Create Stored Procedure that looks at common data errors and warnings

08/12/2024 11:04 am EDT

This stored procedure will find any common data error or warning that can pop up because of a conversion, or even just a simple data update. The script will also give counts for customers to audit.

```
USE [Master_Data_Template]

GO

/****** Object:  StoredProcedure [dbo].[6_VALIDATE_IMPORTED DATA]    Script Date: 8/4/2022 9:04:31 AM ******/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO


/******Instructions**********/

--1. Replace USE [Master_Data_Template] to [YourTargetDatabase]

--2. Update customer, vendor and part Id's

--3.Change "ALTER PROC" TO "CREATE PROC"

--4. Execute query to create procedure in your target database

--5. exec [6_VALIDATE_IMPORTED DATA]

/***************************/


ALTER PROC [dbo].[6_VALIDATE_IMPORTED DATA] AS


BEGIN

    SET NOCOUNT ON;
```

```
/****************2. Update ID's*********************/

declare @customerid int,

@vendorid int,

@partid int


--UPDATE:

set @customerid = 1 --Last CustomerID before import

set @vendorid = 1 --Last VendorID before import

set @partid = 1 --Last PartID before import


--***CUSTOMER & SITE KEY FIELD VALIDATIONS***


--query 1: Check for customers where the terms are set to the null record (N/A)--

Select

AR_Customer.Customer_Number As 'Query 1 - TERM CODE = N/A - "Customer_Number (NO ROWS SHOULD BE RETURNED)"',

CTerm.Term_Code As 'Term'

From

AR_Customer

inner join AR_Term CTerm On AR_Customer.Term_Id = CTerm.Term_id

Where

AR_Customer.Customer_Id > @customerid And

(CTerm.Term_Code = 'N/A')

Order By
```

AR_Customer.Customer_Number

--Query 2: COUNT of customers assigned to each term code --

select

CTerm.Term_Code As 'Query 2 - Term Code', count(CTerm.Term_Code)as "Number of Customers for Term Code - Provide results to the customer to review for accuracy"

From

AR_Customer

inner join AR_Term CTerm On AR_Customer.Term_Id = CTerm.Term_id

Where

AR_Customer.Customer_Id > @customerid

Group By CTerm.Term_Code

--Query 3: Check for customer assigned to a Term Code that does not exist in the Terms Table (No Rows Should be Returned)--

select distinct(a.customer_number) as "Query 3 - AR_Customer - Customer Number - Invalid Term_Id - (NO ROWS SHOULD BE RETURNED)",

a.term_id as "Invalid Term_Id"

from ar_customer a, ar_term b

where a.term_id not in (select term_id from ar_term)

--Query 4: Count of Customers in each Branch where the branch is Active--

select b.branch_code as "Query 4 - AR_Customer - Branch is Active - **Provide results to the customer to review for accuracy**",

```
count(a.branch_id) as "Number of Customers in Active Branch"

from ar_customer a, ar_branch b

where a.branch_id=b.branch_id

and b.inactive='N'

and a.customer_id<>1

group by b.branch_code, a.branch_id
```

```
--Query 5: Count of Customers in each Branch where the branch is Inactive--

select b.branch_code  as "Query 5  - AR_Customer - Branch is Inactive - (NO ROWS SHOULD BE RETURNED)",

count(a.branch_id) as "Number of Customers in Inactive Branch "

from ar_customer a, ar_branch b

where a.branch_id=b.branch_id

and b.inactive='Y'

and a.customer_id<>1

group by b.branch_code, a.branch_id
```

```
--Query 6: Count of Sites in each Branch where the branch is Active--

select b.branch_code  as "Query 6  - AR_Customer_Site - Branch is Active - **Provide results to the customer to review for accuracy**",

count(a.branch_id) as "Number of Sites in Active Branch"

from ar_customer_site a, ar_branch b

where a.branch_id=b.branch_id

and b.inactive='N'
```

and a.customer_id<>1

group by b.branch_code, a.branch_id

--Query 7 Count of Sites in each Branch where the branch is Inactive--

select b.branch_code as "Query 7 - AR_Customer_Site - Branch is Inactive (NO ROWS SHOULD BE RETURNED)",

count(a.branch_id) as "Number of Sites in Inactive Branch"

from ar_customer_site a, ar_branch b

where a.branch_id=b.branch_id

and b.inactive='Y'

and a.customer_id<>1

group by b.branch_code, a.branch_id

--Query 8: Count of Customers by Type of Customer--

select b.type_code as "Query 8 - AR_Customer - Type of Customer - **Provide results to the customer to review for accuracy**",

b.inactive as "Customer Type Inactive?", count(a.customer_type_id) as "Number of Customers"

from ar_customer a, ar_type_of_customer b

where a.customer_type_id=b.type_id

and a.customer_id<>1

group by b.type_code, b.inactive, a.customer_type_id

--Query 9: Listing of customers assigned to Null customer type (Correct the data records)--

select a.customer_number as "Query 9 - AR_Customer - Customer Number - Type of Customer set to Null Node - (NO ROWS SHOULD BE RETURNED)", c.business_name, b.type_code as "Type of Customer"

from ar_customer a, ar_type_of_customer b, ar_customer_bill c

where a.customer_type_id=b.type_id

and a.customer_id=c.customer_id

and b.type_id=1

and a.customer_id<>1

and c.customer_id<>1

order by a.customer_number

--Query 10: Count of Sites in each Tax Group --

select b.taxing_group_code as "Query 10 - AR_Customer_Site - Tax Group - **Provide results to the customer to review for accuracy**",

b.inactive as "Tax Group Inactive Y/N", count(a.tax_group_id) as "Number of Sites in Tax Group"

from ar_customer_site a, ar_taxing_group b

where a.tax_group_id=b.taxing_group_id

and a.customer_id<>1

group by b.taxing_group_code, b.inactive, a.tax_group_id

--Query 11: Count of Sites in Null Tax Group (Correct the data records) --

select b.taxing_group_code   as "Query 11 - AR_Customer_Site - Tax Group - (NO ROWS SHOULD BE RETURNED)",

b.inactive as "Tax Group Inactive Y/N", count(a.tax_group_id) as "Number of Sites in Tax Group"

from ar_customer_site a, ar_taxing_group b

where a.tax_group_id=b.taxing_group_id

and a.customer_id<>1

and b.taxing_group_id=1

group by b.taxing_group_code, b.inactive, a.tax_group_id

--Query 12: Customers/Sites in Null Tax Group (Correct the data records) --

select c.customer_number   as "Query 12 - AR_Customer_Site - Customer Number - Tax Group set to Null Node - (NO ROWS SHOULD BE RETURNED)",

c.customer_name,d.business_name as "Bill To Name",

a.business_name as "Site Name",

b.taxing_group_code, b.inactive as "Tax Group Inactive Y/N"

from ar_customer_site a, ar_taxing_group b, ar_customer c, ar_customer_bill d

where a.tax_group_id=b.taxing_group_id

and a.customer_id<>1

and b.taxing_group_id=1

and a.customer_id=c.customer_id

and c.customer_id=d.customer_id

order by c.customer_number

--Query 13: Count of Sites in each Cycle Tax Group --

select b.taxing_group_code   as "Query 13 - AR_Customer_Site - Cycle Tax Group - **Provide results to the customer to review for accuracy** ",

b.inactive as "Tax Group Inactive Y/N", count(a.cycle_tax_group_id) as "Number of Sites in Cycle Tax Group"

from ar_customer_site a, ar_taxing_group b

where a.cycle_tax_group_id=b.taxing_group_id

and a.customer_id<>1

group by b.taxing_group_code, b.inactive, a.cycle_tax_group_id

```
--Query 14: Count of Sites in Null Cycle Tax Group --

select b.taxing_group_code as "Query 14  - AR_Customer_Site - Cycle Tax Group set to Null Node - (NO ROWS SHOULD BE RETURNED)",

b.inactive as "Tax Group Inactive Y/N", count(a.cycle_tax_group_id) as "Number of Sites in Cycle Tax Group"

from ar_customer_site a, ar_taxing_group b

where a.cycle_tax_group_id=b.taxing_group_id

and a.customer_id<>1

and b.taxing_group_id=1

group by b.taxing_group_code, b.inactive, a.cycle_tax_group_id
```

```
--Query 15: Customers/Sites in Null Cycle Tax Group (Correct the Data Records) --

select c.Customer_number as "Query 15 - AR_Customer_Site - Customer Number - Cycle Tax Group set to Null Node - (NO ROWS SHOULD BE RETURNED) ",

c.customer_name, d.business_name as"Bill To Name", a.business_name as "Site Name",

a.Site_Number, b.taxing_group_code as "Cycle Tax Group Code", b.inactive as "Tax Group:Inactive Y/N"

from ar_customer_site a, ar_taxing_group b, ar_customer c, ar_customer_bill d

where a.cycle_tax_group_id=b.taxing_group_id

and a.customer_id<>1

and b.taxing_group_id=1

and a.customer_id=c.customer_id

and c.customer_id=d.customer_id

order by c.customer_number
```

--Query 16: Customers/Sites with a blank Site Number (Correct the Data Records) --

select c.Customer_number as "Query 16 - AR_Customer_Site - Customer Number - Site Number is Blank (NO ROWS SHOULD BE RETURNED)",

c.customer_name,

d.business_name as"Bill To Name", a.business_name as "Site Name", a.Site_Number

from ar_customer_site a, ar_customer c, ar_customer_bill d

where a.Site_Number = ''

and a.customer_id<>1

and a.customer_id=c.customer_id

and c.customer_id=d.customer_id

order by c.customer_number

--***SYSTEM RECORDS - KEY FIELD VALIDATIONS***

--*** checking warranty, service level, system type, service company, monitoring company, contract form***

--Query 17: Count of Systems by System Type ---

select b.system_code as "Query 17 - AR_Customer_System - System Type **Provide results to the customer to review for accuracy** ",

b.inactive as "System Code Inactive Y/N",

count(a.system_id) as "Number of Systems"

from ar_customer_system a, sy_system b

where a.system_id=b.system_id and a.Customer_Id > @customerid

group by b.system_code, b.inactive, a.system_id

order by b.system_code

--Query 18: Customers/Sites/Systems with Null System Type (Correct the Data Records) --

select c.customer_number as "Query 18 - AR_Customer_System - Customer Number - System Type set to Null Node (NO ROWS SHOULD BE RETURNED)",

d.business_name as"Customer Name", a.business_name as "Site Name",

e.Alarm_account as "System Number", b.system_code as "System Code", b.inactive as "System Type:Inactive Y/N"

from ar_customer_site a, sy_system b, ar_customer c, ar_customer_bill d, ar_customer_system e

where b.system_id=e.system_id

and a.customer_id<>1

and b.system_id=1

and e.customer_id=a.customer_id

and e.customer_id=c.customer_id

and e.customer_id=d.customer_id

and a.customer_id=c.customer_id

and c.customer_id=d.customer_id

and e.customer_site_id=a.customer_site_id

order by c.customer_number

--Query 19: Count of Systems with Panel Type ---

select b.panel_type_code as "Query 19 - AR_Customer_System - Panel Type - **Provide results to the customer to review for accurac",

b.inactive as "Panel Type Code Inactive Y/N",

count(a.panel_type_id) as "Number of Systems with Panel Types"

from ar_customer_system a, sy_panel_type b

where a.panel_type_id=b.panel_type_id and a.Customer_Id > @customerid group by b.panel_type_code, b.inactive, a.panel_type_id

order by b.panel_type_code

--Query 20: Customers/Sites/Systems with Null Panel Type (Fix the Data if any records are returned by the query, if the customer requires a panel type) --

select c.customer_number as "Query 20 - AR_Customer_System - Customer # - Panel Type =Null Node - **Provide results to the customer to review for accuracy**",

c.customer_name as"Customer Name", a.business_name as "Site Name",

e.Alarm_account as "System Number", b.panel_type_code as "Panel Type Code", b.inactive as "Panel Type:Inactive Y/N"

from ar_customer_site a, sy_panel_type b, ar_customer c, ar_customer_bill d, ar_customer_system e

where b.panel_type_id=e.panel_type_id

and a.customer_id<>1

and b.panel_type_id=1

and e.customer_id=a.customer_id

and e.customer_id=c.customer_id

and e.customer_id=d.customer_id

and a.customer_id=c.customer_id

and c.customer_id=d.customer_id

and e.customer_site_id=a.customer_site_id

order by c.customer_number

--Query 21: Count of Systems Contract Form Type ---

select b.contract_form_code as "Query 21 - AR_Customer_System - Contract Type - **Provide results to the customer to review for accuracy**", b.inactive as "Contract Form Code Inactive Y/N",

count(a.contract_form_id) as "Number of Systems with Contract Form"

from ar_customer_system a, ar_contract_form b

where a.contract_form_id=b.contract_form_id and a.Customer_Id > @customerid

group by b.contract_form_code, b.inactive, a.contract_form_id

--Query 22: Customers/Sites/Systems with Null Contract Form Type (Fix the Data if any records are returned by the query, if the customer requires a contract type) --

Select c.customer_number as "Query 22 - AR_Customer_System - Customer#-Contract Type = Null Node - **Provide results to the customer to review for accuracy**",

d.business_name as"Customer Name", a.business_name as "Site Name",

e.Alarm_account as "System Number", b.contract_form_code as "Contract Form Code", b.inactive as "Contract Form:Inactive Y/N"

from ar_customer_site a, ar_contract_form b, ar_customer c, ar_customer_bill d, ar_customer_system e

where b.contract_form_id=e.contract_form_id

and a.customer_id<>1

and b.contract_form_id=1

and e.customer_id=a.customer_id

and e.customer_id=c.customer_id

and e.customer_id=d.customer_id

and a.customer_id=c.customer_id

and c.customer_id=d.customer_id

and e.customer_site_id=a.customer_site_id

order by c.customer_number

--Query 23: Count of Systems Warranty Type ---

select b.warranty_code as "Query 23 - AR_Customer_System - Warranty Type - **Provide results to the customer to review for accuracy**",

b.inactive as "Warranty Code Inactive Y/N", --b.warranty_id,

count(a.warranty_id) as "Number of Systems with Warranty Code", b.description

from ar_customer_system a, sv_warranty b

where a.warranty_id=b.warranty_id and a.Customer_Id > @customerid

group by b.warranty_code, b.inactive, a.warranty_id, b.warranty_id, b.description

--Query 24: Customers/Sites/Systems with Null Warranty Type (No Rows Should be Returned) --

select c.customer_number as "Query 24 - AR_Customer_System - Customer Number - Warranty Type set to Null Node (NO ROWS SHOULD BE RETURNED) ",

d.business_name as"Customer Name", a.business_name as "Site Name",

e.Alarm_account as "System Number", b.warranty_code as "Warranty Code", b.inactive as "Warranty Code:Inactive Y/N",b.warranty_id

from ar_customer_site a, sv_warranty b, ar_customer c, ar_customer_bill d, ar_customer_system e

where b.warranty_id=e.warranty_id

and a.customer_id<>1

and b.warranty_id=1

and e.customer_id=a.customer_id

and e.customer_id=c.customer_id

and e.customer_id=d.customer_id

and a.customer_id=c.customer_id

and c.customer_id=d.customer_id

and e.customer_site_id=a.customer_site_id

order by c.customer_number

--Query 25: Count of Systems by Service Level---

select b.service_level_code as "Query 25 - AR_Customer_System - Service Level - **Provide results to the customer to review for accuracy**",

b.inactive as "Service Level Code Inactive Y/N", --b.service_level_id,

count(a.service_level_id) as "Number of Systems with Service Level Code", b.description

from ar_customer_system a, sv_service_level b

where a.service_level_id=b.service_level_id and a.Customer_Id > @customerid

group by b.service_level_code, b.inactive, a.service_level_id, b.service_level_id, b.description

order by b.service_level_code

--Query 26: Customers/Sites/Systems with Null Service Level (No Rows Should be Returned) --

select c.customer_number as "Query 26 - AR_Customer_System - Customer Number - Service Level set to Null Node - (NO ROWS SHOULD BE RETURNED)",

d.business_name as"Customer Name", a.business_name as "Site Name",

e.Alarm_account as "System Number", b.service_level_code as "Service Level Code",

b.inactive as "Service Level Code:Inactive Y/N", b.service_level_id

from ar_customer_site a, sv_service_level b, ar_customer c, ar_customer_bill d, ar_customer_system e

where b.service_level_id=e.service_level_id

and a.customer_id<>1

and b.service_level_id=1

and e.customer_id=a.customer_id

and e.customer_id=c.customer_id

and e.customer_id=d.customer_id

and a.customer_id=c.customer_id

and c.customer_id=d.customer_id

and e.customer_site_id=a.customer_site_id

order by c.customer_number

--Query 27: Count of Systems by Service Company ---

select b.service_company_code as "Query 27 - AR_Customer_System - Service Company - **Provide results to the customer to review for accuracy**",

b.inactive as "Service Company Code Inactive Y/N", --b.service_company_id,

count(a.service_company_id) as "Number of Systems with Service Company Code", b.description

from ar_customer_system a, sv_service_company b

where a.service_company_id=b.service_company_id and a.Customer_Id > @customerid

group by b.service_company_code, b.inactive, a.service_company_id, b.service_company_id, b.description

order by b.service_company_code

--Query 28: Customers/Sites/Systems with Null Service Company (Fix the Data if any records are returned by the query) --

select c.customer_number as "Query 28 - AR_Customer_System - Service Company set to Null Node - **Provide results to the customer to review for accuracy**",

d.business_name as"Customer Name", a.business_name as "Site Name",

e.Alarm_account as "System Number", b.service_Company_code as "Service Company Code",

b.inactive as "Service Company Code:Inactive Y/N", b.service_Company_id

```
from ar_customer_site a, sv_service_Company b, ar_customer c, ar_customer_bill d, ar_customer_system e

where b.service_Company_id=e.service_Company_id

and a.customer_id<>1

and b.service_Company_id=1

and e.customer_id=a.customer_id

and e.customer_id=c.customer_id

and e.customer_id=d.customer_id

and a.customer_id=c.customer_id

and c.customer_id=d.customer_id

and e.customer_site_id=a.customer_site_id

order by c.customer_number
```

```
--Query 29:  Count of Systems by Alarm Company ---

select  b.alarm_company_code as "Query 29  - AR_Customer_System - Alarm Company - **Provide results to the customer to review for accuracy**",

b.inactive as "Alarm Company Code Inactive Y/N", --b.alarm_company_id,

count(a.alarm_company_id) as "Number of Systems with Alarm Company Code", b.description

from ar_customer_system a, cs_alarm_company b

where a.alarm_company_id=b.alarm_company_id and a.Customer_Id > @customerid

group by b.alarm_company_code, b.inactive, a.alarm_company_id, b.alarm_company_id, b.description

order by b.alarm_company_code
```

--Query 30: Customers/Sites/Systems with Null Alarm Company (Monitored By) (Fix the Data if any records are returned by the query, if the customer requires the Alarm Company) --

select c.customer_number as "Query 30 - AR_Customer_System - Alarm Company set to Null Node -**Provide results to the customer to review for accuracy** ",

d.business_name as"Customer Name", a.business_name as "Site Name",

e.Alarm_account as "System Number", b.alarm_Company_code as "Alarm Company Code",

b.inactive as "Alarm Company Code:Inactive Y/N", b.Alarm_Company_id

from ar_customer_site a, cs_alarm_Company b, ar_customer c, ar_customer_bill d, ar_customer_system e

where b.Alarm_Company_id=e.Alarm_Company_id

and a.customer_id<>1

and e.Alarm_Company_id=1

and e.customer_id=a.customer_id

and e.customer_id=c.customer_id

and e.customer_id=d.customer_id

and a.customer_id=c.customer_id

and c.customer_id=d.customer_id

and e.customer_site_id=a.customer_site_id

order by c.customer_number

--***RECURRING LINE RECORDS - KEY FIELD VALIDATIONS***

--*** checking invoice item codes, billing cycle***

--Query 31: Count of Recurring Lines by Invoice Item Code ---

```
select  b.item_code as "Query 31  - AR_Customer_Recurring - Invoice Item - **Provide results to the customer to review for accuracy**",

b.inactive as "Invoice Item Code Inactive Y/N",

--b.item_id,

count(c.item_id) as "Number of Recurring Lines with Invoice Item Code",

b.description as "Invoice Item Description",

c.user_description as "Recurring Line Description"

from  ar_item b, ar_customer_recurring c

where c.item_id=b.item_id and c.Customer_Id > @customerid

group by b.item_code, b.inactive, b.item_id, b.description, c.User_Description

order by b.item_code
```

```
--Query 32:  Count of Recurring Lines by Invoice Item Code  Where the Recurring Line Description is Blank---

select  b.item_code as "Query 32  - AR_Customer_Recurring - Item Code - User Description is Blank - (NO ROWS SHOULD BE RETURNED)",

b.inactive as "Invoice Item Code Inactive Y/N",

--b.item_id,

count(c.item_id) as "Number of Recurring Lines with Invoice Item Code",

b.description as "Invoice Item Description",

c.user_description as "Recurring Line Description"

from  ar_item b, ar_customer_recurring c

where c.item_id=b.item_id

and c.user_description = ''

group by b.item_code, b.inactive, b.item_id, b.description, c.User_Description

order by b.item_code
```

```
/*QUERY 33 AR_INVOICE_GROUP IS NO LONGER A VALID TABLE */

--Query 33:  Count of Recurring Lines by Invoice Group Number---

--select  a.invoice_Group_number as "Query 33  - AR_Customer_Recurring - Invoice Group - **Provide results to the customer to review for accuracy**",

--b.inactive as "Invoice Group Inactive Y/N", b.group_number, b.group_name,

--count(a.invoice_group_number) as "Number of Recurring Lines with Invoice Group",

--b.group_name as "Invoice Group Name"

--from  AR_Invoice_Group b, ar_customer_recurring a

--where a.invoice_group_number = b.invoice_group_id

--group by a.invoice_group_number, b.inactive, b.group_number, b.group_name

--order by b.Group_Number


--Query 33:  Count of Recurring Lines by Invoice Group Number---

select  a.invoice_Group_number as "Query 30B  - Invoice Group",

count(a.invoice_group_number) as "Number of Recurring Lines with Invoice Group"

--select *

from  ar_customer_recurring a

where a.Customer_Id > @customerid

group by a.invoice_group_number

order by a.invoice_group_number
```

--Query 34: Customers/Sites/Systems with Null Invoice Item Code (Fix the Data if any records are returned by the query) --

select c.customer_number as "Query 34 - AR_Customer_Recurring - Customer Number - Item Code set to Null Node - (NO ROWS SHOULD BE RETURNED)",

d.business_name as"Customer Name", a.business_name as "Site Name",

e.Alarm_account as "System Number", b.item_code as "Invoice Item Code",

b.inactive as "Invoice Item Code:Inactive Y/N", b.item_id

from ar_customer_site a, ar_item b, ar_customer c, ar_customer_bill d, ar_customer_system e,

ar_customer_recurring f

where b.item_id=f.item_id

and a.customer_id<>1

and f.item_id=1

and e.customer_id=a.customer_id

and e.customer_id=c.customer_id

and e.customer_id=d.customer_id

and a.customer_id=c.customer_id

and c.customer_id=d.customer_id

and e.customer_site_id=a.customer_site_id

and f.customer_id=c.customer_id

and f.customer_id=a.customer_id

and f.customer_id=d.customer_id

and f.customer_id=e.customer_id

and f.customer_site_id=a.customer_site_id

and f.customer_system_id=e.customer_system_id

order by c.customer_number

--Query 35: Count of Recurring Lines by Billing Cycle ---

select bill_cycle as "Query 35 - AR_Customer_Recurring - Billing Cycle - **Provide results to the customer to review for accuracy**",

count(bill_cycle) as "Number of Recurring Lines in Billing Cycle"

from ar_customer_recurring

where Customer_Id > @customerid

group by bill_cycle

--Query 36: Recurring Lines Where Cycle Amount =0 (Zero is not a valid value) ---

select cycle_amount as "Query 36 - AR_Customer_Recurring - Cycle Amount = 0 - **Provide results to the customer to review for accuracy**",

count(cycle_amount) as "Number of Recurring Lines", cancelled

from ar_customer_recurring

where cycle_amount =0

and cancelled='N'

and Customer_Id > @customerid

group by cycle_amount, cancelled

--Query 37: Recurring Lines Where Cycle Amount =0 Fix the Data - Should not be any RMR Lines with an amount of Zero---

```
select c.customer_number as "Query 37  - AR_Customer_Recurring - Cust# - Cycle Amount = 0 =invalid -*Provide results to the customer to review for accuracy**",

d.business_name as"Customer Name", a.business_name as "Site Name",

e.Alarm_account as "System Number", b.item_code as "Invoice Item Code", f.cancelled,

f.cycle_amount, f.cycle_start_date, f.next_cycle_date

from ar_customer_site a, ar_item b, ar_customer c, ar_customer_bill d, ar_customer_system e,

ar_customer_recurring f

where b.item_id=f.item_id

and a.customer_id<>1

and f.cycle_amount <=0

and e.customer_id=a.customer_id

and e.customer_id=c.customer_id

and e.customer_id=d.customer_id

and a.customer_id=c.customer_id

and c.customer_id=d.customer_id

and e.customer_site_id=a.customer_site_id

and f.customer_id=c.customer_id

and f.customer_id=a.customer_id

and f.customer_id=d.customer_id

and f.customer_id=e.customer_id

and f.customer_site_id=a.customer_site_id

and f.customer_system_id=e.customer_system_id

order by c.customer_number
```

```sql
--Query 38:  Recurring Lines Where monthly Amount <0  (Not Cancelled) For Review of Valid Free Services ---

select c.customer_number as "Query 38 -   AR_Customer_Recurrin - Customer Number - Monthly Amount <0 - (NO ROWS SHOULD BE RETURNED)",

d.business_name as"Customer Name", a.business_name as "Site Name",

e.Alarm_account as "System Number", b.item_code as "Invoice Item Code", f.cancelled,

f.monthly_amount, f.cycle_start_date, f.next_cycle_date

from ar_customer_site a, ar_item b, ar_customer c, ar_customer_bill d, ar_customer_system e,

ar_customer_recurring f

where b.item_id=f.item_id

and a.customer_id<>1

and f.monthly_amount <0

and f.cancelled = 'N'

and e.customer_id=a.customer_id

and e.customer_id=c.customer_id

and e.customer_id=d.customer_id

and a.customer_id=c.customer_id

and c.customer_id=d.customer_id

and e.customer_site_id=a.customer_site_id

and f.customer_id=c.customer_id

and f.customer_id=a.customer_id

and f.customer_id=d.customer_id

and f.customer_id=e.customer_id

and f.customer_site_id=a.customer_site_id

and f.customer_system_id=e.customer_system_id

order by c.customer_number
```

```
--Query 39:  Recurring Lines Where Next Cycle Date is < Cycle Start Date (Not Cancelled)---

select c.customer_number as "Query 39 -   AR_Customer_Recurring - Cust# - Next Cycle Date < than the Cycle Start Date - (NO ROWS SHOULD BE RETURNED)",

d.business_name as"Customer Name", a.business_name as "Site Name",

e.Alarm_account as "System Number", b.item_code as "Invoice Item Code", f.cancelled,

f.monthly_amount, f.cycle_start_date, f.next_cycle_date

from ar_customer_site a, ar_item b, ar_customer c, ar_customer_bill d, ar_customer_system e, ar_customer_recurring f

where b.item_id=f.item_id

and a.customer_id<>1

and f.next_cycle_date < f.cycle_start_date

and f.cancelled = 'N'

and e.customer_id=a.customer_id

and e.customer_id=c.customer_id

and e.customer_id=d.customer_id

and a.customer_id=c.customer_id

and c.customer_id=d.customer_id

and e.customer_site_id=a.customer_site_id

and f.customer_id=c.customer_id

and f.customer_id=a.customer_id

and f.customer_id=d.customer_id

and f.customer_id=e.customer_id

and f.customer_site_id=a.customer_site_id
```

and f.customer_system_id=e.customer_system_id

order by c.customer_number

--*** CUSTOMER GEOGRAPHICAL TABLES (CITY, STATE, ZIP) - KEY FIELD VALIDATIONS***

--Query 40: GE table id 1 validating (GE_Table1 - City) in AR_CUstomer_Bill - Should be zero rows returned ----

select b.customer_number as "Query 40 - AR_Customer_Bill - Customer Number - City is Invalid - (NO ROWS SHOULD BE RETURNED)",

a.ge_table1_id as "GE Table 1 - City", a.ge_table1_id as "Bill To Table 1 ID"

from ar_customer_bill a, ar_customer b

where a.ge_table1_id not in (select table1_id from ge_table1)

and a.customer_id<>1

and a.customer_id=b.customer_id

order by b.customer_number

--Query 41: GE table id 2 validatiing in AR_Customer_Bill -   (GE_Table2 - State/Province Table) -   Should be zero rows returned ----

select b.customer_number as "Query 41- AR_Customer_Bill - Customer Number - State/Province is Invalid - (NO ROWS SHOULD BE RETURNED)",

a.ge_table2_id as "GE Table 2 - State/Province", a.ge_table2_id as "Bill To Table 2 ID"

from ar_customer_bill a, ar_customer b

where a.ge_table2_id not in (select table2_id from ge_table2)

and a.customer_id<>1

and a.customer_id=b.customer_id

order by b.customer_number

--Query 42: GE table id 3 validating in AR_Customer_Bill - (City/State/Zip table id) Should be zero rows returned ----

select b.customer_number as "Query 42 - AR_Customer_Bill - Customer Number - Zip/Postal Code is invalid - (NO ROWS SHOULD BE RETURNED)",

a.ge_table3_id as "GE Table 3 - ZIP/Postal Code", a.ge_table3_id as "Bill To Table 3 ID"

from ar_customer_bill a, ar_customer b

where a.ge_table3_id not in (select table3_id from ge_table3)

and a.customer_id<>1

and a.customer_id=b.customer_id

order by b.customer_number

--Query 43: GE table id 1 Description validations (City Description) Should be zero rows returned ----

select distinct(b.customer_number) as "Query 43   - AR_Customer_Bill - Customer Number - City - **Provide results to the customer to review for accuracy**",

a.ge1_description as "AR_Customer_Bill - GE Table 1 - (City Description) City Name is invalid",

a.ge_table1_id as "Bill To Table 1 ID"

from ar_customer_bill a, ar_customer b, ge_table1 c where a.ge1_description not in (select description from ge_table1)

and a.customer_id<>1

and a.customer_id=b.customer_id

order by b.customer_number

--Query 44: GE table id 2 Description validations (State Long Description) Should be zero rows returned ----

select b.customer_number as "Query 44 - AR_Customer_BIll - Customer Number - State/Province Long is invalid - (NO ROWS SHOULD BE RETURNED)",

a.ge2_description as "Bill To - GE Table 2 - State Long Description", a.ge_table2_id as "Bill To Table 2 ID"

from ar_customer_bill a, ar_customer b

where a.ge2_description not in (select description from ge_table2)

and a.customer_id<>1

and a.customer_id=b.customer_id

order by b.customer_number

--Query 45: GE table id 2 Description validations (State Short Description) Should be zero rows returned ----

select b.customer_number as "Query 45 - AR_Customer_BIll - Customer Number - State/Province Short is invalid - (NO ROWS SHOULD BE RETURNED)",

a.ge2_short as "GE Table 2 - State Short Description", a.ge_table2_id as "Bill To Table 2 ID"

from ar_customer_bill a, ar_customer b

where a.ge2_short not in (select short from ge_table2)

and a.customer_id<>1

and a.customer_id=b.customer_id

order by b.customer_number

--Query 46: GE table id 3 Description validations (Zipcode Description) Should be zero rows returned ----

select b.customer_number as "Query 46 - AR_Customer_BIll - Customer Number - Zip/Postal Code is invalid (NO ROWS SHOULD BE RETURNED)",

a.ge3_description as "GE Table 3 - Zip/Postal Code Description", a.ge_table3_id as "Bill To Table 3 ID"

from ar_customer_bill a, ar_customer b

where a.ge3_description not in (select description from ge_table3)

and a.customer_id<>1

and a.customer_id=b.customer_id

order by b.customer_number

--*** SITE GEOGRAPHICAL TABLES (CITY, STATE, ZIP) - KEY FIELD VALIDATIONS***

--Query 47: GE table id 1 validations (city table id) ----

select a.customer_number as "Query 47 - AR_Customer_Site - Customer Number - City - **Provide results to the customer to review for accuracy**",

b.business_name as "Site Name", b.ge_table1_id as "Site Table 1 ID"

from ar_customer a, ar_customer_site b

where b.ge_table1_id not in (select table1_id from ge_table1)

and a.customer_id<>1

and a.customer_id=b.customer_id

order by a.customer_number, b.business_name

--Query 48: GE table id 2 validations (state table id) ----

select a.customer_number as "Query 48 - AR_Customer_Site - Customer Number - State/Province is invalid (NO ROWS SHOULD BE RETURNED)",

b.business_name as "Site Name", b.ge_table2_id as "Site Table 2 ID"

from ar_customer a, ar_customer_site b

where b.ge_table2_id not in (select table2_id from ge_table2)

and a.customer_id<>1

and a.customer_id=b.customer_id

order by a.customer_number, b.business_name

--Query 49: GE table id 3 validations (City/State/Zip table id) ----

select a.customer_number as "Query 49 - AR_Customer_Site - Customer Number - Zip/Postal Code is invalid (NO ROWS SHOULD BE RETURNED)",

b.ge_table3_id as "Site Table 3 ID", b.business_name   as "Site Name"

from ar_customer a, ar_customer_site b

where b.ge_table3_id not in (select table3_id from ge_table3)

and a.customer_id<>1

and a.customer_id=b.customer_id

order by a.customer_number, b.business_name

--Query 50: GE table id 1 Description validations (City Description) ----

select a.customer_number as "Query 50 - AR_Customer_Site - Customer Number - City Desc invalid - **Provide results to the customer to review for accuracy**",

b.ge1_description as "Site GE 1 Description", b.business_name as "Site Name",

b.ge_table1_id as "Site Table 1 ID"

from ar_customer a, ar_customer_site b

where b.ge1_description not in (select description from ge_table1)

and a.customer_id<>1

and a.customer_id=b.customer_id

order by a.customer_number, b.business_name

--Query 51: GE table id 2 Description validations (State Short Description) ----

select a.customer_number as "Query 51 - AR_Customer_Site - Customer Number - State/Province Short Description is invalid - (NO ROWS SHOULD BE RETURNED)",

b.ge2_short as "Site GE 2 Short", b.business_name as "Site Name",

b.ge_table2_id as "Site Table 2 ID"

from ar_customer a, ar_customer_site b

where b.ge2_short not in (select short from ge_table2)

and a.customer_id<>1

and a.customer_id=b.customer_id

order by a.customer_number, b.business_name

--Query 52: GE table id 2 Description validations (State Long Description) ----

select a.customer_number as "Query 52 - AR_Customer_Site - Customer Number - State/Province Long Description is invalid - (NO ROWS SHOULD BE RETURNED)",

b.ge2_description as "Site GE 2 Long", b.business_name as "Site Name",

b.ge_table2_id as "Site Table 2 ID"

from ar_customer a, ar_customer_site b

where b.ge2_description not in (select description from ge_table2)

and a.customer_id<>1

and a.customer_id=b.customer_id

order by a.customer_number, b.business_name

--Query 53: GE table id 3 Description validations (Zipcode) ----

select a.customer_number as "Query 53 - AR_Customer_Site - Zip/Postal Code is invalid - (NO ROWS SHOULD BE RETURNED)", b.ge3_description as "Site GE 3 Description", b.business_name as "Site Name",

b.ge_table3_id as "Site Table 3 ID"

from ar_customer a, ar_customer_site b

where b.ge3_description not in (select description from ge_table3)

and a.customer_id<>1

and a.customer_id=b.customer_id

order by a.customer_number, b.business_name

--Query 54: Validate Site City Name (id matches GE3 Table but City Name does not match) ----

--- Checking for possible misspelled Cities --

select a.customer_number as "Query 54 - AR_Customer_Site - City Name - City Name does not match ID in Zip/Postal Code table - (NO ROWS SHOULD BE RETURNED)",

b.ge1_description as "Site GE 1 Description", b.business_name as "Site Name",

b.ge_table1_id as "Site Table 1 ID"

from ar_customer a, ar_customer_site b, ge_table3 c

where b.ge_table3_id=c.table3_id

and c.ge_table2_id in(select table2_id from ge_table2)

and c.ge_table1_id in(select table1_id from ge_table1)

and b.ge1_description not in (select description from ge_table1)

and a.customer_id<>1

and a.customer_id=b.customer_id

order by a.customer_number, b.business_name

--Query 55: Validate Site State (Long Name) (valid ID in GE2 Table but State Long Name does not match) ----

--- Checking for possible misspelled States --

select a.customer_number as "Query 55 - AR_Customer_Site - State/Province - Long Name does not match in State/Province table - (NO ROWS SHOULD BE RETURNED)",

b.ge2_description as "Site GE 2 Long Description", b.business_name as "Site Name",

b.ge_table2_id as "Site Table 2 ID", c.Description as "GE Table 2 Description"

from ar_customer a, ar_customer_site b, ge_table2 c

where b.ge_table2_id=c.table2_id

and b.ge2_description < > c.description

and a.customer_id<>1

and a.customer_id=b.customer_id

order by a.customer_number, b.business_name

--Query 56: Validate Site State (Short Name) (Valid ID in GE2 Table but State Short Name does not match) ----

--- Checking for possible misspelled States --

select a.customer_number as "Query 56 - AR_Customer_Site - State/Province - Short Name does not match in State/Province table - (NO ROWS SHOULD BE RETURNED)",

```
b.ge2_short as "Site GE2 Short", b.business_name as "Site Name",

c.Short as "GE Table 2 Short", c.Table2_Id as "GE Table2 ID"

from ar_customer a, ar_customer_site b, ge_table2 c

where b.ge_table2_id=c.table2_id

and b.ge2_short <> c.short

and a.customer_id<>1

and a.customer_id=b.customer_id

order by a.customer_number, b.business_name
```

```
--Query 57: Validate Site Zipcode (Valid ID in GE3 Table 3 but Zipcode description does not match) ----

---  Checking for possible incorrect zip/postal codes  --

select a.customer_number as "Query 57  - AR_Customer_Site - Zip/Postal Code - Description does not match in Zip/Postal table - (NO ROWS SHOULD BE RETURNED)",

b.ge3_description as "Site GE 3 Description",

 b.business_name as "Site Name", b.GE_Table3_Id as "Site Table 3 ID",

c.description as "GE Table 3 Description",    c.table3_id as "GE Table3 ID"

from ar_customer a, ar_customer_site b, ge_table3 c

where b.ge_table3_id=c.table3_id

and b.ge3_description <> c.description

and a.customer_id<>1

and a.customer_id=b.customer_id

order by a.customer_number, b.business_name
```

/*QUERY 58 & 59 APPEAR TO BE DOING THE SAME THING */

--Query 58: Validate Site Zipcode (Site GE3 id matches GE3 Table but State/Province Long description does not match) ----

--- Checking for possible misspelled Site States (Long State Name) --

select a.customer_number as "QUERY 58-AR_Customer_Site-State/Province-Long Description does not match in State/Province table - (NO ROWS SHOULD BE RETURNED)",

b.business_name as "Site Name",

b.ge_table3_id as "Site Table 3 ID", b.ge_table2_id as "Site Table 2 ID",

b.ge3_description as "Site GE 3 Description (Zip/Postal Code",b.ge2_description as "Site Table 2 Description", b.ge2_short as "Site GE 2 Short",

d.Table2_Id as "GE Table 2 ID", d.Description as "GE Table 2 Description", d.Short as "GE Table 2 Short"

from ar_customer a, ar_customer_site b, ge_table3 c, ge_table2 d, ge_table1 e

where b.ge_table3_id=c.table3_id

and c.GE_Table2_ID = d.Table2_Id

and c.GE_Table1_ID = e.Table1_ID

and b.ge2_description <> d.description

and a.customer_id<>1

and a.customer_id=b.customer_id

order by b.GE2_Description, a.customer_number, b.business_name

--Query 59: Validate Site Zipcode (Site GE3 id matches GE3 Table but State/Province Short description does not match) ----

```
---  Checking for possible misspelled Site States (Short State Name) --

select a.customer_number as "QUERY 59-AR_Customer_Site-State/Province-Short Description does not match in State/Province table - (NO ROWS SHOULD BE RETURNED)",

b.business_name as "Site Name",

b.ge_table3_id as "Site Table 3 ID", b.ge_table2_id as "Site Table 2 ID",

b.ge3_description as "Site GE 3 Description (Zip/Postal Code", b.ge2_description as "Site Table 2 Description", b.ge2_short as "Site GE 2 Short",

d.Table2_Id as "GE Table 2 ID", d.Description as "GE Table 2 Description", d.Short as "GE Table 2 Short"

from ar_customer a, ar_customer_site b, ge_table3 c, ge_table2 d, ge_table1 e

where b.ge_table3_id=c.table3_id

and c.GE_Table2_ID = d.Table2_Id

and c.GE_Table1_ID = e.Table1_ID

and b.ge2_short <> d.short

and a.customer_id<>1

and a.customer_id=b.customer_id

order by b.GE2_short,  a.customer_number, b.business_name
```

```
--Query 60: Zipcodes on Customer Bill To with length less than 5 characters ----

select c.customer_number as "QUERY 60 - AR_Customer_Bill - Customer Number - Zip/Postal Code not 5 Characters - (NO ROWS SHOULD BE RETURNED)",

b.business_name as "Bill To Name", c.Customer_Name as "Customer Name",

g.Description as "ge3_description(zip/postal code)"

from AR_Customer_Bill b

inner join AR_Customer c on b.Customer_Id = c.Customer_Id

left join GE_Table3 g on b.GE_Table3_Id = g.Table3_ID
```

where (len(g.Description) <> 5 and c.Customer_Id > @customerid and g.Country_ID in (1,2)) or (g.[description] not like '%[1-9]%' and c.Customer_Id > @customerid)

order by Customer_Number

--Query 61: Zipcodes on Site with length less than 5 characters ----

select c.customer_number as "QUERY 61 - AR_Customer_Site - Customer Number - Zip/Postal Code not 5 Characters - (NO ROWS SHOULD BE RETURNED)",

b.business_name as "Site Name", g.Description as "ge3_description(Site zip/postal code)"

from AR_Customer_Site b

inner join AR_Customer c on b.Customer_Id = c.Customer_Id

left join GE_Table3 g on b.GE_Table3_Id = g.Table3_ID

where len(g.Description) <> 5 and c.Customer_Id > @customerid and g.Country_ID in (1,2) or (g.[description] not like '%[1-9]%' and c.Customer_Id > @customerid)

order by Customer_Number

--Query 62: Customers without a Bill To record --

select Distinct Customer_Number as "Query 62 - Customer Number - Customers Without a Bill To Record - (NO ROWS SHOULD BE RETURNED) "

from AR_Customer

where Customer_Id not in (select Customer_Id from AR_Customer_Bill)

--Query 62.1: Customers without a Bill To record --

select b.Customer_Number "Query 62.1 - Customer Number - set to receive Email Invoices w/out a valid Email - (NO ROWS SHOULD BE RETURNED) ", a.E_Mail, a.Email_Invoice, a.email_cycle_invoice, a.email_job_invoice, a.email_service_invoice, a.email_other_invoice, a.print_job_invoice, a.print_job_invoice, a.print_job_invoice, a.print_job_invoice

from AR_Customer_Bill a, ar_customer b

where a.email_Cycle_Invoice = 'Y'

and a.Customer_Id = b.Customer_Id

--and E_Mail like '%@%.%'

and Customer_Bill_Id not in (select Customer_Bill_Id from AR_Customer_Bill_Email)

order by a.E_Mail

--Query 63: Customers without a Customer Aging record --

select Distinct Customer_Number as "Query 63 - Customer Number - Customers Without an AR Customer Aging Record - (NO ROWS SHOULD BE RETURNED)"

from AR_Customer

where Customer_Id not in (select Customer_Id from AR_Customer_Aging)

--Query 64: Customers without a site record --

select Distinct Customer_Number as "Query 64 - Customer Number - Customers Without a Site Record - (NO ROWS SHOULD BE RETURNED)"

from AR_Customer

where Customer_Id not in (select Customer_Id from AR_Customer_site)

--Query 65: Count of Subaccounts for each Master Account to verify the correct number of Subaccounts and the count of Master Accounts expected --

select b.Master_Account_Code as "Query 65 - Master Account# - # of Subaccounts/ per MasterAcct- **Provide results to the customer to review for accuracy**",

COUNT(b.Master_Account_Code) as "COUNT of subaccounts",

b.description as "Master Account Description"

from AR_Customer a, AR_Master_Account b

where a.Master_Account_Id >1

and a.Customer_Id <>1

and a.Customer_Id <>b.Customer_Id

and a.Master_Account_Id = b.Master_Account_Id

group by b.Master_Account_Code, b.Description

order by COUNT(b.Master_Account_Code)

/*Query 66 NOT NECESSARY UNLESS CUSTOMER IS USING DEALERS */

--Query 66: Master Accounts with Dealer ID set to 1 in ar_customer --

select distinct a.Customer_Number as "Query 66 - Master Account Customer Number - Dealer ID set to 1 (invalid) - (NO ROWS SHOULD BE RETURNED) ",

a.customer_name as "Customer_Name",

a.Dealer_Id

from AR_Customer a, AR_Master_Account b, AR_Dealer c

where a.customer_id = b.Customer_Id

and a.Dealer_Id = 1

and a.Customer_Id >1

/*Query 67 NOT NECESSARY UNLESS CUSTOMER IS USING DEALERS */

--Query 67: Master Accounts without a record in ar_dealer --

select distinct a.Customer_Number as "Query 67 - Master Account Customer Number - No Record in Dealer Table - (NO ROWS SHOULD BE RETURNED)",

a.customer_name as "Customer_Name",

a.Dealer_Id

from AR_Customer a, AR_Master_Account b

where a.customer_id = b.Customer_Id

and a.Dealer_Id not in (select Dealer_Id from AR_Dealer)

and a.Customer_Id >1

--***Open A/R Invoices & Credits KEY FIELD VALIDATIONS***

--***AR_INVOICE KEY FIELD VALIDATIONS*** --

--Query 68: AR_Invoice - Type_JSCO (all should be O) --

select Distinct b.Type_JSCO as "Query 68 - AR_Invoice - TYPE JSCO - All Should = O",

count(b.type_jsco) as "Number of Invoices by Type"

from ar_customer a, ar_invoice b

where a.customer_id=b.customer_id

and a.customer_id<>1

group by b.Type_JSCO

--Query 69: AR_Invoice - Category (all should be G&A or something similar) Should not be any assigned to N/A --

select Distinct c.Category_Code as "Query 69 - AR_Invoice - Category Code - all should be G&A or something similar - None should be N/A",

count(c.Category_Code) as "Number of Invoices by Category Code"

from ar_customer a, ar_invoice b, AR_Category c

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.Category_Id = c.Category_Id

group by c.Category_Code

--Query 70: AR Setup GL   AR Account Number --

select b.account_code as "Query 70 - AR Posting Account from AR_Setup_GL - All invoices should have the debit to this account", a.ar_id

from ar_setup_gl a, gl_account b

where a.ar_id = b.Account_Id

--Query 71: AR_Invoice - AR Account should be the same as ar_setup_gl.ar_id --

select Distinct c.account_Code as "Query 71 - AR_Invoice - Invoice AR GL Account (ALL Should be the Same)",

count(c.Account_Code) as "Number of Invoices by AR GL Account Code"

from ar_customer a, ar_invoice b, gl_account c where a.customer_id=b.customer_id

and a.customer_id<>1

and b.ar_account_id = c.Account_Id

group by c.Account_Code

--Query 72: AR_Invoice - Amount and Net Due are not equal (should be zero rows returned) --

select Distinct c.account_Code as "Query 72 - AR_Invoice - Invoice AR GL Account - Amount and Net Due are not equal - (NO ROWS SHOULD BE RETURNED)",

count(c.Account_Code) as "Number of Invoices by AR GL Account Code"

--select *

from ar_customer a, ar_invoice b, gl_account c

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.ar_account_id = c.Account_Id

and b.Amount <> b.Net_Due

group by c.Account_Code

--Query 73: AR_Invoice - Count of Invoices by Term Code (None should be N/A) --

select Distinct c.Term_Code as "Query 73 - AR_Invoice - Term Code - (None should be N/A) **Provide results to the customer to review for accuracy**",

count(b.Invoice_Number) as "Number of Invoices by Term Code"

from ar_customer a, ar_invoice b, ar_term c where a.customer_id=b.customer_id

and a.customer_id<>1

and b.Term_Id = c.Term_Id

group by c.Term_Code

--Query 74: AR_Invoice - Invoices where Term Code = N/A (Should be zero rows returned) If rows returned, Fix the Data --

select Distinct a.customer_number as "Query 74 - AR_Invoice - Customer Number - Invoices with Term Code N/A - (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, c.Term_Code

from ar_customer a, ar_invoice b, ar_term c

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.Term_Id = 1

--Query 75: AR_Invoice - Count of Invoices by Invoice Description Code (None should be N/A) --

select Distinct c.inv_desc_Code as "Query 75 - AR_Invoice - Invoice Description Code - None Should be N/A - *Provide results to the customer to review*",

count(b.Invoice_Number) as "Number of Invoices by Invoice Description Code"

from ar_customer a, ar_invoice b, ar_invoice_description c

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.invoice_description_Id = c.inv_desc_Id group by c.inv_desc_code

--Query 76: AR_Invoice - Count of Invoices where register_id = 1 (None should = 1) --

select Distinct a.Customer_Number as "Query 76 - AR_Invoice - Customer Number where Invoice Register ID =1 - (NO ROWS SHOULD BE RETURNED)",

count(b.Invoice_Number) as "Number of Invoices with Reg Id =1"

from ar_customer a, ar_invoice b

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.Register_Id = 1

group by a.Customer_Number

--Query 77: AR_Invoice - Count of Invoices where Complete = N (No Rows Should be Returned) --

select Distinct a.Customer_Number as "Query 77 - AR_Invoice - Customer Number where Invoice Complete = N - All Should be set to Y - (NO ROWS SHOULD BE RETURNED)",

count(b.Invoice_Number) as "Number of Invoices Where Complete = N"

from ar_customer a, ar_invoice b

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.Complete = 'N'

group by a.Customer_Number

--Query 78: AR_Invoice - Invoices with an Invalid Bill To record (No Rows Should be Returned) --

select Distinct a.Customer_Number as "Query 78 - AR_Invoice - Invoice Number where the Bill To is invalid - (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number

from AR_Customer a, ar_invoice b, ar_customer_bill c

where a.customer_id = b.Customer_Id

and c.Customer_Id = a.Customer_Id

and b.Customer_Bill_Id not in (select Customer_Bill_Id from AR_Customer_Bill)

--Query 79: AR_Invoice - Invoices Should not be in Print Queue (No Rows Should be Returned) --

select Distinct a.Customer_Number as "Query 79 - AR_Invoice Customer# - In Print Queue Should be N - (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number

from AR_Customer a, ar_invoice b

where a.customer_id = b.Customer_Id

and b.In_Print_Queue = 'Y'

order by a.Customer_Number, b.Invoice_Number

--Query 80: AR_Invoice - Invoices Should not be in Email Queue (No Rows Should be Returned)--

select Distinct a.Customer_Number as "Query 80 - AR_Invoice - In Email Queue Should be N - (NO ROWS SHOULD BE RETURNED)", b.Invoice_Number

from AR_Customer a, ar_invoice b

where a.customer_id = b.Customer_Id

and b.In_Email_Queue = 'Y'

order by a.Customer_Number, b.Invoice_Number

--Query 81: AR_Invoice - Count of Invoices by Branch --

select Distinct c.Branch_Code as "Query 81 - AR_Invoice - Branch Code - **Provide results to the customer to review for accuracy**",

count(b.Invoice_Number) as "Number of Invoices by Branch"

from ar_customer a, ar_invoice b, ar_branch c

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.Branch_id = c.Branch_Id

group by c.Branch_Code

--Query 82: AR_Invoice where Credit <> 0 (No Rows Should be Returned)--

select Distinct a.Customer_Number as "Query 82 - AR_Invoice - Customer Number - Where Credit <>0 - (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, b.Invoice_Id, b.Amount, b.Credit

from AR_Customer a, ar_invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and b.Credit <>0

order by a.Customer_Number, b.Invoice_Number

--Query 83: AR_Invoice where Payment <> 0 (No Rows Should be Returned)--

select Distinct a.Customer_Number as "Query 83 - AR_Invoice - Customer Number - Where Payment <>0 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, b.Invoice_Id, b.Amount, b.Payment

from AR_Customer a, ar_invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and b.Payment <>0

order by a.Customer_Number, b.Invoice_Number

--Query 84: AR_Invoice where tax_amount <> 0 (No Rows Should be Returned)--

select Distinct a.Customer_Number as "Query 84 - AR_Invoice - Customer Number - Where Tax_Amount <>0 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, b.Invoice_Id, b.Amount, b.Tax_Amount

from AR_Customer a, ar_invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and b.Tax_Amount <>0

order by a.Customer_Number, b.Invoice_Number

--Query 85: AR_Invoice where Advance_Deposit <> 0 (No Rows Should be Returned)--

select Distinct a.Customer_Number as "Query 85 - AR_Invoice - Customer Number - Where Advance_Deposit <>0 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, b.Invoice_Id, b.Amount, b.Advance_Deposit

from AR_Customer a, ar_invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and b.Advance_Deposit <>0

order by a.Customer_Number, b.Invoice_Number

--Query 86: AR_Invoice where Late_Fee_Amt <> 0.00 (No Rows Should be Returned)--

select Distinct a.Customer_Number as "Query 86 - AR_Invoice - Customer Number - Where Late_Fee_Amt <> 0.00 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, b.Invoice_Id, b.Amount, b.Late_Fee_Amt

from AR_Customer a, ar_invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and b.Late_Fee_Amt <> 0.00

order by a.Customer_Number, b.Invoice_Number

--Query 87: AR_Invoice where GST_Tax_Amount <> 0.00 (No Rows Should be Returned)--

select Distinct a.Customer_Number as "Query 87 - AR_Invoice - Customer Number - Where GST_Tax_Amount <> 0.00 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, b.Invoice_Id, b.Amount, b.GST_Tax_Amount

from AR_Customer a, ar_invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and b.GST_Tax_Amount <> 0.00

order by a.Customer_Number, b.Invoice_Number

--Query 88: AR_Invoice where IsTaxable <> N (No Rows Should be Returned)--

select Distinct a.Customer_Number as "Query 88 - AR_Invoice - Customer Number - Where IsTaxable <> N (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, b.Invoice_Id, b.Amount, b.IsTaxable

from AR_Customer a, ar_invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and b.IsTaxable <> 'N'

order by a.Customer_Number, b.Invoice_Number

--Query 89: AR_Invoice where Has_Pending_EFT <> N (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Query 89 - AR_Invoice - Invoice Number - Where Has_Pending_EFT <> N (NO ROWS SHOULD BE RETURNED)",

a.Customer_Number, b.Invoice_Number, b.Invoice_Id, b.Amount, b.Has_Pending_EFT

from AR_Customer a, ar_invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and b.Has_Pending_EFT <> 'N'

order by b.Invoice_Number, a.Customer_Number

--Query 90: AR_Invoice where Department_Id <> 1 (No Rows Should be Returned)--

select Distinct a.Customer_Number as "Query 90 - AR_Invoice - Customer Number - Where Department_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, b.Invoice_Id, b.Amount, b.Department_Id

from AR_Customer a, ar_invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and b.Department_Id <> 1

order by a.Customer_Number, b.Invoice_Number

--Query 91: AR_Invoice where Job_id <> 1 (No Rows Should be Returned)--

select Distinct a.Customer_Number as "Query 91 - AR_Invoice - Customer Number - Where Job_id <> 1 (NO ROWS SHOULD BE RETURNED)", b.Invoice_Number, b.Invoice_Id, b.Amount, b.Job_id

from AR_Customer a, ar_invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and b.Job_id <> 1

order by a.Customer_Number, b.Invoice_Number

--Query 92: AR_Invoice where Service_Ticket_Id <> 1 (No Rows Should be Returned)--

select Distinct a.Customer_Number as "Query 92 - AR_Invoice - Customer Number - Where Service_Ticket_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, b.Invoice_Id, b.Amount, b.Service_Ticket_Id

from AR_Customer a, ar_invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and b.Service_Ticket_Id <> 1

order by a.Customer_Number, b.Invoice_Number

--Query 93: AR_Invoice where Service_Start_Date <> 1899.. (No Rows Should be Returned)--

select Distinct a.Customer_Number as "Query 93 - AR_Invoice - Customer Number - Where Service_Start_Date <> 1899-12-30 00:00:00:000 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, b.Invoice_Id, b.Amount, b.Service_Start_Date

from AR_Customer a, ar_invoice b, AR_Invoice_Item c where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and b.Service_Start_Date <> '1899-12-30 00:00:00:000'

order by a.Customer_Number, b.Invoice_Number

--Query 94: AR_Invoice where Service_End_Date <> 1899.. (No Rows Should be Returned)--

select Distinct a.Customer_Number as "Query 94 - AR_Invoice - Customer Number - Where Service_End_Date <> 1899-12-30 00:00:00:000 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, b.Invoice_Id, b.Amount, b.Service_End_Date

from AR_Customer a, ar_invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and b.Service_End_Date <> '1899-12-30 00:00:00:000'

order by a.Customer_Number, b.Invoice_Number

--Query 95: AR_Invoice where Payment_Date <> 1899.. (No Rows Should be Returned)--

select Distinct a.Customer_Number as "Query 95 - AR_Invoice - Customer Number - Where Payment_Date <> 1899-12-30 00:00:00:000 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, b.Invoice_Id, b.Amount, b.Payment_Date

from AR_Customer a, ar_invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and b.Payment_Date <> '1899-12-30 00:00:00:000'

order by a.Customer_Number, b.Invoice_Number

--*** AR_INVOICE_ITEM KEY VALIDATION FIELDS ** --

--Query 96: AR_Invoice_Item - Count of Invoices by Item Code - (All Item Codes should be the same for data conversion) --

select distinct c.Item_Code as "Query 96 - AR_Invoice_Item - Item Code for Converted Invoices - All Item Codes Should be the Same",

count(c.Item_Code) as "Number of Invoices by Item Code"

from ar_customer a, ar_invoice b, ar_item c, AR_Invoice_Item d

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.Invoice_Id = d.Invoice_Id

and d.Item_Id = c.Item_Id

group by c.Item_Code

order by c.Item_Code

--Query 97: AR_Invoice_Item Where Item Description is blank (No Rows Should be Returned)--

select a.Customer_Number as "Query 97 - AR_Invoice_Item - Customer Number - Where Invoice Item Description is Blank (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, b.Invoice_Id, c.description

from AR_Customer a, ar_invoice b, AR_Invoice_Item c where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Description = ''

order by a.Customer_Number, b.Invoice_Number

--Query 98: AR_Invoice_Item Where the Quantity <> 1 (No Rows Should be Returned)--

select a.Customer_Number as "Query 98 - AR_Invoice_Item - Customer Number - Where Quantity <>1 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, b.Invoice_Id

from AR_Customer a, ar_invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Quantity <>1

order by a.Customer_Number, b.Invoice_Number

--Query 99: AR_Invoice_Item Where the Rate <> Amount (No Rows Should be Returned)--

select a.Customer_Number as "Query 99 - AR_Invoice_Item - Customer Number - Where Rate <> Amount (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, b.Invoice_Id

from AR_Customer a, ar_invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Amount <> c.Rate

order by a.Customer_Number, b.Invoice_Number

--Query 100: AR_Invoice_Item Where the Register ID = 1 (No Rows Should be Returned)--

select a.Customer_Number as "Query 100 - AR_Invoice_Item - Customer Number - Where Invoice Register ID <>1 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Number, b.Invoice_Id, b.Complete, c.Register_Id

from AR_Customer a, ar_invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Register_Id =1

order by a.Customer_Number, b.Invoice_Number, c.register_id

--Query 101: AR_Invoice_Item Where the Part ID <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 101 - AR_Invoice_Item - Invoice Number - Where Part_Id > @partid (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Id, c.Part_Id

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Part_Id <>1

order by b.Invoice_Number

--Query 102: AR_Invoice_Item Where the Part ID <> 2 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 102 - AR_Invoice_Item - Invoice Number - Where Unit_Of_Measure_Id <> 2 (NO ROWS SHOULD BE RETURNED)",

 b.Invoice_Id, c.Unit_Of_Measure_Id

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Unit_Of_Measure_Id <> 2

order by b.Invoice_Number

--

--Query 103: AR_Invoice_Item Where the Job_Install_Id <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 103 - AR_Invoice_Item - Invoice Number - Where Job_Install_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

 b.Invoice_Id, c.Job_Install_Id

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Job_Install_Id <>1

order by b.Invoice_Number

--Query 104: AR_Invoice_Item Where the Job_Equipment_Id <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 104 - AR_Invoice_Item - Invoice Number - Where Job_Equipment_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Id, c.Job_Equipment_Id

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Job_Equipment_Id <>1

order by b.Invoice_Number

--Query 105: AR_Invoice_Item Where the Job_Recurring_Id <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 105 - AR_Invoice_Item - Invoice Number - Where Job_Recurring_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Id, c.Job_Recurring_Id

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Job_Recurring_Id <>1

order by b.Invoice_Number

--Query 106: AR_Invoice_Item Where the Asset_Register_Id <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 106 - AR_Invoice_Item - Invoice Number - Where Asset_Register_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

 b.Invoice_Id, c.Asset_Register_Id

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Asset_Register_Id <>1

order by b.Invoice_Number

--select * from AR_Invoice_Item

--Query 107: AR_Invoice_Item Where the COGS_Register_Id <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 107 - AR_Invoice_Item - Invoice Number - Where COGS_Register_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

 b.Invoice_Id, c.COGS_Register_Id

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.COGS_Register_Id <>1

order by b.Invoice_Number

--Query 108: AR_Invoice_Item Where the Journal_Id <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 108 - AR_Invoice_Item - Invoice Number - Where Journal_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Id, c.Journal_Id

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Journal_Id <>1

order by b.Invoice_Number

--Query 109: AR_Invoice_Item Where the ICO_Register_Id <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 109 - AR_Invoice_Item - Invoice Number - Where ICO_Register_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Id, c.ICO_Register_Id

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.ICO_Register_Id <>1

order by b.Invoice_Number

--Query 110: AR_Invoice_Item Where the Salesperson_Id <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 110 - AR_Invoice_Item - Invoice Number - Where Salesperson_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Id, c.Salesperson_Id

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Salesperson_Id <>1

order by b.Invoice_Number

--Query 111: AR_Invoice_Item Where the Vendor_Id <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 111 - AR_Invoice_Item - Invoice Number - Where Vendor_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Id, c.Vendor_Id

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Vendor_Id > @vendorid

order by b.Invoice_Number

--Query 112: AR_Invoice_Item Where the Royalty_Debit_Id <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 112 - AR_Invoice_Item - Invoice Number - Where Royalty_Debit_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Id, c.Royalty_Debit_Id

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Royalty_Debit_Id <>1

order by b.Invoice_Number

--Query 113: AR_Invoice_Item Where the Royalty_Credit_Id <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 113 - AR_Invoice_Item - Invoice Number - Where Royalty_Credit_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Id, c.Royalty_Credit_Id

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Royalty_Credit_Id <>1

order by b.Invoice_Number

--Query 114: AR_Invoice_Item Where the Inter_Branch_Register_Id <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 114 - AR_Invoice_Item - Invoice Number - Where Inter_Branch_Register_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Id, c.Inter_Branch_Register_Id

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Inter_Branch_Register_Id <>1

order by b.Invoice_Number

--Query 115: AR_Invoice_Item Where the Service_Start_Date <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 115 - AR_Invoice_Item - Invoice Number - Where Service_Start_Date <> 1899-12-30 00:00:00:000 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Id, c.Service_Start_Date

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Service_Start_Date <> '1899-12-30 00:00:00:000'

order by b.Invoice_Number

--Query 116: AR_Invoice_Item Where the Service_End_Date <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 116 - AR_Invoice_Item - Invoice Number - Where Service_End_Date <> 1899-12-30 00:00:00:000 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Id, c.Service_End_Date

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Service_End_Date <> '1899-12-30 00:00:00:000'

order by b.Invoice_Number

--Query 117: AR_Invoice_Item Where the Royalty_End_Date <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 117 - AR_Invoice_Item - Invoice Number - Where Royalty_End_Date <> 1899-12-30 00:00:00:000 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Id, c.Royalty_End_Date

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Royalty_End_Date <> '1899-12-30 00:00:00:000'

order by b.Invoice_Number

--Query 118: AR_Invoice_Item Where the Royalty_Start_Date <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 118 - AR_Invoice_Item - Invoice Number - Where Royalty_Start_Date <> 1899-12-30 00:00:00:000 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Id, c.Royalty_Start_Date

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Royalty_Start_Date <> '1899-12-30 00:00:00:000'

order by b.Invoice_Number

--Query 119: AR_Invoice_Item Where the RMR_Quantity <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 119 - AR_Invoice_Item - Invoice Number - Where RMR_Quantity <> 1.00 (NO ROWS SHOULD BE RETURNED)",

b.Invoice_Id, c.RMR_Quantity

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.RMR_Quantity <> 1.00

order by b.Invoice_Number

```
--Query 120:  AR_Invoice_Item Where the RMR_Quantity_Rate <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 120 - AR_Invoice_Item - Invoice Number - Where RMR_Quantity_Rate <> 0.00 (NO ROWS SHOULD BE RETURNED)",

 b.Invoice_Id, c.RMR_Quantity_Rate

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.RMR_Quantity_Rate <> 0.00

order by b.Invoice_Number
```

```
--Query 121:  AR_Invoice_Item Where the Cycle_Amount <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 121 - AR_Invoice_Item - Invoice Number - Where Cycle_Amount <> 0.00 (NO ROWS SHOULD BE RETURNED)",

 b.Invoice_Id, c.Cycle_Amount

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Cycle_Amount <> 0.00

order by b.Invoice_Number
```

```sql
--Query 122:  AR_Invoice_Item Where the Service_Ticket_Id <> 1 (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 122 - AR_Invoice_Item - Invoice Number - Where Service_Ticket_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

 b.Invoice_Id, c.Service_Ticket_Id

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Service_Ticket_Id <> 1

order by b.Invoice_Number



/* NEED TO CONSULT WITH CAROLYN JOHNSON */


--Query 123:  AR_Invoice_Item Where the Tax_Exempt <> Y (No Rows Should be Returned)--

select distinct b.Invoice_Number as "Invoice_Number",

a.Customer_Number as "Query 123 - AR_Invoice_Item - Invoice Number - Where Tax_Exempt <> Y (NO ROWS SHOULD BE RETURNED)",

 b.Invoice_Id, c.Tax_Exempt

 --select *

from AR_Customer a, AR_Invoice b, AR_Invoice_Item c

where a.customer_id = b.Customer_Id

and c.Invoice_Id = b.Invoice_Id

and c.Tax_Exempt <> 'Y'
```

```
order by b.Invoice_Number

--***AR_CREDIT KEY VALIDATION FIELDS*** --

--Query 124:  AR_Credit - Type_JSCO (all should be O)  --

select b.Type_JSCO as "Query 124  - AR_Credit - TYPE_JSCO (All Should be O) ",

count(b.type_jsco) as "Number of Credits by Type"

from ar_customer a, ar_credit b

where a.customer_id=b.customer_id

and a.customer_id<>1

group by b.Type_JSCO

--Query 125:  AR_Credit - Category (all should be G&A or something similar)  Should not be any assigned to N/A  --

select c.Category_Code as "Query 125   AR_Credit - Category Code (all should be G&A or something similar) None should be N/A ",

count(c.Category_Code) as "Number of Credits by Category Code"

from ar_customer a, AR_Credit b, AR_Category c

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.Category_Id = c.Category_Id
```

group by c.Category_Code

--select * from AR_Category

--Query 126: AR Setup GL Unapplied Credit Memos GL Account Number --

select a.Unap_Credit_Id , b.Account_Code as "Query 126 - AR_Setup_GL Unapplied Credit Memos GL Account - This should be the Debit Account Used on all converted Credits",

b.Description

from ar_setup_gl a, GL_Account b

where a.Unap_Credit_Id = b.Account_Id

--Query 127: AR_Credit - Unapplied Credit Memos GL Account should be the same as ar_setup_gl.unap_Credit_Id --

select b.account_id, c.account_Code as "Query 127 - AR_Credit - Account_id should be the same as ar_setup_gl.unap_Credit_Id",

count(c.Account_Code) as "Number of Credits by Unaplied Credit Memos GL Account Code"

from ar_customer a, AR_Credit b, gl_account c

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.account_id = c.Account_Id

group by b.Account_Id, c.Account_Code

```
--Query 128:  AR_Credit - Amount <=0 (should be zero rows returned)  --

select a.customer_number, b.credit_number as "Query 128  - AR_Credit - Customer Number Where Credit Amount <=0  (NO ROWS SHOULD BE RETURNED)",

b.Amount

from ar_customer a, AR_Credit b

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.Amount <=0

order by b.credit_number
```

```
--Query 129:  AR_Credit - Used Amount >0 (should be zero rows returned)  --

select a.customer_number, b.credit_number as "Query 129  - AR_Credit - Customer Number Where Used_Amount >0 (NO ROWS SHOULD BE RETURNED)",

b.Amount, Used_Amount

from ar_customer a, AR_Credit b

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.Used_Amount >0

order by b.credit_number
```

```
--Query 130:  AR_Credit - Count of Credits by Invoice Description Code (None should be N/A) --

select c.inv_desc_Code as "Query 130  AR_Credit - Invoice Description Code (None should be N/A)",

count(b.Credit_Number) as "Number of Credits by Invoice Description Code"
```

```sql
from ar_customer a, AR_Credit b, ar_invoice_description c
where a.customer_id=b.customer_id
and a.customer_id<>1
and b.invoice_description_Id = c.inv_desc_Id
group by c.inv_desc_code
order by c.Inv_Desc_Code
```

```sql
--Query 131:  AR_Credit - Count of Credits by Credit Reason Code (None should be N/A) --
select c.Credit_Reason_Code as "Query 131   - AR_Credit - Credit Reason Code - (None should be N/A)",
count(b.Credit_Number) as "Number of Credits by Credit Reason Code"
from ar_customer a, AR_Credit b, AR_Credit_Reason c
where a.customer_id=b.customer_id
and a.customer_id<>1
and b.Credit_Reason_Id = c.Credit_Reason_Id
group by c.Credit_Reason_Code
order by c.Credit_Reason_Code
```

```sql
--Query 132: AR_Credit -  Credits Where the Register ID = 1 (No Rows Should be Returned)--
select a.Customer_Number as "Query 132 - AR_Credit - Customer Number Where the Register ID = 1 (NO ROWS SHOULD BE RETURNED)",
b.Credit_Number, b.Credit_Id, b.Complete
from AR_Customer a, ar_credit b
```

```
where a.customer_id = b.Customer_Id

and b.Credit_Id >1

and b.Register_Id =1

order by a.Customer_Number, b.Credit_Number
```

```
--Query 133:  AR_Credit - Count of Credits where Complete = N (No Rows Should be Returned) --

select a.Customer_Number as "Query 133  - AR_Credit - Customer Number Where Complete = N (NO ROWS SHOULD BE RETURNED)",

b.Credit_Number

from ar_customer a, ar_credit b

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.credit_id >1

and b.Complete = 'N'

order by a.Customer_Number
```

```
--Query 134:  AR_Credit with an Invalid Bill To record (No Rows Should be Returned) --

select a.Customer_Number as "Query 134 - AR_Credit - Customer Number Where the Credit Bill To is invalid (NO ROWS SHOULD BE RETURNED)",

b.Credit_Number, b.Customer_Bill_Id

from AR_Customer a, AR_Credit b, ar_customer_bill c

where a.customer_id = b.Customer_Id

and b.Credit_Id >1
```

and c.Customer_Id = a.Customer_Id

and b.Customer_Bill_Id not in (select Customer_Bill_Id from AR_Customer_Bill)

--Query 135: AR_Credit - Count of Credits by Branch --

select c.Branch_Code as "Query 135   AR_Credit - Branch Code - None should be N/A - **Provide results to the customer to review for accuracy**",

count(b.credit_Number) as "Number of Credits by Branch"

from ar_customer a, ar_credit b, ar_branch c

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.credit_id >1

and b.Branch_id = c.Branch_Id

group by c.Branch_Code

--Query 136: AR_Credit - Count of Credits by Department --

select c.Department_Code as "Query 136 - AR_Credit - Department Code - ALL should be N/A ",

b.Department_Id,

count(b.credit_Number) as "Number of Credits by Department"

from ar_customer a, ar_credit b, SY_Department c

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.credit_id >1

```
and b.Department_Id = c.Department_Id

group by c.Department_Code, b.Department_Id




----Query 137:  AR_Credit - Count of Credits  by Job ID --

select c.job_Code as "Query 137  AR_Credit - Job Code  - ALL should be 000 and Job ID should be 1",

b.job_Id,

count(b.credit_Number) as "Number of Credits by Job Code"

from ar_customer a, ar_credit b, oe_job c

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.credit_id >1

and b.job_Id = c.job_Id

group by c.job_Code, b.job_Id




--Query 138:  AR_Credit - Count of Credits  by Tax Amount --

select b.Tax_Amount as "Query 138   AR_Credit - Tax Amount  - ALL should be 0.00 ",

count(b.credit_Number) as "Number of Credits by Tax Amount "

from ar_customer a, ar_credit b

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.credit_id >1
```

```
group by b.Tax_Amount

--Query 139:  AR_Credit - Count of Credits  by Service Start Date --

select b.Service_Start_Date as "Query 139   AR_Credit - Service Start Date  - ALL should be 1899-12-30 00:00:00:000 ",

count(b.credit_Number) as "Number of Credits by Service Start Date "

from ar_customer a, ar_credit b

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.credit_id >1

group by b.Service_Start_Date

--Query 140:  AR_Credit - Count of Credits  by Service End Date --

select b.Service_End_Date as "Query 140   AR_Credit - Service End Date  - ALL should be 1899-12-30 00:00:00:000 ",

count(b.credit_Number) as "Number of Credits by Service End Date "

from ar_customer a, ar_credit b

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.credit_id >1

group by b.Service_End_Date
```

--Query 141: AR_Credit - Count of Credits by Is Taxable --

select b.IsTaxable as "Query 141 AR_Credit - Is_Taxable - ALL should be = N ",

count(b.credit_Number) as "Number of Credits by Is Taxable "

from ar_customer a, ar_credit b

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.credit_id >1

group by b.IsTaxable

--Query 142: AR_Credit - Count of Credits by Service Ticket ID --

select c.Ticket_Number as "Query 142 AR_Credit - Service Ticket Number - ALL should be 0 and Service Ticket ID should be 1",

b.Service_Ticket_Id, count(b.credit_Number) as "Number of Credits by Ticket Number"

from ar_customer a, ar_credit b, SV_Service_Ticket c

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.credit_id >1

and b.Service_Ticket_Id = c.Service_Ticket_Id

group by c.Ticket_Number, b.Service_Ticket_Id

```
--Query 143:  AR_Credit - Count of Credits  by GST_Tax_Amount --

select b.GST_Tax_Amount as "Query 143   AR_Credit  - GST_Tax_Amount - ALL should be 0.00 ",

count(b.credit_Number) as "Number of Credits by GST Tax Amount "

from ar_customer a, ar_credit b

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.credit_id >1

group by b.GST_Tax_Amount
```

```
--Query 144:  AR_Credit - Count of Credits  by UserCode --

     /* UserCode may be set to a Conversion code */

select b.UserCode as "Query 144  AR_Credit - UserCode  - ALL should be Administrator ",

count(b.credit_Number) as "Number of Credits by UserCode "

from ar_customer a, ar_credit b

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.credit_id >1

group by b.UserCode
```

```
--Query 145:  AR_Credit - Count of Credits  by Posting_Date --

select b.Posting_Date as "Query 145   AR_Credit  - Posting_Date  - ALL should be the Same-Verify the date is correct ",

count(b.credit_Number) as "Number of Credits by Posting Date "
```

from ar_customer a, ar_credit b

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.credit_id >1

group by b.Posting_Date

--** Credit Detail Records (ar_credit_item) ** --

--Query 146: AR_Credit_Item - Count of Credits that do not have a record in ar_credit_item --

select count(b.Credit_Number) as "Query 146   AR_Credit - Number of Credits that do not have a record in ar_credit_item (NO ROWS SHOULD BE RETURNED)"

from ar_customer a, AR_Credit b, ar_item c, AR_Credit_Item d

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.Credit_Id = d.Credit_Id

and b.Credit_Id not in (select Credit_Id from AR_Credit_Item)

group by b.Credit_Number

--Query 147: AR_Credit_Item - Count of Credits by Item Code - (All Item Codes should be the same for data conversion) --

select distinct c.Item_Code as "Query 147 - Number of Credits by Item Code - The GL Account should start with a 3", count(c.Item_Code) as "Number of Credits by Item Code", e.Account_Code

from ar_customer a, ar_credit b, ar_item c, AR_credit_Item d, GL_Account e

where a.customer_id=b.customer_id

and a.customer_id<>1

and b.credit_Id = d.credit_Id

and d.Item_Id = c.Item_Id

and c.Account_Id = e.Account_Id

group by c.Item_Code, e.Account_Code

order by c.Item_Code

--Query 148: AR_Credit_Item Where the Item Description is blank (No Rows Should be Returned)--

select a.Customer_Number as "Query 148 - AR_Credit_Item - Where Item Description is Blank (NO ROWS SHOULD BE RETURNED)",

b.Credit_Number, b.Credit_Id

from AR_Customer a, AR_Credit b, AR_Credit_Item c

where a.customer_id = b.Customer_Id

and c.Credit_Id = b.Credit_Id

and c.Description = ''

order by a.Customer_Number, b.Credit_Number

--Query 149: AR_Credit_Item Where the Rate <> Amount (No Rows Should be Returned)--

select distinct b.Credit_Number,

a.Customer_Number as "Query 149 AR_Credit_Item - Where Rate <> Amount (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id

from AR_Customer a, ar_Credit b, AR_Credit_Item c

where a.customer_id = b.Customer_Id

and c.Credit_Id = b.Credit_Id

and c.Amount <> c.Rate

order by b.Credit_Number

--Query 150: AR_Credit_Item Where the Register ID = 1 (No Rows Should be Returned)--

select a.Customer_Number as "Query 150 AR_Credit_Item - Where Register ID = 1 (NO ROWS SHOULD BE RETURNED)" ,

b.Credit_Number, b.Credit_Id, b.Complete, c.Register_Id

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.Register_Id =1

order by a.Customer_Number, b.Credit_Number, c.register_id

--Query 151: AR_Credit_Item Where the Part ID <> 1 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number", a.Customer_Number as "Query 151 AR_Credit_Item - Credit Number - Where Part_Id > @partid (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, b.Complete, c.Register_Id, c.Part_Id

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.part_id <>1

order by a.Customer_Number, b.Credit_Number, c.register_id

--Query 152: AR_Credit_Item Where the Part ID <> 1 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number",

a.Customer_Number as "Query 152 - AR_Credit_Item - Credit Number - Where Unit_Of_Measure_ID <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, c.Unit_Of_Measure_Id, c.Part_Id

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.Unit_Of_Measure_Id <>1

order by b.credit_number

--Query 153: AR_Credit_Item - Count of Credits by Service Start Date --

select c.Service_Start_Date as "Query 153   AR_Credit_Item - Service Start Date - ALL should be 1899-12-30

00:00:00:000 ",

count(b.credit_Number) as "Number of Credit Lines by Service Start Date "

from ar_customer a, ar_credit b, AR_Credit_Item c

where a.customer_id=b.customer_id

and a.customer_id<>1

and c.Credit_Id = b.Credit_Id

and b.credit_id >1

group by c.Service_Start_Date

order by c.Service_Start_Date

--Query 154: AR_Credit_Item - Count of Credits by Service End Date --

select c.Service_End_Date as "Query 154   AR_Credit_Item - Service End Date - ALL should be 1899-12-30 00:00:00:000 ",

count(b.credit_Number) as "Number of Credit Lines by Service End Date "

from ar_customer a, ar_credit b, AR_Credit_Item c

where a.customer_id=b.customer_id

and a.customer_id<>1

and c.Credit_Id = b.Credit_Id

and b.credit_id >1

group by c.Service_End_Date

order by c.Service_End_Date

--Query 155: AR_Credit_Item Where the Asset Register ID <> 1 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number",

a.Customer_Number as "Query 155 - AR_Credit_Item - Credit Number - Where Asset_Register_ID <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, b.Complete, c.Asset_Register_Id

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.Asset_Register_Id <>1

order by a.Customer_Number, b.Credit_Number, c.Asset_Register_Id

--Query 156: AR_Credit_Item Where the COGS Register ID <> 1 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number",

a.Customer_Number as "Query 156 - AR_Credit_Item - Credit Number - Where COGS_Register_ID <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, b.Complete, c.COGS_Register_Id

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.COGS_Register_Id <>1

order by a.Customer_Number, b.Credit_Number, c.COGS_Register_Id

--Query 157: AR_Credit_Item Where the Journal ID <> 1 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number",

a.Customer_Number as "Query 157 AR_Credit_Item - Credit Number - Where Journal_ID <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, b.Complete, c.Journal_Id

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.Journal_Id <>1

order by a.Customer_Number, b.Credit_Number, c.Journal_Id

--Query 158: AR_Credit_Item Where the Master Item ID <> 1 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number",

a.Customer_Number as "Query 158 AR_Credit_Item - Credit Number - Where Master_Item_ID <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, b.Complete, c.Master_Item_Id

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.Master_Item_Id <>1

order by a.Customer_Number, b.Credit_Number, c.Master_Item_Id

--Query 159: AR_Credit_Item Where the ICO Register ID <> 1 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number",

a.Customer_Number as "Query 159 AR_Credit_Item - Credit Number - Where ICO_Register_ID <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, b.Complete, c.ICO_Register_Id

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.ICO_Register_Id <>1

order by a.Customer_Number, b.Credit_Number, c.ICO_Register_Id

--Query 160: AR_Credit_Item Where the Job Install ID <> 1 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number",

a.Customer_Number as "Query 160 - AR_Credit_Item - Credit Number - Where Job_Install_ID <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, b.Complete, c.Job_Install_Id

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.Job_Install_Id <>1

order by a.Customer_Number, b.Credit_Number, c.Job_Install_Id

--Query 161: AR_Credit_Item Where the Job Equipment ID <> 1 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number",

a.Customer_Number as "Query 161 - AR_Credit_Item - Credit Number - Where Job_Equipment_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, b.Complete, c.Job_Equipment_Id

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.Job_Equipment_Id <>1

order by a.Customer_Number, b.Credit_Number, c.Job_Equipment_Id

--Query 162: AR_Credit_Item Where the Job Recurring ID <> 1 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number",

a.Customer_Number as "Query 162 - AR_Credit_Item - Credit Number - Where Job_Recurring_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, b.Complete, c.Job_Recurring_Id

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.Job_Recurring_Id <>1

order by a.Customer_Number, b.Credit_Number, c.Job_Recurring_Id

--Query 163: AR_Credit_Item Where the Salesperson ID < > 1 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number",

a.Customer_Number as "Query 163 - AR_Credit_Item - Credit Number - Where Salesperson_Id < > 1 (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, b.Complete, c.Salesperson_Id

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.Salesperson_Id < >1

order by a.Customer_Number, b.Credit_Number, c.Salesperson_Id

--Query 164: AR_Credit_Item Where the Vendor ID < > 1 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number",

a.Customer_Number as "Query 164 - AR_Credit_Item - Credit Number - Where Vendor_Id < > 1 (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, b.Complete, c.Vendor_Id

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.Vendor_Id > @vendorid

order by a.Customer_Number, b.Credit_Number, c.Vendor_Id

--Query 165: AR_Credit_Item Where the Royalty Credit ID <> 1 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number",

a.Customer_Number as "Query 165 - AR_Credit_Item - Credit Number - Where Royalty_Credit_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, b.Complete, c.Royalty_Credit_Id

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.Royalty_Credit_Id <>1

order by a.Customer_Number, b.Credit_Number, c.Royalty_Credit_Id

--Query 166: AR_Credit_Item Where the Royalty Debit ID <> 1 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number",

a.Customer_Number as "Query 166 - AR_Credit_Item - Credit Number - Where Royalty_Debit_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, b.Complete, c.Royalty_Debit_Id

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.Royalty_Debit_Id <>1

order by a.Customer_Number, b.Credit_Number, c.Royalty_Debit_Id

--Query 167: AR_Credit_Item Where the COGS Amount < > 0.00 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number",

a.Customer_Number as "Query 167 - AR_Credit_Item - Credit Number - Where COGS_Amount < > 0.00 (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, b.Complete, c.COGS_Amount

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.COGS_Amount < >0.00

order by a.Customer_Number, b.Credit_Number, c.COGS_Amount

--Query 168: AR_Credit_Item Where the Monthly Royalty < > 0.00 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number",

a.Customer_Number as "Query 168 - AR_Credit_Item - Credit Number - Where Monthly_Royalty < > 0.00 (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, b.Complete, c.Monthly_Royalty

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.Monthly_Royalty < >0.00

order by a.Customer_Number, b.Credit_Number, c.Monthly_Royalty

--Query 169: AR_Credit_Item Where the Royalty Amount < > 0.00 (No Rows Should be Returned)--

select distinct b.credit_number as "Credit_Number",

a.Customer_Number as "Query 169 - AR_Credit_Item - Credit Number - Where Royalty_Amount < > 0.00 (NO ROWS SHOULD BE RETURNED)",

b.Credit_Id, b.Complete, c.Royalty_Amount

from AR_Customer a, ar_credit b, AR_credit_Item c

where a.customer_id = b.Customer_Id

and c.credit_Id = b.Credit_Id

and c.Royalty_Amount < >0.00

order by a.Customer_Number, b.Credit_Number, c.Royalty_Amount

--Query 170: AR_Credit_Item - Count of Credits by Royalty Start Date --

select c.Royalty_Start_Date as "Query 170   - AR_Credit_Item - Royalty Start Date   - ALL should be 1899-12-30 00:00:00:000 ",

count(b.credit_Number) as "Number of Credit Lines by Royalty Start Date "

from ar_customer a, ar_credit b, AR_Credit_Item c

where a.customer_id=b.customer_id

and a.customer_id< >1

and c.Credit_Id = b.Credit_Id

and b.credit_id >1

group by c.Royalty_Start_Date

order by c.Royalty_Start_Date

--Query 171: AR_Credit_Item - Count of Credits by Royalty End Date --

select c.Royalty_End_Date as "Query 171 - AR_Credit_Item - Royalty End Date - ALL should be 1899-12-30 00:00:00:000 ",

count(b.credit_Number) as "Number of Credit Lines by Royalty End Date "

from ar_customer a, ar_credit b, AR_Credit_Item c

where a.customer_id=b.customer_id

and a.customer_id<>1

and c.Credit_Id = b.Credit_Id

and b.credit_id >1

group by c.Royalty_End_Date

order by c.Royalty_End_Date

--Query 172: AR_Credit_Item - Count of Credits by Cycle Amount (No Rows Should be Returned) --

select c.Cycle_Amount as "Query 172 - AR_Credit_Item - Credit Number - Where Cycle Amount <> 0.00 - (NO ROWS SHOULD BE RETURNED)",

count(b.credit_Number) as "Number of Credit Lines by Cycle Amount "

from ar_customer a, ar_credit b, AR_Credit_Item c

where a.customer_id=b.customer_id

and a.customer_id<>1

and c.Credit_Id = b.Credit_Id

and b.credit_id >1

and c.Cycle_Amount <>0

group by c.Cycle_Amount

order by c.Cycle_Amount

--Query 173: AR_Credit_Item - Count of Credits by Service Ticket Id (No Rows Should be Returned) --

select c.Service_Ticket_Id as "Query 173   AR_Credit_Item - Service Ticket Id - All should be set to 1 (NO ROWS SHOULD BE RETURNED)",

count(b.credit_Number) as "Number of Credit Lines by Service Ticket Id "

from ar_customer a, ar_credit b, AR_Credit_Item c

where a.customer_id=b.customer_id

and a.customer_id<>1

and c.Credit_Id = b.Credit_Id

and b.credit_id >1

and c.Service_Ticket_Id <>1

group by c.Service_Ticket_Id

order by c.Service_Ticket_Id

--Query 174: AR_Credit_Item - Count of Credits by Inter Branch Register Id (No Rows Should be Returned) --

select c.Inter_Branch_Register_Id as "Query 174   - AR_Credit_Item - Where Inter Branch Register Id <> 1 - All should be set to 1 (NO ROWS SHOULD BE RETURNED)",

count(b.credit_Number) as "Number of Credit Lines by Inter Branch Register Id "

from ar_customer a, ar_credit b, AR_Credit_Item c

where a.customer_id=b.customer_id

and a.customer_id<>1

and c.Credit_Id = b.Credit_Id

and b.credit_id >1

and c.Inter_Branch_Register_Id <>1

group by c.Inter_Branch_Register_Id

order by c.Inter_Branch_Register_Id

--***AP_VENDOR FIELD VALIDATIONS***--

--Query 175: AR_Credit_Item - Count of Credits by Tax Exempt (No Rows Should be Returned) --

select c.Tax_Exempt as "Query 175   AR_Credit_Item - Tax Exempt   - All should be set to Y (NO ROWS SHOULD BE RETURNED)",

count(b.credit_Number) as "Number of Credit Lines by Tax Exempt "

from ar_customer a, ar_credit b, AR_Credit_Item c

where a.customer_id=b.customer_id

and a.customer_id<>1

and c.Credit_Id = b.Credit_Id

and b.credit_id >1

and c.Tax_Exempt <> 'Y'

group by c.Tax_Exempt

order by c.Tax_Exempt

--***AP VENDOR FIELD VALIDATIONS***--

--Query 176: AP_Vendor - Company_Name is Blank (NO ROWS SHOULD BE RETURNED) -

select Vendor_Code as "Query 176 - AP_Vendor - Vendor_Code Where Company Name ='' (NO ROWS SHOULD BE RETURNED)" ,

Company_Name

from AP_Vendor

Where Company_Name = ''

and Vendor_Id > @vendorid

Order by Vendor_Code

--Query 177: AP_Vendor - Address_1 is Blank -

select Vendor_Code as "Query 177 - AP_Vendor - Vendor_Code Where Address 1 ='' **Provide results to the customer to review for accuracy**" ,

Address_1

from AP_Vendor

Where Address_1 = ''

and Vendor_Id > @vendorid

Order by Vendor_Code

--Query 178: AP_Vendor - GE_Table1_Id is Invalid (City) (NO ROWS SHOULD BE RETURNED) -

select distinct a.Vendor_Code as "Query 178 - AP_Vendor - Vendor_Code Where GE_Table1_Id is Invalid (City) (NO ROWS SHOULD BE RETURNED)" ,

a.GE_Table1_id as "GE_Table1_ID (City)"

from AP_Vendor a, GE_Table1 b

Where a.GE_Table1_id not in (select Table1_ID from GE_Table1)

and a.Vendor_Id > @vendorid

Order by a.Vendor_Code

--Query 179: AP_Vendor - GE_Table2_Id is Invalid (State/Province) (NO ROWS SHOULD BE RETURNED) -

select distinct a.Vendor_Code as "Query 179 - AP_Vendor - Vendor_Code Where GE_Table2_Id is Invalid (State/Province) (NO ROWS SHOULD BE RETURNED)" ,

a.GE_Table2_id as "GE_Table2_ID (State/Province)"

from AP_Vendor a, GE_Table2 b

Where a.GE_Table2_id not in (select Table2_ID from GE_Table2)

and a.Vendor_Id > @vendorid

Order by a.Vendor_Code

--Query 180: AP_Vendor - GE_Table3_Id is Invalid (Zip/Postal Code) (NO ROWS SHOULD BE RETURNED) -

select distinct a.Vendor_Code as "Query 180 - AP_Vendor - Vendor_Code Where GE_Table3_Id is Invalid (Zip/Postal Code) (NO ROWS SHOULD BE RETURNED)" ,

a.GE_Table3_id as "GE_Table3_ID (Zip/Postal Code)"

from AP_Vendor a, GE_Table3 b

Where a.GE_Table3_id not in (select Table3_ID from GE_Table3) and a.Vendor_Id > @vendorid

Order by a.Vendor_Code

--Query 181: AP_Vendor - Country_Id is Invalid -

select Vendor_Code as "Query 181 - AP_Vendor - Where Country_Id = 1 (NO ROWS SHOULD BE RETURNED)",

Country_Id

from AP_Vendor

Where Country_Id = 1

and Vendor_Id > @vendorid

Order by Vendor_Code

--Query 182: AP_Vendor - Pay_From_Account_id - All Should be 1

select Vendor_Code as "Query 182 - AP_Vendor - Where Pay_From_Account_id <> 1 (All Should be 1) (NO ROWS SHOULD BE RETURNED)",

pay_from_account_Id

from AP_Vendor

Where Pay_From_Account_Id <> 1

and Vendor_Id > @vendorid

Order by Vendor_Code

--Query 183: AP_Vendor - Validate Term_Id

select distinct b.term_code as "Query 183 - AP_Vendor - Term Code - None Should be N/A - **Provide results to the customer to review for accuracy**",

count(a.term_id) as "Count of Vendors by Term Code"

from AP_Vendor a, AR_Term b

Where a.term_id = b.term_id

and a.Vendor_Id > @vendorid

Group by b.Term_Code, a.Term_Id

Order by b.term_code

--Query 184: AP_Vendor - Credit_Limit - All Should be 0.00

select Vendor_Code as "Query 184 - AP_Vendor - Where Credit_Limit <> 0.00 **Provide results to the customer to review for accuracy**",

Credit_Limit

from AP_Vendor

Where Credit_Limit <> 0.00

and Vendor_Id > @vendorid

Order by Vendor_Code

--Query 185: AP_Vendor - Validate Vendor Type

select distinct b.Vendor_Type_Code as "Query 185 - AP_Vendor - Vendor Type - None Should be N/A - **Provide results to the customer to review for accuracy**",

count(b.Vendor_Type_Code) as "Count of Vendors by Vendor_Type"

```
from AP_Vendor a, AP_Vendor_Type b

Where a.Vendor_Type_Id = b.Vendor_Type_Id

and a.Vendor_Id > @vendorid

Group by b.Vendor_Type_Code




--Query 186:  AP_Vendor - Issue_1099 is Valid - All Should be Y or N

select Vendor_Code as "Query 186 - AP_Vendor - Where Issue_1099 <> Y or N (NO ROWS SHOULD BE RETURNED)",

Issue_1099

from AP_Vendor

Where Issue_1099 not in ('N', 'Y')

and Vendor_Id > @vendorid

Order by Vendor_Code




--Query 187:  AP_Vendor - Validate All Vendors are Active

select Vendor_Code as "Query 187 - AP_Vendor - Where Inactive = 'Y' (NO ROWS SHOULD BE RETURNED)",

Inactive

from AP_Vendor

Where Inactive = 'Y'

and Vendor_Id > @vendorid

Order by Vendor_Code
```

--Query 188: AP_Vendor - Checks_Payable_To is Blank (NO ROWS SHOULD BE RETURNED) --

select Vendor_Code as "Query 188 - AP_Vendor - Vendor_Code Where Checks_Payable_To ='' - **Provide results to the customer to review for accuracy**" ,

Checks_Payable_To

--select *

from AP_Vendor

Where Checks_Payable_To = ''

and Vendor_Id > @vendorid

Order by Vendor_Code

--update AP_Vendor

--set Checks_Payable_To = Company_Name

--where Checks_Payable_To = ''

--Query 189: AP_Vendor - Verify Default_Account_Id is Valid --

select Vendor_Code as "Query 189 - AP_Vendor-Vendor_Code Default_Account_Id (N/A is Valid) -*Provide results to the customer to review for accuracy*" ,

b.Account_Code, b.Description

from AP_Vendor a, GL_Account b

Where a.Default_Account_Id = b.Account_Id

and a.Vendor_Id > @vendorid

Order by b.Account_Code, a.Vendor_Code

--Query 190: AP_Vendor - Validate Vendor Branch Code

select distinct b.Branch_Code as "Query 190 - AP_Vendor - Branch Code - None Should be N/A - **Provide results to the customer to review for accuracy**",

count(b.Branch_Code) as "Count of Vendors by Branch Code"

from AP_Vendor a, AR_Branch b

Where a.Branch_Id = b.Branch_Id

and a.Vendor_Id > @vendorid

Group by b.Branch_Code

--Query 191: AP_Vendor - Validate Vendor Category Code

select distinct b.Category_Code as "Query 191 - AP_Vendor - Category Code - None Should be N/A - **Provide results to the customer to review for accuracy**",

count(b.Category_Code) as "Count of Vendors by Category Code"

from AP_Vendor a, AR_Category b

Where a.Category_Id = b.Category_Id

and a.Vendor_Id > @vendorid

Group by b.Category_Code

--Query 192: AP_Vendor - Payment Address_1 is Blank -

select Vendor_Code as "Query 192 - AP_Vendor - Vendor_Code Where Payment Address 1 ='' **Provide results to the customer to review for accuracy**" , Pymt_Address_1

from AP_Vendor

Where Pymt_Address_1 = ''

and Vendor_Id > @vendorid

Order by Vendor_Code

--Query 193: AP_Vendor - Payment GE_Table1_Id is Invalid (City) (NO ROWS SHOULD BE RETURNED) -

select distinct a.Vendor_Code as "Query 193 - AP_Vendor - Vendor_Code Where Pymt_GE_Table1_Id is Invalid (City) (NO ROWS SHOULD BE RETURNED)" ,

a.Pymt_GE_Table1_id as "GE_Table1_ID (City)"

from AP_Vendor a, GE_Table1 b

Where a.Pymt_GE_Table1_id not in (select Table1_ID from GE_Table1)

and a.Vendor_Id > @vendorid

Order by a.Vendor_Code

--Query 194: AP_Vendor - Payment GE_Table2_Id is Invalid (State/Province) (NO ROWS SHOULD BE RETURNED) -

select distinct a.Vendor_Code as "Query 194 - AP_Vendor - Vendor_Code Where Pymt_GE_Table2_Id is Invalid (State/Province) (NO ROWS SHOULD BE RETURNED)" ,

a.Pymt_GE_Table2_id as "GE_Table2_ID (State/Province)"

from AP_Vendor a, GE_Table2 b

Where a.Pymt_GE_Table2_id not in (select Table2_ID from GE_Table2)

and a.Vendor_Id > @vendorid

Order by a.Vendor_Code

--Query 195: AP_Vendor - Payment GE_Table3_Id is Invalid (Zip/Postal Code) (NO ROWS SHOULD BE RETURNED) -

select distinct a.Vendor_Code as "Query 195 - AP_Vendor - Vendor_Code Where Pymt_GE_Table3_Id is Invalid (Zip/Postal Code) (NO ROWS SHOULD BE RETURNED)" ,

a.Pymt_GE_Table3_id as "GE_Table3_ID (Zip/Postal Code)"

from AP_Vendor a, GE_Table3 b

Where a.Pymt_GE_Table3_id not in (select Table3_ID from GE_Table3)

and a.Vendor_Id > @vendorid

Order by a.Vendor_Code

--Query 196: AP_Vendor - GST_Exempt is Valid - All Should be Y or N

select Vendor_Code as "Query 196 - AP_Vendor - Where GST_Exempt <> Y or N (NO ROWS SHOULD BE RETURNED)",

GST_Exempt

from AP_Vendor

Where Issue_1099 not in ('N', 'Y')

and Vendor_Id > @vendorid

Order by Vendor_Code

--Query 197: AP_Vendor - GE1_Description is Invalid (City) (NO ROWS SHOULD BE RETURNED) -

select distinct a.Vendor_Code as "Query 197 - AP_Vendor-Vendor_Code Where City does not match - **Provide results to the customer to review for accuracy**" ,

a.GE1_Description as "Vendor City Name", b.Description as "GE Table 1 Description"

from AP_Vendor a, GE_Table1 b

Where a.GE_Table1_id =b.Table1_ID

and a.GE1_Description <> b.Description

and a.Vendor_Id > @vendorid

Order by a.Vendor_Code

--Query 198: AP_Vendor - GE2_Description is Invalid (State/Province) (NO ROWS SHOULD BE RETURNED) -

select distinct a.Vendor_Code as "Query 198 -AP_Vendor-Vendor_Code Where State/Province does not match - **Provide results to the customer to review for accuracy**" ,

a.GE2_Description as "Vendor State/Province", b.Description as "GE Table 2 Description"

from AP_Vendor a, GE_Table2 b

Where a.GE_Table2_id =b.Table2_ID

and a.GE2_Description <> b.Description

and a.Vendor_Id > @vendorid

Order by a.Vendor_Code

--Query 199: AP_Vendor - GE2_Short is Invalid (State/Province) (NO ROWS SHOULD BE RETURNED) -

select distinct a.Vendor_Code as "Query 199 -AP_Vendor-Vendor_Code Where State/Province does not match - **Provide results to the customer to review for accuracy**" ,

a.GE2_Short as "Vendor State/Province", b.Short as "GE Table 2 Short Description"

from AP_Vendor a, GE_Table2 b

Where a.GE_Table2_id =b.Table2_ID

and a.GE2_Short < > b.Short

and a.Vendor_Id > @vendorid

Order by a.Vendor_Code

--Query 200: AP_Vendor - GE3_Description is Invalid (Zip/Postal Code) (NO ROWS SHOULD BE RETURNED) -

select distinct a.Vendor_Code as "Query 200 -AP_Vendor-Vendor_Code Where Zip/Postal Code- No match -**Provide results to the customer to review for accuracy**" ,

a.GE3_Description as "Vendor Zip/Postal Code", b.Description as "GE Table 3 Description"

from AP_Vendor a, GE_Table3 b

Where a.GE_Table3_id =b.Table3_ID

and a.GE3_Description < > b.Description

and a.Vendor_Id > @vendorid

Order by a.Vendor_Code

--Query 201: AP_Vendor - Payment GE1_Description is Invalid (City) (NO ROWS SHOULD BE RETURNED) -

select distinct a.Vendor_Code as "Query 201 - AP_Vendor-Vendor_Code Where City does not match - **Provide results to the customer to review for accuracy**" ,

a.GE1_Description as "Vendor Payment City Name", b.Description as "GE Table 1 Description"

from AP_Vendor a, GE_Table1 b

Where a.Pymt_GE_Table1_Id =b.Table1_ID

and a.Pymt_GE1_Description < > b.Description

and a.Vendor_Id > @vendorid

Order by a.Vendor_Code

--Query 202: AP_Vendor - Payment GE2_Description is Invalid (State/Province) (NO ROWS SHOULD BE RETURNED) -

select distinct a.Vendor_Code as "Query 202 - AP_Vendor-Vendor_Code Where State/Province - No match - **Provide results to the customer to review for accuracy**" ,

a.Pymt_GE2_Description as "Vendor Payment State/Province", a.GE_Table2_Id as "Vendor Payment State/Prov ID" , b.Description as "GE Table 2 Description", b.Table2_Id

from AP_Vendor a, GE_Table2 b

Where a.Pymt_GE_Table2_Id =b.Table2_ID

and a.Pymt_GE2_Description <> b.Description

and a.Vendor_Id > @vendorid

Order by a.Vendor_Code

--Query 203: AP_Vendor - Payment GE2_Short is Invalid (State/Province) (NO ROWS SHOULD BE RETURNED) -

select distinct a.Vendor_Code as "Query 203 -AP_Vendor-Vendor_Code Where State/Province does not match - **Provide results to the customer to review for accuracy**" ,

a.Pymt_GE2_Short as "Vendor Payment State/Province", b.Short as "GE Table 2 Short Description"

from AP_Vendor a, GE_Table2 b

Where a.GE_Table2_id =b.Table2_ID

and a.Pymt_GE2_Short <> b.Short

and a.Vendor_Id > @vendorid

Order by a.Vendor_Code

--Query 204: AP_Vendor - Payment GE3_Description is Invalid (Zip/Postal Code) (NO ROWS SHOULD BE RETURNED) -

select distinct a.Vendor_Code as "Query 204 -AP_Vendor-Vendor_Code - Payment Zip/Postal Code- No match - **Provide results to the customer to review for accuracy**" ,

a.Pymt_GE3_Description as "Vendor Zip/Postal Code", a.GE_Table3_Id as "Vendor Pymt Table3 ID", b.Description as "GE Table 3 Description", b.Table3_ID as "GE Table 3 ID"

from AP_Vendor a, GE_Table3 b

Where a.GE_Table3_id =b.Table3_ID

and a.Pymt_GE3_Description <> b.Description

and a.Vendor_Id > @vendorid

Order by a.Vendor_Code

--Query 205: AP_Vendor - Currency Id is Valid (NO ROWS SHOULD BE RETURNED) --

select Vendor_Code as "Query 205 - AP_Vendor-Vendor_Code Currency Code -*Provide results to the customer to review for accuracy*" ,

b.Currency_Code, b.Description, a.Currency_Id

from AP_Vendor a, SY_Currency b

Where a.Currency_Id = b.Currency_Id

and a.Vendor_Id > @vendorid

Order by b.Currency_Code, a.Vendor_Code

--Query 206: AP_Vendor - Secure Vendor is Valid - All Should be Y or N

select a.Vendor_Code as "Query 206 - AP_Vendor - Where Secure Vendor < > Y or N - (NO ROWS SHOULD BE RETURNED)",

a.Secure_Vendor, b.Vendor_Type_Code

from AP_Vendor a, AP_Vendor_Type b

Where a.Secure_Vendor not in ('N', 'Y')

and a.Vendor_Id > @vendorid

and a.Vendor_Type_Id = b. Vendor_Type_Id

and b.Vendor_Type_Code < > 'Sales Tax Agency'

Order by a.Vendor_Code

--***AP_INVOICE FIELD VALIDATIONS***--

--Query 207: AP_Invoice - Checking for Duplicate Invoice Numbers per Vendor - (NO ROWS SHOULD BE RETURNED)--

select Vendor_Code as "Query 207 -AP_Vendor-Duplicate Invoice Numbers (NO ROWS SHOULD BE RETURNED)- *Provide results to the customer to review*",

b.Invoice_Number, Count(b.Invoice_Number) as "Number of Bills with Duplicate Invoice Number"

from AP_Vendor a, AP_Invoice b

where a.Vendor_Id = b.Vendor_Id

Group by a.vendor_code, b.invoice_Number

having count(b.invoice_number) >1

--Query 208: AP_Invoice - Count of Duplicate Invoice Numbers per Vendor - (NO ROWS SHOULD BE RETURNED)--

select Vendor_Code as "Query 208 -AP_Vendor-Duplicate Invoice Numbers (NO ROWS SHOULD BE RETURNED)-*Provide results to the customer to review*",

b.Invoice_Number, Count(b.Invoice_Number) as "Number of Bills with Blank Invoice Number"

from AP_Vendor a, AP_Invoice b

where a.Vendor_Id = b.Vendor_Id

and b.Invoice_Number = ''

Group by a.vendor_code, b.invoice_Number

having count(b.invoice_number) >1

--Query 209: AP_Invoice - Count of Vendor Bills by Term Code --

select b.Term_Code as "Query 209 - AP_Vendor_Invoice - Term Code - None Should be N/A - **Provide results to the customer to review for accuracy**",

count(c.Term_Id) as "Count of Vendor Bills by Term Code"

from AP_Vendor a, AR_Term b, AP_Invoice c

Where c.Term_Id = b.Term_Id

AND a.Vendor_Id = C.Vendor_Id

and a.Vendor_Id > @vendorid

Group by b.Term_Code, c.Term_Id

Order by b.Term_Code

--Query 210: AP_Invoice - Count of Vendor Bills by Category Code --

select b.Category_Code as "Query 210 - AP_Vendor_Invoice - Category Code - None Should be N/A - **Provide results to the customer to review for accuracy**",

count(c.Category_Id) as "Count of Vendor Bills by Category Code"

from AP_Vendor a, AR_Category b, AP_Invoice c

Where c.Category_Id = b.Category_Id

AND C.Vendor_Id = A.Vendor_Id

and a.Vendor_Id > @vendorid

Group by b.Category_Code, c.Category_Id

Order by b.Category_Code

--Query 211: AP_Invoice - Vendor_Id is valid and <> 1 --

select b.Invoice_Number as "Query 211 - AP_Vendor_Invoice - Vendor_Id is not Valid - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Vendor_Id =1

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 212: AP_Invoice - Amount <=0 --

select b.Invoice_Number as "Query 212 - AP_Invoice.Invoice_Number - Amount <= 0.00 - (NO ROWS SHOULD BE

RETURNED)",

a.Vendor_Code, b.Amount

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Amount <= 0.00

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 213: AP_Invoice - Verify Amount Paid = 0 --

select b.Invoice_Number as "Query 213 - AP_Invoice.Invoice_Number - Amount_Paid <> 0.00 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Amount_Paid

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Amount_Paid <> 0.00

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 214: AP_Invoice - Verify Net Due = Amount --

select b.Invoice_Number as "Query 214 - AP_Invoice.Invoice_Number - Net Due <> Amount - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Amount, b.Net_Due

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Amount <> b.Net_Due

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 215: AP_Invoice - Verify Status = O --

select b.Invoice_Number as "Query 215 - AP_Invoice.Invoice_Number - Status <> O - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Status

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Net_Due >0

and b.Status <> 'O'

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 216: AP_Invoice - Verify Hold_Payment = N --

select b.Invoice_Number as "Query 216 - AP_Invoice.Invoice_Number - Hold Payment <> N - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Hold_Payment

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Net_Due >0

and b.Hold_Payment <> 'N'

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 217: AP_Invoice - Verify Billed = Y --

select b.Invoice_Number as "Query 217 - AP_Invoice.Invoice_Number - Billed <> Y - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Hold_Payment

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Net_Due >0

and b.Billed <> 'Y'

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 218: AP_Invoice - Verify Received = Y --

select b.Invoice_Number as "Query 218 - AP_Invoice.Invoice_Number - Received <> Y - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Hold_Payment

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Net_Due >0

and b.Received <> 'Y'

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 219: AP_Invoice - Register_Id <> 1 --

select b.Invoice_Number as "Query 219 - AP_Invoice.Invoice_Number - Register Id = 1 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Register_Id

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Net_Due >0

and b.Register_Id = 1

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 220: AP_Invoice - Branch Id is Valid --

select distinct c.Branch_Code as "Query 220 - AP_Invoice - Branch Id is Valid - None Should be N/A -*Provide results to the customer to review*",

Count(c.Branch_Code) as "Number of Bills by Branch"

from AP_Vendor a, AP_Invoice b, AR_Branch c

where a.Vendor_Id = b.Vendor_Id

and b.Branch_Id = c.Branch_Id

and b.Invoice_Id >1

Group by c.Branch_Code

Order by c.Branch_Code

--Query 221: AP_Invoice - Verify Warehouse Id = Default Warehouse --

/*USE THIS TO FIND THE ID FOR THE CUSTOMER'S DEFAULT WAREHOUSE*/

SELECT * FROM IN_Warehouse WHERE Warehouse_Code = 'LIV' --CUSTOMER DEFAULT WAREHOUSE

select b.Invoice_Number as "Query 221 - AP_Invoice.Invoice_Number - Warehouse Id <> 2 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Warehouse_Id

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Warehouse_Id <> 2

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 222: AP_Invoice - Verify Job Id = 1 --

select b.Invoice_Number as "Query 222 - AP_Invoice.Invoice_Number - Job Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Job_Id

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Job_Id <> 1

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 223: AP_Invoice - Verify Service Ticket Id = 1 --

select b.Invoice_Number as "Query 223 - AP_Invoice.Invoice_Number - Service Ticket Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Service_Ticket_Id

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Service_Ticket_Id <> 1

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 224: AP_Invoice - Verify GST Rate = 0 --

select b.Invoice_Number as "Query 224 - AP_Invoice.Invoice_Number - GST Rate <> 0 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.GST_Rate

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.GST_Rate <> 0

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 225: AP_Invoice - Verify GST Amount = 0 --

select b.Invoice_Number as "Query 225 - AP_Invoice.Invoice_Number - GST Amount < > 0 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.GST_Amount

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.GST_Amount < > 0

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 226: AP_Invoice - Verify GST Register Id = 1--

select b.Invoice_Number as "Query 226 - AP_Invoice.Invoice_Number - GST Register Id < > 1 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.GST_Register_Id

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.GST_Register_Id < > 1

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 227: AP_Invoice - Verify Purchase Order Id = 1--

select b.Invoice_Number as "Query 227 - AP_Invoice.Invoice_Number - Purchase Order Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Purchase_Order_Id

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Purchase_Order_Id <> 1

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 228: AP_Invoice - Verify Repair Order Id = 1--

select b.Invoice_Number as "Query 228 - AP_Invoice.Invoice_Number - Repair Order Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Repair_Order_Id

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Repair_Order_Id <> 1

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 229: AP_Invoice - Verify Received Date = 1899-12-30 00:00:00:000 --

/* RETURNED DATES ARE OK IF IT WAS IN THE SOURCE DATE */

select b.Invoice_Number as "Query 229 - AP_Invoice.Invoice_Number - Received Date <> 1899-12-30 00:00:00:000 - (NO ROWS SHOULD BE RETURNED)", a.Vendor_Code, b.Received_Date

```
from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Received_Date <> '1899-12-30 00:00:00:000'

and b.Invoice_Id >1

Order by B.Received_Date, a.Vendor_Code, b.Invoice_Number
```

```
--Query 230:  AP_Invoice - Verify Received Register Id = 1--

select b.Invoice_Number as "Query 230 - AP_Invoice.Invoice_Number - Received Register Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Received_Register_Id

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Received_Register_Id <> 1

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number
```

```
--Query 231:  AP_Invoice - Verify Received By = Administrator --

    /* Received by may be set to a Conversion code */

select b.Invoice_Number as "Query 231 - AP_Invoice.Invoice_Number - Received By <> Administrator - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Received_By
```

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Received_By <> 'Administrator'

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 232: AP_Invoice - Verify Part Total = 0 --

select b.Invoice_Number as "Query 232 - AP_Invoice.Invoice_Number - Part Total <>0 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Part_Total

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Part_Total <> 0

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 233: AP_Invoice - Verify Expense Total <= 0 --

select distinct b.Invoice_Number as "Query 233 - AP_Invoice.Invoice_Number - Expense Total <>0 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Expense_Total

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Expense_Total <= 0

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 234: AP_Invoice - Verify Eligible for Discount = 0 --

select distinct b.Invoice_Number as "Query 234 - AP_Invoice.Invoice_Number - Eligible for Discount <>0 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Eligible_For_Discount

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Eligible_For_Discount <> 0

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 235: AP_Invoice - Currency Id is Valid --

select distinct c.Currency_Code as "Query 235 - AP_Invoice - Currency Id is Valid - None Should be N/A",

Count(c.Currency_Code) as "Number of Bills by Currency Code"

from AP_Vendor a, AP_Invoice b, SY_Currency c

where a.Vendor_Id = b.Vendor_Id

and b.Currency_Id = c.Currency_Id

and b.Invoice_Id >1

Group by c.Currency_Code

Order by c.Currency_Code

--Query 236: AP_Invoice - Verify Exchange Rate = 1 --

select distinct b.Invoice_Number as "Query 236 - AP_Invoice.Invoice_Number - Exchange Rate <> 1 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Exchange_Rate

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Exchange_Rate <> 1

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 237: AP_Invoice - Verify GST Branch Amount = 0 --

select distinct b.Invoice_Number as "Query 237 - AP_Invoice.Invoice_Number - GST Branch Amount <> 0 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.GST_Branch_Amount

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.GST_Branch_Amount <> 0

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 238: AP_Invoice - Verify Manual GST = N --

select distinct b.Invoice_Number as "Query 238 - AP_Invoice.Invoice_Number - Manual GST <> N - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Manual_GST

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Manual_GST <> 'N'

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 239: AP_Invoice - Verify Posting Date is Correct --

select distinct b.Posting_Date as "Query 239 - AP_Invoice - Verify Posting Date is Correct"

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

Order by b.Posting_Date

--Query 240: AP_Invoice - Verify AP_Invoice has a record in AP_Invoice_Expense --

select distinct b.Invoice_Number as "Query 240 - AP Invoice Number - Missing AP Invoice Expense Record - (NO ROWS SHOULD BE RETURNED)"

from AP_Vendor a, AP_Invoice b

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and b.Invoice_Id not in (select Invoice_Id from AP_Invoice_Expense)

Order by b.Invoice_Number

--***AP_INVOICE_EXPENSE FIELD VALIDATIONS***--

--Query 241: AP_Invoice Expense - Verify Account_id is Valid --

select distinct b.Invoice_Number as "Query 241 - AP Invoice Number - Validate Account Id - All Should be the Same",

a.Vendor_Code, d.Account_Code as "GL Account Code - should start with a 3", d.Description as "GL Account Description"

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c, GL_Account d

Where b.Vendor_Id = a.Vendor_Id

and c.Invoice_Id = b.Invoice_Id

and c.Invoice_Id >1

and c.Account_Id <>1

and c.Account_Id=d.Account_Id

Order by b.Invoice_Number

--Query 242: AP_Invoice_Expense - Verify Amount > 0 --

select b.Invoice_Number as "Query 242 - AP_Invoice.Invoice_Number - Amount <=0 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, c.Amount

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and c.Invoice_Id = b.Invoice_Id

and c.Amount <= 0

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 243: AP_Invoice_Expense - Verify Amount = Amount in AP_Invoice --

select distinct b.Invoice_Number as "Query 243 - AP_Invoice.Invoice_Number - < > to Amount in AP_Invoice - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, c.Amount as "AP Invoice Expense Amount", b.Amount as "AP Invoice Amount"

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and c.Invoice_Id = b.Invoice_Id

and c.Amount < > b.Amount

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 244: AP_Invoice_Expense - Verify Description is not Blank --

select distinct b.Invoice_Number as "Query 244 - AP_Invoice_Expense - Description is Blank - (NO ROWS SHOULD BE RETURNED) ",

c.Description

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.Description = ''

Order by b.Invoice_Number

--Query 245: AP_Invoice_Expense - Verify Customer_Id = 1 --

select distinct b.Invoice_Number as "Query 245 - AP_Invoice_Expense - Customer_Id <>1 - (NO ROWS SHOULD BE RETURNED) ",

c.Customer_Id

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.Customer_Id <>1

Order by b.Invoice_Number

--Query 246: AP_Invoice_Expense - Verify Job_Id = 1 --

select distinct b.Invoice_Number as "Query 246 - AP_Invoice_Expense - Job_Id <>1 - (NO ROWS SHOULD BE RETURNED) ",

c.Job_Id

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.Job_Id <>1

Order by b.Invoice_Number

--Query 247: AP_Invoice_Expense - Verify Service_Id = 1 --

select distinct b.Invoice_Number as "Query 247 - AP_Invoice_Expense - Service_Id <>1 - (NO ROWS SHOULD BE RETURNED) ",

c.Service_Id

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.Service_Id <>1

Order by b.Invoice_Number

--Query 248: AP_Invoice_Expense - Verify Register_Id <> 1 --

select distinct b.Invoice_Number as "Query 248 - AP_Invoice_Expense - Register_Id =1 - (NO ROWS SHOULD BE RETURNED) ",

c.Register_Id

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.Register_Id =1

Order by b.Invoice_Number

--Query 249: AP_Invoice_Expense - Validate Category Id --

select b.Category_Code as "Query 249 - AP_Vendor_Invoice - Category Code - None Should be N/A - **Provide results to the customer to review for accuracy**",

count(d.Category_Id) as "Count of Vendor Invoice Expense Records by Category Code"

from AP_Vendor a, AR_Category b, AP_Invoice c, AP_Invoice_Expense d

Where d.Category_Id = b.Category_Id

and a.Vendor_Id > @vendorid

and c.Vendor_Id = a.Vendor_Id

and d.Invoice_Id = c.Invoice_Id

Group by b.Category_Code, d.Category_Id

Order by b.Category_Code

--Query 250: AP_Invoice_Expense - Verify Category Id = Category Id in AP_Invoice --

select distinct b.Invoice_Number as "Query 250 - AP_Invoice_Expense.Category_id - < > to Category Id in AP_Invoice - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, d.Category_Code, c.Category_Id as "AP Invoice Expense Category Id", b.Category_Id as "AP Invoice Category Id"

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c, AR_Category d

Where b.Vendor_Id = a.Vendor_Id

and c.Invoice_Id = b.Invoice_Id

and c.Category_Id < > b.Category_Id

and b.Category_Id = d.Category_Id

and c.Category_Id = d.Category_Id

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 251: AP_Invoice_Expense - Validate Branch Id --

select b.Branch_Code as "Query 251 - AP_Vendor_Invoice - Branch Code - None Should be N/A - **Provide results to the customer to review for accuracy**",

count(d.Branch_Id) as "Count of Vendor Invoice Expense Records by Branch Code"

from AP_Vendor a, AR_Branch b, AP_Invoice c, AP_Invoice_Expense d

Where d.Branch_Id = b.Branch_Id

and a.Vendor_Id > @vendorid

and c.Vendor_Id = a.Vendor_Id

and d.Invoice_Id = c.Invoice_Id

Group by b.Branch_Code, d.Branch_Id

Order by b.Branch_Code

--Query 252: AP_Invoice_Expense - Verify Branch Id = Branch Id in AP_Invoice --

select distinct b.Invoice_Number as "Query 252 - AP_Invoice_Expense.Branch_id - < > to Branch Id in AP_Invoice - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, d.Branch_Code, c.Branch_Id as "AP Invoice Expense Branch Id", b.Branch_Id as "AP Invoice Branch Id"

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c, AR_Branch d

Where b.Vendor_Id = a.Vendor_Id

and c.Invoice_Id = b.Invoice_Id

and c.Branch_Id <> b.Branch_Id

and b.Branch_Id = d.Branch_Id

and c.Branch_Id = d.Branch_Id

and b.Invoice_Id >1

Order by a.Vendor_Code, b.Invoice_Number

--Query 253: AP_Invoice_Expense - Verify Inter_Branch_Id = 1 --

select distinct b.Invoice_Number as "Query 253 - AP_Invoice_Expense - Inter_Branch_Id <>1 - (NO ROWS SHOULD BE RETURNED) ",

c.Inter_Branch_Id

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.Inter_Branch_Id <>1

Order by b.Invoice_Number

--Query 254: AP_Invoice_Expense - Verify PPV_Register_Id = 1 --

select distinct b.Invoice_Number as "Query 254 - AP_Invoice_Expense - PPV_Register_Id <>1 - (NO ROWS SHOULD BE RETURNED) ", c.PPV_Register_Id

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.PPV_Register_Id <>1

Order by b.Invoice_Number

--Query 255: AP_Invoice_Expense - Verify Pass_Thru_Item_Id = 1 --

select distinct b.Invoice_Number as "Query 255 - AP_Invoice_Expense - Pass_Thru_Item_Id <>1 - (NO ROWS SHOULD BE RETURNED) ",

c.Pass_Thru_Item_Id

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.Pass_Thru_Item_Id <>1

Order by b.Invoice_Number

--Query 256: AP_Invoice_Expense - Verify Job_Install_Id = 1 --

select distinct b.Invoice_Number as "Query 256 - AP_Invoice_Expense - Job_Install_Id <>1 - (NO ROWS SHOULD BE RETURNED) ",

c.Job_Install_Id

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.Job_Install_Id <>1

Order by b.Invoice_Number

--Query 257: AP_Invoice_Expense - Verify GST_Register_Id = 1 --

select distinct b.Invoice_Number as "Query 257 - AP_Invoice_Expense - GST_Register_Id <>1 - (NO ROWS SHOULD BE RETURNED) ",

c.GST_Register_Id

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.GST_Register_Id <>1

Order by b.Invoice_Number

--Query 258: AP_Invoice_Expense - Verify GST_Inter_Branch_Id = 1 --

select distinct b.Invoice_Number as "Query 258 - AP_Invoice_Expense - GST_Inter_Branch_Id <>1 - (NO ROWS SHOULD BE RETURNED) ",

c.GST_Inter_Branch_Id

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.GST_Inter_Branch_Id <>1

Order by b.Invoice_Number

--Query 259: AP_Invoice_Expense - Verify Charge_GST = N --

select distinct b.Invoice_Number as "Query 259 - AP_Invoice_Expense - Charge_GST <> N - (NO ROWS SHOULD BE RETURNED) ",

c.Charge_GST

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.Charge_GST <> 'N'

Order by b.Invoice_Number

--Query 260: AP_Invoice_Expense - Verify Pass_Thru_Charge = N --

select distinct b.Invoice_Number as "Query 260 - AP_Invoice_Expense - Pass_Thru_Charge <> N - (NO ROWS SHOULD BE RETURNED) ",

c.Pass_Thru_Charge

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.Pass_Thru_Charge <> 'N'

Order by b.Invoice_Number

--Query 261: AP_Invoice_Expense - Verify Quantity = 1.00 --

select distinct b.Invoice_Number as "Query 261 - AP_Invoice_Expense - Quantity <> 1.00 - (NO ROWS SHOULD BE RETURNED) ",

a.Vendor_Code, c.Quantity

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.Quantity <> 1.00

Order by b.Invoice_Number

--Query 262: AP_Invoice_Expense - Verify GST_Rate = 0 --

select distinct b.Invoice_Number as "Query 262 - AP_Invoice_Expense - GST_Rate <> 0 - (NO ROWS SHOULD BE RETURNED) ",

a.Vendor_Code, c.GST_Rate

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.GST_Rate <> 0

Order by b.Invoice_Number

--Query 263: AP_Invoice_Expense - Verify GST_Amount = 0 --

select distinct b.Invoice_Number as "Query 263 - AP_Invoice_Expense - GST_Amount <> 0 - (NO ROWS SHOULD BE RETURNED) ",

a.Vendor_Code, c.GST_Amount

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.GST_Amount <> 0

Order by b.Invoice_Number

--Query 264: AP_Invoice_Expense - Verify Rate = Amount and <> 0 --

select distinct b.Invoice_Number as "Query 264 - AP_Invoice_Expense - Rate <> Amount and <> 0 - (NO ROWS SHOULD BE RETURNED) ",

a.Vendor_Code, c.Rate, c.Amount

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.Rate < > c.Amount

and c.Rate <=0

Order by b.Invoice_Number

--Query 265: AP_Invoice_Expense - Verify Rate < > 0 --

select distinct b.Invoice_Number as "Query 265 - AP_Invoice_Expense - Rate <= 0 - (NO ROWS SHOULD BE RETURNED) ",

a.Vendor_Code, c.Rate

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.Rate <=0

Order by b.Invoice_Number

--Query 266: AP_Invoice_Expense - Validate Expense_Type_id --

select distinct b.Description as "Query 266 - AP_Vendor_Invoice - Expense Type - None Should be N/A -*Provide results to the customer to review*",

count(b.Expense_Code) as "Count of Vendor Invoice Expense Records by Expense Type Code"

from AP_Vendor a, GL_Expense_Type b, AP_Invoice c, AP_Invoice_Expense d

Where d.Branch_Id = b.Expense_Type_Id

and a.Vendor_Id > @vendorid

and c.Vendor_Id = a.Vendor_Id

and d.Invoice_Id = c.Invoice_Id

Group by b.Expense_Code, b.Description

--Order by b.Expense_Code

--Query 267: AP_Invoice_Expense - Verify Receipt_Expense_Id = 0 --

select distinct b.Invoice_Number as "Query 267 - AP_Invoice_Expense - Receipt_Expense_Id <> 0 - (NO ROWS SHOULD BE RETURNED) ",

a.Vendor_Code, c.Receipt_Expense_Id

from AP_Vendor a, AP_Invoice b, AP_Invoice_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Invoice_Id >1

and c.Invoice_Id = b.Invoice_Id

and c.Receipt_Expense_Id <> 0

Order by b.Invoice_Number

--***AP_CREDIT FIELD VALIDATIONS***--

--Query 268: AP_Credit - Checking for Duplicate Credit Numbers per Vendor - (NO ROWS SHOULD BE RETURNED)--

select Vendor_Code as "Query 268 -AP_Vendor-Duplicate Credit Numbers (NO ROWS SHOULD BE RETURNED)-*Provide results to the customer to review*",

b.Credit_Number, Count(b.Credit_Number) as "Number of Credits with Duplicate Credit Number"

from AP_Vendor a, AP_Credit b

```
where a.Vendor_Id = b.Vendor_Id

Group by a.vendor_code, b.Credit_Number

having count(b.Credit_Number) >1
```

```
--Query 269  AP_Credit - Count of Blank Credit Numbers per Vendor - (NO ROWS SHOULD BE RETURNED)--

select Vendor_Code as "Query 269 -AP_Vendor-Blank Credit Numbers (NO ROWS SHOULD BE RETURNED)-*Provide results to the customer to review*",

b.Credit_Number, Count(b.Credit_Number) as "Number of Credits with Blank Credit Number"

from AP_Vendor a, AP_Credit b

where a.Vendor_Id = b.Vendor_Id

and b.Credit_Number = ''

Group by a.vendor_code, b.Credit_Number

having count(b.Credit_Number) >1
```

```
--Query 270:  AP_Credit - Count of Vendor Credits by Category Code --

select b.Category_Code as "Query 270 - AP_Credit - Category Code - None Should be N/A - **Provide results to the customer to review for accuracy**",

count(c.Category_Id) as "Count of Vendor Credits by Category Code"

from AP_Vendor a, AR_Category b, AP_Credit c

Where c.Category_Id = b.Category_Id

and a.Vendor_Id = c.Vendor_Id

and a.Vendor_Id > @vendorid

Group by b.Category_Code, c.Category_Id
```

Order by b.Category_Code

--Query 271: AP_Credit - Vendor_Id is valid and <> 1 --

select b.Credit_Number as "Query 271 - AP_Credit - Vendor_Id - Invalid Vendor_Id - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.Vendor_Id =1

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 272: AP_Credit - Amount <=0 --

select b.Credit_Number as "Query 272 - AP_Credit.Credit_Number - Amount <= 0.00 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Amount

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.Amount <= 0.00

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 273: AP_Credit - Verify Amount Used = 0 --

select b.Credit_Number as "Query 273 - AP_Credit.Credit_Number - Amount Used <> 0.00 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Amount_Used

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.Amount_Used <> 0.00

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 274: AP_Credit - Verify Balance = Amount --

select b.Credit_Number as "Query 274 - AP_Credit.Credit_Number - Balance <> Amount - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Amount, b.Balance

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.Amount <> b.Balance

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 275: AP_Credit - Register_Id <> 1 --

select b.Credit_Number as "Query 275 - AP_Credit.Credit_Number - Register Id = 1 - (NO ROWS SHOULD BE

RETURNED)",

a.Vendor_Code, b.Register_Id

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.Balance >0

and b.Register_Id = 1

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 276: AP_Credit - Branch Id is Valid --

select distinct c.Branch_Code as "Query 276 - AP_Credit - Branch Id is Valid - None Should be N/A -*Provide results to the customer to review*",

Count(c.Branch_Code) as "Number of Credits by Branch"

from AP_Vendor a, AP_Credit b, AR_Branch c

where a.Vendor_Id = b.Vendor_Id

and b.Branch_Id = c.Branch_Id

and b.Credit_id >1

Group by c.Branch_Code

Order by c.Branch_Code

--Query 277: AP_Credit - Verify Warehouse Id = 1 --

select b.Credit_Number as "Query 277 - AP_Credit.Credit_Number - Warehouse Id <> 1 - (NO ROWS SHOULD BE RETURNED)", a.Vendor_Code, b.Warehouse_Id

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.Warehouse_Id <> 1

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 278: AP_Credit - Verify Job Id = 1 --

select b.Credit_Number as "Query 278 - AP_Credit.Credit_Number - Job Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Job_Id

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.Job_Id <> 1

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 279: AP_Credit - Verify Service Ticket Id = 1 --

select b.Credit_Number as "Query 279 - AP_Credit.Credit_Number - Service Ticket Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Service_Ticket_Id

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.Service_Ticket_Id <> 1

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 280: AP_Credit - Verify GST Rate = 0 --

select b.Credit_Number as "Query 280 - AP_Credit.Credit_Number - GST Rate <> 0 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.GST_Rate

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.GST_Rate <> 0

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 281: AP_Credit - Verify GST Amount = 0 --

select b.Credit_Number as "Query 281 - AP_Credit.Credit_Number - GST Amount <> 0 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.GST_Amount

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.GST_Amount <> 0

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 282: AP_Credit - Verify GST Register Id = 1--

select b.Credit_Number as "Query 282 - AP_Credit.Credit_Number - GST Register Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.GST_Register_Id

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.GST_Register_Id <> 1

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 283 AP_Credit - Verify Purchase Order Id = 1--

select b.Credit_Number as "Query 283 - AP_Credit.Credit_Number - Purchase Order Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Purchase_Order_Id

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.Purchase_Order_Id <> 1

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 284: AP_Credit - Verify Repair Order Id = 1--

select b.Credit_Number as "Query 284 - AP_Credit.Credit_Number - Repair Order Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Repair_Order_Id

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.Repair_Order_Id <> 1

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 285: AP_Credit - Verify Part Total = 0 --

select b.Credit_Number as "Query 285 - AP_Credit.Credit_Number - Part Total <>0 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Part_Total

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.Part_Total <> 0

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 286: AP_Credit - Verify Expense Total = 0 --

select distinct b.Credit_Number as "Query 286 - AP_Credit.Credit_Number - Expense Total <>0 - (NO ROWS SHOULD BE RETURNED)",

```
a.Vendor_Code, b.Expense_Total

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.Expense_Total <> 0

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number
```

```
--Query 287:  AP_Credit - Currency Id is Valid --

select distinct c.Currency_Code as "Query 287 - AP_Credit - Currency Id is Valid - None Should be N/A",

Count(c.Currency_Code) as "Number of Credits by Currency Code"

from AP_Vendor a, AP_Credit b, SY_Currency c

where a.Vendor_Id = b.Vendor_Id

and b.Currency_Id = c.Currency_Id

and b.Credit_id >1

Group by c.Currency_Code

Order by c.Currency_Code
```

```
--Query 288:  AP_Credit - Verify Exchange Rate = 1 --

select distinct b.Credit_Number as "Query 288 - AP_Credit.Credit_Number - Exchange Rate <> 1 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Exchange_Rate

from AP_Vendor a, AP_Credit b
```

Where b.Vendor_Id = a.Vendor_Id

and b.Exchange_Rate <> 1

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 289: AP_Credit - Verify GST Branch Amount = 0 --

select distinct b.Credit_Number as "Query 289 - AP_Credit.Credit_Number - GST Branch Amount <> 0 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.GST_Branch_Amount

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.GST_Branch_Amount <> 0

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 290: AP_Credit - Verify Manual GST = N --

select distinct b.Credit_Number as "Query 290 - AP_Credit.Credit_Number - Manual GST <> N - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, b.Manual_GST

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.Manual_GST <> 'N'

and b.Credit_id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 291: AP_Credit - Verify Posting Date is Correct --

select distinct b.Posting_Date as "Query 291 - AP_Credit - VERIFY POSTING DATE IS CORRECT"

from AP_Vendor a, AP_Credit b

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_id >1

Order by b.Posting_Date

-- ** AP_CREDIT_EXPENSE FIELD VALIDATIONS ** --

--Query 292: AP_Credit Expense - Verify Account_id is Valid --

select distinct b.Credit_Number as "Query 292 - AP_Credit_Expense - AP Credit Number - Validate Account Id - All Should be the Same",

a.Vendor_Code, d.Account_Code as "GL Account Code - should start with a 3", d.Description as "GL Account Description"

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c, GL_Account d

Where b.Vendor_Id = a.Vendor_Id

and c.Credit_Id = b.Credit_Id

and c.Credit_Id >1

and c.Account_Id <>1

and c.Account_Id=d.Account_Id

Order by b.Credit_Number

--Query 293: AP_Credit_Expense - Verify Amount > 0 --

select b.Credit_Number as "Query 293 - AP_Credit_Expense - AP_Credit.Credit_Number - Amount <=0 - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, c.Amount

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and c.Credit_Id = b.Credit_Id

and c.Amount <= 0

and b.Credit_Id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 294: AP_Credit_Expense - Verify Amount = Amount in AP_Credit --

select distinct b.Credit_Number as "Query 294 - AP_Credit_Expense - AP_Credit.Credit_Number - <> to Amount in AP_Credit - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, c.Amount as "AP Credit Expense Amount", b.Amount as "AP Credit Amount"

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and c.Credit_Id = b.Credit_Id

and c.Amount <> b.Amount

and b.Credit_Id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 295: AP_Credit_Expense - Verify Description is not Blank --

select distinct b.Credit_Number as "Query 295 - AP_Credit_Expense - Description is Blank - (NO ROWS SHOULD BE RETURNED) ",

c.Description

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.Description = ''

Order by b.Credit_Number

--Query 296: AP_Credit_Expense - Verify Customer_Id = 1 --

select distinct b.Credit_Number as "Query 296 - AP_Credit_Expense - Customer_Id <>1 - (NO ROWS SHOULD BE RETURNED) ",

c.Customer_Id

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.Customer_Id <>1

Order by b.Credit_Number

```sql
--Query 297:  AP_Credit_Expense - Verify Job_Id = 1 --

select distinct b.Credit_Number as "Query 297 - AP_Credit_Expense - Job_Id <>1 - (NO ROWS SHOULD BE RETURNED) ",

c.Job_Id

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.Job_Id <>1

Order by b.Credit_Number
```

```sql
--Query 298:  AP_Credit_Expense - Verify Service_Ticket_Id = 1 --

select distinct b.Credit_Number as "Query 298 - AP_Credit_Expense - Service_Id <>1 - (NO ROWS SHOULD BE RETURNED) ",

c.Service_Ticket_Id

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.Service_Ticket_Id <>1

Order by b.Credit_Number
```

```sql
--Query 299:  AP_Credit_Expense - Verify Register_Id <> 1 --

select distinct b.Credit_Number as "Query 299 - AP_Credit_Expense - Register_Id =1 - (NO ROWS SHOULD BE RETURNED) ",

c.Register_Id

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.Register_Id =1

Order by b.Credit_Number
```

```sql
--Query 300:  AP_Credit_Expense - Validate Category Id --

select b.Category_Code as "Query 300 - AP_Credit_Expense - Category Code - None Should be N/A - **Provide results to the customer to review for accuracy**",

count(d.Category_Id) as "Count of Vendor Credit Expense Records by Category Code"

from AP_Vendor a, AR_Category b, AP_Credit c, AP_Credit_Expense d

Where d.Category_Id = b.Category_Id

and a.Vendor_Id > @vendorid

and c.Vendor_Id = a.Vendor_Id

and d.Credit_Id = c.Credit_Id

Group by b.Category_Code, d.Category_Id

Order by b.Category_Code
```

--Query 301: AP_Credit_Expense - Verify Category Id = Category Id in AP_Credit --

select distinct b.Credit_Number as "Query 301 - AP_Credit_Expense.Category_id - < > to Category Id in AP_Credit - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, d.Category_Code, c.Category_Id as "AP Credit Expense Category Id", b.Category_Id as "AP Credit Category Id"

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c, AR_Category d

Where b.Vendor_Id = a.Vendor_Id

and c.Credit_Id = b.Credit_Id

and c.Category_Id < > b.Category_Id

and b.Category_Id = d.Category_Id

and c.Category_Id = d.Category_Id

and b.Credit_Id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 302: AP_Credit_Expense - Validate Branch Id --

select b.Branch_Code as "Query 302 - AP_Credit_Expense - Branch Code - None Should be N/A - **Provide results to the customer to review for accuracy**",

count(d.Branch_Id) as "Count of Vendor Credit Expense Records by Branch Code"

from AP_Vendor a, AR_Branch b, AP_Credit c, AP_Credit_Expense d

Where d.Branch_Id = b.Branch_Id

and a.Vendor_Id > @vendorid

and c.Vendor_Id = a.Vendor_Id

and d.Credit_Id = c.Credit_Id

Group by b.Branch_Code, d.Branch_Id

Order by b.Branch_Code

--Query 303: AP_Credit_Expense - Verify Branch Id = Branch Id in AP_Credit --

select distinct b.Credit_Number as "Query 303 - AP_Credit_Expense.Branch_id - <> to Branch Id in AP_Credit - (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Code, d.Branch_Code, c.Branch_Id as "AP Credit Expense Branch Id", b.Branch_Id as "AP Credit Branch Id"

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c, AR_Branch d

Where b.Vendor_Id = a.Vendor_Id

and c.Credit_Id = b.Credit_Id

and c.Branch_Id <> b.Branch_Id

and b.Branch_Id = d.Branch_Id

and c.Branch_Id = d.Branch_Id

and b.Credit_Id >1

Order by a.Vendor_Code, b.Credit_Number

--Query 304: AP_Credit_Expense - Verify Inter_Branch_Id = 1 --

select distinct b.Credit_Number as "Query 304 - AP_Credit_Expense - Inter_Branch_Id <>1 - (NO ROWS SHOULD BE RETURNED) ",

c.Inter_Branch_Id

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.Inter_Branch_Id <>1

Order by b.Credit_Number

--Query 305: AP_Credit_Expense - Verify Pass_Thru_Item_Id = 1 --

select distinct b.Credit_Number as "Query 305 - AP_Credit_Expense - Pass_Thru_Item_Id <>1 - (NO ROWS SHOULD BE RETURNED) ",

c.Pass_Thru_Item_Id

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.Pass_Thru_Item_Id <>1

Order by b.Credit_Number

--Query 306: AP_Credit_Expense - Verify Job_Install_Id = 1 --

select distinct b.Credit_Number as "Query 306 - AP_Credit_Expense - Job_Install_Id <>1 - (NO ROWS SHOULD BE RETURNED) ",

a.Vendor_Code, c.Job_Install_Id

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.Job_Install_Id <>1

Order by b.Credit_Number

--Query 307: AP_Credit_Expense - Verify GST_Register_Id = 1 --

select distinct b.Credit_Number as "Query 307 - AP_Credit_Expense - GST_Register_Id <>1 - (NO ROWS SHOULD BE RETURNED) ",

c.GST_Register_Id

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.GST_Register_Id <>1

Order by b.Credit_Number

--Query 308: AP_Credit_Expense - Verify GST_Inter_Branch_Id = 1 --

select distinct b.Credit_Number as "Query 308 - AP_Credit_Expense - GST_Inter_Branch_Id <>1 - (NO ROWS SHOULD BE RETURNED) ",

c.GST_Inter_Branch_Id

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.GST_Inter_Branch_Id <>1

Order by b.Credit_Number

--Query 309: AP_Credit_Expense - Verify Charge_GST = N --

select distinct b.Credit_Number as "Query 309 - AP_Credit_Expense - Charge_GST <> N - (NO ROWS SHOULD BE RETURNED) ",

c.Charge_GST

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.Charge_GST <> 'N'

Order by b.Credit_Number

--Query 310: AP_Credit_Expense - Verify Pass_Thru_Charge = N --

select distinct b.Credit_Number as "Query 310 - AP_Credit_Expense - Pass_Thru_Charge <> N - (NO ROWS SHOULD BE RETURNED) ",

c.Pass_Thru_Charge

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.Pass_Thru_Charge <> 'N'

Order by b.Credit_Number

--Query 311: AP_Credit_Expense - Verify Quantity = 1.00 --

select distinct b.Credit_Number as "Query 311 - AP_Credit_Expense - Quantity <> 1.00 - (NO ROWS SHOULD BE RETURNED) ",

a.Vendor_Code, c.Quantity

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.Quantity <> 1.00

Order by b.Credit_Number

--Query 312: AP_Credit_Expense - Verify GST_Rate = 0 --

select distinct b.Credit_Number as "Query 312 - AP_Credit_Expense - GST_Rate <> 0 - (NO ROWS SHOULD BE RETURNED) ",

a.Vendor_Code, c.GST_Rate

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.GST_Rate <> 0

Order by b.Credit_Number

--Query 313: AP_Credit_Expense - Verify GST_Amount = 0 --

select distinct b.Credit_Number as "Query 313 - AP_Credit_Expense - GST_Amount < > 0 - (NO ROWS SHOULD BE RETURNED) ",

a.Vendor_Code, c.GST_Amount

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.GST_Amount < > 0

Order by b.Credit_Number

--Query 314: AP_Credit_Expense - Verify Rate = Amount and < > 0 --

select distinct b.Credit_Number as "Query 314 - AP_Credit_Expense - Rate < > Amount and < > 0 - (NO ROWS SHOULD BE RETURNED) ",

a.Vendor_Code, c.Rate, c.Amount

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.Rate < > c.Amount

and c.Rate <=0

Order by b.Credit_Number

```sql
--Query 315:  AP_Credit_Expense - Verify Rate  <> 0 --

select distinct b.Credit_Number as "Query 315 - AP_Credit_Expense - Rate <> 0 - (NO ROWS SHOULD BE RETURNED) ",

a.Vendor_Code, c.Rate

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.Rate <=0

Order by b.Credit_Number
```

```sql
--Query 316:  AP_Credit_Expense - Validate Expense_Type_id --

select distinct b.Expense_Code as "Query 316 - AP_Credit_Expense - Expense Type - None Should be N/A - All Should be O",

count(b.Expense_Code) as "Count of Vendor Credit Expense Records by Expense Type Code"

from AP_Vendor a, GL_Expense_Type b, AP_Credit c, AP_Credit_Expense d

Where d.Branch_Id = b.Expense_Type_Id

and a.Vendor_Id > @vendorid

and c.Vendor_Id = a.Vendor_Id

and d.Credit_Id = c.Credit_Id

Group by b.Expense_Code

Order by b.Expense_Code
```

--Query 317: AP_Credit_Expense - Verify Receipt_Expense_Id = 0 --

select distinct b.Credit_Number as "Query 317 - AP_Credit_Expense - Receipt_Expense_Id <> 0 - (NO ROWS SHOULD BE RETURNED) ",

a.Vendor_Code, c.Receipt_Expense_Id

from AP_Vendor a, AP_Credit b, AP_Credit_Expense c

Where b.Vendor_Id = a.Vendor_Id

and b.Credit_Id >1

and c.Credit_Id = b.Credit_Id

and c.Receipt_Expense_Id <> 0

Order by b.Credit_Number

--***SERVICE TICKET FIELD VALIDATIONS***--

--Query 325: SV_Service_Ticket OPEN Tickets - Verify Ticket Status = OP --

select distinct d.Ticket_Status as "Query 325 - OPEN TICKETS - SV_Service_Ticket.Ticket_Status - All Should be OP",

count(d.ticket_status) as "Count of Tickets by Status"

from AR_customer a, AR_customer_site b, ar_customer_system c, sv_service_ticket d

Where d.Customer_Id = a.Customer_Id

and d.Customer_Site_Id = b.Customer_Site_Id

and d.Customer_System_Id = c.Customer_System_Id

and d.ClosedDate = '1899-12-30 00:00:00:000'

and d.Ticket_Status <> 'CL'

Group by d.Ticket_Status

Order by d.Ticket_Status

--Query 326: SV_Service_Ticket CLOSED Tickets - Verify Ticket Status = CL --

select distinct d.Ticket_Status as "Query 326 - CLOSED TICKETS - SV_Service_Ticket.Ticket_Status - All Should be CL",

count(d.ticket_status) as "Count of Tickets by Status"

from AR_customer a, AR_customer_site b, ar_customer_system c, sv_service_ticket d

Where d.Customer_Id = a.Customer_Id

and d.Customer_Site_Id = b.Customer_Site_Id

and d.Customer_System_Id = c.Customer_System_Id

and d.ClosedDate = '1899-12-30 00:00:00:000'

and d.Ticket_Status = 'CL'

Group by d.Ticket_Status

Order by d.Ticket_Status

--Query 327: SV_Service_Ticket - Count of Duplicate Ticket Numbers - (NO ROWS SHOULD BE RETURNED)--

select distinct d.Ticket_Number as "Query 327 - SV_Service_Ticket-Duplicate Ticket Numbers (NO ROWS SHOULD BE RETURNED)",

Count(d.Ticket_Number) as "Number of Tickets with Same Ticket Number", d.Ticket_Status

from AR_customer a, AR_customer_site b, ar_customer_system c, sv_service_ticket d

where d.Customer_Id = a.Customer_Id

and d.Customer_Site_Id = b.Customer_Site_Id

and d.Customer_System_Id = c.Customer_System_Id

Group by d.Ticket_Number, d.Ticket_Status

having count(d.ticket_number) >1

--Query 328: SV_Service_Ticket - Ticket Customer Id is Valid - < > 1 - (NO ROWS SHOULD BE RETURNED)--

select distinct d.Ticket_Number as "Query 328 - SV_Service_Ticket-Customer Id is Invalid (NO ROWS SHOULD BE RETURNED)",

a.Customer_Number

from AR_customer a, sv_service_ticket d

where d.Customer_Id not in (select Customer_Id from AR_Customer)

and d.Customer_Id < >1

Order by d.Ticket_Number, a.Customer_Number

--Query 329: SV_Service_Ticket - Ticket Customer Site Id is Valid - < > 1 - (NO ROWS SHOULD BE RETURNED)--

select distinct d.Ticket_Number as "Query 329 - SV_Service_Ticket-Customer Site Id is Invalid (NO ROWS SHOULD BE RETURNED)",

a.Customer_Number, b.Customer_Site_Id, b.Site_Number

from AR_customer a, AR_customer_site b, sv_service_ticket d

where d.Customer_Site_Id not in (select Customer_Site_Id from AR_Customer_Site)

and d.Customer_Id > @customerid

and a.Customer_Id > @customerid

and b.Customer_Site_Id < > 1

and d.Customer_Site_Id < > 1

and d.Customer_Id = a.Customer_Id

and b.Customer_Id = a.Customer_Id

and d.Customer_Site_Id = b.Customer_Site_Id

--Query 330: SV_Service_Ticket - Ticket Customer System Id is Valid - <> 1 - (NO ROWS SHOULD BE RETURNED)--

select distinct d.Ticket_Number as "Query 330 - SV_Service_Ticket-Customer System Id is Invalid (NO ROWS SHOULD BE RETURNED)",

a.Customer_Number, b.Customer_Site_Id, b.Site_Number, c.Alarm_Account

from AR_customer a, AR_customer_site b, AR_Customer_System c, sv_service_ticket d

where d.Customer_System_Id not in (select Customer_System_Id from AR_Customer_System)

and d.Customer_Id > @customerid

and a.Customer_Id > @customerid

and c.Customer_Id > @customerid

and b.Customer_Site_Id <> 1

and d.Customer_Site_Id <> 1

and c.Customer_System_Id <> 1

and d.Customer_System_Id <> 1

and d.Customer_Id = a.Customer_Id

and b.Customer_Id = a.Customer_Id

and d.Customer_Site_Id = b.Customer_Site_Id

--Query 331: SV_Service_Ticket - Verify Ticket Creation Date > 01-02-1900 (NO ROWS SHOULD BE RETURNED)--

```
select distinct d.Ticket_Number as "Query 331 - Ticket Creation Date < 01-02-1900 (NO ROWS SHOULD BE RETURNED)",

d.Creation_Date

from AR_customer a, AR_customer_site b, AR_Customer_System c, sv_service_ticket d

where d.Customer_System_Id not in (select Customer_System_Id from AR_Customer_System)

and d.Customer_Id > @customerid

and a.Customer_Id > @customerid

and c.Customer_Id > @customerid

and b.Customer_Site_Id <> 1

and d.Customer_Site_Id <> 1

and c.Customer_Site_Id <> 1

and c.Customer_System_Id <> 1

and d.Customer_System_Id <> 1

and d.Customer_Id = a.Customer_Id

and b.Customer_Id = a.Customer_Id

and d.Customer_Site_Id = b.Customer_Site_Id

and d.Creation_Date < '01-02-1900'



--Query 332: SV_Service_Ticket - Ticket Problem Id is Invalid = 1 - (NO ROWS SHOULD BE RETURNED)--

select distinct d.Ticket_Number as "Query 332 - Ticket Number - Problem Id = 1 (NO ROWS SHOULD BE RETURNED)",

e.Problem_Code

from sv_service_ticket d, SV_Problem e

where d.Service_Ticket_Id <> 1

and d.Problem_Id = e.Problem_Id
```

and d.Problem_Id = 1

--Query 333: SV_Service_Ticket - Tickets Where Scheduled_For < 01-01-1900 - (NO ROWS SHOULD BE RETURNED)--

select distinct Ticket_Number as "Query 333 - Ticket Number - Scheduled_For <> 1899... (NO ROWS SHOULD BE RETURNED)",

Scheduled_For

from sv_service_ticket

where Service_Ticket_Id <>1

and Scheduled_For > '01-01-1900'

--Query 334: SV_Service_Ticket - Tickets Where Last_Service_Tech_Id <>1 - (NO ROWS SHOULD BE RETURNED)--

select distinct Ticket_Number as "Query 334 - Ticket Number - Last_Service_Tech_Id <>1 - (NO ROWS SHOULD BE RETURNED)",

Last_Service_Tech_Id

from sv_service_ticket

where Service_Ticket_Id <>1

and Last_Service_Tech_Id <> 1

--Query 335: SV_Service_Ticket - OPEN TICKETS - Ticket Resolution Id is Invalid = 1 - (NO ROWS SHOULD BE RETURNED)--

select distinct d.Ticket_Number as "Query 335 - Ticket Number RESOLVED TICKETS - Resolution Id <> 1 - Should be 1 (NO ROWS SHOULD BE RETURNED)",

e.Resolution_Code

from sv_service_ticket d, SV_Resolution e

where d.Ticket_Status = 'RS'

and d.Service_Ticket_Id <> 1

and d.Resolution_Id = 1

and d.Resolution_Id = e.Resolution_Id

--Query 336: SV_Service_Ticket - CLOSED TICKETS - Ticket Resolution Id is Invalid = 1 - (NO ROWS SHOULD BE RETURNED)--

select distinct d.Ticket_Number as "Query 336 - Ticket Number CLOSED TICKETS - Resolution Id <> 1 (NO ROWS SHOULD BE RETURNED)",

e.Resolution_Code

from sv_service_ticket d, SV_Resolution e

where d.Service_Ticket_Id <> 1

and d.Ticket_Status = 'CL'

and d.Resolution_Id = e.Resolution_Id

and d.Resolution_Id = 1

--Query 337: SV_Service_Ticket - Tickets Where Billable   = N - (NO ROWS SHOULD BE RETURNED)--

select distinct Ticket_Number as "Query 337 - Ticket Number - Billable = N - (NO ROWS SHOULD BE RETURNED)", Billable

from sv_service_ticket

where Service_Ticket_Id <>1

and Billable <> 'N'

--Query 338: SV_Service_Ticket - Tickets Where Equipment_Charge <> 0 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 338 - Ticket Number - Equipment Charge <> 0 - (NO ROWS SHOULD BE RETURNED)",

Equipment_Charge , Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <>1

and Equipment_Charge <> 0

--Query 339: SV_Service_Ticket - Tickets Where Labor_Charge <> 0 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 339 - Ticket Number - Labor Charge <> 0 - (NO ROWS SHOULD BE RETURNED)",

Labor_Charge , Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <>1

and Labor_Charge <> 0.00

--Query 340: SV_Service_Ticket - Tickets Where Other_Charge <> 0 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 340 - Ticket Number - Other Charge <> 0 - (NO ROWS SHOULD BE RETURNED)",

Other_Charge, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <>1

and Other_Charge <> 0.00

--Query 341: SV_Service_Ticket - Tickets Where Tax_Total <> 0 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 341 - Ticket Number - Tax_Total <> 0 - (NO ROWS SHOULD BE RETURNED)",

TaxTotal, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <>1

and TaxTotal <> 0.00

--Query 342: SV_Service_Ticket - Tickets Where Regular_Hours <> 0 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 342 - Ticket Number - Regular_Hours <> 0 - (NO ROWS SHOULD BE RETURNED)",

Regular_Hours, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <>1

and Regular_Hours <> 0.00

--Query 343: SV_Service_Ticket - Tickets Where Overtime_Hours <> 0 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 343 - Ticket Number - Overtime_Hours <> 0 - (NO ROWS SHOULD BE RETURNED)",

Overtime_Hours, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <>1

and Overtime_Hours <> 0.00

--Query 344: SV_Service_Ticket - Tickets Where Holiday_Hours <> 0 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 344 - Ticket Number - Holiday_Hours <> 0 - (NO ROWS SHOULD BE RETURNED)",

Holiday_Hours, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <>1

and Holiday_Hours <> 0.00

--Query 345: SV_Service_Ticket - Tickets Where Invoice_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 345 - Ticket Number - Invoice_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

Invoice_Id, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <>1

and Invoice_Id <> 1

--Query 346: SV_Service_Ticket - Tickets Where Regular_Rate   <> 0 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 346 - Ticket Number - Regular_Rate <> 0 - (NO ROWS SHOULD BE RETURNED)",

Regular_Rate, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Regular_Rate <> 0

--Query 347: SV_Service_Ticket - Tickets Where Overtime_Rate   <> 0 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 347 - Ticket Number - Overtime_Rate <> 0 - (NO ROWS SHOULD BE RETURNED)",

Overtime_Rate, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Overtime_Rate <> 0

--Query 348: SV_Service_Ticket - Tickets Where Holiday_Rate   <> 0 - (NO ROWS SHOULD BE RETURNED)", select distinct Ticket_Number as "Query 348 - Ticket Number - Holiday_Rate <> 0 - (NO ROWS SHOULD BE RETURNED)",

Holiday_Rate, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Holiday_Rate <> 0

--Query 349: SV_Service_Ticket - Tickets Where Bypass_Warranty <> N - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 349 - Ticket Number - Bypass_Warranty <> N - (NO ROWS SHOULD BE RETURNED)",

Bypass_Warranty, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Bypass_Warranty <> 'N'

--Query 350: SV_Service_Ticket - Tickets Where Bypass_Service_Level <> N - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 350 - Ticket Number - Bypass_ServiceLevel <> N - (NO ROWS SHOULD BE RETURNED)",

Bypass_ServiceLevel, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Bypass_ServiceLevel <> 'N'

--Query 351: SV_Service_Ticket - Tickets Where IsInspection <> N - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 351 - Ticket Number - IsInspection <> N - (NO ROWS SHOULD BE RETURNED)",

IsInspection, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and IsInspection <> 'N'

--Query 352: SV_Service_Ticket - Tickets Where Manual_Labor <> N - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 352 - Ticket Number - Manual_Labor <> N - (NO ROWS SHOULD BE RETURNED)",

Manual_Labor, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Manual_Labor <> 'N'

--Query 353: SV_Service_Ticket - Tickets Where Remittance <> N - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 353 - Ticket Number - Remittance <> N - (NO ROWS SHOULD BE RETURNED)",

Remittance, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Remittance <> 'N'

--Query 354: SV_Service_Ticket - Tickets Where Payment_Received <> N - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 354 - Ticket Number - Payment_Received <> N - (NO ROWS SHOULD BE RETURNED)",

Payment_Received, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Payment_Received <> 'N'

--Query 355: SV_Service_Ticket - Tickets Where Sub_Problem_Id = 1 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 355 - Ticket Number - Sub_Problem_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

Sub_Problem_Id, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Sub_Problem_Id <> 1

--Query 356: SV_Service_Ticket - Tickets Where Entered_By   <> 'Administrator' - (NO ROWS SHOULD BE RETURNED)",

/* ENTERED BY CAN BE DIFFERENT DEPENDING ON DATA SOURCE */

select distinct Ticket_Number as "Query 356 - Ticket Number - Entered_By <> Administrator - (NO ROWS SHOULD BE RETURNED)",

Entered_By, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Entered_By <> 'Administrator'

--Query 357: SV_Service_Ticket - Tickets Where UserCode <> 'Administrator' - (NO ROWS SHOULD BE RETURNED)",

/* USERCODE CAN BE SET TO A CONVERSION CODE */

select distinct Ticket_Number as "Query 357 - Ticket Number - UserCode <> Administrator - (NO ROWS SHOULD BE RETURNED)",

UserCode, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and UserCode <> 'Administrator'

--Query 358: SV_Service_Ticket - Tickets Where Edit_TimeStamp <> Creation_Date - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 358 - Ticket Number - Edit_TimeStamp <> Creation_Date - (NO ROWS SHOULD BE RETURNED)", Edit_TimeStamp, Creation_Date, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Creation_Date <> Edit_Timestamp

--Query 359: SV_Service_Ticket - Tickets Where CustomerComments = '' - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 359 - Ticket Number - CustomerComments = '' - (NO ROWS SHOULD BE RETURNED)",

CustomerComments, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and CustomerComments = ''

--Query 360: SV_Service_Ticket - Tickets Where Number_Of_Dispatches <> 0 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 360 - Ticket Number - Number_Of_Dispatches <> 0 - (NO ROWS SHOULD BE RETURNED)",

Number_Of_Dispatches, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Number_Of_Dispatches <> 0

--Query 361: SV_Service_Ticket - Tickets Where Customer_CC_Id <> 0 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 361 - Ticket Number - Customer_CC_Id <> 0 - (NO ROWS SHOULD BE RETURNED)",

Customer_CC_Id, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Customer_CC_Id <> 0

--Query 362: SV_Service_Ticket - Tickets Where Customer_Bank_Id <> 0 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 362 - Ticket Number - Customer_Bank_Id <> 0 - (NO ROWS SHOULD BE RETURNED)",

Customer_Bank_Id, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Customer_Bank_Id <> 0

--Query 363: SV_Service_Ticket - Tickets Where Ticket_Status_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 363 - Ticket Number - Ticket_Status_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

Ticket_Status_Id, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Ticket_Status_Id <> 1

--Query 364: SV_Service_Ticket - Tickets Where Customer_EFT_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 364 - Ticket Number - Customer_EFT_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

Customer_EFT_Id, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Customer_EFT_Id <> 1

--Query 365: SV_Service_Ticket - Tickets Where Inspection_Id   <> 1 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 365 - Ticket Number - Inspection_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

Inspection_Id, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Inspection_Id <> 1

--Query 366: SV_Service_Ticket - Tickets Where Service_Ticket_Group_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 366 - Ticket Number - Service_Ticket_Group_Id <> 1 - (NO ROWS SHOULD BE RETURNED)", Service_Ticket_Group_Id, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Service_Ticket_Group_Id <> 1

--Query 367: SV_Service_Ticket - Tickets Where Service_Coordinator_Employee_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 367 - Ticket Number - Service_Coordinator_Employee_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

Service_Coordinator_Employee_Id, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Service_Coordinator_Employee_Id <> 1

--Query 368: SV_Service_Ticket - Tickets Where Inspection_Incremented <> N - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 368 - Ticket Number - Inspection_Incremented <> N - (NO ROWS SHOULD BE RETURNED)",

Inspection_Incremented, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and Inspection_Incremented <> 'N'

--Query 369: SV_Service_Ticket - Tickets Where ByPass_TicketServiceCompany <> N - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 369 - Ticket Number - ByPass_TicketServiceCompany <> N - (NO ROWS SHOULD BE RETURNED)",

ByPass_TicketServiceCompany, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <> 1

and ByPass_TicketServiceCompany <> 'N'

--Query 370: SV_Service_Ticket - Tickets Where Estimated_Length <> SV_Problem.estimated_time - (NO ROWS SHOULD BE RETURNED)",

select distinct a.Ticket_Number as "Query 370 - Ticket Number - Estimated_Length <> SV_Problem.estimated_time - (NO ROWS SHOULD BE RETURNED)",

b.problem_Code, b.Estimated_Time as "Problem Code Estimated_Time", a.Estimated_Length as "Ticket Estimated_Length"

from sv_service_ticket a, SV_Problem b

where a.Service_Ticket_Id <> 1

and a.Problem_Id = b.Problem_Id

and a.Estimated_Length <> b.Estimated_Time

--Query 371: SV_Service_Ticket - Tickets Where Priority_Id <> SV_Problem.Priority_Id - (NO ROWS SHOULD BE RETURNED)", select distinct a.Ticket_Number as "Query 371 - Ticket Number - Priority_Id <> SV_Problem.Priority_Id - (NO ROWS SHOULD BE RETURNED)",

b.problem_Code, b.Priority_Id as "Problem Code Priority_Id", a.Priority_Id as "Ticket Priority_Id"

from sv_service_ticket a, SV_Problem b

where a.Service_Ticket_Id <> 1

and a.Problem_Id = b.Problem_Id

and a.Priority_Id <> b.Priority_Id

--Query 372: SV_Service_Ticket - Tickets Where Expertise_Level <> SV_Problem.Expertise_Level - (NO ROWS SHOULD BE RETURNED)",

select distinct a.Ticket_Number as "Query 372 - Ticket Number - Expertise_Level <> SV_Problem.Expertise_Level - (NO ROWS SHOULD BE RETURNED)",

b.problem_Code, b.Expertise_Level as "Problem Code Expertise_Level", a.Expertise_Level as "Ticket Expertise_Level"

from sv_service_ticket a, SV_Problem b

where a.Service_Ticket_Id <> 1

and a.Problem_Id = b.Problem_Id

and a.Expertise_Level <> b.Expertise_Level

--Query 373: SV_Service_Ticket - Tickets Where Trip_Charge <> System Service_Level - (NO ROWS SHOULD BE RETURNED)",

select distinct a.Ticket_Number as "Query 373 - Ticket Number - Trip_Charge <> System Service_Level - (NO ROWS SHOULD BE RETURNED)",

b.service_Level_code, a.Ticket_Status, a.service_level_id as "Ticket Service_Level_Id", a.Trip_Charge as "Ticket Trip_Charge", c.Service_Level_Id as "System Service_Level_Id", b.Trip_Charge as "Service Level Trip_Charge"

from sv_service_ticket a, SV_Service_Level b, AR_Customer_System c

where a.Service_Ticket_Id <> 1

and a.Service_Level_Id = c.Service_Level_Id

and a.Service_Level_Id = b.Service_Level_Id

and b.Service_Level_Id <> 1

and a.Service_Level_Id <> 1

and c.Service_Level_Id <> 1

and a.Trip_Charge <> b.Trip_Charge

--Query 374: SV_Service_Ticket - Tickets Where Ticket Service_Company_Id <> System Service_Company_Id - (NO ROWS SHOULD BE RETURNED)",

select distinct a.Ticket_Number as "Query 374 - Ticket Number - Service_Company_Id <> System Service_Company_Id - (NO ROWS SHOULD BE RETURNED)",

a.Ticket_Status, a.Service_Company_Id as "Ticket Service_Company_Id",

c.Service_Company_Id as "System Service_Company_Id", b.Service_Company_Code as "System Service_Company_Code", b.Service_Company_Id

from sv_service_ticket a, SV_Service_Company b, AR_Customer_System c

where a.Service_Ticket_Id <> 1

and a.Customer_System_Id = c.Customer_System_Id

and a.Service_Company_Id = b.Service_Company_Id

and a.Service_Company_Id <> c.Service_Company_Id

and b.Service_Company_Id <> 1

and a.Service_Company_Id <> 1

and c.Service_Company_Id <> 1

and a.IsInspection = 'N'

--Query 375: SV_Service_Ticket - Validate Category_Id - (NO ROWS SHOULD BE RETURNED)--

select distinct a.Ticket_Number as "Query 375 - Ticket Number - Validate Category_id <> 1 (NO ROWS SHOULD BE RETURNED)",

b.Category_Code

from sv_service_ticket a, AR_Category b

where a.Service_Ticket_Id <> 1

and a.Category_id = b.Category_Id

and a.Category_Id =1

--Query 376: SV_Service_Ticket - Validate Customer_Bill_Id - (NO ROWS SHOULD BE RETURNED)--

select distinct a.Ticket_Number as "Query 376 - Ticket Number - Validate Customer_Bill_Id <> 1 (NO ROWS SHOULD BE RETURNED)",

a.customer_bill_id as "Ticket Customer_Bill_Id", b.Customer_Bill_Id as "Customer Customer_Bill_Id"

from sv_service_ticket a, AR_Customer_Bill b, ar_customer c, AR_Customer_Site d, AR_Customer_System e

where a.Service_Ticket_Id <> 1

and c.Master_Account_Id = 1

and b.Customer_Id = c.Customer_Id

and a.Customer_Bill_Id = b.Customer_Bill_Id

and a.Customer_Site_Id = d.Customer_Site_Id

and a.Customer_Bill_Id = d.Customer_Bill_Id

and a.Customer_System_Id = e.Customer_System_Id and a.Customer_Bill_Id = 1

--Query 377: SV_Service_Ticket - OPEN TICKETS Where Resolved_Date <> 1899... - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 377 - Ticket Number - OPEN TICKETS Where Resolved_Date <> 1899... - (NO ROWS SHOULD BE RETURNED)",

Resolved_Date

from sv_service_ticket

where Service_Ticket_Id <>1

and Ticket_Status = 'OP'

and Resolved_Date > '01-01-1900'

--Query 378: SV_Service_Ticket - CLOSED TICKETS Where Resolved_Date < 1-1-1900 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 378 - Ticket Number - CLOSED TICKETS Where Resolved_Date <1-1-1900 - (NO ROWS SHOULD BE RETURNED)",

Resolved_Date, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <>1

and Ticket_Status = 'CL'

and Resolved_Date < '01-01-1900'

--Query 379: SV_Service_Ticket - Validate Route_Id - (NO ROWS SHOULD BE RETURNED)--

select distinct a.Route_Id as "Query 379 - Ticket Number - Validate Route (If the Customer is not using Routes, all Should be 1)",

Count(a.Route_Id) as "Count of Tickets by Route_Id"

from sv_service_ticket a, AR_Customer_Bill b, ar_customer c, AR_Customer_Site d, AR_Customer_System e, SV_Routes f

where a.Service_Ticket_Id <> 1

and c.Master_Account_Id = 1

and b.Customer_Id = c.Customer_Id

and a.Customer_Bill_Id = b.Customer_Bill_Id

and a.Customer_Site_Id = d.Customer_Site_Id

and a.Customer_Bill_Id = d.Customer_Bill_Id

and a.Customer_System_Id = e.Customer_System_Id

and e.Route_Id = f.Route_Id

and a.Route_Id = e.Route_Id

Group by a.Route_Id

--Query 380: SV_Service_Ticket - CLOSED TICKETS Where ClosedDate < 1-1-1900 - (NO ROWS SHOULD BE RETURNED)",

select distinct Ticket_Number as "Query 380 - Ticket Number - CLOSED TICKETS Where Closed_Date <1-1-1900 - (NO ROWS SHOULD BE RETURNED)",

ClosedDate, Ticket_Status

from sv_service_ticket

where Service_Ticket_Id <>1

and Ticket_Status = 'CL'

and ClosedDate < '01-01-1900'

--*** IN_PART FIELD VALIDATION QUERIES***--

--Query 400: IN_Part - Checking for Duplicate Part Codes - (NO ROWS SHOULD BE RETURNED)--

```
select Part_Code as "Query 400 - IN_Part - Duplicate Part_Code (NO ROWS SHOULD BE RETURNED)",

Count(Part_Code) as "Number of Duplicate Part Code"

from IN_Part

where Part_Id > @partid

Group by Part_Code

having count(Part_Code) >1
```

--Query 401: IN_Part - Checking for Blank Description - (NO ROWS SHOULD BE RETURNED)--

```
select Part_Code as "Query 401 - IN_Part - Checking for Blank Description (NO ROWS SHOULD BE RETURNED)",

Description

from IN_Part

where Part_Id > @partid

and Description = ''

Order by Part_Code
```

--Query 402: IN_Part - Checking for Blank Detail - (NO ROWS SHOULD BE RETURNED)--

select Part_Code as "Query 402 - IN_Part - Checking for Blank Detail (NO ROWS SHOULD BE RETURNED)",

Detail

from IN_Part

where Part_Id > @partid

and Detail = ''

Order by Part_Code

--Query 403: IN_Part - Checking for Unit_Of_Measure_Id <> 2 - (NO ROWS SHOULD BE RETURNED)--

select Part_Code as "Query 403 - IN_Part - Checking for Unit_Of_Measure_Id <> 2 (NO ROWS SHOULD BE RETURNED)",

Unit_Of_Measure_id

from IN_Part

where Part_Id > @partid

and Unit_Of_Measure_Id <> 2

Order by Part_Code

--Query 404: IN_Part - Checking for Non_Value_Part <> N - (NO ROWS SHOULD BE RETURNED)--

select Part_Code as "Query 404 - IN_Part - Checking for Non_Value_Part <> N (NO ROWS SHOULD BE RETURNED)",

Non_Value_Part

from IN_Part

where Part_Id > @partid

and Non_Value_Part = 'Y'

```
Order by Part_Code

--select * from in_part where part_code = '001000S4'

--Query 405: IN_Part - Checking for Purchase_UOM_Id <> 2 - (NO ROWS SHOULD BE RETURNED)--

select Part_Code as "Query 405 - IN_Part - Checking for Purchase_UOM_Id <> 2 (NO ROWS SHOULD BE RETURNED)",

Purchase_UOM_Id

--select *

from IN_Part

where Part_Id > @partid

and Purchase_UOM_Id <> 2

Order by Part_Code

--Query 406: IN_Part - Checking for Issue_UOM_Id <> 2 - (NO ROWS SHOULD BE RETURNED)--

select Part_Code as "Query 406 - IN_Part - Checking for Issue_UOM_Id <> 2 (NO ROWS SHOULD BE RETURNED)",

Issue_UOM_Id

from IN_Part

where Part_Id > @partid

and Issue_UOM_Id <> 2

Order by Part_Code
```

--Query 407: IN_Part - Checking If Service_Price =0 0 - (NO ROWS SHOULD BE RETURNED)--

select Part_Code as "Query 407 - IN_Part - Checking If Service_Price = 0 (NO ROWS SHOULD BE RETURNED)",

Service_Price

from IN_Part

where Part_Id > @partid

and Service_Price = 0

Order by Part_Code

--Query 408: IN_Part - Checking for Blank Service_Description - (NO ROWS SHOULD BE RETURNED)--

select Part_Code as "Query 408 - IN_Part - Checking for Blank Service_Description (NO ROWS SHOULD BE RETURNED)",

Service_Description

from IN_Part

where Part_Id > @partid

and Service_Description = ''

Order by Part_Code

--Query 409: IN_Part - Checking If Sales_Price = 0 - (NO ROWS SHOULD BE RETURNED)--

select a.Part_Code as "Query 409 - IN_Part - Checking If Sales_Price = 0 (NO ROWS SHOULD BE RETURNED)",

a.Sales_Price, b.Product_Line_Code

from IN_Part a, IN_Product_Line b

where a.Part_Id > @partid

and a.Sales_Price = 0

and a.Product_Line_Id = b.Product_Line_Id

Order by b.Product_Line_Code, a.Part_Code

--Query 410: IN_Part - Checking for Blank Sales_Description - (NO ROWS SHOULD BE RETURNED)--

select Part_Code as "Query 410 - IN_Part - Checking for Blank Sales_Description (NO ROWS SHOULD BE RETURNED)",

Sales_Description

from IN_Part

where Part_Id > @partid

and Sales_Description = ''

Order by Part_Code

--Query 411: IN_Part - Checking for Inactive = Y - (NO ROWS SHOULD BE RETURNED)--

select a.Part_Code as "Query 411 - IN_Part - Checking for Inactive = Y (NO ROWS SHOULD BE RETURNED)",

a.Inactive, b.Product_Line_Code

from IN_Part a, IN_Product_Line b

where Part_Id > @partid

and a.Product_Line_Id = b.Product_Line_Id

and a.Inactive = 'Y'

Order by b.Product_Line_Code, a.Part_Code

--Query 412: IN_Part - Checking for Service_Part = N - (NO ROWS SHOULD BE RETURNED)--

select a.Part_Code as "Query 412 - IN_Part - Checking for Service_Part = N (NO ROWS SHOULD BE RETURNED)",

a.Service_Part, b.Product_Line_Code

from IN_Part a, IN_Product_Line b

where Part_Id > @partid

and a.Product_Line_Id = b.Product_Line_Id

and a.Service_Part = 'N'

Order by b.Product_Line_Code, a.Part_Code

--Query 413: IN_Part - Checking for Sales_Part = N - (NO ROWS SHOULD BE RETURNED)--

select a.Part_Code as "Query 413 - IN_Part - Checking for Sales_Part = N (NO ROWS SHOULD BE RETURNED)",

a.Sales_Part, b.Product_Line_Code

from IN_Part a, IN_Product_Line b

where Part_Id > @partid

and a.Product_Line_Id = b.Product_Line_Id

and a.Sales_Part = 'N'

Order by b.Product_Line_Code, a.Part_Code

--Query 414: IN_Part - Checking for Special_Order = Y - (*Provide results to the customer to review*)--

select a.Part_Code as "Query 414 - IN_Part - Checking for Special_Order = Y (*Provide results to the customer to review*)",

Special_Order , b.Product_Line_Code

from IN_Part a, IN_Product_Line b

where Part_Id > @partid

and Special_Order = 'Y'

and a.Product_Line_Id = b.Product_Line_Id

Order by b.Product_Line_Code, a.Part_Code

--Query 415: IN_Part - Checking for Dir_Expense_Account_Id <> 1 - (*Provide results to the customer to review*)--

select a.Part_Code as "Query 415 - IN_Part - Checking for Dir_Expense_Account_Id <> 1 (*Provide results to the customer to review*)",

Dir_Expense_Account_Id, c.Account_Code as "GL Account", c.Description as "GL Account Descr", b.Product_Line_Code

from IN_Part a, IN_Product_Line b, GL_Account c

where Part_Id > @partid

and Dir_Expense_Account_Id <> 1

and a.Dir_Expense_Account_Id = c.Account_Id

and a.Product_Line_Id = b.Product_Line_Id

Order by b.Product_Line_Code, a.Part_Code

--Query 416: IN_Part - Checking for Customer_Equipment_Breakout <> N - (NO ROWS SHOULD BE RETURNED)--

select Part_Code as "Query 416 - IN_Part - Checking for Customer_Equipment_Breakout <> N (NO ROWS SHOULD

BE RETURNED)",

Customer_Equipment_Breakout

from IN_Part

where Part_Id > @partid

and Customer_Equipment_Breakout <> 'N'

Order by Part_Code

--Query 417: IN_Part - Checking for Freeze_Purchasing <> N - (NO ROWS SHOULD BE RETURNED)--

select Part_Code as "Query 417 - IN_Part - Checking for Freeze_Purchasing <> N (NO ROWS SHOULD BE RETURNED)",

Freeze_Purchasing

from IN_Part

where Part_Id > @partid

and Freeze_Purchasing <> 'N'

Order by Part_Code

--Query 418: IN_Part - Checking for Income_Account_Id <> 1 - (*Provide results to the customer to review*)--

select a.Part_Code as "Query 418 - IN_Part - Checking for Income_Account_Id <> 1 (*Provide results to the customer to review*)",

Income_Account_Id, c.Account_Code as "GL Account", c.Description as "GL Account Descr", b.Product_Line_Code

from IN_Part a, IN_Product_Line b, GL_Account c

where Part_Id > @partid

and Income_Account_Id <> 1

and a.Income_Account_Id = c.Account_Id

and a.Product_Line_Id = b.Product_Line_Id

Order by b.Product_Line_Code, a.Part_Code

--Query 419: IN_Part - Validate Item_Id <> 1 - (NO ROWS SHOULD BE RETURNED)--

select a.Part_Code as "Query 419 - IN_Part - Validate Item_Id <> 1 ((NO ROWS SHOULD BE RETURNED)",

Item_Id, c.Account_Code as "GL Account", c.Description as "GL Account Descr", b.Product_Line_Code

from IN_Part a, IN_Product_Line b, GL_Account c

where Part_Id > @partid

and Item_Id = 1

and c.Account_Id <> 1

and a.Item_Id = c.Account_Id

and a.Product_Line_Id = b.Product_Line_Id

Order by b.Product_Line_Code, a.Part_Code

--Query 420: IN_Part - Validate Product_Line_Id <> 1 - (NO ROWS SHOULD BE RETURNED)--

select a.Part_Code as "Query 420 - IN_Part - Validate Product_Line_Id <> 1 ((NO ROWS SHOULD BE RETURNED)",

b.Product_Line_Code

from IN_Part a, IN_Product_Line b

where a.Part_Id > @partid

and a.Product_Line_Id = 1

and a.Product_Line_Id = b.Product_Line_Id

Order by b.Product_Line_Code, a.Part_Code

--Query 421: IN_Part - Count of Part Code by Manufacturer_Code - (*Provide results to the customer to review*)--

select b.Manufacturer_Code as "Query 421 - IN_Part - Count of Part Code by Manufacturer_Code (*Provide results to the customer to review*)",

Count(b.Manufacturer_Code) as "Count of Parts by Manufacturer Code"

from IN_Part a, IN_Manufacturer b

where a.Part_Id > @partid

and a. Manufacturer_Id = b.Manufacturer_Id

Group by b.Manufacturer_Code

Order by b.Manufacturer_Code

--Query 422: IN_Part - Checking for PPV_Account_Id <> 1 - (NO ROWS SHOULD BE RETURNED)--

select a.Part_Code as "Query 422 - IN_Part - Checking for PPV_Account_Id <> 1 ((NO ROWS SHOULD BE RETURNED)",

PPV_Account_Id, c.Account_Code as "GL Account", c.Description as "GL Account Descr", b.Product_Line_Code

from IN_Part a, IN_Product_Line b, GL_Account c

where Part_Id > @partid

and PPV_Account_Id = 1

and a.PPV_Account_Id = c.Account_Id

and a.Product_Line_Id = b.Product_Line_Id

Order by b.Product_Line_Code, a.Part_Code

--Query 423: IN_Part - Count of Part Code by Job_Use_default - (*Provide results to the customer to review*)--

select Job_Use_default as "Query 423 - IN_Part - Count by Job_Use_default S Indicates STOCK ITEM FOR JOBS (*Provide results to the customer to review*)",

Count(Job_Use_default) as "Count of Parts by Manufacturer Code"

from IN_Part where Part_Id > @partid

Group by Job_Use_default

Order by Job_Use_default

--Query 424: IN_Part - Job_Use_default - ((*Provide results to the customer to review*)--

select a.Part_Code as "Query 424 - IN_Part - Job_Use_default ((*Provide results to the customer to review*)",

a.Job_Use_default, b.Product_Line_Code

from IN_Part a, IN_Product_Line b

where Part_Id > @partid

and a.Product_Line_Id = b.Product_Line_Id

Order by a.Job_Use_Default, a.Part_Code

--Query 425: IN_Part - Count of Part Code by Customer_Equipment - (*Provide results to the customer to review*)--

select Customer_Equipment as "Query 425 - IN_Part - Count by Customer_Equipment (*Provide results to the customer to review*)",

Count(Customer_Equipment) as "Count of Parts by Customer_Equipment Flag"

from IN_Part where Part_Id > @partid

Group by Customer_Equipment

Order by Customer_Equipment

--Query 426: IN_Part - Part Code Where Customer_Equipment = N - (*Provide results to the customer to review*)--

select Part_Code as "Query 426 - IN_Part - Customer_Equipment = N (*Provide results to the customer to review*)",

Customer_Equipment

from IN_Part where Part_Id > @partid

and Customer_Equipment = 'N'

Order by Customer_Equipment, Part_Code

--Query 427: IN_Part - Validate Purchase_Description is Not Blank - (NO ROWS SHOULD BE RETURNED)--

select Part_Code as "Query 427 - IN_Part - Validate Purchase_Description Not Blank (NO ROWS SHOULD BE RETURNED)",

purchase_description

from IN_Part

where Part_Id > @partid

and Purchase_Description = ''

Order by Part_Code

--Query 428: IN_Part - Validate Part Purchase_Description = Part Supplier Description - (NO ROWS SHOULD BE RETURNED)--

```
select a.Part_Code as "Query 428 - IN_Part - Part Purchase_Description = Part Supplier Description (NO ROWS SHOULD BE RETURNED)",

a.purchase_description as "Part Purchase_Description", b.Description as "Primary Vendor Part Description"

from IN_Part a, IN_Part_Supplier b

where a.Part_Id > @partid

and a.part_id = b.part_id

and b.Is_Primary = 'Y'

and a.Purchase_Description <> b.Description

Order by Part_Code




--Query 429: IN_Part - Validate Primary Vendor_Id - (NO ROWS SHOULD BE RETURNED)--

select a.Part_Code as "Query 429 - IN_Part - Validate Primary_Vendor_Id (NO ROWS SHOULD BE RETURNED)",

a.Primary_Vendor_Id as "Part Primary_Vendor_Id", b.Vendor_Id as "Supplier Primary Vendor_Id"

from IN_Part a, IN_Part_Supplier b, ap_Vendor c

where a.Part_Id > @partid

and a.part_id = b.part_id

and b.Is_Primary = 'Y'

and a.Primary_Vendor_Id <> c.Vendor_Id

and b.Vendor_Id = c.Vendor_Id

and a.Primary_Vendor_Id <> 1

and b.Vendor_Id <> 1

and c.Vendor_Id <> 1

Order by Part_Code
```

--Query 430: IN_Part - Validate Part Primary Vendor_Id - (NO ROWS SHOULD BE RETURNED)--

select distinct a.Part_Code as "Query 430 - IN_Part - Validate Part Primary_Vendor_Id (NO ROWS SHOULD BE RETURNED)",

a.Primary_Vendor_Id as "Part Primary_Vendor_Id", b.vendor_code

from IN_Part a, ap_Vendor b

where a.Part_Id > @partid

and a.Primary_Vendor_Id = b.Vendor_Id

and a.Primary_Vendor_Id = 1

and b.Vendor_Id <> 1

Order by Part_Code

--Query 431: IN_Part - Validate Part Vendor_Part = Supplier Vendor_Part_Number - (NO ROWS SHOULD BE RETURNED)--

select a.Part_Code as "Query 431 - IN_Part - Validate Part Vendor_Part = Supplier Vendor_Part_Number (NO ROWS SHOULD BE RETURNED)",

a.Vendor_Part as "Part Vendor_Part", b.Vendor_Part_Number as "Supplier Vendor_Part_Number"

from IN_Part a, IN_Part_Supplier b

where a.Part_Id > @partid

and a.part_id = b.part_id

and b.Is_Primary = 'Y'

and a.Vendor_Part <> b.Vendor_Part_Number

and a.Primary_Vendor_Id <> 1

and b.Vendor_Id <> 1

Order by Part_Code

--*** IN_PART_SUPPLIER FIELD VALIDATION QUERIES***--

--Query 440: IN_Part_Supplier - Validate Vendor_Id - (NO ROWS SHOULD BE RETURNED)--

select distinct a.Part_Code as "Query 440 - Part_Code - IN_Part_Supplier - Validate Vendor_Id (NO ROWS SHOULD BE RETURNED)",

b.Vendor_Id as "Part Supplier Vendor_Id", c.vendor_code

from IN_Part a, IN_Part_Supplier b, ap_Vendor c

where a.Part_Id > @partid

and a.Primary_Vendor_Id = b.Vendor_Id

and a.Primary_Vendor_Id <> 1

and b.Vendor_Id = c.Vendor_Id

and b.Vendor_Id = 1

Order by Part_Code

--Query 441: IN_Part_Supplier - Validate Vendor_Part_Number Not Blank - (NO ROWS SHOULD BE RETURNED)--

select distinct Vendor_Part_Number as "Query 441 - Validate Vendor_Part_Number is Not Blank (NO ROWS SHOULD BE RETURNED)"

from IN_Part_Supplier

where Part_Id > @partid

and Vendor_Part_Number = ''

Order by Vendor_Part_Number

--Query 442: IN_Part_Supplier - Validate Description Not Blank - (NO ROWS SHOULD BE RETURNED)--

select distinct Description as "Query 442 - Validate Part Supplier Description is Not Blank (NO ROWS SHOULD BE RETURNED)"

from IN_Part_Supplier

where Part_Id > @partid

and Description = ''

Order by Description

--Query 443: IN_Part_Supplier - Validate Cost <> 0 - (NO ROWS SHOULD BE RETURNED)--

select distinct b.Part_Code as "Query 443 - Validate Supplier Cost <> 0 (NO ROWS SHOULD BE RETURNED)",

a.Cost as "Part Supplier Cost"

from IN_Part_Supplier a, in_part b

where a.Part_Id > @partid

and b.Part_Id > @partid

and a.Part_Id = b.Part_Id

and a.Cost = 0

Order by b.Part_Code

--Query 444: IN_Part_Supplier - Validate List_Price <> 0 - (NO ROWS SHOULD BE RETURNED)--

```
select distinct b.Part_Code as "Query 444 - Validate Supplier List_Price <> 0 (NO ROWS SHOULD BE RETURNED)",

a.List_Price as "Part Supplier List_Price"

from IN_Part_Supplier a, in_part b

where a.Part_Id > @partid

and b.Part_Id > @partid

and a.Part_Id = b.Part_Id

and a.List_Price = 0

Order by b.Part_Code



--Query 445: IN_Part_Supplier - Validate Quantity <> 0 - (NO ROWS SHOULD BE RETURNED)--

select distinct b.Part_Code as "Query 445 - Validate Supplier Quantity <> 0 (NO ROWS SHOULD BE RETURNED)",

a.Quantity as "Part Supplier Quantity"

from IN_Part_Supplier a, in_part b

where a.Part_Id > @partid

and b.Part_Id > @partid

and a.Part_Id = b.Part_Id

and a.Quantity <> 1

Order by b.Part_Code



--*** AP_PURCHASE_ORDER FIELD VALIDATION QUERIES***--
```

```
--Query 460: AP_Purchase_Order - Checking for Duplicate PO Numbers - (NO ROWS SHOULD BE RETURNED)--

select PO_Number as "Query 460 - AP_Purchase_Order - Checking for Duplicate PO_Number (NO ROWS SHOULD BE RETURNED)",

Count(PO_Number) as "Number of Duplicate PO Numbers"

from AP_Purchase_Order

where Purchase_Order_Id <> 1

Group by PO_Number

having count(PO_Number) >1
```

```
--Query 461: AP_Purchase_Order - Verify Type_JSO = O - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 461 - AP_Purchase_Order - Verify Type_JSO <> O (NO ROWS SHOULD BE RETURNED)",

Type_JSO

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Type_JSO <> 'O'

Order by PO_Number
```

```
--Query 462: AP_Purchase_Order - Verify Job_Id = 1 - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 462 - AP_Purchase_Order - Verify Job_Id = 1 (NO ROWS SHOULD BE RETURNED)",

Job_Id

from AP_Purchase_Order
```

where Purchase_Order_Id <> 1

and Job_Id <> 1

Order by PO_Number

--Query 463: AP_Purchase_Order - Verify Service_Id = 1 - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 463 - AP_Purchase_Order - Verify Service_Id = 1 (NO ROWS SHOULD BE RETURNED)",

Service_Id

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Service_Id <> 1

Order by PO_Number

--Query 464: AP_Purchase_Order - Verify Closed = N - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 464 - AP_Purchase_Order - Verify Closed = N (NO ROWS SHOULD BE RETURNED)",

Closed

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Closed <> 'N'

Order by PO_Number

--Query 465: AP_Purchase_Order - Verify On_Back_Order = N - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 465 - AP_Purchase_Order - Verify On_Back_Order = N (NO ROWS SHOULD BE RETURNED)",

On_Back_Order

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and On_Back_Order <> 'N'

Order by PO_Number

--Query 466: AP_Purchase_Order - Verify UserCode = 'Administrator' - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 466 - AP_Purchase_Order - Verify UserCode = 'Administrator' (NO ROWS SHOULD BE RETURNED)",

UserCode

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and UserCode <> 'Administrator'

Order by PO_Number

--Query 467: AP_Purchase_Order - Verify Edit_UserCode = 'Administrator' - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 467 - AP_Purchase_Order - Verify Edit_UserCode = 'Administrator' (NO ROWS SHOULD BE RETURNED)",

Edit_UserCode

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Edit_UserCode <> 'Administrator'

Order by PO_Number

--Query 468: AP_Purchase_Order - Verify Exchange_Rate = 1.00 - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 468 - AP_Purchase_Order - Verify Exchange_Rate = 1.00 (NO ROWS SHOULD BE RETURNED)",

Exchange_Rate

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Exchange_Rate <> 1.00

Order by PO_Number

--Query 469: AP_Purchase_Order - Verify Vendor_Acknowledged = 0 - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 469 - AP_Purchase_Order - Verify Vendor_Acknowledged = 0 (NO ROWS SHOULD BE RETURNED)",

Vendor_Acknowledged

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Vendor_Acknowledged <> 0

Order by PO_Number

--Query 470: AP_Purchase_Order - Verify Integration_Status = O - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 470 - AP_Purchase_Order - Verify Integration_Status = O (NO ROWS SHOULD BE RETURNED)",

Integration_Status

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Integration_Status <> 'O'

Order by PO_Number

--Query 471: AP_Purchase_Order - Verify Vendor_Acknowledged_Date = 1899... - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 471 - AP_Purchase_Order - Verify Vendor_Acknowledged_Date = 1899... (NO ROWS SHOULD BE RETURNED)",

Vendor_Acknowledged_Date

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Vendor_Acknowledged_Date <> '1899-12-30 00:00:00:000'

Order by PO_Number

--Query 472: AP_Purchase_Order - Verify Due_Date = 1899... - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 472 - AP_Purchase_Order - Verify Due_Date = 1899... (NO ROWS SHOULD BE RETURNED)", Due_Date

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Due_Date <> '1899-12-30 00:00:00:000'

Order by PO_Number

--Query 473: AP_Purchase_Order - Verify Shipping_Date = 1899... - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 473 - AP_Purchase_Order - Verify Shipping_Date = 1899... (NO ROWS SHOULD BE RETURNED)",

Shipping_Date

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Shipping_Date <> '1899-12-30 00:00:00:000'

Order by PO_Number

--Query 474: AP_Purchase_Order - Verify Acknowledged_Date = 1899... - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 474 - AP_Purchase_Order - Verify Acknowledged_Date = 1899... (NO ROWS SHOULD BE RETURNED)",

Acknowledged_Date

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Acknowledged_Date <> '1899-12-30 00:00:00:000'

Order by PO_Number

--Query 475: AP_Purchase_Order - Verify Shipping_Address not Blank if Warehouse_Id >1 - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 475 - AP_Purchase_Order - Verify Shipping_Address not Blank if Warehouse_Id >1 (NO ROWS SHOULD BE RETURNED)",

Warehouse_Id, Shipping_Address

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Warehouse_Id > 1

and Shipping_Address = ''

Order by PO_Number

--Query 476: AP_Purchase_Order - Validate Category_Id - (NO ROWS SHOULD BE RETURNED)--

select distinct a.PO_Number as "Query 476 - AP_Purchase_Order - Validate Category_Id (NO ROWS SHOULD BE RETURNED)",

b.Category_Code

from AP_Purchase_Order a, AR_Category b

where a.Purchase_Order_Id <> 1

and a.Category_Id = b.Category_Id

and a.Category_Id = 1

and b.Category_Id <> 1

Order by PO_Number

--Query 477: AP_Purchase_Order - Validate Vendor_Id - (NO ROWS SHOULD BE RETURNED)--

select distinct a.PO_Number as "Query 477 - AP_Purchase_Order - Validate Vendor_Id (NO ROWS SHOULD BE RETURNED)",

b.Vendor_Code

from AP_Purchase_Order a, AP_Vendor b

where a.Purchase_Order_Id <> 1

and a.Vendor_Id = b.Vendor_Id

and a.Vendor_Id = 1

and b.Vendor_Id <> 1

Order by PO_Number

--Query 478: AP_Purchase_Order - Validate Shipping_Method_Id - (NO ROWS SHOULD BE RETURNED)--

select distinct a.PO_Number as "Query 478 - AP_Purchase_Order - Validate Shipping_Method_Id (NO ROWS SHOULD BE RETURNED)",

b.Shipping_Method_Code

from AP_Purchase_Order a, AP_Shipping_Method b

where a.Purchase_Order_Id <> 1

and a.Shipping_Method_Id = b.Shipping_Method_Id

and a.Warehouse_Id >1

and a.Shipping_Method_Id = 1

and b.Shipping_Method_Id <> 1

Order by PO_Number

```
--Query 479: AP_Purchase_Order - Validate Warehouse_Id - (NO ROWS SHOULD BE RETURNED)--

select distinct a.PO_Number as "Query 479 - AP_Purchase_Order - Validate Warehouse_Id (NO ROWS SHOULD BE RETURNED)",

b.Warehouse_Code

from AP_Purchase_Order a, IN_Warehouse b

where a.Purchase_Order_Id <> 1

and a.Warehouse_Id = b.Warehouse_Id

and a.Warehouse_Id =1

and b.Warehouse_Id <> 1

Order by PO_Number
```

```
--Query 480: AP_Purchase_Order - Validate Branch_Id - (NO ROWS SHOULD BE RETURNED)--

select distinct a.PO_Number as "Query 480 - AP_Purchase_Order - Validate Branch_Id (NO ROWS SHOULD BE RETURNED)",

b.Branch_Code

from AP_Purchase_Order a, AR_Branch b

where a.Purchase_Order_Id <> 1

and a.Branch_Id = b.Branch_Id

and a.Branch_Id = 1

and b.Branch_Id <> 1

Order by PO_Number
```

--Query 481: AP_Purchase_Order - Verify Order_Date > 1-1-1900 - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 481 - AP_Purchase_Order - Verify Order_Date > 1-1-1900 (NO ROWS SHOULD BE RETURNED)",

Order_Date

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Order_Date < '1-1-1900'

Order by PO_Number

--Query 482: AP_Purchase_Order - Verify Total_Cost > 0 - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 482 - AP_Purchase_Order - Verify Total_Cost > 0 (NO ROWS SHOULD BE RETURNED)",

Total_Cost

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Closed = 'N'

and Total_Cost <= 0

Order by PO_Number

--Query 483: AP_Purchase_Order - Verify Part_Total + Expense_Total = Total_Cost - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 483 - AP_Purchase_Order - Verify Part_Total + Expense_Total = Total_Cost (NO ROWS SHOULD BE RETURNED)",

Part_Total, Expense_Total

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Closed = 'N'

and Part_Total + Expense_Total <> Total_Cost

Order by PO_Number

--Query 484: AP_Purchase_Order - Verify Direct_Expense = Y when Warehouse_Id = 1 - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 484 - AP_Purchase_Order - Verify Direct_Expense = Y when Warehouse_Id = 1 (NO ROWS SHOULD BE RETURNED)",

Direct_Expense, Warehouse_Id

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Closed = 'N'

and Warehouse_Id = 1

and Direct_Expense = 'N'

Order by PO_Number

--Query 485 AP_Purchase_Order - Verify Entered_Date > 1-1-1900 - (NO ROWS SHOULD BE RETURNED)--

select distinct PO_Number as "Query 485 - AP_Purchase_Order - Verify Entered_Date > 1-1-1900 (NO ROWS SHOULD BE RETURNED)", Entered_Date, Order_Date

from AP_Purchase_Order

where Purchase_Order_Id <> 1

and Closed = 'N'

and Entered_Date <= '1-1-1900'

Order by PO_Number

--Query 486: AP_Purchase_Order - Validate Currency_Id - (NO ROWS SHOULD BE RETURNED)--

select distinct a.PO_Number as "Query 486 - AP_Purchase_Order - Validate Currency_Id (NO ROWS SHOULD BE RETURNED)",

b.Currency_Code

from AP_Purchase_Order a, SY_Currency b

where a.Purchase_Order_Id <> 1

and a.Currency_Id = b.Currency_Id

and a.Currency_Id = 1

and b.Currency_Id <> 1

Order by PO_Number

--*** AP_PURCHASE_ORDER_EXPENSE FIELD VALIDATION QUERIES***--

--Query 500: AP_Purchase_Order_Expense - Verify a record exists in AP_Purchase_Order - (NO ROWS SHOULD BE RETURNED)--

select distinct Purchase_Order_Expense_Id as "Query 500 - AP_Purchase_Order_Expense - Verify a record exists in

AP_Purchase_Order (NO ROWS SHOULD BE RETURNED)"

from AP_Purchase_Order_Expense

where Purchase_Order_Id <> 1

and Purchase_Order_Id not in (select Purchase_Order_Id from ap_purchase_order)

Order by Purchase_Order_Expense_Id

--Query 501: AP_Purchase_Order_Expense - Validate Account_Id - (*Provide results to the customer to review for accuracy*)--

select distinct b.Account_Code as "Query 501 - AP_Purchase_Order GL Account - Validate Account_Id (*Provide results to the customer to review for accuracy*)",

b.Description, c.PO_Number

from AP_Purchase_Order_Expense a, GL_Account b, AP_Purchase_Order c

where a.Purchase_Order_Id <> 1

and a.Account_Id = b.Account_Id

and b.Account_Id <> 1

and a.Purchase_Order_Id = c.Purchase_Order_Id

Order by b.Account_Code

--Query 502: AP_Purchase_Order_Expense - Validate Category_Id - (*Provide results to the customer to review for accuracy*)--

select distinct b.Category_Code as "Query 502 - AP_Purchase_Order - Validate Category_Id (*Provide results to the customer to review for accuracy*)",

Count(b.Category_Code) as "Count of Purchase Orders by Category"

from AP_Purchase_Order_Expense a, AR_Category b, AP_Purchase_Order c where a.Purchase_Order_Id <> 1

and a.Category_Id = b.Category_Id

and b.Category_Id <> 1

and a.Purchase_Order_Id = c.Purchase_Order_Id

Group by b.Category_Code

Order by b.Category_Code

--Query 503: AP_Purchase_Order_Expense - Validate Category_Id - (*Provide results to the customer to review for accuracy*)--

select distinct b.Category_Code as "Query 503 - AP_Purchase_Order - Validate Category_Id (*Provide results to the customer to review for accuracy*)",

c.PO_Number

from AP_Purchase_Order_Expense a, AR_Category b, AP_Purchase_Order c

where a.Purchase_Order_Id <> 1

and a.Category_Id = b.Category_Id

and b.Category_Id <> 1

and a.Purchase_Order_Id = c.Purchase_Order_Id

Order by b.Category_Code

--Query 504: AP_Purchase_Order_Expense - Validate Expense_Type_Id - (NO ROWS SHOULD BE RETURNED)--

select distinct a.Purchase_Order_Id as "Query 504 - AP_Purchase_Order - Validate Job_Expense_Code (NO ROWS SHOULD BE RETURNED)",

a.Job_Expense_Code

from AP_Purchase_Order_Expense a, GL_Expense_Type b

where a.Purchase_Order_Id <> 1

and a.Job_Expense_Code not in (select Expense_Code from GL_Expense_Type)

Order by a.Purchase_Order_Id

--Query 505: AP_Purchase_Order_Expense - Validate Expense_Type_Id - (NO ROWS SHOULD BE RETURNED)--

select distinct a.Purchase_Order_Id as "Query 505 - AP_Purchase_Order - Validate Expense_Type_Id (NO ROWS SHOULD BE RETURNED)",

a.Expense_Type_Id

from AP_Purchase_Order_Expense a, GL_Expense_Type b

where a.Purchase_Order_Id <> 1

and a.Expense_Type_Id not in (select Expense_Type_Id from GL_Expense_Type)

Order by a.Purchase_Order_Id

--Query 506: AP_Purchase_Order_Expense - Verify Amount = Quantity x Rate - (NO ROWS SHOULD BE RETURNED)--

select distinct a.Purchase_Order_Expense_Id as "Query 506 - AP_Purchase_Order_Expense - Verify Amount = Quantity x Rate (NO ROWS SHOULD BE RETURNED)",

a.Purchase_Order_Expense_Id, b.Purchase_Order_Id, b.PO_Number, a.Quantity, a.Rate, a.Amount

from AP_Purchase_Order_Expense a, AP_Purchase_Order b

where a.Purchase_Order_Id <> 1

and b.Purchase_Order_Id <> 1

and a.Purchase_Order_Id = b.Purchase_Order_Id and a.Quantity * a.rate <> a.Amount

and a.Purchase_Order_Expense_Id <> 1

Order by a.Purchase_Order_Expense_Id

--Query 507: AP_Purchase_Order_Expense - Verify Description Not Blank - (NO ROWS SHOULD BE RETURNED)--

select distinct a.Purchase_Order_Expense_Id as "Query 507 - AP_Purchase_Order_Expense - Verify Description Not Blank (NO ROWS SHOULD BE RETURNED)",

a.Description, b.PO_Number

from AP_Purchase_Order_Expense a, AP_Purchase_Order b

where a.Purchase_Order_Id <> 1

and b.Purchase_Order_Id <> 1

and a.Purchase_Order_Id = b.Purchase_Order_Id

and a.Description = ''

Order by a.Purchase_Order_Expense_Id

--Query 508: AP_Purchase_Order_Expense - Verify Customer_Id = 1 - (NO ROWS SHOULD BE RETURNED)--

select distinct a.Purchase_Order_Expense_Id as "Query 508 - AP_Purchase_Order_Expense - Verify Customer_Id = 1 (NO ROWS SHOULD BE RETURNED)",

a.Customer_Id, b.PO_Number

from AP_Purchase_Order_Expense a, AP_Purchase_Order b

where a.Purchase_Order_Id <> 1

and b.Purchase_Order_Id <> 1

and a.Purchase_Order_Id = b.Purchase_Order_Id and a.Customer_Id > @customerid

Order by a.Purchase_Order_Expense_Id

--Query 509: AP_Purchase_Order_Expense - Verify Service_Id = 1 - (NO ROWS SHOULD BE RETURNED)--

select distinct a.Purchase_Order_Expense_Id as "Query 509 - AP_Purchase_Order_Expense - Verify Service_Id = 1 (NO ROWS SHOULD BE RETURNED)",

a.Service_Id, b.PO_Number

from AP_Purchase_Order_Expense a, AP_Purchase_Order b

where a.Purchase_Order_Id <> 1

and b.Purchase_Order_Id <> 1

and a.Purchase_Order_Id = b.Purchase_Order_Id

and a.Service_Id <> 1

Order by a.Purchase_Order_Expense_Id

--Query 510: AP_Purchase_Order_Expense - Verify Job_Id = 1 - (NO ROWS SHOULD BE RETURNED)--

select distinct a.Purchase_Order_Expense_Id as "Query 510 - AP_Purchase_Order_Expense - Verify Job_Id = 1 (NO ROWS SHOULD BE RETURNED)",

a.Job_Id, b.PO_Number

from AP_Purchase_Order_Expense a, AP_Purchase_Order b

where a.Purchase_Order_Id <> 1

and b.Purchase_Order_Id <> 1

and a.Purchase_Order_Id = b.Purchase_Order_Id

and a.Job_Id <> 1

Order by a.Purchase_Order_Expense_Id

--Query 511: AP_Purchase_Order_Expense - Verify Quantity > 0 - (NO ROWS SHOULD BE RETURNED)--

select distinct a.Purchase_Order_Expense_Id as "Query 510 - AP_Purchase_Order_Expense - Verify Quantity > 0 (NO ROWS SHOULD BE RETURNED)",

a.Quantity, b.PO_Number

from AP_Purchase_Order_Expense a, AP_Purchase_Order b

where a.Purchase_Order_Id <> 1

and b.Purchase_Order_Id <> 1

and a.Purchase_Order_Id = b.Purchase_Order_Id

and a.Quantity <=0

Order by a.Purchase_Order_Expense_Id

--Query 512: AP_Purchase_Order_Expense - Verify Rate > 0 - (NO ROWS SHOULD BE RETURNED)--

select distinct a.Purchase_Order_Expense_Id as "Query 512 - AP_Purchase_Order_Expense - Verify Rate > 0 (NO ROWS SHOULD BE RETURNED)",

a.Rate, b.PO_Number

from AP_Purchase_Order_Expense a, AP_Purchase_Order b

where a.Purchase_Order_Id <> 1

and b.Purchase_Order_Id <> 1

and a.Purchase_Order_Id = b.Purchase_Order_Id

and a.Rate <=0

Order by a.Purchase_Order_Expense_Id

--*** AP_PURCHASE_ORDER_PARTS FIELD VALIDATION QUERIES***--

--Query 520: AP_Purchase_Order_Parts - Verify a record exists in AP_Purchase_Order - (NO ROWS SHOULD BE RETURNED)--

select distinct Purchase_Order_Parts_Id as "Query 520 - AP_Purchase_Order_Parts - Verify a record exists in AP_Purchase_Order (NO ROWS SHOULD BE RETURNED)"

from AP_Purchase_Order_Parts

where Purchase_Order_Id <> 1

and Purchase_Order_Id not in (select Purchase_Order_Id from ap_purchase_order)

Order by Purchase_Order_Parts_Id

--Query 521: AP_Purchase_Order_Parts - Validate Part_Id - (*Provide results to the customer to review for accuracy*)--

select distinct b.Part_Code as "Query 521 - AP_Purchase_Order_Parts - Validate Part_Id (*Provide results to the customer to review for accuracy*)",

c.PO_Number

from AP_Purchase_Order_Parts a, IN_Part b, AP_Purchase_Order c

where a.Purchase_Order_Id <> 1

and a.Part_Id not in (select part_id from in_part)

and b.Part_Id > @partid

and a.Purchase_Order_Id = c.Purchase_Order_Id

Order by b.Part_Code

--Query 522: AP_Purchase_Order_Parts - Verify Quantity > 0 - (NO ROWS SHOULD BE RETURNED)--

select distinct a.Purchase_Order_Parts_Id as "Query 522 - AP_Purchase_Order_Parts - Verify Quantity > 0 (NO ROWS SHOULD BE RETURNED)",

b.PO_Number, a.Quantity

from AP_Purchase_Order_Parts a, AP_Purchase_Order b

where a.Purchase_Order_Id <> 1

and b.Purchase_Order_Id <> 1

and a.Purchase_Order_Id = b.Purchase_Order_Id

and a.Quantity <=0

Order by a.Purchase_Order_Parts_Id

--Query 523: AP_Purchase_Order_Parts - Verify Unit_Of_Measure_Id = 2 - (NO ROWS SHOULD BE RETURNED)--

select distinct Purchase_Order_Parts_Id as "Query 523 - AP_Purchase_Order_Parts - Verify Unit_Of_Measure_Id = 2 (NO ROWS SHOULD BE RETURNED)",

Unit_Of_Measure_Id

from AP_Purchase_Order_Parts

where Purchase_Order_Id <> 1

and Unit_Of_Measure_Id not in (1,2)

Order by Purchase_Order_Parts_Id

--Query 524: AP_Purchase_Order_Parts - Verify Cost > 0 - (NO ROWS SHOULD BE RETURNED)--

select distinct Purchase_Order_Parts_Id as "Query 524 - AP_Purchase_Order_Parts - Verify Cost > 0 (NO ROWS SHOULD BE RETURNED)",

Cost

from AP_Purchase_Order_Parts

where Purchase_Order_Id <> 1

and Cost <=0

Order by Purchase_Order_Parts_Id

--Query 525: AP_Purchase_Order_Parts - Verify KitId = 0 - (NO ROWS SHOULD BE RETURNED)--

select distinct Purchase_Order_Parts_Id as "Query 525 - AP_Purchase_Order_Parts - Verify KitId = 0 (NO ROWS SHOULD BE RETURNED)",

KitId

from AP_Purchase_Order_Parts

where Purchase_Order_Id <> 1

and KitId <> 0

Order by Purchase_Order_Parts_Id

--Query 526: AP_Purchase_Order_Parts - Verify On_Back_Order = N - (NO ROWS SHOULD BE RETURNED)--

select distinct Purchase_Order_Parts_Id as "Query 526 - AP_Purchase_Order_Parts - Verify On_Back_Order = N (NO ROWS SHOULD BE RETURNED)",

On_Back_Order

from AP_Purchase_Order_Parts

```
where Purchase_Order_Id <> 1

and On_Back_Order <> 'N'

Order by Purchase_Order_Parts_Id
```

```
--Query 527: AP_Purchase_Order_Parts - Verify Phase_Id = 2 - (NO ROWS SHOULD BE RETURNED)--

select distinct Purchase_Order_Parts_Id as "Query 527 - AP_Purchase_Order_Parts - Verify Phase_Id = 2 (NO ROWS SHOULD BE RETURNED)",

Phase_Id

from AP_Purchase_Order_Parts

where Purchase_Order_Id <> 1

and Phase_Id <> 2

Order by Purchase_Order_Parts_Id
```

```
--Query 528: AP_Purchase_Order_Parts - Verify Vendor_Part_Code Not Blank - (NO ROWS SHOULD BE RETURNED)--

select distinct Purchase_Order_Parts_Id as "Query 528 - AP_Purchase_Order_Parts - Verify Vendor_Part_Code Not Blank (NO ROWS SHOULD BE RETURNED)",

Vendor_Part_Code

from AP_Purchase_Order_Parts

where Purchase_Order_Id <> 1

and Vendor_Part_Code = ''

Order by Purchase_Order_Parts_Id
```

--Query 529: AP_Purchase_Order_Parts - Verify Vendor_Part_Description Not Blank - (NO ROWS SHOULD BE RETURNED)--

select distinct Purchase_Order_Parts_Id as "Query 529 - AP_Purchase_Order_Parts - Verify Vendor_Part_Description Not Blank (NO ROWS SHOULD BE RETURNED)",

Vendor_Part_Description

from AP_Purchase_Order_Parts

where Purchase_Order_Id <> 1

and Vendor_Part_Description = ''

Order by Purchase_Order_Parts_Id

--Query 530: AP_Purchase_Order_Parts - Verify IsKit = N - (NO ROWS SHOULD BE RETURNED)--

select distinct Purchase_Order_Parts_Id as "Query 530 - AP_Purchase_Order_Parts - Verify IsKit = N (NO ROWS SHOULD BE RETURNED)",

IsKit

from AP_Purchase_Order_Parts

where Purchase_Order_Id <> 1

and IsKit <> 'N'

Order by Purchase_Order_Parts_Id

--Query 531: AP_Purchase_Order_Parts - Verify Kit_Quantity Not Blank - (NO ROWS SHOULD BE RETURNED)--

select distinct Purchase_Order_Parts_Id as "Query 531 - AP_Purchase_Order_Parts - Verify Kit_Quantity Not Blank (NO ROWS SHOULD BE RETURNED)",

Kit_Quantity

from AP_Purchase_Order_Parts

where Purchase_Order_Id <> 1

and Kit_Quantity <> 0

Order by Purchase_Order_Parts_Id

---SUMMARY LIST OF QUERIES ---

--***CUSTOMER & SITE KEY FIELD VALIDATIONS***

--query 1: Check for customers where the terms are set to the null record (N/A)--

--Query 2: COUNT of customers assigned to each term code --

--Query 3: Check for customer assigned to a Term Code that does not exist in the Terms Table (No Rows Should be Returned)--

--Query 4: Count of Customers in each Branch where the branch is Active--

--Query 5: Count of Customers in each Branch where the branch is Inactive--

--Query 6: Count of Sites in each Branch where the branch is Active--

--Query 7 Count of Sites in each Branch where the branch is Inactive--

--Query 8: Count of Customers by Type of Customer--

--Query 9: Listing of customers assigned to Null customer type (Correct the data records)--

--Query 10: Count of Sites in each Tax Group --

--Query 11: Count of Sites in Null Tax Group (Correct the data records) --

--Query 12: Customers/Sites in Null Tax Group (Correct the data records) --

--Query 13: Count of Sites in each Cycle Tax Group --

--Query 14: Count of Sites in Null Cycle Tax Group --

--Query 15: Customers/Sites in Null Cycle Tax Group (Correct the Data Records) --

--Query 16: Customers/Sites with a blank Site Number (Correct the Data Records) --

--***SYSTEM RECORDS - KEY FIELD VALIDATIONS***

--*** checking warranty, service level, system type, service company, monitoring company, contract form***

--Query 17: Count of Systems by System Type ---

--Query 18: Customers/Sites/Systems with Null System Type (Correct the Data Records) --

--Query 19: Count of Systems with Panel Type ---

--Query 20: Customers/Sites/Systems with Null Panel Type (Fix the Data if any records are returned by the query, if the customer requires a panel type) --

--Query 21: Count of Systems Contract Form Type ---

--Query 22: Customers/Sites/Systems with Null Contract Form Type (Fix the Data if any records are returned by the query, if the customer requires a contract type) --

--Query 23: Count of Systems Warranty Type ---

--Query 24: Customers/Sites/Systems with Null Warranty Type (No Rows Should be Returned) --

--Query 25: Count of Systems by Service Level---

--Query 26: Customers/Sites/Systems with Null Service Level (No Rows Should be Returned) --

--Query 27: Count of Systems by Service Company ---

--Query 28: Customers/Sites/Systems with Null Service Company (Fix the Data if any records are returned by the query) --

--Query 29: Count of Systems by Alarm Company ---

--Query 30: Customers/Sites/Systems with Null Alarm Company (Monitored By) (Fix the Data if any records are returned by the query, if the customer requires the Alarm Company) --

--***RECURRING LINE RECORDS - KEY FIELD VALIDATIONS***

--*** checking invoice item codes, billing cycle***

--Query 31: Count of Recurring Lines by Invoice Item Code ---

--Query 32: Count of Recurring Lines by Invoice Item Code Where the Recurring Line Description is Blank---

--Query 33: Count of Recurring Lines by Invoice Group Number---

--Query 34: Customers/Sites/Systems with Null Invoice Item Code (Fix the Data if any records are returned by the query) --

--Query 35: Count of Recurring Lines by Billing Cycle ---

--Query 36: Recurring Lines Where Cycle Amount =0 (Zero is not a valid value) ---

--Query 37: Recurring Lines Where Cycle Amount =0 Fix the Data - Should not be any RMR Lines with an amount of Zero---

--Query 38: Recurring Lines Where monthly Amount <0 (Not Cancelled) For Review of Valid Free Services ---

--Query 39: Recurring Lines Where Next Cycle Date is < Cycle Start Date (Not Cancelled)---

--*** CUSTOMER GEOGRAPHICAL TABLES (CITY, STATE, ZIP) - KEY FIELD VALIDATIONS***

--Query 40: GE table id 1 validating (GE_Table1 - City) in AR_CUstomer_Bill - Should be zero rows returned ----

--Query 41: GE table id 2 validatiing in AR_Customer_Bill -   (GE_Table2 - State/Province Table) -   Should be zero rows returned ----

--Query 42: GE table id 3 validating in AR_Customer_Bill - (City/State/Zip table id) Should be zero rows returned ----

--Query 43: GE table id 1 Description validations (City Description) Should be zero rows returned ----

--Query 44: GE table id 2 Description validations (State Long Description) Should be zero rows returned ----

--Query 45: GE table id 2 Description validations (State Short Description) Should be zero rows returned ----

--Query 46: GE table id 3 Description validations (Zipcode Description) Should be zero rows returned ----

--*** SITE GEOGRAPHICAL TABLES (CITY, STATE, ZIP) - KEY FIELD VALIDATIONS***

--Query 47: GE table id 1 validations (city table id) ----

--Query 48: GE table id 2 validations (state table id) ----

--Query 49: GE table id 3 validations (City/State/Zip table id) ----

--Query 50: GE table id 1 Description validations (City Description) ----

--Query 51: GE table id 2 Description validations (State Short Description) ----

--Query 52: GE table id 2 Description validations (State Long Description) ----

--Query 53: GE table id 3 Description validations (Zipcode) ----

--Query 54: Validate Site City Name (id matches GE3 Table but City Name does not match) ----

--- Checking for possible misspelled Cities --

--Query 55: Validate Site State (Long Name) (valid ID in GE2 Table but State Long Name does not match) ----

--- Checking for possible misspelled States --

--Query 56: Validate Site State (Short Name) (Valid ID in GE2 Table but State Short Name does not match) ----

--- Checking for possible misspelled States --

--Query 57: Validate Site Zipcode (Valid ID in GE3 Table 3 but Zipcode description does not match) ----

--- Checking for possible incorrect zip/postal codes --

--Query 58: Validate Site Zipcode (Site GE3 id matches GE3 Table but State/Province Long description does not match) ----

--- Checking for possible misspelled Site States (Long State Name) --

--Query 59: Validate Site Zipcode (Site GE3 id matches GE3 Table but State/Province Short description does not match) ----

--- Checking for possible misspelled Site States (Short State Name) --

--Query 60: Zipcodes on Customer Bill To with length less than 5 characters ----

--Query 61: Zipcodes on Site with length less than 5 characters ----

--Query 62: Customers without a Bill To record --

--Query 63: Customers without a Customer Aging record --

--Query 64: Customers without a site record --

--Query 65: Count of Subaccounts for each Master Account to verify the correct number of Subaccounts and the count of Master Accounts expected --

--Query 66: Master Accounts with Dealer ID set to 1 in ar_customer --

--Query 67: Master Accounts without a record in ar_dealer --

--***Open A/R Invoices & Credits KEY FIELD VALIDATIONS***

--***AR_INVOICE KEY FIELD VALIDATIONS*** --

--Query 68: AR_Invoice - Type_JSCO (all should be O) --

--Query 69: AR_Invoice - Category (all should be G&A or something similar) Should not be any assigned to N/A --

--Query 70: AR Setup GL   AR Account Number --

--Query 71: AR_Invoice - AR Account should be the same as ar_setup_gl.ar_id --

--Query 72: AR_Invoice - Amount and Net Due are not equal (should be zero rows returned) --

--Query 73: AR_Invoice - Count of Invoices by Term Code (None should be N/A) --

--Query 74: AR_Invoice - Invoices where Term Code = N/A (Should be zero rows returned) If rows returned, Fix the Data --

--Query 75: AR_Invoice - Count of Invoices by Invoice Description Code (None should be N/A) --

--Query 76: AR_Invoice - Count of Invoices where register_id = 1 (None should = 1) --

--Query 77: AR_Invoice - Count of Invoices where Complete = N (No Rows Should be Returned) --

--Query 78: AR_Invoice - Invoices with an Invalid Bill To record (No Rows Should be Returned) --

--Query 79: AR_Invoice - Invoices Should not be in Print Queue (No Rows Should be Returned) --

--Query 80: AR_Invoice - Invoices Should not be in Email Queue (No Rows Should be Returned)--

--Query 81: AR_Invoice - Count of Invoices by Branch --

--Query 82: AR_Invoice where Credit <> 0 (No Rows Should be Returned)--

--Query 83: AR_Invoice where Payment <> 0 (No Rows Should be Returned)--

--Query 84: AR_Invoice where tax_amount < > 0 (No Rows Should be Returned)--

--Query 85: AR_Invoice where Advance_Deposit < > 0 (No Rows Should be Returned)--

--Query 86: AR_Invoice where Late_Fee_Amt < > 0.00 (No Rows Should be Returned)--

--Query 87: AR_Invoice where GST_Tax_Amount < > 0.00 (No Rows Should be Returned)--

--Query 88: AR_Invoice where IsTaxable < > N (No Rows Should be Returned)--

--Query 89: AR_Invoice where Has_Pending_EFT < > N (No Rows Should be Returned)--

--Query 90: AR_Invoice where Department_Id < > 1 (No Rows Should be Returned)--

--Query 91: AR_Invoice where Job_id < > 1 (No Rows Should be Returned)--

--Query 92: AR_Invoice where Service_Ticket_Id < > 1 (No Rows Should be Returned)--

--Query 93: AR_Invoice where Service_Start_Date < > 1899.. (No Rows Should be Returned)--

--Query 94: AR_Invoice where Service_End_Date < > 1899.. (No Rows Should be Returned)--

--Query 95: AR_Invoice where Payment_Date < > 1899.. (No Rows Should be Returned)--

--*** AR_INVOICE_ITEM KEY VALIDATION FIELDS ** --

--Query 96: AR_Invoice_Item - Count of Invoices by Item Code - (All Item Codes should be the same for data conversion) --

--Query 97: AR_Invoice_Item Where Item Description is blank (No Rows Should be Returned)--

--Query 98: AR_Invoice_Item Where the Quantity <> 1 (No Rows Should be Returned)--

--Query 99: AR_Invoice_Item Where the Rate <> Amount (No Rows Should be Returned)--

--Query 100: AR_Invoice_Item Where the Register ID = 1 (No Rows Should be Returned)--

--Query 101: AR_Invoice_Item Where the Part ID <> 1 (No Rows Should be Returned)--

--Query 102: AR_Invoice_Item Where the Part ID <> 1 (No Rows Should be Returned)--

--Query 103: AR_Invoice_Item Where the Job_Install_Id <> 1 (No Rows Should be Returned)--

--Query 104: AR_Invoice_Item Where the Job_Equipment_Id <> 1 (No Rows Should be Returned)--

--Query 105: AR_Invoice_Item Where the Job_Recurring_Id <> 1 (No Rows Should be Returned)--

--Query 106: AR_Invoice_Item Where the Asset_Register_Id <> 1 (No Rows Should be Returned)--

--Query 107: AR_Invoice_Item Where the COGS_Register_Id <> 1 (No Rows Should be Returned)--

--Query 108: AR_Invoice_Item Where the Journal_Id <> 1 (No Rows Should be Returned)--

--Query 109: AR_Invoice_Item Where the ICO_Register_Id <> 1 (No Rows Should be Returned)--

--Query 110: AR_Invoice_Item Where the Salesperson_Id <> 1 (No Rows Should be Returned)--

--Query 111: AR_Invoice_Item Where the Vendor_Id <> 1 (No Rows Should be Returned)--

--Query 112: AR_Invoice_Item Where the Royalty_Debit_Id <> 1 (No Rows Should be Returned)--

--Query 113: AR_Invoice_Item Where the Royalty_Credit_Id <> 1 (No Rows Should be Returned)--

--Query 114: AR_Invoice_Item Where the Inter_Branch_Register_Id <> 1 (No Rows Should be Returned)--

--Query 115: AR_Invoice_Item Where the Service_Start_Date <> 1 (No Rows Should be Returned)--

--Query 116: AR_Invoice_Item Where the Service_End_Date <> 1 (No Rows Should be Returned)--

--Query 117: AR_Invoice_Item Where the Royalty_End_Date <> 1 (No Rows Should be Returned)--

--Query 118: AR_Invoice_Item Where the Royalty_Start_Date <> 1 (No Rows Should be Returned)--

--Query 119: AR_Invoice_Item Where the RMR_Quantity <> 1 (No Rows Should be Returned)--

--Query 120: AR_Invoice_Item Where the RMR_Quantity_Rate <> 1 (No Rows Should be Returned)--

--Query 121: AR_Invoice_Item Where the Cycle_Amount < > 1 (No Rows Should be Returned)--

--Query 122: AR_Invoice_Item Where the Service_Ticket_Id < > 1 (No Rows Should be Returned)--

--Query 123: AR_Invoice_Item Where the Tax_Exempt < > 1 (No Rows Should be Returned)--

--***AR_CREDIT KEY VALIDATION FIELDS*** --

--Query 124: AR_Credit - Type_JSCO (all should be O) --

--Query 125: AR_Credit - Category (all should be G&A or something similar) Should not be any assigned to N/A --

--Query 126: AR Setup GL Unapplied Credit Memos GL Account Number --

--Query 127: AR_Credit - Unapplied Credit Memos GL Account should be the same as ar_setup_gl.unap_Credit_Id --

--Query 128: AR_Credit - Amount <=0 (should be zero rows returned) --

--Query 129: AR_Credit - Used Amount >0 (should be zero rows returned) --

--Query 130: AR_Credit - Count of Credits by Invoice Description Code (None should be N/A) --

--Query 131: AR_Credit - Count of Credits by Credit Reason Code (None should be N/A) --

--Query 132: AR_Credit - Credits Where the Register ID = 1 (No Rows Should be Returned)--

--Query 133: AR_Credit - Count of Credits where Complete = N (No Rows Should be Returned) --

--Query 134: AR_Credit with an Invalid Bill To record (No Rows Should be Returned) --

--Query 135: AR_Credit - Count of Credits by Branch --

--Query 136: AR_Credit - Count of Credits by Department --

--Query 137: AR_Credit - Count of Credits by Job ID --

--Query 138: AR_Credit - Count of Credits by Tax Amount --

--Query 139: AR_Credit - Count of Credits by Service Start Date --

--Query 140: AR_Credit - Count of Credits by Service End Date --

--Query 141: AR_Credit - Count of Credits by Is Taxable --

--Query 142: AR_Credit - Count of Credits by Service Ticket ID --

--Query 143: AR_Credit - Count of Credits by GST_Tax_Amount --

--Query 144: AR_Credit - Count of Credits by UserCode --

--Query 145: AR_Credit - Count of Credits by Posting_Date --

--** Credit Detail Records (ar_credit_item) ** --

--Query 146: AR_Credit_Item - Count of Credits that do not have a record in ar_credit_item --

--Query 147: AR_Credit_Item - Count of Credits by Item Code - (All Item Codes should be the same for data conversion) --

--Query 148: AR_Credit_Item Where the Item Description is blank (No Rows Should be Returned)--

--Query 149: AR_Credit_Item Where the Rate <> Amount (No Rows Should be Returned)--

--Query 150: AR_Credit_Item Where the Register ID = 1 (No Rows Should be Returned)--

--Query 151: AR_Credit_Item Where the Part ID <> 1 (No Rows Should be Returned)--

--Query 152: AR_Credit_Item Where the Part ID <> 1 (No Rows Should be Returned)--

--Query 153: AR_Credit_Item - Count of Credits by Service Start Date --

--Query 154: AR_Credit_Item - Count of Credits by Service End Date --

--Query 155: AR_Credit_Item Where the Asset Register ID <> 1 (No Rows Should be Returned)--

--Query 156: AR_Credit_Item Where the COGS Register ID <> 1 (No Rows Should be Returned)--

--Query 157: AR_Credit_Item Where the Journal ID <> 1 (No Rows Should be Returned)--

--Query 158: AR_Credit_Item Where the Master Item ID <> 1 (No Rows Should be Returned)--

--Query 159: AR_Credit_Item Where the ICO Register ID <> 1 (No Rows Should be Returned)--

--Query 160: AR_Credit_Item Where the Job Install ID <> 1 (No Rows Should be Returned)--

--Query 161: AR_Credit_Item Where the Job Equipment ID <> 1 (No Rows Should be Returned)--

--Query 162: AR_Credit_Item Where the Job Recurring ID <> 1 (No Rows Should be Returned)--

--Query 163: AR_Credit_Item Where the Salesperson ID <> 1 (No Rows Should be Returned)--

--Query 164: AR_Credit_Item Where the Vendor ID <> 1 (No Rows Should be Returned)--

--Query 165: AR_Credit_Item Where the Royalty Credit ID <> 1 (No Rows Should be Returned)--

--Query 166: AR_Credit_Item Where the Royalty Debit ID <> 1 (No Rows Should be Returned)--

--Query 167: AR_Credit_Item Where the COGS Amount <> 0.00 (No Rows Should be Returned)--

--Query 168: AR_Credit_Item Where the Monthly Royalty <> 0.00 (No Rows Should be Returned)--

--Query 169: AR_Credit_Item Where the Royalty Amount <> 0.00 (No Rows Should be Returned)--

--Query 170: AR_Credit_Item - Count of Credits by Royalty Start Date --

--Query 171: AR_Credit_Item - Count of Credits by Royalty End Date --

--Query 172: AR_Credit_Item - Count of Credits by Cycle Amount (No Rows Should be Returned) --

--Query 173: AR_Credit_Item - Count of Credits by Service Ticket Id (No Rows Should be Returned) --

--Query 174: AR_Credit_Item - Count of Credits by Inter Branch Register Id (No Rows Should be Returned) --

--***AP_VENDOR FIELD VALIDATIONS***--

--Query 175: AR_Credit_Item - Count of Credits by Tax Exempt (No Rows Should be Returned) --

--***AP VENDOR FIELD VALIDATIONS***--

--Query 176: AP_Vendor - Company_Name is Blank (NO ROWS SHOULD BE RETURNED) -

--Query 177: AP_Vendor - Address_1 is Blank -

--Query 178: AP_Vendor - GE_Table1_Id is Invalid (City) (NO ROWS SHOULD BE RETURNED) -

--Query 179: AP_Vendor - GE_Table2_Id is Invalid (State/Province) (NO ROWS SHOULD BE RETURNED) -

--Query 180: AP_Vendor - GE_Table3_Id is Invalid (Zip/Postal Code) (NO ROWS SHOULD BE RETURNED) -

--Query 181: AP_Vendor - Country_Id is Invalid -

--Query 182: AP_Vendor - Pay_From_Account_id - All Should be 1

--Query 183: AP_Vendor - Validate Term_Id

--Query 184: AP_Vendor - Credit_Limit - All Should be 0.00

--Query 185: AP_Vendor - Validate Vendor Type

--Query 186: AP_Vendor - Issue_1099 is Valid - All Should be Y or N

--Query 187: AP_Vendor - Validate All Vendors are Active

--Query 188: AP_Vendor - Checks_Payable_To is Blank (NO ROWS SHOULD BE RETURNED) --

--Query 189: AP_Vendor - Verify Default_Account_Id is Valid --

--Query 190: AP_Vendor - Validate Vendor Branch Code

--Query 191: AP_Vendor - Validate Vendor Category Code

--Query 192: AP_Vendor - Payment Address_1 is Blank -

--Query 193: AP_Vendor - Payment GE_Table1_Id is Invalid (City) (NO ROWS SHOULD BE RETURNED) -

--Query 194: AP_Vendor - Payment GE_Table2_Id is Invalid (State/Province) (NO ROWS SHOULD BE RETURNED) -

--Query 195: AP_Vendor - Payment GE_Table3_Id is Invalid (Zip/Postal Code) (NO ROWS SHOULD BE RETURNED) -

--Query 196: AP_Vendor - GST_Exempt is Valid - All Should be Y or N

--Query 197: AP_Vendor - GE1_Description is Invalid (City) (NO ROWS SHOULD BE RETURNED) -

--Query 198: AP_Vendor - GE2_Description is Invalid (State/Province) (NO ROWS SHOULD BE RETURNED) -

--Query 199: AP_Vendor - GE2_Short is Invalid (State/Province) (NO ROWS SHOULD BE RETURNED) -

--Query 200: AP_Vendor - GE3_Description is Invalid (Zip/Postal Code) (NO ROWS SHOULD BE RETURNED) -

--Query 201: AP_Vendor - Payment GE1_Description is Invalid (City) (NO ROWS SHOULD BE RETURNED) -

--Query 202: AP_Vendor - Payment GE2_Description is Invalid (State/Province) (NO ROWS SHOULD BE RETURNED) -

--Query 203: AP_Vendor - Payment GE2_Short is Invalid (State/Province) (NO ROWS SHOULD BE RETURNED) -

--Query 204: AP_Vendor - Payment GE3_Description is Invalid (Zip/Postal Code) (NO ROWS SHOULD BE RETURNED) -

--Query 205: AP_Vendor - Currency Id is Valid (NO ROWS SHOULD BE RETURNED) --

--Query 206: AP_Vendor - Secure Vendor is Valid - All Should be Y or N

--***AP_INVOICE FIELD VALIDATIONS***--

--Query 207: AP_Invoice - Checking for Duplicate Invoice Numbers per Vendor - (NO ROWS SHOULD BE RETURNED)--

--Query 208: AP_Invoice - Count of Duplicate Invoice Numbers per Vendor - (NO ROWS SHOULD BE RETURNED)--

--Query 209: AP_Invoice - Count of Vendor Bills by Term Code --

--Query 210: AP_Invoice - Count of Vendor Bills by Category Code --

--Query 211: AP_Invoice - Vendor_Id is valid and <> 1 --

--Query 212: AP_Invoice - Amount <=0 --

--Query 213: AP_Invoice - Verify Amount Paid = 0 --

--Query 214: AP_Invoice - Verify Net Due = Amount --

--Query 215: AP_Invoice - Verify Status = O --

--Query 216: AP_Invoice - Verify Hold_Payment = N --

--Query 217: AP_Invoice - Verify Billed = Y --

--Query 218: AP_Invoice - Verify Received = Y --

--Query 219: AP_Invoice - Register_Id <> 1 --

--Query 220: AP_Invoice - Branch Id is Valid --

--Query 221: AP_Invoice - Verify Warehouse Id = 1 --

--Query 222: AP_Invoice - Verify Job Id = 1 --

--Query 223: AP_Invoice - Verify Service Ticket Id = 1 --

--Query 224: AP_Invoice - Verify GST Rate = 0 --

--Query 225: AP_Invoice - Verify GST Amount = 0 --

--Query 226: AP_Invoice - Verify GST Register Id = 1--

--Query 227: AP_Invoice - Verify Purchase Order Id = 1--

--Query 228: AP_Invoice - Verify Repair Order Id = 1--

--Query 229: AP_Invoice - Verify Received Date = 1899-12-30 00:00:00:000 --

--Query 230: AP_Invoice - Verify Received Register Id = 1--

--Query 231: AP_Invoice - Verify Received By = Administrator --

--Query 232: AP_Invoice - Verify Part Total = 0 --

--Query 233: AP_Invoice - Verify Expense Total = 0 --

--Query 234: AP_Invoice - Verify Eligible for Discount = 0 --

--Query 235: AP_Invoice - Currency Id is Valid --

--Query 236: AP_Invoice - Verify Exchange Rate = 1 --

--Query 237: AP_Invoice - Verify GST Branch Amount = 0 --

--Query 238: AP_Invoice - Verify Manual GST = N --

--Query 239: AP_Invoice - Verify Posting Date is Correct --

--Query 240: AP_Invoice - Verify AP_Invoice has a record in AP_Invoice_Expense --

--***AP_INVOICE_EXPENSE FIELD VALIDATIONS***--

--Query 241: AP_Invoice Expense - Verify Account_id is Valid --

--Query 242: AP_Invoice_Expense - Verify Amount > 0 --

--Query 243: AP_Invoice_Expense - Verify Amount = Amount in AP_Invoice --

--Query 244: AP_Invoice_Expense - Verify Description is not Blank --

--Query 245: AP_Invoice_Expense - Verify Customer_Id = 1 --

--Query 246: AP_Invoice_Expense - Verify Job_Id = 1 --

--Query 247: AP_Invoice_Expense - Verify Service_Id = 1 --

--Query 248: AP_Invoice_Expense - Verify Register_Id <> 1 --

--Query 249: AP_Invoice_Expense - Validate Category Id --

--Query 250: AP_Invoice_Expense - Verify Category Id = Category Id in AP_Invoice --

--Query 251: AP_Invoice_Expense - Validate Branch Id --

--Query 252: AP_Invoice_Expense - Verify Branch Id = Branch Id in AP_Invoice --

--Query 253: AP_Invoice_Expense - Verify Inter_Branch_Id = 1 --

--Query 254: AP_Invoice_Expense - Verify PPV_Register_Id = 1 --

--Query 255: AP_Invoice_Expense - Verify Pass_Thru_Item_Id = 1 --

--Query 256: AP_Invoice_Expense - Verify Job_Install_Id = 1 --

--Query 257: AP_Invoice_Expense - Verify GST_Register_Id = 1 --

--Query 258: AP_Invoice_Expense - Verify GST_Inter_Branch_Id = 1 --

--Query 259: AP_Invoice_Expense - Verify Charge_GST = N --

--Query 260: AP_Invoice_Expense - Verify Pass_Thru_Charge = N --

--Query 261: AP_Invoice_Expense - Verify Quantity = 1.00 --

--Query 262: AP_Invoice_Expense - Verify GST_Rate = 0 --

--Query 263: AP_Invoice_Expense - Verify GST_Amount = 0 --

--Query 264: AP_Invoice_Expense - Verify Rate = Amount and <> 0 --

--Query 265: AP_Invoice_Expense - Verify Rate <> 0 --

--Query 266: AP_Invoice_Expense - Validate Expense_Type_id --

--Query 267: AP_Invoice_Expense - Verify Receipt_Expense_Id = 0 --

--***AP_CREDIT FIELD VALIDATIONS***--

--Query 268: AP_Credit - Checking for Duplicate Credit Numbers per Vendor - (NO ROWS SHOULD BE RETURNED)-

-

--Query 269 AP_Credit - Count of Blank Credit Numbers per Vendor - (NO ROWS SHOULD BE RETURNED)--

--Query 270: AP_Credit - Count of Vendor Credits by Category Code --

--Query 271: AP_Credit - Vendor_Id is valid and <> 1 --

--Query 272: AP_Credit - Amount <=0 --

--Query 273: AP_Credit - Verify Amount Used = 0 --

--Query 274: AP_Credit - Verify Balance = Amount --

--Query 275: AP_Credit - Register_Id <> 1 --

--Query 276: AP_Credit - Branch Id is Valid --

--Query 277: AP_Credit - Verify Warehouse Id = 1 --

--Query 278: AP_Credit - Verify Job Id = 1 --

--Query 279: AP_Credit - Verify Service Ticket Id = 1 --

--Query 280: AP_Credit - Verify GST Rate = 0 --

--Query 281: AP_Credit - Verify GST Amount = 0 --

--Query 282: AP_Credit - Verify GST Register Id = 1--

--Query 283 AP_Credit - Verify Purchase Order Id = 1--

--Query 284: AP_Credit - Verify Repair Order Id = 1--

--Query 285: AP_Credit - Verify Part Total = 0 --

--Query 286: AP_Credit - Verify Expense Total = 0 --

--Query 287: AP_Credit - Currency Id is Valid --

--Query 288: AP_Credit - Verify Exchange Rate = 1 --

--Query 289: AP_Credit - Verify GST Branch Amount = 0 --

--Query 290: AP_Credit - Verify Manual GST = N --

--Query 291: AP_Credit - Verify Posting Date is Correct --

-- ** AP_CREDIT_EXPENSE FIELD VALIDATIONS ** --

--Query 292: AP_Credit Expense - Verify Account_id is Valid --

--Query 293: AP_Credit_Expense - Verify Amount > 0 --

--Query 294: AP_Credit_Expense - Verify Amount = Amount in AP_Credit --

--Query 295: AP_Credit_Expense - Verify Description is not Blank --

--Query 296: AP_Credit_Expense - Verify Customer_Id = 1 --

--Query 297: AP_Credit_Expense - Verify Job_Id = 1 --

--Query 298: AP_Credit_Expense - Verify Service_Ticket_Id = 1 --

--Query 299: AP_Credit_Expense - Verify Register_Id <> 1 --

--Query 300: AP_Credit_Expense - Validate Category Id --

--Query 301: AP_Credit_Expense - Verify Category Id = Category Id in AP_Credit --

--Query 302: AP_Credit_Expense - Validate Branch Id --

--Query 303: AP_Credit_Expense - Verify Branch Id = Branch Id in AP_Credit --

--Query 304: AP_Credit_Expense - Verify Inter_Branch_Id = 1 --

--Query 305: AP_Credit_Expense - Verify Pass_Thru_Item_Id = 1 --

--Query 306: AP_Credit_Expense - Verify Job_Install_Id = 1 --

--Query 307: AP_Credit_Expense - Verify GST_Register_Id = 1 --

--Query 308: AP_Credit_Expense - Verify GST_Inter_Branch_Id = 1 --

--Query 309: AP_Credit_Expense - Verify Charge_GST = N --

--Query 310: AP_Credit_Expense - Verify Pass_Thru_Charge = N --

--Query 311: AP_Credit_Expense - Verify Quantity = 1.00 --

--Query 312: AP_Credit_Expense - Verify GST_Rate = 0 --

--Query 313: AP_Credit_Expense - Verify GST_Amount = 0 --

--Query 314: AP_Credit_Expense - Verify Rate = Amount and <> 0 --

--Query 315: AP_Credit_Expense - Verify Rate <> 0 --

--Query 316: AP_Credit_Expense - Validate Expense_Type_id --

--Query 317: AP_Credit_Expense - Verify Receipt_Expense_Id = 0 --

--***SERVICE TICKET FIELD VALIDATIONS***--

--Query 325: SV_Service_Ticket OPEN Tickets - Verify Ticket Status = OP --

--Query 326: SV_Service_Ticket CLOSED Tickets - Verify Ticket Status = CL --

--Query 327: SV_Service_Ticket - Count of Duplicate Ticket Numbers - (NO ROWS SHOULD BE RETURNED)--

--Query 328: SV_Service_Ticket - Ticket Customer Id is Valid - <> 1 - (NO ROWS SHOULD BE RETURNED)--

--Query 329: SV_Service_Ticket - Ticket Customer Site Id is Valid - <> 1 - (NO ROWS SHOULD BE RETURNED)--

--Query 330: SV_Service_Ticket - Ticket Customer System Id is Valid - <> 1 - (NO ROWS SHOULD BE RETURNED)--

--Query 331: SV_Service_Ticket - Verify Ticket Creation Date > 01-02-1900 (NO ROWS SHOULD BE RETURNED)--

--Query 332: SV_Service_Ticket - Ticket Problem Id is Invalid = 1 - (NO ROWS SHOULD BE RETURNED)--

--Query 333: SV_Service_Ticket - Tickets Where Scheduled_For < 01-01-1900 - (NO ROWS SHOULD BE RETURNED)--

--Query 334: SV_Service_Ticket - Tickets Where Last_Service_Tech_Id <>1 - (NO ROWS SHOULD BE RETURNED)--

--Query 335: SV_Service_Ticket - OPEN TICKETS - Ticket Resolution Id is Invalid = 1 - (NO ROWS SHOULD BE RETURNED)--

--Query 336: SV_Service_Ticket - CLOSED TICKETS - Ticket Resolution Id is Invalid = 1 - (NO ROWS SHOULD BE RETURNED)--

--Query 337: SV_Service_Ticket - Tickets Where Billable = N - (NO ROWS SHOULD BE RETURNED)--

--Query 338: SV_Service_Ticket - Tickets Where Equipment_Charge <> 0 - (NO ROWS SHOULD BE RETURNED)",

--Query 339: SV_Service_Ticket - Tickets Where Labor_Charge <> 0 - (NO ROWS SHOULD BE RETURNED)",

--Query 340: SV_Service_Ticket - Tickets Where Other_Charge <> 0 - (NO ROWS SHOULD BE RETURNED)",

--Query 341: SV_Service_Ticket - Tickets Where Tax_Total <> 0 - (NO ROWS SHOULD BE RETURNED)",

--Query 342: SV_Service_Ticket - Tickets Where Regular_Hours <> 0 - (NO ROWS SHOULD BE RETURNED)",

--Query 343: SV_Service_Ticket - Tickets Where Overtime_Hours <> 0 - (NO ROWS SHOULD BE RETURNED)",

--Query 344: SV_Service_Ticket - Tickets Where Holiday_Hours <> 0 - (NO ROWS SHOULD BE RETURNED)",

--Query 345: SV_Service_Ticket - Tickets Where Invoice_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

--Query 346: SV_Service_Ticket - Tickets Where Regular_Rate <> 0 - (NO ROWS SHOULD BE RETURNED)",

--Query 347: SV_Service_Ticket - Tickets Where Overtime_Rate <> 0 - (NO ROWS SHOULD BE RETURNED)",

--Query 348: SV_Service_Ticket - Tickets Where Holiday_Rate <> 0 - (NO ROWS SHOULD BE RETURNED)",

--Query 349: SV_Service_Ticket - Tickets Where Bypass_Warranty <> N - (NO ROWS SHOULD BE RETURNED)",

--Query 350: SV_Service_Ticket - Tickets Where Bypass_Service_Level <> N - (NO ROWS SHOULD BE RETURNED)",

--Query 351: SV_Service_Ticket - Tickets Where IsInspection <> N - (NO ROWS SHOULD BE RETURNED)",

--Query 352: SV_Service_Ticket - Tickets Where Manual_Labor <> N - (NO ROWS SHOULD BE RETURNED)",

--Query 353: SV_Service_Ticket - Tickets Where Remittance <> N - (NO ROWS SHOULD BE RETURNED)",

--Query 354: SV_Service_Ticket - Tickets Where Payment_Received <> N - (NO ROWS SHOULD BE RETURNED)",

--Query 355: SV_Service_Ticket - Tickets Where Sub_Problem_Id = 1 - (NO ROWS SHOULD BE RETURNED)",

--Query 356: SV_Service_Ticket - Tickets Where Entered_By <> 'Administrator' - (NO ROWS SHOULD BE RETURNED)",

--Query 357: SV_Service_Ticket - Tickets Where UserCode <> 'Administrator' - (NO ROWS SHOULD BE RETURNED)",

--Query 358: SV_Service_Ticket - Tickets Where Edit_TimeStamp <> Creation_Date - (NO ROWS SHOULD BE RETURNED)",

--Query 359: SV_Service_Ticket - Tickets Where CustomerComments = '' - (NO ROWS SHOULD BE RETURNED)",

--Query 360: SV_Service_Ticket - Tickets Where Number_Of_Dispatches <> 0 - (NO ROWS SHOULD BE RETURNED)",

--Query 361: SV_Service_Ticket - Tickets Where Customer_CC_Id <> 0 - (NO ROWS SHOULD BE RETURNED)",

--Query 362: SV_Service_Ticket - Tickets Where Customer_Bank_Id <> 0 - (NO ROWS SHOULD BE RETURNED)",

--Query 363: SV_Service_Ticket - Tickets Where Ticket_Status_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

--Query 364: SV_Service_Ticket - Tickets Where Customer_EFT_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

--Query 365: SV_Service_Ticket - Tickets Where Inspection_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

--Query 366: SV_Service_Ticket - Tickets Where Service_Ticket_Group_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

--Query 367: SV_Service_Ticket - Tickets Where Service_Coordinator_Employee_Id <> 1 - (NO ROWS SHOULD BE RETURNED)",

--Query 368: SV_Service_Ticket - Tickets Where Inspection_Incremented <> N - (NO ROWS SHOULD BE RETURNED)",

--Query 369: SV_Service_Ticket - Tickets Where ByPass_TicketServiceCompany <> N - (NO ROWS SHOULD BE RETURNED)",

--Query 370: SV_Service_Ticket - Tickets Where Estimated_Length <> SV_Problem.estimated_time - (NO ROWS SHOULD BE RETURNED)",

--Query 371: SV_Service_Ticket - Tickets Where Priority_Id <> SV_Problem.Priority_Id - (NO ROWS SHOULD BE RETURNED)",

--Query 372: SV_Service_Ticket - Tickets Where Expertise_Level <> SV_Problem.Expertise_Level - (NO ROWS SHOULD BE RETURNED)",

--Query 373: SV_Service_Ticket - Tickets Where Trip_Charge <> System Service_Level - (NO ROWS SHOULD BE RETURNED)",

--Query 374: SV_Service_Ticket - Tickets Where Ticket Service_Company_Id <> System Service_Company_Id - (NO ROWS SHOULD BE RETURNED)",

--Query 375: SV_Service_Ticket - Validate Category_Id - (NO ROWS SHOULD BE RETURNED)--

--Query 376: SV_Service_Ticket - Validate Customer_Bill_Id - (NO ROWS SHOULD BE RETURNED)--

--Query 377: SV_Service_Ticket - OPEN TICKETS Where Resolved_Date <> 1899... - (NO ROWS SHOULD BE RETURNED)",

--Query 378: SV_Service_Ticket - CLOSED TICKETS Where Resolved_Date < 1-1-1900 - (NO ROWS SHOULD BE RETURNED)",

--Query 379: SV_Service_Ticket - Validate Route_Id - (NO ROWS SHOULD BE RETURNED)--

--Query 380: SV_Service_Ticket - CLOSED TICKETS Where ClosedDate < 1-1-1900 - (NO ROWS SHOULD BE RETURNED)",

--*** IN_PART FIELD VALIDATION QUERIES***--

--Query 400: IN_Part - Checking for Duplicate Part Codes - (NO ROWS SHOULD BE RETURNED)--

--Query 401: IN_Part - Checking for Blank Description - (NO ROWS SHOULD BE RETURNED)--

--Query 402: IN_Part - Checking for Blank Detail - (NO ROWS SHOULD BE RETURNED)--

--Query 403: IN_Part - Checking for Unit_Of_Measure_Id <> 2 - (NO ROWS SHOULD BE RETURNED)--

--Query 404: IN_Part - Checking for Non_Value_Part <> N - (NO ROWS SHOULD BE RETURNED)--

--Query 405: IN_Part - Checking for Purchase_UOM_Id <> 2 - (NO ROWS SHOULD BE RETURNED)--

--Query 406: IN_Part - Checking for Issue_UOM_Id <> 2 - (NO ROWS SHOULD BE RETURNED)--

--Query 407: IN_Part - Checking If Service_Price =0 0 - (NO ROWS SHOULD BE RETURNED)--

--Query 408: IN_Part - Checking for Blank Service_Description - (NO ROWS SHOULD BE RETURNED)--

--Query 409: IN_Part - Checking If Sales_Price = 0 - (NO ROWS SHOULD BE RETURNED)--

--Query 410: IN_Part - Checking for Blank Sales_Description - (NO ROWS SHOULD BE RETURNED)--

--Query 411: IN_Part - Checking for Inactive = Y - (NO ROWS SHOULD BE RETURNED)--

--Query 412: IN_Part - Checking for Service_Part = N - (NO ROWS SHOULD BE RETURNED)--

--Query 413: IN_Part - Checking for Sales_Part = N - (NO ROWS SHOULD BE RETURNED)--

--Query 414: IN_Part - Checking for Special_Order = Y - (*Provide results to the customer to review*)--

--Query 415: IN_Part - Checking for Dir_Expense_Account_Id <> 1 - (*Provide results to the customer to review*)--

--Query 416: IN_Part - Checking for Customer_Equipment_Breakout <> N - (NO ROWS SHOULD BE RETURNED)--

--Query 417: IN_Part - Checking for Freeze_Purchasing <> N - (NO ROWS SHOULD BE RETURNED)--

--Query 418: IN_Part - Checking for Income_Account_Id <> 1 - (*Provide results to the customer to review*)--

--Query 419: IN_Part - Validate Item_Id <> 1 - (NO ROWS SHOULD BE RETURNED)--

--Query 420: IN_Part - Validate Product_Line_Id <> 1 - (NO ROWS SHOULD BE RETURNED)--

--Query 421: IN_Part - Count of Part Code by Manufacturer_Code - (*Provide results to the customer to review*)--

--Query 422: IN_Part - Checking for PPV_Account_Id <> 1 - (NO ROWS SHOULD BE RETURNED)--

--Query 423: IN_Part - Count of Part Code by Job_Use_default - (*Provide results to the customer to review*)--

--Query 424: IN_Part - Job_Use_default - ((*Provide results to the customer to review*)--

--Query 425: IN_Part - Count of Part Code by Customer_Equipment - (*Provide results to the customer to review*)--

--Query 426: IN_Part - Part Code Where Customer_Equipment = N - (*Provide results to the customer to review*)--

--Query 427: IN_Part - Validate Purchase_Description is Not Blank - (NO ROWS SHOULD BE RETURNED)--

--Query 428: IN_Part - Validate Part Purchase_Description = Part Supplier Description - (NO ROWS SHOULD BE RETURNED)--

--Query 429: IN_Part - Validate Primary Vendor_Id - (NO ROWS SHOULD BE RETURNED)--

--Query 430: IN_Part - Validate Part Primary Vendor_Id - (NO ROWS SHOULD BE RETURNED)--

--Query 431: IN_Part - Validate Part Vendor_Part = Supplier Vendor_Part_Number - (NO ROWS SHOULD BE RETURNED)--

--*** IN_PART_SUPPLIER FIELD VALIDATION QUERIES***--

--Query 440: IN_Part_Supplier - Validate Vendor_Id - (NO ROWS SHOULD BE RETURNED)--

--Query 441: IN_Part_Supplier - Validate Vendor_Part_Number Not Blank - (NO ROWS SHOULD BE RETURNED)--

--Query 442: IN_Part_Supplier - Validate Description Not Blank - (NO ROWS SHOULD BE RETURNED)--

--Query 443: IN_Part_Supplier - Validate Cost <> 0 - (NO ROWS SHOULD BE RETURNED)--

--Query 444: IN_Part_Supplier - Validate List_Price <> 0 - (NO ROWS SHOULD BE RETURNED)--

--Query 445: IN_Part_Supplier - Validate Quantity <> 0 - (NO ROWS SHOULD BE RETURNED)--

--*** AP_PURCHASE_ORDER FIELD VALIDATION QUERIES***--

--Query 460: AP_Purchase_Order - Checking for Duplicate PO Numbers - (NO ROWS SHOULD BE RETURNED)--

--Query 461: AP_Purchase_Order - Verify Type_JSO = O - (NO ROWS SHOULD BE RETURNED)--

--Query 462: AP_Purchase_Order - Verify Job_Id = 1 - (NO ROWS SHOULD BE RETURNED)--

--Query 463: AP_Purchase_Order - Verify Service_Id = 1 - (NO ROWS SHOULD BE RETURNED)--

--Query 464: AP_Purchase_Order - Verify Closed = N - (NO ROWS SHOULD BE RETURNED)--

--Query 465: AP_Purchase_Order - Verify On_Back_Order = N - (NO ROWS SHOULD BE RETURNED)--

--Query 466: AP_Purchase_Order - Verify UserCode = 'Administrator' - (NO ROWS SHOULD BE RETURNED)--

--Query 467: AP_Purchase_Order - Verify Edit_UserCode = 'Administrator' - (NO ROWS SHOULD BE RETURNED)--

--Query 468: AP_Purchase_Order - Verify Exchange_Rate = 1.00 - (NO ROWS SHOULD BE RETURNED)--

--Query 469: AP_Purchase_Order - Verify Vendor_Acknowledged = 0 - (NO ROWS SHOULD BE RETURNED)--

--Query 470: AP_Purchase_Order - Verify Integration_Status = O - (NO ROWS SHOULD BE RETURNED)--

--Query 471: AP_Purchase_Order - Verify Vendor_Acknowledged_Date = 1899... - (NO ROWS SHOULD BE RETURNED)--

--Query 472: AP_Purchase_Order - Verify Due_Date = 1899... - (NO ROWS SHOULD BE RETURNED)--

--Query 473: AP_Purchase_Order - Verify Shipping_Date = 1899... - (NO ROWS SHOULD BE RETURNED)--

--Query 474: AP_Purchase_Order - Verify Acknowledged_Date = 1899... - (NO ROWS SHOULD BE RETURNED)--

--Query 475: AP_Purchase_Order - Verify Shipping_Address not Blank if Warehouse_Id >1 - (NO ROWS SHOULD BE RETURNED)--

--Query 476: AP_Purchase_Order - Validate Category_Id - (NO ROWS SHOULD BE RETURNED)--

--Query 477: AP_Purchase_Order - Validate Vendor_Id - (NO ROWS SHOULD BE RETURNED)--

--Query 478: AP_Purchase_Order - Validate Shipping_Method_Id - (NO ROWS SHOULD BE RETURNED)--

--Query 479: AP_Purchase_Order - Validate Warehouse_Id - (NO ROWS SHOULD BE RETURNED)--

--Query 480: AP_Purchase_Order - Validate Branch_Id - (NO ROWS SHOULD BE RETURNED)--

--Query 481: AP_Purchase_Order - Verify Order_Date > 1-1-1900 - (NO ROWS SHOULD BE RETURNED)--

--Query 482: AP_Purchase_Order - Verify Total_Cost > 0 - (NO ROWS SHOULD BE RETURNED)--

--Query 483: AP_Purchase_Order - Verify Part_Total + Expense_Total = Total_Cost - (NO ROWS SHOULD BE RETURNED)--

--Query 484: AP_Purchase_Order - Verify Direct_Expense = Y when Warehouse_Id = 1 - (NO ROWS SHOULD BE RETURNED)--

--Query 485 AP_Purchase_Order - Verify Entered_Date > 1-1-1900 - (NO ROWS SHOULD BE RETURNED)--

--Query 486: AP_Purchase_Order - Validate Currency_Id - (NO ROWS SHOULD BE RETURNED)--

--*** AP_PURCHASE_ORDER_EXPENSE FIELD VALIDATION QUERIES***--

--Query 500: AP_Purchase_Order_Expense - Verify a record exists in AP_Purchase_Order - (NO ROWS SHOULD BE RETURNED)--

--Query 501: AP_Purchase_Order_Expense - Validate Account_Id - (*Provide results to the customer to review for accuracy*)--

--Query 502: AP_Purchase_Order_Expense - Validate Category_Id - (*Provide results to the customer to review for accuracy*)--

--Query 503: AP_Purchase_Order_Expense - Validate Category_Id - (*Provide results to the customer to review for accuracy*)--

--Query 504: AP_Purchase_Order_Expense - Validate Expense_Type_Id - (NO ROWS SHOULD BE RETURNED)--

--Query 505: AP_Purchase_Order_Expense - Validate Expense_Type_Id - (NO ROWS SHOULD BE RETURNED)--

--Query 506: AP_Purchase_Order_Expense - Verify Amount = Quantity x Rate - (NO ROWS SHOULD BE RETURNED)--

--Query 507: AP_Purchase_Order_Expense - Verify Description Not Blank - (NO ROWS SHOULD BE RETURNED)--

--Query 508: AP_Purchase_Order_Expense - Verify Customer_Id = 1 - (NO ROWS SHOULD BE RETURNED)--

--Query 509: AP_Purchase_Order_Expense - Verify Service_Id = 1 - (NO ROWS SHOULD BE RETURNED)--

--Query 510: AP_Purchase_Order_Expense - Verify Job_Id = 1 - (NO ROWS SHOULD BE RETURNED)--

--Query 511: AP_Purchase_Order_Expense - Verify Quantity > 0 - (NO ROWS SHOULD BE RETURNED)--

--Query 512: AP_Purchase_Order_Expense - Verify Rate > 0 - (NO ROWS SHOULD BE RETURNED)--

--*** AP_PURCHASE_ORDER_PARTS FIELD VALIDATION QUERIES***--

--Query 520: AP_Purchase_Order_Parts - Verify a record exists in AP_Purchase_Order - (NO ROWS SHOULD BE RETURNED)--

--Query 521: AP_Purchase_Order_Parts - Validate Part_Id - (*Provide results to the customer to review for accuracy*)--

--Query 522: AP_Purchase_Order_Parts - Verify Quantity > 0 - (NO ROWS SHOULD BE RETURNED)--

--Query 523: AP_Purchase_Order_Parts - Verify Unit_Of_Measure_Id = 2 - (NO ROWS SHOULD BE RETURNED)--

--Query 524: AP_Purchase_Order_Parts - Verify Cost > 0 - (NO ROWS SHOULD BE RETURNED)--

--Query 525: AP_Purchase_Order_Parts - Verify KitId = 0 - (NO ROWS SHOULD BE RETURNED)--

--Query 526: AP_Purchase_Order_Parts - Verify On_Back_Order = N - (NO ROWS SHOULD BE RETURNED)--

--Query 527: AP_Purchase_Order_Parts - Verify Phase_Id = 2 - (NO ROWS SHOULD BE RETURNED)--

--Query 528: AP_Purchase_Order_Parts - Verify Vendor_Part_Code Not Blank - (NO ROWS SHOULD BE RETURNED)--

--Query 529: AP_Purchase_Order_Parts - Verify Vendor_Part_Description Not Blank - (NO ROWS SHOULD BE RETURNED)--

--Query 530: AP_Purchase_Order_Parts - Verify IsKit = N - (NO ROWS SHOULD BE RETURNED)--

--Query 531: AP_Purchase_Order_Parts - Verify Kit_Quantity Not Blank - (NO ROWS SHOULD BE RETURNED)--

END