

Practical SQL for the A/P Functional User

This document contains several practical SQL scripts for use with the Oracle Payables Application.

Warning: Oracle Corporation does not condone the direct manipulation of Oracle Application tables through the use of SQL or any other programmatic method. Doing so may void your Oracle Support agreement.

Disclaimer: These scripts are for educational purposes only and carry no warranties.

Below is a breakdown of the SQL scripts.

Finding Dead Vendors – vendors with no invoice or PO activity or only activity is cancelled invoices

- Vendor list of vendors never used - no Invoices and no PO's
- Vendor list of vendors with only activity being cancelled invoices
- Vendors with % in their names

Vendor Merge assistance

- Vendor list sorted by Zip Code, street address
- Duplicate vendor listing by Zip Code/street address
- Fix duplicate invoices preventing a vendor merge
- Vendor Unmerge

Vendor Site Delete

- Delete unused vendor sites

Finding Dead Vendors

During initial Oracle Payables implementation typically all vendors in the old system are copied over to Oracle. Many of these vendors are never used in Oracle.

How can you get a list of them?

Vendors Never Used

```
--Vendors with no PO's nor Invoices

select vend.segment1 Vendor#,
       vendor_name,
       sites.vendor_site_code,
       vend.end_date_active VEND_INACTIVE,
       sites.inactive_date SITE_INACTIVE
from po_vendors vend,
     po_vendor_sites_all sites,
     ap_invoices_all inv,
     po_headers_all PO
where  vend.vendor_id = sites.vendor_id
and    sites.vendor_site_id = inv.vendor_site_id(+)
and    inv.vendor_site_id is null
and    sites.vendor_site_id = po.vendor_site_id(+)
and    po.vendor_site_id is null
order by
       vendor_name;
```

```
--Vendors with no Invoices

select segment1 Vendor#,
       vendor_name,
       vendor_site_code,
       end_date_active VEND_INACTIVE,
       inactive_date SITE_INACTIVE
from po_vendors vend,
     po_vendor_sites_all sites,
     ap_invoices_all inv
where  vend.vendor_id = sites.vendor_id
and    sites.vendor_site_id = inv.vendor_site_id(+)
and    inv.vendor_site_id is null
order by
       vendor_name;
```

Vendors with Only Cancelled Invoices

--List of Vendor sites with cancelled invoices as only activity

```
select vend.vendor_name,
       site.vendor_site_code,
       site.inactive_date
from po_vendors vend,
     po_vendor_sites_all site
where   vend.vendor_id = site.vendor_id
and     exists
-- Cancelled invoices
      (select 'a'
       from ap_invoices_all inall
       where site.vendor_site_id = inall.vendor_site_id
           and inall.cancelled_date is not null)
and     not exists
-- Good invoices
      (select 'a'
       from ap_invoices_all inall
       where site.vendor_site_id = inall.vendor_site_id
           and inall.cancelled_date is null)
order by
       vendor_name;
```

Vendors with % in their names

Many times vendors are accidentally created when the user thinks they are in query mode. This results in a new vendor ending with the infamous % sign. These are difficult to find because you cannot query with the % being one of the meaningful search characters.

This query will find all of those vendors.

```
-- Find Vendors with % in the Vendor Name

select vendor_name,
       segment1 vendor_number
from po_vendors
where  vendor_name like '%/%%' escape '/';
```

Vendor Merge Assistance

Vendor list sorted by Zip Code, street address

Oracle provides a report listing vendors with the first xx characters of the vendor name being the same. This is intended to assist in finding duplicate vendors. The problem with this report is the fewer numbers you use to run the report the more similar names you get. The larger number skips over many of the duplicates. Also, it cannot identify duplicates where the names are not even similar.

The best way to find duplicate vendors is to use the street address. This report lists all vendors sorted by zip code and street address. It will generate a large report if you have a large number of vendors. It also requires that you visually review the report for potential duplicates – it is only a listing.

--Vendor Listing by zip code and street address

```
select vend.vendor_name VENDOR_NAME,
       vend.segment1 VENDOR_NUM,
       site.vendor_site_code SITE_CODE,
       site.address_line1 STREET,
       site.city CITY,
       site.zip ZIP
from po_vendors vend,
     po_vendor_sites_all site
where vend.vendor_id = site.vendor_id
order by
       site.zip,
       site.address_line1;
```

Duplicate Vendor List

This report looks for all vendors having the same zip code and the first xx characters of the street address being the same.

Change the number as indicated to change the number of characters compared in the street address – currently set at 20

Add additional Where clauses to exclude common addresses such as your company's places of business where employees are likely to receive expense checks.

Note: Each pair of duplicates will be listed twice. I tried to write this so the duplicate pair would not be included but the script ran too long. I'm sure there is someone out there who can tune this. I would appreciate a copy if someone does.

Example:

<u>Original</u>	<u>Duplicate</u>
John Smith	Smith, John
Smith, John	John Smith

```
--Duplicate Vendor List
select pv1.vendor_name,
       pv2.vendor_name DUP_VENDOR_NAME,
       pv1.segment1 VENDOR_ID,
       pv2.segment1 DUP_VENDOR_ID,
       pvsa1.vendor_site_code SITE_CODE,
       pvsa2.vendor_site_code SITE_CODE,
       pvsa1.address_line1,
       pvsa2.address_line1 DUP_ADDRESS_LINE1,
       pvsa1.zip
from   po_vendors pv1,
       po_vendors pv2,
       po_vendor_sites_all pvsa1,
       po_vendor_sites_all pvsa2
where  pvsa1.vendor_site_id <> pvsa2.vendor_site_id
--Change the last number at the end of the next two
--lines to the number of characters to be tested in
--the address line for duplicates                xx
and    substr(replace(pvsa1.address_line1, ' '),1,20) =
       substr(replace(pvsa2.address_line1, ' '),1,20)
and    pvsa1.zip = pvsa2.zip
and    pv1.vendor_id = pvsa1.vendor_id
and    pv2.vendor_id = pvsa2.vendor_id
and    pv1.vendor_id <> pv2.vendor_id
--Below: Enter street addresses to be ignored
--Common addresses are employee offices
and    pvsa1.address_line1 <> '123 Main Street'
order by 1;
```

Fix Duplicate Invoices

Often you are prevented from running vendor merge because the two vendors have common invoice numbers that will create a duplicate invoice situation if merged. Typically you have posted and paid these invoices so it is difficult (and tedious) to change them.

Below are 2 scripts. One will list these potential duplicates and the other will modify them by appending "-Merge" to the end of the invoice to be merged.

Prerequisite: You must have already entered the vendor merge data using Oracle Vendor merge form. This program uses the table AP_DUPLICATE_VENDORS_ALL to determine what vendor invoices may be in duplicate. It will search this table for any vendors that have not been processed and look for duplicate invoices. It will then append the "-Merge" to the invoice number so it will now be unique.

```
--List duplicate invoice numbers to be changed so the
--vendors can be merged
select dupl.entry_id,
       inv1.invoice_num,
       inv1.invoice_num||' -Merge' ||to_char(dupl.entry_id) NEW_NUM,
       vend1.vendor_name GOOD_VENDOR,
       site1.vendor_site_code GOOD_SITE,
       vend2.vendor_name DUP_VENDOR,
       site2.vendor_site_code DUP_SITE
from   ap_duplicate_vendors_all dupl,
       ap_duplicate_vendors_all dup2,
       ap_invoices_all inv1,
       po_vendors vend1,
       po_vendor_sites_all site1,
       ap_invoices_all inv2,
       po_vendors vend2,
       po_vendor_sites_all site2
where  dupl.process_flag = 'N'
and    dupl.entry_id = dup2.entry_id
and    inv1.vendor_id = dupl.vendor_id
and    inv1.vendor_id = vend1.vendor_id
and    dupl.vendor_site_id = site1.vendor_site_id
and    inv2.vendor_id = dup2.duplicate_vendor_id
and    inv2.vendor_id = vend2.vendor_id
and    dup2.duplicate_vendor_site_id = site2.vendor_site_id
and    dup2.duplicate_vendor_site_id = inv2.vendor_site_id
and    inv1.invoice_num = inv2.invoice_num;
```

```

-- Change invoice number for invoices that are duplicates
-- in the vendor merge process
update ap_invoices_all inv
set
    inv.invoice_num =
    (select inv2.invoice_num || '-Merge' || to_char(dup1.entry_id)
     from ap_duplicate_vendors_all dup1,
          ap_duplicate_vendors_all dup2,
          ap_invoices_all inv1,
          ap_invoices_all inv2
     where
        dup1.process_flag = 'N'
        and dup1.entry_id = dup2.entry_id
        and inv1.vendor_id = dup1.vendor_id
        and inv2.vendor_id = dup2.duplicate_vendor_id
        and dup2.duplicate_vendor_site_id = inv2.vendor_site_id
        and inv1.invoice_num = inv2.invoice_num
        and inv.invoice_id = inv2.invoice_id
        and inv1.invoice_id != inv2.invoice_id)
where
    inv.invoice_id in
    (select inv2.invoice_id
     from ap_duplicate_vendors_all dup1,
          ap_duplicate_vendors_all dup2,
          ap_invoices_all inv1,
          ap_invoices_all inv2
     where
        dup1.process_flag = 'N'
        and dup1.entry_id = dup2.entry_id
        and inv1.vendor_id = dup1.vendor_id
        and inv2.vendor_id = dup2.duplicate_vendor_id
        and dup2.duplicate_vendor_site_id = inv2.vendor_site_id
        and inv1.invoice_num = inv2.invoice_num);

commit;

```

Vendor Unmerge

On some occasions users find they have accidentally, in error, intentionally, ... merged the wrong vendors. Once this has happened there is now Oracle supported remedy. In the past most people have said it is impossible to find all of the original invoices that were merged even if you knew the original vendor/sites. This is because the invoice now looks like it was originally keyed to the new vendor/site.

There is now a solution. There is a little know table that keeps a history of all merged and scheduled to be merged vendors – AP_DUPLICATE_VENDORS_ALL. Although a list of merged invoices is not kept there is a way to find them.

The vendor merge program uses the AP_DUPLICATE_VENDORS_ALL table as a workbench to determine what invoices are to be moved to another vendor. At the time of commit, Oracle places the system date in the LAST_UPDATE_DATE column of the invoice record and the vendor/site duplicate information in the AP_DUPLICATE_VENDORS_ALL record. The date is stored in day/month/year/hour/minute/second format. This means the invoice last update date and the program run time are exactly the same. The chances of any other invoices having been merged with the same origin and destination vendor/sites are quite remote. This makes it easy to find the merged invoices.

Prerequisites: The invoice must not have been updated since the vendor merge process. This is unlikely for these types of invoices. If it has be "updated" it will be passed over for unmerging.

You must know the origin and destination vendor/site pairs.

This pair must have been merged at one time.

The origin and destination vendor/site pairs are hard coded in the SQL code and must be changed each time. A qualified developer can modify this code to permit request parameters at run time.

There are 2 unmerge scripts below. One if for invoices and the other is for PO's. If you do not use purchasing then you can ignore the second one.

The merge process inactivates the vendor site by putting the system date in the inactive date field. You will need to remove this date if you want to use this site after unmerging the vendor.

```

-- SQL script to show source and listing of invoices to
-- be "Unmerged"
-- VEND1 and SITE1 are the merged to vendor and site
-- VEND1 and SITE1 are where the invoices are now
-- VEND2 and SITE2 are the Original vendor and site prior
-- to the merge
SELECT VEND1.VENDOR_NAME FROM_VENDOR,
       SITE1.VENDOR_SITE_CODE FROM_SITE,
       VEND2.VENDOR_NAME TO_VENDOR,
       SITE2.VENDOR_SITE_CODE TO_SITE, INVOICE_DATE,
       INVOICE_NUM,
       TO_CHAR(INVOICE_AMOUNT, '9999999.99') INVOICE_AMOUNT
FROM PO_VENDORS VEND1,
     PO_VENDOR_SITES_ALL SITE1,
     PO_VENDORS VEND2,
     PO_VENDOR_SITES_ALL SITE2,
     AP_INVOICES_ALL INV,
     AP_DUPLICATE_VENDORS_ALL DUPS
WHERE
       VEND1.VENDOR_ID = SITE1.VENDOR_ID
AND    VEND2.VENDOR_ID = SITE2.VENDOR_ID
-- There must be a record on file of the original merge
AND    VEND1.VENDOR_ID = DUPS.VENDOR_ID
AND    VEND2.VENDOR_ID = DUPS.DUPLICATE_VENDOR_ID
AND    SITE1.VENDOR_SITE_ID = DUPS.VENDOR_SITE_ID
AND    SITE2.VENDOR_SITE_ID = DUPS.DUPLICATE_VENDOR_SITE_ID
-- select invoices in the FROM_VENDOR_SITE
AND    SITE1.VENDOR_SITE_ID = INV.VENDOR_SITE_ID
-- Vendor/site selection criteria
-- VEND1 and SITE1 are where the invoices are now
AND    VEND1.VENDOR_NAME = 'SMITH, JOHN'
AND    SITE1.VENDOR_SITE_CODE = 'ATLANTA'
-- VEND2 and SITE2 are where the invoices were before the merge
AND    VEND2.VENDOR_NAME = 'JOHN SMITH'
AND    SITE2.VENDOR_SITE_CODE = 'DECATUR'
-- Look for invoices with same LAST_UPDATE_DATE as the date
-- of the original vendor merge.
-- Merge record stored in AP_DUPLICATE_VENDORS
AND    TO_DATE(INV.LAST_UPDATE_DATE) =
       TO_DATE(DUPS.LAST_UPDATE_DATE)
ORDER BY
       INVOICE_NUM;

```

```

--SQL script to "Unmerge Invoices"
UPDATE AP_INVOICES_ALL
SET     VENDOR_ID =
        (SELECT VEND.VENDOR_ID
         FROM PO_VENDORS VEND
-- Where the invoices were originally
        WHERE VENDOR_NAME = 'JOHN SMITH'),
        VENDOR_SITE_ID =
        (SELECT SITE.VENDOR_SITE_ID
         FROM PO_VENDOR_SITES_ALL SITE,
              PO_VENDORS VEND
        WHERE VEND.VENDOR_ID = SITE.VENDOR_ID
-- Where the invoices were originally
        AND VENDOR_NAME = 'JOHN SMITH'
        AND VENDOR_SITE_CODE = 'DECATUR')
WHERE
        VENDOR_ID =
        (SELECT VEND.VENDOR_ID
         FROM PO_VENDORS VEND
-- Where the invoices are now
        WHERE VENDOR_NAME = 'SMITH, JOHN')
AND     VENDOR_SITE_ID =
        (SELECT SITE.VENDOR_SITE_ID
         FROM PO_VENDOR_SITES_ALL SITE, PO_VENDORS VEND
        WHERE VEND.VENDOR_ID = SITE.VENDOR_ID
-- Where the invoices are now
        AND VENDOR_NAME = 'SMITH, JOHN'
        AND VENDOR_SITE_CODE = 'ATLANTA')
-- Look for invoices with same LAST_UPDATE_DATE as the date
-- of the original vendor merge.
-- Merge record stored in AP_DUPLICATE_VENDORS
AND     TO_DATE(LAST_UPDATE_DATE) =
        (SELECT DISTINCT(TO_DATE(DUPS.LAST_UPDATE_DATE))
         FROM AP_DUPLICATE_VENDORS_ALL DUPS,
              PO_VENDORS VEND1,
              PO_VENDOR_SITES_ALL SITE1,
              PO_VENDORS VEND2,
              PO_VENDOR_SITES_ALL SITE2,
              AP_INVOICES_ALL INV
        WHERE VEND1.VENDOR_ID = SITE1.VENDOR_ID
        AND  VEND2.VENDOR_ID = SITE2.VENDOR_ID
        AND  VEND1.VENDOR_ID = DUPS.VENDOR_ID
        AND  VEND2.VENDOR_ID = DUPS.DUPLICATE_VENDOR_ID
        AND  SITE1.VENDOR_SITE_ID = DUPS.VENDOR_SITE_ID
        AND  SITE2.VENDOR_SITE_ID =
              DUPS.DUPLICATE_VENDOR_SITE_ID
        AND  SITE1.VENDOR_SITE_ID = INV.VENDOR_SITE_ID
-- VEND1 and SITE1 are where the invoices are now
        AND  VEND1.VENDOR_NAME = 'SMITH, JOHN'
        AND  SITE1.VENDOR_SITE_CODE = 'ATLANTA')
-- VEND2 and SITE2 are where the invoices were before the merge
        AND  VEND2.VENDOR_NAME = 'JOHN SMITH'

```

```

        AND SITE2.VENDOR_SITE_CODE = 'DECATUR');
-- SQL script to "Unmerge PO'S"
UPDATE PO_HEADERS_ALL
SET     VENDOR_ID =
        (SELECT VEND.VENDOR_ID
         FROM PO_VENDORS VEND
-- Where the invoices were originally
         WHERE VENDOR_NAME = 'JOHN SMITH'),
        VENDOR_SITE_ID =
        (SELECT SITE.VENDOR_SITE_ID
         FROM PO_VENDOR_SITES_ALL SITE,
              PO_VENDORS VEND
         WHERE VEND.VENDOR_ID = SITE.VENDOR_ID
-- Where the invoices were originally
         AND VENDOR_NAME = 'JOHN SMITH'
         AND VENDOR_SITE_CODE = 'DECATUR')
WHERE VENDOR_ID =
        (SELECT VEND.VENDOR_ID
         FROM PO_VENDORS VEND
-- Where the invoices are now
         WHERE VENDOR_NAME = 'SMITH, JOHN')
AND     VENDOR_SITE_ID =
        (SELECT SITE.VENDOR_SITE_ID
         FROM PO_VENDOR_SITES_ALL SITE, PO_VENDORS VEND
         WHERE VEND.VENDOR_ID = SITE.VENDOR_ID
-- Where the invoices are now
         AND VENDOR_NAME = 'SMITH, JOHN'
         AND VENDOR_SITE_CODE = 'ATLANTA')
AND     TO_DATE(LAST_UPDATE_DATE) =
        (SELECT DISTINCT(TO_DATE(DUPS.LAST_UPDATE_DATE))
         FROM AP_DUPLICATE_VENDORS_ALL DUPS,
              PO_VENDORS VEND1,
              PO_VENDOR_SITES_ALL SITE1,
              PO_VENDORS VEND2,
              PO_VENDOR_SITES_ALL SITE2,
              AP_INVOICES_ALL INV
         WHERE VEND1.VENDOR_ID = SITE1.VENDOR_ID
         AND VEND2.VENDOR_ID = SITE2.VENDOR_ID
         AND VEND1.VENDOR_ID = DUPS.VENDOR_ID
         AND VEND2.VENDOR_ID = DUPS.DUPLICATE_VENDOR_ID
         AND SITE1.VENDOR_SITE_ID = DUPS.VENDOR_SITE_ID
         AND SITE2.VENDOR_SITE_ID =
              DUPS.DUPLICATE_VENDOR_SITE_ID
         AND SITE1.VENDOR_SITE_ID = INV.VENDOR_SITE_ID
-- Where the invoices are now
         AND VEND1.VENDOR_NAME = 'SMITH, JOHN'
         AND SITE1.VENDOR_SITE_CODE = 'ATLANTA'
-- Where the invoices were originally
         AND VEND2.VENDOR_NAME = 'JOHN SMITH'
         AND SITE2.VENDOR_SITE_CODE = 'DECATUR');

COMMIT;

```

Vendor Site Delete

After merging your vendors, identifying the inactive ones and merging cancelled invoices to a dummy vendor, you are still stuck with vendor sites that are still on the system. The merge process only puts an inactive date on the vendor site and does not purge it from the system.

This SQL script will delete all vendor sites not having any invoices or PO's and have an inactive date prior to today.

Again, there are two scripts. One to list the delete candidates prior to deleting and one to perform the delete.

Prerequisite: You must inactivate any vendor sites you want deleted – set vendor inactive date to a date earlier than today.

```
-- List of Vendor Sites with no invoices, no PO's
-- and an inactive date less than today
Select sites.vendor_site_id,
       vend.segment1 Vendor#,
       vendor_name,
       sites.vendor_site_code,
       vend.end_date_active VEND_INACTIVE,
       sites.inactive_date SITE_INACTIVE
from po_vendor_sites_all sites,
     po_vendors vend,
     ap_invoices_all inv,
     po_headers_all PO
where vend.vendor_id = sites.vendor_id
and sites.vendor_site_id = inv.vendor_site_id(+)
and inv.vendor_site_id is null
and sites.vendor_site_id = po.vendor_site_id(+)
and po.vendor_site_id is null
and sites.inactive_date < sysdate
order by
       vendor_name,
       vendor_site_code;
```

```
-- Delete Vendor Sites with no invoices, no PO's
-- and an inactive date less than today
delete from po_vendor_sites_all posite
where posite.vendor_site_id in
  (Select sites.vendor_site_id
   from po_vendor_sites_all sites,
        po_vendors vend,
        ap_invoices_all inv,
        po_headers_all PO
   where vend.vendor_id = sites.vendor_id
        and sites.vendor_site_id = inv.vendor_site_id(+)
        and inv.vendor_site_id is null
        and sites.vendor_site_id = po.vendor_site_id(+)
        and po.vendor_site_id is null
        and sites.inactive_date < sysdate);

commit;
```