NAAUG 2004

New Titles: SQL programs
written by Anca Coza, McGill University Libraries

new_title_daily.sql

/*----------------------------------------------
- PL/SQL program to detect NEW TITLES/ update z30_inventory_number_date/ insert lines in
  mgu01.z07
- first change for ItemProcessStatus to BLANK (excepting the initial null - old books)
- tables : z30/z30h
- created: 2003 September
- modified : 2004 January
- version : ALEPH daily
----------------------------------------------*/

SET LINESIZE 50
SET SERVEROUTPUT ON SIZE 1000000

DECLARE

  -- local variable used in the program
  first_occ CHAR(15);       /* ADM */
  prev_occ CHAR(9):='000000000';  /* to avoid double BIB record number in z07 */
  entry_date number(8);   /* previous date */
  prev_ips char(2);   /* previous IPS */
  seq number(1):=0;    /* sequence for z07_sequence - last digit */
  cnt_z07 number(2):=0;   /* count for existing BIB numbers in z07 */

  -- declaring the cursor for selecting the information to be checked
  -- get the list from z30h only with Process Status changes from z30h, for BOOKs not in process
  -- and that haven't been previously updated with an accession date

CURSOR ips_cur IS
  select
    substr(z30h_rec_key, 1, 15) ADM,
    z30h_h_date H_DATE,
    z30h_h_hour H_HOUR, -- added Febr 6 - 0019999060
    z30h_item_process_status IPS
  from
    mgu50.z30h
  where
    z30h_h_reason like '%Process%'
    and substr(z30h_rec_key, 1, 15) IN
    (select
      z30_rec_key
    from
      mgu50.z30
    where
      /* ... */

/* ... */
--z30_barcode like '3%' and -- eliminated on January 27!!
z30_material = 'BOOK'
and z30_item_process_status IS NULL
and z30_inventory_number_date = 0 -- added Jan 19
)
order by
1,2,3;

--record of the cursor type
ips_rec ips_cur%ROWTYPE;
BEGIN
--open ips cursor
IF NOT ips_cur%ISOPEN THEN
OPEN ips_cur;
END IF;

-- fetch the first record
FETCH ips_cur INTO ips_rec;
-- make disable the accession_date trigger for update z30_inventory_number_date:
execute immediate 'ALTER TRIGGER "MGU50"."KEEP_ACC_DATE_UPD" DISABLE';

WHILE ips_cur%FOUND LOOP
    first_occ := ips_rec.ADM;
    entry_date := ips_rec.H_DATE;
    prev_ips := ips_rec.IPS;

    -- insulate the case "INITIAL ItemProcessStatus is null" - the book was in collection before 2000

    WHILE prev_ips is null LOOP
        -- skip all the other records for this ADM
        LOOP
            -- fetch next record
            FETCH ips_cur INTO ips_rec;

            -- exit from the internal LOOP (a new ADM)
            exit when ips_rec.ADM <> first_occ or not ips_cur%FOUND;
        END LOOP;

        -- populate variables with new values (the new ADM)
        first_occ := ips_rec.ADM;
        entry_date := ips_rec.H_DATE;
        prev_ips := ips_rec.IPS;
    END LOOP;

END LOOP;
END LOOP;
-- last fetch reached a new ADM with first IPS not null
-- so, there are changes in process status in z30h - new items!!

WHILE ips_rec.ADM = first_occ AND ips_rec.IPS is not null AND ips_cur%FOUND
LOOP
-- this loop exits at first null ItemProcessStatus OR when it fetches a new ADM
IF ips_rec.IPS is not null THEN
-- populate IPS and the date from current record
prev_ips := ips_rec.IPS;
entry_date := ips_rec.H_DATE;
END IF;
-- next record
FETCH ips_cur INTO ips_rec;
END LOOP;

/* the wanted accession date would be taken:
- from the last record for current ADMIN OR
- from the record before changing to IPS null

accession_date update will be done only for today's date + yesterday's night and not for
reserve items */

-- initial seed:
-- IF entry_date >= 20030601 and first_occ < '009000000000000' THEN

IF entry_date = TO_NUMBER(TO_char(SYSDATE,'YYYYMMDD'))
 OR entry_date = TO_NUMBER(TO_char(SYSDATE-1,'YYYYMMDD'))
 AND first_occ < '009000000000000' THEN

DBMS_OUTPUT.PUT_LINE(first_occ||prev_ips||entry_date);

-- update the accession_date
EXECUTE IMMEDIATE
 'update mgu50.z30
 set z30_inventory_number_date= :1
 where z30_rec_key = :2'
USING
 entry_date, first_occ;
-- insert updated bib record numbers in z07 but only distinct BIBs

prev_occ := substr(first_occ,1,9);  --- keep previous BIB

-- if BIB number already exists in z07, skip it

EXECUTE IMMEDIATE
  'SELECT COUNT(*) FROM MGU01.Z07
   WHERE Z07_REC_KEY = :1'
INTO cnt_z07
USING prev_occ;

IF cnt_z07 = 0 THEN
  -- BIB number not in z07:

  EXECUTE IMMEDIATE
    'insert into mgu01.z07 (Z07_REC_KEY,Z07_SEQUENCE,z07_LEVEL,Z07_HISTORY)
     values (:1,''2009''||to_char(sysdate,''MMDDHH24MISS'')|| :2 ,''MGU01''||:3)'
  USING prev_occ, seq, prev_occ;

  ---DBMS_OUTPUT.PUT_LINE (prev_occ||' added in z07');

  -- generating the sequence 0-9

  IF seq = 9 THEN
    seq:=0;
  ELSE
    seq:=seq + 1;
  END IF;

ELSE

  DBMS_OUTPUT.PUT_LINE (prev_occ||' already exists in z07');

END IF;

END IF;

END LOOP;

CLOSE ips_cur;

-- make all changes permanently :

COMMIT;

-- turn on the accession_date trigger :

execute immediate 'ALTER TRIGGER "MGU50".KEEP_ACC_DATE_UPD* ENABLE';
EXCEPTION
WHEN others THEN
ROLLBACK;
execute immediate 'ALTER TRIGGER "MGU50"."KEEP_ACC_DATE_UPD" ENABLE';
DBMS_OUTPUT.PUT_LINE('Exit with error message: '"||SQLCODE||'" ||SQLERRM);
DBMS_OUTPUT.PUT_LINE('Please run the job again!');
END;
/

new_title_revise.sql (weekly)

/*----------------------------------------------
- PL/SQL program to detect NEW TITLES/ update z30_inventory_number_date/ insert lines in mgu01.z07
- first change for ItemProcessStatus to BLANK (excepting the initial null - old books)
- tables : z30/z30h
- created: 2003 September
- modified : 2004 January
- version : ALEPH check
----------------------------------------------*/

SET LINESIZE 50
SET SERVEROUTPUT ON SIZE 1000000

DECLARE
-- local variable used in the program
first_occ CHAR(15); /* ADM */
prev_occ CHAR(9):='000000000'; /* to avoid double BIB record number in z07 */
entry_date number(8); /* previos date */
prev_ips char(2); /* previous IPS */
seq number(1):=0; /* sequence for z07_sequence - last digit */
cnt_z07 number(2):=0; /* count for existing BIB numbers in z07 */
acc_date number(8);

-- declaring the cursor for selecting the information to be checked
-- get the list from z30h only with Process Status changes from z30h, for BOOKs not in process
-- and that haven't been previously updated with an accession date
CURSOR ips_cur IS
select
substr(z30h_rec_key, 1, 15) ADM,
z30h_h_date H_DATE,
z30h_h_hour H_HOUR,
z30h_item_process_status IPS
from
mgu50.z30h
where
z30h_h_reason like '%Process%'
and substr(z30h_rec_key, 1, 15) IN
(select...
z30_rec_key
from
mgu50.z30
where
--z30_barcode like '3%' and
z30_material = 'BOOK'
and z30_item_process_status IS NULL
--and z30_inventory_number_date = 0 -- added Jan 19
)
order by
  1,2,3;

--record of the cursor type
ips_rec ips_cur%ROWTYPE;
BEGIN

--open ips cursor
IF NOT ips_cur%ISOPEN THEN
  OPEN ips_cur;
END IF;

-- fetch the first record
FETCH ips_cur INTO ips_rec;

-- make disable the accession_date trigger for update z30_inventory_number_date:
execute immediate 'ALTER TRIGGER "MGU50"."KEEP_ACC_DATE_UPD" DISABLE';

WHILE ips_cur%FOUND LOOP

  first_occ := ips_rec.ADM;
  entry_date := ips_rec.H_DATE;
  prev_ips := ips_rec.IPS;

  -- insulate the case "INITIAL ItemProcessStatus is null" - the book was in collection before 2000
  WHILE prev_ips is null LOOP

    -- skip all the other records for this ADM
    LOOP
      -- fetch next record
      FETCH ips_cur INTO ips_rec;

      -- exit from the internal LOOP (a new ADM)
      exit when ips_rec.ADM <> first_occ or not ips_cur%FOUND;
    END LOOP;

  END WHILE;

  -- populate variables with new values (the new ADM)
```sql
first_occ := ips_rec.ADM;
entry_date := ips_rec.H_DATE;
prev_ips := ips_rec.IPS;

END LOOP;

-- last fetch reached a new ADM with first IPS not null
-- so, there are changes in process status in z30h - new items!!

WHILE ips_rec.ADM = first_occ AND ips_rec.IPS is not null AND ips_cur%FOUND
LOOP
-- this loop exits at first null ItemProcessStatus OR when it fetches a new ADM
IF ips_rec.IPS is not null THEN
-- populate IPS and the date from current record
    prev_ips := ips_rec.IPS;
    entry_date := ips_rec.H_DATE;
END IF;
-- next record
    FETCH ips_cur INTO ips_rec;
END LOOP;

/* the wanted accession date would be taken:
- from the last record for current ADMIN OR
- from the record before changing to IPS null

accession_date update will be done only for today's date + yesterday's night and not for
reserve items */

IF entry_date >= 20030601 and entry_date <
    TO_NUMBER(TO_char(SYSDATE,'YYYYMMDD'))
    AND first_occ < '009000000000000' THEN
-- !!! un-comment next line to have the update list:
-- DBMS_OUTPUT.PUT_LINE(first_occ||prev_ips||entry_date);
-- update the accession_date
    EXECUTE IMMEDIATE
        'SELECT z30_inventory_number_date
         FROM MGU50.Z30
         WHERE Z30_REC_KEY = :1'
    INTO
```
acc_date

USING
first_occ;

IF acc_date <> entry_date THEN

DBMS_OUTPUT.PUT_LINE(first_occ||' wrong date ''||acc_date||' correct date ''||entry_date);

-- update the accession_date

EXECUTE IMMEDIATE
'update mgu50.z30
    set z30_inventory_number_date= :1
    where z30_rec_key = :2'
USING
entry_date, first_occ;

-- insert updated bib record numbers in z07 but only distinct BIBs

prev_occ := substr(first_occ,1,9);  --- keep previous BIB

-- if BIB number already exists in z07, skip it

EXECUTE IMMEDIATE
'SELECT COUNT(*) FROM MGU01.Z07
    WHERE Z07_REC_KEY = :1'
INTO
cnt_z07
USING
prev_occ;

IF cnt_z07 = 0 THEN

-- BIB number not in z07:

EXECUTE IMMEDIATE
'insert into mgu01.z07
    (Z07_REC_KEY,Z07_SEQUENCE,z07_LEVEL,Z07_HISTORY) values
    (:1,''2009''||to_char(sysdate,''MMDDHH 24MISS'')|| :2 ,1 ,''MGU01''||:3)
USING
prev_occ, seq, prev_occ;

DBMS_OUTPUT.PUT_LINE (prev_occ||' added in z07'); /* test

for entries in z07*/

-- generating the sequence 0-9

IF seq = 9 THEN
    seq:=0;
ELSE
    seq:=seq + 1;
END IF;
ELSE
   DBMS_OUTPUT.PUT_LINE (prev_occ||' already exists in z07');
END IF;
END IF;
END IF;
END LOOP;
CLOSE ips_cur;

-- make all changes permanently :
COMMIT;

-- turn on the accession_date trigger :
execute immediate 'ALTER TRIGGER "MGU50"."KEEP_ACC_DATE_UPD" ENABLE';
EXCEPTION
WHEN others THEN
   ROLLBACK;
   execute immediate 'ALTER TRIGGER "MGU50"."KEEP_ACC_DATE_UPD" ENABLE';
   DBMS_OUTPUT.PUT_LINE('Exit with error message: '||SQLCODE||' '||SQLERRM);
   DBMS_OUTPUT.PUT_LINE('Please run the job again!');
END;
/

new_title_revise_gift.sql (weekly)

/*----------------------------------------------
- PL/SQL program to detect NEW TITLES/ update z30_inventory_number_date/ insert lines in mgu01.z07
- first change for ItemProcessStatus to BLANK (skipping the initial ips null)
- tables : z30/z30h
- created: 2003 September
- modified : 2004 February
- version : ALEPH gifts
----------------------------------------------*/

SET LINESIZE 50
SET SERVEROUTPUT ON SIZE 1000000

DECLARE
   -- local variable used in the program
first_occ CHAR(15); /* ADM */
prev_occ CHAR(9):='000000000'; /* to avoid double BIB record number in z07 */
entry_date number(8); /* previous date */
prev_ips char(2); /* previous IPS */
seq number(1):=0; /* sequence for z07_sequence - last digit */
cnt_z07 number(2):=0; /* count for existing BIB numbers in z07 */
acc_date number(8);

-- declaring the cursor for selecting the information to be checked
-- get the list from z30h only with Process Status changes from z30h, for BOOKs not in process
-- and that haven't been previously updated with an accession date

CURSOR ips_cur IS
select
    substr(z30h_rec_key, 1, 15) ADM,
    z30h_h_date H_DATE,
    z30h_h_hour H_HOUR,
    z30h_item_process_status IPS
from
    mgu50.z30h
where
    z30h_h_reason like '%Process%'
and substr(z30h_rec_key, 1, 15) IN
    (select
        z30_rec_key
    from
        mgu50.z30, mgu50.z68
    where
        --z30_barcode like '3%' and
        z30_material = 'BOOK'
        and z30_item_process_status IS NULL
        and z68_order_number = z30_order_number
        and z68_method_of_aquisition = 'G'
    )
order by
    1,2,3;

-- record of the cursor type
ips_rec ips_cur%ROWTYPE;
BEGIN
-- open ips cursor
IF NOT ips_cur%ISOPEN THEN
    OPEN ips_cur;
END IF;
-- fetch the first record
FETCH ips_cur INTO ips_rec;
-- make disable the accession_date trigger for update z30_inventory_number_date:
execute immediate 'ALTER TRIGGER "MGU50"."KEEP_ACC_DATE_UPD" DISABLE';
WHILE ips_cur%FOUND LOOP
    first_occ := ips_rec.ADM;
    entry_date := ips_rec.H_DATE;
    prev_ips := ips_rec.IPS;
    
    -- skip the "INITIAL ItemProcessStatus is null"
    WHILE prev_ips IS NULL AND ips_cur%FOUND LOOP
        -- skip all the other records for this ADM
        -- LOOP
        -- fetch next record
        --DBMS_OUTPUT.PUT_LINE(first_occ||prev_ips||entry_date||'skiped');
        FETCH ips_cur INTO ips_rec;
        
        -- exit from the internal LOOP (a new ADM)
        -- exit when ips_rec.ADM <> first_occ or not ips_cur%FOUND;
        -- END LOOP;
        
        -- populate variables with new values (the new ADM)
        first_occ := ips_rec.ADM;
        entry_date := ips_rec.H_DATE;
        prev_ips := ips_rec.IPS;
    END LOOP;
    
    -- last fetch reached first IPS not null
    WHILE ips_rec.ADM = first_occ AND ips_rec.IPS IS NOT NULL AND ips_cur%FOUND LOOP
        -- this loop exits at first null ItemProcessStatus OR when it fetches a new ADM
        IF ips_rec.IPS IS NOT NULL THEN
            -- populate IPS and the date from current record
            prev_ips := ips_rec.IPS;
            entry_date := ips_rec.H_DATE;
        END IF;
        
        --DBMS_OUTPUT.PUT_LINE(first_occ||prev_ips||entry_date||'same ADM');
        -- next record
        FETCH ips_cur INTO ips_rec;
    END LOOP;
/* the wanted accession date would be taken:
- from the last record for current ADMIN OR
- from the record before changing to IPS null

accession_date update will be done only for today’s date + yesterday's night and not for
reserve items */

IF entry_date >= 20030601 and first_occ < '009000000000000' THEN
    -- !!! un-comment next line to have the update list:
    -- DBMS_OUTPUT.PUT_LINE(first_occ||prev_ips||entry_date);
    EXECUTE IMMEDIATE
        'SELECT z30_inventory_number_date
         FROM MGU50.Z30
         WHERE Z30_REC_KEY = :1'
    INTO acc_date
    USING first_occ;

    IF acc_date <> entry_date THEN
        -- update the accession_date
        DBMS_OUTPUT.PUT_LINE(first_occ||prev_ips||entry_date);
        EXECUTE IMMEDIATE
            'update mgu50.z30
             set z30_inventory_number_date = :1
             where z30_rec_key = :2'
        USING entry_date, first_occ;

        -- insert updated bib record numbers in z07 but only distinct BIBs
        IF substr(first_occ,1,9) <> prev_occ THEN
            prev_occ := substr(first_occ,1,9);  --- keep previous BIB
            -- in the case that BIB number already exists in z07, skip it
            EXECUTE IMMEDIATE
                'SELECT COUNT(*) FROM MGU01.Z07
                 WHERE Z07_REC_KEY = :1'
            INTO cnt_z07
            USING prev_occ;
            IF cnt_z07 = 0 THEN

        END IF;
    END IF;
END IF;
-- BIB number not in z07:

EXECUTE IMMEDIATE 'insert into mgu01.z07
(Z07_REC_KEY,Z07_SEQUENCE,z07_LEVEL,Z07_HISTORY) values
(:1,''2009''||to_char(sysdate,''MMDDHH 24MISS'')|| :2 ,1 ,''MGU01''||:3')
USING prev_occ, seq, prev_occ;

-- generating the sequence 0-9

IF seq = 9 THEN
  seq:=0;
ELSE
  seq:=seq + 1;
END IF;

ELSE
  DBMS_OUTPUT.PUT_LINE (prev_occ||' already exists in z07');
END IF;

END IF;
END IF;
END IF;

WHILE ips_rec.ADM = first_occ AND ips_cur%FOUND LOOP

FETCH ips_cur INTO ips_rec;

prev_ips := ips_rec.IPS;
entry_date := ips_rec.H_DATE;

-- DBMS_OUTPUT.PUT_LINE(first_occ||prev_ips||entry_date||' skiped after');

END LOOP;
END LOOP;
CLOSE ips_cur;

-- make all changes permanently :
COMMIT;

-- turn on the accession_date trigger :
Anca describes her SQL:

The PL/SQL program needs ALEPH_ADMIN user for running.

Why? It needs the following system privileges:

- ALTER ANY TRIGGER (to make disable/enable KEEP_ACC_DATE_UPD trigger)
- SELECT ANY TABLE (select from mgu50.z30/z30h)
- UPDATE ANY TABLE (update mgu50.z30.z30_inventory_number_date)
- INSERT ANY TABLE (insert new lines in mgu01.z07)

It is scheduled now to be executed every night at 7p.m. in job_list to fill in the accession date for today's new entries and what has left from yesterday.

What the program does?

Search in Item_history table (z30h) for all changes in Process_status, only for Books and with the present Process_status blank.

```sql
select
  substr(z30h_rec_key, 1, 15) ADM,
  z30h_h_date H_DATE,
  z30h_item_process_status IPS
from
  mgu50.z30h
where
  z30h_h_reason like '%Process%'
and substr(z30h_rec_key, 1, 15) IN
  (select z30_rec_key
    from mgu50.z30
    where
      z30_material = 'BOOK'
      and z30_item_process_status IS NULL
      and z30_inventory_number_date = 0 -- added Jan 19
  )
order by
  1, 2;
```
The items with "initial process status NULL" in z30h are rejected because they are old, and we have no information about their previous process status and the date when it was changed.

For the others, I checked for the first process_status Null.

The accession date is taken from the line preceding the first IPS null.

See here a sample for BIB number 001750343 - item history:

<table>
<thead>
<tr>
<th>Action</th>
<th>Date</th>
<th>Process Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>OI</td>
<td>20001106</td>
<td>Process Status</td>
</tr>
<tr>
<td>OR</td>
<td>20001106</td>
<td>Process Status</td>
</tr>
<tr>
<td>IP</td>
<td>20001115</td>
<td>Process Status</td>
</tr>
<tr>
<td>TR</td>
<td>20001122</td>
<td>Process Status</td>
</tr>
<tr>
<td></td>
<td>20030909</td>
<td>Process Status</td>
</tr>
<tr>
<td></td>
<td>20030910</td>
<td>Process Status</td>
</tr>
</tbody>
</table>

----- we should stop here, taking this date as an entry date in library!!

In the case there are no Process-status null in z30h, the accession date will be the last date from z30h when the process status was changed.

Note: The previous process_status before last change is usually IP / TR/ SB but could be a lot of other cases when the change was done from CA, MS, OI, OR, BP, RP, RV, UP, CT (Pat accepted these cases!)

After detecting the accession date the program will:

- display the list(ADM, date, previous Item_process_status) (only for test)
- update mgu50.z30_inventory_number_date with the new accession date
  execute immediate
  'update mgu50.z30
  set z30_inventory_number_date= :1
  where z30_rec_key = :2'
  using entry_date, first_occ;
- insert new lines in mgu01.z07 to reindex affected bib numbers (only after a check if that BIB number is not already in z07)
  execute immediate
  'insert into mgu01.z07
  (Z07_REC_KEY,Z07_SEQUENCE,Z07_LEVEL,Z07_HISTORY) values
  (:1,"2009"||to_char(sysdate,"MMDDHH24MISS")|| :2 ,1 ,"MGU01"||:3)'
  using prev_occ, seq, prev_occ;

The lines inserted in mgu01.z07:

| z07_rec_key | -- bib record number(9 characters) |
| z07_sequence | -- Timestamp : '2009' + MMDDHH24MISS + seq(1) (15 characters) ('2009' lower priority for reindex) |
| z07_level   | -- 1 |
| z07_history | -- MGU01 + bib record number(9 characters) |
The initial seed program is the same as the daily one, except the period of time considered:

The line:

IF entry_date = TO_NUMBER(TO_char(SYSDATE,'YYYYMMDD')) or entry_date =
TO_NUMBER(TO_char(SYSDATE-1,'YYYYMMDD')) AND first_occ < '0090000000000000' THEN

should be replaced with:

if entry_date >= 20030601 and first_occ < '0090000000000000' then

Problem: Missing accession date (new_title_check.sql program)

Because the seeding program was run once on Dec 17 for all new entries since June 1st, at that
moment were some of this items back in process or with a "B" barcode again, so those where not
considered in the initial "select".

After a while they were back "on the shelf" so they could recovered - 48 items in this case.

Some of them had the accession date deleted, because the trigger was activated on Jan. 7.

The check program should be run periodically to see if there are new modification in item process
status (to Null)/ item_history (deleted lines)