**HRMS: Absence Management (Leave) Queries**

**To get approved leaves**:

         SELECT pat.NAME absence\_type, paa.date\_start leave\_start\_date,  
                paa.date\_end leave\_end\_date, paa.absence\_days, paa.abs\_information\_category,  
                paa.abs\_information2, paa.abs\_information3,paa.abs\_information4  
           FROM per\_absence\_attendances paa,  
                per\_absence\_attendance\_types pat,  
                per\_all\_people\_f papf  
          WHERE paa.absence\_attendance\_type\_id =  
                                                pat.absence\_attendance\_type\_id  
            AND papf.person\_id = paa.person\_id  
            AND TRUNC (paa.date\_start) BETWEEN TRUNC  
                                                    (papf.effective\_start\_date)  
                                           AND TRUNC (papf.effective\_end\_date)  
            AND TRUNC (p\_date) BETWEEN TRUNC (paa.date\_start)  
                                   AND TRUNC (NVL (paa.date\_end,  
                                                   paa.date\_start  
                                                  )  
                                             )  
            AND papf.person\_id = :p\_person\_id;

**To get pending leaves:**

      SELECT hat.transaction\_id, hats.transaction\_step\_id

        FROM hr\_api\_transactions hat, hr\_api\_transaction\_steps hats

       WHERE hat.transaction\_ref\_table = 'PER\_ABSENCE\_ATTENDANCES'

         AND hat.transaction\_group = 'ABSENCE\_MGMT'

         AND hat.transaction\_identifier = 'ABSENCES'

         AND hat.transaction\_ref\_id IS NOT NULL

         AND hat.status = 'Y'

         AND hat.transaction\_id = hats.transaction\_id

         AND hat.CREATOR\_PERSON\_ID = :p\_person\_id;

To delete pending leaves:

As far as i know there is no API to delete pending leaves.

If you are aware of any API used to delete pending leaves, please let me know.

Solution1:

This is my own way of deleting the leaves

CREATE OR REPLACE PROCEDURE xx\_delete\_pending\_leaves

AS

   CURSOR c1

   IS

      SELECT hat.transaction\_id, hats.transaction\_step\_id

        FROM hr\_api\_transactions hat, hr\_api\_transaction\_steps hats

       WHERE hat.transaction\_ref\_table = 'PER\_ABSENCE\_ATTENDANCES'

         AND hat.transaction\_group = 'ABSENCE\_MGMT'

         AND hat.transaction\_identifier = 'ABSENCES'

         AND hat.transaction\_ref\_id IS NOT NULL

         AND hat.status = 'Y'

         AND hat.transaction\_id = hats.transaction\_id;

--For a particular employee/pass the person\_id values as a cursor parameter

         --AND hat.CREATOR\_PERSON\_ID = 11923;

   --AND hats.creator\_person\_id = p\_person\_id;  -- Add date conditon later and person ID if required

   l\_transaction\_id        NUMBER;

   l\_transaction\_step\_id   NUMBER;

   l\_error                 VARCHAR2 (300);

BEGIN

   OPEN c1;

   LOOP

      FETCH c1

       INTO l\_transaction\_id, l\_transaction\_step\_id;

      EXIT WHEN c1%NOTFOUND;

      IF l\_transaction\_step\_id IS NOT NULL

      THEN

         DELETE FROM hr\_api\_transaction\_values

               WHERE transaction\_step\_id = l\_transaction\_step\_id;

         DBMS\_OUTPUT.put\_line

                     (   'Deleted Transaction Value of transaction step id: '

                      || l\_transaction\_step\_id

                     );

      END IF;

      IF l\_transaction\_id IS NOT NULL

      THEN

         DELETE FROM hr\_api\_transaction\_steps

               WHERE transaction\_id = l\_transaction\_id;

         DBMS\_OUTPUT.put\_line

                           (   'Deleted Transaction step of transaction id: '

                            || l\_transaction\_id

                           );

         DELETE FROM hr\_api\_transactions

               WHERE transaction\_id = l\_transaction\_id;

         DBMS\_OUTPUT.put\_line ('Deleted Transaction ID: ' || l\_transaction\_id);

      END IF;

   END LOOP;

   CLOSE c1;

   COMMIT;

EXCEPTION

   WHEN OTHERS

   THEN

      l\_error := SUBSTR (SQLERRM, 1, 200);

      DBMS\_OUTPUT.put\_line (   'Other Error in xx\_delete\_pending\_leaves- '

                            || l\_error

                           );

--p\_out := 'Other Error in xx\_delete\_pending\_leaves- '||l\_error; -- Removed OUT parameter

-- Write FND\_LOG for concurrent program log

END xx\_delete\_pending\_leaves;

Solution 2:

We can enable/disable the update/delete icon in Absence Management - HR self service responsibility. How ?

Solution: Get query from jsp page (OAF) (See Standard Query below) and change the decode value of cancel\_icon, Confirm\_icon and update\_icon values by passing 'HrCancelEnabled/HrCancelDisabled' OR 'HrUpdateEnabled/HrUpdateDisabled' values and replace with the custom query.

Tables Used in Standard Query are:

       per\_absence\_attendances,

       per\_absence\_attendance\_types,

       per\_abs\_attendance\_types\_tl,

       hr\_lookups,

       hr\_api\_transactions,

       hr\_api\_transaction\_steps

Standard Query:

SELECT  NVL(paa.date\_start,paa.date\_projected\_start) start\_date

        ,NVL(paa.date\_end,paa.date\_projected\_end) end\_date

       ,paattl.name absence\_type

       ,paat.absence\_attendance\_type\_id absence\_attendance\_type\_id

       ,fcl.meaning  absence\_category

       ,paat.absence\_category  absence\_category\_code

       ,paa.absence\_hours

       ,(SELECT meaning from hr\_lookups where 'A' = lookup\_code(+) and 'LEAVE\_STATUS' = lookup\_type(+))

       approval\_status

       , 'A' approval\_status\_code

       ,decode(paa.date\_start,null,

                                   (SELECT meaning from fnd\_lookup\_values

                                    where lookup\_type ='ABSENCE\_STATUS' and lookup\_code ='PLANNED'

 and language = userenv('LANG')),

                                   (SELECT meaning from fnd\_lookup\_values

                                    where lookup\_type ='ABSENCE\_STATUS' and lookup\_code ='CONFIRMED'

and language = userenv('LANG')))

        absence\_status

       , decode(paa.date\_start,null,'PLANNED','CONFIRMED') absence\_status\_code

       ,(nvl((SELECT 'Y'

     from fnd\_attached\_documents

     where entity\_name='PER\_ABSENCE\_ATTENDANCES'

     and pk1\_value = to\_char(paa.absence\_attendance\_id) and rownum = 1),'N')) supporting\_documents

       ,decode (

        paa.date\_start,null,

        decode(paa.date\_projected\_start, null, 'HrCancelDisabled', 'HrCancelEnabled'),

        decode(

               sign(trunc(paa.date\_start)-trunc(sysdate)),1,'HrCancelEnabled',

                                                                              decode(paa.date\_end,null,'HrCancelEnabled','HrCancelDisabled')

              )

       )

        cancel\_icon

       ,decode(paa.date\_end,NULL,

          decode(paa.date\_start,Null, 'HrConfirmEnabled','HrConfirmDisabled'), 'HrConfirmDisabled' )

         confirm\_icon

       ,decode ( paa.date\_start,null,

          'HrUpdateEnabled',

           decode(sign(nvl(paa.date\_end,sysdate+1)-sysdate),1,'HrUpdateEnabled','HrUpdateDisabled'))

         update\_icon

       ,NULL

         details\_icon

       ,paa.absence\_attendance\_id

       ,null transaction\_id

      ,to\_char(paa.absence\_attendance\_id) supportingDocKey

      ,paa.absence\_days

FROM per\_absence\_attendances paa

     ,per\_absence\_attendance\_types paat

     ,per\_abs\_attendance\_types\_tl paattl

     ,hr\_lookups fcl

WHERE paa.person\_id = :1

      and paa.business\_group\_id+0 = :2

      and paa.absence\_attendance\_type\_id = paat.absence\_attendance\_type\_id

      and paat.absence\_attendance\_type\_id = paattl.absence\_attendance\_type\_id

      and paattl.language = userenv('LANG')

      and fcl.lookup\_type(+) = 'ABSENCE\_CATEGORY'

      and paat.absence\_category = fcl.lookup\_code(+)

      and ((hr\_api.return\_legislation\_code(paat.business\_group\_id) = 'GB'

         and nvl(paat.absence\_category,'#')  not in

('M','GB\_PAT\_ADO','GB\_PAT\_BIRTH','GB\_ADO'))

         or

        (hr\_api.return\_legislation\_code(paat.business\_group\_id) <> 'GB'

         and nvl(paat.absence\_category,'#') not in

('GB\_PAT\_ADO','GB\_PAT\_BIRTH','GB\_ADO')))

      and not exists

      ( select 'e'

        from hr\_api\_transactions t

         WHERE

        t.selected\_person\_id = paa.person\_id

        and t.CREATOR\_PERSON\_ID  = nvl(:3,t.CREATOR\_PERSON\_ID)

        and t.transaction\_ref\_table='PER\_ABSENCE\_ATTENDANCES'

         and t.transaction\_ref\_id = paa.absence\_attendance\_id

         and not(hr\_absutil\_ss.getabsencetype(t.transaction\_id,    NULL) IS NULL

         and t.status = 'W')

         and t.status not in ('D','E')

      )

union all

select hr\_absutil\_ss.getStartDate(hat.transaction\_id,null) start\_date

       ,hr\_absutil\_ss.getEndDate(hat.transaction\_id,null) end\_date

       ,hr\_absutil\_ss.getAbsenceType(hat.transaction\_id,null) absence\_type

       , to\_number(hats.Information5) absence\_attendance\_type\_id

      ,hr\_absutil\_ss.getAbsenceCategory(hat.transaction\_id,null)  absence\_category

      , hats.Information6 absence\_category\_code

       ,hr\_absutil\_ss.getAbsenceHoursDuration(hat.transaction\_id,null)  absence\_hours

       ,hr\_absutil\_ss.getApprovalStatus(hat.transaction\_id,null) approval\_status

       ,hr\_absutil\_ss.getApprovalStatusCode(hat.transaction\_id,null) approval\_status\_code

       ,hr\_absutil\_ss.getAbsenceStatus(hat.transaction\_id,null) absence\_status

       , hats.Information9 absence\_status\_code

       ,hr\_absutil\_ss.hasSupportingDocuments(hat.transaction\_id,hat.TRANSACTION\_REF\_ID) supporting\_documents

       ,hr\_absutil\_ss.isCancelAllowed(hat.transaction\_id,null,hat.status) cancel\_icon

       ,hr\_absutil\_ss.isConfirmAllowed(hat.transaction\_id,null) confirm\_icon

     ,hr\_absutil\_ss.isUpdateAllowed(hat.transaction\_id,null,hat.status) update\_icon

       ,null  details\_icon

      ,hat.TRANSACTION\_REF\_ID absence\_attendance\_id

       ,hat.transaction\_id transaction\_id

       ,to\_char(hat.TRANSACTION\_REF\_ID||'\_'||hat.transaction\_id) supportingDocKey

      ,hr\_absutil\_ss.getAbsenceDaysDuration(hat.transaction\_id,null)  absence\_days

from hr\_api\_transactions hat

     ,hr\_api\_transaction\_steps hats

where hat.TRANSACTION\_REF\_TABLE='PER\_ABSENCE\_ATTENDANCES'

and hat.TRANSACTION\_GROUP='ABSENCE\_MGMT'

and hat.TRANSACTION\_IDENTIFIER='ABSENCES'

and hat.TRANSACTION\_REF\_ID is not null

and hat.SELECTED\_PERSON\_ID =:4

and hat.CREATOR\_PERSON\_ID  = nvl(:5,hat.CREATOR\_PERSON\_ID)

and hat.transaction\_id=hats.transaction\_id(+)

and hat.status not in ('D','E')

and not (hr\_absutil\_ss.getabsencetype(hat.transaction\_id,null) is null and hat.status='W')