# Oracle Fusion HDL and BI Report Deliverable

## Document Information

|  |  |
| --- | --- |
| Document Title | Fusion HDL and Report Specification - Employee Element Entry |
| Author | Abdulrahman |
| Date Created | 2025-07-15 |
| Last Updated | 2025-07-15 |
| Fusion Module | HCM > Payroll |
| Deliverable Type | HDL Data Loader / BI Report |
| Target Environment | TEST |
| Status | Delivered |

## Section 1: HDL Data Loader Summary

|  |  |
| --- | --- |
| Object Type | Element Entry |
| HDL File Name | ElementEntry.dat |
| File Format | .dat (Text) |
| Source System Owner | LEGACY |
| Source System ID | ELEM001 |
| Key Fields Used | AssignmentNumber, ElementName |
| Supported Actions | Insert |
| Dependencies | Employee Assignment must exist |

Sample HDL Snippet:

METADATA|ElementEntry|SourceSystemOwner|SourceSystemId|AssignmentNumber|ElementName|EntryType|InputValueName|EntryValue  
MERGE|ElementEntry|LEGACY|ELEM001|E12345|Housing Allowance|EARNINGS|Amount|5000

## Section 2: BI/OTBI Report Details

|  |  |
| --- | --- |
| Report Name | Active Employees by BU |
| Report Path | /Custom/HCM/Reports/ActiveEmployees.xdo |
| Report Type | BI Publisher |
| Subject Area (OTBI) | Workforce Management - Person Real Time |
| Parameters | Business Unit, Effective Date |
| Output Format | Excel |
| Scheduled? (Y/N) | Yes - Weekly every Monday 8 AM |
| Security/Access | HR Specialist Role |

Business Purpose:

Used to validate active employees by business unit for payroll processing.

## Attachments

- HDL File (`ElementEntry.dat`)

- BI Report Definition (`.xdo`)

- User Guide

- UAT Sign-Off Sheet

## Notes & Dependencies

- Ensure employee assignments are loaded before element entries.  
- Report returns only active assignments within effective date range.

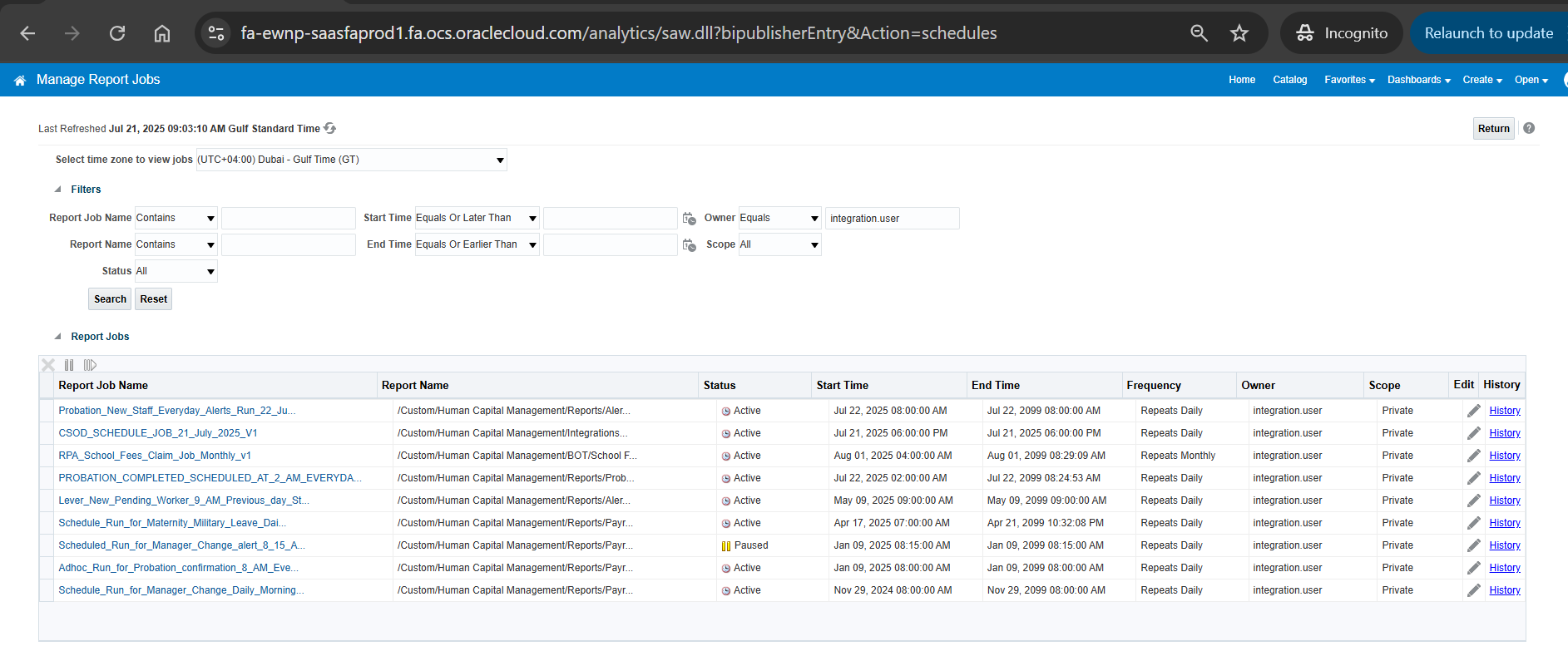
## Change Log

|  |  |  |
| --- | --- | --- |
| Date | Modified By | Change Summary |
| 2025-07-15 | Abdulrahman | Initial version created |

integration.user

details

following reports are scheduled using Integration. User



|  |  |  |
| --- | --- | --- |
| S.no | Report Name | Path |
| 1 | PROBATION\_ALERT | |  |  |  | | --- | --- | --- | |  |  | /Custom/Human Capital Management/Reports/Alerts/PROBATION\_ALERT.xdo | |
| 2 | ALD Employee Master Report CSOD | /Custom/Human Capital Management/Integrations/ALD Employee Master Report CSOD.xdo |
| 3 | School Fee Request Unprocess Entries Report | /Custom/Human Capital Management/BOT/School Fee Request Unprocess Entries Report.xdo |
| 4 | Probation status to performance management team | /Custom/Human Capital Management/Reports/Probation Reports/PerformanceDocumentWorkerAndManagerRatingsProb.xdo |
| 5 | Lever Integration Alert | /Custom/Human Capital Management/Reports/Alerts/ALDAR\_LEVER\_INTEGRATION\_ALERT.xdo |
| 6 | Maternity and Military Leave Alert | /Custom/Human Capital Management/Reports/Payroll Reports/Integrations/Alerts/ALDAR\_MAT\_MIL\_ALERT.xdo |
| 7 | Email Alert to employees on Probation confirmation or Extension | Custom/Human Capital Management/Reports/Payroll Reports/Integrations/Alerts/ALD Appraisal Probation Confirmation Extension Report.xdo |
| 8 | Manager Change Alert | /Custom/Human Capital Management/Reports/Payroll Reports/Integrations/Alerts/ALDAR\_NEW\_MANAGER\_ALERT.xdo |

task/process list what you are currently handling

Bank Account Reconciliation Report

PAM Access

Pam Access is given by Security Team and they can clone or grant the same user to anyone you want to assign so this user access details are not dependent upon me sharing credentials, this access can be given by asking security team to giving access my user on PAM Server

commission + costing month end process documents

shared it in the shared Folder

HDL List & files

All HDL files are shared on the Transition Knowledge Folder

list of interfaces/integrations

Name of Integration Status

1 Costing GL from Fusion to EBS Active ADBH still using

2 Bank Account Automation Active

3 Business Travel for Executive Dashboard Active

4 Fusion to EBS Employee Creation Active

5 Cornerstone Integration for Active Employees Active

6 Entomo Integration for Active employees Active

7 Visa Management System Not Active

8 OrgVue Integraton with Fusion HCM Active

9 Probation confirmation Active

10 Sailpoint Integration Active

11 Silent Bank Account Automation Active

12 Lever Integration automation Active

13 Hyperion Monthly Report Integration with Hyperion Active

14 Zoho Integration with Employee Absences Active

Technical deliverable Alerts documents

ALD Business Travel Perdiem Report

Data Model

Local query

select

papf.person\_number,

ppnf.full\_name,

pd.name department,

prrv.result\_value,

/\* pivt.name,\*/

prr.start\_date proration\_start\_date,

prr.end\_date proration\_end\_date,

abn.start\_date leave\_st\_date,

abn.end\_date leave\_ed\_date,

abn.ATTRIBUTE2 travel\_purpose,

abn.ATTRIBUTE5 destination,

abn.ATTRIBUTE\_DATE1 destination\_st\_date,

abn.ATTRIBUTE\_DATE2 destination\_ed\_date

from

per\_all\_people\_f papf,

per\_person\_names\_f ppnf,

per\_periods\_of\_service ppos,

per\_all\_assignments\_m asg,

per\_departments pd,

ANC\_PER\_ABS\_ENTRIES ABN,

ANC\_ABSENCE\_TYPES\_F\_TL ABTYP,

pay\_payroll\_assignments pasg,

pay\_payroll\_rel\_actions ppra,

pay\_payroll\_actions ppa,

pay\_run\_results prr,

pay\_run\_result\_values prrv,

pay\_element\_types\_tl petl,

pay\_input\_values\_tl pivt

where

papf.person\_id = ppnf.person\_id

and papf.person\_id = ppos.person\_id

and papf.person\_id = asg.person\_id

and asg.period\_of\_service\_id = ppos.period\_of\_service\_id

and asg.organization\_id = pd.organization\_id(+)

AND PAPF.PERSON\_ID = ABN.PERSON\_ID

AND ABN.ABSENCE\_TYPE\_ID = ABTYP.ABSENCE\_TYPE\_ID

and abtyp.language = 'US'

/\*AND ABTYP.NAME IN ('Business Travel')

and abn.ATTRIBUTE1 = 'Local Business Travel'

\*/

and asg.assignment\_id = pasg.hr\_assignment\_id

and pasg.payroll\_relationship\_id = ppra.payroll\_relationship\_id

and ppra.payroll\_action\_id = ppa.payroll\_action\_id

and prr.payroll\_rel\_action\_id = ppra.payroll\_rel\_action\_id

and prr.run\_result\_id = prrv.run\_result\_id

and prr.element\_type\_id = petl.element\_type\_id

and prrv.input\_value\_id = pivt.input\_value\_id

and petl.language = 'US'

and pivt.language = 'US'

and petl.element\_name

IN

(

'Overseas Business Travel Entitlement Result',

'Overseas Business Travel Entitlement Retroactive' ,

'Local and Overseas Business Travel Adjustment'

) /\*= 'Business Travel Entitlement Result' \*/

and pivt.name = 'Net Pay'

and ppra.run\_type\_id is not NULL

and ppra.retro\_component\_id is null

and ppa.action\_type in ('Q','R')

and ppa.action\_status = 'C'

and ppra.action\_status = 'C'

and pd.status='A' /\*Added by Abdulrahman 04-Jun-2025 \*/

and ppnf.name\_type = 'GLOBAL'

and asg.assignment\_type = 'E'

and asg.effective\_latest\_change = 'Y'

and asg.primary\_flag = 'Y'

and trunc(prr.start\_date) = trunc(abn.start\_date)

/\*and trim(abn.attribute1) = trim(:p\_travel\_type) \*/

and trunc(last\_day(ppa.effective\_date)) between papf.effective\_start\_date and papf.effective\_end\_date

and trunc(last\_day(ppa.effective\_date)) between ppnf.effective\_start\_date and ppnf.effective\_end\_date

and trunc(last\_day(ppa.effective\_date)) between asg.effective\_start\_date and asg.effective\_end\_date

and trunc(last\_day(ppa.effective\_date)) between abtyp.effective\_start\_date and abtyp.effective\_end\_date

and trunc(last\_day(ppa.effective\_date)) between pd.effective\_start\_date and pd.effective\_end\_date /\*Added by Abdulrahman 04-Jun-2025 \*/

and trunc(last\_day(ppa.effective\_date)) = TRUNC(last\_day(TO\_DATE(TO\_CHAR(:p\_date,'yyyy-MM-dd'),'yyyy-MM-dd') ) )

and papf.person\_number = nvl(:p\_person\_number,papf.person\_number)

and ppos.date\_start = (select max(ppos1.date\_start)

from

per\_periods\_of\_service ppos1

where

ppos.person\_id = ppos1.person\_id

and trunc(ppos1.date\_start) <= trunc(last\_day(ppa.effective\_date))

)

AND TO\_CHAR(ASG.business\_unit\_id) IN (

SELECT

TO\_CHAR(substr(a.ROLE\_NAME,7))

FROM

per\_roles\_dn\_vl a,

per\_user\_roles b,

per\_users c

WHERE

a.role\_id = b.role\_id

AND b.user\_id = c.user\_id

AND c.username = fnd\_global.user\_name

AND b.active\_flag = 'Y'

AND a.ROLE\_NAME IN ('CORPT-300000003250079',

'ADRES-300000003250653',

'GATWY-300000003250571',

'TEYSR-300000003250489',

'INVST-300000003250325',

'PRJCT-300000003250243',

'DEVLP-300000003250161',

'ADBUH-300000003250407',

'DUHLG-300000003250735')

)

Foreign

SELECT

x.person\_number,

x.full\_name,

x.element,

x.concatenated\_segments,

debit\_or\_credit,

DECODE(x.debit\_or\_credit,'C',x.costed\_value) credit\_value,

DECODE(x.debit\_or\_credit,'D',x.costed\_value) debit\_value,

to\_number(DECODE(x.debit\_or\_credit,'D',x.costed\_value,0)) -

to\_number(DECODE(x.debit\_or\_credit,'C',x.costed\_value,0)) NET,

X.INSTANCE\_NAME instance\_name,

x.payroll\_action\_id,

x.PAYROLL\_REL\_ACTION\_ID,

x.assignment\_id,

x.person\_id,

x.effective\_date,

x.account,

x.consolidation\_set\_name,

(select FLV.DESCRIPTION

from FND\_FLEX\_VALUES\_VL FLV,

FND\_FLEX\_VALUE\_SETS FLS

where

FLS.FLEX\_VALUE\_SET\_ID = FLV.FLEX\_VALUE\_SET\_ID

AND FLV.ENABLED\_FLAG = 'Y'

AND UPPER(FLS.FLEX\_VALUE\_SET\_NAME) = 'ALDAR\_GL\_ACCOUNT'

and FLV.FLEX\_VALUE = x.account

)account\_name,

(

SELECT DECODE(group1.segment1,'Y','yes','N','No')

FROM per\_people\_groups group1

WHERE group1.people\_group\_id (+) = x.people\_group\_id

) people\_grp\_uae\_national,

NVL((SELECT meaning FROM fnd\_lookup\_values WHERE language = 'US' AND lookup\_type = 'HZ\_DOMAIN\_SUFFIX\_LIST' AND lookup\_code =(select pc.legislation\_code from per\_citizenships pc where pc.person\_id = X.person\_id

and pc.CITIZENSHIP\_STATUS ='A'

AND NVL(pc.citizenship\_id,1) = NVL(

(SELECT MIN(citizenship\_id)

FROM per\_citizenships

WHERE person\_id = X.person\_id

AND citizenship\_status = 'A'

),1)

) and rownum=1) ,

(SELECT initcap(meaning)

FROM hr\_lookups

WHERE lookup\_type = 'NATIONALITY'

AND lookup\_code = (select pc.legislation\_code from per\_citizenships pc where pc.person\_id = X.person\_id

and pc.CITIZENSHIP\_STATUS ='A'

AND NVL(pc.citizenship\_id,1) = NVL(

(SELECT MIN(citizenship\_id)

FROM per\_citizenships

WHERE person\_id = X.person\_id

AND citizenship\_status = 'A'

),1)

)

AND enabled\_flag = 'Y' and rownum=1

)

) NATIONALITY,

( SELECT HAOU.Attribute1

FROM

HR\_ALL\_ORGANIZATION\_UNITS HAOU

where HAOU.ORGANIZATION\_ID = X.ASG\_ORGANIZATION\_ID

AND trunc(sysdate) BETWEEN HAOU.effective\_start\_date AND HAOU.effective\_end\_date) dept1,

( SELECT HAOU.Attribute2

FROM

HR\_ALL\_ORGANIZATION\_UNITS HAOU

where HAOU.ORGANIZATION\_ID = X.ASG\_ORGANIZATION\_ID

AND trunc(sysdate) BETWEEN HAOU.effective\_start\_date AND HAOU.effective\_end\_date) dept2,

( SELECT HAOU.Attribute3

FROM

HR\_ALL\_ORGANIZATION\_UNITS HAOU

where HAOU.ORGANIZATION\_ID = X.ASG\_ORGANIZATION\_ID

AND trunc(sysdate) BETWEEN HAOU.effective\_start\_date AND HAOU.effective\_end\_date) dept3,

x.element\_entry\_id

FROM

(

SELECT

papf.person\_number,

ppnf.full\_name,

ppa.effective\_date,

asg.people\_group\_id,

-- petl.element\_name element,

petl.REPORTING\_NAME element, ---Added By Abdulrahman on 28-Feb-2024 for LTIP Employee Contribution to show instead of Other Deduction Results element

TRIM(pca.segment1

|| '.'

|| pca.segment2

|| '.'

|| pca.segment3

|| '.'

|| pca.segment4

|| '.'

|| pca.segment5

|| '.'

|| pca.segment6

||'.'

||pca.segment7

||'.'

||pca.segment8

) concatenated\_segments,

debit\_or\_credit,

SUM(costed\_value) costed\_value,

PLE.organization\_id,

PFI.INSTANCE\_NAME,

pca.segment4 account,

pcs.consolidation\_set\_name,

ppa.payroll\_action\_id,

papf.person\_id,

asg.ORGANIZATION\_ID ASG\_ORGANIZATION\_ID,

ppar.PAYROLL\_REL\_ACTION\_ID,

asg.assignment\_id

,

(select prr.element\_entry\_id from pay\_payroll\_actions ppa, PAY\_PAYROLL\_REL\_ACTIONS ppra, PAY\_PAY\_RELATIONSHIPS\_DN pprd , pay\_run\_results prr where

ppa.payroll\_action\_id = ppra.payroll\_action\_id

and ppra.PAYROLL\_RELATIONSHIP\_ID= pprd.PAYROLL\_RELATIONSHIP\_ID

and prr.payroll\_rel\_action\_id = ppra.payroll\_rel\_action\_id

and sysdate between pprd.start\_date and pprd.end\_date

and pprd.person\_id = papf.person\_id and rownum=1

) element\_entry\_id

FROM

per\_all\_assignments\_m asg,

per\_all\_people\_f papf,

per\_periods\_of\_service ppos,

per\_person\_names\_f ppnf,

fun\_all\_business\_units\_v fbu,

pay\_costs pc,

pay\_cost\_allocation\_keyflex pca,

pay\_payroll\_rel\_actions ppar,

pay\_payroll\_actions ppa,

pay\_rel\_groups\_dn rel,

pay\_requests pr,

pay\_flow\_instances pfi,

pay\_consolidation\_sets pcs,

pay\_input\_values\_f pivf,

pay\_element\_types\_tl petl,

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Added Element link table\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

-- pay\_element\_links\_f pel,

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Added Element link table\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

per\_legal\_employers ple

WHERE

papf.person\_id = asg.person\_id

and papf.person\_id = ppnf.person\_id

AND papf.person\_id = ppos.person\_id

AND ppos.period\_of\_service\_id = asg.period\_of\_service\_id

and fbu.bu\_id(+) =asg.business\_unit\_id

AND rel.assignment\_id = asg.assignment\_id

AND ppar.payroll\_relationship\_id = rel.payroll\_relationship\_id

AND ppa.payroll\_action\_id = ppar.payroll\_action\_id

AND ppar.payroll\_rel\_action\_id = pc.payroll\_rel\_action\_id

and pca.cost\_allocation\_keyflex\_id = pc.cost\_allocation\_keyflex\_id

AND pr.pay\_request\_id (+) = ppa.pay\_request\_id

AND pfi.flow\_instance\_id (+) = pr.flow\_instance\_id

and pcs.consolidation\_set\_id = ppa.consolidation\_set\_id

AND pivf.input\_value\_id = pc.input\_value\_id

AND pivf.element\_type\_id = petl.element\_type\_id

AND ple.organization\_id = asg.legal\_entity\_id

AND ppa.action\_status = 'C'

-- AND ppa.run\_type\_id IS NOT NULL

AND ppar.retro\_component\_id IS NULL

AND petl.language = 'US'

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Added Element link table\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

-- and pel.element\_type\_id = petl.element\_type\_id

-- and pel.effective\_end\_date =to\_date('4712-12-31','YYYY-MM-DD')

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Added Element link table\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

AND ppa.action\_type IN ('Q','R','S' )

--and petl.REPORTING\_NAME = 'Annual Leave Accrual Result'

and ppnf.name\_type = 'GLOBAL'

AND asg.assignment\_type = 'E'

AND asg.primary\_flag = 'Y'

AND asg.effective\_latest\_change = 'Y'

AND least(nvl(ppos.actual\_termination\_date, trunc(last\_day(ppa.effective\_date))),trunc(last\_day(ppa.effective\_date))) BETWEEN asg.effective\_start\_date AND asg.effective\_end\_date

AND least(nvl(ppos.actual\_termination\_date, trunc(last\_day(ppa.effective\_date))),trunc(last\_day(ppa.effective\_date))) BETWEEN papf.effective\_start\_date AND papf.effective\_end\_date

AND least(nvl(ppos.actual\_termination\_date, trunc(last\_day(ppa.effective\_date))),trunc(last\_day(ppa.effective\_date))) BETWEEN ppnf.effective\_start\_date AND ppnf.effective\_end\_date

AND least(nvl(ppos.actual\_termination\_date, trunc(last\_day(ppa.effective\_date))),trunc(last\_day(ppa.effective\_date))) BETWEEN pivf.effective\_start\_date AND pivf.effective\_end\_date

AND least(nvl(ppos.actual\_termination\_date, trunc(last\_day(ppa.effective\_date))),trunc(last\_day(ppa.effective\_date))) BETWEEN ple.effective\_start\_date AND ple.effective\_end\_date

and trunc(last\_day(ppa.effective\_date)) = trunc(last\_day(TO\_DATE(TO\_CHAR(:p\_date,'yyyy-MM-dd'),'yyyy-MM-dd')))

/\* and fbu.bu\_name = nvl(:p\_bu\_name,fbu.bu\_name) \*/

AND ppos.date\_start = (

SELECT

MAX(ppos1.date\_start)

FROM

per\_periods\_of\_service ppos1

WHERE

ppos1.person\_id = ppos.person\_id

-- AND ppos1.period\_of\_service\_id = asg.period\_of\_service\_id

AND ppos1.date\_start <= trunc(last\_day(ppa.effective\_date))

AND TO\_CHAR(ASG.business\_unit\_id) IN (

SELECT

TO\_CHAR(substr(a.ROLE\_NAME,7))

FROM

per\_roles\_dn\_vl a,

per\_user\_roles b,

per\_users c

WHERE

a.role\_id = b.role\_id

AND b.user\_id = c.user\_id

AND c.username = fnd\_global.user\_name

AND b.active\_flag = 'Y'

AND a.ROLE\_NAME IN ('CORPT-300000003250079',

'ADRES-300000003250653',

'GATWY-300000003250571',

'TEYSR-300000003250489',

'INVST-300000003250325',

'PRJCT-300000003250243',

'DEVLP-300000003250161',

'ADBUH-300000003250407',

'DUHLG-300000003250735')

)

)

GROUP BY

papf.person\_number,

ppnf.full\_name,

-- petl.element\_name,

petl.REPORTING\_NAME,

TRIM(pca.segment1

|| '.'

|| pca.segment2

|| '.'

|| pca.segment3

|| '.'

|| pca.segment4

|| '.'

|| pca.segment5

|| '.'

|| pca.segment6

||'.'

||pca.segment7

||'.'

||pca.segment8

),

debit\_or\_credit,

PLE.organization\_id,

pfi.instance\_name,

ppa.payroll\_action\_id,

ppa.effective\_date,

pca.segment4,

pcs.consolidation\_set\_name,

asg.people\_group\_id,

papf.person\_id,

asg.ORGANIZATION\_ID,

asg.assignment\_id,

ppar.PAYROLL\_REL\_ACTION\_ID

) x

where 1=2 and x.element IN

('Overseas Business Travel Entitlement Result',

'Overseas Business Travel Entitlement Retroactive' ,

'Local and Overseas Business Travel Adjustment' /\* ,

'Local Business Travel One Day Entitlement Result',

'Local Business Travel More Than 1 Day Entitlement Result',

'Local Business Travel More Than 1 Day Entitlement Retroactive',

'Local Business Travel One Day Entitlement Retroactive' \*/

)

Layout



ALD Appraisal Probation Confirmation Extension Report

Data Model

select \* from (SELECT TO\_CHAR(SYSDATE,'DD-Mon-RRRR', 'NLS\_DATE\_LANGUAGE = American') TIME, PNAME.FULL\_NAME, PNAME.FIRST\_NAME,

to\_char(paam.DATE\_PROBATION\_END+90,'DD-MON-YYYY', 'NLS\_DATE\_LANGUAGE = American') PROBATION\_EXTENSION\_END\_DATE,

(select name from hr\_all\_positions\_f\_tl hapft where position\_id = PAAM.POSITION\_ID and

hapft.language='US' and sysdate between EFFECTIVE\_START\_DATE and EFFECTIVE\_END\_DATE ) POSITION,

(select pea\_emp.EMAIL\_ADDRESS from PER\_EMAIL\_ADDRESSES pea\_emp where pea\_emp.person\_id = paam.person\_id and pea\_emp.EMAIL\_TYPE='W1' and rownum=1) EMP\_EMAIL\_ADDRESS ,

(select pea\_emp.EMAIL\_ADDRESS from PER\_EMAIL\_ADDRESSES pea\_emp , PER\_ASSIGNMENT\_SUPERVISORS\_F pasf\_new where pea\_emp.person\_id = pasf\_new.manager\_id and pasf\_new.person\_id = paam.person\_id

and sysdate between pasf\_new.EFFECTIVE\_START\_DATE

and pasf\_new.effective\_end\_date

AND pea\_emp.EMAIL\_TYPE='W1' and rownum=1) NEW\_MANAGER\_EMAIL\_ADDRESS,

PAAM.probation\_period,

DECODE(PAAM.probation\_unit,'D','Days','M','Months') units\_probatn, to\_char(paam.DATE\_PROBATION\_END ,'DD-MON-YYYY', 'NLS\_DATE\_LANGUAGE = American') DATE\_PROBATION\_END,

(select to\_char(min(date\_start),'DD-MON-YYYY', 'NLS\_DATE\_LANGUAGE = American') from per\_periods\_of\_service ppos where ppos.PERIOD\_TYPE ='E' and ppos.period\_of\_service\_id = paam.period\_of\_service\_id ) date\_PROBATION\_start,

RatingLevelPEO.RATING\_DESCRIPTION RATING,

TO\_CHAR(EvaluationEO.last\_update\_date,'YYYY-MM-DD') last\_update\_date ,

EvaluationEO.EVALUATION\_ID,

(SELECT acl.action\_name FROM per\_actions\_vl acl WHERE acl.action\_id = (

SELECT ac.action\_id FROM per\_action\_occurrences ac WHERE ac.action\_occurrence\_id = (

(SELECT MAX(i.action\_occurrence\_id)

FROM per\_periods\_of\_service i

WHERE i.person\_id = PAAM.person\_id

AND i.period\_of\_service\_id = PAAM.period\_of\_service\_id

AND i.primary\_flag = 'Y'

AND i.period\_type IN ('E', 'C')

)))) ACTION\_CODE

FROM PER\_ALL\_ASSIGNMENTS\_M PAAM,

per\_person\_names\_f pname,

HRA\_EVALUATIONS EvaluationEO,

HRA\_EVAL\_SECTIONS EvalSectionEO,

HRA\_EVAL\_RATINGS EvalRatingEO,

HRT\_RATING\_LEVELS\_VL RatingLevelPEO,

HRA\_EVAL\_PARTICIPANTS EvalParticipantEO

WHERE PAAM.PERSON\_ID = PNAME.PERSON\_ID

AND PAAM.PERSON\_ID = EvaluationEO.WORKER\_ID

AND EvaluationEO.EVALUATION\_ID=EvalSectionEO.EVALUATION\_ID

AND RatingLevelPEO.BUSINESS\_GROUP\_ID = EvalRatingEO.BUSINESS\_GROUP\_ID AND

EvalSectionEO.SECTION\_TYPE\_CODE ='OS' AND

EvalSectionEO.EVAL\_SECTION\_ID=EvalRatingEO.REFERENCE\_ID AND

EvalRatingEO.REFERENCE\_TYPE='SECTION' AND

EvaluationEO.EVALUATION\_ID=EvalParticipantEO.EVALUATION\_ID AND

EvaluationEO.MANAGER\_ID=EvalParticipantEO.PERSON\_ID AND

EvalRatingEO.EVAL\_PARTICIPANT\_ID=EvalParticipantEO.EVAL\_PARTICIPANT\_ID AND

EvalRatingEO.PERFORMANCE\_RATING\_ID=RatingLevelPEO.RATING\_LEVEL\_ID

/\* AND RatingLevelPEO.RATING\_DESCRIPTION ='Extension of Probation' \*/

AND TO\_CHAR(EvaluationEO.last\_update\_date,'YYYY-MM-DD') = to\_char(SYSDATE-1,'YYYY-MM-DD')

/\* to be uncommented after one time run 13May2025

and paam.person\_id in (select person\_id from per\_all\_people\_f where person\_number='3330')

and EvaluationEO.last\_update\_date between sysdate-90 and sysdate

\*/

AND RatingLevelPEO.RATING\_DESCRIPTION IN ( 'Satisfactory Completion of Probation', 'Extension of Probation' )

AND EvaluationEO.STATUS\_CODE='COMP'

/\*AND EvaluationEO.EVALUATION\_ID in (300000047544529 , 300000043050205 ) \*/

AND EvalParticipantEO.ROLE\_TYPE\_CODE='MANAGER'

AND EvalParticipantEO.PARTICIPATION\_STATUS\_CODE='COMP'

AND pname.name\_type = 'GLOBAL'

AND SYSDATE BETWEEN pname.EFFECTIVE\_START\_DATE and pname.EFFECTIVE\_END\_DATE

AND PAAM.assignment\_type IN ( 'E','P' )

AND PAAM.primary\_flag = 'Y'

AND PAAM.effective\_latest\_change = 'Y'

AND PAAM.assignment\_status\_type = 'ACTIVE'

AND sysdate between paam.effective\_start\_date and paam.effective\_end\_date

) where action\_code ='End Probation Period' /\* not in ('Resignation', 'Global Transfer') \*/

Layout



Aldar Probation Expiry Alert to Manager

Data Model

select

d.employee\_number,

d.employee\_name,

d.date\_probation\_end,

d.line\_manager\_email,

d.line\_manager\_name,

d.PROBATION\_END\_DATE,

d.bu\_name,

d.legal\_employer,

d.HRBP\_name,

d.HRBP\_RESPONSIBILITY\_name,

d.HRBP\_email,

d.raman\_Alzuraiqi\_emails

from

(

select

papf.person\_number employee\_number,

ppnf.display\_name employee\_name,

papf.START\_DATE,

(select

pjft.name

from

per\_jobs\_f\_tl pjft

where

pjft.job\_id = paam.job\_id

AND pjft.language= 'US'

AND TRUNC(SYSDATE) BETWEEN pjft.effective\_start\_date AND pjft.effective\_end\_date) job\_name,

paam.date\_probation\_end,

to\_char(paam.date\_probation\_end,'dd-Mon-yyyy','NLS\_DATE\_LANGUAGE = AMERICAN') probation\_end\_date,

(SELECT

LISTAGG(pea.email\_address,',') WITHIN GROUP(ORDER BY pea.email\_address)

FROM

per\_all\_people\_f papf3,

per\_assignment\_supervisors\_f pasf,

per\_email\_addresses pea

WHERE

pasf.manager\_id = papf3.person\_id

AND pasf.person\_id = papf.person\_id

AND trunc(SYSDATE) BETWEEN pasf.effective\_start\_date AND pasf.effective\_end\_date

AND trunc(SYSDATE) BETWEEN papf3.effective\_start\_date AND papf3.effective\_end\_date

and pasf.manager\_type = 'LINE\_MANAGER'

AND pea.email\_address\_id = papf3.primary\_email\_id )line\_manager\_email,

(SELECT

ppnf.full\_name

FROM

per\_person\_names\_f ppnf,

per\_assignment\_supervisors\_f pasf

WHERE

pasf.manager\_id = ppnf.person\_id

AND pasf.person\_id = papf.person\_id

and pasf.assignment\_id = paam.assignment\_id

AND trunc(SYSDATE) BETWEEN pasf.effective\_start\_date AND pasf.effective\_end\_date

AND ppnf.name\_type = 'GLOBAL'

AND trunc(SYSDATE) BETWEEN ppnf.effective\_start\_date AND ppnf.effective\_end\_date

AND pasf.manager\_type = 'LINE\_MANAGER')line\_manager\_name,

ple.name legal\_employer,

fabu.bu\_name,

(SELECT

-- ppnfv.full\_name

LISTAGG(distinct ppnfv.full\_name ,',') WITHIN GROUP(ORDER BY ppnfv.full\_name )

FROM

per\_person\_names\_f\_v ppnfv,

per\_asg\_responsibilities par,

per\_all\_people\_f papfc

WHERE

papfc.person\_id = par.person\_id

AND ppnfv.person\_id = papfc.person\_id

AND par.RESPONSIBILITY\_TYPE = 'ALDAR\_HRBP'

AND SYSDATE BETWEEN ppnfv.effective\_start\_date AND ppnfv.effective\_end\_date

AND SYSDATE BETWEEN papfc.effective\_start\_date AND papfc.effective\_end\_date

AND nvl(par.business\_unit\_id,nvl(paam.business\_unit\_id,1) ) = nvl(paam.business\_unit\_id,1)

AND nvl(par.legal\_entity\_id,nvl(paam.legal\_entity\_id,1) ) = nvl(paam.legal\_entity\_id,1)

AND nvl(par.organization\_id,nvl(paam.organization\_id,1) ) = nvl(paam.organization\_id,1)

AND nvl(par.location\_id,nvl(paam.location\_id,1) ) = nvl(paam.location\_id,1)

AND nvl(par.position\_id,nvl(paam.position\_id,1) ) = nvl(paam.position\_id,1)

AND nvl(par.job\_id,nvl(paam.job\_id,1) ) = nvl(paam.job\_id,1)

)HRBP\_name,

(SELECT

LISTAGG(distinct pea.email\_address,',') WITHIN GROUP(ORDER BY pea.email\_address)

FROM

per\_person\_names\_f\_v ppnfv,

per\_asg\_responsibilities par,

per\_all\_people\_f papfc,

per\_email\_addresses pea

WHERE

papfc.person\_id = par.person\_id

AND ppnfv.person\_id = papfc.person\_id

AND pea.email\_address\_id = papfc.primary\_email\_id

AND par.RESPONSIBILITY\_TYPE = 'ALDAR\_HRBP'

AND SYSDATE BETWEEN ppnfv.effective\_start\_date AND ppnfv.effective\_end\_date

AND SYSDATE BETWEEN papfc.effective\_start\_date AND papfc.effective\_end\_date

AND nvl(par.business\_unit\_id,nvl(paam.business\_unit\_id,1) ) = nvl(paam.business\_unit\_id,1)

AND nvl(par.legal\_entity\_id,nvl(paam.legal\_entity\_id,1) ) = nvl(paam.legal\_entity\_id,1)

AND nvl(par.organization\_id,nvl(paam.organization\_id,1) ) = nvl(paam.organization\_id,1)

AND nvl(par.location\_id,nvl(paam.location\_id,1) ) = nvl(paam.location\_id,1)

AND nvl(par.position\_id,nvl(paam.position\_id,1) ) = nvl(paam.position\_id,1)

AND nvl(par.job\_id,nvl(paam.job\_id,1) ) = nvl(paam.job\_id,1)

)HRBP\_email,

(SELECT

--par.RESPONSIBILITY\_name

LISTAGG(distinct par.RESPONSIBILITY\_name,',') WITHIN GROUP(ORDER BY par.RESPONSIBILITY\_name)

FROM

per\_person\_names\_f\_v ppnfv,

per\_asg\_responsibilities par,

per\_all\_people\_f papfc

WHERE

papfc.person\_id = par.person\_id

AND ppnfv.person\_id = papfc.person\_id

AND par.RESPONSIBILITY\_TYPE = 'ALDAR\_HRBP'

AND SYSDATE BETWEEN ppnfv.effective\_start\_date AND ppnfv.effective\_end\_date

AND SYSDATE BETWEEN papfc.effective\_start\_date AND papfc.effective\_end\_date

AND nvl(par.business\_unit\_id,nvl(paam.business\_unit\_id,1) ) = nvl(paam.business\_unit\_id,1)

AND nvl(par.legal\_entity\_id,nvl(paam.legal\_entity\_id,1) ) = nvl(paam.legal\_entity\_id,1)

AND nvl(par.organization\_id,nvl(paam.organization\_id,1) ) = nvl(paam.organization\_id,1)

AND nvl(par.location\_id,nvl(paam.location\_id,1) ) = nvl(paam.location\_id,1)

AND nvl(par.position\_id,nvl(paam.position\_id,1) ) = nvl(paam.position\_id,1)

AND nvl(par.job\_id,nvl(paam.job\_id,1) ) = nvl(paam.job\_id,1)

)HRBP\_RESPONSIBILITY\_name,

-----Raman Ganes,Ahmad Alzuraiqi emails----

(SELECT

LISTAGG(pea.email\_address,',') WITHIN GROUP(ORDER BY pea.email\_address)

FROM

per\_all\_people\_f papf3,

per\_email\_addresses pea

WHERE

papf3.person\_number in ('511','408')

AND trunc(SYSDATE) BETWEEN papf3.effective\_start\_date AND papf3.effective\_end\_date

AND pea.email\_address\_id = papf3.primary\_email\_id

)raman\_Alzuraiqi\_emails

FROM

per\_all\_people\_f papf,

per\_all\_assignments\_m paam,

per\_person\_names\_f ppnf,

fun\_all\_business\_units\_v fabu,

PER\_LEGAL\_EMPLOYERS ple

where

papf.person\_id = ppnf.person\_id

AND papf.person\_id = paam.person\_id

and fabu.bu\_id=paam.business\_unit\_id

and ple.organization\_id(+) = paam.legal\_entity\_id

AND ppnf.name\_type = 'GLOBAL'

AND paam.primary\_flag = 'Y'

AND paam.effective\_latest\_change = 'Y'

AND paam.assignment\_status\_type = 'ACTIVE'

AND paam.assignment\_type = 'E'

AND trunc(SYSDATE) BETWEEN papf.effective\_start\_date AND papf.effective\_end\_date

AND trunc(SYSDATE) BETWEEN ppnf.effective\_start\_date AND ppnf.effective\_end\_date

AND trunc(SYSDATE) BETWEEN paam.effective\_start\_date AND paam.effective\_end\_date

AND TRUNC(sysdate) BETWEEN ple.effective\_start\_date AND ple.effective\_end\_date

order by to\_number( papf.person\_number )

)d

where

1=1

and trunc(date\_probation\_end) = trunc(sysdate)+14

and d.date\_probation\_end is not null

and d.line\_manager\_email is not null

order by to\_number(d.employee\_number)

Layout



ALDAR\_LEVER\_INTEGRATION\_ALERT

Data Model

/\*

CREATION DATE:11-SEP-2024

CREATED BY: ABDULRAHMAN MOHAMMED

DESCRIPTION: New User in Fusion Created with Lever Integration with Fusion

MODIFIED BY :

MODIFIED DATE:

MODIFICATION REASON:

\*/

SELECT XX.\*

FROM

(SELECT papf.PERSON\_NUMBER EMPLOYEE\_NUMBER,

TO\_CHAR(SYSDATE,'DDMMRRHH24MI') TIME,

PNAME.display\_name EMP\_NAME,

(select

ple.name

from PER\_LEGAL\_EMPLOYERS ple

where ple.ORGANIZATION\_ID = asg.LEGAL\_ENTITY\_ID

and sysdate between ple.effective\_start\_date and ple.effective\_end\_date

) legal\_employer ,

(select

ple.name

from PER\_LEGAL\_EMPLOYERS ple

where ple.ORGANIZATION\_ID = asg.LEGAL\_ENTITY\_ID

and sysdate between ple.effective\_start\_date and ple.effective\_end\_date

) MOL\_ENTITY,

( select

bu.bu\_name

from fun\_all\_business\_units\_v bu

where

1=1

AND bu.bu\_id = asg.business\_unit\_id

) BUSINESS\_UNIT,

TO\_CHAR((SELECT MIN(date\_start)

FROM PER\_PERIODS\_OF\_SERVICE ppos1

WHERE ppos1.person\_id = ppos.person\_id

),'DD-Mon-YYYY', 'NLS\_DATE\_LANGUAGE = American') DOJ,

POS.NAME POSITION,

ORG.name DEPARTMENT,

(select EMAIL\_ADDRESS

from

PER\_EMAIL\_ADDRESSES

where PERSON\_ID= papf.PERSON\_ID

AND EMAIL\_TYPE ='W1'

and rownum =1

) EMAIL\_ADDRESS,

PAPF.CREATION\_DATE,

ASG.ASSIGNMENT\_TYPE ,ASG.PRIMARY\_FLAG , asg.EFFECTIVE\_LATEST\_CHANGE , asg.assignment\_status\_type, ppos.date\_start

FROM PER\_PERSON\_NAMES\_F PNAME,

PER\_ALL\_PEOPLE\_F PAPF,

PER\_ALL\_ASSIGNMENTS\_M ASG,

PER\_PERIODS\_OF\_SERVICE PPOS,

HR\_ALL\_ORGANIZATION\_UNITS ORG,

HR\_ALL\_POSITIONS\_F\_TL POS,

HRC\_INTEGRATION\_KEY\_MAP HIKM

WHERE PNAME.PERSON\_ID = PAPF.PERSON\_ID

AND ASG.PERSON\_ID = PAPF.PERSON\_ID

AND PPOS.PERSON\_ID = PAPF.PERSON\_ID

AND HIKM.surrogate\_id = papf.PERSON\_ID

AND HIKM.object\_name = 'Person'

AND HIKM.SOURCE\_SYSTEM\_OWNER='LEVER'

AND TRUNC(PAPF.CREATION\_DATE) = TRUNC(SYSDATE) /\* to be executed for same day\*/

AND ORG.ORGANIZATION\_ID(+) = ASG.ORGANIZATION\_ID

AND ASG.position\_id =POS.position\_id(+)

AND NVL(POS.language,'US') ='US'

AND TRUNC(sysdate) BETWEEN papf.EFFECTIVE\_START\_DATE AND papf.EFFECTIVE\_END\_DATE

AND TRUNC(sysdate) BETWEEN PNAME.EFFECTIVE\_START\_DATE AND PNAME.EFFECTIVE\_END\_DATE

AND TRUNC(sysdate) BETWEEN ASG.EFFECTIVE\_START\_DATE AND ASG.EFFECTIVE\_END\_DATE

AND TRUNC(sysdate) BETWEEN POS.effective\_start\_date AND POS.effective\_end\_date

AND papf.PERSON\_NUMBER = NVL(:P\_EMPNO,papf.PERSON\_NUMBER )

/\* AND NVL(org.organization\_id,00000) =NVL(:p\_dept,NVL(org.organization\_id,00000))

AND NVL(:p\_date\_from,SYSDATE) <=NVL(:p\_date\_to,SYSDATE) \*/

AND asg.business\_unit\_id = NVL(:p\_emp\_cat,asg.business\_unit\_id)

AND PNAME.NAME\_TYPE = 'GLOBAL'

AND ASG.ASSIGNMENT\_TYPE = 'P'

AND ASG.PRIMARY\_FLAG = 'N'

AND asg.EFFECTIVE\_LATEST\_CHANGE = 'Y'

AND asg.assignment\_status\_type ='ACTIVE'

AND ppos.date\_start =

(SELECT MAX(date\_start)

FROM PER\_PERIODS\_OF\_SERVICE ppos1

WHERE ppos1.person\_id = ppos.person\_id

)

ORDER BY to\_number(employee\_number)

)XX

Layout



ALDAR\_NEW\_MANAGER\_ALERT

SELECT

\*

from (

SELECT TO\_CHAR(SYSDATE,'DDMMRRHH24MI') TIME,

(SELECT listagg(meaning,',') EMAIL\_GROUP\_CC\_HRBP

FROM fnd\_lookup\_values flv

WHERE flv.enabled\_flag = 'Y'

AND flv.LANGUAGE = 'US'

AND FLV.lookup\_type='ALDAR\_HRBP\_LEVER\_NEW\_EMP\_GROUP'

) EMAIL\_GROUP\_CC\_HRBP,

(SELECT listagg(meaning,',') EMAIL\_GROUP\_CC\_HRBP

FROM fnd\_lookup\_values flv

WHERE flv.enabled\_flag = 'Y'

AND flv.LANGUAGE = 'US'

AND FLV.lookup\_type='ADRES\_HRBP\_NEW\_EMP\_GROUP'

) ADRES\_EMAIL\_GROUP\_CC\_HRBP,

(SELECT listagg(meaning,',') EMAIL\_GROUP\_CC\_HRBP

FROM fnd\_lookup\_values flv

WHERE flv.enabled\_flag = 'Y'

AND flv.LANGUAGE = 'US'

AND FLV.lookup\_type='C2I\_HRBP\_NEW\_EMP\_GROUP'

) C2I\_EMAIL\_GROUP\_CC\_HRBP,

-- NVL((

-- SELECT NVL(flv.description,'ALDAR') FROM FND\_LOOKUP\_VALUES flv, PER\_LEGAL\_EMPLOYERS ple WHERE trim(upper(ple.name)) = trim(upper(flv.meaning))

-- and ha.LEGAL\_ENTITY\_ID = ple.ORGANIZATION\_ID

-- and flv.LOOKUP\_TYPE ='ALDAR\_MOL\_ENTITY' and ple.status='A' and rownum =1 ),'ADRES') business\_unit,

(select

ple.name

from

PER\_LEGAL\_EMPLOYERS ple

where

ple.organization\_id = ha.LEGAL\_ENTITY\_ID

and trunc(sysdate) between ple.effective\_start\_date and ple.effective\_end\_date

)business\_unit,

(select papf.person\_number from per\_all\_people\_f papf where papf.person\_id = ha.person\_id and rownum=1) employee\_number ,

(select ppnf.full\_name from per\_person\_names\_f ppnf where ppnf.person\_id = ha.person\_id and ppnf.name\_type='GLOBAL' and rownum=1) employee\_name ,

(select pea\_emp.EMAIL\_ADDRESS from PER\_EMAIL\_ADDRESSES pea\_emp where pea\_emp.person\_id = ha.person\_id and pea\_emp.EMAIL\_TYPE='W1' and rownum=1) EMP\_EMAIL\_ADDRESS ,

(select ppnf.full\_name from per\_person\_names\_f ppnf where ppnf.person\_id = pasf\_new.manager\_id and ppnf.name\_type='GLOBAL' and rownum=1) AS new\_manager\_name ,

(select pea\_emp.EMAIL\_ADDRESS from PER\_EMAIL\_ADDRESSES pea\_emp where pea\_emp.person\_id = pasf\_old.manager\_id and pea\_emp.EMAIL\_TYPE='W1' and rownum=1) OLD\_MANAGER\_EMAIL\_ADDRESS,

(select pea\_emp.EMAIL\_ADDRESS from PER\_EMAIL\_ADDRESSES pea\_emp where pea\_emp.person\_id = pasf\_new.manager\_id and pea\_emp.EMAIL\_TYPE='W1' and rownum=1) NEW\_MANAGER\_EMAIL\_ADDRESS,

(select ppnf.full\_name from per\_person\_names\_f ppnf where ppnf.person\_id = pasf\_old.manager\_id and ppnf.name\_type='GLOBAL' and rownum=1) AS old\_manager\_name,

to\_char(pasf\_new.EFFECTIVE\_START\_DATE,'DD-Mon-YYYY', 'NLS\_DATE\_LANGUAGE = American') AS change\_date

FROM

per\_all\_assignments\_m ha,

PER\_ASSIGNMENT\_SUPERVISORS\_F pasf\_old ,

PER\_ASSIGNMENT\_SUPERVISORS\_F pasf\_new

where ha.assignment\_id =pasf\_new.assignment\_id

and ha.person\_id = pasf\_new.person\_id

and trunc(pasf\_new.EFFECTIVE\_START\_DATE) = trunc(sysdate-1)

and trunc(pasf\_old.effective\_end\_date) = trunc(sysdate-2)

and ha.assignment\_id =pasf\_old.assignment\_id(+)

and ha.person\_id = pasf\_old.person\_id(+)

and ha.assignment\_type IN ( 'E','P' )

AND ha.primary\_flag = 'Y'

AND ha.effective\_latest\_change = 'Y'

AND ha.assignment\_status\_type = 'ACTIVE'

and sysdate between ha.effective\_start\_date and ha.effective\_end\_date

)

where business\_unit like 'Aldar%'

Layout



ALDAR\_MAT\_MIL\_ALERT

This report is scheduled to send notification for staff who have started Maternity or Military Leave

Data Model

/\*

CREATION DATE:17-MAR-2016

CREATED BY: V.SRIDURGA

DESCRIPTION: Leave Details Report

MODIFIED BY :

MODIFIED DATE:

MODIFICATION REASON:

\*/

SELECT XX.\*

FROM

(SELECT papf.PERSON\_NUMBER EMPLOYEE\_NUMBER,

TO\_CHAR(SYSDATE,'DDMMRRHH24MI') TIME,

A1.NAME ABSENCE\_TYPE,

TO\_CHAR(A2.START\_DATE,'dd-Mon-yyyy') start\_dt,

TO\_CHAR(A2.END\_DATE,'dd-Mon-yyyy') end\_dt,

/\*TO\_CHAR(to\_date(TO\_CHAR(:p\_date\_to,'yyyy-MM-dd'),'yyyy-MM-dd'),'dd-Mon-yyyy') to\_date,

TO\_CHAR(to\_date(TO\_CHAR(:p\_date\_from,'yyyy-MM-dd'),'yyyy-MM-dd'),'dd-Mon-yyyy') from\_date,\*/

PNAME.display\_name EMP\_NAME,

TO\_CHAR(PPOS.DATE\_START,'DD-Mon-YYYY') DOJ,

POS.NAME POSITION,

ORG.name DEPARTMENT,

(select EMAIL\_ADDRESS

from

PER\_EMAIL\_ADDRESSES

where PERSON\_ID= papf.PERSON\_ID

AND EMAIL\_TYPE ='W1'

and rownum =1

) EMAIL\_ADDRESS,

a2.duration,

(

CASE

WHEN --(TRUNC(to\_date(TO\_CHAR(A2.START\_DATE,'yyyy-MM-dd'),'yyyy-MM-dd'))>=(NVL(TRUNC(to\_date(TO\_CHAR(:p\_date\_from,'yyyy-MM-dd'),'yyyy-MM-dd')),TRUNC(to\_date(TO\_CHAR(sysdate,'yyyy-MM-dd'),'yyyy-MM-dd')))))

(TRUNC(to\_date(TO\_CHAR(A2.END\_DATE,'yyyy-MM-dd'),'yyyy-MM-dd'))<=(NVL(TRUNC(to\_date(TO\_CHAR(:p\_date\_to,'yyyy-MM-dd'),'yyyy-MM-dd')),TRUNC(to\_date(TO\_CHAR(sysdate,'yyyy-MM-dd'),'yyyy-MM-dd')))))

THEN 'COMPLETED'

WHEN (TRUNC(to\_date(TO\_CHAR(A2.START\_DATE,'yyyy-MM-dd'),'yyyy-MM-dd'))>=(NVL(TRUNC(to\_date(TO\_CHAR(:p\_date\_to,'yyyy-MM-dd'),'yyyy-MM-dd')),TRUNC(to\_date(TO\_CHAR(sysdate,'yyyy-MM-dd'),'yyyy-MM-dd')))))

THEN 'PLANNED'

WHEN (TRUNC(to\_date(TO\_CHAR(A2.START\_DATE,'yyyy-MM-dd'),'yyyy-MM-dd'))<=(NVL(TRUNC(to\_date(TO\_CHAR(:p\_date\_to,'yyyy-MM-dd'),'yyyy-MM-dd')),TRUNC(to\_date(TO\_CHAR(sysdate,'yyyy-MM-dd'),'yyyy-MM-dd')))))

AND (TRUNC(to\_date(TO\_CHAR(A2.END\_DATE,'yyyy-MM-dd'),'yyyy-MM-dd')) >=(NVL(TRUNC(to\_date(TO\_CHAR(:p\_date\_to,'yyyy-MM-dd'),'yyyy-MM-dd')),TRUNC(to\_date(TO\_CHAR(sysdate,'yyyy-MM-dd'),'yyyy-MM-dd')))))

THEN 'IN PROGRESS'

END)STATUS,

/\*(SELECT SUBSTR(NAME,1,INSTR(NAME,'BU',-1,1)-2)

FROM hrw\_business\_units

WHERE business\_unit\_id=asg.business\_unit\_id

)company\*/

-- (SELECT meaning

-- FROM hr\_lookups

-- WHERE lookup\_type = 'EMP\_CAT'

-- AND LOOKUP\_CODE = asg.employment\_category

-- ) entity,

a2.ABSENCE\_STATUS\_CD,

NVL(

(SELECT max(DEI\_ATTRIBUTE\_DATE1)

FROM hr\_documents\_of\_record hdr,

HR\_DOCUMENT\_TYPES\_TL hdt

WHERE hdr.document\_type\_id =hdt.document\_type\_id

AND hdr.person\_id =papf.person\_id

AND HDT.LANGUAGE ='US'

AND HDT.DOCUMENT\_TYPE ='Duty Resumption'

AND TO\_CHAR(hdr.DEI\_ATTRIBUTE1) =TO\_CHAR(A2.PER\_ABSENCE\_ENTRY\_ID)

), NVL(

(SELECT MIN(HDR.DEI\_ATTRIBUTE\_DATE3)

FROM hr\_documents\_of\_record hdr,

HR\_DOCUMENT\_TYPES\_TL hdt

WHERE hdr.document\_type\_id =hdt.document\_type\_id

AND hdr.person\_id =papf.person\_id

AND HDT.LANGUAGE ='US'

AND HDT.DOCUMENT\_TYPE ='Duty Resumption - Admin'

AND TRUNC(hdr.DEI\_ATTRIBUTE\_DATE1) =TRUNC(A2.START\_DATE)

AND TRUNC(hdr.DEI\_ATTRIBUTE\_DATE2) =TRUNC(A2.END\_DATE)

),(

CASE

WHEN (SELECT TRUNC(MAX(ANC.START\_DATE))

FROM ANC\_PER\_ABS\_ENTRIES anc

WHERE anc.person\_id = PAPF.person\_id

AND upper(anc.APPROVAL\_STATUS\_CD) = DECODE(upper(anc.ABSENCE\_STATUS\_CD),upper('SUBMITTED'),upper('APPROVED'),upper('ORA\_WITHDRAWN'),upper('ORA\_AWAIT\_AWAIT'))

AND TRUNC(ANC.START\_DATE) <=

(SELECT MIN(DEI\_ATTRIBUTE\_DATE3)

FROM hr\_documents\_of\_record hdr,

HR\_DOCUMENT\_TYPES\_TL hdt

WHERE hdr.document\_type\_id =hdt.document\_type\_id

AND hdr.person\_id =papf.person\_id

AND HDT.LANGUAGE ='US'

AND HDT.DOCUMENT\_TYPE ='Duty Resumption - Admin'

AND TRUNC(hdr.DEI\_ATTRIBUTE\_DATE3) >=TRUNC(A2.END\_DATE)

AND NOT EXISTS

(SELECT 1

FROM ANC\_PER\_ABS\_ENTRIES ABS1

WHERE ABS1.PERSON\_ID =PAPF.PERSON\_ID

AND TRUNC(ABS1.START\_DATE)=TRUNC(hdr.DEI\_ATTRIBUTE\_DATE1)

AND TRUNC(ABS1.END\_DATE) =TRUNC(hdr.DEI\_ATTRIBUTE\_DATE2)

)

) ) =TRUNC(A2.START\_DATE)

THEN

(SELECT MIN(DEI\_ATTRIBUTE\_DATE3)

FROM hr\_documents\_of\_record hdr,

HR\_DOCUMENT\_TYPES\_TL hdt

WHERE hdr.document\_type\_id =hdt.document\_type\_id

AND hdr.person\_id =papf.person\_id

AND HDT.LANGUAGE ='US'

AND HDT.DOCUMENT\_TYPE ='Duty Resumption - Admin'

AND TRUNC(hdr.DEI\_ATTRIBUTE\_DATE3) >=TRUNC(A2.END\_DATE)

AND NOT EXISTS

(SELECT 1

FROM ANC\_PER\_ABS\_ENTRIES ABS1

WHERE ABS1.PERSON\_ID =PAPF.PERSON\_ID

AND TRUNC(ABS1.START\_DATE)=TRUNC(hdr.DEI\_ATTRIBUTE\_DATE1)

AND TRUNC(ABS1.END\_DATE) =TRUNC(hdr.DEI\_ATTRIBUTE\_DATE2)

)

)

ELSE NULL

END ))) duty\_resumption

--a4.name abs\_reason

FROM PER\_PERSON\_NAMES\_F PNAME,

PER\_ALL\_PEOPLE\_F PAPF,

PER\_ALL\_ASSIGNMENTS\_M ASG,

PER\_PERIODS\_OF\_SERVICE PPOS,

--PAY\_PAYROLL\_ASSIGNMENTS d2,

--PAY\_ASSIGNED\_PAYROLLS\_DN d1,

HR\_ALL\_ORGANIZATION\_UNITS ORG,

ANC\_ABSENCE\_TYPES\_F\_TL A1,

ANC\_PER\_ABS\_ENTRIES A2,

ANC\_ABSENCE\_TYPE\_REASONS\_F a3,

ANC\_ABSENCE\_REASONS\_F\_TL a4,

HR\_ALL\_POSITIONS\_F\_TL POS

WHERE PNAME.PERSON\_ID = PAPF.PERSON\_ID

AND ASG.PERSON\_ID = PAPF.PERSON\_ID

AND PPOS.PERSON\_ID = PAPF.PERSON\_ID

--AND d2.hr\_assignment\_id = asg.assignment\_id

--AND d1.payroll\_term\_id = d2.payroll\_term\_id

AND ORG.ORGANIZATION\_ID(+) = ASG.ORGANIZATION\_ID

AND A1.ABSENCE\_TYPE\_ID = A2.ABSENCE\_TYPE\_ID

AND A2.PERSON\_ID = PAPF.PERSON\_ID

AND A2.ABSENCE\_TYPE\_REASON\_ID = a3.ABSENCE\_TYPE\_REASON\_ID(+)

AND a3.ABSENCE\_REASON\_ID =a4.ABSENCE\_REASON\_ID(+)

AND ASG.position\_id =POS.position\_id(+)

AND NVL(A1.LANGUAGE,'US') ='US'

AND NVL(A4.LANGUAGE,'US') ='US'

AND NVL(POS.language,'US') ='US'

AND upper(A2.APPROVAL\_STATUS\_CD) = DECODE(upper(a2.ABSENCE\_STATUS\_CD),upper('SUBMITTED'),upper('APPROVED'),upper('ORA\_WITHDRAWN'),upper('ORA\_AWAIT\_AWAIT'))

--AND upper(a2.ABSENCE\_STATUS\_CD) <> upper('ORA\_WITHDRAWN')

AND TRUNC(A2.start\_date) BETWEEN papf.EFFECTIVE\_START\_DATE AND papf.EFFECTIVE\_END\_DATE

AND TRUNC(A2.start\_date) BETWEEN PNAME.EFFECTIVE\_START\_DATE AND PNAME.EFFECTIVE\_END\_DATE

AND TRUNC(A2.start\_date) BETWEEN ASG.EFFECTIVE\_START\_DATE AND ASG.EFFECTIVE\_END\_DATE

AND TRUNC(A2.start\_date) BETWEEN NVL(POS.effective\_start\_date,A2.START\_DATE) AND NVL(POS.effective\_end\_date,A2.START\_DATE)

--AND TRUNC(A2.start\_date) BETWEEN NVL(d1.start\_date,TRUNC(A2.start\_date,'Month')) AND NVL(d1.end\_date,last\_day(A2.start\_date))

--AND TRUNC(A2.start\_date) BETWEEN d2.start\_date AND d2.end\_date

/\* AND (((NVL(TRUNC(to\_date(TO\_CHAR(:p\_date\_from,'yyyy-MM-dd'),'yyyy-MM-dd')),TRUNC(to\_date(TO\_CHAR(sysdate,'yyyy-MM-dd'),'yyyy-MM-dd')))) BETWEEN A2.start\_date AND A2.end\_date)

OR ((NVL(TRUNC(to\_date(TO\_CHAR(:p\_date\_to,'yyyy-MM-dd'),'yyyy-MM-dd')),TRUNC(to\_date(TO\_CHAR(sysdate,'yyyy-MM-dd'),'yyyy-MM-dd')))) BETWEEN A2.start\_date AND A2.END\_DATE)

OR (((NVL(TRUNC(to\_date(TO\_CHAR(:p\_date\_from,'yyyy-MM-dd'),'yyyy-MM-dd')),TRUNC(to\_date(TO\_CHAR(sysdate,'yyyy-MM-dd'),'yyyy-MM-dd')))) < A2.start\_date)

AND (((NVL(TRUNC(to\_date(TO\_CHAR(:p\_date\_to,'yyyy-MM-dd'),'yyyy-MM-dd')),TRUNC(to\_date(TO\_CHAR(sysdate,'yyyy-MM-dd'),'yyyy-MM-dd')))) > A2.END\_DATE))))

\*/

AND papf.PERSON\_NUMBER = NVL(:P\_EMPNO,papf.PERSON\_NUMBER )

AND NVL(org.organization\_id,00000) =NVL(:p\_dept,NVL(org.organization\_id,00000))

AND NVL(:p\_date\_from,SYSDATE) <=NVL(:p\_date\_to,SYSDATE)

AND asg.business\_unit\_id = NVL(:p\_emp\_cat,asg.business\_unit\_id)

AND A1.NAME in ('Military Service Leave','Maternity Leave')

AND trunc(A2.CREATION\_DATE) =TRUNC(SYSDATE-1) /\*between trunc(sysdate,'YYYY') and trunc(sysdate) \*/

AND PNAME.NAME\_TYPE = 'GLOBAL'

AND ASG.ASSIGNMENT\_TYPE = 'E'

AND ASG.PRIMARY\_FLAG = 'Y'

AND asg.EFFECTIVE\_LATEST\_CHANGE = 'Y'

AND asg.assignment\_status\_type ='ACTIVE'

AND ppos.date\_start =

(SELECT MAX(date\_start)

FROM PER\_PERIODS\_OF\_SERVICE ppos1

WHERE ppos1.person\_id = ppos.person\_id

/\*AND ppos1.date\_start <= NVL(last\_day(to\_date(TO\_CHAR(:p\_date\_to,'yyyy-MM-dd'),'yyyy-MM-dd')),sysdate)\*/

)

ORDER BY to\_number(employee\_number),

a2.start\_date

)XX

/\*WHERE ((NVL(:P\_STATUS,'ALL')='ALL'

AND XX.STATUS IN('COMPLETED','IN PROGRESS','PLANNED'))

OR (:P\_STATUS ='AVAILED'

AND XX.STATUS IN('COMPLETED'))

OR (:P\_STATUS ='APPLIED'

AND XX.STATUS IN('PLANNED','IN PROGRESS')))

ORDER BY EMPLOYEE\_NUMBER \*/

Layout



PROBATION\_ALERT

This alert sends notification for all new joiners who joined a day before

Data Model

SELECT

PERSON\_NUMBER ,

person\_id,

LEGALEMPLOYER ,

BUSINESSUNIT ,

EMP\_NAME ,

EMAIL\_ADDRESS ,

ORIGINAL\_DATE\_OF\_HIRE,

DIFF,

SUPERVISOR\_EMAIL\_ADDRESS

FROM

(

SELECT

PERSON\_NUMBER ,

person\_id,

LEGALEMPLOYER ,

BUSINESSUNIT ,

EMP\_NAME ,

EMAIL\_ADDRESS ,

ORIGINAL\_DATE\_OF\_HIRE,

TRUNC(SYSDATE) - TRUNC(ORIGINAL\_DATE\_OF\_HIRE) DIFF,

SUPERVISOR\_EMAIL\_ADDRESS

FROM

(

SELECT papf.person\_number,

papf.person\_id,

hauft.NAME LegalEmployer,

(select hauft.NAME

from HR\_ORGANIZATION\_UNITS\_F\_TL hauft

where hauft.organization\_id = paam.business\_unit\_id

AND SYSDATE BETWEEN hauft.effective\_start\_date AND hauft.effective\_end\_date ) BusinessUnit,

(select emp\_name.full\_name from per\_person\_names\_f emp\_name

WHERE PAPF.person\_id = emp\_name.person\_id

AND emp\_name.name\_type = 'GLOBAL'

AND trunc(SYSDATE) BETWEEN emp\_name.effective\_start\_date AND emp\_name.effective\_end\_date) EMP\_NAME,

(select min (ppos.DATE\_START) from per\_periods\_of\_service PPOS WHERE ppos.PERIOD\_TYPE ='E' and PPOS.PERSON\_ID = PAPF.PERSON\_ID AND PPOS.PERIOD\_OF\_SERVICE\_ID = PAAM.PERIOD\_OF\_SERVICE\_ID) original\_date\_of\_hire ,

( SELECT

listagg(pea.email\_address ,',')

FROM

per\_email\_addresses pea

WHERE

papf.person\_id = pea.person\_id AND pea.email\_type='W1' AND TRUNC (SYSDATE) BETWEEN PEA.DATE\_FROM AND NVL(PEA.DATE\_TO,TO\_DATE('31-12-4712','DD-MM-RRRR')) ) EMAIL\_ADDRESS,

( SELECT

listagg (pea.email\_address ,',')

FROM

per\_email\_addresses pea ,

per\_assignment\_supervisors\_f pasf

WHERE

pea.person\_id = pasf.manager\_id

AND papf.person\_id = pasf.person\_id

AND pasf.assignment\_id = paam.assignment\_id

AND pasf.manager\_type = 'LINE\_MANAGER'

AND pea.email\_type='W1'

AND trunc(SYSDATE) BETWEEN pasf.effective\_start\_date AND pasf.effective\_end\_date

AND TRUNC (SYSDATE) BETWEEN PEA.DATE\_FROM AND NVL(PEA.DATE\_TO,TO\_DATE('31-12-4712','DD-MM-RRRR')) ) SUPERVISOR\_EMAIL\_ADDRESS

FROM HR\_ORG\_UNIT\_CLASSIFICATIONS\_F houcf,

HR\_ALL\_ORGANIZATION\_UNITS\_F haouf,

HR\_ORGANIZATION\_UNITS\_F\_TL hauft,

per\_all\_assignments\_m paam,

per\_all\_people\_f papf, (select min (ppos.original\_date\_of\_hire) original\_date\_of\_hire, ppos.PERSON\_ID from per\_periods\_of\_service PPOS where ppos.PERIOD\_TYPE ='E' group by ppos.PERSON\_ID) ppos

WHERE haouf.ORGANIZATION\_ID = houcf.ORGANIZATION\_ID

AND haouf.ORGANIZATION\_ID = hauft.ORGANIZATION\_ID

AND haouf.EFFECTIVE\_START\_DATE BETWEEN houcf.EFFECTIVE\_START\_DATE AND houcf.EFFECTIVE\_END\_DATE

AND hauft.LANGUAGE = 'US'

AND hauft.EFFECTIVE\_START\_DATE = haouf.EFFECTIVE\_START\_DATE

AND hauft.EFFECTIVE\_END\_DATE = haouf.EFFECTIVE\_END\_DATE

AND houcf.CLASSIFICATION\_CODE = 'HCM\_LEMP'

AND SYSDATE BETWEEN hauft.effective\_start\_date AND hauft.effective\_end\_date

AND hauft.organization\_id = paam.legal\_entity\_id

and paam.person\_id = papf.person\_id

and ppos.PERSON\_ID = PAPF.PERSON\_ID

-- AND paam.period\_of\_service\_id = PPOS.PERIOD\_OF\_SERVICE\_ID

and paam.primary\_assignment\_flag = 'Y'

and paam.assignment\_type = 'E'

and paam.effective\_latest\_change = 'Y'

and sysdate between paam.effective\_start\_date and paam.effective\_end\_date

and sysdate between papf.effective\_start\_date and papf.effective\_end\_date

-- and papf.person\_number = nvl(:personnumber,papf.person\_number)

-- and papf.person\_number = nvl(:personnumber,papf.person\_number)

-- and papf.effective\_start\_date between to\_date('01-01-2009','DD-MM-RRRR') and sysdate

AND TO\_CHAR(paam.business\_unit\_id) IN (

SELECT

TO\_CHAR(substr(a.ROLE\_NAME,7))

FROM

per\_roles\_dn\_vl a,

per\_user\_roles b,

per\_users c

WHERE

a.role\_id = b.role\_id

AND b.user\_id = c.user\_id

AND c.username = fnd\_global.user\_name

AND b.active\_flag = 'Y'

AND a.ROLE\_NAME IN ('CORPT-300000003250079',

'ADRES-300000003250653',

'GATWY-300000003250571',

'TEYSR-300000003250489',

'INVST-300000003250325',

'PRJCT-300000003250243',

'DEVLP-300000003250161',

'ADBUH-300000003250407',

'DUHLG-300000003250735')

)

order by papf.person\_number asc,hauft.name asc nulls first

)

)

WHERE DIFF =:DIFF

Layout



Probation process documents

Requirement: Automate updating Probation Confirmed DFF flag to “Yes” for staff whose appraisal is “Completed Successfully”

Develop Report to query for all staff who have been successfully Probation Confirmed as of day before

And develop HDL Fast Formula scripts to update Probation Confirmed Flag to “Yes”

Following are the objects created

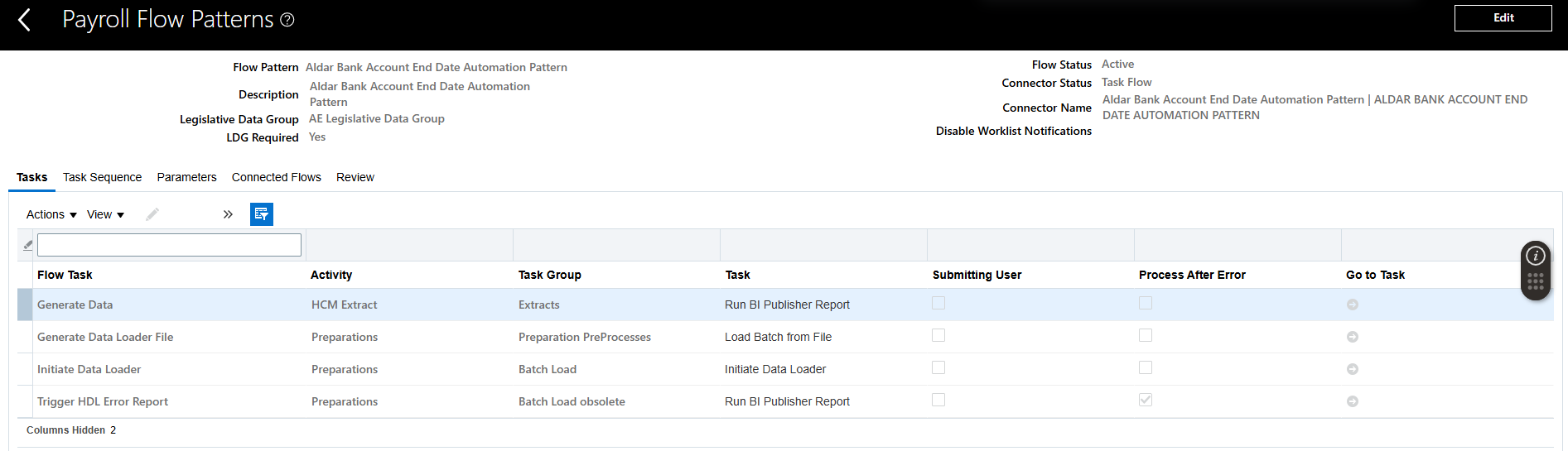
Bank Account Automation Process

Requirement: Automate updating Bank Account and bank for staff who have completed approval using document of records

Develop Report to query for all staff who have been successfully Completed Bank Account approval as of day before

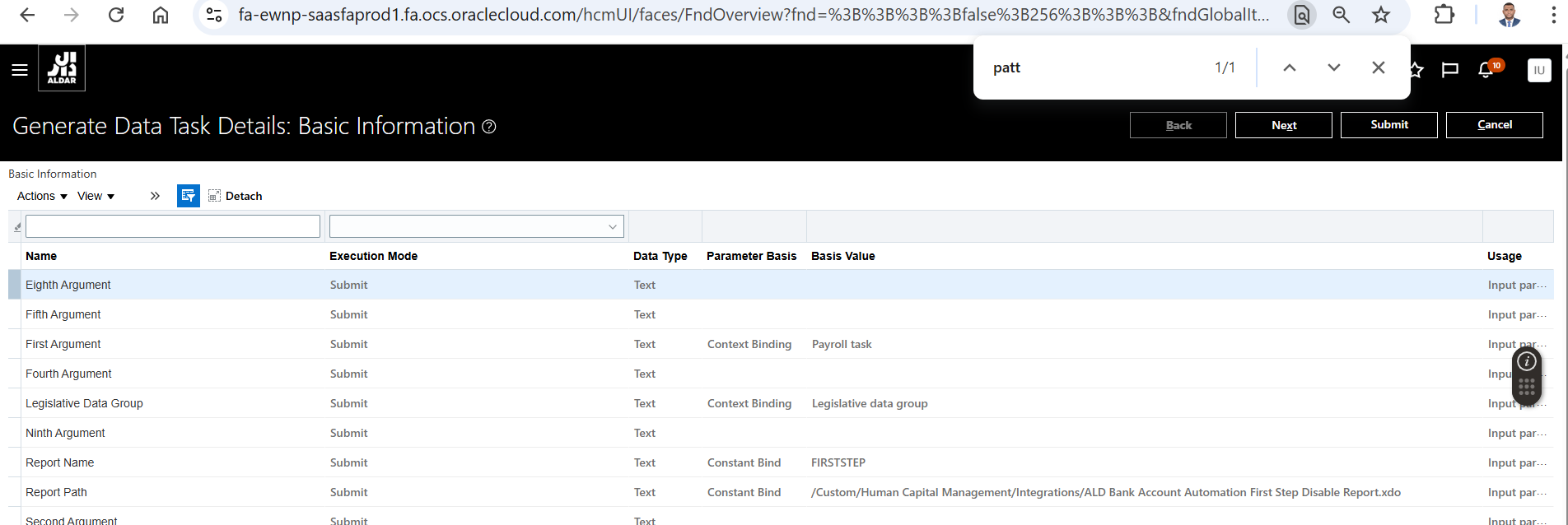
And develop HDL Fast Formula scripts to end date existing bank account and create new bank account and make it primary.

Following are the objects created

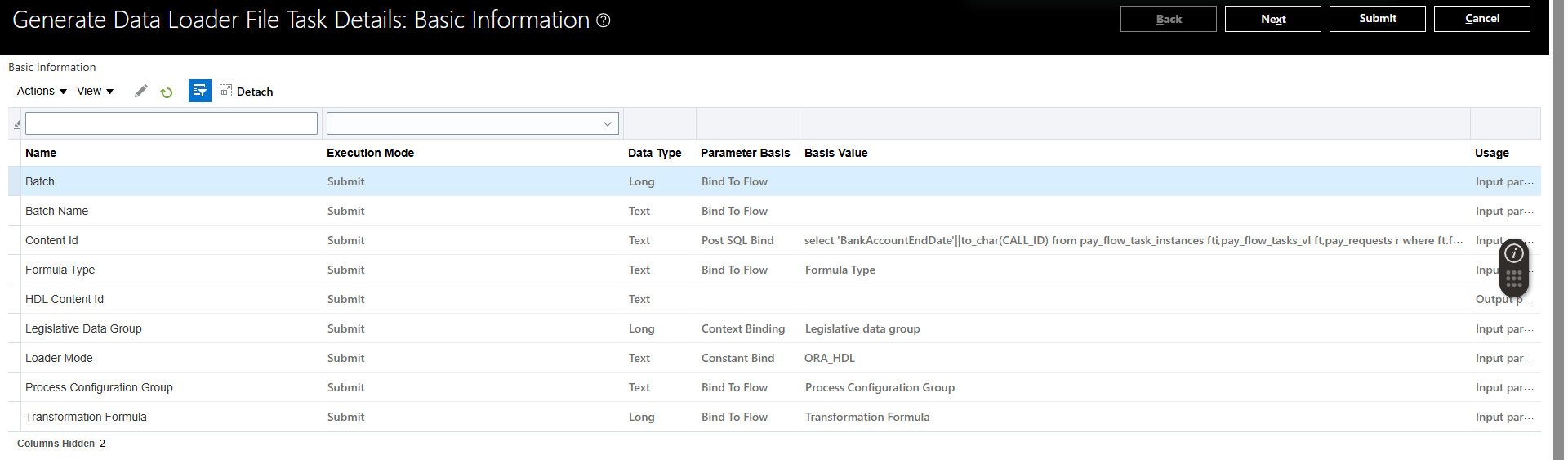


Step 1. Generate Data (from report)

path



Step 1. Generate Data Loader File parameters



Initiate Data Loader

A screenshot of a computer

AI-generated content may be incorrect.

select 'BankAccountEndDate'||to\_char(CALL\_ID) from pay\_flow\_task\_instances fti,pay\_flow\_tasks\_vl ft,pay\_requests r where ft.flow\_task\_name = 'Generate Data' and ft.base\_flow\_task\_id = fti.base\_flow\_task\_id and fti.flow\_task\_instance\_id = r.flow\_task\_instance\_id and r.call\_type= 'ESS' and fti.flow\_instance\_id = :pFlowInstanceId

Aldar Bank Account Create Automation Pattern

A screenshot of a computer

AI-generated content may be incorrect.

Step 1. Generate Data

/Custom/Human Capital Management/Integrations/ALD\_Bank\_Account\_Automation\_Second\_Step\_Create\_Payment\_Report.xdo

A screenshot of a computer

AI-generated content may be incorrect.

Step 2. Generate Data Loader File

select 'BankAccountCreateDate'||to\_char(CALL\_ID) from pay\_flow\_task\_instances fti,pay\_flow\_tasks\_vl ft,pay\_requests r where ft.flow\_task\_name = 'Generate Data' and ft.base\_flow\_task\_id = fti.base\_flow\_task\_id and fti.flow\_task\_instance\_id = r.flow\_task\_instance\_id and r.call\_type= 'ESS' and fti.flow\_instance\_id = :pFlowInstanceId

A screenshot of a computer

AI-generated content may be incorrect.

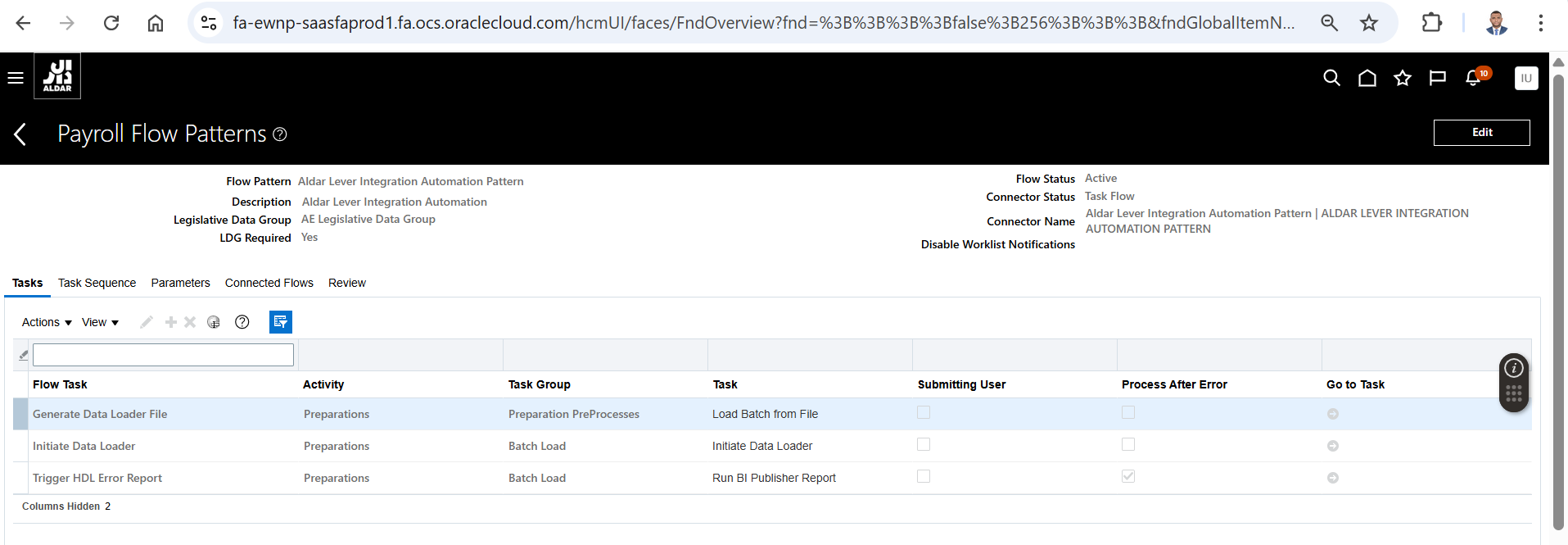
Fast Formula for HDL

For

End Date Bank Account HDL

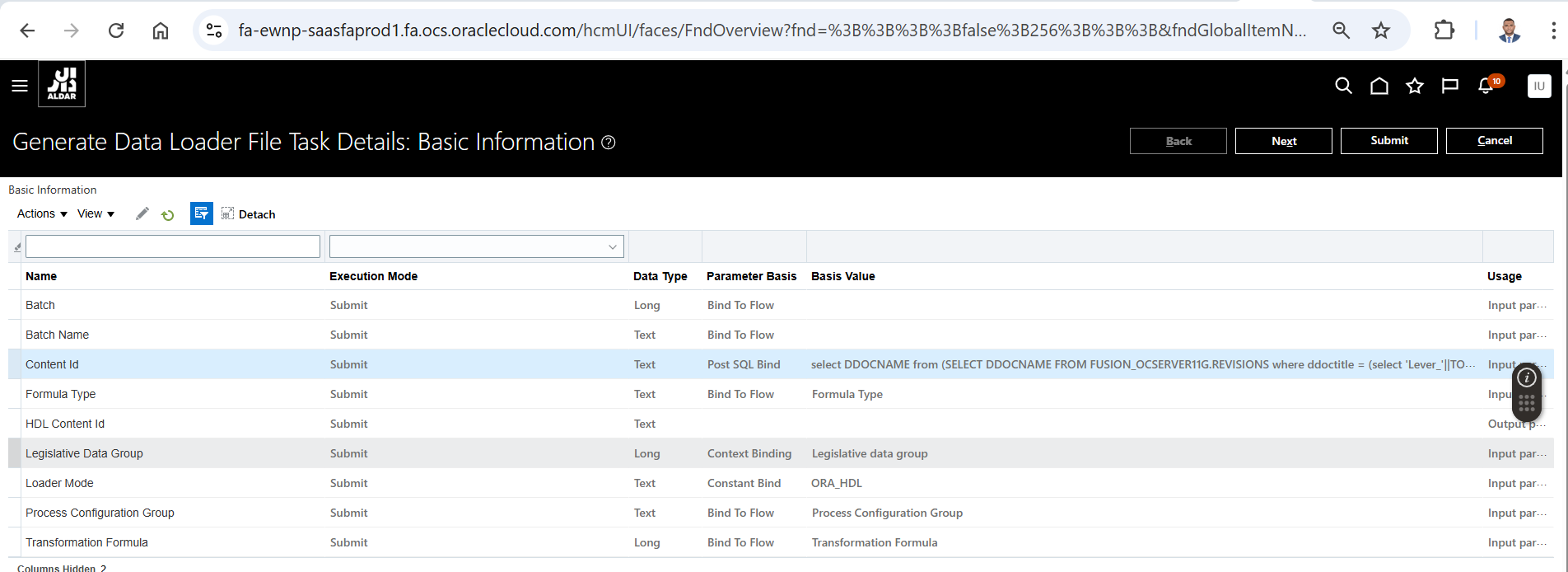
Create Bank Account HDL

Lever Employee Import Automation Process



Step 1. Generate Data Loader File

select DDOCNAME from (SELECT DDOCNAME FROM FUSION\_OCSERVER11G.REVISIONS where ddoctitle = (select 'Lever\_'||TO\_CHAR(sysdate-1,'DD\_MM\_YYYY', 'NLS\_DATE\_LANGUAGE = American') ||'.txt' from dual) order by DCREATEDATE desc) where rownum=1



Step 2. Initiate Data Loader

A screenshot of a computer

AI-generated content may be incorrect.

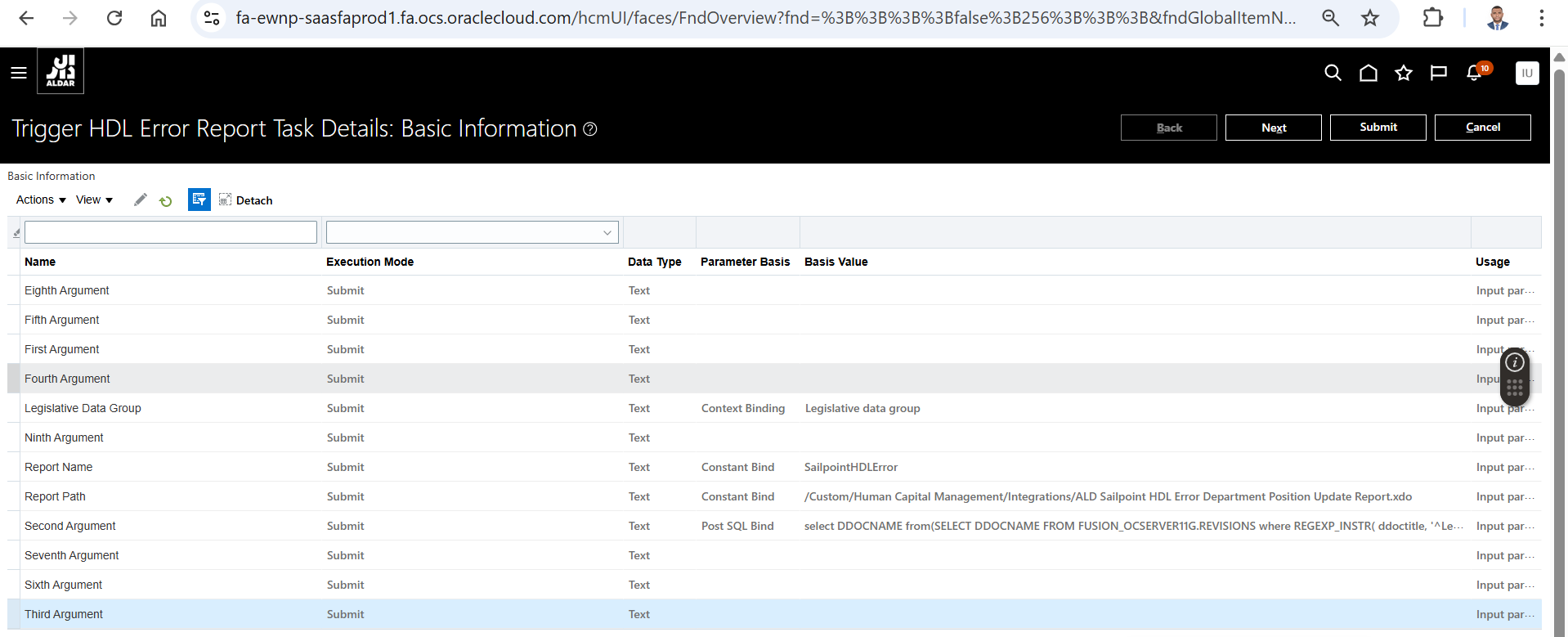
Step 3. Trigger HDL Error Report

Report Path

/Custom/Human Capital Management/Integrations/ALD Sailpoint HDL Error Department Position Update Report.xdo

Second Argument

select DDOCNAME from(SELECT DDOCNAME FROM FUSION\_OCSERVER11G.REVISIONS where REGEXP\_INSTR( ddoctitle, '^Lever\_Scheduled\_Job\_[0-9]{1,2}\_(AM|PM)\_Everyday\_[0-9]{2}\_[A-Z][a-z]{2}\_[0-9]{4}\_go\_live\_V[0-9]+ : [0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}:[0-9]{2}:[0-9]{2}\.[0-9]+\.zip$')>0 order by DCREATEDATE desc) where rownum=1



Fast Formula for HDL

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*

\* FORMULA NAME: ALDAR\_LEVER\_EMPLOYEE\_INTEGRATION\_GENERATE\_HDL\_FROM\_TXT

\* FORMULA TYPE: HCM Data Loader

\* DESCRIPTION: This formula will create HDL from TXT File

\* CHANGE HISTORY:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Version Date Created By Comments

-----------------------------------------------------------------------

1.0 14-May-2024 Abdulrahman Mohammed Initial Version

2.0 06-May-2025 Abdulrahman Mohammed Updated to create Pending worker with Position Code

2.1 12-May-2025 Abdulrahman Mohammed Updated to create Pending worker with Contract happens

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

INPUTS ARE OPERATION (text), LINENO (number), LINEREPEATNO (number),POSITION1 (text), POSITION2 (text), POSITION3 (text), POSITION4 (text), POSITION5 (text), POSITION6 (text),POSITION7 (text), POSITION8 (text), POSITION9 (text), POSITION10 (text), POSITION11 (text), POSITION12 (text),POSITION13 (text), POSITION14 (text), POSITION15 (text), POSITION16 (text), POSITION17 (text), POSITION18 (text), POSITION19 (text), POSITION20 (text)

DEFAULT FOR LINENO IS 1

DEFAULT FOR LINEREPEATNO IS 1

DEFAULT FOR LINEREPEAT IS 'Y'

/\* Company Name|First Name|Last Name|Team|Department|Grade|Job Title|Job Level|Requisition Name|Salary|Candidate Hire Date|Date of Birth|Gender|Nationality|Nationality Type|Martial Status|Contract Type \*/

DEFAULT FOR POSITION1 IS 'NO DATA' /\*Existing Field with Data ALDAR \*/ /\*New field with Data ALDAR Company Name \*/

DEFAULT FOR POSITION11 IS 'NO DATA' /\*EffectiveStartDate Candidate Hire Date ---Earlier was Address Line 1\*/

DEFAULT FOR POSITION12 IS 'NO DATA' /\*Date of Birth -- Earlier was AddressLine2 \*/

DEFAULT FOR POSITION2 IS 'NO DATA' /\*First Name --Existing Field with Data EffectiveStartDate \*/

DEFAULT FOR POSITION3 IS 'NO DATA' /\*Last Name ---Existing Field with Data DateOfBirth \*/

DEFAULT FOR POSITION4 IS 'NO DATA' /\*Data Team ---Existing Field with Data FirstName \*/

DEFAULT FOR POSITION5 IS 'NO DATA' /\*Business UnitDepartment --Existing Field with Data LastName \*/

DEFAULT FOR POSITION6 IS 'NO DATA' /\*Grade --Existing Field with Data LastName \*/

DEFAULT FOR POSITION7 IS 'NO DATA' /\*Job Title \*/

DEFAULT FOR POSITION8 IS 'NO DATA' /\*Job Level\*/

DEFAULT FOR POSITION9 IS 'NO DATA' /\* Position Name --Existing Field with Data PhoneNumber \*/

DEFAULT FOR POSITION10 IS 'NO DATA' /\*Salary --Existing Field with Data AddressType HM \*/

DEFAULT FOR POSITION13 IS 'NO DATA' /\*Gender --Existing New field with Data\*/

DEFAULT FOR POSITION14 IS 'NO DATA' /\*Nationality ---Existing Field with Data Region2\*/

DEFAULT FOR POSITION15 IS 'NO DATA' /\*Nationality Type \*/

DEFAULT FOR POSITION16 IS 'NO DATA' /\*Martial Status \*/

DEFAULT FOR POSITION17 IS 'NO DATA' /\*Contract Type ---Existing Field with Data Citizenship Country \*/

DEFAULT FOR POSITION18 IS 'NO DATA' /\*Position Code \*/

DEFAULT FOR POSITION19 IS 'NO DATA' /\*Position Code \*/

DEFAULT FOR POSITION20 IS 'NO DATA' /\*Position Code \*/

utc\_date = GET\_SYSDATE()

cst\_date = NEW\_TIME(utc\_date, 'GMT', 'CST')

IF OPERATION='FILETYPE' THEN

OUTPUTVALUE='DELIMITED'

ELSE IF OPERATION='DELIMITER' THEN

OUTPUTVALUE='|'

ELSE IF OPERATION='READ' THEN

OUTPUTVALUE='NONE'

ELSE IF OPERATION = 'NUMBEROFBUSINESSOBJECTS' THEN

(

OUTPUTVALUE = '13'

RETURN OUTPUTVALUE

)

ELSE IF OPERATION = 'METADATALINEINFORMATION' THEN

(

METADATA1[1] = 'Worker' /\*FileName\*/ /\*Reserved\*/

METADATA1[2] = 'Worker' /\*FileDiscriminator\*/ /\*Reserved\*/

METADATA1[3] = 'SourceSystemOwner'

METADATA1[4] = 'SourceSystemId'

METADATA1[5] = 'EffectiveStartDate'

METADATA1[6] = 'EffectiveEndDate'

METADATA1[7] = 'StartDate'

METADATA1[8] = 'DateOfBirth'

METADATA1[9] = 'ActionCode'

METADATA2[1] = 'Worker' /\*FileName\*/ /\*Reserved\*/

METADATA2[2] = 'PersonAddress' /\*FileDiscriminator\*/ /\*Reserved\*/

METADATA2[3] = 'SourceSystemOwner'

METADATA2[4] = 'SourceSystemId'

METADATA2[5] = 'PersonId(SourceSystemId)'

METADATA2[6] = 'EffectiveStartDate'

METADATA2[7] = 'EffectiveEndDate'

METADATA2[8] = 'AddressType' /\* HOME\*/

METADATA2[9] = 'AddressLine1' /\* Villa 29 \*/

METADATA2[10] = 'AddressLine2' /\* Al Hiwaylah Street Sector 7 \*/

METADATA2[11] = 'Region1' /\* Baniyas East \*/

METADATA2[12] = 'Region2' /\* Abu Dhabi \*/

METADATA2[13] = 'Region3' /\* Abu Dhabi \*/

METADATA2[14] = 'Country' /\* AE \*/

METADATA2[15] = 'PostalCode' /\* 123314 \*/

METADATA3[1] = 'Worker' /\*FileName\*/ /\*Reserved\*/

METADATA3[2] = 'PersonCitizenship' /\*FileDiscriminator\*/

METADATA3[3] = 'SourceSystemOwner'

METADATA3[4] = 'SourceSystemId'

METADATA3[5] = 'PersonId(SourceSystemId)'

METADATA3[6] = 'DateFrom'

METADATA3[7] = 'DateTo'

METADATA3[8] = 'LegislationCode'

METADATA3[9] = 'CitizenshipStatus'

METADATA4[1] = 'Worker' /\*FileName\*/ /\*Reserved\*/

METADATA4[2] = 'PersonLegislativeData' /\*FileDiscriminator\*/

METADATA4[3] = 'SourceSystemOwner'

METADATA4[4] = 'SourceSystemId'

METADATA4[5] = 'PersonId(SourceSystemId)'

METADATA4[6] = 'EffectiveStartDate'

METADATA4[7] = 'EffectiveEndDate'

METADATA4[8] = 'LegislationCode'

METADATA4[9] = 'MaritalStatus'

METADATA4[10] = 'Sex'

METADATA5[1] = 'Worker' /\*FileName\*/ /\*Reserved\*/

METADATA5[2] = 'PersonName' /\*FileDiscriminator\*/

METADATA5[3] = 'SourceSystemOwner'

METADATA5[4] = 'SourceSystemId'

METADATA5[5] = 'PersonId(SourceSystemId)'

METADATA5[6] = 'EffectiveStartDate'

METADATA5[7] = 'EffectiveEndDate'

METADATA5[8] = 'LegislationCode'

METADATA5[9] = 'NameType'

METADATA5[10] = 'FirstName'

METADATA5[11] = 'LastName'

METADATA5[12] = 'Title'

METADATA6[1] = 'Worker' /\*FileName\*/ /\*Reserved\*/

METADATA6[2] = 'PersonPhone' /\*FileDiscriminator\*/

METADATA6[3] = 'SourceSystemOwner'

METADATA6[4] = 'SourceSystemId'

METADATA6[5] = 'PersonId(SourceSystemId)'

METADATA6[6] = 'DateFrom'

METADATA6[7] = 'DateTo'

METADATA6[8] = 'PhoneType'

METADATA6[9] = 'LegislationCode'

METADATA6[10] = 'PhoneNumber'

METADATA7[1] = 'Worker' /\*FileName\*/ /\*Reserved\*/

METADATA7[2] = 'PersonReligion' /\*FileDiscriminator\*/

METADATA7[3] = 'SourceSystemOwner'

METADATA7[4] = 'SourceSystemId'

METADATA7[5] = 'PersonId(SourceSystemId)'

METADATA7[6] = 'LegislationCode'

METADATA7[7] = 'Religion'

METADATA7[8] = 'PrimaryFlag'

METADATA8[1] = 'Worker' /\*FileName\*/ /\*Reserved\*/

METADATA8[2] = 'WorkRelationship' /\*FileDiscriminator\*/

METADATA8[3] = 'SourceSystemOwner'

METADATA8[4] = 'SourceSystemId'

METADATA8[5] = 'PersonId(SourceSystemId)'

METADATA8[6] = 'LegalEmployerName'

METADATA8[7] = 'DateStart'

/\*METADATA8[8] = 'PrimaryFlag' \*/

METADATA8[8] = 'WorkerType'

METADATA8[9] = 'OnMilitaryServiceFlag'

METADATA8[10] = 'RehireRecommendationFlag'

METADATA8[11] = 'ActionCode'

METADATA8[12] = 'ReadyToConvert'

METADATA9[1] = 'Worker' /\*FileName\*/ /\*Reserved\*/

METADATA9[2] = 'WorkTerms' /\*FileDiscriminator\*/

METADATA9[3] = 'SourceSystemOwner'

METADATA9[4] = 'SourceSystemId'

METADATA9[5] = 'PersonId(SourceSystemId)'

METADATA9[6] = 'EffectiveStartDate'

METADATA9[7] = 'WorkerType'

/\* METADATA9[7] = 'EffectiveEndDate' \*/

METADATA9[8] = 'EffectiveSequence'

METADATA9[9] = 'EffectiveLatestChange'

METADATA9[10] = 'DateStart'

/\* METADATA9[10] = 'AssignmentName' \*/

METADATA9[11] = 'AssignmentNumber'

METADATA9[12] = 'AssignmentStatusTypeCode'

METADATA9[13] = 'PersonTypeCode'

/\* METADATA9[13] = 'AssignmentType' \*/

METADATA9[14] = 'BusinessUnitShortCode'

METADATA9[15] = 'PositionOverrideFlag'

METADATA9[16] = 'PrimaryWorkTermsFlag'

METADATA9[17] = 'PeriodOfServiceId(SourceSystemId)'

METADATA9[18] = 'LegalEmployerName'

/\* METADATA9[19] = 'SystemPersonType' \*/

METADATA9[20] = 'ActionCode'

METADATA9[21] = 'ProposedUserPersonType'

METADATA10[1] = 'Worker' /\*FileName\*/ /\*Reserved\*/

METADATA10[2] = 'Contract' /\*FileDiscriminator\*/

METADATA10[3] = 'SourceSystemOwner'

METADATA10[4] = 'SourceSystemId'

METADATA10[5] = 'PersonId(SourceSystemId)'

METADATA10[6] = 'AssignmentId(SourceSystemId)'

METADATA10[7] = 'ContractType'

METADATA10[8] = 'EffectiveStartDate'

METADATA10[9] = 'AssignmentNumber'

METADATA10[10] = 'ContractNumber'

METADATA11[1] = 'Worker' /\*FileName\*/ /\*Reserved\*/

METADATA11[2] = 'Assignment' /\*FileDiscriminator\*/

METADATA11[3] = 'SourceSystemOwner'

METADATA11[4] = 'SourceSystemId'

METADATA11[5] = 'ActionCode'

METADATA11[6] = 'EffectiveStartDate'

METADATA11[7] = 'EffectiveSequence'

METADATA11[8] = 'EffectiveLatestChange'

METADATA11[9] = 'WorkTermsAssignmentId(SourceSystemId)'

METADATA11[10] = 'PersonTypeCode'

METADATA11[11] = 'BusinessUnitShortCode'

METADATA11[12] = 'PrimaryAssignmentFlag'

METADATA11[13] = 'JobCode'

METADATA11[14] = 'WorkTermsNumber'

METADATA11[15] = 'GradeCode'

/\* METADATA11[16] = 'PositionOverrideFlag'\*/

METADATA11[17] = 'AssignmentCategory'

METADATA11[18] = 'PersonId(SourceSystemId)'

METADATA11[19] = 'PeriodOfServiceId(SourceSystemId)'

METADATA11[20] = 'DateStart'

METADATA11[21] = 'DepartmentName'

METADATA11[22] = 'PositionCode'

METADATA11[23] = 'WorkerType'

METADATA11[24] = 'AssignmentStatusTypeCode'

METADATA11[25] = 'LegalEmployerName'

METADATA11[26] = 'ProposedUserPersonType'

METADATA11[27] = 'ProjectedStartDate'

METADATA11[28] = 'HourlySalariedCode'

METADATA11[29] = 'FullPartTime'

METADATA11[30] = 'PermanentTemporary'

METADATA12[1] = 'Salary' /\*FileName\*/ /\*Reserved\*/

METADATA12[2] = 'Salary' /\*FileDiscriminator\*/

METADATA12[3] = 'SourceSystemOwner'

METADATA12[4] = 'SourceSystemId'

METADATA12[5] = 'AssignmentId(SourceSystemId)'

METADATA12[6] = 'DateFrom'

METADATA12[7] = 'SalaryAmount'

METADATA12[8] = 'MultipleComponents'

METADATA12[9] = 'SalaryBasisName'

METADATA12[10] = 'ActionCode'

/\* METADATA13[1] = 'Position'

METADATA13[2] = 'Position'

METADATA13[3] = 'SourceSystemOwner'

METADATA13[4] = 'SourceSystemId'

METADATA13[5] = 'EffectiveStartDate'

METADATA13[6] = 'EffectiveEndDate'

METADATA13[7] = 'Name'

METADATA13[8] = 'PositionCode'

METADATA13[9] = 'BusinessUnitName'

METADATA13[10] = 'ActiveStatus'

METADATA13[11] = 'JobCode'

METADATA13[12] = 'JobSetCode'

METADATA13[13] = 'DepartmentName'

METADATA13[14] = 'HiringStatus'

METADATA13[15] = 'PositionType'

METADATA13[16] = 'FTE'

METADATA13[17] = 'HeadCount'

METADATA13[18] = 'FullPartTime'

METADATA13[19] = 'WorkingHours'

METADATA13[20] = 'Frequency'

\*/

METADATA13[1] = 'AssignedPayroll' /\*FileName\*/ /\*Reserved\*/

METADATA13[2] = 'AssignedPayroll' /\*FileDiscriminator\*/

METADATA13[3] = 'SourceSystemOwner'

METADATA13[4] = 'SourceSystemId'

METADATA13[5] = 'EffectiveStartDate'

METADATA13[6] = 'AssignmentId(SourceSystemId)'

METADATA13[7] = 'PayrollDefinitionCode'

METADATA13[8] = 'LegislativeDataGroupName'

METADATA13[9] = 'StartDate'

RETURN METADATA1, METADATA2 , METADATA3, METADATA4, METADATA5, METADATA6 ,METADATA7 ,METADATA8,METADATA9 ,METADATA10 ,METADATA11, METADATA12, METADATA13

)

ELSE IF (OPERATION='MAP' AND POSITION1='ALDAR') THEN

/\*HDL Related Outputs\*/

(

FileName = 'Worker'

BusinessOperation = 'MERGE'

SourceSystemOwner = 'LEVER'

EffectiveStartDate = trim(POSITION11)

EffectiveEndDate = '4712/12/31'

DateFrom= trim(POSITION11)

DateTo= '4712/12/31'

LC\_TIME =TO\_CHAR(cst\_date ,'YYYYMMDDHH24MI')

LC\_LEGAL\_EMPLOYER = 'Aldar Properties'

LC\_PAYROLL\_NAME = 'Aldar Development Payroll'

"PersonId(SourceSystemId)" = 'LEVER\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

RET = ESS\_LOG\_WRITE('Start of Map Operations time LC\_TIME :' ||LC\_TIME )

/\*Assigning Payroll and Legal Employer based on business unit \*/

IF trim(POSITION5) ='Avobar' THEN

(

LC\_LEGAL\_EMPLOYER = 'Avobar Restaurant - Sole Proprietorship L.L.C' /\*'AVOBAR RESTAURANT - SOLE PROPRIETORSHIP L.L.C.' \*/

LC\_PAYROLL\_NAME = 'C2i Payroll'

)

IF trim(POSITION5) ='Aldar Development' THEN

(

LC\_LEGAL\_EMPLOYER = 'Aldar Development - LLC - OPC' /\*'Aldar Properties' \*/

LC\_PAYROLL\_NAME = 'Aldar Development Payroll'

)

IF trim(POSITION5) ='Teyaseer' THEN

(

LC\_LEGAL\_EMPLOYER = 'Aldar Development - LLC - OPC' /\*'Teyaseer' \*/

LC\_PAYROLL\_NAME = 'Teyaseer Payroll'

)

IF trim(POSITION5) ='Aldar Dubai Holding' THEN

(

LC\_LEGAL\_EMPLOYER = 'Aldar Dubai Holding'

LC\_PAYROLL\_NAME = 'Aldar Development Payroll'

)

IF trim(POSITION5) ='Aldar Corporate' THEN

(

LC\_LEGAL\_EMPLOYER = 'Aldar Properties PJSC' /\*'Aldar Corporate' \*/

LC\_PAYROLL\_NAME = 'Aldar Corporate Payroll'

)

IF trim(POSITION5) ='C2i Corporate' THEN

(

LC\_LEGAL\_EMPLOYER = 'C2I Holding - Sole Proprietorship L.L.C.' /\*'Cloud Spaces Sole Proprietorship LLC - Branch'\*/

LC\_PAYROLL\_NAME = 'C2i Payroll'

)

IF trim(POSITION5) ='C2i Mancity' THEN

(

LC\_LEGAL\_EMPLOYER = 'C2I Holding - Sole Proprietorship L.L.C.'

LC\_PAYROLL\_NAME = 'C2i Payroll'

)

IF trim(POSITION5) ='Cloud Circle Mall' THEN

(

LC\_LEGAL\_EMPLOYER = 'Cloud Spaces Business Center L.L.C.'

LC\_PAYROLL\_NAME = 'C2i Payroll'

)

IF trim(POSITION5) ='Aldar Projects' THEN

(

LC\_LEGAL\_EMPLOYER = 'Aldar Projects LLC' /\*'Aldar Projects' \*/

LC\_PAYROLL\_NAME = 'Aldar Projects Payroll'

)

IF trim(POSITION5) ='Cloud Yas Mall' THEN

(

LC\_LEGAL\_EMPLOYER = 'Cloud Spaces - Sole Proprietorship L.L.C.'

LC\_PAYROLL\_NAME = 'C2i Payroll'

)

IF trim(POSITION5) ='Cloud ADGM' THEN

(

LC\_LEGAL\_EMPLOYER = 'Cloud Spaces - Sole Proprietorship L.L.C.'

LC\_PAYROLL\_NAME = 'C2i Payroll'

)

IF trim(POSITION5) ='Cloud WTC' THEN

(

LC\_LEGAL\_EMPLOYER = 'Cloud Spaces - Sole Proprietorship L.L.C.'

LC\_PAYROLL\_NAME = 'C2i Payroll'

)

IF trim(POSITION5) ='ADRES' THEN

(

LC\_LEGAL\_EMPLOYER = 'ADRES'

LC\_PAYROLL\_NAME = 'Advanced Real Estate Services – Sole Proprietorship LLC'

)

/\* IF trim(POSITION5) ='Huna' THEN

(

LC\_LEGAL\_EMPLOYER = 'C2i Holding - Sole Proprietorship LLC - Branch (Huna)'

LC\_PAYROLL\_NAME = 'C2i Payroll'

)\*/

IF trim(POSITION5) ='Aldar Estates' THEN

(

LC\_LEGAL\_EMPLOYER = 'Aldar Estates'

LC\_PAYROLL\_NAME = 'Aldar Development Payroll'

)

IF trim(POSITION5) ='Aldar Investment' /\*'Aldar Investment Property Management'\*/ THEN

(

LC\_LEGAL\_EMPLOYER = 'Aldar Investment Properties LLC - Standalone' /\*'Aldar Investment Property Management'\*/

LC\_PAYROLL\_NAME = 'AIPM Payroll'

)

/\* IF trim(POSITION5) ='Abu Dhabi Business Hub' THEN

(

LC\_LEGAL\_EMPLOYER = 'ABU DHABI BUSINESS HUB LLC - SOLE PROPRIETORSHIP L.L.C.'

LC\_PAYROLL\_NAME = 'ADBH Payroll'

)\*/

IF trim(POSITION5) ='Ether Yas Mall' THEN

(

LC\_LEGAL\_EMPLOYER = 'Ether By Cloud Spaces - Sole Proprietorship L.L.C'

LC\_PAYROLL\_NAME = 'C2i Payroll'

)

/\* IF trim(POSITION5) ='Teyaseer' THEN

(

LC\_LEGAL\_EMPLOYER = 'Aldar Development - LLC - OPC'

LC\_PAYROLL\_NAME = 'Gateway Payroll'

) \*/

IF trim(POSITION5) ='Cloud Corporate' THEN

(

LC\_LEGAL\_EMPLOYER = 'Cloud Spaces - Sole Proprietorship L.L.C.'

LC\_PAYROLL\_NAME = 'C2i Payroll'

)

IF trim(POSITION5) ='Cloud WTC' THEN

(

LC\_LEGAL\_EMPLOYER = 'Cloud Spaces - Sole Proprietorship L.L.C.'

LC\_PAYROLL\_NAME = 'C2i Payroll'

)

IF trim(POSITION5) ='Cloud WTC' THEN

(

LC\_LEGAL\_EMPLOYER = 'Cloud Spaces - Sole Proprietorship L.L.C.'

LC\_PAYROLL\_NAME = 'C2i Payroll'

)

IF trim(POSITION5) ='C2i Yas Food Hall' THEN

(

LC\_LEGAL\_EMPLOYER = 'C2I Holding - Sole Proprietorship L.L.C.'

LC\_PAYROLL\_NAME = 'C2i Payroll'

)

/\*METADATA|Position|SourceSystemOwner|SourceSystemId|EffectiveStartDate|EffectiveEndDate|Name|PositionCode|BusinessUnitName|ActiveStatus|JobCode|JobSetCode|DepartmentName|HiringStatus|PositionType|FTE|HeadCount|FullPartTime|WorkingHours|Frequency

MERGE|Position|LEVER|TECHNOLOGY LEAD - TECHNOLOGY|2021/01/01|4712/12/31|Technology Lead - Technology|Technology Lead - Technology|ADRES|A|HEAD|COMMON|Technology|APPROVED|NONE|100|100|FULL\_TIME|40|W

\*/

/\*

IF LINEREPEATNO=1 THEN

(

FileName='Position'

RET = ESS\_LOG\_WRITE('LINEREPEATNO 1 before update :' ||TO\_CHAR(LINEREPEATNO))

LINEREPEATNO=LINEREPEATNO+1

RET = ESS\_LOG\_WRITE('LINEREPEATNO 1 after update :' ||TO\_CHAR(LINEREPEATNO))

RET = ESS\_LOG\_WRITE('LINENO 1 :' ||TO\_CHAR(LINENO))

FileDiscriminator = 'Position'

dummy = ESS\_LOG\_WRITE('FileDiscriminator after change : '||FileDiscriminator)

SourceSystemId = 'LEVER\_POSITION\_' || trim(upper(POSITION9))

EffectiveStartDate ='1951/01/01'

EffectiveEndDate ='4712/12/31'

Name =trim(POSITION9)

PositionCode = 'LEVER\_POSITION' ||'\_'||LC\_TIME trim(UPPER(SUBSTR(POSITION9,1,30)))

BusinessUnitName = trim(POSITION5)

ActiveStatus = 'A'

JobCode =trim(UPPER(SUBSTR(POSITION8,1,30)))

JobSetCode = 'COMMON'

DepartmentName =trim(POSITION4)

HiringStatus ='APPROVED'

PositionType ='NONE'

FTE ='100'

HeadCount ='100'

FullPartTime ='FULL\_TIME'

WorkingHours ='40'

Frequency ='W'

LINEREPEAT = 'Y'

RETURN

BusinessOperation,FileName,FileDiscriminator,SourceSystemOwner,SourceSystemId,EffectiveStartDate,EffectiveEndDate,Name,PositionCode,BusinessUnitName,ActiveStatus,JobCode,JobSetCode,DepartmentName,HiringStatus,PositionType,FTE,HeadCount,FullPartTime,WorkingHours,Frequency,LINEREPEATNO,LINEREPEAT

)

\*/

IF LINEREPEATNO=1 THEN

(

RET = ESS\_LOG\_WRITE('LINEREPEATNO 2 before update :' ||TO\_CHAR(LINEREPEATNO))

LINEREPEATNO=LINEREPEATNO+1

RET = ESS\_LOG\_WRITE('LINEREPEATNO 2 after update :' ||TO\_CHAR(LINEREPEATNO))

RET = ESS\_LOG\_WRITE('LINENO 2 :' ||TO\_CHAR(LINENO))

FileDiscriminator = 'Worker' /\*FileDiscriminator\*/

dummy = ESS\_LOG\_WRITE('FileDiscriminator after change 2: '||FileDiscriminator)

SourceSystemId = 'LEVER\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

StartDate = trim(POSITION11)

DateStart = trim(POSITION11)

DateOfBirth = trim(POSITION12)

ActionCode = 'ADD\_PEN\_WKR'

LINEREPEAT = 'Y'

RETURN

BusinessOperation,FileName,FileDiscriminator,SourceSystemOwner,SourceSystemId,EffectiveStartDate,EffectiveEndDate,StartDate,DateOfBirth,ActionCode,LINEREPEATNO,LINEREPEAT

)

IF LINEREPEATNO=2 THEN

(

RET = ESS\_LOG\_WRITE('LINEREPEATNO 3 :' ||TO\_CHAR(LINEREPEATNO))

RET = ESS\_LOG\_WRITE('ORIGINAL LINENO 3 :' ||TO\_CHAR(LINENO ))

LINEREPEATNO = LINEREPEATNO +1

FileDiscriminator = 'PersonAddress' /\*FileDiscriminator\*/

dummy = ESS\_LOG\_WRITE('FileDiscriminator after change 3: '||FileDiscriminator)

SourceSystemId = 'LEVER\_ADDRESS\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

"PersonId(SourceSystemId)" = 'LEVER\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

AddressType = 'HOME'

AddressLine1 = '0000' /\*trim(POSITION11) \*/

AddressLine2 = '0000' /\*trim(POSITION12) \*/

Region1 = 'Abu Dhabi' /\*trim(POSITION13)\*/

Region2 = 'Abu Dhabi' /\*trim(POSITION14)\*/

Region3 = 'Abu Dhabi' /\*trim(POSITION15)\*/

Country = 'AE' /\*trim(POSITION15)\*/

PostalCode = '51133' /\*trim(POSITION16)\*/

LINEREPEATNO=LINEREPEATNO+1

LINEREPEAT = 'Y'

RETURN BusinessOperation,FileName,FileDiscriminator,SourceSystemOwner,SourceSystemId,"PersonId(SourceSystemId)", EffectiveStartDate,EffectiveEndDate,AddressType,AddressLine1 ,AddressLine2 , Region1, Region2, Region3, Country , PostalCode ,LINEREPEATNO,LINEREPEAT

)

IF LINEREPEATNO=3 THEN

(

RET = ESS\_LOG\_WRITE('LINEREPEATNO 4 :' ||TO\_CHAR(LINEREPEATNO))

RET = ESS\_LOG\_WRITE('ORIGINAL LINENO 4 :' ||TO\_CHAR(LINENO ))

FileDiscriminator = 'PersonCitizenship' /\*FileDiscriminator\*/

dummy = ESS\_LOG\_WRITE('FileDiscriminator after change 4: '||FileDiscriminator)

SourceSystemId = 'LEVER\_CITIZEN\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

"PersonId(SourceSystemId)" = 'LEVER\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

dummy = ESS\_LOG\_WRITE('PersonId(SourceSystemId) after change 4: '||"PersonId(SourceSystemId)")

DateFrom= trim(POSITION11)

DateTo= '4712/12/31'

LegislationCode= trim(UPPER(SUBSTR(POSITION14,1,2)))

CitizenshipStatus= 'A'

LINEREPEATNO=LINEREPEATNO+1

LINEREPEAT = 'Y'

RETURN BusinessOperation,FileName,FileDiscriminator,SourceSystemOwner,SourceSystemId,"PersonId(SourceSystemId)",DateFrom ,DateTo , LegislationCode , CitizenshipStatus ,LINEREPEATNO,LINEREPEAT

)

IF LINEREPEATNO=4 THEN

(

RET = ESS\_LOG\_WRITE('LINEREPEATNO 5 :' ||TO\_CHAR(LINEREPEATNO))

RET = ESS\_LOG\_WRITE('ORIGINAL LINENO 5 :' ||TO\_CHAR(LINENO ))

FileDiscriminator = 'PersonLegislativeData' /\*FileDiscriminator\*/

dummy = ESS\_LOG\_WRITE('FileDiscriminator after change 5: '||FileDiscriminator)

SourceSystemId = 'LEVER\_LEG\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

"PersonId(SourceSystemId)" = 'LEVER\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

LegislationCode= 'AE'

IF trim(UPPER(POSITION16))='MARRIED' THEN

( MaritalStatus='M')

ELSE

( MaritalStatus='S')

IF trim(UPPER(POSITION13))='MALE' THEN

( Sex='M')

ELSE IF trim(UPPER(POSITION13))='FEMALE' THEN

( Sex='F' )

LINEREPEATNO=LINEREPEATNO+1

LINEREPEAT = 'Y'

RETURN BusinessOperation,FileName,FileDiscriminator,SourceSystemOwner,SourceSystemId,"PersonId(SourceSystemId)",EffectiveStartDate,EffectiveEndDate,LegislationCode,MaritalStatus,Sex ,LINEREPEATNO,LINEREPEAT

)

IF LINEREPEATNO=5 THEN

(

RET = ESS\_LOG\_WRITE('LINEREPEATNO 6 :' ||TO\_CHAR(LINEREPEATNO))

RET = ESS\_LOG\_WRITE('ORIGINAL LINENO 6 :' ||TO\_CHAR(LINENO ))

FileDiscriminator = 'PersonName' /\*FileDiscriminator\*/

dummy = ESS\_LOG\_WRITE('FileDiscriminator after change 6: '||FileDiscriminator)

SourceSystemId = 'LEVER\_NAME\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

"PersonId(SourceSystemId)" = 'LEVER\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

LegislationCode= 'AE'

NameType='GLOBAL'

FirstName=trim(POSITION2)

/\*IF ISNULL(POSITION3)='Y' THEN

(

LastName='.'

)

ELSE \*/

LastName=trim(POSITION3)

IF trim(UPPER(POSITION13))='MALE' THEN

( Title ='MR.')

ELSE IF trim(UPPER(POSITION13))='FEMALE' THEN

( Title ='MS.')

ELSE IF (trim(UPPER(POSITION13))='FEMALE' AND trim(UPPER(POSITION16))='MARRIED' ) THEN

( Title ='MRS.')

LINEREPEATNO=LINEREPEATNO+1

LINEREPEAT = 'Y'

RETURN BusinessOperation,FileName,FileDiscriminator,SourceSystemOwner,SourceSystemId,"PersonId(SourceSystemId)",EffectiveStartDate,EffectiveEndDate,LegislationCode,NameType, FirstName, LastName, Title ,LINEREPEATNO,LINEREPEAT

)

IF LINEREPEATNO=6 THEN

(

RET = ESS\_LOG\_WRITE('LINEREPEATNO 9 :' ||TO\_CHAR(LINEREPEATNO))

RET = ESS\_LOG\_WRITE('ORIGINAL LINENO 9 :' ||TO\_CHAR(LINENO ))

FileDiscriminator = 'WorkRelationship' /\*FileDiscriminator\*/

dummy = ESS\_LOG\_WRITE('FileDiscriminator after change 9: '||FileDiscriminator)

SourceSystemId = 'LEVER\_WR\_SERVICE\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

"PersonId(SourceSystemId)" = 'LEVER\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

LegalEmployerName = LC\_LEGAL\_EMPLOYER /\* Aldar Properties LEGAL EMPLOYER HAS TO BE DERIVED FROM BUSINESS UNIT \*/

DateStart = trim(POSITION11)

/\* PrimaryFlag = 'Y' \*/

WorkerType = 'P' /\*'E' \*/

OnMilitaryServiceFlag = 'N'

RehireRecommendationFlag = 'Y'

ActionCode = 'ADD\_PEN\_WKR'

ReadyToConvert = 'Yes'

LINEREPEATNO=LINEREPEATNO+1

LINEREPEAT = 'Y'

RETURN BusinessOperation,FileName,FileDiscriminator,SourceSystemOwner,SourceSystemId,"PersonId(SourceSystemId)",LegalEmployerName,DateStart/\*,PrimaryFlag\*/ ,WorkerType,OnMilitaryServiceFlag,RehireRecommendationFlag,ActionCode,ReadyToConvert,LINEREPEATNO,LINEREPEAT

)

IF LINEREPEATNO=7 THEN

(

RET = ESS\_LOG\_WRITE('LINEREPEATNO 10 :' ||TO\_CHAR(LINEREPEATNO))

RET = ESS\_LOG\_WRITE('ORIGINAL LINENO 10 :' ||TO\_CHAR(LINENO ))

FileDiscriminator = 'WorkTerms' /\*FileDiscriminator\*/

dummy = ESS\_LOG\_WRITE('FileDiscriminator after change 10: '||FileDiscriminator)

SourceSystemId = 'LEVER\_WT\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

"PersonId(SourceSystemId)" = 'LEVER\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

EffectiveSequence ='1'

EffectiveLatestChange = 'Y'

DateStart = trim(POSITION11)

/\* AssignmentName = 'ET'||LC\_TIME \*/

AssignmentNumber = 'ET'||LC\_TIME

AssignmentStatusTypeCode = 'ACTIVE\_PROCESS'

/\* AssignmentType ='ET' \*/

BusinessUnitShortCode = trim(POSITION5) /\*Aldar Development\*/

/\* PositionOverrideFlag = 'Y' \*/

PrimaryWorkTermsFlag = 'Y'

"PeriodOfServiceId(SourceSystemId)" = 'LEVER\_WR\_SERVICE\_' || trim(upper(POSITION2))||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

LegalEmployerName = LC\_LEGAL\_EMPLOYER /\*Aldar Properties LEGAL EMPLOYER HAS TO BE DERIVED BASED ON BUSINESS UNIT\*/

/\*SystemPersonType = 'EMP' \*/

ActionCode = 'ADD\_PEN\_WKR'

PersonTypeCode = 'Pending Worker'

ProposedUserPersonType = 'Employee'

WorkerType = 'P'

LINEREPEATNO=LINEREPEATNO+1

LINEREPEAT = 'Y'

RETURN BusinessOperation,FileName,FileDiscriminator,SourceSystemOwner,SourceSystemId,"PersonId(SourceSystemId)",EffectiveStartDate /\*,EffectiveEndDate\*/ ,EffectiveSequence,EffectiveLatestChange,DateStart /\*,AssignmentName\*/,AssignmentNumber ,AssignmentStatusTypeCode /\*,AssignmentType\*/ ,BusinessUnitShortCode/\*,PositionOverrideFlag\*/ ,PrimaryWorkTermsFlag,"PeriodOfServiceId(SourceSystemId)",LegalEmployerName/\*,SystemPersonType\*/,ActionCode,PersonTypeCode,ProposedUserPersonType,WorkerType,LINEREPEATNO,LINEREPEAT

)

IF LINEREPEATNO=8 THEN

(

RET = ESS\_LOG\_WRITE('LINEREPEATNO 11 :' ||TO\_CHAR(LINEREPEATNO))

RET = ESS\_LOG\_WRITE('ORIGINAL LINENO 11 :' ||TO\_CHAR(LINENO ))

FileDiscriminator = 'Contract' /\*FileDiscriminator\*/

dummy = ESS\_LOG\_WRITE('FileDiscriminator after change 11: '||FileDiscriminator)

SourceSystemId = 'LEVER\_CONTRACT\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

"PersonId(SourceSystemId)" = 'LEVER\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

"AssignmentId(SourceSystemId)" = 'LEVER\_WT\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

ContractType = 'Long Term'

AssignmentNumber = 'ET'||LC\_TIME

ContractNumber = 'CONT'||LC\_TIME

LINEREPEATNO=LINEREPEATNO+1

LINEREPEAT = 'Y'

RETURN BusinessOperation,FileName,FileDiscriminator,SourceSystemOwner,SourceSystemId,"PersonId(SourceSystemId)","AssignmentId(SourceSystemId)",ContractType,EffectiveStartDate,AssignmentNumber,ContractNumber,LINEREPEATNO,LINEREPEAT

)

IF LINEREPEATNO=9 THEN

(

RET = ESS\_LOG\_WRITE('LINEREPEATNO 12 :' ||TO\_CHAR(LINEREPEATNO))

RET = ESS\_LOG\_WRITE('ORIGINAL LINENO 12 :' ||TO\_CHAR(LINENO ))

FileDiscriminator = 'Assignment' /\*FileDiscriminator\*/

dummy = ESS\_LOG\_WRITE('FileDiscriminator after change 12: '||FileDiscriminator)

SourceSystemId = 'LEVER\_ASSIGN\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

ActionCode = 'ADD\_PEN\_WKR'

EffectiveStartDate = trim(POSITION11)

EffectiveSequence ='1'

EffectiveLatestChange = 'Y'

"WorkTermsAssignmentId(SourceSystemId)" = 'LEVER\_WT\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

AssignmentName = 'E'||LC\_TIME

AssignmentNumber = 'E'||LC\_TIME

AssignmentStatusTypeCode = 'ACTIVE\_PROCESS'

PersonTypeCode = 'Pending Worker' /\*'Employee'\*/

BusinessUnitShortCode = trim(POSITION5)

LegalEmployerName = LC\_LEGAL\_EMPLOYER /\* Aldar Properties LEGAL EMPLOYER HAS TO BE DERIVED FROM BUSINESS UNIT \*/

PrimaryAssignmentFlag = 'Y'

JobCode = trim(UPPER(SUBSTR(POSITION8,1,30)))

PositionCode = trim(POSITION20)

GradeCode = trim(POSITION6)

/\*PositionOverrideFlag = 'N' \*/

AssignmentCategory = trim(POSITION17)

DepartmentName = trim(POSITION4)

WorkTermsNumber = 'ET'||LC\_TIME

"PersonId(SourceSystemId)" = 'LEVER\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

"PeriodOfServiceId(SourceSystemId)" = 'LEVER\_WR\_SERVICE\_' || trim(upper(POSITION2))||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

DateStart = trim(POSITION11)

ProjectedStartDate = trim(POSITION11)

WorkerType ='P' /\*'E' \*/

PersonTypeCode ='Pending Worker'

HourlySalariedCode ='Salaried'

FullPartTime = 'Full time'

PermanentTemporary ='Regular'

LINEREPEATNO=LINEREPEATNO+1

LINEREPEAT = 'Y'

RETURN BusinessOperation,FileName,FileDiscriminator,SourceSystemOwner,SourceSystemId,ActionCode,EffectiveStartDate,EffectiveSequence,EffectiveLatestChange,"WorkTermsAssignmentId(SourceSystemId)",AssignmentName,AssignmentNumber,AssignmentStatusTypeCode,PersonTypeCode,BusinessUnitShortCode,LegalEmployerName,ProposedUserPersonType,PrimaryAssignmentFlag,JobCode,GradeCode,AssignmentCategory,PositionCode/\*, PositionOverrideFlag \*/ , DepartmentName, WorkTermsNumber, "PersonId(SourceSystemId)","PeriodOfServiceId(SourceSystemId)",DateStart,ProjectedStartDate,WorkerType,HourlySalariedCode,FullPartTime,PermanentTemporary, LINEREPEATNO,LINEREPEAT

)

/\*METADATA|AssignedPayroll|EffectiveStartDate|AssignmentNumber|PayrollDefinitionCode|LegislativeDataGroupName|StartDate

MERGE|AssignedPayroll|2019/03/01|E10001|Weekly Payroll|Vision US LDG|2019/03/01

\*/

/\* Need to code based on unit id to default PayrollDefinitionCode

ADRES Payroll

AIPM Payroll

Aldar Corporate Payroll

Aldar Development Payroll

Aldar Projects Payroll

C2i Payroll

Gateway Payroll

Teyaseer Payroll

\*/

IF LINEREPEATNO=10 THEN

(

FileName = 'AssignedPayroll'

RET = ESS\_LOG\_WRITE('LINEREPEATNO 13 :' ||TO\_CHAR(LINEREPEATNO))

RET = ESS\_LOG\_WRITE('ORIGINAL LINENO 13 :' ||TO\_CHAR(LINENO ))

FileDiscriminator = 'AssignedPayroll' /\*FileDiscriminator\*/

dummy = ESS\_LOG\_WRITE('FileDiscriminator after change 13: '||FileDiscriminator)

SourceSystemId = 'LEVER\_ASS\_PAYROLL' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

EffectiveStartDate = trim(POSITION11)

StartDate = trim(POSITION11)

"AssignmentId(SourceSystemId)" = 'LEVER\_ASSIGN\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

PayrollDefinitionCode = LC\_PAYROLL\_NAME /\*'Aldar Development Payroll' \*/

LegislativeDataGroupName = 'AE Legislative Data Group'

LINEREPEATNO=LINEREPEATNO+1

LINEREPEAT = 'Y'

RETURN BusinessOperation,FileName,FileDiscriminator,SourceSystemOwner,SourceSystemId,EffectiveStartDate,"AssignmentId(SourceSystemId)",PayrollDefinitionCode,LegislativeDataGroupName,StartDate,LINEREPEATNO,LINEREPEAT

)

/\* MERGE|Salary|2024/04/01||38577|N|E465|Aldar Basic Salary|CHANGE\_SALARY \*/

IF LINEREPEATNO=11 THEN

(

FileName = 'Salary'

RET = ESS\_LOG\_WRITE('LINEREPEATNO 14 :' ||TO\_CHAR(LINEREPEATNO))

RET = ESS\_LOG\_WRITE('ORIGINAL LINENO 14 :' ||TO\_CHAR(LINENO ))

FileDiscriminator = 'Salary' /\*FileDiscriminator\*/

dummy = ESS\_LOG\_WRITE('FileDiscriminator after change 14: '||FileDiscriminator)

SourceSystemId = 'LEVER\_SAL\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

DateFrom = trim(POSITION11)

SalaryAmount = trim(UPPER(POSITION10))

MultipleComponents = 'N'

"AssignmentId(SourceSystemId)" = 'LEVER\_ASSIGN\_' || trim(upper(POSITION2)) ||'\_'||LC\_TIME /\*\_2024\_05\_20'\*/

/\* AssignmentNumber = 'E'||LC\_TIME \*/

SalaryBasisName = 'Aldar Basic Salary'

ActionCode = 'ADD\_PEN\_WKR'

/\*LINEREPEATNO=LINEREPEATNO+1\*/

LINEREPEAT = 'N'

RETURN BusinessOperation,FileName,FileDiscriminator,SourceSystemOwner,SourceSystemId,DateFrom,SalaryAmount,MultipleComponents,"AssignmentId(SourceSystemId)",SalaryBasisName,ActionCode,LINEREPEATNO,LINEREPEAT

)

)

ELSE

OUTPUTVALUE='NONE'

RETURN OUTPUTVALUE

/\* End Formula Text \*/

/\*

MODIFIED FORMULA NAME: ALDAR\_PROBATION\_CONFIRMATION\_AUTOMATION\_STEP\_1

FORMULA TYPE: HCM Data Loader

DESCRIPTION: This formula will create Worker.dat based on probation confirmation report text

created by Abdulrahman 10-Mar-2025

\*/

INPUTS ARE OPERATION (text), LINENO (number), LINEREPEATNO (number),POSITION1 (text), POSITION2 (text), POSITION3 (text), POSITION4 (text), POSITION5 (text), POSITION6 (text),POSITION7 (text), POSITION8 (text), POSITION9 (text), POSITION10 (text), POSITION11 (text), POSITION12 (text),POSITION13 (text)

DEFAULT FOR LINENO IS 1

DEFAULT FOR LINEREPEATNO IS 1

/\* ActionCode|PERSON\_NUMBER|PersonId(SOURCE\_SYSTEM\_ID)|EFFECTIVE\_START\_DATE|DATE\_START|WORKER\_TYPE|ACTION\_CODE|REASON\_CODE|LEGAL\_EMPLOYER\_NAME|BUSINESS\_UNIT|PeriodOfServiceId(SourceSystemId)|WorkTermsAssignmentId(SourceSystemId)|AssignmentId(SourceSystemId) \*/

/\* END\_PROBATION|PERSON\_NUMBER|

DEFAULT FOR POSITION1 IS 'NO DATA' /\* New field with Data Field with Data END\_PROBATION \*/

DEFAULT FOR POSITION2 IS 'NO DATA' /\* PERSON\_NUMBER \*/

DEFAULT FOR POSITION3 IS 'NO DATA' /\* PersonId(SOURCE\_SYSTEM\_ID) \*/

DEFAULT FOR POSITION4 IS 'NO DATA' /\* EFFECTIVE\_START\_DATE \*/

DEFAULT FOR POSITION5 IS 'NO DATA' /\* DATE\_START \*/

DEFAULT FOR POSITION6 IS 'NO DATA' /\* WORKER\_TYPE\*/

DEFAULT FOR POSITION7 IS 'NO DATA' /\* ACTION\_CODE \*/

DEFAULT FOR POSITION8 IS 'NO DATA' /\* REASON\_CODE\*/

DEFAULT FOR POSITION9 IS 'NO DATA' /\* LEGAL\_EMPLOYER\_NAME \*/

DEFAULT FOR POSITION10 IS 'NO DATA' /\* BUSINESS\_UNIT \*/

DEFAULT FOR POSITION11 IS 'NO DATA' /\* PeriodOfServiceId(SourceSystemId) \*/

DEFAULT FOR POSITION12 IS 'NO DATA' /\* WorkTermsAssignmentId(SourceSystemId) \*/

DEFAULT FOR POSITION13 IS 'NO DATA' /\* AssignmentId(SourceSystemId) \*/

utc\_date = GET\_SYSDATE()

cst\_date = NEW\_TIME(utc\_date, 'GMT', 'CST')

IF OPERATION = 'FILETYPE' THEN

OUTPUTVALUE = 'DELIMITED'

ELSE IF OPERATION = 'DELIMITER' THEN

OUTPUTVALUE = '|'

ELSE IF OPERATION = 'READ' THEN

OUTPUTVALUE = 'NONE'

ELSE IF OPERATION = 'NUMBEROFBUSINESSOBJECTS' THEN

OUTPUTVALUE = '2'

ELSE IF OPERATION = 'METADATALINEINFORMATION' THEN

(

METADATA1[1] = 'Worker' /\*FileName\*/ /\*Reserved\*/

METADATA1[2] = 'WorkTerms'

METADATA1[3] = 'PersonNumber'

METADATA1[4] = 'PersonId(SourceSystemId)'

METADATA1[5] = 'PeriodOfServiceId(SourceSystemId)'

METADATA1[6] = 'EffectiveLatestChange'

METADATA1[7] = 'EffectiveSequence'

METADATA1[8] = 'EffectiveStartDate'

METADATA1[9] = 'DateStart'

METADATA1[10] = 'WorkerType'

METADATA1[11] = 'ActionCode'

METADATA1[12] = 'ReasonCode'

METADATA1[13] = 'SourceSystemOwner'

METADATA1[14] = 'SourceSystemId'

METADATA1[15] = 'PrimaryWorkTermsFlag'

METADATA1[16] = 'LegalEmployerName'

METADATA2[1] = 'Worker' /\*FileName\*/ /\*Reserved\*/

METADATA2[2] = 'Assignment'

METADATA2[3] = 'PersonNumber'

METADATA2[4] = 'PersonId(SourceSystemId)'

METADATA2[5] = 'PeriodOfServiceId(SourceSystemId)'

METADATA2[6] = 'EffectiveStartDate'

METADATA2[7] = 'EffectiveSequence'

METADATA2[8] = 'EffectiveLatestChange'

METADATA2[9] = 'DateStart'

METADATA2[10] = 'WorkerType'

METADATA2[11] = 'ActionCode'

METADATA2[12] = 'ReasonCode'

METADATA2[13] = 'WorkTermsAssignmentId(SourceSystemId)'

METADATA2[14] = 'SourceSystemId'

METADATA2[15] = 'SourceSystemOwner'

METADATA2[16] = 'PrimaryFlag'

METADATA2[17] = 'BusinessUnitShortCode'

METADATA2[18] = 'LegalEmployerName'

METADATA2[19] = 'FLEX:PER\_ASG\_DF'

METADATA2[20] = 'probationConfirmed(PER\_ASG\_DF=Global Data Elements)'

RETURN METADATA1, METADATA2

)

ELSE IF (OPERATION='MAP' AND POSITION1='END\_PROBATION') THEN

(

FileName = 'Worker'

BusinessOperation = 'MERGE'

SourceSystemOwner = 'FUSION'

ld\_EffectiveStartDate = trim(POSITION4)

/\*DateStart = trim(DATE\_START) \*/

ActionCode = 'END\_PROBATION'

PrimaryWorkTermsFlag = 'Y'

IF LINEREPEATNO=1 THEN

(

RET = ESS\_LOG\_WRITE('LINEREPEATNO 1 :' ||TO\_CHAR(LINEREPEATNO))

RET = ESS\_LOG\_WRITE('ORIGINAL LINENO 1 :' ||TO\_CHAR(LINENO ))

FileDiscriminator = 'WorkTerms' /\*FileDiscriminator\*/

dummy = ESS\_LOG\_WRITE('FileDiscriminator after change 10: '||FileDiscriminator)

PersonNumber = TRIM(POSITION2)

"PersonId(SourceSystemId)" = TRIM(POSITION3)

"PeriodOfServiceId(SourceSystemId)" = TRIM(POSITION11)

EffectiveLatestChange = 'Y'

EffectiveSequence = '1'

EffectiveStartDate = trim(POSITION4)

DateStart = trim(POSITION5)

WorkerType = 'E'

ActionCode = 'END\_PROBATION'

SourceSystemOwner = 'FUSION'

SourceSystemId = trim(POSITION12)

PrimaryWorkTermsFlag = 'Y'

LegalEmployerName = trim(POSITION9) /\*Aldar Properties LEGAL EMPLOYER HAS TO BE DERIVED BASED ON BUSINESS UNIT\*/

SystemPersonType = 'EMP'

LINEREPEATNO=LINEREPEATNO+1

LINEREPEAT = 'Y'

RETURN BusinessOperation,FileName,FileDiscriminator,PersonNumber,"PersonId(SourceSystemId)","PeriodOfServiceId(SourceSystemId)",EffectiveLatestChange,EffectiveSequence,EffectiveStartDate,DateStart, WorkerType, ActionCode, SourceSystemOwner, SourceSystemId, PrimaryWorkTermsFlag, LegalEmployerName, SystemPersonType, LINEREPEATNO, LINEREPEAT

)

IF LINEREPEATNO=2 THEN

(

RET = ESS\_LOG\_WRITE('LINEREPEATNO 2 :' ||TO\_CHAR(LINEREPEATNO))

RET = ESS\_LOG\_WRITE('ORIGINAL LINENO 2 :' ||TO\_CHAR(LINENO ))

FileDiscriminator = 'Assignment' /\*FileDiscriminator\*/

dummy = ESS\_LOG\_WRITE('FileDiscriminator after change 2: '||FileDiscriminator)

PersonNumber = TRIM(POSITION2)

"PersonId(SourceSystemId)" = TRIM(POSITION3)

"PeriodOfServiceId(SourceSystemId)" = TRIM(POSITION11)

EffectiveStartDate = trim(POSITION4)

EffectiveSequence = '1'

EffectiveLatestChange = 'Y'

DateStart = trim(POSITION5)

WorkerType ='E'

ActionCode = 'END\_PROBATION'

"WorkTermsAssignmentId(SourceSystemId)" = trim(POSITION12)

SourceSystemId = trim(POSITION13)

SourceSystemOwner = 'FUSION'

PrimaryFlag = 'Y'

BusinessUnitShortCode = trim(POSITION10)

LegalEmployerName = trim(POSITION9)

"FLEX:PER\_ASG\_DF" = 'Global Data Elements'

"probationConfirmed(PER\_ASG\_DF=Global Data Elements)" = 'Y'

LINEREPEATNO=LINEREPEATNO+1

LINEREPEAT = 'Y'

RETURN BusinessOperation,FileName,FileDiscriminator,PersonNumber, "PersonId(SourceSystemId)","PeriodOfServiceId(SourceSystemId)", EffectiveStartDate, EffectiveSequence, EffectiveLatestChange, DateStart,WorkerType, ActionCode, "WorkTermsAssignmentId(SourceSystemId)", SourceSystemId, SourceSystemOwner, PrimaryFlag ,BusinessUnitShortCode, LegalEmployerName, "FLEX:PER\_ASG\_DF","probationConfirmed(PER\_ASG\_DF=Global Data Elements)", LINEREPEATNO, LINEREPEAT

)

)

ELSE

OUTPUTVALUE = 'NONE'

RETURN OUTPUTVALUE

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*

\* FORMULA NAME: GENERATE\_PERSONACCRUALENTRY\_HDL\_FROM\_TXT

\* FORMULA TYPE: HCM Data Loader

\* DESCRIPTION: This formula will create Worker HDL File from TXT File

\* CHANGE HISTORY:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Version Date Created By Comments

-----------------------------------------------------------------------

1.0 14-Jul-2023 Abdulrahman Mohammed Initial Version

2.0 03-Nov-2023 Abdulrahman Mohammed Added AD Last working date field

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Inputs \*/

INPUTS ARE OPERATION (text), LINENO (number), LINEREPEATNO (number),POSITION1 (text), POSITION2 (text), POSITION3 (text), POSITION4 (text),

POSITION5 (text), POSITION6 (text), POSITION7 (text), POSITION8 (text), POSITION9 (text), POSITION10 (text), POSITION11 (text), POSITION12 (text), POSITION13 (text), POSITION14 (text),POSITION15 (text),POSITION16 (text),POSITION17 (text)

DEFAULT FOR POSITION1 IS 'NO DATA'

DEFAULT FOR POSITION2 IS 'NO DATA'

DEFAULT FOR POSITION3 IS 'NO DATA'

DEFAULT FOR POSITION4 IS 'NO DATA'

DEFAULT FOR POSITION5 IS 'NO DATA'

DEFAULT FOR POSITION6 IS 'NO DATA'

DEFAULT FOR POSITION7 IS 'NO DATA'

DEFAULT FOR POSITION8 IS 'NO DATA'

DEFAULT FOR POSITION9 IS 'NO DATA'

DEFAULT FOR POSITION10 IS 'NO DATA'

DEFAULT FOR POSITION11 IS 'NO DATA'

DEFAULT FOR POSITION12 IS 'NO DATA'

DEFAULT FOR POSITION13 IS 'NO DATA'

DEFAULT FOR POSITION14 IS 'NO DATA'

DEFAULT FOR POSITION15 IS 'NO DATA'

DEFAULT FOR POSITION16 IS 'NO DATA'

DEFAULT FOR POSITION17 IS 'NO DATA'

DEFAULT FOR LINEREPEATNO IS 1

IF OPERATION='FILETYPE' THEN

OUTPUTVALUE='DELIMITED'

ELSE IF OPERATION='DELIMITER' THEN

OUTPUTVALUE='|'

ELSE IF OPERATION='READ' THEN

OUTPUTVALUE='NONE'

ELSE IF OPERATION = 'NUMBEROFBUSINESSOBJECTS' THEN

(

OUTPUTVALUE= '1'

RETURN OUTPUTVALUE

)

ELSE IF OPERATION = 'METADATALINEINFORMATION' THEN

(

METADATA1[1] = 'Worker' /\*FileName\*/ /\*Reserved\*/

METADATA1[2] = 'WorkerExtraInfo' /\*FileDiscriminator\*/ /\*Reserved\*/

METADATA1[3] = 'EffectiveStartDate'

METADATA1[4] = 'EffectiveEndDate'

METADATA1[5] = 'PersonNumber'

METADATA1[6] = 'PeiInformationCategory'

METADATA1[7] = 'CategoryCode'

METADATA1[8] = 'InformationType'

METADATA1[9] = 'EFF\_CATEGORY\_CODE'

METADATA1[10] = 'FLEX:PER\_PERSON\_EIT\_EFF'

METADATA1[11] = 'adDisplayDepartmentName(PER\_PERSON\_EIT\_EFF=Sailpoint\_Active\_Directory\_Upd)'

METADATA1[12] = 'isStaffBelowAvp(PER\_PERSON\_EIT\_EFF=Sailpoint\_Active\_Directory\_Upd)'

METADATA1[13] = 'isStaffPartOfCmOrSalesTeam(PER\_PERSON\_EIT\_EFF=Sailpoint\_Active\_Directory\_Upd)'

METADATA1[14] = 'adTitle(PER\_PERSON\_EIT\_EFF=Sailpoint\_Active\_Directory\_Upd)'

METADATA1[15] = 'SourceSystemOwner'

METADATA1[16] = 'SourceSystemId'

METADATA1[17] = 'adLastWorkingDate(PER\_PERSON\_EIT\_EFF=Sailpoint\_Active\_Directory\_Upd)'

RETURN METADATA1

)

ELSE IF OPERATION='MAP' THEN

/\*HDL Related Outputs\*/

(

IF LINEREPEATNO = 1 THEN

(

IF POSITION4 <> 'PersonNumber' THEN

(

LINEREPEAT = 'Y'

FileName = 'Worker'

BusinessOperation = 'MERGE'

FileDiscriminator = 'WorkerExtraInfo' /\*trim(POSITION1)\*/

EffectiveStartDate = trim(POSITION3)

EffectiveEndDate = trim(POSITION4)

PersonNumber = trim(POSITION5)

PeiInformationCategory = trim(POSITION6)

CategoryCode = trim(POSITION7)

InformationType = trim(POSITION8)

"EFF\_CATEGORY\_CODE" = trim(POSITION9)

"FLEX:PER\_PERSON\_EIT\_EFF" = trim(POSITION10)

"adDisplayDepartmentName(PER\_PERSON\_EIT\_EFF=Sailpoint\_Active\_Directory\_Upd)" = trim(POSITION11)

"isStaffBelowAvp(PER\_PERSON\_EIT\_EFF=Sailpoint\_Active\_Directory\_Upd)" = trim(POSITION12)

"isStaffPartOfCmOrSalesTeam(PER\_PERSON\_EIT\_EFF=Sailpoint\_Active\_Directory\_Upd)" = trim(POSITION13)

"adTitle(PER\_PERSON\_EIT\_EFF=Sailpoint\_Active\_Directory\_Upd)" = trim(POSITION14)

SourceSystemOwner = trim(POSITION15)

"SourceSystemId" = trim(POSITION16)

"adLastWorkingDate(PER\_PERSON\_EIT\_EFF=Sailpoint\_Active\_Directory\_Upd)" = trim(POSITION17)

RETURN BusinessOperation,FileName,FileDiscriminator,EffectiveStartDate,EffectiveEndDate,PersonNumber,PeiInformationCategory,CategoryCode,InformationType,"EFF\_CATEGORY\_CODE","FLEX:PER\_PERSON\_EIT\_EFF","adDisplayDepartmentName(PER\_PERSON\_EIT\_EFF=Sailpoint\_Active\_Directory\_Upd)","isStaffBelowAvp(PER\_PERSON\_EIT\_EFF=Sailpoint\_Active\_Directory\_Upd)","isStaffPartOfCmOrSalesTeam(PER\_PERSON\_EIT\_EFF=Sailpoint\_Active\_Directory\_Upd)","adTitle(PER\_PERSON\_EIT\_EFF=Sailpoint\_Active\_Directory\_Upd)",SourceSystemOwner,"SourceSystemId","adLastWorkingDate(PER\_PERSON\_EIT\_EFF=Sailpoint\_Active\_Directory\_Upd)",LINEREPEAT,LINEREPEATNO

)

)

)

ELSE

OUTPUTVALUE='NONE'

RETURN OUTPUTVALUE

/\* End Formula Text \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*

\* FORMULA NAME: GENERATE\_EXTERNALBANKACCOUNTENDDATE\_HDL\_FROM\_TXT

\* FORMULA TYPE: HCM Data Loader

\* DESCRIPTION: This formula will create ExternalBankAccount HDL File for Enddating Employee existing Bank Account from TXT File

\* CHANGE HISTORY:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Version Date Created By Comments

-----------------------------------------------------------------------

1.0 1-Jan-2024 Abdulrahman Mohammed Initial Version

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

INPUTS ARE OPERATION (text), LINENO (number), LINEREPEATNO (number),POSITION1 (text), POSITION2 (text), POSITION3 (text), POSITION4 (text), POSITION5 (text), POSITION6 (text)

DEFAULT FOR POSITION1 IS 'NO DATA'

DEFAULT FOR POSITION2 IS 'NO DATA'

DEFAULT FOR POSITION3 IS 'NO DATA'

DEFAULT FOR POSITION4 IS 'NO DATA'

DEFAULT FOR POSITION5 IS 'NO DATA'

DEFAULT FOR POSITION6 IS 'NO DATA'

DEFAULT FOR LINEREPEATNO IS 1

IF OPERATION='FILETYPE' THEN

OUTPUTVALUE='DELIMITED'

ELSE IF OPERATION='DELIMITER' THEN

OUTPUTVALUE='|'

ELSE IF OPERATION='READ' THEN

OUTPUTVALUE='NONE'

ELSE IF OPERATION = 'NUMBEROFBUSINESSOBJECTS' THEN

(

OUTPUTVALUE = '3'

RETURN OUTPUTVALUE

)

ELSE IF OPERATION = 'METADATALINEINFORMATION' THEN

(

METADATA1[1] = 'ExternalBankAccount' /\*FileName\*/ /\*Reserved\*/

METADATA1[2] = 'ExternalBankAccount' /\*FileDiscriminator\*/ /\*Reserved\*/

METADATA1[3] = 'ExternalBankAccountId'

METADATA1[4] = 'InactiveFlag'

METADATA2[1] = 'ExternalBankAccount' /\*FileName\*/ /\*Reserved\*/

METADATA2[2] = 'ExternalBankAccountOwner' /\*FileDiscriminator\*/

METADATA2[3] = 'ExternalBankAccountId'

METADATA2[4] = 'ExternalBankAccountOwnerId'

METADATA2[5] = 'PersonNumber'

METADATA2[6] = 'PrimaryFlag'

METADATA3[1] = 'PersonalPaymentMethod' /\*FileName\*/ /\*Reserved\*/

METADATA3[2] = 'PersonalPaymentMethod' /\*FileDiscriminator\*/

METADATA3[3] = 'PersonalPaymentMethodId'

METADATA3[4] = 'EffectiveStartDate'

METADATA3[5] = 'EffectiveEndDate'

RETURN METADATA1, METADATA2, METADATA3

)

ELSE IF OPERATION='MAP' THEN

IF POSITION2='ExternalBankAccount' THEN

(

FileName = 'ExternalBankAccount'

BusinessOperation = 'MERGE'

FileDiscriminator = 'ExternalBankAccount' /\*trim(POSITION2)\*/

ExternalBankAccountId = trim(POSITION3)

InactiveFlag = trim(POSITION4)

RETURN BusinessOperation,FileName,FileDiscriminator,ExternalBankAccountId,InactiveFlag

)

IF POSITION2='ExternalBankAccountOwner' THEN

(

FileName = 'ExternalBankAccount'

BusinessOperation = 'MERGE'

FileDiscriminator = 'ExternalBankAccountOwner' /\*trim(POSITION2)\*/

ExternalBankAccountId = trim(POSITION3)

ExternalBankAccountOwnerId = trim(POSITION4)

PersonNumber = trim(POSITION5)

PrimaryFlag = trim(POSITION6)

RETURN BusinessOperation,FileName,FileDiscriminator,ExternalBankAccountId,ExternalBankAccountOwnerId,PersonNumber,PrimaryFlag

)

IF POSITION2='PersonalPaymentMethod' THEN

(

FileName = 'PersonalPaymentMethod'

BusinessOperation = 'MERGE'

FileDiscriminator = 'PersonalPaymentMethod' /\*trim(POSITION2)\*/

PersonalPaymentMethodId = trim(POSITION3)

EffectiveStartDate = trim(POSITION4)

EffectiveEndDate = trim(POSITION5)

RETURN BusinessOperation,FileName,FileDiscriminator,PersonalPaymentMethodId,EffectiveStartDate,EffectiveEndDate

)

ELSE

OUTPUTVALUE='NONE'

RETURN OUTPUTVALUE

/\* End Formula Text \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*

\* FORMULA NAME: ALDAR\_BANK\_ACC\_AUTOMATION\_STEP\_2\_GENERATE\_EXTERNAL\_ACC\_CREATE\_HDL\_FROM\_TXT

\* FORMULA TYPE: HCM Data Loader

\* DESCRIPTION: This formula will create ExternalBankAccount HDL File for Creating Employee existing Bank Account from TXT File

\* CHANGE HISTORY:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Version Date Created By Comments

-----------------------------------------------------------------------

1.0 1-Jan-2024 Abdulrahman Mohammed Initial Version

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

INPUTS ARE OPERATION (text), LINENO (number), LINEREPEATNO (number),POSITION1 (text), POSITION2 (text), POSITION3 (text), POSITION4 (text), POSITION5 (text), POSITION6 (text),POSITION7 (text), POSITION8 (text), POSITION9 (text), POSITION10 (text), POSITION11 (text), POSITION12 (text), POSITION13 (text), POSITION14 (text), POSITION15 (text), POSITION16 (text), POSITION17 (text), POSITION18 (text)

DEFAULT FOR POSITION1 IS 'NO DATA'

DEFAULT FOR POSITION2 IS 'NO DATA'

DEFAULT FOR POSITION1 IS 'NO DATA'

DEFAULT FOR POSITION2 IS 'NO DATA'

DEFAULT FOR POSITION3 IS 'NO DATA'

DEFAULT FOR POSITION4 IS 'NO DATA'

DEFAULT FOR POSITION5 IS 'NO DATA'

DEFAULT FOR POSITION6 IS 'NO DATA'

DEFAULT FOR POSITION7 IS 'NO DATA'

DEFAULT FOR POSITION8 IS 'NO DATA'

DEFAULT FOR POSITION9 IS 'NO DATA'

DEFAULT FOR POSITION10 IS 'NO DATA'

DEFAULT FOR POSITION11 IS 'NO DATA'

DEFAULT FOR POSITION12 IS 'NO DATA'

DEFAULT FOR POSITION13 IS 'NO DATA'

DEFAULT FOR POSITION14 IS 'NO DATA'

DEFAULT FOR POSITION15 IS 'NO DATA'

DEFAULT FOR POSITION16 IS 'NO DATA'

DEFAULT FOR POSITION17 IS 'NO DATA'

DEFAULT FOR POSITION18 IS 'NO DATA'

DEFAULT FOR LINEREPEATNO IS 1

IF OPERATION='FILETYPE' THEN

OUTPUTVALUE='DELIMITED'

ELSE IF OPERATION='DELIMITER' THEN

OUTPUTVALUE='|'

ELSE IF OPERATION='READ' THEN

OUTPUTVALUE='NONE'

ELSE IF OPERATION = 'NUMBEROFBUSINESSOBJECTS' THEN

(

OUTPUTVALUE = '3'

RETURN OUTPUTVALUE

)

ELSE IF OPERATION = 'METADATALINEINFORMATION' THEN

(

/\* METADATA|ExternalBankAccount|SourceSystemOwner|SourceSystemId|AccountNumber|IBAN|BankBranchName|BankName|CountryCode|CurrencyCode|AccountName \*/

METADATA1[1] = 'ExternalBankAccount' /\*FileName\*/ /\*Reserved\*/

METADATA1[2] = 'ExternalBankAccount' /\*FileDiscriminator\*/ /\*Reserved\*/

METADATA1[3] = 'SourceSystemOwner'

METADATA1[4] = 'SourceSystemId'

METADATA1[5] = 'AccountNumber'

METADATA1[6] = 'IBAN'

METADATA1[7] = 'BankBranchName'

METADATA1[8] = 'BankName'

METADATA1[9] = 'CountryCode'

METADATA1[10] = 'CurrencyCode'

METADATA1[11] = 'AccountName'

/\* METADATA|ExternalBankAccountOwner|SourceSystemOwner|SourceSystemId|PersonNumber|AccountNumber|CountryCode|CurrencyCode|BankBranchName|BankName|ExternalBankAccountId(SourceSystemId) \*/

METADATA2[1] = 'ExternalBankAccount' /\*FileName\*/ /\*Reserved\*/

METADATA2[2] = 'ExternalBankAccountOwner' /\*FileDiscriminator\*/

METADATA2[3] = 'SourceSystemOwner'

METADATA2[4] = 'SourceSystemId'

METADATA2[5] = 'PersonNumber'

METADATA2[6] = 'AccountNumber'

METADATA2[7] = 'CountryCode'

METADATA2[8] = 'CurrencyCode'

METADATA2[9] = 'BankBranchName'

METADATA2[10] = 'BankName'

METADATA2[11] = 'ExternalBankAccountId(SourceSystemId)'

/\* METADATA|PersonalPaymentMethod|PersonalPaymentMethodId|LegislativeDataGroupName|AssignmentNumber|PersonalPaymentMethodCode|EffectiveStartDate|EffectiveEndDate|PaymentAmountType|Amount|Percentage|ProcessingOrder|OrganizationPaymentMethodCode|BankName|BankBranchName|BankCountryCode|BankAccountNumber \*/

METADATA3[1] = 'PersonalPaymentMethod' /\*FileName\*/ /\*Reserved\*/

METADATA3[2] = 'PersonalPaymentMethod' /\*FileDiscriminator\*/

METADATA3[3] = 'LegislativeDataGroupName'

METADATA3[4] = 'AssignmentNumber'

METADATA3[5] = 'PersonalPaymentMethodCode'

METADATA3[6] = 'EffectiveStartDate'

METADATA3[7] = 'EffectiveEndDate'

METADATA3[8] = 'PaymentAmountType'

/\* METADATA3[9] = 'Amount' \*/

METADATA3[9] = 'Percentage'

/\*METADATA3[10] = 'ProcessingOrder'\*/

/\*METADATA3[10] = 'OrganizationPaymentMethodCode' \*/

METADATA3[10] = 'BankName'

METADATA3[11] = 'BankBranchName'

METADATA3[12] = 'BankCountryCode'

METADATA3[13] = 'BankAccountNumber'

METADATA3[14] = 'LegislativeDataGroupId'

METADATA3[15] = 'OrganizationPaymentMethodId'

METADATA3[16] = 'SourceSystemOwner'

METADATA3[17] = 'SourceSystemId'

RETURN METADATA1, METADATA2, METADATA3

)

ELSE IF OPERATION='MAP' THEN

IF POSITION2='ExternalBankAccount' THEN

(

FileName = 'ExternalBankAccount'

BusinessOperation = 'MERGE'

FileDiscriminator = 'ExternalBankAccount' /\*trim(POSITION2)\*/

SourceSystemOwner = 'ALDAR'

SourceSystemId = trim(POSITION4)

AccountNumber = trim(POSITION5)

IBAN = trim(POSITION6)

BankBranchName = trim(POSITION7)

BankName = trim(POSITION8)

CountryCode= 'AE'

CurrencyCode = 'AED'

AccountName = trim(POSITION11)

RETURN BusinessOperation,FileName,FileDiscriminator,SourceSystemOwner,SourceSystemId,AccountNumber,IBAN,BankBranchName,BankName,CountryCode,CurrencyCode,AccountName

)

IF POSITION2='ExternalBankAccountOwner' THEN

(

/\* MERGE|ExternalBankAccountOwner|ALDAR|E2\_BAO|25|10150|AE|AED|MAIN BRANCH|Emirates NBD|E2\_BA \*/

FileName = 'ExternalBankAccount'

BusinessOperation = 'MERGE'

FileDiscriminator = 'ExternalBankAccountOwner' /\*trim(POSITION2)\*/

SourceSystemOwner = 'ALDAR'

SourceSystemId = trim(POSITION4)

PersonNumber = trim(POSITION5)

AccountNumber = trim(POSITION6)

CountryCode= 'AE'

CurrencyCode = 'AED'

BankBranchName = trim(POSITION9)

BankName = trim(POSITION10)

"ExternalBankAccountId(SourceSystemId)" = trim(POSITION11)

RETURN BusinessOperation,FileName,FileDiscriminator,SourceSystemOwner,SourceSystemId,PersonNumber,AccountNumber,CountryCode,CurrencyCode,BankBranchName,BankName,"ExternalBankAccountId(SourceSystemId)"

)

IF POSITION2='PersonalPaymentMethod' THEN

(

FileName = 'PersonalPaymentMethod'

/\*old MERGE|PersonalPaymentMethod|3000000|AE Legislative Data Group|E2|Bank Transfer|2024/01/01|4712/12/31|P||100|1|Bank Transfer|Emirates NBD|MAIN BRANCH|AE|10150625 \*/

/\*New Template METADATA|PersonalPaymentMethod|LegislativeDataGroupName|AssignmentNumber|PersonalPaymentMethodCode|EffectiveStartDate|EffectiveEndDate|PaymentAmountType|Amount|Percentage|OrganizationPaymentMethodCode|BankName|BankBranchName|BankCountryCode|BankAccountNumber

MERGE|PersonalPaymentMethod|AE Legislative Data Group|E116|Bank Transfer|2024/10/01|4712/12/31|P||100|Bank Transfer|Abu Dhabi Commercial Bank|MAIN BRANCH|AE|490030011856932920001

\*/

BusinessOperation = 'MERGE'

FileDiscriminator = 'PersonalPaymentMethod' /\*trim(POSITION2)\*/

LegislativeDataGroupName = trim(POSITION3)

AssignmentNumber = trim(POSITION4)

PersonalPaymentMethodCode = 'Bank Transfer'

EffectiveStartDate = trim(POSITION6)

EffectiveEndDate = '4712/12/31' /\*trim(POSITION7) \*/

PaymentAmountType = 'P'

/\* Amount ='' \*/

Percentage = '100'

/\*ProcessingOrder = '1'

OrganizationPaymentMethodCode = 'Bank Transfer' \*/

BankName = trim(POSITION12)

BankBranchName = trim(POSITION13)

BankCountryCode = 'AE'

BankAccountNumber = trim(POSITION15)

LegislativeDataGroupId = trim(POSITION16)

OrganizationPaymentMethodId = trim(POSITION17)

SourceSystemOwner = trim(POSITION18) /\*'ALDAR' \*/

SourceSystemId = trim(POSITION5)

RETURN BusinessOperation,FileName,FileDiscriminator,LegislativeDataGroupName, AssignmentNumber, PersonalPaymentMethodCode, EffectiveStartDate,EffectiveEndDate,PaymentAmountType /\*,Amount\*/ ,Percentage /\*ProcessingOrder ,OrganizationPaymentMethodCode\*/,BankName,BankBranchName,BankCountryCode,BankAccountNumber,LegislativeDataGroupId,OrganizationPaymentMethodId ,SourceSystemOwner, SourceSystemId

)

ELSE

OUTPUTVALUE='NONE'

RETURN OUTPUTVALUE

/\* End Formula Text \*/

Prerequisite Setup for Enabling Lever to upload file to UCM Server

roles needed for an audit user to call the following webservices: fscmRestApi, fndAuditRESTService, audittrail, getaudithistory

SOLUTION

1. Create a custom role in Security Console with Internal Auditor role plus add the ASE\_REST\_SERVICE\_ACCESS\_IDENTITY\_INTEGRATION\_PRIV privilege.

2. Add this custom role to the user.

so we have added above role to this user and

XXAD\_REST\_API\_UPLOAD\_DOCS\_LEVER

lever\_fusion\_integration\_user

Lever2024#

https://fa-ewnp-test-saasfaprod1.fa.ocs.oraclecloud.com/fscmRestApi/resources/11.13.18.05/erpintegrations

{

"OperationName":"uploadFileToUCM",

"DocumentContent":"VGVzdGluZ19maWxlX3NhbXBsZQ==",

"DocumentAccount":"hcm/common/content",

"ContentType":"zip",

"FileName":"Test.zip",

"DocumentId":null

}

