

Introduction to Databases and SQL – Part 2 Testing your Knowledge

Office of Internal Audit

Kate Head | April 12, 2022



SQL Syntax – Order of Operations

Remember: the database will read the query in the following order:

- **FROM**: Specifies the table to select the columns from
- **WHERE**: Filters the data allowing you to select only some rows
- **GROUP BY**: Summarizes records with the same value(s).
- **SELECT**: Specifies the columns or Select* all the columns.
- **ORDER BY**: Sorts the data based on a particular column(s) use **DESC** or **ASC** to set the sort order.

Understanding ALIAS

Sets an “alias” for a table name or column.

```
SELECT table_name AS alias_name  
FROM table_name
```

```
Example: SELECT PS_JOB AS A  
FROM PS_JOB
```

When reading Query language the first instance of the alias may look like this

A. PS_JOB

Example 1 – PS_JOB_CURR_VW

```
SELECT A.EMPLID
,A.EMPL_RCD
,A.EFFDT
,A.BUSINESS_UNIT
,A.EMPL_STATUS
,A.HR_STATUS
,A.DEPTID
,A.JOBCODE
,A.LOCATION
,A.POSITION_NBR
,A.ACTION
,A.ACTION_REASON
,A.COMP_FREQUENCY
,A.COMPRATE
,A.CURRENCY_CD
,A.SAL_ADMIN_PLAN
,A.GRADE
,A.COMPANY
,A.PAY_SYSTEM_FLG
,A.PAYGROUP
,A.REG_TEMP
,A.FULL_PART_TIME
,A.SETID_DEPT
,A.SETID_JOBCODE
,A.SETID_LOCATION
FROM PS_JOB A
WHERE A.EFFDT = ( SELECT MAX (C.EFFDT) FROM PS_JOB C WHERE C.EMPLID = A.EMPLID AND C.EMPL_RCD = A.EMPL_RCD AND C.EFFDT <=
TO_DATE(TO_CHAR(SYSDATE,'YYYY-MM-DD'),'YYYY-MM-DD'))
AND A.EFFSEQ = (SELECT MAX(D.EFFSEQ) FROM PS_JOB D WHERE D.EMPLID = A.EMPLID AND D.EMPL_RCD = A.EMPL_RCD AND D.EFFDT = A.EFFDT)
AND HR_STATUS = 'A';
```

In this example we are extracting a subset of records from the JOB table which represents one record for each unique EMPLID, EMPL_RCD combination. We want only the latest effective date and effective sequenced record.

We don't want any future dated records.

This will provide us with the job date on the most current job record for each unique EMPLID EMPL_RCD.

Step One: Determine Table & Columns

JOB Table (ALIAS A)

EMPL_RCD

EFFDT

BUSINESS UNIT

EMPLY_STATUS

HR_STATUS

DEPTID

JOBCODE

LOCATION

POSITION_NUMBER

ACTION

Remaining Columns too long to list

FROM

SELECT

This data will be in the out put file which is generated after the **WHERE** Clause is applied.

JOB Table (Alias C)

EMPLID

EMPL_RCD

EFFDT



```
WHERE A.EFFDT = ( SELECT MAX
(C.EFFDT) FROM PS_JOB C WHERE
C.EMPLID = A.EMPLID AND
C.EMPL_RCD = A.EMPL_RCD AND
C.EFFDT <=
TO_DATE(TO_CHAR(SYSDATE,'YYYY-
MM-DD'),'YYYY-MM-DD'))
```

JOB Table (Alias A)

EMPLID

EMPL_RCD

EFF_DT (is set at Maximums using ALIAS C)

EFF_SEQ (is set a Maximum using ALIAS D)

HR_STATUS (must be A)

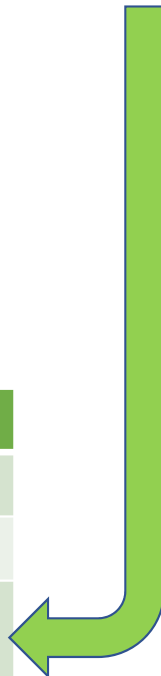
JOB Table (Alias D)

EMPLID

EMPL_RCD

EFFDT (uses the already set maximum effective date in A)

EFF_SEQ



```
AND A.EFFSEQ = (SELECT
MAX(D.EFFSEQ) FROM PS_JOB D
WHERE D.EMPLID = A.EMPLID AND
D.EMPL_RCD = A.EMPL_RCD AND
D.EFFDT = A.EFFDT)
AND HR_STATUS = 'A';
```

Example 2 – U_PAYACTUALS

```
SELECT A.COMPANY, A.PAYGROUP, TO_CHAR(A.PAY_END_DT,'YYYY-MM-DD'), A.PAGE_NUM, A.LINE_NUM, A.OFF_CYCLE, A.SEPCHK, A.EMPLID, A.FIRST_NAME, A.LAST_NAME,
A.EMPL_RCD, A.U_HR_DEPTID, A.POSITION_NBR, A.JOBCODE, A.CA_GL_INTFC_RUN, A.U_PAY_PERIOD, A.REVERSED, A.FISCAL_YEAR, A.ACCOUNTING_PERIOD, A.ACCT_CD,
A.OPERATING_UNIT, A.FUND_CODE, A.ACCOUNT, A.DEPTID, A.PRODUCT, A.CHARTFIELD1, A.PROJECT_ID, A.U_BUD_ENT, A.U_FUND_ID, A.U_SAMAS_ACCOUNT, A.U_DIST_SOURCE,
A.U_DIST_CODE1, A.U_DIST_CODE2, A.U_DIST_CODE3, A.U_DIST_AMOUNT, A.HP_RETDIST_SEQ_NO, A.HP_RETDIST_STATUS, TO_CHAR(A.HP_RETDIST_PAY_DT,'YYYY-MM-DD'),
A.U_PERCENT_DISTRIB, A.SAL_ADMIN_PLAN
FROM PS_U_PAYACTUALS_VW A, PS_EMPLMT_SRCH_QRY A1
WHERE ( A.EMPLID = A1.EMPLID
AND A.EMPL_RCD = A1.EMPL_RCD
AND A1.OPRID = 'KHEAD'
AND ( A.PAY_END_DT BETWEEN TO_DATE(:1,'YYYY-MM-DD') AND TO_DATE(:2,'YYYY-MM-DD')
AND A.PAYGROUP LIKE :3
AND A.EMPLID LIKE :4
AND to_char( A.EMPL_RCD) LIKE :5
AND A.OPERATING_UNIT LIKE :6
AND A.FUND_CODE LIKE :7
AND A.ACCOUNT LIKE :8
AND A.DEPTID LIKE :9
AND A.PRODUCT LIKE :10
AND A.CHARTFIELD1 LIKE :11
AND A.PROJECT_ID LIKE :12
AND A.U_BUD_ENT LIKE :13
AND A.U_FUND_ID LIKE :14
AND A.JOBCODE LIKE :15
AND A.SAL_ADMIN_PLAN LIKE :16
AND A.U_HR_DEPTID LIKE :17 ))
```

In this example we are looking a SQL Code which is used to create a table of paycheck actual amounts.

In this example the user of the query can either reflect they want all the records or a subset of the record.

When user runs the Query they enter the selection criteria

Min Pay Period End Date

Max Pay Period End Date

Pay Group

Empl ID

Empl Record

Operating Unit

Fund Code

Account

FAST-GL DeptID

Product

Chartfield 1

Project/Grant

Budget Entity

Fund Id

Job Code

Salary Administration Plan

GEMS DeptID

% is a a wildcard used in SQL Server. Other version of SQL use different wildcards to represent "all character fields" WHERE *fieldname* LIKE % will pull all as long as the fieldname is a character.

The LIKE command allows the "user input" to be applied. This is like using a variable when scripting in IDEA or ACL.

Step One: Table & Columns

PS_U_PAYACTUALS_VW (ALIAS A)

COMPANY

PAYGROUP

PAY_END_DT

FIRST_NAME

LAST_NAME

EMPL_RCD

U_HR_DEPT_ID

POSITION_NBR

JOBCODE

CA_GL_INTFC_RUN

Remaining Columns to long to list

PS_EMPLMT_SRCH_QRY

EMPLID

EMPL_RCD

OPERID

FROM

SELECT

This data will be in the output file which is generated after the **WHERE** Clause is applied.

FROM **PS_U_PAYACTUALS_VW** A (Table 1 – A Alias),
PS_EMPLMT_SRCH_QRY A1 (Table 2 – A1 Alias)

Step Two: Where Clause

PS_U_PAYACTUALS_VW
(ALIAS A)

EMPLID

EMPL_RCD

OPRID

PS_EMPLMT_SRCH_QRY

EMPLID

EMPL_RCD

OPERID

WHERE A.EMPLID = A1.EMPLID
AND A.EMPL_RCD = A1.EMPL_RCD
AND A1.OPRID = 'KHEAD'

PS_EMPLMT_SRCH_QRY is added automatically to the online SQL. The record involves the department tree to allow users to see data they are allowed to.

Step Three: Identify User Defined Variables (LIKE)

Field /Column	LIKE	User Input	SQL CODE
A.PAY_END_DT	1 2	% or Date begin % or Date end	AND (A.PAY_END_DT BETWEEN TO_DATE(:1,'YYYY-MM-DD') AND TO_DATE(:2,'YYYY-MM-DD')
A. PAYGROUP	3	% or Pay group	AND A.PAYGROUP LIKE :3
A. EMPLID	4	% or specific EMPLID	AND A.EMPLID LIKE :4
To Char (E.EMPL.RD)*	5	% or specific EMPLRCD	AND to_char(A.EMPL_RCD) LIKE :5
A. OPERATING_UNIT	6	% or Operating Unit	AND A.OPERATING_UNIT LIKE :6
A. FUND_CODE	7	% or Fund code	AND A.FUND_CODE LIKE :7
A. ACCOUNT	8	% or Account	AND A.ACCOUNT LIKE :8
A. DEPTID	9	% or Department ID	AND A.DEPTID LIKE :9

*Since this field was numeric it was converted to Character

Variables

Field /Column	LIKE	User Input	SQL CODE
A. PRODUCT	10	% or	AND A.PRODUCT LIKE :10
A. CHARTFIELD1	11	% or initiative (chartfield 1)	AND A.CHARTFIELD1 LIKE :11
A. PROJECT_ID	12	% or Project ID	AND A.PROJECT_ID LIKE :12
A. U_BUD_ENT	13	% or budget entity	AND A.U_BUD_ENT LIKE :13
A. U_FUND_ID	14	% or Fund id	AND A.U_FUND_ID LIKE :14
A. JOBCODE	15	% or job code	AND A.JOBCODE LIKE :15
A. SAL_ADM_PLAN	16	% or salary admin plan	AND A.SAL_ADMIN_PLAN LIKE :16
A. U_HR_DEPTID	17	% or HR department ID	AND A.U_HR_DEPTID LIKE :17

*Since this field was numeric it was converted to Character

Example 3 – U_PAYROLL_POSTING_LEDGER

```

SELECT D.RUN_ID, TO_CHAR(A.PAY_END_DT,'YYYY-MM-DD'), A.U_HR_DEPTID, B.LAST_NAME, B.FIRST_NAME, A.EMPLID, A.EMPL_RCD, A.POSITION_NBR, A.JOBCODE, C.SAL_ADMIN_PLAN, C.PAYGROUP, C.FTE, A.SEPCHK, A.OFF_CYCLE,
A.HP_RETDIST_SEQ_NO, A.HP_RETDIST_STATUS, TO_CHAR(A.HP_RETDIST_PAY_DT,'YYYY-MM-DD'), TO_CHAR(A.HP_CORRECTED_DT,'YYYY-MM-DD'), A.OPERATING_UNIT, A.DEPTID, A.FUND_CODE, A.PRODUCT, A.CHARTFIELD1,
A.PROJECT_ID, MAX(E.U_SAMAS_ACCOUNT), A.U_PERCENT_DISTRIB, C.COMPRATE, A.U_AP_EARNS, A.U_FAC_EARNS, A.U_FAC_ADJ_EARNS, A.U_USPS_EARNS, A.U_OPS_EARNS, A.U_GA_EARNS, A.U_GA_PHD_EARNS,
A.U_POSTDOC_EARNS, A.U_RES_EARNS, A.U_ADMIN_EARNS, A.U_BONUS_EARNS, A.U_CELL_ALLOWANCE, A.U_DATA_ALLOWANCE, A.U_HEALTH_MATCH, A.U_LIFE_MATCH, A.U_RETIRE_MATCH, A.U_FICA_MATCH,
A.U_PRE_TAX_DED_AMT, A.U_LEAVE_ALLOC_AMT, A.U_DISABILITY_AMT, (A.U_AP_EARNS + A.U_ADMIN_EARNS + A.U_BONUS_EARNS + A.U_FAC_EARNS + A.U_OPS_EARNS + A.U_GA_EARNS + A.U_GA_PHD_EARNS +
A.U_FAC_ADJ_EARNS + A.U_POSTDOC_EARNS + A.U_RES_EARNS + A.U_USPS_EARNS + A.U_CELL_ALLOWANCE + A.U_DATA_ALLOWANCE + A.U_HEALTH_MATCH + A.U_LIFE_MATCH + A.U_RETIRE_MATCH + A.U_DISABILITY_AMT +
A.U_PRE_TAX_DED_AMT + A.U_LEAVE_ALLOC_AMT + A.U_FICA_MATCH), D.CA_GL_INTFC_RUN
FROM PS_U_PAYDIST_EDT_V A, PS_PERSONAL_DATA B, PS_PERALL_SEC_QRY B1, PS_JOB C, PS_EMPLMT_SRCH_QRY C1, PS_PAY_CALENDAR D, PS_ACCT_CD_TBL E
WHERE (B.EMPLID = B1.EMPLID
AND B1.OPRID = 'KHEAD'
AND C.EMPLID = C1.EMPLID
AND C.EMPL_RCD = C1.EMPL_RCD
AND C1.OPRID = 'KHEAD'
AND (A.PAY_END_DT BETWEEN TO_DATE(:1,'YYYY-MM-DD') AND TO_DATE(:2,'YYYY-MM-DD')
AND (A.U_HR_DEPTID LIKE :3
AND A.DEPTID LIKE :4)
AND (A.PROJECT_ID LIKE :5
AND A.CHARTFIELD1 LIKE :6
AND A.PRODUCT LIKE :7
AND A.FUND_CODE LIKE :8)
AND A.EMPLID LIKE :9
AND B.EMPLID = A.EMPLID
AND C.EMPLID = A.EMPLID
AND C.EMPL_RCD = A.EMPL_RCD
AND C.EFFDT =
(SELECT MAX(C_ED.EFFDT) FROM PS_JOB C_ED
WHERE C.EMPLID = C_ED.EMPLID
AND C.EMPL_RCD = C_ED.EMPL_RCD
AND C_ED.EFFDT <= A.PAY_END_DT)
AND C.EFFSEQ =
(SELECT MAX(C_ES.EFFSEQ) FROM PS_JOB C_ES
WHERE C.EMPLID = C_ES.EMPLID
AND C.EMPL_RCD = C_ES.EMPL_RCD
AND C.EFFDT = C_ES.EFFDT)
AND A.COMPANY = D.COMPANY
AND A.PAYGROUP = D.PAYGROUP
AND A.PAY_END_DT = D.PAY_END_DT
AND E.OPERATING_UNIT = A.OPERATING_UNIT
AND E.FUND_CODE = A.FUND_CODE
AND E.PRODUCT = A.PRODUCT
AND E.PROJECT_ID = A.PROJECT_ID
AND E.CHARTFIELD1 = A.CHARTFIELD1
AND E.DEPTID_CF = A.DEPTID ))

```



This example is a query used to run the payroll posting ledger. It allows user defined fields which are interpreted in the same manner as the example 2.

This example comes from Peoplesoft Query Manager. Each of the fields are given an Heading Text which is not in this SQL Script Example above due to its length. The can be found in tab Fields in Query Manager

PeopleSoft Query Manager Example

< Records
Query Manager

Records
Query
Expressions
Prompts
Fields
Criteria
Having
View SQL
Run

Query Name **91_U_PAYROLL_POSTING_LEDGER** Description **Must use % in any blank field** Feed ▾

Click folder next to record to show fields. Check fields to add to query. Uncheck fields to remove from query. Add additional records by clicking the records tab. When finished click the fields tab.

Chosen Records

Alias	Record	
A	U_PAYDIST_EDT_V - MergedDistrib for Ern Ded TX	Hierarchy Join ▾
B	PERSONAL_DATA - PERSONAL_DATA for Rptng	Hierarchy Join ▾
C	JOB - EE Job History	Hierarchy Join ▾
D	PAY_CALENDAR - Pay Calendar	Hierarchy Join ▾
E	ACCT_CD_TBL - Account Codes	Hierarchy Join ▾

Expand All Records
Collapse All Records

Save
[Save As](#)
[New Query](#)
[Preferences](#)
[Properties](#)

[Publish as Feed](#)
[Publish as Pivot Grid](#)
[New Union](#)

Return To Search

People Soft Query Manager Example

New Window | Help

Records Query Expressions Prompts **Fields** Criteria Having View SQL Run

Query Name U_PAYROLL_POSTING_LEDGER Description Must use % in any blank field Feed

/ew field properties, or use field as criteria in query statement. Reorder / Sort

ields

Q 1-49 of 49 View All

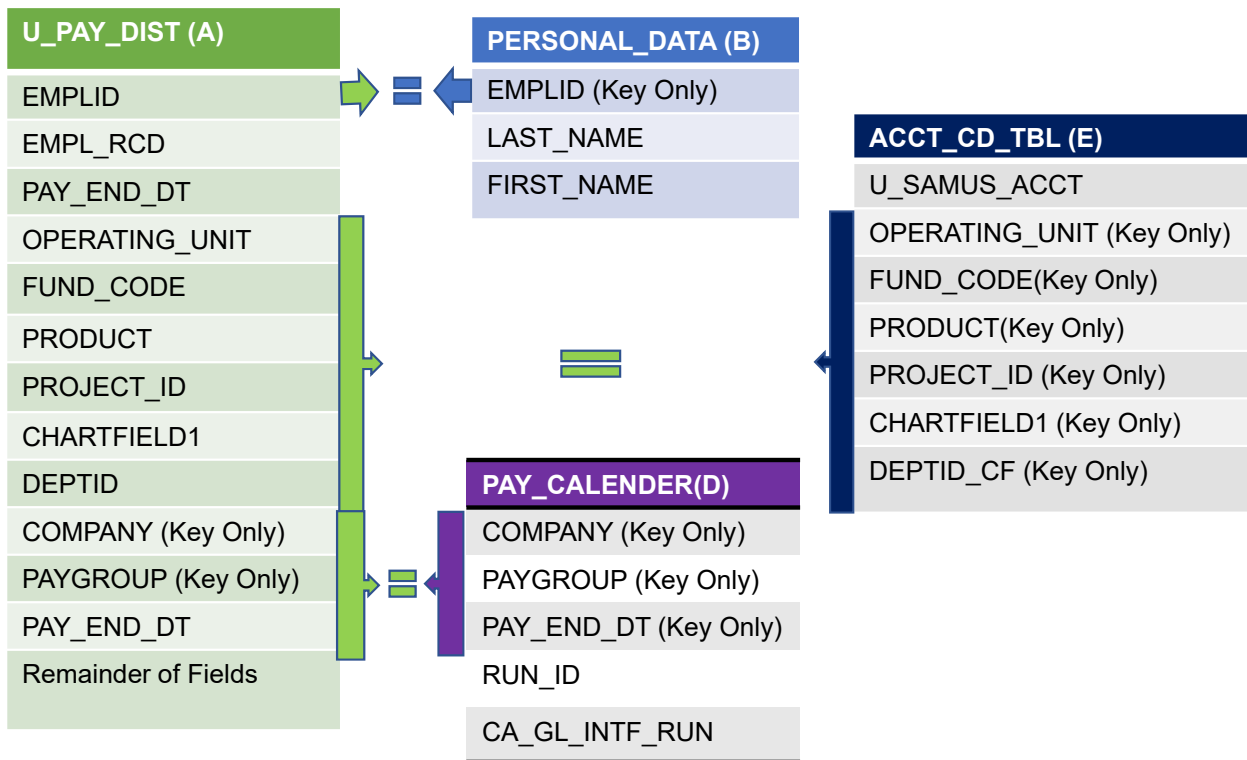
Col	Record.Fieldname	Format	Ord	XLAT	Agg	Heading Text	Add Criteria	Edit	Delete
1	D.RUN_ID - Pay Run ID	Char10				Pay Period No.		Edit	—
2	A.PAY_END_DT - Pay Period End Date	Date	1			Pay Period End Date		Edit	—
3	A.U_HR_DEPTID - HR Department	Char10	2			HR Department		Edit	—
4	B.LAST_NAME - Last Name	Char30	3			Last Name		Edit	—
5	B.FIRST_NAME - First Name	Char30	4			First Name		Edit	—
6	A.EMPLID - Empl ID	Char11				ID		Edit	—
7	A.EMPL_RCD - Empl Record	Num3 0	5			Empl Record		Edit	—
8	A.POSITION_NBR - Position Number	Char8				Position		Edit	—
9	A.JOBCODE - Job Code	Char6				Job Code		Edit	—
10	C.SAL_ADMIN_PLAN - Salary Administration Plan	Char4				Sal Plan		Edit	—
11	C.PAYGROUP - Pay Group	Char3				Pay Group on Job		Edit	—
12	C.FTE - FTE	Num2 6				FTE		Edit	—

Step One: Table and their Primary Keys

Alias	Table	Primary Keys*
A	U_PAYDIST_EDT	COMPANY, PAYGROUP, PAY_END_DT, OFF_CYCLE, PAGE_NUM, LINE_NUM, SEPCHK, EMPLID, EMPLRCD
B	PERSONAL_DATA	EMPLID
B1	PS_PERALL_SEC_QR	Used for security purposes
C	JOB	EMPLID, EMPLRCD, EFFDT, EFFSEQ
C1	PS_EMPLMT_SRCH_QRY	Used for security purposes
D	PAY_CALENDAR	COMPANY, PAYGROUP, PAY_END_DT
E	ACCT_CD_TBL	FDM_HASH

* Keys are used to join tables or in WHERE clauses to select records

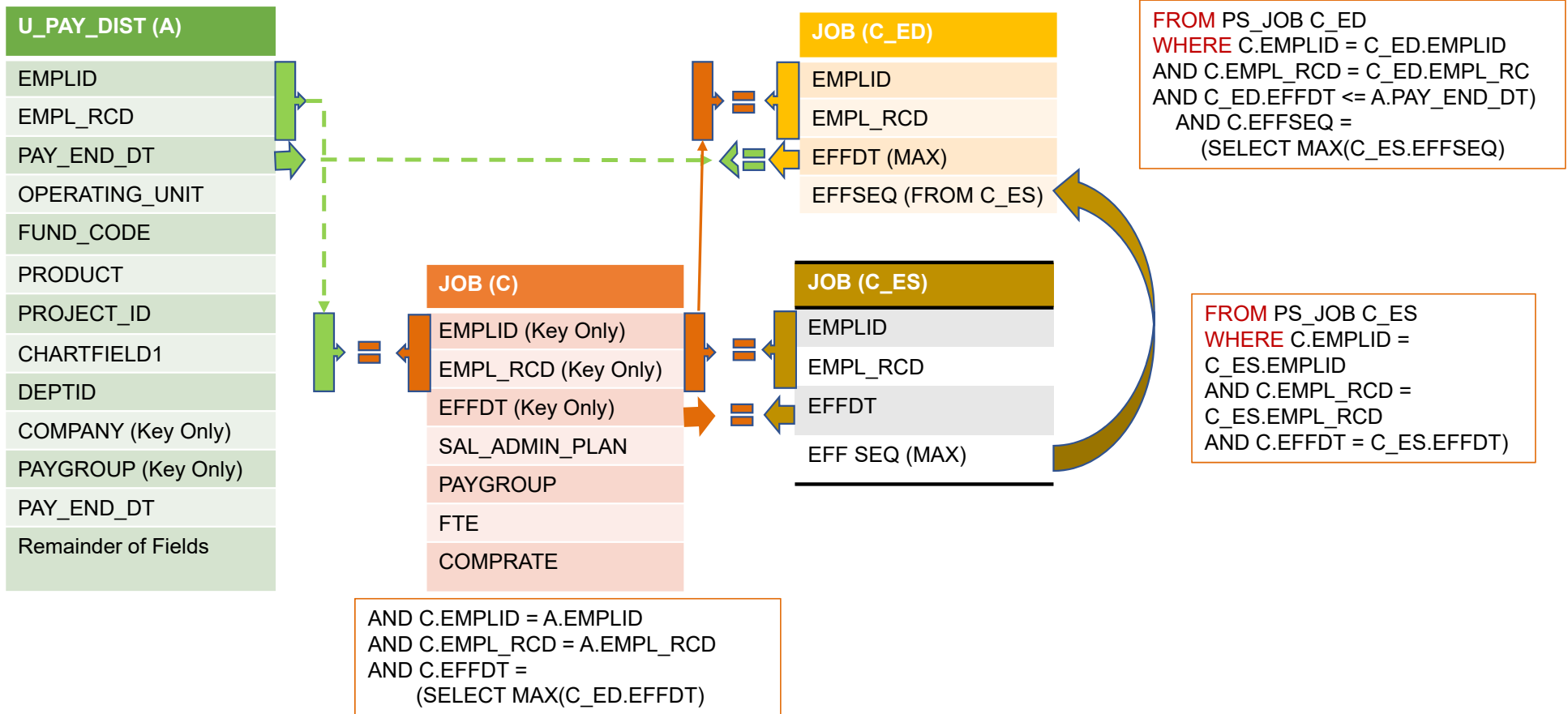
Step One: Tables and Columns



For connection to JOB table see next page

JOB (C)
EMPLID (Key Only)
EMPL_RCD (Key Only)
EFF_DT (Key Only)
SAL_ADMIN_PLAN
PAYGROUP
FTE
COMPRATE

Step One: Tables and Columns



Example 3 – U_PAYROLL_POSTING_LEDGER (Second half of Script)

GROUP BY D.RUN_ID, A.PAY_END_DT, A.U_HR_DEPTID, B.LAST_NAME, B.FIRST_NAME, A.EMPLID,
A.EMPL_RCD, A.POSITION_NBR, A.JOBCODE, C.SAL_ADMIN_PLAN, C.PAYGROUP, C.FTE, A.SEPCHK,
A.OFF_CYCLE, A.HP_RETDIST_SEQ_NO, A.HP_RETDIST_STATUS, A.HP_RETDIST_PAY_DT,
A.HP_CORRECTED_DT, A.OPERATING_UNIT, A.DEPTID, A.FUND_CODE, A.PRODUCT, A.CHARTFIELD1,
A.PROJECT_ID, A.U_PERCENT_DISTRIB, C.COMPRATE, A.U_AP_EARNS, A.U_FAC_EARNS,
A.U_FAC_ADJ_EARNS, A.U_USPS_EARNS, A.U_OPS_EARNS, A.U_GA_EARNS, A.U_GA_PHD_EARNS,
A.U_POSTDOC_EARNS, A.U_RES_EARNS, A.U_ADMIN_EARNS, A.U_BONUS_EARNS, A.U_CELL_ALLOWANCE,
A.U_DATA_ALLOWANCE, A.U_HEALTH_MATCH, A.U_LIFE_MATCH, A.U_RETIRE_MATCH, A.U_FICA_MATCH,
A.U_PRE_TAX_DED_AMT, A.U_LEAVE_ALLOC_AMT, A.U_DISABILITY_AMT, (**A.U_AP_EARNS +
A.U_ADMIN_EARNS + A.U_BONUS_EARNS + A.U_FAC_EARNS + A.U_OPS_EARNS + A.U_GA_EARNS +
A.U_GA_PHD_EARNS + A.U_FAC_ADJ_EARNS + A.U_POSTDOC_EARNS + A.U_RES_EARNS +
A.U_USPS_EARNS + A.U_CELL_ALLOWANCE + A.U_DATA_ALLOWANCE + A.U_HEALTH_MATCH +
A.U_LIFE_MATCH + A.U_RETIRE_MATCH + A.U_DISABILITY_AMT + A.U_PRE_TAX_DED_AMT +
A.U_LEAVE_ALLOC_AMT + A.U_FICA_MATCH**), D.CA_GL_INTFC_RUN
ORDER BY 2, 3, 4, 5, 7, 15, 19, 20, 21, 22, 23, 24

This is one single field which accumulates all these fields into a Total Amount

ORDER BY

ORDERS the columns
based on the order in
the SELECT statement

#	Column
2	A.PAY_END_DT
3	A.U_HR_DEPTID
4	B.LAST_NAME,
5	B.FIRST_NAME,
7	A.EMPL_RCD
15	A.HP_RETDIST_SEQ_NO
19	A.OPERATING_UNIT,
20	A.DEPTID
21	A.FUND_CODE
22	A.PRODUCT
23	A. CHARTFIELD1 (INITIATIVE)
24	A. PROJECT ID