



UCOP Data Users Group

June 20 & July 18, 2018

Exercise Goal 1

- ▶ **(ENROLL and UAD: Student Admissions Data)**
 - ❑ Write a SQL query to join enrollment and UAD to count the number of enrolled undergraduates in Fall 2016 by campus, citizenship status(2cat) and parent education level.

Execution steps to join Enrollment and UAD

- ▶ Build query from enrollment star using enrollment fact and associated dimensions from STUD_BI (QUERY1).
- ▶ Build query from UAD star using application fact and associated dimensions from UAD_BI (QUERY2).
- ▶ Join Query1 to query 2 using cross schema join (LEFT OUTER JOIN) using campus+studentid key (CMP_STUD_ID).
- ▶ Group the data set based on the required elements

Syntax to join Enrollment and UAD

```
SELECT
ENROLL_STAR.ACAD_YR, ENROLL_STAR.CAMPUS_NAME,
UAD_STAR.CITIZENSHIP2CAT,
UAD_STAR.PARENT1_EDUCATION_LVL_DESC,
COUNT(*)
FROM
(QUERY1) as ENROLL_STAR
LEFT OUTER JOIN
(QUERY2) as UAD_STAR
ON ENROLL_STAR.CMP_STUD_ID = UAD_STAR.CMP_STUD_ID
GROUP BY
ENROLL_STAR.ACAD_YR, ENROLL_STAR.CAMPUS_NAME,
UAD_STAR.CITIZENSHIP2CAT,
UAD_STAR.PARENT1_EDUCATION_LVL_DESC
```

Execution steps to write a query

- ▶ Find the required fact and dimension tables
- ▶ Find the joining surrogate key columns
- ▶ Find the fact and dimension non key columns required in the query
- ▶ Build the query by joining the fact table with all the required dimensions using surrogate keys.
- ▶ Add the filter conditions on the dimension columns based on the required values.
- ▶ Finally group the data set based on the mentioned columns and generate the aggregated result set.

Query 1: Enrollment star Query

▶ Rules:

- Academic Year = 2016
- Term Name = Fall
- Campus name = LOS ANGELES
- Record Type = 3WK
- Major Component Number = 1 (select only primary major record)
- Student Level Code = 1, 2, 3, 4, L, S (Only Undergraduates)
- Create Campus Student ID:
- `CMP_STUD_ID = campus code||student id (student_d).`

Query 1 – Technical Requirements

- ▶ **Schema:**
 - STUD_BI
- ▶ **Table Name:**
 - ENROLLMENT_F
 - ACADEMIC_SUB_TERM_D
 - STUDENT_D
 - CAMPUS_LOCATION_D
 - STUDENT_LEVEL_D
- ▶ **Column Names:**
 - Enrollment Fact: major component number
 - Sub Term dimension: Academic year, Term name and Record type
 - Campus Dimension: Campus long description
 - Student Dimension: Student ID and Campus Code
 - Student Level Dimension: Student Level Code

Query 2: UAD star Query

▶ Rules:

➤ Create Campus Student ID:

- `CMP_STUD_ID = registration campus code || registration id (applicant_d)`

Query 2 – Technical Requirements

▶ Schema:

- UAD_BI

▶ Table Name:

- APPLICATION_F
- APPLICANT_D
- TERM_D
- PARENT_EDUCATION_RESIDENCY_D

▶ Column Names:

- Applicant dimension: Registered campus code, Registration ID, Citizenship Status 2CAT
- Parent education dimension : Parent 1 education level code description

Expected Result Set

ACAD_YR	CMP_LOC_LNG_DESC	AL_CTZN_STAT_2CAT_DESC	PAR1_EDU_LVL_CD_DESC	COUNT
2016	LOS ANGELES	Domestic	Four-Year College/University Graduate	6513
2016	LOS ANGELES	Domestic	High School Graduate	3169
2016	LOS ANGELES	Domestic	No High School	2328
2016	LOS ANGELES	Domestic	No Selection	1878
2016	LOS ANGELES	Domestic	Postgraduate Study	7715
2016	LOS ANGELES	Domestic	Some College/University	2491
2016	LOS ANGELES	Domestic	Some High School	1664
2016	LOS ANGELES	Domestic	Two-Year College/University Graduate	928
2016	LOS ANGELES	Foreign	Four-Year College/University Graduate	1525
2016	LOS ANGELES	Foreign	High School Graduate	397
2016	LOS ANGELES	Foreign	No High School	179
2016	LOS ANGELES	Foreign	No Selection	224
2016	LOS ANGELES	Foreign	Postgraduate Study	1244
2016	LOS ANGELES	Foreign	Some College/University	224
2016	LOS ANGELES	Foreign	Some High School	171
2016	LOS ANGELES	Foreign	Two-Year College/University Graduate	150
2016	LOS ANGELES	(null)	(null)	74

SQL Query

--07/18/2018 UAD and enrollment

```

SELECT
ENROLL_STAR.ACAD_SUB_T ACAD_YR,
ENROLL_STAR.CMP_LOC_LOC1_LNG_DESC,
UAD_STAR.AL_CTZN_STAT_ADM_2CAT_DESC,
UAD_STAR.PAR_EDU_RSDNCY_PAR1_EDU_LVL_CD_DESC,
count(ENROLL_STAR.CMP_STUD_ID)
FROM
(
SELECT *
FROM (
SELECT
s_d.STUD_LOC_CMP_CD||s_d.STUD_ID as CMP_STUD_ID, c1_d.CMP_LOC_LOC1_LNG_DESC,
ast_d.ACAD_SUB_T ACAD_YR, ast_d.ACAD_SUB_T_REC_TY, ast_d.ACAD_SUB_T_NAM, s1_d.STUD_LVL_CD
FROM STUD_BI.ENROLLMENT_F e_f
INNER JOIN STUD_BI.ACADEMIC_SUB_TERM_D ast_d ON e_f.ACAD_SUB_T_KEY = ast_d.ACAD_SUB_T_KEY
INNER JOIN STUD_BI.STUDENT_D s_d ON e_f.STUD_KEY = s_d.STUD_KEY
INNER JOIN STUD_BI.CAMPUS_LOCATION_D c1_d ON c1_d.CMP_LOC_LOC1_CD = s_d.STUD_LOC_CMP_CD
INNER JOIN STUD_BI.STUDENT_LEVEL_D s1_d ON e_f.STUD_LVL_KEY = s1_d.STUD_LVL_KEY
WHERE ast_d.ACAD_SUB_T ACAD_YR in (2016)
AND ast_d.ACAD_SUB_T_REC_TY = '3WK'
AND c1_d.CMP_LOC_LOC1_LNG_DESC IN( 'LOS ANGELES')
AND ast_d.ACAD_SUB_T_NAM = 'Fall'
AND e_f.ENRL_MAJ_CMPNT_NUM = 1
AND s1_d.STUD_LVL_CD in ('1', '2', '3', '4', 'L', 'S')
)
ORDER BY CMP_STUD_ID, ACAD_SUB_T ACAD_YR, ACAD_SUB_T_NAM
) AS ENROLL_STAR
LEFT OUTER JOIN --Left Outer join gives all the records from first query and all the matching records from the second query based on the join condition
(
SELECT
a_f.APPL_ID, a_d.AL_REGD_CMP_CD||a_d.AL_REGD_ID as CMP_STUD_ID, t_d.T ACAD_YR, t_d.T_NAM,
a_d.AL_CTZN_STAT_ADM_2CAT_DESC, per_d.PAR_EDU_RSDNCY_PAR1_EDU_LVL_CD_DESC
FROM UAD_BI.APPLICATION_F a_f
INNER JOIN UAD_BI.APPLICANT_D a_d ON a_f.AL_KEY = a_d.AL_KEY
INNER JOIN UAD_BI.TERM_D t_d ON a_f.T_KEY = t_d.T_KEY
INNER JOIN UAD_BI.PARENT_EDUCATION_RESIDENCY_D per_d ON a_f.PAR_EDU_RSDNCY_KEY = per_d.PAR_EDU_RSDNCY_KEY
ORDER BY t_d.T ACAD_YR DESC, t_d.T_NAM DESC
) AS UAD_STAR
ON ENROLL_STAR.CMP_STUD_ID = UAD_STAR.CMP_STUD_ID
group by ENROLL_STAR.ACAD_SUB_T ACAD_YR, ENROLL_STAR.CMP_LOC_LOC1_LNG_DESC, UAD_STAR.AL_CTZN_STAT_ADM_2CAT_DESC, UAD_STAR.PAR_EDU_RSDNCY_PAR1_EDU_LVL_CD_DESC
order by ENROLL_STAR.ACAD_SUB_T ACAD_YR, ENROLL_STAR.CMP_LOC_LOC1_LNG_DESC, UAD_STAR.AL_CTZN_STAT_ADM_2CAT_DESC, UAD_STAR.PAR_EDU_RSDNCY_PAR1_EDU_LVL_CD_DESC

```