

# UCOP Data Users Group March 9th, 2020

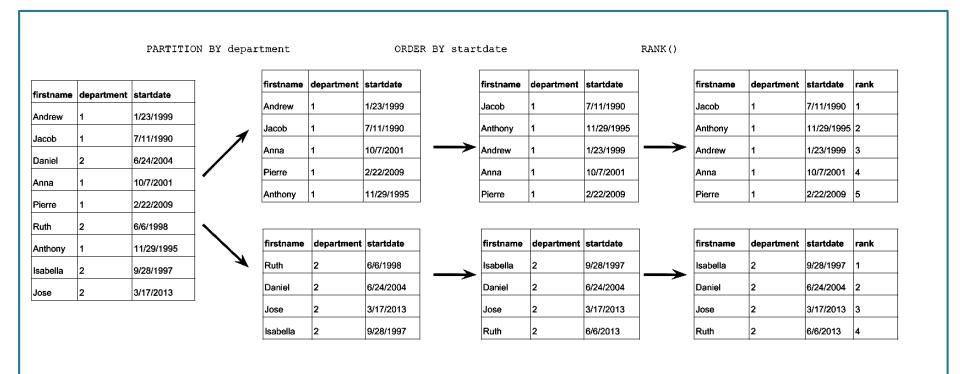


### **OLAP Functions**

- OLAP (Online Analytical Processing) functions allow you to flexibly create subgroups in your query by partition the data sets.
- It can be used in the SELECT statements in SQL to effectively perform arithmetic, analytical, ranking functions, find lead and lag values, cumulative values by subgroups of the entire data set.
- It simplifies the SQL query significantly.



# **OLAP Function diagram**





# Syntax

SELECT Function({c1}) OVER (PARTITION BY c2,c3,... {ORDER BY c4, c5...}) AS Oc1, c2, c3, c6, c7,... FROM T1

• T1 has columns c1, c2, c3, c4, c5, c6, c7.

- OLAP Functions:
- ROW\_NUMBER, RANK, DENSE\_RANK
- FIRST\_VALUE, LAST\_VALUE, NTH\_VALUE
- SUM, AVG, MAX, MIN, COUNT
- LEAD, LAG



# Exercise Goal – Enrollment Star

- Write a SQL query to select the following
- 1. EOT if available or 3WK other wise
- 2. Find the earliest, latest year and term enrolled
- 3. Cumulative total number of fall terms enrolled
- 4. Set flag to 1 whenever the student level is changed, else set to 0



### **OLAP Functions used**

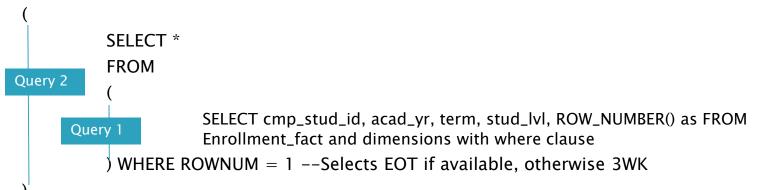
- 1. EOT if available or 3WK other wise ROW\_NUMBER()
- 2. Find the earliest, latest year and term enrolled FIRST\_VALUE()



## **Technical Requirements**

### Query format:

#### SELECT \*, F\_ACAD\_YEAR, F\_TERM\_CD, L\_ACAD\_YEAR, L\_TERM\_CD FROM





### **Execution steps**

- **Base Query:**
- Write query from enrollment star using enrollment fact and associated dimensions.
- Create a column CMP\_STUD\_ID (campus code||student id)

### • QUERY1:

- Use ROW\_NUMBER() partition by CMP\_STUD\_ID, ACAD\_YR, TERM order by RECORD\_TYPE desc
- □ Filter the rows where ROW\_NUMBER = 1

### • QUERY2:

- Use FIRST\_VALUE() partition by CMP\_STUD\_ID order by ACAD\_YR, TERM asc to determine the earliest year/term enrolled & order by ACAD\_YR desc, TERM desc to determine the latest year/term enrolled
- Create columns F\_ACAD\_YR & F\_TERM for earliest year/term enrolled
- Create columns L\_ACAD\_YR & L\_TERM for latest year/term enrolled



## **Technical Requirements**

- Schema:
  - STUD\_BI
- Table Name:
  - ENROLLMENT\_F
  - ACADEMIC\_SUB\_TERM\_D
  - STUDENT\_D
  - STUDENT\_LEVEL\_D
- Column Names:
  - Enrollment fact: Enrollment major component number
  - Academic sub term dimension: Academic year, term, record type
  - Student dimension: Student ID
  - Campus Location dimension: Campus code
  - Student Level dimension: Student level code
- Row Filter Rules for the Base Query:
  - Major component number = 1
  - Student level code in ('5', '6','7', '8')
  - Stud ID in ('10000001') [Stud id here is a random number used for example purposes, it is not a real stud id]



### **Technical Requirements**

- ▶ BASE QUERY & QUERY 1 (*ROWNUM*)
  - ROW\_NUMBER() OVER (PARTITION BY *cmp\_stud\_id*, *year*, *term* order by *rec\_type* desc)
- QUERY2 (F\_ACAD\_YEAR, F\_TERM\_CD)
  - FIRST\_VALUE(*year*) OVER (PARTITION BY *cmp\_stud\_id* order by *year* asc, *term* asc)
  - FIRST\_VALUE(*term*) OVER (PARTITION BY *cmp\_stud\_id* order by *year* asc, *term* asc)
- QUERY2 (L\_ACAD\_YEAR, L\_TERM\_CD)
  - FIRST\_VALUE(*year*) OVER (PARTITION BY *cmp\_stud\_id* order by *year* desc, *term* desc)
  - FIRST\_VALUE(*term*) OVER (PARTITION BY *cmp\_stud\_id* order by *year* desc, *term* desc)



### **Expected results – Query**

```
SELECT A CMP STUD ID, A ACAD SUB T REC TY, A ACAD YEAR, A TERM CD, A STUD LVL CD, F ACAD YEAR, F TERM CD, L ACAD YEAR, L TERM CD
 FROM
SELECT *,
     --select earliest yr /term for every degree enrolled
     FIRST VALUE (A ACAD YEAR) OVER (PARTITION BY A CMP STUD ID order by A ACAD YEAR asc, A TERM CD asc) as F ACAD YEAR,
     --select earliest yr /term for every degree enrolled
     FIRST VALUE (A TERM CD) OVER (PARTITION BY A CMP STUD ID order by A ACAD YEAR asc, A TERM CD asc) as F TERM CD,
     --select latest yr /term for every degree enrolled
     FIRST VALUE (A ACAD YEAR) OVER (PARTITION BY A CMP STUD ID order by A ACAD YEAR desc, A TERM CD desc) as L ACAD YEAR,
     --select latest yr /term for every degree enrolled
     FIRST VALUE (A TERM CD) OVER (PARTITION BY A CMP STUD ID order by A ACAD YEAR desc, A TERM CD desc) as L TERM CD
     FROM
     (
         SELECT * FROM
         ( --Select EOT if available, otherwise select 3WK record type
             SELECT *, ROW NUMBER() OVER (PARTITION BY A CMP STUD ID, A ACAD YEAR, A TERM CD order by A ACAD SUB T REC TY desc) as ROWNUM
             FROM
             (
                 SELECT
                 ast d.ACAD SUB T REC TY as A ACAD SUB T REC TY, ast d.ACAD SUB T ACAD YR as A ACAD YEAR, ast d.ACAD SUB T CD as A TERM CD,
                 TRIM(cl d.CMP LOC LOC1 CD||s d.STUD ID) as A CMP STUD ID, s d.STUD LOC CMP CD as A STUD LOC CMP CD,
                 TRIM(s d.STUD ID) as A STUD ID, sl d.STUD LVL CD as A 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 cl d
                                                                         ON e f.CMP LOC KEY = cl d.CMP LOC KEY
                 INNER JOIN STUD BI.STUDENT LEVEL D sl d
                                                                        ON e f.STUD LVL KEY = sl d.STUD LVL KEY
                 WHERE
                 --Select only the graduate degree students.
                 sl d.STUD LVL CD in ('5', '6', '7', '8', 'P')
                 -- Select only the primary major record. This will ensure the grain of the query is one row per student, year, and term.
                 AND e f.ENRL MAJ CMPNT NUM = 1
                 --Stud id here is a random number used for example purposes, it is not a real stud id
                 AND s d.STUD ID IN ('10000001')
             )
         ) WHERE ROWNUM = 1
```



### **Expected results – Output**

A_CMP_STUD_ID	A_ACAD_SUB_T_REC_TY	A_ACAD_YEAR	A_TERM_CD	A_STUD_LVL_CD	F_ACAD_YEAR	F_TERM_CD	L_ACAD_YEAR	L_TERM_CD
10000001	EOT	2006 3		5	2006 3		2012 3	
10000001	EOT	2006 4		5	2006 3		2012 3	
1000001	EOT	2007 2		5	2006 3		2012 3	
1000001	EOT	2007 3		5	2006 3		2012 3	
10000001	EOT	2007 4		5	2006 3		2012 3	
1000001	EOT	2008 2		5	2006 3		2012 3	
10000001	EOT	2008 3		6	2006 3		2012 3	
1000001	EOT	2008 4		6	2006 3		2012 3	
10000001	EOT	2009 2		6	2006 3		2012 3	
1000001	EOT	2009 3		6	2006 3		2012 3	
10000001	EOT	2009 4		6	2006 3		2012 3	
1000001	EOT	2010 2		6	2006 3		2012 3	
10000001	EOT	2010 3		7	2006 3		2012 3	
1000001	EOT	2010 4		7	2006 3		2012 3	
10000001	EOT	201	12	7	20	063	201	.2.3
10000001	EOT	201	13	7	20	063	201	.2.3
1000001	EOT	201	14	7	20	063	201	.23
1000001	EOT	2012 2		7	2006 3		2012 3	
10000001	EOT	201	23	7	20	06 3	201	23

Stud id here is a random number used for example purposes, it is not a real stud id.

- Query picks EOT is available, else 3WK is selected.
- A\_ACAD\_YR & A\_TERM\_CD is the year/term enrolled.
- F\_ACAD\_YR & F\_TERM\_CD is the earliest year/term enrolled.
- L\_ACAD\_YR & L\_TERM\_CD is the latest year/term enrolled.
- The grain of the query is one record per student, year and term enrolled.