



# UCOP Data Users Group

August 15, 2018

## Exercise Goal

### ▶ **Financial Aid Star:**

- ❑ Write a SQL query to find the top three funds awarded and their average paid amount to most students through scholarship, grant, and loan awards in academic year 2014 and 2015.

# 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.
- ▶ Group the data set based on the mentioned columns and generate the aggregated result set.
- ▶ Rank the result set based on the required attributes.

# Query – Technical Requirements

- ▶ **Schema:**
  - STUD\_BI
- ▶ **Table Name:**
  - STUDENT\_FINANCIAL\_AID\_F
  - STUDENT\_D
  - FILE\_CYCLE\_D
  - AWARD\_D
  - AWARD\_REFERENCE\_D
  - FUND\_D
- ▶ **Column Names:**
  - Financial Aid fact: Paid to date amount
  - File cycle dimension: Academic year, and Final file flag
  - Student dimension: Student ID and Campus Code
  - Award dimension: Award type code and description
  - Award reference dimension: Award acceptance code
  - Fund dimension: Fund title

# Query 1: Enrollment star Query

## ▶ Rules:

- ▶ Award Type Code = '1', '2', '3'
- ▶ Award reference acceptance code = 'A'
- ▶ File cycle Final File = 'Y'
- ▶ File Cycle Academic Year = 2014, 2015
- ▶ Create Campus Student ID:
  - `CMP_STUD_ID = campus code||student id (student_d).`

# Query – Building steps

- ▶ Join the financial aid fact to its associated dimensions and grab the required attributes.
- ▶ Add filter on the required dimension attributes.
- ▶ Calculate the following aggregate columns grouped by academic year, award type description, and fund title.
  - Average of the paid to date amount(two points decimal)
  - Count of distinct CMP\_STUD\_ID.
- ▶ Rank the data set by academic year and award type description in the descending order of the student count.
- ▶ Generate the final result set displaying only the top three ranked rows for every academic year and award type description.

# Cumulative columns

- ▶ Average of paid amount:
  - Use function AVG to calculate average
  - Use function DECIMAL to format the average value to two decimal points (2 digits after the decimal point)
  - Syntax: `DECIMAL(AVG(paid_amount), m, n)`
  - m is the number of digits before decimal point and n is the number of digits after decimal point
- ▶ Count of Distinct studentIDs:
  - Distinct count is necessary to avoid duplicate count of student ids for multiple accounts.
  - Example: `COUNT(DISTINCT cmp_stud_id)`

# Ranking Methods

- ▶ Types of Ranking Methods:
  - ROW\_NUMBER()
  - RANK()
  - DENSE\_RANK()



# Ranking Methods – ROW\_NUMBER()

- ▶ Syntax:

**ROW\_NUMBER() OVER (PARTITION BY year, award\_type  
ORDER BY student\_count desc) AS ROWNUM1**

Academic_Year	Award_Type_Desc	Fund_Title	Student_Count	ROWNUM1
2014	Grant	education fee	5000	1
2014	Grant	Tution	4000	2
2014	Grant	Pell Grant	4000	3
2014	Grant	Student Tution	3000	4
2014	Loan	Pell Grant	6000	1
2014	Loan	education fee	5000	2
2014	Loan	Cal Grant	4000	3

# Ranking Methods – RANK()

- ▶ Syntax:

**RANK() OVER (PARTITION BY year, award\_type ORDER BY student\_count desc) AS RANK1**

Academic Year	Award Type Desc	Fund Title	Student Count	RANK1
2014	Grant	education fee	5000	1
2014	Grant	Tution	4000	2
2014	Grant	Pell Grant	4000	2
2014	Grant	Student Tution	3000	4
2014	Loan	Pell Grant	6000	1
2014	Loan	education fee	5000	2
2014	Loan	Cal Grant	4000	3

# Ranking Methods – DENSE\_RANK()

- ▶ Syntax:

**DENSE\_RANK() OVER (PARTITION BY year, award\_type  
ORDER BY student\_count desc) AS DENSE\_RANK1**

Academic Year	Award Type Desc	Fund Title	Student Count	DENSE_RANK1
2014	Grant	education fee	5000	1
2014	Grant	Tution	4000	2
2014	Grant	Pell Grant	4000	2
2014	Grant	Student Tution	3000	3
2014	Loan	Pell Grant	6000	1
2014	Loan	education fee	5000	2
2014	Loan	Cal Grant	4000	3

# Expected Result Set

FILE_CYCLE_ACAD_YR	AWRD_TY_DESC	FD_FD_TTL	AVG_PAID_AMT	COUNT_STUDENTS
2014 Grant		TUITION	6622.45	46455
2014 Grant		UNIVERSITY TUITION INCOME	7149.14	14761
2014 Grant		Unknown Fund	9133.93	28682
2014 Loan		CURRENT FUNDS BAL SHEET CONTROL	16100.79	15619
2014 Loan		FEDERAL STUDENT LOAN FUNDS	3775.87	10538
2014 Loan		Unknown Fund	1039.55	13810
2014 Scholarship/Fellowship/Traineeship		GENERAL FUNDS	4889.76	5232
2014 Scholarship/Fellowship/Traineeship		TUITION	8854.27	9183
2014 Scholarship/Fellowship/Traineeship		Unknown Fund	3136.94	7447
2015 Grant		TUITION	6715.98	50524
2015 Grant		UNIVERSITY TUITION INCOME	7507.00	14435
2015 Grant		Unknown Fund	11832.43	14806
2015 Loan		CURRENT FUNDS BAL SHEET CONTROL	16223.17	15476
2015 Loan		FEDERAL STUDENT LOAN FUNDS	3984.93	10015
2015 Loan		Unknown Fund	789.68	13052
2015 Scholarship/Fellowship/Traineeship		GENERAL FUNDS	4898.44	5528
2015 Scholarship/Fellowship/Traineeship		TUITION	10229.01	7782
2015 Scholarship/Fellowship/Traineeship		Unknown Fund	3155.06	5834

# Query

```

/*Write a SQL query to find the top three funds awarded and their average paid amount to most students through
scholarship, grant, and loan awards in academic year 2014 and 2015*/
SELECT FILE_CYCLE_ACAD_YR, AWRD_TY_DESC, FD_FD_TTL, AVG_PAID_AMT, COUNT_STUDENTS
FROM
(
SELECT FILE_CYCLE_ACAD_YR, AWRD_TY_DESC, FD_FD_TTL, AVG_PAID_AMT, COUNT_STUDENTS,
DENSE_RANK() OVER(PARTITION BY FILE_CYCLE_ACAD_YR, AWRD_TY_DESC order by COUNT_STUDENTS desc) as ROWNUM1
FROM
(
SELECT FILE_CYCLE_ACAD_YR, AWRD_TY_DESC, FD_FD_TTL, count(CMP_STUD_ID) as COUNT_STUDENTS,
DECIMAL(AVG(TOTAL_AMT), 10,2) as AVG_PAID_AMT
FROM (
SELECT CMP_STUD_ID, FILE_CYCLE_ACAD_YR, FD_FD_TTL, AWRD_TY_DESC, SUM(STUD_FINL_AID_AWRD_PD_TO_DT_AMT) AS TOTAL_AMT
FROM
(
SELECT
s_d.STUD_LOC_CMP_CD||s_d.STUD_ID as "CMP_STUD_ID",
fc_d.FILE_CYCLE_MTH_SHRT_NAM, fc_d.FILE_CYCLE_DPNDT_ON_REC_TY, fc_d.FILE_CYCLE_ACAD_YR,
aw_d.AWRD_CD, aw_d.AWRD_TY_DESC,
fai_f.STUD_FINL_AID_AWRD_PD_TO_DT_AMT,
f_d.FD_FD_TTL
FROM STUD_BI.STUDENT_FINANCIAL_AID_F fai_f
INNER JOIN STUD_BI.STUDENT_D s_d ON fai_f.STUD_KEY = s_d.STUD_KEY
INNER JOIN STUD_BI.CAMPUS_LOCATION_D cl_d ON fai_f.CMP_LOC_KEY = cl_d.CMP_LOC_KEY
INNER JOIN STUD_BI.FILE_CYCLE_D fc_d ON fai_f.FILE_CYCLE_KEY = fc_d.FILE_CYCLE_KEY
INNER JOIN STUD_BI.AWARD_D aw_d ON fai_f.AWRD_KEY = aw_d.AWRD_KEY
INNER JOIN STUD_BI.AWARD_REFERENCE_D awr_d ON fai_f.AWRD_REF_KEY = awr_d.AWRD_REF_KEY
INNER JOIN STUD_BI.FUND_D f_d ON fai_f.FD_KEY = f_d.FD_KEY
WHERE
aw_d.AWRD_TY_CD in('1', '2', '3')
AND awr_d.AWRD_REF_AWRD_ACPTC_CD = 'A'
AND fc_d.FILE_CYCLE_FN_FILE_FL = 'Y'
AND fc_d.FILE_CYCLE_ACAD_YR IN (2014, 2015)
)
)
GROUP BY CMP_STUD_ID, FILE_CYCLE_ACAD_YR, FD_FD_TTL, AWRD_TY_DESC
)
GROUP by FILE_CYCLE_ACAD_YR, AWRD_TY_DESC, FD_FD_TTL
)
)
WHERE ROWNUM1 < 4
ORDER BY FILE_CYCLE_ACAD_YR, AWRD_TY_DESC, FD_FD_TTL, AVG_PAID_AMT, COUNT_STUDENTS

```