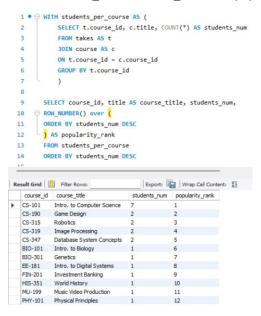
1) OLAP

 Based on the number of total students enrolled in each course, rank the courses from most popular to least popular. The result should include the columns: course_id, course_title, students_num, and popularity_rank



2. Analyze the academic performance of students across courses and departments. Show the average grade by department, courses within that department, and overall average grade across all departments and courses. Use the following conversions from grade letters to numbers: ['A','A+','A-'] → 100, ['B','B+','B-'] → 90, ['C','C+','C-'] → 80, ['D','D+','D-'] → 60, and 0 for all the other letters. The result should include the columns: department, course, grades_average.

```
1 • ⊖ WITH num_grades AS (
 2
          SELECT t.id, t.course_id, c.title, c.dept_name,
 3
             WHEN grade IN ('A', 'A+', 'A-') THEN 100
 5
             WHEN grade IN ('B', 'B+', 'B-') THEN 90
             WHEN grade IN ('C', 'C+', 'C-') THEN 80
 6
            WHEN grade in ('D', 'D+', 'D-') THEN 60
             ELSE 0
 8
 9
          END AS num_grade
10
          FROM takes AS t
          JOIN course AS c
11
12
          ON t.course_id = c.course_id
13
14
15
       SELECT dept_name AS department, title AS course, AVG(num_grade) AS grades_average
16
      FROM num_grades
      GROUP BY dept_name, course WITH rollup;
17
   department course
                                          grades_average
                Genetics
Biology
                                          0.0000
   Biology
               Intro. to Biology
                                      100.0000
   Biology
                                          50.0000
   Comp. Sci. Database System Concepts 100.0000
   Comp. Sci. Game Design
                                          95.0000
   Comp. Sci. Image Processing
                                          95.0000
                Intro. to Computer Science 78.5714
    Comp. Sci.
    Comp. Sci.
                Robotics
                                          95.0000
    Comp. Sci.
                                          88.0000
   Elec. Eng.
                Intro. to Digital Systems
                                          80.0000
                NULL
   Elec. Eng.
                                          80,0000
                Investment Banking
                                          80,0000
   Finance
   Finance
                                          80.0000
   History
                World History
                                          90.0000
   History
                                          90.0000
                Music Video Production
   Music
                                          100.0000
                NULL
   Music
                                          100.0000
   Physics
                Physical Principles
                                          90.0000
                NULL
   Physics
                                          90.0000
                NULL
   NULL
                                          84.5455
```

3. For each semester identify the top 3 students that received the highest grades in that semester (across all the courses they took in that semester). The result should include the columns: year, semester, student_name, grade

```
1 • WITH grades_per_semester AS (

SELECT t.ID, s.name, t.semester, t.year, t.grade,

ROW_NUMBER() over (

PARTITION BY t.semester, t.year

ORDER BY t.grade
) AS grade_rank
FROM takes AS t

JOIN student AS s

ON s.ID = t.ID

ORDER BY t.semester, t.year
)

SELECT year, semester, name AS student_name, grade

FROM grades_per_semester

WHERE grade_rank <= 3;
```

