1) OLAP

1. 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

```
WITH students_per_course AS (
    SELECT t.course_id, c.title, COUNT(*) AS students_num
    FROM takes AS t
    JOIN course AS c
    ON t.course_id = c.course_id
    GROUP BY t.course_id
    )

SELECT course_id, title AS course_title, students_num,
ROW_NUMBER() over (
ORDER BY students_num DESC
) AS popularity_rank
FROM students_per_course
ORDER BY students_num DESC
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

course_id	course_title	students_num	popularity_rank
CS-101	Intro. to Computer Science	7	1
CS-190	Game Design	2	2
CS-315	Robotics	2	3
CS-319	Image Processing	2	4
CS-347	Database System Concepts	2	5
BIO-101	Intro. to Biology	1	6
BIO-301	Genetics	1	7
EE-181	Intro. to Digital Systems	1	8
FIN-201	Investment Banking	1	9
HIS-351	World History	1	10
MU-199	Music Video Production	1	11
PHY-101	Physical Principles	1	12

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.

```sql
WITH num_grades AS (
    SELECT t.id, t.course_id, c.title, c.dept_name,
    CASE
        WHEN grade IN ('A', 'A+', 'A-') THEN 100
        WHEN grade IN ('B', 'B+', 'B-') THEN 90
        WHEN grade IN ('C', 'C+', 'C-') THEN 80
        WHEN grade in ('D', 'D+', 'D-') THEN 60
        ELSE 0
    END AS num_grade
    FROM takes AS t
    JOIN course AS c
    ON t.course_id = c.course_id
    )

SELECT dept_name AS department, title AS course, AVG(num_grade) AS grades_average
FROM num_grades
GROUP BY dept_name, course WITH rollup;
```

department	course	grades_average
Biology	Genetics	0.0000
Biology	Intro. to Biology	100.0000
Biology	NULL	50.0000
Comp. Sci.	Database System Concepts	100.0000
Comp. Sci.	Game Design	95.0000
Comp. Sci.	Image Processing	95.0000
Comp. Sci.	Intro. to Computer Science	78.5714
Comp. Sci.	Robotics	95.0000
Comp. Sci.	NULL	88.0000
Elec. Eng.	Intro. to Digital Systems	80.0000
Elec. Eng.	NULL	80.0000
Finance	Investment Banking	80.0000
Finance	NULL	80.0000
History	World History	90.0000
History	NULL	90.0000
Music	Music Video Production	100.0000
Music	NULL	100.0000
Physics	Physical Principles	90.0000
Physics	NULL	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

```sql
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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

year	semester	student_name	grade
2017	Fall	Zhang	A
2017	Fall	Shankar	A
2017	Fall	Brown	A
2017	Spring	Shankar	A
2017	Spring	Williams	B+
2017	Spring	Aoi	C
2018	Spring	Shankar	A
2018	Spring	Brown	A
2018	Spring	Sanchez	A-
2017	Summer	Tanaka	A
2018	Summer	Tanaka	NULL