- -- Using window functions
- -- 1. Using ROW_NUMBER() to list all students along with a row number based on their enrollment date in ascending order

SELECT

STU_TBL.student_id AS ID,

STU_TBL.student_name AS NAME,

ENR_TBL.enrollment_date AS ENR_DATE,

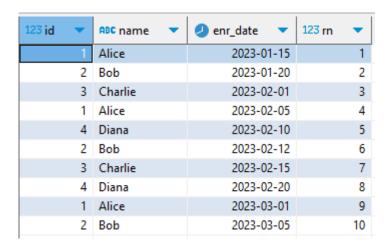
ROW_NUMBER() OVER (ORDER BY ENR_TBL.enrollment_date ASC) AS RN

FROM

Enrollments ENR_TBL

JOIN

Students STU_TBL ON ENR_TBL.student_id = STU_TBL.student_id;



-- 2. Using RANK() to rank students based on the number of courses they are enrolled in, handling ties by assigning the same rank

SELECT

NAME,

COUNT_COURSES,

RANK() OVER (ORDER BY COUNT_COURSES DESC) AS RANKING

```
FROM (

SELECT

STU_TBL.student_name AS NAME,

COUNT(ENR_TBL.course_id) AS COUNT_COURSES

FROM

Students STU_TBL

LEFT JOIN

Enrollments ENR_TBL ON STU_TBL.student_id = ENR_TBL.student_id

GROUP BY

STU_TBL.student_name
) AS STU_COURSE_COUNT;
```

ABC name	123 count_courses	123 ranking
Alice	3	1
Bob	3	1
Charlie	2	3
Diana	2	3

-- 3. Using DENSE_RANK() to determine the dense rank of courses based on their enrollment count across all students

```
SELECT

COURSE,

ENROLL_COUNT,

DENSE_RANK() OVER (ORDER BY ENROLL_COUNT DESC) AS DRANK

FROM (

SELECT

CRS_TBL.course_name AS COURSE,

COUNT(ENR_TBL.student_id) AS ENROLL_COUNT

FROM

Courses CRS_TBL

LEFT JOIN
```

Enrollments ENR_TBL ON CRS_TBL.course_id = ENR_TBL.course_id GROUP BY

CRS_TBL.course_name) AS CRS_ENROLL_COUNT;

ABC course	123 enroll_count	123 drank
Calculus I	3	1
Introduction to C	3	1
Data Structures	2	2
World History	1	3
Biology Basics	1	3