-- 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
   s.student_id AS student_id,
   s.student_name AS student_name,
   e.enrollment_date AS enrollment_date,
   ROW_NUMBER() OVER (
      ORDER BY
         e.enrollment_date ASC
   ) AS row_number
FROM
   enrollments e
   JOIN students s ON e.student_id = s.student_id;
```

123 id	ABC name	enr_date	123 rn
1	Alice	2023-01-15	1
2	Bob	2023-01-20	2
3	Charlie	2023-02-01	3
1	Alice	2023-02-05	4
4	Diana	2023-02-10	5
2	Bob	2023-02-12	6
3	Charlie	2023-02-15	7
4	Diana	2023-02-20	8
1	Alice	2023-03-01	9
2	Bob	2023-03-05	10

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

```
SELECT
   student_name,
   course_count,
```

```
    RANK() OVER (
        ORDER BY
            course_count DESC
    ) AS ranking
FROM
    (
        SELECT
            s.student_name AS student_name,
            COUNT(e.course_id) AS course_count
        FROM
            students s
            LEFT JOIN enrollments e ON s.student_id = e.student_id
        GROUP BY
            s.student_name
    ) AS student_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_name,
    enrollment_count,
    DENSE_RANK() OVER (
        ORDER BY
            enrollment_count DESC
    ) AS dense_rank
```

```
FROM
  (
    SELECT
      c.course_name AS course_name,
      COUNT(e.student_id) AS enrollment_count
    FROM
      courses c
      LEFT JOIN enrollments e ON c.course_id = e.course_id
    GROUP BY
      c.course_name
  ) AS course_enrollment_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