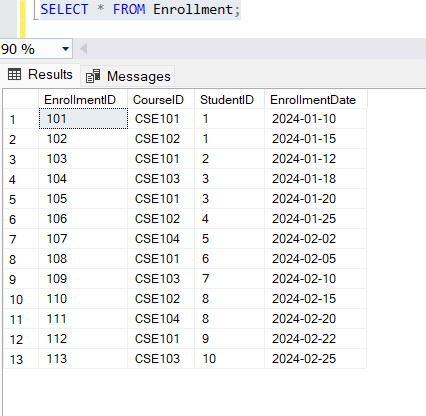
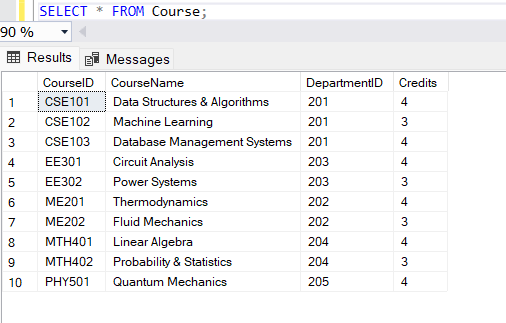
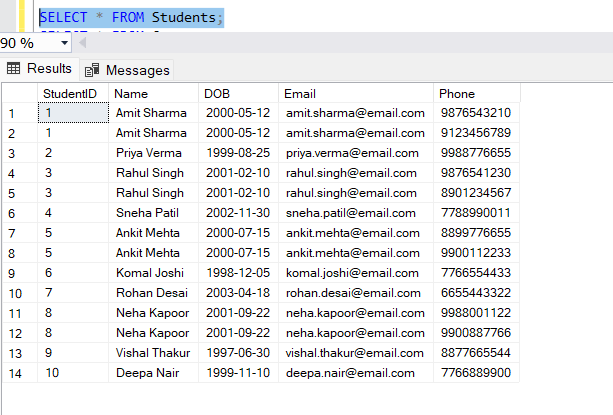
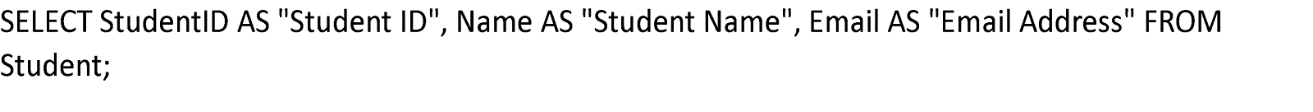
**Assignment 4**

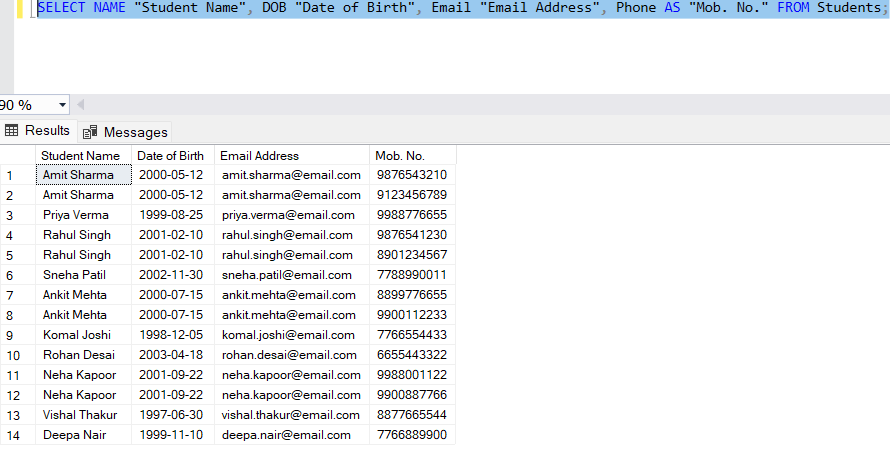
**Overview:** This is a continuation to the previous assignment, The goal of this sub-assignment is to teach students how to retrieve and manipulate data in Oracle SQL, using basic SQL statements, WHERE clauses, sorting, group functions, and Joins. The assignment covers a variety of SQL operations, allowing students to practice filtering, sorting, grouping, and joining data across different tables.

1. SQL Query Basics
   * **Using the SELECT Statement:** Write queries to retrieve all columns from tables like Student, Course, and Enrollment, etc.

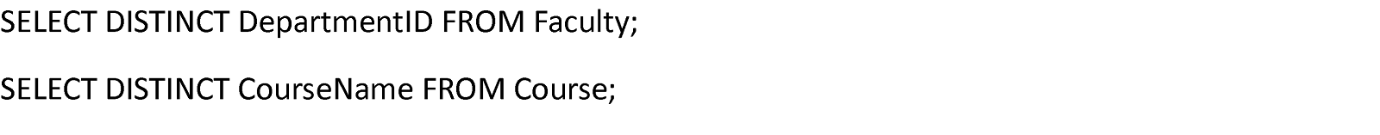


* **Column Aliases:** Use column aliases to provide readable column names in the output.
  + **Example Query:** (experiment using different column names and tables)

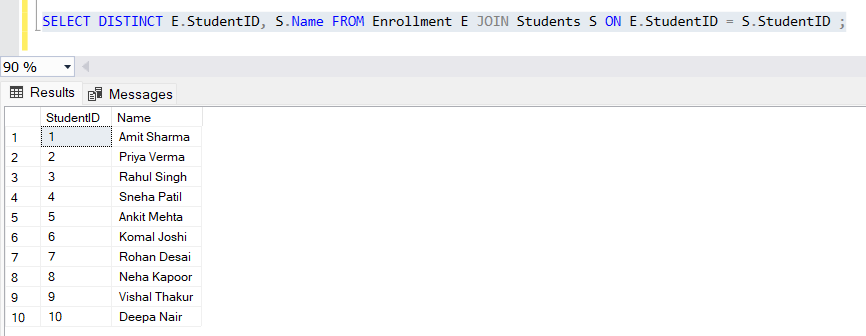
****



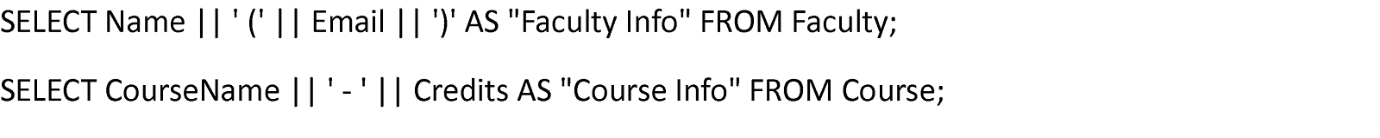
* **DISTINCT and UNIQUE Operators:** Retrieve unique records from Courses and other tables using the DISTINCT or UNIQUE operator.
  + **Example Query:** (experiment using different column names and tables)

****

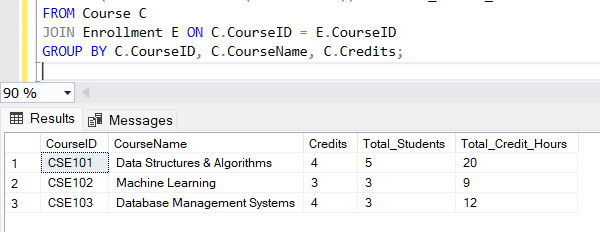
* Retrieve distinct student names from the Enrollment table.



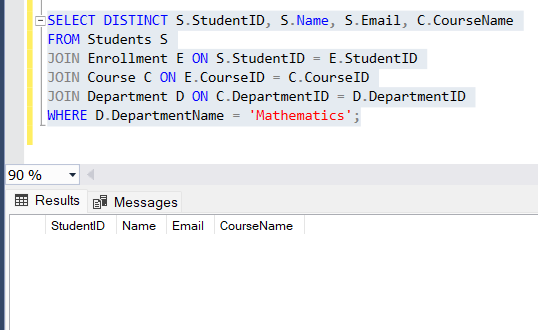
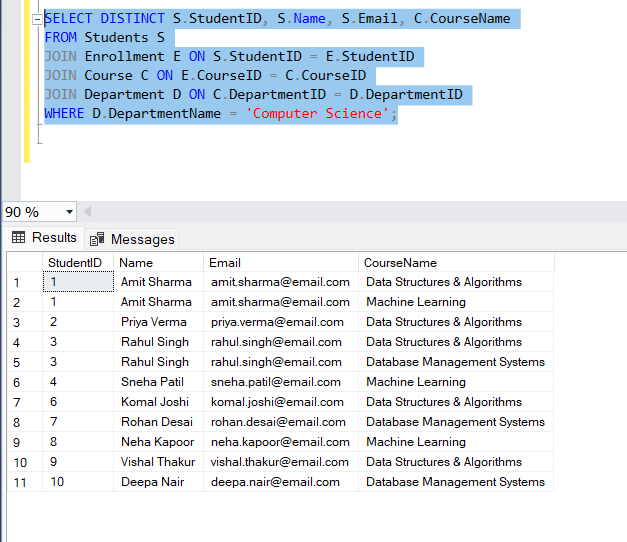
* **Concatenation and Arithmetic Expressions:** Use the concatenation operator (||) and arithmetic operators (+, -, \*, /) to combine data or perform basic math on data.
  + **Example Query:** (experiment using different column names and tables)

****

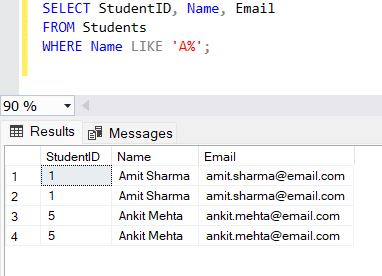
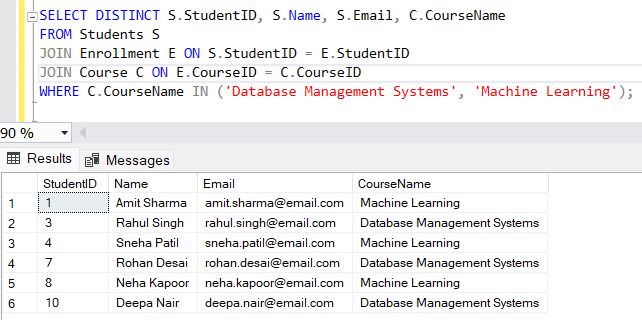
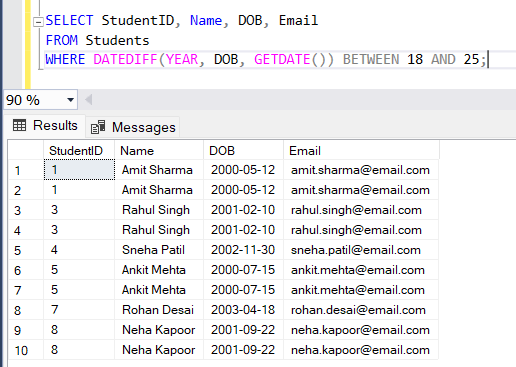
* Calculate total credits for students enrolled in courses (Credits \* Number of students)

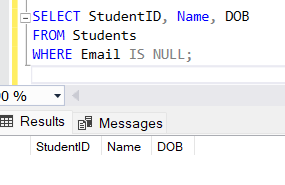


1. Filtering Data:
   * **WHERE Clause and Comparison Operators:** Write queries using the WHERE clause to filter data based on specific conditions.
     + Retrieve students enrolled in the 'CS' department
     + Retrieve students enrolled in courses offered by the 'Mathematics' department:

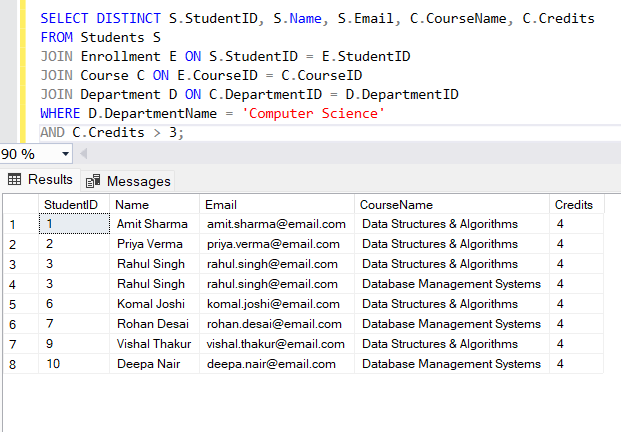


* **BETWEEN...AND, IN, LIKE, IS NULL:** Use different filtering operators,
  + Retrieve students whose age is between 18 and 25.
  + Retrieve students enrolled in specific courses.
  + Retrieve students whose names start with 'A'.
  + Retrieve students who don't have an email listed (null email).

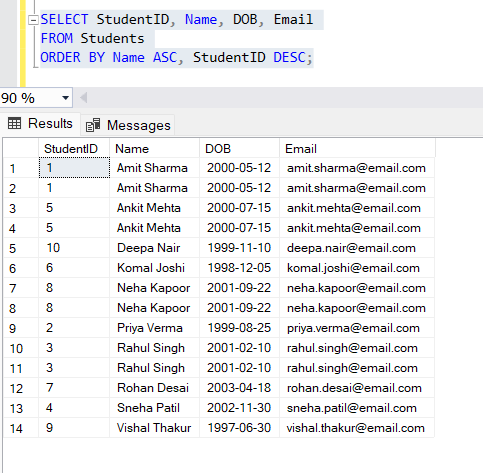




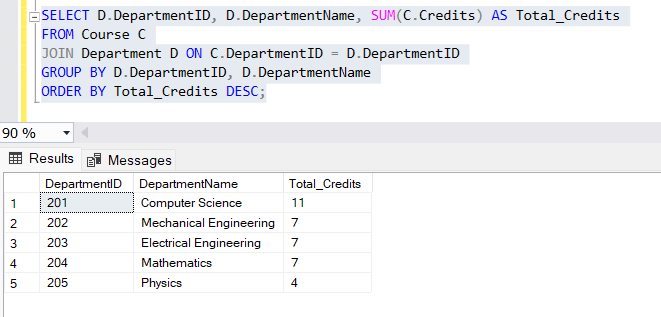
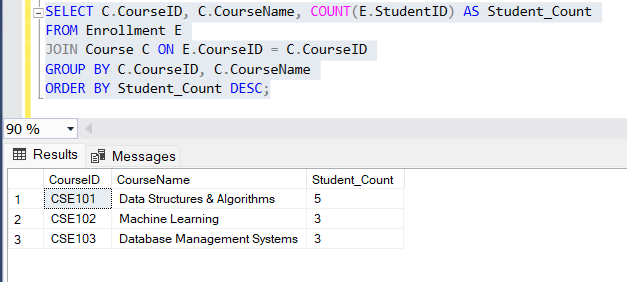
* **Logical Operators:** Use logical operators to combine multiple conditions in your queries.
  + Retrieve students from the 'CS' department who are enrolled in courses that have more than 3 credits



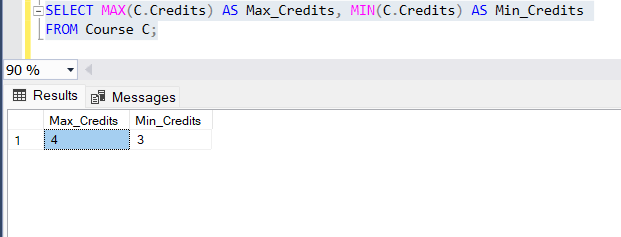
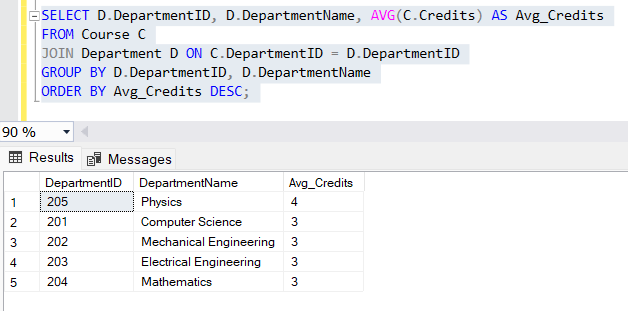
1. Sorting and Grouping Data:
   * **ORDER BY Clause:**
     + Sort students by their names in ascending order and by their StudentID in descending order
     + Sort courses by the CourseName in alphabetical order



* **GROUP BY and HAVING Clauses:**
  + Group students by their course and count the number of students in each course
  + Group courses by department and find the total number of credits per department

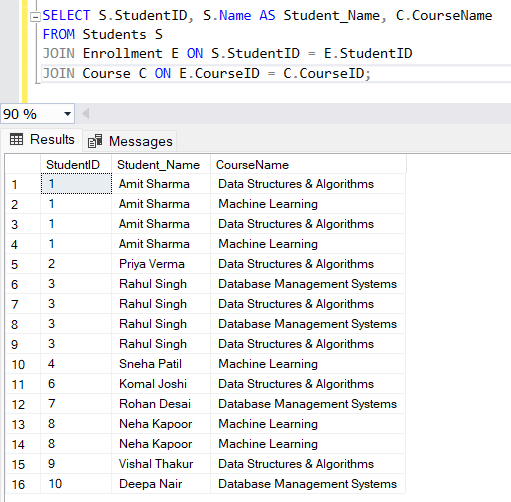


* **Using Group Functions:**
  + Calculate the average number of credits for each department
  + Find the maximum and minimum credits available for courses

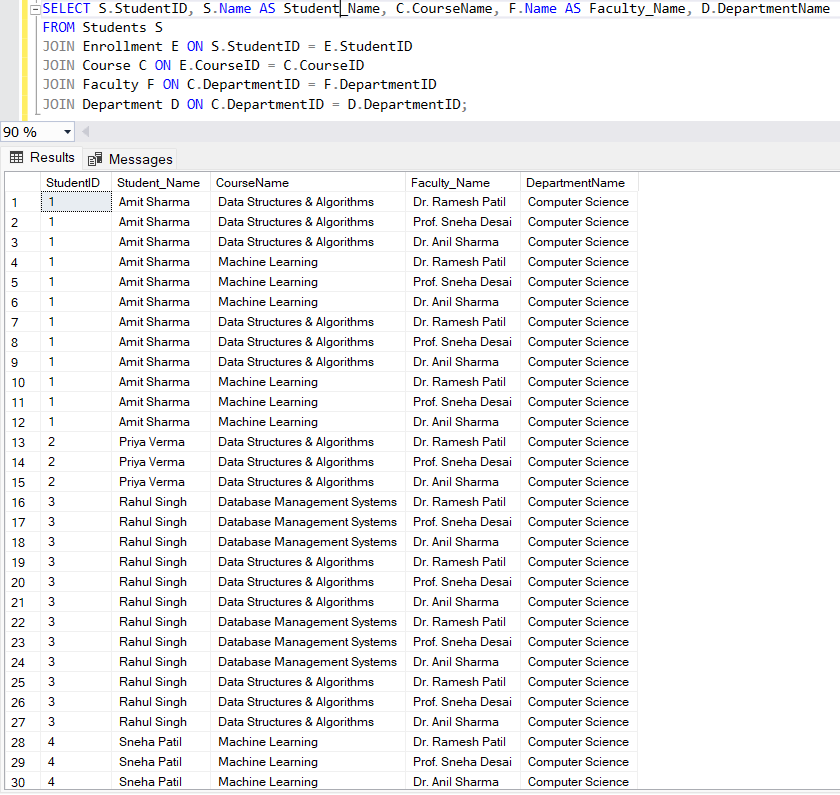


1. Using Joins:
   * **Types of Joins:**
     + **INNER JOIN:** Retrieve students and their respective courses (using inner join)
     + **OUTER JOIN (LEFT, RIGHT, FULL):** Retrieve all students and the courses they are enrolled in (including students not enrolled in any course)
     + **SELF JOIN:** Retrieve pairs of students enrolled in the same course (self join)
     + **CROSS JOIN:** Retrieve all combinations of students and courses (cross join)

* **Handling Ambiguous Column Names:** When multiple tables have columns with the same name (e.g., StudentID), use table aliases to disambiguate
  + Retrieve student names and their course names (handling ambiguous column names)



* **Multiple Join Operations:**
  + Retrieve students, courses, and faculty details with multiple joins



1. Deliverables (How and what to submit):
   * Students should take complete screenshots or log the output from their SQL queries executed on the database.
     + Provide output for each query you write (e.g., a screenshot of the result).
     + Make sure the test results correspond to the queries, showing that the queries return the expected results.
     + Submit the screenshots/logs as part of the .zip file or as separate images in the submission.
     + Comments should also include any assumptions or considerations (e.g., if specific data is missing or if there are any optimizations).