



DATABASE PROGRAMMING

Instructions

- Complete all tasks below.
- **TOPIC:** ADVANCED SQL QUERIES

QUESTION

1. INNER JOIN

- Write a query to retrieve all employees along with their corresponding department names from the `employees` and `departments` tables.

2. LEFT JOIN

- Write a query to list all customers and their orders, showing customers even if they have no orders.

3. Subquery in the SELECT Clause

- Write a query to select all products along with the average price of products in the same category.

4. Correlated Subquery

- Write a query to find all employees whose salary is greater than the average salary of their department.

5. UNION

- Write a query to combine results from the `students` and `teachers` tables, showing names in a single list.

6. GROUP BY with HAVING

- Write a query to find departments with more than 10 employees and list their total salaries.

7. CASE Statement

- Write a query to categorize employees based on their salary. For example: "High", "Medium", "Low".

8. CTE (Common Table Expression)

- Write a CTE to find the top 5 products by sales, and then select from this CTE to show their names and total sales.

9. Window Functions

- Write a query to rank employees based on their salaries and show the rank along with their names.

10. Creating and Using Temporary Tables

- Write a query to create a temporary table that stores products with a stock level below 10, and then select from this temporary table.

11. Full Outer Join

- Write a query to retrieve all customers and their orders, including customers with no orders and orders with no associated customers.

12. EXISTS Clause

- Write a query to find all students who have enrolled in at least one course.

13. CROSS JOIN

- Write a query to generate a list of all possible combinations of students and courses.

14. JSON Functions (if the database supports it)

- Write a query to extract specific fields from a JSON column in the `users` table.

15. Index Usage

- Write a query that demonstrates the improvement in performance when using an index on a frequently queried column.

END