Centre for Development of Advanced Computing (C-DAC) Bangalore

Post Graduate Diploma in Advanced Computing (PG-DAC)

August 2025 Batch

Lab Assessment

Module: Database Technologies

Date & Time: 29 Sep 2025, 0930 hrs - -1130 hrs

Duration: 2 hrs

MySQL Table Creation and Data

```
CREATE TABLE Students (
    student_id INT PRIMARY KEY,
    name VARCHAR(50),
    age INT,
    join_date DATE,
    grade CHAR(1),
    course_id INT
);

CREATE TABLE Courses (
    course_id INT PRIMARY KEY,
    course_name VARCHAR(50),
    fees DECIMAL(10,2)
);

INSERT INTO Students VALUES
(1, 'Alice', 22, '2021-03-10', 'A', 101),
(2, 'Bob', 25, '2020-12-15', 'B', 102),
(3, 'Charlie', 23, '2022-05-20', 'A', 101),
(4, 'David', 24, '2021-07-01', 'C', 103),
(5, 'Emma', 21, '2023-01-11', 'A', 102),
(6, 'Frank', 26, '2022-08-15', 'B', 103),
(7, 'Grace', 22, '2021-11-05', 'A', 101),
(8, 'Hannah', 23, '2023-02-20', 'C', 102);

INSERT INTO Courses VALUES
(101, 'Data Science', 30000),
(102, 'Web Development', 25000),
(103, 'AI & ML', 40000),
(104, 'Cyber Security', 35000);
```

SQL Problem Statements

Q1. List all customers who joined after 2021-01-01. -- 2 Marks

Q2. Display customers aged above 27. -- 2 Marks

Q3. List all orders with more than one item. -- 2 Marks

Q4. Display orders with price greater than 30000. -- 2 Marks

Q5. Find the total quantity of orders for each customer. -- 4 Marks

Q6. Show the total revenue generated by each customer. -- 4 Marks

Q7. Display the most expensive product ordered by each customer. -- 4 Marks

Q8. Write a stored procedure GetOrdersByCustomer that takes a customer ID as an IN parameter and returns the total number of orders placed by that customer as an OUT parameter. -- 5 Marks

Q9. Write a function CalculateDiscount that takes order price as input and returns 10% discount amount. -- 5 Marks

MongoDB Sample Data (**orders** collection)

```
[
 { "order_id": 401, "customer": "Arun", "product": "Laptop", "quantity": 2, "price": 60000 },
 { "order_id": 402, "customer": "Meena", "product": "Mobile", "quantity": 1, "price": 25000 },
 { "order_id": 403, "customer": "Ravi", "product": "Tablet", "quantity": 3, "price": 18000 },
 { "order_id": 404, "customer": "Divya", "product": "Headphones", "quantity": 1, "price": 2000 },
 { "order_id": 405, "customer": "Kiran", "product": "Laptop", "quantity": 2, "price": 60000 },
 { "order_id": 406, "customer": "Arun", "product": "Tablet", "quantity": 1, "price": 18000 },
 { "order_id": 407, "customer": "Meena", "product": "Headphones", "quantity": 2, "price": 2000 }
]
```

MongoDB Problem Statements

Q10. Display all orders placed by customer "Arun". -- 2 Marks

Q11. Find all orders where price > 30000. -- 2 Marks

Q12. Display all orders where the customer is either "Arun" or "Meena". -- 2 Marks

Q13. Show only the customer name and product ordered. -- 2 Marks

Q14. Use an aggregate function to calculate the total quantity of all products ordered. -- 2 Marks