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, 1400 hrs - -1600 hrs

Duration: 2 hrs

MySQL Table Creation and Data

```
CREATE TABLE Customers (
    customer_id INT PRIMARY KEY,
    name VARCHAR(50),
    age INT,
    join_date DATE
);

CREATE TABLE Orders (
    order_id INT PRIMARY KEY,
    customer_id INT,
    product VARCHAR(50),
    quantity INT,
    price DECIMAL(10,2),
    order_date DATE,
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);

INSERT INTO Customers VALUES
(301, 'Arun', 28, '2021-02-10'),
```

(302, 'Meena', 25, '2020-11-05'),

(303, 'Ravi', 30, '2022-01-15'),

(304, 'Divya', 26, '2021-07-20'),

(305, 'Kiran', 27, '2023-03-05'),

(306, 'Sonia', 29, '2022-09-10');

INSERT INTO Orders VALUES

(401, 301, 'Laptop', 2, 60000, '2023-01-10'),

(402, 302, 'Mobile', 1, 25000, '2023-02-15'),

(403, 303, 'Tablet', 3, 18000, '2023-03-20'),

(404, 304, 'Headphones', 1, 2000, '2023-04-05'),

(405, 305, 'Laptop', 2, 60000, '2023-05-12'),

(406, 301, 'Tablet', 1, 18000, '2023-06-01'),

(407, 302, 'Headphones', 2, 2000, '2023-06-15');

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