Task 6 :

Subqueries and Nested Queries

Objective: Use subqueries in SELECT, WHERE, and FROM

Tools : DB Browser for SQLite / MySQL Workbench

Deliverables : SQL queries with nested logic

```
-- 1. Create Customers table
CREATE TABLE Customers (
    customer_id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
    city TEXT NOT NULL
);

-- 2. Create Orders table
CREATE TABLE Orders (
    order_id INTEGER PRIMARY KEY,
    customer_id INTEGER,
    amount REAL,
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);

-- 3. Insert sample data into Customers
INSERT INTO Customers (customer_id, name, city) VALUES
(1, 'Alice', 'Delhi'),
```

(2, 'Bob', 'Mumbai'),

(3, 'Charlie', 'Delhi'),

(4, 'Diana', 'Kolkata');

Select * from Customers;

customer_id	name	city
1	Alice	Delhi
2	Bob	Mumbai
3	Charlie	Delhi
4	Diana	Kolkata

-- 4. Insert sample data into Orders

INSERT INTO Orders (order_id, customer_id, amount) VALUES

(101, 1, 5000),

(102, 2, 3000),

(103, 1, 7000),

(104, 3, 2000);

Select * from Orders;

```
SELECT * FROM Orders;
```

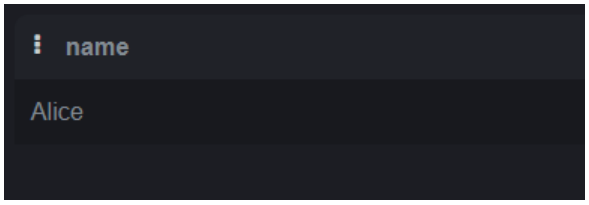
order_id	customer_id	amount
101	1	5000
102	2	3000
103	1	7000
104	3	2000

```
-- 5. Subquery in SELECT - Total spent by each customer
SELECT name,
    (SELECT SUM(amount)
     FROM Orders
     WHERE Orders.customer_id = Customers.customer_id) AS total_spent
FROM Customers;
```

name	total_spent
Alice	12000
Bob	3000
Charlie	2000
Diana	NULL

```
-- 6. Subquery in WHERE - Customers who spent more than 5000
SELECT name
FROM Customers
WHERE customer_id IN (
    SELECT customer_id
    FROM Orders
    GROUP BY customer_id
```

```
    HAVING SUM(amount) > 5000
);
```



name
Alice

```
-- 7. Subquery in FROM - Average order amount per customer
SELECT sub.customer_id, sub.avg_amount
FROM (
    SELECT customer_id, AVG(amount) AS avg_amount
    FROM Orders
    GROUP BY customer_id
) AS sub;
```

customer_id	avg_amount
1	6000
2	3000
3	2000