Task 3: SQL for Data Analysis

Mock Database Tables

1. users table:

user_id	name	email
1	Aisha	aisha@mail.com
2	Rohan	rohan@mail.com
3	Meera	meera@mail.com

2. orders table:

order_id	user_id	product	amount	order_date
101	1	Shoes	2000	2024-05-01
102	2	T-shirt	500	2024-05-02
103	1	Watch	1500	2024-05-03
104	3	Headphones	1200	2024-05-03
105	2	Sunglasses	800	2024-05-04

SQL Queries and Explanation

1. SELECT with WHERE

```
SELECT * FROM orders WHERE amount > 1000;
```

2. INNER JOIN

```
SELECT users.name, orders.product, orders.amount
FROM users
JOIN orders ON users.user_id = orders.user_id;
```

3. GROUP BY with SUM and AVG

```
SELECT user_id, SUM(amount) AS total_spent, AVG(amount) AS avg_spent
FROM orders
```

```
GROUP BY user_id;
```

4. Subquery

```
SELECT name FROM users
WHERE user_id IN (
  SELECT user_id FROM orders GROUP BY user_id HAVING SUM(amount) > 2000
);
```

5. Create a View

```
CREATE VIEW high_value_orders AS
SELECT * FROM orders WHERE amount > 1000;
```

6. Create an Index

```
CREATE INDEX idx_user_id ON orders(user_id);
```

Interview Questions - Simple Answers

1. WHERE filters rows before grouping, HAVING filters after grouping.

2. Joins: INNER (common records), LEFT (all from left), RIGHT (all from right).

3. Average revenue per user: AVG(total_amount) after GROUP BY user_id.

4. Subqueries are queries inside another query (used for filtering, etc.).

5. To optimize: use indexes, avoid SELECT *, filter early with WHERE.

6. A view is a virtual table made from a SELECT query.

7. Handle NULLs using IS NULL, IS NOT NULL, or functions like COALESCE().