1️⃣ Difference between WHERE and HAVING

Feature	WHERE	HAVING
Use	Filters rows **before grouping**	Filters groups **after GROUP BY**
Works on	Individual rows	Aggregate functions (like SUM, AVG)
Example	SELECT * FROM orders WHERE city='Delhi';	SELECT city, SUM(sales) FROM orders GROUP BY city HAVING SUM(sales) > 10000;

👉 In short:

- **WHERE** → filters rows.
- **HAVING** → filters grouped data.

2️⃣ Different types of JOINS

Type	Description	Example
INNER JOIN	Returns rows that match in both tables	Common customers who have orders
LEFT JOIN	All rows from left table + matching rows from right	All customers, even those with no orders
RIGHT JOIN	All rows from right table + matching rows from left	All orders, even if no customer info
FULL JOIN	All rows from both tables	Combines left and right (all data)
CROSS JOIN	Every row from first table paired with every row of second	Used rarely, creates all combinations

---

3️⃣ Calculate Average Revenue Per User (ARPU)

Formula:

SELECT

```
  SUM(revenue) / COUNT(DISTINCT customer_id) AS avg_revenue_per_user
FROM orders;
```

👉 This divides total revenue by total number of unique users.

4️⃣ What are Subqueries?

- A **subquery** is a query **inside another query**.
- It helps to use the result of one query as input to another.

Example:

```
SELECT customer_name
FROM customers
WHERE customer_id IN (
  SELECT customer_id FROM orders WHERE total_amount > 10000
);
```

Here, the inner query finds customers with large orders, and the outer query fetches their names.

5️⃣ How to Optimize a SQL Query

Some best practices:

1. **Use indexes** on columns used in WHERE, JOIN, and ORDER BY.
2. **Avoid SELECT *** (fetch only required columns).
3. **Use proper JOINs** instead of subqueries when possible.
4. **Use LIMIT** to reduce data fetched.
5. **Analyze execution plan** (EXPLAIN in PostgreSQL/MySQL).
6. **Avoid functions on indexed columns** inside WHERE.

6️⃣ What is a View in SQL?

- A **View** is a **virtual table** created from a query.
- It doesn't store data itself — it shows data from one or more tables.

Example:

CREATE VIEW customer_sales AS

SELECT customer_id, SUM(total_amount) AS total_spent

FROM orders

GROUP BY customer_id;

Then you can use:

SELECT * FROM customer_sales;

7️⃣ How to Handle NULL Values in SQL

You can handle NULL using:

1. **IS NULL / IS NOT NULL**
2. SELECT * FROM customers WHERE phone IS NULL;
3. **COALESCE()** → replaces NULL with a default value
4. SELECT COALESCE(phone, 'Not Available') AS phone_number FROM customers;
5. **IFNULL() / NVL()** → depends on SQL version, similar to COALESCE.