15 Advanced SQL Concepts With Examples (2025 Edition)

Table of Contents

1. Standard Operations

Basic SQL operations for filtering, sorting, and modifying table structure using WHERE, ORDER BY, and ALTER TABLE.

2. Subqueries / Nested Queries

A subquery is a query within another query. Useful for filtering based on results of another table.

Example:

```
SELECT customer_name
FROM fresher.customers
WHERE customer_id IN (
  SELECT customer_id
  FROM fresher.orders
  WHERE order_date >= DATE_SUB(NOW(), INTERVAL 1 MONTH)
);
```

NOW() returns current timestamp.

DATE_SUB(NOW(), INTERVAL 1 MONTH) returns date one month ago.

3. Joins (INNER, LEFT, RIGHT, CROSS)

INNER JOIN: Returns matching records from both tables.

LEFT JOIN: Returns all records from left table, matching records from right.

RIGHT JOIN: Returns all from right table, matching from left.

CROSS JOIN: Returns all combinations.

Example:

```
SELECT s.student_name, c.course_name
FROM fresher.students s
INNER JOIN fresher.courses c ON s.student_id = c.student_id;
```

4. UNION Operation

Combines results of two queries, removing duplicates.

Example:

```
SELECT customer_name FROM customers WHERE country = 'USA'
UNION
SELECT customer_name FROM customers WHERE country = 'Canada';
```

5. Aggregate Functions with GROUP BY

Used to perform calculations on grouped data.

Example:

```
SELECT product_category, SUM(order_amount) AS total_sales
```

```
FROM orders
GROUP BY product_category;
```

6. Window Functions

Used to rank or compare rows without grouping.

Example:

```
SELECT customer_name, order_amount,
ROW_NUMBER() OVER (ORDER BY order_amount DESC) AS rank_
FROM orders;
```

7. String Manipulation

Common functions: CONCAT(), SUBSTRING(), REPLACE()

Example:

```
SELECT CONCAT(first_name, ' ', last_name) AS full_name
FROM employees;
```

8. User-Defined Functions (UDFs)

Used to define custom logic.

Example:

```
CREATE FUNCTION calculate_discount(price DECIMAL(10,2), discount DECIMAL(5,2))
RETURNS DECIMAL(10,2)
RETURN price - (price * discount);
```

Used as:

```
SELECT product_name, calculate_discount(price, discount_rate)
```

```
FROM fresher.products;
```

9. Temporary Tables

Session-based temporary storage for intermediate results.

Example:

```
CREATE TEMPORARY TABLE temp_orders (
  order_id INT,
  customer_name VARCHAR(100),
  total_amount DECIMAL(10,2)
);
```

10. Stored Procedures

Reusable SQL blocks.

Example:

```
DELIMITER $$
CREATE PROCEDURE get_todays_orders()
BEGIN
  CREATE TEMPORARY TABLE todays_orders AS
  SELECT * FROM orders WHERE order_date = CURDATE();
  SELECT * FROM todays_orders;
END$$
DELIMITER ;
```

11. External Query Filter / Predicate Pushdown

Optimization technique to push filtering logic to the source system.

Example:

```
CREATE PROCEDURE get_external_orders(IN target_date DATE)
BEGIN
  SELECT * FROM external_db.orders WHERE order_date = target_date;
END;
```

Benefit: Reduces transferred data, increases performance.