



SQL Query

SELECT Queries

Fetch All Rows

- Query: `SELECT * FROM [tablename]`
- Fetches all rows in a table.

Fetch Specified Columns

- Query: `SELECT id, name FROM [tablename]`
- Fetches only the specified columns from the table.

Alias a Column

- Query: `SELECT id as product_id FROM [tablename]`
- Fetches the column with the specified name, using an alias.

Fetch Specific Row

- Query: `SELECT * FROM [tablename] WHERE id = x;`
- Fetches a specific row based on the condition `id = x`.

Multiple Conditions (AND)

- Query: `SELECT * FROM [tablename] WHERE inventory > 0 AND price < 20;`
- Retrieves rows that satisfy both conditions: `inventory > 0` and `price < 20`.

Multiple Conditions (OR)

- Query: `SELECT * FROM [tablename] WHERE price > 100 OR price < 20;`
- Retrieves rows that satisfy either condition: `price > 100` or `price < 20`.

Filter by Multiple IDs

- **Query:** `SELECT * FROM [tablename] WHERE id IN (1,2,3);`
- Retrieves rows where the `id` matches any of the specified values (1, 2, 3).

Filter by Partial String Match (Starts With)

- **Query:** `SELECT * FROM products WHERE name LIKE 'TV%';`
- Retrieves products whose names start with 'TV'.

Filter by Partial String Match (Ends With)

- **Query:** `SELECT * FROM products WHERE name LIKE '%e';`
- Retrieves products whose names end with 'e'.

Filter by String Match (Does Not End With)

- **Query:** `SELECT * FROM products WHERE name NOT LIKE '%e';`
- Retrieves products whose names do not end with 'e'.

Filter by String Match (Contains in the Middle)

- **Query:** `SELECT * FROM products WHERE name NOT LIKE '%en%';`
- Retrieves products whose names contain 'en' somewhere in the middle.

ORDER BY and LIMIT Queries

Sort by Price (Ascending)

- **Query:** `SELECT * FROM products ORDER BY price;`
- Arranges rows in ascending order of price.

Sort by Price (Descending)

- **Query:** `SELECT * FROM products ORDER BY price DESC;`
- Arranges rows in descending order of price.

Sort by Inventory (Descending) and Price (Ascending)

- **Query:** `SELECT * FROM products ORDER BY inventory DESC, price ASC;`
- Sorts rows first by inventory in descending order and then by price in ascending order as a tiebreaker.

Limit Results

- **Query:** `SELECT * FROM products LIMIT 10;`
- Retrieves the first 10 rows from the result set.

Limit Results with a Condition

- **Query:** `SELECT * FROM products WHERE price > 100 LIMIT 2;`
- Retrieves the first 2 rows that meet the condition `price > 100`.

Offset Results

- **Query:** `SELECT * FROM products ORDER BY id LIMIT 5 OFFSET 2;`
- Retrieves the first 5 rows, skipping the first 2 rows as specified by the `OFFSET` clause.

INSERT and UPDATE Queries

Insert a New Row

- **Query:** `INSERT INTO products (name, price, inventory) VALUES ('tortilla', 4, 1000);`
- Inserts a new row with specified values into the 'products' table.

Insert with Return

- **Query:** `INSERT INTO products (name, price, inventory) VALUES ('car', 10000, 10) RETURNING *;`
- Inserts a new row with specified values into the 'products' table and returns the inserted row.

Delete Rows with Condition

- **Query:** `DELETE FROM products WHERE inventory = 0 RETURNING *;`

- Deletes rows from the 'products' table where the `inventory` is 0 and returns the deleted rows.

Update Rows with Condition

- **Query:** `UPDATE products SET name = 'keyboard RGB', price = 50 WHERE id = 28 RETURNING *;`
- Updates rows in the 'products' table with specified values where the `id` is 28 and returns the updated rows.