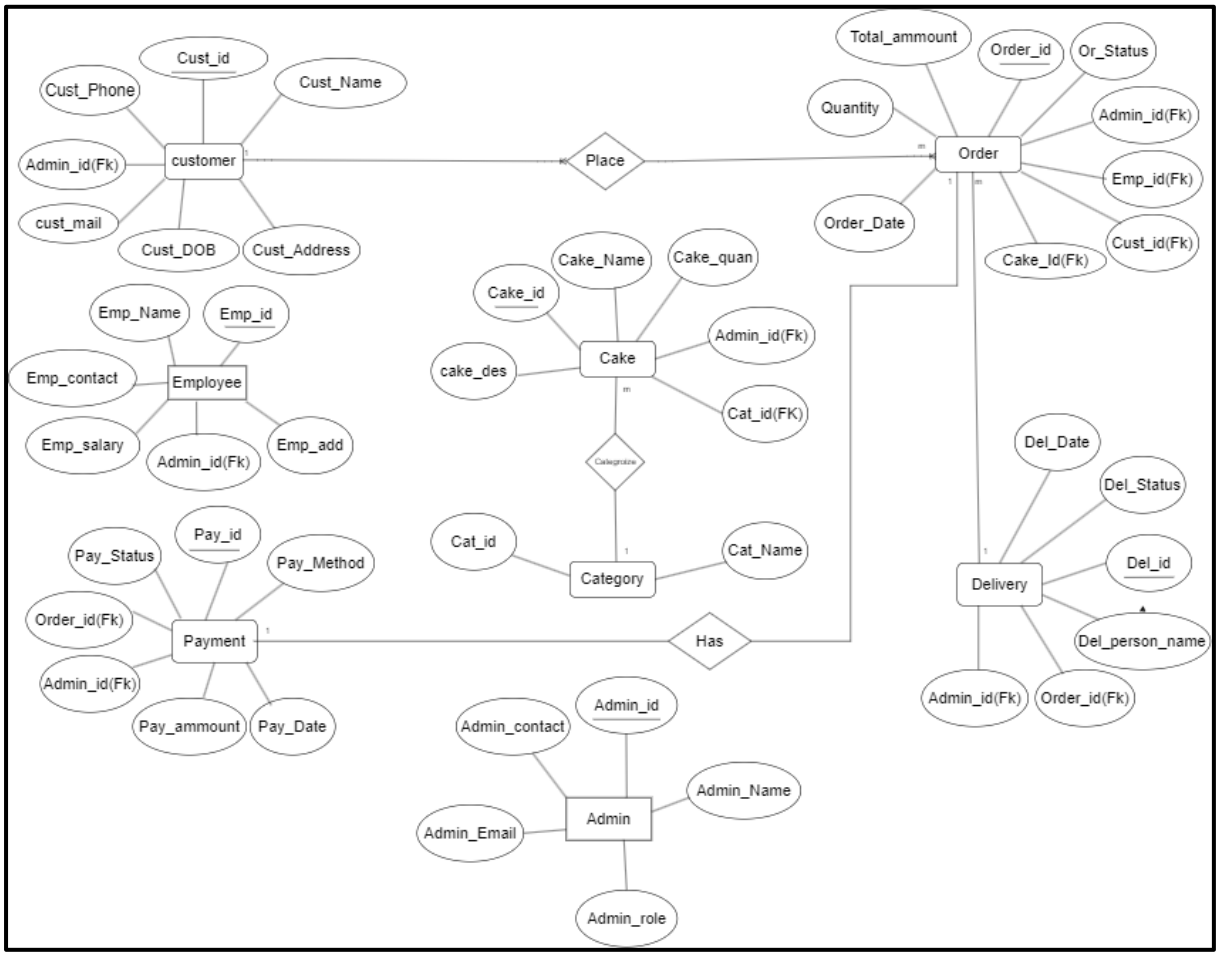
# MySQL Documentation for Cake Shop Database

## Introduction

This documentation explains how the cake shop database is designed and used. It helps manage the main operations of the shop, such as keeping track of employees, customer information, products, orders, and deliveries. Each table in the database has a specific purpose to organize and store data properly. The tables are linked with primary and foreign keys to ensure data can be easily connected and retrieved. You will find examples of common SQL queries for adding, updating, or finding data. There are also instructions for backing up and restoring the database to keep data safe. This guide is meant for anyone who needs to understand or work with the cake shop database.

# ER Diagram




The ER diagram shows how a cake shop database is organized. Here are the main parts:

1. Customer: This section stores details about each customer, like their contact information. Each customer is connected to an admin who manages them.
2. Employee: This part holds information about each employee, such as their name and contact details. Every employee is also connected to an admin.
3. Admin: The admin is the central role in this database. Admins manage customers, employees, orders, and more.
4. Cake: This section stores details about each cake, including its name and quantity. Each cake belongs to a specific category, like “birthday cakes” or “wedding cakes.”

5. Category: This part groups cakes into different categories. Each category can have many cakes.
6. Order: Orders track what a customer buys. Each order links to the customer, the cake, and the employees handling it.
7. Payment: Payments are connected to each order and contain details like the payment amount and method. Each payment is managed by an admin.
8. Delivery: This section tracks the delivery of orders, including the delivery person and the delivery status. Each delivery is managed by an admin.

Key Relationships

- Customers place orders.
- Cakes are categorized by type.
- Categories are managed by admins.

# Normalization

## 1NF

| cust_name | cust_address | cust_phone | cake_name | categoery | cake_price | quantity | total_amount | order_status | emp_name | emp_contact | del_person_name | del_status | pay_method | pay_status | pay_date | pay_amount |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rohan Sharma | 12 MG Road, Pune | 9123456789 | Black Forest Cake | birthday cake | 500 | 2 | 1000 | Completed | Ajit Pandy | 9123456781 | Ajit Pandey | Delivered | Credit Card | Paid | 01-11-2024 | 1000 |
| Sana Patel | 56 Baker Street, Mumbai | 9223456789 | Pineapple Pastry | pastry | 100 | 4 | 400 | Pending | Dorababu Shinde | 9223456781 | Dorababu Shinde | Pending | Cash | Pending | 02-11-2024 | 400 |
| Karan Gupta | 78 Park Avenue, Delhi | 9323456789 | Chocolate Truffle Cake | chocolate cake | 700 | 1 | 700 | Shipped | Ovee Karekar | 9823456781 | Ajit Pandey | Out for Delivery | Online Transfer | Paid | 03-11-2024 | 700 |
| Priya Iyer | 45 Residency Road, Bangalore | 9423456789 | Almond Cookie, gold cookie | cookies | 50,60 | [3],[2] | [150][160] | Delivered | Ganesh Balugade | 9923456781 | Dorababu Shinde | Delivered | Debit Card | Paid | 04-11-2024 | [150][160] |

1NF :
1) Every column/attribute need to have a single value
2) Each row should be unique. Either through a single or multiple columns not mandatory to have primary key

| cust_name | cust_address | cust_phone | cake_name | categoery | cake_price | quantity | total_amount | order_status | emp_name | emp_contact | del_person_name | del_status | pay_method | pay_status | pay_date | pay_amount |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rohan Sharma | 12 MG Road, Pune | 9123456789 | Black Forest Cake | birthday cake | 500 | 2 | 1000 | Completed | Ajit Pandy | 9123456781 | Ajit Pandey | Delivered | Credit Card | Paid | 01-11-2024 | 1000 |
| Sana Patel | 56 Baker Street, Mumbai | 9223456789 | Pineapple Pastry | pastry | 100 | 4 | 400 | Pending | Dorababu Shinde | 9223456781 | Dorababu Shinde | Pending | Cash | Pending | 02-11-2024 | 400 |
| Karan Gupta | 78 Park Avenue, Delhi | 9323456789 | Chocolate Truffle Cake | chocolate cake | 700 | 1 | 700 | Shipped | Ovee Karekar | 9823456781 | Ajit Pandey | Out for Delivery | Online Transfer | Paid | 03-11-2024 | 700 |
| Priya Iyer | 45 Residency Road, Bangalore | 9423456789 | Almond Cookie | cookies | 50 | 3 | 150 | Delivered | Ganesh Balugade | 9923456781 | Dorababu Shinde | Delivered | Debit Card | Paid | 04-11-2024 | 150 |
| Priya Iyer | 45 Residency Road, Bangalore | 9423456789 | Gold Cookie | cookies | 60 | 2 | 160 | Delivered | Ganesh Balugade | 9923456781 | Dorababu Shinde | Delivered | Debit Card | Paid | 04-11-2024 | 160 |

## 2NF

2NF :
1) must be in 1 NF
2) All non key attributes must be fully dependent on candidate key
if a non key column is partially dependent on candidate key ( subset of columns forming candidate key) then split them into separate tables
3) Every table should have primarykey and relationship between the tables should be formed using foreign key

candidate key
set of columns which uniquely identify a record
a table can have multiple candidate keys

non key columns
columns whichare not part of the candidate key or primary key

Partial dependency
if your candidate key is combinatio of 2 columns or multiple columns then every on key column which Are not part of the

1. Customer Table

| cus_id(pk) | cust_name | cust_address | cust_phone |
|---|---|---|---|
| c1 | Rohan Sharma | 12 MG Road, Pune | 9123456789 |
| c2 | Sana Patel | 56 Baker Street, Mumbai | 9223456789 |
| c3 | Karan Gupta | 78 Park Avenue, Delhi | 9323456789 |
| c4 | Priya Iyer | 45 Residency Road, Bangalore | 9423456789 |
| c4 | Priya Iyer | 45 Residency Road, Bangalore | 9423456789 |

2. Cake Table

| cake_id(pk) | cake_name | categoery | cake_price |
|---|---|---|---|
| 1 | Black Forest Cake | birthday cake | 500 |
| 2 | Pineapple Pastry | pastry | 100 |
| 3 | Chocolate Truffle Cake | chocolate cake | 700 |
| 4 | Almond Cookie | cookies | 50 |
| 5 | Gold Cookie | cookies | 60 |

3. **Order Details Table**

| order_id(pk) | quantity | total_amount | order_status |
|---|---|---|---|
| 1 | 2 | 1000 | Completed |
| 2 | 4 | 400 | Pending |
| 3 | 1 | 700 | Shipped |
| 4 | 3 | 150 | Delivered |
| 5 | 2 | 160 | Delivered |

4.employee table

| emp_id(pk) | emp_name | emp_contact |
|---|---|---|
| 1 | Ajit Pandy | 9123456781 |
| 2 | Dorababu Shinde | 9223456781 |
| 3 | Ovee Karekar | 9823456781 |
| 4 | Ganesh | 9923456781 |
| 4 | Ganesh | 9923456781 |

5.delivery table

| del_id(pk) | del_person_name | del_status |
|---|---|---|
| 1 | Ajit Pandey | Delivered |
| 2 | Dorababu Shinde | Pending |
| 3 | Ajit Pandey | Out for Delivery |
| 4 | Dorababu Shinde | Delivered |
| 5 | Dorababu Shinde | Delivered |

6.payment table

| pay_id(pk) | pay_method | pay_status | pay_date | pay_amount |
|---|---|---|---|---|
| 1 | Credit Card | Paid | 01-11-2024 | 1000 |
| 2 | Cash | Pending | 02-11-2024 | 400 |
| 3 | Online Transfer | Paid | 03-11-2024 | 700 |
| 4 | Debit Card | Paid | 04-11-2024 | 150 |
| 5 | Debit Card | Paid | 04-11-2024 | 160 |

| cus_id(pk) | cake_id(pk) | order_id(pk | emp_id(pk) | del_id(pk) | pay_id(pk) |
|---|---|---|---|---|---|
| c1 | 1 | 1 | 1 | 1 | 1 |
| c2 | 2 | 2 | 2 | 2 | 2 |
| c3 | 3 | 3 | 3 | 3 | 3 |
| c4 | 4 | 4 | 4 | 4 | 4 |
| c4 | 5 | 5 | 4 | 5 | 5 |

## 3NF

**3NF**

1.Must be in 2NF.
2.Avoid Transitive Dependencies.

1. **Customer Table**

| cus_id(pk) | cust_name | cust_address | cust_phone |
|---|---|---|---|
| c1 | Rohan Sharma | 12 MG Road, Pune | 9123456789 |
| c2 | Sana Patel | 56 Baker Street, Mumbai | 9223456789 |
| c3 | Karan Gupta | 78 Park Avenue, Delhi | 9323456789 |
| c4 | Priya Iyer | 45 Residency Road, Bangalore | 9423456789 |
| c4 | Priya Iyer | 45 Residency Road, Bangalore | 9423456789 |

2. **Cake Table**

| cake_id(pk) | cake_name | categoery | cake_price |
|---|---|---|---|
| 1 | Black Forest Cake | birthday cake | 500 |
| 2 | Pineapple Pastry | pastry | 100 |
| 3 | Chocolate Truffle Cake | chocolate cake | 700 |
| 4 | Almond Cookie | cookies | 50 |
| 5 | Gold Cookie | cookies | 60 |

3. **Order Details Table**

| order_id(pk) | quantity | total_amount | order_status |
|---|---|---|---|
| 1 | 2 | 1000 | Completed |
| 2 | 4 | 400 | Pending |
| 3 | 1 | 700 | Shipped |
| 4 | 3 | 150 | Delivered |
| 5 | 2 | 160 | Delivered |

4.employee table

| emp_id(pk) | emp_name | emp_contact |
|---|---|---|
| 1 | Ajit Pandy | 9123456781 |
| 2 | Dorababu Shinde | 9223456781 |
| 3 | Ovee Karekar | 9823456781 |
| 4 | Ganesh | 9923456781 |
| 4 | Ganesh | 9923456781 |

| cat_id | categoery |
|---|---|
| 1 | birthday cake |
| 2 | pastry |
| 3 | chocolate cake |
| 4 | cookies |
| 4 | cookies |

| cus_id(pk) | cake_id(pk) | order_id(pk | emp_id(pk) | del_id(pk) | pay_id(pk) | cat_id |
|---|---|---|---|---|---|---|
| c1 | 1 | 1 | 1 | 1 | 1 | 1 |
| c2 | 2 | 2 | 2 | 2 | 2 | 2 |
| c3 | 3 | 3 | 3 | 3 | 3 | 3 |
| c4 | 4 | 4 | 4 | 4 | 4 | 4 |
| c4 | 5 | 5 | 4 | 5 | 5 | 4 |

# Database Overview

**Database Name**: cake_shop

**Total Tables**: 8

**Tables in "cake_shop"**: admin , cake , category , customer , delivery , employee , orders , payment.

## 1. Admin Table

Structure:

```
CREATE TABLE Admin (
  admin_id INT PRIMARY KEY,
  admin_name VARCHAR(100),
  admin_contact VARCHAR(15),
  admin_email VARCHAR(100),
  admin_role VARCHAR(50) );
```

Inserted Data:

```
mysql> INSERT INTO Admin (admin_id, admin_name, admin_contact, admin_email, admin_role)
    -> VALUES (1, 'nilam balugade', '9324604591', 'nilambalugade@gmail.com', 'Manager');
Query OK, 1 row affected (0.01 sec)

mysql> select * from admin;
+----------+----------------+---------------+------------------------+------------+
| admin_id | admin_name     | admin_contact | admin_email            | admin_role |
+----------+----------------+---------------+------------------------+------------+
|        1 | nilam balugade | 9324604591    | nilambalugade@gmail.com | Manager    |
+----------+----------------+---------------+------------------------+------------+
1 row in set (0.00 sec)
```

## 2. Category Table

Structure:

```
CREATE TABLE Category (
cat_id INT PRIMARY KEY,
```

cat_name VARCHAR(100));

Inserted Data:

```
mysql> INSERT INTO Category (cat_id, cat_name)
    -> VALUES
    -> (1, 'Cakes'),
    -> (2, 'Pastries'),
    -> (3, 'Cookies'),
    -> (4, 'Cupcakes'),
    -> (5, 'Brownies'),
    -> (6, 'Tarts'),
    -> (7, 'Muffins'),
    -> (8, 'Cheesecakes'),
    -> (9, 'Donuts'),
    -> (10, 'Bread');
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Category;
+--------+-------------+
| cat_id | cat_name    |
+--------+-------------+
|      1 | Cakes       |
|      2 | Pastries    |
|      3 | Cookies     |
|      4 | Cupcakes    |
|      5 | Brownies    |
|      6 | Tarts       |
|      7 | Muffins     |
|      8 | Cheesecakes |
|      9 | Donuts      |
|     10 | Bread       |
+--------+-------------+
10 rows in set (0.00 sec)
```

## 3. Customer Table

Structure:

CREATE TABLE Customer (

cust_id INT PRIMARY KEY,

cust_name VARCHAR(100),

cust_address VARCHAR(255),

cust_phone VARCHAR(15),

cust_email VARCHAR(100),

cust_dob DATE,

admin_id INT,

FOREIGN KEY (admin_id) REFERENCES Admin(admin_id) );

Inserted Data:

```
mysql> INSERT INTO Customer (cust_id, cust_name, cust_address, cust_phone, cust_email, cust_dob, admin_id)
    -> VALUES
    -> (1, 'Rohan Sharma', '12 MG Road, Pune', '9123456789', 'rohan.sharma@example.com', '1993-05-15', 1),
    -> (2, 'Sana Patel', '56 Baker Street, Mumbai', '9223456789', 'sana.patel@example.com', '1990-08-22', 1),
    -> (3, 'Karan Gupta', '78 Park Avenue, Delhi', '9323456789', 'karan.gupta@example.com', '1988-11-30', 1),
    -> (4, 'Priya Iyer', '45 Residency Road, Bangalore', '9423456789', 'priya.iyer@example.com', '1995-03-12', 1),
    -> (5, 'Amit Choudhury', '67 Tulip Street, Kolkata', '9523456789', 'amit.choudhury@example.com', '1991-09-10', 1),
    -> (6, 'Neha Singh', '89 Lakeview Street, Chennai', '9623456789', 'neha.singh@example.com', '1994-04-18', 1),
    -> (7, 'Vikram Malhotra', '12 Rajpath Road, Jaipur', '9723456789', 'vikram.malhotra@example.com', '1987-12-24', 1),
    -> (8, 'Ananya Roy', '23 Maple Avenue, Lucknow', '9823456789', 'ananya.roy@example.com', '1992-06-05', 1),
    -> (9, 'Ravi Kumar', '34 MG Avenue, Hyderabad', '9923456789', 'ravi.kumar@example.com', '1985-01-15', 1),
    -> (10, 'Meera Desai', '78 Orchid Street, Ahmedabad', '9023456789', 'meera.desai@example.com', '1993-02-22', 1);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Customer;
+---------+----------------+-----------------------------+------------+-----------------------------+------------+----------+
| cust_id | cust_name      | cust_address                | cust_phone | cust_email                  | cust_dob   | admin_id |
+---------+----------------+-----------------------------+------------+-----------------------------+------------+----------+
|       1 | Rohan Sharma   | 12 MG Road, Pune            | 9123456789 | rohan.sharma@example.com    | 1993-05-15 |        1 |
|       2 | Sana Patel     | 56 Baker Street, Mumbai     | 9223456789 | sana.patel@example.com      | 1990-08-22 |        1 |
|       3 | Karan Gupta    | 78 Park Avenue, Delhi       | 9323456789 | karan.gupta@example.com     | 1988-11-30 |        1 |
|       4 | Priya Iyer     | 45 Residency Road, Bangalore | 9423456789 | priya.iyer@example.com      | 1995-03-12 |        1 |
|       5 | Amit Choudhury | 67 Tulip Street, Kolkata    | 9523456789 | amit.choudhury@example.com  | 1991-09-10 |        1 |
|       6 | Neha Singh     | 89 Lakeview Street, Chennai | 9623456789 | neha.singh@example.com      | 1994-04-18 |        1 |
|       7 | Vikram Malhotra | 12 Rajpath Road, Jaipur     | 9723456789 | vikram.malhotra@example.com | 1987-12-24 |        1 |
|       8 | Ananya Roy     | 23 Maple Avenue, Lucknow    | 9823456789 | ananya.roy@example.com      | 1992-06-05 |        1 |
|       9 | Ravi Kumar     | 34 MG Avenue, Hyderabad     | 9923456789 | ravi.kumar@example.com      | 1985-01-15 |        1 |
|      10 | Meera Desai    | 78 Orchid Street, Ahmedabad | 9023456789 | meera.desai@example.com     | 1993-02-22 |        1 |
+---------+----------------+-----------------------------+------------+-----------------------------+------------+----------+
10 rows in set (0.00 sec)
```

## 4. Employee Table

Structure:

CREATE TABLE Employee (

emp_id INT PRIMARY KEY,

emp_name VARCHAR(100),

emp_contact VARCHAR(15),

emp_add VARCHAR(255),

emp_salary DECIMAL(10, 2),

admin_id INT,

FOREIGN KEY (admin_id) REFERENCES Admin(admin_id) );

Inserted Data:

```
mysql> INSERT INTO Employee (emp_id, emp_name, emp_contact, emp_add, emp_salary, admin_id)
    -> VALUES
    -> (1, 'Ajit Pandy', '9123456781', '23 Worker Street, Kalyan', 35000.00, 1),
    -> (2, 'Dorababu Shinde', '9223456781', '67 Employee Lane, Kalyan', 40000.00, 1),
    -> (3, 'Ovee Karekar', '9823456781', '23 Rose Avenue, Kalyan', 42000.00, 1),
    -> (4, 'Ganesh Balugade', '9923456781', '50 Circular Road, Kalyan', 33000.00, 1),
    -> (5, 'Shree Balugade', '9023456781', '89 Lotus Street, Kalyan', 37000.00, 1);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select * from Employee;
+--------+-----------------+-------------+--------------------------+------------+----------+
| emp_id | emp_name        | emp_contact | emp_add                  | emp_salary | admin_id |
+--------+-----------------+-------------+--------------------------+------------+----------+
|      1 | Ajit Pandy      | 9123456781  | 23 Worker Street, Kalyan |   35000.00 |        1 |
|      2 | Dorababu Shinde | 9223456781  | 67 Employee Lane, Kalyan |   40000.00 |        1 |
|      3 | Ovee Karekar    | 9823456781  | 23 Rose Avenue, Kalyan   |   42000.00 |        1 |
|      4 | Ganesh Balugade | 9923456781  | 50 Circular Road, Kalyan |   33000.00 |        1 |
|      5 | Shree Balugade  | 9023456781  | 89 Lotus Street, Kalyan  |   37000.00 |        1 |
+--------+-----------------+-------------+--------------------------+------------+----------+
5 rows in set (0.00 sec)
```

## 5. Cake Table

Structure:

```
CREATE TABLE Cake (
    cake_id INT PRIMARY KEY,
    cake_name VARCHAR(100),
    cake_price DECIMAL(10, 2),
    cake_quantity VARCHAR(20),
    cake_desc VARCHAR(255),
    cat_id INT,
    admin_id INT,
    FOREIGN KEY (cat_id) REFERENCES Category(cat_id),
    FOREIGN KEY (admin_id) REFERENCES Admin(admin_id) );
```

Inserted Data:

```
mysql> INSERT INTO Cake (cake_id, cake_name, cake_price, cake_quantity, cake_desc, cat_id, admin_id)
    -> VALUES
    -> (1, 'Black Forest Cake', 500.00, '2 kg', 'Layers of chocolate and cream', 1, 1),
    -> (2, 'Pineapple Pastry', 100.00, '50 pc', 'Delicious pineapple flavor', 2, 1),
    -> (3, 'Chocolate Truffle Cake', 700.00, '1.5 kg', 'Rich truffle with dark chocolate', 1, 1),
    -> (4, 'Almond Cookie', 50.00, '1 kg', 'Crunchy almond delight', 3, 1),
    -> (5, 'Red Velvet Cupcake', 150.00, '30 pc', 'Moist red velvet with cream cheese frosting', 4, 1),
    -> (6, 'Walnut Brownie', 120.00, '40 pc', 'Rich brownie with walnuts', 5, 1),
    -> (7, 'Fruit Tart', 200.00, '1 kg', 'Tart with fresh fruits', 6, 1),
    -> (8, 'Blueberry Muffin', 80.00, '60 pc', 'Moist muffin with blueberries', 7, 1),
    -> (9, 'Classic Cheesecake', 300.00, '1.8 kg', 'Creamy and rich cheesecake', 8, 1),
    -> (10, 'Chocolate Donut', 40.00, '120 pc', 'Soft donut with chocolate glaze', 9, 1);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Cake;
+---------+----------------------+------------+---------------+-----------------------------------------+--------+----------+
| cake_id | cake_name            | cake_price | cake_quantity | cake_desc                               | cat_id | admin_id |
+---------+----------------------+------------+---------------+-----------------------------------------+--------+----------+
|       1 | Black Forest Cake    |     500.00 | 2 kg          | Layers of chocolate and cream           |      1 |        1 |
|       2 | Pineapple Pastry     |     100.00 | 50 pc         | Delicious pineapple flavor              |      2 |        1 |
|       3 | Chocolate Truffle Cake |     700.00 | 1.5 kg        | Rich truffle with dark chocolate        |      1 |        1 |
|       4 | Almond Cookie        |      50.00 | 1 kg          | Crunchy almond delight                  |      3 |        1 |
|       5 | Red Velvet Cupcake   |     150.00 | 30 pc         | Moist red velvet with cream cheese frosting |      4 |        1 |
|       6 | Walnut Brownie       |     120.00 | 40 pc         | Rich brownie with walnuts               |      5 |        1 |
|       7 | Fruit Tart           |     200.00 | 1 kg          | Tart with fresh fruits                  |      6 |        1 |
|       8 | Blueberry Muffin     |      80.00 | 60 pc         | Moist muffin with blueberries           |      7 |        1 |
|       9 | Classic Cheesecake   |     300.00 | 1.8 kg        | Creamy and rich cheesecake              |      8 |        1 |
|      10 | Chocolate Donut      |      40.00 | 120 pc        | Soft donut with chocolate glaze         |      9 |        1 |
+---------+----------------------+------------+---------------+-----------------------------------------+--------+----------+
10 rows in set (0.00 sec)
```

## 6. Orders Table

Structure:

```
CREATE TABLE Orders (
    order_id INT PRIMARY KEY,
    cust_id INT,
    cake_id INT,
    order_date DATE,
    quantity INT,
    total_amount DECIMAL(10, 2),
    order_status VARCHAR(50),
    emp_id INT,
    admin_id INT,
    FOREIGN KEY (cust_id) REFERENCES Customer(cust_id),
    FOREIGN KEY (cake_id) REFERENCES Cake(cake_id),
    FOREIGN KEY (emp_id) REFERENCES Employee(emp_id),
```

FOREIGN KEY (admin_id) REFERENCES Admin(admin_id) );

Inserted Data:

```
mysql> INSERT INTO Orders (order_id, cust_id, cake_id, order_date, quantity, total_amount, order_status, emp_id,
    -> VALUES
    -> (1, 1, 1, '2024-11-01', 2, 1000.00, 'Completed', 1, 1),
    -> (2, 2, 2, '2024-11-02', 4, 400.00, 'Pending', 2, 1),
    -> (3, 3, 3, '2024-11-03', 1, 700.00, 'Shipped', 3, 1),
    -> (4, 4, 4, '2024-11-04', 3, 150.00, 'Delivered', 4, 1),
    -> (5, 5, 5, '2024-11-05', 2, 300.00, 'Processing', 5, 1),
    -> (6, 6, 6, '2024-11-06', 3, 360.00, 'Completed', 4, 1),
    -> (7, 7, 7, '2024-11-07', 1, 200.00, 'Pending', 3, 1),
    -> (8, 8, 8, '2024-11-08', 4, 320.00, 'Shipped', 1, 1),
    -> (9, 9, 9, '2024-11-09', 2, 600.00, 'Completed', 2, 1),
    -> (10, 10, 10, '2024-11-10', 5, 200.00, 'Processing', 2, 1);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Orders;
+----------+---------+---------+------------+----------+--------------+--------------+--------+----------+
| order_id | cust_id | cake_id | order_date | quantity | total_amount | order_status | emp_id | admin_id |
+----------+---------+---------+------------+----------+--------------+--------------+--------+----------+
|        1 |       1 |       1 | 2024-11-01 |        2 |      1000.00 | Completed    |      1 |        1 |
|        2 |       2 |       2 | 2024-11-02 |        4 |       400.00 | Pending      |      2 |        1 |
|        3 |       3 |       3 | 2024-11-03 |        1 |       700.00 | Shipped      |      3 |        1 |
|        4 |       4 |       4 | 2024-11-04 |        3 |       150.00 | Delivered    |      4 |        1 |
|        5 |       5 |       5 | 2024-11-05 |        2 |       300.00 | Processing   |      5 |        1 |
|        6 |       6 |       6 | 2024-11-06 |        3 |       360.00 | Completed    |      4 |        1 |
|        7 |       7 |       7 | 2024-11-07 |        1 |       200.00 | Pending      |      3 |        1 |
|        8 |       8 |       8 | 2024-11-08 |        4 |       320.00 | Shipped      |      1 |        1 |
|        9 |       9 |       9 | 2024-11-09 |        2 |       600.00 | Completed    |      2 |        1 |
|       10 |      10 |      10 | 2024-11-10 |        5 |       200.00 | Processing   |      2 |        1 |
+----------+---------+---------+------------+----------+--------------+--------------+--------+----------+
10 rows in set (0.00 sec)
```

## 7. Payment Table

Structure:

CREATE TABLE Payment (

pay_id INT PRIMARY KEY,

pay_method VARCHAR(50),

pay_status VARCHAR(50),

pay_date DATE,

pay_amount DECIMAL(10, 2),

order_id INT,

admin_id INT,

FOREIGN KEY (order_id) REFERENCES Orders(order_id),

FOREIGN KEY (admin_id) REFERENCES Admin(admin_id) );

Inserted Data:

```
mysql> INSERT INTO Payment (pay_id, pay_method, pay_status, pay_date, pay_amount, order_id, admin_id)
    -> VALUES
    -> (1, 'Credit Card', 'Paid', '2024-11-01', 1000.00, 1, 1),
    -> (2, 'Cash', 'Pending', '2024-11-02', 400.00, 2, 1),
    -> (3, 'Online Transfer', 'Paid', '2024-11-03', 700.00, 3, 1),
    -> (4, 'Debit Card', 'Paid', '2024-11-04', 150.00, 4, 1),
    -> (5, 'Cash', 'Pending', '2024-11-05', 300.00, 5, 1),
    -> (6, 'UPI', 'Paid', '2024-11-06', 360.00, 6, 1),
    -> (7, 'Credit Card', 'Pending', '2024-11-07', 200.00, 7, 1),
    -> (8, 'Cash', 'Paid', '2024-11-08', 320.00, 8, 1),
    -> (9, 'credit Card', 'Paid', '2024-11-09', 600.00, 9, 1),
    -> (10, 'Debit Card','Paid', '2024-11-10', 200.00, 10, 1);
Query OK, 10 rows affected (0.02 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql>
mysql> select * from Payment;
+--------+-----------------+------------+------------+------------+----------+----------+
| pay_id | pay_method      | pay_status | pay_date   | pay_amount | order_id | admin_id |
+--------+-----------------+------------+------------+------------+----------+----------+
|      1 | Credit Card     | Paid       | 2024-11-01 |    1000.00 |        1 |        1 |
|      2 | Cash            | Pending    | 2024-11-02 |     400.00 |        2 |        1 |
|      3 | Online Transfer | Paid       | 2024-11-03 |     700.00 |        3 |        1 |
|      4 | Debit Card      | Paid       | 2024-11-04 |     150.00 |        4 |        1 |
|      5 | Cash            | Pending    | 2024-11-05 |     300.00 |        5 |        1 |
|      6 | UPI             | Paid       | 2024-11-06 |     360.00 |        6 |        1 |
|      7 | Credit Card     | Pending    | 2024-11-07 |     200.00 |        7 |        1 |
|      8 | Cash            | Paid       | 2024-11-08 |     320.00 |        8 |        1 |
|      9 | credit Card     | Paid       | 2024-11-09 |     600.00 |        9 |        1 |
|     10 | Debit Card      | Paid       | 2024-11-10 |     200.00 |       10 |        1 |
+--------+-----------------+------------+------------+------------+----------+----------+
10 rows in set (0.00 sec)
```

## 8. Delivery Table

Structure:

CREATE TABLE Delivery (

del_id INT PRIMARY KEY,

del_person_name VARCHAR(100),

del_date DATE,

del_status VARCHAR(50),

order_id INT,

admin_id INT,

FOREIGN KEY (order_id) REFERENCES Orders(order_id),

FOREIGN KEY (admin_id) REFERENCES Admin(admin_id) );

## Inserted Data:

```
mysql> INSERT INTO Delivery (del_id, del_person_name, del_date, del_status, order_id, admin_id)
    -> VALUES
    -> (1, 'Ajit Pandey', '2024-11-21', 'Delivered', 1, 1),
    -> (2, 'Dorababu Shinde', '2024-11-22', 'Pending', 2, 1),
    -> (3, 'Ajit Pandey', '2024-11-23', 'Out for Delivery', 3, 1),
    -> (4, 'Dorababu Shinde', '2024-11-24', 'Delivered', 4, 1),
    -> (5, 'Ajit Pandey', '2024-11-25', 'Cancelled', 5, 1),
    -> (6, 'Dorababu Shinde', '2024-11-26', 'Pending', 6, 1),
    -> (7, 'Ajit Pandey', '2024-11-27', 'Delivered', 7, 1),
    -> (8, 'Dorababu Shinde', '2024-11-28', 'Out for Delivery', 8, 1),
    -> (9, 'Ajit Pandey', '2024-11-29', 'Pending', 9, 1),
    -> (10, 'Dorababu Shinde', '2024-11-30', 'Delivered', 10, 1);
Query OK, 10 rows affected (0.01 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select * from Delivery;
+--------+-----------------+------------+------------------+----------+----------+
| del_id | del_person_name | del_date   | del_status       | order_id | admin_id |
+--------+-----------------+------------+------------------+----------+----------+
|      1 | Ajit Pandey     | 2024-11-21 | Delivered        |        1 |        1 |
|      2 | Dorababu Shinde | 2024-11-22 | Pending          |        2 |        1 |
|      3 | Ajit Pandey     | 2024-11-23 | Out for Delivery |        3 |        1 |
|      4 | Dorababu Shinde | 2024-11-24 | Delivered        |        4 |        1 |
|      5 | Ajit Pandey     | 2024-11-25 | Cancelled        |        5 |        1 |
|      6 | Dorababu Shinde | 2024-11-26 | Pending          |        6 |        1 |
|      7 | Ajit Pandey     | 2024-11-27 | Delivered        |        7 |        1 |
|      8 | Dorababu Shinde | 2024-11-28 | Out for Delivery |        8 |        1 |
|      9 | Ajit Pandey     | 2024-11-29 | Pending          |        9 |        1 |
|     10 | Dorababu Shinde | 2024-11-30 | Delivered        |       10 |        1 |
+--------+-----------------+------------+------------------+----------+----------+
10 rows in set (0.00 sec)
```

# Constraints in SQL

**PRIMARY KEY:** Uniquely identifies each record in a table.

**FOREIGN KEY:** Establishes a relationship between two tables.

**UNIQUE:** Ensures all values in a column are unique.

**CHECK:** Ensures values meet a specified condition.

**DEFAULT:** Provides a default value for a column when none is specified.

**NOT NULL:** Ensures a column cannot have a NULL value.

**AUTO_INCREMENT:** Automatically increments the value of a column, typically used for primary keys.

## 1. PRIMARY KEY

A primary key is a special column (or set of columns) in a database table that:

- **Uniquely identifies a row**: Every row in the table must have a unique value in this column.
- **No NULL values**: Cannot leave this column empty (NULL).
- **One per table**: A table can only have one primary key.

Ensures that the column(s) have unique values and cannot contain NULL.

Example:

```
CREATE TABLE Cake (
    cake_id INT  primary key , -- Primary Key for unique identification
```

cake_name VARCHAR(100),

cake_price DECIMAL(10, 2),

cake_quantity VARCHAR(20),

cake_desc VARCHAR(255));

```
mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | YES  |     | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

## 2. UNIQUE

A **UNIQUE** constraint makes sure that all values in a column are different.

- **No duplicates**: Each value in that column has to be unique.
- **One NULL allowed**: You can have one empty (NULL) value if needed.

Ensures that all values in a column are unique.

Example:

CREATE TABLE Cake (

cake_id INT  primary key ,

cake_name VARCHAR(100) UNIQUE, --No duplicate cake_name allowed

cake_price DECIMAL(10, 2),

cake_quantity VARCHAR(20),

cake_desc VARCHAR(255));

```
mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | YES  | UNI | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

## 3. NOT NULL

- **Must have a value**: Every row in the column needs to have a value; it cannot be left empty.
- **No NULL values**: You cannot store NULL (empty) in that column.

Ensures that a column cannot contain NULL values.

Example:

CREATE TABLE Cake (

cake_id INT  primary key ,

cake_name VARCHAR(100) UNIQUE,

cake_price DECIMAL(10, 2) NOT NULL , -- Price must be a decimal value, not null

cake_quantity VARCHAR(20),

cake_desc VARCHAR(255));

## 4. CHECK

- **Validates values**: Ensures that values in a column meet a specified condition.

Ensures that the values in a column satisfy a specific condition.
Example:

```
CREATE TABLE Cake (
    cake_id INT  primary key ,
    cake_name VARCHAR(100) UNIQUE,
    cake_price DECIMAL(10, 2) NOT NULL CHECK (cake_price >= 0),  -- Price cannot be NULL and must be non-negative
    cake_quantity VARCHAR(20),
    cake_desc VARCHAR(255));
```

## 5. DEFAULT

- **Automatic value**: Sets a default value for a column if no value is provided.

Provides a default value for a column when no value is provided during an insert.

Example:

```
CREATE TABLE Cake (
    cake_id INT  primary key ,
    cake_name VARCHAR(100) UNIQUE,
    cake_price DECIMAL(10, 2) NOT NULL CHECK (cake_price >= 0),
```

cake_quantity VARCHAR(20) DEFAULT '1 kg’,  -- Default value for quantity is '1 kg'

cake_desc VARCHAR(255));

```
mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | YES  | UNI | NULL    |       |
| cake_price    | decimal(10,2) | NO   |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | 1 kg    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

## 6. FOREIGN KEY

- **Links tables:** Connects a column in one table to a column in another table.
- **Refers to a PRIMARY KEY:** The foreign key column points to the primary key of another table.
- **Ensures data integrity:** Helps maintain the relationship between tables.

Ensures referential integrity by linking a column to the primary key or unique column of another table.

Example:

CREATE TABLE Cake (

cake_id INT  primary key ,

cake_name VARCHAR(100) UNIQUE,

cake_price DECIMAL(10, 2) NOT NULL CHECK (cake_price >= 0),

cake_quantity VARCHAR(20) DEFAULT '1 kg’,

cake_desc VARCHAR(255),

category_id INT, -- Foreign key column for category reference

FOREIGN KEY (category_id) REFERENCES Category(cat_id) ); -- Foreign key constraint

```
mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | YES  | UNI | NULL    |       |
| cake_price    | decimal(10,2) | NO   |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | 1 kg    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| category_id   | int           | YES  | MUL | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

# DDL (Data Definition Language)

**DDL** is a part of SQL used to define and manage database structures. The main commands include:

1. **CREATE**: Used to create a new database object, such as a table.
2. **ALTER**: Used to modify an existing database object, like adding or changing a column.
3. **DROP**: Used to delete a database object, such as a table.
4. **TRUNCATE**: Used to remove all records from a table without deleting the table itself.

Here are the primary DDL commands:

## 1. CREATE

**Used to create new database objects like tables, views, schemas, and databases.**

### 1. Basic Table Creation

```
CREATE TABLE TableName (
column1 datatype,
column2 datatype,
column3 datatype,
 ... );
```

### 2. Simple Table Creation

```
CREATE TABLE Cake (
 cake_id INT,
 cake_name VARCHAR(100),
cake_price   DECIMAL(10, 2),
cake_quantity VARCHAR(20),
cake_desc VARCHAR(255)
);
```

### 3. Table Creation with Constraints

```
CREATE TABLE Cake (
    cake_id INT  primary key , -- Primary Key for unique identification
    cake_name VARCHAR(100) UNIQUE, --No duplicate cake_name allowed
    cake_price DECIMAL(10, 2) NOT NULL ,  -- Price must be a decimal value, not null
    cake_quantity VARCHAR(20) DEFAULT '1 kg', -- Default value for quantity is '1 kg'
    cake_desc VARCHAR(255),
    category_id INT, -- Foreign key column for category reference
    FOREIGN KEY (category_id) REFERENCES Category(cat_id) );  -- Foreign key constraint
```

### 4. Table Creation with Multiple Constraints on a Single Column

```
CREATE TABLE Cake (
    cake_id INT  primary key ,
```

cake_name VARCHAR(100) UNIQUE,

cake_price DECIMAL(10, 2) NOT NULL CHECK (cake_price >= 0), -- Price cannot be NULL and must be non-negative

cake_quantity VARCHAR(20) DEFAULT '1 kg',

cake_desc VARCHAR(255),

category_id INT,

FOREIGN KEY (category_id) REFERENCES Category(cat_id) );

## 2. ALTER

**ALTER TABLE command for adding, modifying, dropping constraints, and changing the structure of a table**

### 1. Add Constraints

#### a. Add a NOT NULL Constraint

column cannot store NULL values.

Query: alter table cake

modify cake_name varchar(100) not null;

Example:

```
mysql> alter table cake
    -> modify cake_name varchar(100) not null;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | YES  |     | NULL    |       |
| cake_name     | varchar(100)  | NO   |     | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

## b. Add a primary key Constraint

that the column or combination of columns uniquely identifies rows in the table.

Query: alter table cake add constraint primary key (cake_id);

Example:

```
mysql> alter table cake
    -> add constraint primary key (cake_id);
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | NO   | UNI | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | 400.00  |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| category_id   | int           | YES  | MUL | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

## c. Add a unique Constraint

that all values in the column are unique.

Query: alter table cake

add constraint unique_cake_name unique (cake_name);

Example:

```
mysql> alter table cake
    -> add constraint unique_cake_name unique (cake_name);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | NO   | UNI | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

d. Add a check Constraint

that values in the column satisfy a specific condition (e.g., cake price should be greater than 0).

Query: alter table cake

add constraint check_price check (cake_price > 0);

Example:

```
mysql> alter table cake
    -> add constraint check_price check (cake_price > 0);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | NO   | UNI | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

## e. Add a default Constraint

This sets a default value for the column if no value is specified during insertion.

Query: alter table cake

alter column cake_price set default 400;

Example:

```
mysql> alter table cake
    -> alter column cake_price set default 400;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | NO   | UNI | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | 400.00  |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

## f. Add a foreign key Constraint

that a column in one table matches a value from the primary key in another table.

Query: alter table cake

add constraint fk_category foreign key (category_id) references category(category_id);

Example:

```
mysql> CREATE TABLE category (
    ->     category_id INT PRIMARY KEY,
    ->     category_name VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> desc category;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| category_id   | int          | NO   | PRI | NULL    |       |
| category_name | varchar(100) | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

```
mysql> ALTER TABLE cake
    -> ADD COLUMN category_id INT;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | NO   | UNI | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | 400.00  |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| category_id   | int           | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> alter table cake
    -> add constraint fk_category foreign key (category_id) references category(category_id);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | NO   | UNI | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | 400.00  |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| category_id   | int           | YES  | MUL | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

## 2. Drop Constraints

### a. Drop a NOT NULL Constraint

The NOT NULL constraint can be removed using the ALTER COLUMN command.

Query: ALTER TABLE Cake MODIFY cake_name VARCHAR(100) NULL;

Example:

```
mysql> ALTER TABLE Cake
    -> MODIFY cake_name VARCHAR(100) NULL;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | YES  | UNI | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | 400.00  |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| category_id   | int           | YES  | MUL | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

## b. Drop a UNIQUE Constraint

If you no longer want the UNIQUE constraint on a column:

Query: alter table cake

drop constraint unique_cake_name;

Example:

```
mysql> alter table cake
    -> drop constraint unique_cake_name;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | YES  |     | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | 400.00  |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| category_id   | int           | YES  | MUL | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

## c. Drop a check Constraint

Query: alter table cake

drop constraint check_price;

Example:

```
mysql> alter table cake
    -> drop constraint check_price;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | YES  | UNI | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | 400.00  |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| category_id   | int           | YES  | MUL | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

## d. Drop a default Constraint

Query: alter table cake

alter column cake_price drop default;

Example:

```
mysql> alter table cake
    -> alter column cake_price drop default;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | YES  |     | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| category_id   | int           | YES  | MUL | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

f. Drop a primary key Constraint

Query: alter table cake drop primary key;

Example:

```
mysql> alter table cake drop primary key;
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   |     | NULL    |       |
| cake_name     | varchar(100)  | YES  |     | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| category_id   | int           | YES  | MUL | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

## 3. Modifying , Renaming Tables and Columns

1. add a column in an existing table

Query: alter table cake add cake_color varchar(20);

Example:

```
mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | NO   | UNI | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| cat_id        | int           | YES  | MUL | NULL    |       |
| admin_id      | int           | YES  | MUL | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
7 rows in set (0.00 sec)

mysql> alter table cake add cake_color varchar(20);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | NO   | UNI | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| cat_id        | int           | YES  | MUL | NULL    |       |
| admin_id      | int           | YES  | MUL | NULL    |       |
| cake_color    | varchar(20)   | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

## 2. Add a column after a specific existing column

Query: alter table cake add cake_size varchar(20) after cake_name;

Example:

```
mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | NO   | UNI | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| cat_id        | int           | YES  | MUL | NULL    |       |
| admin_id      | int           | YES  | MUL | NULL    |       |
| cake_color    | varchar(20)   | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
8 rows in set (0.00 sec)

mysql> alter table cake add cake_size varchar(20) after cake_name;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | NO   | UNI | NULL    |       |
| cake_size     | varchar(20)   | YES  |     | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| cat_id        | int           | YES  | MUL | NULL    |       |
| admin_id      | int           | YES  | MUL | NULL    |       |
| cake_color    | varchar(20)   | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
9 rows in set (0.00 sec)
```

## 3. Add a column at the top (first position) in the table

Query: alter table cake add cake_shape varchar(20) first;

Example:

```
mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | NO   | UNI | NULL    |       |
| cake_size     | varchar(20)   | YES  |     | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| cat_id        | int           | YES  | MUL | NULL    |       |
| admin_id      | int           | YES  | MUL | NULL    |       |
| cake_color    | varchar(20)   | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
9 rows in set (0.00 sec)

mysql> alter table cake add cake_shape varchar(20) first;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_shape    | varchar(20)   | YES  |     | NULL    |       |
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | NO   | UNI | NULL    |       |
| cake_size     | varchar(20)   | YES  |     | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| cat_id        | int           | YES  | MUL | NULL    |       |
| admin_id      | int           | YES  | MUL | NULL    |       |
| cake_color    | varchar(20)   | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
10 rows in set (0.00 sec)
```

## 4. Change the data type of an existing column

Query: alter table cake modify cake_price decimal(15, 2);

Example:

```
mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_shape    | varchar(20)   | YES  |     | NULL    |       |
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | NO   | UNI | NULL    |       |
| cake_size     | varchar(20)   | YES  |     | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| cat_id        | int           | YES  | MUL | NULL    |       |
| admin_id      | int           | YES  | MUL | NULL    |       |
| cake_color    | varchar(20)   | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
10 rows in set (0.00 sec)

mysql> alter table cake modify cake_price decimal(15, 2);
Query OK, 10 rows affected (0.10 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_shape    | varchar(20)   | YES  |     | NULL    |       |
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | NO   | UNI | NULL    |       |
| cake_size     | varchar(20)   | YES  |     | NULL    |       |
| cake_price    | decimal(15,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| cat_id        | int           | YES  | MUL | NULL    |       |
| admin_id      | int           | YES  | MUL | NULL    |       |
| cake_color    | varchar(20)   | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
10 rows in set (0.00 sec)
```

## 5. Change an existing column's name

Query: alter table cake change cake_desc cake_description varchar(255);

Example:

```
mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_shape    | varchar(20)   | YES  |     | NULL    |       |
| cake_id       | int           | NO   | PRI | NULL    |       |
| cake_name     | varchar(100)  | NO   | UNI | NULL    |       |
| cake_size     | varchar(20)   | YES  |     | NULL    |       |
| cake_price    | decimal(15,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| cat_id        | int           | YES  | MUL | NULL    |       |
| admin_id      | int           | YES  | MUL | NULL    |       |
| cake_color    | varchar(20)   | YES  |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
10 rows in set (0.00 sec)

mysql> alter table cake change cake_desc cake_description varchar(255);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| cake_shape       | varchar(20)   | YES  |     | NULL    |       |
| cake_id          | int           | NO   | PRI | NULL    |       |
| cake_name        | varchar(100)  | NO   | UNI | NULL    |       |
| cake_size        | varchar(20)   | YES  |     | NULL    |       |
| cake_price       | decimal(15,2) | YES  |     | NULL    |       |
| cake_quantity    | varchar(20)   | YES  |     | NULL    |       |
| cake_description | varchar(255)  | YES  |     | NULL    |       |
| cat_id           | int           | YES  | MUL | NULL    |       |
| admin_id         | int           | YES  | MUL | NULL    |       |
| cake_color       | varchar(20)   | YES  |     | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
10 rows in set (0.00 sec)
```

## 6. Change both the data type and column name at the same time

Query: alter table cake change cake_quantity cake_weight varchar(20);

Example:

```
mysql> desc cake;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| cake_shape       | varchar(20)   | YES  |     | NULL    |       |
| cake_id          | int           | NO   | PRI | NULL    |       |
| cake_name        | varchar(100)  | NO   | UNI | NULL    |       |
| cake_size        | varchar(20)   | YES  |     | NULL    |       |
| cake_price       | decimal(15,2) | YES  |     | NULL    |       |
| cake_quantity    | varchar(20)   | YES  |     | NULL    |       |
| cake_description | varchar(255)  | YES  |     | NULL    |       |
| cat_id           | int           | YES  | MUL | NULL    |       |
| admin_id         | int           | YES  | MUL | NULL    |       |
| cake_color       | varchar(20)   | YES  |     | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
10 rows in set (0.00 sec)

mysql> alter table cake change cake_quantity cake_weight varchar(20);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| cake_shape       | varchar(20)   | YES  |     | NULL    |       |
| cake_id          | int           | NO   | PRI | NULL    |       |
| cake_name        | varchar(100)  | NO   | UNI | NULL    |       |
| cake_size        | varchar(20)   | YES  |     | NULL    |       |
| cake_price       | decimal(15,2) | YES  |     | NULL    |       |
| cake_weight      | varchar(20)   | YES  |     | NULL    |       |
| cake_description | varchar(255)  | YES  |     | NULL    |       |
| cat_id           | int           | YES  | MUL | NULL    |       |
| admin_id         | int           | YES  | MUL | NULL    |       |
| cake_color       | varchar(20)   | YES  |     | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
10 rows in set (0.00 sec)
```

## 7. Add multiple columns at the same time

Query: alter table cake add cake_rating int(5), add cake_flavor varchar(100);

Example:

```
mysql> desc cake;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| cake_shape       | varchar(20)   | YES  |     | NULL    |       |
| cake_id          | int           | NO   | PRI | NULL    |       |
| cake_name        | varchar(100)  | NO   | UNI | NULL    |       |
| cake_size        | varchar(20)   | YES  |     | NULL    |       |
| cake_price       | decimal(15,2) | YES  |     | NULL    |       |
| cake_weight      | varchar(20)   | YES  |     | NULL    |       |
| cake_description | varchar(255)  | YES  |     | NULL    |       |
| cat_id           | int           | YES  | MUL | NULL    |       |
| admin_id         | int           | YES  | MUL | NULL    |       |
| cake_color       | varchar(20)   | YES  |     | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
10 rows in set (0.00 sec)

mysql> alter table cake add cake_rating int(5), add cake_flavor varchar(100);
Query OK, 0 rows affected, 1 warning (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 1

mysql> desc cake;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| cake_shape       | varchar(20)   | YES  |     | NULL    |       |
| cake_id          | int           | NO   | PRI | NULL    |       |
| cake_name        | varchar(100)  | NO   | UNI | NULL    |       |
| cake_size        | varchar(20)   | YES  |     | NULL    |       |
| cake_price       | decimal(15,2) | YES  |     | NULL    |       |
| cake_weight      | varchar(20)   | YES  |     | NULL    |       |
| cake_description | varchar(255)  | YES  |     | NULL    |       |
| cat_id           | int           | YES  | MUL | NULL    |       |
| admin_id         | int           | YES  | MUL | NULL    |       |
| cake_color       | varchar(20)   | YES  |     | NULL    |       |
| cake_rating      | int           | YES  |     | NULL    |       |
| cake_flavor      | varchar(100)  | YES  |     | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
12 rows in set (0.00 sec)
```

## 8. Rearrange columns

Query: alter table cake modify column cake_shape varchar(20) after cake_color;

Example:

```
mysql> desc cake;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| cake_shape       | varchar(20)   | YES  |     | NULL    |       |
| cake_id          | int           | NO   | PRI | NULL    |       |
| cake_name        | varchar(100)  | NO   | UNI | NULL    |       |
| cake_size        | varchar(20)   | YES  |     | NULL    |       |
| cake_price       | decimal(15,2) | YES  |     | NULL    |       |
| cake_weight      | varchar(20)   | YES  |     | NULL    |       |
| cake_description | varchar(255)  | YES  |     | NULL    |       |
| cat_id           | int           | YES  | MUL | NULL    |       |
| admin_id         | int           | YES  | MUL | NULL    |       |
| cake_color       | varchar(20)   | YES  |     | NULL    |       |
| cake_rating      | int           | YES  |     | NULL    |       |
| cake_flavor      | varchar(100)  | YES  |     | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
12 rows in set (0.00 sec)

mysql> alter table cake modify column cake_shape varchar(20) after cake_color;
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| cake_id          | int           | NO   | PRI | NULL    |       |
| cake_name        | varchar(100)  | NO   | UNI | NULL    |       |
| cake_size        | varchar(20)   | YES  |     | NULL    |       |
| cake_price       | decimal(15,2) | YES  |     | NULL    |       |
| cake_weight      | varchar(20)   | YES  |     | NULL    |       |
| cake_description | varchar(255)  | YES  |     | NULL    |       |
| cat_id           | int           | YES  | MUL | NULL    |       |
| admin_id         | int           | YES  | MUL | NULL    |       |
| cake_color       | varchar(20)   | YES  |     | NULL    |       |
| cake_shape       | varchar(20)   | YES  |     | NULL    |       |
| cake_rating      | int           | YES  |     | NULL    |       |
| cake_flavor      | varchar(100)  | YES  |     | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
12 rows in set (0.00 sec)
```

## 9. Change the name of the table

Query: alter table cake rename to cake_details;

Example:

```
mysql> desc cake;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| cake_id          | int          | NO   | PRI | NULL    |       |
| cake_name        | varchar(100) | NO   | UNI | NULL    |       |
| cake_size        | varchar(20)  | YES  |     | NULL    |       |
| cake_price       | decimal(15,2)| YES  |     | NULL    |       |
| cake_weight      | varchar(20)  | YES  |     | NULL    |       |
| cake_description | varchar(255) | YES  |     | NULL    |       |
| cat_id           | int          | YES  | MUL | NULL    |       |
| admin_id         | int          | YES  | MUL | NULL    |       |
| cake_color       | varchar(20)  | YES  |     | NULL    |       |
| cake_shape       | varchar(20)  | YES  |     | NULL    |       |
| cake_rating      | int          | YES  |     | NULL    |       |
| cake_flavor      | varchar(100) | YES  |     | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
12 rows in set (0.00 sec)

mysql> alter table cake rename to cake_det;
Query OK, 0 rows affected (0.04 sec)

mysql> desc cake_det;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| cake_id          | int          | NO   | PRI | NULL    |       |
| cake_name        | varchar(100) | NO   | UNI | NULL    |       |
| cake_size        | varchar(20)  | YES  |     | NULL    |       |
| cake_price       | decimal(15,2)| YES  |     | NULL    |       |
| cake_weight      | varchar(20)  | YES  |     | NULL    |       |
| cake_description | varchar(255) | YES  |     | NULL    |       |
| cat_id           | int          | YES  | MUL | NULL    |       |
| admin_id         | int          | YES  | MUL | NULL    |       |
| cake_color       | varchar(20)  | YES  |     | NULL    |       |
| cake_shape       | varchar(20)  | YES  |     | NULL    |       |
| cake_rating      | int          | YES  |     | NULL    |       |
| cake_flavor      | varchar(100) | YES  |     | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
12 rows in set (0.00 sec)
```

## 10. Delete a column from an existing table

Query: alter table cake_det drop cake_rating;

Example:

```
mysql> desc cake_det;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| cake_id          | int           | NO   | PRI | NULL    |       |
| cake_name        | varchar(100)  | NO   | UNI | NULL    |       |
| cake_price       | decimal(15,2) | YES  |     | NULL    |       |
| cake_description | varchar(255)  | YES  |     | NULL    |       |
| cat_id           | int           | YES  | MUL | NULL    |       |
| admin_id         | int           | YES  | MUL | NULL    |       |
| cake_rating      | int           | YES  |     | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
7 rows in set (0.00 sec)

mysql> alter table cake_det drop cake_rating;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake_det;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| cake_id          | int           | NO   | PRI | NULL    |       |
| cake_name        | varchar(100)  | NO   | UNI | NULL    |       |
| cake_price       | decimal(15,2) | YES  |     | NULL    |       |
| cake_description | varchar(255)  | YES  |     | NULL    |       |
| cat_id           | int           | YES  | MUL | NULL    |       |
| admin_id         | int           | YES  | MUL | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

## 4.Drop

### 1. Drop a Table

This will delete both the table and its data (structure and content are permanently removed).

Query: drop table cake;

Example:

```
mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   |     | NULL    |       |
| cake_name     | varchar(100)  | YES  |     | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| category_id   | int           | YES  | MUL | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> drop table cake;
Query OK, 0 rows affected (0.02 sec)

mysql> select * from cake;
ERROR 1146 (42S02): Table 'cake_documentation.cake' doesn't exist
mysql> desc cake;
ERROR 1146 (42S02): Table 'cake_documentation.cake' doesn't exist
mysql>
```

## 2. Drop a Column from a Table

This removes a specific column from an existing table. The data in that column is permanently

Query: alter table cake drop column cake_desc;

Example:

```
mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   |     | NULL    |       |
| cake_name     | varchar(100)  | YES  |     | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| cake_desc     | varchar(255)  | YES  |     | NULL    |       |
| category_id   | int           | YES  | MUL | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> alter table cake drop column cake_desc;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cake;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| cake_id       | int           | NO   |     | NULL    |       |
| cake_name     | varchar(100)  | YES  |     | NULL    |       |
| cake_price    | decimal(10,2) | YES  |     | NULL    |       |
| cake_quantity | varchar(20)   | YES  |     | NULL    |       |
| category_id   | int           | YES  | MUL | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

## 3. Drop a Database

This deletes an entire database, including all of its tables and data.

Query: drop database cake_documentation;

Example:

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| cake_documentation |
| cake_shop          |
| cake_shop_query    |
| classicmodels      |
| information_schema |
| mysql              |
| nilam              |
| nilam_practice     |
| performance_schema |
| swiggy             |
| sys                |
+--------------------+
11 rows in set (0.00 sec)

mysql> drop database  cake_documentation;
Query OK, 1 row affected (0.02 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| cake_shop          |
| cake_shop_query    |
| classicmodels      |
| information_schema |
| mysql              |
| nilam              |
| nilam_practice     |
| performance_schema |
| swiggy             |
| sys                |
+--------------------+
10 rows in set (0.00 sec)
```

## 5. TRUNCATE

Truncate Table (Delete all data from a table but keep the structure)

Query: truncate table cake;

Example:

```
mysql> INSERT INTO cake (cake_id, cake_name, cake_price, cake_quantity, cake_desc, category_id)
    -> VALUES (1, 'Chocolate Cake', 500.00, '1 kg', 'A rich chocolate cake with creamy frosting', 1),
    ->        (2, 'Vanilla Cake', 450.00, '1 kg', 'A soft and light vanilla-flavored cake', 2),
    ->        (3, 'Strawberry Cake', 550.00, '1.5 kg', 'A fresh strawberry cake topped with cream', 3);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from cake;
+---------+----------------+------------+---------------+----------------------------------------+-------------+
| cake_id | cake_name      | cake_price | cake_quantity | cake_desc                              | category_id |
+---------+----------------+------------+---------------+----------------------------------------+-------------+
|       1 | Chocolate Cake |     500.00 | 1 kg          | A rich chocolate cake with creamy frosting |           1 |
|       2 | Vanilla Cake   |     450.00 | 1 kg          | A soft and light vanilla-flavored cake |           2 |
|       3 | Strawberry Cake |     550.00 | 1.5 kg        | A fresh strawberry cake topped with cream |           3 |
+---------+----------------+------------+---------------+----------------------------------------+-------------+
3 rows in set (0.00 sec)

mysql> truncate table cake;
Query OK, 0 rows affected (0.05 sec)

mysql> select * from cake;
Empty set (0.00 sec)
```

- **Faster than DELETE**: TRUNCATE does not generate individual row delete operations, making it faster than DELETE.
- **No WHERE Clause**: TRUNCATE will delete all rows in the table. It is not selective.

# DQL (Data Query Language)

DQL is a subset of SQL used to query and retrieve data from a database. The primary command in DQL is:

1. **SELECT**: Used to retrieve data from one or more tables based on specified criteria.

## 1. SELECT:

Retrieve data from one or more tables.

### 1. Basic Selection:

Retrieve all columns from the cake table.

Query: select * from cake;

Example:

```
mysql> select * from cake;
+---------+-----------------------+------------+-----------------------------------------+--------+----------+
| cake_id | cake_name             | cake_price | cake_description                        | cat_id | admin_id |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream           |      1 |        1 |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor              |      2 |        1 |
|       3 | Chocolate Truffle Cake|     700.00 | Rich truffle with dark chocolate        |      1 |        1 |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight                  |      3 |        1 |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting |  4 |        1 |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts               |      5 |        1 |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                  |      6 |        1 |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries           |      7 |        1 |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake              |      8 |        1 |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze         |      9 |        1 |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+
10 rows in set (0.00 sec)
```

### 2. Selecting Specific Columns:

Retrieve only the cake_name and cake_price columns.

Query: select cake_name, cake_price from cake;

Example:

```
mysql> select cake_name, cake_price from cake;
+------------------------+------------+
| cake_name              | cake_price |
+------------------------+------------+
| Black Forest Cake      |     550.00 |
| Pineapple Pastry       |     150.00 |
| Chocolate Truffle Cake |     700.00 |
| Almond Cookie          |      50.00 |
| Red Velvet Cupcake     |     200.00 |
| Walnut Brownie         |     170.00 |
| Fruit Tart             |     250.00 |
| Blueberry Muffin       |      80.00 |
| Classic Cheesecake     |     350.00 |
| Chocolate Donut        |      40.00 |
+------------------------+------------+
10 rows in set (0.00 sec)
```

## 2. WHERE clause

### 1.Simple Condition:

Find cakes with a price greater than 100.

Query: select * from cake where cake_price > 400;

Example:

```
mysql> select * from cake where cake_price > 400;
+---------+------------------------+------------+---------------------------------+--------+----------+
| cake_id | cake_name              | cake_price | cake_description                | cat_id | admin_id |
+---------+------------------------+------------+---------------------------------+--------+----------+
|       1 | Black Forest Cake      |     550.00 | Layers of chocolate and cream   |      1 |        1 |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate |      1 |        1 |
+---------+------------------------+------------+---------------------------------+--------+----------+
2 rows in set (0.00 sec)
```

### 2. Equality Condition:

Retrieve cakes that belong to category 1.

Query: select * from cake where cat_id = 1;

Example:

```
mysql> select * from cake where cat_id = 1;
+---------+-----------------------+------------+---------------------------------+--------+----------+
| cake_id | cake_name             | cake_price | cake_description                | cat_id | admin_id |
+---------+-----------------------+------------+---------------------------------+--------+----------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream   |      1 |        1 |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate |      1 |        1 |
+---------+-----------------------+------------+---------------------------------+--------+----------+
2 rows in set (0.00 sec)
```

## 3.Multiple Conditions with AND:

Find cakes that are priced above 100 and belong to category 1.

Query: select * from cake where cake_price > 100 and cat_id = 1;

Example:

```
mysql> select * from cake where cake_price > 500 and cat_id = 1;
+---------+-----------------------+------------+---------------------------------+--------+----------+
| cake_id | cake_name             | cake_price | cake_description                | cat_id | admin_id |
+---------+-----------------------+------------+---------------------------------+--------+----------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream   |      1 |        1 |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate |      1 |        1 |
+---------+-----------------------+------------+---------------------------------+--------+----------+
2 rows in set (0.00 sec)
```

## 4. Multiple Conditions with OR:

Find cakes that have a quantity of "2 kg" or are priced below 200.

Query: select * from cake where cake_quantity = '2 kg' or cake_price < 200;

Example:

```
mysql> select * from cake where cake_quantity = '2 kg' or cake_price < 200;
+---------+------------------+------------+---------------------------------+--------+----------+---------------+
| cake_id | cake_name        | cake_price | cake_description                | cat_id | admin_id | cake_quantity |
+---------+------------------+------------+---------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake |     550.00 | Layers of chocolate and cream   |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry  |     150.00 | Delicious pineapple flavor      |      2 |        1 | 50 pc         |
|       4 | Almond Cookie     |      50.00 | Crunchy almond delight          |      3 |        1 | 1 kg          |
|       6 | Walnut Brownie    |     170.00 | Rich brownie with walnuts       |      5 |        1 | 40 pc         |
|       8 | Blueberry Muffin  |      80.00 | Moist muffin with blueberries   |      7 |        1 | 60 pc         |
|      10 | Chocolate Donut   |      40.00 | Soft donut with chocolate glaze |      9 |        1 | 120 pc        |
+---------+------------------+------------+---------------------------------+--------+----------+---------------+
6 rows in set (0.00 sec)
```

# 3. Arithmetic, Comparison, and Logical Operators

## 1. Arithmetic Operators

### 1. Addition (+)

Query: select cake_id, cake_name, cake_price, cake_price + 50  AS new_price from cake;

Example:

```
mysql> select cake_id, cake_name,cake_price, cake_price + 50  AS new_price from cake;
+---------+------------------------+------------+-----------+
| cake_id | cake_name              | cake_price | new_price |
+---------+------------------------+------------+-----------+
|       1 | Black Forest Cake      |     550.00 |    600.00 |
|       2 | Pineapple Pastry       |     150.00 |    200.00 |
|       3 | Chocolate Truffle Cake |     700.00 |    750.00 |
|       4 | Almond Cookie          |      50.00 |    100.00 |
|       5 | Red Velvet Cupcake     |     200.00 |    250.00 |
|       6 | Walnut Brownie         |     170.00 |    220.00 |
|       7 | Fruit Tart             |     250.00 |    300.00 |
|       8 | Blueberry Muffin       |      80.00 |    130.00 |
|       9 | Classic Cheesecake     |     350.00 |    400.00 |
|      10 | Chocolate Donut        |      40.00 |     90.00 |
+---------+------------------------+------------+-----------+
10 rows in set (0.00 sec)
```

2. Subtraction (-)

Query: select cake_id, cake_name, cake_price ,cake_price - 50  AS new_price from cake;

Example:

```
mysql> select cake_id, cake_name,cake_price, cake_price - 50  AS new_price from cake;
+---------+------------------------+------------+-----------+
| cake_id | cake_name              | cake_price | new_price |
+---------+------------------------+------------+-----------+
|       1 | Black Forest Cake      |     550.00 |    500.00 |
|       2 | Pineapple Pastry       |     150.00 |    100.00 |
|       3 | Chocolate Truffle Cake |     700.00 |    650.00 |
|       4 | Almond Cookie          |      50.00 |      0.00 |
|       5 | Red Velvet Cupcake     |     200.00 |    150.00 |
|       6 | Walnut Brownie         |     170.00 |    120.00 |
|       7 | Fruit Tart             |     250.00 |    200.00 |
|       8 | Blueberry Muffin       |      80.00 |     30.00 |
|       9 | Classic Cheesecake     |     350.00 |    300.00 |
|      10 | Chocolate Donut        |      40.00 |    -10.00 |
+---------+------------------------+------------+-----------+
10 rows in set (0.00 sec)
```

3. Multiplication (*)

Query: select cake_id, cake_name, cake_price ,cake_price * 50  AS new_price from cake;

Example:

```
mysql> select cake_id, cake_name,cake_price, cake_price * 2  AS new_price from cake;
+---------+-------------------------+------------+-----------+
| cake_id | cake_name               | cake_price | new_price |
+---------+-------------------------+------------+-----------+
|       1 | Black Forest Cake       |     550.00 |   1100.00 |
|       2 | Pineapple Pastry        |     150.00 |    300.00 |
|       3 | Chocolate Truffle Cake  |     700.00 |   1400.00 |
|       4 | Almond Cookie           |      50.00 |    100.00 |
|       5 | Red Velvet Cupcake      |     200.00 |    400.00 |
|       6 | Walnut Brownie          |     170.00 |    340.00 |
|       7 | Fruit Tart              |     250.00 |    500.00 |
|       8 | Blueberry Muffin        |      80.00 |    160.00 |
|       9 | Classic Cheesecake      |     350.00 |    700.00 |
|      10 | Chocolate Donut         |      40.00 |     80.00 |
+---------+-------------------------+------------+-----------+
10 rows in set (0.00 sec)
```

4. Division (/)

Query: select cake_id, cake_name, cake_price ,cake_price / 50  AS new_price from cake;

Example:

```
mysql> select cake_id, cake_name,cake_price, cake_price / 2  AS new_price from cake;
+---------+-------------------------+------------+------------+
| cake_id | cake_name               | cake_price | new_price  |
+---------+-------------------------+------------+------------+
|       1 | Black Forest Cake       |     550.00 | 275.000000 |
|       2 | Pineapple Pastry        |     150.00 |  75.000000 |
|       3 | Chocolate Truffle Cake  |     700.00 | 350.000000 |
|       4 | Almond Cookie           |      50.00 |  25.000000 |
|       5 | Red Velvet Cupcake      |     200.00 | 100.000000 |
|       6 | Walnut Brownie          |     170.00 |  85.000000 |
|       7 | Fruit Tart              |     250.00 | 125.000000 |
|       8 | Blueberry Muffin        |      80.00 |  40.000000 |
|       9 | Classic Cheesecake      |     350.00 | 175.000000 |
|      10 | Chocolate Donut         |      40.00 |  20.000000 |
+---------+-------------------------+------------+------------+
10 rows in set (0.00 sec)
```

5. Modulus (%)

Query: select cake_id, cake_name, cake_price ,cake_price % 50  AS new_price from cake;

Example:

```
mysql> select cake_id, cake_name,cake_price, cake_price % 100  AS new_price from cake;
+---------+------------------------+------------+-----------+
| cake_id | cake_name              | cake_price | new_price |
+---------+------------------------+------------+-----------+
|       1 | Black Forest Cake      |     550.00 |     50.00 |
|       2 | Pineapple Pastry       |     150.00 |     50.00 |
|       3 | Chocolate Truffle Cake |     700.00 |      0.00 |
|       4 | Almond Cookie          |      50.00 |     50.00 |
|       5 | Red Velvet Cupcake     |     200.00 |      0.00 |
|       6 | Walnut Brownie         |     170.00 |     70.00 |
|       7 | Fruit Tart             |     250.00 |     50.00 |
|       8 | Blueberry Muffin       |      80.00 |     80.00 |
|       9 | Classic Cheesecake     |     350.00 |     50.00 |
|      10 | Chocolate Donut        |      40.00 |     40.00 |
+---------+------------------------+------------+-----------+
10 rows in set (0.00 sec)
```

## 2. Comparison Operators

1.Equal (=)

Find cakes priced exactly at 150:

Query: select * from cake where cake_price = 150;

Example:

```
mysql> select * from cake where cake_price = 150;
+---------+-----------------+------------+---------------------------+--------+----------+---------------+
| cake_id | cake_name       | cake_price | cake_description          | cat_id | admin_id | cake_quantity |
+---------+-----------------+------------+---------------------------+--------+----------+---------------+
|       2 | Pineapple Pastry |     150.00 | Delicious pineapple flavor |      2 |        1 | 50 pc         |
+---------+-----------------+------------+---------------------------+--------+----------+---------------+
1 row in set (0.00 sec)
```

2.Not Equal (<> or !=)

Find cakes that are not priced at 150:

Query: select * from cake where cake_price <> 150;

Example:

```
mysql> select * from cake where cake_price <> 150;
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream           |      1 |        1 | 2 kg          |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
9 rows in set (0.00 sec)
```

## 3.Greater Than (>)

Find cakes that cost more than 200:

Query: select * from cake where cake_price > 200;

Example:

```
mysql> select * from cake where cake_price > 200;
+---------+-----------------------+------------+---------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+---------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream   |      1 |        1 | 2 kg          |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate |      1 |        1 | 1.5 kg        |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits          |      6 |        1 | 1 kg          |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake      |      8 |        1 | 1.8 kg        |
+---------+-----------------------+------------+---------------------------------+--------+----------+---------------+
4 rows in set (0.00 sec)
```

## 4.Less Than (<)

Find cakes that cost less than 200:

Query: select * from cake where cake_price < 200;

Example:

```
mysql> select * from cake where cake_price < 200;
+---------+-----------------+------------+---------------------------------+--------+----------+---------------+
| cake_id | cake_name       | cake_price | cake_description                | cat_id | admin_id | cake_quantity |
+---------+-----------------+------------+---------------------------------+--------+----------+---------------+
|       2 | Pineapple Pastry |     150.00 | Delicious pineapple flavor      |      2 |        1 | 50 pc         |
|       4 | Almond Cookie   |      50.00 | Crunchy almond delight          |      3 |        1 | 1 kg          |
|       6 | Walnut Brownie  |     170.00 | Rich brownie with walnuts       |      5 |        1 | 40 pc         |
|       8 | Blueberry Muffin |      80.00 | Moist muffin with blueberries   |      7 |        1 | 60 pc         |
|      10 | Chocolate Donut |      40.00 | Soft donut with chocolate glaze |      9 |        1 | 120 pc        |
+---------+-----------------+------------+---------------------------------+--------+----------+---------------+
5 rows in set (0.00 sec)
```

5.Greater Than or Equal To (>=)

Find cakes priced at 350 or more:

Query: select * from cake where cake_price >= 350;

Example:

```
mysql> select * from cake where cake_price >= 350;
+---------+----------------------+------------+---------------------------------+--------+----------+---------------+
| cake_id | cake_name            | cake_price | cake_description                | cat_id | admin_id | cake_quantity |
+---------+----------------------+------------+---------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake    |     550.00 | Layers of chocolate and cream   |      1 |        1 | 2 kg          |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate |      1 |        1 | 1.5 kg        |
|       9 | Classic Cheesecake   |     350.00 | Creamy and rich cheesecake      |      8 |        1 | 1.8 kg        |
+---------+----------------------+------------+---------------------------------+--------+----------+---------------+
3 rows in set (0.00 sec)
```

6. Less Than or Equal To (<=)

Find cakes priced at 100 or less:

Query: select * from cake where cake_price <= 100;

Example:

```
mysql> select * from cake where cake_price <= 100;
+---------+-----------------+------------+--------------------------------+--------+----------+---------------+
| cake_id | cake_name       | cake_price | cake_description               | cat_id | admin_id | cake_quantity |
+---------+-----------------+------------+--------------------------------+--------+----------+---------------+
|       4 | Almond Cookie   |      50.00 | Crunchy almond delight         |      3 |        1 | 1 kg          |
|       8 | Blueberry Muffin |      80.00 | Moist muffin with blueberries  |      7 |        1 | 60 pc         |
|      10 | Chocolate Donut |      40.00 | Soft donut with chocolate glaze |      9 |        1 | 120 pc        |
+---------+-----------------+------------+--------------------------------+--------+----------+---------------+
3 rows in set (0.00 sec)
```

7. Between

Find cakes with prices between 100 and 300:

Query: select * from cake where cake_price between 100 and 300;

Example:

```
mysql> select * from cake where cake_price between 100 and 300;
+---------+-------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name         | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+-------------------+------------+-----------------------------------------+--------+----------+---------------+
|       2 | Pineapple Pastry  |     150.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       5 | Red Velvet Cupcake|     200.00 | Moist red velvet with cream cheese frosting |  4 |        1 | 30 pc         |
|       6 | Walnut Brownie    |     170.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart        |     250.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
+---------+-------------------+------------+-----------------------------------------+--------+----------+---------------+
4 rows in set (0.00 sec)
```

8. In

Find cakes that have a price of either 150, 250, or 350:

Query: select * from cake where cake_price in (150, 250, 350);

Example:

```
mysql> select * from cake where cake_price in (150, 250, 350);
+---------+-------------------+------------+---------------------------+--------+----------+---------------+
| cake_id | cake_name         | cake_price | cake_description          | cat_id | admin_id | cake_quantity |
+---------+-------------------+------------+---------------------------+--------+----------+---------------+
|       2 | Pineapple Pastry  |     150.00 | Delicious pineapple flavor |      2 |        1 | 50 pc         |
|       7 | Fruit Tart        |     250.00 | Tart with fresh fruits     |      6 |        1 | 1 kg          |
|       9 | Classic Cheesecake|     350.00 | Creamy and rich cheesecake |      8 |        1 | 1.8 kg        |
+---------+-------------------+------------+---------------------------+--------+----------+---------------+
3 rows in set (0.00 sec)
```

## 3.Logical Operators

1. **AND**

Retrieve cakes with a price over 200 *and* in category cat_id 1:

Query: select * from cake where cake_price > 200 and cat_id = 1;

Example:

```
mysql> select * from cake where cake_price > 200 and cat_id = 1;
+---------+---------------------+------------+---------------------------------+--------+----------+---------------+
| cake_id | cake_name           | cake_price | cake_description                | cat_id | admin_id | cake_quantity |
+---------+---------------------+------------+---------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake   |     550.00 | Layers of chocolate and cream   |      1 |        1 | 2 kg          |
|       3 | Chocolate Truffle Cake |   700.00 | Rich truffle with dark chocolate |     1 |        1 | 1.5 kg        |
+---------+---------------------+------------+---------------------------------+--------+----------+---------------+
2 rows in set (0.00 sec)
```

2. **OR**

Retrieve cakes with a price under 100 *or* in category cat_id 2:

Query: select * from cake where cake_price < 100 or cat_id = 2;

Example:

```
mysql> select * from cake where cake_price < 100 or cat_id = 2;
+---------+------------------+------------+---------------------------------+--------+----------+---------------+
| cake_id | cake_name        | cake_price | cake_description                | cat_id | admin_id | cake_quantity |
+---------+------------------+------------+---------------------------------+--------+----------+---------------+
|       2 | Pineapple Pastry |     150.00 | Delicious pineapple flavor      |      2 |        1 | 50 pc         |
|       4 | Almond Cookie    |      50.00 | Crunchy almond delight          |      3 |        1 | 1 kg          |
|       8 | Blueberry Muffin |      80.00 | Moist muffin with blueberries   |      7 |        1 | 60 pc         |
|      10 | Chocolate Donut  |      40.00 | Soft donut with chocolate glaze |      9 |        1 | 120 pc        |
+---------+------------------+------------+---------------------------------+--------+----------+---------------+
4 rows in set (0.00 sec)
```

### 3. **NOT**

Retrieve cakes that are *not* in category cat_id 3:

Query: select * from cake where not cat_id = 3;

Example:

```
mysql> select * from cake where not cat_id = 3;
+---------+-----------------------+------------+-----------------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                              | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+-----------------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream                 |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor                    |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate              |      1 |        1 | 1.5 kg        |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting   |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts                     |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                        |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries                 |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake                    |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze               |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+-----------------------------------------------+--------+----------+---------------+
9 rows in set (0.00 sec)
```

## 4. Range operator

### 1. **BETWEEN**

Find cakes priced between 100 and 300:

Query: select * from cake where cake_price between 100 and 300;

Example:

```
mysql> select * from cake where cake_price between 100 and 300;
+---------+-------------------+------------+-----------------------------------------------+--------+----------+---------------+
| cake_id | cake_name         | cake_price | cake_description                              | cat_id | admin_id | cake_quantity |
+---------+-------------------+------------+-----------------------------------------------+--------+----------+---------------+
|       2 | Pineapple Pastry  |     150.00 | Delicious pineapple flavor                    |      2 |        1 | 50 pc         |
|       5 | Red Velvet Cupcake |     200.00 | Moist red velvet with cream cheese frosting   |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie    |     170.00 | Rich brownie with walnuts                     |      5 |        1 | 40 pc         |
|       7 | Fruit Tart        |     250.00 | Tart with fresh fruits                        |      6 |        1 | 1 kg          |
+---------+-------------------+------------+-----------------------------------------------+--------+----------+---------------+
4 rows in set (0.00 sec)
```

## 2. NOT BETWEEN

Find cakes priced *outside* the range of 100 and 300:

Query: select * from cake where cake_price not between 100 and 300;

Example:

```
mysql> select * from cake where cake_price not between 100 and 300;
+---------+-----------------------+------------+---------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+---------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream   |      1 |        1 | 2 kg          |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight          |      3 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries   |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake      |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+---------------------------------+--------+----------+---------------+
6 rows in set (0.00 sec)
```

## 5. List operator

### 1.IN

Checks if a value matches any value in a specified list.

Query: select * from cake where cat_id in (1, 2);

Example:

```
mysql> select * from cake where cat_id in (1, 2);
+---------+-----------------------+------------+---------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+---------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream   |      1 |        1 | 2 kg          |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate |      1 |        1 | 1.5 kg        |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor      |      2 |        1 | 50 pc         |
+---------+-----------------------+------------+---------------------------------+--------+----------+---------------+
3 rows in set (0.00 sec)
```

### 2.NOT IN

Checks if a value does not match any value in a specified list.

Query: select * from cake where cat_id not in (3, 4);

Example:

```
mysql> select * from cake where cat_id not in (3, 4);
+---------+-----------------------+------------+---------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+---------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream   |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor      |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate |      1 |        1 | 1.5 kg        |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts       |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits          |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries   |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake      |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+---------------------------------+--------+----------+---------------+
8 rows in set (0.00 sec)
```

## 6. Using ORDER BY, DISTINCT and TOP

### 1. ORDER BY

1. To sort the cakes by price in ascending order:

Query: select * from cake order by cake_price;

Example:

```
mysql> select * from cake;
+---------+-----------------------+------------+---------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                      | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+---------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream         |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor            |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate      |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight                |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts             |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries         |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake            |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze       |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+---------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> select * from cake order by cake_price;
+---------+-----------------------+------------+---------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                      | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+---------------------------------------+--------+----------+---------------+
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze       |      9 |        1 | 120 pc        |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight                |      3 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries         |      7 |        1 | 60 pc         |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor            |      2 |        1 | 50 pc         |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts             |      5 |        1 | 40 pc         |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                |      6 |        1 | 1 kg          |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake            |      8 |        1 | 1.8 kg        |
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream         |      1 |        1 | 2 kg          |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate      |      1 |        1 | 1.5 kg        |
+---------+-----------------------+------------+---------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)
```

2. Or to sort by price in descending order:

Query: select * from cake order by cake_price desc;

Example:

```
mysql> select * from cake;
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                           | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream              |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor                 |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate           |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight                     |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts                  |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                     |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries              |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake                 |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze            |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> select * from cake order by cake_price desc;
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                           | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate           |      1 |        1 | 1.5 kg        |
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream              |      1 |        1 | 2 kg          |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake                 |      8 |        1 | 1.8 kg        |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                     |      6 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts                  |      5 |        1 | 40 pc         |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor                 |      2 |        1 | 50 pc         |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries              |      7 |        1 | 60 pc         |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight                     |      3 |        1 | 1 kg          |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze            |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)
```

## 2. DISTINCT

To get a list of unique categories (cat_id) of cakes:

Query: select distinct cat_id from cake;

Example:

```
mysql> select * from cake;
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                           | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream              |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor                 |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate           |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight                     |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts                  |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                     |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries              |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake                 |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze            |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> select distinct cat_id from cake;
+--------+
| cat_id |
+--------+
|      1 |
|      2 |
|      3 |
|      4 |
|      5 |
|      6 |
|      7 |
|      8 |
|      9 |
+--------+
9 rows in set (0.00 sec)
```

## 3. TOP

Get the top 3 most expensive cakes:

1. For SQL Server:

Query: select top 3 * from cake order by cake_price desc;

Example:

2. For MySQL (use LIMIT instead of TOP):

Query: select * from cake cake_price limit 3;

Example:

```
mysql> select * from cake;
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream           |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> select * from cake cake_price limit 3;
+---------+-----------------------+------------+---------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+---------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream   |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor      |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate |      1 |        1 | 1.5 kg        |
+---------+-----------------------+------------+---------------------------------+--------+----------+---------------+
3 rows in set (0.00 sec)
```

## 7. NULL and IS NOT NULL

### 1. IS NULL

To find rows where a column has no value (e.g., cake_description is empty):

Query: select * from cake where cake_description is null;

Example:

```
mysql> select * from cake where cake_description is null;
Empty set (0.00 sec)
```

This returns all cakes that have no description.

## 2. IS NOT NULL

To find rows where a column has a value (e.g., cake_description is filled in):

Query: select * from cake where cake_description is not null;

Example:

```
mysql> select * from cake where cake_description is not null;
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name            | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake    |     550.00 | Layers of chocolate and cream           |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry     |     150.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie        |      50.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake   |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie       |     170.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart           |     250.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin     |      80.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake   |     350.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut      |      40.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)
```

## 8. CASE statement

- `CASE` starts the statement.
- `WHEN` specifies a condition.
- `THEN` specifies the result if the condition is true.
- `ELSE` specifies the result if none of the conditions are true (optional).
- `END` ends the statement.

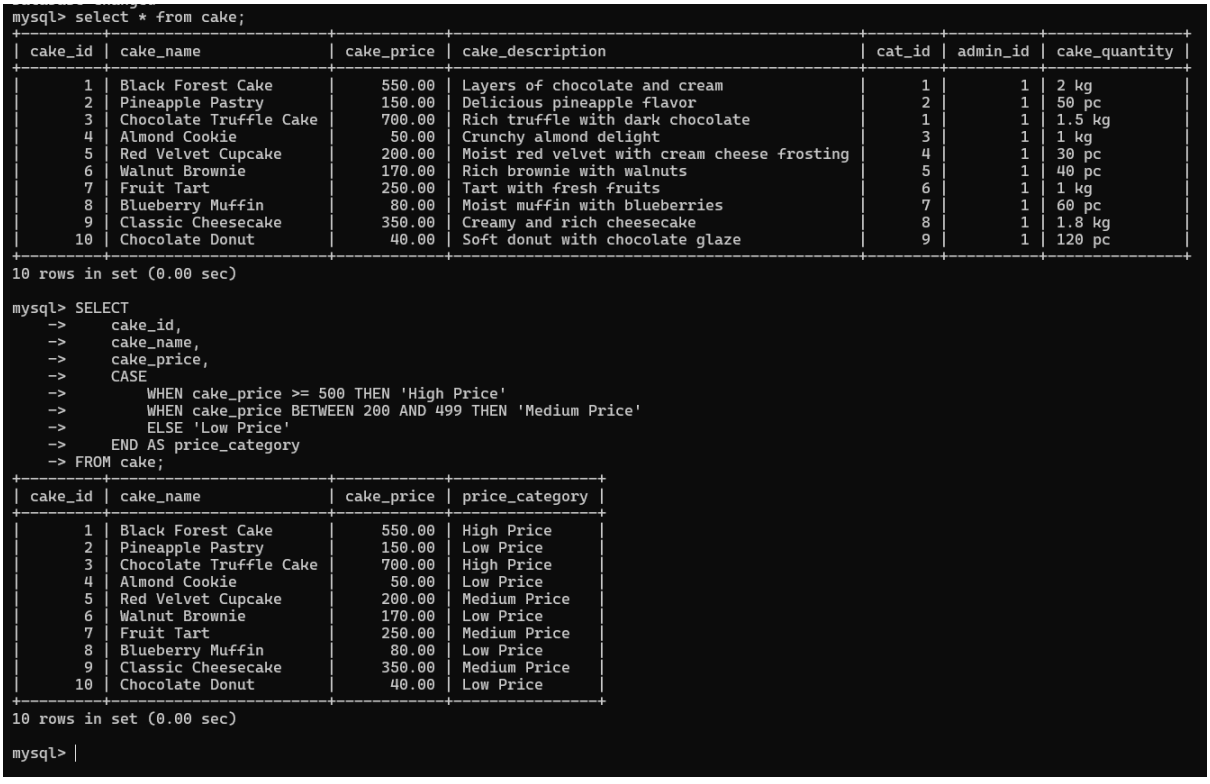
Basic Syntax:

```
CASE
   WHEN condition1 THEN result1
   WHEN condition2 THEN result2
   ...
   ELSE result
END
```

Query: SELECT

```
    cake_id,
    cake_name,
    cake_price,
    CASE
        WHEN cake_price >= 500 THEN 'High Price'
        WHEN cake_price BETWEEN 200 AND 499 THEN 'Medium Price'
        ELSE 'Low Price'
    END AS price_category
FROM cake;
```

Example:



```
mysql> select * from cake;
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream           |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake|     700.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting |  4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> SELECT
    ->     cake_id,
    ->     cake_name,
    ->     cake_price,
    ->     CASE
    ->         WHEN cake_price >= 500 THEN 'High Price'
    ->         WHEN cake_price BETWEEN 200 AND 499 THEN 'Medium Price'
    ->         ELSE 'Low Price'
    ->     END AS price_category
    -> FROM cake;
+---------+-----------------------+------------+----------------+
| cake_id | cake_name             | cake_price | price_category |
+---------+-----------------------+------------+----------------+
|       1 | Black Forest Cake     |     550.00 | High Price     |
|       2 | Pineapple Pastry      |     150.00 | Low Price      |
|       3 | Chocolate Truffle Cake|     700.00 | High Price     |
|       4 | Almond Cookie         |      50.00 | Low Price      |
|       5 | Red Velvet Cupcake    |     200.00 | Medium Price   |
|       6 | Walnut Brownie        |     170.00 | Low Price      |
|       7 | Fruit Tart            |     250.00 | Medium Price   |
|       8 | Blueberry Muffin      |      80.00 | Low Price      |
|       9 | Classic Cheesecake    |     350.00 | Medium Price   |
|      10 | Chocolate Donut       |      40.00 | Low Price      |
+---------+-----------------------+------------+----------------+
10 rows in set (0.00 sec)

mysql>
```

## 9. Like operator

### LIKE Variations

1. % for zero or more characters.
2. _ for exactly one character.
3. Combining % and _ for flexible patterns.
4. Escaping % or _ when they’re literal characters.

## 1. % Wildcard: Match Any Sequence of Characters

1.Match any text containing "cake" anywhere:

Query: SELECT * FROM cake WHERE cake_name LIKE '%cake%';

Example:

```
mysql> SELECT * FROM cake ;
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                           | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream              |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor                 |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake|     700.00 | Rich truffle with dark chocolate           |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight                     |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting|      4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts                  |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                     |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries              |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake                 |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze            |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM cake WHERE cake_name LIKE '%cake%';
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                           | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream              |      1 |        1 | 2 kg          |
|       3 | Chocolate Truffle Cake|     700.00 | Rich truffle with dark chocolate           |      1 |        1 | 1.5 kg        |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting|      4 |        1 | 30 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake                 |      8 |        1 | 1.8 kg        |
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
4 rows in set (0.00 sec)
```

2. Match any text that starts with "Choco":

Query: SELECT * FROM cake WHERE cake_name LIKE 'Choco%';

Example:

```
mysql> SELECT * FROM cake ;
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                           | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream              |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor                 |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate           |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight                     |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts                  |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                     |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries              |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake                 |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze            |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM cake WHERE cake_name LIKE 'Choco%';
+---------+-----------------------+------------+-----------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                  | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+-----------------------------------+--------+----------+---------------+
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze   |      9 |        1 | 120 pc        |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate  |      1 |        1 | 1.5 kg        |
+---------+-----------------------+------------+-----------------------------------+--------+----------+---------------+
2 rows in set (0.00 sec)
```

3. Match any text that ends with "Cake":

Query: SELECT * FROM cake WHERE cake_name LIKE '%Cake';

Example:

```
mysql> SELECT * FROM cake ;
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                           | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream              |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor                 |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate           |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight                     |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts                  |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                     |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries              |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake                 |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze            |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM cake WHERE cake_name LIKE '%Cake';
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                           | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream              |      1 |        1 | 2 kg          |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate           |      1 |        1 | 1.5 kg        |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake                 |      8 |        1 | 1.8 kg        |
+---------+-----------------------+------------+--------------------------------------------+--------+----------+---------------+
4 rows in set (0.00 sec)
```

## 2. _ Wildcard: Match Exactly One Character

1. Match any four-letter word ending in "ake":

Query: SELECT * FROM cake WHERE cake_name LIKE '_ake';

Example:

```
mysql> SELECT * FROM cake ;
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream           |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake|     700.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting |  4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM cake WHERE cake_name LIKE '_ake';
Empty set (0.00 sec)
```

2. Match words where the third character is "o":

Query: SELECT * FROM cake WHERE cake_name LIKE '__o%';

Example:

```
mysql> SELECT * FROM cake ;
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream           |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake|     700.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting |  4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM cake WHERE cake_name LIKE '__o%';
+---------+-----------------------+------------+-----------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                  | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+-----------------------------------+--------+----------+---------------+
|       3 | Chocolate Truffle Cake|     700.00 | Rich truffle with dark chocolate  |      1 |        1 | 1.5 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze   |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+-----------------------------------+--------+----------+---------------+
2 rows in set (0.00 sec)
```

## 3. Combining % and _ for Complex Patterns

You can combine % and _ wildcards for more specific searches.

1. Match words that start with "C" and have exactly five characters:

Query: SELECT * FROM cake WHERE cake_name LIKE 'C____';

Example:

```
mysql> SELECT * FROM cake ;
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name            | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake    |     550.00 | Layers of chocolate and cream           |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry     |     150.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie        |      50.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake   |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie       |     170.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart           |     250.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin     |      80.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake   |     350.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut      |      40.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM cake WHERE cake_name LIKE 'C____';
Empty set (0.00 sec)

mysql> SELECT * FROM cake WHERE cake_name LIKE 'C_______';
Empty set (0.00 sec)

mysql> SELECT * FROM cake WHERE cake_name LIKE 'C_____________';
+---------+---------------+------------+-----------------------------+--------+----------+---------------+
| cake_id | cake_name     | cake_price | cake_description            | cat_id | admin_id | cake_quantity |
+---------+---------------+------------+-----------------------------+--------+----------+---------------+
|      10 | Chocolate Donut |      40.00 | Soft donut with chocolate glaze |      9 |        1 | 120 pc        |
+---------+---------------+------------+-----------------------------+--------+----------+---------------+
1 row in set (0.00 sec)
```

2. Match words that start with "Ch" and contain at least one more character:

Query: SELECT * FROM cake WHERE cake_name LIKE 'Ch_%';

Example:

```
mysql> SELECT * FROM cake ;
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name            | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake    |     550.00 | Layers of chocolate and cream           |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry     |     150.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie        |      50.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake   |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie       |     170.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart           |     250.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin     |      80.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake   |     350.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut      |      40.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM cake WHERE cake_name LIKE 'Ch_%';
+---------+----------------------+------------+-----------------------------+--------+----------+---------------+
| cake_id | cake_name            | cake_price | cake_description            | cat_id | admin_id | cake_quantity |
+---------+----------------------+------------+-----------------------------+--------+----------+---------------+
|      10 | Chocolate Donut      |      40.00 | Soft donut with chocolate glaze |      9 |        1 | 120 pc        |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate |      1 |        1 | 1.5 kg        |
+---------+----------------------+------------+-----------------------------+--------+----------+---------------+
2 rows in set (0.00 sec)
```

3. Match words containing "nut" where it is followed by exactly one character:

Query: SELECT * FROM cake WHERE cake_name LIKE '%nut_';

Example:

```
mysql> SELECT * FROM cake ;
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream           |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake|     700.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting |  4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> SELECT * FROM cake WHERE cake_name LIKE '%nut_';
Empty set (0.00 sec)

mysql> SELECT * FROM cake WHERE cake_name LIKE '%nut';
+---------+----------------+------------+--------------------------------+--------+----------+---------------+
| cake_id | cake_name      | cake_price | cake_description               | cat_id | admin_id | cake_quantity |
+---------+----------------+------------+--------------------------------+--------+----------+---------------+
|      10 | Chocolate Donut |      40.00 | Soft donut with chocolate glaze |      9 |        1 | 120 pc        |
+---------+----------------+------------+--------------------------------+--------+----------+---------------+
1 row in set (0.00 sec)
```

# DML (Data Manipulation Language)

DML is a part of SQL used to manipulate and manage the data stored in a database. The main commands include:

1. **INSERT**: Used to add new records into a table.
2. **UPDATE**: Used to modify existing records in a table.
3. **DELETE**: Used to remove specific records from a table.
4. **SELECT**: Used to retrieve data from one or more tables.

## 1. SELECT:

Retrieves all columns or specific columns from a table.

Query: select * from  cake;

Example:

```
mysql> SELECT * FROM cake;
+---------+-----------------------+------------+-----------------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                              | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+-----------------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     550.00 | Layers of chocolate and cream                 |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry      |     150.00 | Delicious pineapple flavor                    |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake|     700.00 | Rich truffle with dark chocolate              |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      50.00 | Crunchy almond delight                        |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     200.00 | Moist red velvet with cream cheese frosting   |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     170.00 | Rich brownie with walnuts                     |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     250.00 | Tart with fresh fruits                        |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      80.00 | Moist muffin with blueberries                 |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     350.00 | Creamy and rich cheesecake                    |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      40.00 | Soft donut with chocolate glaze               |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+-----------------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)
```

Query: select cake_id,cake_name,cake_price from cake;

Example:

```
mysql> select cake_id,cake_name,cake_price from cake;
+---------+-------------------------+------------+
| cake_id | cake_name               | cake_price |
+---------+-------------------------+------------+
|       1 | Black Forest Cake       |     550.00 |
|       2 | Pineapple Pastry        |     150.00 |
|       3 | Chocolate Truffle Cake  |     700.00 |
|       4 | Almond Cookie           |      50.00 |
|       5 | Red Velvet Cupcake      |     200.00 |
|       6 | Walnut Brownie          |     170.00 |
|       7 | Fruit Tart              |     250.00 |
|       8 | Blueberry Muffin        |      80.00 |
|       9 | Classic Cheesecake      |     350.00 |
|      10 | Chocolate Donut         |      40.00 |
+---------+-------------------------+------------+
10 rows in set (0.00 sec)
```

## 2.Update

### 1. Simple UPDATE

Update a single column in a specific row using the WHERE clause.

Query: update cake set cake_price = 600 where cake_id = 1;

Example:

```
mysql> SELECT * FROM cake;
+---------+-------------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name               | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+-------------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake       |     550.00 | Layers of chocolate and cream           |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry        |     150.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake  |     700.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie           |      50.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake      |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie          |     170.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart              |     250.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin        |      80.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake      |     350.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut         |      40.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+-------------------------+------------+-----------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> update cake set cake_price = 600 where cake_id = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM cake;
+---------+-------------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name               | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+-------------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake       |     600.00 | Layers of chocolate and cream           |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry        |     150.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake  |     700.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie           |      50.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake      |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie          |     170.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart              |     250.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin        |      80.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake      |     350.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut         |      40.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+-------------------------+------------+-----------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)
```

## 2. Update Multiple Columns

Update multiple columns in one statement.

Query: update cake set cake_price = 650, cake_quantity = '3 kg'

where cake_id = 1;

Example:

```
mysql> SELECT * FROM cake;
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name            | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake    |     600.00 | Layers of chocolate and cream           |      1 |        1 | 2 kg          |
|       2 | Pineapple Pastry     |     150.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie        |      50.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake   |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie       |     170.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart           |     250.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin     |      80.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake   |     350.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut      |      40.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> update cake set cake_price = 650, cake_quantity = '3 kg' where cake_id=1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM cake;
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name            | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake    |     650.00 | Layers of chocolate and cream           |      1 |        1 | 3 kg          |
|       2 | Pineapple Pastry     |     150.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |     700.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie        |      50.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake   |     200.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie       |     170.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart           |     250.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin     |      80.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake   |     350.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut      |      40.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)
```

## 3. Conditional Update Using where

Update rows that meet a certain condition. This can target a specific subset of rows.

Query: update cake set cake_price = cake_price * 0.9 where cake_price > 500;

Example:

```
mysql> SELECT * FROM cake;
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name            | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake    |     650.00 | Layers of chocolate and cream           |      1 |        1 | 3 kg          |
|       2 | Pineapple Pastry     |     150.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |   700.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie        |      50.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake   |     200.00 | Moist red velvet with cream cheese frosting |  4 |        1 | 30 pc         |
|       6 | Walnut Brownie       |     170.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart           |     250.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin     |      80.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake   |     350.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut      |      40.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> update cake set cake_price = cake_price * 0.9 where cake_price > 500;
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> SELECT * FROM cake;
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name            | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake    |     585.00 | Layers of chocolate and cream           |      1 |        1 | 3 kg          |
|       2 | Pineapple Pastry     |     150.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |   630.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie        |      50.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake   |     200.00 | Moist red velvet with cream cheese frosting |  4 |        1 | 30 pc         |
|       6 | Walnut Brownie       |     170.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart           |     250.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin     |      80.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake   |     350.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut      |      40.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+----------------------+------------+-----------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)
```

## 3.DELETE

### 1. Simple delete

Delete a specific row based on a condition.

Query: delete from cake_details where ca_id = 1;

Example:

```
mysql> select * from cake_details;
+-------+-------------+-----------------------+----------+-------------+
| ca_id | ca_name     | ca_desc               | ca_price | category_id |
+-------+-------------+-----------------------+----------+-------------+
|     2 | black forest | this is very sweet cake |   400.00 |         101 |
|     4 | chocolavacake | this very sweet cake  |   600.00 |         102 |
|     5 | redvelvetcake | this very sweet cake  |   800.00 |         103 |
+-------+-------------+-----------------------+----------+-------------+
3 rows in set (0.00 sec)

mysql> delete from cake_details where ca_id = 2;
Query OK, 1 row affected (0.01 sec)

mysql> select * from cake_details;
+-------+-------------+-----------------------+----------+-------------+
| ca_id | ca_name     | ca_desc               | ca_price | category_id |
+-------+-------------+-----------------------+----------+-------------+
|     4 | chocolavacake | this very sweet cake  |   600.00 |         102 |
|     5 | redvelvetcake | this very sweet cake  |   800.00 |         103 |
+-------+-------------+-----------------------+----------+-------------+
2 rows in set (0.00 sec)
```

## 2. Delete All Rows in a Table

Deletes all rows without a where clause.

Query: delete from cake_details;

Example:

```
mysql> select * from cake_details;
+-------+--------------+---------------------+----------+-------------+
| ca_id | ca_name      | ca_desc             | ca_price | category_id |
+-------+--------------+---------------------+----------+-------------+
|     4 | chocolavacake | this very sweet cake |   600.00 |         102 |
|     5 | redvelvetcake | this very sweet cake |   800.00 |         103 |
+-------+--------------+---------------------+----------+-------------+
2 rows in set (0.00 sec)

mysql> delete from cake_details;
Query OK, 2 rows affected (0.01 sec)

mysql> select * from cake_details;
Empty set (0.00 sec)

mysql> desc cake_details;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| ca_id       | int          | NO   | PRI | NULL    |       |
| ca_name     | varchar(20)  | YES  |     | NULL    |       |
| ca_desc     | varchar(100) | YES  |     | NULL    |       |
| ca_price    | decimal(10,2) | YES  |     | NULL    |       |
| category_id | int          | YES  | MUL | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

## 4.INSERT:

## 1. Simple insert Statement

Query: insert into cake2 (cake_id, cake_name, cake_price, cake_description, cat_id, admin_id, cake_quantity)

values (11, 'Lemon Cake', 300, 'Fresh lemon-flavored cake', 1, 2, '1 kg');

Example:

```
mysql> create table cake2 (
    ->     cake_id int primary key,
    ->     cake_name varchar(50) not null,
    ->     cake_price decimal(10, 2),
    ->     cake_description text,
    ->     cat_id int,
    ->     admin_id int,
    ->     cake_quantity varchar(20)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> insert into cake2 (cake_id, cake_name, cake_price, cake_description, cat_id, admin_id, cake_quantity)
    -> values (11, 'Lemon Cake', 300, 'Fresh lemon-flavored cake', 1, 2, '1 kg');
Query OK, 1 row affected (0.01 sec)

mysql> select * from cake2;
+---------+------------+------------+--------------------------+--------+----------+---------------+
| cake_id | cake_name  | cake_price | cake_description         | cat_id | admin_id | cake_quantity |
+---------+------------+------------+--------------------------+--------+----------+---------------+
|      11 | Lemon Cake |     300.00 | Fresh lemon-flavored cake |      1 |        2 | 1 kg          |
+---------+------------+------------+--------------------------+--------+----------+---------------+
1 row in set (0.00 sec)
```

## 2. Insert Without Specifying Column Names

Query: insert into cake2 values (12, 'Banana Bread', 180, 'Homemade banana bread', 2, 2, '500 g');

Example:

```
mysql> insert into cake2 values (12, 'Banana Bread', 180, 'Homemade banana bread', 2, 2, '500 g');
Query OK, 1 row affected (0.01 sec)

mysql> select * from cake2;
+---------+--------------+------------+--------------------------+--------+----------+---------------+
| cake_id | cake_name    | cake_price | cake_description         | cat_id | admin_id | cake_quantity |
+---------+--------------+------------+--------------------------+--------+----------+---------------+
|      11 | Lemon Cake   |     300.00 | Fresh lemon-flavored cake |      1 |        2 | 1 kg          |
|      12 | Banana Bread |     180.00 | Homemade banana bread    |      2 |        2 | 500 g         |
+---------+--------------+------------+--------------------------+--------+----------+---------------+
2 rows in set (0.00 sec)
```

## 3. Insert Multiple Rows

Query: insert into cake2 (cake_id, cake_name, cake_price, cake_description, cat_id, admin_id, cake_quantity)

values

(13, 'Carrot Cake', 250, 'Cake with grated carrots', 3, 2, '1.2 kg'),

(14, 'Vanilla Cupcake', 50, 'Classic vanilla flavor', 4, 2, '20 pc');

Example:

```
mysql> insert into cake2 (cake_id, cake_name, cake_price, cake_description, cat_id, admin_id, cake_quantity)
    -> values
    ->     (13, 'Carrot Cake', 250, 'Cake with grated carrots', 3, 2, '1.2 kg'),
    ->     (14, 'Vanilla Cupcake', 50, 'Classic vanilla flavor', 4, 2, '20 pc');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select * from cake2;
+---------+----------------+------------+--------------------------+--------+----------+---------------+
| cake_id | cake_name      | cake_price | cake_description         | cat_id | admin_id | cake_quantity |
+---------+----------------+------------+--------------------------+--------+----------+---------------+
|      11 | Lemon Cake     |     300.00 | Fresh lemon-flavored cake |      1 |        2 | 1 kg          |
|      12 | Banana Bread   |     180.00 | Homemade banana bread     |      2 |        2 | 500 g         |
|      13 | Carrot Cake    |     250.00 | Cake with grated carrots  |      3 |        2 | 1.2 kg        |
|      14 | Vanilla Cupcake |      50.00 | Classic vanilla flavor    |      4 |        2 | 20 pc         |
+---------+----------------+------------+--------------------------+--------+----------+---------------+
4 rows in set (0.00 sec)
```

## 4. Insert Multiple Rows Without Specifying Column Names

Query: insert into cake2 values

(15, 'Carrot Cake', 250, 'Cake with grated carrots', 3, 2, '1.2 kg'),

(16, 'Vanilla Cupcake', 50, 'Classic vanilla flavor', 4, 2, '20 pc');

Example:

```
mysql> insert into cake2 values
    ->     (15, 'Carrot Cake', 250, 'Cake with grated carrots', 3, 2, '1.2 kg'),
    ->     (16, 'Vanilla Cupcake', 50, 'Classic vanilla flavor', 4, 2, '20 pc');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select * from cake2;
+---------+----------------+------------+--------------------------+--------+----------+---------------+
| cake_id | cake_name      | cake_price | cake_description         | cat_id | admin_id | cake_quantity |
+---------+----------------+------------+--------------------------+--------+----------+---------------+
|      11 | Lemon Cake     |     300.00 | Fresh lemon-flavored cake |      1 |        2 | 1 kg          |
|      12 | Banana Bread   |     180.00 | Homemade banana bread     |      2 |        2 | 500 g         |
|      13 | Carrot Cake    |     250.00 | Cake with grated carrots  |      3 |        2 | 1.2 kg        |
|      14 | Vanilla Cupcake |      50.00 | Classic vanilla flavor    |      4 |        2 | 20 pc         |
|      15 | Carrot Cake    |     250.00 | Cake with grated carrots  |      3 |        2 | 1.2 kg        |
|      16 | Vanilla Cupcake |      50.00 | Classic vanilla flavor    |      4 |        2 | 20 pc         |
+---------+----------------+------------+--------------------------+--------+----------+---------------+
6 rows in set (0.00 sec)
```

## 5. Insert with Default Values

Query: insert into cake2 (cake_id, cake_name, cake_price, cake_description, cat_id, admin_id)

values (18, 'Strawberry Cake',250, 'Strawberry-flavored cake', 1, 2);

Example:

```
mysql> alter table cake2
    -> modify cake_quantity varchar(20) default '1 pc';
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> insert into cake2 (cake_id, cake_name, cake_price, cake_description, cat_id, admin_id)
    -> values (18, 'Strawberry Cake',250, 'Strawberry-flavored cake', 1, 2);
Query OK, 1 row affected (0.01 sec)

mysql> select * from cake2;
+---------+-----------------+------------+---------------------------+--------+----------+---------------+
| cake_id | cake_name       | cake_price | cake_description          | cat_id | admin_id | cake_quantity |
+---------+-----------------+------------+---------------------------+--------+----------+---------------+
|      11 | Lemon Cake      |     300.00 | Fresh lemon-flavored cake |      1 |        2 | 1 kg          |
|      12 | Banana Bread    |     180.00 | Homemade banana bread     |      2 |        2 | 500 g         |
|      13 | Carrot Cake     |     250.00 | Cake with grated carrots  |      3 |        2 | 1.2 kg        |
|      14 | Vanilla Cupcake |      50.00 | Classic vanilla flavor    |      4 |        2 | 20 pc         |
|      15 | Carrot Cake     |     250.00 | Cake with grated carrots  |      3 |        2 | 1.2 kg        |
|      16 | Vanilla Cupcake |      50.00 | Classic vanilla flavor    |      4 |        2 | 20 pc         |
|      18 | Strawberry Cake |     250.00 | Strawberry-flavored cake  |      1 |        2 | 1 pc          |
+---------+-----------------+------------+---------------------------+--------+----------+---------------+
7 rows in set (0.00 sec)

mysql> desc cake2;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| cake_id          | int           | NO   | PRI | NULL    |       |
| cake_name        | varchar(50)   | NO   |     | NULL    |       |
| cake_price       | decimal(10,2) | YES  |     | NULL    |       |
| cake_description | text          | YES  |     | NULL    |       |
| cat_id           | int           | YES  |     | NULL    |       |
| admin_id         | int           | YES  |     | NULL    |       |
| cake_quantity    | varchar(20)   | YES  |     | 1 pc    |       |
+------------------+---------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

# Unlock the Built in Functions

## 1. String Function

String functions manipulate or extract information from text values.

### 1.CONCAT(str1, str2, ...):

Concatenates two or more strings together.

Concatenate cake_name and cake_description:

Query: select concat(cake_name, ' - ', cake_description) as cake_info from cake;

Example:

```
mysql> select concat(cake_name, ' - ', cake_description) as cake_info from cake;
+-----------------------------------------------------------------+
| cake_info                                                       |
+-----------------------------------------------------------------+
| Black Forest Cake - Layers of chocolate and cream               |
| Pineapple Pastry - Delicious pineapple flavor                   |
| Chocolate Truffle Cake - Rich truffle with dark chocolate       |
| Almond Cookie - Crunchy almond delight                          |
| Red Velvet Cupcake - Moist red velvet with cream cheese frosting |
| Walnut Brownie - Rich brownie with walnuts                      |
| Fruit Tart - Tart with fresh fruits                             |
| Blueberry Muffin - Moist muffin with blueberries                |
| Classic Cheesecake - Creamy and rich cheesecake                 |
| Chocolate Donut - Soft donut with chocolate glaze               |
+-----------------------------------------------------------------+
10 rows in set (0.00 sec)
```

### 2. UPPER(str):

Converts a string to uppercase.

Convert cake_name to uppercase:

Query:

Example: select upper(cake_name) from cake;

```
mysql> select upper(cake_name) from cake;
+------------------------+
| upper(cake_name)       |
+------------------------+
| ALMOND COOKIE          |
| BLACK FOREST CAKE      |
| BLUEBERRY MUFFIN       |
| CHOCOLATE DONUT        |
| CHOCOLATE TRUFFLE CAKE |
| CLASSIC CHEESECAKE     |
| FRUIT TART             |
| PINEAPPLE PASTRY       |
| RED VELVET CUPCAKE     |
| WALNUT BROWNIE         |
+------------------------+
10 rows in set (0.00 sec)
```

## 3. LOWER(str):

Converts a string to lowercase.

Convert cake_name to lowercase:

Query: select lower(cake_name) from cake;

Example:

```
mysql> select lower(cake_name) from cake;
+-----------------------+
| lower(cake_name)      |
+-----------------------+
| almond cookie         |
| black forest cake     |
| blueberry muffin      |
| chocolate donut       |
| chocolate truffle cake |
| classic cheesecake    |
| fruit tart            |
| pineapple pastry      |
| red velvet cupcake    |
| walnut brownie        |
+-----------------------+
10 rows in set (0.00 sec)
```

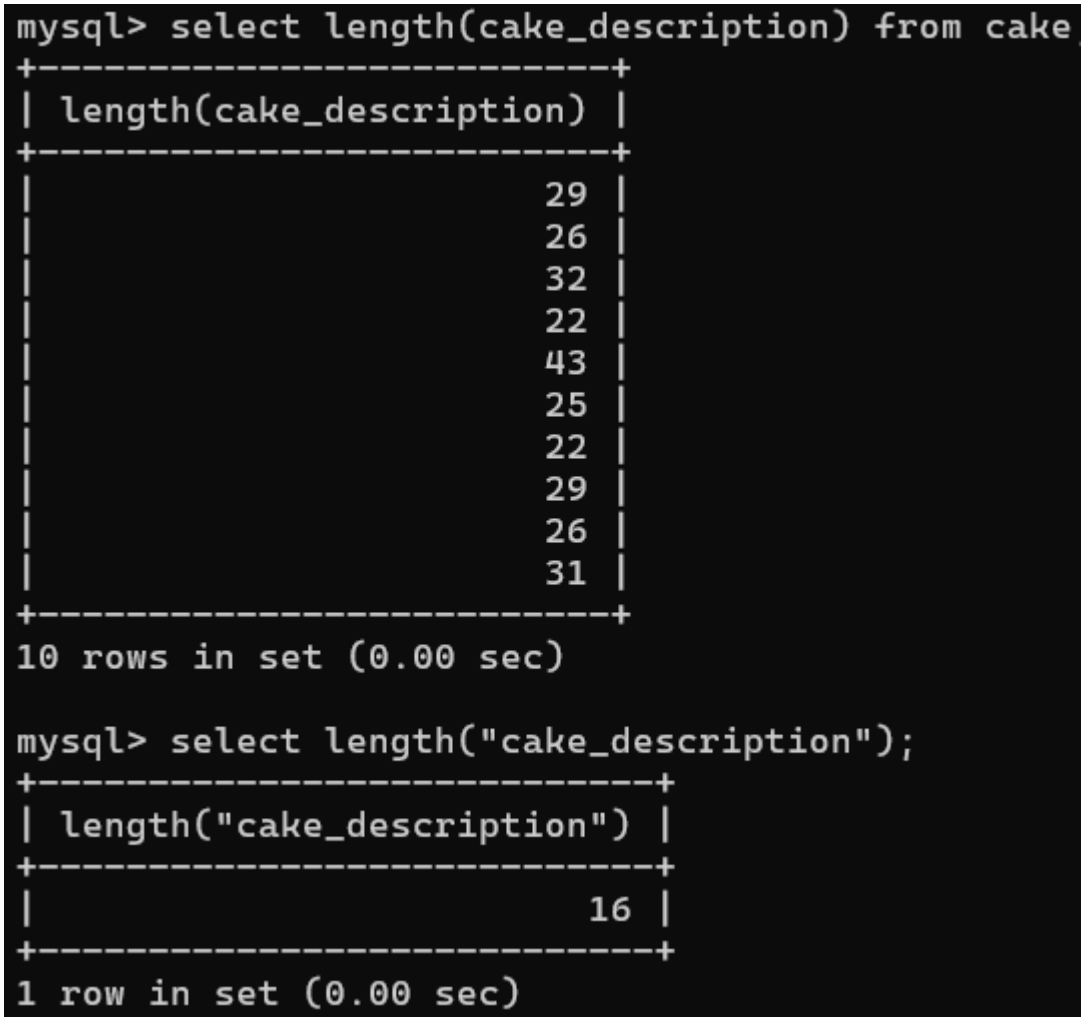
## 4. LENGTH(str):

Returns the length of a string.

Get the length of the cake description:

Query: select length(cake_description) from cake;

Example:



```
mysql> select length(cake_description) from cake
+---------------------------+
| length(cake_description) |
+---------------------------+
|                        29 |
|                        26 |
|                        32 |
|                        22 |
|                        43 |
|                        25 |
|                        22 |
|                        29 |
|                        26 |
|                        31 |
+---------------------------+
10 rows in set (0.00 sec)

mysql> select length("cake_description");
+----------------------------+
| length("cake_description") |
+----------------------------+
|                         16 |
+----------------------------+
1 row in set (0.00 sec)
```

## 5. SUBSTRING(str, start, length):

Extracts a substring from a string.

Extract the first five characters of the cake name:

Query:

Example: select substring(cake_name, 1, 5) from cake;

```
mysql> select substring(cake_name, 1, 5) from cake;
+---------------------------+
| substring(cake_name, 1, 5) |
+---------------------------+
| Almon                     |
| Black                     |
| Blueb                     |
| Choco                     |
| Choco                     |
| Class                     |
| Fruit                     |
| Pinea                     |
| Red V                     |
| Walnu                     |
+---------------------------+
10 rows in set (0.00 sec)

mysql> select substring(cake_name, 2, 5) from cake;
+---------------------------+
| substring(cake_name, 2, 5) |
+---------------------------+
| lmond                     |
| lack                      |
| luebe                     |
| hocol                     |
| hocol                     |
| lassi                     |
| ruit                      |
| ineap                     |
| ed Ve                     |
| alnut                     |
+---------------------------+
10 rows in set (0.00 sec)

mysql> select substring("cake_name", 2, 5);
+-----------------------------+
| substring("cake_name", 2, 5) |
+-----------------------------+
| ake_n                       |
+-----------------------------+
1 row in set (0.00 sec)
```

## 6. Replace

Replace occurrences of 'Cake' with 'Delight' in cake name:

Query: select replace(cake_name, 'Cake', 'Delight') from cake;

Example:

```
mysql> select cake_name from cake;
+-----------------------+
| cake_name             |
+-----------------------+
| Almond Cookie         |
| Black Forest Cake     |
| Blueberry Muffin      |
| Chocolate Donut       |
| Chocolate Truffle Cake |
| Classic Cheesecake    |
| Fruit Tart            |
| Pineapple Pastry      |
| Red Velvet Cupcake    |
| Walnut Brownie        |
+-----------------------+
10 rows in set (0.00 sec)

mysql> select replace(cake_name, 'Cake', 'Delight') from cake;
+--------------------------------------+
| replace(cake_name, 'Cake', 'Delight') |
+--------------------------------------+
| Almond Cookie                        |
| Black Forest Delight                 |
| Blueberry Muffin                     |
| Chocolate Donut                      |
| Chocolate Truffle Delight            |
| Classic Cheesecake                   |
| Fruit Tart                           |
| Pineapple Pastry                     |
| Red Velvet Cupcake                   |
| Walnut Brownie                       |
+--------------------------------------+
10 rows in set (0.00 sec)
```

## 2. Math Function

### 1. ROUND(num, decimal_places):

Rounds a numeric value to a specified number of decimal places.

Round cake_price to one decimal place:

Query: select round(cake_price, 1) from cake;

Example: SELECT cake_price, ROUND(cake_price, 1) AS round_price FROM cake;

```
mysql> SELECT cake_price, ROUND(cake_price, 1) AS round_price FROM cake;
+------------+-------------+
| cake_price | round_price |
+------------+-------------+
|     400.00 |       400.0 |
|     135.00 |       135.0 |
|     363.00 |       363.0 |
|      45.00 |        45.0 |
|     100.00 |       100.0 |
|      73.00 |        73.0 |
|     525.00 |       525.0 |
|     372.00 |       372.0 |
|     615.00 |       615.0 |
|     400.00 |       400.0 |
|     400.00 |       400.0 |
|     400.00 |       400.0 |
+------------+-------------+
12 rows in set (0.00 sec)
```

## 2. **CEIL(num)**:

Rounding the cake price up to the nearest integer:

Query: select cake_price, ceil(cake_price) as ceil_price from cake;

Example:

```
mysql> select cake_price, ceil(cake_price) as ceil_price from cake;
+------------+------------+
| cake_price | ceil_price |
+------------+------------+
|     400.00 |        400 |
|     135.00 |        135 |
|     363.00 |        363 |
|      45.00 |         45 |
|     100.00 |        100 |
|      73.00 |         73 |
|     525.00 |        525 |
|     372.00 |        372 |
|     615.00 |        615 |
|     400.00 |        400 |
|     400.00 |        400 |
|     400.00 |        400 |
+------------+------------+
12 rows in set (0.00 sec)
```

## 3. **FLOOR()**

Rounding the cake price down to the nearest integer:

Query: select cake_price , floor(cake_price) as floor_price from cake;

Example:

```
mysql> select cake_price , floor(cake_price) as floor_price from cake;
+------------+-------------+
| cake_price | floor_price |
+------------+-------------+
|     400.00 |         400 |
|     135.00 |         135 |
|     363.00 |         363 |
|      45.00 |          45 |
|     100.00 |         100 |
|      73.00 |          73 |
|     525.00 |         525 |
|     372.00 |         372 |
|     615.00 |         615 |
|     400.00 |         400 |
|     400.00 |         400 |
|     400.00 |         400 |
+------------+-------------+
12 rows in set (0.00 sec)
```

## 4. **ABS()**

Calculating the absolute value of a negative number:

Query: select cake_price , abs(cake_price) as absolute_price from cake where cake_id=1;

Example:

```
mysql> select cake_price , abs(cake_price) as absolute_price from cake where cake_id=1;
+------------+----------------+
| cake_price | absolute_price |
+------------+----------------+
|     400.00 |         400.00 |
+------------+----------------+
1 row in set (0.00 sec)
```

## 5. **POWER()**

Raising the cake price to a specific exponent:

Query: select cake_price, power(cake_price, 2) as price_squared from cake where cake_id=2;

Example:

```
mysql> select cake_price, power(cake_price, 2) as price_squared from cake where cake_id=2;
+------------+---------------+
| cake_price | price_squared |
+------------+---------------+
|     135.00 |         18225 |
+------------+---------------+
1 row in set (0.00 sec)
```

## 3. Date Function

1. **CURRENT_DATE** – Returns the current date (without time):

Example:

```
mysql> select current_date;
+--------------+
| current_date |
+--------------+
| 2024-11-14   |
+--------------+
1 row in set (0.00 sec)
```

2. **CURRENT_TIME** – Returns the current time (without date):

Example:

```
mysql> select current_time;
+--------------+
| current_time |
+--------------+
| 09:19:51     |
+--------------+
1 row in set (0.00 sec)
```

3. **CURRENT_TIMESTAMP** – Returns the current date and time:

Example:

```
mysql> select current_timestamp;
+---------------------+
| current_timestamp   |
+---------------------+
| 2024-11-14 09:20:26 |
+---------------------+
1 row in set (0.00 sec)
```

**4. DATEDIFF()** – Calculates the difference between two dates

Example:

```
mysql> select datediff('2024-10-5','2030-06-08');
+-------------------------------------+
| datediff('2024-10-5','2030-06-08') |
+-------------------------------------+
|                               -2072 |
+-------------------------------------+
1 row in set (0.00 sec)

mysql> select datediff('2024-10-5','2010-06-08');
+-------------------------------------+
| datediff('2024-10-5','2010-06-08') |
+-------------------------------------+
|                                5233 |
+-------------------------------------+
1 row in set (0.00 sec)
```

## 4. Aggregate Function

**combined query:**

To show the count, sum, average, minimum, and maximum values for cake_price in a single query:

Example:

```
mysql> select
    ->     count(*) as total_cakes,
    ->     sum(cake_price) as total_price,
    ->     avg(cake_price) as average_price,
    ->     min(cake_price) as cheapest_price,
    ->     max(cake_price) as most_expensive_price
    -> from cake;
+-------------+-------------+---------------+---------------+----------------------+
| total_cakes | total_price | average_price | cheapest_price | most_expensive_price |
+-------------+-------------+---------------+---------------+----------------------+
|          12 |     3828.00 |    319.000000 |          45.00 |               615.00 |
+-------------+-------------+---------------+---------------+----------------------+
1 row in set (0.00 sec)
```

## 5.GROUP BY WITH HAVING CLAUSE

1.GROUP BY

**GROUP BY** is used for grouping data based on one or more columns.

They are often used together to perform aggregate functions like COUNT(), SUM(), AVG(), MIN(), MAX() on grouped data

Query: select cake_price,sum(cake_price) as total_price from cake group by cake_price;

Example:

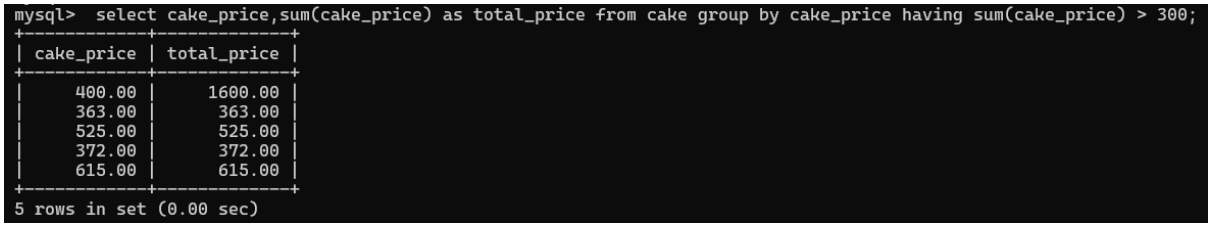
```
mysql> select cake_price,sum(cake_price) as total_price from cake group by cake_price;
+------------+-------------+
| cake_price | total_price |
+------------+-------------+
|     400.00 |     1600.00 |
|     135.00 |      135.00 |
|     363.00 |      363.00 |
|      45.00 |       45.00 |
|     100.00 |      100.00 |
|      73.00 |       73.00 |
|     525.00 |      525.00 |
|     372.00 |      372.00 |
|     615.00 |      615.00 |
+------------+-------------+
9 rows in set (0.00 sec)
```

## 2.GROUP BY WITH HAVING CLAUSE

**HAVING** is used to filter the grouped data based on conditions involving aggregate functions (like SUM(), COUNT(), AVG(), etc.).

Query: select cake_price,sum(cake_price) as total_price from cake group by cake_price having sum(cake_price) > 300;

Example:



```
mysql>  select cake_price,sum(cake_price) as total_price from cake group by cake_price having sum(cake_price) > 300;
+------------+-------------+
| cake_price | total_price |
+------------+-------------+
|     400.00 |     1600.00 |
|     363.00 |      363.00 |
|     525.00 |      525.00 |
|     372.00 |      372.00 |
|     615.00 |      615.00 |
+------------+-------------+
5 rows in set (0.00 sec)
```

# Subquery

A **subquery** is a query within another SQL query.

Types of subquery:

1. Single-Row Subquery:
2. Multi-Row Subquery:

Subqueries can be used in various SQL clauses:

1. **SELECT:** Subqueries can be used to retrieve values for columns or expressions.
2. **FROM:** Subqueries can be used to define virtual tables (derived tables) for the main query.
3. **WHERE:** Subqueries can filter rows based on conditions.
4. **HAVING:** Subqueries can filter groups based on aggregate conditions.
5. **INSERT, UPDATE, DELETE:** Subqueries can also be used in DML statements to manipulate data.

## 1. Single-Row Subquery

Returns only one row as a result.

Often used with comparison operators like =, >, <,=<,>=

Retrieve cakes with a price equal to the highest price in the cake table.

Query: select cake_name, cake_price from cake where cake_price = (select max(cake_price) from cake);

Example:

```
mysql> select * from cake;
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |      58.50 | Layers of chocolate and cream           |      1 |        1 | 3 kg          |
|       2 | Pineapple Pastry      |     135.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |      63.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      45.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     180.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     153.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     225.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      72.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     315.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      36.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> select cake_name, cake_price from cake where cake_price = (select max(cake_price) from cake);
+-------------------+------------+
| cake_name         | cake_price |
+-------------------+------------+
| Classic Cheesecake |     315.00 |
+-------------------+------------+
1 row in set (0.00 sec)
```

## 2. Multi-Row Subquery

Returns multiple rows.

Typically used with operators like IN, ANY, or ALL.

Find cakes where the price matches any price in the cake2 table.

Query: select cake_price from cake where cake_quantity in ('3 kg','1 kg');

Example:

```
mysql> select * from cake;
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |      58.50 | Layers of chocolate and cream           |      1 |        1 | 3 kg          |
|       2 | Pineapple Pastry      |     135.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |      63.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      45.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     180.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |     153.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     225.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |      72.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     315.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |      36.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
10 rows in set (0.00 sec)

mysql> select cake_price from cake where cake_quantity in ('3kg','1kg');
Empty set (0.00 sec)

mysql> select cake_price from cake where cake_quantity in ('3 kg','1 kg');
+------------+
| cake_price |
+------------+
|      58.50 |
|      45.00 |
|     225.00 |
+------------+
3 rows in set (0.00 sec)
```

# Connecting Data for Insights (Joins)

A **JOIN** in SQL is used to combine data from two or more tables based on a related column between them.

1. **INNER JOIN:** Returns only the matching rows from both tables.
2. **LEFT JOIN (LEFT OUTER JOIN):** Returns all rows from the left table and the matching rows from the right table (with NULL for non-matching rows from the right).
3. **RIGHT JOIN (RIGHT OUTER JOIN):** Returns all rows from the right table and the matching rows from the left table (with NULL for non-matching rows from the left).
4. **FULL JOIN (FULL OUTER JOIN):** Returns all rows from both tables, with NULL for missing matches.
5. **CROSS JOIN**: Returns every possible combination of rows between two tables (Cartesian product).
6. **SELF JOIN:** Joins a table with itself, used for hierarchical or self-referential data
7. **NATURAL JOIN:** It automatically joins tables on columns with the same name and same data type.
8. **Equi Join:** Uses exact equality between columns.
9. **Non-Equi Join:** Uses range or inequality comparison between columns.

## 1. inner join

returns only the matching rows from both tables.

Query: select cake.cake_name, cake.cake_price, category.cat_name from cake inner join category on cake.cat_id = category.cat_id;

Example:

```
mysql> select cake.cake_name, cake.cake_price, category.cat_name from cake inner join category on cake.cat_id = category.cat_id;
+-----------------------+------------+-------------+
| cake_name             | cake_price | cat_name    |
+-----------------------+------------+-------------+
| Black Forest Cake     |     358.50 | Cakes       |
| Chocolate Truffle Cake |     363.00 | Cakes       |
| Vanilla Cake          |     400.00 | Cakes       |
| Strawberry Cake       |     420.00 | Cakes       |
| Pineapple Pastry      |     135.00 | Pastries    |
| Almond Cookie         |      45.00 | Cookies     |
| Red Velvet Cupcake    |     100.00 | Cupcakes    |
| Walnut Brownie        |      73.00 | Brownies    |
| Fruit Tart            |     525.00 | Tarts       |
| Blueberry Muffin      |     372.00 | Muffins     |
| Classic Cheesecake    |     615.00 | Cheesecakes |
| Chocolate Donut       |     136.00 | Donuts      |
+-----------------------+------------+-------------+
12 rows in set (0.00 sec)
```

## 2. **left join (left outer join)**

returns all rows from the left table and matching rows from the right table.

Query: select cake.cake_name, cake.cake_price, category.cat_name

from cake left join category on cake.cat_id = category.cat_id;

Example:

```
mysql> select cake.cake_name, cake.cake_price, category.cat_name from cake left join category on cake.cat_id = category.cat_id;
+-----------------------+------------+-------------+
| cake_name             | cake_price | cat_name    |
+-----------------------+------------+-------------+
| Black Forest Cake     |     358.50 | Cakes       |
| Pineapple Pastry      |     135.00 | Pastries    |
| Chocolate Truffle Cake |     363.00 | Cakes       |
| Almond Cookie         |      45.00 | Cookies     |
| Red Velvet Cupcake    |     100.00 | Cupcakes    |
| Walnut Brownie        |      73.00 | Brownies    |
| Fruit Tart            |     525.00 | Tarts       |
| Blueberry Muffin      |     372.00 | Muffins     |
| Classic Cheesecake    |     615.00 | Cheesecakes |
| Chocolate Donut       |     136.00 | Donuts      |
| Vanilla Cake          |     400.00 | Cakes       |
| Strawberry Cake       |     420.00 | Cakes       |
+-----------------------+------------+-------------+
12 rows in set (0.00 sec)
```

## 3. **right join (right outer join)**

returns all rows from the right table and matching rows from the left table.

Query: select cake.cake_name, cake.cake_price, category.cat_name from cake right join category on cake.cat_id = category.cat_id;

Example:

```
mysql> select cake.cake_name, cake.cake_price, category.cat_name from cake right join category on cake.cat_id = category.cat_id;
+-----------------------+------------+-------------+
| cake_name             | cake_price | cat_name    |
+-----------------------+------------+-------------+
| Black Forest Cake     |     358.50 | Cakes       |
| Chocolate Truffle Cake|     363.00 | Cakes       |
| Vanilla Cake          |     400.00 | Cakes       |
| Strawberry Cake       |     420.00 | Cakes       |
| Pineapple Pastry      |     135.00 | Pastries    |
| Almond Cookie         |      45.00 | Cookies     |
| Red Velvet Cupcake    |     100.00 | Cupcakes    |
| Walnut Brownie        |      73.00 | Brownies    |
| Fruit Tart            |     525.00 | Tarts       |
| Blueberry Muffin      |     372.00 | Muffins     |
| Classic Cheesecake    |     615.00 | Cheesecakes |
| Chocolate Donut       |     136.00 | Donuts      |
| NULL                  |       NULL | Bread       |
+-----------------------+------------+-------------+
13 rows in set (0.00 sec)
```

## 4. full join (full outer join)

returns all rows from both tables.

Query: select cake.cake_name, cake.cake_price, category.cat_name from cake left join category on cake.cat_id = category.cat_id union select cake.cake_name, cake.cake_price, category.cat_name from cake right join category on cake.cat_id = category.cat_id;

Example:

```
ysql> select cake.cake_name, cake.cake_price, category.cat_name from cake left join category on cake.cat_id = category.cat_id union
   -> select cake.cake_name, cake.cake_price, category.cat_name from cake right join category on cake.cat_id = category.cat_id;
+-----------------------+------------+-------------+
| cake_name             | cake_price | cat_name    |
+-----------------------+------------+-------------+
| Black Forest Cake     |     358.50 | Cakes       |
| Pineapple Pastry      |     135.00 | Pastries    |
| Chocolate Truffle Cake|     363.00 | Cakes       |
| Almond Cookie         |      45.00 | Cookies     |
| Red Velvet Cupcake    |     100.00 | Cupcakes    |
| Walnut Brownie        |      73.00 | Brownies    |
| Fruit Tart            |     525.00 | Tarts       |
| Blueberry Muffin      |     372.00 | Muffins     |
| Classic Cheesecake    |     615.00 | Cheesecakes |
| Chocolate Donut       |     136.00 | Donuts      |
| Vanilla Cake          |     400.00 | Cakes       |
| Strawberry Cake       |     420.00 | Cakes       |
| NULL                  |       NULL | Bread       |
+-----------------------+------------+-------------+
3 rows in set (0.00 sec)
```

## 5. **cross join**

returns every possible combination of rows between two tables

Query: select cake.cake_name, category.cat_name from cake

cross join category;

Example:

```
mysql>
mysql> select cake.cake_name, category.cat_name from cake cross join category;
+-----------------------+-----------+
| cake_name             | cat_name  |
+-----------------------+-----------+
| Almond Cookie         | Bread     |
| Almond Cookie         | Donuts    |
| Almond Cookie         | Cheesecakes |
| Almond Cookie         | Muffins   |
| Almond Cookie         | Tarts     |
| Almond Cookie         | Brownies  |
| Almond Cookie         | Cupcakes  |
| Almond Cookie         | Cookies   |
| Almond Cookie         | Pastries  |
| Almond Cookie         | Cakes     |
| Black Forest Cake     | Bread     |
| Black Forest Cake     | Donuts    |
| Black Forest Cake     | Cheesecakes |
| Black Forest Cake     | Muffins   |
| Black Forest Cake     | Tarts     |
| Black Forest Cake     | Brownies  |
| Black Forest Cake     | Cupcakes  |
| Black Forest Cake     | Cookies   |
| Black Forest Cake     | Pastries  |
| Black Forest Cake     | Cakes     |
| Blueberry Muffin      | Bread     |
| Blueberry Muffin      | Donuts    |
| Blueberry Muffin      | Cheesecakes |
| Blueberry Muffin      | Muffins   |
| Blueberry Muffin      | Tarts     |
| Blueberry Muffin      | Brownies  |
| Blueberry Muffin      | Cupcakes  |
| Blueberry Muffin      | Cookies   |
| Blueberry Muffin      | Pastries  |
| Blueberry Muffin      | Cakes     |
| Chocolate Donut       | Bread     |
| Chocolate Donut       | Donuts    |
| Chocolate Donut       | Cheesecakes |
| Chocolate Donut       | Muffins   |
| Chocolate Donut       | Tarts     |
| Chocolate Donut       | Brownies  |
| Chocolate Donut       | Cupcakes  |
| Chocolate Donut       | Cookies   |
| Chocolate Donut       | Pastries  |
| Chocolate Donut       | Cakes     |
| Chocolate Truffle Cake | Bread     |
| Chocolate Truffle Cake | Donuts    |
| Chocolate Truffle Cake | Cheesecakes |
| Chocolate Truffle Cake | Muffins   |
| Chocolate Truffle Cake | Tarts     |
| Chocolate Truffle Cake | Brownies  |
| Chocolate Truffle Cake | Cupcakes  |
| Chocolate Truffle Cake | Cookies   |
| Chocolate Truffle Cake | Pastries  |
| Chocolate Truffle Cake | Cakes     |
| Classic Cheesecake    | Bread     |
| Classic Cheesecake    | Donuts    |
| Classic Cheesecake    | Cheesecakes |
| Classic Cheesecake    | Muffins   |
| Classic Cheesecake    | Tarts     |
| Classic Cheesecake    | Brownies  |
| Classic Cheesecake    | Cupcakes  |
| Classic Cheesecake    | Cookies   |
| Classic Cheesecake    | Pastries  |
| Classic Cheesecake    | Cakes     |
| Fruit Tart            | Bread     |
| Fruit Tart            | Donuts    |
| Fruit Tart            | Cheesecakes |
| Fruit Tart            | Muffins   |
| Fruit Tart            | Tarts     |
| Fruit Tart            | Brownies  |
| Fruit Tart            | Cupcakes  |
| Fruit Tart            | Cookies   |
| Fruit Tart            | Pastries  |
| Fruit Tart            | Cakes     |
| Pineapple Pastry      | Bread     |
| Pineapple Pastry      | Donuts    |
| Pineapple Pastry      | Cheesecakes |
| Pineapple Pastry      | Muffins   |
| Pineapple Pastry      | Tarts     |
| Pineapple Pastry      | Brownies  |
| Pineapple Pastry      | Cupcakes  |
| Pineapple Pastry      | Cookies   |
| Pineapple Pastry      | Pastries  |
| Pineapple Pastry      | Cakes     |
| Red Velvet Cupcake    | Bread     |
| Red Velvet Cupcake    | Donuts    |
| Red Velvet Cupcake    | Cheesecakes |
| Red Velvet Cupcake    | Muffins   |
| Red Velvet Cupcake    | Tarts     |
| Red Velvet Cupcake    | Brownies  |
| Red Velvet Cupcake    | Cupcakes  |
| Red Velvet Cupcake    | Cookies   |
| Red Velvet Cupcake    | Pastries  |
| Red Velvet Cupcake    | Cakes     |
| Strawberry Cake       | Bread     |
| Strawberry Cake       | Donuts    |
| Strawberry Cake       | Cheesecakes |
| Strawberry Cake       | Muffins   |
| Strawberry Cake       | Tarts     |
| Strawberry Cake       | Brownies  |
| Strawberry Cake       | Cupcakes  |
| Strawberry Cake       | Cookies   |
| Strawberry Cake       | Pastries  |
| Strawberry Cake       | Cakes     |
| Vanilla Cake          | Bread     |
| Vanilla Cake          | Donuts    |
| Vanilla Cake          | Cheesecakes |
| Vanilla Cake          | Muffins   |
| Vanilla Cake          | Tarts     |
| Vanilla Cake          | Brownies  |
| Vanilla Cake          | Cupcakes  |
| Vanilla Cake          | Cookies   |
| Vanilla Cake          | Pastries  |
| Vanilla Cake          | Cakes     |
| Walnut Brownie        | Bread     |
| Walnut Brownie        | Donuts    |
| Walnut Brownie        | Cheesecakes |
| Walnut Brownie        | Muffins   |
| Walnut Brownie        | Tarts     |
| Walnut Brownie        | Brownies  |
| Walnut Brownie        | Cupcakes  |
| Walnut Brownie        | Cookies   |
| Walnut Brownie        | Pastries  |
| Walnut Brownie        | Cakes     |
+-----------------------+-----------+
120 rows in set (0.00 sec)

mysql>
```

## 6. self join

joins a table with itself.

Query: select e.emp_name as Employee_Name, m.emp_name as Manager_Name

from employees e

join employees m on e.repots_to = m.emp_id;

Example:

```
mysql> select * from employees;
+--------+-----------------+-------------+--------------------------+------------+-----------+
| emp_id | emp_name        | emp_contact | emp_add                  | emp_salary | repots_to |
+--------+-----------------+-------------+--------------------------+------------+-----------+
|      1 | Ajit Pandy      | 9123456781  | 23 Worker Street, Kalyan |   35000.00 |         6 |
|      2 | Dorababu Shinde | 9223456781  | 67 Employee Lane, Kalyan |   40000.00 |         7 |
|      3 | Praju Kedar     | 9823456781  | 23 Rose Avenue, Kalyan   |   42000.00 |         6 |
|      4 | Ganesh Balugade | 9923456781  | 50 Circular Road, Kalyan |   33000.00 |         7 |
|      5 | Shree Balugade  | 9023456781  | 89 Lotus Street, Kalyan  |   37000.00 |         6 |
|      6 | Nilam Balugade  | 9324604591  | Thane                    |    4000.00 |      NULL |
|      7 | Ovee Karekar    | 9324604590  | Thane                    |   50000.00 |      NULL |
+--------+-----------------+-------------+--------------------------+------------+-----------+
7 rows in set (0.00 sec)

mysql> select e.emp_name as Employee_Name, m.emp_name as Manager_Name
    -> from employees e
    -> join employees m on e.repots_to = m.emp_id;
+-----------------+----------------+
| Employee_Name   | Manager_Name   |
+-----------------+----------------+
| Ajit Pandy      | Nilam Balugade |
| Dorababu Shinde | Ovee Karekar   |
| Praju Kedar     | Nilam Balugade |
| Ganesh Balugade | Ovee Karekar   |
| Shree Balugade  | Nilam Balugade |
+-----------------+----------------+
5 rows in set (0.00 sec)
```

## 7. **natural join**

automatically joins tables on columns with the same name and data type.

Query: select * from cake natural join category order by cake_id;

Example:

```
mysql> select * from cake natural join category order by cake_id;
+--------+---------+-----------------------+------------+-----------------------------------------+----------+---------------+-------------+
| cat_id | cake_id | cake_name             | cake_price | cake_description                        | admin_id | cake_quantity | cat_name    |
+--------+---------+-----------------------+------------+-----------------------------------------+----------+---------------+-------------+
|      1 |       1 | Black Forest Cake     |     358.50 | Layers of chocolate and cream           |        1 | 3 kg          | Cakes       |
|      2 |       2 | Pineapple Pastry      |     135.00 | Delicious pineapple flavor              |        1 | 50 pc         | Pastries    |
|      1 |       3 | Chocolate Truffle Cake |     363.00 | Rich truffle with dark chocolate        |        1 | 1.5 kg        | Cakes       |
|      3 |       4 | Almond Cookie         |      45.00 | Crunchy almond delight                  |        1 | 1 kg          | Cookies     |
|      4 |       5 | Red Velvet Cupcake    |     100.00 | Moist red velvet with cream cheese frosting |        1 | 30 pc         | Cupcakes    |
|      5 |       6 | Walnut Brownie        |      73.00 | Rich brownie with walnuts               |        1 | 40 pc         | Brownies    |
|      6 |       7 | Fruit Tart            |     525.00 | Tart with fresh fruits                  |        1 | 1 kg          | Tarts       |
|      7 |       8 | Blueberry Muffin      |     372.00 | Moist muffin with blueberries           |        1 | 60 pc         | Muffins     |
|      8 |       9 | Classic Cheesecake    |     615.00 | Creamy and rich cheesecake              |        1 | 1.8 kg        | Cheesecakes |
|      9 |      10 | Chocolate Donut       |     136.00 | Soft donut with chocolate glaze         |        1 | 120 pc        | Donuts      |
|      1 |      11 | Vanilla Cake          |     400.00 | NULL                                    |        1 | NULL          | Cakes       |
|      1 |      12 | Strawberry Cake       |     420.00 | NULL                                    |        1 | NULL          | Cakes       |
+--------+---------+-----------------------+------------+-----------------------------------------+----------+---------------+-------------+
12 rows in set (0.00 sec)
```

## 8. **equi join**

uses exact equality between columns. (usually =)

Query: select cake.cake_name, category.cat_name from cake join category on cake.cat_id = category.cat_id;

Example:

```
mysql> select cake.cake_name, category.cat_name from cake join category on cake.cat_id = category.cat_id;
+------------------------+-------------+
| cake_name              | cat_name    |
+------------------------+-------------+
| Black Forest Cake      | Cakes       |
| Chocolate Truffle Cake | Cakes       |
| Vanilla Cake           | Cakes       |
| Strawberry Cake        | Cakes       |
| Pineapple Pastry       | Pastries    |
| Almond Cookie          | Cookies     |
| Red Velvet Cupcake     | Cupcakes    |
| Walnut Brownie         | Brownies    |
| Fruit Tart             | Tarts       |
| Blueberry Muffin       | Muffins     |
| Classic Cheesecake     | Cheesecakes |
| Chocolate Donut        | Donuts      |
+------------------------+-------------+
12 rows in set (0.00 sec)
```

## 9. **non-equi join**

uses range or inequality comparison between columns.
(like <, >, <=, >=, !=)

Query: select cake.cake_name, category.cat_name from cake inner join category on cake.cat_id > category.cat_id;

Example:

```
mysql> select cake.cake_name, category.cat_name from cake inner join category on cake.cat_id > category.cat_id;
+--------------------+-------------+
| cake_name          | cat_name    |
+--------------------+-------------+
| Pineapple Pastry   | Cakes       |
| Almond Cookie      | Cakes       |
| Red Velvet Cupcake | Cakes       |
| Walnut Brownie     | Cakes       |
| Fruit Tart         | Cakes       |
| Blueberry Muffin   | Cakes       |
| Classic Cheesecake | Cakes       |
| Chocolate Donut    | Cakes       |
| Almond Cookie      | Pastries    |
| Red Velvet Cupcake | Pastries    |
| Walnut Brownie     | Pastries    |
| Fruit Tart         | Pastries    |
| Blueberry Muffin   | Pastries    |
| Classic Cheesecake | Pastries    |
| Chocolate Donut    | Pastries    |
| Red Velvet Cupcake | Cookies     |
| Walnut Brownie     | Cookies     |
| Fruit Tart         | Cookies     |
| Blueberry Muffin   | Cookies     |
| Classic Cheesecake | Cookies     |
| Chocolate Donut    | Cookies     |
| Walnut Brownie     | Cupcakes    |
| Fruit Tart         | Cupcakes    |
| Blueberry Muffin   | Cupcakes    |
| Classic Cheesecake | Cupcakes    |
| Chocolate Donut    | Cupcakes    |
| Fruit Tart         | Brownies    |
| Blueberry Muffin   | Brownies    |
| Classic Cheesecake | Brownies    |
| Chocolate Donut    | Brownies    |
| Blueberry Muffin   | Tarts       |
| Classic Cheesecake | Tarts       |
| Chocolate Donut    | Tarts       |
| Classic Cheesecake | Muffins     |
| Chocolate Donut    | Muffins     |
| Chocolate Donut    | Cheesecakes |
+--------------------+-------------+
36 rows in set (0.00 sec)
```

# view

A **view** is the SQL query that specifies the data and structure of a view.

Simple View :Simple Views are based on a single table.

Complex View : Complex Views involve multiple tables joined together or include aggregations , subqueries or functions.

1.simple view

Query: create view basic_cake_info as select cake_id, cake_name, cake_price from cake;

Example:

```
mysql> create view basic_cake_info as
    -> select cake_id, cake_name, cake_price
    -> from cake;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from basic_cake_info;
+---------+----------------------+------------+
| cake_id | cake_name            | cake_price |
+---------+----------------------+------------+
|       1 | Black Forest Cake    |     358.50 |
|       2 | Pineapple Pastry     |     135.00 |
|       3 | Chocolate Truffle Cake |     363.00 |
|       4 | Almond Cookie        |      45.00 |
|       5 | Red Velvet Cupcake   |     100.00 |
|       6 | Walnut Brownie       |      73.00 |
|       7 | Fruit Tart           |     525.00 |
|       8 | Blueberry Muffin     |     372.00 |
|       9 | Classic Cheesecake   |     615.00 |
|      10 | Chocolate Donut      |     136.00 |
|      11 | Vanilla Cake         |     400.00 |
|      12 | Strawberry Cake      |     420.00 |
+---------+----------------------+------------+
12 rows in set (0.00 sec)
```

## 2.complex view

Query: create view complex_cake

As

select cake.cake_name, category.cat_name from cake join category on cake.cat_id = category.cat_id;

Example:

```
mysql> create view complex_cake
    -> as
    -> select cake.cake_name, category.cat_name from cake join category on cake.cat_id = category.cat_id;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from complex_cake;
+------------------------+-------------+
| cake_name              | cat_name    |
+------------------------+-------------+
| Black Forest Cake      | Cakes       |
| Chocolate Truffle Cake | Cakes       |
| Vanilla Cake           | Cakes       |
| Strawberry Cake        | Cakes       |
| Pineapple Pastry       | Pastries    |
| Almond Cookie          | Cookies     |
| Red Velvet Cupcake     | Cupcakes    |
| Walnut Brownie         | Brownies    |
| Fruit Tart             | Tarts       |
| Blueberry Muffin       | Muffins     |
| Classic Cheesecake     | Cheesecakes |
| Chocolate Donut        | Donuts      |
+------------------------+-------------+
12 rows in set (0.00 sec)
```

## 3. DML using views

DML statements (INSERT, UPDATE, DELETE) are used to manipulate data in tables.

## 1.insert

Query: create view cake_info as

select cake_id, cake_name, cake_price from cake;

insert into basic_cake_info (cake_id, cake_name, cake_price)

values (101, 'Chocolate Cake', 150);

Example:

```
mysql> create view cake_info as
    -> select cake_id, cake_name, cake_price from cake;
Query OK, 0 rows affected (0.12 sec)

mysql> insert into basic_cake_info (cake_id, cake_name, cake_price)
    -> values (101, 'Chocolate Cake', 150);
Query OK, 1 row affected (0.01 sec)

mysql> select * from cake_info;
+---------+----------------------+------------+
| cake_id | cake_name            | cake_price |
+---------+----------------------+------------+
|       1 | Black Forest Cake    |     358.50 |
|       2 | Pineapple Pastry     |     135.00 |
|       3 | Chocolate Truffle Cake |     363.00 |
|       4 | Almond Cookie        |      45.00 |
|       5 | Red Velvet Cupcake   |     100.00 |
|       6 | Walnut Brownie       |      73.00 |
|       7 | Fruit Tart           |     525.00 |
|       8 | Blueberry Muffin     |     372.00 |
|       9 | Classic Cheesecake   |     615.00 |
|      10 | Chocolate Donut      |     136.00 |
|      11 | Vanilla Cake         |     400.00 |
|      12 | Strawberry Cake      |     420.00 |
|     101 | Chocolate Cake       |     150.00 |
+---------+----------------------+------------+
13 rows in set (0.01 sec)
```

## 2.update

Query: update cake_info set cake_id=13 where cake_price=150;

Example:

```
mysql> select * from cake_info;
+---------+-----------------------+------------+
| cake_id | cake_name             | cake_price |
+---------+-----------------------+------------+
|       1 | Black Forest Cake     |     358.50 |
|       2 | Pineapple Pastry      |     135.00 |
|       3 | Chocolate Truffle Cake |     363.00 |
|       4 | Almond Cookie         |      45.00 |
|       5 | Red Velvet Cupcake    |     100.00 |
|       6 | Walnut Brownie        |      73.00 |
|       7 | Fruit Tart            |     525.00 |
|       8 | Blueberry Muffin      |     372.00 |
|       9 | Classic Cheesecake    |     615.00 |
|      10 | Chocolate Donut       |     136.00 |
|      11 | Vanilla Cake          |     400.00 |
|      12 | Strawberry Cake       |     420.00 |
|     101 | Chocolate Cake        |     150.00 |
+---------+-----------------------+------------+
13 rows in set (0.00 sec)

mysql> update cake_info set cake_id=13 where cake_price=150;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from cake_info;
+---------+-----------------------+------------+
| cake_id | cake_name             | cake_price |
+---------+-----------------------+------------+
|       1 | Black Forest Cake     |     358.50 |
|       2 | Pineapple Pastry      |     135.00 |
|       3 | Chocolate Truffle Cake |     363.00 |
|       4 | Almond Cookie         |      45.00 |
|       5 | Red Velvet Cupcake    |     100.00 |
|       6 | Walnut Brownie        |      73.00 |
|       7 | Fruit Tart            |     525.00 |
|       8 | Blueberry Muffin      |     372.00 |
|       9 | Classic Cheesecake    |     615.00 |
|      10 | Chocolate Donut       |     136.00 |
|      11 | Vanilla Cake          |     400.00 |
|      12 | Strawberry Cake       |     420.00 |
|      13 | Chocolate Cake        |     150.00 |
+---------+-----------------------+------------+
13 rows in set (0.00 sec)
```

## 3.delete

Query: delete from basic_cake_info where cake_id = 13;

Example:

```
mysql> delete from basic_cake_info where cake_id = 13;
Query OK, 1 row affected (0.01 sec)

mysql> select * from cake_info;
+---------+-------------------------+-------------+
| cake_id | cake_name               | cake_price |
+---------+-------------------------+-------------+
|       1 | Black Forest Cake       |     358.50 |
|       2 | Pineapple Pastry        |     135.00 |
|       3 | Chocolate Truffle Cake  |     363.00 |
|       4 | Almond Cookie           |      45.00 |
|       5 | Red Velvet Cupcake      |     100.00 |
|       6 | Walnut Brownie          |      73.00 |
|       7 | Fruit Tart              |     525.00 |
|       8 | Blueberry Muffin        |     372.00 |
|       9 | Classic Cheesecake      |     615.00 |
|      10 | Chocolate Donut         |     136.00 |
|      11 | Vanilla Cake            |     400.00 |
|      12 | Strawberry Cake         |     420.00 |
+---------+-------------------------+-------------+
12 rows in set (0.00 sec)
```

## 4. Constraints in view

### 1.check constraint in view

Query: create view cake_check as select * from cake where cake_price=400 with check option;

Example:

```
mysql> select * from cake;
+---------+-------------------------+------------+-------------------------------------------+--------+----------+---------------+
| cake_id | cake_name               | cake_price | cake_description                          | cat_id | admin_id | cake_quantity |
+---------+-------------------------+------------+-------------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake       |     358.50 | Layers of chocolate and cream             |      1 |        1 | 3 kg          |
|       2 | Pineapple Pastry        |     135.00 | Delicious pineapple flavor                |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake  |     363.00 | Rich truffle with dark chocolate          |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie           |      45.00 | Crunchy almond delight                    |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake      |     100.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie          |      73.00 | Rich brownie with walnuts                 |      5 |        1 | 40 pc         |
|       7 | Fruit Tart              |     525.00 | Tart with fresh fruits                    |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin        |     372.00 | Moist muffin with blueberries             |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake      |     615.00 | Creamy and rich cheesecake                |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut         |     400.00 | Soft donut with chocolate glaze           |      9 |        1 | 120 pc        |
|      11 | Vanilla Cake            |     400.00 | NULL                                      |      1 |        1 | NULL          |
|      12 | Strawberry Cake         |     400.00 | NULL                                      |      1 |        1 | NULL          |
+---------+-------------------------+------------+-------------------------------------------+--------+----------+---------------+
12 rows in set (0.00 sec)

mysql>  create view cake_check
    -> as
    -> select * from cake where cake_price=400 with check option;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from cake_check;
+---------+----------------+------------+-----------------------------------+--------+----------+---------------+
| cake_id | cake_name      | cake_price | cake_description                  | cat_id | admin_id | cake_quantity |
+---------+----------------+------------+-----------------------------------+--------+----------+---------------+
|      10 | Chocolate Donut |     400.00 | Soft donut with chocolate glaze |      9 |        1 | 120 pc        |
|      11 | Vanilla Cake    |     400.00 | NULL                              |      1 |        1 | NULL          |
|      12 | Strawberry Cake |     400.00 | NULL                              |      1 |        1 | NULL          |
+---------+----------------+------------+-----------------------------------+--------+----------+---------------+
3 rows in set (0.00 sec)
```

## 2.unqiue constraint (distinct)

Query: create view cake_unqiue as select distinct cake_price from cake;

Example:

```
mysql> select * from cake;
+---------+----------------------+------------+-------------------------------------------+--------+----------+---------------+
| cake_id | cake_name            | cake_price | cake_description                          | cat_id | admin_id | cake_quantity |
+---------+----------------------+------------+-------------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake    |     358.50 | Layers of chocolate and cream             |      1 |        1 | 3 kg          |
|       2 | Pineapple Pastry     |     135.00 | Delicious pineapple flavor                |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |     363.00 | Rich truffle with dark chocolate          |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie        |      45.00 | Crunchy almond delight                    |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake   |     100.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie       |      73.00 | Rich brownie with walnuts                 |      5 |        1 | 40 pc         |
|       7 | Fruit Tart           |     525.00 | Tart with fresh fruits                    |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin     |     372.00 | Moist muffin with blueberries             |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake   |     615.00 | Creamy and rich cheesecake                |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut      |     400.00 | Soft donut with chocolate glaze           |      9 |        1 | 120 pc        |
|      11 | Vanilla Cake         |     400.00 | NULL                                      |      1 |        1 | NULL          |
|      12 | Strawberry Cake      |     400.00 | NULL                                      |      1 |        1 | NULL          |
+---------+----------------------+------------+-------------------------------------------+--------+----------+---------------+
12 rows in set (0.00 sec)

mysql> create view cake_unqiue
    -> as
    -> select distinct cake_price from cake;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from cake_unqiue;
+------------+
| cake_price |
+------------+
|     358.50 |
|     135.00 |
|     363.00 |
|      45.00 |
|     100.00 |
|      73.00 |
|     525.00 |
|     372.00 |
|     615.00 |
|     400.00 |
+------------+
10 rows in set (0.00 sec)
```

## 3.not null constraint in view

Query: create view cake_not_null1 as select cake_description from cake where cake_description is not null;Example:

```
mysql> select cake_description from cake;
+-------------------------------------------+
| cake_description                          |
+-------------------------------------------+
| Layers of chocolate and cream             |
| Delicious pineapple flavor                |
| Rich truffle with dark chocolate          |
| Crunchy almond delight                    |
| Moist red velvet with cream cheese frosting |
| Rich brownie with walnuts                 |
| Tart with fresh fruits                    |
| Moist muffin with blueberries             |
| Creamy and rich cheesecake                |
| Soft donut with chocolate glaze           |
| NULL                                      |
| NULL                                      |
+-------------------------------------------+
12 rows in set (0.00 sec)

mysql> create view cake_not_null1
    -> as
    -> select cake_description from cake where cake_description is not null;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from cake_not_null1;
+-------------------------------------------+
| cake_description                          |
+-------------------------------------------+
| Layers of chocolate and cream             |
| Delicious pineapple flavor                |
| Rich truffle with dark chocolate          |
| Crunchy almond delight                    |
| Moist red velvet with cream cheese frosting |
| Rich brownie with walnuts                 |
| Tart with fresh fruits                    |
| Moist muffin with blueberries             |
| Creamy and rich cheesecake                |
| Soft donut with chocolate glaze           |
+-------------------------------------------+
10 rows in set (0.01 sec)
```

## 4.default constraint in view

Query: create view default_description1 AS

select cake_id,cake_name,cake_price,ifnull(cake_description, 'No description available') AS cake_description,cat_id,admin_id,

cake_quantity

from cake;

Example:

```
mysql> select * from cake;
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     358.50 | Layers of chocolate and cream           |      1 |        1 | 3 kg          |
|       2 | Pineapple Pastry      |     135.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake|     363.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      45.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     100.00 | Moist red velvet with cream cheese frosting |  4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |      73.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     525.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |     372.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     615.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |     400.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
|      11 | Vanilla Cake          |     400.00 | NULL                                    |      1 |        1 | NULL          |
|      12 | Strawberry Cake       |     400.00 | NULL                                    |      1 |        1 | NULL          |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
12 rows in set (0.00 sec)

mysql> create view default_description1 AS
    -> select
    ->     cake_id,
    ->     cake_name,
    ->     cake_price,
    ->     ifnull(cake_description, 'No description available') AS cake_description,
    ->     cat_id,
    ->     admin_id,
    ->     cake_quantity
    -> from cake;
Query OK, 0 rows affected (0.01 sec)

mysql> select cake_description from default_description1;
+-----------------------------------------+
| cake_description                        |
+-----------------------------------------+
| Layers of chocolate and cream           |
| Delicious pineapple flavor              |
| Rich truffle with dark chocolate        |
| Crunchy almond delight                  |
| Moist red velvet with cream cheese frosting |
| Rich brownie with walnuts               |
| Tart with fresh fruits                  |
| Moist muffin with blueberries           |
| Creamy and rich cheesecake              |
| Soft donut with chocolate glaze         |
| No description available                |
| No description available                |
+-----------------------------------------+
12 rows in set (0.00 sec)
```

# Stored Procedure

A **Stored Procedure** is a set of SQL statements that can be stored and executed on the database server.

## 1.simple store procedure

Query: delimiter $$

```
create procedure get_all_cake()

begin

    select * from cake;

end $$

delimiter ;

call get_all_cake();
```

Example:

```
mysql> delimiter $$
mysql> create procedure get_all_cake()
    -> begin
    -> select * from cake;
    -> end $$
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> call get_all_cake();
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
| cake_id | cake_name             | cake_price | cake_description                        | cat_id | admin_id | cake_quantity |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
|       1 | Black Forest Cake     |     358.50 | Layers of chocolate and cream           |      1 |        1 | 3 kg          |
|       2 | Pineapple Pastry      |     135.00 | Delicious pineapple flavor              |      2 |        1 | 50 pc         |
|       3 | Chocolate Truffle Cake |     363.00 | Rich truffle with dark chocolate        |      1 |        1 | 1.5 kg        |
|       4 | Almond Cookie         |      45.00 | Crunchy almond delight                  |      3 |        1 | 1 kg          |
|       5 | Red Velvet Cupcake    |     100.00 | Moist red velvet with cream cheese frosting |      4 |        1 | 30 pc         |
|       6 | Walnut Brownie        |      73.00 | Rich brownie with walnuts               |      5 |        1 | 40 pc         |
|       7 | Fruit Tart            |     525.00 | Tart with fresh fruits                  |      6 |        1 | 1 kg          |
|       8 | Blueberry Muffin      |     372.00 | Moist muffin with blueberries           |      7 |        1 | 60 pc         |
|       9 | Classic Cheesecake    |     615.00 | Creamy and rich cheesecake              |      8 |        1 | 1.8 kg        |
|      10 | Chocolate Donut       |     400.00 | Soft donut with chocolate glaze         |      9 |        1 | 120 pc        |
|      11 | Vanilla Cake          |     400.00 | NULL                                    |      1 |        1 | NULL          |
|      12 | Strawberry Cake       |     400.00 | NULL                                    |      1 |        1 | NULL          |
+---------+-----------------------+------------+-----------------------------------------+--------+----------+---------------+
12 rows in set (0.00 sec)

Query OK, 0 rows affected (0.11 sec)
```

## 2. in Parameter:

Query: delimiter $$

create procedure cake_id(in cakeid int)

begin

select cake_name,cake_price from cake where cake_id=cakeid;

end $$

Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;

mysql> call cake_id(2);

Example:

```
mysql> delimiter $$
mysql> create procedure cake_id(in cakeid int)
    -> begin
    -> select cake_name,cake_price from cake where cake_id=cakeid;
    -> end $$
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> call cake_id(2);
+------------------+------------+
| cake_name        | cake_price |
+------------------+------------+
| Pineapple Pastry |     135.00 |
+------------------+------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> call cake_id(3);
+-----------------------+------------+
| cake_name             | cake_price |
+-----------------------+------------+
| Chocolate Truffle Cake |     363.00 |
+-----------------------+------------+
1 row in set (0.00 sec)
```

## 3. Out Parameter:

Query: delimiter $$

create procedure cake_price(out max_price decimal(10,2))

begin

select max(cake_price) into max_price from cake;

end $$

Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;

mysql> call cake_price(@max_price);

Query OK, 1 row affected (0.00 sec)

mysql> select @max_price;

Example:

```
mysql> delimiter $$
mysql> create procedure cake_price(out max_price decimal(10,2))
    -> begin
    -> select max(cake_price) into max_price from cake;
    -> end $$
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> call cake_price(@max_price);
Query OK, 1 row affected (0.00 sec)

mysql> select @max_price;
+------------+
| @max_price |
+------------+
|     615.00 |
+------------+
1 row in set (0.00 sec)

mysql>
```

## 4. In and Out Parameter

Query: delimiter $$

```
create procedure cake_in_out(in p_cake_id int, out p_cake_price
decimal(10,2))

begin

    select cake_price

    into p_cake_price

    from cake

    where cake_id = p_cake_id;

end $$

delimiter ;

call cake_in_out(2, @p_cake_price);

select @p_cake_price;
```

Example:

```
mysql> delimiter $$
mysql> create procedure cake_in_out(in p_cake_id int,out p_cake_price decimal(10,2))
    -> begin
    -> select cake_price into p_cake_price from cake where cake_id = p_cake_id;
    -> end $$
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> call cake_in_out(2,@p_cake_price);
Query OK, 1 row affected (0.00 sec)

mysql> select @p_cake_price;
+---------------+
| @p_cake_price |
+---------------+
|        135.00 |
+---------------+
1 row in set (0.00 sec)

mysql> call cake_in_out(5,@p_cake_price);
Query OK, 1 row affected (0.00 sec)

mysql> select @p_cake_price;
+---------------+
| @p_cake_price |
+---------------+
|        100.00 |
+---------------+
1 row in set (0.00 sec)
```

# Cursor

**A cursor is a database object used to fetch and process rows from a query result set, one row at a time.**

Key Concepts:

1. Declaration: A cursor must first be declared and associated with a SELECT statement.
2. Opening: The cursor is opened to establish the context and fetch rows.
3. Fetching: Retrieves one row at a time from the result set.
4. Closing: Closes the cursor to free resources.

Syntax of cursor:

```
delimiter $$


create procedure procedure_name()

begin

    -- Declare variables

    declare done int default 0;

    declare var_name data_type;


    -- Declare cursor

    declare cursor_name cursor for

    select column_name

    from table_name;


    -- Declare a handler for the cursor
```

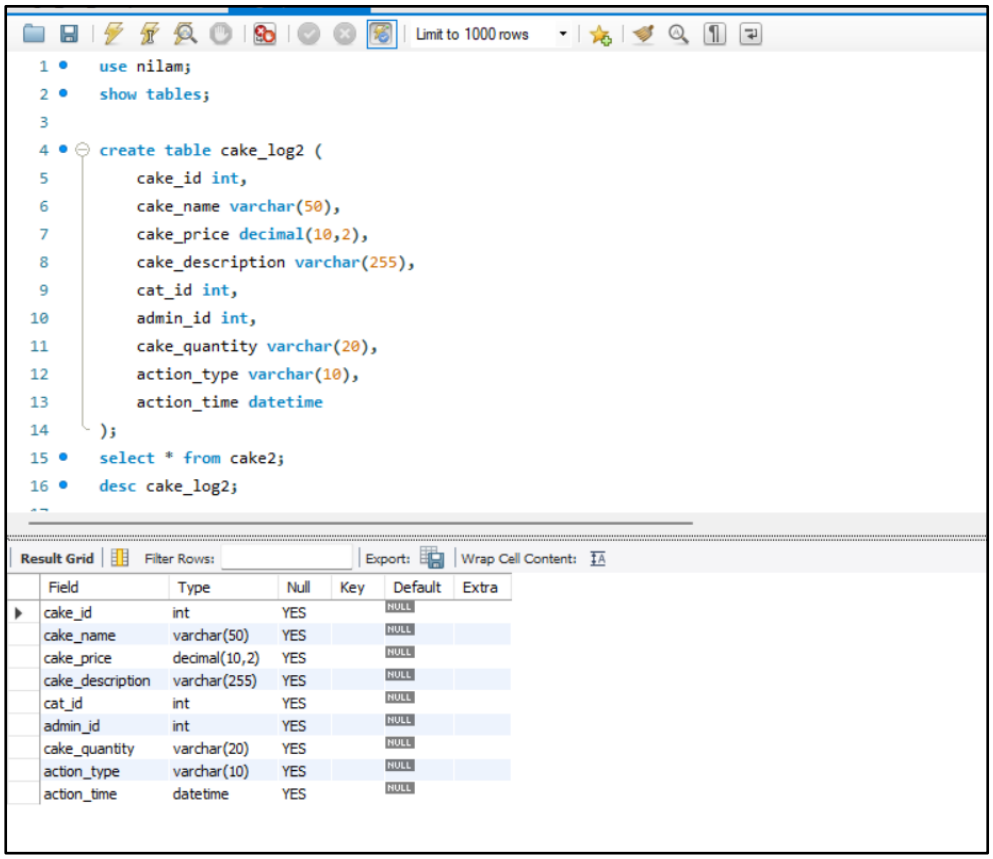
```
    declare continue handler for not found set done = 1;

    -- Open the cursor
    open cursor_name;

    -- Loop through rows fetched by the cursor
    fetch_loop: loop
        fetch cursor_name into var_name;
        if done then
            leave fetch_loop;
        end if;

        -- Process each row here
        -- Example: insert into another_table values (var_name);
    end loop;

    -- Close the cursor
    close cursor_name;
end$$

delimiter ;
```

# Automating SQL (Triggers)

A trigger is a special kind of rule in a database that automatically runs a specific action when certain changes (like adding, updating, or deleting data) happen to a table.

1. **BEFORE Triggers**: Executed before an operation on the table (used for data validation or modification).
2. **AFTER Triggers**: Executed after an operation on the table (commonly used for logging, auditing, or cascading changes).
3. **INSERT Triggers**: Fired when a new row is inserted into the table.
4. **UPDATE Triggers**: Fired when an existing row is updated.
5. **DELETE Triggers**: Fired when a row is deleted from the table.

First create backup table cake_log2



## 1. Trigger: log_after_insert2

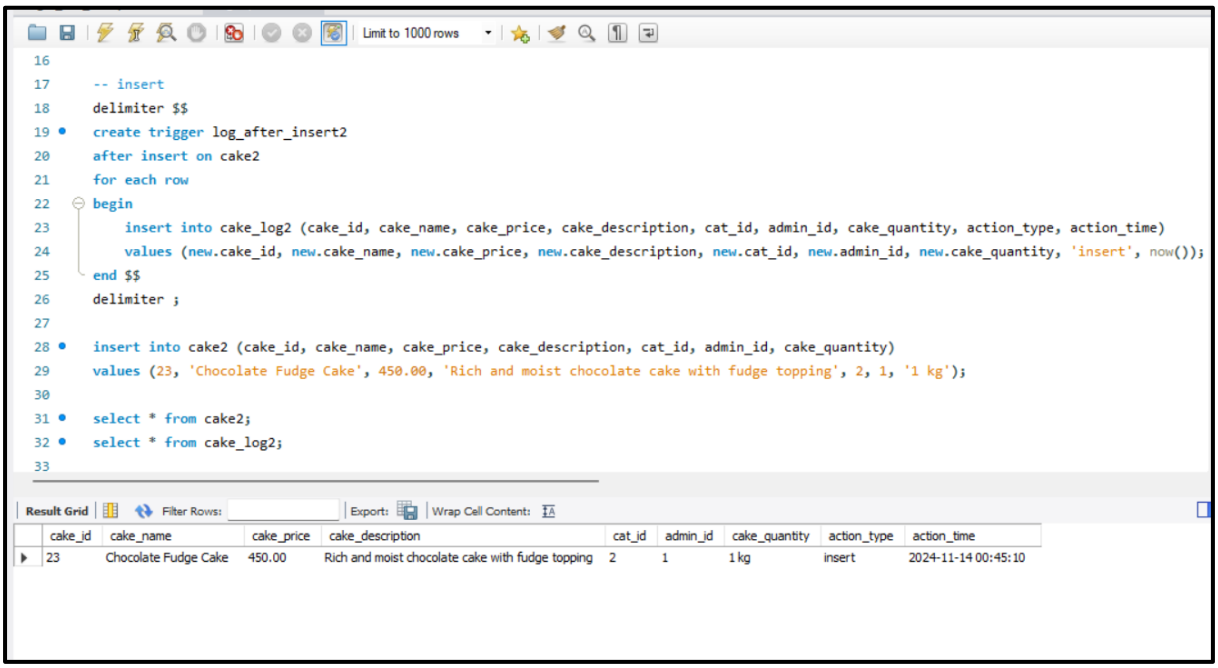
Event: After Insert

Table: cake2

Description:

This trigger logs the details of each inserted row in the cake2 table into the cake_log2 table.

Example:



```
16
17    -- insert
18    delimiter $$
19    create trigger log_after_insert2
20    after insert on cake2
21    for each row
22    begin
23        insert into cake_log2 (cake_id, cake_name, cake_price, cake_description, cat_id, admin_id, cake_quantity, action_type, action_time)
24        values (new.cake_id, new.cake_name, new.cake_price, new.cake_description, new.cat_id, new.admin_id, new.cake_quantity, 'insert', now());
25    end $$
26    delimiter ;
27
28    insert into cake2 (cake_id, cake_name, cake_price, cake_description, cat_id, admin_id, cake_quantity)
29    values (23, 'Chocolate Fudge Cake', 450.00, 'Rich and moist chocolate cake with fudge topping', 2, 1, '1 kg');
30
31    select * from cake2;
32    select * from cake_log2;
33
```

Result Grid

| cake_id | cake_name | cake_price | cake_description | cat_id | admin_id | cake_quantity | action_type | action_time |
|---|---|---|---|---|---|---|---|---|
| 23 | Chocolate Fudge Cake | 450.00 | Rich and moist chocolate cake with fudge topping | 2 | 1 | 1 kg | insert | 2024-11-14 00:45:10 |

## 2. Trigger: log_after_update2

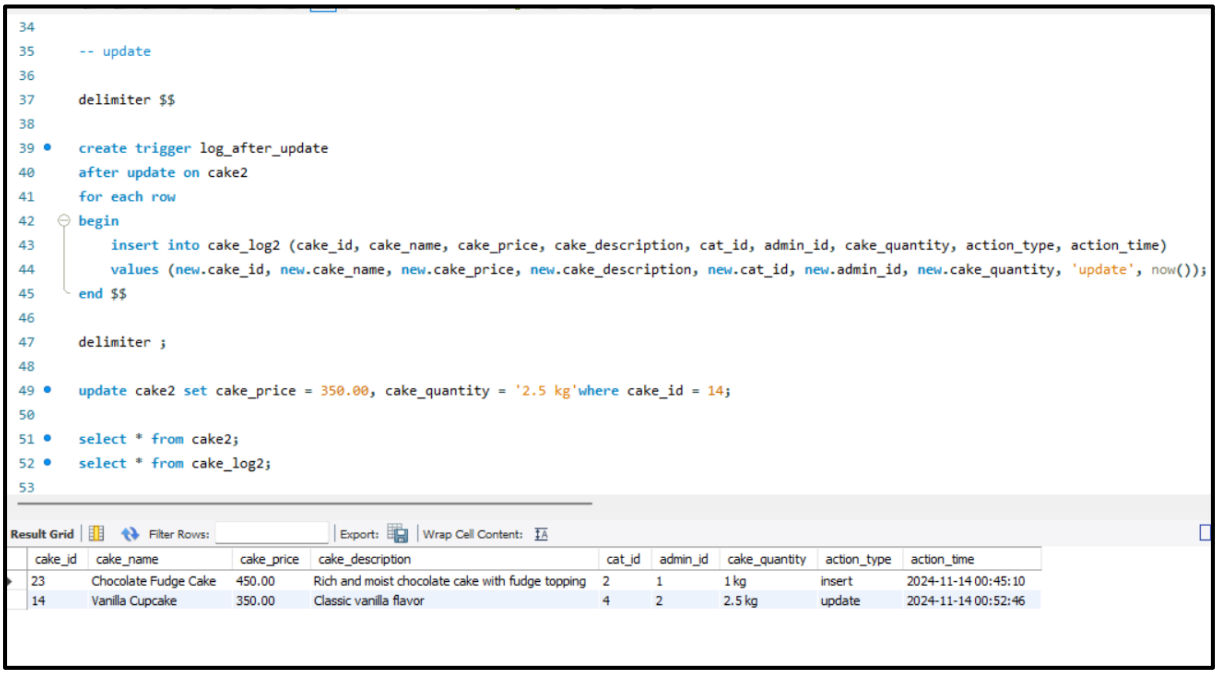
Event: After Update

Table: cake2

Description:

This trigger logs the details of each updated row from the cake2 table into the cake_log2 table.

Example:



```
34
35      -- update
36
37      delimiter $$
38
39  •   create trigger log_after_update
40      after update on cake2
41      for each row
42      begin
43          insert into cake_log2 (cake_id, cake_name, cake_price, cake_description, cat_id, admin_id, cake_quantity, action_type, action_time)
44          values (new.cake_id, new.cake_name, new.cake_price, new.cake_description, new.cat_id, new.admin_id, new.cake_quantity, 'update', now());
45      end $$
46
47      delimiter ;
48
49  •   update cake2 set cake_price = 350.00, cake_quantity = '2.5 kg'where cake_id = 14;
50
51  •   select * from cake2;
52  •   select * from cake_log2;
53
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| cake_id | cake_name | cake_price | cake_description | cat_id | admin_id | cake_quantity | action_type | action_time |
|---|---|---|---|---|---|---|---|---|
| 23 | Chocolate Fudge Cake | 450.00 | Rich and moist chocolate cake with fudge topping | 2 | 1 | 1 kg | insert | 2024-11-14 00:45:10 |
| 14 | Vanilla Cupcake | 350.00 | Classic vanilla flavor | 4 | 2 | 2.5 kg | update | 2024-11-14 00:52:46 |

## 3. Trigger: log_after_delete2

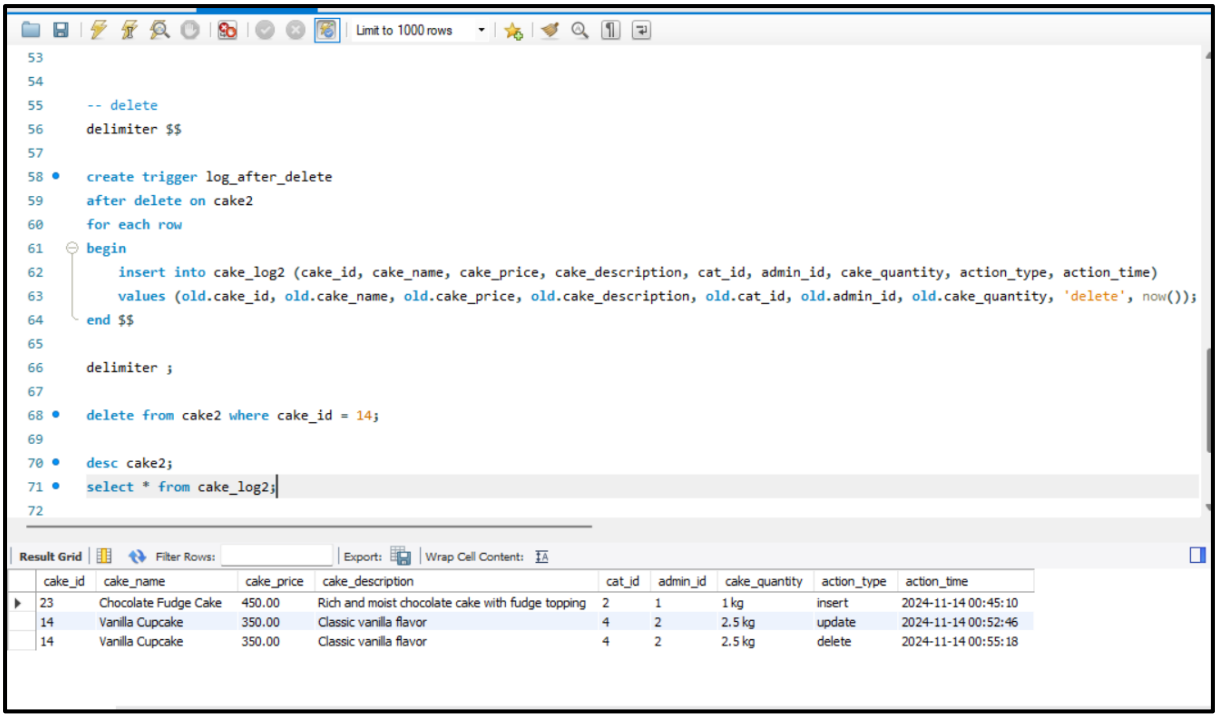
Event: After Delete

Table: cake2

Description:

This trigger logs the details of each deleted row from the cake2 table into the cake_log2 table.

Example:



```
53
54
55      -- delete
56      delimiter $$
57
58  •   create trigger log_after_delete
59      after delete on cake2
60      for each row
61      begin
62          insert into cake_log2 (cake_id, cake_name, cake_price, cake_description, cat_id, admin_id, cake_quantity, action_type, action_time)
63          values (old.cake_id, old.cake_name, old.cake_price, old.cake_description, old.cat_id, old.admin_id, old.cake_quantity, 'delete', now());
64      end $$
65
66      delimiter ;
67
68  •   delete from cake2 where cake_id = 14;
69
70  •   desc cake2;
71  •   select * from cake_log2;
72
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| cake_id | cake_name | cake_price | cake_description | cat_id | admin_id | cake_quantity | action_type | action_time |
|---|---|---|---|---|---|---|---|---|
| 23 | Chocolate Fudge Cake | 450.00 | Rich and moist chocolate cake with fudge topping | 2 | 1 | 1 kg | insert | 2024-11-14 00:45:10 |
| 14 | Vanilla Cupcake | 350.00 | Classic vanilla flavor | 4 | 2 | 2.5 kg | update | 2024-11-14 00:52:46 |
| 14 | Vanilla Cupcake | 350.00 | Classic vanilla flavor | 4 | 2 | 2.5 kg | delete | 2024-11-14 00:55:18 |