Cake Shop

Management Database

**Overview:-**

Effective cake shop management is crucial for ensuring smooth operations, customer satisfaction, and overall business success. In a competitive market, a cake shop must not only produce high-quality baked goods but also maintain efficient workflows, manage inventory effectively, and provide excellent customer service. A robust inventory management system helps track stock levels and reorder points, preventing waste and ensuring freshness while minimizing the risk of overstocking or stockouts. Additionally, managing customer relationships is essential for building loyalty and repeat business; by leveraging customer data, the shop can personalize marketing efforts and tailor offerings to enhance the shopping experience. Overall, effective management combines operational efficiency, inventory control, and customer relationship management to foster a thriving business.

To create a comprehensive data model for a cake shop that encompasses conceptual, logical, and physical aspects, we can break down the steps as follows:

**1. Conceptual Data Model**

The conceptual data model outlines the high-level structure of the data without delving into specific implementations. It identifies the main entities, their relationships, and attributes.

**Entities:**

* **Cakes**
* **Pastries**
* **Decorations**
* **Customers**
* **Orders**
* **Order Details**
* **Payments**
* **Inventory**
* **Employees**
* **Delivery**

**Relationships:**

* **Customers** can place **Orders**.
* **Orders** can include multiple **Order Details** (which can refer to Cakes, Pastries, or Decorations).
* **Employees** handle **Orders** and **Payments**.
* **Orders** are associated with **Payments**.
* **Delivery** is linked to **Orders**.
* **Cakes**, **Pastries**, and **Decorations** are managed within **Inventory**.

**2. Logical Data Model**

The logical data model specifies the structure of the database in a way that includes the detailed attributes and types, while still remaining independent of a specific database management system.

**Entities and Attributes:**

1. **Cakes**
   * cake\_id (PK, INT)
   * flavour (VARCHAR)
   * weight\_kg (DECIMAL)
   * price (DECIMAL)
   * description (TEXT)
   * inventory\_id (FK, INT)
2. **Pastries**
   * pastry\_id (PK, INT)
   * flavour (VARCHAR)
   * price (DECIMAL)
   * description (TEXT)
   * inventory\_id (FK, INT)
3. **Decorations**
   * decor\_id (PK, INT)
   * item\_name (VARCHAR)
   * price (DECIMAL)
   * description (TEXT)
   * inventory\_id (FK, INT)
4. **Customers**
   * customer\_id (PK, INT)
   * customer\_name (VARCHAR)
   * contact\_info (VARCHAR)
   * address (TEXT)
5. **Orders**
   * order\_id (PK, INT)
   * customer\_id (FK, INT)
   * employee\_id (FK, INT)
   * total\_price (DECIMAL)
   * order\_date (DATE)
6. **Order Details**
   * order\_detail\_id (PK, INT)
   * order\_id (FK, INT)
   * item\_type (ENUM: 'Cake', 'Pastry', 'Decoration')
   * item\_id (INT)
   * quantity (INT)
   * price (DECIMAL)
7. **Payments**
   * payment\_id (PK, INT)
   * order\_id (FK, INT)
   * employee\_id (FK, INT)
   * payment\_method (ENUM: 'Cash', 'Card', 'UPI')
   * payment\_status (ENUM: 'Full-Paid', 'Half-Paid', 'Cancelled', 'Pending')
   * payment\_date (DATE)
8. **Inventory**
   * inventory\_id (PK, INT)
   * item\_name (VARCHAR)
   * quantity (INT)
   * unit (VARCHAR)
   * reorder\_level (INT)
9. **Employees**
   * employee\_id (PK, INT)
   * name (VARCHAR)
   * role (VARCHAR)
   * contact\_info (VARCHAR)
   * salary (DECIMAL)
10. **Delivery**
    * delivery\_id (PK, INT)
    * order\_id (FK, INT)
    * employee\_id (FK, INT)
    * delivery\_address (TEXT)
    * delivery\_date (DATE)
    * delivery\_status (VARCHAR)

**3. Physical Data Model**

The physical data model provides the actual implementation details, including data types and constraints as needed for a specific database management system.

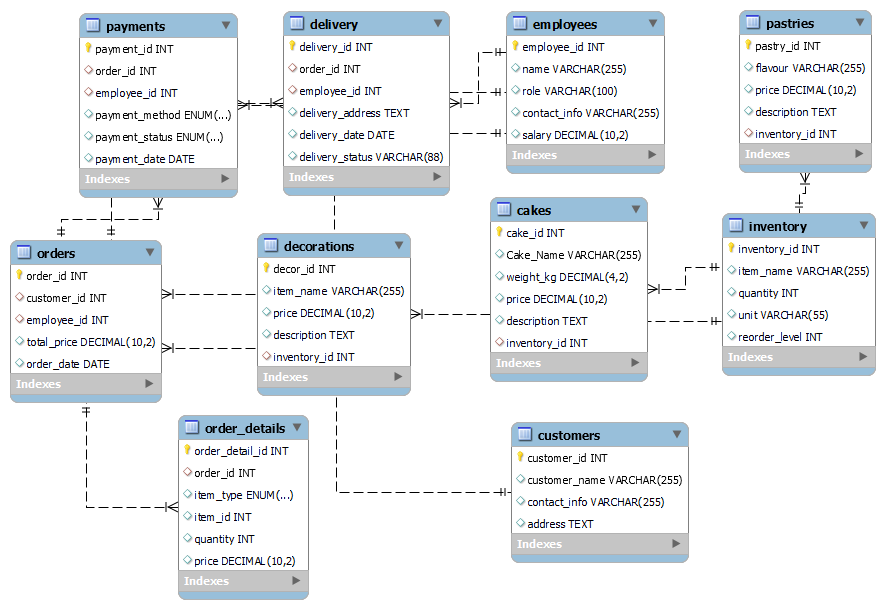
This is generally what you have provided in your SQL code. Here’s a brief overview of the physical aspects:

* Each table is created with specified data types (e.g., INT, VARCHAR, DECIMAL, TEXT, DATE).
* Primary keys are defined for each table.
* Foreign key constraints are set to maintain referential integrity.
* Use of ENUM for fields that have a defined set of possible values (e.g., payment\_method, payment\_status, item\_type).

**SQL Implementation**

Here’s a brief recap of your SQL implementation, which is correctly structured to represent the physical model:

**ER Diagram**

****

**Create Tables**

-- Cakes Table

CREATE TABLE Cakes (

cake\_id INT PRIMARY KEY,

flavour VARCHAR(255),

weight\_kg DECIMAL(4,2),

price DECIMAL(10,2),

description TEXT,

inventory\_id INT,

FOREIGN KEY (inventory\_id) REFERENCES Inventory(inventory\_id)

);

-- Pastries Table

CREATE TABLE Pastries (

pastry\_id INT PRIMARY KEY,

flavour VARCHAR(255),

price DECIMAL(10,2),

description TEXT,

inventory\_id INT,

FOREIGN KEY (inventory\_id) REFERENCES Inventory(inventory\_id)

);

-- Decorations Table

CREATE TABLE Decorations (

decor\_id INT PRIMARY KEY,

item\_name VARCHAR(255),

price DECIMAL(10,2),

description TEXT,

inventory\_id INT,

FOREIGN KEY (inventory\_id) REFERENCES Inventory(inventory\_id)

);

-- Customers Table

CREATE TABLE Customers (

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(255),

contact\_info VARCHAR(255),

address TEXT

);

-- Orders Table

CREATE TABLE Orders (

order\_id INT PRIMARY KEY,

customer\_id INT,

employee\_id INT,

total\_price DECIMAL(10,2),

order\_date DATE,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),

FOREIGN KEY (employee\_id) REFERENCES Employees(employee\_id)

);

-- Order Details Table

CREATE TABLE Order\_Details (

order\_detail\_id INT PRIMARY KEY,

order\_id INT,

item\_type ENUM('Cake', 'Pastry', 'Decoration'),

item\_id INT,

quantity INT,

price DECIMAL(10,2),

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id)

);

-- Payments Table

CREATE TABLE Payments (

payment\_id INT PRIMARY KEY,

order\_id INT,

employee\_id INT,

payment\_method ENUM('Cash', 'Card', 'UPI'),

payment\_status ENUM('Full-Paid', 'Half-Paid', 'Cancelled', 'Pending'),

payment\_date DATE,

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id),

FOREIGN KEY (employee\_id) REFERENCES Employees(employee\_id)

);

-- Inventory Table

CREATE TABLE Inventory (

inventory\_id INT PRIMARY KEY,

item\_name VARCHAR(255),

quantity INT,

unit VARCHAR(55),

reorder\_level INT

);

-- Employees Table

CREATE TABLE Employees (

employee\_id INT PRIMARY KEY,

name VARCHAR(255),

role VARCHAR(100),

contact\_info VARCHAR(255),

salary DECIMAL(10,2)

);

-- Delivery Table

CREATE TABLE Delivery (

delivery\_id INT PRIMARY KEY,

order\_id INT,

employee\_id INT,

delivery\_address TEXT,

delivery\_date DATE,

delivery\_status VARCHAR(88),

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id),

FOREIGN KEY (employee\_id) REFERENCES Employees(employee\_id)

);

**Insert Values**

-- Insert into **Cakes**

INSERT INTO Cakes VALUES

(1, 'Chocolate', 0.50, 300.00, 'Chocolate cake for small celebrations', 1),

(2, 'Vanilla', 1.00, 600.00, 'Vanilla cake for birthday parties', 2),

(3, 'Strawberry', 2.00, 1200.00, 'Strawberry cake for anniversaries', 3),

(4, 'Red Velvet', 3.00, 1800.00, 'Large red velvet cake for big events', 4),

(5, 'Black Forest', 5.00, 3000.00, 'Black Forest cake for grand celebrations', 5),

(6, 'Pineapple', 0.50, 350.00, 'Small pineapple cake for gatherings', 6),

(7, 'Butterscotch', 1.50, 900.00, 'Butterscotch cake for medium-sized parties', 7),

(8, 'Coffee', 2.50, 1400.00, 'Coffee-flavored cake for special events', 8),

(9, 'Fruit Cake', 3.00, 1800.00, 'Fruit cake for big celebrations', 9),

(10, 'Mango', 4.00, 2200.00, 'Mango cake for large birthday parties', 10),

(11, 'Oreo', 5.00, 3500.00, 'Oreo cake for special occasions', 11),

(12, 'Blueberry', 0.50, 400.00, 'Small blueberry cake for casual events', 12),

(13, 'Lemon', 1.00, 600.00, 'Lemon cake for small celebrations', 13),

(14, 'Cheesecake', 3.00, 2800.00, 'Large cheesecake for special events', 14),

(15, 'Tiramisu', 2.00, 1500.00, 'Tiramisu cake for intimate parties', 15),

(16, 'Carrot Cake', 1.50, 800.00, 'Carrot cake for family gatherings', 16),

(17, 'Marble Cake', 0.50, 350.00, 'Small marble cake for casual celebrations', 17),

(18, 'Raspberry', 4.50, 2800.00, 'Raspberry cream cake for big occasions', 18),

(19, 'Peanut Butter', 2.50, 1300.00, 'Peanut butter cake for small events', 19),

(20, 'Almond', 5.00, 3800.00, 'Almond cake for large celebrations', 20),

(21, 'Honey Cake', 1.00, 750.00, 'Honey cake for small family events', 21),

(22, 'Orange Cake', 2.00, 1200.00, 'Orange cake for birthday parties', 22),

(23, 'Coconut Cake', 3.50, 1800.00, 'Coconut cake for special events', 23),

(24, 'Pistachio Cake', 4.00, 2500.00, 'Pistachio cake for large gatherings', 24),

(25, 'Walnut Cake', 0.50, 350.00, 'Small walnut cake for casual events', 25),

(26, 'Mint Chocolate', 5.00, 4000.00, 'Mint chocolate cake for big celebrations', 26),

(27, 'Banana Cake', 1.50, 900.00, 'Banana cake for medium-sized parties', 27),

(28, 'Sponge Cake', 2.50, 1400.00, 'Sponge cake for large gatherings', 28),

(29, 'Hazelnut Cake', 3.00, 2200.00, 'Hazelnut cake for special occasions', 29),

(30, 'Mango', 5.00, 3500.00, 'Large mango cake for anniversaries or big parties', 30);

-- Insert into **Pastries** (30 rows)

INSERT INTO Pastries (pastry\_id, flavour, price, description, inventory\_id) VALUES

(1, 'Chocolate', 50.00, 'Chocolate pastry', 1),

(2, 'Vanilla', 45.00, 'Vanilla pastry', 2),

(3, 'Butterscotch', 55.00, 'Butterscotch pastry', 3),

(4, 'Mango', 60.00, 'Mango flavored pastry', 4),

(5, 'Strawberry', 50.00, 'Fresh strawberry pastry', 5),

(6, 'Pineapple', 55.00, 'Pineapple pastry', 6),

(7, 'Blueberry', 65.00, 'Blueberry pastry', 7),

(8, 'Coffee', 70.00, 'Coffee-flavored pastry', 8),

(9, 'Red Velvet', 75.00, 'Red Velvet pastry', 9),

(10, 'Caramel', 60.00, 'Caramel pastry', 10),

(11, 'Oreo', 80.00, 'Oreo pastry', 11),

(12, 'Banana', 50.00, 'Banana pastry', 12),

(13, 'Hazelnut', 85.00, 'Hazelnut pastry', 13),

(14, 'Mint', 70.00, 'Mint-flavored pastry', 14),

(15, 'Raspberry', 65.00, 'Raspberry pastry', 15),

(16, 'Blackberry', 60.00, 'Blackberry pastry', 16),

(17, 'Peanut Butter', 75.00, 'Peanut butter pastry', 17),

(18, 'Coconut', 55.00, 'Coconut pastry', 18),

(19, 'Almond', 80.00, 'Almond pastry', 19),

(20, 'Walnut', 75.00, 'Walnut pastry', 20),

(21, 'Orange', 55.00, 'Orange-flavored pastry', 21),

(22, 'Pistachio', 85.00, 'Pistachio pastry', 22),

(23, 'Sponge', 50.00, 'Sponge pastry', 23),

(24, 'Cheesecake', 90.00, 'Cheesecake pastry', 24),

(25, 'Tiramisu', 100.00, 'Tiramisu pastry', 25),

(26, 'Mousse', 85.00, 'Chocolate mousse pastry', 26),

(27, 'Marble', 60.00, 'Marble pastry', 27),

(28, 'Carrot', 55.00, 'Carrot-flavored pastry', 28),

(29, 'Apple', 70.00, 'Apple-flavored pastry', 29),

(30, 'Lemon', 65.00, 'Lemon-flavored pastry', 30);

-- Insert into **Decorations** (first 8 rows)

INSERT INTO Decorations (decor\_id, item\_name, price, description, inventory\_id) VALUES

(1, 'Balloon Set', 150.00, 'Balloons for decoration', 1),

(2, 'Normal Candles', 30.00, 'Candles for cake', 4),

(3, 'Number Candles', 50.00, 'Candles for cake', 20),

(4, 'Banners', 120.00, 'Celebration banners', 5),

(5, 'Ribbons', 80.00, 'Ribbons for cake and decor', 7),

(6, 'Sparkle ', 55.00, 'Sparkles for Cake ', 8),

(7, 'Big Ballon ', 45.00, 'Big ballon ', 5);

-- Insert into **Customers**

INSERT INTO Customers (customer\_id, customer\_name, contact\_info, address) VALUES

(1, 'Rahul Sharma', '9876543210', 'Mumbai, India'),

(2, 'Pooja Singh', '9123456789', 'Delhi, India'),

(3, 'Rahul Sharma', '9123456710', 'Mumbai, India'), -- Duplicate name, different details

(4, 'Amit Verma', '9876543123', 'Bangalore, India'),

(5, 'Sneha Patil', '9812345678', 'Mumbai, India'),

(6, 'Rahul Sharma', '9812345123', 'Pune, India'), -- Duplicate name again

(7, 'Anjali Desai', '9988776655', 'Mumbai, India'),

(8, 'Sameer Khan', '9876541234', 'Mumbai, India'),

(9, 'Riya Mehta', '8765432109', 'Mumbai, India'),

(10, 'Rohan Gupta', '9912345678', 'Mumbai, India'),

(11, 'Aditi Joshi', '9700000001', 'Mumbai, India'),

(12, 'Rahul Sharma', '9000000002', 'Mumbai, India'), -- Another duplicate

(13, 'Vikram Singh', '9912345689', 'Mumbai, India'),

(14, 'Priya Chavan', '9988776644', 'Mumbai, India'),

(15, 'Tarun Yadav', '7777777777', 'Mumbai, India'),

(16, 'Kajal Patil', '6666666666', 'Mumbai, India'),

(17, 'Neha Sinha', '5555555555', 'Mumbai, India'),

(18, 'Rakesh Joshi', '4444444444', 'Mumbai, India'),

(19, 'Suresh Naik', '3333333333', 'Mumbai, India'),

(20, 'Rahul Sharma', '2222222222', 'Mumbai, India'), -- Another entry for Rahul

(21, 'Karan Bhatia', '1111111111', 'Mumbai, India'),

(22, 'Snehal Khatri', '8888888888', 'Mumbai, India'),

(23, 'Dinesh Patil', '9999998888', 'Mumbai, India'),

(24, 'Asha Verma', '7777776666', 'Mumbai, India'),

(25, 'Ajay Rathi', '8888885555', 'Mumbai, India'),

(26, 'Raghav Sharma', '6666664444', 'Mumbai, India'),

(27, 'Vani Mehta', '5555553333', 'Mumbai, India'),

(28, 'Aarav Singh', '4444442222', 'Mumbai, India'),

(29, 'Isha Joshi', '3333331111', 'Mumbai, India'),

(30, 'Ravi Patel', '9999999999', 'Ahmedabad, India'),

(31, 'Nikita Gupta', '9988773322', 'Mumbai, India'),

(32, 'Kriti Agarwal', '9933445566', 'Mumbai, India'),

(33, 'Rajesh Rathi', '9988776655', 'Mumbai, India'),

(34, 'Seema Das', '8899001122', 'Mumbai, India'),

(35, 'Amit Soni', '8833445566', 'Mumbai, India'),

(36, 'Neelam Thakur', '9988773344', 'Mumbai, India'),

(37, 'Prakash Desai', '8844556677', 'Mumbai, India'),

(38, 'Anita Roy', '9911223344', 'Mumbai, India'),

(39, 'Shivam Tiwari', '9922334455', 'Mumbai, India'),

(40, 'Meera Kaur', '9999888777', 'Mumbai, India'),

(41, 'Gaurav Yadav', '9999990000', 'Mumbai, India'),

(42, 'Ritesh Bhardwaj', '7777777771', 'Mumbai, India'),

(43, 'Simran Singh', '6666667771', 'Mumbai, India'),

(44, 'Harsh Kumar', '8888880000', 'Mumbai, India'),

(45, 'Tanya Joshi', '1112223333', 'Mumbai, India'),

(46, 'Kunal Khanna', '2221110000', 'Mumbai, India'),

(47, 'Riya Sharma', '3334445555', 'Mumbai, India'),

(48, 'Ankita Bansal', '4445556666', 'Mumbai, India'),

(49, 'Ravi Sharma', '5556667777', 'Mumbai, India'),

(50, 'Ananya Verma', '6667778888', 'Mumbai, India');

-- Insert into **Orders**

INSERT INTO Orders (order\_id, customer\_id, employee\_id, total\_price, order\_date) VALUES

(1, 1, 1, 1500.00, '2024-10-01'),

(2, 2, 2, 900.00, '2024-10-03'),

(3, 3, 3, 1100.00, '2024-10-05'),

(4, 4, 4, 1750.00, '2024-10-06'),

(5, 5, 5, 1300.00, '2024-10-07'),

(6, 6, 1, 950.00, '2024-10-08'),

(7, 7, 2, 1200.00, '2024-10-09'),

(8, 8, 3, 600.00, '2024-10-10'),

(9, 9, 4, 1550.00, '2024-10-12'),

(10, 10, 5, 850.00, '2024-10-13'),

(11, 6, 1, 2500.00, '2024-10-18'),

(12, 7, 2, 3200.00, '2024-10-18'),

(13, 8, 3, 4000.00, '2024-10-19'),

(14, 9, 4, 2300.00, '2024-10-20'),

(15, 10, 5, 2900.00, '2024-10-20'),

(16, 11, 1, 3500.00, '2024-10-21'),

(17, 12, 2, 2700.00, '2024-10-21'),

(18, 13, 3, 1800.00, '2024-10-22'),

(19, 14, 4, 1400.00, '2024-10-22'),

(20, 15, 5, 1600.00, '2024-10-22'),

(21, 16, 6, 2100.00, '2024-10-23'),

(22, 17, 7, 2200.00, '2024-10-23'),

(23, 18, 1, 2000.00, '2024-10-24'),

(24, 19, 2, 1900.00, '2024-10-24'),

(25, 20, 3, 2400.00, '2024-10-25'),

(26, 21, 4, 1750.00, '2024-10-25'),

(27, 22, 5, 1850.00, '2024-10-26'),

(28, 23, 6, 1950.00, '2024-10-26'),

(29, 24, 7, 2050.00, '2024-10-27'),

(30, 25, 1, 2150.00, '2024-10-27');

-- Insert into **Order\_Details**

INSERT INTO Order\_Details (order\_detail\_id, order\_id, item\_type, item\_id, quantity, price) VALUES

(1, 1, 'Cake', 1, 1, 500.00),

(2, 1, 'Pastry', 1, 4, 200.00),

(3, 2, 'Cake', 2, 1, 1000.00),

(4, 2, 'Decoration', 1, 1, 150.00),

(5, 3, 'Cake', 3, 1, 600.00),

(6, 3, 'Pastry', 2, 2, 400.00),

(7, 4, 'Cake', 4, 1, 700.00),

(8, 4, 'Decoration', 2, 1, 120.00),

(9, 5, 'Cake', 5, 1, 800.00),

(10, 5, 'Pastry', 3, 3, 600.00),

(11, 6, 'Cake', 6, 1, 900.00),

(12, 6, 'Decoration', 3, 1, 150.00),

(13, 7, 'Cake', 7, 1, 1000.00),

(14, 7, 'Pastry', 4, 2, 400.00),

(15, 8, 'Cake', 8, 1, 1100.00),

(16, 8, 'Decoration', 4, 1, 160.00),

(17, 9, 'Cake', 9, 1, 1200.00),

(18, 9, 'Pastry', 5, 2, 300.00),

(19, 10, 'Cake', 10, 1, 1300.00),

(20, 10, 'Decoration', 5, 1, 170.00),

(21, 11, 'Cake', 11, 1, 1400.00),

(22, 11, 'Pastry', 6, 3, 450.00),

(23, 12, 'Cake', 12, 1, 1500.00),

(24, 12, 'Decoration', 6, 1, 180.00),

(25, 13, 'Cake', 13, 1, 1600.00),

(26, 13, 'Pastry', 7, 2, 600.00),

(27, 14, 'Cake', 14, 1, 1700.00),

(28, 14, 'Decoration', 7, 1, 190.00),

(29, 15, 'Cake', 15, 1, 1800.00),

(30, 15, 'Pastry', 8, 3, 720.00),

(31, 16, 'Cake', 16, 1, 1900.00),

(32, 16, 'Decoration', 8, 1, 200.00),

(33, 17, 'Cake', 17, 1, 2000.00),

(34, 17, 'Pastry', 9, 2, 300.00),

(35, 18, 'Cake', 18, 1, 2100.00);

-- Insert into **Payments**

INSERT INTO Payments (payment\_id, order\_id, employee\_id, payment\_method, payment\_status, payment\_date) VALUES

(1, 1, 1, 'UPI', 'Full-Paid', '2024-10-01'),

(2, 2, 2, 'Cash', 'Full-Paid', '2024-10-02'),

(3, 3, 1, 'Card', 'Half-Paid', '2024-10-03'),

(4, 4, 2, 'UPI', 'Full-Paid', '2024-10-04'),

(5, 5, 3, 'UPI', 'Pending', '2024-10-05'),

(6, 6, 1, 'Cash', 'Full-Paid', '2024-10-06'),

(7, 7, 2, 'Card', 'Full-Paid', '2024-10-07'),

(8, 8, 3, 'UPI', 'Half-Paid', '2024-10-08'),

(9, 9, 4, 'Cash', 'Full-Paid', '2024-10-09'),

(10, 10, 5, 'Card', 'Pending', '2024-10-10'),

(11, 11, 1, 'UPI', 'Full-Paid', '2024-10-11'),

(12, 12, 2, 'Cash', 'Full-Paid', '2024-10-12'),

(13, 13, 3, 'Card', 'Half-Paid', '2024-10-13'),

(14, 14, 4, 'UPI', 'Full-Paid', '2024-10-14'),

(15, 15, 5, 'Cash', 'Pending', '2024-10-15'),

(16, 16, 1, 'UPI', 'Full-Paid', '2024-10-16'),

(17, 17, 2, 'Card', 'Full-Paid', '2024-10-17'),

(18, 18, 3, 'Cash', 'Half-Paid', '2024-10-18'),

(19, 19, 4, 'UPI', 'Full-Paid', '2024-10-19'),

(20, 20, 5, 'Card', 'Pending', '2024-10-20'),

(21, 21, 1, 'Cash', 'Full-Paid', '2024-10-21'),

(22, 22, 2, 'UPI', 'Full-Paid', '2024-10-22'),

(23, 23, 3, 'Card', 'Half-Paid', '2024-10-23'),

(24, 24, 4, 'Cash', 'Full-Paid', '2024-10-24'),

(25, 25, 5, 'UPI', 'Pending', '2024-10-25'),

(26, 26, 1, 'Cash', 'Full-Paid', '2024-10-26'),

(27, 27, 2, 'Card', 'Full-Paid', '2024-10-27'),

(28, 28, 3, 'UPI', 'Half-Paid', '2024-10-28'),

(29, 29, 4, 'Cash', 'Full-Paid', '2024-10-29'),

(30, 30, 5, 'Cash', 'Full-Paid', '2024-10-30');

-- Insert into **Inventory**

INSERT INTO Inventory (inventory\_id, item\_name, quantity, unit, reorder\_level) VALUES

(1, 'Chocolate Cake Ingredients', 50, 'kg', 10),

(2, 'Vanilla Cake Ingredients', 60, 'kg', 15),

(3, 'Pastry Ingredients', 100, 'pieces', 20),

(4, 'Decoration Items', 150, 'pieces', 30),

(5, 'Strawberry Cake Ingredients', 40, 'kg', 8),

(6, 'Lemon Cake Ingredients', 30, 'kg', 6),

(7, 'Fruit Topping', 70, 'kg', 12),

(8, 'Whipped Cream', 80, 'kg', 18),

(9, 'Sprinkles', 200, 'pieces', 50),

(10, 'Cake Boxes', 300, 'pieces', 100),

(11, 'Chocolate Syrup', 20, 'liters', 5),

(12, 'Vanilla Extract', 25, 'liters', 6),

(13, 'Almond Flour', 15, 'kg', 3),

(14, 'Butter', 40, 'kg', 10),

(15, 'Eggs', 120, 'pieces', 30),

(16, 'Sugar', 50, 'kg', 12),

(17, 'Flour', 100, 'kg', 20),

(18, 'Baking Powder', 30, 'kg', 5),

(19, 'Salt', 15, 'kg', 3),

(20, 'Baking Soda', 10, 'kg', 2),

(21, 'Cooking Oil', 50, 'liters', 10),

(22, 'Milk', 100, 'liters', 20),

(23, 'Cream Cheese', 30, 'kg', 7),

(24, 'Piping Bags', 200, 'pieces', 40),

(25, 'Chocolate Chips', 45, 'kg', 8),

(26, 'Fondant', 25, 'kg', 5),

(27, 'Gelatin', 10, 'kg', 2),

(28, 'Cocoa Powder', 15, 'kg', 3),

(29, 'Caramel Sauce', 12, 'liters', 3),

(30, 'Mango Cake Ingredients', 20, 'kg', 5);

-- Insert into **Employees**

INSERT INTO Employees (employee\_id, name, role, contact\_info, salary) VALUES

(1, 'Amit Kumar', 'Cashier', '9823456789', 20000.00),

(2, 'Sunita Gupta', 'Baker', '9812345678', 25000.00),

(3, 'Rajesh Mehta', 'Delivery Staff', '9876543211', 18000.00),

(4, 'Rani Sinha', 'Delivery Staff', '9821345678', 22000.00),

(5, 'Anil Sharma', 'Manager', '9812456789', 30000.00), -- 1 Manager

(6, 'Neha Desai', 'Baker', '9812345123', 24000.00), -- 2nd Baker

(7, 'Rakesh Jain', 'Delivery Staff', '9823415678', 18000.00), -- 2nd Delivery Staff

(8, 'Rahul Yadav', 'Cashier', '9823456123', 21000.00), -- 2nd Cashier

(9, 'Sneha Patil', 'Cashier', '9812345670', 20000.00), -- 3rd Cashier

(10, 'Pooja Rathi', 'Cleaning Staff', '9812365789', 15000.00); -- 1 Cleaning Staff

-- Insert into **Delivery**

INSERT INTO Delivery (delivery\_id, order\_id, employee\_id, delivery\_address, delivery\_date, delivery\_status) VALUES

(1, 1, 3, 'Mumbai, India', '2024-10-02', 'Delivered'),

(2, 2, 4, 'Delhi, India', '2024-10-03', 'Pending'),

(3, 3, 3, 'Mumbai, India', '2024-10-04', 'Delivered'),

(4, 4, 3, 'Bangalore, India', '2024-10-05', 'Dispatched'),

(5, 5, 3, 'Mumbai, India', '2024-10-06', 'Pending'),

(6, 6, 4, 'Mumbai, India', '2024-10-07', 'On Delivery'),

(7, 7, 2, 'Mumbai, India', '2024-10-08', 'Completed'),

(8, 8, 1, 'Mumbai, India', '2024-10-09', 'Dispatched'),

(9, 9, 5, 'Mumbai, India', '2024-10-10', 'Pending'),

(10, 10, 2, 'Mumbai, India', '2024-10-11', 'On Delivery'),

(11, 11, 3, 'Mumbai, India', '2024-10-12', 'Delivered'),

(12, 12, 4, 'Mumbai, India', '2024-10-13', 'Pending'),

(13, 13, 5, 'Mumbai, India', '2024-10-14', 'Dispatched'),

(14, 14, 1, 'Mumbai, India', '2024-10-15', 'On Delivery'),

(15, 15, 2, 'Mumbai, India', '2024-10-16', 'Completed'),

(16, 16, 3, 'Mumbai, India', '2024-10-17', 'Delivered'),

(17, 17, 4, 'Mumbai, India', '2024-10-18', 'Pending'),

(18, 18, 5, 'Mumbai, India', '2024-10-19', 'Dispatched'),

(19, 19, 1, 'Mumbai, India', '2024-10-20', 'On Delivery'),

(20, 20, 2, 'Mumbai, India', '2024-10-21', 'Completed'),

(21, 21, 3, 'Mumbai, India', '2024-10-22', 'Delivered'),

(22, 22, 4, 'Mumbai, India', '2024-10-23', 'Pending'),

(23, 23, 5, 'Mumbai, India', '2024-10-24', 'Dispatched'),

(24, 24, 1, 'Mumbai, India', '2024-10-25', 'On Delivery'),

(25, 25, 2, 'Mumbai, India', '2024-10-26', 'Completed'),

(26, 26, 3, 'Mumbai, India', '2024-10-27', 'Delivered'),

(27, 27, 4, 'Mumbai, India', '2024-10-28', 'Pending'),

(28, 28, 5, 'Mumbai, India', '2024-10-29', 'Dispatched'),

(29, 29, 1, 'Mumbai, India', '2024-10-30', 'On Delivery'),

(30, 30, 2, 'Ahmedabad, India', '2024-10-31', 'Dispatched');

**Select & Desc**

**SELECT**

select \* from Cakes ;

select \* from Pastries ;

select \* from Decorations ;

select \* from Customers ;

select \* from Orders ; --

select \* from Order\_Details ; --

select \* from Payments ; --

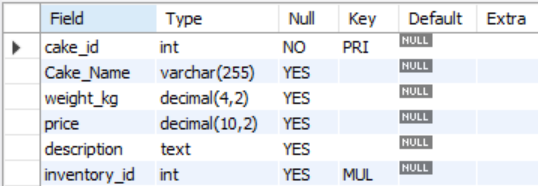
select \* from Inventory ;

select \* from Employees ;

select \* from Delivery ; --

**DESCRIBE**

Desc Cakes ;



Desc Pastries ;

A screenshot of a computer

Description automatically generated

Desc Decorations ;

A screenshot of a computer

Description automatically generated

Desc Customers ;

A screenshot of a computer

Description automatically generated

Desc Orders ;

A screenshot of a computer

Description automatically generated

Desc Order\_Details ;

A screenshot of a computer

Description automatically generated

Desc Payments ;

A screenshot of a computer

Description automatically generated

Desc Inventory ;

A screenshot of a computer

Description automatically generated

Desc Employees ;

A screenshot of a computer

Description automatically generated

Desc Delivery ;

A screenshot of a computer

Description automatically generated

-- change column name of cakes tables

ALTER TABLE Cakes

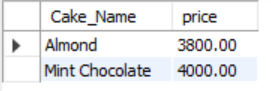
RENAME COLUMN flavour TO Cake\_Name;

-- To find all cakes with a price above 3500?

SELECT Cake\_Name, price

FROM Cakes

WHERE price > 3500;

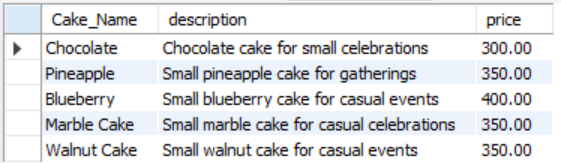


-- To find cakes that are best for small celebrations price below 500 ?

SELECT Cake\_Name, description, price

FROM Cakes

WHERE price < 500;



-- To count how many types of cakes are available:

SELECT COUNT(\*) AS total\_cakes

FROM Cakes;

A close-up of a number

Description automatically generated

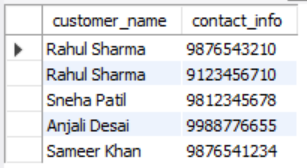
-- To find 5 customers from Mumbai:

SELECT customer\_name, contact\_info

FROM Customers

WHERE address LIKE '%Mumbai%'

limit 5;



-- To find customers who have the name 'Rahul Sharma'?

SELECT \*

FROM Customers

WHERE customer\_name = 'Rahul Sharma';

A screenshot of a computer

Description automatically generated

-- to display cakes in descending order of their price and weight ?

SELECT Cake\_Name, weight\_kg, price

FROM Cakes

ORDER BY price DESC

limit 10;

A screenshot of a computer

Description automatically generated

-- To find the cheapest cake:

SELECT Cake\_Name, MIN(price) AS lowest\_price

FROM Cakes

GROUP BY flavour

LIMIT 5;

A screenshot of a computer

Description automatically generated

-- To find all cakes that weigh more than 2 kg?

SELECT Cake\_Name, weight\_kg

FROM Cakes

WHERE weight\_kg > 2.00

limit 5;

A screenshot of a computer

Description automatically generated

-- To find cakes meant for special events?

SELECT Cake\_Name, description

FROM Cakes

WHERE description LIKE '%special occasions%';

A screenshot of a computer

Description automatically generated

-- Find all customers who have placed orders?

SELECT DISTINCT Customers.customer\_id, customer\_name, contact\_info, address

FROM Customers

JOIN Orders ON Customers.customer\_id = Orders.customer\_id limit 5;

A screenshot of a computer

Description automatically generated

-- Get total orders made by each customer ?

SELECT Customers.customer\_name, COUNT(Orders.order\_id) AS total\_orders

FROM Customers

JOIN Orders ON Customers.customer\_id = Orders.customer\_id

GROUP BY Customers.customer\_name limit 3;

A screenshot of a computer

Description automatically generated

-- Show all orders, including customer names, total price, and order dates:

SELECT Orders.order\_id, Customers.customer\_name, Orders.total\_price, Orders.order\_date

FROM Orders

JOIN Customers ON Orders.customer\_id = Customers.customer\_id;



-- Find the total sales made in a given month ?

SELECT SUM(total\_price) AS total\_sales

FROM Orders

WHERE MONTH(order\_date) = 10 AND YEAR(order\_date) = 2024;

A screenshot of a computer

Description automatically generated

-- List all employees along with their roles and contact information:

SELECT employee\_id, name, role, contact\_info,salary

FROM Employees;

A screenshot of a computer

Description automatically generated

-- Find all pending or unpaid payments:

SELECT Payments.payment\_id, Payments.payment\_status, Payments.payment\_date, Orders.order\_id, Customers.customer\_name

FROM Payments

JOIN Orders ON Payments.order\_id = Orders.order\_id

JOIN Customers ON Orders.customer\_id = Customers.customer\_id

WHERE Payments.payment\_status IN ('Half-Paid', 'Pending');

A screenshot of a computer

Description automatically generated

-- Display all deliveries along with delivery status and date:

SELECT delivery\_id, Orders.order\_id, Customers.customer\_name, Delivery.delivery\_address, Delivery.delivery\_date, Delivery.delivery\_status

FROM Delivery

JOIN Orders ON Delivery.order\_id = Orders.order\_id

JOIN Customers ON Orders.customer\_id = Customers.customer\_id;

A screenshot of a computer

Description automatically generated

-- which pastry is customer buying ?

SELECT

C.customer\_name,

PA.flavour AS pastry\_name,

PA.price AS pastry\_price

FROM Customers C

JOIN Orders O ON C.customer\_id = O.customer\_id

JOIN Order\_Details OD ON O.order\_id = OD.order\_id

JOIN Pastries PA ON OD.item\_id = PA.pastry\_id

WHERE OD.item\_type = 'Pastry' limit 5;

A screenshot of a computer

Description automatically generated

-- Top 3 Max salary of cashier

SELECT name,salary

FROM Employees

WHERE role = 'Cashier'

ORDER BY salary DESC

LIMIT 3;

A screenshot of a computer

Description automatically generated

**Conclusion**

This design establishes a well-structured relational database for a cake shop, allowing for comprehensive management of inventory, products, customer orders, payments, and employee data. With clearly defined relationships and fields, this system is highly scalable, supporting day-to-day operations, stock control, and tracking sales data, ultimately providing insights into business performance and growth.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* THE END \*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*