Pizza Shop Database Cloud Setup and Overview and Execution Documentation

Cloud Setup (AWS RDS)

The Pizza Shop Database is hosted on Amazon Web Services (AWS) RDS (Relational Database Service) using MySQL 8.0 as the database engine. The cloud setup allows secure access for the frontend and backend applications to communicate with the database.

Steps Taken for Cloud Setup

1. **Created an AWS RDS MySQL Instance**
   a. Selected MySQL 8.0 as the database engine.
   b. Chose a cost-effective instance type.
   c. Set the database name as `pizza_shop`.
   d. Configured the admin username and password for access.
2. **Configured Network & Security**
   a. Set up a Virtual Private Cloud (VPC) for database security.
   b. Created a security group and modified inbound rules to allow connections from MySQL Workbench and the application.
3. **Connected AWS RDS to MySQL Workbench**
   a. Retrieved the database endpoint from AWS RDS.
   b. Used MySQL Workbench to connect to the database remotely.
   c. Verified the connection and created database tables.

Database Structure & Purpose

The Pizza Shop Database is designed to: Store customer details
Manage orders and order items
Track order statuses
Process payments
Store pizza menu items

Tables in the database:

Table Name	Purpose
customers	Stores customer details (name, phone, address, email).
orders	Stores order details, special instructions, and total price.
order_items	Links orders to pizzas, storing quantity of each pizza in an order.
order_status_updates	Tracks status changes for each order.
payments	Stores payment details for each order.
pizzas	Stores pizza menu, including name, description, price, and optional image.

Key Functionalities Implemented

1. Order Processing

- Customers can place orders by selecting pizzas.
- The database stores customer information and assigns a unique order ID.

2. Order Status Tracking

- Orders start as Pending and automatically update to Preparing using a trigger.
- Backend can update status to Out for Delivery and Delivered.

3. Payment Management

- Each order can be paid using Cash, Card, or Online Payment.
- The total order amount is calculated and stored.

Stored Procedures & Triggers

Stored Procedure: `GetOrderDetails`

- Retrieves customer name, phone, order date, status, and ordered pizzas for a specific order.

Stored Procedure: `UpdateOrderTotal`

- Automatically updates the total order cost based on the pizzas and quantities ordered.

Trigger: `after_order_insert`

- Automatically inserts an order status update when a new order is placed without backend intervention, setting the initial status to Preparing.

Indexing for Performance Optimization

- Indexing on frequently searched columns like phone (customers), status(orders) for faster queries.
- Foreign keys used for data consistency and referential integrity.

Execution Steps: How to Set Up & Run the SQL File Locally

Prerequisites

- Install **MySQL Server** (Version 8.0 or later).
- Install **MySQL Workbench** (Optional)
- Ensure **MySQL is running** on your local machine.
- Place the **pizza_shop.sql** file in a known directory (e.g., `Downloads` folder).

Method 1: Using MySQL Command Line (Recommended)

Open **MySQL Command Line Client**.
Enter the **MySQL root password** when prompted.

Switch to your **Downloads** folder (or wherever the file is stored).

Run the SQL file using:

SOURCE C:/Users/your-username/Downloads/pizza_shop.sql;

The script will **create the database, tables, and insert sample data**.

💡 **To verify the database:**

- Use `USE pizza_shop;`
- Run `SHOW TABLES;` to list all created tables.
- Run `SELECT * FROM orders;` to view inserted orders.

Method 2: Using MySQL Workbench

Open **MySQL Workbench**.

Click on your **local MySQL connection**.
Open a **new query tab** (File → New Query).
□Click File → Open SQL Script and **select pizza_shop.sql**.
Click **Run** (Do NOT click "Execute" in production!).

Troubleshooting Common Issues

Error: "Table already exists"

Solution:

- Drop existing tables before running the script.

DROP DATABASE IF EXISTS pizza_shop;

Error: "Duplicate entry for key customers.phone"

Solution:

- Ensure you **don't insert the same sample data multiple times**.
- Use TRUNCATE TABLE customers; before re-running the script.

🔹 **Error: "Foreign key constraint fails"**

Solution:

- Check if pizzas table exists before inserting order_items.
- Try disabling foreign key checks temporarily:

```
SET FOREIGN_KEY_CHECKS=0;
```