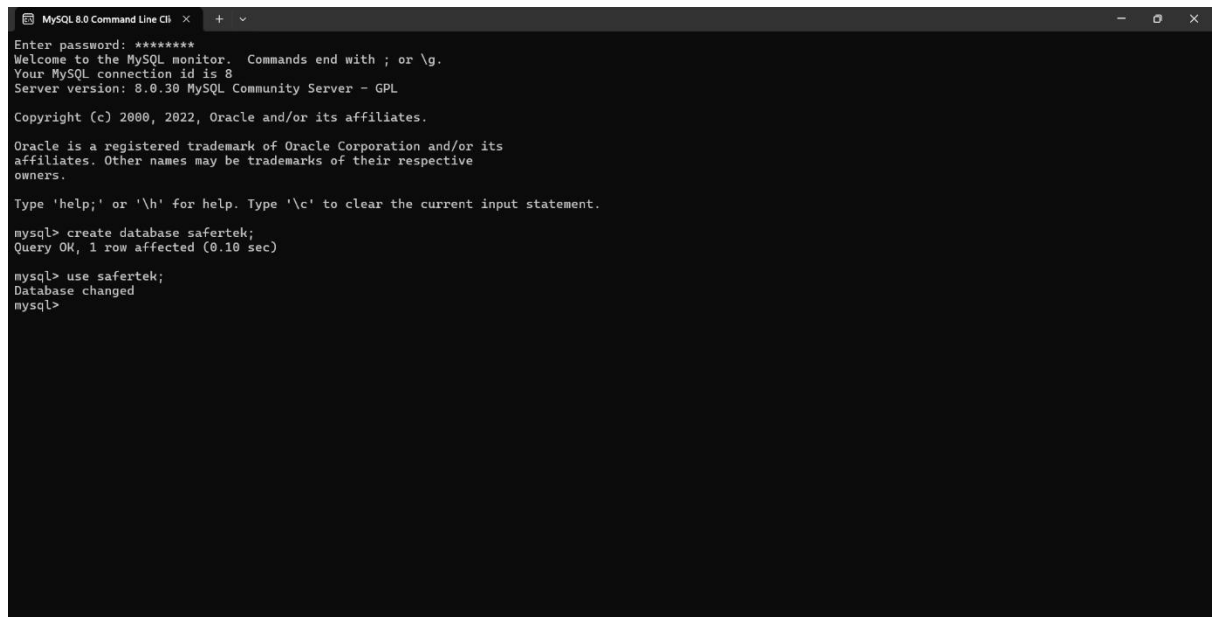


Safertek SQLBackend

ID: 2100032420

NAME: Venkataramana Baratam

1. CREATE DATABASE safertek; USE safertek;



```
MySQL 8.0 Command Line Cli x + v
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.30 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database safertek;
Query OK, 1 row affected (0.10 sec)

mysql> use safertek;
Database changed
mysql>
```

2. CREATE TABLE Customers (CustomerID INT PRIMARY KEY, FirstName VARCHAR(50), LastName VARCHAR(50), Email VARCHAR(100), DateOfBirth DATE);

CREATE TABLE Products (ProductID INT PRIMARY KEY, ProductName VARCHAR(100), Price DECIMAL(10, 2));

CREATE TABLE Orders (OrderID INT PRIMARY KEY, CustomerID INT, OrderDate DATE, FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID));

CREATE TABLE OrderItems (OrderItemID INT PRIMARY KEY, OrderID INT, ProductID INT, Quantity INT, FOREIGN KEY (OrderID) REFERENCES Orders(OrderID), FOREIGN KEY (ProductID) REFERENCES Products(ProductID));

```
MySQL 8.0 Command Line C...
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use safartek;
Database changed
mysql> CREATE TABLE Customers (
  -> CustomerID INT PRIMARY KEY,
  -> FirstName VARCHAR(50),
  -> LastName VARCHAR(50),
  -> Email VARCHAR(100),
  -> DateOfBirth DATE
  -> );
Query OK, 0 rows affected (0.13 sec)

mysql> CREATE TABLE Products (
  -> ProductID INT PRIMARY KEY,
  -> ProductName VARCHAR(100),
  -> Price DECIMAL(10, 2)
  -> );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE Orders (
  -> OrderID INT PRIMARY KEY,
  -> CustomerID INT,
  -> OrderDate DATE,
  -> FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
  -> );
Query OK, 0 rows affected (0.07 sec)

mysql> show tables;
+-----+
| Tables_in_safartek |
+-----+
| customers           |
| orders              |
| products            |
+-----+
3 rows in set (0.06 sec)

mysql>
```

3. INSERT INTO Customers (CustomerID, FirstName, LastName, Email, DateOfBirth) VALUES (1, 'John', 'Doe', 'john.doe@example.com', '1985-01-15'), (2, 'Jane', 'Smith', 'jane.smith@example.com', '1990-06-20');

INSERT INTO Products (ProductID, ProductName, Price) VALUES (1, 'Laptop', 1000.00), (2, 'Smartphone', 600.00), (3, 'Headphones', 100.00);

INSERT INTO Orders (OrderID, CustomerID, OrderDate) VALUES (1, 1, '2023-01-10'), (2, 2, '2023-01-12');

INSERT INTO OrderItems (OrderItemID, OrderID, ProductID, Quantity) VALUES (1, 1, 1, 1), (2, 1, 3, 2), (3, 2, 2, 1), (4, 2, 3, 1);

```
MySQL 8.0 Command Line C...
| products
+-----+
3 rows in set (0.06 sec)

mysql> CREATE TABLE OrderItems (
  -> OrderItemID INT PRIMARY KEY,
  -> OrderID INT,
  -> ProductID INT,
  -> Quantity INT,
  -> FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
  -> FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
  -> );
Query OK, 0 rows affected (0.09 sec)

mysql> INSERT INTO Customers (CustomerID, FirstName, LastName, Email, DateOfBirth) VALUES
  -> (1, 'John', 'Doe', 'john.doe@example.com', '1985-01-15'),
  -> (2, 'Jane', 'Smith', 'jane.smith@example.com', '1990-06-20');
Query OK, 2 rows affected (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Products (ProductID, ProductName, Price) VALUES
  -> (1, 'Laptop', 1000.00),
  -> (2, 'Smartphone', 600.00),
  -> (3, 'Headphones', 100.00);
Query OK, 3 rows affected (0.01 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Orders (OrderID, CustomerID, OrderDate) VALUES
  -> (1, 1, '2023-01-10'),
  -> (2, 2, '2023-01-12');
Query OK, 2 rows affected (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> INSERT INTO OrderItems (OrderItemID, OrderID, ProductID, Quantity) VALUES
  -> (1, 1, 1, 1),
  -> (2, 1, 3, 2),
  -> (3, 2, 2, 1),
  -> (4, 2, 3, 1);
Query OK, 4 rows affected (0.02 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

4. To list all customers from the Customers table:

Query: `SELECT * FROM Customers;`

```
mysql> select * from Customers;
+-----+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | Email | DateOfBirth |
+-----+-----+-----+-----+-----+
| 1 | John | Doe | john.doe@example.com | 1985-01-15 |
| 2 | Jane | Smith | jane.smith@example.com | 1990-06-20 |
+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> select * from Products;
+-----+-----+-----+
| ProductID | ProductName | Price |
+-----+-----+-----+
| 1 | Laptop | 1000.00 |
| 2 | Smartphone | 600.00 |
| 3 | Headphones | 100.00 |
+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> select * from Orders;
+-----+-----+-----+
| OrderID | CustomerID | OrderDate |
+-----+-----+-----+
| 1 | 1 | 2023-01-10 |
| 2 | 2 | 2023-01-12 |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from OrderItems;
+-----+-----+-----+-----+
| OrderItemID | OrderID | ProductID | Quantity |
+-----+-----+-----+-----+
| 1 | 1 | 1 | 1 |
| 2 | 1 | 3 | 2 |
| 3 | 2 | 2 | 1 |
| 4 | 2 | 3 | 1 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> |
```

5. To find all orders placed in January 2023:

Query: `SELECT * FROM Orders WHERE OrderDate >= '2023-01-01' AND OrderDate <= '2023-01-31';`

```
mysql> SELECT * FROM Orders WHERE OrderDate >= '2023-01-01' AND OrderDate <= '2023-01-31';
+-----+-----+-----+
| OrderID | CustomerID | OrderDate |
+-----+-----+-----+
| 1 | 1 | 2023-01-10 |
| 2 | 2 | 2023-01-12 |
+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> |
```

6. To get the details of each order, including the customer's name and email.

Query: `SELECT O.OrderID, O.OrderDate, C.FirstName, C.LastName, C.Email FROM Orders O JOIN Customers C ON O.CustomerID = C.CustomerID;`

```
mysql> SELECT O.OrderID, O.OrderDate, C.FirstName, C.LastName, C.Email
-> FROM Orders O JOIN Customers C ON O.CustomerID = C.CustomerID;
+-----+-----+-----+-----+-----+
| OrderID | OrderDate | FirstName | LastName | Email |
+-----+-----+-----+-----+-----+
| 1 | 2023-01-10 | John | Doe | john.doe@example.com |
| 2 | 2023-01-12 | Jane | Smith | jane.smith@example.com |
+-----+-----+-----+-----+-----+
2 rows in set (0.02 sec)

mysql> |
```

7. To list the products purchased in a specific order (e.g., OrderID = 1).

Query: `SELECT P.ProductName, O.Quantity FROM OrderItems O JOIN Products P ON O.ProductID = P.ProductID WHERE O.OrderID = 1;`

```
mysql> SELECT P.ProductName, O.Quantity FROM OrderItems O
-> JOIN Products P ON O.ProductID = P.ProductID WHERE O.OrderID = 1;
+-----+-----+
| ProductName | Quantity |
+-----+-----+
| Laptop | 1 |
| Headphones | 2 |
+-----+-----+
2 rows in set (0.01 sec)

mysql> |
```

8. To calculate the total amount spent by each customer.

Query: SELECT C.CustomerID, C.FirstName, C.LastName, C.Email, SUM(P.Price * OI.Quantity) AS TotalAmountSpent FROM Customers C JOIN Orders O ON C.CustomerID = O.CustomerID JOIN OrderItems OI ON O.OrderID = OI.OrderID JOIN Products P ON OI.ProductID = P.ProductID GROUP BY C.CustomerID, C.FirstName, C.LastName, C.Email;

```
mysql> SELECT C.CustomerID, C.FirstName, C.LastName, C.Email, SUM(P.Price * OI.Quantity) AS TotalAmountSpent FROM Customers C JOIN Orders O ON C.CustomerID = O.CustomerID
-> JOIN OrderItems OI ON O.OrderID = OI.OrderID JOIN Products P ON OI.ProductID = P.ProductID GROUP BY C.CustomerID, C.FirstName, C.LastName, C.Email;
+-----+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | Email | TotalAmountSpent |
+-----+-----+-----+-----+-----+
| 1 | John | Doe | john.doe@example.com | 1200.00 |
| 2 | Jane | Smith | jane.smith@example.com | 700.00 |
+-----+-----+-----+-----+-----+
2 rows in set (0.02 sec)

mysql> |
```

9. To find the most popular product (the one that has been ordered the most).

Query: SELECT P.ProductID, P.ProductName, SUM(OI.Quantity) AS TotalQuantityOrdered FROM Products P JOIN OrderItems OI ON P.ProductID = OI.ProductID GROUP BY P.ProductID, P.ProductName ORDER BY TotalQuantityOrdered DESC LIMIT 1;

```
mysql> SELECT P.ProductID, P.ProductName, SUM(OI.Quantity) AS TotalQuantityOrdered
-> FROM Products P JOIN OrderItems OI ON P.ProductID = OI.ProductID
-> GROUP BY P.ProductID, P.ProductName ORDER BY TotalQuantityOrdered DESC LIMIT 1;
+-----+-----+-----+
| ProductID | ProductName | TotalQuantityOrdered |
+-----+-----+-----+
| 3 | Headphones | 3 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> |
```

10. To get the total number of orders and the total sales amount for each month in 2023.

Query: SELECT YEAR(OrderDate) AS Year, MONTH(OrderDate) AS Month, COUNT(*) AS TotalOrders, SUM(Price * Quantity) AS TotalSalesAmount FROM Orders O JOIN OrderItems ON O.OrderID = OrderItems.OrderID JOIN Products ON OrderItems.ProductID = Products.ProductID WHERE YEAR(OrderDate) = 2023 GROUP BY YEAR(OrderDate), MONTH(OrderDate) ORDER BY YEAR(OrderDate), MONTH(OrderDate);

```
mysql> SELECT YEAR(OrderDate) AS Year, MONTH(OrderDate) AS Month, COUNT(*) AS TotalOrders, SUM(Price * Quantity) AS TotalSalesAmount FROM Orders O
-> JOIN OrderItems ON O.OrderID = OrderItems.OrderID JOIN Products ON OrderItems.ProductID = Products.ProductID WHERE YEAR(OrderDate) = 2023 GROUP BY Y
EAR(OrderDate), MONTH(OrderDate) ORDER BY YEAR(OrderDate), MONTH(OrderDate);
+-----+-----+-----+-----+
| Year | Month | TotalOrders | TotalSalesAmount |
+-----+-----+-----+-----+
| 2023 | 1 | 4 | 1900.00 |
+-----+-----+-----+-----+
1 row in set (0.01 sec)

mysql> |
```

11. To find customers who have spent more than \$1000.

Query: SELECT C.CustomerID, C.FirstName, C.LastName, C.Email, SUM(P.Price * OI.Quantity) AS TotalSpent FROM Customers C JOIN Orders O ON C.CustomerID = O.CustomerID JOIN OrderItems OI ON O.OrderID = OI.OrderID JOIN Products P ON OI.ProductID = P.ProductID GROUP BY C.CustomerID, C.FirstName, C.LastName, C.Email HAVING TotalSpent > 1000;

```
mysql> SELECT C.CustomerID, C.FirstName, C.LastName, C.Email, SUM(P.Price * OI.Quantity) AS TotalSpent FROM Customers C JOIN Orders O ON C.CustomerID = O.CustomerID JOIN OrderItems OI ON O.OrderID = OI.OrderID JOIN Products P ON OI.ProductID = P.ProductID
-> GROUP BY C.CustomerID, C.FirstName, C.LastName, C.Email HAVING TotalSpent > 1000;
+-----+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | Email | TotalSpent |
+-----+-----+-----+-----+-----+
| 1 | John | Doe | john.doe@example.com | 1200.00 |
+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

mysql> |
```