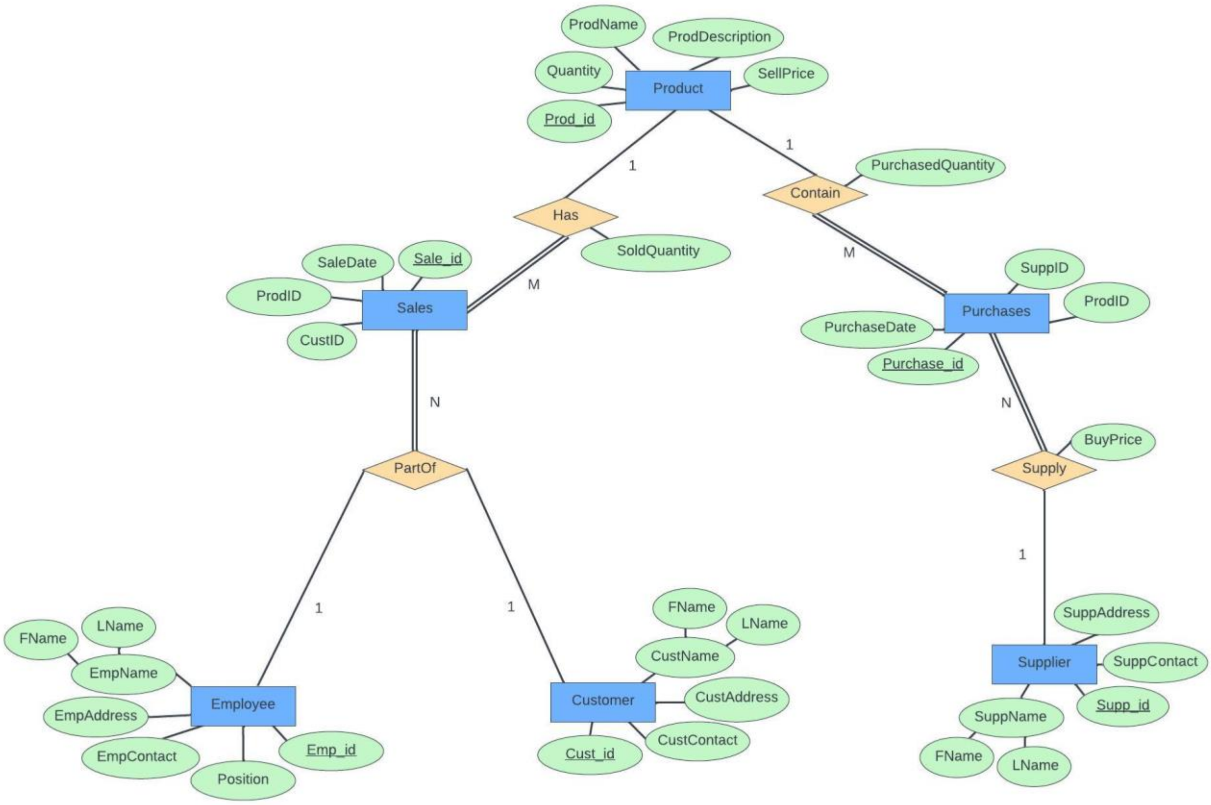
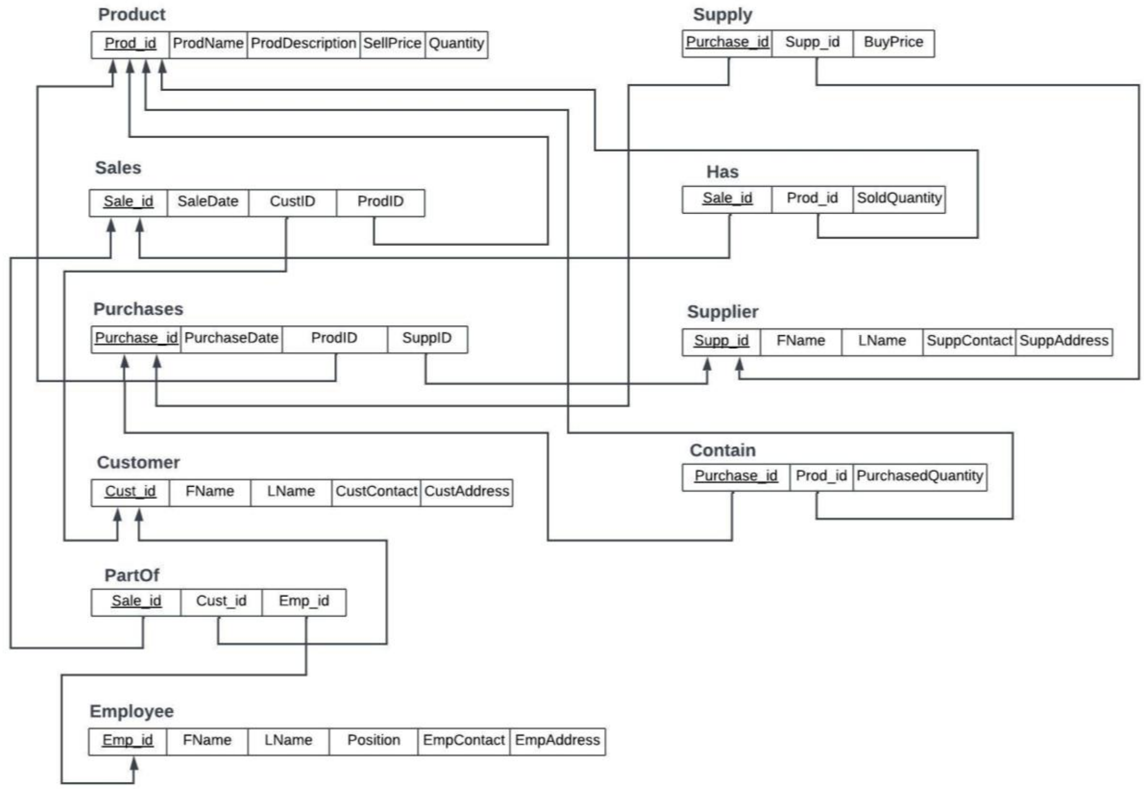
Supermarket Management System

Shardul Shingare

ER Diagram:



Relational Schema:


Product
Prod_id | ProdName | ProdDescription | SellPrice | Quantity
Supply
Purchase_id | Supp_id | BuyPrice
Sales
Sale_id | SaleDate | CustID | ProdID
Has
Sale_id | Prod_id | SoldQuantity
Purchases
Purchase_id | PurchaseDate | ProdID | SuppID
Supplier
Supp_id | FName | LName | SuppContact | SuppAddress
Customer
Cust_id | FName | LName | CustContact | CustAddress
Contain
Purchase_id | Prod_id | PurchasedQuantity
PartOf
Sale_id | Cust_id | Emp_id
Employee
Emp_id | FName | LName | Position | EmpContact | EmpAddress

This relational schema is already in 3NF, I have pointed out the reasons for this below. Moreover, I have broken down the Employee, Customer, Supplier names into FName and LName because they were composite attributes.

product

Prod_id	ProdName	ProdDescription	SellPrice	Quantity

Since every attribute in the 'product' relation is atomic, it is already in 1NF. Only one attribute constitutes the primary key here, hence there does not exist any partial dependency and so the relation is in 2NF. The ProdName, ProdDescription, SellPrice, Quantity attributes are functionally dependent on only the Prod_id in this relation and hence, there exists no transitive dependency and so the relation is in 3NF.

customer

Cust_id	FName	LName	CustContact	CustAddress

Every attribute in the 'customer' relation is atomic, so it is in 1NF. Only one attribute constitutes the primary key (Cust_id), and all other attributes are dependent on this, hence there does not exist any partial dependency and so the relation is in 2NF. The FName, LName, CustContact, CustAddress attributes are functionally dependent on only the Cust_id in this relation and hence, there exists no transitive dependency and so the relation is in 3NF.

purchases

Purchase_id	PurchaseDate	ProdID	SuppID

Every attribute in the ‘purchases’ relation is atomic, so it is in 1NF. Only one attribute constitutes the primary key (Purchase_id), and all other attributes are dependent on this, hence there does not exist any partial dependency and so the relation is in 2NF. The PurchaseDate, ProdID, SuppID attributes are functionally dependent on only the Purchase_id in this relation and hence, there exists no transitive dependency and so the relation is in 3NF.

supplier

Supp_id	FName	LName	SuppContact	SuppAddress

Every attribute in the ‘supplier’ relation is atomic, so it is in 1NF. Only Supp_id constitutes the primary key and all other attributes are dependent on this, hence there does not exist any partial dependency and so the relation is in 2NF. The FName, LName, SuppContact, SuppAddress attributes are functionally dependent only on the Supp_id in this relation and hence, there exists no transitive dependency and so the relation is in 3NF.

employee

Emp_id	FName	LName	Position	EmplContact	EmplAddress

Every attribute in the ‘employee’ relation is atomic, so it is in 1NF. Only Emp_id constitutes the primary key and all other attributes are dependent on this, hence there does not exist any partial dependency and so the relation is in 2NF. The FName, LName, Position, EmplContact, EmplAddress attributes are functionally dependent only on the Emp_id in this relation and hence, there exists no transitive dependency and so the relation is in 3NF.

sales

Sale_id	SaleDate	CustID	ProdID

Every attribute in the ‘sales’ relation is atomic, so it is in 1NF. Only Sale_id constitutes the primary key and all other attributes are dependent on this, hence there does not exist any partial dependency and so the relation is in 2NF. The SaleDate, CustID and ProdID attributes are functionally dependent only on the Sale_id and hence, there exists no transitive dependency and so the relation is in 3NF.

contain

Purchase_id	ProdID	PurchasedQuantity

Every attribute in the ‘contain’ relation is atomic, so it is in 1NF. Only Purchase_id constitutes the primary key and all other attributes are dependent on this, hence there does not exist any partial dependency and so the relation is in 2NF. The ProdID and PurchasedQuantity attributes are functionally dependent only on the Purchase_id in this relation and hence, there exists no transitive dependency and so the relation is in 3NF.

supply

Purchase_id	Supp_id	BuyPrice

Every attribute in the ‘supply’ relation is atomic, so it is in 1NF. Only Purchase_id constitutes the primary key and all other attributes are dependent on this, hence there does not exist any partial dependency and so the relation is in 2NF. The Supp_id and BuyPrice attributes are functionally dependent only on the Purchase_id in this relation and hence, there exists no transitive dependency and so the relation is in 3NF.

has

Sale_id	Prod_id	SoldQuantity

Every attribute in the ‘has’ relation is atomic, so it is in 1NF. Only Sale_id constitutes the primary key and all other attributes are dependent on this, hence there does not exist any partial dependency and so the relation is in 2NF. The Prod_id and SoldQuantity attributes are functionally dependent only on the Sale_id in this relation and hence, there exists no transitive dependency and so the relation is in 3NF.

partof

Sale_id	Cust_id	Emp_id

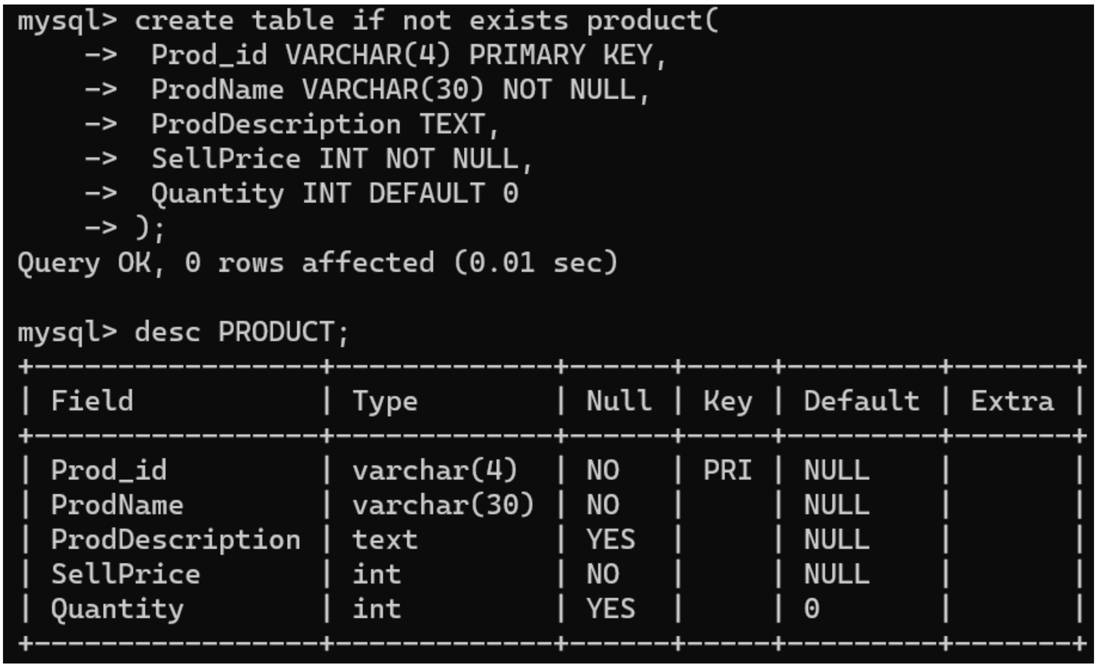
Every attribute in the ‘partof’ relation is atomic, so it is in 1NF. Only Sale_id constitutes the primary key and all other attributes are dependent on this, hence there does not exist any partial dependency and so the relation is in 2NF. The Cust_id and Emp_id attributes are functionally dependent only on the Sale_id in this relation and hence, there exists no transitive dependency and so the relation is in 3NF.

The constraints for each relation can be viewed in the SM_makingtables.sql file.

User requests (queries and their outputs):

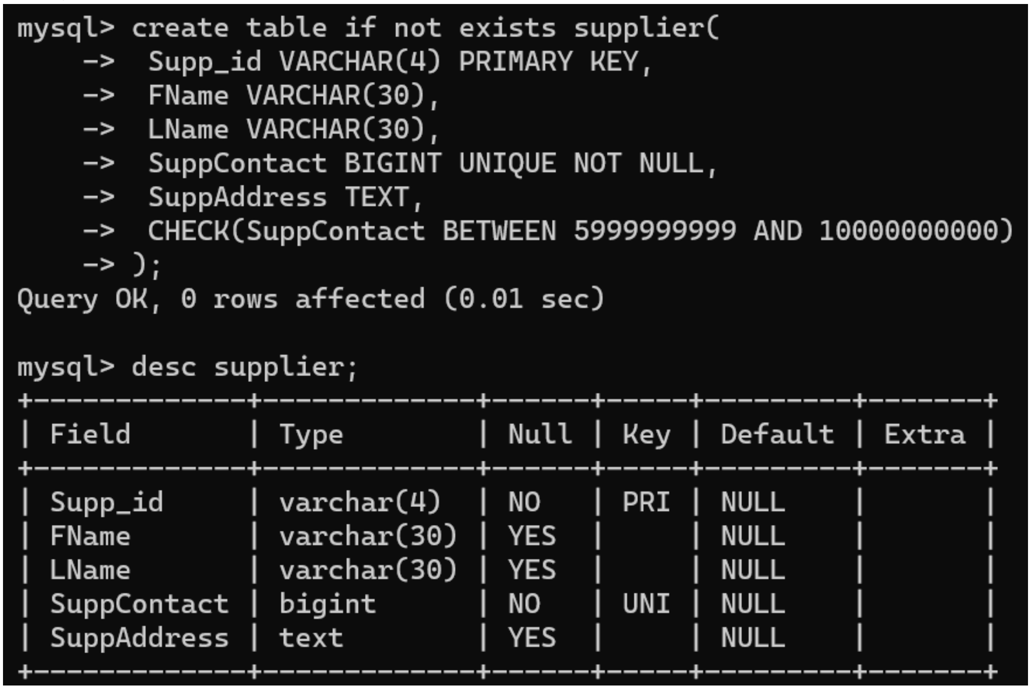
(First ten queries i.e. the table creation and foreign key queries are in the SM_makingtables.sql file, the rest ten are in the SM_queries.sql file)

1. Create a table for products, including columns for product name, description, price, and quantity in stock.



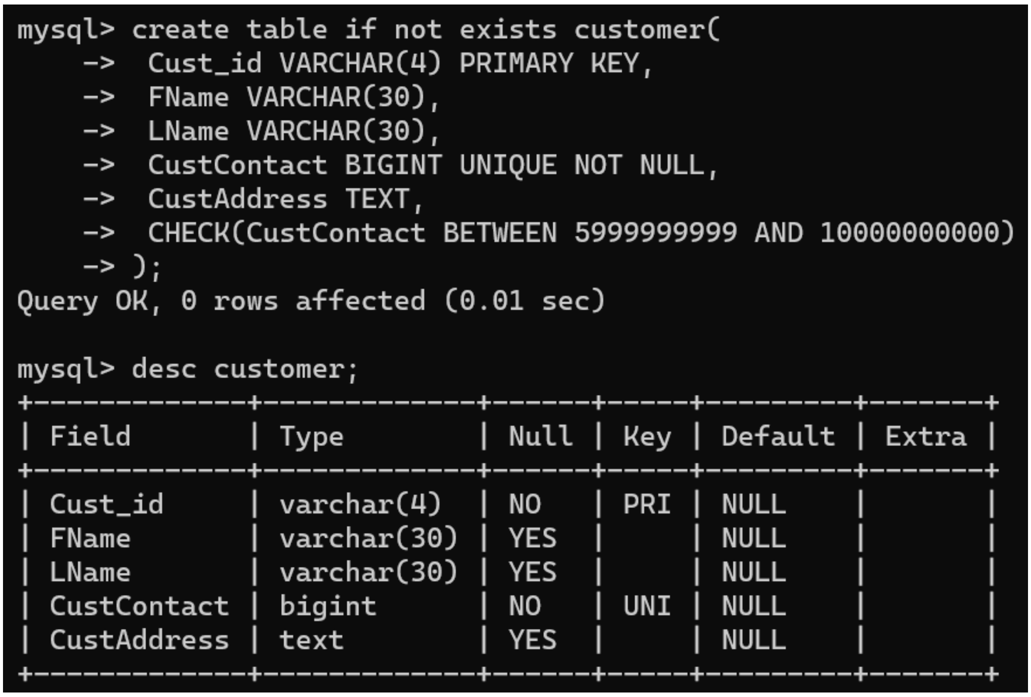
```
mysql> create table if not exists product(
    ->  Prod_id VARCHAR(4) PRIMARY KEY,
    ->  ProdName VARCHAR(30) NOT NULL,
    ->  ProdDescription TEXT,
    ->  SellPrice INT NOT NULL,
    ->  Quantity INT DEFAULT 0
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> desc PRODUCT;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| Prod_id         | varchar(4)  | NO   | PRI | NULL    |       |
| ProdName        | varchar(30) | NO   |     | NULL    |       |
| ProdDescription | text        | YES  |     | NULL    |       |
| SellPrice       | int         | NO   |     | NULL    |       |
| Quantity        | int         | YES  |     | 0       |       |
+-----------------+-------------+------+-----+---------+-------+
```

2. Create a table for suppliers, including columns for supplier name, address, and contact information.



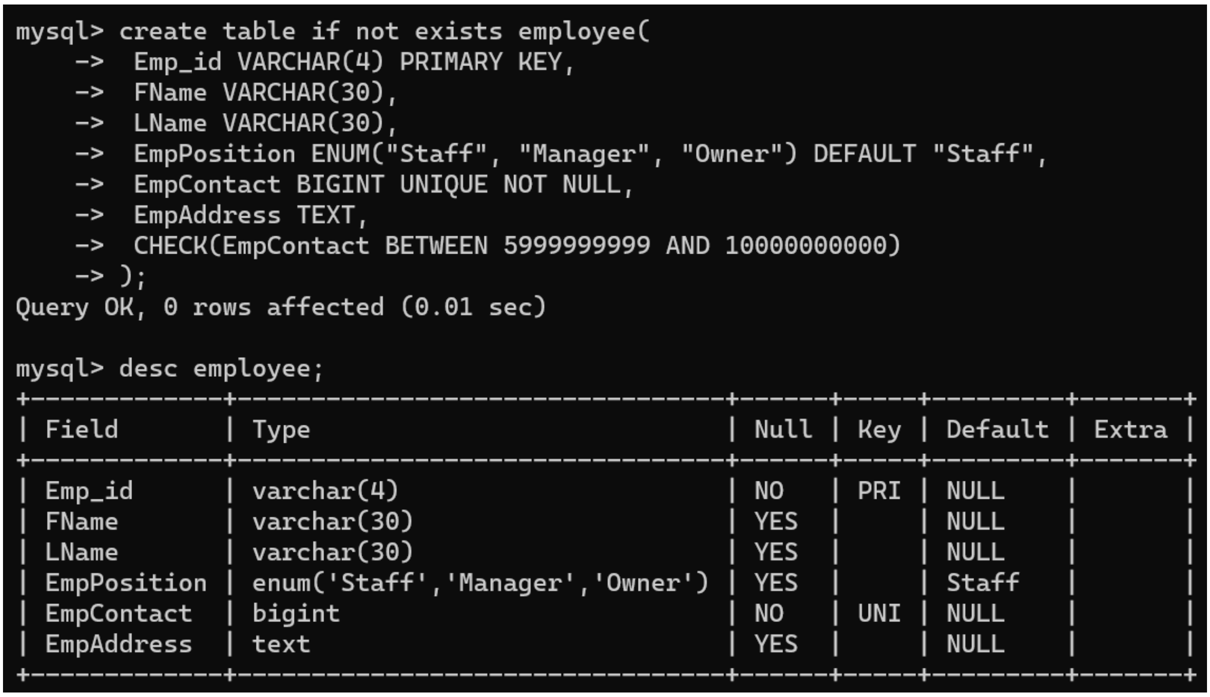
```
mysql> create table if not exists supplier(
    ->  Supp_id VARCHAR(4) PRIMARY KEY,
    ->  FName VARCHAR(30),
    ->  LName VARCHAR(30),
    ->  SuppContact BIGINT UNIQUE NOT NULL,
    ->  SuppAddress TEXT,
    ->  CHECK(SuppContact BETWEEN 5999999999 AND 10000000000)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> desc supplier;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Supp_id     | varchar(4)  | NO   | PRI | NULL    |       |
| FName       | varchar(30) | YES  |     | NULL    |       |
| LName       | varchar(30) | YES  |     | NULL    |       |
| SuppContact | bigint      | NO   | UNI | NULL    |       |
| SuppAddress | text        | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
```

3. Create a table for customers, including columns for customer name, address, and contact information.



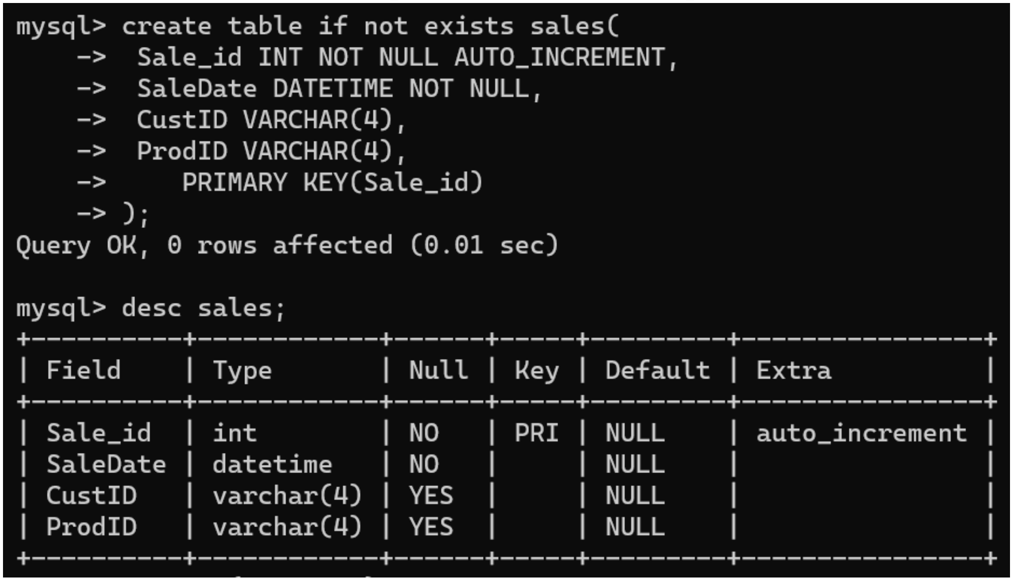
```
mysql> create table if not exists customer(
    ->  Cust_id VARCHAR(4) PRIMARY KEY,
    ->  FName VARCHAR(30),
    ->  LName VARCHAR(30),
    ->  CustContact BIGINT UNIQUE NOT NULL,
    ->  CustAddress TEXT,
    ->  CHECK(CustContact BETWEEN 5999999999 AND 10000000000)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> desc customer;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Cust_id     | varchar(4)  | NO   | PRI | NULL    |       |
| FName       | varchar(30) | YES  |     | NULL    |       |
| LName       | varchar(30) | YES  |     | NULL    |       |
| CustContact | bigint      | NO   | UNI | NULL    |       |
| CustAddress | text        | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
```

4. Create a table for employees, including columns for employee name, address, contact information, and job position.



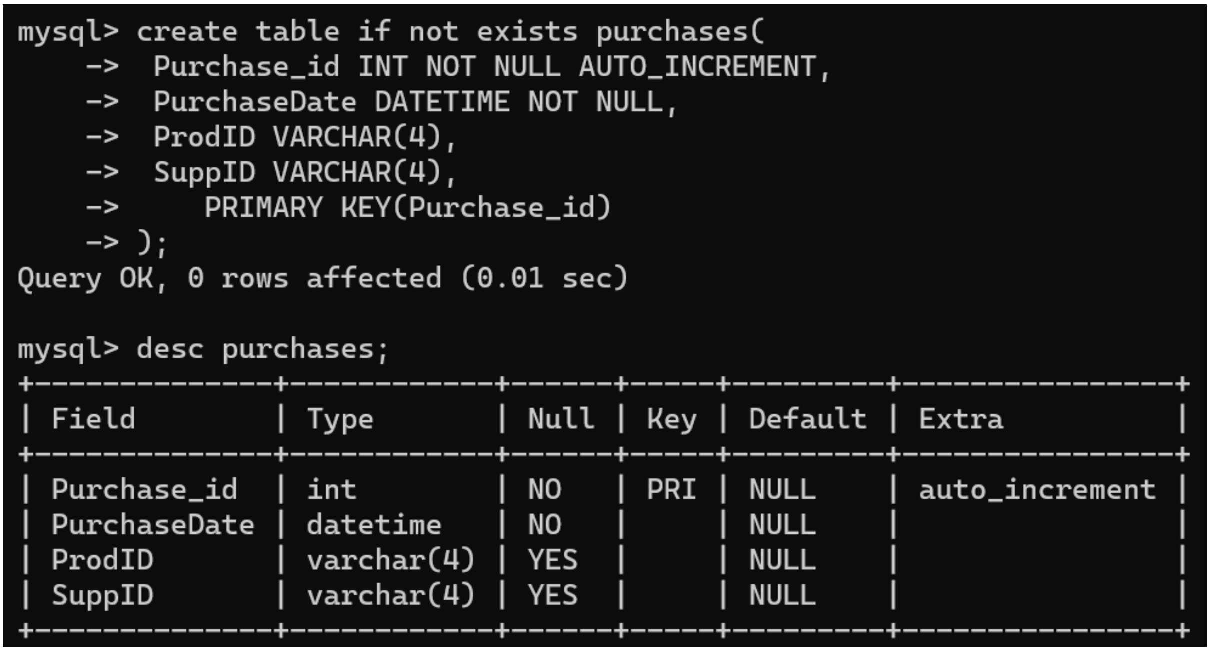
```
mysql> create table if not exists employee(
    ->  Emp_id VARCHAR(4) PRIMARY KEY,
    ->  FName VARCHAR(30),
    ->  LName VARCHAR(30),
    ->  EmpPosition ENUM("Staff", "Manager", "Owner") DEFAULT "Staff",
    ->  EmpContact BIGINT UNIQUE NOT NULL,
    ->  EmpAddress TEXT,
    ->  CHECK(EmpContact BETWEEN 5999999999 AND 10000000000)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> desc employee;
+-------------+--------------------------------+------+-----+---------+-------+
| Field       | Type                           | Null | Key | Default | Extra |
+-------------+--------------------------------+------+-----+---------+-------+
| Emp_id      | varchar(4)                     | NO   | PRI | NULL    |       |
| FName       | varchar(30)                    | YES  |     | NULL    |       |
| LName       | varchar(30)                    | YES  |     | NULL    |       |
| EmpPosition | enum('Staff','Manager','Owner') | YES  |     | Staff   |       |
| EmpContact  | bigint                         | NO   | UNI | NULL    |       |
| EmpAddress  | text                           | YES  |     | NULL    |       |
+-------------+--------------------------------+------+-----+---------+-------+
```

5. Create a table for sales, including columns for the date of the sale, the customer who made the purchase, and the products sold.



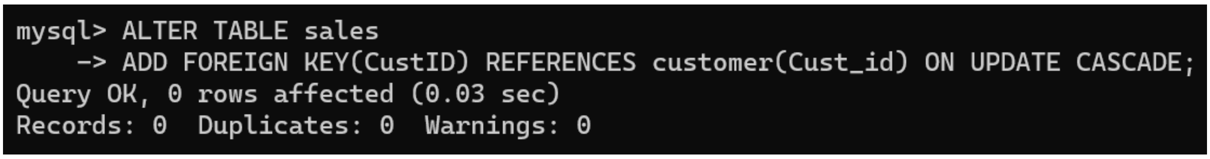
```
mysql> create table if not exists sales(
    ->  Sale_id INT NOT NULL AUTO_INCREMENT,
    ->  SaleDate DATETIME NOT NULL,
    ->  CustID VARCHAR(4),
    ->  ProdID VARCHAR(4),
    ->     PRIMARY KEY(Sale_id)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> desc sales;
+----------+------------+------+-----+---------+----------------+
| Field    | Type       | Null | Key | Default | Extra          |
+----------+------------+------+-----+---------+----------------+
| Sale_id  | int        | NO   | PRI | NULL    | auto_increment |
| SaleDate | datetime   | NO   |     | NULL    |                |
| CustID   | varchar(4) | YES  |     | NULL    |                |
| ProdID   | varchar(4) | YES  |     | NULL    |                |
+----------+------------+------+-----+---------+----------------+
```

6. Create a table for purchases, including columns for the date of the purchase, the supplier who provided the product, and the products purchased.



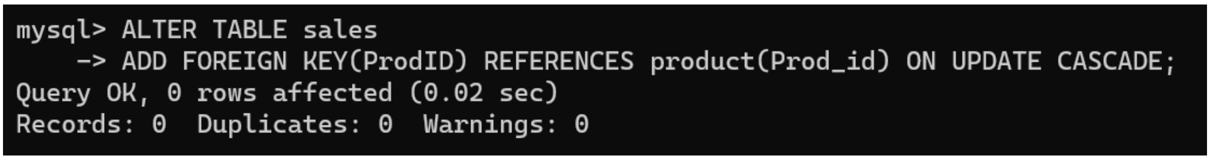
```
mysql> create table if not exists purchases(
    ->  Purchase_id INT NOT NULL AUTO_INCREMENT,
    ->  PurchaseDate DATETIME NOT NULL,
    ->  ProdID VARCHAR(4),
    ->  SuppID VARCHAR(4),
    ->     PRIMARY KEY(Purchase_id)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> desc purchases;
+--------------+------------+------+-----+---------+----------------+
| Field        | Type       | Null | Key | Default | Extra          |
+--------------+------------+------+-----+---------+----------------+
| Purchase_id  | int        | NO   | PRI | NULL    | auto_increment |
| PurchaseDate | datetime   | NO   |     | NULL    |                |
| ProdID       | varchar(4) | YES  |     | NULL    |                |
| SuppID       | varchar(4) | YES  |     | NULL    |                |
+--------------+------------+------+-----+---------+----------------+
```

7. Create a foreign key constraint between the sales table and the customers table, linking them on the customer ID.



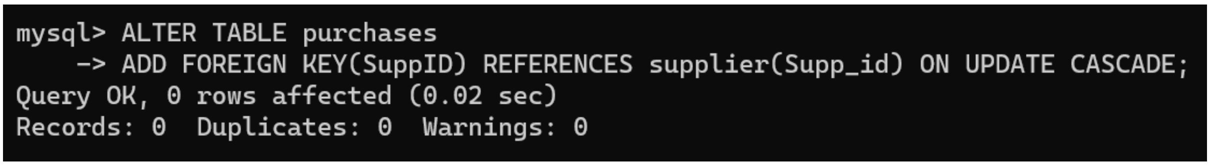
```
mysql> ALTER TABLE sales
    -> ADD FOREIGN KEY(CustID) REFERENCES customer(Cust_id) ON UPDATE CASCADE;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

8. Create a foreign key constraint between the sales table and the products table, linking them on the product ID.



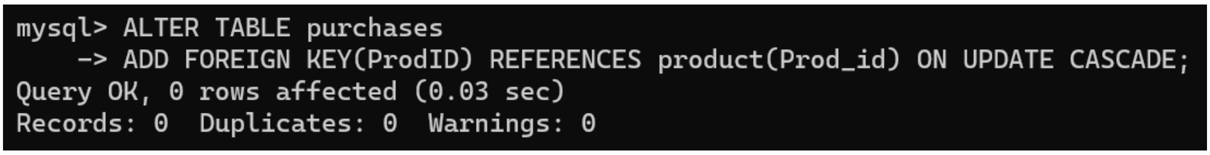
```
mysql> ALTER TABLE sales
    -> ADD FOREIGN KEY(ProdID) REFERENCES product(Prod_id) ON UPDATE CASCADE;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

9. Create a foreign key constraint between the purchases table and the suppliers table, linking them on the supplier ID.



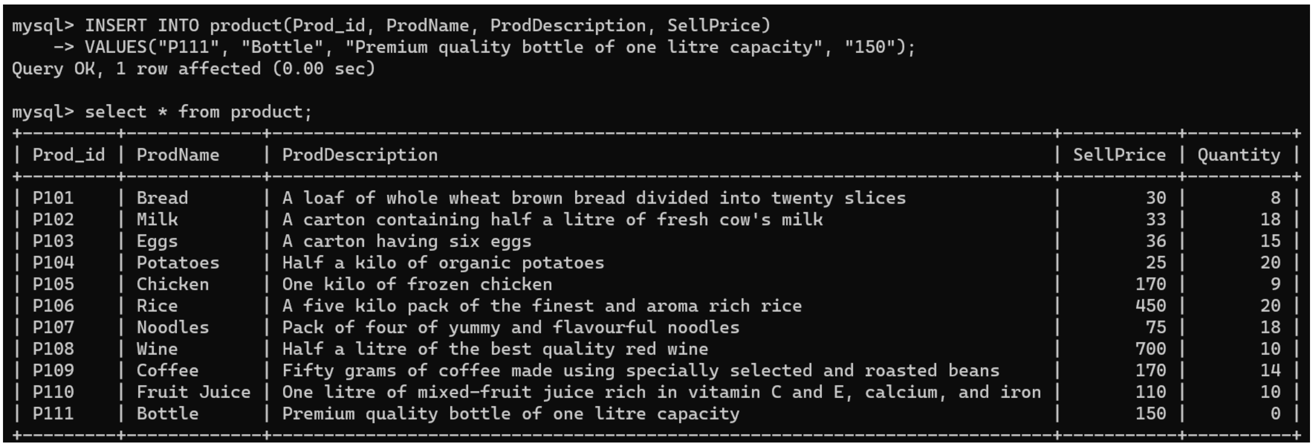
```
mysql> ALTER TABLE purchases
    -> ADD FOREIGN KEY(SuppID) REFERENCES supplier(Supp_id) ON UPDATE CASCADE;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

10. Create a foreign key constraint between the purchases table and the products table, linking them on the product ID.



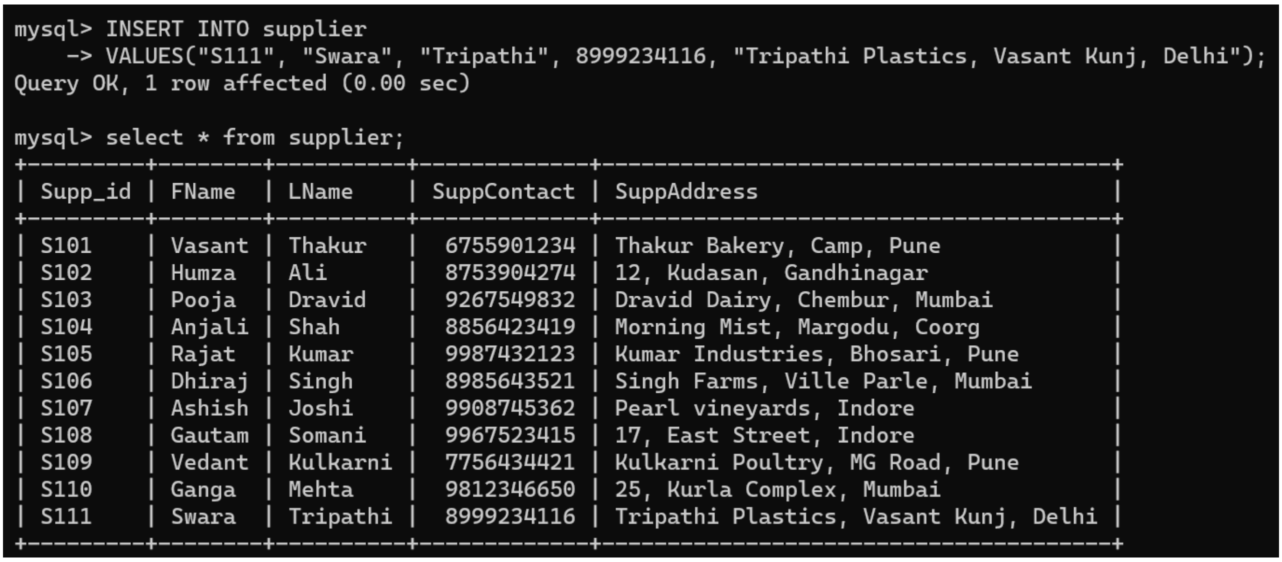
```
mysql> ALTER TABLE purchases
    -> ADD FOREIGN KEY(ProdID) REFERENCES product(Prod_id) ON UPDATE CASCADE;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

11. Insert a new product into the products table



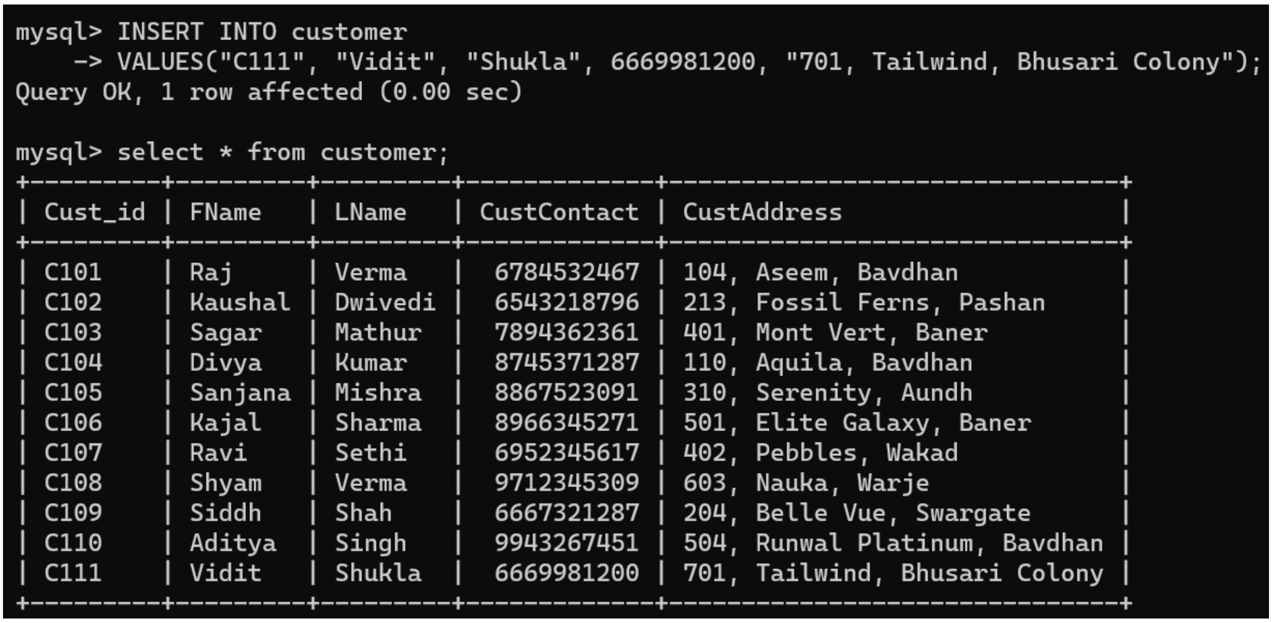
```
mysql> INSERT INTO product(Prod_id, ProdName, ProdDescription, SellPrice)
    -> VALUES("P111", "Bottle", "Premium quality bottle of one litre capacity", "150");
Query OK, 1 row affected (0.00 sec)

mysql> select * from product;
+---------+-------------+---------------------------------------------------------------------------+-----------+----------+
| Prod_id | ProdName    | ProdDescription                                                           | SellPrice | Quantity |
+---------+-------------+---------------------------------------------------------------------------+-----------+----------+
| P101    | Bread       | A loaf of whole wheat brown bread divided into twenty slices              |        30 |        8 |
| P102    | Milk        | A carton containing half a litre of fresh cow's milk                      |        33 |       18 |
| P103    | Eggs        | A carton having six eggs                                                  |        36 |       15 |
| P104    | Potatoes    | Half a kilo of organic potatoes                                           |        25 |       20 |
| P105    | Chicken     | One kilo of frozen chicken                                                |       170 |        9 |
| P106    | Rice        | A five kilo pack of the finest and aroma rich rice                        |       450 |       20 |
| P107    | Noodles     | Pack of four of yummy and flavourful noodles                              |        75 |       18 |
| P108    | Wine        | Half a litre of the best quality red wine                                 |       700 |       10 |
| P109    | Coffee      | Fifty grams of coffee made using specially selected and roasted beans     |       170 |       14 |
| P110    | Fruit Juice | One litre of mixed-fruit juice rich in vitamin C and E, calcium, and iron |       110 |       10 |
| P111    | Bottle      | Premium quality bottle of one litre capacity                              |       150 |        0 |
+---------+-------------+---------------------------------------------------------------------------+-----------+----------+
```

12. Insert a new supplier into the suppliers table



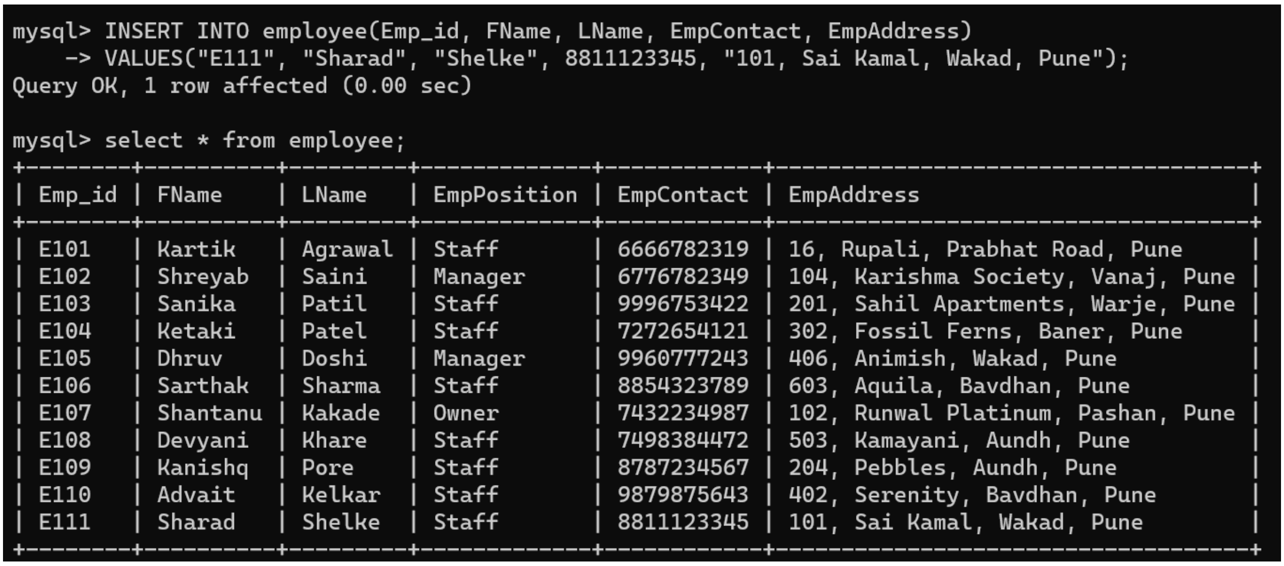
```
mysql> INSERT INTO supplier
    -> VALUES("S111", "Swara", "Tripathi", 8999234116, "Tripathi Plastics, Vasant Kunj, Delhi");
Query OK, 1 row affected (0.00 sec)

mysql> select * from supplier;
+---------+--------+----------+-------------+---------------------------------------+
| Supp_id | FName  | LName    | SuppContact | SuppAddress                           |
+---------+--------+----------+-------------+---------------------------------------+
| S101    | Vasant | Thakur   |  6755901234 | Thakur Bakery, Camp, Pune             |
| S102    | Humza  | Ali      |  8753904274 | 12, Kudasan, Gandhinagar              |
| S103    | Pooja  | Dravid   |  9267549832 | Dravid Dairy, Chembur, Mumbai         |
| S104    | Anjali | Shah     |  8856423419 | Morning Mist, Margodu, Coorg          |
| S105    | Rajat  | Kumar    |  9987432123 | Kumar Industries, Bhosari, Pune       |
| S106    | Dhiraj | Singh    |  8985643521 | Singh Farms, Ville Parle, Mumbai      |
| S107    | Ashish | Joshi    |  9908745362 | Pearl vineyards, Indore               |
| S108    | Gautam | Somani   |  9967523415 | 17, East Street, Indore               |
| S109    | Vedant | Kulkarni |  7756434421 | Kulkarni Poultry, MG Road, Pune       |
| S110    | Ganga  | Mehta    |  9812346650 | 25, Kurla Complex, Mumbai             |
| S111    | Swara  | Tripathi |  8999234116 | Tripathi Plastics, Vasant Kunj, Delhi |
+---------+--------+----------+-------------+---------------------------------------+
```

13. Insert a new customer into the customers table.



```
mysql> INSERT INTO customer
    -> VALUES("C111", "Vidit", "Shukla", 6669981200, "701, Tailwind, Bhusari Colony");
Query OK, 1 row affected (0.00 sec)

mysql> select * from customer;
+---------+---------+---------+-------------+--------------------------------+
| Cust_id | FName   | LName   | CustContact | CustAddress                    |
+---------+---------+---------+-------------+--------------------------------+
| C101    | Raj     | Verma   |  6784532467 | 104, Aseem, Bavdhan            |
| C102    | Kaushal | Dwivedi |  6543218796 | 213, Fossil Ferns, Pashan      |
| C103    | Sagar   | Mathur  |  7894362361 | 401, Mont Vert, Baner          |
| C104    | Divya   | Kumar   |  8745371287 | 110, Aquila, Bavdhan           |
| C105    | Sanjana | Mishra  |  8867523091 | 310, Serenity, Aundh           |
| C106    | Kajal   | Sharma  |  8966345271 | 501, Elite Galaxy, Baner       |
| C107    | Ravi    | Sethi   |  6952345617 | 402, Pebbles, Wakad            |
| C108    | Shyam   | Verma   |  9712345309 | 603, Nauka, Warje              |
| C109    | Siddh   | Shah    |  6667321287 | 204, Belle Vue, Swargate       |
| C110    | Aditya  | Singh   |  9943267451 | 504, Runwal Platinum, Bavdhan  |
| C111    | Vidit   | Shukla  |  6669981200 | 701, Tailwind, Bhusari Colony  |
+---------+---------+---------+-------------+--------------------------------+
```

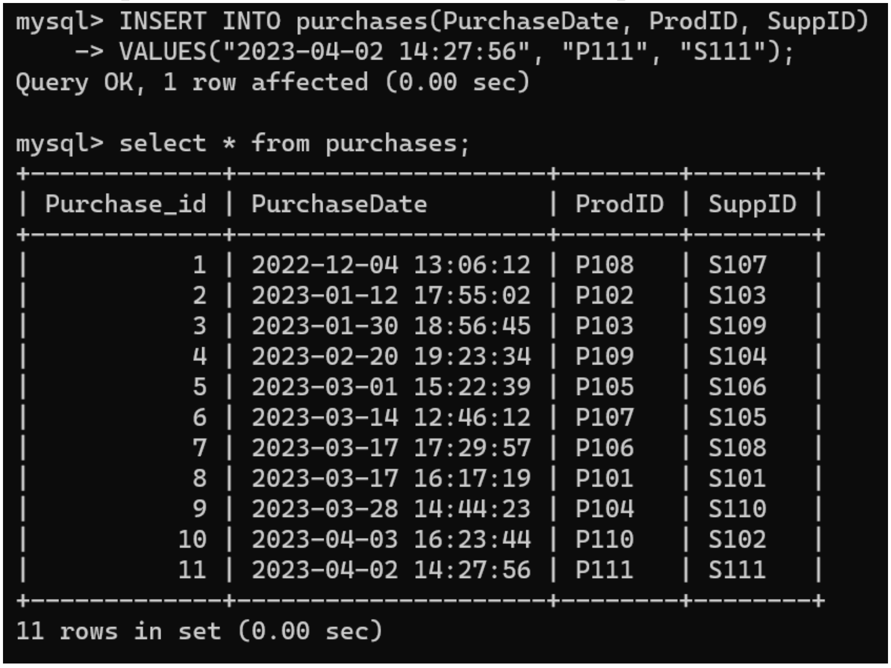
14. Insert a new employee into the employees table.



```
mysql> INSERT INTO employee(Emp_id, FName, LName, EmpContact, EmpAddress)
    -> VALUES("E111", "Sharad", "Shelke", 8811123345, "101, Sai Kamal, Wakad, Pune");
Query OK, 1 row affected (0.00 sec)

mysql> select * from employee;
+--------+----------+---------+-------------+------------+-------------------------------------+
| Emp_id | FName    | LName   | EmpPosition | EmpContact | EmpAddress                          |
+--------+----------+---------+-------------+------------+-------------------------------------+
| E101   | Kartik   | Agrawal | Staff       | 6666782319 | 16, Rupali, Prabhat Road, Pune      |
| E102   | Shreyab  | Saini   | Manager     | 6776782349 | 104, Karishma Society, Vanaj, Pune  |
| E103   | Sanika   | Patil   | Staff       | 9996753422 | 201, Sahil Apartments, Warje, Pune  |
| E104   | Ketaki   | Patel   | Staff       | 7272654121 | 302, Fossil Ferns, Baner, Pune      |
| E105   | Dhruv    | Doshi   | Manager     | 9960777243 | 406, Animish, Wakad, Pune           |
| E106   | Sarthak  | Sharma  | Staff       | 8854323789 | 603, Aquila, Bavdhan, Pune          |
| E107   | Shantanu | Kakade  | Owner       | 7432234987 | 102, Runwal Platinum, Pashan, Pune  |
| E108   | Devyani  | Khare   | Staff       | 7498384472 | 503, Kamayani, Aundh, Pune          |
| E109   | Kanishq  | Pore    | Staff       | 8787234567 | 204, Pebbles, Aundh, Pune           |
| E110   | Advait   | Kelkar  | Staff       | 9879875643 | 402, Serenity, Bavdhan, Pune        |
| E111   | Sharad   | Shelke  | Staff       | 8811123345 | 101, Sai Kamal, Wakad, Pune         |
+--------+----------+---------+-------------+------------+-------------------------------------+
```

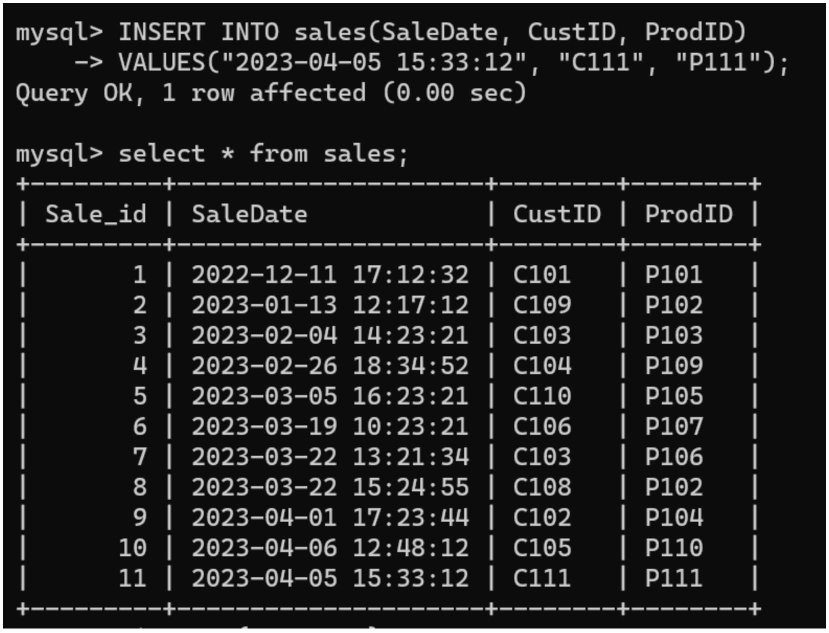
For the next 2 queries, I am considering the example that bottle is firstly purchased from the supplier by the supermarket and then sold to customer, so I am executing query 16 before 15.

16. Insert a new purchase into the purchases table, including the supplier ID, the product ID, and the date of the purchase:



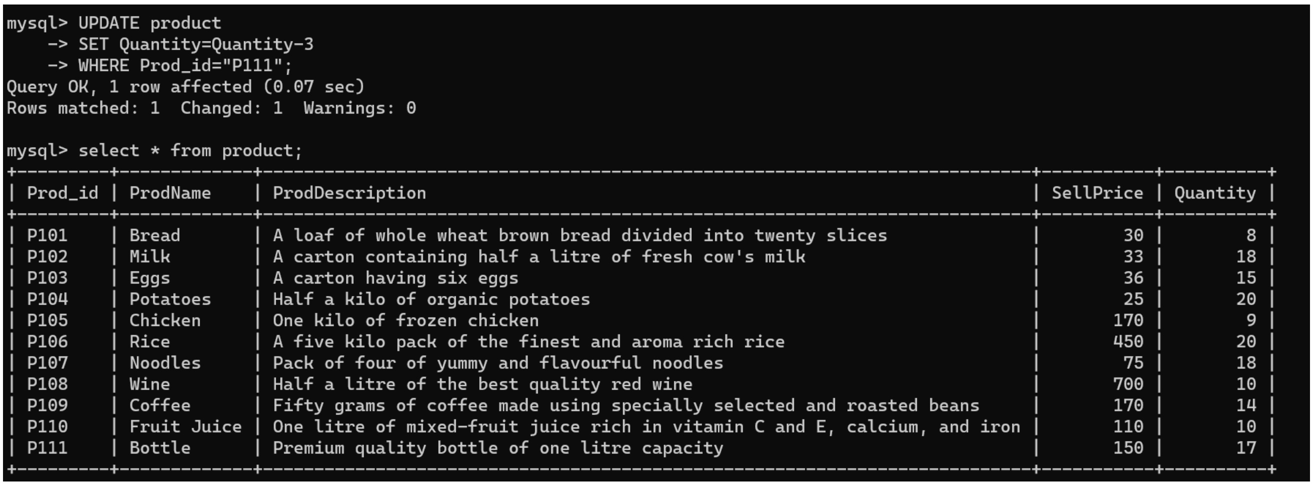
```
mysql> INSERT INTO purchases(PurchaseDate, ProdID, SuppID)
    -> VALUES("2023-04-02 14:27:56", "P111", "S111");
Query OK, 1 row affected (0.00 sec)

mysql> select * from purchases;
+-------------+---------------------+--------+--------+
| Purchase_id | PurchaseDate        | ProdID | SuppID |
+-------------+---------------------+--------+--------+
|           1 | 2022-12-04 13:06:12 | P108   | S107   |
|           2 | 2023-01-12 17:55:02 | P102   | S103   |
|           3 | 2023-01-30 18:56:45 | P103   | S109   |
|           4 | 2023-02-20 19:23:34 | P109   | S104   |
|           5 | 2023-03-01 15:22:39 | P105   | S106   |
|           6 | 2023-03-14 12:46:12 | P107   | S105   |
|           7 | 2023-03-17 17:29:57 | P106   | S108   |
|           8 | 2023-03-17 16:17:19 | P101   | S101   |
|           9 | 2023-03-28 14:44:23 | P104   | S110   |
|          10 | 2023-04-03 16:23:44 | P110   | S102   |
|          11 | 2023-04-02 14:27:56 | P111   | S111   |
+-------------+---------------------+--------+--------+
11 rows in set (0.00 sec)
```

15. Insert a new sale into the sales table, including the customer ID, the product ID, and the date of the sale:



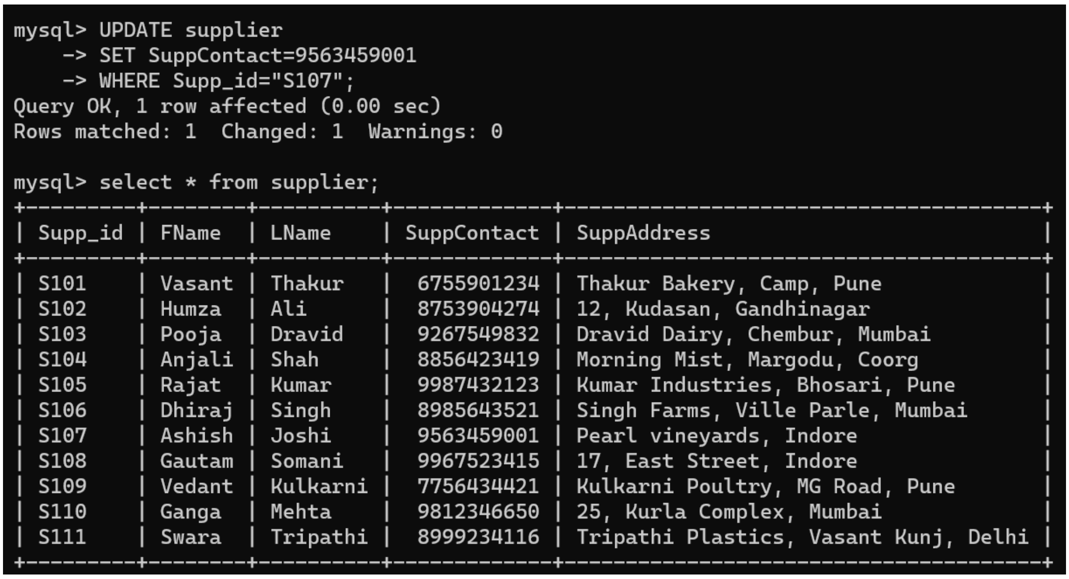
```
mysql> INSERT INTO sales(SaleDate, CustID, ProdID)
    -> VALUES("2023-04-05 15:33:12", "C111", "P111");
Query OK, 1 row affected (0.00 sec)

mysql> select * from sales;
+---------+---------------------+--------+--------+
| Sale_id | SaleDate            | CustID | ProdID |
+---------+---------------------+--------+--------+
|       1 | 2022-12-11 17:12:32 | C101   | P101   |
|       2 | 2023-01-13 12:17:12 | C109   | P102   |
|       3 | 2023-02-04 14:23:21 | C103   | P103   |
|       4 | 2023-02-26 18:34:52 | C104   | P109   |
|       5 | 2023-03-05 16:23:21 | C110   | P105   |
|       6 | 2023-03-19 10:23:21 | C106   | P107   |
|       7 | 2023-03-22 13:21:34 | C103   | P106   |
|       8 | 2023-03-22 15:24:55 | C108   | P102   |
|       9 | 2023-04-01 17:23:44 | C102   | P104   |
|      10 | 2023-04-06 12:48:12 | C105   | P110   |
|      11 | 2023-04-05 15:33:12 | C111   | P111   |
+---------+---------------------+--------+--------+
```

17. Update the quantity in stock for a specific product in the products table



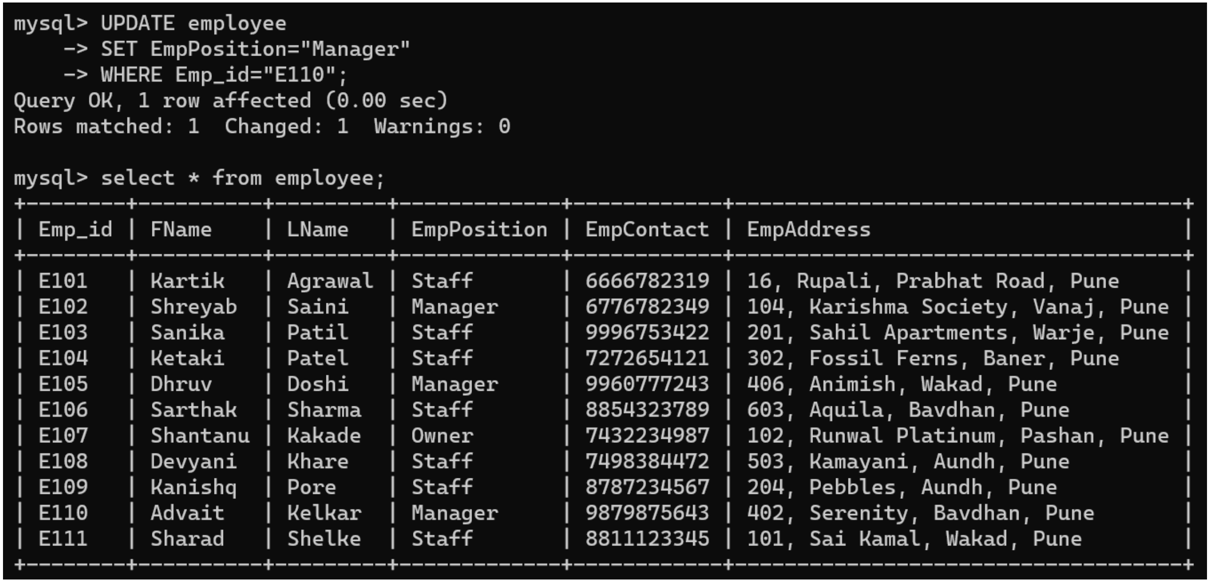
```
mysql> UPDATE product
    -> SET Quantity=Quantity-3
    -> WHERE Prod_id="P111";
Query OK, 1 row affected (0.07 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from product;
+---------+-------------+---------------------------------------------------------------------------+-----------+----------+
| Prod_id | ProdName    | ProdDescription                                                           | SellPrice | Quantity |
+---------+-------------+---------------------------------------------------------------------------+-----------+----------+
| P101    | Bread       | A loaf of whole wheat brown bread divided into twenty slices              |        30 |        8 |
| P102    | Milk        | A carton containing half a litre of fresh cow's milk                      |        33 |       18 |
| P103    | Eggs        | A carton having six eggs                                                  |        36 |       15 |
| P104    | Potatoes    | Half a kilo of organic potatoes                                           |        25 |       20 |
| P105    | Chicken     | One kilo of frozen chicken                                                |       170 |        9 |
| P106    | Rice        | A five kilo pack of the finest and aroma rich rice                        |       450 |       20 |
| P107    | Noodles     | Pack of four of yummy and flavourful noodles                              |        75 |       18 |
| P108    | Wine        | Half a litre of the best quality red wine                                 |       700 |       10 |
| P109    | Coffee      | Fifty grams of coffee made using specially selected and roasted beans     |       170 |       14 |
| P110    | Fruit Juice | One litre of mixed-fruit juice rich in vitamin C and E, calcium, and iron |       110 |       10 |
| P111    | Bottle      | Premium quality bottle of one litre capacity                              |       150 |       17 |
+---------+-------------+---------------------------------------------------------------------------+-----------+----------+
```

18. Update the contact information for a specific supplier in the suppliers table:



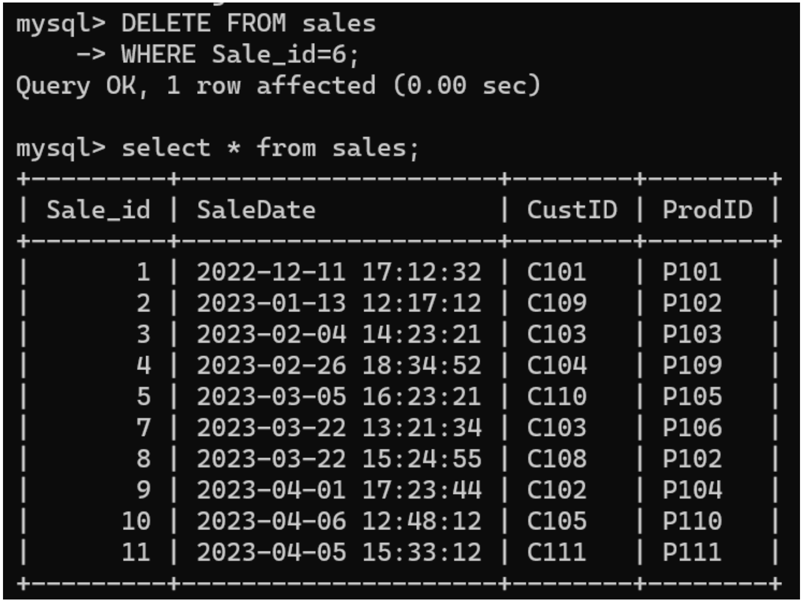
```
mysql> UPDATE supplier
    -> SET SuppContact=9563459001
    -> WHERE Supp_id="S107";
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from supplier;
+---------+--------+----------+-------------+---------------------------------------+
| Supp_id | FName  | LName    | SuppContact | SuppAddress                           |
+---------+--------+----------+-------------+---------------------------------------+
| S101    | Vasant | Thakur   |  6755901234 | Thakur Bakery, Camp, Pune             |
| S102    | Humza  | Ali      |  8753904274 | 12, Kudasan, Gandhinagar              |
| S103    | Pooja  | Dravid   |  9267549832 | Dravid Dairy, Chembur, Mumbai         |
| S104    | Anjali | Shah     |  8856423419 | Morning Mist, Margodu, Coorg          |
| S105    | Rajat  | Kumar    |  9987432123 | Kumar Industries, Bhosari, Pune       |
| S106    | Dhiraj | Singh    |  8985643521 | Singh Farms, Ville Parle, Mumbai      |
| S107    | Ashish | Joshi    |  9563459001 | Pearl vineyards, Indore               |
| S108    | Gautam | Somani   |  9967523415 | 17, East Street, Indore               |
| S109    | Vedant | Kulkarni |  7756434421 | Kulkarni Poultry, MG Road, Pune       |
| S110    | Ganga  | Mehta    |  9812346650 | 25, Kurla Complex, Mumbai             |
| S111    | Swara  | Tripathi |  8999234116 | Tripathi Plastics, Vasant Kunj, Delhi |
+---------+--------+----------+-------------+---------------------------------------+
```

19. Update the job position for a specific employee in the employees table:



```
mysql> UPDATE employee
    -> SET EmpPosition="Manager"
    -> WHERE Emp_id="E110";
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from employee;
+--------+----------+---------+-------------+------------+------------------------------------+
| Emp_id | FName    | LName   | EmpPosition | EmpContact | EmpAddress                         |
+--------+----------+---------+-------------+------------+------------------------------------+
| E101   | Kartik   | Agrawal | Staff       | 6666782319 | 16, Rupali, Prabhat Road, Pune     |
| E102   | Shreyab  | Saini   | Manager     | 6776782349 | 104, Karishma Society, Vanaj, Pune |
| E103   | Sanika   | Patil   | Staff       | 9996753422 | 201, Sahil Apartments, Warje, Pune |
| E104   | Ketaki   | Patel   | Staff       | 7272654121 | 302, Fossil Ferns, Baner, Pune     |
| E105   | Dhruv    | Doshi   | Manager     | 9960777243 | 406, Animish, Wakad, Pune          |
| E106   | Sarthak  | Sharma  | Staff       | 8854323789 | 603, Aquila, Bavdhan, Pune         |
| E107   | Shantanu | Kakade  | Owner       | 7432234987 | 102, Runwal Platinum, Pashan, Pune |
| E108   | Devyani  | Khare   | Staff       | 7498384472 | 503, Kamayani, Aundh, Pune         |
| E109   | Kanishq  | Pore    | Staff       | 8787234567 | 204, Pebbles, Aundh, Pune          |
| E110   | Advait   | Kelkar  | Manager     | 9879875643 | 402, Serenity, Bavdhan, Pune       |
| E111   | Sharad   | Shelke  | Staff       | 8811123345 | 101, Sai Kamal, Wakad, Pune        |
+--------+----------+---------+-------------+------------+------------------------------------+
```

20. Delete a specific sale from the sales table based on the sale ID:



```
mysql> DELETE FROM sales
    -> WHERE Sale_id=6;
Query OK, 1 row affected (0.00 sec)

mysql> select * from sales;
+---------+---------------------+--------+--------+
| Sale_id | SaleDate            | CustID | ProdID |
+---------+---------------------+--------+--------+
|       1 | 2022-12-11 17:12:32 | C101   | P101   |
|       2 | 2023-01-13 12:17:12 | C109   | P102   |
|       3 | 2023-02-04 14:23:21 | C103   | P103   |
|       4 | 2023-02-26 18:34:52 | C104   | P109   |
|       5 | 2023-03-05 16:23:21 | C110   | P105   |
|       7 | 2023-03-22 13:21:34 | C103   | P106   |
|       8 | 2023-03-22 15:24:55 | C108   | P102   |
|       9 | 2023-04-01 17:23:44 | C102   | P104   |
|      10 | 2023-04-06 12:48:12 | C105   | P110   |
|      11 | 2023-04-05 15:33:12 | C111   | P111   |
+---------+---------------------+--------+--------+
```

The SM_operations_report.sql file contains thirteen SQL queries that I have come up with as a report of the operations of the supermarket.