ASSIGNMENT – 4

CSA0593 – DBMS

NAME – S.VARSHA

REG NO – 192311425

DATE – 16:11:2024

1. Develop a database to manage products, suppliers, customers, and orders.

- Model tables for products, suppliers, customers, and orders.
- Write stored procedures for placing orders, updating product quantities, and recording new suppliers.
- Implement triggers to update product availability based on stock levels and to notify when items are low in stock.
- Write SQL queries to analyze top-selling products, supplier reliability, and customer purchasing trends.

ANSWER:

CONCEPTUAL ER DIAGRAM:

1. **Entities**: Products, Suppliers, Customers, Orders.
2. **Relationships**: Suppliers supply Products, Customers place Orders, Orders include multiple Products.
3. **Constraints**: Foreign keys link Products to Suppliers, Orders to Customers, and Orders to Products via a junction table.

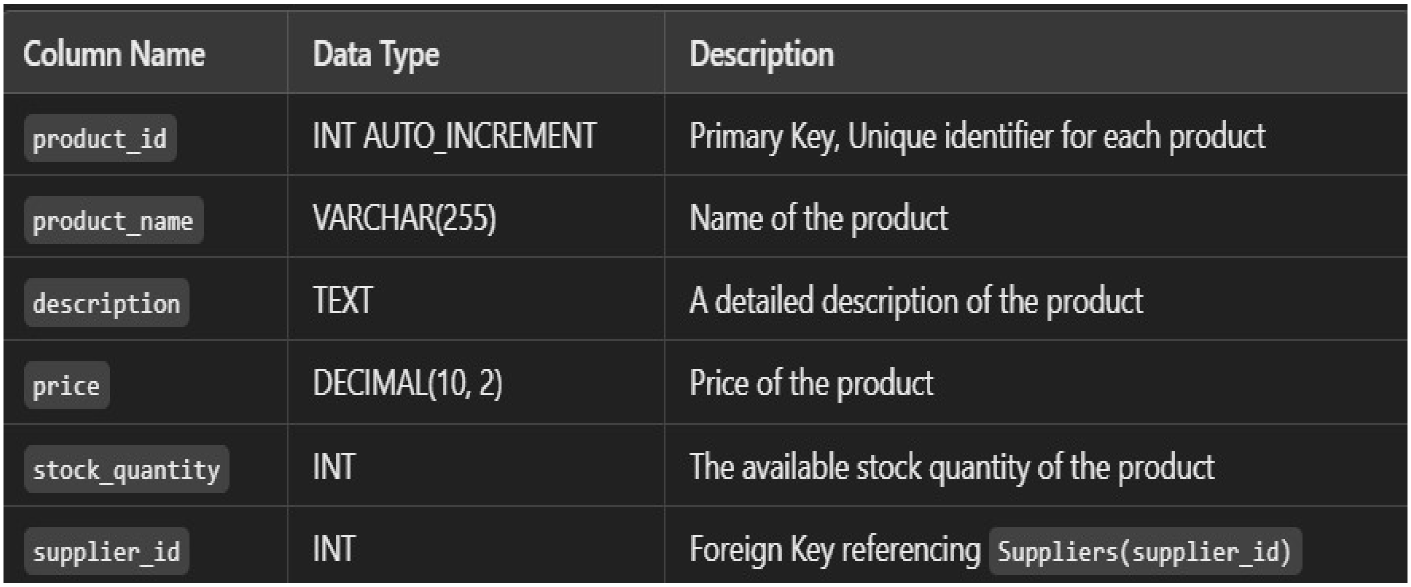
CODE:

```
Create Suppliers Table
CREATE TABLE Suppliers (
    supplier_id INT AUTO_INCREMENT PRIMARY KEY,
    supplier_name VARCHAR(255) NOT NULL,
    contact_name VARCHAR(100),
    phone_number VARCHAR(20),
    email VARCHAR(100),
    address VARCHAR(255)
);

-- Create Products Table
CREATE TABLE Products (
    product_id INT AUTO_INCREMENT PRIMARY KEY,
    product_name VARCHAR(255) NOT NULL,
    description TEXT,
    price DECIMAL(10, 2) NOT NULL,
    stock_quantity INT NOT NULL,
    supplier_id INT,
    FOREIGN KEY (supplier_id) REFERENCES Suppliers(supplier_id)
```

```
);

-- Create Customers Table
CREATE TABLE Customers (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(100) NOT NULL,
    last_name VARCHAR(100) NOT NULL,
    email VARCHAR(100),
    phone_number VARCHAR(20),
    address VARCHAR(255)
);

-- Create Orders Table
CREATE TABLE Orders (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    order_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    status ENUM('Pending', 'Shipped', 'Delivered', 'Canceled') DEFAULT 'Pending',
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
```

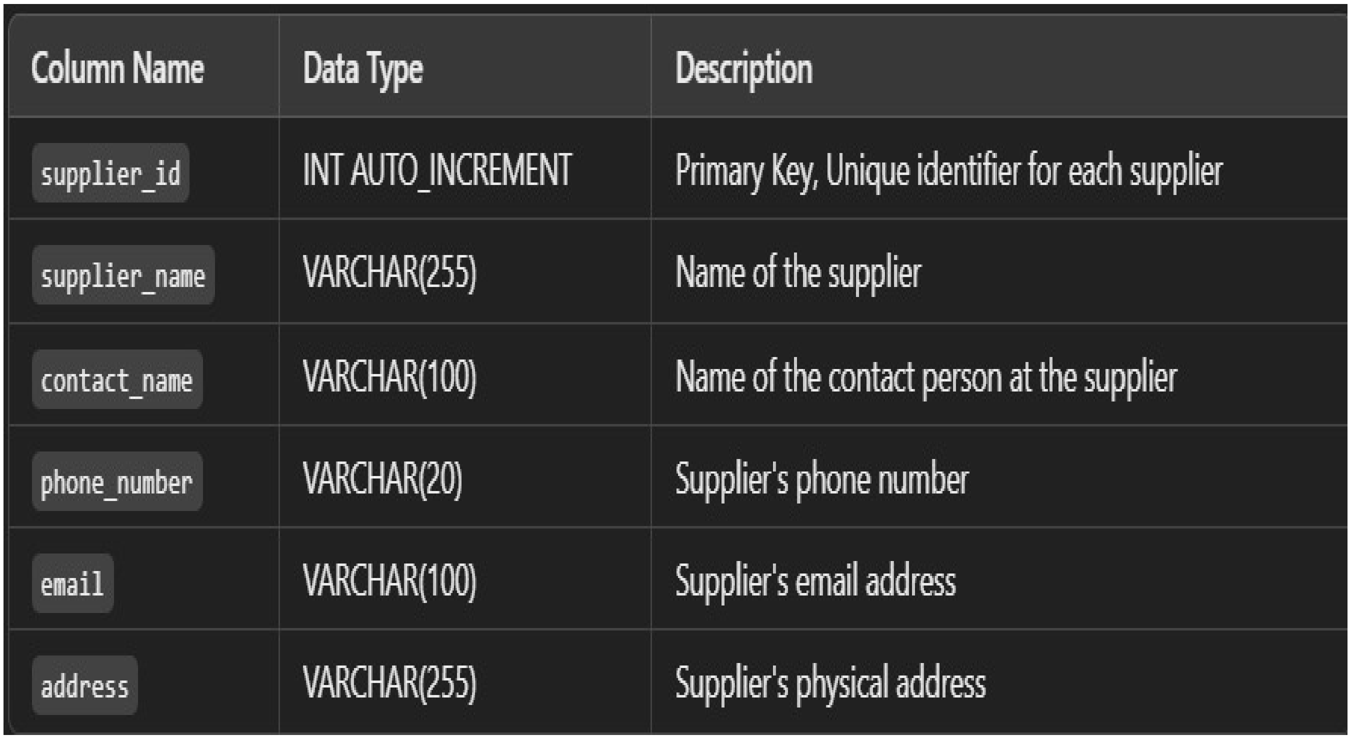
```
);

-- Create OrderDetails Table
CREATE TABLE OrderDetails (
    order_detail_id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT,
    product_id INT,
    quantity INT NOT NULL,
    unit_price DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (order_id) REFERENCES Orders(order_id),
    FOREIGN KEY (product_id) REFERENCES
Products(product_id)
);
```

PRODUCT TABLE



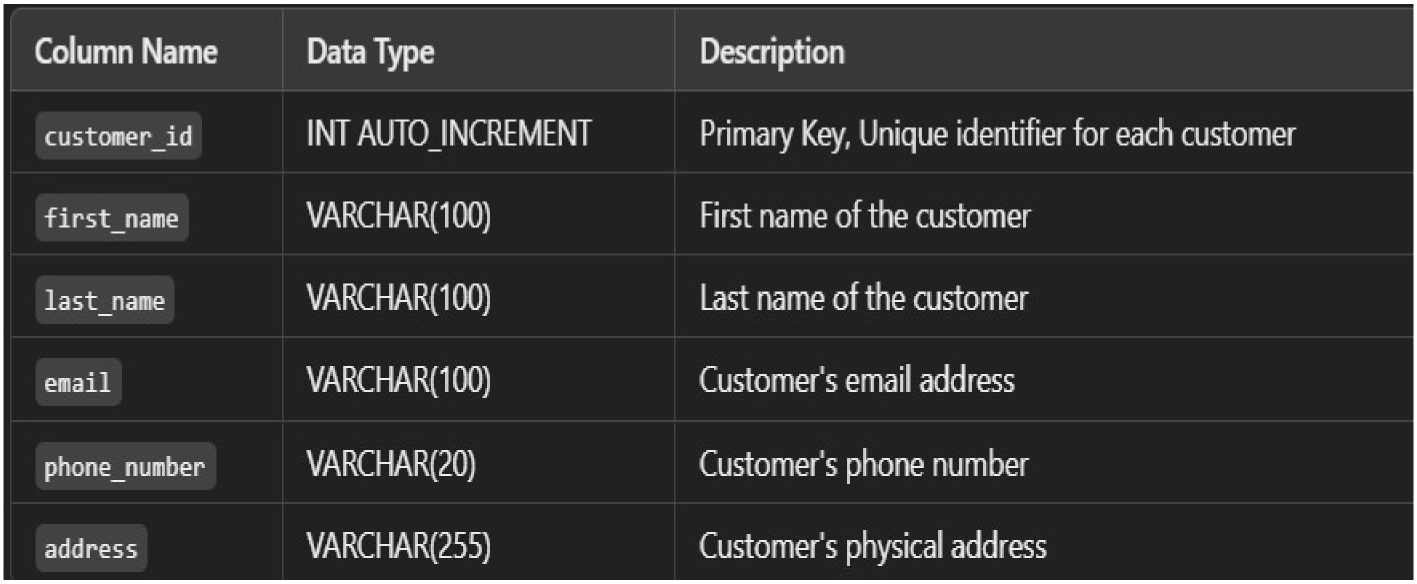
Column Name	Data Type	Description
`product_id`	INT AUTO_INCREMENT	Primary Key, Unique identifier for each product
`product_name`	VARCHAR(255)	Name of the product
`description`	TEXT	A detailed description of the product
`price`	DECIMAL(10, 2)	Price of the product
`stock_quantity`	INT	The available stock quantity of the product
`supplier_id`	INT	Foreign Key referencing `Suppliers(supplier_id)`

SUPPILER TABLE



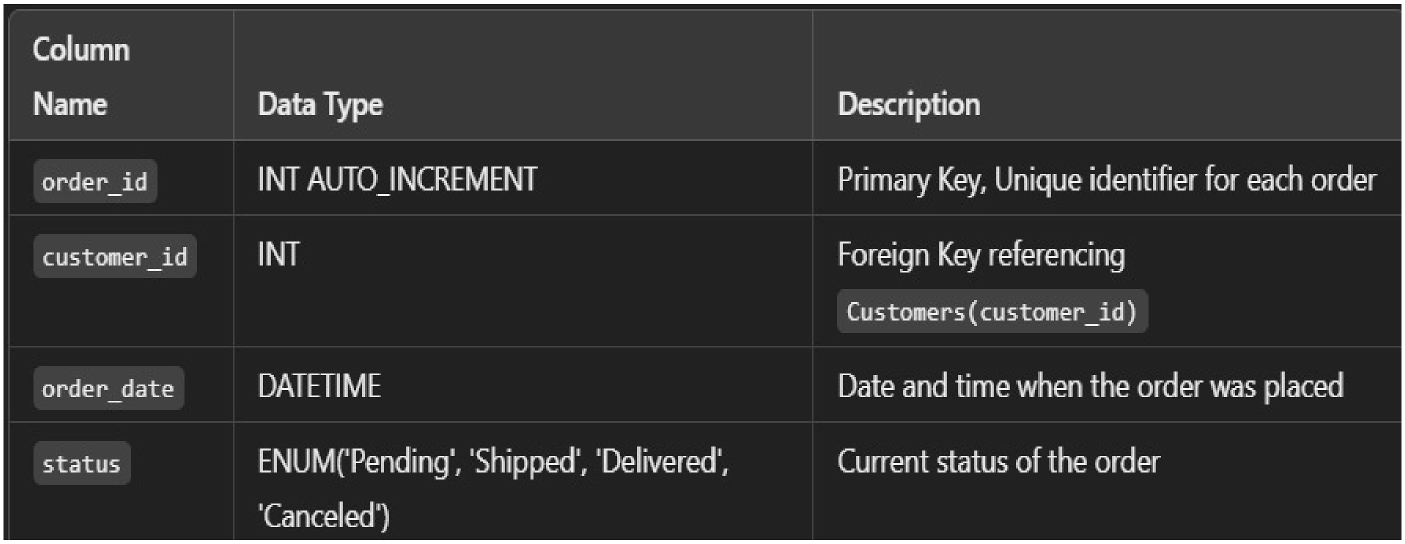
Column Name	Data Type	Description
supplier_id	INT AUTO_INCREMENT	Primary Key, Unique identifier for each supplier
supplier_name	VARCHAR(255)	Name of the supplier
contact_name	VARCHAR(100)	Name of the contact person at the supplier
phone_number	VARCHAR(20)	Supplier's phone number
email	VARCHAR(100)	Supplier's email address
address	VARCHAR(255)	Supplier's physical address

CUSTOMER TABLE



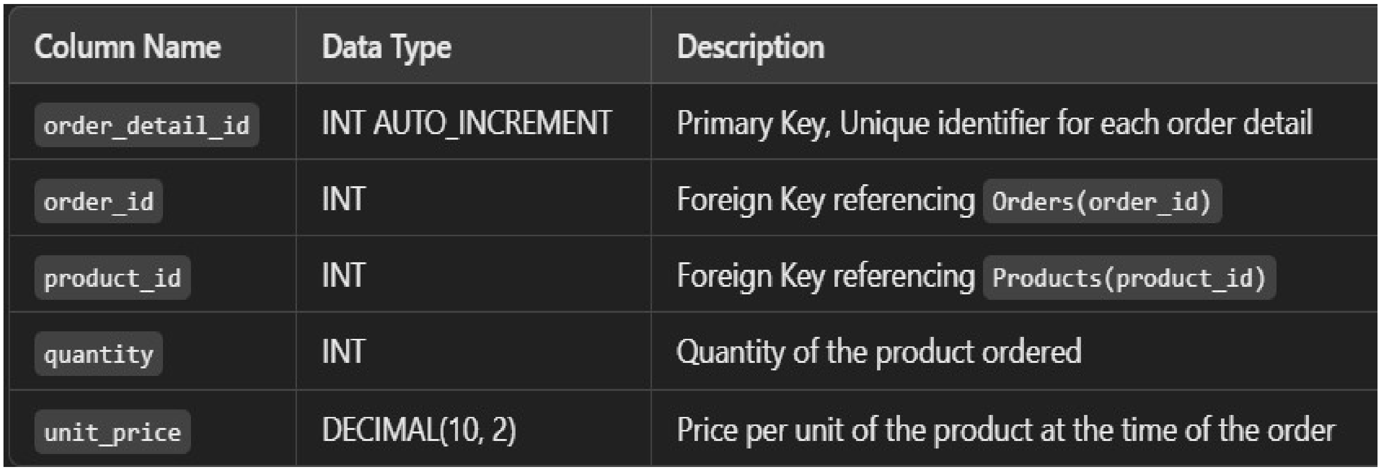
Column Name	Data Type	Description
customer_id	INT AUTO_INCREMENT	Primary Key, Unique identifier for each customer
first_name	VARCHAR(100)	First name of the customer
last_name	VARCHAR(100)	Last name of the customer
email	VARCHAR(100)	Customer's email address
phone_number	VARCHAR(20)	Customer's phone number
address	VARCHAR(255)	Customer's physical address

ORDER TABLE



Column Name	Data Type	Description
`order_id`	INT AUTO_INCREMENT	Primary Key, Unique identifier for each order
`customer_id`	INT	Foreign Key referencing `Customers(customer_id)`
`order_date`	DATETIME	Date and time when the order was placed
`status`	ENUM('Pending', 'Shipped', 'Delivered', 'Canceled')	Current status of the order

ORDER DETAIL TABLE:



Column Name	Data Type	Description
`order_detail_id`	INT AUTO_INCREMENT	Primary Key, Unique identifier for each order detail
`order_id`	INT	Foreign Key referencing `Orders(order_id)`
`product_id`	INT	Foreign Key referencing `Products(product_id)`
`quantity`	INT	Quantity of the product ordered
`unit_price`	DECIMAL(10, 2)	Price per unit of the product at the time of the order

CONCLUSION

The above tables model the relationships between products, suppliers, customers, and orders. The structure supports efficient querying and updates to track stock, manage customer orders, and store relevant supplier information.

Products are linked to Suppliers via a foreign key.

Customers place Orders, and OrderDetails associates Products with Orders.The database allows for easy tracking of stock levels, customer orders, and supplier information.