

SQL Code for Creating Database, Table Objects and An ER Diagram Illustrating How the Objects Are Connected

-- Creating database called sales_group

CREATE DATABASE IF NOT EXISTS sales_group;

USE sales_group;

-- The database has 9 table objects

-- Now creating all tables for the sales_group database

-- All foreign keys are set to on delete cascade

-- Creating the categories table, this table will details about the product categories

CREATE TABLE categories (

CategoryID **SMALLINT PRIMARY KEY NOT NULL**,

CategoryName **VARCHAR(20) NOT NULL**,

Description **VARCHAR(60) NOT NULL**

);

-- Creating the geo_location table, this table has details of the cities and countries involved in sales

CREATE TABLE geo_location (

City **VARCHAR(20) PRIMARY KEY NOT NULL**,

Country **VARCHAR(15)**

);

-- Creating shippers table, this table has details about the shippers who carry out the logistics of the products

CREATE TABLE shippers (

ShipperID **SMALLINT PRIMARY KEY NOT NULL**,

CompanyName **VARCHAR(20)**,

Phone **VARCHAR(20)**

);

-- Creating suppliers table, this table has details for the companies that supply the business with products that the business sales

```
CREATE TABLE suppliers (  
    SupplierID SMALLINT PRIMARY KEY,  
    CompanyName VARCHAR(40),  
    ContactName VARCHAR(30),  
    ContactTitle VARCHAR(30),  
    Address VARCHAR(50),  
    City VARCHAR(20),  
    Country VARCHAR(15),  
    Phone VARCHAR(20)  
);
```

-- Creating the product table, this table of the products sold

```
CREATE TABLE product (  
    ProductID SMALLINT PRIMARY KEY NOT NULL,  
    ProductName VARCHAR(50),  
    SupplierID SMALLINT,  
    CategoryID SMALLINT,  
    QuantityPerUnit VARCHAR(50),  
    UnitPrice DECIMAL(6 , 2 ),  
    UnitsInStock SMALLINT,  
    UnitsOnOrder SMALLINT,  
    ReorderLevel SMALLINT,  
    Discontinued VARCHAR(6),  
    FOREIGN KEY (SupplierID)  
        REFERENCES suppliers (SupplierID)  
        ON DELETE CASCADE,  
    FOREIGN KEY (CategoryID)  
        REFERENCES categories (CategoryID)  
        ON DELETE CASCADE  
);
```

-- Creating customer table, this table has details of the customer that purchase the products

```
CREATE TABLE customers (  
    CustomerID VARCHAR(5) PRIMARY KEY NOT NULL,  
    CompanyName VARCHAR(40) NOT NULL,  
    ContactName VARCHAR(30) NOT NULL,  
    ContactTitle VARCHAR(35) NOT NULL,  
    Address VARCHAR(50) NOT NULL,  
    City VARCHAR(50),  
    Country VARCHAR(50) NOT NULL,  
    Phone VARCHAR(20) NULL,  
    FOREIGN KEY (City)  
        REFERENCES geo_location (City)  
        ON DELETE CASCADE  
);
```

-- Creating employees table, this table has details of the employees of the business

```
CREATE TABLE employees (  
    EmployeeID SMALLINT PRIMARY KEY NOT NULL,  
    LastName VARCHAR(10) NOT NULL,  
    FirstName VARCHAR(10) NOT NULL,  
    Title VARCHAR(30) NOT NULL,  
    TitleOfCourtesy VARCHAR(5) NOT NULL,  
    BirthDate DATE NOT NULL,  
    HireDate DATE NOT NULL,  
    Address VARCHAR(50) NOT NULL,  
    City VARCHAR(20) NOT NULL,  
    Country VARCHAR(15) NOT NULL,  
    HomePhone VARCHAR(20) NULL,  
    ReportsTo SMALLINT NULL,  
    FOREIGN KEY (City)  
        REFERENCES geo_location (City)  
        ON DELETE CASCADE  
);
```

-- Creating orders tables, this table has details about the sales done by the business

```
CREATE TABLE orders (  
  OrderID SMALLINT PRIMARY KEY NOT NULL,  
  CustomerID VARCHAR(20),  
  EmployeeID SMALLINT,  
  OrderDate DATE,  
  RequiredDate DATE,  
  ShippedDate DATE,  
  ShipVia SMALLINT,  
  Freight DECIMAL(5 , 2 ),  
  ShipName VARCHAR(50),  
  ShipAddress VARCHAR(50),  
  ShipCity VARCHAR(50),  
  ShipCountry,  
  FOREIGN KEY (CustomerID)  
    REFERENCES customers (CustomerID)  
    ON DELETE CASCADE,  
  FOREIGN KEY (EmployeeID)  
    REFERENCES employees (EmployeeID)  
    ON DELETE CASCADE,  
  FOREIGN KEY (ShipVia)  
    REFERENCES shippers (ShipperID)  
    ON DELETE CASCADE,  
  FOREIGN KEY (ShipCountry)  
    REFERENCES geo_location (City)  
    ON DELETE CASCADE  
);
```

-- Creating order_details, this table has finer details on the orders made by customers

CREATE TABLE order_details (

OrderID **SMALLINT**,

ProductID **SMALLINT**,

UnitPrice **DECIMAL**(6 , 2),

Quantity **SMALLINT**,

Discount **DECIMAL**(3 , 2),

Discount_Amount **DECIMAL**(6 , 2),

Total_Sales **DECIMAL**(10 , 2),

FOREIGN KEY (OrderID)

REFERENCES orders (OrderID)

ON DELETE CASCADE,

FOREIGN KEY (ProductID)

REFERENCES product (ProductID)

ON DELETE CASCADE

);

-- The ER Diagram

