**Q1**. Create a Data Dictionary for the TPAG scenario.

**Ans:**

| Data Dictionary | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Table Name** | **Attribute Name** | **Contents** | **Type** | **Format** | **Range** | **Required** | **PK or FK** | **FK Referenced Table** |
| **Customer** | CustomerID | Customer Identification Number and account number | TEXT | XXX00000 | XXX0000000-XXX9999999 | Y | PK | |
| | Cus_FName | First Name | VARCHAR(20) | Xxxxxxxx | Variable length with maximum 20 characters | Y | | |
| | Cus_MName | Middle Name | VARCHAR(15) | Xxxxxxxx | | | | |
| | Cus_LName | Last Name | VARCHAR(15) | Xxxxxxxx | | Y | | |
| | Billing_Street | Billing Address-Street Number and name | VARCHAR(100) | 1807 Harlequin Pl | | Y | | |
| | Billing_Apt | Billing Address-Apartment or unit number | VARCHAR(20) | Villa 30 | | Y | | |
| | Billing_City | Billing Address- City name | VARCHAR(50) | Allen | | Y | | |
| | Billing_State | Billing Address- State (as abbreviations) | VARCHAR(5) | TX | | Y | | |
| | Billing_Zip | Billing Address-Zip Code | VARCHAR(10) | 75003 | | Y | | |
| | Cus_Phone | Customer phone/contact number | VARCHAR(15) | +1(469)3018444 | | | | |
| | Cus_Email | Customer email address | VARCHAR(50) | Xxx@gmail.com | | Y | | |
| **Orders** | OrderID | Unique number for each generated order | INT | 45678 | 00000-99999 | Y | PK | |
| | CustomerID | Customer Identification Number and account number | TEXT | XXX00000 | XXX0000000-XXX9999999 | Y | FK | Customer |
| | SalesRepID | ID for a sales representative | INT | 1002111 | 1000000-1009999 | Y | FK | SalesRep |
| | ShipmentID | Identification number for a shipment | INT | 7777777 | 0000000-9999999 | Y | FK | Shipment |
| | Order_Date | The date on which an order is placed | TEXT | YYYY-MM-DD | | Y | | |
| | Ord_Amount | Total amount of the order placed | DECIMAL(10,2) | 100.91 | The maximum value that can be stored is 9999999.99 | Y | | |
| **ArtWorks** | ArtID | Each type of Artwork is identified by an Item Code | VARCHAR(20) | XXX012345 | XXX000000-XXX999999 | Y | PK | |
| | Art_Des | Description of the Artwork- size, weight type etc. | VARCHAR(100) | | | Y | | |
| | Art_Price | Price of each Artwork | DECIMAL(10,2) | 100.91 | The maximum value that can be stored is 9999999.99 | Y | | |
| | Art_Qty | The quantity of each Artwork in stock | INT | 566 | | Y | | |
| **Vehicle** | VehicleID | Identity code/number for each vehicle assigned by TPAG | VARCHAR(15) | XXX-0101 | XXX-0000 to XXX-9999 | Y | PK | |
| | DriverID | Unique number to identify each diver. | INT | 1002133 | 1000000-1009999 | Y | FK | Driver |
| | VLicense_No | License plate number of each vehicle | VARCHAR(10) | XXX-1234 | | Y | | |
| | VLicense_Exp | Expiration date of license of a vehicle | TEXT | YYYY-MM-DD | | Y | | |
| | Inspection_Exp | Expiration date of last inspection of a vehicle | TEXT | YYYY-MM-DD | | Y | | |
| **Shipment** | ShipmentID | Identification number for a shipment | INT | 456789 | 000000-999999 | Y | PK | |
| | OrderID | Unique number for each generated order | INT | 45678 | 00000-99999 | Y | FK | Orders |
| | Del_Street | Delivery Address-Street Number and name | VARCHAR(100) | 1807 Harlequin Pl | | Y | | |
| | Del_Apt | Delivery Address-Apartment or unit number | VARCHAR(20) | Villa 30 | | Y | | |
| | Del_City | Delivery Address- City name | VARCHAR(50) | Allen | | Y | | |
| | Del_State | Delivery Address- State | VARCHAR(5) | TX | | Y | | |
| | Del_Zip | Delivery Address-Zip Code | VARCHAR(10) | 75003 | | Y | | |
| | Del_Date | Due date the shipment to be delivered | TEXT | YYYY-MM-DD | | Y | | |
| | VehicleID | Each vehicle has a number assigned by TPAG | VARCHAR(15) | XXX-0101 | | Y | FK | Vehicle |
| **Employee** | EmployeeID | Unique number to identify each employee | INT | 1002133 | 1000000-1009999 | Y | PK | |
| | SSN | Social Security Number | VARCHAR(11) | 123-11-1234 | | Y | | |
| | Emp_Fname | Employee first name | VARCHAR(20) | Xxxxxxxx | | Y | | |
| | Emp_Mname | Employee middle name | VARCHAR(15) | Xxxxxxxx | | | | |
| | Emp_Lname | Employee last name | VARCHAR(15) | Xxxxxxxx | | Y | | |
| | DoB | Employee date of birth | TEXT | YYYY-MM-DD | | Y | | |
| | Emp_Type | Employee type (driver or sales representative) | VARCHAR(25) | Driver/ Sales Representative | | Y | | |
| | Emp_Salary | Employee monthly salary | DECIMAL(10,2) | 1234.99 | The maximum value that can be stored is 9999999.99 | Y | | |
| | Emp_Street | Employee residence address | VARCHAR(100) | 1807 Harlequin Pl | | Y | | |
| | Emp_Apt | Employee residence address | VARCHAR(20) | Villa 30 | | Y | | |
| | Emp_City | Employee residence address -city | VARCHAR(50) | Allen | | Y | | |
| | Emp_State | Employee residence address-state | VARCHAR(5) | TX | | Y | | |
| | Emp_Zip | Employee residence address- zip code | VARCHAR(10) | 75003 | | Y | | |
| | Emp_Phone | Employee contact number | VARCHAR(15) | +1(469)3018444 | | Y | | |
| **SalesRep** | SalesRepID | Employee Id that representing a sales representative | INT | 1002111 | 1000000-1009999 | Y | PK and FK | Employee |
| | Commission | Commission for a sales representative on processing order | DECIMAL(10,2) | 234.32 | The maximum value that can be stored is 9999999.99 | | | |
| **Driver** | DriverID | Unique number to identify each diver. This DriverID is also the employee id for the driver as part of the Employee entity. | INT | 1002133 | 1000000-1009999 | Y | PK and FK | Employee |
| | DL_No | The valid driving license number | VARCHAR(20) | 12345678 | | Y | | |
| | DL_Exp_Date | The expiration date of the driving license | TEXT | YYYY-MM-DD | | Y | | |
| | VehicleID | The vehicle number assigned to the driver | VARCHAR(15) | XXX-0101 | XXX-0000 to XXX-9999 | Y | FK | Vehicle |
| **Assign** | OrderID | Unique number for each generated order | INT | 45678 | 00000-99999 | Y | PK and FK | Orders |
| | ArtID | Each type of Artwork is identified by an Item Code | VARCHAR(20) | XXX012345 | XXX000000-XXX999999 | Y | PK and FK | ArtWorks |
| | Qty_Ordered | Quantity of the items ordered associated with each order | INT | 32 | | Y | | |

| | |
|---|---|
| **PK** | Primary Key |
| **FK** | Foreign Key |
| **TEXT** | Represents variable-length character strings |
| **VARCHAR** | Variable character length data (1-2000 characters) |
| **INTEGER** | Used for whole numbers, both positive and negative. |
| **DECIMAL** | Stores numbers with a maximum of 10 digits in total, including 2 decimal places to the right of the decimal point. |
| **TEXT** | Used to define dates |

**Q2**. Create Tables in SQLite Database

**Ans:**

**Creating Database Assignment_4.db and Tables:**

```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\samih>cd Downloads

C:\Users\samih\Downloads>cd Dbms

C:\Users\samih\Downloads\Dbms>sqlite3
SQLite version 3.42.0 2023-05-16 12:36:15
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .quit

C:\Users\samih\Downloads\Dbms>sqlite3 Assignment_4.db
SQLite version 3.42.0 2023-05-16 12:36:15
Enter ".help" for usage hints.
sqlite> CREATE TABLE Customer (
(x1...> CustomerID TEXT PRIMARY KEY NOT NULL,
(x1...> Cus_FName VARCHAR(20) NOT NULL,
(x1...> Cus_MName VARCHAR(15),
(x1...> Cus_LName VARCHAR(15) NOT NULL,
(x1...> Billing_Street VARCHAR(100) NOT NULL,
(x1...> Billing_Apt VARCHAR(50) NOT NULL,
(x1...> Billing_City VARCHAR(50) NOT NULL,
(x1...> Billing_State VARCHAR(5) NOT NULL,
(x1...> Billing_Zip VARCHAR(10) NOT NULL,
(x1...> Cus_Phone VARCHAR(15),
(x1...> Cus_Email VARCHAR(50) NOT NULL
```

```
sqlite> CREATE TABLE Employee (
(x1...> EmployeeID INT PRIMARY KEY NOT NULL,
(x1...> SSN VARCHAR(11) NOT NULL,
(x1...> Emp_FName VARCHAR(20) NOT NULL,
(x1...> Emp_MName VARCHAR(15),
(x1...> Emp_LName VARCHAR(15) NOT NULL,
(x1...> DoB TEXT NOT NULL,
(x1...> Emp_Type VARCHAR(25) NOT NULL,
(x1...> Emp_Salary DECIMAL(10,2) NOT NULL,
(x1...> Emp_Street VARCHAR(100) NOT NULL,
(x1...> Emp_Apt VARCHAR(20) NOT NULL,
(x1...> Emp_City VARCHAR(50) NOT NULL,
(x1...> Emp_State VARCHAR(5) NOT NULL,
(x1...> Emp_Zip VARCHAR(10) NOT NULL,
(x1...> Emp_Phone VARCHAR(15) NOT NULL
```

```
sqlite> CREATE TABLE ArtWorks (
(x1...> ArtID VARCHAR(20) PRIMARY KEY NOT NULL,
(x1...> Art_Des VARCHAR(100) NOT NULL,
(x1...> Art_Price DECIMAL(10,2) NOT NULL,
(x1...> Art_Qty INTEGER NOT NULL
```

```
sqlite> CREATE TABLE Orders (
(x1...> OrderID INTEGER PRIMARY KEY NOT NULL,
(x1...> CustomerID TEXT NOT NULL,
(x1...> SalesRepID INTEGER NOT NULL,
(x1...> ShipmentID INTEGER NOT NULL,
(x1...> Order_Date TEXT NOT NULL,
(x1...> Order_Amount DECIMAL(10,2),
(x1...> FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID) ON DELETE CASCADE ON UPDATE CASCADE,
(x1...> FOREIGN KEY (SalesRepID) REFERENCES SalesRep (SalesRepID) ON DELETE CASCADE ON UPDATE CASCADE,
(x1...> FOREIGN KEY (ShipmentID) REFERENCES Shipment (ShipmentID) ON DELETE CASCADE ON UPDATE CASCADE
```

```
sqlite> CREATE TABLE Vehicle (
(x1...> VehicleID VARCHAR(15) PRIMARY KEY NOT NULL,
(x1...> DriverID INTEGER NOT NULL,
(x1...> VLicense_No VARCHAR(10) NOT NULL,
(x1...> VLicense_Exp TEXT NOT NULL,
(x1...> Inspection_Exp TEXT NOT NULL,
(x1...> FOREIGN KEY (DriverID) REFERENCES Driver (DriverID) ON DELETE CASCADE ON UPDATE CASCADE
```

```
sqlite> CREATE TABLE Shipment (
(x1...> ShipmentID INTEGER PRIMARY KEY NOT NULL,
(x1...> OrderID INTEGER NOT NULL,
(x1...> VehicleID VARCHAR(15) NOT NULL,
(x1...> Del_Street VARCHAR(100) NOT NULL,
(x1...> Del_Apt VARCHAR(20) NOT NULL,
(x1...> Del_City VARCHAR(50) NOT NULL,
(x1...> Del_State VARCHAR(5) NOT NULL,
(x1...> Del_Zip VARCHAR(10) NOT NULL,
(x1...> Del_Date TEXT NOT NULL,
(x1...> FOREIGN KEY (OrderID) REFERENCES Orders (OrderID) ON DELETE CASCADE ON UPDATE CASCADE,
(x1...> FOREIGN KEY (VehicleID) REFERENCES Vehicle (VehicleID) ON DELETE CASCADE ON UPDATE CASCADE
(x1...> );
```

```
sqlite> CREATE TABLE SalesRep (
(x1...> SalesRepID INT PRIMARY KEY NOT NULL,
(x1...> Commission DECIMAL(10,2) NOT NULL,
(x1...> FOREIGN KEY (SalesRepID) REFERENCES Employee (SalesRepID) ON DELETE CASCADE ON UPDATE CASCADE
(x1...> );
```

```
sqlite> CREATE TABLE Driver (
(x1...> DriverID INT PRIMARY KEY NOT NULL,
(x1...> VehicleID VARCHAR(15) NOT NULL,
(x1...> DL_No VARCHAR(20) NOT NULL,
(x1...> DL_Exp_Date TEXT NOT NULL,
(x1...> FOREIGN KEY (VehicleID) REFERENCES Vehicle (VehicleID) ON DELETE CASCADE ON UPDATE CASCADE,
(x1...> FOREIGN KEY (DriverID) REFERENCES Employee (DriverID) ON DELETE CASCADE ON UPDATE CASCADE
```

```
sqlite> CREATE TABLE Assign (
(x1...> OrderID INTEGER NOT NULL,
(x1...> ArtID VARCHAR(20) NOT NULL,
(x1...> Qty_Ordered INTEGER NOT NULL,
(x1...> PRIMARY KEY (OrderID, ArtID),
(x1...> FOREIGN KEY (ArtID) REFERENCES ArtWorks (ArtID) ON DELETE CASCADE ON UPDATE CASCADE,
(x1...> FOREIGN KEY (OrderID) REFERENCES Orders (OrderID) ON DELETE CASCADE ON UPDATE CASCADE
(x1...> );
```

**SELECT Statement to Verify Creating Tables:**

```
sqlite> .header on
sqlite> .mode column
sqlite> .table
ArtWorks  Customer  Employee  SalesRep  Vehicle
Assign    Driver    Orders    Shipment
sqlite> SELECT * FROM artworks;
sqlite> SELECT * FROM customer;
sqlite> SELECT * FROM employee;
sqlite> SELECT * FROM salesrep;
sqlite> SELECT * FROM vehicle;
sqlite> SELECT * FROM assign;
sqlite> SELECT * FROM driver;
sqlite> SELECT * FROM orders;
sqlite> SELECT * FROM shipment;
```

**PRAGMA table_info(<Table_Name>) to Show Table Schema:**

```
sqlite> PRAGMA table_info(artworks);
cid  name       type           notnull  dflt_value  pk
---  ---------  -------------  -------  ----------  --
0    ArtID      VARCHAR(20)    1                    1
1    Art_Des    VARCHAR(100)   1                    0
2    Art_Price  DECIMAL(10,2)  1                    0
3    Art_Qty    INTEGER        1                    0
sqlite>
```

```
sqlite> PRAGMA table_info(customer);
cid  name           type           notnull  dflt_value  pk
---  -------------  -------------  -------  ----------  --
0    CustomerID     TEXT           1                    1
1    Cus_FName      VARCHAR(20)    1                    0
2    Cus_MName      VARCHAR(15)    0                    0
3    Cus_LName      VARCHAR(15)    1                    0
4    Billing_Street VARCHAR(100)   1                    0
5    Billing_Apt    VARCHAR(50)    1                    0
6    Billing_City   VARCHAR(50)    1                    0
7    Billing_State  VARCHAR(5)     1                    0
8    Billing_Zip    VARCHAR(10)    1                    0
9    Cus_Phone      VARCHAR(15)    0                    0
10   Cus_Email      VARCHAR(50)    1                    0
sqlite>
```

```
sqlite> PRAGMA table_info(employee);
cid  name         type           notnull  dflt_value  pk
---  -----------  -------------  -------  ----------  --
0    EmployeeID   INT            1                    1
1    SSN          VARCHAR(11)    1                    0
2    Emp_FName    VARCHAR(20)    1                    0
3    Emp_MName    VARCHAR(15)    0                    0
4    Emp_LName    VARCHAR(15)    1                    0
5    DoB          TEXT           1                    0
6    Emp_Type     VARCHAR(25)    1                    0
7    Emp_Salary   DECIMAL(10,2)  1                    0
8    Emp_Street   VARCHAR(100)   1                    0
9    Emp_Apt      VARCHAR(20)    1                    0
10   Emp_City     VARCHAR(50)    1                    0
11   Emp_State    VARCHAR(5)     1                    0
12   Emp_Zip      VARCHAR(10)    1                    0
13   Emp_Phone    VARCHAR(15)    1                    0
sqlite>
```

```
sqlite> PRAGMA table_info(salesrep);
cid  name         type           notnull  dflt_value  pk
---  -----------  -------------  -------  ----------  --
0    SalesRepID   INT            1                    1
1    Commission   DECIMAL(10,2)  1                    0
sqlite>
```

```
sqlite> PRAGMA table_info(vehicle);
cid  name           type          notnull  dflt_value  pk
---  -------------  ------------  -------  ----------  --
0    VehicleID      VARCHAR(15)   1                    1
1    DriverID       INTEGER       1                    0
2    VLicense_No    VARCHAR(10)   1                    0
3    VLicense_Exp   TEXT          1                    0
4    Inspection_Exp TEXT          1                    0
sqlite>
```

```
sqlite> PRAGMA table_info(assign);
cid  name         type          notnull  dflt_value  pk
---  -----------  ------------  -------  ----------  --
0    OrderID      INTEGER       1                    1
1    ArtID        VARCHAR(20)   1                    2
2    Qty_Ordered  INTEGER       1                    0
sqlite>
```

```
sqlite> PRAGMA table_info(driver);
cid  name         type         notnull  dflt_value  pk
---  -----------  -----------  -------  ----------  --
0    DriverID     INT          1                    1
1    VehicleID    VARCHAR(15)  1                    0
2    DL_No        VARCHAR(20)  1                    0
3    DL_Exp_Date  TEXT         1                    0
```

```
sqlite> PRAGMA table_info(orders);
cid  name          type           notnull  dflt_value  pk
---  ------------  -------------  -------  ----------  --
0    OrderID       INTEGER        1                    1
1    CustomerID    TEXT           1                    0
2    SalesRepID    INTEGER        1                    0
3    ShipmentID    INTEGER        1                    0
4    Order_Date    TEXT           1                    0
5    Order_Amount  DECIMAL(10,2)  0                    0
```

```
sqlite> PRAGMA table_info(shipment);
cid  name         type           notnull  dflt_value  pk
---  ----------  ------------  -------  ----------  --
0    ShipmentID  INTEGER        1                    1
1    OrderID     INTEGER        1                    0
2    VehicleID   VARCHAR(15)    1                    0
3    Del_Street  VARCHAR(100)   1                    0
4    Del_Apt     VARCHAR(20)    1                    0
5    Del_City    VARCHAR(50)    1                    0
6    Del_State   VARCHAR(5)     1                    0
7    Del_Zip     VARCHAR(10)    1                    0
8    Del_Date    TEXT           1                    0
sqlite> .save Assignment_4
sqlite>
```

**Queries:**

```
CREATE TABLE Customer (
	CustomerID TEXT PRIMARY KEY NOT NULL,
	Cus_FName VARCHAR(20) NOT NULL,
	Cus_MName VARCHAR(15),
	Cus_LName VARCHAR(15) NOT NULL,
	Billing_Street VARCHAR(100) NOT NULL,
	Billing_Apt VARCHAR(50) NOT NULL,
	Billing_City VARCHAR(50) NOT NULL,
	Billing_State VARCHAR(5) NOT NULL,
	Billing_Zip VARCHAR(10) NOT NULL,
	Cus_Phone VARCHAR(15),
	Cus_Email VARCHAR(50) NOT NULL
);
CREATE TABLE Employee (
	EmployeeID INT PRIMARY KEY NOT NULL,
	SSN VARCHAR(11) NOT NULL,
	Emp_FName VARCHAR(20) NOT NULL,
	Emp_MName VARCHAR(15),
	Emp_LName VARCHAR(15) NOT NULL,
	DoB TEXT NOT NULL,
	Emp_Type VARCHAR(25) NOT NULL,
	Emp_Salary DECIMAL(10,2) NOT NULL,
	Emp_Street VARCHAR(100) NOT NULL,
	Emp_Apt VARCHAR(20) NOT NULL,
	Emp_City VARCHAR(50) NOT NULL,
	Emp_State VARCHAR(5) NOT NULL,
	Emp_Zip VARCHAR(10) NOT NULL,
	Emp_Phone VARCHAR(15) NOT NULL
);
CREATE TABLE ArtWorks (
	ArtID VARCHAR(20) PRIMARY KEY NOT NULL,
	Art_Des VARCHAR(100) NOT NULL,
	Art_Price DECIMAL(10,2) NOT NULL,
	Art_Qty INTEGER NOT NULL
);
CREATE TABLE Orders (
	OrderID INTEGER PRIMARY KEY NOT NULL,
	CustomerID TEXT NOT NULL,
	SalesRepID INTEGER NOT NULL,
	ShipmentID INTEGER NOT NULL,
	Order_Date TEXT NOT NULL,
	Order_Amount DECIMAL(10,2),
	FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID) ON DELETE CASCADE ON UPDATE CASCADE,
	FOREIGN KEY (SalesRepID) REFERENCES SalesRep (SalesRepID) ON DELETE CASCADE ON UPDATE CASCADE,
	FOREIGN KEY (ShipmentID) REFERENCES Shipment (ShipmentID) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE Vehicle (
	VehicleID VARCHAR(15) PRIMARY KEY NOT NULL,
	DriverID INTEGER NOT NULL,
	VLicense_No VARCHAR(10) NOT NULL,
	VLicense_Exp TEXT NOT NULL,
	Inspection_Exp TEXT NOT NULL,
	FOREIGN KEY (DriverID) REFERENCES Driver (DriverID) ON DELETE CASCADE ON UPDATE CASCADE
);
CREATE TABLE Shipment (
	ShipmentID INTEGER PRIMARY KEY NOT NULL,
	OrderID INTEGER NOT NULL,
	VehicleID VARCHAR(15) NOT NULL,
	Del_Street VARCHAR(100) NOT NULL,
	Del_Apt VARCHAR(20) NOT NULL,
	Del_City VARCHAR(50) NOT NULL,
	Del_State VARCHAR(5) NOT NULL,
	Del_Zip VARCHAR(10) NOT NULL,
	Del_Date TEXT NOT NULL,
	FOREIGN KEY (OrderID) REFERENCES Orders (OrderID) ON DELETE CASCADE ON UPDATE CASCADE,
	FOREIGN KEY (VehicleID) REFERENCES Vehicle (VehicleID) ON DELETE CASCADE ON UPDATE CASCADE
);
CREATE TABLE SalesRep (
	SalesRepID INT PRIMARY KEY NOT NULL,
	Commission DECIMAL(10,2) NOT NULL,
	FOREIGN KEY (SalesRepID) REFERENCES Employee (SalesRepID) ON DELETE CASCADE ON UPDATE CASCADE
);
CREATE TABLE Driver (
	DriverID INT PRIMARY KEY NOT NULL,
	VehicleID VARCHAR(15) NOT NULL,
	DL_No VARCHAR(20) NOT NULL,
	DL_Exp_Date TEXT NOT NULL,
	FOREIGN KEY (VehicleID) REFERENCES Vehicle (VehicleID) ON DELETE CASCADE ON UPDATE CASCADE,
	FOREIGN KEY (DriverID) REFERENCES Employee (DriverID) ON DELETE CASCADE ON UPDATE CASCADE
);
CREATE TABLE Assign (
	OrderID INTEGER NOT NULL,
	ArtID VARCHAR(20) NOT NULL,
	Qty_Ordered INTEGER NOT NULL,
	PRIMARY KEY (OrderID, ArtID),
	FOREIGN KEY (ArtID) REFERENCES ArtWorks (ArtID) ON DELETE CASCADE ON UPDATE CASCADE,
	FOREIGN KEY (OrderID) REFERENCES Orders (OrderID) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
.Header on
.Mode column
SELECT * FROM artworks;
SELECT * FROM customer;
SELECT * FROM employee;
SELECT * FROM salesrep;
SELECT * FROM vehicle;
SELECT * FROM assign;
SELECT * FROM driver;
SELECT * FROM orders;
SELECT * FROM shipment;
PRAGMA table_info(artworks);
PRAGMA table_info(customer);
PRAGMA table_info(employee);
PRAGMA table_info(salesrep);
PRAGMA table_info(vehicle);
PRAGMA table_info(assign);
PRAGMA table_info(driver);
PRAGMA table_info(orders);
PRAGMA table_info(shipment);
.save Assignment_4
```