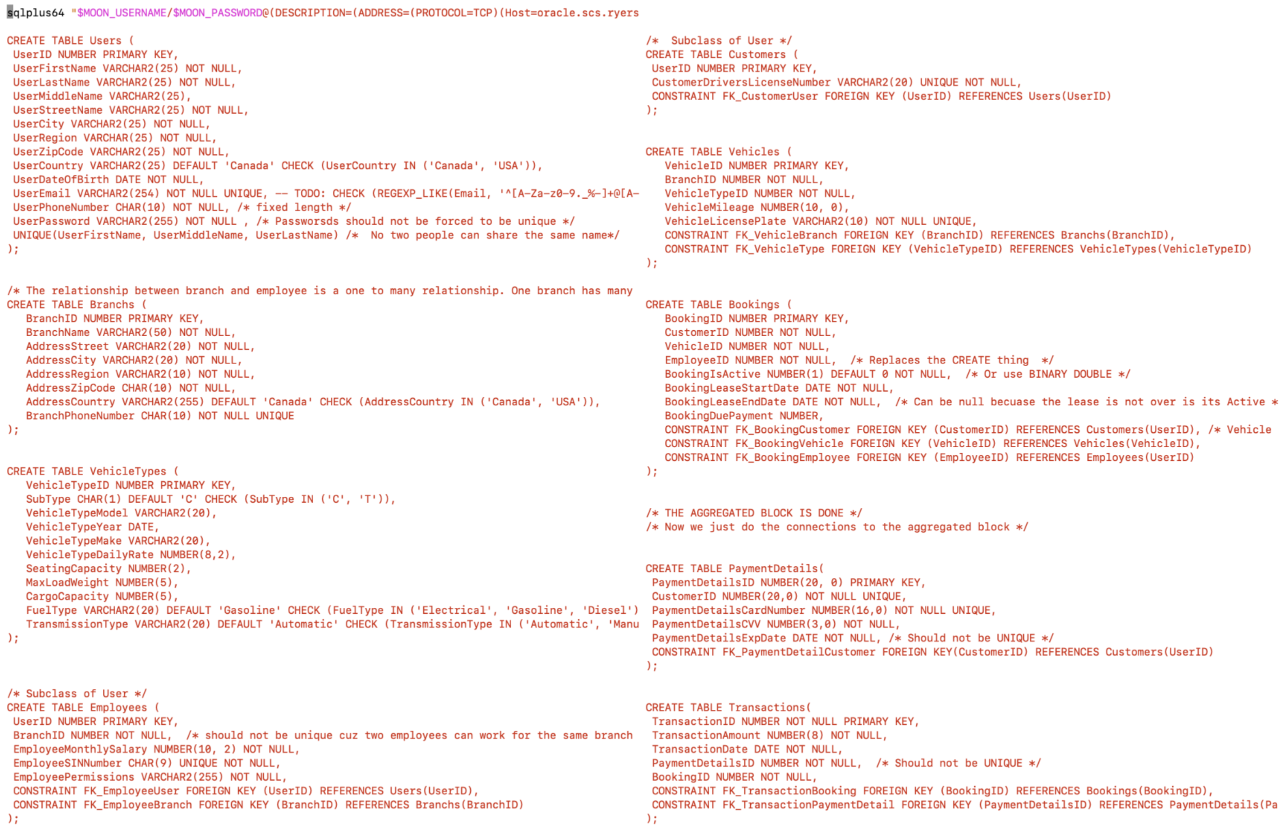
Menu

Program:

```
#!/bin/sh

# README ################

# 1. Set necessary env variables for the scs.oracle credentials
# export MOON_USERNAME=""
# export MOON_PASSWORD=""

# 2. Run script with bash as shown below
# bash menu.sh

# END ###################

Pause()
{
 read -n1 -r -p "Press any key to continue..."
 echo ""
}

MainMenu()
{
while [ "$CHOICE" != "START" ]
do
#    clear
    echo "================================================================"
    echo "| Oracle All Inclusive Tool |"
    echo "| Main Menu - Select Desired Operation(s): |"
    echo "| <CTRL-Z Anytime to Enter Interactive CMD Prompt> |"
    echo "----------------------------------------------------------------"
    echo " $IS_SELECTEDM M) View Manual"
    echo " "
    echo " $IS_SELECTED1 1) Drop Tables"
    echo " $IS_SELECTED2 2) Create Tables"
    echo " $IS_SELECTED3 3) Populate Tables"
    echo " $IS_SELECTED4 4) Query Tables"
    echo " $IS_SELECTED4 5) Manual Entry"
    echo " "
    echo " $IS_SELECTEDX X) Force/Stop/Kill Oracle DB"
    echo " "
    echo " $IS_SELECTEDE E) End/Exit"
    echo "Choose: "
    read CHOICE
    if [ "$CHOICE" = "0" ]
    then
        echo "Nothing Here"
    Pause
    elif [ "$CHOICE" = "1" ]
    then
        bash drop_tables.sh
            Pause
    elif [ "$CHOICE" = "2" ]
    then
    bash create_tables.sh
    Pause
    elif [ "$CHOICE" = "3" ]
    then
        bash populate_tables.sh
        Pause
    elif [ "$CHOICE" = "4" ]
    then
        bash queries.sh
        Pause
    elif [ "$CHOICE" = "5" ]
    then
        bash manual.sh
        Pause
    elif [ "$CHOICE" = "E" ]
    then
        exit
    fi
    done
}

#--COMMENTS BLOCK--
# Main Program
#--COMMENTS BLOCK--

ProgramStart()
{
    # StartMessage
    while [ 1 ]
    do
        MainMenu
    done
}

ProgramStart
```

Create Tables

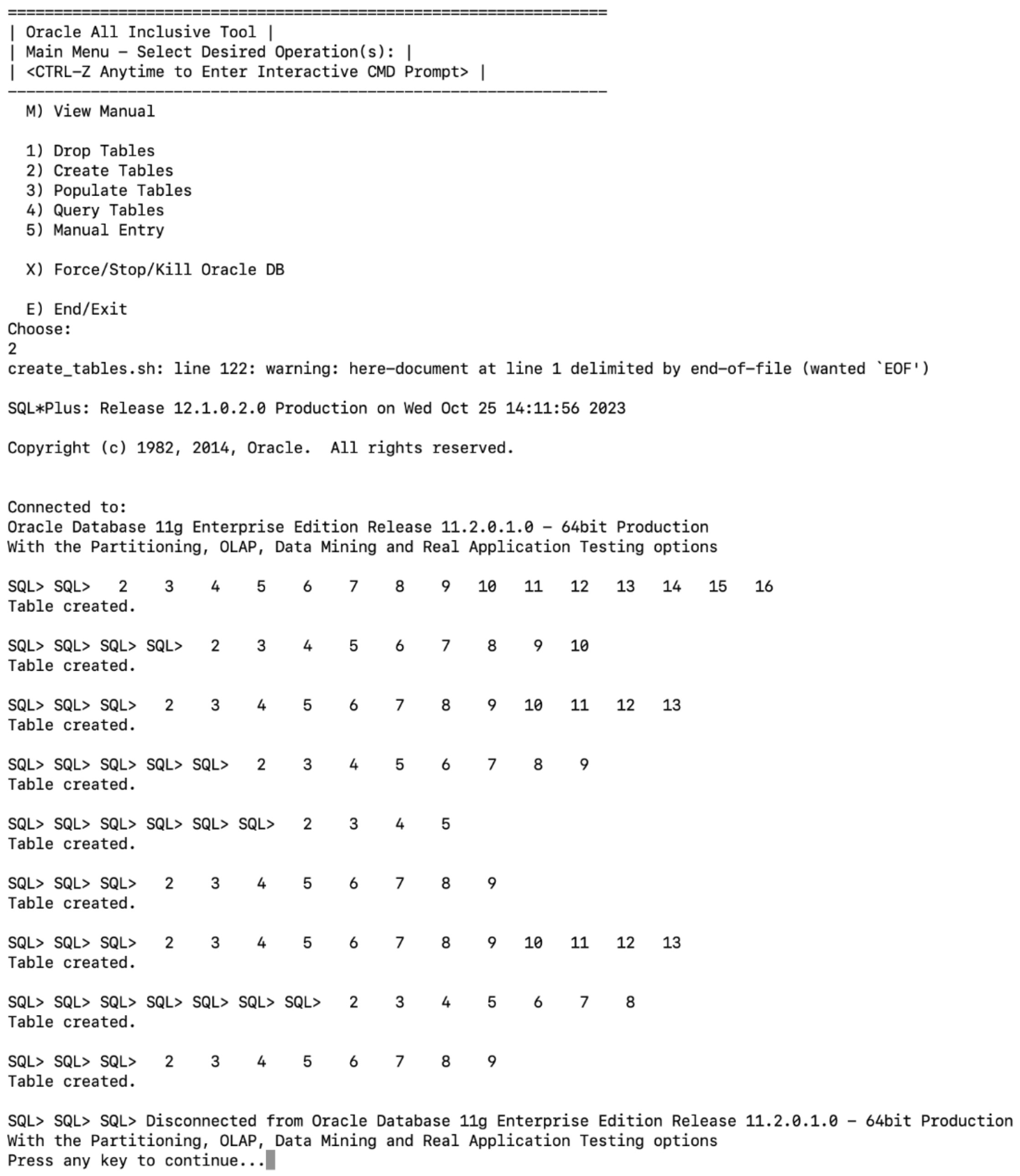
Program:



```
sqlplus64 "$MOON_USERNAME/$MOON_PASSWORD@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryers

CREATE TABLE Users (
 UserID NUMBER PRIMARY KEY,
 UserFirstName VARCHAR2(25) NOT NULL,
 UserLastName VARCHAR2(25) NOT NULL,
 UserMiddleName VARCHAR2(25),
 UserStreetName VARCHAR2(25) NOT NULL,
 UserCity VARCHAR2(25) NOT NULL,
 UserRegion VARCHAR(25) NOT NULL,
 UserZipCode VARCHAR2(25) NOT NULL,
 UserCountry VARCHAR2(25) DEFAULT 'Canada' CHECK (UserCountry IN ('Canada', 'USA')),
 UserDateOfBirth DATE NOT NULL,
 UserEmail VARCHAR2(254) NOT NULL UNIQUE, -- TODO: CHECK (REGEXP_LIKE(Email, '^[A-Za-z0-9._%-]+@[A-
 UserPhoneNumber CHAR(10) NOT NULL, /* fixed length */
 UserPassword VARCHAR2(255) NOT NULL , /* Passworsds should not be forced to be unique */
 UNIQUE(UserFirstName, UserMiddleName, UserLastName) /*  No two people can share the same name*/
);


/* The relationship between branch and employee is a one to many relationship. One branch has many
CREATE TABLE Branchs (
   BranchID NUMBER PRIMARY KEY,
   BranchName VARCHAR2(50) NOT NULL,
   AddressStreet VARCHAR2(20) NOT NULL,
   AddressCity VARCHAR2(20) NOT NULL,
   AddressRegion VARCHAR2(10) NOT NULL,
   AddressZipCode CHAR(10) NOT NULL,
   AddressCountry VARCHAR2(255) DEFAULT 'Canada' CHECK (AddressCountry IN ('Canada', 'USA')),
   BranchPhoneNumber CHAR(10) NOT NULL UNIQUE
);


CREATE TABLE VehicleTypes (
   VehicleTypeID NUMBER PRIMARY KEY,
   SubType CHAR(1) DEFAULT 'C' CHECK (SubType IN ('C', 'T')),
   VehicleTypeModel VARCHAR2(20),
   VehicleTypeYear DATE,
   VehicleTypeMake VARCHAR2(20),
   VehicleTypeDailyRate NUMBER(8,2),
   SeatingCapacity NUMBER(2),
   MaxLoadWeight NUMBER(5),
   CargoCapacity NUMBER(5),
   FuelType VARCHAR2(20) DEFAULT 'Gasoline' CHECK (FuelType IN ('Electrical', 'Gasoline', 'Diesel')
   TransmissionType VARCHAR2(20) DEFAULT 'Automatic' CHECK (TransmissionType IN ('Automatic', 'Manu
);


/* Subclass of User */
CREATE TABLE Employees (
 UserID NUMBER PRIMARY KEY,
 BranchID NUMBER NOT NULL,  /* should not be unique cuz two employees can work for the same branch
 EmployeeMonthlySalary NUMBER(10, 2) NOT NULL,
 EmployeeSINNumber CHAR(9) UNIQUE NOT NULL,
 EmployeePermissions VARCHAR2(255) NOT NULL,
 CONSTRAINT FK_EmployeeUser FOREIGN KEY (UserID) REFERENCES Users(UserID),
 CONSTRAINT FK_EmployeeBranch FOREIGN KEY (BranchID) REFERENCES Branchs(BranchID)
);
```

```
/*  Subclass of User */
CREATE TABLE Customers (
 UserID NUMBER PRIMARY KEY,
 CustomerDriversLicenseNumber VARCHAR2(20) UNIQUE NOT NULL,
 CONSTRAINT FK_CustomerUser FOREIGN KEY (UserID) REFERENCES Users(UserID)
);


CREATE TABLE Vehicles (
   VehicleID NUMBER PRIMARY KEY,
   BranchID NUMBER NOT NULL,
   VehicleTypeID NUMBER NOT NULL,
   VehicleMileage NUMBER(10, 0),
   VehicleLicensePlate VARCHAR2(10) NOT NULL UNIQUE,
   CONSTRAINT FK_VehicleBranch FOREIGN KEY (BranchID) REFERENCES Branchs(BranchID),
   CONSTRAINT FK_VehicleType FOREIGN KEY (VehicleTypeID) REFERENCES VehicleTypes(VehicleTypeID)
);


CREATE TABLE Bookings (
   BookingID NUMBER PRIMARY KEY,
   CustomerID NUMBER NOT NULL,
   VehicleID NUMBER NOT NULL,
   EmployeeID NUMBER NOT NULL,  /* Replaces the CREATE thing  */
   BookingIsActive NUMBER(1) DEFAULT 0 NOT NULL,  /* Or use BINARY DOUBLE */
   BookingLeaseStartDate DATE NOT NULL,
   BookingLeaseEndDate DATE NOT NULL,  /* Can be null becuase the lease is not over is its Active *
   BookingDuePayment NUMBER,
   CONSTRAINT FK_BookingCustomer FOREIGN KEY (CustomerID) REFERENCES Customers(UserID), /* Vehicle
   CONSTRAINT FK_BookingVehicle FOREIGN KEY (VehicleID) REFERENCES Vehicles(VehicleID),
   CONSTRAINT FK_BookingEmployee FOREIGN KEY (EmployeeID) REFERENCES Employees(UserID)
);


/* THE AGGREGATED BLOCK IS DONE */
/* Now we just do the connections to the aggregated block */


CREATE TABLE PaymentDetails(
 PaymentDetailsID NUMBER(20, 0) PRIMARY KEY,
 CustomerID NUMBER(20,0) NOT NULL UNIQUE,
 PaymentDetailsCardNumber NUMBER(16,0) NOT NULL UNIQUE,
 PaymentDetailsCVV NUMBER(3,0) NOT NULL,
 PaymentDetailsExpDate DATE NOT NULL, /* Should not be UNIQUE */
 CONSTRAINT FK_PaymentDetailCustomer FOREIGN KEY(CustomerID) REFERENCES Customers(UserID)
);


CREATE TABLE Transactions(
 TransactionID NUMBER NOT NULL PRIMARY KEY,
 TransactionAmount NUMBER(8) NOT NULL,
 TransactionDate DATE NOT NULL,
 PaymentDetailsID NUMBER NOT NULL,  /* Should not be UNIQUE */
 BookingID NUMBER NOT NULL,
 CONSTRAINT FK_TransactionBooking FOREIGN KEY (BookingID) REFERENCES Bookings(BookingID),
 CONSTRAINT FK_TransactionPaymentDetail FOREIGN KEY (PaymentDetailsID) REFERENCES PaymentDetails(Pa
);
```

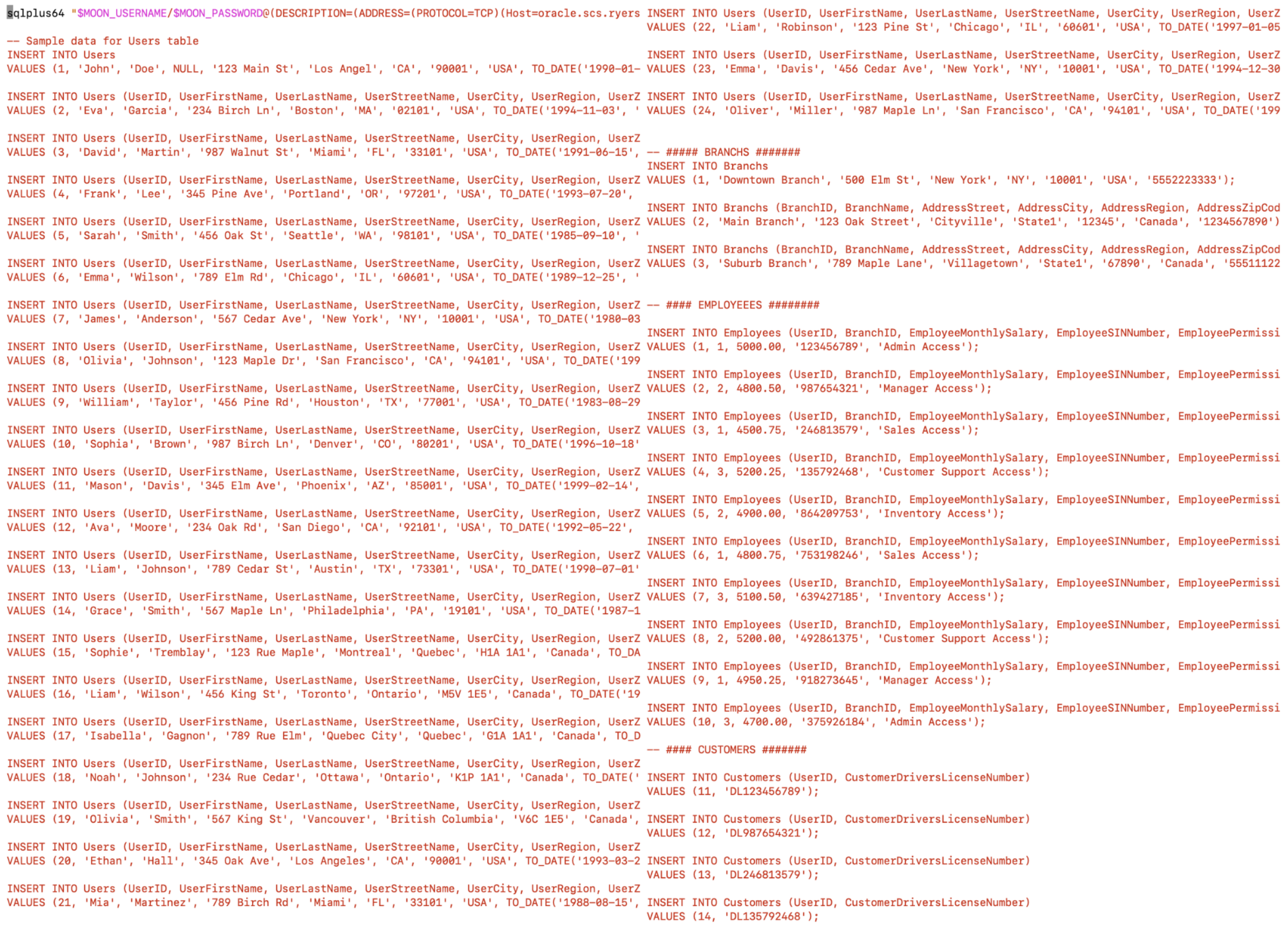
Output:



```
=================================================================
| Oracle All Inclusive Tool |
| Main Menu - Select Desired Operation(s): |
| <CTRL-Z Anytime to Enter Interactive CMD Prompt> |
-----------------------------------------------------------------
  M) View Manual

  1) Drop Tables
  2) Create Tables
  3) Populate Tables
  4) Query Tables
  5) Manual Entry

  X) Force/Stop/Kill Oracle DB

  E) End/Exit
Choose:
2
create_tables.sh: line 122: warning: here-document at line 1 delimited by end-of-file (wanted `EOF')

SQL*Plus: Release 12.1.0.2.0 Production on Wed Oct 25 14:11:56 2023

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>   2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
Table created.

SQL> SQL> SQL> SQL>   2    3    4    5    6    7    8    9   10
Table created.

SQL> SQL> SQL>   2    3    4    5    6    7    8    9   10   11   12   13
Table created.

SQL> SQL> SQL> SQL> SQL>   2    3    4    5    6    7    8    9
Table created.

SQL> SQL> SQL> SQL> SQL> SQL>   2    3    4    5
Table created.

SQL> SQL> SQL>   2    3    4    5    6    7    8    9
Table created.

SQL> SQL> SQL>   2    3    4    5    6    7    8    9   10   11   12   13
Table created.

SQL> SQL> SQL> SQL> SQL> SQL> SQL>   2    3    4    5    6    7    8
Table created.

SQL> SQL> SQL>   2    3    4    5    6    7    8    9
Table created.

SQL> SQL> SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Press any key to continue...
```

Populate Tables

Program:



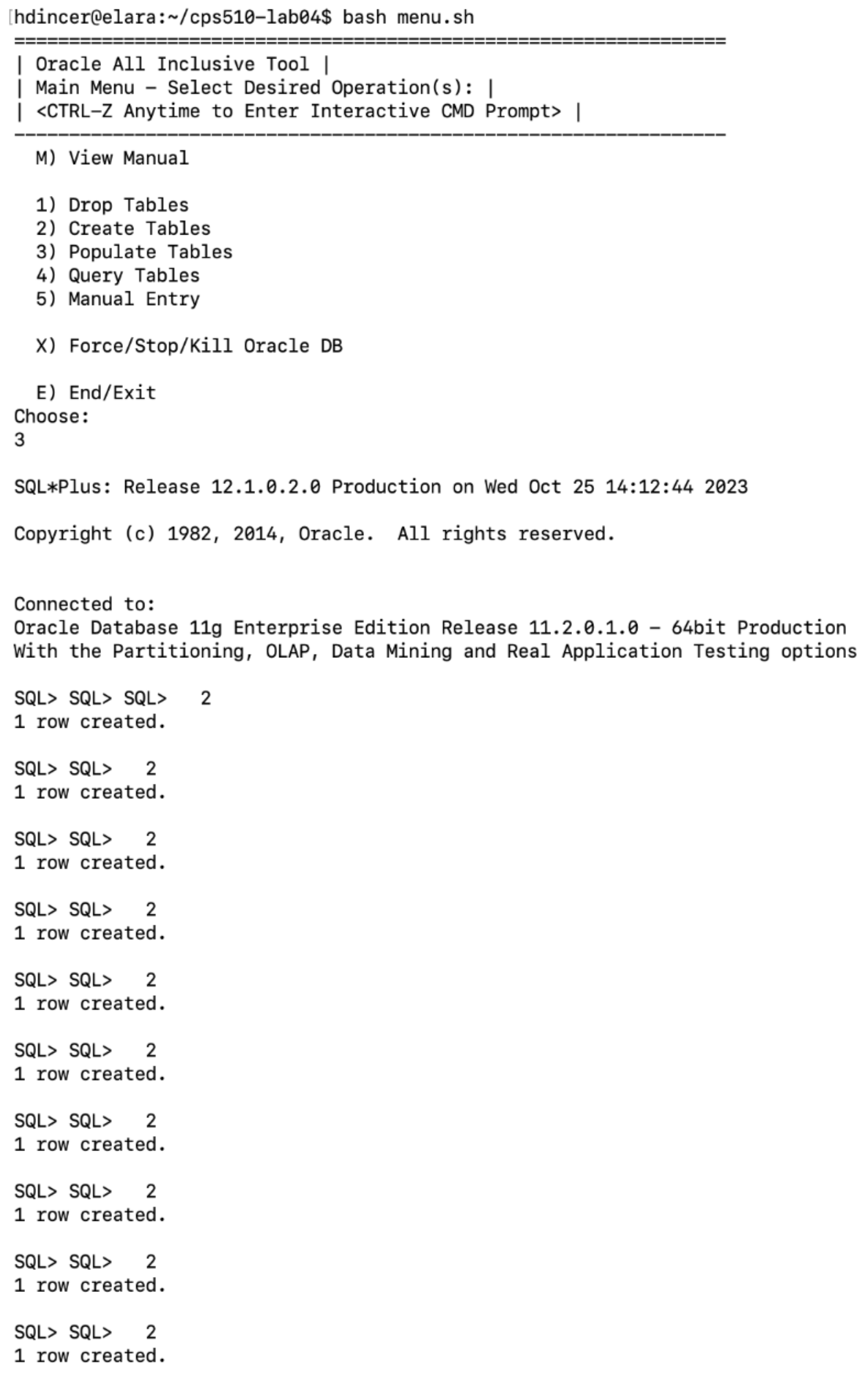
```
sqlplus64 "$MOON_USERNAME/$MOON_PASSWORD@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryers

-- Sample data for Users table
INSERT INTO Users
VALUES (1, 'John', 'Doe', NULL, '123 Main St', 'Los Angel', 'CA', '90001', 'USA', TO_DATE('1990-01-

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (2, 'Eva', 'Garcia', '234 Birch Ln', 'Boston', 'MA', '02101', 'USA', TO_DATE('1994-11-03', '

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (3, 'David', 'Martin', '987 Walnut St', 'Miami', 'FL', '33101', 'USA', TO_DATE('1991-06-15',

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (4, 'Frank', 'Lee', '345 Pine Ave', 'Portland', 'OR', '97201', 'USA', TO_DATE('1993-07-20',

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (5, 'Sarah', 'Smith', '456 Oak St', 'Seattle', 'WA', '98101', 'USA', TO_DATE('1985-09-10', '

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (6, 'Emma', 'Wilson', '789 Elm Rd', 'Chicago', 'IL', '60601', 'USA', TO_DATE('1989-12-25', '

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (7, 'James', 'Anderson', '567 Cedar Ave', 'New York', 'NY', '10001', 'USA', TO_DATE('1980-03

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (8, 'Olivia', 'Johnson', '123 Maple Dr', 'San Francisco', 'CA', '94101', 'USA', TO_DATE('199

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (9, 'William', 'Taylor', '456 Pine Rd', 'Houston', 'TX', '77001', 'USA', TO_DATE('1983-08-29

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (10, 'Sophia', 'Brown', '987 Birch Ln', 'Denver', 'CO', '80201', 'USA', TO_DATE('1996-10-18'

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (11, 'Mason', 'Davis', '345 Elm Ave', 'Phoenix', 'AZ', '85001', 'USA', TO_DATE('1999-02-14',

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (12, 'Ava', 'Moore', '234 Oak Rd', 'San Diego', 'CA', '92101', 'USA', TO_DATE('1992-05-22',

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (13, 'Liam', 'Johnson', '789 Cedar St', 'Austin', 'TX', '73301', 'USA', TO_DATE('1990-07-01'

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (14, 'Grace', 'Smith', '567 Maple Ln', 'Philadelphia', 'PA', '19101', 'USA', TO_DATE('1987-1

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (15, 'Sophie', 'Tremblay', '123 Rue Maple', 'Montreal', 'Quebec', 'H1A 1A1', 'Canada', TO_DA

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (16, 'Liam', 'Wilson', '456 King St', 'Toronto', 'Ontario', 'M5V 1E5', 'Canada', TO_DATE('19

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (17, 'Isabella', 'Gagnon', '789 Rue Elm', 'Quebec City', 'Quebec', 'G1A 1A1', 'Canada', TO_D

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (18, 'Noah', 'Johnson', '234 Rue Cedar', 'Ottawa', 'Ontario', 'K1P 1A1', 'Canada', TO_DATE('

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (19, 'Olivia', 'Smith', '567 King St', 'Vancouver', 'British Columbia', 'V6C 1E5', 'Canada',

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (20, 'Ethan', 'Hall', '345 Oak Ave', 'Los Angeles', 'CA', '90001', 'USA', TO_DATE('1993-03-2

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (21, 'Mia', 'Martinez', '789 Birch Rd', 'Miami', 'FL', '33101', 'USA', TO_DATE('1988-08-15',
```

```
INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (22, 'Liam', 'Robinson', '123 Pine St', 'Chicago', 'IL', '60601', 'USA', TO_DATE('1997-01-05

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (23, 'Emma', 'Davis', '456 Cedar Ave', 'New York', 'NY', '10001', 'USA', TO_DATE('1994-12-30

INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (24, 'Oliver', 'Miller', '987 Maple Ln', 'San Francisco', 'CA', '94101', 'USA', TO_DATE('199

-- ##### BRANCHS #######
INSERT INTO Branchs
VALUES (1, 'Downtown Branch', '500 Elm St', 'New York', 'NY', '10001', 'USA', '5552223333');

INSERT INTO Branchs (BranchID, BranchName, AddressStreet, AddressCity, AddressRegion, AddressZipCod
VALUES (2, 'Main Branch', '123 Oak Street', 'Cityville', 'State1', '12345', 'Canada', '1234567890')

INSERT INTO Branchs (BranchID, BranchName, AddressStreet, AddressCity, AddressRegion, AddressZipCod
VALUES (3, 'Suburb Branch', '789 Maple Lane', 'Villagetown', 'State1', '67890', 'Canada', '55511122

-- #### EMPLOYEEES ########

INSERT INTO Employees (UserID, BranchID, EmployeeMonthlySalary, EmployeeSINNumber, EmployeePermissi
VALUES (1, 1, 5000.00, '123456789', 'Admin Access');

INSERT INTO Employees (UserID, BranchID, EmployeeMonthlySalary, EmployeeSINNumber, EmployeePermissi
VALUES (2, 2, 4800.50, '987654321', 'Manager Access');

INSERT INTO Employees (UserID, BranchID, EmployeeMonthlySalary, EmployeeSINNumber, EmployeePermissi
VALUES (3, 1, 4500.75, '246813579', 'Sales Access');

INSERT INTO Employees (UserID, BranchID, EmployeeMonthlySalary, EmployeeSINNumber, EmployeePermissi
VALUES (4, 3, 5200.25, '135792468', 'Customer Support Access');

INSERT INTO Employees (UserID, BranchID, EmployeeMonthlySalary, EmployeeSINNumber, EmployeePermissi
VALUES (5, 2, 4900.00, '864209753', 'Inventory Access');

INSERT INTO Employees (UserID, BranchID, EmployeeMonthlySalary, EmployeeSINNumber, EmployeePermissi
VALUES (6, 1, 4800.75, '753198246', 'Sales Access');

INSERT INTO Employees (UserID, BranchID, EmployeeMonthlySalary, EmployeeSINNumber, EmployeePermissi
VALUES (7, 3, 5100.50, '639427185', 'Inventory Access');

INSERT INTO Employees (UserID, BranchID, EmployeeMonthlySalary, EmployeeSINNumber, EmployeePermissi
VALUES (8, 2, 5200.00, '492861375', 'Customer Support Access');

INSERT INTO Employees (UserID, BranchID, EmployeeMonthlySalary, EmployeeSINNumber, EmployeePermissi
VALUES (9, 1, 4950.25, '918273645', 'Manager Access');

INSERT INTO Employees (UserID, BranchID, EmployeeMonthlySalary, EmployeeSINNumber, EmployeePermissi
VALUES (10, 3, 4700.00, '375926184', 'Admin Access');

-- #### CUSTOMERS #######

INSERT INTO Customers (UserID, CustomerDriversLicenseNumber)
VALUES (11, 'DL123456789');

INSERT INTO Customers (UserID, CustomerDriversLicenseNumber)
VALUES (12, 'DL987654321');

INSERT INTO Customers (UserID, CustomerDriversLicenseNumber)
VALUES (13, 'DL246813579');

INSERT INTO Customers (UserID, CustomerDriversLicenseNumber)
VALUES (14, 'DL135792468');
```

```
INSERT INTO Users (UserID, UserFirstName, UserLastName, UserStreetName, UserCity, UserRegion, UserZ
VALUES (21, 'Mia', 'Martinez', '789 Birch Rd', 'Miami', 'FL', '33101', 'USA', TO_DATE('1988-08-15',


INSERT INTO Customers (UserID, CustomerDriversLicenseNumber)
VALUES (15, 'DL864209753');

INSERT INTO Customers (UserID, CustomerDriversLicenseNumber)
VALUES (16, 'DL753198246');

INSERT INTO Customers (UserID, CustomerDriversLicenseNumber)
VALUES (17, 'DL639427185');

INSERT INTO Customers (UserID, CustomerDriversLicenseNumber)
VALUES (18, 'DL492861375');

INSERT INTO Customers (UserID, CustomerDriversLicenseNumber)
VALUES (19, 'DL918273645');

INSERT INTO Customers (UserID, CustomerDriversLicenseNumber)
VALUES (20, 'DL375926184');

INSERT INTO Customers (UserID, CustomerDriversLicenseNumber)
VALUES (21, 'DL654321123');

INSERT INTO Customers (UserID, CustomerDriversLicenseNumber)
VALUES (22, 'DL987774321');

INSERT INTO Customers (UserID, CustomerDriversLicenseNumber)
VALUES (23, 'DL123253789');

INSERT INTO Customers (UserID, CustomerDriversLicenseNumber)
VALUES (24, 'DL567890123');


-- ##### VEHICLE TYPES #######
INSERT INTO VehicleTypes (VehicleTypeID, VehicleTypeModel, VehicleTypeYear, VehicleTypeMake, Vehicl
VALUES (1, 'Sedan', TO_DATE('2023-01-01', 'YYYY-MM-DD'), 'Toyota', 50.00, 5, NULL, NULL, 'Gasoline'

INSERT INTO VehicleTypes (VehicleTypeID, VehicleTypeModel, VehicleTypeYear, VehicleTypeMake, Vehicl
VALUES (2, 'SUV', TO_DATE('2023-01-01', 'YYYY-MM-DD'), 'Ford', 60.00, 7, NULL, NULL, 'Gasoline', 'A

INSERT INTO VehicleTypes (VehicleTypeID, VehicleTypeModel, VehicleTypeYear, VehicleTypeMake, Vehicl
VALUES (3, 'Truck', TO_DATE('2023-01-01', 'YYYY-MM-DD'), 'Chevrolet', 70.00, 3, 5000, 20, 'Gasoline

INSERT INTO VehicleTypes (VehicleTypeID, VehicleTypeModel, VehicleTypeYear, VehicleTypeMake, Vehicl
VALUES (4, 'Compact', TO_DATE('2023-01-01', 'YYYY-MM-DD'), 'Honda', 45.00, 4, NULL, NULL, 'Gasoline

INSERT INTO VehicleTypes (VehicleTypeID, VehicleTypeModel, VehicleTypeYear, VehicleTypeMake, Vehicl
VALUES (5, 'Van', TO_DATE('2023-01-01', 'YYYY-MM-DD'), 'Nissan', 65.00, 8, NULL, 25, 'Gasoline', 'A

-- ##### VEHICLES #######
INSERT INTO Vehicles (VehicleID, BranchID, VehicleTypeID, VehicleMileage, VehicleLicensePlate)
VALUES (1, 1, 1, 1000, 'AB123CD');

INSERT INTO Vehicles (VehicleID, BranchID, VehicleTypeID, VehicleMileage, VehicleLicensePlate)
VALUES (2, 2, 2, 1200, 'DE456FG');

INSERT INTO Vehicles (VehicleID, BranchID, VehicleTypeID, VehicleMileage, VehicleLicensePlate)
VALUES (3, 1, 3, 800, 'GH789IJ');

INSERT INTO Vehicles (VehicleID, BranchID, VehicleTypeID, VehicleMileage, VehicleLicensePlate)
VALUES (4, 3, 4, 1500, 'KL012MN');

INSERT INTO Vehicles (VehicleID, BranchID, VehicleTypeID, VehicleMileage, VehicleLicensePlate)
VALUES (5, 2, 5, 2000, 'OP345QR');
```

```
VALUES (13, 'DL246813579');

INSERT INTO Customers (UserID, CustomerDriversLicenseNumber)
VALUES (14, 'DL135792468');

INSERT INTO Vehicles (VehicleID, BranchID, VehicleTypeID, VehicleMileage, VehicleLicensePlate)
VALUES (6, 1, 1, 900, 'ST678UV');

INSERT INTO Vehicles (VehicleID, BranchID, VehicleTypeID, VehicleMileage, VehicleLicensePlate)
VALUES (7, 3, 2, 1100, 'WX901YZ');

INSERT INTO Vehicles (VehicleID, BranchID, VehicleTypeID, VehicleMileage, VehicleLicensePlate)
VALUES (8, 2, 3, 750, 'AB234CD');

INSERT INTO Vehicles (VehicleID, BranchID, VehicleTypeID, VehicleMileage, VehicleLicensePlate)
VALUES (9, 1, 4, 1400, 'DE567FG');

INSERT INTO Vehicles (VehicleID, BranchID, VehicleTypeID, VehicleMileage, VehicleLicensePlate)
VALUES (10, 3, 5, 1800, 'GH890IJ');

INSERT INTO Vehicles (VehicleID, BranchID, VehicleTypeID, VehicleMileage, VehicleLicensePlate)
VALUES (11, 1, 1, 950, 'KL123MN');

INSERT INTO Vehicles (VehicleID, BranchID, VehicleTypeID, VehicleMileage, VehicleLicensePlate)
VALUES (12, 2, 2, 1250, 'OP456QR');

INSERT INTO Vehicles (VehicleID, BranchID, VehicleTypeID, VehicleMileage, VehicleLicensePlate)
VALUES (13, 1, 3, 780, 'ST678VW');

INSERT INTO Vehicles (VehicleID, BranchID, VehicleTypeID, VehicleMileage, VehicleLicensePlate)
VALUES (14, 3, 4, 1480, 'XZ901YZ');

INSERT INTO Vehicles (VehicleID, BranchID, VehicleTypeID, VehicleMileage, VehicleLicensePlate)
VALUES (15, 2, 5, 1900, 'AB345CE');


-- Sample data for Bookings table
INSERT INTO Bookings
VALUES (1, 18, 2, 8, 1, TO_DATE('2023-09-20', 'YYYY-MM-DD'), TO_DATE('2023-09-25', 'YYYY-MM-DD'), 2

INSERT INTO Bookings (BookingID, CustomerID, VehicleID, EmployeeID, BookingIsActive, BookingLeaseSt
VALUES (2, 15, 1, 5, 1, TO_DATE('2023-10-20', 'YYYY-MM-DD'), TO_DATE('2023-10-27', 'YYYY-MM-DD'), 2

INSERT INTO Bookings (BookingID, CustomerID, VehicleID, EmployeeID, BookingIsActive, BookingLeaseSt
VALUES (3, 18, 3, 3, 1, TO_DATE('2023-10-24', 'YYYY-MM-DD'), TO_DATE('2023-10-29', 'YYYY-MM-DD'), 2

INSERT INTO Bookings (BookingID, CustomerID, VehicleID, EmployeeID, BookingIsActive, BookingLeaseSt
VALUES (4, 19, 5, 2, 1, TO_DATE('2023-10-19', 'YYYY-MM-DD'), TO_DATE('2023-10-29', 'YYYY-MM-DD'), 3

INSERT INTO Bookings (BookingID, CustomerID, VehicleID, EmployeeID, BookingIsActive, BookingLeaseSt
VALUES (5, 15, 7, 3, 1, TO_DATE('2023-10-19', 'YYYY-MM-DD'), TO_DATE('2023-10-29', 'YYYY-MM-DD'), 3

-- Sample data for PaymentDetails table
INSERT INTO PaymentDetails
VALUES (1, 15, 1234567890123456, 123, TO_DATE('2025-12-31', 'YYYY-MM-DD'));

 -- Sample data for Transactions table (additional 5 transactions)
INSERT INTO Transactions
VALUES (1, 150.00, TO_DATE('2023-09-22', 'YYYY-MM-DD'), 1, 1);



exit;
EOF
```

Output:

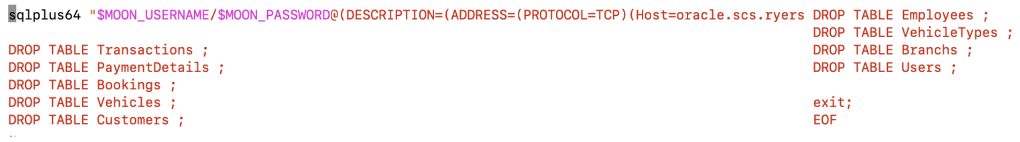


```
hdincer@elara:~/cps510-lab04$ bash menu.sh
=================================================================
| Oracle All Inclusive Tool |
| Main Menu - Select Desired Operation(s): |
| <CTRL-Z Anytime to Enter Interactive CMD Prompt> |
-----------------------------------------------------------------
  M) View Manual

  1) Drop Tables
  2) Create Tables
  3) Populate Tables
  4) Query Tables
  5) Manual Entry

  X) Force/Stop/Kill Oracle DB

  E) End/Exit
Choose:
3

SQL*Plus: Release 12.1.0.2.0 Production on Wed Oct 25 14:12:44 2023

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.
```

```
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL> SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL>   2
1 row created.

SQL> SQL> SQL>   2
1 row created.

SQL> SQL> SQL>   2
1 row created.

SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> Disconnected from Oracle Database 11g
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Press any key to continue...
```

Drop Tables

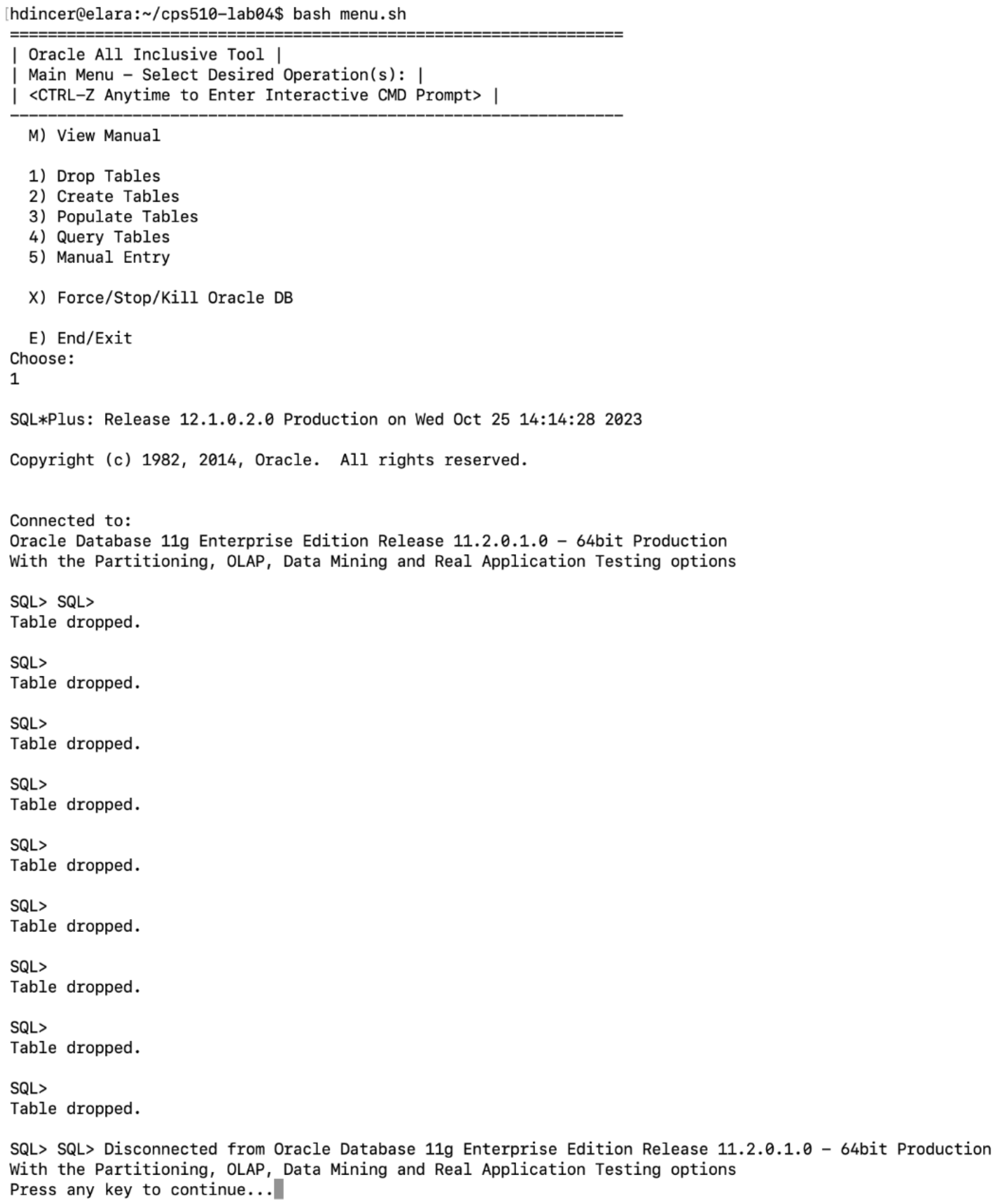
Program:



```
sqlplus64 "$MOON_USERNAME/$MOON_PASSWORD@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryers

DROP TABLE Transactions ;
DROP TABLE PaymentDetails ;
DROP TABLE Bookings ;
DROP TABLE Vehicles ;
DROP TABLE Customers ;
```

```
DROP TABLE Employees ;
DROP TABLE VehicleTypes ;
DROP TABLE Branchs ;
DROP TABLE Users ;

exit;
EOF
```

Output:



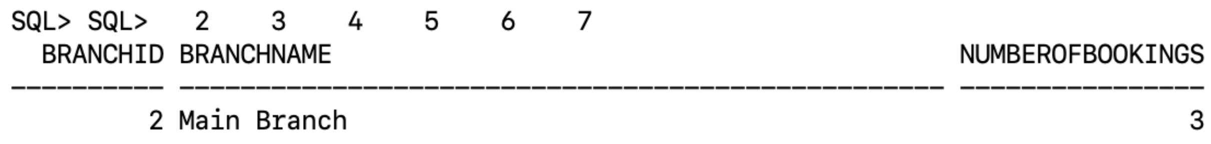
```
hdincer@elara:~/cps510-lab04$ bash menu.sh
=================================================================
| Oracle All Inclusive Tool |
| Main Menu - Select Desired Operation(s): |
| <CTRL-Z Anytime to Enter Interactive CMD Prompt> |
-----------------------------------------------------------------
  M) View Manual

  1) Drop Tables
  2) Create Tables
  3) Populate Tables
  4) Query Tables
  5) Manual Entry

  X) Force/Stop/Kill Oracle DB

  E) End/Exit
Choose:
1

SQL*Plus: Release 12.1.0.2.0 Production on Wed Oct 25 14:14:28 2023

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL> SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Press any key to continue...
```

ADVANCED QUERIES

1. This query finds branches with more than 3 active bookings and retrieves their information.

```
SELECT b.BranchID, b.BranchName, COUNT(*) AS NumberOfBookings
FROM Branchs b
JOIN Employees e ON b.BranchID = e.BranchID
JOIN Bookings bk ON e.UserID = bk.EmployeeID
WHERE bk.BookingIsActive = 1
GROUP BY b.BranchID, b.BranchName
HAVING COUNT(*) > 3;
```

Output:

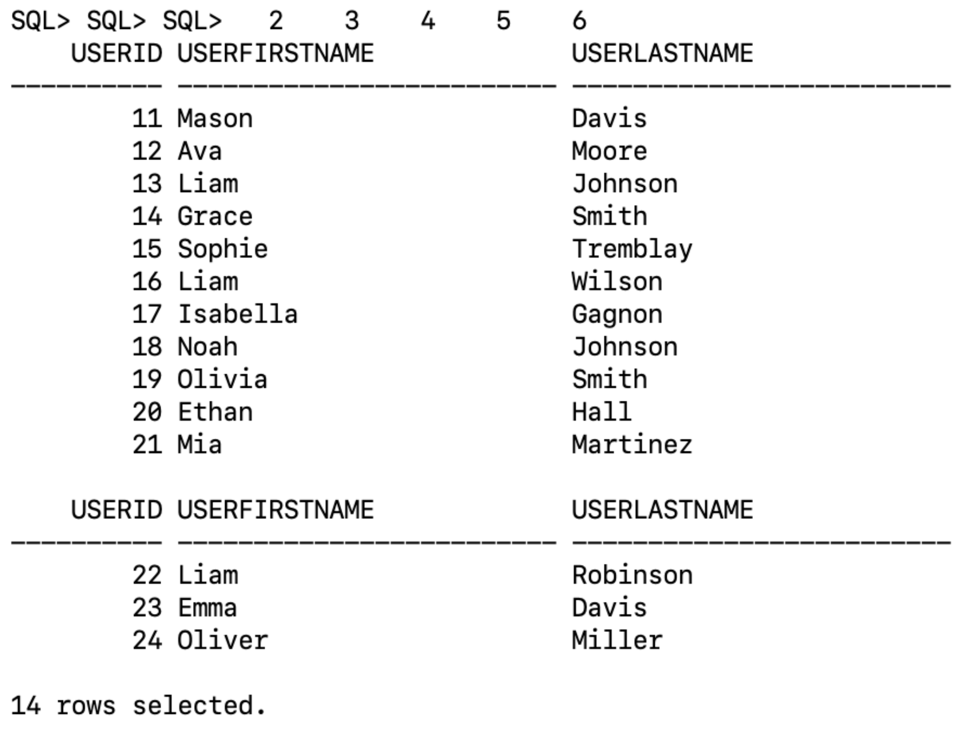


```
SQL> SQL>   2    3    4    5    6    7
  BRANCHID BRANCHNAME                                         NUMBEROFBOOKINGS
---------- -------------------------------------------------- ----------------
         2 Main Branch                                                       3
```

2.This query uses MINUS to retrieve customers who have not made any bookings:

```
SELECT u.UserID, u.UserFirstName, u.UserLastName
FROM Users u
JOIN Customers c ON u.UserID = c.UserID
MINUS
SELECT b.CustomerID, NULL ,NULL
FROM Bookings b;
```

Output:



```
SQL> SQL> SQL>   2    3    4    5    6
    USERID USERFIRSTNAME             USERLASTNAME
---------- ------------------------- -------------------------
        11 Mason                     Davis
        12 Ava                       Moore
        13 Liam                      Johnson
        14 Grace                     Smith
        15 Sophie                    Tremblay
        16 Liam                      Wilson
        17 Isabella                  Gagnon
        18 Noah                      Johnson
        19 Olivia                    Smith
        20 Ethan                     Hall
        21 Mia                       Martinez

    USERID USERFIRSTNAME             USERLASTNAME
---------- ------------------------- -------------------------
        22 Liam                      Robinson
        23 Emma                      Davis
        24 Oliver                    Miller

14 rows selected.
```

3.Retrieves High Salary Employees and Frequent Customers from Users, Customers, and Employees table and Union the result. (Significant Users)

```
SELECT 'High-Salary Employees' AS Category, Employees.UserID,
Users.UserFirstName, Users.UserLastName
FROM Employees
JOIN Users ON Employees.UserID = Users.UserID
WHERE Employees.EmployeeMonthlySalary > 4000
UNION
SELECT 'Frequent Customers' AS Category, Customers.UserID,
Users.UserFirstName, Users.UserLastName
FROM Customers
JOIN Users ON Customers.UserID = Users.UserID
WHERE Customers.UserID IN (
    SELECT Bookings.CustomerID
    FROM Bookings
    GROUP BY Bookings.CustomerID
    HAVING COUNT(*) >= 2
);
```

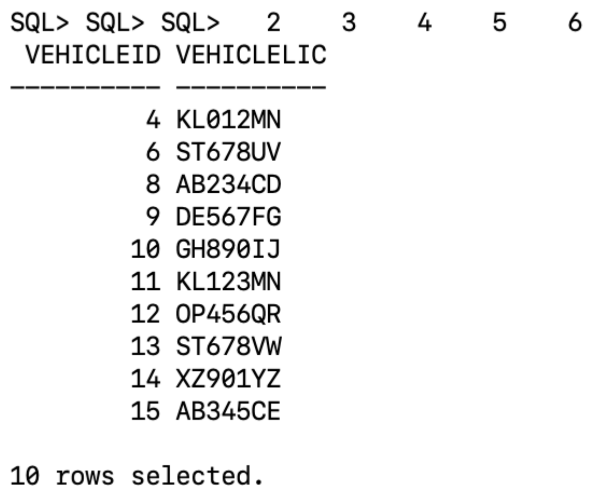
Output:

```
SQL> SQL>   2    3    4    5    6    7    8    9   10   11   12   13   14
CATEGORY                  USERID USERFIRSTNAME
--------------------- ---------- -------------------------
USERLASTNAME
-------------------------
Frequent Customers            15 Sophie
Tremblay

Frequent Customers            18 Noah
Johnson

High-Salary Employees          1 John
Doe


CATEGORY                  USERID USERFIRSTNAME
--------------------- ---------- -------------------------
USERLASTNAME
-------------------------
High-Salary Employees          2 Eva
Garcia

High-Salary Employees          3 David
Martin

High-Salary Employees          4 Frank
Lee


CATEGORY                  USERID USERFIRSTNAME
--------------------- ---------- -------------------------
USERLASTNAME
-------------------------
High-Salary Employees          5 Sarah
Smith

High-Salary Employees          6 Emma
Wilson

High-Salary Employees          7 James
Anderson


CATEGORY                  USERID USERFIRSTNAME
--------------------- ---------- -------------------------
USERLASTNAME
-------------------------
High-Salary Employees          8 Olivia
Johnson

High-Salary Employees          9 William
Taylor

High-Salary Employees         10 Sophia
Brown


12 rows selected.
```

4. LIST ALL VEHICLES THAT HAVE NEVER BEEN BOOKED:

```
SELECT v.VehicleID, v.VehicleLicensePlate
FROM Vehicles v
MINUS
SELECT DISTINCT b.VehicleID, v.VehicleLicensePlate
FROM Bookings b
JOIN Vehicles v ON b.VehicleID = v.VehicleID;
```

Output:

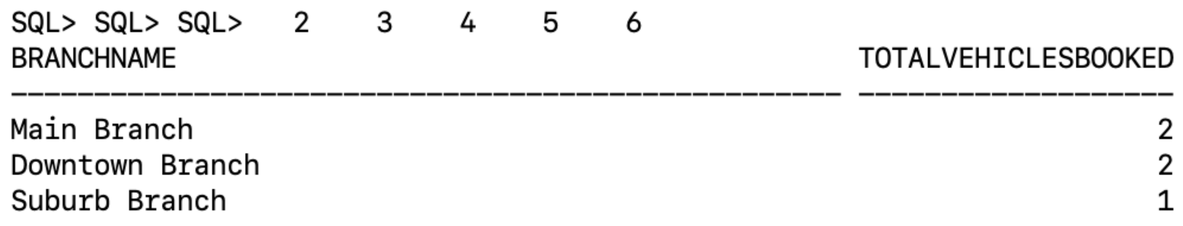


```
SQL> SQL> SQL>   2    3    4    5    6
 VEHICLEID VEHICLELIC
---------- ----------
         4 KL012MN
         6 ST678UV
         8 AB234CD
         9 DE567FG
        10 GH890IJ
        11 KL123MN
        12 OP456QR
        13 ST678VW
        14 XZ901YZ
        15 AB345CE

10 rows selected.
```

5. AMOUNT OF VEHICLES CURRENTLY BOOKED OUT BY A BRANCH

```
SELECT b.BranchName, COUNT(v.VehicleID) AS TotalVehiclesBooked
FROM Branchs b
JOIN Vehicles v ON b.BranchID = v.BranchID
JOIN Bookings bk ON v.VehicleID = bk.VehicleID
WHERE bk.BookingIsActive = 1
GROUP BY b.BranchName;
```

Output:

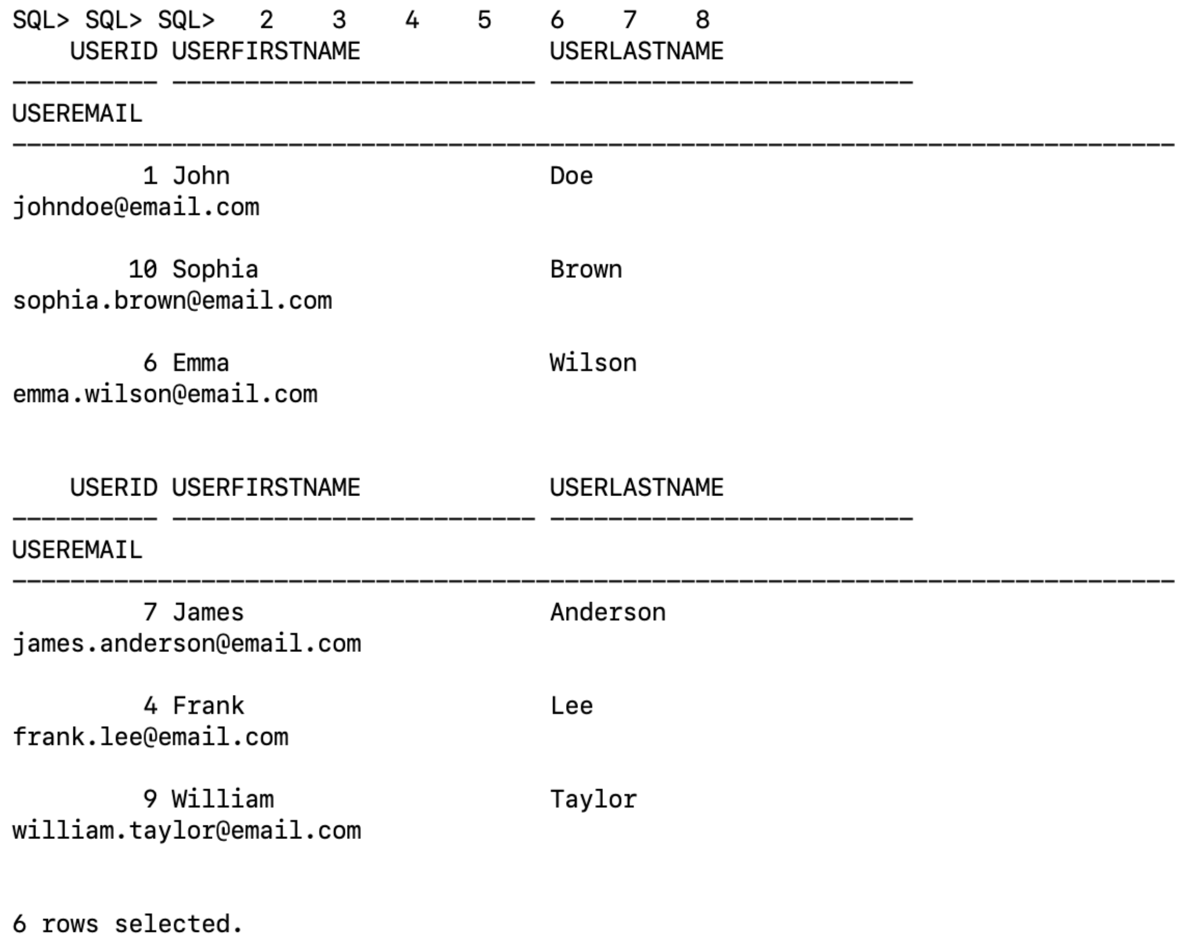


```
SQL> SQL> SQL>   2    3    4    5    6
BRANCHNAME                                         TOTALVEHICLESBOOKED
-------------------------------------------------- -------------------
Main Branch                                                          2
Downtown Branch                                                      2
Suburb Branch                                                        1
```

6.A list of Users that are employees that have not made any bookings.

```
SELECT u.UserID, u.UserFirstName, u.UserLastName, u.UserEmail
FROM Users u
JOIN Employees e ON u.UserID = e.UserID
WHERE NOT EXISTS (
    SELECT 1
    FROM Bookings b
    WHERE b.EmployeeID = e.UserID
);
```

Output:



```
SQL> SQL> SQL>   2    3    4    5    6    7    8
    USERID USERFIRSTNAME             USERLASTNAME
---------- ------------------------- -------------------------
USEREMAIL
--------------------------------------------------------------------------------
         1 John                      Doe
johndoe@email.com

        10 Sophia                    Brown
sophia.brown@email.com

         6 Emma                      Wilson
emma.wilson@email.com


    USERID USERFIRSTNAME             USERLASTNAME
---------- ------------------------- -------------------------
USEREMAIL
--------------------------------------------------------------------------------
         7 James                     Anderson
james.anderson@email.com

         4 Frank                     Lee
frank.lee@email.com

         9 William                   Taylor
william.taylor@email.com


6 rows selected.
```