**A close up of a logo

Description automatically generated**

**DATABASE PROJECT**

**CAR RENTAL SYSTEM**

BUS 601 Project  
Instructor: Tariq Islam

Date : March 2nd 2020

**Submitted by :**

**ANCHAL BHAMORE**

**INDEX**

1. Project detail------------------------------------------- 3
2. Customer Requirements---------------------------- 4
3. Installation --------------------------------------------- 4
4. Creating Database------------------------------------ 5
5. Creating User ID---------------------------------------- 5
6. Granting privileges and roles------------------------ 6
7. [Introduction--------------------------------------------- 5](#_TOC_250008)
8. [Entities ---------------------------------------------------](#_TOC_250007)  8
9. [Relations -------------------------------------------------](#_TOC_250006)  9
10. [Planning and Designing------------------------------ 10](#_TOC_250005)
11. [ERD Diagram ------------------------------------------ 11](#_TOC_250004)
12. Create, Insert and Constraints -------------------- 12
13. SQL Queries ------------------------------------------ 35
14. Advanced SQL Queries----------------------------- 40
15. Report-------------------------------------------------- 49

**Description of the Project:**

A Wonderland car rental company wants to develop a relation database to monitor customers, rentals, pickup date, maintenance and employees.

CRC's fleet consists of cars of different types. A car is described via a unique code (VIN), a description, color, brand and model. CRC should also store in this database its customers. A customer is described by a unique ID, SSN, Name (First, Last), email, mobile phone number and lives in a state and country. A rental is described by a unique reservation number, it has an amount and contains the pickup date and the return date. Entity-Relationship Diagram (ERD) Use the Entity-Relationship Diagram (ERD) to model entities, relationships, attributes, cardinalities, and all necessary constraints. Use any tool you like to draw the ERD.

**Mission**

Our mission is to satisfy our customer’s rental-car needs while delivering the best quality, service and value; we believe the ideal system is a human-centered user-experience modeled according to a consistent object-oriented model from the front-end of the application to the back-end of the database.  Concordantly, we believe both the entity relationship diagram and relational database-model specifications required by the rental-car management inventory system will seek to:

1. Proactively maintain and monitor the database system to prevent any potential issues.
2. Accurately diagnose and forecast the database system health and capacity to maximize return on investment for the rental-car inventory management system and promptly respond to the stakeholder’s support request(s) to remove any work bottleneck on DBA side.
3. Persistently seek innovative ways to improve DBA work efficiently yet effectively.

**CUSTOMER REQUIREMENTS**

1. Car rental agency should have collection of cars
2. Each car should belong to a car category
3. Customer based on car category preference, rents a car
4. Customer will select a car from the suggestions and should be able to reserve it for rent.
5. Car rental rate will be calculated based on the selected make and model
6. Which is the most preferable rental office location for customers.
7. Car model with the highest mileage and reservations

Lets Start.

**INSTALLATION of ORACLE 12C**

**Step 1**

We will start with installing the Oracle which will help us in creating a database in SQL server.

Download it from website : <https://www.oracle.com/database/12c-database/>

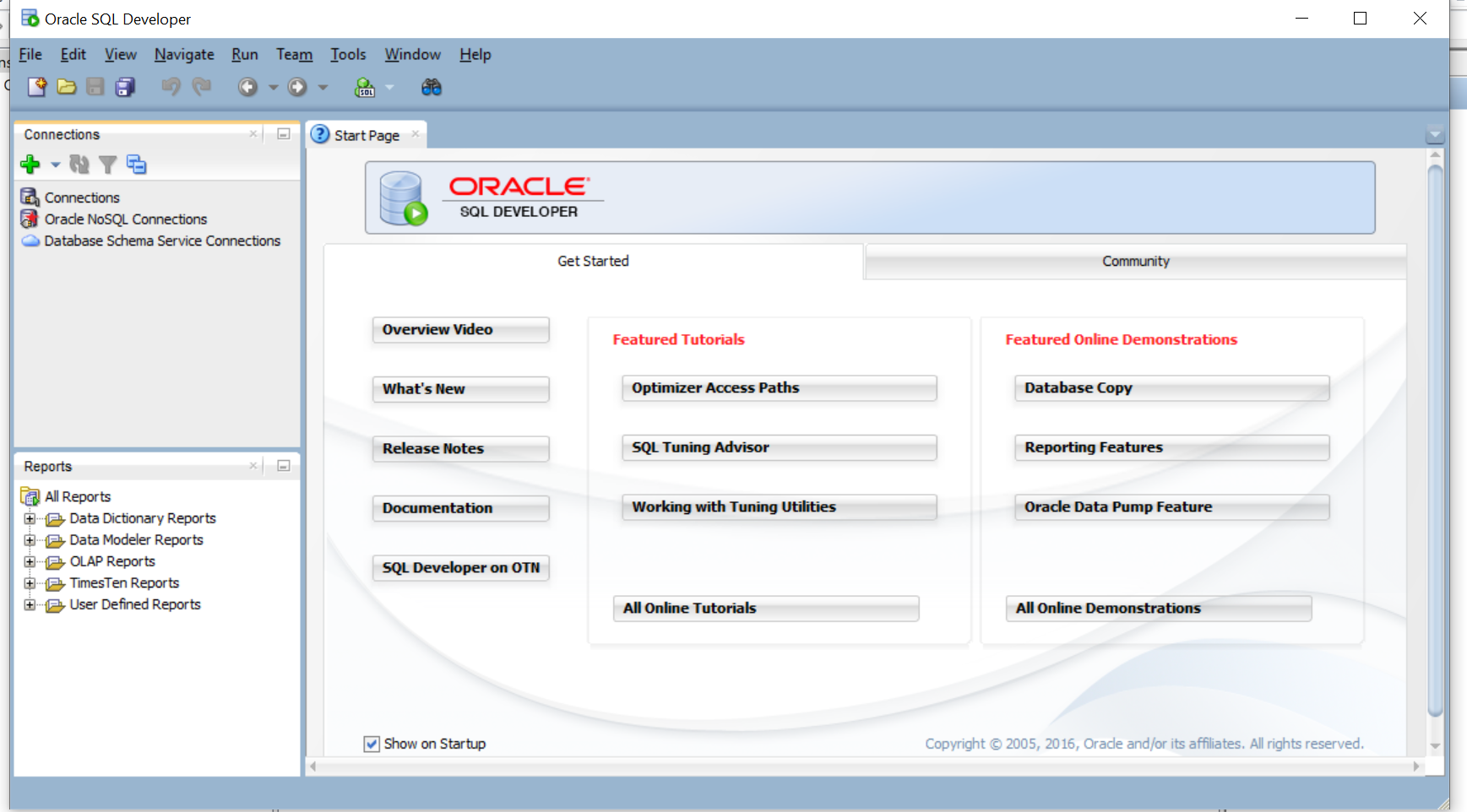
Install ORACLE 12C

Follow steps as when and then required.

After installing we will be able to start developing database and queries in SQL.

Follow the steps in the website above and after installing go to Command Prompt and type “sqlplus / as sysdba”.(  Sysdba have administrative privileges required to perform high-level administrative operations such as creating(users or database or instance for example), starting up, shutting down, backing up, or recovering the database.)

For this project we are using Command Prompt as our developer tool. To learn in deep about the queries and database and its data types.



After Installing we will open the command prompt and will create a USER in SQL Server in which we will create database.

**USER ID : C##abbhamore**

**PASSWORD: teacoffee**

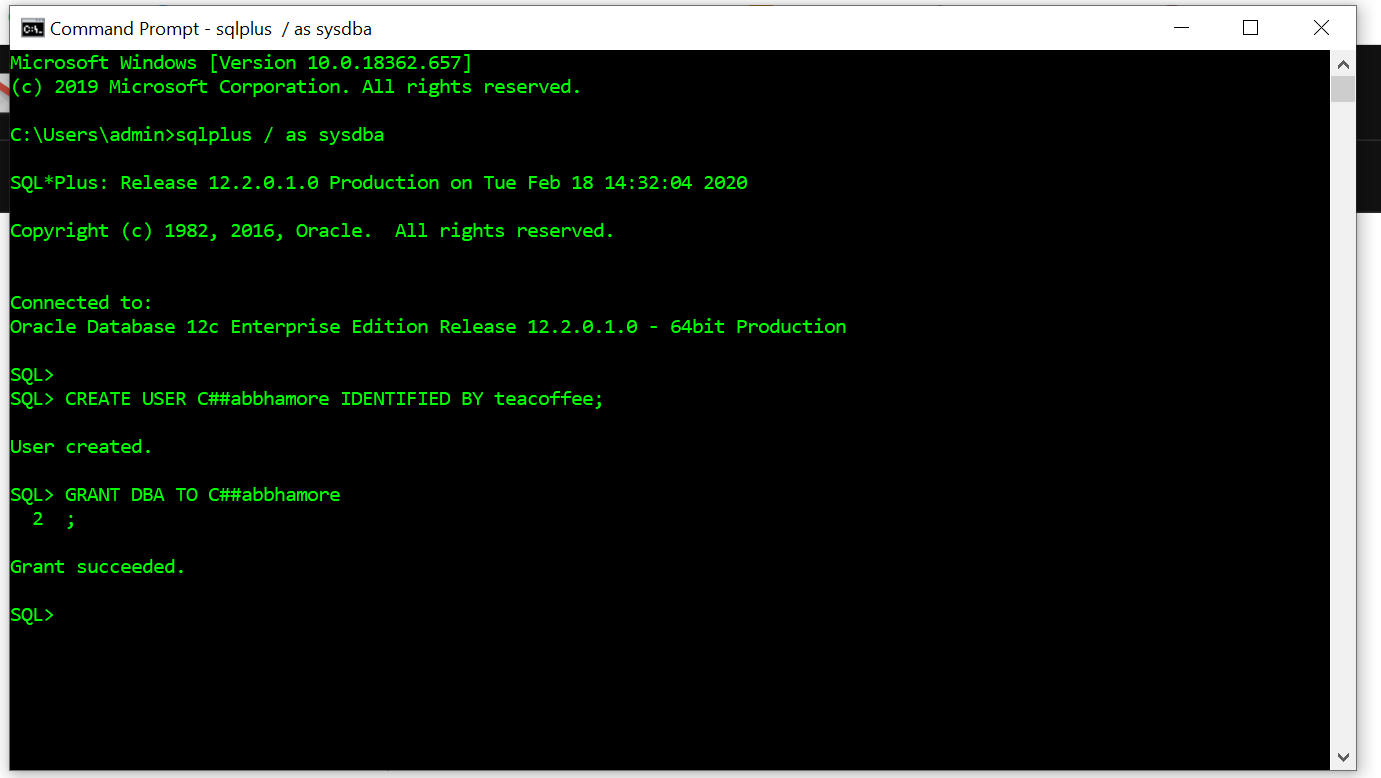
Steps to login to database please refer below screenshot.

A screenshot of a computer

Description automatically generated

Once the user is created we will then grant the DBA and other necessary privileges to the **USER C##abbhamore**

GRANTING DBA All privilieges and roles like insert and delete is granted to **USER = C##abbhamore**



A picture containing indoor, person, wall, sitting

Description automatically generated

A screenshot of a computer

Description automatically generated

Here we have created the DBA and granted the access and privileges to our USER = C##abbhamore

Lets Start coding!! Hurray

# **ENTITIES**

### Customer:

Customer will be the one who is using car rental system for reserving a car. He can be a member of the system or a non-member of the system. Customer entity will store details like customer driving license number, email, address, name, and phone number.

### Vehicle:

Vehicle entity will have list of Vehicle available in the system. Each Vehicle will be associated with a make and model. Vehicle will have attributes like make, model, mileage and license plate, release year and color.

### Employee:

Every person working for wonderland car rental system, position at employee work in the company, employee’s SSN. Their first and last name with their employee ID in the company associated with that employee.

### Reservation

Each car booking will be monitored in the entity called reservation. Reservation will have attributes like reservation ID, which is primary key. Rental office ID, date of reservation, Rate for specific car and customer ID will be including in this entity.

### Rental Office Location

### Every car is rented from specific location and depends on the customer booking car from that location. Office is accompanied with its city name state code and zip code which takes in reservations from customer and assign the preferable car to clients

### Maintenance Log

### This entity will take car of different cars and its maintenance log book. Vehicle requiring the repair and type of procedure followed for each vehicle. The car will have date on which it is repaired and employeeID who took the responsibility to repair it.

# **RELATIONS**

### Vehicle to Reservation :

Every Vehicle is associated with a VIN. VIN and reservation ID is associated with customer who is making reservation for that vehicle. Here it has one to one relation. The relation name is ‘Reserves’.

### Reservation to Accesory:

Once customer . There can be case like booking is cancelled in that case no bill will be associated with the booking. The relation name is ‘Includes.

### Maintenance to Employee:

The employee is responsible for maintiang a vehicle repair who will manage the procedure of vehicle repair so here employee manages the maintenance hence the relation name is ‘manages’.

### Reservation to Customer:

Customer makes reservation for vehicle rental . The relation name is ‘Makes’.

### Employee to Rental Office location

### Employee working at the office location from where the car is booked is mange by the employee table and rental office will keep details of that office. The relation name here is ‘Employs’.

### 

### LOGICAL VIEW OF ENTITY RELATIONSHIP DIAGRAM

# **FUNCTIONAL DEPENDENCIES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ENTITIES** | **Primar Keys** | **Foreign keys** | **Foreign Key 2** | **Unique Keys** |
| **VEHICLE** | VIN | - | - | License Plate |
| **CUSTOMER** | CID | - | - | Driver License |
| **EMPLOYEE** | EID | - | - | SSN |
| **MAINTENANCE LOG** | MID | VIN | EID | - |
| **RENTAL OFFICE ID** | ROID | - | - | - |
| **RESERVATION** | RID | CID | ROID | - |
| **RESERVES** | VIN, RID | VIN | RID | - |
| **INCLUDES** | Sr no, RID | Sr no | RID | - |
| **EMPLOYING** | ROID, EID | ROID | EID | - |
| **ACCESORY** | Serial no | - | - | - |
| **CUSTOMER PHONE NO** | Customer Phone no | CID | - | - |

**Abbreviations:**

|  |  |
| --- | --- |
| **Abbreviation** | **Full Forms** |
| CID | Customer ID |
| VIN | Vehicle identification number |
| EID | Employee ID |
| RID | Reservation ID |
| ROID | Rental office ID |
| Sr no | Serial no |
| MID | Maintenance ID |

# **SQL STATEMENTS**

**Create table Statements**

SQL> CREATE TABLE VEHICLE

(

VIN VARCHAR(30) NOT NULL,

Make VARCHAR(30) NOT NULL,

Model VARCHAR(30) NOT NULL,

Year INT NOT NULL,

Color CHAR(30) NOT NULL,

LicensePlate VARCHAR(20) NOT NULL,

Mileage INT NOT NULL,

PRIMARY KEY (VIN),

UNIQUE (LicensePlate)

);

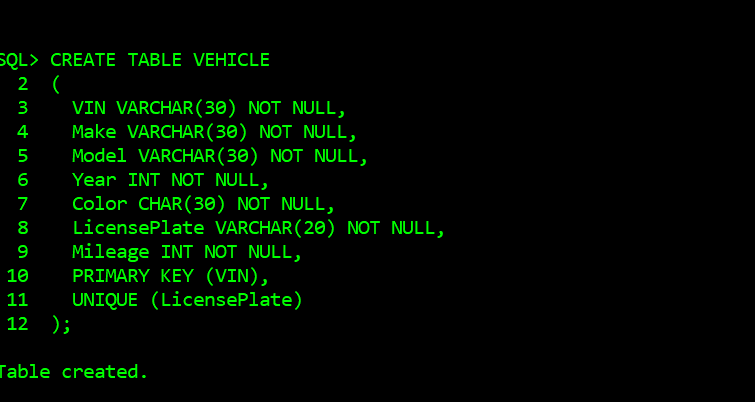


Table created.

SQL> CREATE TABLE EMPLOYEE

(

EmployeeFName VARCHAR(100) NOT NULL,

EmployeeID INT NOT NULL,

EmployeeLName VARCHAR(100) NOT NULL,

Title VARCHAR(100) NOT NULL,

SSN INT NOT NULL,

PRIMARY KEY (EmployeeID),

UNIQUE (SSN)

);

A screenshot of a computer

Description automatically generated

Table created.

SQL> CREATE TABLE ACCESSORY

(

SerialNumber INT NOT NULL,

Type VARCHAR(50) NOT NULL,

Cost DECIMAL(20,2) NOT NULL,

Quantity INT NOT NULL,

PRIMARY KEY (SerialNumber)

);

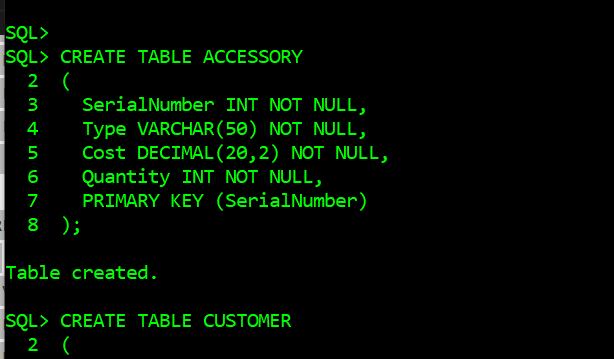


Table created.

SQL> CREATE TABLE CUSTOMER

(

CustomerID INT NOT NULL,

CustomerFName VARCHAR(100) NOT NULL,

CustomerLName VARCHAR(100) NOT NULL,

CustomerEmail VARCHAR(200) NOT NULL,

DriversLicense VARCHAR(50) NOT NULL,

RewardNumber INT,

PRIMARY KEY (CustomerID),

UNIQUE (DriversLicense)

);

A screenshot of a computer

Description automatically generated

Table created.

SQL> CREATE TABLE RENTAL\_OFFICE\_LOCATION

(

RentalOfficeID INT NOT NULL,

StreetAddress VARCHAR(200) NOT NULL,

City VARCHAR(100) NOT NULL,

State VARCHAR(100) NOT NULL,

ZipCode INT NOT NULL,

PRIMARY KEY (RentalOfficeID)

);

A close up of a screen

Description automatically generated

Table created.

SQL> CREATE TABLE MAINTENANCE\_LOG

(

Maintenance\_ID INT NOT NULL,

EmployeeID INT NOT NULL,

VIN VARCHAR(30) NOT NULL,

Maintenance\_Date DATE NOT NULL,

Maintenance\_Procedure VARCHAR(200) NOT NULL,

PRIMARY KEY (Maintenance\_ID),

FOREIGN KEY (VIN) REFERENCES VEHICLE(VIN),

FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE(EmployeeID)

);

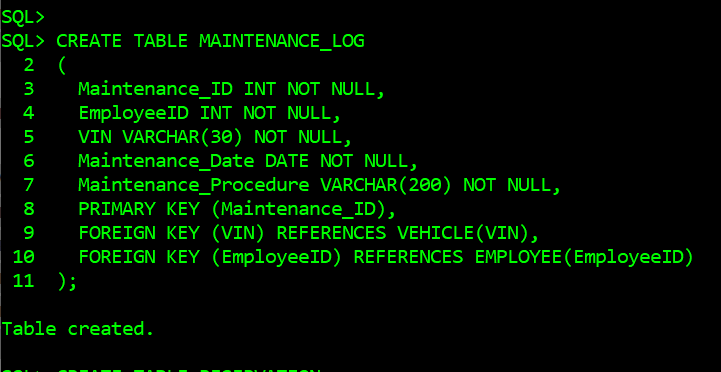


Table created.

SQL> CREATE TABLE RESERVATION

(

ReservationID INT NOT NULL,

Pick\_Up\_Date DATE NOT NULL,

Return\_Date DATE NOT NULL,

Daily\_Rate DECIMAL(20,2) NOT NULL,

Total DECIMAL(20,2) NOT NULL,

CustomerID INT NOT NULL,

RentalOfficeID INT NOT NULL,

PRIMARY KEY (ReservationID),

FOREIGN KEY (CustomerID) REFERENCES CUSTOMER(CustomerID),

FOREIGN KEY (RentalOfficeID) REFERENCES RENTAL\_OFFICE\_LOCATION(RentalOfficeID)

);

A close up of a logo

Description automatically generated

Table created.

SQL>

SQL> CREATE TABLE EMPLOYING

(

RentalOfficeID INT NOT NULL,

EmployeeID INT NOT NULL,

PRIMARY KEY (RentalOfficeID, EmployeeID),

FOREIGN KEY (RentalOfficeID) REFERENCES RENTAL\_OFFICE\_LOCATION(RentalOfficeID),

FOREIGN KEY (EmployeeID) REFERENCES EMPLOYEE(EmployeeID)

);

A close up of a logo

Description automatically generated

Table created.

SQL> CREATE TABLE RESERVES

(

VIN VARCHAR(30) NOT NULL,

ReservationID INT NOT NULL,

PRIMARY KEY (VIN, ReservationID),

FOREIGN KEY (VIN) REFERENCES VEHICLE(VIN),

FOREIGN KEY (ReservationID) REFERENCES RESERVATION(ReservationID)

);

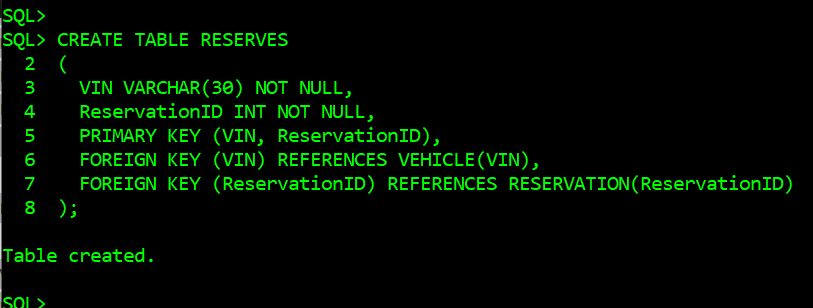


Table created.

SQL> CREATE TABLE INCLUDES

(

SerialNumber INT NOT NULL,

ReservationID INT NOT NULL,

PRIMARY KEY (SerialNumber, ReservationID),

FOREIGN KEY (SerialNumber) REFERENCES ACCESSORY(SerialNumber),

FOREIGN KEY (ReservationID) REFERENCES RESERVATION(ReservationID)

);

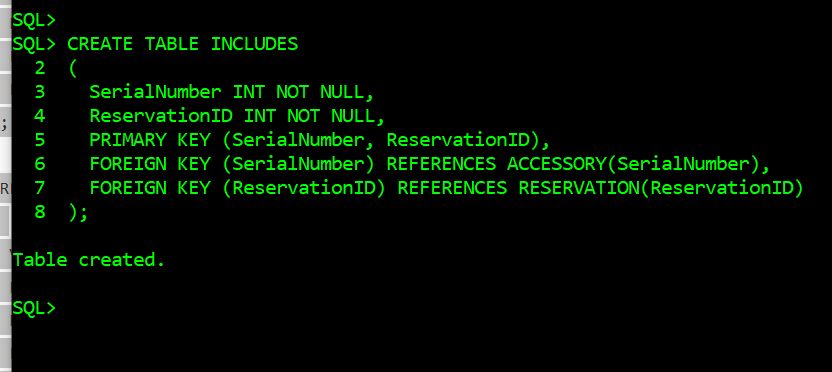


Table created.

**Insert SQL Statements**

**INSERT INTO CUSTOMER**

INSERT INTO CUSTOMER VALUES (000001, 'Alia', 'Bhat', 'bhatalia@jld.org', 999990011, 0000011);

INSERT INTO CUSTOMER VALUES (000007, 'anshul', 'bhamore', 'bhamoreanshul@jld.org', '999990017', '0000018');

INSERT INTO CUSTOMER VALUES (000002, 'priyanka', 'chopra', 'choprapriyanka@jld.org', '999990012', '0000012');

INSERT INTO CUSTOMER VALUES (000003, 'sharukh', 'khan', 'khansharukh@jld.org', '999990013', '0000013');

INSERT INTO CUSTOMER VALUES (000004, 'salman', 'khan', 'khansalman@jld.org', '999990014', '0000014');

INSERT INTO CUSTOMER VALUES (000005, 'aamir', 'khan', 'khanaamir@jld.org', '999990015', '0000015');

INSERT INTO CUSTOMER VALUES (000006, 'anchal', 'bhamore’, 'bhamoreanchal@jld.org', '999990016', '0000016')

INSERT INTO CUSTOMER VALUES (000008, 'ankur', 'bhamore', 'bhamoreankur@jld.org', '999990018', '0000018');

INSERT INTO CUSTOMER VALUES (000009, 'rajkishore', 'bhamore', 'bhamorerajkishore@jld.org', '999990019', '0000019');

INSERT INTO CUSTOMER VALUES (000010, 'devang', 'pawar', 'pawardevang@jld.org', '999990020', '0000020');

INSERT INTO CUSTOMER VALUES (000011, 'vicky', 'pardeshi', 'pardeshivicky@jld.org', '999990021', '0000021');

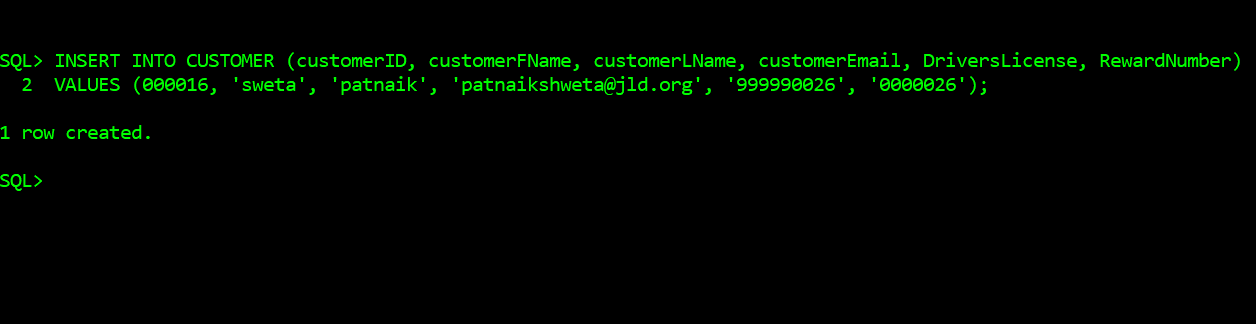
INSERT INTO CUSTOMER VALUES (000012, 'rohit', 'kargave', 'kargaverohit@jld.org', '999990022', '0000022');

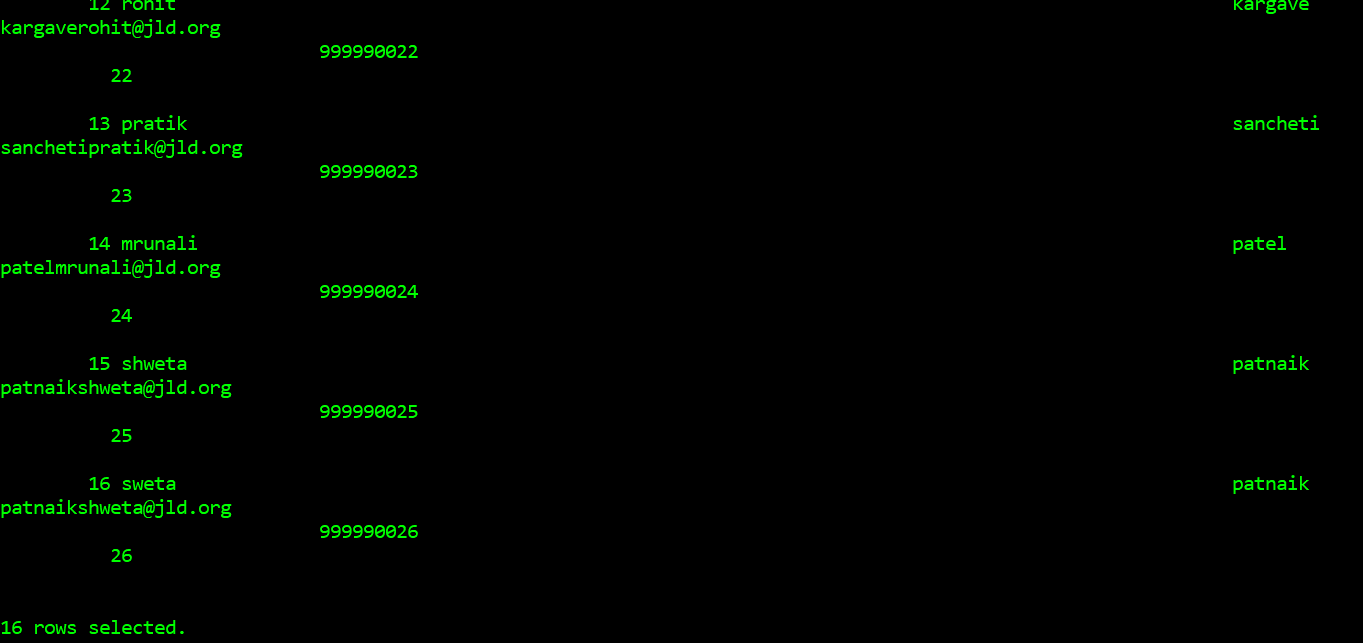
INSERT INTO CUSTOMER VALUES (000013, 'pratik', 'sancheti', 'sanchetipratik@jld.org', '999990023', '0000023');

INSERT INTO CUSTOMER VALUES (000014, 'mrunali', 'patel', 'patelmrunali@jld.org', '999990024', '0000024');

INSERT INTO CUSTOMER VALUES (000015, 'shweta', 'patnaik', 'patnaikshweta@jld.org', '999990025', '0000025');

INSERT INTO CUSTOMER VALUES (000016, 'sweta', 'patnaik', 'patnaikshweta@jld.org', '999990026', '0000026');





I**NSERT INTO RENTAL\_OFFICE\_LOCATION**

INSERT INTO RENTAL\_OFFICE\_LOCATION VALUES ( 01, '123 east street', 'cleveland', 'OH', '44114') ;

INSERT INTO RENTAL\_OFFICE\_LOCATION VALUES ( 02, '124 east street', 'cleveland', 'OH', '44114') ;

INSERT INTO RENTAL\_OFFICE\_LOCATION VALUES ( 03, '125 east street', 'cleveland', 'OH', '44114') ;

INSERT INTO RENTAL\_OFFICE\_LOCATION VALUES ( 03 , '126 east street', 'cleveland', 'OH', '44114');

INSERT INTO RENTAL\_OFFICE\_LOCATION VALUES ( 04, '127 east street', 'cleveland', 'OH', '44114') ;

INSERT INTO RENTAL\_OFFICE\_LOCATION VALUES ( 05, '128 east street', 'cleveland', 'OH', '44114');

INSERT INTO RENTAL\_OFFICE\_LOCATION VALUES ( 05, '129 east street', 'cleveland', 'OH', '44114') ;

INSERT INTO RENTAL\_OFFICE\_LOCATION VALUES ( 06, '130 east street', 'cleveland', 'OH', '44114');

INSERT INTO RENTAL\_OFFICE\_LOCATION VALUES ( 07, '131 east street', 'cleveland', 'OH', '44114') ;

INSERT INTO RENTAL\_OFFICE\_LOCATION VALUES ( 08, '132 east street', 'cleveland', 'OH', '44114');

INSERT INTO RENTAL\_OFFICE\_LOCATION VALUES ( 09, '133 east street', 'cleveland', 'OH', '44114') ;

INSERT INTO RENTAL\_OFFICE\_LOCATIONVALUES ( 11, '135 east street', 'cleveland', 'OH', '44114') ;

INSERT INTO RENTAL\_OFFICE\_LOCATION VALUES ( 12, '136 east street', 'cleveland', 'OH', '44114');

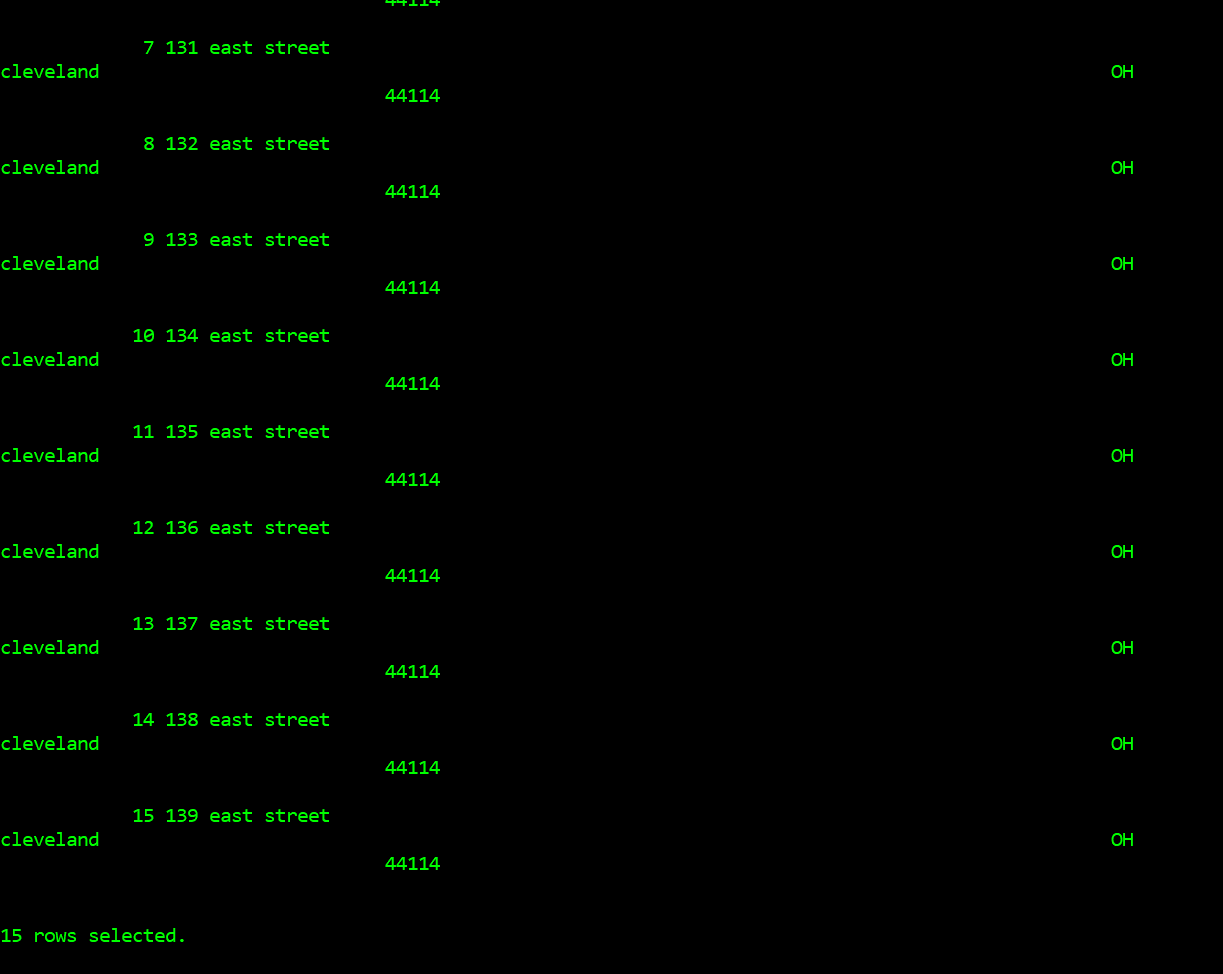
INSERT INTO RENTAL\_OFFICE\_LOCATION VALUES ( 13, '137 east street', 'cleveland', 'OH', '44114') ;

INSERT INTO RENTAL\_OFFICE\_LOCATION VALUES ( 14, '138 east street', 'cleveland', 'OH', '44114');

INSERT INTO RENTAL\_OFFICE\_LOCATION VALUES ( 15, '139 east street', 'cleveland', 'OH', '44114') ;

A hand holding a cell phone screen with text

Description automatically generated



**INSERT INTO the EMPLOYEE table**

INSERT INTO EMPLOYEE VALUES (11, 'Victor', 'Stone', 'Office Manager', '999990006');

INSERT INTO EMPLOYEE VALUES (12, 'Arthur', 'Curry', 'Office Manager', '999990007');

INSERT INTO EMPLOYEE VALUES (13, 'Kara', 'Danvers', 'Office Manager', '999990008');

INSERT INTO EMPLOYEE VALUES (14, 'Pamela', 'Isley', 'Office Manager', '999990009');

INSERT INTO EMPLOYEE VALUES (15, 'Barbara', 'Gordon', 'Customer Care Specialist', '999990010');

INSERT INTO EMPLOYEE VALUES (16, 'Harleen', 'Quinzel', 'Customer Care Specialist', '999990011');

INSERT INTO EMPLOYEE VALUES (17, 'John', 'Jones', 'Customer Care Specialist', '999990012');

INSERT INTO EMPLOYEE VALUES (18, 'Oliver', 'Queen', 'Customer Care Specialist', '999990014');

INSERT INTO EMPLOYEE VALUES (20, 'Clark', 'Kent', 'Mechanic', '999990016');

INSERT INTO EMPLOYEE VALUES (21, 'Lex', 'Luthor', 'Mechanic', '999990017');

INSERT INTO EMPLOYEE VALUES (22, 'alex', 'Luthor', 'Mechanic', '999990017');

INSERT INTO EMPLOYEE VALUES (23, 'Lexa', 'Luthor', 'analyst', '999990018');

INSERT INTO EMPLOYEE VALUES (24, 'Lexo', 'Luthoron', 'analyst', '999990019');

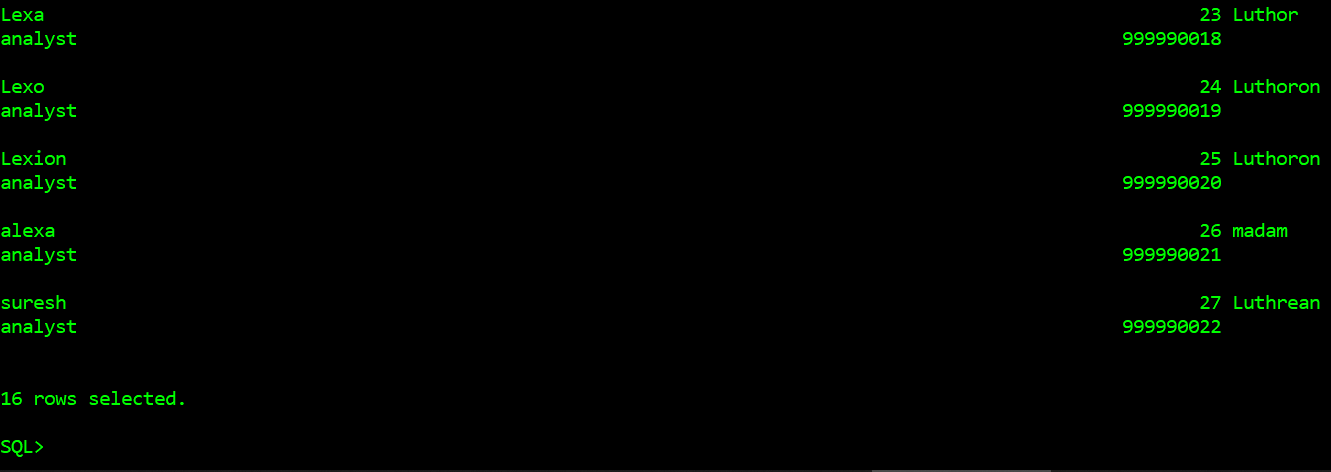
INSERT INTO EMPLOYEE VALUES (25, 'Lexion', 'Luthoron', 'analyst', '999990020');

INSERT INTO EMPLOYEE VALUES (26, 'alexa', 'madam', 'analyst', '999990021');

INSERT INTO EMPLOYEE VALUES (27, 'suresh', 'Luthrean', 'analyst', '999990022');

A close up of a screen

Description automatically generated



**INSERT INTO VEHICLE TABLE**

INSERT INTO VEHICLE VALUES ('55555LLLLL', 'Kia', 'soul', '2011', 'Red', 'EE44FF99', '350');

INSERT INTO VEHICLE VALUES ('55555MMMMM', 'Ford', 'Fusion', '2012', 'Silver', 'EE66FF99', '316');

INSERT INTO VEHICLE VALUES ('55555NNNNN', 'Ford', 'Escape', '2015', 'Black', 'EE77FF99', '308');

INSERT INTO VEHICLE VALUES ('55555OOOOO', 'Ford', 'Explorer', '2016', 'Grey', 'EE88FF99', '298');

INSERT INTO VEHICLE VALUES ('11111AAAAA', 'Honda', 'Accord', '2014', 'Gold', 'AA11BB22', '27365');

INSERT INTO VEHICLE VALUES ('33333CCCCC', 'Chevy', 'Tahoe', '2012', 'Black', 'CC33DD44', '19999');

INSERT INTO VEHICLE VALUES ('44444DDDDD', 'Audi', 'A4', '2017', 'Silver', 'DD44EE55', '567');

INSERT INTO VEHICLE VALUES ('55555EEEEE', 'Dodge', 'Charger', '2015', 'Purple', 'EE55FF66', '302');

INSERT INTO VEHICLE VALUES ('66666FFFFF', 'Dodge', 'Challenger', '2015', 'Purple', 'EE55FF77', '264');

INSERT INTO VEHICLE VALUES ('55555GGGGG', 'Honda', 'Accord', '2015', 'Black', 'EE55FF88', '328');

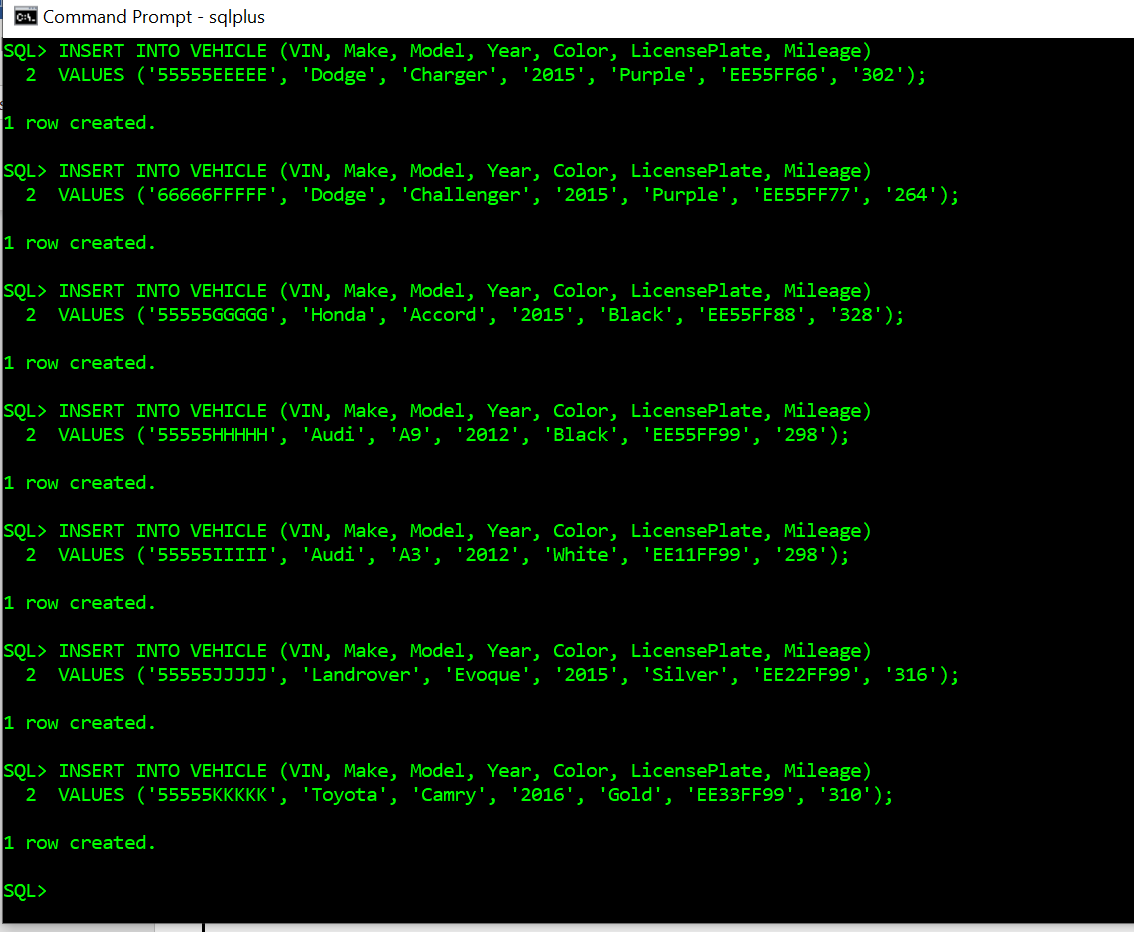
INSERT INTO VEHICLE VALUES ('55555HHHHH', 'Audi', 'A9', '2012', 'Black', 'EE55FF99', '298');

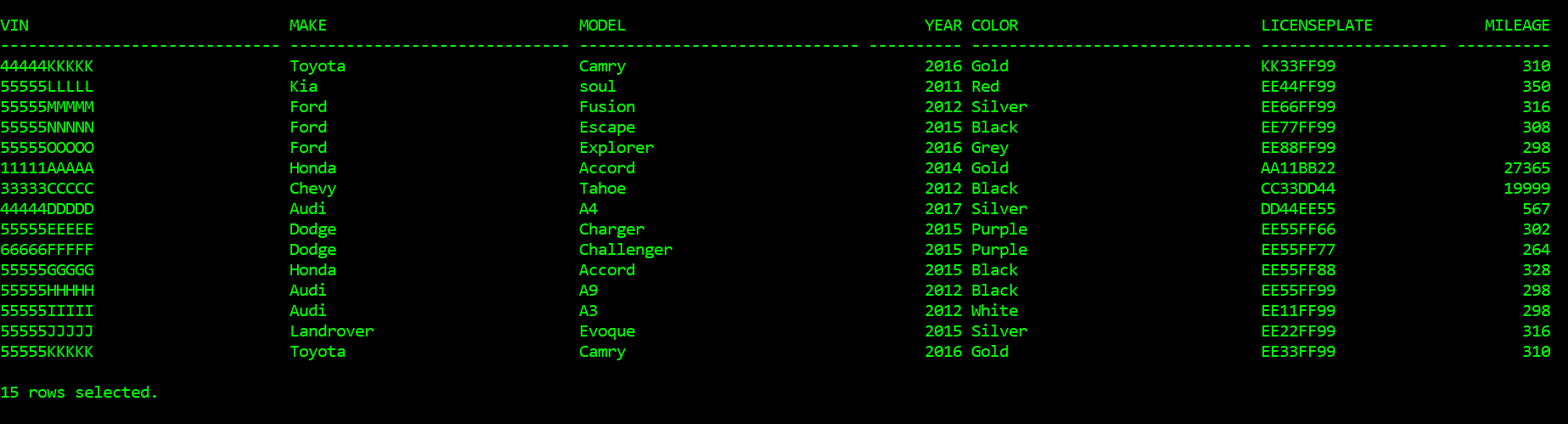
INSERT INTO VEHICLE VALUES ('55555IIIII', 'Audi', 'A3', '2012', 'White', 'EE11FF99', '298');

INSERT INTO VEHICLE VALUES ('55555JJJJJ', 'Landrover', 'Evoque', '2015', 'Silver', 'EE22FF99', '316');

INSERT INTO VEHICLE VALUES ('55555KKKKK', 'Toyota', 'Camry', '2016', 'Gold', 'EE33FF99', '310');

INSERT INTO VEHICLE VALUES ('44444KKKKK', 'Toyota', 'Camry', '2016', 'Gold', 'KK33FF99', '310');





**INSERT ACCESORY TABLE**

INSERT INTO ACCESSORY VALUES (12212, 'GPS', '25', '1');

INSERT INTO ACCESSORY VALUES (12321, 'GPS', '30', '1');

INSERT INTO ACCESSORY VALUES (12334, 'GPS', '25', '1');

INSERT INTO ACCESSORY VALUES (12355, 'GPS', '25', '1');

INSERT INTO ACCESSORY VALUES (12423, 'GPS', '31', '1');

INSERT INTO ACCESSORY VALUES (12234, 'GPS', '25', '1');

INSERT INTO ACCESSORY VALUES (19234, 'GPS', '25', '1');

INSERT INTO ACCESSORY VALUES (18234, 'GPS', '25', '1');

INSERT INTO ACCESSORY VALUES (17234, 'GPS', '30', '1');

INSERT INTO ACCESSORY VALUES (16234, 'GPS', '31', '1');

INSERT INTO ACCESSORY VALUES (15234, 'GPS', '30', '1');

INSERT INTO ACCESSORY VALUES (14234, 'GPS', '31', '1');

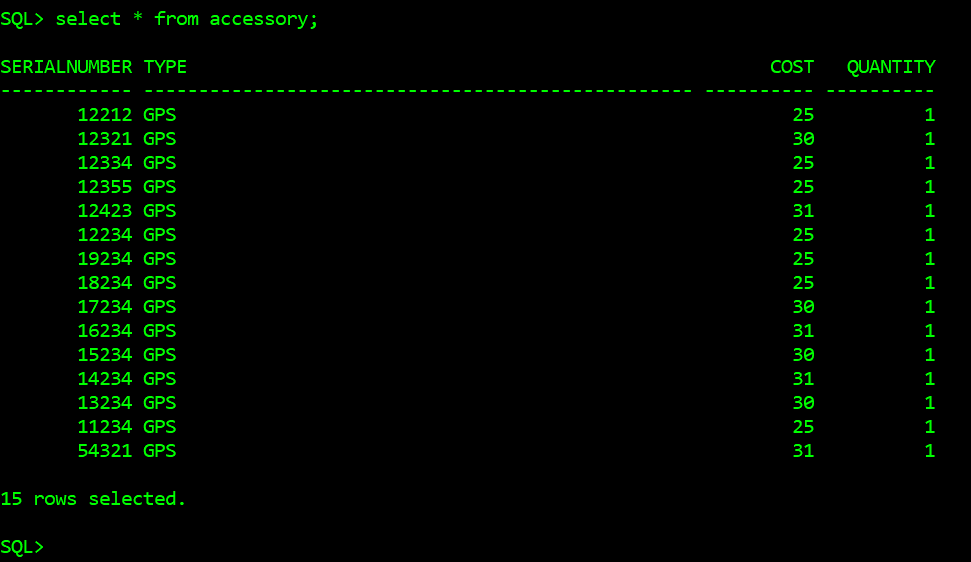
INSERT INTO ACCESSORY VALUES (13234, 'GPS', '30', '1');

INSERT INTO ACCESSORY VALUES (11234, 'GPS', '25', '1');

INSERT INTO ACCESSORY VALUES (54321, 'GPS', '31', '1');

A screenshot of a cell phone

Description automatically generated



**INSERT INTO RESERVATION TABLE**

INSERT INTO RESERVATION (ReservationID, Pick\_Up\_Date, Return\_Date, Daily\_Rate, Total, CustomerID, RentalOfficeID)

INSERT INTO RESERVATION VALUES (07, '06jun20', '13jun20' , 125.00, 875.00, 000001, 01);

INSERT INTO RESERVATION VALUES (08, '07mar20', '14mar20', 125.00, 875.00, 000002, 04);

INSERT INTO RESERVATION VALUES (09, '05apr20', '15apr20', 125.00, 1250.00, 000003, 05);

INSERT INTO RESERVATION VALUES (10, '05jun20', '15jun20', 125.00, 1250.00, 000004, 01);

INSERT INTO RESERVATION VALUES (11, '21jun20', '23jun20', 125.00, 250.00, 000005, 08);

INSERT INTO RESERVATION VALUES (12, '01mar20', '08mar20', 95.00, 665.00, 000006, 09);

INSERT INTO RESERVATION VALUES (13, '10jun20', '12jun20', 125.00, 250.00, 000007, 04);

INSERT INTO RESERVATION VALUES (14, '05apr20', '07apr20', 80.00, 160.00, 000008, 08);

INSERT INTO RESERVATION VALUES (15, '19jun20', '21jun20', 150.00, 300.00, 000009, 09);

INSERT INTO RESERVATION VALUES (16, '22jul20', '27jul20', 115.00, 575.00, 000010, 09);

INSERT INTO RESERVATION VALUES (17, '19aug20', '23aug20', 95.00, 380.00, 000011, 03);

INSERT INTO RESERVATION VALUES (18, '19aug20', '21aug20', 95.00, 190.00, 000012, 02);

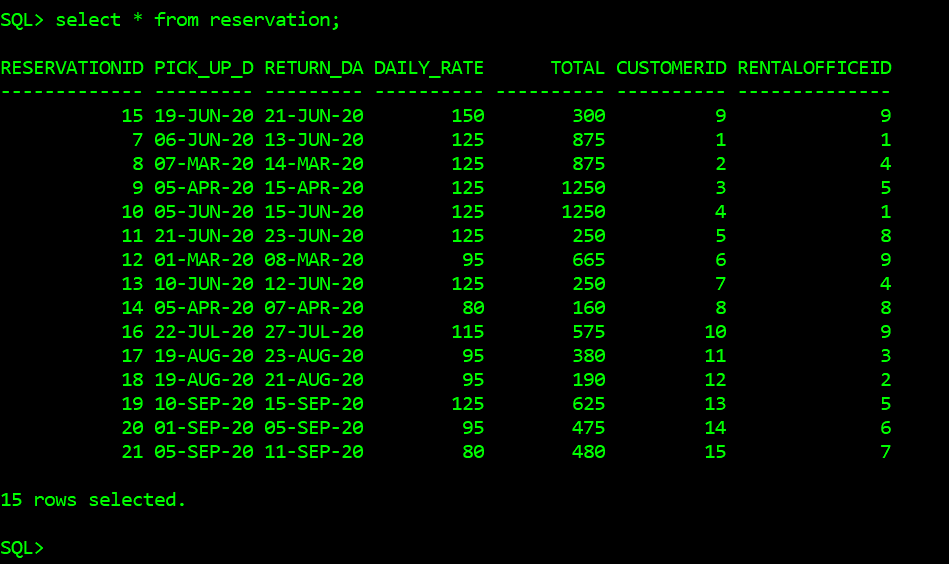
INSERT INTO RESERVATION VALUES (19, '10sep20', '15sep20', 125.00, 625.00, 000013, 05);

INSERT INTO RESERVATION VALUES (20, '01sep20', '05sep20', 95.00, 475.00, 000014, 06);

INSERT INTO RESERVATION VALUES (21, '05sep20', '11sep20', 80.00, 480.00, 000015, 07);

A screen shot of a smart phone

Description automatically generated



**INSERT INTO EMPLOYING TABLE**

INSERT INTO EMPLOYING VALUES ('01', '11');

INSERT INTO EMPLOYING VALUES ('02', '12');

INSERT INTO EMPLOYING VALUES ('03', '13');

INSERT INTO EMPLOYING VALUES ('04', '14');

INSERT INTO EMPLOYING VALUES ('06', '16');

INSERT INTO EMPLOYING VALUES ('07', '17');

INSERT INTO EMPLOYING VALUES ('08', '18');

INSERT INTO EMPLOYING VALUES ('09', '19');

INSERT INTO EMPLOYING VALUES ('01', '20');

INSERT INTO EMPLOYING VALUES ('02', '21');

INSERT INTO EMPLOYING VALUES ('02', '22');

INSERT INTO EMPLOYING VALUES ('04', '23');

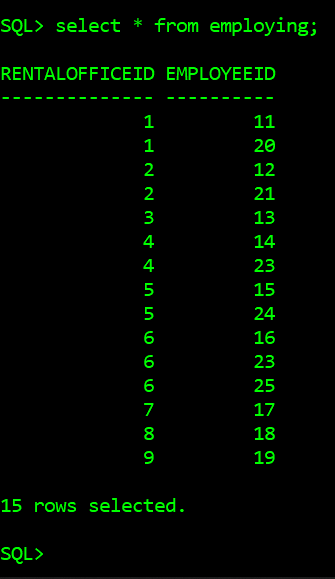
INSERT INTO EMPLOYING VALUES ('05', '24');

INSERT INTO EMPLOYING VALUES ('06', '25');

INSERT INTO EMPLOYING VALUES ('06', '23);

A screenshot of a cell phone

Description automatically generated



**INSERT INTO TABLE MAINTENANCE\_LOG**

INSERT INTO MAINTENANCE\_LOG VALUES (77, 12, '55555LLLLL', '07jan20', 'Oil Change');

INSERT INTO MAINTENANCE\_LOG VALUES (78, 18, '55555MMMMM', '07jan20', 'Tire Replacement');

INSERT INTO MAINTENANCE\_LOG VALUES (79, 16, '55555NNNNN', '07feb20', 'Annual Maintenance');

INSERT INTO MAINTENANCE\_LOG VALUES (80, 25, '55555OOOOO', '07jun20', 'Alignment Check');

INSERT INTO MAINTENANCE\_LOG VALUES (81, 12, '11111AAAAA', '07feb20', 'Alignment Check');

INSERT INTO MAINTENANCE\_LOG VALUES (82, 24, '33333CCCCC', '07mar20', 'Alignment Check');

INSERT INTO MAINTENANCE\_LOG VALUES (83, 15, '44444DDDDD', '07feb20', 'Alignment Check');

INSERT INTO MAINTENANCE\_LOG VALUES (84, 13, '55555EEEEE', '07mar20', 'Alignment Check');

INSERT INTO MAINTENANCE\_LOG VALUES (85, 24, '66666FFFFF', '07dec20', 'Oil Change');

INSERT INTO MAINTENANCE\_LOG VALUES (86, 20, '55555GGGGG', '07dec20', 'Oil Change');

INSERT INTO MAINTENANCE\_LOG VALUES (87, 21, '55555HHHHH', '07feb20', 'Oil Change');

INSERT INTO MAINTENANCE\_LOG VALUES (88, 14, '55555IIIII', '07jul20', 'Tire Replacement');

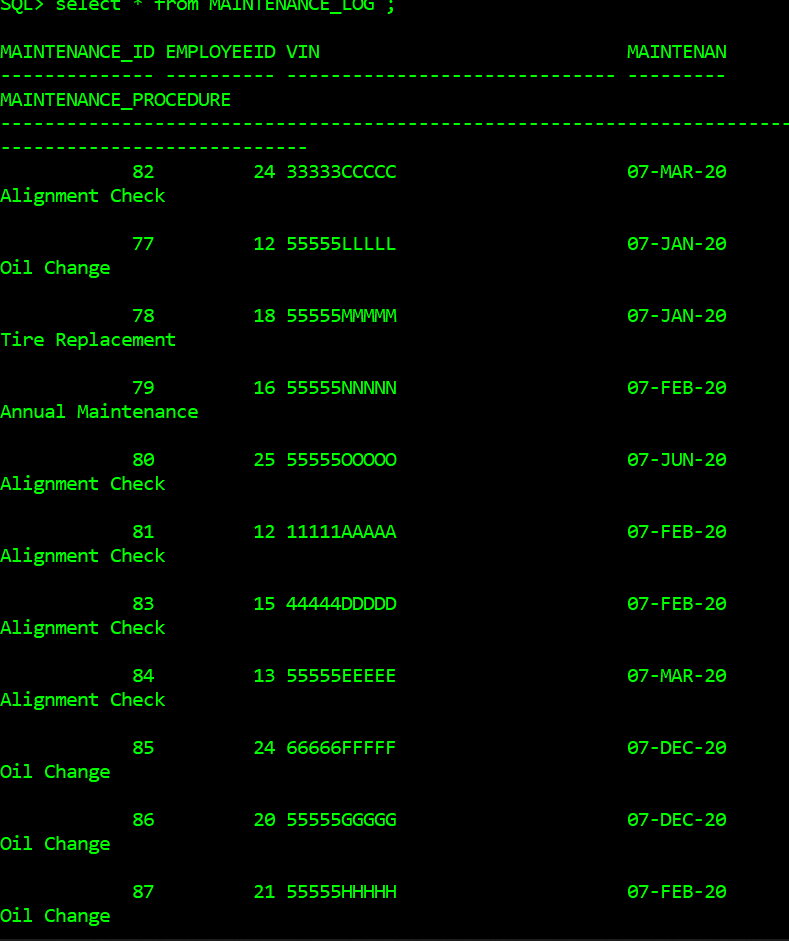
INSERT INTO MAINTENANCE\_LOG VALUES (89, 14, '55555JJJJJ', '07nov20', 'Oil Change');

INSERT INTO MAINTENANCE\_LOG VALUES (90, 17, '55555KKKKK', '07nov20', 'Tire Replacement');

INSERT INTO MAINTENANCE\_LOG VALUES (91, 18, '11111AAAAA', '07feb20', 'Oil Change');

A picture containing indoor

Description automatically generated



**INSERT INTO RESERVES**

INSERT INTO RESERVES VALUES ('11111AAAAA', 07);

INSERT INTO RESERVES VALUES ('55555KKKKK', 08);

INSERT INTO RESERVES VALUES ('55555JJJJJ', 09);

INSERT INTO RESERVES VALUES ('55555IIIII', 10);

INSERT INTO RESERVES VALUES ('55555HHHHH', 11);

INSERT INTO RESERVES VALUES ('55555GGGGG', 12);

INSERT INTO RESERVES VALUES ('55555FFFFF, 13);

INSERT INTO RESERVES VALUES ('55555EEEEE, 14);

INSERT INTO RESERVES VALUES ('55555DDDDD', 15);

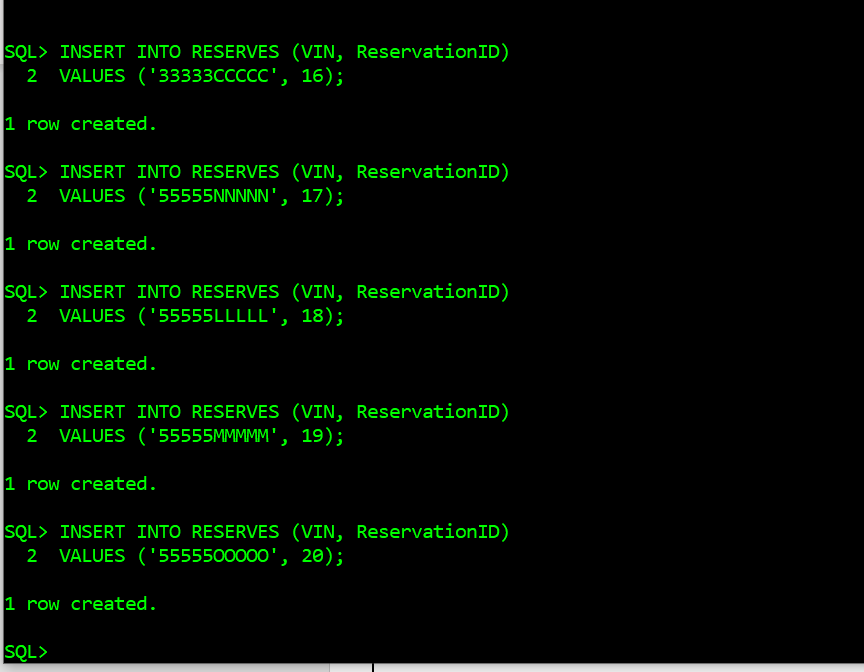
INSERT INTO RESERVES VALUES ('33333CCCCC', 16);

INSERT INTO RESERVES VALUES ('55555NNNNN', 17);

INSERT INTO RESERVES VALUES ('55555LLLLL', 18);

INSERT INTO RESERVES VALUES ('55555MMMMM', 19);

INSERT INTO RESERVES VALUES ('55555OOOOO', 20);



**INSERT INTO INCLUDES**

INSERT INTO INCLUDES VALUES (12212, 07);

INSERT INTO INCLUDES VALUES (12321, 08);

INSERT INTO INCLUDES VALUES (12334, 09);

INSERT INTO INCLUDES VALUES (12355, 10);

INSERT INTO INCLUDES VALUES (12423, 11);

INSERT INTO INCLUDES VALUES (12234, 12);

INSERT INTO INCLUDES VALUES (19234, 13);

INSERT INTO INCLUDES VALUES (18234, 14);

INSERT INTO INCLUDES VALUES (16232, 15);

INSERT INTO INCLUDES VALUES (16234, 16);

INSERT INTO INCLUDES VALUES (15234, 17);

INSERT INTO INCLUDES VALUES (14234, 18);

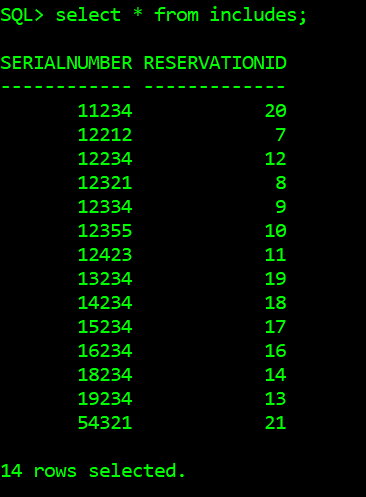
INSERT INTO INCLUDES VALUES (13234, 19);

INSERT INTO INCLUDES VALUES (11234, 20);

INSERT INTO INCLUDES VALUES (54321, 21);

A picture containing sitting, indoor

Description automatically generated



**Here all the tables have been created in a database, entered entities and its attributes with all the data. In futher steps we will run queries relation to different set of operations, function and select statements.**

**Some more queries on FROM,ORDER BY, WHERE and ON clause.Subqueries will be relation to the client’s requirement. So let’s start with the SQL Queries in Command prompt.**

**SQL QUERIES**

1. This query will count all records within the model column using the count-function returning the number of records within a table grouped by MODEL from the VEHICLE-table, which will then be displayed as an alias-table entitled, “Highest mileage”.

**SYNTAX**

**SELECT** MODEL, COUNT (\*) AS "HIGHEST MILEAGE"

**FROM** VEHICLE

**WHERE** MILEAGE > 300

**GROUP** **BY** MODEL;

A screenshot of a cell phone

Description automatically generated

1. Query to find the oldest reservation done for any car is by using the select minimum function where pick\_up\_date is the earliest date among all of them.

**SYNTAX**

**SELECT** \* **FROM** RESERVATION

**WHERE** Pick\_Up\_Date = (SELECT MIN(Pick\_Up\_Date) FROM RESERVATION);

A picture containing wall, person

Description automatically generated

1. This query will count all records withing the daily rates column using between function returning the Daily\_rate between 100-120 within a table

**SYNTAX**

**SELECT** \* **FROM** RESERVATION

**WHERE** DAILY\_RATE BETWEEN 100 AND 120;

A picture containing wall, object, green

Description automatically generated

1. **SEQUENCE CREATING**

To create a sequence on the database we will set the minimum and maximum values of the car rental rate and will keep it increasing after every quarter by 20 where the value starts with 100 and cache will not be more than 10 to save space.

**SYNTAX**

**CREATE** SEQUENCE CAR\_RENTAL\_RATE

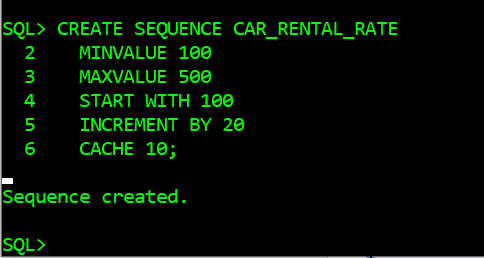
**MINVALUE** 100

**MAXVALUE** 500

**START** WITH 100

**INCREMENT** BY 20

**CACHE** 10;



SQUENCE CREATION

1. **SEQUENCE ALTERATION**

Question : If the car rate starts with 100 and you want it to get 223 as your next car rate. We need to alter the sequence increment rate by writing the below command. We would execute the following command :

**SYNTAX**

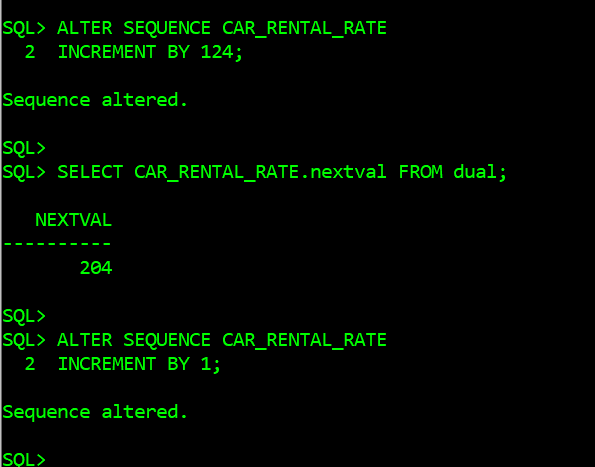
**ALTER** SEQUENCE CAR\_RENTAL\_RATE

**INCREMENT** BY 19;

**SELECT** CAR\_RENTAL\_RATE.nextval FROM dual;

**ALTER** SEQUENCE CAR\_RENTAL\_RATE

**INCREMENT** BY 1;



1. This query will display the total-revenue collected per state by utiliziing an implicit-join format in addition to alias tables to effectively display the total-revenue collected by state corresponding to the "RentalOfficeID" column contained within the two separate tables, joining or "zipping-up" the data separated between the tables.

**SYNTAX**

**SELECT** STATE, Total

**FROM** RENTAL\_OFFICE\_LOCATION r,

RESERVATION i

**WHERE** r.RentalOfficeID=i.RentalOfficeID

**ORDER** **BY** Total;

A screenshot of a cell phone

Description automatically generated

**ADVANCED SQL QUERIES**

1. **INNER JOIN**

Customer making reseravation of vehicles with their make and model is join in a table with the help of Inner join function. This query is advance query for operator like inner join.

**SYNTAX**

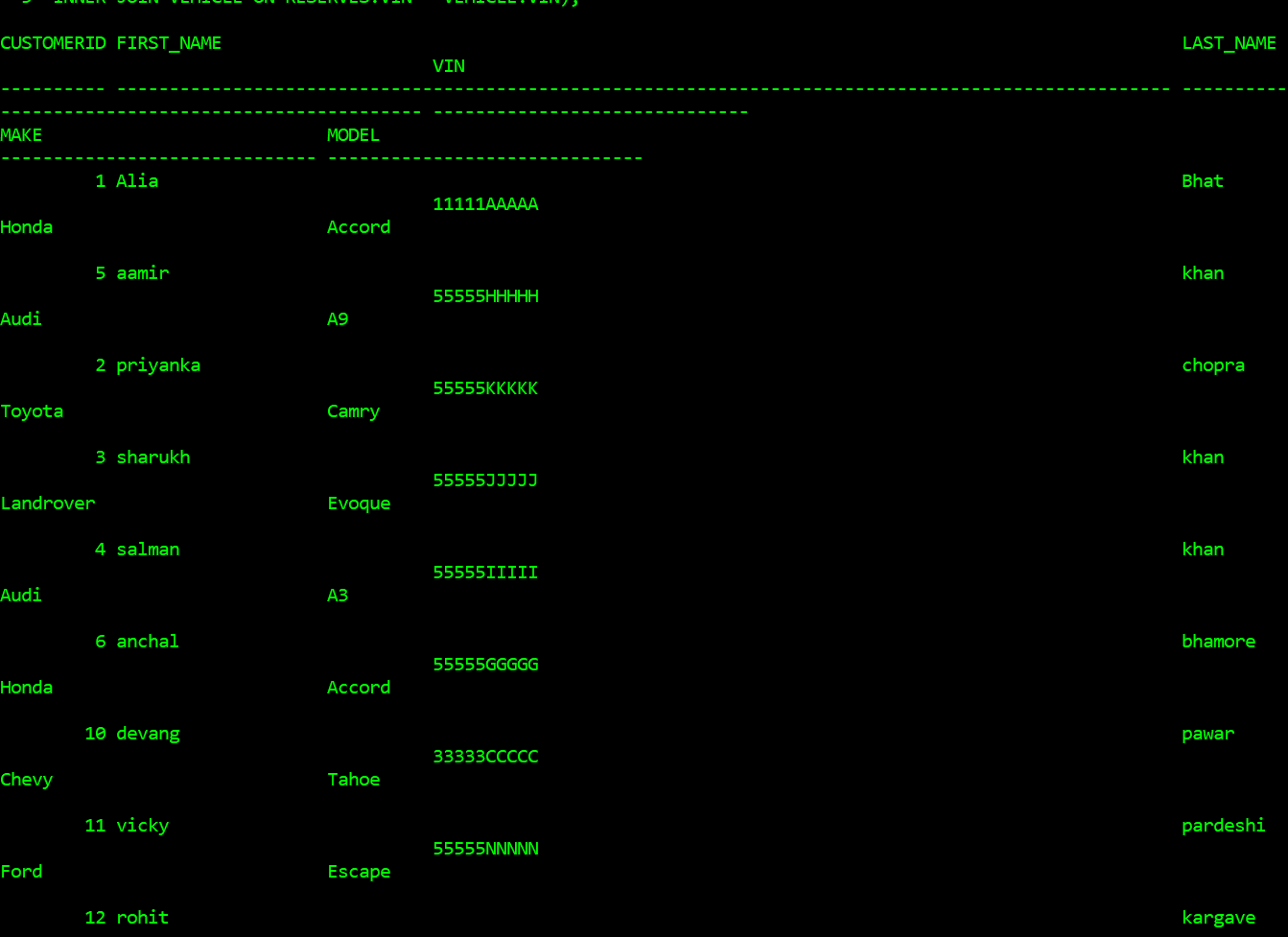
**SELECT** CUSTOMER.CustomerID, CUSTOMER.CustomerFname **AS** First\_Name, CUSTOMER.CustomerLname **AS** Last\_Name, RESERVES.VIN, VEHICLE.Make, VEHICLE.Model

**FROM** (((CUSTOMER

**INNER** **JOIN** RESERVATION **ON** CUSTOMER.CustomerID **=** RESERVATION.CustomerID)

**INNER** **JOIN** RESERVES **ON** RESERVATION.ReservationID **=** RESERVES.ReservationID)

**INNER** **JOIN** VEHICLE **ON** RESERVES.VIN **=** VEHICLE.VIN);



1. **Sum of Rental Office Reservation by Sales Descending**

To find the total number of sale done by the rental office in descending order, where we will use SUM and INNER JOIN.

ON below query will be like :

**SYNTAX**

**SELECT** SUM(RESERVATION.Total) **AS** Total\_Sales,RENTAL\_OFFICE\_LOCATION.RentalOfficeID

**FROM** RESERVATION

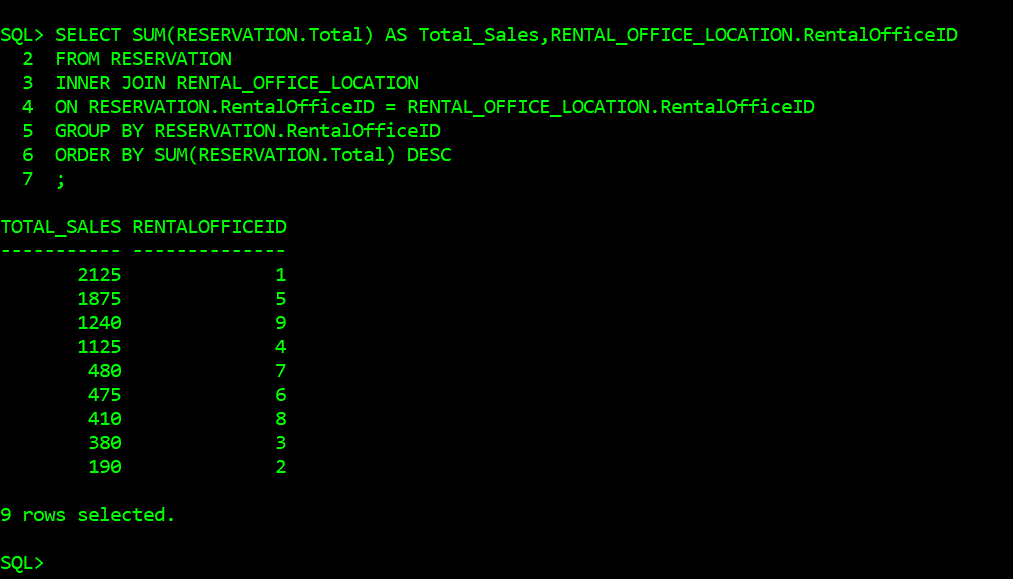
**INNER** **JOIN** RENTAL\_OFFICE\_LOCATION

**ON** RESERVATION.RentalOfficeID = RENTAL\_OFFICE\_LOCATION.RentalOfficeID

**GROUP** **BY** RESERVATION.RentalOfficeID

**ORDER** **BY** SUM(RESERVATION.Total) **DESC**;

|  |  |
| --- | --- |
| **Total Sales** | **Rental Office ID** |
| 2125 | 1 |
| 1875 | 5 |
| 1240 | 9 |
| 1125 | 4 |
| 480 | 7 |
| 475 | 6 |
| 410 | 8 |
| 380 | 3 |
| 190 | 2 |



1. **Show vehicles with maintenance record**

Vehicle came in on date for maintenance, which is handled by employee of that rental office. Below query will help us in finding the vehicle and its maintenance type and which vehicle belong customer and their VIN.

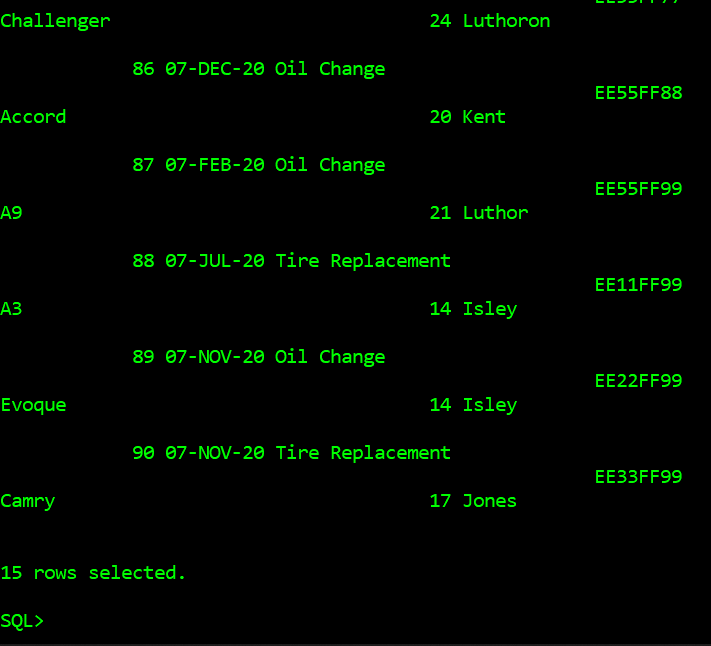
**SYNTAX**

**SELECT** MAINTENANCE\_LOG.Maintenance\_ID, MAINTENANCE\_LOG.Maintenance\_Date, MAINTENANCE\_LOG.Maintenance\_Procedure, VEHICLE.LicensePlate, VEHICLE.Model, EMPLOYEE.EmployeeID, EMPLOYEE.EmployeeLName **AS** LastName

**FROM** MAINTENANCE\_LOG, VEHICLE, EMPLOYEE

**WHERE** MAINTENANCE\_LOG.EmployeeID = EMPLOYEE.EmployeeID

**AND** MAINTENANCE\_LOG.VIN = VEHICLE.VIN;



1. **Show prices of reservations with the accessory cost added**

This query will give us the output where it’s the total amount of price a customer will pay for the vehicle reservation after adding the accessory cost to their booking

**SYNTAX**

**SELECT** RESERVATION.ReservationID, (RESERVATION.Total + ACCESSORY.Cost) **AS** TotalCost, CustomerFName **AS** FirstName

**FROM** RESERVATION, INCLUDES, ACCESSORY, CUSTOMER

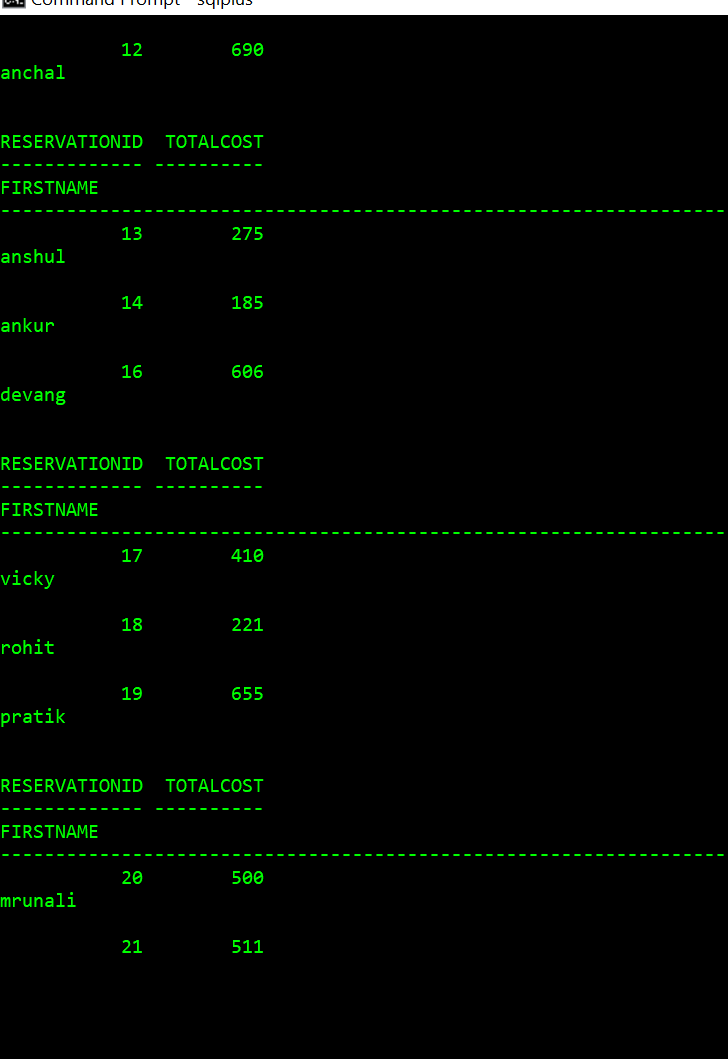
**WHERE** RESERVATION.ReservationID = INCLUDES.ReservationID **AND**

INCLUDES.SerialNumber = ACCESSORY.SerialNumber **AND**

CUSTOMER.CustomerID = RESERVATION.CustomerID

**ORDER BY** RESERVATION.ReservationID;

|  |  |  |
| --- | --- | --- |
| **First name** | **Reservation ID** | **Total Cost** |
| Ankur | 14 | 185 |
| Anshul | 13 | 275 |
| Vicky | 17 | 410 |
| Devang | 16 | 606 |
| Shweta | 21 | 511 |
| Pratik | 19 | 655 |
| Aamir | 11 | 281 |
| Anchal | 12 | 690 |
| Mrunali | 20 | 500 |
| Rohit | 18 | 221 |
| Alia | 7 | 900 |
| Priyanka | 8 | 905 |
| Sharukh | 9 | 1275 |
| Salman | 10 | 1275 |



1. This query will pull vehicle and customer identification information from existing reservations, and display them in order based on the total cost of the reservation.

**SYNTAX**

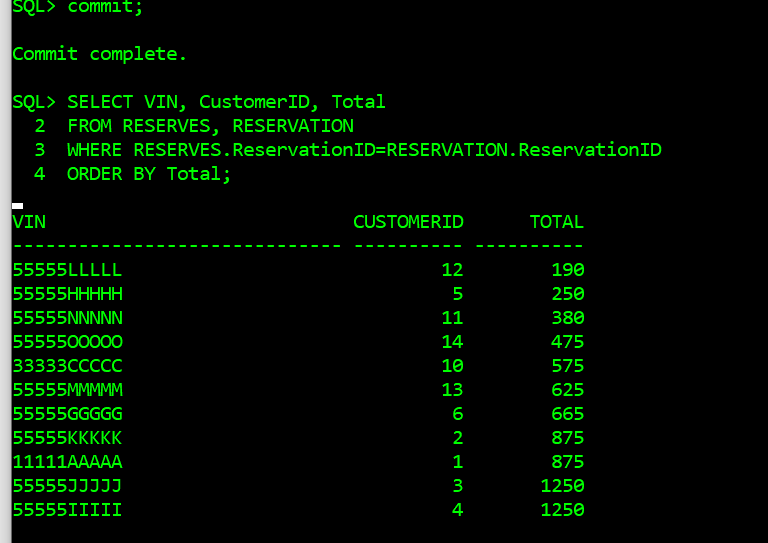
**SELECT** VIN, CustomerID, Total

**FROM** RESERVES, RESERVATION

**WHERE** RESERVES.ReservationID=RESERVATION.ReservationID

**ORDER** **BY** Total;

|  |  |  |
| --- | --- | --- |
| **VIN** | **CID** | **TOTAL** |
| 55555LLLLL | 12 | 190 |
| 55555HHHHH | 5 | 250 |
| 55555NNNNN | 11 | 380 |
| 55555OOOOO | 14 | 475 |
| 33333CCCCC | 10 | 575 |
| 55555MMMMM | 13 | 625 |
| 55555GGGGG | 6 | 665 |
| 55555KKKKK | 2 | 875 |
| 11111AAAAA | 1 | 875 |
| 55555JJJJJ | 3 | 1250 |
| 55555IIIII | 4 | 1250 |



1. **UNION QUERY**

Query to get the data information of certain column from table and union them with the other column from other table. Here EmployeeID and EmployeeFName of employee table where title of employee is office manager is getting union with the customerID and Driverslicense of customer table .

**SYNTAX**

**SELECT** EmployeeID, EmployeeFName

**FROM** EMPLOYEE

**WHERE** title = ‘office manager’

**UNION ALL**

**SELECT** customerID, DriversLicense

**FROM** Customer

**WHERE** customerID > 000011

**ORDER BY** 2;

A screenshot of a cell phone

Description automatically generated

1. The query should only display the models within the above range, which this case should be two different makes and the two corresponding models accordingly: "available make and models with mileage at least 10,000 miles but less than 20,000 miles."

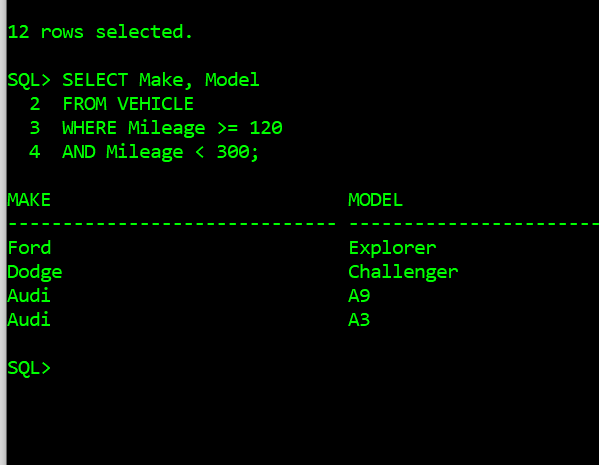
**SYNTAX**

**SELECT** Make, Model

**FROM** VEHICLE

**WHERE** Mileage **>=** 120

**AND** Mileage **<** 300;



1. This query will count all of the employed positions by selecting the Title Column from the Employee-table using the COUNT method in conjucnction with the DISTINCT method to effectively

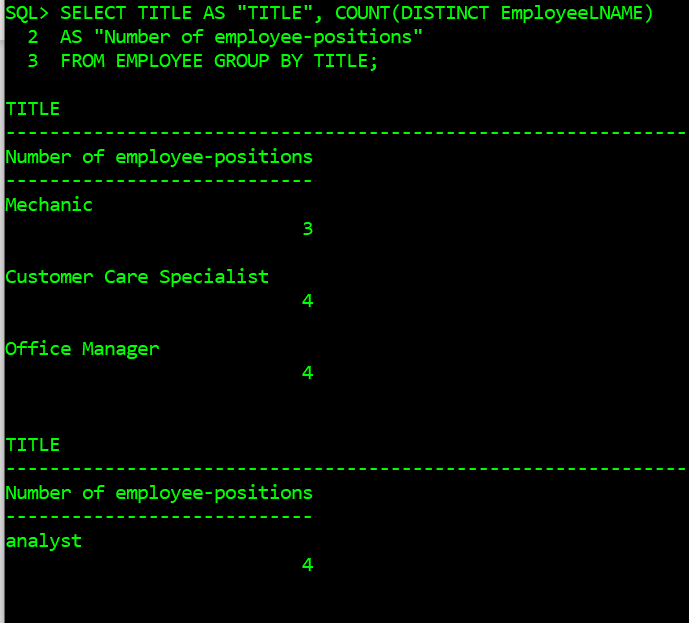
count all unique last-names then grouping the count of unique last names corresponding to those employee-positions by Title from the EMPLOYEE-Table displayed in an alias-table "COUNT OF EMPLOYEES."

**SYNTAX**

**SELECT** TITLE AS "TITLE", COUNT(DISTINCT EmployeeLNAME)

**AS** "Number of employee-positions"

**FROM** EMPLOYEE **GROUP** **BY** TITLE;



**FINAL REPORT**

Above Diagram show total cost of reservation made by every vehicle based on vehicle identification number (VIN) and customerID (CID). Here customer number 4 has made the expensive booking among all whereas customer number 5 has made the cheapest reservation with car rental company.

pie chart depicts that the maximum no of sale which is 26% has been done by rental office number 1 followed by number 5 with totaling of 23% of total sales. Rental office number 2 and 3 need to improve their sales to generate revenue.

**Vehicle Requiring maximum maintenance**

Honda accord and soul with maximum number of maintenance, where accord need two time oil change and 1 time alignment check and soul need 3 different maintenance procedure of annual maint, oil change and tire replacement. therefore, they should improve the vehicle condition for both honda accord and soul.

Customer Devang and Anchal has made booking from rental office number 9. Rental office number 5 and 8 also manages to book atleast two vehicles but rest of them only able to make single reservation. Hence Improvement at all other rental location is needed.