CS 6360.001 Database Design

FINAL PROJECT REPORT

TEAM NUMBER – 12 (UBER-2)

TEAM MEMBERS:

MOHAMED ABRAR (MXA190079)

KAMESHWARI SOUNDARARAJAN (KXS210013)

SARA TABASSI (SXT200083)

Table of Contents

PROJECT DESCRIPTION:

Uber is a transportation company with an app that allows passengers to book a ride and drivers to charge fares and get paid. More specifically, Uber is a ridesharing company that hires independent contractors as drivers. It's one of many services today that contribute to the sharing economy, supplying a means of connecting existing resources instead of providing the physical resources themselves.

How does it work?

Uber links passengers with drivers using the Uber app. Generally, the drivers own their own car. Uber offers rides under a dynamic pricing model for both drivers and passengers. Passengers needing a ride can use the app to hail a driver with an estimated price that is dependent on the destination as well as the demand at the time.

Uber incentivizes drivers to pick up more fares in peak busy hours by paying more during those times. This means that riders are charged more at busy times in order to help ensure the needed number of drivers are available. During holidays, such as New Year's Eve, a passenger can expect to pay a high price. However, unlike with a taxi, they can expect that the price will attract a driver, as opposed to a car that might never have shown up from a taxi service.

Important Components of the System

Drivers and **Customers** are the two most significant actors in the domain, and they constitute the core foundation of the Database Model.

Driver - a person over the age of 18 who has a valid, unexpired driver's license and a Social Security number, as well as a registered car under insurance.
Customer - a person who has an Uber account and needs to be picked up from one location and dropped off at another.

Uber's System Operation :

Uber is a real-time application that enables customers to request a taxi from their present position to their desired location.

- The customer is qualified for an Uber promotion based on their account, which they can take advantage of by entering the promo code.

- The customer requests a ride and has the option of selecting the type of ride they want (UberX, UberXL).

- Based on the type of ride selected, the customer can also see the expected fare pricing.

- The customer is assigned a ride based on the driver's availability in the present location.

PROJECT REQUIREMENTS:

> A **User** entity has been formed, with the user type being either a customer or a driver, both of whom are registered with Uber and are considered Uber Users. Common details about Uber users such as name, date of birth, address, email, and phone number is stored in the User entity. Every user holds a unique ID (UberID) for identification purposes.

> Based on points obtained a customer is categorized into 4 types namely silver, gold, diamond, platinum. This information is stored in CustomerType attribute under **customer**.

> The specific details of a driver such as driver license number (DLNo), driver license expiry date (DLExpiry_date), SSN, date, and the duration of work hours (Login_time, Logout_time) are stored in **driver** entity.

> After completing every trip, the driver receives his/her share. Every detail from the trip such as the date, no.of trips, distance, amount all are recorded using a unique ID (shareID).

> **Vehicle** entity contains all the details about the car owned by a driver. It includes unique vehicle ID (VID), vehicle license no (LicenseNo), model of car (Model), type of the vehicle (VehicleType), year of manufacture (ManufYear), date of purchase (PurchaseDate), color, and the status if the vehicle has passengers on board or not (VehicleStatus).

> Apart from these the insurance details of the car: Insurance number (InsuranceNo) and the expiry date of the vehicle insurance (ExpiryDate) is stored in the **Insurance** entity, which will be accessed at the time of investigation or for renewal remainders to the drivers.

> Every **trip** request made by a customer holds a unique trip ID (TID) and the details like the trip request time (Book_time), distance from the pickup to drop-off location (Distance), fare estimation based on route (Estimated_fare). Uber in general allows sharing a ride; so, there could be a possibility of having multiple pickup locations as well as multiple drop-off points in a single trip.

> Every trip request falls into one of the following categories: Completed trips (Completed_trips), Incomplete trips (In_completed_trips).

> **Completed_trips** will have the updated fare (Fare), tax based on the cost (tax), duration of the trip, tips awarded by customer (Tip), pickup time and the drop-off time of a customer.

> After completing every trip, the customer can rate their experience by giving the details such as the rating of the driver (DriverRating), suggestions from the customer (CustomerFeedback). Even every driver can rate their customer (CustomerRating) and give feedback about their customer (DriverFeedback) and every **rating** holds a unique ID (RatingID) for future processing.

> If the trip is incomplete the reason for the cancelation and the cancelled time is stored in the **In_completed_trips** entity.

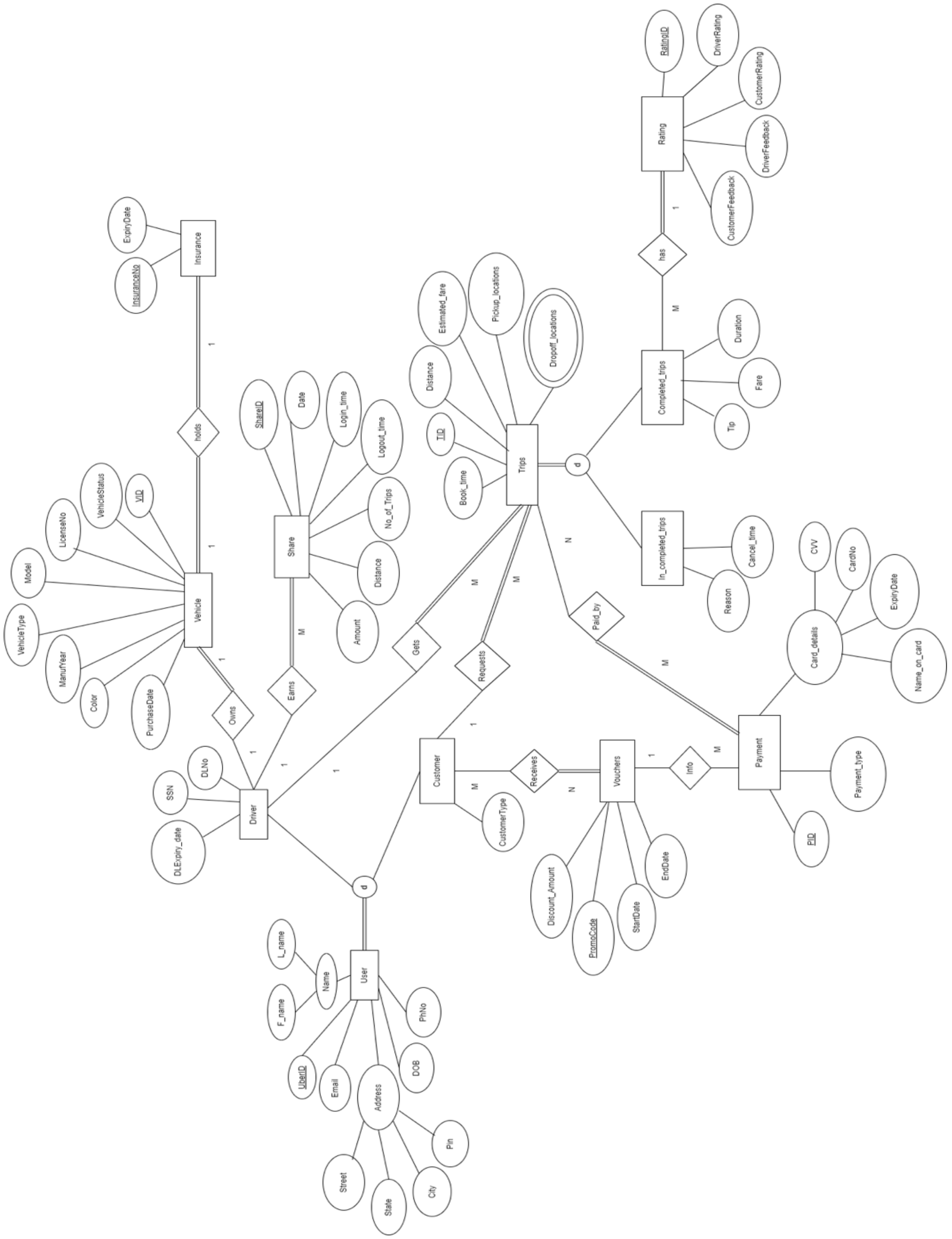
> A customer may hold a coupon code which reduces the amount that has to paid by him/her. Every voucher holds a unique ID (PromoCode), amount of possible discount

(Discount_Amount), starting and the ending date of the coupon is recorded in the **Vouchers** entity.

> Each transaction is recorded in the **payment**. Every transaction holds a unique payment ID (PID), card details of the customer for making the payment such as the CVV, CardNo, ExpiryDate, Name_on_card. The payment can be completed in one of the 2 ways personal payment or business payment which is given in Payment_type attribute.

EER DIAGRAM:

UBER EER-DIAGRAM



(a) One-to-One binary relationships:

1. **VEHICLE ---holds--- INSURANCE:** Total Participation on both sides. Each VEHICLE will have only one INSURANCE and each INSURANCE is associated with only one VEHICLE.

2. **DRIVER ---owns--- VEHICLE:** Total Participation on VEHICLE; Partial Participation on DRIVER. Each DRIVER owns one VEHICLE and each VEHICLE is owned by one DRIVER.

(b) One-to-Many binary relationships:

1. **DRIVER ---earns--- SHARE:** Total Participation on SHARE; Partial Participation on DRIVER.
   One DRIVER earns many SHARES in a particular day and each SHARE is associated with only one DRIVER.

2. **DRIVER ---gets---- TRIPS:** Total Participation on TRIPS; Partial Participation on DRIVER.
   One DRIVER gets many TRIPS and each TRIP is assigned to only one DRIVER.

3. **CUSTOMER ---requests ---- TRIPS:** Total Participation on TRIPS; Partial Participation on CUSTOMER.
   One CUSTOMER can request for multiple TRIPS and each TRIP is requested by only one CUSTOMER.

4. **COMPLETED_TRIPS ---has--- RATING:** Total Participation on RATING; Partial Participation on COMPLETED_TRIPS.
   Each completed trip has one RATING and many completed trips can have the same RATING.

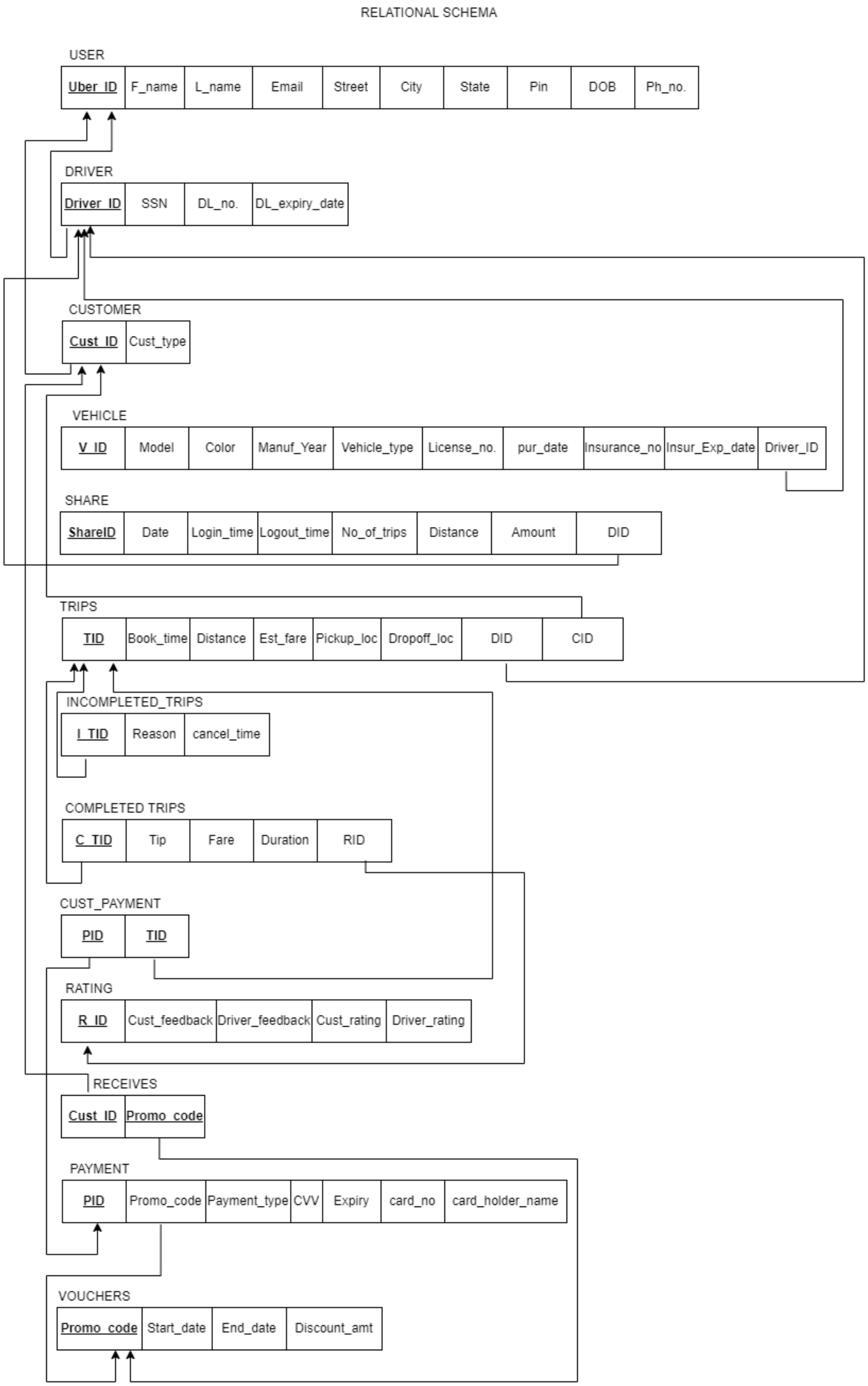
5. **VOUCHERS ---info--- PAYMENT**: Partial Participation on both sides.
   A particular VOUCHER can be used by different customers in their PAYMENT and each PAYMENT can have only one VOUCHER applied to it.

(c) Many-to-Many binary relationships:

1. **TRIPS ---paid_by---- PAYMENT:** Total Participation on PAYMENT; Partial Participation on TRIPS. A TRIP can be paid by multiple customers (PAYMENT) Ex; UBER POOL and one customer's PAYMENT information can be used for multiple trips.

2. **CUSTOMER ---receives--- VOUCHER:** Total Participation on VOUCHERS; Partial Participation on CUSTOMER. Each CUSTOMER can receive multiple vouchers and a particular VOUCHER can be given to many CUSTOMERS.

MAPPING EER DIAGRAM TO RELATIONAL MODEL:



RELATIONAL SCHEMA

USER

Uber_ID	F_name	L_name	Email	Street	City	State	Pin	DOB	Ph_no.

DRIVER

Driver_ID	SSN	DL_no.	DL_expiry_date

CUSTOMER

Cust_ID	Cust_type

VEHICLE

V_ID	Model	Color	Manuf_Year	Vehicle_type	License_no.	pur_date	Insurance_no	Insur_Exp_date	Driver_ID

SHARE

ShareID	Date	Login_time	Logout_time	No_of_trips	Distance	Amount	DID

TRIPS

TID	Book_time	Distance	Est_fare	Pickup_loc	Dropoff_loc	DID	CID

INCOMPLETED_TRIPS

I_TID	Reason	cancel_time

COMPLETED TRIPS

C_TID	Tip	Fare	Duration	RID

CUST_PAYMENT

PID	TID

RATING

R_ID	Cust_feedback	Driver_feedback	Cust_rating	Driver_rating

RECEIVES

Cust_ID	Promo_code

PAYMENT

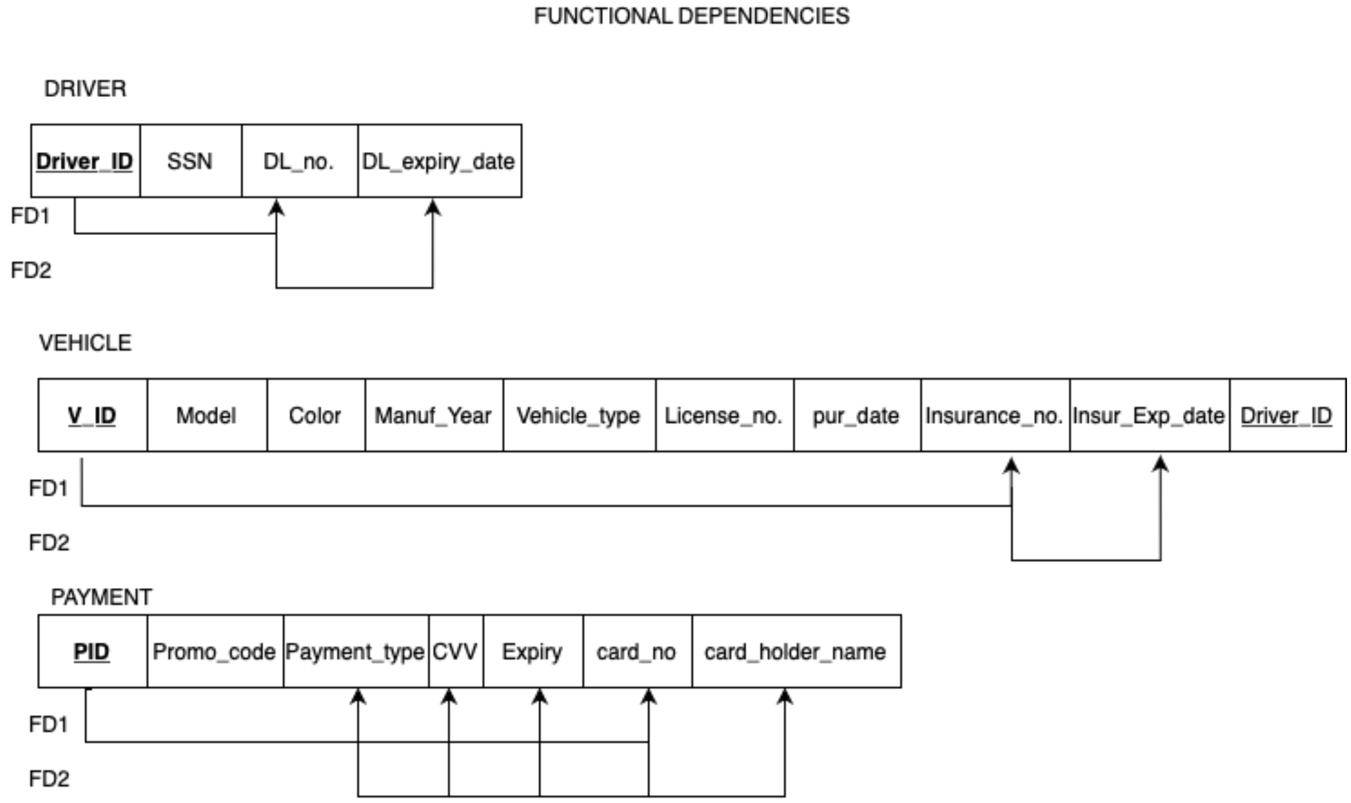
PID	Promo_code	Payment_type	CVV	Expiry	card_no	card_holder_name

VOUCHERS

Promo_code	Start_date	End_date	Discount_amt

FUNCTIONAL DEPENDENCIES AND NORMALIZATION:

All our tables contain atomic values and there exists no partial dependency in the tables. Hence, our schema is **already in 1NF and 2NF.**



- > **Driver** { **Driver_ID** , SSN, DL_no, DL_expiry_date }
  FD1: **Driver_ID--**-> DL_no,
  FD2: DL_no ---> DL_expiry_date

Here FD2 violates 3NF as there exists a transitive dependency.

So, the new tables are:
Driver{ **Driver_ID**, SSN, *DL_no,* }

Driver_License_Info{ **DL_no** , DL_expiry_date }

- > **Vehicle** { **V_ID**, *Driver_ID*, Model, Color , Manuf_Year, Pur_Date, License_no,Insurance_no, Insur_Exp_date}

FD1: **V_ID**---> Insurance_no

FD2: Insurance_no > Insur_Exp_date

Here FD2 violates 3NF as there exists a transitive dependency.

So, the new tables are:

Vehicle { **V_ID**, *Driver_ID*, Model, Color , Manuf_Year, Pur_Date, License_no, Insurance_no}

Insurance_Info{ **Insurance_no**, Insur_Exp_date}

- **Payment**{ **PID**, Promo_code, Payment_Type, Card_no, Card_holder_name, CVV, Expiry}

  FD1: **PID** ---> Card_no
  FD2: Card_no ---> CVV, Expiry, Payment_Type, Card_holder_name

Here FD2 violates 3NF as there exists a transitive dependency.

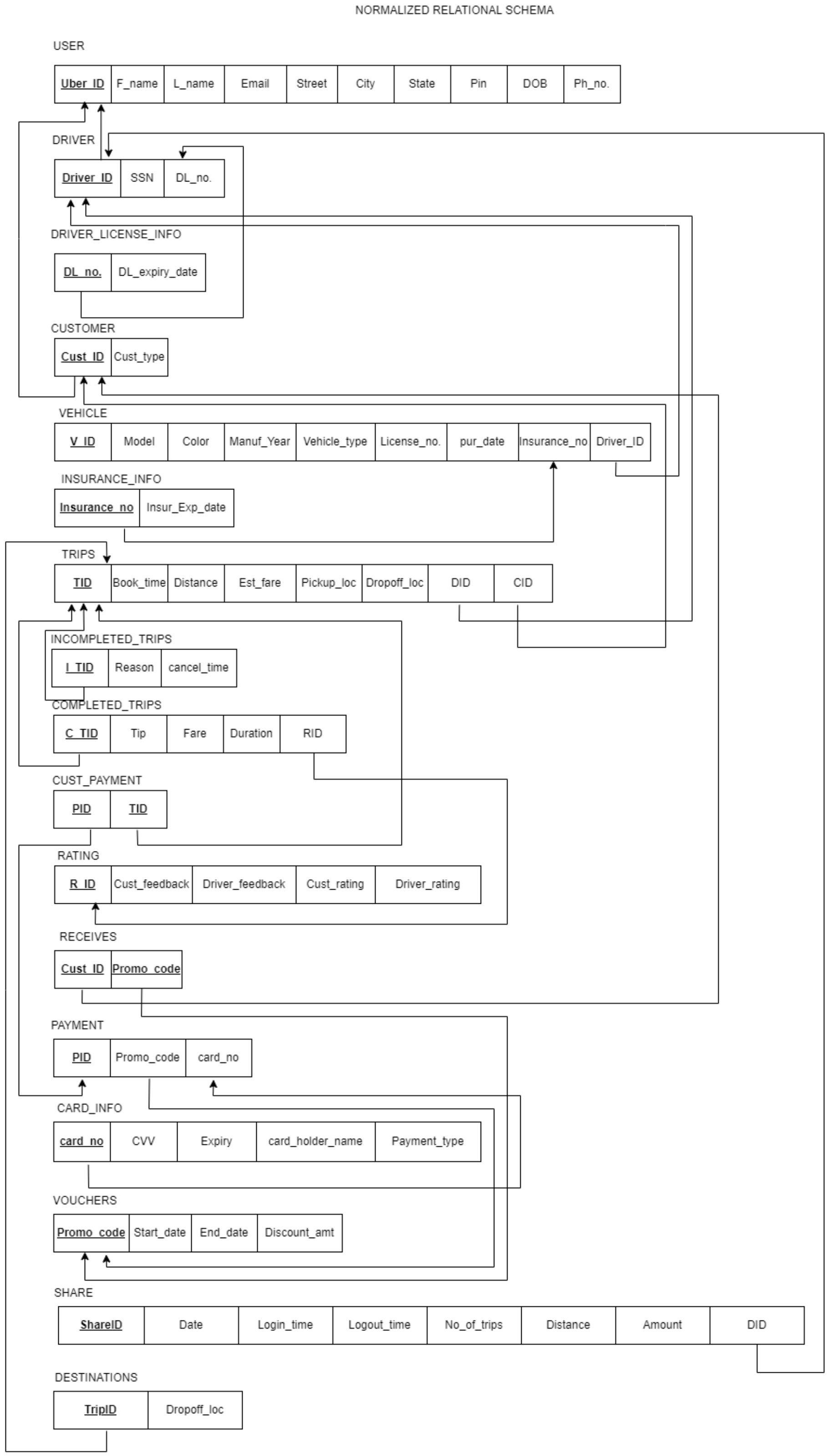
So, the new tables are:

Payment{ **PID** , Promo_code ,Card_no }

Card_Info { **Card_no**, CVV, Expiry, Payment_Type, Card_holder_name }

*Primary Key - **Bold**
*Foreign Key - *Italics and Underlined*

FINAL RELATIONAL SCHEMA AFTER NORMALIZATION:



NORMALIZED RELATIONAL SCHEMA

USER

Uber_ID	F_name	L_name	Email	Street	City	State	Pin	DOB	Ph_no.

DRIVER

Driver_ID	SSN	DL_no.

DRIVER_LICENSE_INFO

DL_no.	DL_expiry_date

CUSTOMER

Cust_ID	Cust_type

VEHICLE

V_ID	Model	Color	Manuf_Year	Vehicle_type	License_no.	pur_date	Insurance_no	Driver_ID

INSURANCE_INFO

Insurance_no	Insur_Exp_date

TRIPS

TID	Book_time	Distance	Est_fare	Pickup_loc	Dropoff_loc	DID	CID

INCOMPLETED_TRIPS

I_TID	Reason	cancel_time

COMPLETED_TRIPS

C_TID	Tip	Fare	Duration	RID

CUST_PAYMENT

PID	**TID**

RATING

R_ID	Cust_feedback	Driver_feedback	Cust_rating	Driver_rating

RECEIVES

Cust_ID	**Promo_code**

PAYMENT

PID	Promo_code	card_no

CARD_INFO

card_no	CVV	Expiry	card_holder_name	Payment_type

VOUCHERS

Promo_code	Start_date	End_date	Discount_amt

SHARE

ShareID	Date	Login_time	Logout_time	No_of_trips	Distance	Amount	DID

DESTINATIONS

TripID	Dropoff_loc

SQL CODE FOR CREATE TABLE & INSERT VALUES:

- **UBER_USER:**

```
CREATE TABLE UBER_USER
(
Uber_ID INT NOT NULL,
F_name VARCHAR(50) NOT NULL,
L_Name VARCHAR(50) NOT NULL,
Ph_no INT NOT NULL,
Email VARCHAR(50) NOT NULL,
Street VARCHAR(50) NOT NULL,
City VARCHAR(30) NOT NULL,
State_name VARCHAR(30),
Pin INT NOT NULL,
DOB DATE NOT NULL,
PRIMARY KEY(Uber_ID)
);

//INSERTING VALUES INTO THE TABLES

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(1111, 'Jim','Williams',1324567890, 'jw@gmail.com',
'Coit','Dallas','TX',78945,TO_DATE('1990-12-3','YYYY-MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(2222, 'Jim','Williams',1324567890, 'jw@gmail.com',
'Coit','Dallas','TX',78945,TO_DATE('1990-12-3','YYYY-MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(3333, 'Jim','Williams',1324567890, 'jw@gmail.com',
'Coit','Dallas','TX',78945,TO_DATE('1990-12-3','YYYY-MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(4444, 'Jim','Williams',1324567890, 'jw@gmail.com',
'Coit','Dallas','TX',78945,TO_DATE('1990-12-3','YYYY-MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(5555, 'Jim','Williams',1324567890, 'jw@gmail.com',
'Coit','Dallas','TX',78945,TO_DATE('1990-12-3','YYYY-MM-DD'));
```

```
INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(0147, 'Jim','Williams',1324567890, 'jw@gmail.com',
'Coit','Dallas','TX',78945,TO_DATE('1990-12-3','YYYY-MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(4656, 'Julie','Swan', 7418529630,
'js@gmail.com','Frankford','Austin', 'TX', 45678, TO_DATE('1980-
5-9','YYYY-MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(8520, 'Joey','Buffay',2583691470,
'jb@gmail.com','Campbell','Richardson','TX',
15926,TO_DATE('1996-7-19','YYYY-MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(4928,'John','Smith', 3216549870,
'js@gmail.com','Ricky','Blach Springs','TX',59268,
TO_DATE('1989-9-28','YYYY-MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(2598, 'Jack', 'Chandler', 3698521479,
'jc@gmail.com','Frankford','Green Lake','TX',35786,
TO_DATE('1987-12-19','YYYY-MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(4859, 'George', 'Matt',2581472580, 'gm@gmail.com',
'Campbell', 'Austin','TX', 24356, TO_DATE('1986-9-26','YYYY-MM-
DD'));


INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(2693, 'Rachel','Black',7946138520,
'rb@gmail.com','Frankford','Green Lake','TX',9874,
TO_DATE('1999-6-19','YYYY-MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
```

```
VALUES(1234, 'Jaden','Will',3690926498,
'jw@gmail.com','Campbell', 'Austin','TX',1458, TO_DATE('1982-10-
16','YYYY-MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(9685, 'John','Corner',4679792585,
'jc@gmail.com','Coit','Richardson','TX',8552, TO_DATE('1992-11-
14','YYYY-MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(1948, 'Megan', 'Fox',5948592636, 'mf@gmail.com','
Campbell','Blanch Springs','TX', 3629, TO_DATE('1994-8-
19','YYYY-MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(1034, 'kyle', 'Fox',5938502636, 'kf@gmail.com','
Campbell','ranch rings','TX', 3629, TO_DATE('1999-8-19','YYYY-
MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(2345, 'john', 'cole',6638502636, 'jc@gmail.com','
mound','waco','TX', 3629, TO_DATE('1999-3-30','YYYY-MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(3456, 'sara', 'jamie',5928902636, 'sj@gmail.com','
Campbell','ranch rings','TX', 3629, TO_DATE('2000-4-20','YYYY-
MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(4567, 'dan', 'bill',9123402636, 'db@gmail.com','
Campbell','ranch rings','TX', 3629, TO_DATE('1999-8-19','YYYY-
MM-DD'));

INSERT INTO UBER_USER(UBER_ID, F_name, L_Name, Ph_no, Email,
Street, City, State_name, Pin, DOB)
VALUES(5678, 'loan', 'cox',5938502636, 'lc@gmail.com','
Courtyrds','houston','TX', 3629, TO_DATE('1979-8-19','YYYY-MM-
DD'));
```

- **DRIVER:**

```
CREATE TABLE DRIVER
(
DID INT NOT NULL,
SSN INT NOT NULL,
DLNo INT Unique NOT NULL,
PRIMARY KEY(DID),
FOREIGN KEY(DID) REFERENCES UBER_USER(Uber_ID)
);

//INSERTING VALUES INTO THE TABLES

INSERT INTO DRIVER(DID, SSN, DLNo)
VALUES(0147, 7418529639, 7894);

INSERT INTO DRIVER(DID, SSN, DLNo)
VALUES(8520, 7946138596, 9636);

INSERT INTO DRIVER(DID, SSN, DLNo)
VALUES(4928, 5696897462, 8594);

INSERT INTO DRIVER(DID, SSN, DLNo)
VALUES(2598, 594875869, 2983);

INSERT INTO DRIVER(DID, SSN, DLNo)
VALUES(4859, 9685744589, 9518);

INSERT INTO DRIVER(DID, SSN, DLNo)
VALUES(1111, 9675744589, 9076);

INSERT INTO DRIVER(DID, SSN, DLNo)
VALUES(2222, 6985744589, 9556);

INSERT INTO DRIVER(DID, SSN, DLNo)
VALUES(3333, 9685744589, 3493 );

INSERT INTO DRIVER(DID, SSN, DLNo)
VALUES(4444, 9634744589, 2310);

INSERT INTO DRIVER(DID, SSN, DLNo)
VALUES(5555, 7985744589, 9012);
```

- **DRIVER_LICENSE:**

```
CREATE TABLE Driver_License
(
DLNo INT NOT NULL,
DL_expiry_date DATE NOT NULL,
PRIMARY KEY(DLNo),
FOREIGN KEY (DLNo) REFERENCES Driver(DLNo)
);

//INSERTING VALUES INTO THE TABLES

INSERT INTO DRIVER_LICENSE(DLNo, DL_expiry_date)
VALUES(7894, TO_DATE('2023-1-5','YYYY-MM-DD'));

INSERT INTO DRIVER_LICENSE(DLNo, DL_expiry_date)
VALUES(9636, TO_DATE('2024-11-19','YYYY-MM-DD'));

INSERT INTO DRIVER_LICENSE(DLNo, DL_expiry_date)
VALUES(8594, TO_DATE('2023-5-9','YYYY-MM-DD'));

INSERT INTO DRIVER_LICENSE(DLNo, DL_expiry_date)
VALUES(2983, TO_DATE('2024-8-19','YYYY-MM-DD'));

INSERT INTO DRIVER_LICENSE(DLNo, DL_expiry_date)
VALUES(9518, TO_DATE('2024-11-19','YYYY-MM-DD'));
```

- **CUSTOMER:**

```
CREATE TABLE Customer
(
 CID INT NOT NULL,
 CustomerType VARCHAR(15) NOT NULL ,
 PRIMARY KEY(CID),
 FOREIGN KEY (CID) REFERENCES Uber_User(Uber_ID)
);

//INSERTING VALUES INTO THE TABLES


INSERT INTO Customer(CID, CustomerType)
VALUES(4656,'Gold');

INSERT INTO Customer(CID, CustomerType)
VALUES(2693, 'Diamond');
```

```
INSERT INTO Customer(CID, CustomerType)
VALUES(1234, 'Platinum');

INSERT INTO Customer(CID, CustomerType)
VALUES(9685, 'Silver');

INSERT INTO Customer(CID, CustomerType)
VALUES(1948, 'Diamond');

INSERT INTO Customer(CID, CustomerType)
VALUES(1034,'Gold');

INSERT INTO Customer(CID, CustomerType)
VALUES(2345, 'Diamond');

INSERT INTO Customer(CID, CustomerType)
VALUES(3456, 'Platinum');

INSERT INTO Customer(CID, CustomerType)
VALUES(4567, 'Silver');

INSERT INTO Customer(CID, CustomerType)
VALUES(5678, 'Diamond');
```

- **VOUCHERS:**

```
CREATE TABLE VOUCHERS
(
Promocode INT NOT NULL,
Start_date DATE,
End_date DATE NOT NULL,
Discount int,
PRIMARY KEY(Promocode)
);

//INSERTING VALUES INTO THE TABLES

INSERT INTO VOUCHERS(Promocode, Start_date, End_date, Discount)
VALUES(96326, TO_DATE('2021-11-29','YYYY-MM-DD'), TO_DATE('2021-
12-30','YYYY-MM-DD'), 20);


INSERT INTO VOUCHERS(Promocode, Start_date, End_date, Discount)
```

```
VALUES(29598, TO_DATE('2021-12-1','YYYY-MM-DD'), TO_DATE('2021-
12-30','YYYY-MM-DD'), 10);


INSERT INTO VOUCHERS(Promocode, Start_date, End_date, Discount)
VALUES(59689, TO_DATE('2021-11-29','YYYY-MM-DD'),TO_DATE('2021-
12-31','YYYY-MM-DD'), 25);

INSERT INTO VOUCHERS(Promocode, Start_date, End_date, Discount)
VALUES(59152, TO_DATE('2021-11-29','YYYY-MM-DD'), TO_DATE('2021-
12-30','YYYY-MM-DD'), 30);


INSERT INTO VOUCHERS(Promocode, Start_date, End_date, Discount)
VALUES(22584, TO_DATE('2021-11-30','YYYY-MM-DD'), TO_DATE('2021-
12-31','YYYY-MM-DD'), 15);
```

- **PAYMENT:**

```
CREATE TABLE Payment
(
 PID int NOT NULL,
 CardNo int Unique NOT NULL,
 Couponcode int NOT NULL,
 PRIMARY KEY(PID),
 FOREIGN KEY (Couponcode) REFERENCES VOUCHERS(Promocode)
);

//INSERTING VALUES INTO THE TABLES

INSERT INTO Payment(PID, CardNo, Couponcode)
VALUES(16592, 1959483629, 96326);


INSERT INTO Payment(PID, CardNo, Couponcode)
VALUES(59682, 5789123963, 29598);

INSERT INTO Payment(PID, CardNo,Couponcode)
VALUES(32165, 9876548521, 59689);

INSERT INTO Payment(PID, CardNo,Couponcode)
VALUES(49165, 4875598658,59152);

INSERT INTO Payment(PID, CardNo, Couponcode)
VALUES(29487, 4956785896, 22584);
```

- **CARD_INFO:**

```
CREATE TABLE Card_info
(
 CardNo int NOT NULL,
 CVV int NOT NULL,
 Card_holder_name VARCHAR(30) NOT NULL,
 Expiry DATE NOT NULL,
 Payment_type varchar(50) NOT NULL,
 PRIMARY KEY(CardNo),
 FOREIGN KEY (CardNo) REFERENCES Payment(CardNo)
);

//INSERTING VALUES INTO THE TABLES

INSERT INTO Card_info(CardNo, CVV, Card_holder_name,
Expiry,Payment_type)
VALUES(1959483629, 456, 'Julie', TO_DATE('2022-12-11','YYYY-MM-
DD'), 'Personalpayment');

INSERT INTO Card_info(CardNo, CVV, Card_holder_name, Expiry,
Payment_type)
VALUES(5789123963, 951, 'Rachel', TO_DATE('2023-5-19','YYYY-MM-
DD'), 'Personalpayment');

INSERT INTO Card_info(CardNo, CVV, Card_holder_name, Expiry,
Payment_type)
VALUES(9876548521, 258, 'Jaden', TO_DATE('2023-12-11','YYYY-MM-
DD'), 'Businesspayment');

INSERT INTO Card_info(CardNo, CVV, Card_holder_name, Expiry,
Payment_type)
VALUES(4875598658, 357,'John', TO_DATE('2023-4-19','YYYY-MM-
DD'), 'Personalpayment');

INSERT INTO Card_info(CardNo, CVV, Card_holder_name, Expiry,
Payment_type)
VALUES(4956785896, 396, 'Megan', TO_DATE('2023-8-11','YYYY-MM-
DD'), 'Personalpayment');
```

- **TRIPS:**

```
CREATE TABLE Trips
(
 TID INT NOT NULL,
 Distance_D float NOT NULL,
 Pickup_loc varchar(30) NOT NULL,
 Book_time TIMESTAMP NOT NULL,
 Est_fare float NOT NULL,
 DID INT NOT NULL,
 CustID INT NOT NULL,
 PRIMARY KEY(TID),
 FOREIGN KEY (DID) REFERENCES DRIVER(DID),
 FOREIGN KEY(CustID) REFERENCES Customer(CID) ON DELETE CASCADE
);

//INSERTING VALUES INTO THE TABLES

INSERT INTO Trips(TID, Distance_D, Pickup_loc,
Book_time,Est_fare,DID,CustID)
VALUES(96857, 15, 'Green Fields', TO_TIMESTAMP('2021-11-2
11:10:11', 'YYYY-MM-DD HH:MI:SS'),20,0147,1034);

INSERT INTO Trips(TID, Distance_D, Pickup_loc,
Book_time,Est_fare,DID,CustID)
VALUES(14745, 25, 'BlanchSprings', TO_TIMESTAMP('2021-11-12
10:28:19', 'YYYY-MM-DD HH:MI:SS'),35,0147,2345);

INSERT INTO Trips(TID, Distance_D, Pickup_loc,
Book_time,Est_fare,DID,CustID)
VALUES(35265, 14, 'Austin', TO_TIMESTAMP('2021-8-12 11:11:13',
'YYYY-MM-DD HH:MI:SS'),25,8520,3456);

INSERT INTO Trips(TID, Distance_D, Pickup_loc,
Book_time,Est_fare,DID,CustID)
VALUES(15598, 26, 'Richardson', TO_TIMESTAMP('2021-11-18
8:10:00', 'YYYY-MM-DD HH:MI:SS'),13,8520,4567);

INSERT INTO Trips(TID, Distance_D, Pickup_loc,
Book_time,Est_fare,DID,CustID)
VALUES(26625, 19, 'Irving', TO_TIMESTAMP('2021-11-25 9:25:19',
'YYYY-MM-DD HH:MI:SS'),69.9,4928,5678);


INSERT INTO Trips(TID, Distance_D, Pickup_loc,
Book_time,Est_fare,DID,CustID)
VALUES(59784, 20, 'Rockwall', TO_TIMESTAMP('2021-11-25 9:45:19',
'YYYY-MM-DD HH:MI:SS'),32.5,4928,4656);

INSERT INTO Trips(TID, Distance_D, Pickup_loc,
Book_time,Est_fare,DID, CustID)
```

```
VALUES(26854, 18, 'Irving', TO_TIMESTAMP('2021-1-19 9:25:19',
'YYYY-MM-DD HH:MI:SS'),23.2,2598,2693);

INSERT INTO Trips(TID, Distance_D, Pickup_loc,
Book_time,Est_fare,DID,CustID)
VALUES(41785, 24, 'Rockwall', TO_TIMESTAMP('2021-12-1 11:25:19',
'YYYY-MM-DD HH:MI:SS'),21,2598,1234);


INSERT INTO Trips(TID, Distance_D, Pickup_loc,
Book_time,Est_fare,DID,CustID)
VALUES(55784, 16, 'Irving', TO_TIMESTAMP('2021-4-16 11:35:19',
'YYYY-MM-DD HH:MI:SS'),14,4859,9685);


INSERT INTO Trips(TID, Distance_D, Pickup_loc,
Book_time,Est_fare,DID,CustID)
VALUES(67948, 45, 'Austin', TO_TIMESTAMP('2021-6-27 10:30:16',
'YYYY-MM-DD HH:MI:SS'),17.5,4859,1948);
```

- **DESTINATIONS:**

```
CREATE TABLE Destinations
(
TripID int NOT NULL,
Dropoff_loc varchar(30) NOT NULL,
PRIMARY KEY(TripID),
FOREIGN KEY(TripID) REFERENCES Trips(TID) ON DELETE CASCADE
);

//INSERTING VALUES INTO THE TABLES


INSERT INTO Destinations(TripID, Dropoff_loc)
VALUES(96857, 'Austin');

INSERT INTO Destinations(TripID, Dropoff_loc)
VALUES(14745,'Irving');

INSERT INTO Destinations(TripID, Dropoff_loc)
VALUES(35265, 'Blanch Springs');

INSERT INTO Destinations(TripID, Dropoff_loc)
VALUES(15598, 'Green fields');

INSERT INTO Destinations(TripID, Dropoff_loc)
```

```
VALUES(26625, 'Richardson');
```

- **VEHICLE:**

```
CREATE TABLE Vehicle
(
 VID INT NOT NULL,
 DrID int NOT NULL,
 Model varchar(50) NOT NULL,
 Color varchar(20) NOT NULL,
 ManufYear INT NOT NULL,
 PurDate DATE NOT NULL,
 LicenseNo INT NOT NULL,
 VehicleType varchar(30) NOT NULL,
 InsuranceNo INT Unique NOT NULL,
 PRIMARY KEY (VID),
 FOREIGN KEY(DrID) REFERENCES Driver(DID) ON DELETE CASCADE
);

//INSERTING VALUES INTO THE TABLES


INSERT INTO Vehicle(VID, DrID, Model, Color, ManufYear, PurDate,
LicenseNo, VehicleType,InsuranceNo)
VALUES(12345, 0147, 'Atlas', 'Black', 2020, TO_DATE('2020-8-
9','YYYY-MM-DD'), 78946, 'SUV',49795);


INSERT INTO Vehicle(VID, DrID, Model, Color, ManufYear, PurDate,
LicenseNo, VehicleType,InsuranceNo)
VALUES(85296, 8520, 'Bolt','White', 2018, TO_DATE('2018-6-
19','YYYY-MM-DD'), 15987, 'Minivan',45259);


INSERT INTO Vehicle(VID, DrID, Model, Color, ManufYear, PurDate,
LicenseNo, VehicleType,InsuranceNo)
VALUES(49168, 4928, 'Camry','Black','2021',TO_DATE('2021-5-
27','YYYY-MM-DD'), 98748, 'Sports',21658);

INSERT INTO Vehicle(VID, DrID, Model, Color, ManufYear, PurDate,
LicenseNo, VehicleType,InsuranceNo)
VALUES(25874, 2598, 'Civic','Metallic', 2020, TO_DATE('2020-3-
14','YYYY-MM-DD'), 25879, 'Sedan',18957);

INSERT INTO Vehicle(VID, DrID, Model, Color, ManufYear, PurDate,
LicenseNo, VehicleType,InsuranceNo)
```

```
VALUES(49768, 4859, 'Elantra GT','Red', 2018, TO_DATE('2018-9-
19','YYYY-MM-DD'), 16769, 'Sedan',49268);
```

- **INSURANCE:**

```
CREATE TABLE Insurance
(
 InsuranceNo INT NOT NULL,
 InsuranceExpiry DATE NOT NULL,
 PRIMARY KEY(InsuranceNo),
 FOREIGN KEY(InsuranceNo) REFERENCES Vehicle(InsuranceNo) ON
DELETE CASCADE
);

//INSERTING VALUES INTO THE TABLES


INSERT INTO Insurance(InsuranceNo, InsuranceExpiry)
VALUES(49795, TO_DATE('2026-5-7','YYYY-MM-DD'));



INSERT INTO Insurance(InsuranceNo, InsuranceExpiry)
VALUES(45259,TO_DATE('2025-9-20','YYYY-MM-DD'));


INSERT INTO Insurance(InsuranceNo, InsuranceExpiry)
VALUES(21658, TO_DATE('2029-2-28','YYYY-MM-DD'));


INSERT INTO Insurance(InsuranceNo, InsuranceExpiry)
VALUES(18957, TO_DATE('2022-9-16','YYYY-MM-DD'));



INSERT INTO Insurance(InsuranceNo, InsuranceExpiry)
VALUES(49268, TO_DATE('2024-3-2','YYYY-MM-DD'));
```

- **DRIVER_SHARE:**

```
CREATE TABLE Driver_Share
(
 ShareID INT NOT NULL,
 DT DATE NOT NULL,
 LoginTime TIMESTAMP NOT NULL,
 LogoutTime TIMESTAMP NOT NULL,
 Distance_travelled INT NOT NULL,
 No_of_Trips INT NOT NULL,
 Salary INT NOT NULL,
 Dri_ID INT NOT NULL,
 PRIMARY KEY(ShareID),
 FOREIGN KEY (Dri_ID) REFERENCES Driver(DID) ON DELETE CASCADE
);

//INSERTING VALUES INTO THE TABLES

INSERT INTO Driver_Share(ShareID, DT, LoginTime, LogoutTime,
Distance_travelled,No_of_Trips, Salary, Dri_ID)
VALUES(96857, TO_DATE('2021-11-2' , 'YYYY-MM-DD'),
TO_TIMESTAMP('2021-11-12 7:28:19', 'YYYY-MM-DD HH:MI:SS'),
TO_TIMESTAMP('2021-11-12 12:50:19', 'YYYY-MM-DD HH:MI:SS'),
220,3, 800, 0147);

INSERT INTO Driver_Share(ShareID, DT, LoginTime, LogoutTime,
Distance_travelled,No_of_Trips, Salary, Dri_ID)
VALUES(49758, TO_DATE('2021-11-12', 'YYYY-MM-DD'),
TO_TIMESTAMP('2021-11-12 7:00:19', 'YYYY-MM-DD HH:MI:SS'),
TO_TIMESTAMP('2021-11-12 12:58:19', 'YYYY-MM-DD HH:MI:SS'),
200,5, 740, 8520);

INSERT INTO Driver_Share(ShareID, DT, LoginTime, LogoutTime,
Distance_travelled, No_of_Trips,Salary, Dri_ID)
VALUES(25896, TO_DATE('2021-8-12', 'YYYY-MM-DD'),
TO_TIMESTAMP('2021-8-12 9:11:13', 'YYYY-MM-DD HH:MI:SS'),
TO_TIMESTAMP('2021-8-12 11:50:13', 'YYYY-MM-DD HH:MI:SS'), 69,
10, 300, 4928);


INSERT INTO Driver_Share(ShareID, DT, LoginTime, LogoutTime,
Distance_travelled, No_of_Trips,Salary, Dri_ID)
VALUES(13469, TO_DATE('2021-11-18', 'YYYY-MM-DD'),
TO_TIMESTAMP('2021-11-18 8:00:00', 'YYYY-MM-DD HH:MI:SS'),
TO_TIMESTAMP('2021-11-18 12:10:00', 'YYYY-MM-DD HH:MI:SS'),
120,15,  600, 2598);


INSERT INTO Driver_Share(ShareID, DT, LoginTime, LogoutTime,
Distance_travelled,No_of_Trips, Salary, Dri_ID)
VALUES(55824, TO_DATE('2021-11-25', 'YYYY-MM-DD'),
TO_TIMESTAMP('2021-11-18 7:10:00', 'YYYY-MM-DD HH:MI:SS'),
TO_TIMESTAMP('2021-11-18 12:40:00', 'YYYY-MM-DD HH:MI:SS'),
150,7, 600, 4859);
```

- **CUST_PAYMENT:**

```
CREATE TABLE Paid_by
(
    PID INT NOT NULL,
    TID INT NOT NULL,
    PRIMARY KEY(PID,TID),
    FOREIGN KEY(PID) REFERENCES PAYMENT(PID) ON DELETE CASCADE,
    FOREIGN KEY(TID) REFERENCES Trips(TID) ON DELETE CASCADE
);

//INSERTING VALUES INTO THE TABLES


INSERT INTO Paid_by(PID, TID)
VALUES( 16592, 96857);


INSERT INTO Paid_by(PID, TID)
VALUES(59682,14745);


INSERT INTO Paid_by(PID, TID)
VALUES(32165,35265);

INSERT INTO Paid_by(PID, TID)
VALUES(49165,15598);

INSERT INTO Paid_by(PID, TID)
VALUES(29487,26625);
```

- **INCOMPLETE_TRIPS:**

```
CREATE TABLE In_completed_trips
(
ITID int NOT NULL,
CancelTime TIMESTAMP NOT NULL,
Reason varchar(30) NOT NULL,
PRIMARY KEY(ITID),
FOREIGN KEY (ITID) REFERENCES Trips(TID) ON DELETE CASCADE
);
```

```
INSERT INTO In_completed_trips(ITID, CancelTime, Reason)
VALUES(59784, TO_TIMESTAMP('2021-12-7 11:10:00', 'YYYY-MM-DD
HH:MI:SS'), 'Late');


INSERT INTO In_completed_trips(ITID, CancelTime, Reason)

VALUES(26854, TO_TIMESTAMP('2021-11-9 12:50:00', 'YYYY-MM-DD
HH:MI:SS'), 'Unauthorized User');



INSERT INTO In_completed_trips(ITID, CancelTime, Reason)

VALUES(41785, TO_TIMESTAMP('2021-11-16 10:16:00', 'YYYY-MM-DD
HH:MI:SS'), 'Late');



INSERT INTO In_completed_trips(ITID, CancelTime, Reason)

VALUES(55784, TO_TIMESTAMP('2021-8-26 11:56:00', 'YYYY-MM-DD
HH:MI:SS'), 'Destination not clear');



INSERT INTO In_completed_trips(ITID, CancelTime, Reason)

VALUES(67948, TO_TIMESTAMP('2021-8-16 12:19:29', 'YYYY-MM-DD
HH:MI:SS'), 'Distance cant be covered');
```

- **RATING:**

```
CREATE TABLE Rating
(
 RatingID int NOT NULL,
 DriverRating int NOT NULL,
 CustomerRating int NOT NULL,
 DriverFeedback varchar(15) NOT NULL,
 CustomerFeedback varchar(15) NOT NULL,
 PRIMARY KEY(RatingID)
);

//INSERTING VALUES INTO THE TABLES
```

```
INSERT INTO
RATING(RatingID,DriverRating,CustomerRating,DriverFeedback,Custo
merFeedback)
VALUES(201,5,5,'good cust','good driver');

INSERT INTO
RATING(RatingID,DriverRating,CustomerRating,DriverFeedback,Custo
merFeedback)
VALUES(200,5,4,'Cust was good','was helpful');

INSERT INTO
RATING(RatingID,DriverRating,CustomerRating,DriverFeedback,Custo
merFeedback)
VALUES(212,5,3,'good','rude');

INSERT INTO
RATING(RatingID,DriverRating,CustomerRating,DriverFeedback,Custo
merFeedback)
VALUES(312,5,2,'good','unclean car');

INSERT INTO
RATING(RatingID,DriverRating,CustomerRating,DriverFeedback,Custo
merFeedback)
VALUES(213,3,2,'rude','lost route');
```

- **COMPLETED_TRIPS**

```
CREATE TABLE Completed_trips
(
 CTID int NOT NULL,
 Duration float NOT NULL,
 Tip float NOT NULL,
 Fare float NOT NULL,
 RID int NOT NULL,
 PRIMARY KEY(CTID),
 FOREIGN KEY (CTID) REFERENCES Trips(TID) ON DELETE CASCADE,
 FOREIGN KEY (RID) REFERENCES Rating(RatingID) ON DELETE CASCADE
);

//INSERTING VALUES INTO THE TABLES

INSERT INTO COMPLETED_TRIPS(CTID,Duration,Tip,Fare,RID)
VALUES(96857,0.5,5,20,201);
```

```
INSERT INTO COMPLETED_TRIPS(CTID,Duration,Tip,Fare,RID)
VALUES(14745,0.9,6,67,200);

INSERT INTO COMPLETED_TRIPS(CTID,Duration,Tip,Fare,RID)
VALUES(35265,1.2,3.5,101.2,212);

INSERT INTO COMPLETED_TRIPS(CTID,Duration,Tip,Fare,RID)
VALUES(15598,0.2,1,20.2,312);

INSERT INTO COMPLETED_TRIPS(CTID,Duration,Tip,Fare,RID)
VALUES(26625,0.5,1.3,40.4,213);
```

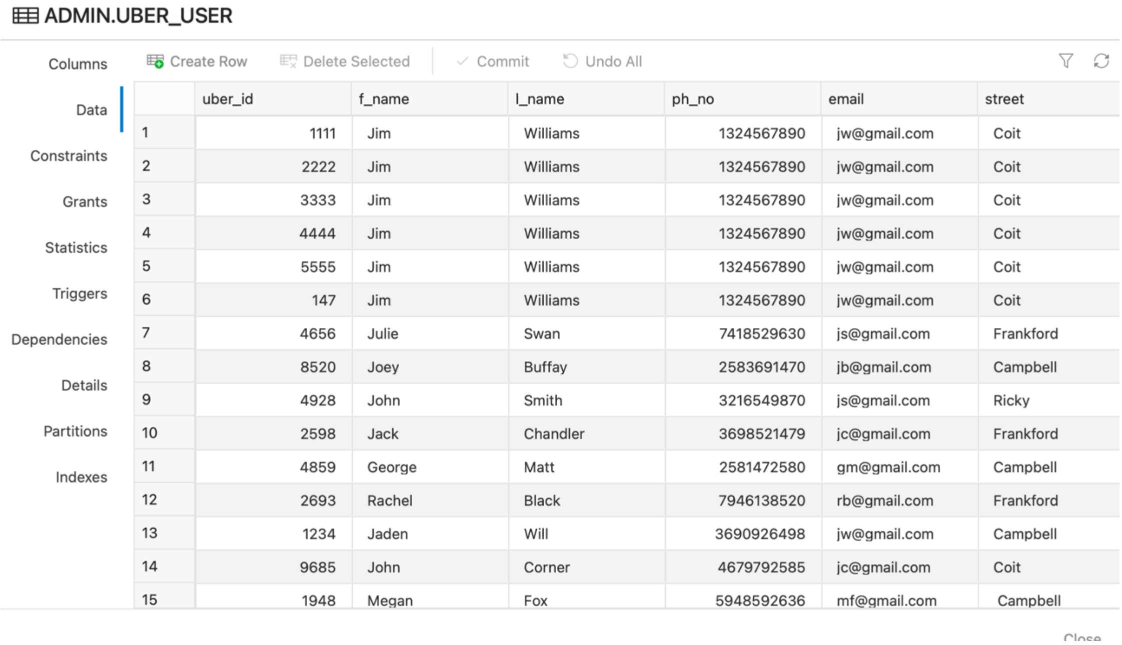
- **RECEIVES:**

```
CREATE TABLE Receives
(
    CustID int NOT NULL,
    Promocode int NOT NULL,
    PRIMARY KEY(CustID, Promocode),
    FOREIGN KEY(CustID) REFERENCES Customer(CID) ON DELETE
CASCADE,
    FOREIGN KEY(Promocode) REFERENCES VOUCHERS(Promocode) ON
DELETE CASCADE
);

//INSERTING VALUES INTO THE TABLES

INSERT INTO Receives(CustID,Promocode)
VALUES(4656,96326);

INSERT INTO Receives(CustID,Promocode)
VALUES(2693,29598);

INSERT INTO Receives(CustID,Promocode)
VALUES(1234,59689);

INSERT INTO Receives(CustID,Promocode)
VALUES(9685,59152);

INSERT INTO Receives(CustID,Promocode)
VALUES(1948,22584);
```

TABLES:

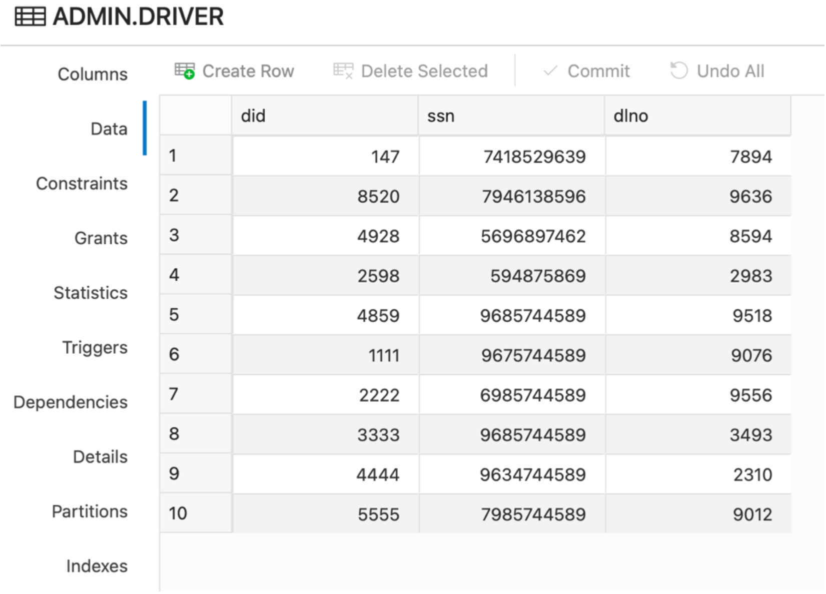
Below are snapshots of all the tables that were created on ORACLE SQL Developer.

UBER_USER:



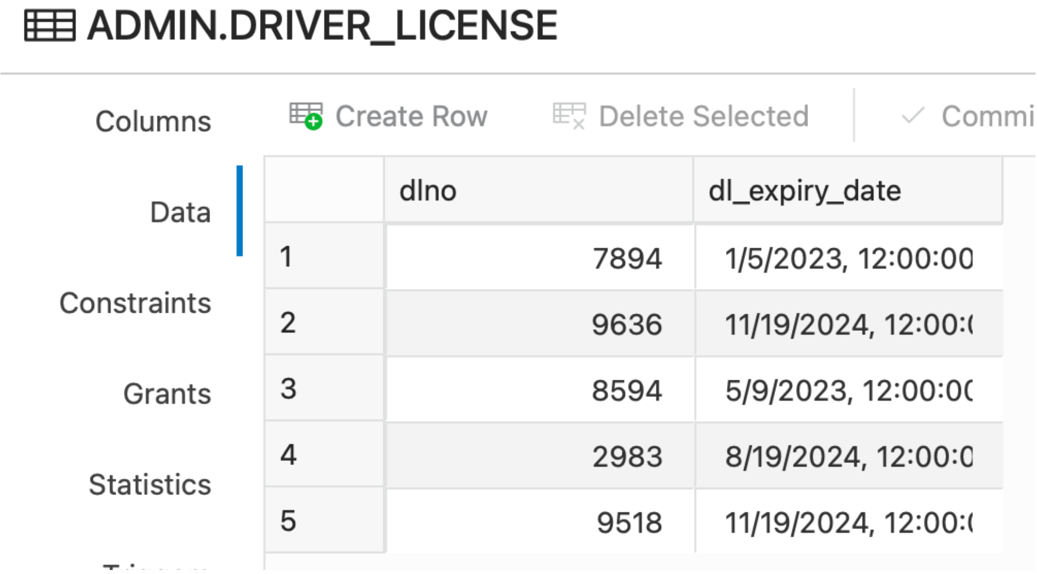
	uber_id	f_name	l_name	ph_no	email	street
1	1111	Jim	Williams	1324567890	jw@gmail.com	Coit
2	2222	Jim	Williams	1324567890	jw@gmail.com	Coit
3	3333	Jim	Williams	1324567890	jw@gmail.com	Coit
4	4444	Jim	Williams	1324567890	jw@gmail.com	Coit
5	5555	Jim	Williams	1324567890	jw@gmail.com	Coit
6	147	Jim	Williams	1324567890	jw@gmail.com	Coit
7	4656	Julie	Swan	7418529630	js@gmail.com	Frankford
8	8520	Joey	Buffay	2583691470	jb@gmail.com	Campbell
9	4928	John	Smith	3216549870	js@gmail.com	Ricky
10	2598	Jack	Chandler	3698521479	jc@gmail.com	Frankford
11	4859	George	Matt	2581472580	gm@gmail.com	Campbell
12	2693	Rachel	Black	7946138520	rb@gmail.com	Frankford
13	1234	Jaden	Will	3690926498	jw@gmail.com	Campbell
14	9685	John	Corner	4679792585	jc@gmail.com	Coit
15	1948	Megan	Fox	5948592636	mf@gmail.com	Campbell

DRIVER:



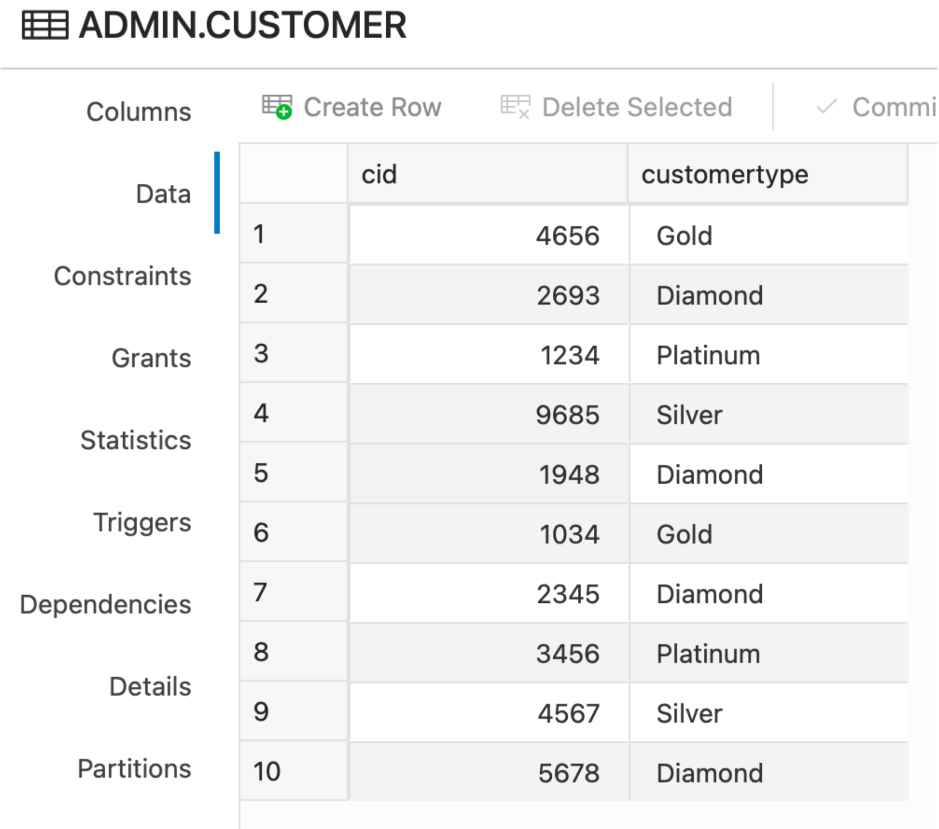
	did	ssn	dlno
1	147	7418529639	7894
2	8520	7946138596	9636
3	4928	5696897462	8594
4	2598	594875869	2983
5	4859	9685744589	9518
6	1111	9675744589	9076
7	2222	6985744589	9556
8	3333	9685744589	3493
9	4444	9634744589	2310
10	5555	7985744589	9012

DRIVER_LICENSE:



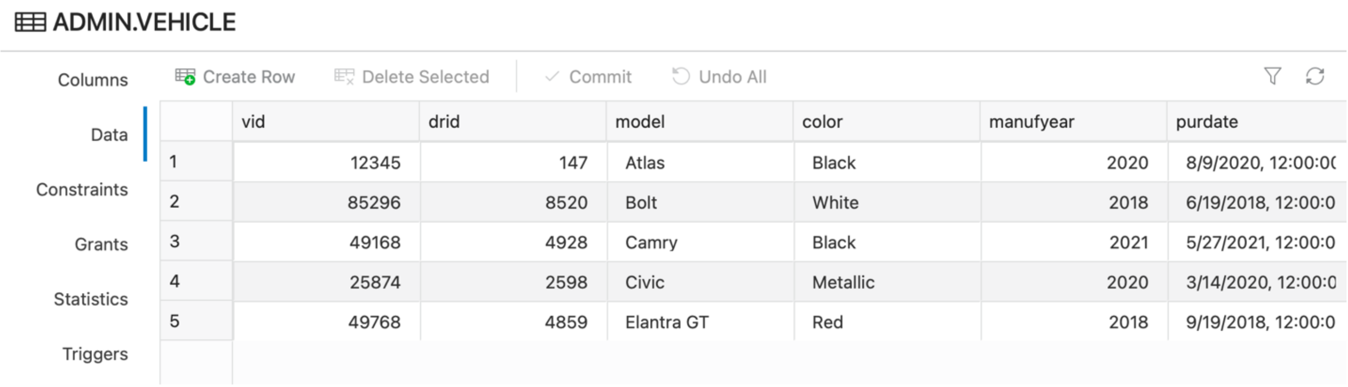
	dlno	dl_expiry_date
1	7894	1/5/2023, 12:00:00
2	9636	11/19/2024, 12:00:
3	8594	5/9/2023, 12:00:0
4	2983	8/19/2024, 12:00:0
5	9518	11/19/2024, 12:00:

CUSTOMER:



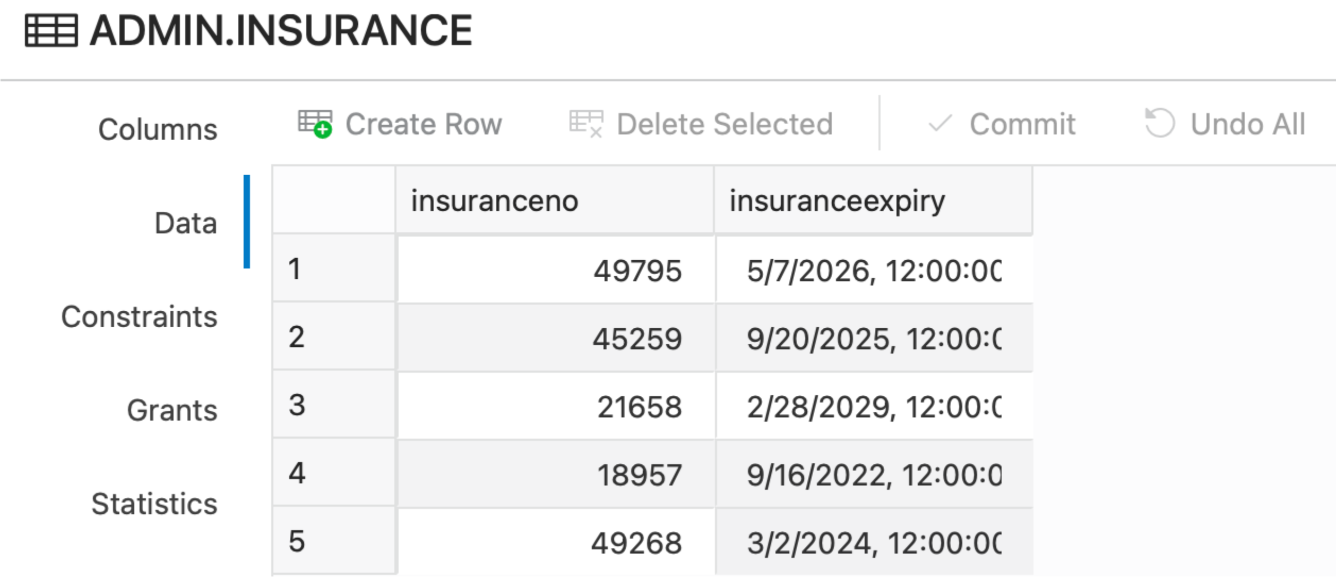
	cid	customertype
1	4656	Gold
2	2693	Diamond
3	1234	Platinum
4	9685	Silver
5	1948	Diamond
6	1034	Gold
7	2345	Diamond
8	3456	Platinum
9	4567	Silver
10	5678	Diamond

VEHICLE:



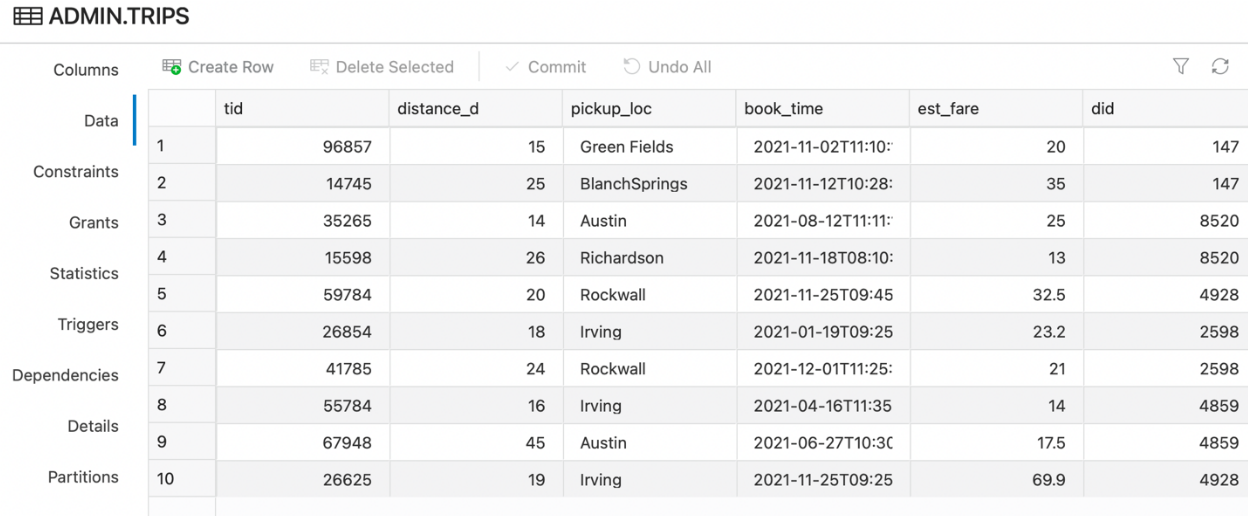
	vid	drid	model	color	manufyear	purdate
1	12345	147	Atlas	Black	2020	8/9/2020, 12:00:0
2	85296	8520	Bolt	White	2018	6/19/2018, 12:00:0
3	49168	4928	Camry	Black	2021	5/27/2021, 12:00:0
4	25874	2598	Civic	Metallic	2020	3/14/2020, 12:00:0
5	49768	4859	Elantra GT	Red	2018	9/19/2018, 12:00:0

INSURANCE:



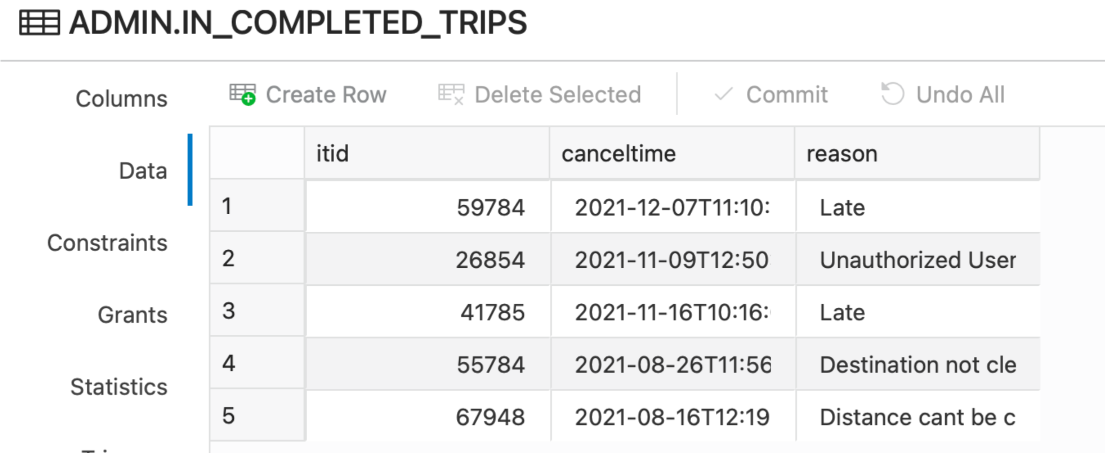
	insuranceno	insuranceexpiry
1	49795	5/7/2026, 12:00:0
2	45259	9/20/2025, 12:00:
3	21658	2/28/2029, 12:00:
4	18957	9/16/2022, 12:00:0
5	49268	3/2/2024, 12:00:0

TRIPS:



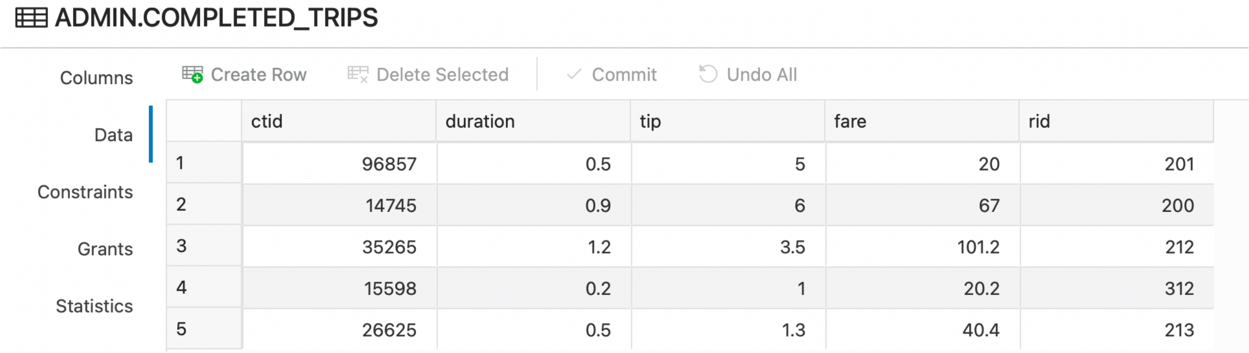
	tid	distance_d	pickup_loc	book_time	est_fare	did
1	96857	15	Green Fields	2021-11-02T11:10:	20	147
2	14745	25	BlanchSprings	2021-11-12T10:28:	35	147
3	35265	14	Austin	2021-08-12T11:11:	25	8520
4	15598	26	Richardson	2021-11-18T08:10:	13	8520
5	59784	20	Rockwall	2021-11-25T09:45	32.5	4928
6	26854	18	Irving	2021-01-19T09:25	23.2	2598
7	41785	24	Rockwall	2021-12-01T11:25:	21	2598
8	55784	16	Irving	2021-04-16T11:35	14	4859
9	67948	45	Austin	2021-06-27T10:3	17.5	4859
10	26625	19	Irving	2021-11-25T09:25	69.9	4928

IN_COMPLETED_TRIPS:



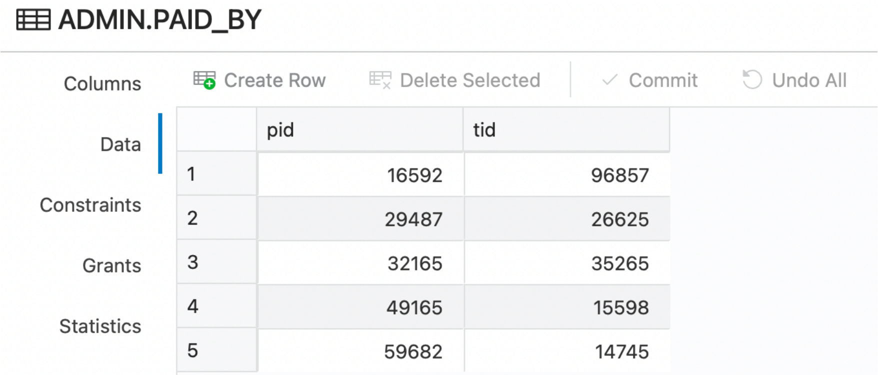
	itid	canceltime	reason
1	59784	2021-12-07T11:10:	Late
2	26854	2021-11-09T12:50	Unauthorized User
3	41785	2021-11-16T10:16:	Late
4	55784	2021-08-26T11:56	Destination not cle
5	67948	2021-08-16T12:19	Distance cant be c

COMPLETED_TRIPS:


ADMIN.COMPLETED_TRIPS

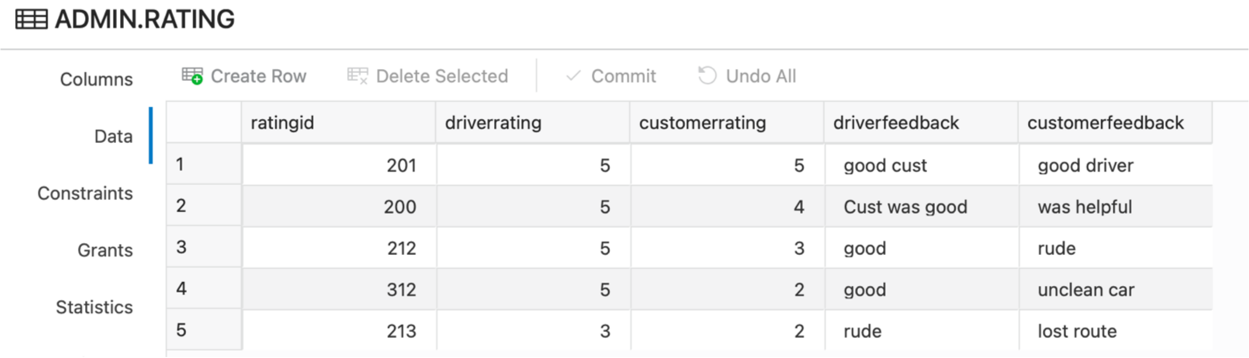
	ctid	duration	tip	fare	rid
1	96857	0.5	5	20	201
2	14745	0.9	6	67	200
3	35265	1.2	3.5	101.2	212
4	15598	0.2	1	20.2	312
5	26625	0.5	1.3	40.4	213

CUST_PAYMENT:


ADMIN.PAID_BY

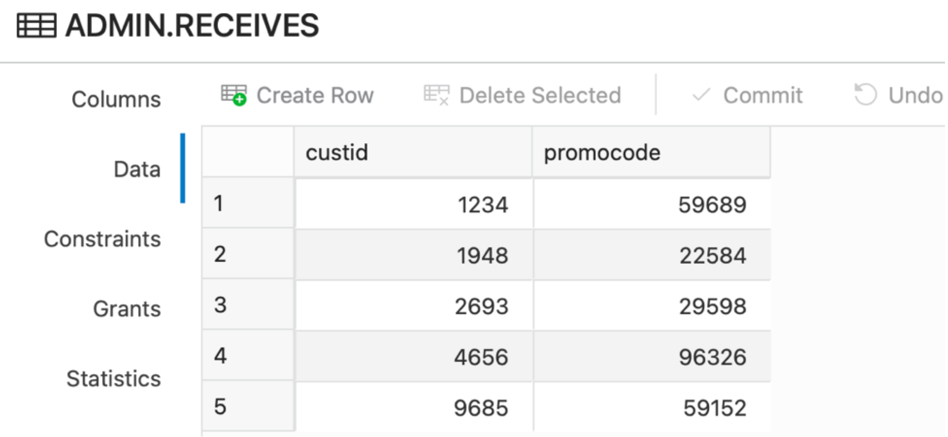
	pid	tid
1	16592	96857
2	29487	26625
3	32165	35265
4	49165	15598
5	59682	14745

RATING:


ADMIN.RATING

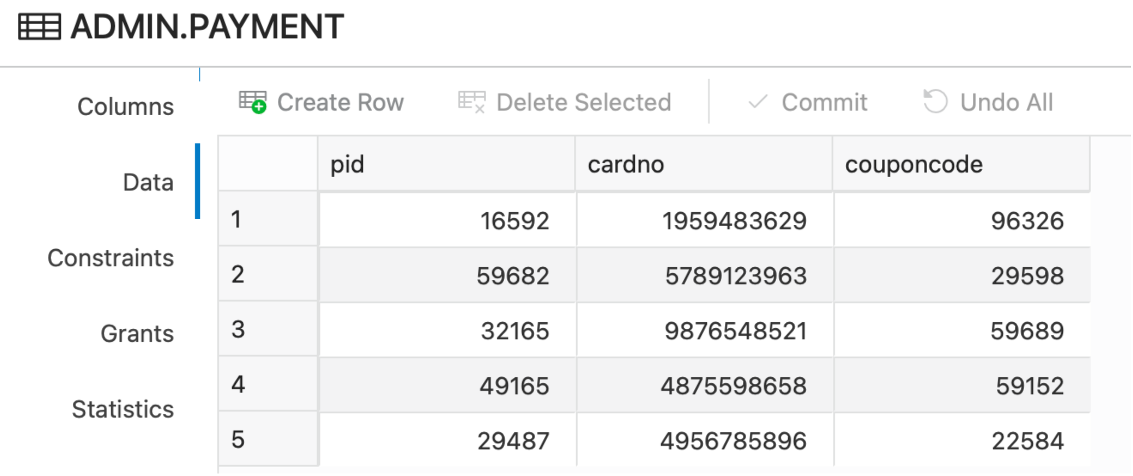
	ratingid	driverrating	customerrating	driverfeedback	customerfeedback
1	201	5	5	good cust	good driver
2	200	5	4	Cust was good	was helpful
3	212	5	3	good	rude
4	312	5	2	good	unclean car
5	213	3	2	rude	lost route

CUST_RECEIVES_PROMOCODE:


ADMIN.RECEIVES

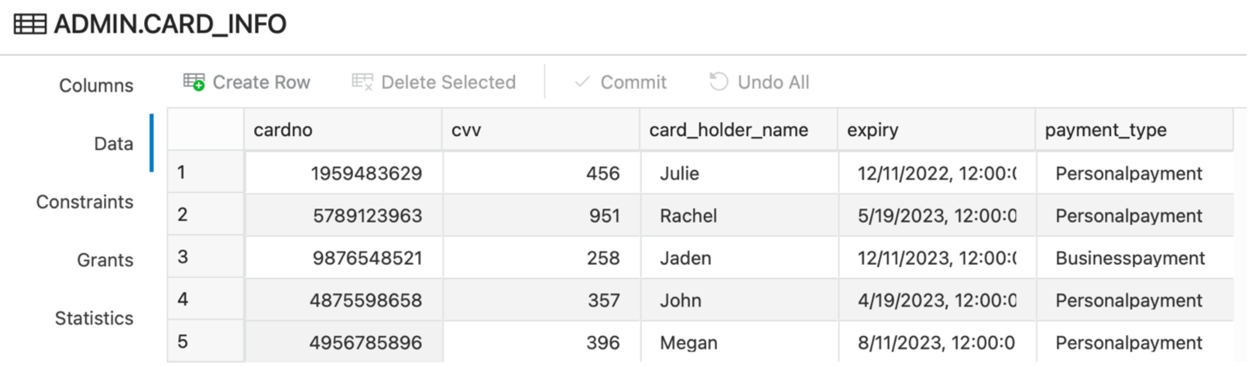
	custid	promocode
1	1234	59689
2	1948	22584
3	2693	29598
4	4656	96326
5	9685	59152

PAYMENT:


ADMIN.PAYMENT

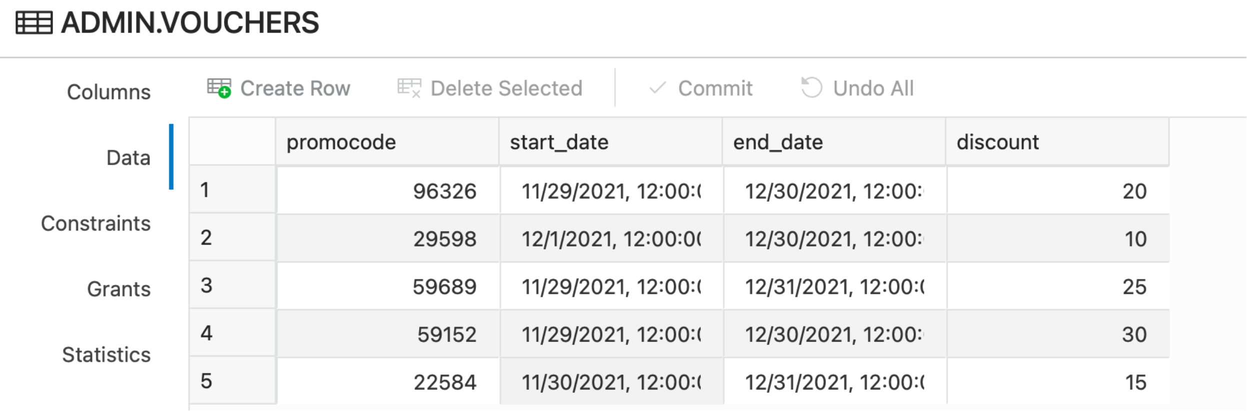
	pid	cardno	couponcode
1	16592	1959483629	96326
2	59682	5789123963	29598
3	32165	9876548521	59689
4	49165	4875598658	59152
5	29487	4956785896	22584

CARD_INFO:


ADMIN.CARD_INFO

	cardno	cvv	card_holder_name	expiry	payment_type
1	1959483629	456	Julie	12/11/2022, 12:00:0	Personalpayment
2	5789123963	951	Rachel	5/19/2023, 12:00:0	Personalpayment
3	9876548521	258	Jaden	12/11/2023, 12:00:0	Businesspayment
4	4875598658	357	John	4/19/2023, 12:00:0	Personalpayment
5	4956785896	396	Megan	8/11/2023, 12:00:0	Personalpayment

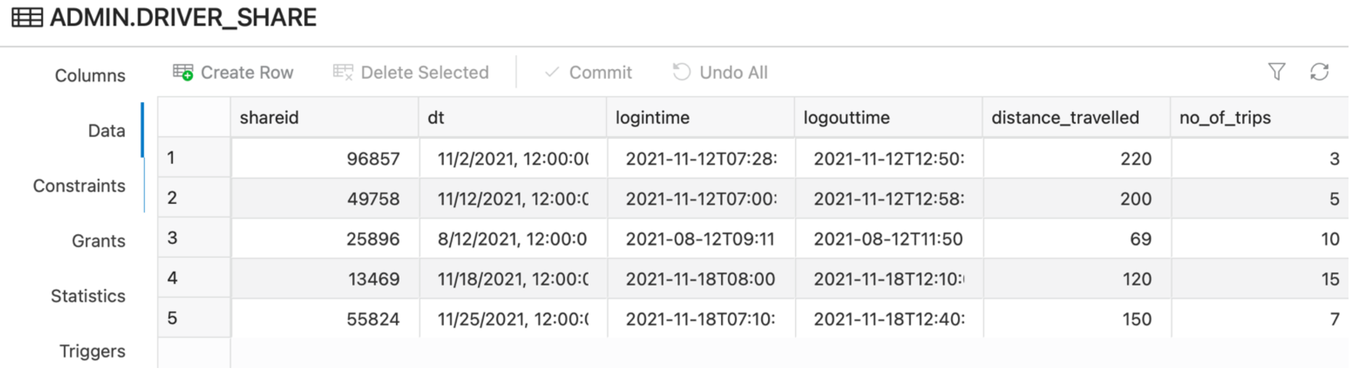
VOUCHERS:



ADMIN.VOUCHERS

	promocode	start_date	end_date	discount
1	96326	11/29/2021, 12:00:0	12/30/2021, 12:00:	20
2	29598	12/1/2021, 12:00:0	12/30/2021, 12:00:	10
3	59689	11/29/2021, 12:00:0	12/31/2021, 12:00:0	25
4	59152	11/29/2021, 12:00:0	12/30/2021, 12:00:	30
5	22584	11/30/2021, 12:00:0	12/31/2021, 12:00:0	15

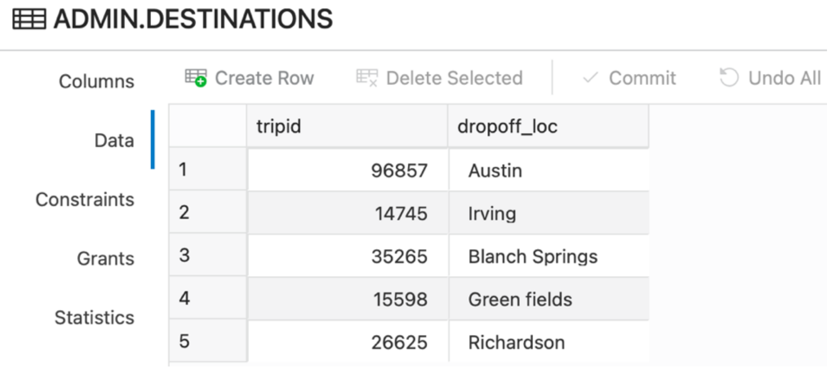
DRIVER_SHARE:



ADMIN.DRIVER_SHARE

	shareid	dt	logintime	logouttime	distance_travelled	no_of_trips
1	96857	11/2/2021, 12:00:0	2021-11-12T07:28:	2021-11-12T12:50:	220	3
2	49758	11/12/2021, 12:00:0	2021-11-12T07:00:	2021-11-12T12:58:	200	5
3	25896	8/12/2021, 12:00:0	2021-08-12T09:11	2021-08-12T11:50	69	10
4	13469	11/18/2021, 12:00:0	2021-11-18T08:00	2021-11-18T12:10:	120	15
5	55824	11/25/2021, 12:00:0	2021-11-18T07:10:	2021-11-18T12:40:	150	7

DESTINATIONS:



ADMIN.DESTINATIONS

	tripid	dropoff_loc
1	96857	Austin
2	14745	Irving
3	35265	Blanch Springs
4	15598	Green fields
5	26625	Richardson

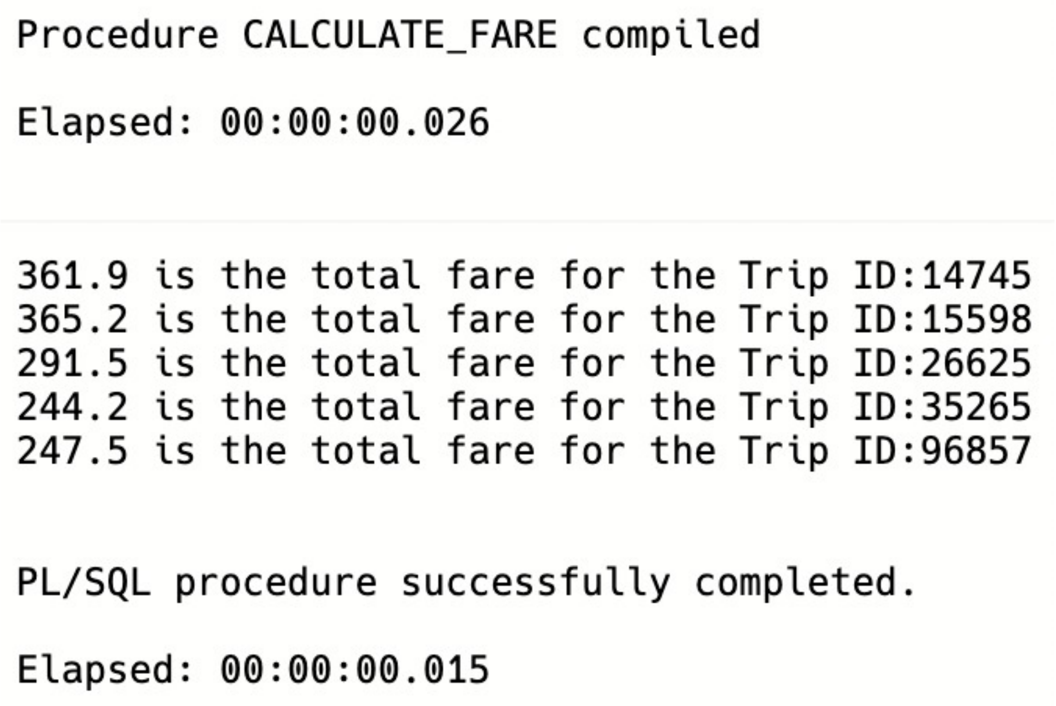
PL/SQL:

Executing STORED PROCEDURES on ORACLE SQL DEVELOPER:

1. **Stored Procedure to calculate TOTAL FARE of a given ride:**

```
create or replace PROCEDURE Calculate_Fare(Base_fare IN
number, Cost_per_mile IN number, Cost_per_min IN
number, Surge IN number,Tax IN number) AS

CURSOR Trip_total_fare IS
SELECT C.CTID as TID,C.Duration as
duration,t.Distance_D as distance
FROM COMPLETED_TRIPS C, Trips t
WHERE t.TID=C.CTID;
thisTrip Trip_total_fare%rowtype;
thisTotalFare TRIPS.EST_FARE%Type;

BEGIN
OPEN Trip_total_fare;
LOOP
FETCH Trip_total_fare INTO thisTrip;
EXIT WHEN (Trip_total_fare%NOTFOUND);
thisTotalFare:= (Base_fare + Tax +
Cost_per_mile*thisTrip.distance +
Cost_per_min*thisTrip.duration )*(1 +Surge);
dbms_output.put_line(thisTotalFare || ' is the total
fare for the Trip ID:' || thisTrip.TID); END LOOP;
CLOSE Trip_total_fare;
END;

Begin
Calculate_Fare(5,1,1,10,2);
End;
```

OUTPUT:



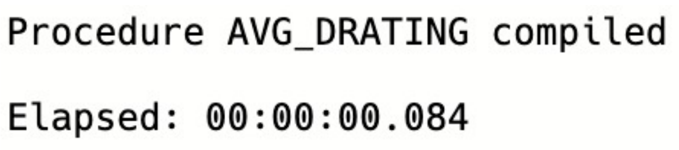
```
Procedure CALCULATE_FARE compiled

Elapsed: 00:00:00.026


361.9 is the total fare for the Trip ID:14745
365.2 is the total fare for the Trip ID:15598
291.5 is the total fare for the Trip ID:26625
244.2 is the total fare for the Trip ID:35265
247.5 is the total fare for the Trip ID:96857


PL/SQL procedure successfully completed.

Elapsed: 00:00:00.015
```

2. **Stored Procedure to calculate AVERAGE RATING of all drivers:**

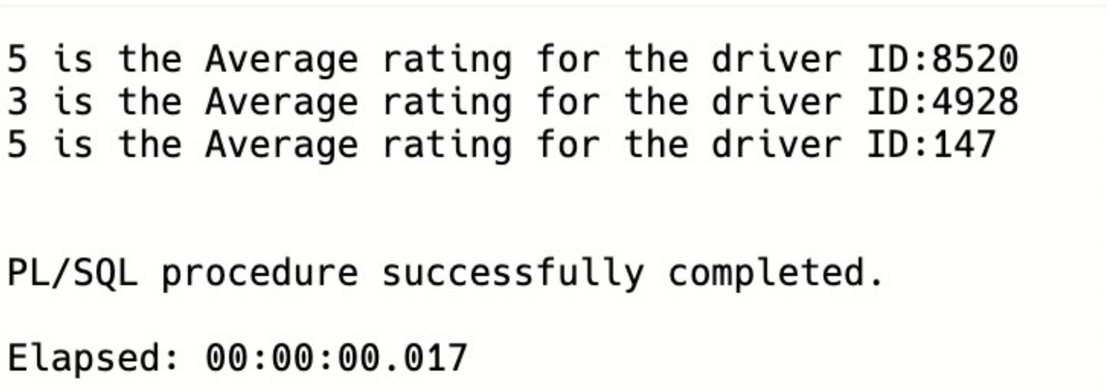
```
create or replace PROCEDURE Avg_DRating AS
CURSOR DrivRating IS SELECT AVG(R.Driverrating) as
AvgRating, D.DID FROM Rating R, COMPLETED_TRIPS C,
TRIPS T, DRIVER D WHERE R.RATINGID=C.RID AND
T.TID=C.CTID AND T.DID = D. DID
GROUP BY T.DID;
thisRating DrivRating%ROWTYPE;

BEGIN
OPEN DrivRating;
LOOP
FETCH DrivRating INTO thisRating;
EXIT WHEN (DrivRating%NOTFOUND);
dbms_output.put_line(thisRating.AvgRating || ' is the
Average rating for the driver ID:' || thisRating.DID);
END LOOP;
```

```
CLOSE DrivRating;
END;

begin
Avg_DRating;
End;
```

OUTPUT:



```
Procedure AVG_DRATING compiled

Elapsed: 00:00:00.084
```



```
5 is the Average rating for the driver ID:8520
3 is the Average rating for the driver ID:4928
5 is the Average rating for the driver ID:147


PL/SQL procedure successfully completed.

Elapsed: 00:00:00.017
```

Executing TRIGGERS on ORACLE SQL DEVELOPER:

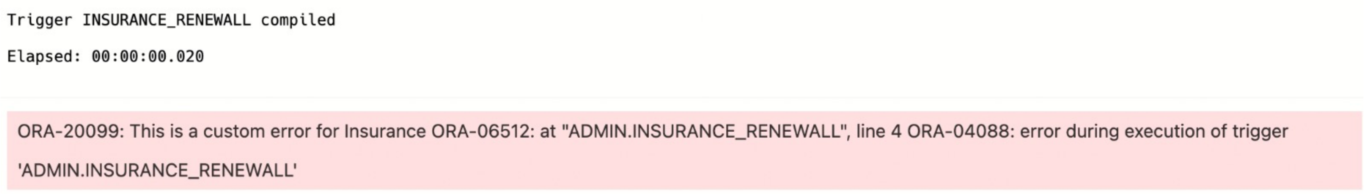
1. **Trigger to check that the INSURANCE for the vehicle must not be expired:**

```
create or replace TRIGGER Insurance_RenewalL before
insert or update
on INSURANCE for each row
Begin

if (:new.InsuranceExpiry < sysdate) then
raise_application_error( -20099, 'This is a custom
error for Insurance'); end if;
End;
```

QUERY:

```
Update INSURANCE set INSURANCEEXPIRY =
TO_DATE(‘2020-04-16’,’YYYY-MM-DD’) where
INSURANCENO=49795;
```

OUTPUT:



```
Trigger INSURANCE_RENEWALL compiled

Elapsed: 00:00:00.020
```

```
ORA-20099: This is a custom error for Insurance ORA-06512: at "ADMIN.INSURANCE_RENEWALL", line 4 ORA-04088: error during execution of trigger
'ADMIN.INSURANCE_RENEWALL'
```

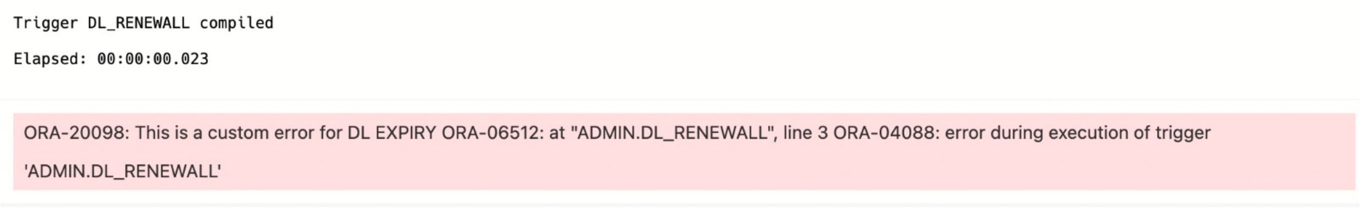
2. **Trigger to check that DRIVER’S LICENSE should not be expired:**

```
create or replace TRIGGER DL_Renewall before insert or
update
on DRIVER_LICENSE for each row
Begin
if (:new.DL_expiry_date < sysdate) then
raise_application_error( -20098, 'This is a custom
error for DL EXPIRY'); end if;
End;
```

QUERY:

```
update DRIVER_LICENSE set DL_EXPIRY_DATE =
TO_DATE('2020-04-16','YYYY-MM-DD') where DLNo= 2983;
```

OUTPUT:



```
Trigger DL_RENEWALL compiled

Elapsed: 00:00:00.023
```

```
ORA-20098: This is a custom error for DL EXPIRY ORA-06512: at "ADMIN.DL_RENEWALL", line 3 ORA-04088: error during execution of trigger
'ADMIN.DL_RENEWALL'
```