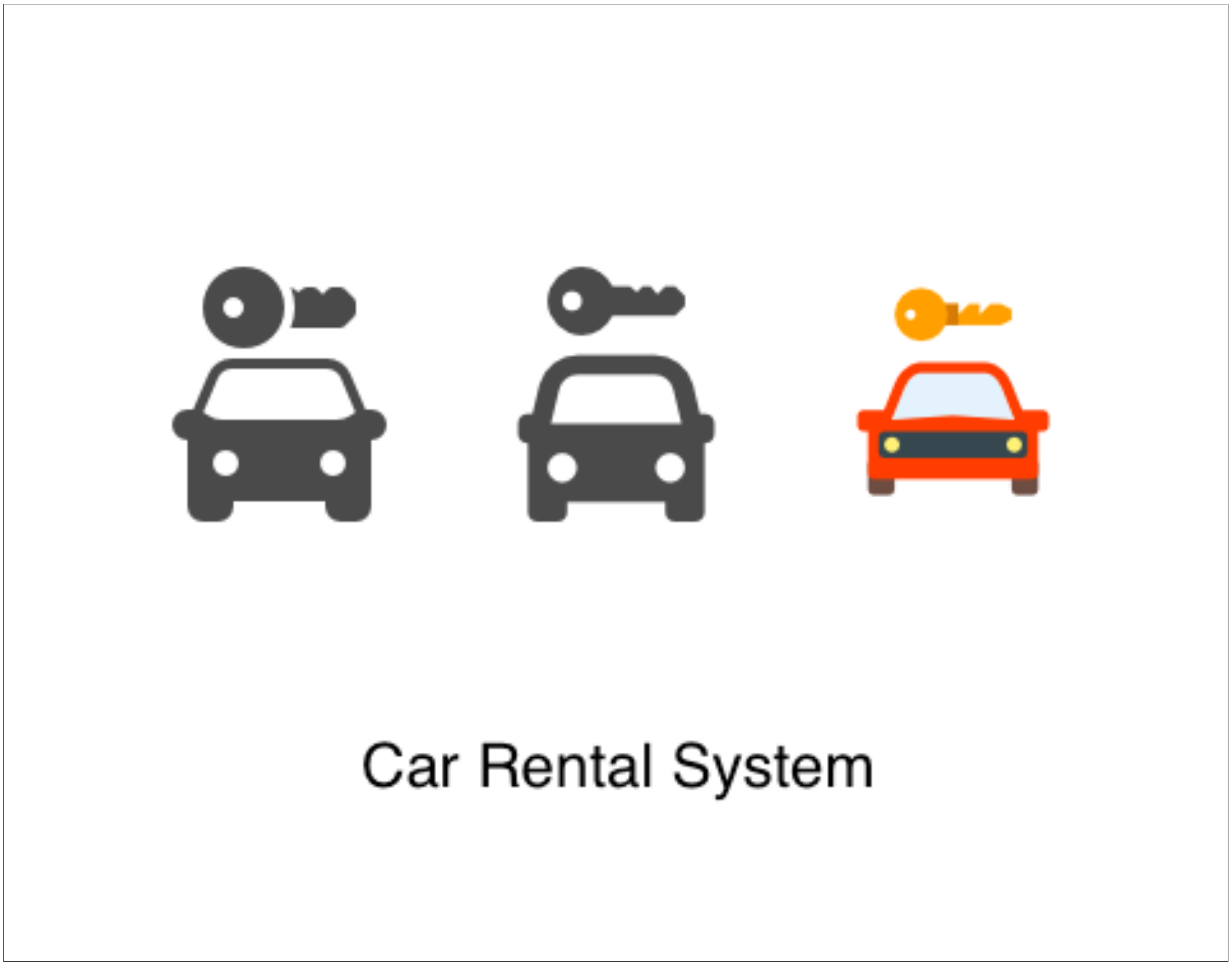
CAR RENTAL SYSTEM



Database Design Project

Prepared by : Soorya Prasanna Ravichandran (sxr152130)

Srikanth Kannan (sxk141931)

Subject Code : CS 6360.005

December 8, 2015

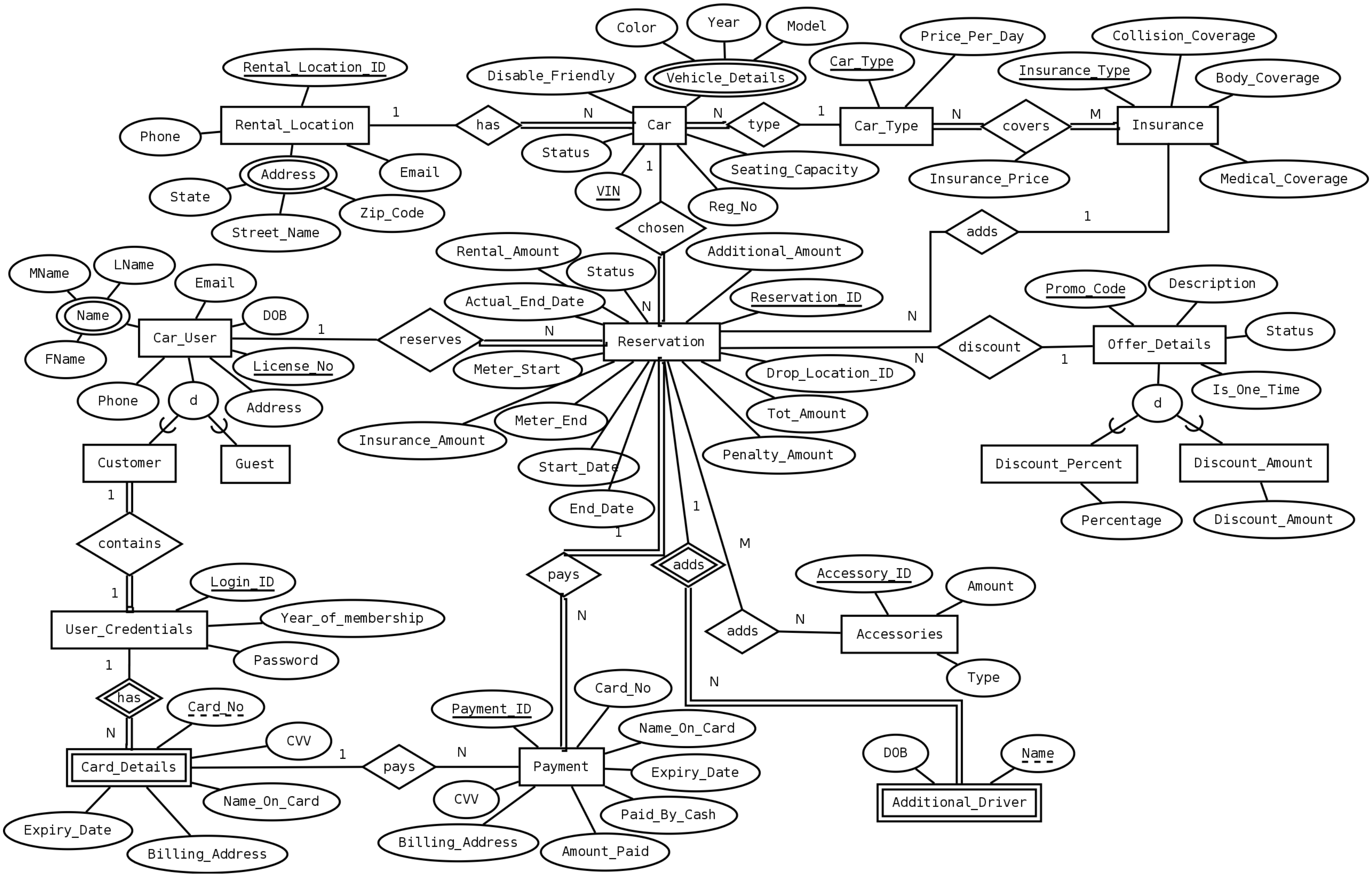
CONTENTS

DATA REQUIREMENTS

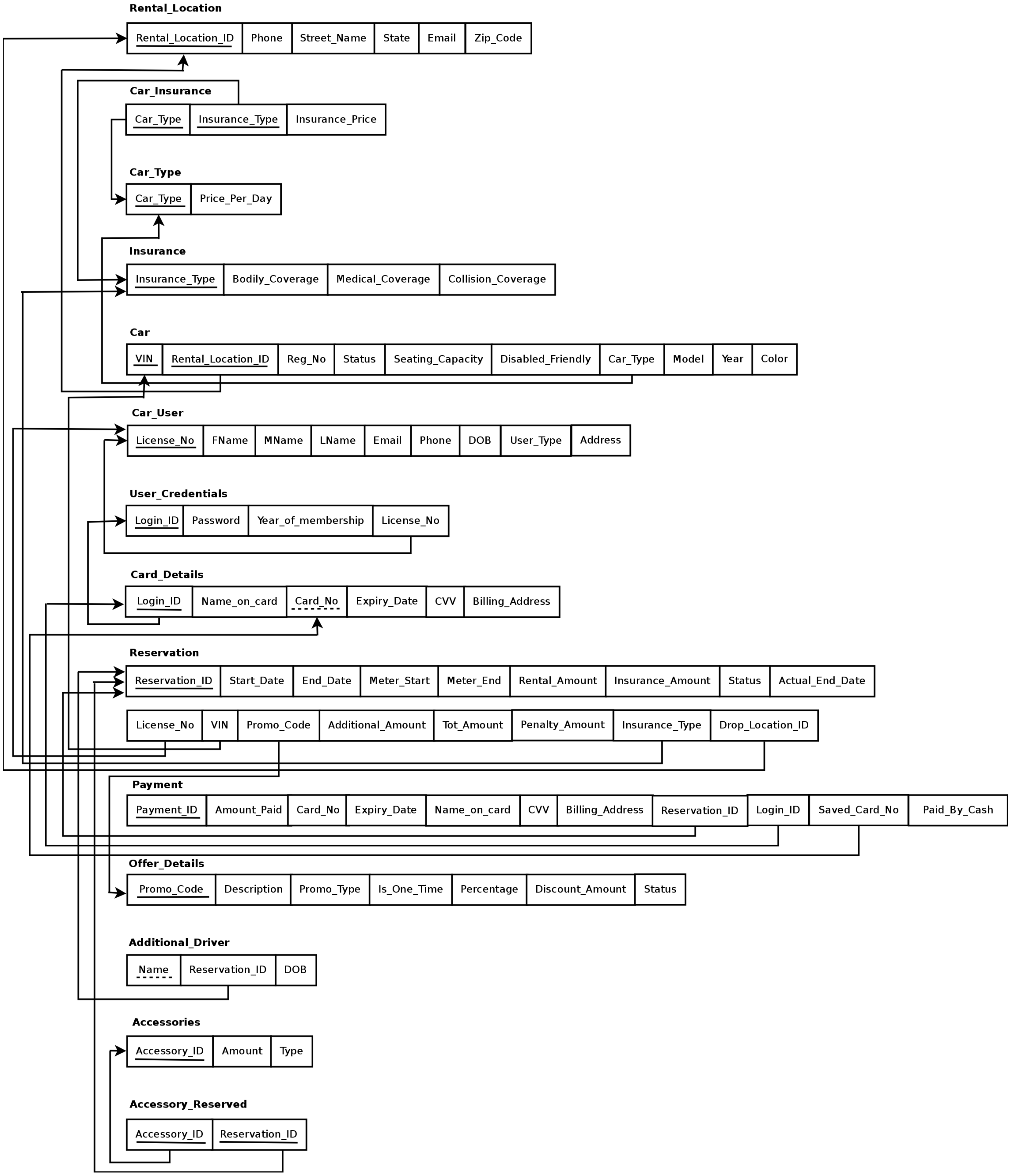
- The Car Rental Database needs to keep track of three main entities namely User, Car and Reservation.
- Car can be rented from a rental location with a specific address of each rental location. A rental location keeps track of contact phone number, contact email and rental location address with street name, state and zip code.
- A rental location has a number of cars for rental. Each car is described by VIN, Registration number, status as ‘Available’ and ‘Unavailable’, seating capacity of the car, whether it has disabled friendly feature, car color, car model and car manufactured year.
- Car has a parameter as Car Type. Car type can be ‘Economy’, ‘Standard’, ’SUV’, ‘Premium’ and ‘Minivan’. Car Type defines the rental price per day. A user can take Insurance per day for the rental car. There are different types of insurance each having different medical insurance coverage, collision coverage and car bodily coverage. Insurance Types are ‘Liability’ and ‘Comprehensive’. Car type and Insurance Type drives the Insurance price per day.
- A user can reserve a car for a number of days. He can use any valid promotional code which is maintained by status. When a user books a car he mentions the start date and end date for which he needs the car. The end date will be hypothetical at the time of reservation and updated with actual end date when the car is returned. If the Actual End Date is greater than End Date mentioned initially, then a penalty is imposed. The total amount is calculated based on start date, end date, rental price per day, insurance price per day and promotional code if any used.
- A user can cancel a reserved car before he/she rented the car. A reservation can have status as ‘Reserved’, ‘Completed’ and ‘Cancelled’. When the car is reserved, status will be in ‘Reserved’ Status. Once the car rental of the reservation is completed with due amount paid, the status will be ‘Completed’.
- A user can drop the car in any location. Car will be available for future use from the location where it is dropped.
- A user is categorized as guest and customer. User can continue reserving car as guest as long as he has not registered as customer. A user is uniquely identified by his/her license number. User information consists of his name as Fist name, middle name and last name, Email address, physical address, date of birth and contact number.
- A registered customer will be provided with login id and password. A customer can save his credit/ debit card details for future payment.

- Partial payment can be made at the time of reservation and the rest must be paid by the user during car return when actual end date is known. If user is a customer, he/she can pay through saved debit/credit card details. A user irrespective of whether guest or customer has an option to pay by cash.
- A user can add any accessories as part of his/her reservation. Accessories can be ‘Car Seat’ and ‘GPS’. A user can add as many accessories he/she needs as part of the reservation.
- Additional driver can be added as part of his/her reservation. For each additional driver there will be an additional charges.

ENTITY RELATIONSHIP DIAGRAM



MAPPING



Rental_Location

Rental_Location_ID	Phone	Street_Name	State	Email	Zip_Code

Car_Insurance

Car_Type	Insurance_Type	Insurance_Price

Car_Type

Car_Type	Price_Per_Day

Insurance

Insurance_Type	Bodily_Coverage	Medical_Coverage	Collision_Coverage

Car

VIN	Rental_Location_ID	Reg_No	Status	Seating_Capacity	Disabled_Friendly	Car_Type	Model	Year	Color

Car_User

License_No	FName	MName	LName	Email	Phone	DOB	User_Type	Address

User_Credentials

Login_ID	Password	Year_of_membership	License_No

Card_Details

Login_ID	Name_on_card	Card_No	Expiry_Date	CVV	Billing_Address

Reservation

Reservation_ID	Start_Date	End_Date	Meter_Start	Meter_End	Rental_Amount	Insurance_Amount	Status	Actual_End_Date

License_No	VIN	Promo_Code	Additional_Amount	Tot_Amount	Penalty_Amount	Insurance_Type	Drop_Location_ID

Payment

Payment_ID	Amount_Paid	Card_No	Expiry_Date	Name_on_card	CVV	Billing_Address	Reservation_ID	Login_ID	Saved_Card_No	Paid_By_Cash

Offer_Details

Promo_Code	Description	Promo_Type	Is_One_Time	Percentage	Discount_Amount	Status

Additional_Driver

Name	Reservation_ID	DOB

Accessories

Accessory_ID	Amount	Type

Accessory_Reserved

Accessory_ID	Reservation_ID

FUNCTIONAL DEPENDENCIES

- In Rental car location, Rental_Location_ID is the primary key

  Rental_Location_ID —> {Phone, Email, Street_Name, State, Zip_Code}
- Type of the car defines the rental price of the car per day

  Car_Type —> Price_Per_Day
- Type of the insurance defines the insurance coverage

  Insurance_Type —> {Bodily_Coverage, Medical_Coverage, Collision_Coverage}
- Insurance Type and Car Type defines the Insurance price per day

  {Car_Type, Insurance_Type} —> {Insurance_Price}
- A user is defined by his/her License_No

  {License_No} —> {FName, Mname, Lname, Email, Address, Phone, DOB, User_Type}
- In a user credential, Login_ID defines the rest of the attributes in the entity

  {Login_ID} —> {Password, Year_Of_Membership, License_No}
- Login_ID and Card_No in Card_Details defines complete card information

  {Login_ID, Card_No} —> {Name_On_Card, Expiry_Date, CVV, Billing_Address}
- Reservation_ID drives all the other attributes in Reservation relation

  Reservation_ID —> {Start_Date, End_Date, Meter_Start, Meter_End, Rental_Amount, Insurance_Amount, Actual_End_Date, Status, License_No, VIN, Promo_Code, Additional_Amount, Tot_Amount, Insurance_Type, Penalty_Amount, Drop_Location_ID}
- Payment_ID is the primary key of Payment relation

  Payment_ID —> {Amount_Paid, Card_No, Expiry_Date, Name_On_Card, CVV, Billing_Address, Reservation_ID, Login_ID, Saved_Card_No, Paid_By_Cash}
- Promo_Code defines other attributes in Offer_Details relation

  Promo_Code —> {Description, Promo_Type, Is_One_Time, Percentage, Discounted_Amount, Status}
- Additional_Drivers relation

  {Reservation_ID,Name} —> DOB
- In Accessories relation, Accessory_ID drives the type and amount of the accessory

  Accessory_ID —> {Type, Amount}

Note: Since our mapping was already in 3NF, we did not decompose it further

SQL STATEMENTS

```
--------------------------------------------
-- Start of Coding
--------------------------------------------
--------------------------------------------
-- DROP STATEMENTS
--------------------------------------------
drop table RENTAL_LOCATION cascade constraints purge;
drop table CAR_TYPE cascade constraints purge;
drop table INSURANCE cascade constraints purge;
drop table CAR_INSURANCE cascade constraints purge;
drop table CAR_USER cascade constraints purge;
drop table USER_CREDENTIALS cascade constraints purge;
drop table CARD_DETAILS cascade constraints purge;
drop table CAR cascade constraints purge;
drop table RESERVATION cascade constraints purge;
drop table PAYMENT cascade constraints purge;
drop table OFFER_DETAILS cascade constraints purge;
drop table ADDITIONAL_DRIVER cascade constraints purge;
drop table ACCESSORIES cascade constraints purge;
drop table ACCESSORY_RESERVED cascade constraints purge;
commit;
```

```
-------------------------------------------------
-- CREATE TABLE STATEMENTS
-------------------------------------------------
CREATE TABLE RENTAL_LOCATION
(
  Rental_Location_ID INT PRIMARY KEY,
  Phone CHAR(10) NOT NULL,
  Email VARCHAR(25),
  Street_Name VARCHAR(40) NOT NULL,
  State CHAR(2) NOT NULL,
  Zip_Code CHAR(6) NOT NULL
);

CREATE TABLE CAR_TYPE
(
  Car_Type VARCHAR(15) PRIMARY KEY,
  Price_Per_Day NUMBER(8,2) NOT NULL
);

CREATE TABLE INSURANCE
(
  Insurance_Type VARCHAR(15) PRIMARY KEY,
  Bodily_Coverage NUMBER(8,2) NOT NULL,
  Medical_Coverage NUMBER(8,2) NOT NULL,
  Collision_Coverage NUMBER(8,2) NOT NULL
);
```

```
CREATE TABLE CAR_INSURANCE
(
  Car_Type VARCHAR(15),
  Insurance_Type VARCHAR(15),
  Insurance_Price NUMBER(8,2) NOT NULL,
  PRIMARY KEY(Car_Type,Insurance_Type),
  CONSTRAINT CARTYPEFK
  FOREIGN KEY (Car_Type) REFERENCES CAR_TYPE(Car_Type)
              ON DELETE CASCADE,
  CONSTRAINT INSURANCETYPEFK
  FOREIGN KEY (Insurance_Type) REFERENCES INSURANCE(Insurance_Type)
              ON DELETE CASCADE
);

CREATE TABLE CAR_USER
(
  License_No VARCHAR(15) PRIMARY KEY,
  Fname VARCHAR(15) NOT NULL,
  Mname VARCHAR(1),
  Lname VARCHAR(15) NOT NULL,
  Email VARCHAR(25) NOT NULL UNIQUE,
  Address VARCHAR(100) NOT NULL,
  Phone CHAR(10) NOT NULL,
  DOB DATE NOT NULL,
  User_Type VARCHAR(10) NOT NULL
);
```

```
CREATE TABLE USER_CREDENTIALS
(
  Login_ID VARCHAR(15) PRIMARY KEY,
  Password VARCHAR(15) NOT NULL,
  Year_Of_Membership Char(4) NOT NULL ,
  License_No VARCHAR(15) NOT NULL,
  CONSTRAINT USRLIC
  FOREIGN KEY (License_No) REFERENCES CAR_USER(License_No)
              ON DELETE CASCADE
);

CREATE TABLE CARD_DETAILS
(
  Login_ID VARCHAR(15) NOT NULL,
  Name_On_Card VARCHAR(50) NOT NULL,
  Card_No CHAR(16) NOT NULL,
  Expiry_Date DATE NOT NULL,
  CVV CHAR(3) NOT NULL,
  Billing_Address VARCHAR(50) NOT NULL,
  PRIMARY KEY(Login_ID,Card_No),
  CONSTRAINT USRCARDFK
  FOREIGN KEY (Login_ID) REFERENCES USER_CREDENTIALS(Login_ID)
              ON DELETE CASCADE
);
```

```
CREATE TABLE CAR
(
  VIN CHAR(17) PRIMARY KEY,
  Rental_Location_ID INT NOT NULL,
  Reg_No VARCHAR(15) UNIQUE,
  Status VARCHAR(15) NOT NULL,
  Seating_Capacity INT NOT NULL,
  Disability_Friendly CHAR(1),
  Car_Type VARCHAR(15) NOT NULL,
  Model VARCHAR(20),
  Year CHAR(4),
  Color VARCHAR(10),
  CONSTRAINT CARVINTYPEFK
  FOREIGN KEY (Car_Type) REFERENCES CAR_TYPE(Car_Type)
              ON DELETE CASCADE,
  CONSTRAINT CARVINRENTALFK
  FOREIGN KEY (Rental_Location_ID) REFERENCES RENTAL_LOCATION(Rental_Location_ID)
              ON DELETE CASCADE
);

CREATE TABLE OFFER_DETAILS
(
  Promo_Code VARCHAR(15) PRIMARY KEY,
  Description VARCHAR(50),
  Promo_Type VARCHAR(20) NOT NULL,
  Is_One_Time CHAR(1),
  Percentage DECIMAL(5,2),
  Discounted_Amount NUMBER(8,2),
  Status VARCHAR(10) NOT NULL);
```

```
CREATE TABLE RESERVATION
(
  Reservation_ID INT PRIMARY KEY,
  Start_Date DATE NOT NULL,
  End_Date DATE NOT NULL,
  Meter_Start INT NOT NULL,
  Meter_End INT,
  Rental_Amount NUMBER(8,2) NOT NULL,
  Insurance_Amount NUMBER(8,2) NOT NULL,
  Actual_End_Date DATE NULL,
  Status VARCHAR(10) NOT NULL,
  License_No VARCHAR(15) NOT NULL,
  VIN CHAR(17) NOT NULL,
  Promo_Code VARCHAR(15),
  Additional_Amount NUMBER(8,2),
  Tot_Amount NUMBER(8,2) NOT NULL,
  Insurance_Type VARCHAR(15),
  Penalty_Amount NUMBER(8,2),
  Drop_Location_ID INT,
  CONSTRAINT RSERVLOCATIONFK
  FOREIGN KEY (Drop_Location_ID) REFERENCES RENTAL_LOCATION(Rental_Location_ID)
              ON DELETE CASCADE,
  CONSTRAINT RESLICENSEFK
  FOREIGN KEY (License_No) REFERENCES CAR_USER(License_No)
              ON DELETE CASCADE,
  CONSTRAINT VINRESERVATIONFK
  FOREIGN KEY (VIN) REFERENCES CAR(VIN)
              ON DELETE CASCADE,
  CONSTRAINT PROMORESERVATIONFK
  FOREIGN KEY (Promo_Code) REFERENCES OFFER_DETAILS(Promo_Code)
              ON DELETE CASCADE,
  CONSTRAINT INSURESERVATIONFK
  FOREIGN KEY (Insurance_Type) REFERENCES INSURANCE(Insurance_Type)
              ON DELETE CASCADE
);

CREATE TABLE PAYMENT
(
  Payment_ID INT PRIMARY KEY,
  Amount_Paid NUMBER(8,2) NOT NULL,
  Card_No CHAR(16),
  Expiry_Date DATE,
  Name_On_Card VARCHAR(50),
  CVV CHAR(3),
  Billing_Address VARCHAR(50),
  Reservation_ID INT NOT NULL,
  Login_ID VARCHAR(15),
  Saved_Card_No CHAR(16),
  Paid_By_Cash CHAR(1),
  CONSTRAINT PAYMENTRESERVATIONFK
  FOREIGN KEY (Reservation_ID) REFERENCES RESERVATION(Reservation_ID)
              ON DELETE CASCADE,
  CONSTRAINT PAYMENTLOGINFK
  FOREIGN KEY (Login_ID,Saved_Card_No) REFERENCES CARD_DETAILS(Login_ID,Card_No)
              ON DELETE CASCADE
);
CREATE TABLE ADDITIONAL_DRIVER
(
  Reservation_ID INT,
  NAME VARCHAR(50) NOT NULL,
  DOB DATE NOT NULL,
  PRIMARY KEY(Reservation_ID,NAME),
  CONSTRAINT ADDTIONALFK
  FOREIGN KEY (Reservation_ID) REFERENCES RESERVATION(Reservation_ID)
              ON DELETE CASCADE
);

CREATE TABLE ACCESSORIES
(
  Accessory_ID INT PRIMARY KEY,
  Type VARCHAR(15) NOT NULL,
  Amount NUMBER(8,2) NOT NULL
);

CREATE TABLE ACCESSORY_RESERVED
(
  Accessory_ID INT,
  Reservation_ID INT,
  PRIMARY KEY(Accessory_ID,Reservation_ID),
  CONSTRAINT ACCESSORYRESERVFK
  FOREIGN KEY (Reservation_ID) REFERENCES RESERVATION(Reservation_ID)
              ON DELETE CASCADE,
  CONSTRAINT ACCESSFK
  FOREIGN KEY (Accessory_ID) REFERENCES ACCESSORIES(Accessory_ID)
              ON DELETE CASCADE);commit;
```

```
-------------------------------------------------
-- INSERT STATEMENTS
-------------------------------------------------
INSERT ALL
INTO RENTAL_LOCATION
(Rental_Location_ID,Phone,Email,Street_Name,State,Zip_Code)
VALUES
(101,'9726031111','adams12@gmail.com','980 Addison Road, Dallas','TX',75123)
INTO RENTAL_LOCATION
(Rental_Location_ID,Phone,Email,Street_Name,State,Zip_Code)
VALUES
(102,'9726032222','bobw@gmail.com',' 111, Berlington Road, Dallas','TX',75243)
INTO RENTAL_LOCATION
(Rental_Location_ID,Phone,Email,Street_Name,State,Zip_Code)
VALUES
(103,'9721903121','patric.clever@gmail.com',' 9855 Shadow Way, Dallas','TX',
75211)
INTO RENTAL_LOCATION
(Rental_Location_ID,Phone,Email,Street_Name,State,Zip_Code)
VALUES
(104,'721903121',NULL,'434 Harrodswood Road, Irving','TX',76512)
INTO RENTAL_LOCATION
(Rental_Location_ID,Phone,Email,Street_Name,State,Zip_Code)
VALUES
(105,'5026981045','julier@gmail.com','7788 internal Drive, Irving','TX',77888)
SELECT * FROM Dual;
```

```
INSERT ALL
INTO CAR_TYPE
(Car_Type,Price_Per_Day)
VALUES
('Economy',19.95)
INTO CAR_TYPE
(Car_Type,Price_Per_Day)
VALUES
('Standard',29.95)
INTO CAR_TYPE
(Car_Type,Price_Per_Day)
VALUES
('SUV',89.95)
INTO CAR_TYPE
(Car_Type,Price_Per_Day)
VALUES
('MiniVan',109.95)
INTO CAR_TYPE
(Car_Type,Price_Per_Day)
VALUES
('Premium',149.95)
SELECT * FROM dual;

INSERT ALL
INTO INSURANCE
(Insurance_Type,Bodily_Coverage,Medical_Coverage,Collision_Coverage)
VALUES
('Liability',25000.00,50000.00,0.00)
INTO INSURANCE
```

```
(Insurance_Type,Bodily_Coverage,Medical_Coverage,Collision_Coverage)
VALUES
('Comprehensive',50000.00,50000.00,50000.00)
SELECT * FROM dual;

INSERT ALL
INTO CAR_INSURANCE
(Car_Type,Insurance_Type,Insurance_Price)
VALUES
('Economy','Liability',9.99)
INTO CAR_INSURANCE
(Car_Type,Insurance_Type,Insurance_Price)
VALUES
('Standard','Liability',10.99)
INTO CAR_INSURANCE
(Car_Type,Insurance_Type,Insurance_Price)
VALUES
('SUV','Liability',12.99)
INTO CAR_INSURANCE
(Car_Type,Insurance_Type,Insurance_Price)
VALUES
('MiniVan','Liability',14.99)
INTO CAR_INSURANCE
(Car_Type,Insurance_Type,Insurance_Price)
VALUES
('Premium','Liability',19.99)
INTO CAR_INSURANCE
(Car_Type,Insurance_Type,Insurance_Price)
VALUES
```

```
('Economy','Comprehensive',19.99)
INTO CAR_INSURANCE
(Car_Type,Insurance_Type,Insurance_Price)
VALUES
('Standard','Comprehensive',19.99)
INTO CAR_INSURANCE
(Car_Type,Insurance_Type,Insurance_Price)
VALUES
('SUV','Comprehensive',24.99)
INTO CAR_INSURANCE
(Car_Type,Insurance_Type,Insurance_Price)
VALUES
('MiniVan','Comprehensive',29.99)
INTO CAR_INSURANCE
(Car_Type,Insurance_Type,Insurance_Price)
VALUES
('Premium','Comprehensive',49.99)
SELECT * FROM dual;

INSERT ALL
INTO CAR_USER
(License_No,FName,MName,Lname,Email,Address,Phone,DOB,USER_TYPE)
VALUES
('E12905109','Patrick','G','Cleaver','patric.c@yahoo.com','1701 N.Campbell Rd,
Dallas, TX-75243','5022196058',TO_DATE('1970/01/10', 'yyyy/mm/dd'),'Guest')
INTO CAR_USER
(License_No,FNAME,MNAME,LNAME,Email,Address,Phone,DOB,USER_TYPE)
VALUES
('C11609103','Courtney',NULL,'Rollins','courtney.r@hotmail.com','1530 S.Campbell
Rd','4697891045',TO_DATE('1990/03/20', 'yyyy/mm/dd'),'Customer')
```

```
INTO CAR_USER

(License_No,FNAME,MNAME,LNAME,Email,Address,Phone,DOB,USER_TYPE)

VALUES

('G30921561','Glenn',NULL,'Tucker','glenn.t@hotmail.com','101 Meritline
drive','8590125607',TO_DATE('1964/11/11', 'yyyy/mm/dd'),'Customer')

INTO CAR_USER

(License_No,FNAME,MNAME,LNAME,Email,Address,Phone,DOB,USER_TYPE)

VALUES

('R12098127','Ron',NULL,'Harper','ron.harper@hotmail.com','43 Greenville
Road','2048015647',TO_DATE('1987/04/24', 'yyyy/mm/dd'),'Guest')

INTO CAR_USER

(License_No,FNAME,MNAME,LNAME,Email,Address,Phone,DOB,USER_TYPE)

VALUES

('M12098127','Manoj',NULL,'Punwani','manoj123@gmail.com','43 Greenville
Road','2048015647',TO_DATE('1987/04/24', 'yyyy/mm/dd'),'Customer')

SELECT * FROM dual;


INSERT ALL

INTO USER_CREDENTIALS

(Login_ID,Password,Year_Of_Membership,License_No)

VALUES

('courtney90','bc125ac','2009','C11609103')

INTO USER_CREDENTIALS

(Login_ID,Password,Year_Of_Membership,License_No)

VALUES

('glenn64','macpro99','2011','G30921561')

INTO USER_CREDENTIALS

(Login_ID,Password,Year_Of_Membership,License_No)

VALUES

('manoj87','windows99','2008','M12098127')
```

```
SELECT * FROM dual;

INSERT ALL
INTO CARD_DETAILS
(Login_ID,Name_On_Card,Card_No,Expiry_Date,CVV,Billing_Address)
VALUES
('courtney90','Courtney Rollins','4735111122223333',TO_DATE('2018/01/15', 'yyyy/mm/dd'),'833','1530 S.Campbell Rd, Dallas, TX 75251')
INTO CARD_DETAILS
(Login_ID,Name_On_Card,Card_No,Expiry_Date,CVV,Billing_Address)
VALUES
('manoj87','Manoj Punwani','4233908110921001',TO_DATE('2019/12/31', 'yyyy/mm/dd'),'419','9855 Shadow Way, TX 75243')
SELECT * FROM dual;

INSERT ALL
INTO CAR
(VIN,Rental_Location_ID,Reg_No,Status,Seating_Capacity,Disability_Friendly,Car_Type,Model,Year,Color)
VALUES
('F152206785240289',101,'TXF101','Available', 5,'N','Economy','Mazda3','2007','Gold')
INTO CAR
(VIN,Rental_Location_ID,Reg_No,Status,Seating_Capacity,Disability_Friendly,Car_Type,Model,Year,Color)
VALUES
('T201534710589051',101,'KYQ101','Available',5,'Y','Standard','Toyota Camry','2012','Grey')
INTO CAR
(VIN,Rental_Location_ID,Reg_No,Status,Seating_Capacity,Disability_Friendly,Car_Type,Model,Year,Color)
VALUES
```

```
('E902103289341098',102,'XYZ671','Available',
5,NULL,'Premium','BMW','2015','Black')

INTO CAR

(VIN,Rental_Location_ID,Reg_No,Status,Seating_Capacity,Disability_Friendly,Car_Type,Model,Year,Color)

VALUES

('R908891209418173',103,'DOP391','Unavailable',7,NULL,'SUV','Acura
MDX','2014','White')

INTO CAR

(VIN,Rental_Location_ID,Reg_No,Status,Seating_Capacity,Disability_Friendly,Car_Type,Model,Year,Color)

VALUES

('N892993994858292',104,'RAC829','Available',
15,NULL,'MiniVan','Sienna','2013','Black')

SELECT * FROM dual;


INSERT ALL

INTO OFFER_DETAILS

(PROMO_CODE,DESCRIPTION,PROMO_TYPE,IS_ONE_TIME,PERCENTAGE,DISCOUNTED_AMOUNT,Status)

VALUES

('CHRISTMAS10','Christmas 10% offer','Percentage','N',10.00,NULL,'Available')

INTO OFFER_DETAILS

(PROMO_CODE,DESCRIPTION,PROMO_TYPE,IS_ONE_TIME,PERCENTAGE,DISCOUNTED_AMOUNT,Status)

VALUES

('July25','July $25.00 discount','Discounted Amount','Y',NULL,25.00,'Expired')

INTO OFFER_DETAILS

(PROMO_CODE,DESCRIPTION,PROMO_TYPE,IS_ONE_TIME,PERCENTAGE,DISCOUNTED_AMOUNT,Status)

VALUES

('LaborDay5','Labor Day $5.00 offer','Discounted Amount','Y',NULL,5.00,'Expired')
```

```
INTO OFFER_DETAILS

(PROMO_CODE,DESCRIPTION,PROMO_TYPE,IS_ONE_TIME,PERCENTAGE,DISCOUNTED_AMOUNT,Statu
s)

VALUES

('NewYear10','New Year 10% offer','Percentage','N',10.00,NULL,'Available')

INTO OFFER_DETAILS

(PROMO_CODE,DESCRIPTION,PROMO_TYPE,IS_ONE_TIME,PERCENTAGE,DISCOUNTED_AMOUNT,Statu
s)

VALUES

('VeteranDay15','New Year 15% offer','Percentage','N',15.00,NULL,'Expired')

SELECT * FROM dual;


INSERT ALL

INTO RESERVATION

(Reservation_ID,Start_Date,End_Date,Meter_Start,Meter_End,Rental_Amount,Insurance
_Amount,Status,Actual_End_Date,License_No,VIN,Promo_Code,Additional_Amount,Tot_Am
ount,Penalty_Amount,Insurance_Type,Drop_Location_ID)

VALUES

(1,TO_DATE('2015/11/06', 'yyyy/mm/dd'),TO_DATE('2015/11/12', 'yyyy/mm/dd'),
81256,81300,119.70,9.95,'Completed',TO_DATE('2015/11/12', 'yyyy/mm/
dd'),'E12905109','F152206785240289',NULL,NULL,129.65,0.00,'Liability',101)

INTO RESERVATION

(Reservation_ID,Start_Date,End_Date,Meter_Start,Meter_End,Rental_Amount,Insurance
_Amount,Status,Actual_End_Date,License_No,VIN,Promo_Code,Additional_Amount,Tot_Am
ount,Penalty_Amount,Insurance_Type,Drop_Location_ID)

VALUES

(2,TO_DATE('2015/10/20', 'yyyy/mm/dd'),TO_DATE('2015/10/24', 'yyyy/mm/dd'),
76524,76590,119.80,9.95,'Completed',TO_DATE('2015/10/24', 'yyyy/mm/
dd'),'C11609103','T201534710589051',NULL,NULL,129.75,0.00,'Liability',101)

INTO RESERVATION

(Reservation_ID,Start_Date,End_Date,Meter_Start,Meter_End,Rental_Amount,Insurance
_Amount,Status,Actual_End_Date,License_No,VIN,Promo_Code,Additional_Amount,Tot_Am
ount,Penalty_Amount,Insurance_Type,Drop_Location_ID)

VALUES
```

```
(3,TO_DATE('2015/12/06', 'yyyy/mm/dd'),TO_DATE('2015/12/12', 'yyyy/mm/dd'),
82001,NULL,
659.40,29.95,'Reserved',NULL,'C11609103','N892993994858292','NewYear10',NULL,
689.35,0.00,'Comprehensive',104)

INTO RESERVATION

(Reservation_ID,Start_Date,End_Date,Meter_Start,Meter_End,Rental_Amount,Insurance
_Amount,Status,Actual_End_Date,License_No,VIN,Promo_Code,Additional_Amount,Tot_Am
ount,Penalty_Amount,Insurance_Type,Drop_Location_ID)

VALUES

(4,TO_DATE('2015/09/01', 'yyyy/mm/dd'),TO_DATE('2015/09/02', 'yyyy/mm/dd'),
51000,51100,89.95,24.95,'Completed',TO_DATE('2015/09/02', 'yyyy/mm/
dd'),'C11609103','R908891209418173',NULL,NULL,114.90,0.00,'Comprehensive',103)

INTO RESERVATION

(Reservation_ID,Start_Date,End_Date,Meter_Start,Meter_End,Rental_Amount,Insurance
_Amount,Status,Actual_End_Date,License_No,VIN,Promo_Code,Additional_Amount,Tot_Am
ount,Penalty_Amount,Insurance_Type,Drop_Location_ID)

VALUES

(5,TO_DATE('2015/08/13', 'yyyy/mm/dd'),TO_DATE('2015/08/15', 'yyyy/mm/dd'),
51000,51100,299.00,99.9,'Completed',TO_DATE('2015/08/15', 'yyyy/mm/
dd'),'R12098127','E902103289341098',NULL,NULL,398.90,0.00,'Comprehensive',105)

SELECT * FROM dual;


INSERT ALL

INTO PAYMENT

(Payment_ID,Amount_Paid,Card_NO,Expiry_Date,Name_On_Card,CVV,Billing_Address,Rese
rvation_ID,Login_ID,Saved_Card_No,Paid_By_Cash)

VALUES

(1001,129.65,'4735111122223333',TO_DATE('2018/01/15', 'yyyy/mm/dd'),'Patric
Clever','100','1530 S.Campbell Rd, Dallas, TX 75251',1,NULL,NULL,NULL)

INTO PAYMENT

(Payment_ID,Amount_Paid,Card_NO,Expiry_Date,Name_On_Card,CVV,Billing_Address,Rese
rvation_ID,Login_ID,Saved_Card_No,Paid_By_Cash)

VALUES

(1002,300.00,NULL,NULL,NULL,NULL,NULL,5,NULL,NULL,'Y')

INTO PAYMENT
```

```
(Payment_ID,Amount_Paid,Card_NO,Expiry_Date,Name_On_Card,CVV,Billing_Address,Reservation_ID,Login_ID,Saved_Card_No,Paid_By_Cash)
VALUES
(1003,98.90,NULL,NULL,NULL,NULL,NULL,5,NULL,NULL,'Y')
INTO PAYMENT
(Payment_ID,Amount_Paid,Card_NO,Expiry_Date,Name_On_Card,CVV,Billing_Address,Reservation_ID,Login_ID,Saved_Card_No,Paid_By_Cash)
VALUES
(1004,689.35,NULL,NULL,NULL,NULL,NULL,3,'courtney90','4735111122223333',NULL)
INTO PAYMENT
(Payment_ID,Amount_Paid,Card_NO,Expiry_Date,Name_On_Card,CVV,Billing_Address,Reservation_ID,Login_ID,Saved_Card_No,Paid_By_Cash)
VALUES
(1005,114.91,NULL,NULL,NULL,NULL,NULL,4,NULL,NULL,'Y')
SELECT * FROM dual;

INSERT ALL
INTO ADDITIONAL_DRIVER
(Reservation_ID,Name,DOB)
VALUES
(1,'William Smith',TO_DATE('1970/07/15', 'yyyy/mm/dd'))
INTO ADDITIONAL_DRIVER
(Reservation_ID,Name,DOB)
VALUES
(2,'Green Taylor',TO_DATE('1987/06/15', 'yyyy/mm/dd'))
INTO ADDITIONAL_DRIVER
(Reservation_ID,Name,DOB)
VALUES
(2,'Robert Moore',TO_DATE('1990/12/17', 'yyyy/mm/dd'))
INTO ADDITIONAL_DRIVER
```

```
(Reservation_ID,Name,DOB)
VALUES
(4,'Brad Cook',TO_DATE('1966/12/12', 'yyyy/mm/dd'))
INTO ADDITIONAL_DRIVER
(Reservation_ID,Name,DOB)
VALUES
(5,'Steve Fouts',TO_DATE('1976/05/28', 'yyyy/mm/dd'))
SELECT * FROM Dual;

INSERT ALL
INTO ACCESSORIES
(Accessory_ID,Type,Amount)
VALUES
(1,'GPS Navigation',49.95)
INTO ACCESSORIES
(Accessory_ID,Type,Amount)
VALUES
(2,'GPS Navigation',49.95)
INTO ACCESSORIES
(Accessory_ID,Type,Amount)
VALUES
(3,'GPS Navigation',49.95)
INTO ACCESSORIES
(Accessory_ID,Type,Amount)
VALUES
(4,'Baby Seater',29.95)
INTO ACCESSORIES
(Accessory_ID,Type,Amount)
VALUES
```

```
(5,'Baby Seater',29.95)
SELECT * FROM dual;

INSERT ALL
INTO ACCESSORY_RESERVED
(Accessory_ID,Reservation_ID)
VALUES
(1,1)
INTO ACCESSORY_RESERVED
(Accessory_ID,Reservation_ID)
VALUES
(1,4)
INTO ACCESSORY_RESERVED
(Accessory_ID,Reservation_ID)
VALUES
(5,5)
INTO ACCESSORY_RESERVED
(Accessory_ID,Reservation_ID)
VALUES
(5,2)
INTO ACCESSORY_RESERVED
(Accessory_ID,Reservation_ID)
VALUES
(2,4)
SELECT * FROM dual;
commit;
```

```
-------------------------------------------------
-- PL/SQL
-------------------------------------------------
```

/*

1. <u>UpdateStatus</u>

This procedure is called as daily batch process (preferably a nightly automatic batch process). The purpose of this procedure is, when the user has paid the total amount and credit is realized by respective credit card bank, following are the updates that need to happen. The reason it is better to be a nightly batch process is, usually a credit card transaction gets confirmed in mid-night and once it is confirmed, the application can insert a payment record based on the web service communication with credit card agency. Once Payment record is created in our car rental system following actions can be taken as part of the procedure.

a. Update the status of the Reservation record as ‘Completed’

b. Update the Status of the car as ‘Available’

c. Update the Rental_Location_ID of Car to the Drop_Location_ID of the reservation as the car will be available going forward from where it is dropped

d. Update the status of the promo_code if used, to ‘Expired’, if it is one time use promo_code

*/

```
-- 1. Update Reservation, Promo_Code and Car status with Drop Location ID
CREATE OR REPLACE PROCEDURE UpdateStatus AS
BEGIN
DECLARE
  thisReservation RESERVATION%ROWTYPE;
CURSOR reservationCursor IS
    SELECT R.* FROM RESERVATION R WHERE R.STATUS = 'Reserved' AND
    R.TOT_AMOUNT <= (SELECT SUM(AMOUNT_PAID) FROM PAYMENT P WHERE
P.RESERVATION_ID = R.RESERVATION_ID)
    FOR UPDATE OF STATUS;
BEGIN
  OPEN reservationCursor;
```

```
LOOP
  FETCH reservationCursor INTO thisReservation;
   EXIT WHEN (reservationCursor%NOTFOUND);
  -- Update Reservation Status
  UPDATE RESERVATION SET STATUS = 'Completed'
  WHERE CURRENT OF reservationCursor;
  -- Update Promo_Code Status
  UPDATE OFFER_DETAILS SET STATUS = 'Expired'
  WHERE PROMO_CODE = thisReservation.PROMO_CODE AND IS_ONE_TIME = 'Y';
  -- Update Rental_Location Of Car as Drop Location ID and Car Status
  UPDATE CAR SET RENTAL_LOCATION_ID = thisReservation.Drop_Location_ID,STATUS = 'Available'
  WHERE CAR.VIN = thisReservation.VIN;
END LOOP;
CLOSE reservationCursor;
END;
END UpdateStatus;

-- Call the procedure
SET SERVEROUTPUT ON
BEGIN
UpdateStatus;
END;
```

/*

2. IncreaseInsuranceRate

This procedure updates the insurance price of each car based on the percentage that is passed to the procedure. This procedure can be referred by the application, whenever there is a rise in car insurance rate

*/

```
-- 2. Increase the insurance price by given percentage
CREATE OR REPLACE PROCEDURE IncreaseInsuranceRate(percentage IN NUMBER) AS
BEGIN
DECLARE
  thisCarInsurance CAR_Insurance%ROWTYPE;
CURSOR carInsuranceCursor IS
    SELECT * FROM CAR_INSURANCE
    FOR UPDATE OF Insurance_Price;
BEGIN
  OPEN carInsuranceCursor;
LOOP
  FETCH carInsuranceCursor INTO thisCarInsurance;
   EXIT WHEN (carInsuranceCursor%NOTFOUND);

  UPDATE CAR_INSURANCE SET Insurance_Price = (Insurance_Price +
(Insurance_Price*percentage/100))
  WHERE CURRENT OF carInsuranceCursor;
END LOOP;
CLOSE carInsuranceCursor;
END;
END IncreaseInsuranceRate;
```

/*

3. *Print Reservations*

This PL/SQL prints the list of car that are reserved by the user, past end date and not yet returned

*/

```
-- 3. Print Reservations whose End Date is in the past and Car not yet returned

SET SERVEROUTPUT ON

DECLARE

  thisVIN CAR.VIN%TYPE;

  thisLicenseNo CAR_USER.License_No%TYPE;

  thisCarUser CAR_USER%ROWTYPE;

  CURSOR openReservationCursor IS

    SELECT VIN,License_No FROM RESERVATION WHERE STATUS = 'Reserved' AND END_DATE
< SYSDATE AND ACTUAL_END_DATE IS NULL;

BEGIN

  OPEN openReservationCursor;

LOOP

  FETCH openReservationCursor INTO thisVIN,thisLicenseNo;

   EXIT WHEN (openReservationCursor%NOTFOUND);

  SELECT * INTO thisCarUser FROM CAR_USER WHERE License_No = thisLicenseNo;

  dbms_output.put_line( 'User ' || thisCarUser.Fname || ' ' || thisCarUser.Mname
|| ' ' || thisCarUser.Lname || '  ' || 'whose email id is ' ||  thisCarUser.Email
||

  ' has not returned the car with VIN ' || thisVIN ||'.');

END LOOP;

  CLOSE openReservationCursor;

END;
```

/*

4. Print Promo Code

This PL/SQL prints the list of promo_codes which are NOT one time use, and the number of times those promo_codes are used

*/

```
-- 4. Print Promo code which are NOT of type One Time and number of times it is
used till now.

SET SERVEROUTPUT ON

DECLARE

  thispromoCode OFFER_DETAILS.Promo_Code%TYPE;

  thispromoCodeCount INT;

--  thisLicenseNo CAR_USER.License_No%TYPE;

--  thisCarUser CAR_USER%ROWTYPE;

  CURSOR openOfferDetailsCursor IS

    SELECT Promo_Code FROM OFFER_DETAILS WHERE Is_One_Time <> 'Y';

BEGIN

  OPEN openOfferDetailsCursor;

LOOP

  FETCH openOfferDetailsCursor INTO thispromoCode;

   EXIT WHEN (openOfferDetailsCursor%NOTFOUND);

  SELECT COUNT(*) INTO thispromoCodeCount FROM RESERVATION WHERE PROMO_CODE =
thispromoCode;

  dbms_output.put_line( 'Promo Code ' || thispromoCode || ' has been used ' ||
thispromoCodeCount || ' times.');

END LOOP;

  CLOSE openOfferDetailsCursor;

END;

--------------------------------------------

-- End of Coding

--------------------------------------------
```