# CAR RENTAL SYSTEM


TEAM:

Sumanth Pobala (71652304)

Bharath Kumar Karre (11072574)


SUBJECT: DATABASE SYSTEMS

FINAL PROJECT REPORT

# TABLE OF CONTENTS

## 1. INTRODUCTION TO CAR RENTAL SYSTEM

As a part of out Final Project in Database Systems we have chosen Car Rental System Database Design where customers can rent a car based on the model and make of the car. We have design a system where customers can pick-up and drop-off cars at different designated locations. If the car is not returned in time there will be a late fee on the rental car. There is an option called Car Rental Insurance where customers can buy the insurance and save the bill if they meet an accidents or something happens to the vehicle or them. They can use a discount coupon code on their final amount which is billed according the number of days that a car has been taken. If customer enrolls in a membership, a 10% of discount is automatically applied to the final bill.

**Used IDE's:** Oracle express 10G, Oracle Developer edition

### 1.1. REQUIREMENTS OF THE SYSTEM

i. Car rental agency must have a collection of cars so that customer chooses on his own interest.
ii. Every car should be designated to a type of car and to a particular location.
iii. Every customer based on his location and the type of car he wants to drive, rents a car.
iv. Customer would select a car based on the suggestions and should be able to successfully reserve the car.
v. At the moment customer reserves the car, they can choose a optional Insurance plan and if they have a valid discount code they can apply to the reservation while reserving the car.
vi. In this system, if a customer is already a member of the car agency and have a valid Membership ID they can avail an automatic discount of 10% and can apply one discount coupon.
vii. A bill is generated when the car is returned back to the agency.
viii. If a customer returns the car after the due date, they will be charged with a late fee and added to the final bill.
ix. In the bill a tax of 8.25% is added which also includes the late fee charge and its tax.
x. When the car is returned, it will available for the next booking.
xi. A reservation can be cancelled until 5days before the actual pickup.
xii. Car agency have different types of coupons like weekend, corporate, etc., Car price will be calculated based on the model and make of the car.

## 2. ENTITIES

### i. Customer

The customer is the one who uses a car hire system to book a car. He may be a system member or a non-system member. Membership id is going to be part of the system. Details like the customer license number, email address, name and telephone number are stored in the customer entity.

### ii. Car

Car entity has a list of cars on the system. Every car is linked to the category of a car, and the car has characteristics such as mark, model, km and registration number. Car also has a separate flag to monitor the car's availability.

### iii. Car Category

Each car is classified as a car. The price is based on the category of car. Price is determined. Car category shall have attributes, no baggage, name, cost per day and late charge per hour, like no person.

### iv. Location

The location here indicates the location of the car. The customer can pick up the vehicle from the specific location and have the same or different location. Attributes such as the location id, name and address are available to the location.

### v. Booking

Each car reservation will be monitored in the entity called booking. Booking will have attributes like booking id, from date and time of booking and due return date and time and actual return date and time of the booking and booking status. This booking amount might also include rental insurance and discount code.

### vi. Billing

A bill will be generated on the particular reservation when a customer returns a car. Billing has attributes such as Bill ID, bill date, bill status, total late fee, tax amount, and total amount.

### vii. Discount

While the bill is generated, customers can apply discount code. Each code of discount has different percentage of discount. Discount will have discount code, name, expiry date and percentage discount attributes.

viii. **Car rental insurance**

The customer may already have or can purchase a car rental insurance while reserving the car. Car rental insurance has attributes, such as insurance code, type of insurance, name and daily costs.

## 3. RELATIONS

i. **Car to Car Category**

Each vehicle will be associated with an car category. Once the customer selects an Car, the costs will be derived daily from the vehicle category of the selected vehicle.

ii. **Car to Location**

In a certain location, the customer will pick or drop the car. At this particular location, the customer can pick up or drop the car. Cars are therefore present at a place. The name of the relation is ' Current location '.

iii. **Booking to Billing**

Once the customer returns a bill for each reservation will be generated. In case of cancelation of the reservation, no bill is linked to the reservation. The name ' Gives' is the relationship.

iv. **Booking to Discount**

When a customer books a car, the customer can apply a discount code. This discount applies to the total amount of the bill after tax and late fee. The amount total is reduced by a certain percentage based on the discount code. The name 'Has ' is the relationship.

v. **Booking to Car Rental Insurance**

During the reservation, customers can select rental insurance to cover, cover damages based on the type of insurance coverage. The name of the relation is ' Includes. '

vi. **Booking to Location**

The customer can select a car from a specific location for rent. The name of the relation is ' Pick up location '

vii. **Booking to Location**

In a specific location, customers can drop off rental car. The name of the relationship is "Drop off location."

viii. **Customer to car to Booking**

ix. The customer is going to choose a rented car. So both the car and the reservation are related to the customer. The relationship between these three entities is a ternary relationship and the name of the relation is "Rents."

**4. Assumptions**

i. Only one car reservation at a time is associated with each booking.

ii. The car should be available in a given location in the system.

iii. The discount code may or may not be applied to billing.

iv. Not every reservation is related to billing due to cancelled reservations.

v. Reservations may or may not have insurance on rental because a customer may be insured for him / her.

**5. ER or Entity - Relationship Diagram**

**5.1. Entities**

- Customer
- Car Category
- Location
- Booking
- Billing
- Discount Car Rental Insurance

**5.2. Relationships**

- Car to Car Category
- Car to Location
- Booking to Billing
- Booking to Discount
- Booking to Car Rental Insurance
- Booking to Location
- Customer to car to Booking

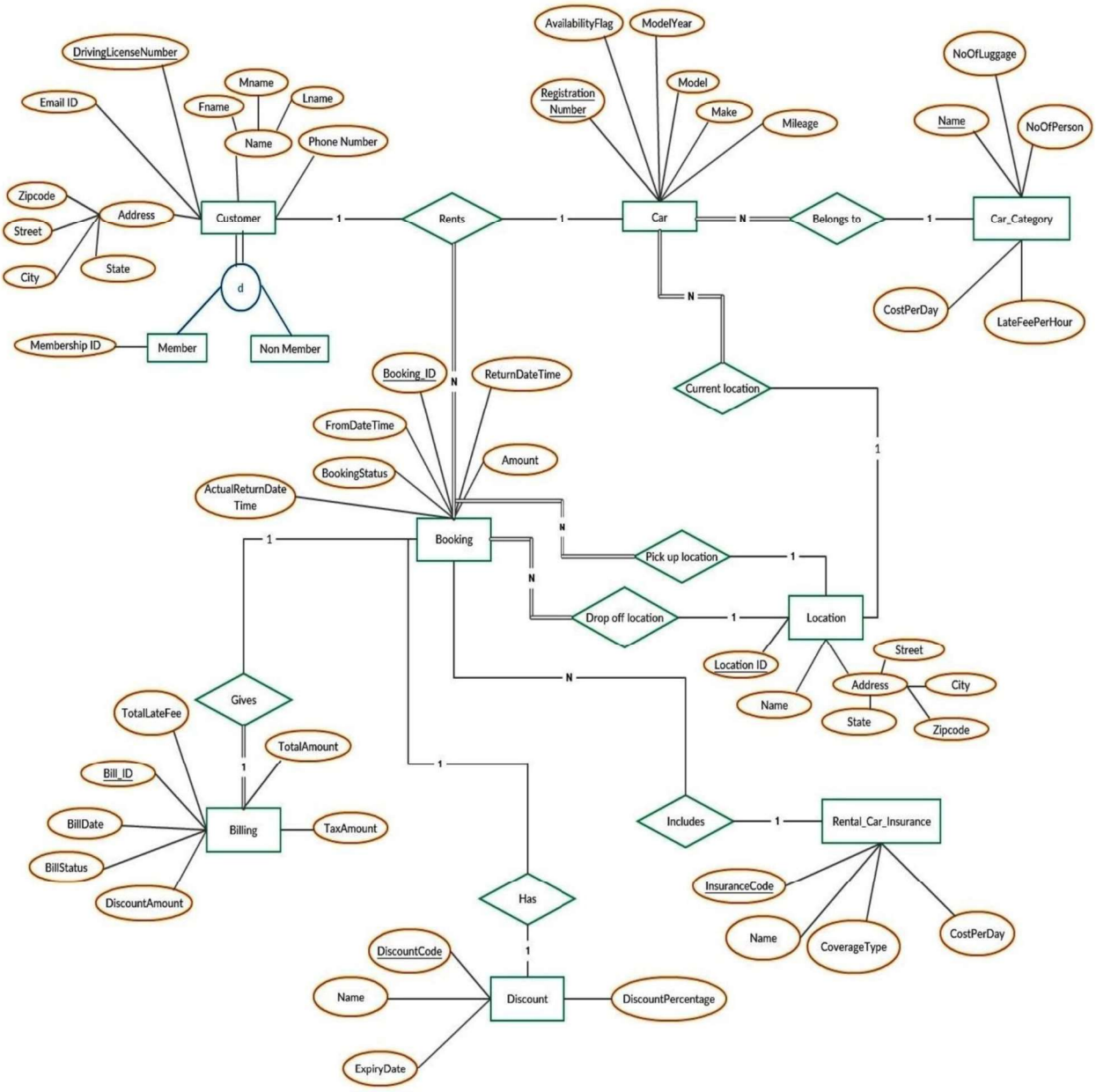


**Figure 1 ER Diagram of car Rental System**

## 6. FUNCTIONAL DEPENDENCIES

### a) Customer_Details Relation:

- DL_number -> Fname, Mname, Lname, Phone_number, Email_id, Street, City, State, Zipcode, Membership_id, Membership_type
- Zipcode -> State,City

**b) Car Relation:**

- Registration_number -> Model, Make, Model_year, Car_category_name, Loc_id, Mileage, Availability_flag
- Model -> Make

**c) Car_Category Relation:**

- Category_name -> No_of_luggage, No_of_person, Cost_per_day, Late_fee_per_hour

**d) Location _Details Relation:**

- Location_id -> Name,Street,City,State,Zipcode
- Zipcode -> State,City

**e) Booking_Details Relation:**

- Booking_id -> From_dt_time, Ret_dt_time, Amount, Booking_status, Pickup_loc, Drop_loc, Reg_num, DL_num, Ins_code, Act_ret_dt_time, Discount_code

**f) Billing_Details Relation:**

- Bill_id -> Bill_date, Bill_status, Discount_amt, Total_amt, Tax_amt, Booking_id, Total_late_fee

**g) Discount_Details Relation:**

- Discount_code -> Discount_name,Expiry_date,Discount_percentage
- Discount_name -> Discount_code ,Expiry_date,Discount_percentage

**h) Rental_Car_Insurance Relation:**

- Insurance_code -> Insurance_name,Coverage_type,Cost_per_day
- Insurance_name -> Insurance_code ,Coverage_type,Cost_per_day

## 6.1. FUNCTIONAL DEPENDENCIES THAT VIOLATE RULES

The Following transitive dependencies exist in the relational schema.

- **Customer_Details Relation**

DL_number -> Zipcode

Zipcode -> State,City

- **Car Relation**

Registration_number -> Model_name

Model_name -> Make

- **Location_Details Relation**

Location_id -> Zipcode

Zipcode -> State,City

## 7. Final Relation Schema

For convenience, we have de-normalized the relation into 3NF from 2NF. Now we are representing the relation in second normal form.

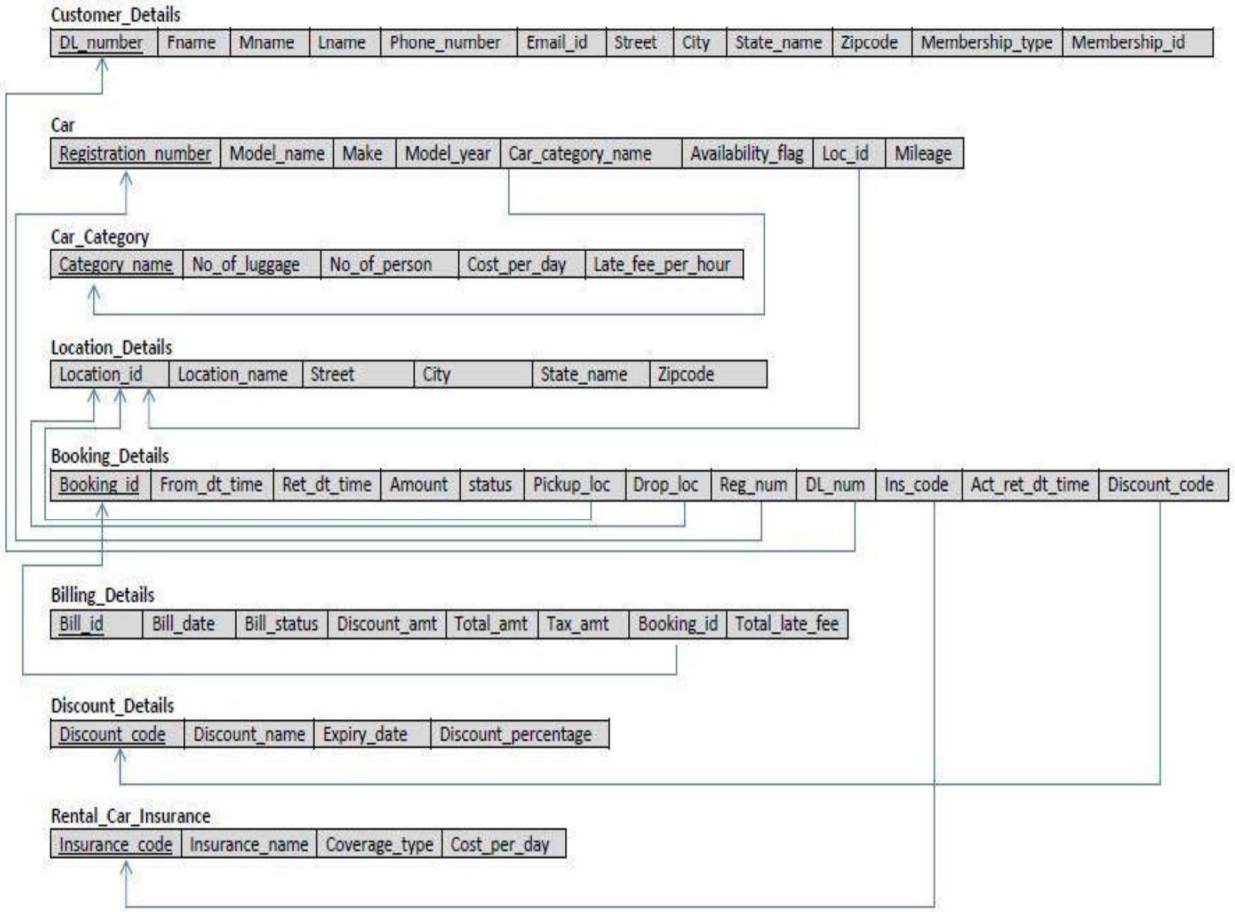


Figure 2 Final Relation Schema

## 8. SQL STATEMENTS

### 8.1. CREATE TABLES

#### i. CUSTOMER_DETAILS

```
CREATE TABLE CUSTOMER_DETAILS
( DL_NUMBER CHAR(8) NOT NULL,
FNAME VARCHAR(25) NOT NULL,
MNAME VARCHAR(15),
LNAME VARCHAR(25) NOT NULL,
PHONE_NUMBER NUMBER(10) NOT NULL,
EMAIL_ID VARCHAR(30) NOT NULL,
STREET VARCHAR(30) NOT NULL,
CITY VARCHAR(20) NOT NULL,
STATE_NAME VARCHAR(20) NOT NULL,
ZIPCODE NUMBER(5) NOT NULL,
MEMBERSHIP_TYPE CHAR(1) DEFAULT 'N' NOT NULL,
MEMBERSHIP_ID CHAR(5),
CONSTRAINT CUSTOMERPK
PRIMARY KEY (DL_NUMBER) );
```

#### ii. CAR_CATEGORY

```
CREATE TABLE CAR_CATEGORY
( CATEGORY_NAME VARCHAR(25) NOT NULL,
NO_OF_LUGGAGE INTEGER NOT NULL,
NO_OF_PERSON INTEGER NOT NULL,
COST_PER_DAY NUMBER(5,2) NOT NULL,
LATE_FEE_PER_HOUR NUMBER(5,2) NOT NULL,
CONSTRAINT CARCATEGORYPK
PRIMARY KEY (CATEGORY_NAME) );
```

#### iii. LOCATION_DETAILS

```
CREATE TABLE LOCATION_DETAILS
( LOCATION_ID CHAR(4) NOT NULL,
LOCATION_NAME VARCHAR(50) NOT NULL,
STREET VARCHAR(30) NOT NULL,
```

```
CITY VARCHAR(20) NOT NULL,
STATE_NAME VARCHAR(20) NOT NULL,
ZIPCODE NUMBER(5) NOT NULL,
CONSTRAINT LOCATIONPK
PRIMARY KEY (LOCATION_ID)
);
```

**iv. CAR**

```
CREATE TABLE CAR
( REGISTRATION_NUMBER CHAR(7) NOT NULL,
MODEL_NAME VARCHAR(25) NOT NULL,
MAKE VARCHAR(25) NOT NULL,
MODEL_YEAR NUMBER(4) NOT NULL,
MILEAGE INTEGER NOT NULL,
CAR_CATEGORY_NAME VARCHAR(25) NOT NULL,
LOC_ID CHAR(4) NOT NULL,
AVAILABILITY_FLAG CHAR(1) NOT NULL,
CONSTRAINT CARPK
PRIMARY KEY (REGISTRATION_NUMBER),
CONSTRAINT CARFK1
FOREIGN KEY (CAR_CATEGORY_NAME) REFERENCES
CAR_CATEGORY(CATEGORY_NAME),
CONSTRAINT CARFK2
FOREIGN KEY (LOC_ID) REFERENCES LOCATION_DETAILS(LOCATION_ID)
);
```

**v. DISCOUNT_DETAILS**

```
CREATE TABLE DISCOUNT_DETAILS
( DISCOUNT_CODE CHAR(4) NOT NULL,
DISCOUNT_NAME VARCHAR(25) NOT NULL,
EXPIRY_DATE DATE NOT NULL,
DISCOUNT_PERCENTAGE NUMBER(4,2) NOT NULL,
CONSTRAINT DISCOUNTPK
PRIMARY KEY (DISCOUNT_CODE),
```

```
CONSTRAINT DISCOUNTSK
UNIQUE (DISCOUNT_NAME)
);
```

### vi. RENTAL_CAR_INSURANCE

```
CREATE TABLE RENTAL_CAR_INSURANCE
( INSURANCE_CODE CHAR(4) NOT NULL,
INSURANCE_NAME VARCHAR(50) NOT NULL,
COVERAGE_TYPE VARCHAR(200) NOT NULL,
COST_PER_DAY NUMBER(4,2) NOT NULL,
CONSTRAINT INSURANCEPK
PRIMARY KEY (INSURANCE_CODE),
CONSTRAINT INSURANCESK
UNIQUE (INSURANCE_NAME)
);
```

### vii. BOOKING_DETAILS

```
CREATE TABLE BOOKING_DETAILS
( BOOKING_ID CHAR(5) NOT NULL,
FROM_DT_TIME TIMESTAMP NOT NULL,
RET_DT_TIME TIMESTAMP NOT NULL,
AMOUNT NUMBER(10,2) NOT NULL,
BOOKING_STATUS CHAR(1) NOT NULL,
PICKUP_LOC CHAR(4) NOT NULL,
DROP_LOC CHAR(4) NOT NULL,
REG_NUM CHAR(7) NOT NULL,
DL_NUM CHAR(8) NOT NULL,
INS_CODE CHAR(4),
ACT_RET_DT_TIME TIMESTAMP,
DISCOUNT_CODE CHAR(4),
CONSTRAINT BOOKINGPK
PRIMARY KEY (BOOKING_ID),
CONSTRAINT BOOKINGFK1
FOREIGN KEY (PICKUP_LOC) REFERENCES LOCATION_DETAILS(LOCATION_ID),
```

```
CONSTRAINT BOOKINGFK2
FOREIGN KEY (DROP_LOC) REFERENCES LOCATION_DETAILS(LOCATION_ID),
CONSTRAINT BOOKINGFK3
FOREIGN KEY (REG_NUM) REFERENCES CAR(REGISTRATION_NUMBER),
CONSTRAINT BOOKINGFK4
FOREIGN KEY (DL_NUM) REFERENCES CUSTOMER_DETAILS(DL_NUMBER),
CONSTRAINT BOOKINGFK5
FOREIGN KEY (INS_CODE) REFERENCES
RENTAL_CAR_INSURANCE(INSURANCE_CODE),
CONSTRAINT BOOKINGFK6
FOREIGN KEY (DISCOUNT_CODE) REFERENCES
DISCOUNT_DETAILS(DISCOUNT_CODE)
);
```

**viii. BILLING_DETAILS**

```
CREATE TABLE BILLING_DETAILS
( BILL_ID CHAR(6) NOT NULL,
BILL_DATE DATE NOT NULL,
BILL_STATUS CHAR(1) NOT NULL,
DISCOUNT_AMOUNT NUMBER(10,2) NOT NULL,
TOTAL_AMOUNT NUMBER(10,2) NOT NULL,
TAX_AMOUNT NUMBER(10,2) NOT NULL,
BOOKING_ID CHAR(5) NOT NULL,
TOTAL_LATE_FEE NUMBER(10,2) NOT NULL,
CONSTRAINT BILLINGPK
PRIMARY KEY (BILL_ID),
CONSTRAINT BILLINGFK1
FOREIGN KEY (BOOKING_ID) REFERENCES BOOKING_DETAILS(BOOKING_ID)
);
```

## 8.2 INSERT SQL STATEMENTS

### 8.2.1. CUSTOMER_DETAILS

```
INSERT INTO CUSTOMER_DETAILS VALUES('F1234554', 'NAVEEN',
NULL,'RAJ','4696004267', 'naveen@gmail.com','700 CAMPBELL RD',
'RICHARDSON','TEXAS',75080,'M','M1001');
INSERT INTO CUSTOMER_DETAILS VALUES('F9764521', 'NIVEDITIIA',
NULL,'VARADHA CHANDRASEKARAN','4696478596', 'nivi07@gmail.com',
'800 RENNER RD','RICHARDSON','TEXAS',75080,'M','M1002');
INSERT INTO CUSTOMER_DETAILS VALUES('F2345611', 'SURESH',
'KUMAR','GOPALAKRISHNAN','8189187546', 'suresh2234@gmail.com',
'6547 CANOGA AVE','CANOGA PARK','CALIFORNIA',91303,'N',NULL);
INSERT INTO CUSTOMER_DETAILS VALUES('R8763578', 'MARK',
NULL,'HUFF','7345678902', 'markhuff@gmail.com','1445 ROSS AVE',
'DALLAS','TEXAS',75202,'N',NULL);
INSERT INTO CUSTOMER_DETAILS VALUES('I3478953', 'MARK',
'S','TOWNSEND','9872563478', 'markstown@gmail.com','7825 MCCALLUM BLVD',
'DALLAS','TEXAS',75252,'M','M1003');
INSERT INTO CUSTOMER_DETAILS VALUES('E7521097', 'MITA',
NULL,'RANA','9098123429', 'mitarana@gmail.com','367 MEANDERING WAY',
'HOUSTON','TEXAS',76245,'N',NULL);
INSERT INTO CUSTOMER_DETAILS VALUES('T0981237', 'DANISH',
NULL,'HASSAN','6712890345', 'danishhasan@gmail.com','888 PRESTON ROAD',
'DULLES','VIRGINIA',92367,'M','M1004');
INSERT INTO CUSTOMER_DETAILS VALUES('F0091266', 'MIKE',
NULL,'BOYEAR','7892340918', 'mikeboy@gmail.com','1007 DALLAS PARKWAY',
'DALLAS','TEXAS',72212,'N',NULL);
INSERT INTO CUSTOMER_DETAILS VALUES('P1234567', 'CHRIS',
NULL,'ALEXANDER','9902489', 'chrisalex@gmail.com','2256 WALL STREET',
'NEWARK','NEW JERSEY',65289,'M','M1005');
INSERT INTO CUSTOMER_DETAILS VALUES('V5690245',
'VELA','R','REYNALDO','9908762514', 'reyvela@gmail.com','0099 ALMA ROAD',
'DULLES','VIRGINIA',97325,'N',NULL);
```

```
INSERT INTO CUSTOMER_DETAILS VALUES('A1234567', 'THARUN',
NULL,'BOGI','3132660795', 'tharun@gmail.com','700 Meadows',
'Dearborn','MICHIGAN',48126,'M','M0001');
INSERT INTO CUSTOMER_DETAILS VALUES('B1234567', 'SUMANTH',
NULL,'POBALA','3132564859', 'spobala@gmail.com', '208 MEADOWS
DR','Dearborn','MICHIGAN',48126,'M','M0002');
INSERT INTO CUSTOMER_DETAILS VALUES('C1234567', 'BHARATH',
'KUMAR','KARRE','3132661415', 'bharathk@gmail.com', 'Q208 HEATHER
DR','MEADOWS','MICHIGAN',48126,'N',NULL);
INSERT INTO CUSTOMER_DETAILS VALUES('J8763578', 'YASHWANTH',
NULL,'KANDIMALLA','313145614', 'yashwanthk@gmail.com','s207 HEATHER DR',
'MEADOWS','MICHIGAN',48126,'N',NULL);
INSERT INTO CUSTOMER_DETAILS VALUES('D1234567', 'KARTHIK',
'CH','REDDY','313256489', 'karthikshe@gmail.com','Q208 HEATHER DR',
'MEADOWS','MICHIGAN',48126,'M','M0003');
INSERT INTO CUSTOMER_DETAILS VALUES('E1234567', 'SWATHI',
NULL,'PRIYA','313265478', 'swathip@gmail.com','ASD 1234',
'DEARBORN','MICHIGAN',48126,'N',NULL);
INSERT INTO CUSTOMER_DETAILS VALUES('E7521097', 'AMULYA',
NULL,'NAYAK','6712890345', 'amulyakn@gmail.com','F 212',
'MEADOWS','MICHIGAN',48126,'M','M0004');
INSERT INTO CUSTOMER_DETAILS VALUES('G1234567', 'SUHASINI',
NULL,'KL','31324568', 'suhasinikl@gmail.com','c 123456',
'MEADOWS','MICHIGAN',48126,'N',NULL);
INSERT INTO CUSTOMER_DETAILS VALUES('H1234567', 'VENKATA',
NULL,'KAMESHWAR','9902489', 'kameshwar@gmail.com','Q 208', 'HEATHER DR
APPTS','MICHIGAN',48126,'M','M0005');
INSERT INTO CUSTOMER_DETAILS VALUES('I1234567',
'TEJA','R','DURISETI','1234567895', 'tejad@gmail.com','F 209',
'HEATHER','VIRGINIA',97325,'N',NULL);
```

### 8.2.2. CAR_CATEGORY

```
INSERT INTO CAR_CATEGORY VALUES('ECONOMY',2,5,30,0.9);
INSERT INTO CAR_CATEGORY VALUES('COMPACT',3,5,32,0.96);
INSERT INTO CAR_CATEGORY VALUES('MID SIZE',3,5,35,1.05);
INSERT INTO CAR_CATEGORY VALUES('STANDARD',3,5,38,1.14);
INSERT INTO CAR_CATEGORY VALUES('FULL SIZE',4,5,40,1.2);
INSERT INTO CAR_CATEGORY VALUES('LUXURY CAR',5,5,75,2.25);
INSERT INTO CAR_CATEGORY VALUES('MID SIZE SUV',2,5,36,1.08);
INSERT INTO CAR_CATEGORY VALUES('STANDARD SUV',3,5,40,1.2);
INSERT INTO CAR_CATEGORY VALUES('FULL SIZE SUV',2,8,60,1.8);
INSERT INTO CAR_CATEGORY VALUES('MINI VAN',5,7,70,2.1);
```

### 8.2.3. LOCATION_DETAILS

```
INSERT INTO LOCATION_DETAILS VALUES('L101','DALLAS LOVE FIELD AIRPORT',
'Herb Kelleher Way','Dallas','Texas',75235);
INSERT INTO LOCATION_DETAILS VALUES('L102','LOS ANGELES INTL AIRPORT',
'World Way','Los Angeles','California',90045);
INSERT INTO LOCATION_DETAILS VALUES('L103','DALLAS/ FORT WORTH INTL
AIRPORT',
'International Pkwy','DFW Airport','Texas',75261);
INSERT INTO LOCATION_DETAILS VALUES('L104','WEST IIOUSTON AIRPORT',
'Groschke Rd','Houston','Texas',77094);
INSERT INTO LOCATION_DETAILS VALUES('L105','WASHINGTON DULLES INTL
AIRPORT',
'Saarinen Cir','Dulles','Virginia',20166);
INSERT INTO LOCATION_DETAILS VALUES('L106','NEWARK LIBERTY INTL AIRPORT',
'Brewster Rd','Newark','New Jersey',07114);
INSERT INTO LOCATION_DETAILS VALUES('L107','SALT LAKE CITY INTL AIRPORT',
'N Terminal Dr','Salt Lake City','Utah',84122);
```

### 8.2.4. CAR_VALUES

```
INSERT INTO CAR VALUES('ABX1234','CIVIC','HONDA',
2014,10000,'ECONOMY','L101','A');
INSERT INTO CAR VALUES('SDF4567','FIESTA','FORD',
```

```
2015,15000,'ECONOMY','L102','N');
INSERT INTO CAR VALUES('WER3245','ACCENT','HYUNDAI',
2014,12356,'ECONOMY','L103','A');
INSERT INTO CAR VALUES('GLZ2376','COROLLA','TOYOTA',
2016,5000,'ECONOMY','L104','A');
INSERT INTO CAR VALUES('HJK1234','CIVIC','HONDA',
2015,20145,'ECONOMY','L102','N');
INSERT INTO CAR VALUES('GLS7625','FOCUS','FORD',
2014,12000,'COMPACT','L107','A');
INSERT INTO CAR VALUES('FKD8202','GOLF','VOLKSWAGAN',
2016,9000,'COMPACT','L106','A');
INSERT INTO CAR VALUES('HNX1890','PRIUS','TOYOTA',
2015,15690,'COMPACT','L105','N');
INSERT INTO CAR VALUES('KJS1983','PRIUS','TOYOTA',
2014,20900,'COMPACT','L104','A');
INSERT INTO CAR VALUES('SDL9356','FOCUS','FORD',
2016,10009,'COMPACT','L103','A');
INSERT INTO CAR VALUES('OTY7293','CRUZE','CHEVROLET',
2016,17800,'MID SIZE','L102','A');
INSERT INTO CAR VALUES('QWE4562','LEGACY','SUBARU',
2012,13420,'MID SIZE','L101','A');
INSERT INTO CAR VALUES('CXZ2356','AVENGER','DODGE',
2015,5000,'MID SIZE','L102','A');
INSERT INTO CAR VALUES('ASD9090','ACCORD','HONDA',
2016,200,'MID SIZE','L103','A');
INSERT INTO CAR VALUES('UYT3981','LEGACY','SUBARU',
2013,16750,'MID SIZE','L104','A');
INSERT INTO CAR VALUES('TRE9726','200','CHRYSTLER',
2012,14320,'STANDARD','L105','A');
INSERT INTO CAR VALUES('HGF5628','TAURUS','FORD',
2013,15540,'STANDARD','L106','A');
```

```
INSERT INTO CAR VALUES('LKJ7253','200','CHRYSTLER',
2014,16300,'STANDARD','L107','A');
INSERT INTO CAR VALUES('VBN6283','TAURUS','FORD',
2015,17500,'STANDARD','L101','A');
INSERT INTO CAR VALUES('POI7281','200','CHRYSTLER',
2016,18830,'STANDARD','L102','N');
INSERT INTO CAR VALUES('MNB8654','FALCON','FORD',
2012,10900,'FULL SIZE','L103','A');
INSERT INTO CAR VALUES('UHV9786','IMPALA','CHEVROLET',
2013,11500,'FULL SIZE','L104','A');
INSERT INTO CAR VALUES('EFB5427','WAYFARER','FORD',
2014,14350,'FULL SIZE','L105','A');
INSERT INTO CAR VALUES('PLM9873','IMPALA','CHEVROLET',
2015,18900,'FULL SIZE','L106','A');
INSERT INTO CAR VALUES('WDV2458','FALCON','FORD',
2016,5600,'FULL SIZE','L107','A');
INSERT INTO CAR VALUES('QSC8709','MKZ','LINCOLN',

2012,18700,'LUXURY CAR','L101','A');
INSERT INTO CAR VALUES('TGB8961','GENESIS','HYUNDAI',
2013,17620,'LUXURY CAR','L102','A');
INSERT INTO CAR VALUES('MKU0172','TLX','ACURA',
2014,12345,'LUXURY CAR','L103','A');
INSERT INTO CAR VALUES('CFT1908','328I','BMW',
2015,10800,'LUXURY CAR','L104','A');
INSERT INTO CAR VALUES('WHM7619','AVALON','TOYOTA',
2016,7800,'LUXURY CAR','L105','A');
INSERT INTO CAR VALUES('WLZ8955','ESCAPE','FORD',
2012,19800,'MID SIZE SUV','L106','A');
INSERT INTO CAR VALUES('QIO7621','EQUINOX','CHEVROLET',
2013,17560,'MID SIZE SUV','L107','A');
```

```
INSERT INTO CAR VALUES('YSN1927','PATHFINDER','NISSAN',
2014,14390,'MID SIZE SUV','L101','A');
INSERT INTO CAR VALUES('EDM8610','GLA','MERCEDEZ BENZ',
2015,12900,'MID SIZE SUV','L102','A');
INSERT INTO CAR VALUES('AHK7325','RAV4','TOYOTA',
2016,3400,'MID SIZE SUV','L103','A');
INSERT INTO CAR VALUES('OHZ0976','EDGE','FORD',
2012,27890,'STANDARD SUV','L104','A');
INSERT INTO CAR VALUES('RKS9862','TAHOE','CHEVROLET',
2013,20390,'STANDARD SUV','L105','A');
INSERT INTO CAR VALUES('WIJ6190','EDGE','FORD',
2014,18700,'STANDARD SUV','L106','A');
INSERT INTO CAR VALUES('ZDT8612','TAHOE','CHEVROLET',
2015,14300,'STANDARD SUV','L107','A');
INSERT INTO CAR VALUES('LDJ7719','EDGE','FORD',
2016,5690,'STANDARD SUV','L101','A');
INSERT INTO CAR VALUES('UIA8709','EXPEDITION','FORD',
2012,19870,'FULL SIZE SUV','L102','A');
INSERT INTO CAR VALUES('WKJ7972','SEQUOIA','TOYOTA',
2013,14500,'FULL SIZE SUV','L103','A');
INSERT INTO CAR VALUES('JLS1097','SUBURBAN','CHEVROLET',
2014,13290,'FULL SIZE SUV','L104','A');
INSERT INTO CAR VALUES('UHJ6782','EXPEDITION','FORD',
2015,11750,'FULL SIZE SUV','L105','A');
INSERT INTO CAR VALUES('XBM6822','SUBURBAN','CHEVROLET',
2016,3400,'FULL SIZE SUV','L106','A');
INSERT INTO CAR VALUES('SHK7767','QUEST','NISSAN
2012,23478,'MINI VAN','L107','A');
INSERT INTO CAR VALUES('JSL7920','ODYSSEY','HONDA',
2013,19320,'MINI VAN','L106','A');
INSERT INTO CAR VALUES('PAJ5289','GRAND CARAVAN','DODGE',
```

2014,23478,'MINI VAN','L105','A');

INSERT INTO CAR VALUES('TSJ6290','QUEST','NISSAN',

2015,13200,'MINI VAN','L104','A');

INSERT INTO CAR VALUES('MWO9296','ODYSSEY','HONDA',

2016,2300,'MINI VAN','L103','A');

### 8.2.5. DISCOUNT_DETAILS

INSERT INTO DISCOUNT_DETAILS VALUES ('D678','IBM CORPORATE',

to_date('2018-01-25','YYYY-MM-DD'),25);

INSERT INTO DISCOUNT_DETAILS VALUES ('D234','CTS CORPORATE',

to_date('2019-09-02','YYYY-MM-DD'),20);

INSERT INTO DISCOUNT_DETAILS VALUES ('D756','HOLIDAY SPECIAL',

to_date('2017-10-29','YYYY-MM-DD'),10);

INSERT INTO DISCOUNT_DETAILS VALUES ('D109','WEEKLY RENTALS',

to_date('2020-11-09','YYYY-MM-DD'),25);

INSERT INTO DISCOUNT_DETAILS VALUES ('D972','ONE WAY SPECIAL',

to_date('2016-12-15','YYYY-MM-DD'),20);

INSERT INTO DISCOUNT_DETAILS VALUES ('D297','UPGRADE SPECIAL',

to_date('2018-02-18','YYYY-MM-DD'),20);

### 8.2.6. RENTAL_CAR_INSURANCE

INSERT INTO RENTAL_CAR_INSURANCE VALUES('I201', 'COLLISION DAMAGE WAIVER',

'Covers theft and total damage to the rental car',15);

INSERT INTO RENTAL_CAR_INSURANCE VALUES('I202',

'SUPPLEMENTAL LIABILITY PROTECTION', 'Covers damage done to others',12);

INSERT INTO RENTAL_CAR_INSURANCE VALUES('I203',

'PERSONAL ACCIDENT INSURANCE', 'Covers medical costs for driver and passengers',10);

INSERT INTO RENTAL_CAR_INSURANCE VALUES('I204',

'PERSONAL EFFECTS COVERAGE', 'Covers theft of personal belongings',5);

### 8.2.7. BOOKING_DETAILS

INSERT INTO BOOKING_DETAILS VALUES('B1001',TO_TIMESTAMP('2016-01-20 10:00:00',

```
'YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2016-01-25 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
150,'R','L101','L101','ABX1234','F1234554',NULL,
TO_TIMESTAMP('2016-01-25 10:00:00', 'YYYY-MM-DD HH24:MI:SS'),'D756');
INSERT INTO BOOKING_DETAILS VALUES('B1002',TO_TIMESTAMP('2016-01-21
11:00:00',
'YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2016-01-24 10:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
90,'C','L102','L102','SDF4567','T0981237',NULL,NULL,NULL);
INSERT INTO BOOKING_DETAILS VALUES('B1003',TO_TIMESTAMP('2016-02-10
13:00:00',
'YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2016-01-15 13:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
450,'R','L101','L101','QSC8709','R8763578','I201',
TO_TIMESTAMP('2016-01-15 13:00:00', 'YYYY-MM-DD HH24:MI:SS'),NULL);
INSERT INTO BOOKING_DETAILS VALUES('B1004',TO_TIMESTAMP('2016-04-24
13:00:00',
'YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2016-04-25 20:30:00', 'YYYY-MM-DD
HH24:MI:SS'),
48,'R','L106','L106','WLZ8955','F0091266','I202',
TO_TIMESTAMP('2016-04-23 20:30:00', 'YYYY-MM-DD HH24:MI:SS'),'D234');
INSERT INTO BOOKING_DETAILS VALUES('B1005',TO_TIMESTAMP('2016-04-18
09:00:00',
'YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2016-04-25 09:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
266,'B','L102','L106','POI7281','P1234567',NULL,NULL,'D972');
INSERT INTO BOOKING_DETAILS VALUES('B1006',TO_TIMESTAMP('2016-04-21
17:00:00',
'YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2016-04-25 17:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
168,'B','L105','L107','HNX1890','V5690245','I203',NULL,'D234');
```

```
INSERT INTO BOOKING_DETAILS VALUES('B1007',TO_TIMESTAMP('2016-04-16 08:00:00',
'YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2016-04-25 17:00:00', 'YYYY-MM-DD HH24:MI:SS'),
405,'B','L102','L102','SDF4567','I3478953','I201',NULL,'D756');
INSERT INTO BOOKING_DETAILS VALUES('B1008',TO_TIMESTAMP('2016-04-11 08:00:00',
'YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2016-04-25 17:00:00', 'YYYY-MM-DD HH24:MI:SS'),
630,'B','L102','L102','HJK1234','T0981237','I201',NULL,'D756');
INSERT INTO BOOKING_DETAILS VALUES('B1010',TO_TIMESTAMP('2018-01-21 11:00:00', 'YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2018-01-24 10:00:00', 'YYYY-MM-DD HH24:MI:SS'),90,'C','L102','L102','SDF4567','T0981237',NULL,NULL,NULL);
INSERT INTO BOOKING_DETAILS VALUES('B1011',TO_TIMESTAMP('2018-02-10 13:00:00', 'YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2018-01-15 13:00:00', 'YYYY-MM-DD HH24:MI:SS'),450,'R','L101','L101','QSC8709','R8763578','I201',TO_TIMESTAMP('2016-01-15 13:00:00', 'YYYY-MM-DD HH24:MI:SS'),NULL);
INSERT INTO BOOKING_DETAILS VALUES('B1012',TO_TIMESTAMP('2018-04-24 13:00:00', 'YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2018-04-25 20:30:00', 'YYYY-MM-DD HH24:MI:SS'),48,'R','L106','L106','WLZ8955','F0091266','I202',TO_TIMESTAMP('2018-04-23 20:30:00', 'YYYY-MM-DD HH24:MI:SS'),'D234');
INSERT INTO BOOKING_DETAILS VALUES('B1013',TO_TIMESTAMP('2018-04-18 09:00:00', 'YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2018-04-25 09:00:00', 'YYYY-MM-DD HH24:MI:SS'),266,'B','L109','L106','POI7281','P1234567',NULL,NULL,'D972');
INSERT INTO BOOKING_DETAILS VALUES('B1014',TO_TIMESTAMP('2018-04-21 17:00:00', 'YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2018-04-25 17:00:00', 'YYYY-MM-DD HH24:MI:SS'),168,'B','L105','L107','HNX1890','V5690245','I203',NULL,'D234');
INSERT INTO BOOKING_DETAILS VALUES('B1015',TO_TIMESTAMP('2018-04-16 08:00:00', 'YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2018-04-25 17:00:00', 'YYYY-MM-DD HH24:MI:SS'),405,'B','L102','L102','SDF4567','I3478953','I201',NULL,'D756');
```

INSERT INTO BOOKING_DETAILS VALUES('B1016',TO_TIMESTAMP('2018-04-11 08:00:00', 'YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2018-04-25 17:00:00', 'YYYY-MM-DD HH24:MI:SS'),630,'B','L108','L108','HJK1234','T0981237','I201',NULL,'D756');

### 8.2.8. BILLING_VALUES

INSERT INTO BILLING_DETAILS VALUES('BL1001',to_date('2016-01-25','YYYY-MM-DD'), 'P',24.36 ,138.03,12.38 ,'B1001',0);

INSERT INTO BILLING_DETAILS VALUES('BL1002',to_date('2016-01-15','YYYY-MM-DD'), 'P',0 ,487.13 ,12.38 ,'B1003',0);

INSERT INTO BILLING_DETAILS VALUES('BL1003',to_date('2016-04-24','YYYY-MM-DD'), 'P',10.39 ,41.57 ,3.96 ,'B1004',0);

INSERT INTO BILLING_DETAILS VALUES('BL1004',to_date('2018-04-24','YYYY-MM-DD'),'P',15.39 ,57.57 ,5.96 ,'B1012',0);

INSERT INTO BILLING_DETAILS VALUES('BL1005',to_date('2018-01-15','YYYY-MM-DD'),'P',36 ,51.57 ,2.96 ,'B1009',0);

INSERT INTO BILLING_DETAILS VALUES('BL1006',to_date('2018-04-25','YYYY-MM-DD'),'P',36.4 ,150.7 ,14.28 ,'B1014',5.7);

INSERT INTO BILLING_DETAILS VALUES('BL1006',to_date('2018-04-25','YYYY-MM-DD'),'P',36.4 ,150.7 ,14.28 ,'B1014',5.7);

INSERT INTO BILLING_DETAILS VALUES('BL1007',to_date('2018-01-24','YYYY-MM-DD'),'P',0 ,97.5 ,7.5 ,'B1010',0);

INSERT INTO BILLING_DETAILS VALUES('BL1008',to_date('2018-01-24','YYYY-MM-DD'),'P',0 ,488.25 ,38.25 ,'B1011',0);

## 9. PL/SQL STATEMENTS

### 9.1. STORED_PROCEDURE 1: CALCULATE_LATE_FEE_AND_TAX

Due to the date and time of your reservation, your real time and date of return, your car registration number and reservation amount, this method calculates the total late fee by means of the late fee per hour, the date and time of your return and the actual date and time the car was loaded for you. The amount is added to the late fee and the total tax is determined. When you have received the late fee.

```
--------------------------------------------------------------------------------
--Procedure Name: CALCULATE_LATE_FEE_AND_TAX
--This stored procedure is used to calculate the total late fee and tax.
--------------------------------------------------------------------------------
```

```
CREATE OR REPLACE PROCEDURE CALCULATE_LATE_FEE_AND_TAX
(actualReturnDateTime IN BOOKING_DETAILS.ACT_RET_DT_TIME%TYPE,
ReturnDateTime IN BOOKING_DETAILS.RET_DT_TIME%TYPE,
regNum IN BOOKING_DETAILS.REG_NUM%TYPE,
amount IN BOOKING_DETAILS.AMOUNT%TYPE,
totalLateFee OUT BILLING_DETAILS.TOTAL_AMOUNT%TYPE,
totalTax OUT BILLING_DETAILS.TAX_AMOUNT%TYPE ) AS
--local declarations
lateFeePerHour CAR_CATEGORY.LATE_FEE_PER_HOUR%TYPE;
hourDifference DECIMAL(10,2);
BEGIN
SELECT LATE_FEE_PER_HOUR INTO lateFeePerHour
FROM CAR_CATEGORY CC INNER JOIN CAR C ON CC.CATEGORY_NAME =
C.CAR_CATEGORY_NAME WHERE C.REGISTRATION_NUMBER = regNum;
IF actualReturnDateTime > ReturnDateTime THEN
hourDifference := (TO_DATE (TO_CHAR (actualReturnDateTime,
'dd/mm/yyyy hh24:mi:ss'), 'dd/mm/yyyy hh24:mi:ss')
- TO_DATE (TO_CHAR (ReturnDateTime, 'dd/mm/yyyy hh24:mi:ss')
,'dd/mm/yyyy hh24:mi:ss'))*(24);
totalLateFee := hourDifference * lateFeePerHour;
ELSE
totalLateFee := 0;
END IF;
totalTax := (amount + totalLateFee)*0.0825;
END;
/
```

### 9.2. STORED_PROCEDURE 2: CALCULATE_DISCOUNT_AMOUNT

This procedure calculates the discount amount based on the discount code used by the customer when booking the rental car, given the driver's license number, total amount and the discount code. Customers with membership ID receive a further 10% discount. The discount is calculated on the total sum obtained after the late payment and the amount of tax have been added.

- For customers with No-Membership:
  Discount amount = total amount * ((discountPercentage)/100)
- For members:
  Discount amount = total amount * ((discountPercentage+10)/100)

```
--------------------------------------------------------------------------------
--Procedure Name: CALCULATE_DISCOUNT_AMOUNT
--This stored procedure calculates the discount amount for a booking.
--------------------------------------------------------------------------------
CREATE OR REPLACE PROCEDURE CALCULATE_DISCOUNT_AMOUNT
(dlNum IN CUSTOMER_DETAILS.DL_NUMBER%TYPE,
amount IN BILLING_DETAILS.TOTAL_AMOUNT%TYPE,
discountCode IN DISCOUNT_DETAILS.DISCOUNT_CODE%TYPE,
discountAmt OUT BILLING_DETAILS.DISCOUNT_AMOUNT%TYPE) AS
--local declarations
memberType CUSTOMER_DETAILS.MEMBERSHIP_TYPE%TYPE;
discountPercentage DISCOUNT_DETAILS.DISCOUNT_PERCENTAGE%TYPE;
BEGIN
SELECT MEMBERSHIP_TYPE INTO memberType FROM CUSTOMER_DETAILS
WHERE DL_NUMBER = dlNum;
IF NVL(discountCode,'NULL') <> 'NULL' THEN
SELECT DISCOUNT_PERCENTAGE INTO discountPercentage
FROM DISCOUNT_DETAILS WHERE DISCOUNT_CODE = discountCode;
IF memberType = 'M' THEN
discountAmt := amount * ((discountPercentage+10)/100);
ELSE
discountAmt := amount * (discountPercentage/100);
END IF;
ELSE
IF memberType = 'M' THEN
discountAmt := amount * 0.1;
```

```
ELSE
discountAmt := 0;
END IF;
END IF;
END;
/
```

### 9.3 Stored Procedure 3: GENERATE_REVENUE_REPORT

Based on the date, location and automobile classes, this process generates a monthly return report. The no of cars and the total revenue for each location are calculated and a monthly report is produced. The cursor is used for report generation by this stored procedure.

```
--------------------------------------------------------------------------------
--Procedure Name: GENERATE_REVENUE_REPORT
--This stored procedure calculates and generates the monthly revenue report.
--------------------------------------------------------------------------------
CREATE OR REPLACE PROCEDURE GENERATE_REVENUE_REPORT AS
--local declarations
thisLocationID LOCATION_DETAILS.LOCATION_ID%TYPE;
currentLocationID LOCATION_DETAILS.LOCATION_ID%TYPE;
locationName LOCATION_DETAILS.LOCATION_NAME%TYPE;
thisCategoryName CAR_CATEGORY.CATEGORY_NAME%TYPE;
thisNoOfCars integer; thisRevenue DECIMAL(15,2);
--Cursor declaration
CURSOR CURSOR_REPORT IS SELECT TABLE1.LOCATIONID, TABLE1.CATNAME ,
TABLE1.NOOFCARS,SUM(NVL((TABLE2.AMOUNT),0)) AS REVENUE
FROM (SELECT LC.LID AS LOCATIONID, LC.CNAME AS CATNAME ,
COUNT(C.REGISTRATION_NUMBER) AS NOOFCARS FROM (SELECT
L.LOCATION_ID AS LID, CC.CATEGORY_NAME AS CNAME FROM
CAR_CATEGORY CC CROSS JOIN LOCATION_DETAILS L) LC LEFT OUTER JOIN
CAR C ON LC.CNAME = C.CAR_CATEGORY_NAME AND LC.LID = C.LOC_ID
GROUP BY LC.LID, LC.CNAME ORDER BY LC.LID) TABLE1 LEFT OUTER JOIN
```

```
(SELECT BC.PLOC AS PICKLOC,BC.CNAME AS CNAMES, SUM(BL.TOTAL_AMOUNT) AS
AMOUNT FROM (SELECT B.PICKUP_LOC AS PLOC, C1.CAR_CATEGORY_NAME AS
CNAME,
B.BOOKING_ID AS BID FROM BOOKING_DETAILS B INNER JOIN CAR C1 ON
B.REG_NUM = C1.REGISTRATION_NUMBER) BC INNER JOIN BILLING_DETAILS BL
ON BC.BID = BL.BOOKING_ID WHERE
(to_date (SYSDATE,'dd-MM-yyyy') - to_date(BL.BILL_DATE,'dd-MM-yyyy'))
<=30 GROUP BY BC.PLOC,BC.CNAME ORDER BY BC.PLOC) TABLE2
ON TABLE1.LOCATIONID=TABLE2.PICKLOC AND TABLE1.CATNAME =
TABLE2.CNAMES
GROUP BY TABLE1.LOCATIONID, TABLE1.CATNAME, TABLE1.NOOFCARS
ORDER BY TABLE1.LOCATIONID;
BEGIN
dbms_output.put_line(' ');
dbms_output.put_line('Revenue Report');
dbms_output.put_line('**************');

dbms_output.put_line(' ');
OPEN CURSOR_REPORT;
FETCH CURSOR_REPORT INTO thisLocationID, thisCategoryName,
thisNoOfCars, thisRevenue;
IF CURSOR_REPORT%NOTFOUND THEN
dbms_output.put_line('No Report to be Generated');
ELSE
currentLocationID := thisLocationID;
<<LABEL_NEXTLOC>>
SELECT LOCATION_NAME INTO locationName from LOCATION_DETAILS
WHERE LOCATION_ID = currentLocationID;
dbms_output.put_line('Location Name: '|| locationName);
dbms_output.put_line(' ');
dbms_output.put_line('Car Category' || ' '||'Number of Cars'
```

```
||' '|| 'Revenue');
dbms_output.put_line('------------' || ' '||'--------------'
||' '|| '-------');
dbms_output.put_line(thisCategoryName ||
RPAD(' ', (16 - LENGTH(thisCategoryName)))||thisNoOfCars
||RPAD(' ', (18 - LENGTH(thisNoOfCars)))|| thisRevenue);
LOOP
FETCH CURSOR_REPORT INTO thisLocationID, thisCategoryName,
thisNoOfCars, thisRevenue;
EXIT WHEN (CURSOR_REPORT%NOTFOUND);
IF thisLocationID = currentLocationID THEN
dbms_output.put_line(thisCategoryName ||
RPAD(' ', (16 - LENGTH(thisCategoryName)))||thisNoOfCars
||RPAD(' ', (18 - LENGTH(thisNoOfCars)))|| thisRevenue);
ELSE
currentLocationID := thisLocationID;
dbms_output.put_line(' ');
dbms_output.put_line('**********************
*****************************************
***************
**********************');
dbms_output.put_line(' ');
GOTO LABEL_NEXTLOC;
END IF;
END LOOP;
END IF;
END;
/
```

### 9.4 Trigger 1: GENERATE_BILLING

When the actual return date is updated and the reservation status updated to ' R ' in the Booking Details tab, this trigger insers tuples into the table Billing Details. When the rental car is returned, it generates Bill. This will occur once a row in the Booking Details table is updated.

```
--------------------------------------------------------------------------------------
--Trigger Name: GENERATE_BILLING
--This trigger generates the bill and inserts a row in Billing_Details table
--------------------------------------------------------------------------------------
CREATE OR REPLACE TRIGGER GENERATE_BILLING
AFTER UPDATE ON BOOKING_DETAILS
FOR EACH ROW
WHEN (NVL(TO_CHAR(NEW.ACT_RET_DT_TIME),'NULL') <> 'NULL' AND
NEW.BOOKING_STATUS ='R')
DECLARE
-- declaration section
lastBillId BILLING_DETAILS.BILL_ID%TYPE;
newBillId BILLING_DETAILS.BILL_ID%TYPE;
discountAmt BILLING_DETAILS.DISCOUNT_AMOUNT%TYPE;
totalLateFee BILLING_DETAILS.TOTAL_LATE_FEE%TYPE;
totalTax BILLING_DETAILS.TAX_AMOUNT%TYPE;
totalAmountBeforeDiscount BILLING_DETAILS.TOTAL_AMOUNT%TYPE;
finalAmount BILLING_DETAILS.TOTAL_AMOUNT%TYPE;
BEGIN
SELECT BILL_ID INTO lastBillId FROM ( SELECT BILL_ID, ROWNUM AS
RN FROM BILLING_DETAILS)
WHERE RN= (SELECT MAX(ROWNUM) FROM BILLING_DETAILS);
newBillId := 'BL' || TO_CHAR(TO_NUMBER(SUBSTR(lastBillId,3))+1);
CALCULATE_LATE_FEE_AND_TAX(:NEW.ACT_RET_DT_TIME, :NEW.RET_DT_TIME,
:NEW.REG_NUM,:NEW.AMOUNT, totalLateFee, totalTax);
totalAmountBeforeDiscount := :NEW.AMOUNT + totalLateFee + totalTax;
```

```
CALCULATE_DISCOUNT_AMOUNT(:NEW.DL_NUM, totalAmountBeforeDiscount,
:NEW.DISCOUNT_CODE, discountAmt);
finalAmount := totalAmountBeforeDiscount - discountAmt;
--insert new bill into the billing_details table
INSERT INTO BILLING_DETAILS
(BILL_ID,BILL_DATE,BILL_STATUS,DISCOUNT_AMOUNT,
TOTAL_AMOUNT,TAX_AMOUNT,BOOKING_ID,TOTAL_LATE_FEE)
VALUES (newBillId,to_date(SYSDATE,'YYYY-MM-DD'),'P',
discountAmt,finalAmount,totalTax,:NEW.BOOKING_ID,totalLateFee);
END;
/
```

### 9.5 Trigger 2: UPDATE_CAR_DETAILS

This trigger will update the car's flag available, mileage and location when the actual date of return is updated or a reservation cancelled. When a row is updated in the Booking Detail table, this is triggered.

```
--------------------------------------------------------------------------------------
--Trigger Name: UPDATE_CAR_DETAILS
--This trigger updates the availability flag, mileage and location in the car table
--when the car is returned.
--------------------------------------------------------------------------------------
CREATE OR REPLACE TRIGGER UPDATE_CAR_DETAILS
AFTER UPDATE ON BOOKING_DETAILS
FOR EACH ROW
WHEN (NVL(TO_CHAR(NEW.ACT_RET_DT_TIME),'NULL') <> 'NULL' OR
NEW.BOOKING_STATUS ='C')
DECLARE
BEGIN
IF :NEW.BOOKING_STATUS ='C' THEN
UPDATE CAR SET AVAILABILITY_FLAG = 'A' , LOC_ID = :NEW.PICKUP_LOC WHERE
REGISTRATION_NUMBER = :NEW.REG_NUM;
ELSE
IF NVL(TO_CHAR(:NEW.ACT_RET_DT_TIME),'NULL') <> 'NULL' THEN
UPDATE CAR SET AVAILABILITY_FLAG = 'A' , LOC_ID = :NEW.DROP_LOC,
MILEAGE = MILEAGE+GET_MILEAGE WHERE REGISTRATION_NUMBER =
:NEW.REG_NUM;
END IF;
END IF;
END;
/
```

# 10. SAMPLE OUTPUT SCREENS

## 10.1. CUSTOMER DETAILS

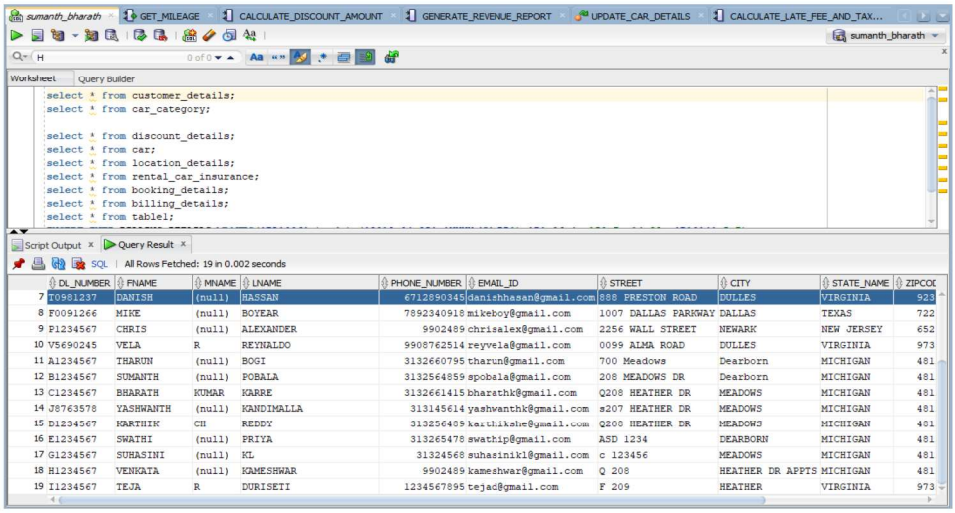


Figure 3 Customer Details Snapshot

## 10.2. CAR_CATEGORY

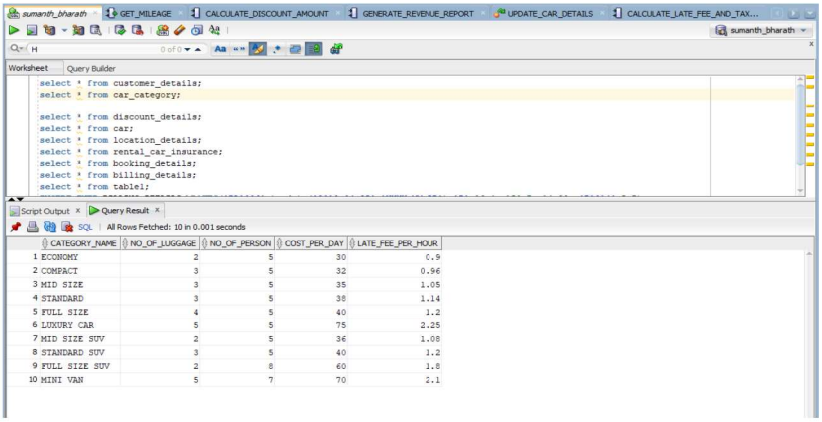


Figure 4 Car Category Snapshot

## 10.3. DISCOUNT-DETAILS

Script Output | Query Result

SQL | All Rows Fetched: 6 in 0.001 seconds

| | DISCOUNT_CODE | DISCOUNT_NAME | EXPIRY_DATE | DISCOUNT_PERCENTAGE |
|---|---|---|---|---|
| 1 | D678 | IBM CORPORATE | 25-JAN-18 | 25 |
| 2 | D234 | CTS CORPORATE | 02-SEP-19 | 20 |
| 3 | D756 | HOLIDAY SPECIAL | 29-OCT-17 | 10 |
| 4 | D109 | WEEKLY RENTALS | 09-NOV-20 | 25 |
| 5 | D972 | ONE WAY SPECIAL | 15-DEC-16 | 20 |
| 6 | D297 | UPGRADE SPECIAL | 18-FEB-18 | 20 |

Figure 5 Discount Details Snapshot

## 10.4. CAR

Script Output | Query Result

SQL | Fetched 50 rows in 0.002 seconds

| | REGISTRATION_NUMBER | MODEL_NAME | MAKE | MODEL_YEAR | MILEAGE | CAR_CATEGORY_NAME | LOC_ID | AVAILABILITY_FLAG |
|---|---|---|---|---|---|---|---|---|
| 1 | ABX1234 | CIVIC | HONDA | 2014 | 10000 | ECONOMY | L101 | A |
| 2 | SDF4567 | FIESTA | FORD | 2015 | 15000 | ECONOMY | L102 | N |
| 3 | WER3245 | ACCENT | HYUNDAI | 2014 | 12356 | ECONOMY | L103 | A |
| 4 | GLZ2376 | COROLLA | TOYOTA | 2016 | 5000 | ECONOMY | L104 | A |
| 5 | HJK1234 | CIVIC | HONDA | 2015 | 20145 | ECONOMY | L102 | N |
| 6 | GLS7625 | FOCUS | FORD | 2014 | 12000 | COMPACT | L107 | A |
| 7 | FKD8202 | GOLF | VOLKSWAGAN | 2016 | 9000 | COMPACT | L106 | A |
| 8 | HNX1890 | PRIUS | TOYOTA | 2015 | 15690 | COMPACT | L105 | N |
| 9 | KJS1983 | PRIUS | TOYOTA | 2014 | 20900 | COMPACT | L104 | A |
| 10 | SDL9356 | FOCUS | FORD | 2016 | 10009 | COMPACT | L103 | A |
| 11 | OTY7293 | CRUZE | CHEVROLET | 2016 | 17800 | MID SIZE | L102 | A |
| 12 | QWE4562 | LEGACY | SUBARU | 2012 | 13420 | MID SIZE | L101 | A |
| 13 | CXZ2356 | AVENGER | DODGE | 2015 | 5000 | MID SIZE | L102 | A |
| 14 | ASD9090 | ACCORD | HONDA | 2016 | 200 | MID SIZE | L103 | A |

Figure 6 Car Snapshot

## 10.5. LOCATION_DETAILS

Script Output × Query Result ×

SQL | All Rows Fetched: 9 in 0.002 seconds

| | LOCATION_ID | LOCATION_NAME | STREET | CITY | STATE_NAME | ZIPCODE |
|---|---|---|---|---|---|---|
| 1 | L101 | DALLAS LOVE FIELD AIRPORT | Herb Kelleher Way | Dallas | Texas | 75235 |
| 2 | L102 | LOS ANGELES INTL AIRPORT | World Way | Los Angeles | California | 90045 |
| 3 | L103 | DALLAS/ FORT WORTH INTL AIRPORT | International Pkwy | DFW Airport | Texas | 75261 |
| 4 | L104 | WEST HOUSTON AIRPORT | Groschke Rd | Houston | Texas | 77094 |
| 5 | L105 | WASHINGTON DULLES INTL AIRPORT | Saarinen Cir | Dulles | Virginia | 20166 |
| 6 | L106 | NEWARK LIBERTY INTL AIRPORT | Brewster Rd | Newark | New Jersey | 7114 |
| 7 | L107 | SALT LAKE CITY INTL AIRPORT | N Terminal Dr | Salt Lake City | Utah | 84122 |
| 8 | L108 | DETROIT METROPOLITAN AIRPORT | N Terminal | Wayne Country | Michigan | 67945 |
| 9 | L109 | Portland International Airrport | Terminal 3 | Portland | Oregon | 17396 |

Figure 7 Location Details Snapshot

## 10.6. RENTAL_CAR_INSURANCE

Script Output × Query Result ×

SQL | All Rows Fetched: 4 in 0 seconds

| | INSURANCE_CODE | INSURANCE_NAME | COVERAGE_TYPE | COST_PER_DAY |
|---|---|---|---|---|
| 1 | I201 | COLLISION DAMAGE WAIVER | Covers theft and total damage to the rental car | 15 |
| 2 | I202 | SUPPLEMENTAL LIABILITY PROTECTION | Covers damage done to others | 12 |
| 3 | I203 | PERSONAL ACCIDENT INSURANCE | Covers medical costs for driver and passengers | 10 |
| 4 | I204 | PERSONAL EFFECTS COVERAGE | Covers theft of personal belongings | 5 |

Figure 8 Rental Car Insurance Snapshot

## 10.7. CAR_CATEGORY

Script Output × Query Result ×

SQL | Fetched 50 rows in 0.003 seconds

| | LOCATIONID | CATNAME | NOOFCARS |
|---|---|---|---|
| 1 | L101 | COMPACT | 0 |
| 2 | L101 | ECONOMY | 1 |
| 3 | L101 | FULL SIZE | 0 |
| 4 | L101 | FULL SIZE SUV | 0 |
| 5 | L101 | LUXURY CAR | 1 |
| 6 | L101 | MID SIZE | 1 |
| 7 | L101 | MID SIZE SUV | 1 |
| 8 | L101 | MINI VAN | 0 |
| 9 | L101 | STANDARD | 1 |
| 10 | L101 | STANDARD SUV | 1 |
| 11 | L102 | COMPACT | 0 |
| 12 | L102 | ECONOMY | 2 |
| 13 | L102 | FULL SIZE | 0 |
| 14 | L102 | FULL SIZE SUV | 1 |

Figure 9 Car Category Snapshot

## 10.8. BOOKING_DETAILS

Script Output | Query Result

SQL | All Rows Fetched: 16 in 0.001 seconds

| | BOOKING_ID | FROM_DT_TIME | RET_DT_TIME | AMOUNT | BOOKING_STATUS | PICKUP_LOC | DROP_LOC | REG_NUM | DL_NUM | INS_C |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | B1001 | 20-JAN-16 10.00.00.000000000 AM | 25-JAN-16 10.00.00.000000000 AM | 150 | R | L101 | L101 | ABX1234 | F1234554 | (null) |
| 2 | B1002 | 21-JAN-16 11.00.00.000000000 AM | 24-JAN-16 10.00.00.000000000 AM | 90 | C | L102 | L102 | SDF4567 | T0981237 | (null) |
| 3 | B1003 | 10-FEB-16 01.00.00.000000000 PM | 15-JAN-16 01.00.00.000000000 PM | 450 | R | L101 | L101 | QSC8709 | R8763578 | I201 |
| 4 | B1004 | 24-APR-16 01.00.00.000000000 PM | 25-APR-16 08.30.00.000000000 PM | 48 | R | L106 | L106 | WLZ8955 | F0091266 | I202 |
| 5 | B1005 | 18-APR-16 09.00.00.000000000 AM | 25-APR-16 09.00.00.000000000 AM | 266 | B | L102 | L106 | POI7281 | P1234567 | (null) |
| 6 | B1006 | 21-APR-16 05.00.00.000000000 PM | 25-APR-16 05.00.00.000000000 PM | 168 | B | L105 | L107 | HNX1890 | V5690245 | I203 |
| 7 | B1007 | 16-APR-16 08.00.00.000000000 AM | 25-APR-16 05.00.00.000000000 PM | 405 | B | L102 | L102 | SDF4567 | I3478953 | I201 |
| 8 | B1008 | 11-APR-16 08.00.00.000000000 AM | 25-APR-16 05.00.00.000000000 PM | 630 | B | L102 | L102 | HJK1234 | T0981237 | I201 |
| 9 | B1009 | 20-JAN-18 10.00.00.000000000 AM | 25-JAN-18 10.00.00.000000000 AM | 150 | R | L101 | L101 | ABX1234 | F1234554 | (null) |
| 10 | B1010 | 21-JAN-18 11.00.00.000000000 AM | 24-JAN-18 10.00.00.000000000 AM | 90 | C | L102 | L102 | SDF4567 | T0981237 | (null) |
| 11 | B1011 | 10-FEB-18 01.00.00.000000000 PM | 15-JAN-18 01.00.00.000000000 PM | 450 | R | L101 | L101 | QSC8709 | R8763578 | I201 |
| 12 | B1012 | 24-APR-18 01.00.00.000000000 PM | 25-APR-18 08.30.00.000000000 PM | 48 | R | L106 | L106 | WLZ8955 | F0091266 | I202 |
| 13 | B1013 | 18-APR-18 09.00.00.000000000 AM | 25-APR-18 09.00.00.000000000 AM | 266 | B | L109 | L106 | POI7281 | P1234567 | (null) |

Figure 10 Booking Details Snapshot

## 10.9. FINAL OUTPUT

Script Output | Query Result

SQL | All Rows Fetched: 8 in 0.001 seconds

| | BILL_ID | BILL_DATE | BILL_STATUS | DISCOUNT_AMOUNT | TOTAL_AMOUNT | TAX_AMOUNT | BOOKING_ID | TOTAL_LATE_FEE |
|---|---|---|---|---|---|---|---|---|
| 1 | BL1006 | 25-APR-18 | P | 36.4 | 150.7 | 14.28 | B1014 | 5.7 |
| 2 | BL1007 | 24-JAN-18 | P | 0 | 97.5 | 7.5 | B1010 | 0 |
| 3 | BL1008 | 24-JAN-18 | P | 0 | 488.25 | 38.25 | B1011 | 0 |
| 4 | BL1001 | 25-JAN-16 | P | 24.36 | 138.03 | 12.38 | B1001 | 0 |
| 5 | BL1002 | 15-JAN-16 | P | 0 | 487.13 | 12.38 | B1003 | 0 |
| 6 | BL1003 | 24-APR-16 | P | 10.39 | 41.57 | 3.96 | B1004 | 0 |
| 7 | BL1004 | 24-APR-18 | P | 15.39 | 57.57 | 5.96 | B1012 | 0 |
| 8 | BL1005 | 15-JAN-18 | P | 36 | 51.57 | 2.96 | B1009 | 0 |

Figure 11 Final Output of Billing Details Snapshot

## 11. CONCLUSION

Through this project we have acquired the knowledge in designing an Entity-Relationship Diagram with all the rules and how to design a best relation schema. During this course of the project we have learnt how to map relations to ER diagram and how to derive the functional dependencies and how to normalize the relation schema. We have learnt how to create a database and connect it to the created server. We have used the concepts of Stored Procedures and trigger concepts and came to know how to effectively use them to trigger the values and procedures. Finally through this project we have enlightened ourselves in how to store and manipulate the data.