Auto Parts Sales

(CS 6360 – 005)

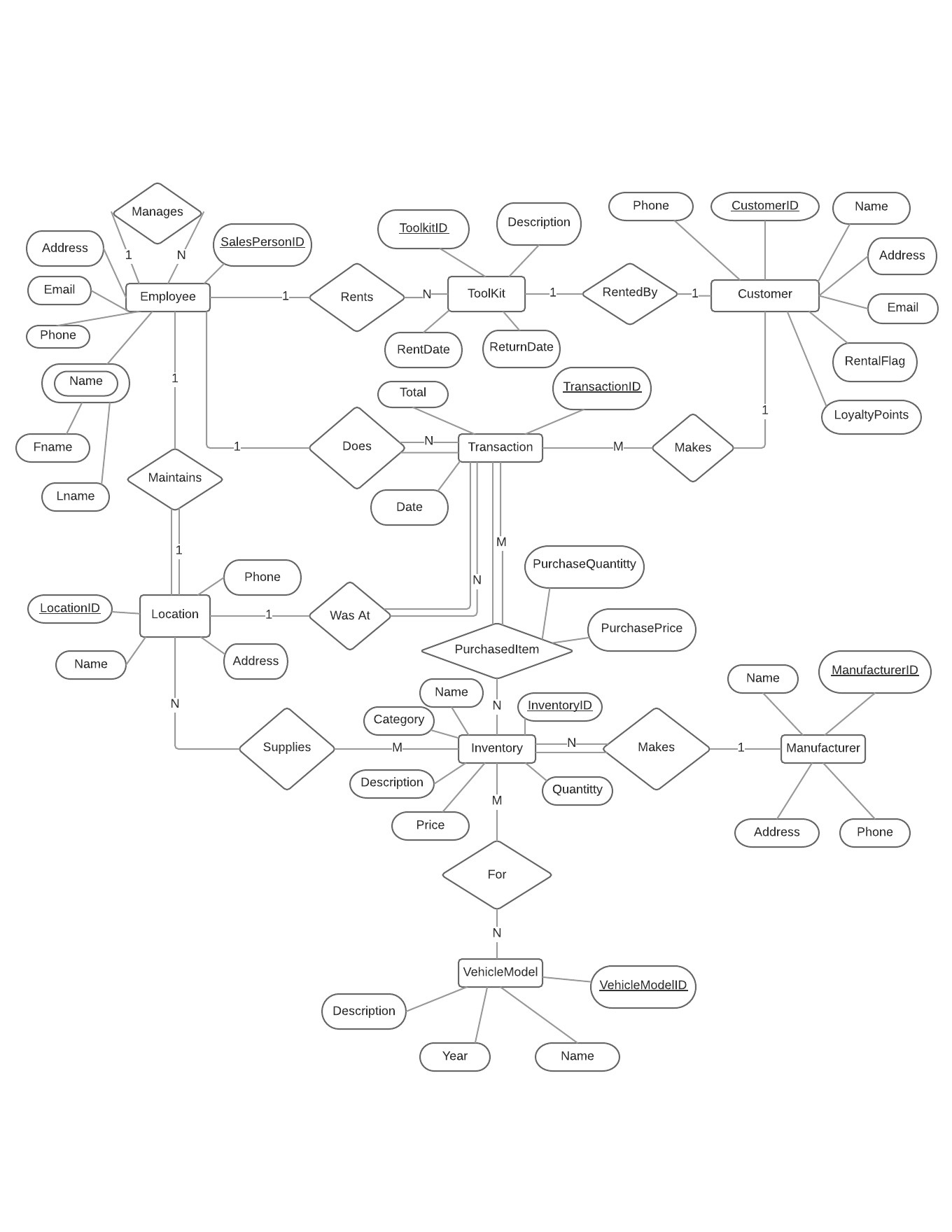
Apurva Patel

Himanshu Parashar

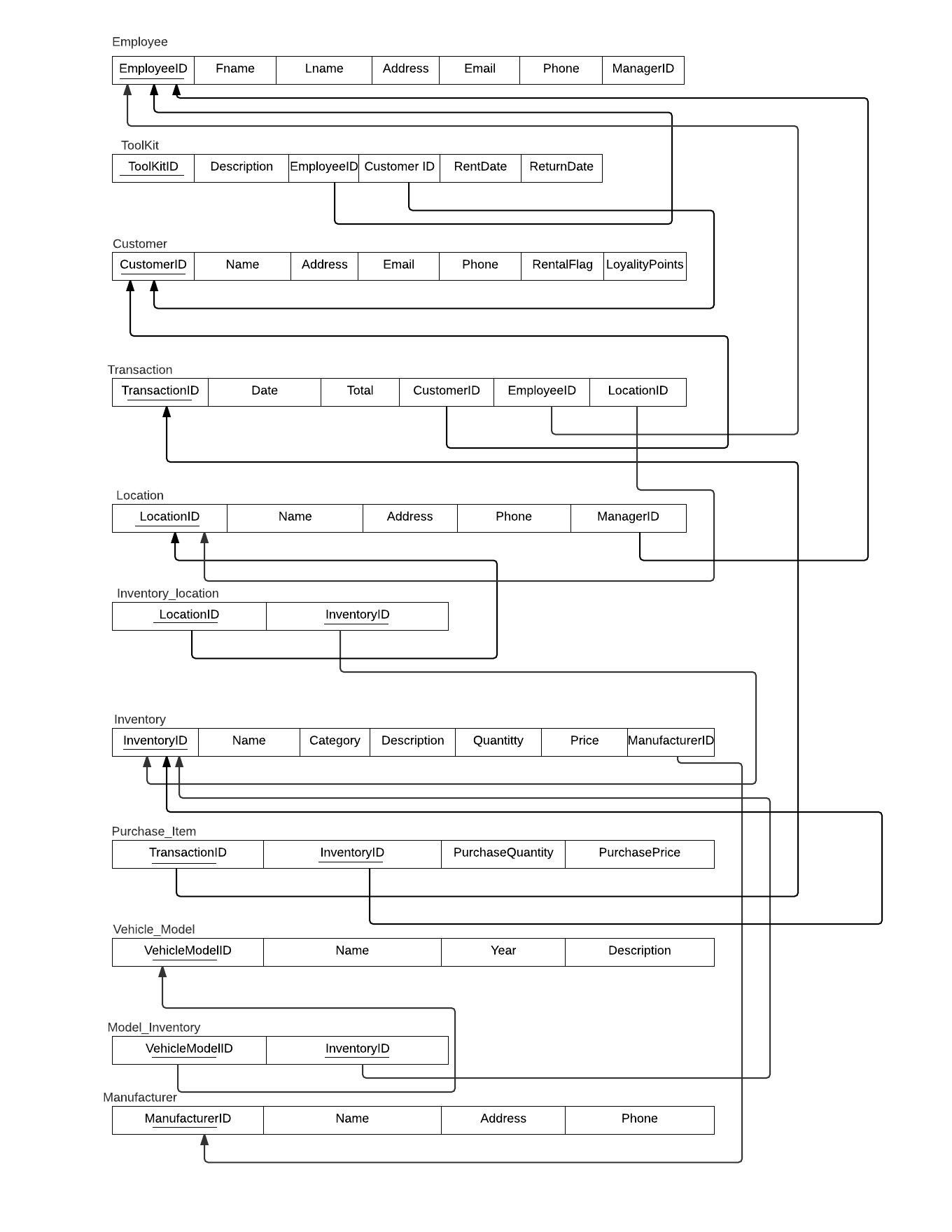
Satwant Singh

**NOTE:** The tables are host on Satwant’s account (SXS149531).

1. **Data Requirements**
   * An auto parts sales system needs to be able to keep track of current inventory, sales transactions, and current rentals. It also needs to keep track of the registered users, employees, and store locations.
   * There can be multiple store locations. Each location has a unique id, name, address, phone number, and a unique manager assigned to it.
   * Auto parts sales stores have two types of employees, salesperson and a manager. The distinction between the two is that the manager has a store that they manage and every salesperson has a manager assigned to them. Each employee has a unique id, name, address, phone, and an email. Managers do not have a manager supervising them.
   * There can be two types of customers, registered and unregistered customers. Unregistered customers can only be identified by the sales transaction they are involved in. Registered customers can be identified by their unique id. The system also keeps track of registered customers' names, addresses, phones, emails, and loyalty points. Both registered and unregistered customers can be make a sales transaction but only registered customers can rent toolkits from a store. The system also keeps track of whether a registered user is currently renting a toolkit as only 1 toolkit per customer is allowed at a time.
   * Each sales transaction that occurs in the system, has a unique id, the id of the salesperson who conducted the transaction, time, total, the id of the store location, and optionally the id of a registered customer. If a customer does not provide their customer id at the time of purchase, the sales transaction assumes they are unregistered. This is similar to presenting a rewards card at a supermarket. The transaction also keeps track of all the items purchased and their quantity and price.
   * Each rental transaction has an id of the toolkit, id of the salesperson, rent date, return deadline and an id of the registered customer. Only registered customers are allowed to rent toolkits. Each rental transaction also have flags for the registered customer to signify that the customer has an active rental. This is because the system will only allow one toolkit rental at a time. Customers can earn points for purchasing from the store and spend them to rent toolkits.
   * The system also keeps track of all inventory across all locations. Each item in the inventory has a unique id, name, description, price, available quantity, category, location, manufacturer, and a vehicle it is for.
   * The system also keeps track of different manufacturers of parts. Each manufacturer has an ID, name, address, and phone.
   * Vehicle models are also kept track of by their unique id, name, year, and description.
2. **ER Diagram**

****

1. **Relational Schema (Already in 3NF)**



1. **Functional Dependencies**

EmployeeID -> Fname, Lname, Address, Email, Phone, ManagerID

ToolKitID -> Description, EmployeeID, CustomerID, RentDate, ReturnDate

CustomerID -> Name, Address, Email, Phone, RentalFlag, LoyalityPoints

TransactionID -> Date, Total, CustomerID, EmployeeID, LocationID

LocationID -> Name, Address, Phone, ManagerID

InventoryID -> Name, Category, Description, Quantity, Price, ManufacturerID

TransactionID, InventoryID -> PurchaseQuantity, PurchasePrice

VehicleModelID -> Name, Year, Description

ManufacturerID -> Name, Address, Phone

1. **Database Normalization Rules**

Since the initial schema does not have any transitive dependencies and every attribute is depended on the key, there does not need to be any decomposition. The only thing needed is a new table for each of the 3 M-to-N relationships with foreign keys to both entities’ primary keys as the keys of the new table.

1. **Final Relational Schema**

Due to the initial relational schema already being in 3NF, the final relational schema is the same as the initial relational schema in step 4.

1. **Create Statements (SQL)**

Create Table Employee

(

Employee\_id VARCHAR2(10) NOT NULL,

Fname VARCHAR2(35),

Lname VARCHAR2(35),

Address VARCHAR2(35) DEFAULT 'NO PUBLISHER' ,

Email VARCHAR2(35),

Phone\_no NUMBER(10),

Manager\_id Varchar2(10),

CONSTRAINT EMPPK PRIMARY KEY (Employee\_id)

);

Create Table Customer

(

Cust\_id VARCHAR2(10) NOT NULL,

Name VARCHAR2(35),

Address VARCHAR2(35) DEFAULT 'NO PUBLISHER' ,

Email VARCHAR2(35),

Phone\_no NUMBER(10),

Rental\_Flag NUMBER(1) ,

Loyalty\_Points NUMBER(10),

CONSTRAINT CUSTPK PRIMARY KEY (Cust\_id)

);

Create Table Tool\_kit

(

ToolKit\_id VARCHAR2(10) NOT NULL,

Description VARCHAR2(35),

Employee\_id Varchar2(10),

Customer\_id Varchar2(10),

RentDate date,

ReturnDate date,

Constraint TOOLPK Primary Key (ToolKit\_id),

Constraint ToolFK1 FOREIGN key (Employee\_Id) references Employee(Employee\_id) on delete set null,

constraint ToolFK2 FOREIGN key (Customer\_id) references Customer(Cust\_id) on delete set null

);

Create table Transaction

(

Txn\_id VARCHAR2(10) NOT NULL,

Txn\_date Date,

Total Number(10,2),

Customer\_id Varchar2(10),

Employee\_id Varchar2(10) not null ,

Location\_id varchar2 (10) not null ,

Constraint TxnPK primary key (txn\_id),

constraint txnFK1 FOREIGN key (Employee\_id) references employee(employee\_id) ,

constraint txnFK2 FOREIGN key (Customer\_id) references customer(Cust\_id),

constraint txnFK3 FOREIGN key (Location\_id) references Location(Loc\_id)

);

Create Table Location

(

Loc\_id VARCHAR2(10) NOT NULL ,

name VARCHAR2(35),

Address VARCHAR2(35),

Phone Number(10),

Manager\_id Varchar(10) not Null ,

constraint LocPK primary key (Loc\_id),

constraint LocFK FOREIGN key (manager\_id) references employee(employee\_id)

);

Create table Manufacturer

(

Man\_id VARCHAR2(10) NOT NULL,

Name VARCHAR2(35),

Address VARCHAR2(35) DEFAULT 'NO PUBLISHER' ,

Phone\_no NUMBER(10),

constraint MANPK primary key (Man\_id)

);

Create Table Vehicle\_model

(

Vehicle\_mod\_id VARCHAR2 (10) not null ,

Vehicle\_Name VARCHAR2(35) not null ,

year number(4) ,

description varchar(35),

constraint VehPK primary key (Vehicle\_mod\_id)

);

Create table inventory

(

Inventory\_id VARCHAR2(10) NOT NULL,

Name VARCHAR2(35),

category varchar2(35),

description varchar(35),

quantity number(6),

price number(10,2),

manufacturer\_id VARCHAR2(10) not null ,

constraint InvPK primary key(Inventory\_id) ,

constraint INVFK FOREIGN key (manufacturer\_id) REFERENCES Manufacturer(Man\_id) on delete cascade

);

create table inventory\_location

(

Location\_id VARCHAR2(10) NOT NULL,

Inventory\_id VARCHAR2(10) NOT NULL,

constraint INVLOCPK primary key (Location\_id,Inventory\_id),

constraint INVLOCFK1 FOREIGN key(location\_id) references Location(Loc\_id) on delete cascade,

constraint INVLOCFK2 FOREIGN key(Inventory\_id) references Inventory(Inventory\_id) on delete cascade

);

create table purchase\_item

(

txn\_id varchar2(10) not null ,

inventory\_id varchar2(10) not null,

purchase\_quantity number(6),

purchase\_price number(10,2),

constraint PURCHSITMPK primary key (txn\_id,inventory\_id),

constraint PURCHSITMFK1 FOREIGN key (txn\_id) REFERENCES transaction(txn\_id) on delete cascade,

constraint PURCHSITMFK2 FOREIGN key (inventory\_id) REFERENCES Inventory(Inventory\_id)

);

create table Model\_inventory

(

Vehicle\_Model\_ID varchar2(10) not null ,

inventory\_id varchar2(10) not null,

constraint MODINV PRIMARY key (Vehicle\_Model\_ID,inventory\_id),

constraint MODELINVFK1 FOREIGN key (inventory\_id) REFERENCES Inventory(Inventory\_id) on delete cascade,

constraint MODELINVFK2 FOREIGN key (Vehicle\_Model\_ID) REFERENCES Vehicle\_model(Vehicle\_mod\_id) on delete cascade

);

**Insert Statements (SQL) to add tuples**

Insert into CUSTOMER (CUST\_ID,NAME,ADDRESS,EMAIL,PHONE\_NO,RENTAL\_FLAG,LOYALTY\_POINTS) values ('0000000000','Himanshu','123 Home Road','hm@gmail.com',9111234,0,100);

Insert into CUSTOMER (CUST\_ID,NAME,ADDRESS,EMAIL,PHONE\_NO,RENTAL\_FLAG,LOYALTY\_POINTS) values ('0000000001','George','45 Street','grg@gmail.com',9371377321,0,230);

Insert into CUSTOMER (CUST\_ID,NAME,ADDRESS,EMAIL,PHONE\_NO,RENTAL\_FLAG,LOYALTY\_POINTS) values ('0000000002','Harvey','34 Street','harv@gmail.com',663716717,0,590);

Insert into CUSTOMER (CUST\_ID,NAME,ADDRESS,EMAIL,PHONE\_NO,RENTAL\_FLAG,LOYALTY\_POINTS) values ('0000000003','Mike','43 Triangle Street','mike@gmail.com',7761361763,0,300);

Insert into CUSTOMER (CUST\_ID,NAME,ADDRESS,EMAIL,PHONE\_NO,RENTAL\_FLAG,LOYALTY\_POINTS) values ('0000000004','Tony','12 Pine Street','tony@gmail.com',921938813,0,500);

Insert into EMPLOYEE (EMPLOYEE\_ID,ADDRESS,EMAIL,PHONE\_NO,MANAGER\_ID,FNAME,LNAME) values ('1234567890','123 Street','kevin@work.com',5556688,null,'Kevin','Bob');

Insert into EMPLOYEE (EMPLOYEE\_ID,ADDRESS,EMAIL,PHONE\_NO,MANAGER\_ID,FNAME,LNAME) values ('1234567891','34 Road','manager@work.com',1230987,'1234567890','Satwant','Singh');

Insert into EMPLOYEE (EMPLOYEE\_ID,ADDRESS,EMAIL,PHONE\_NO,MANAGER\_ID,FNAME,LNAME) values ('1234560092','Apt456 ,Mccallum Road,Texas','Bob123@work.com',9756058545,'1234567890','Bob','White');

Insert into EMPLOYEE (EMPLOYEE\_ID,ADDRESS,EMAIL,PHONE\_NO,MANAGER\_ID,FNAME,LNAME) values ('1234560095','Apt983 ,Creek Road,Plano','GeorgeK3@work.com',5756058545,null,'George','Manager');

Insert into EMPLOYEE (EMPLOYEE\_ID,ADDRESS,EMAIL,PHONE\_NO,MANAGER\_ID,FNAME,LNAME) values ('1234561075','345 ,Street x ,Richardson','KumarG3@work.com',4756058545,'1234560095','Gaurav','Kumar');

Insert into EMPLOYEE (EMPLOYEE\_ID,ADDRESS,EMAIL,PHONE\_NO,MANAGER\_ID,FNAME,LNAME) values ('1234551099','345 ,Street x ,Richardson','KumarA3@work.com',4756058545,'1234551078','Akash','Kumar');

Insert into EMPLOYEE (EMPLOYEE\_ID,ADDRESS,EMAIL,PHONE\_NO,MANAGER\_ID,FNAME,LNAME) values ('1234551049','898 , road g , texas','Manager4@work.com',7864534009,null,'Manager','Singh');

Insert into EMPLOYEE (EMPLOYEE\_ID,ADDRESS,EMAIL,PHONE\_NO,MANAGER\_ID,FNAME,LNAME) values ('1234561098','7545 Collin Creek ,plano','Manager9@work.com',7737814829,null,'Marie','Brown');

Insert into EMPLOYEE (EMPLOYEE\_ID,ADDRESS,EMAIL,PHONE\_NO,MANAGER\_ID,FNAME,LNAME) values ('1234551078','7777 Mccallum blvd ,dallas','Manager10@work.com',7374367583,null,'Jason','Black');

Insert into EMPLOYEE (EMPLOYEE\_ID,ADDRESS,EMAIL,PHONE\_NO,MANAGER\_ID,FNAME,LNAME) values ('1234551089','7878, Street downtown 8','EmployeeApurva@work.com',9786345982,'1234551078','Apurva','Patel');

Insert into VEHICLE\_MODEL (VEHICLE\_MOD\_ID,VEHICLE\_NAME,YEAR,DESCRIPTION) values ('KOEN111337','ONE:1',2015,'Imported');

Insert into VEHICLE\_MODEL (VEHICLE\_MOD\_ID,VEHICLE\_NAME,YEAR,DESCRIPTION) values ('PORSCHE911','918',2014,'Hybrid');

Insert into VEHICLE\_MODEL (VEHICLE\_MOD\_ID,VEHICLE\_NAME,YEAR,DESCRIPTION) values ('LAMBOKING1','Sesto Elemento',2011,'Track only');

Insert into VEHICLE\_MODEL (VEHICLE\_MOD\_ID,VEHICLE\_NAME,YEAR,DESCRIPTION) values ('FERRARI012','FXX-K',2015,'Track only');

Insert into VEHICLE\_MODEL (VEHICLE\_MOD\_ID,VEHICLE\_NAME,YEAR,DESCRIPTION) values ('McLaren','P1',2014,'Limited Edition');

Insert into MANUFACTURER (MAN\_ID,NAME,ADDRESS,PHONE\_NO) values ('BMW1200001','BMW Corp','Street1337,Berlin,Germany',5620017654);

Insert into MANUFACTURER (MAN\_ID,NAME,ADDRESS,PHONE\_NO) values ('Lambo31002','Lamborghini Corp ,Volkswagen Group',' Sant''Agata Bolognese, Italy',8725961102);

Insert into MANUFACTURER (MAN\_ID,NAME,ADDRESS,PHONE\_NO) values ('Ferrari031','Ferrari S.P.A',' Maranello, Italy',6225961102);

Insert into MANUFACTURER (MAN\_ID,NAME,ADDRESS,PHONE\_NO) values ('Mclaren0p1','McLaren Automotive','Woking,surey,Britain',3225961102);

Insert into MANUFACTURER (MAN\_ID,NAME,ADDRESS,PHONE\_NO) values ('Chevy0p123','General Motors','Detroit, USA',9225961102);

Insert into LOCATION (LOC\_ID,NAME,ADDRESS,PHONE,MANAGER\_ID) values ('LOC001','Dallas','35 Collin Creek',98217827,'1234567890');

Insert into LOCATION (LOC\_ID,NAME,ADDRESS,PHONE,MANAGER\_ID) values ('LOC002','New York','43 Manhattan Drive',531237872,'1234560095');

Insert into LOCATION (LOC\_ID,NAME,ADDRESS,PHONE,MANAGER\_ID) values ('LOC003','Seattle','56 Street',8123382173,'1234551049');

Insert into LOCATION (LOC\_ID,NAME,ADDRESS,PHONE,MANAGER\_ID) values ('LOC004','Miami','12 Street',55464566,'1234561098');

Insert into LOCATION (LOC\_ID,NAME,ADDRESS,PHONE,MANAGER\_ID) values ('LOC005','San Francsico','23 Street',8329218387,'1234551078');

Insert into INVENTORY (INVENTORY\_ID,NAME,CATEGORY,DESCRIPTION,QUANTITY,PRICE,MANUFACTURER\_ID) values ('INV001','Wheels','Wheels','Life time warranty',10000,5000,'BMW1200001');

Insert into INVENTORY (INVENTORY\_ID,NAME,CATEGORY,DESCRIPTION,QUANTITY,PRICE,MANUFACTURER\_ID) values ('INV002','GearBox','Transmission','Automatic or manual',350,30000,'Lambo31002');

Insert into INVENTORY (INVENTORY\_ID,NAME,CATEGORY,DESCRIPTION,QUANTITY,PRICE,MANUFACTURER\_ID) values ('INV003','Brakes','Braking','Power Brakes',250,800,'Ferrari031');

Insert into INVENTORY (INVENTORY\_ID,NAME,CATEGORY,DESCRIPTION,QUANTITY,PRICE,MANUFACTURER\_ID) values ('INV004','HeadLights','Chasis','Xeon Lights',500,3000,'Mclaren0p1');

Insert into INVENTORY (INVENTORY\_ID,NAME,CATEGORY,DESCRIPTION,QUANTITY,PRICE,MANUFACTURER\_ID) values ('INV005','Turbo','Engine','Supercharger',50,70000,'Chevy0p123');

Insert into INVENTORY\_LOCATION (LOCATION\_ID,INVENTORY\_ID) values ('LOC001','INV005');

Insert into INVENTORY\_LOCATION (LOCATION\_ID,INVENTORY\_ID) values ('LOC002','INV004');

Insert into INVENTORY\_LOCATION (LOCATION\_ID,INVENTORY\_ID) values ('LOC003','INV001');

Insert into INVENTORY\_LOCATION (LOCATION\_ID,INVENTORY\_ID) values ('LOC004','INV002');

Insert into INVENTORY\_LOCATION (LOCATION\_ID,INVENTORY\_ID) values ('LOC005','INV002');

Insert into INVENTORY\_LOCATION (LOCATION\_ID,INVENTORY\_ID) values ('LOC005','INV003');

Insert into MODEL\_INVENTORY (VEHICLE\_MODEL\_ID,INVENTORY\_ID) values ('KOEN111337','INV001');

Insert into MODEL\_INVENTORY (VEHICLE\_MODEL\_ID,INVENTORY\_ID) values ('PORSCHE911','INV002');

Insert into MODEL\_INVENTORY (VEHICLE\_MODEL\_ID,INVENTORY\_ID) values ('LAMBOKING1','INV003');

Insert into MODEL\_INVENTORY (VEHICLE\_MODEL\_ID,INVENTORY\_ID) values ('FERRARI012','INV004');

Insert into MODEL\_INVENTORY (VEHICLE\_MODEL\_ID,INVENTORY\_ID) values ('McLaren','INV005');

Insert into MODEL\_INVENTORY (VEHICLE\_MODEL\_ID,INVENTORY\_ID) values ('LAMBOKING1','INV001');

Insert into MODEL\_INVENTORY (VEHICLE\_MODEL\_ID,INVENTORY\_ID) values ('FERRARI012','INV002');

Insert into MODEL\_INVENTORY (VEHICLE\_MODEL\_ID,INVENTORY\_ID) values ('McLaren','INV001');

Insert into TOOL\_KIT (TOOLKIT\_ID,DESCRIPTION,EMPLOYEE\_ID,CUSTOMER\_ID,RENTDATE,RETURNDATE) values ('Kit0006712','Kit of Location001',null,null,null,null);

Insert into TOOL\_KIT (TOOLKIT\_ID,DESCRIPTION,EMPLOYEE\_ID,CUSTOMER\_ID,RENTDATE,RETURNDATE) values ('Kit0006714','Kit 2 of Location001',null,null,null,null);

Insert into TOOL\_KIT (TOOLKIT\_ID,DESCRIPTION,EMPLOYEE\_ID,CUSTOMER\_ID,RENTDATE,RETURNDATE) values ('Kit0006716','Kit of Location002',null,null,null,null);

Insert into TOOL\_KIT (TOOLKIT\_ID,DESCRIPTION,EMPLOYEE\_ID,CUSTOMER\_ID,RENTDATE,RETURNDATE) values ('Kit0006715','Kit of Location003',null,null,null,null);

Insert into TOOL\_KIT (TOOLKIT\_ID,DESCRIPTION,EMPLOYEE\_ID,CUSTOMER\_ID,RENTDATE,RETURNDATE) values ('Kit0006719','Kit of Location004',null,null,null,null);

Insert into TOOL\_KIT (TOOLKIT\_ID,DESCRIPTION,EMPLOYEE\_ID,CUSTOMER\_ID,RENTDATE,RETURNDATE) values ('Kit0006710','Kit of Location005',null,null,null,null);

Insert into TRANSACTION (TXN\_ID,TXN\_DATE,TOTAL,CUSTOMER\_ID,EMPLOYEE\_ID,LOCATION\_ID) values ('txnLoc1120',to\_date('02-DEC-15','DD-MON-RR'),40000,'0000000000','1234567891','LOC001');

Insert into TRANSACTION (TXN\_ID,TXN\_DATE,TOTAL,CUSTOMER\_ID,EMPLOYEE\_ID,LOCATION\_ID) values ('txnLoc1121',to\_date('06-DEC-15','DD-MON-RR'),50000,'0000000001','1234560092','LOC001');

Insert into TRANSACTION (TXN\_ID,TXN\_DATE,TOTAL,CUSTOMER\_ID,EMPLOYEE\_ID,LOCATION\_ID) values ('txnLoc2120',to\_date('25-NOV-15','DD-MON-RR'),35000,'0000000002','1234551099','LOC002');

Insert into TRANSACTION (TXN\_ID,TXN\_DATE,TOTAL,CUSTOMER\_ID,EMPLOYEE\_ID,LOCATION\_ID) values ('txnLoc2100',to\_date('04-DEC-15','DD-MON-RR'),71000,'0000000003','1234561075','LOC002');

Insert into TRANSACTION (TXN\_ID,TXN\_DATE,TOTAL,CUSTOMER\_ID,EMPLOYEE\_ID,LOCATION\_ID) values ('txnLoc1123',to\_date('21-OCT-15','DD-MON-RR'),2900,'0000000004','1234551089','LOC005');

Insert into TRANSACTION (TXN\_ID,TXN\_DATE,TOTAL,CUSTOMER\_ID,EMPLOYEE\_ID,LOCATION\_ID) values ('txnLoc3110',to\_date('24-NOV-15','DD-MON-RR'),5700,null,'1234551089','LOC005');

Insert into TRANSACTION (TXN\_ID,TXN\_DATE,TOTAL,CUSTOMER\_ID,EMPLOYEE\_ID,LOCATION\_ID) values ('txnLoc3111',to\_date('28-OCT-15','DD-MON-RR'),88000,null,'1234551089','LOC005');

Insert into TRANSACTION (TXN\_ID,TXN\_DATE,TOTAL,CUSTOMER\_ID,EMPLOYEE\_ID,LOCATION\_ID) values ('txnLoc2125',to\_date('08-DEC-15','DD-MON-RR'),93000,null,'1234561075','LOC002');

Insert into PURCHASE\_ITEM (TXN\_ID,INVENTORY\_ID,PURCHASE\_QUANTITY,PURCHASE\_PRICE) values ('txnLoc1120','INV001',8,100045);

Insert into PURCHASE\_ITEM (TXN\_ID,INVENTORY\_ID,PURCHASE\_QUANTITY,PURCHASE\_PRICE) values ('txnLoc1121','INV002',1,89000);

Insert into PURCHASE\_ITEM (TXN\_ID,INVENTORY\_ID,PURCHASE\_QUANTITY,PURCHASE\_PRICE) values ('txnLoc2120','INV003',3,45000);

Insert into PURCHASE\_ITEM (TXN\_ID,INVENTORY\_ID,PURCHASE\_QUANTITY,PURCHASE\_PRICE) values ('txnLoc2100','INV004',5,87000);

Insert into PURCHASE\_ITEM (TXN\_ID,INVENTORY\_ID,PURCHASE\_QUANTITY,PURCHASE\_PRICE) values ('txnLoc3110','INV003',1,50501);

Insert into PURCHASE\_ITEM (TXN\_ID,INVENTORY\_ID,PURCHASE\_QUANTITY,PURCHASE\_PRICE) values ('txnLoc3111','INV003',2,43435);

Insert into PURCHASE\_ITEM (TXN\_ID,INVENTORY\_ID,PURCHASE\_QUANTITY,PURCHASE\_PRICE) values ('txnLoc2125','INV005',7,77700);

Insert into PURCHASE\_ITEM (TXN\_ID,INVENTORY\_ID,PURCHASE\_QUANTITY,PURCHASE\_PRICE) values ('txnLoc1123','INV005',10,165000);

1. **Stored Procedures (PL/SQL)**

---PROCEDURE FOR TRANSACTION (GIVE LOYALTY POINTS FOR TRANSACTION AND UPDATE INVENTORY)---

create or replace PROCEDURE SALES\_TRANSACTION(INPUTTXNID IN TRANSACTION.TXN\_ID%TYPE)

AS

CUSID CUSTOMER.CUST\_ID%TYPE;

INVENTORYID INVENTORY.INVENTORY\_ID%TYPE;

ITEM\_TABLE PURCHASE\_ITEM%ROWTYPE;

QUANTY INVENTORY.QUANTITY%TYPE;

POINTS NUMBER ;

CURSOR INVNTRYUPDT

IS

SELECT \* FROM PURCHASE\_ITEM WHERE TXN\_ID = INPUTTXNID ;

BEGIN

OPEN INVNTRYUPDT;

LOOP

FETCH INVNTRYUPDT INTO ITEM\_TABLE;

EXIT WHEN (INVNTRYUPDT%NOTFOUND);

SELECT QUANTITY INTO QUANTY FROM INVENTORY WHERE INVENTORY\_ID=ITEM\_TABLE.INVENTORY\_ID ;

IF (ITEM\_TABLE.TXN\_ID = INPUTTXNID AND ITEM\_TABLE.PURCHASE\_QUANTITY <= QUANTY )

THEN

UPDATE INVENTORY SET QUANTITY = (QUANTITY - ITEM\_TABLE.PURCHASE\_QUANTITY) WHERE INVENTORY\_ID=ITEM\_TABLE.INVENTORY\_ID ;

END IF;

END LOOP;

CLOSE INVNTRYUPDT;

SELECT TOTAL INTO POINTS FROM TRANSACTION WHERE TXN\_ID = INPUTTXNID ;

SELECT CUSTOMER\_ID INTO CUSID FROM TRANSACTION WHERE TXN\_ID = INPUTTXNID;

UPDATE CUSTOMER SET LOYALTY\_POINTS = LOYALTY\_POINTS + POINTS where CUST\_ID = CUSID ;

END;

--- TESTING THE ABOVE PROCEDURE ----

set SERVEROUTPUT ON

begin SALES\_TRANSACTION('txnLoc2100'); end;

---PROCEDURE FOR RENTING A TOOL KIT (GIVE A START AND THE LAST DAY TO RETURN DATE FOR A GIVEN TOOLKIT FOR A CUSTOMER AND MARK CUSTOMER FOR HAVING A TOOLKIT RENTED)-----

create or replace PROCEDURE RENT\_TOOLKIT(INPUTKITID IN TOOL\_KIT.TOOLKIT\_ID%TYPE ,INPUTEMPLOYEEID IN EMPLOYEE.EMPLOYEE\_ID%TYPE ,INPUTCUSTOMERID IN CUSTOMER.CUST\_ID%TYPE )

AS

CUS\_ID CUSTOMER.CUST\_ID%TYPE;

RENTFLAG CUSTOMER.RENTAL\_FLAG%TYPE;

POINTS CUSTOMER.LOYALTY\_POINTS%TYPE;

BEGIN

SELECT CUSTOMER\_ID INTO CUS\_ID FROM TOOL\_KIT WHERE TOOLKIT\_ID = INPUTKITID;

SELECT RENTAL\_FLAG INTO RENTFLAG FROM CUSTOMER WHERE CUST\_ID= INPUTCUSTOMERID;

SELECT LOYALTY\_POINTS INTO POINTS FROM CUSTOMER WHERE CUST\_ID=INPUTCUSTOMERID;

IF (CUS\_ID IS NULL AND RENTFLAG = 0 AND POINTS >= 100)

THEN

UPDATE TOOL\_KIT SET RENTDATE= SYSDATE , CUSTOMER\_ID = INPUTCUSTOMERID , EMPLOYEE\_ID =INPUTEMPLOYEEID,RETURNDATE =trunc(sysdate)+POINTS WHERE TOOLKIT\_ID=INPUTKITID;

UPDATE CUSTOMER SET RENTAL\_FLAG = 1 WHERE CUST\_ID = INPUTCUSTOMERID ;

END IF ;

END ;

--- TESTING THE ABOVE PROCEDURE ----

set serveroutput on

begin RENT\_TOOLKIT('Kit0006712','1234567891','0000000003');end;

---PROCEDURE FOR RETURN TOOL KIT (RETURN TOOLKIT AND SUBTRACT LOYALTY POINTS)----

create or replace PROCEDURE RETURN\_TOOLKIT(INPUTKITID IN TOOL\_KIT.TOOLKIT\_ID%TYPE)

AS

CUSID CUSTOMER.CUST\_ID%TYPE;

RENTDAYS NUMBER;

BEGIN

SELECT CUSTOMER\_ID INTO CUSID FROM TOOL\_KIT WHERE TOOLKIT\_ID = INPUTKITID;

SELECT ROUND(SYSDATE - TK.RENTDATE)INTO RENTDAYS FROM TOOL\_KIT TK WHERE TK.TOOLKIT\_ID = INPUTKITID;

UPDATE CUSTOMER SET RENTAL\_FLAG=0,LOYALTY\_POINTS=LOYALTY\_POINTS-RENTDAYS WHERE CUST\_ID=CUSID;

UPDATE TOOL\_KIT SET RENTDATE = null , CUSTOMER\_ID = null , EMPLOYEE\_ID = null ,ReturnDate = null where ToolKit\_id =INPUTKITID;

END;

--- TESTING THE ABOVE PROCEDURE ----

set serveroutput on

begin RETURN\_TOOLKIT('Kit0006712');end;

---PROCEDURE FOR REVENUE REPORT (GENERATE REVENUE REPORT FOR GIVEN DATES) ---

create or replace PROCEDURE Revenue\_Report(from\_date in Date,till\_date in Date)

as

txn\_total NUMBER;

txn\_count number;

thistxn transaction%ROWTYPE;

Cursor txns is select \* from TRANSACTION where TXN\_DATE between from\_date and till\_date;

begin

select sum(total),count(\*) into txn\_total,txn\_count from TRANSACTION where TXN\_DATE between from\_date and till\_date;

open txns;

loop

fetch txns into thistxn;

exit when (txns%NOTFOUND);

dbms\_output.put\_line('TransactionID:' || thistxn.txn\_id ||'Date:'|| thistxn.txn\_date||'Price:$'||thistxn.total);

end loop;

close txns;

dbms\_output.put\_line('In total '|| txn\_count||' transactions between '||from\_date||' and '||till\_date||'generating revenue of $'||txn\_total);

end;

--- TESTING THE ABOVE PROCEDURE ----

set serveroutput on

begin Revenue\_Report('01-OCT-15','31-DEC-15');end;