AUTO PARTS STORE

CS6360-002 FINAL PROJECT

PRATHIK KULKARNI (pvk170130)

SUMIN REDDY GUJJA (sxg172130)

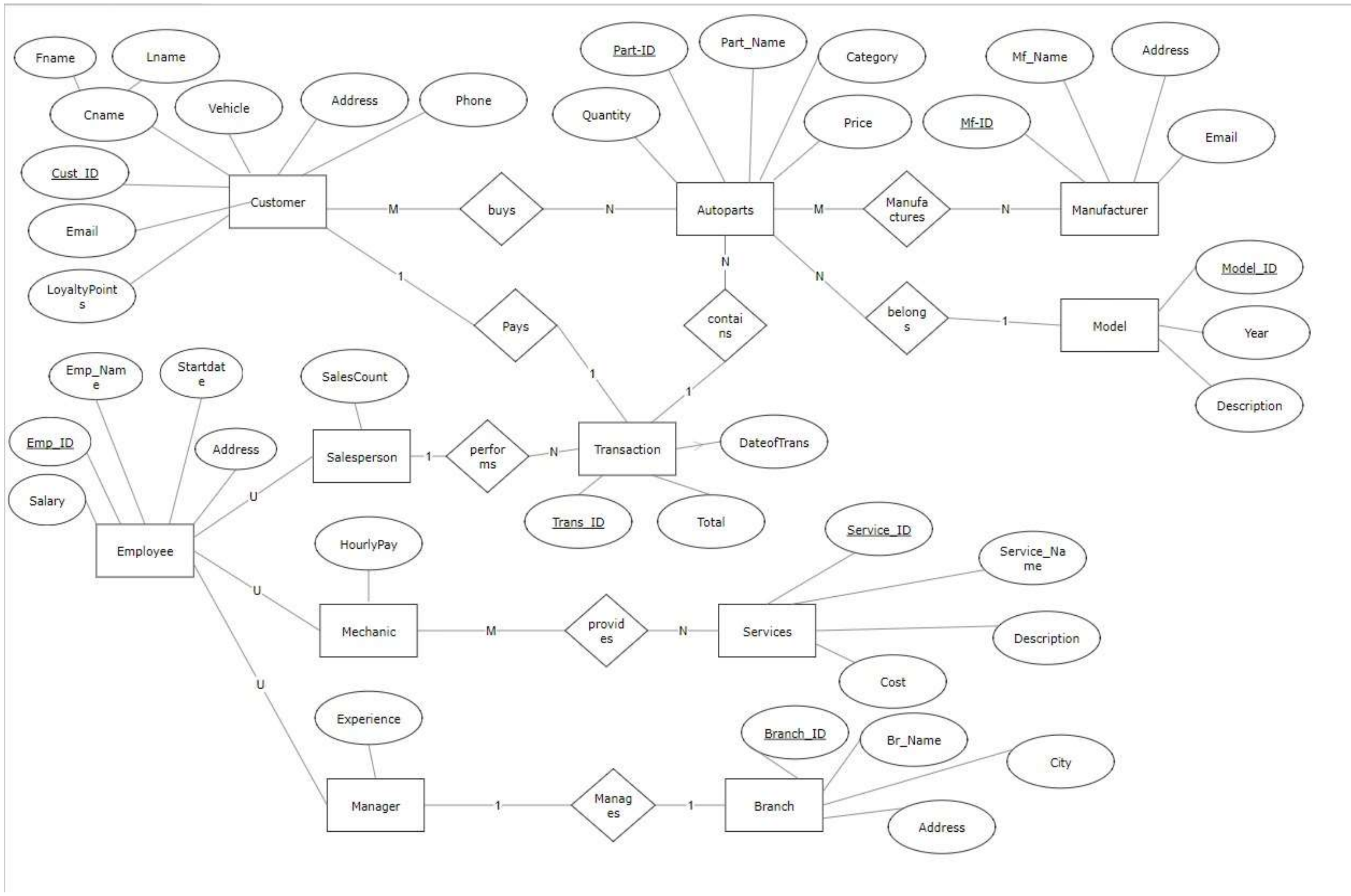
VENKATESH GOTIMUKUL (vxg173330)

TABLE OF CONTENTS

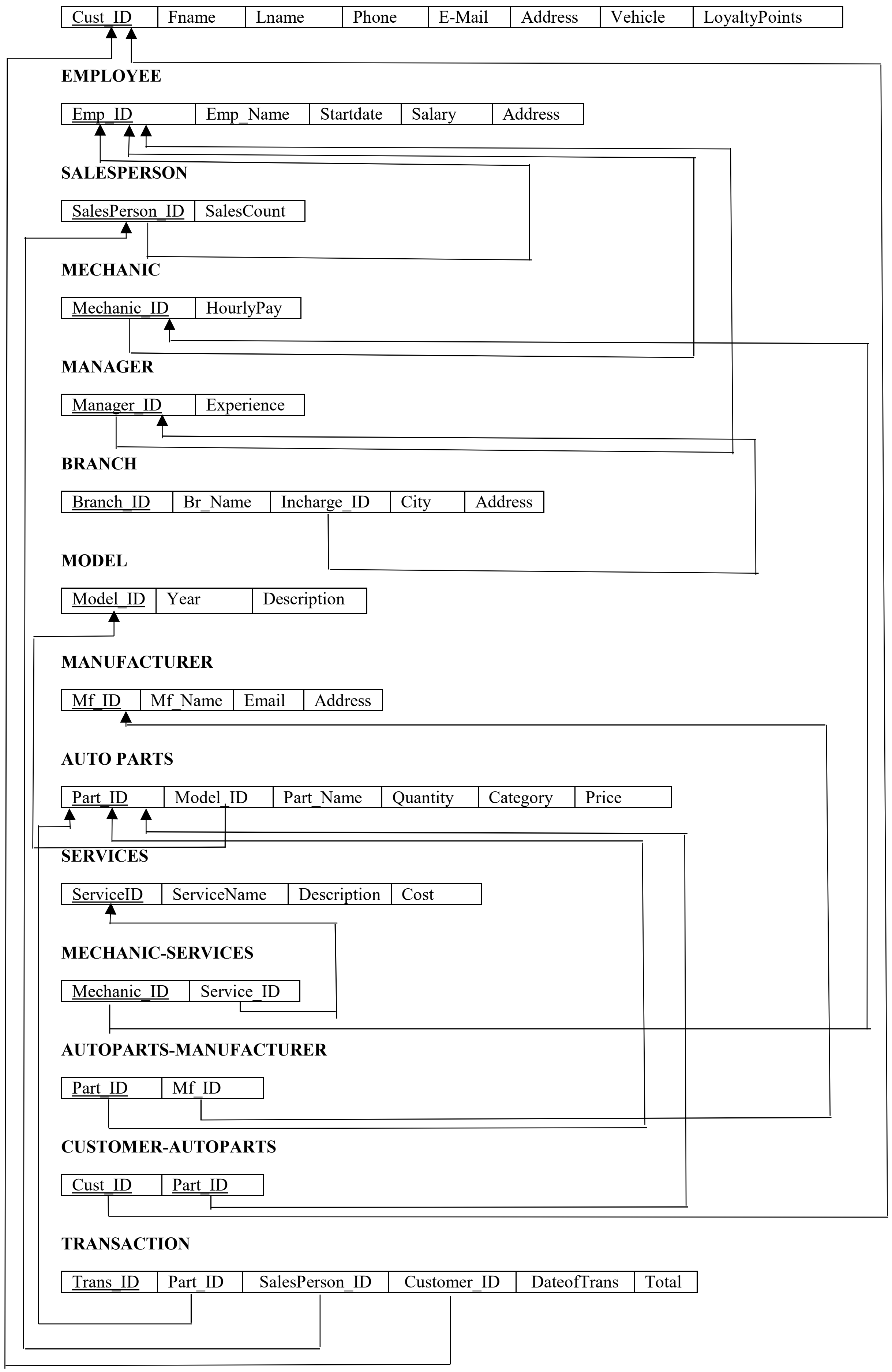
I. Data requirements

1. An auto parts store consists of auto parts which belong to a certain category and are manufactured by different manufacturers.
2. The auto parts store system should keep track of auto parts available, sales transactions and other auxiliary services provided by the store. It also keeps track of its customers, employees and locations.
3. The auto parts part store has different branches in different locations. The inventory quantities may be different for different branches. Each branch consists of a unique id, name, address, city and manager assigned to it.
4. There are three types of employees, salesperson- who performs the sales transaction for auto parts, mechanic- who provides repair services for the customer and manager- who manages specific branch. Each employee has a unique id, name, start date, phone, address and salary. Salesperson and mechanics have managers assigned to them.
5. Salesperson generates an invoice when an auto part is to be sold to a customer, and the customer pays the amount specified in the invoice to complete the transaction. Each sales transaction has unique id, customer involved, salesperson who sells the auto part, total amount and auto parts list.
6. Customers are divided into new customers and returning customers. The customers will have unique id, name, vehicle owned, address, phone and email. Returning customers are awarded loyalty points.
7. Auto parts inventory keeps track of the available auto parts. It is updated once auto part is sold or when a new auto part is obtained from the manufacturer.
8. The system also keeps track of different manufacturers. They have unique id, name, address, phone and email.
9. The different vehicle models for which auto parts are available is also kept track off. They have unique id, name and description.
10. The auxiliary services provided are also kept track off. They have unique id, name, description and cost.

II. ER Diagram



III. Relational Mapping(already in 3NF)



CUSTOMER

Cust_ID	Fname	Lname	Phone	E-Mail	Address	Vehicle	LoyaltyPoints

EMPLOYEE

Emp_ID	Emp_Name	Startdate	Salary	Address

SALESPERSON

SalesPerson_ID	SalesCount

MECHANIC

Mechanic_ID	HourlyPay

MANAGER

Manager_ID	Experience

BRANCH

Branch_ID	Br_Name	Incharge_ID	City	Address

MODEL

Model_ID	Year	Description

MANUFACTURER

Mf_ID	Mf_Name	Email	Address

AUTO PARTS

Part_ID	Model_ID	Part_Name	Quantity	Category	Price

SERVICES

ServiceID	ServiceName	Description	Cost

MECHANIC-SERVICES

Mechanic_ID	Service_ID

AUTOPARTS-MANUFACTURER

Part_ID	Mf_ID

CUSTOMER-AUTOPARTS

Cust_ID	Part_ID

TRANSACTION

Trans_ID	Part_ID	SalesPerson_ID	Customer_ID	DateofTrans	Total

IV. Functional Dependencies

Cust_ID->Fname, Lname, Phone, E-mail, Address, Vehicle, Loyalty_Points

Emp_ID-> Emp_Name, Startdate, Salary, Address

SalesPerson_ID-> SalesCount

Mechanic_ID-> HourlyPay

Manager_ID-> Experience

Branch_ID-> Br_Name, Incharge_ID, City, Address

Model_ID-> Year, Description

Mf_ID-> Mf_Name, Email, Address

Part_ID-> Model_ID, Part_Name, Quantity, Category, Price

ServiceID-> ServiceName, Description, Cost

Trans_ID-> Part_ID, SalesPerson_ID, Customer_ID, DateofTrans, Total

V. Database Normalization

- The initial schema does not have any transitive dependencies and every attribute is dependent on the primary key, there does not need to be any decomposition.
- 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 has to be created.
- The relationships with 1-to-N and 1-to-1 are five in number and there is no need to create a new table for those relationships instead adding the primary key reference is enough.

VI. 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.

VII. Create Statements (SQL)

```
drop table customer cascade constraints;

Create Table CUSTOMER

(

Cust_ID VARCHAR(10) NOT NULL,

Fname   VARCHAR(35),

Lname   VARCHAR(35),

Address VARCHAR(35),
Email VARCHAR(35),
Vehicle VARCHAR(35),
Phone VARCHAR (10),
LoyaltyPoints VARCHAR (10),
CONSTRAINT CUSTPK PRIMARY KEY(Cust_ID)
);

drop table employee cascade constraints;
Create Table EMPLOYEE
(
Emp_ID VARCHAR(10) NOT NULL,
Emp_Name VARCHAR(35),
Address VARCHAR(35),
Startdate DATE,
Salary VARCHAR(10),
CONSTRAINT EMPPK PRIMARY KEY (Emp_ID)
);

drop table salesperson cascade constraints;
Create Table SALESPERSON
(
SalesCount VARCHAR(30),
Salesperson_ID VARCHAR(10) unique,
CONSTRAINT SPFK FOREIGN KEY (Salesperson_ID) references EMPLOYEE(Emp_ID)
);

drop table mechanic;
Create Table MECHANIC
(
HourlyPay VARCHAR(30),
Mechanic_ID VARCHAR(10) unique,
CONSTRAINT MEFK FOREIGN KEY (Mechanic_ID)references EMPLOYEE(Emp_ID)
);
```

```
alter table mechanic add constraint mechanic_ID unique (mechanic_ID);

drop table manager;
Create Table MANAGER
(
Experience VARCHAR(30),
Manager_ID VARCHAR(10) UNIQUE,
CONSTRAINT MAFK FOREIGN KEY (Manager_ID)references EMPLOYEE(Emp_ID)
);

drop table BRANCH;
Create Table BRANCH
(
Branch_ID VARCHAR(10) NOT NULL,
Br_Name VARCHAR(35),
Incharge_ID VARCHAR(10),
Address VARCHAR(35),
City VARCHAR (50)NOT NULL,
CONSTRAINT BRANCHPK PRIMARY KEY (Branch_ID),
CONSTRAINT BRANCHFK FOREIGN key (Incharge_ID) references MANAGER(Manager_ID)
);

drop table model cascade constraints;
Create Table MODEL
(
Model_ID VARCHAR(10) NOT NULL,
Year VARCHAR(10) NOT NULL,
Description VARCHAR(30),
CONSTRAINT MODELPK PRIMARY KEY(Model_ID)
);

drop table manufacturer cascade constraints;
```

```
Create table Manufacturer

(

Mf_ID VARCHAR(10),

Mf_Name VARCHAR(35) NOT NULL,

Email VARCHAR(50),

Address VARCHAR(35) DEFAULT 'NOT AVAILABLE',

CONSTRAINT MANPK PRIMARY KEY(Mf_ID)

);


drop table autoparts cascade CONSTRAINTS;

Create table AUTOPARTS

(

Part_ID VARCHAR(10),

Mod_ID VARCHAR(8),

Part_Name VARCHAR(35) NOT NULL,

Quantity VARCHAR(30),

Category VARCHAR(30),

Price VARCHAR(30),

CONSTRAINT AUTOPK PRIMARY KEY(Part_ID),

CONSTRAINT AUTOFK FOREIGN KEY(Mod_ID) REFERENCES MODEL(Model_ID)

);


drop table services CASCADE CONSTRAINTS;

Create table SERVICES

(

Service_ID VARCHAR(10),

Service_Name VARCHAR(35) NOT NULL,

Cost VARCHAR (30),

Description VARCHAR(30),

CONSTRAINT SERVPK PRIMARY KEY(Service_ID)

);


drop table transaction cascade constraints;
```

```
Create table TRANSACTION
(
Trans_ID VARCHAR(10),
DateofTrans Date NOT NULL,
Total VARCHAR (10),
Customer_ID VARCHAR(10),
Salesperson_ID VARCHAR(10),
Part_ID VARCHAR(10),
CONSTRAINT TRANSPK PRIMARY KEY (Trans_ID),
CONSTRAINT TRANSFK1 FOREIGN key (Customer_ID) references CUSTOMER(Cust_ID),
CONSTRAINT TRANSFK2 FOREIGN key (Salesperson_ID) references SALESPERSON(Salesperson_ID),
CONSTRAINT TRANSFK4 FOREIGN KEY (Part_ID) references AUTOPARTS(Part_ID)
);

drop table mechanic_services cascade constraints;
Create table mechanic_services
(
Mechanic_ID VARCHAR(10),
Service_ID VARCHAR(10),
CONSTRAINT MSFK1 FOREIGN key (Mechanic_ID) references MECHANIC(Mechanic_ID),
CONSTRAINT MSFK2 FOREIGN key (Service_ID) references SERVICES(Service_ID),
CONSTRAINT MSPK PRIMARY KEY(Mechanic_ID,Service_ID)
);

drop table autoparts_manufacturer cascade constraints;
Create table autoparts_manufacturer
(
Part_ID VARCHAR(10),
Mf_ID VARCHAR(10),
CONSTRAINT AMFK1 FOREIGN key (Part_ID) references AUTOPARTS(Part_ID),
CONSTRAINT AMFK2 FOREIGN key (Mf_ID) references MANUFACTURER(Mf_ID),
CONSTRAINT AMPK PRIMARY KEY(Part_ID,Mf_ID)
);
```

```
drop table customer_autoparts cascade constraints;

Create table customer_autoparts

(

Cust_ID VARCHAR(10),

Part_ID VARCHAR(10),

CONSTRAINT CAFK1 FOREIGN key (Cust_ID) references CUSTOMER(Cust_ID),

CONSTRAINT CAFK2 FOREIGN key (Part_ID) references AUTOPARTS(Part_ID),

CONSTRAINT CAPK PRIMARY KEY(Cust_ID,Part_ID)

);
```

VIII. Insert Statements (SQL)

```
desc customer;

select * from customer;

insert into customer values ('c1','john','cole','frankford','c1@gmail.com','Ford',8989898989,25);

insert into customer values ('c2','chris','davis','mccallum','c2@gmail.com','Mustang',8989898990,30);

insert into customer values ('c3','billy','chuck','palencia','c3@gmail.com','camaro',8989898991,35);

select * from customer;


desc employee;

select * from employee;

insert into employee values('e1','morrison','ashwood', '1998-01-01','10000');

insert into employee values('e2','david','estates',' 1999-02-02','20000');

insert into employee values('e3','jennifer','campbell', ' 2000-03-03','30000');

insert into employee values('e4','jenny','coit', ' 2001-04-04' ,'40000');

insert into employee values('e5','kathy','northplane', ' 2002-05-05' ,'50000');

insert into employee values('e6','christopher','plano', ' 2003-06-06','55000');

insert into employee values('e7','krish','richardson', ' 2004-07-07','60000');

insert into employee values('e8','daniel','frisco', ' 2005-08-08','65000');

insert into employee values('e9','ethan','freshno', ' 2006-09-09','70000');

select * from employee;


desc salesperson;
```

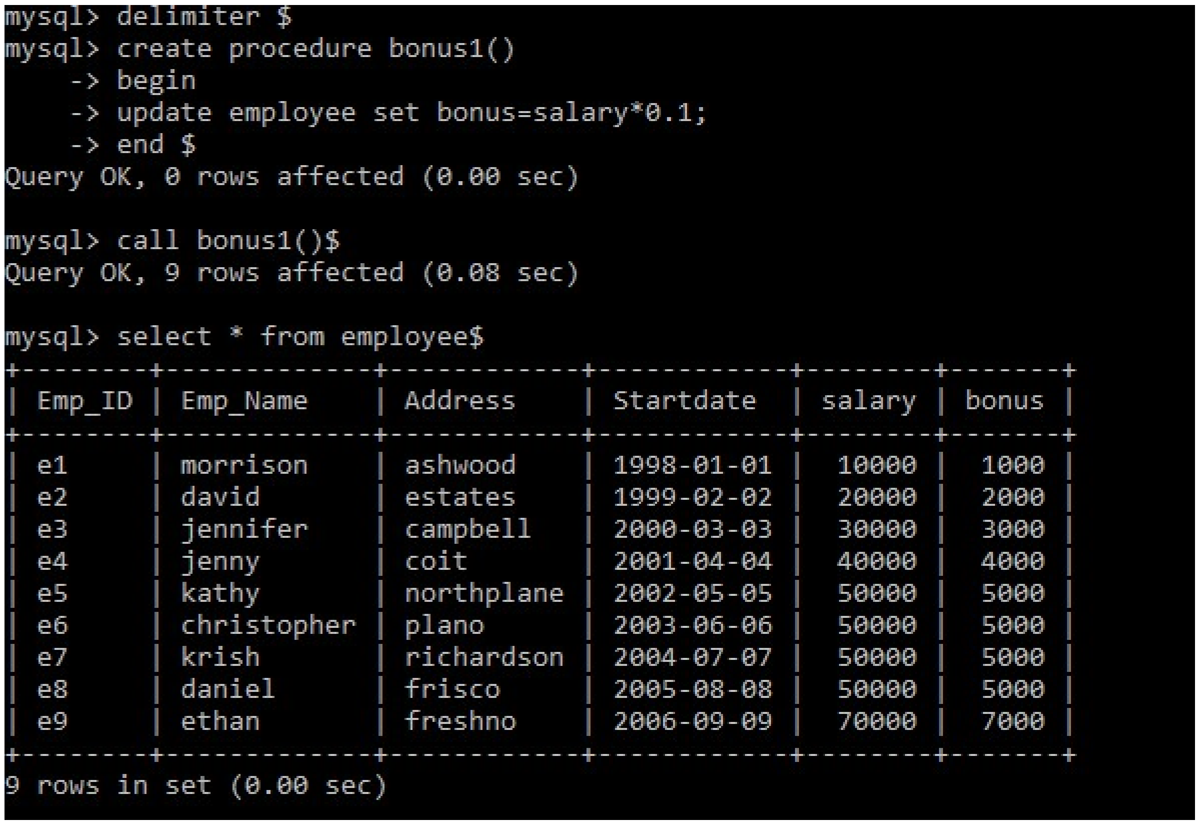
```
select * from salesperson;
insert into salesperson values(50,'e1');
insert into salesperson values(100,'e2');
insert into salesperson values(150,'e3');
select * from salesperson;

desc mechanic;
select * from mechanic;
insert into mechanic values(10,'e4');
insert into mechanic values(15,'e5');
insert into mechanic values(20,'e6');
select * from mechanic;

desc manager;
select * from manager;
insert into manager values(1,'e7');
insert into manager values(2,'e8');
insert into manager values(3,'e9');

desc branch;
select * from branch;
insert into branch values('b1','cloes','e8','inwood','dallas');
insert into branch values('b2','joes','e9','northwood','seattle');
insert into branch values('b3','harrys','e7','westwood','mckinney');
select * from branch;

desc model;
select * from model;
insert into model values('m1',2001,'sedan');
insert into model values('m2',2002,'hatchback');
insert into model values('m3',2005,'SUV');
select * from model;
```

```
desc manufacturer;
select * from manufacturer;
insert into manufacturer values('mf1','courtney','mf1@gmail.com','washington');
insert into manufacturer values('mf2','phoebe','mf2@gmail.com','houston');
insert into manufacturer values('mf3','martha','mf3@gmail.com','southlake');
select * from manufacturer;

desc autoparts;
select * from autoparts;
insert into autoparts values('p1','m2','gear','20','gearsystem','500');
insert into autoparts values('p2','m3','tyre','25','locomotion','1000');
insert into autoparts values('p3','m1','windshield','30','transparentscreens','1500');
select * from autoparts;
desc services;
select * from services;
insert into services values('s1','washing',1000,'deep cleanse');
insert into services values('s2','wiping',1200,'drying');
insert into services values('s3','oil change',1400,'lubrication');
select * from services;

desc transaction;
select * from transaction;
insert into transaction values('t1', '2006-09-09',2500,'c2','e2','p2');
insert into transaction values('t2', '2006-10-10',2700,'c3','e3','p3');
insert into transaction values('t3', '2006-11-11',2900,'c1','e1','p1');
select * from transaction;

desc mechanic_services;
select * from mechanic_services;
insert into mechanic_services values('e5','s2');
insert into mechanic_services values('e6','s3');
insert into mechanic_services values('e4','s1');
select * from mechanic_services;
```

```
desc autoparts_manufacturer;

select * from autoparts_manufacturer;

insert into autoparts_manufacturer values('p2','mf2');

insert into autoparts_manufacturer values('p3','mf3');

insert into autoparts_manufacturer values('p1','mf1');

select * from autoparts_manufacturer;


desc customer_autoparts;

select * from customer_autoparts;

insert into customer_autoparts values('c2','p2');

insert into customer_autoparts values('c3','p3');

insert into customer_autoparts values('c1','p1');

select * from customer_autoparts;
```

IX. PL/SQL Stored Procedures

```
Alter table employee add column bonus int;

-----PROCEDURE FOR EMPLOYEE (GIVE THE BONUS AMOUNT FOR EMPLOYEES)-----

delimiter $

create procedure bonus1()

begin

update employee set bonus=salary*0.1;

end $

--EXECUTION—

Call bonus1()$
```

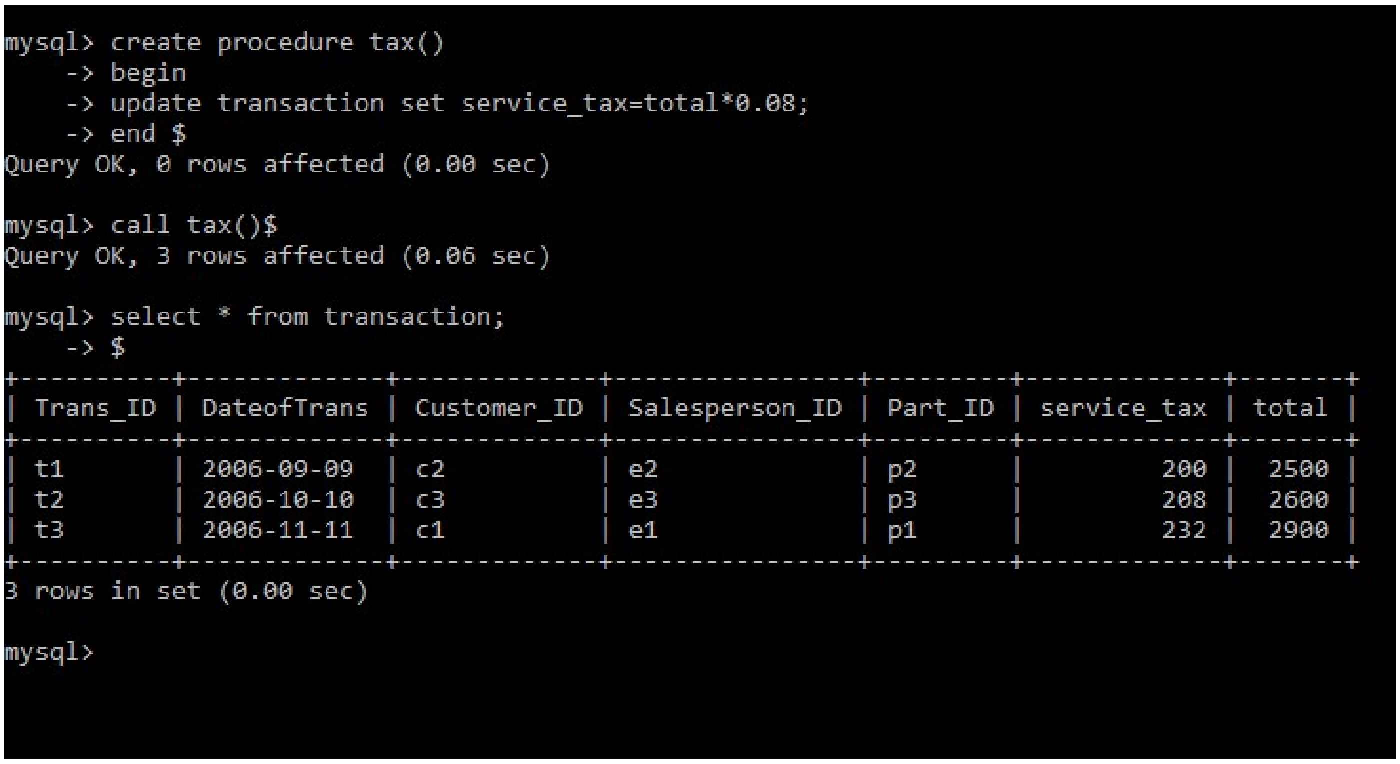


```
mysql> delimiter $
mysql> create procedure bonus1()
    -> begin
    -> update employee set bonus=salary*0.1;
    -> end $
Query OK, 0 rows affected (0.00 sec)

mysql> call bonus1()$
Query OK, 9 rows affected (0.08 sec)

mysql> select * from employee$
+--------+-------------+------------+------------+--------+-------+
| Emp_ID | Emp_Name    | Address    | Startdate  | salary | bonus |
+--------+-------------+------------+------------+--------+-------+
| e1     | morrison    | ashwood    | 1998-01-01 |  10000 |  1000 |
| e2     | david       | estates    | 1999-02-02 |  20000 |  2000 |
| e3     | jennifer    | campbell   | 2000-03-03 |  30000 |  3000 |
| e4     | jenny       | coit       | 2001-04-04 |  40000 |  4000 |
| e5     | kathy       | northplane | 2002-05-05 |  50000 |  5000 |
| e6     | christopher | plano      | 2003-06-06 |  50000 |  5000 |
| e7     | krish       | richardson | 2004-07-07 |  50000 |  5000 |
| e8     | daniel      | frisco     | 2005-08-08 |  50000 |  5000 |
| e9     | ethan       | freshno    | 2006-09-09 |  70000 |  7000 |
+--------+-------------+------------+------------+--------+-------+
9 rows in set (0.00 sec)
```

```
Alter table transaction add column service_tax int;
```

-----PROCEDURE FOR TRANSACTION (GIVE THE SERVICE_TAX AMOUNT FOR TRANSACTION)-----

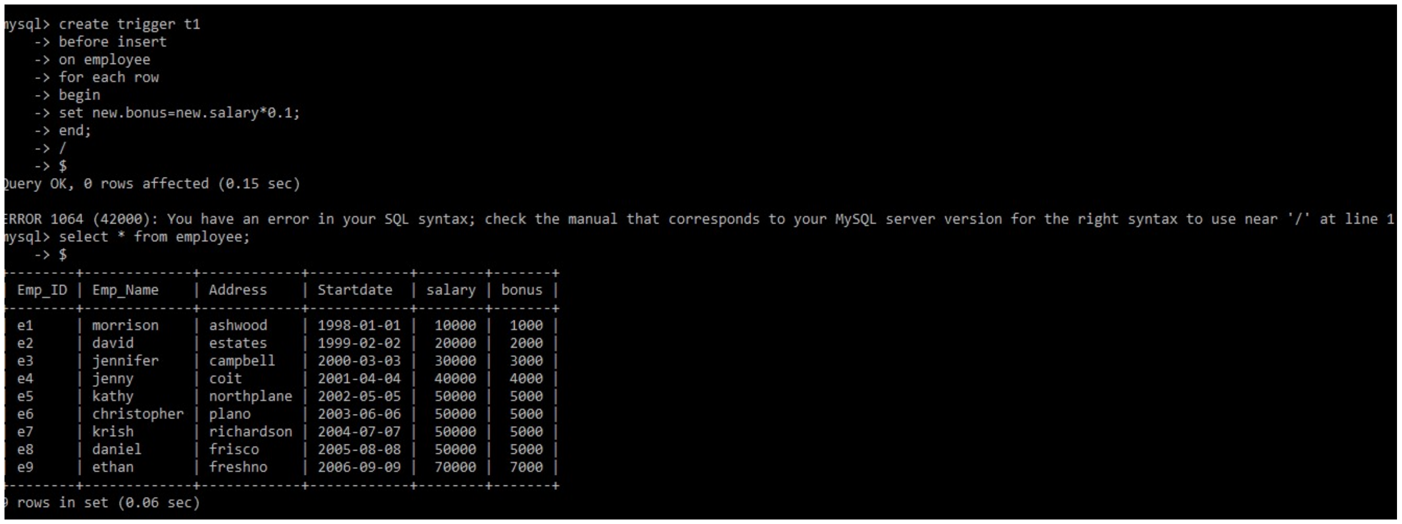
```
delimiter $

create procedure tax()

begin

update transaction set service_tax=total*0.08;

end $

--EXECUTION—

Call tax()$
```



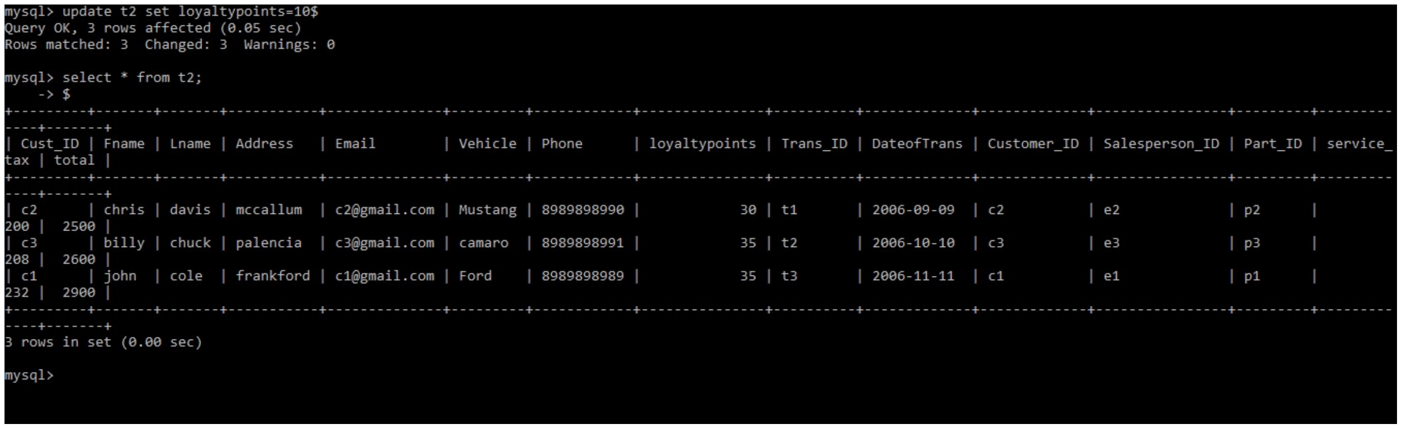
```
mysql> create procedure tax()
    -> begin
    -> update transaction set service_tax=total*0.08;
    -> end $
Query OK, 0 rows affected (0.00 sec)

mysql> call tax()$
Query OK, 3 rows affected (0.06 sec)

mysql> select * from transaction;
    -> $
+----------+------------+-------------+----------------+---------+-------------+-------+
| Trans_ID | DateofTrans | Customer_ID | Salesperson_ID | Part_ID | service_tax | total |
+----------+------------+-------------+----------------+---------+-------------+-------+
| t1       | 2006-09-09 | c2          | e2             | p2      |         200 |  2500 |
| t2       | 2006-10-10 | c3          | e3             | p3      |         208 |  2600 |
| t3       | 2006-11-11 | c1          | e1             | p1      |         232 |  2900 |
+----------+------------+-------------+----------------+---------+-------------+-------+
3 rows in set (0.00 sec)

mysql>
```

X. PL/SQL Triggers

```
-----TRIGGER FOR CUSTOMER-TRANSACTION JOIN TABLE(UPDATE THE LOYALTY POINTS FOR
CUSTOMERS ON TRANSACTION)-----


create table t2 as (select * from customer a, transaction b where
a.Cust_ID=b.Customer_ID)$

create trigger t4

before update

on t2

for each row

set new.loyaltypoints=old.loyaltypoints+5

$

--EXECUTION—

update t2 set loyaltypoints=10$
```



```
ysql> create trigger t1
    -> before insert
    -> on employee
    -> for each row
    -> begin
    -> set new.bonus=new.salary*0.1;
    -> end;
    -> /
    -> $
Query OK, 0 rows affected (0.15 sec)

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '/' at line 1
ysql> select * from employee;
    -> $
--------+-------------+-----------+------------+--------+-------+
 Emp_ID | Emp_Name    | Address   | Startdate  | salary | bonus |
--------+-------------+-----------+------------+--------+-------+
 e1     | morrison    | ashwood   | 1998-01-01 |  10000 |  1000 |
 e2     | david       | estates   | 1999-02-02 |  20000 |  2000 |
 e3     | jennifer    | campbell  | 2000-03-03 |  30000 |  3000 |
 e4     | jenny       | coit      | 2001-04-04 |  40000 |  4000 |
 e5     | kathy       | northplane| 2002-05-05 |  50000 |  5000 |
 e6     | christopher | plano     | 2003-06-06 |  50000 |  5000 |
 e7     | krish       | richardson| 2004-07-07 |  50000 |  5000 |
 e8     | daniel      | frisco    | 2005-08-08 |  50000 |  5000 |
 e9     | ethan       | freshno   | 2006-09-09 |  70000 |  7000 |
--------+-------------+-----------+------------+--------+-------+
 rows in set (0.06 sec)
```



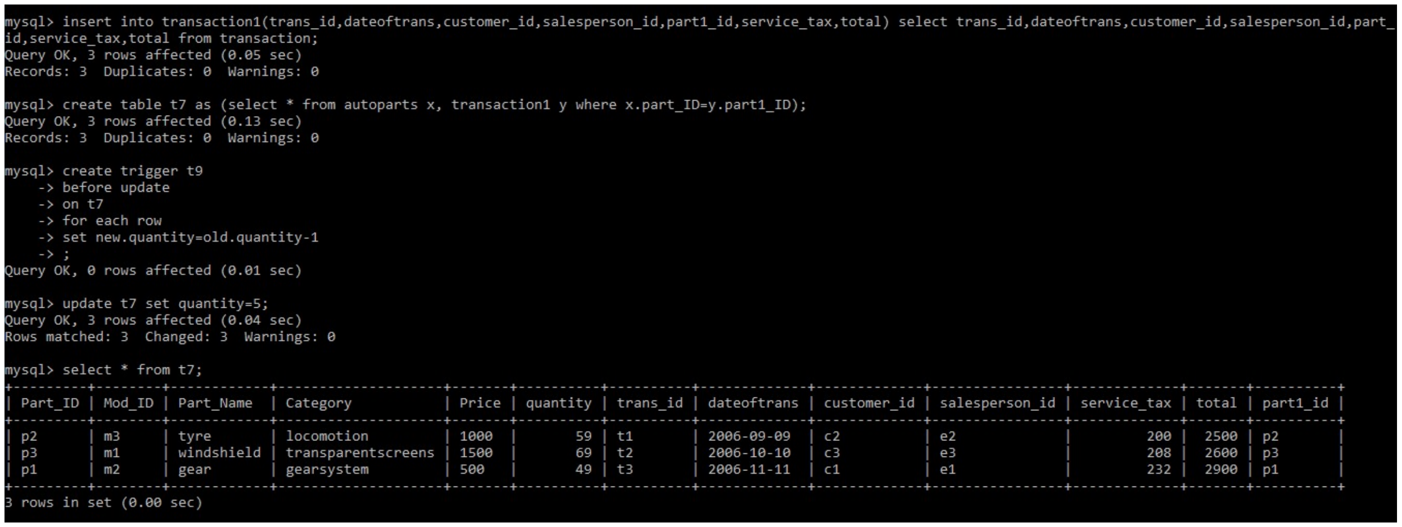
```
mysql> update t2 set loyaltypoints=10$
Query OK, 3 rows affected (0.05 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> select * from t2;
    -> $
+---------+-------+-------+----------+-------------+---------+------------+---------------+----------+-------------+-------------+----------------+---------+--------
----+-------+
| Cust_ID | Fname | Lname | Address  | Email       | Vehicle | Phone      | loyaltypoints | Trans_ID | DateofTrans | Customer_ID | Salesperson_ID | Part_ID | service_
tax | total |
+---------+-------+-------+----------+-------------+---------+------------+---------------+----------+-------------+-------------+----------------+---------+--------
----+-------+
| c2      | chris | davis | mccallum | c2@gmail.com | Mustang | 8989898990 |            30 | t1       | 2006-09-09  | c2          | e2             | p2      |
200 |  2500 |
| c3      | billy | chuck | palencia | c3@gmail.com | camaro  | 8989898991 |            35 | t2       | 2006-10-10  | c3          | e3             | p3      |
208 |  2600 |
| c1      | john  | cole  | frankford | c1@gmail.com | Ford    | 8989898989 |            35 | t3       | 2006-11-11  | c1          | e1             | p1      |
232 |  2900 |
+---------+-------+-------+----------+-------------+---------+------------+---------------+----------+-------------+-------------+----------------+---------+--------
----+-------+
3 rows in set (0.00 sec)

mysql>
```

-----TRIGGER FOR AUTOPART-TRANSACTION JOIN TABLE(UPDATE THE QUANTITY IN AUTOPARTS ON TRANSACTION)-----

```
create table t7 as (select * from autoparts x, transaction1 y where
x.part_ID=y.part1_ID);

create trigger t9

before update

on t7

for each row

set new.quantity=old.quantity-1

;
```

--EXECUTION—

```
update t7 set quantity=5;
```



```
mysql> insert into transaction1(trans_id,dateoftrans,customer_id,salesperson_id,part1_id,service_tax,total) select trans_id,dateoftrans,customer_id,salesperson_id,part_
id,service_tax,total from transaction;
Query OK, 3 rows affected (0.05 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> create table t7 as (select * from autoparts x, transaction1 y where x.part_ID=y.part1_ID);
Query OK, 3 rows affected (0.13 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> create trigger t9
    -> before update
    -> on t7
    -> for each row
    -> set new.quantity=old.quantity-1
    -> ;
Query OK, 0 rows affected (0.01 sec)

mysql> update t7 set quantity=5;
Query OK, 3 rows affected (0.04 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> select * from t7;
+---------+--------+------------+--------------------+-------+----------+----------+-------------+-------------+----------------+-------------+-------+----------+
| Part_ID | Mod_ID | Part_Name  | Category           | Price | quantity | trans_id | dateoftrans | customer_id | salesperson_id | service_tax | total | part1_id |
+---------+--------+------------+--------------------+-------+----------+----------+-------------+-------------+----------------+-------------+-------+----------+
| p2      | m3     | tyre       | locomotion         | 1000  |       59 | t1       | 2006-09-09  | c2          | e2             |         200 |  2500 | p2       |
| p3      | m1     | windshield | transparentscreens | 1500  |       69 | t2       | 2006-10-10  | c3          | e3             |         208 |  2600 | p3       |
| p1      | m2     | gear       | gearsystem         | 500   |       49 | t3       | 2006-11-11  | c1          | e1             |         232 |  2900 | p1       |
+---------+--------+------------+--------------------+-------+----------+----------+-------------+-------------+----------------+-------------+-------+----------+
3 rows in set (0.00 sec)
```