**Database Management Systems**

**Assignment One:**

**Database Design and Implementation**

Halima Ware :H00178418

Emaan Siddiqui :H00202694

Muhammad Maaz :H00177899

Cassandra Fernandez :H00200702

Abdallah Mohamed :H00176120

**Abstract**

Involves the design and implementation of a relational database (using MySQL)

# R1 Scenario and Conceptual model (10 marks)

Description of extended scenario:

Our courier company arranges for the collection and delivery of packages for clients where each client can have multiple orders and multiple delivery locations ie home, workplace etc. The company delivers to 3 main areas: Sharjah, Ajman, and Dubai. Each employee is assigned to a vehicle by using the vehicle plate number. The employees are categorized into 3 departments: manager, technician and driver.

In our database we created the following tables:



This ER diagram represents the conceptual model of our scenario. It has one-to many and many-to-many relationships:

* In the case of 1-M, for example one client can have multiple orders and multiple orders can have one client.
* In the case of M-M, for example, many orders can have many rates and in that another relationship is set where 1 order has many order rates depending on the location it is being delivered to. So the same order would be charged less if it was going to Sharjah instead of Abu Dhabi.

Our ER diagram also shows the recursive relationship we have in the employee’s table. In this case, a manager, which is also and employee, supervises another employee. Furthermore, we have a variety of participation constraint for example not all employees are drivers.

We also made use of derived attributes for example the rate is calculated by the location, composite attributes, composite keys, repeating attributes, use of generalization and specialization.

# R2 Translation into Relational Schema (10 marks)

In the next stage, we had to create a relational schema from our conceptual ER diagram. It includes the foreign keys, and primary keys.

Below is our RM and how we developed it further with the data dictionary.

Employee (

empID: integer (5);

firstName: string (30);

lastName: string (30)

)

Department (

deptID: interger (5);

role: string (20)

)

Vehicles (

vechiclePlate: string (6)

)

Area (

areaCode: integer (5);

area: string (30)

)

Delivery (

d\_ID: integer (5);

firstName: string (30)

)

Orders (

orderID: integer (5);

)

OrderRate (

rateID: integer (5);

location: string (50);

rate: integer (5,3);

)

Client (

cid: integer (5);

firstName: string (30);

lastName: string (30);

)



Refer to data dictionary:

In the employee table, we added a foreign key, Dept.ID, from the Table department because

of the relationship. In one case, the manager manages a department and so we need a

foreign key in the employee table to illustrate

this relationship

The vehicle table has 3 foreign keys:

Employee id – because we need to know what employee is driving what car

Department id –

Order id-

Area has 4 foreign keys:

Client has one foreign key and that’s the order id so we can know what order belongs to what client

Orders table has client id so we know what order belongs to what client, it also has the delivery id where we can tell where to deliver the package and lastly it has the rate id so as to price accordingly how much the transaction will be depending on where they live.

# R3 Implementation of the Schema in MySQL (5 marks)

Here we had to create the tables either using phpMyAdmin or through the SQL command line, using the innoDB storage engine. Below are the tables we created.

CREATE TABLE Department(

deptID INT(5) NOT NULL,

role VARCHAR(20) NOT NULL,

PRIMARY KEY(deptID)

) ENGINE = INNODB;

CREATE TABLE orders(

orderID INT(5) NOT NULL,

ccID INT(5) NOT NULL,

ddID INT(5) NOT NULL,

rrID INT(5) NOT NULL,

PRIMARY KEY(orderID),

FOREIGN KEY(ccID) REFERENCES CLIENT(clientID),

FOREIGN KEY(ddID) REFERENCES Delivery(devID),

FOREIGN KEY(rrID) REFERENCES OrderRate(rateID)

) ENGINE=INNODB;

CREATE TABLE CLIENT(

clientID INT(5) NOT NULL,

firstName VARCHAR(30) NOT NULL,

lastName VARCHAR(30) NOT NULL,

odrID INT(5) NOT NULL,

PRIMARY KEY(clientID),

FOREIGN KEY(odrID) REFERENCES Orders(orderID)

) ENGINE = INNODB;

CREATE TABLE OrderRate(

rateID INT(5) NOT NULL,

rate CHAR(10) NOT NULL,

location VARCHAR(150) NOT NULL,

PRIMARY KEY(rateID)

) ENGINE = INNODB;

CREATE TABLE Delivery(

devID INT(5) NOT NULL,

firstName VARCHAR(30) NOT NULL,

ordrID INT(5) NOT NULL,

emID INT(5) NOT NULL,

rID INT(5) NOT NULL,

vID VARCHAR(6) NOT NULL,

PRIMARY KEY(devID),

FOREIGN KEY(ordrID) REFERENCES Orders(orderID),

FOREIGN KEY(emID) REFERENCES Employee(empID),

FOREIGN KEY(rID) REFERENCES OrderRate(rateID),

FOREIGN KEY(vID) REFERENCES Vehicles(vehiclePlate)

) ENGINE = INNODB;

CREATE TABLE Employee(

empID INT(5) NOT NULL,

firstName VARCHAR(20) NOT NULL,

lastName VARCHAR(20) NOT NULL,

dID INT(5) NOT NULL,

PRIMARY KEY(empID),

FOREIGN KEY(dID) REFERENCES Department(deptID)

) ENGINE = INNODB;

CREATE TABLE Vehicles(

vehiclePlate VARCHAR(6) NOT NULL,

eID INT(5) NOT NULL,

depID INT(5) NOT NULL,

ordID INT(5) NOT NULL,

PRIMARY KEY(vehiclePlate),

FOREIGN KEY(eID) REFERENCES Employee(empID),

FOREIGN KEY(depID) REFERENCES Department(deptID),

FOREIGN KEY(ordID) REFERENCES Orders(orderID)

) ENGINE = INNODB;

CREATE TABLE AREA(

areaCode CHAR(3) NOT NULL,

AREA VARCHAR(50) NOT NULL,

orID INT(5) NOT NULL,

vPlate VARCHAR(6) yNOT NULL,

dlID INT(5) NOT NULL,

cID INT(5) NOT NULL,

PRIMARY KEY(areaCode),

FOREIGN KEY(orID) REFERENCES Orders(orderID),

FOREIGN KEY(vPLate) REFERENCES Vehicles(vehiclePlate),

FOREIGN KEY(dlID) REFERENCES Delivery(devID),

FOREIGN KEY(cID) REFERENCES CLIENT(clientID)

) ENGINE=INNODB;

CREATE TABLE Rate(

rrateID INT(5) NOT NULL,

aCode CHAR(3) NOT NULL,

orrderID INT(5) NOT NULL,

FOREIGN KEY(rrateID) REFERENCES OrderRate(rateID),

FOREIGN KEY(aCode) REFERENCES AREA(areaCode),

FOREIGN KEY(orrderID) REFERENCES Orders(orderID)

) ENGINE = INNODB;

# R4 Loading Data

Next we had to load the data using insert statements,

INSERT INTO Department (deptID, role) VALUES

('' , 'Manager'),

('' , 'Technician'),

('' , 'Driver');

INSERT INTO Employee (empID, firstName, lastName, dID) VALUES

('' , 'Talal' ,'Shiekh' ,''),

('' , 'Mohammed' ,'Hamdan' ,''),

('' , 'Hind' ,'Zantout' ,''),

('' , 'Hani' ,'Ahmad' ,'');

INSERT INTO Vehicles (vehiclePlate, eID, depID, ordID) VALUES

('L56109' , '' ,'' ,''),

('A00001' , '' ,'' ,''),

('EF4567' , '' ,'' ,'');

INSERT INTO Delivery (devID, firstName, ordrID, emID, rID, vID) VALUES

('', 'Abdullah' , '', '', '', ''),

('', 'Maaz' , '', '', '', ''),

('', 'Halima' , '', '', '', ''),

('', 'Taan-E' , '', '', '', ''),

('', 'Cass' , '', '', '', '');

INSERT INTO Area (areaCode, area, orID, vPlate, dlID, cID) VALUES

('SHJ', 'Sharjah, Nassriyah' , '', '', '', ''),

('DXB', 'Dubai, Barsha' , '', '', '', ''),

('ABD', 'Abudhabi, Mussfah' , '', '', '', '');

INSERT INTO Client (clientID, firstName, lastName, odrID) VALUES

('' , 'Abdullah' ,'Abdelwahab' ,''),

('' , 'Monster' ,'Sandwich' ,''),

('' , 'Emaan' ,'Nananana' ,''),

('' , 'Maaz' ,'Barz' ,''),

('' , 'Halima' ,'Lima' ,''),

('' , 'Aliyu' ,'Hafiz' ,''),

('' , 'Bello' ,'Dangiwa' ,'');

INSERT INTO Orders (orderID, ccID, ddID, rrID) VALUES

('' , '' ,'' ,''),

('' , '' ,'' ,''),

('' , '' ,'' ,'');

INSERT INTO OrderRate (rateID, rate, location) VALUES

('' , '20 AED' ,'Sharjah' ),

('' , '30 AED' ,'Dubai' ),

('' , '40 AED' ,'Abudhabi' );

INSERT INTO Rate (rrateID, aCode, orrderID) VALUES

('' , '' ,'' ),

('' , '' ,'' ),

('' , '' ,'' );

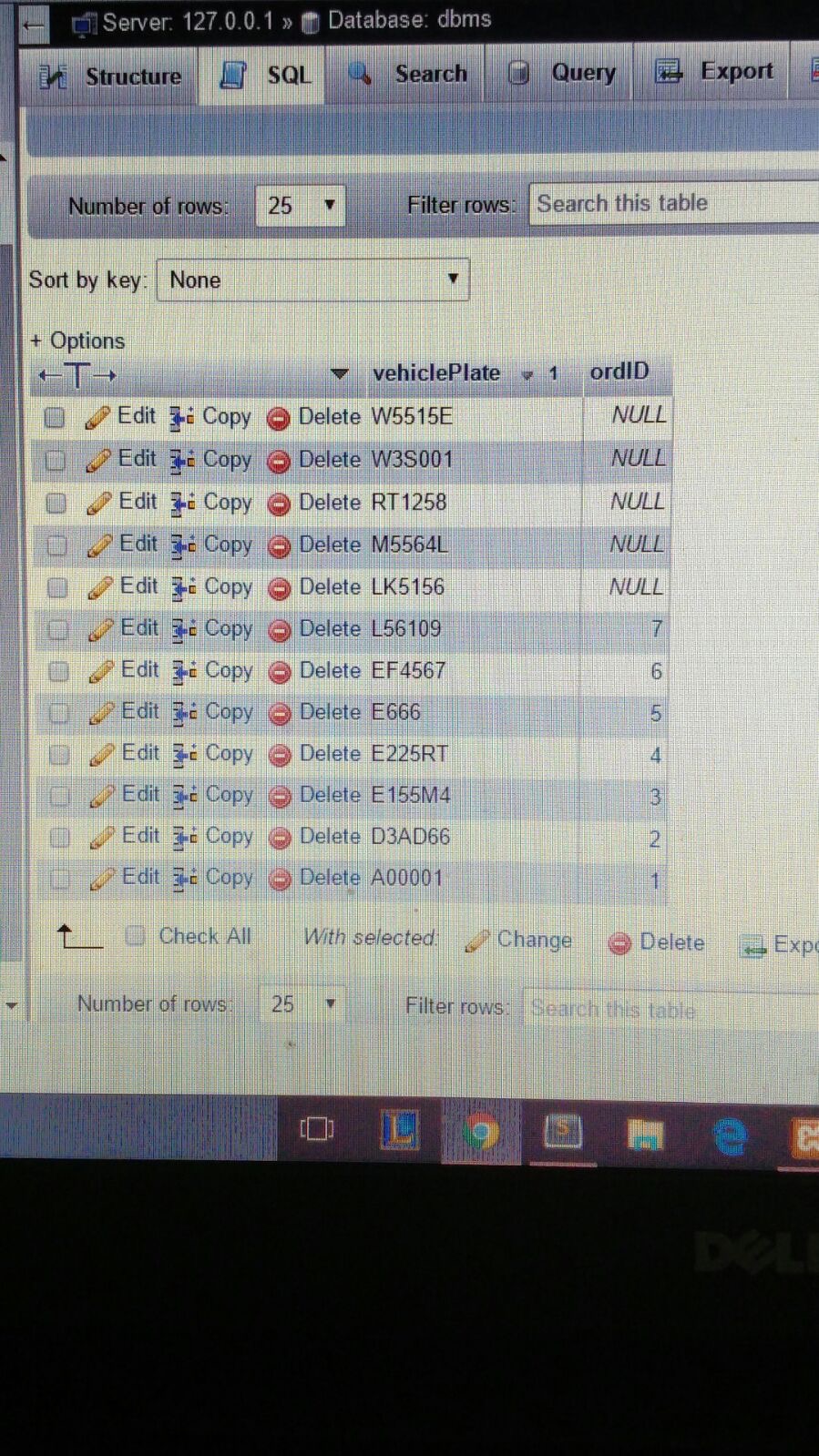
An example would be the vehicles table;

INSERT INTO Vehicles (vehiclePlate, eID, depID, ordID) VALUES

('L56109' , '' ,'' ,''),

('A00001' , '' ,'' ,''),

('EF4567' , '' ,'' ,'');



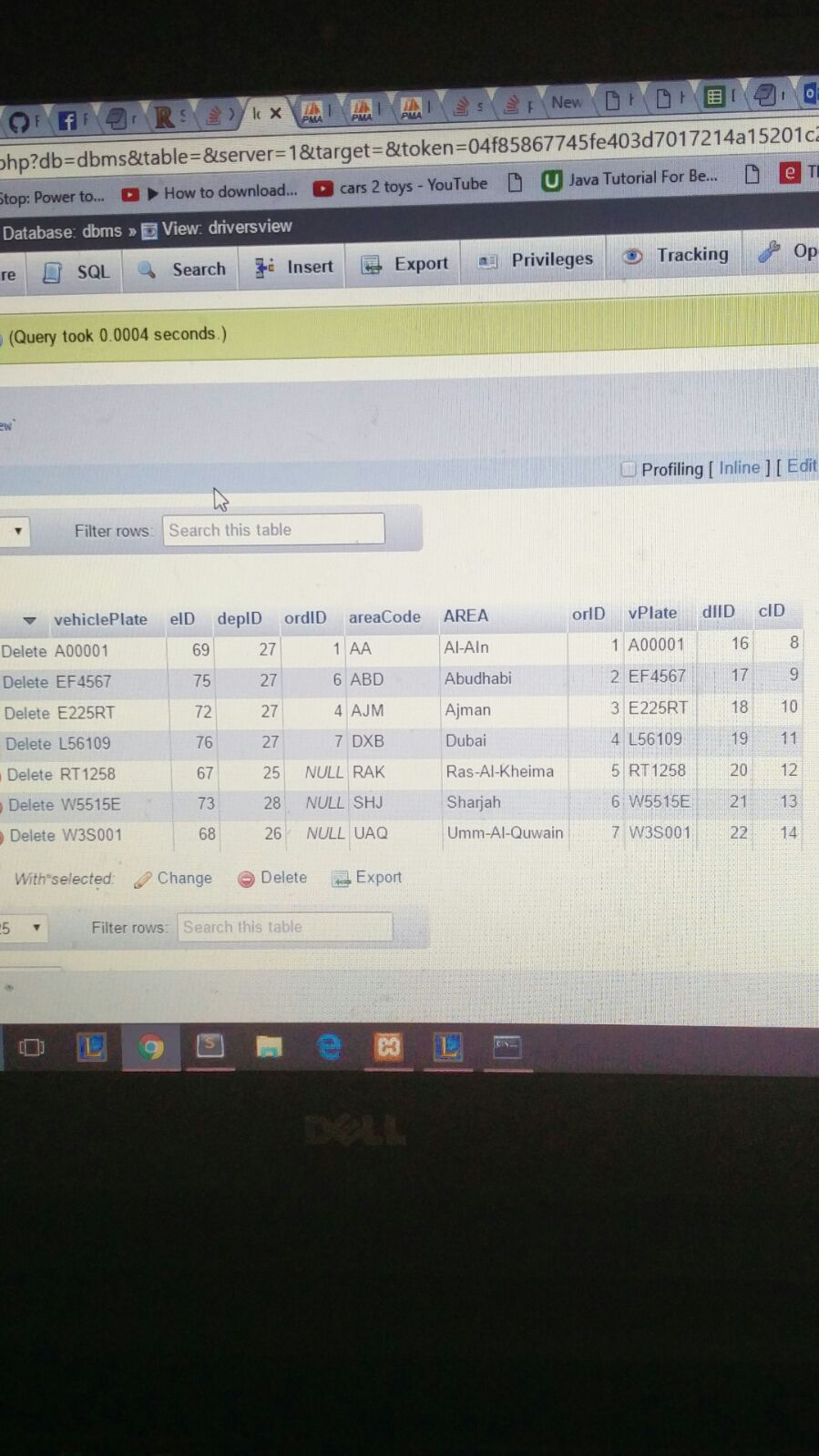
SELECT vehiclePlate,COUNT(ordID)

FROM vehicle order by vehiclePlate desc;

# R5 Roles, permissions and views

# Drivers View

He can see the order ID and the location and area code.



# Managers View

The manager can see the clients and their orders

# R6 Queries over the Database

Below are **some** of the queries implemented in our table.

@COMBINES VEHICLES AND AREA

CREATE VIEW DriversView

AS SELECT \*

FROM Vehicles V, AREA A

WHERE V.vehiclePlate=A.vPlate;

CREATE VIEW ManagersView

AS SELECT \*

FROM client C,orders O

WHERE C.odrID=O.orderID;

@finds out how man vehicles go to each area

SELECT AREA,count(vehiclePlate)

FROM DBvehicleandarea

ORDER BY count(vehiclePlate);

@amount of employees in each Department

SELECT Department.role, count(Employee.empID)

FROM Department

INNER JOIN Employee

ON Department.deptID=Employee.dID;

# R7 Indexes

The indexes were created on the tables vehicle, orders and employees as given below:

