CSC 472 Introduction to Database Systems
Project Database Design and Implementation

- *Akshay Rajshekar Shiraguppi(arajs2@uis.edu)*

1. **Project description and purpose: Choose an application that you think is useful that has relevance in database system design. Provide a very brief description of your project. The database must have unnormalized relations so that you can come out with 1NF and 2NF. The database must have minimum of three entity sets in the final or normalized form. Each entity set must have at least 2 attributes.**

⇨ The Project is about the Registration of the Vehicles. In any country the registration of vehicles is mandatory. Registration plays an important role to identify vehicles and their owners. Vehicle registration is a compulsory procedure incurred all over the world. After the registration of the vehicle, the owner of the vehicle gets the Number Plate which is unique for each and every vehicle. The project is about how the registration takes place. The vehicle registration process where a customer selects a vehicle which he wants to buy then he will go to registration office for registering his vehicle. There is a certain amount to be paid at the time of registration as registration fee. The registration officer will then take the record of the Engine No and Chassis No(VIN) and allots a unique number to the customer after certain period of time.

The Required Fields for this project are:

Customer ID, customer name, customer phone number, customer address, vehicle make, vehicle model, vehicle color, vehicle engine no, vehicle chassis no, date of registration, amount, registration number, registration area, registration valid from, registration valid till.

Here in this database we are creating four tables.

- customer (Customer_ID, Name, Street, CITY, Phone_No)
- vehicles (Engine_No, Chassis_No, Make, Model, Color)
- RTO (S_No, Date_OF_REG, Amount)
- Number_Plate (Reg_NO, Owner_Name, Valid_From, Valid_Till, Reg_Area)

In this data base customer buys a vehicle then he registers the vehicle with the RTO later the RTO assigns the Number_Plate then the owner(customer) receives the Number_Plate.

2. **Unnormalized Form: Show the unnormalized relations. Specify the name of each relation followed by its attributes separated by commas and placed in parenthesis.**

⇨ customer (Customer_ID, Name, Address,)

RTO (Engine_No, Chassis_No, Make, Model, Color, S_No, Date_OF_REG, Amount, Owner_Name, Valid_From, Valid_Till, Reg_Area)

Here in the customer table Address is in the unnormalized form. The RTO table is also can be divided Into different tables like vehicle, RTO and Licence_Plate so that the we can minimize data redundancy.

3. **Normalized Form: Perform normalization or decomposition. Eliminate any partial functional dependencies, transitive functional dependencies, repetitive repetition, and anomalies. Show or explain how you perform the normalization or decomposition process.**

⇨ To perform normalization I have changed the customer address into two attributes i.e. street and city so that we can get it into normal form. I also decomposed the RTO table into Vehicles , RTO and Licence_Plate so that data will be in the normal form and we can eliminate functional and transitive dependencies.

```
SQL> create table customer
  2  (Customer_ID varchar2(15),
  3  Name varchar2(15) not null,
  4  Street varchar2(15) not null,
  5  City varchar2(15) not null,
  6  Phone_No varchar2(15) not null,
  7  primary key(Customer_ID));
```

```
SQL> create table vehicle
  2  (Engine_No varchar2(15),
  3  Chassis_No varchar2(15),
  4  Make varchar2(15) not null,
  5  Model varchar2(15) not null,
  6  Color varchar2(15) not null,
  7  primary key(Engine_No));
```

```
SQL> create table RTO
  2  (S_No varchar2(15),
  3  Date_OF_REG date,
  4  Amount number not null,
  5  primary key(S_No));
```

```
SQL> create table Number_Plate
```

```
  2  (Reg_No varchar2(15),
  3  Owner_Name varchar2(15) not null,
  4  Valid_From varchar2(15) not null,
  5  Valid_Till varchar2(15) not null,
  6  Reg_Area varchar2(15) not null,
  7  primary key(Reg_No));

SQL> create table Buys
  2  (Customer_ID varchar2(15),
  3  Engine_No varchar2(15),
  4  Chassis_No varchar2(15),
  5  primary key(Customer_ID, Engine_No),
  6  foreign key(Engine_No) references vehicle(Engine_No)
  7  foreign key(Customer_ID) references customer(Customer_ID));

SQL> create table Registration
  2  (Engine_No varchar2(15),
  3  Chassis_No varchar2(15),
  4  S_No varchar2(15),
  5  primary key(Engine_No,  S_No),
  6  foreign key(Engine_No) references vehicle(Engine_No),
  7  foreign key(S_No) references RTO(S_No));

SQL> create table Assigns
  2  (S_No varchar2(15),
  3  Reg_No varchar2(15),
  4  primary key(S_No, Reg_No),
  5  foreign key(S_No) references RTO(S_No),
  6  foreign key(Reg_No) references Number_Plate(Reg_No));
```

```
SQL> create table receive
  2  (Reg_No varchar2(15),
  3  Customer_ID varchar2(15),
  4  primary key(Reg_No, Customer_ID),
  5  foreign key(Reg_No) references Number_Plate(Reg_No),
  6     foreign key(Customer_ID) references customer(Customer_ID));
```

4. Show all the schemas of the resulting or normalized relations in your database. Specify the name followed by attributes separated by commas and placed in parenthesis for each relation. Underline the primary key(s).

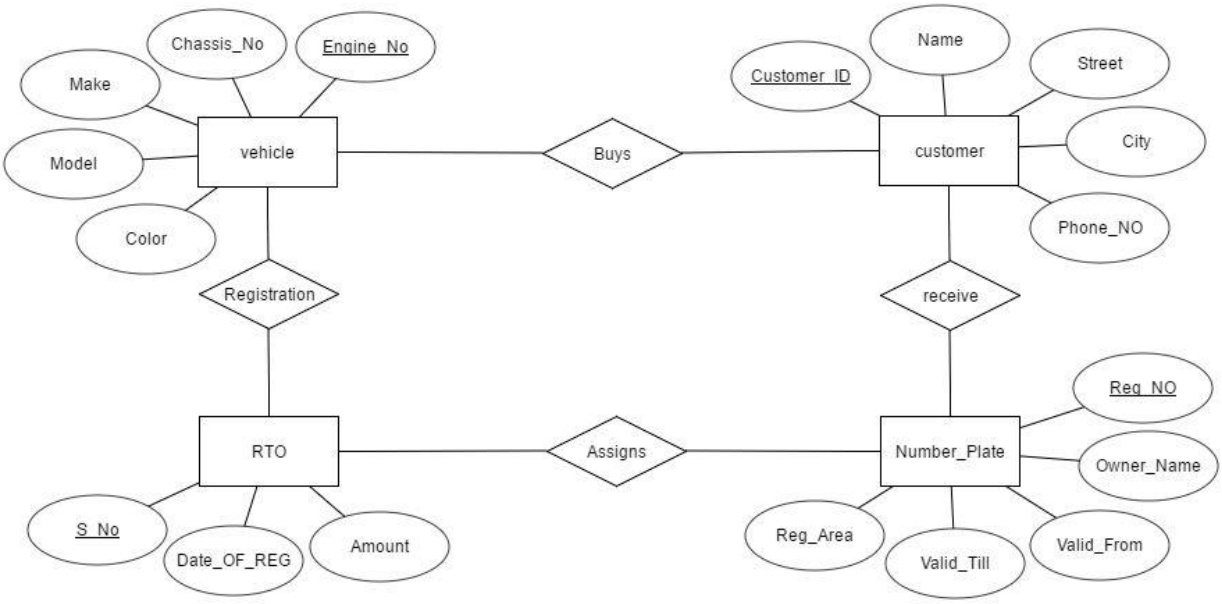
It will be simply in the form of:
relation_name(attribute1, attribute2, ….)

⇨

- customer (Customer_ID, Name, Street, CITY, Phone_No)
- vehicles (Engine_No, Chassis_No, Make, Model, Color)
- RTO (S_No, Date_OF_REG, Amount)
- Number_Plate (Reg_NO, Owner_Name, Valid_From, Valid_Till, Reg_Area)

5. Basic ER diagram. Produce a basic ER diagram of your database which reflects entity sets, attributes, and relationships among them. Underline the primary key on the ER diagram.

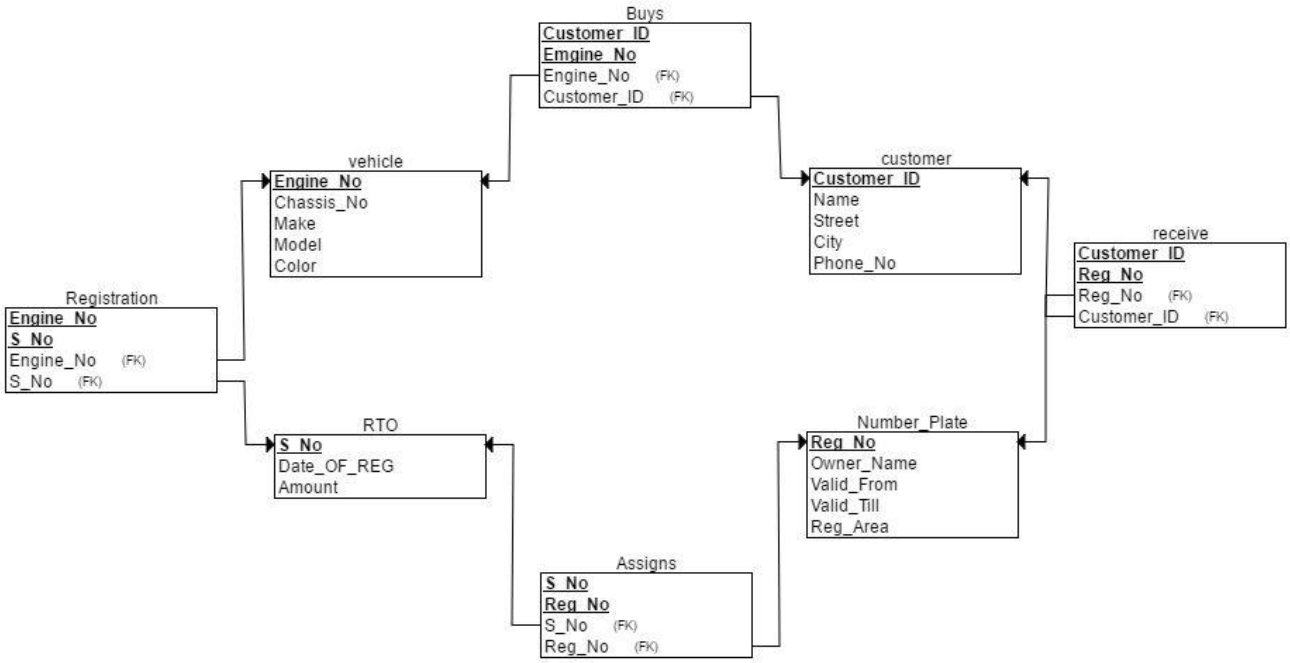
⇨



5. **Relational Database Schema. Convert the ER diagram into a relational database schema. Underline primary keys.**

⇨

.



6. **Algebraic Statements. Formulate any two algebraic queries that are relevant to the application you have chosen.**

⇨ Algebraic Statements:

SQL> select * from RTO where amount > 1500;

σ amount **>** 1500(RTO)

SQL> select * from vehicle where color = 'Red';

σ color **=** ’Red’(vehicle)

SQL> select Owner_Name, Reg_No from Number_Plate;

π Owner_Name, Reg_No (Number_Plate)

SQL> select S_No, Date_OF_REG from RTO where amount > 1500;

π S_No, Date_OF_REG (***σ*** amount **>**100 (RTO))

7. SQL Statements. Formulate any two queries in SQL statements that are relevant to the application you have chosen .

⇨

SQL> select * from vehicle where color = 'White';

```
ENGINE_NO          CHASSIS_NO         MAKE                 MODEL
COLOR
---------------- ---------------- ---------------- ------------
--- ---------------
KLAJD1254          GFCFCY5318         Sokda                Octavia
White
```

SQL> select Owner_Name, S_No from Number_Plate, Assigns where(Assigns.Reg_No = Number_Plate.Reg_No);

```
OWNER_NAME       S_NO
---------------- ---------------
Akshay           1236
Raj              1234
Aravind          1235
```

SQL> select Date_OF_REG, Amount, Engine_No from RTO, Registration where(RTO.S_No = Registration.S_No);

```
DATE_OF_R     AMOUNT ENGINE_NO
--------- ---------- ---------------
09-APR-17       1500 KLAJD1254
15-JAN-07       1600 KLAJD1545
15-DEC-13       1700 KLAJD9874
```

SQL> select * from RTO where amount > 1500;

```
S_NO             DATE_OF_R     AMOUNT
---------------- --------- ----------
1235             15-JAN-07       1600
1236             15-DEC-13       1700
```

9. XPath Expression. Convert two algebraic queries into XPath expressions.

⇨ SQL> select * from RTO where amount > 1500;

σ amount **>** 1500(RTO)
/vehicleregistration/RTO[Amount > 1500]

SQL> select * from vehicle where color = 'Red';

σ color = ’Red’(vehicle)
/Vehicleregistration/vehicle[color = ‘Red’]

10. Create Relations. Create two relations using SQL statements including constraints, i.e. show the SQL commands how to create the relations.

⇨

SQL> select Date_OF_REG, Amount, Engine_No from RTO, Registration where (RTO.S_No = Registration.S_No);

```
DATE_OF_R     AMOUNT ENGINE_NO
--------- ---------- ---------------
09-APR-17       1500 KLAJD1254
15-JAN-07       1600 KLAJD1545
15-DEC-13       1700 KLAJD9874
```

SQL> select Owner_Name, S_No from Number_Plate, Assigns where (Assigns.Reg_No = Number_Plate.Reg_No);

```
OWNER_NAME      S_NO
--------------- ---------------
Akshay          1236
Raj             1234
Aravind         1235
```

SQL> select Name, Phone_No, Engine_No from customer, Buys where (Buys.Customer_Id = Customer.Customer_ID);

```
NAME            PHONE_NO        ENGINE_NO
--------------- --------------- ---------------
Akshay           9575757575      KLAJD1254
Raj               9757585855      KLAJD9874
Aravind          9949945124      KLAJD1545
```

11. Populate Relations. Insert three records in at least two relations, i.e. show SQL commands on how to populate the relations.

⇨ **CUSTOMER TABLE:** The following commands are used to insert data into the customer table:

SQL> insert into customer values ('0123', 'Akshay', 'Whitefield', 'Bengaluru’, '9575757575');

SQL> insert into customer values ('0124', ‘Raj', 'Noida', 'Delhi', '9757585855');

SQL> insert into customer values ('0125', 'Aravind', 'Marina', 'Chennai', '8545452577');

VEHICLE TABLE: The following commands are used to insert data into vehicle table:

SQL> insert into vehicle values ('KLAJD1254', 'GFCFCY5318', 'Sokda', 'Octavia', 'White');

SQL> insert into vehicle values ('KLAJD1545', 'GFCFCY3578', 'Honda', 'Accord', 'Red');

SQL> insert into vehicle values ('KLAJD9874', 'GFCFCY1259', 'Chevrolet', 'Cruze', 'Black');

RTO TABLE: The following commands are used to insert data into RTO table:

SQL> insert into RTO values('1234','09-apr-2017','1500');

SQL> insert into RTO values('1235','15-jan-2007','1600');

SQL> insert into RTO values('1236','15-dec-2013','1700');

NUMBER_PLATE: The following commands are used to insert data into NUMBER_PLATE table:

SQL> insert into Number_Plate values ('TS02BD0017', 'Akshay', '2012','2030','Bengaluru');

SQL> insert into Number_Plate values ('AP37AG2678', 'Raj', '2007','2028','Delhi');

SQL> insert into Number_Plate values ('TS07CH7977', 'Aravind', '2013','2031','Chennai');

12. Briefly describe what were the least and most challenging parts of the above and explain why?.

⇨ The least challenging part is to insert values into the tables. The most challenging part is to decide on which database I have to design after deciding the model and selection of attributes is also one of the difficult task.