**PROJECT REPORT**

**On**

**TAXI SERVICE MANAGEMENT SYSTEM**

**For the partial fulfilment of Degree of B. Tech in Computer Science**

**Course: UCS310 Database and Management System**

**Martin Kaushal (Roll No. 102017135)**

**Mudrika Jain (Roll No. 102017143)**

**Nandini Setia (Roll No. 102017121)**

**Sudhit Soni (Roll No. 102017137)**

**(Students of 2nd Year B.Tech/B.E. (Computer Science)**

**Under the Guidance of:**

**Mr. Hemant Kumar Gianey**

Thapar Institute of Engineering & Technology, Patiala



**Submitted to**

**THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY**

**(Deemed to be University)**

**PATIALA – 147004**

**PUNJAB – INDIA**

**2022**

Y

INDEX

[REQUIREMENTS 3](#_Toc103544982)

[MODELLING OF REQUIREMENTS AS ER-DIAGRAM 4](#_Toc103544983)

[ASSUMPTIONS: 4](#_Toc103544984)

[MAPPING OF ER-DIAGRAM IN RELATIONAL SCHEMA 5](#_Toc103544985)

ER TO TABLES…………………………………………………………………………………………………………………………………….6

[SQL STATEMENS FOR TABLE CREATION 8](#_Toc103544986)

[SQL STATEMENS FOR INSERT COMMANDS 10](#_Toc103544988)

OUTPUT SCREENSHOTS…………………………………………………………………………………………………………………….12

[NORMALIZATION OF RELATIONAL SCHEMA 14](#_Toc103544995)

# REQUIREMENTS

#### The Taxi Service Database involves around three main entities Taxi, User and Trip.

#### Taxi can be booked for a specific location with a specific address by a User. User has a unique User\_id, a Contact\_no and an Email.

#### A Taxi Service has a number of taxis for service. Each taxi is described by Taxi\_id, Registration\_no, Model, Manufactured year and Status.

#### Taxi has a parameter Taxi\_type. It can be ‘Economy’, ‘Standard’, ’SUV’, ‘Premium’ and ‘Minivan’. Taxi\_type defines the price per hour.

#### A User can reserve a taxi for a number of hours/days. He can use any valid promotional code.

#### A user is uniquely identified by his/her User\_id. User information consists of his name as first name, last name, address, age and contact number.

#### When a user books a taxi and starts the trip by the driver the start time automatically updated by the system.

#### When the trip ends, the end trip time also automatically updated in the database by the system.

#### A unique bill is generated with a Bill\_no after a trip ends which has the information of user, driver, amount, date.

#### The total amount and net amount are calculated based on start time, end time, taxi price per hour and promotional code if any.

#### A taxi is categorized as Individual Owner and Taxi Service Company. Every taxi has a owner and he/she can give his/her car for the taxi service. Every owner has SSN and name. For the taxi service company information like tcs\_id and tsc\_name will also be there.

#### A registered user will be provided with a login id and password. A customer can save his credit/debit card details for future payment.

#### Partial payment can also be made at the time of booking and the balance must be paid by the user at the end of the trip.

#### If user is a customer, he/she can pay through saved debit/credit card details

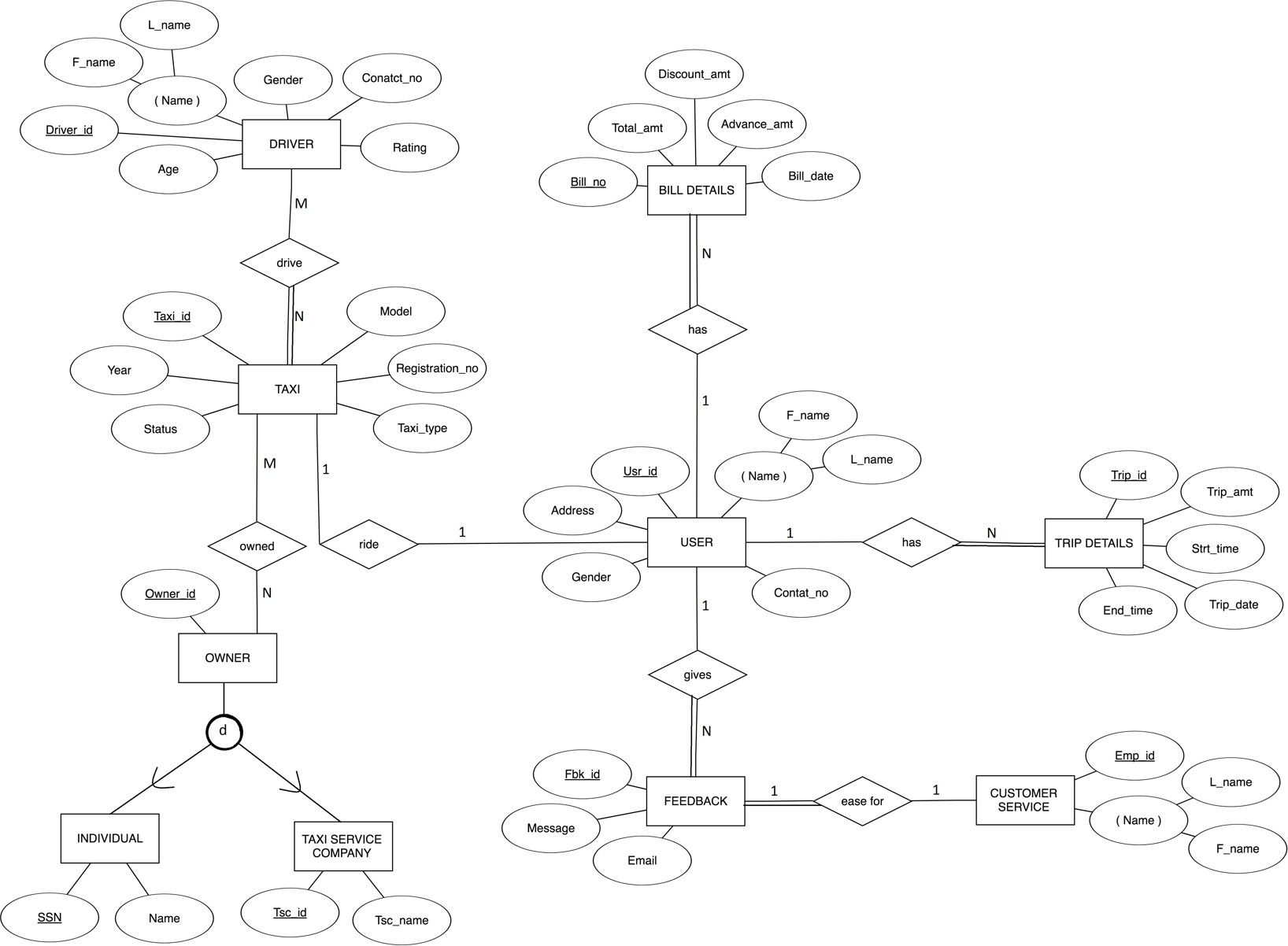
#### A taxi can be drive by a driver. Driver has uniquely identified by the Driver\_id. Other information consists of name, gender, contact\_no, rating and age.

#### After the trip over a unique trip\_id is generated for that particular trip. Along with all the necessary trip\_details such as amount, date etc.

#### Users can also the give the feedback/rating for the trip they traveled into it. The feedback can be a message or rating out five for the driver who is giving trip to that user.

#### Feedback can be taking by the customer service center representative. They have the information like emp id, name and email.

# MODELLING OF REQUIREMENTS AS ER-DIAGRAM



## ASSUMPTIONS:

#### Many drivers can drive many taxis (M:N)

#### Many owners can give many taxis at a time (M:N)

#### One customer service representative can take one feedback at a time (1:1)

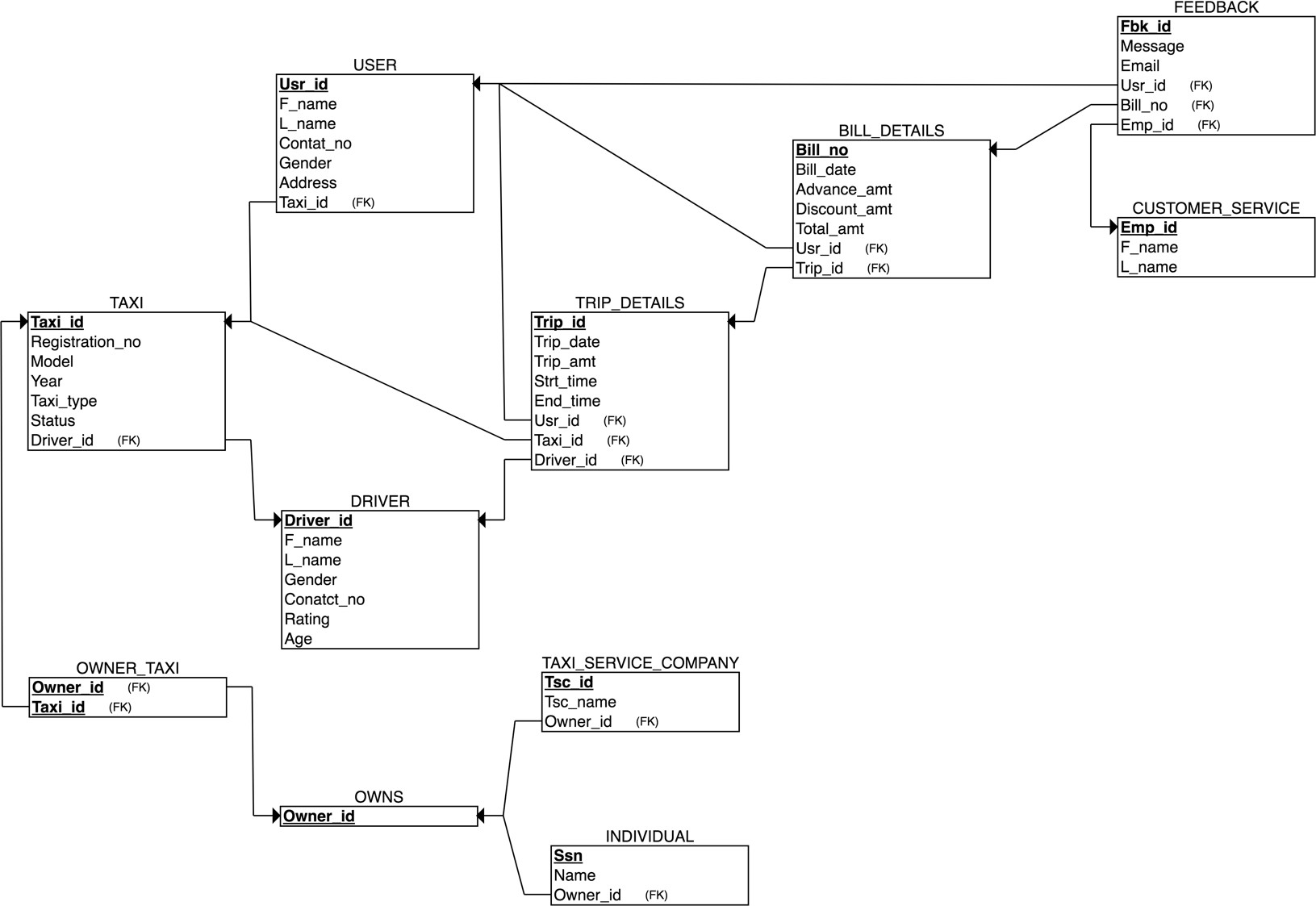
#### Single user can have multiple trips details (1:N)

#### Single user can have multiple bills details (1:N)

#### Single user can give many feedbacks (1:N)

#### Single user can ride in one taxi at a time (1:1)

# MAPPING OF ER-DIAGRAM IN RELATIONAL SCHEMA



ER TO TABLE :-

### TAXI

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Taxi\_id | Registration\_no | Taxi\_Model | Taxi\_Year | Taxi\_typ e | Status | Driver\_id |

#### Primary Key: Taxi\_id

#### Foreign Keys: Driver\_id

### USER\_TBL

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Usr\_id | F\_name | L\_name | Contat\_no | Gender | Address | Taxi\_id |

#### Primary Key: Usr\_id

#### Foreign Keys: Taxi\_id

### DRIVER

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Driver\_id | F\_name | L\_name | Gender | Conatct\_no | Rating | Age |

#### Primary Key: Driver\_id

#### Foreign Keys: NA

### TRIP\_DETAILS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trip\_id | Trip\_date | Trip\_amt | Driver\_id | Usr\_id |

|  |  |  |
| --- | --- | --- |
| Taxi\_id | Strt\_time | End\_time |

#### Primary Key: Trip\_id

#### Foreign Keys: Taxi\_id, Usr\_id, Driver\_id

### BILL\_DETAILS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Bill\_no | Bill\_date | Advance\_amt | Discount\_amt | Total\_amt | Usr\_id | Trip\_id |

#### Primary Key: Bill\_no

#### Foreign Keys: Usr\_id, Trip\_id

### CUSTOMER\_SERVICE

|  |  |  |
| --- | --- | --- |
| Emp\_id | F\_name | L\_name |

#### Primary Key: Emp\_id

#### Foreign Keys: NA

### FEEDBACK

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fbk\_id | Message | Email | Emp\_id | Usr\_id | Trip\_id |

#### Primary Key: Fbk\_id

#### Foreign Keys: Usr\_id, Emp\_id, Trip\_id

### OWNER\_TAXI

Taxi\_id

Owner\_id

#### Primary Key: Owner\_id, Taxi\_id

#### Foreign Keys: Owner\_id, Taxi\_id

### OWNS

No\_Cars

Owner\_id

#### Primary Key: Owner\_id

#### Foreign Keys: NA

### INDIVIDUAL

|  |  |  |
| --- | --- | --- |
| Ssn | Name | Owner\_id |

#### Primary Key: Ssn

#### Foreign Keys: Owner\_id

### TAXI\_SERVICE\_COMPANY

|  |  |  |
| --- | --- | --- |
| Tsc\_id | Tsc\_name | Owner\_id |

#### Primary Key: Tsc\_id

#### Foreign Keys: Owner\_id

# SQL STATEMENS FOR TABLE CREATION

-- Table Creation

CREATE TABLE\_TAXI (

Taxi\_id integer NOT NULL, Registration\_no VARCHAR(20), Taxi\_Model VARCHAR(20), Taxi\_Year DATE,

Taxi\_type VARCHAR(20), Status VARCHAR(20),

Driver\_id integer, PRIMARY KEY (Taxi\_id), UNIQUE (Registration\_no), foreign key(Driver\_id) references DRIVER(Driver\_id) ON DELETE CASCADE

);

CREATE TABLE USER\_TBL (

Usr\_id integer NOT NULL, F\_name VARCHAR(20), L\_name VARCHAR(20),

Contat\_no integer, Gender VARCHAR(10),

Address VARCHAR(50), Taxi\_id integer, PRIMARY KEY (Usr\_id),

FOREIGN KEY (Taxi\_id) REFERENCES TAXI(Taxi\_id) ON DELETE CASCADE

);

CREATE TABLE DRIVER (

Driver\_id integer NOT NULL, F\_name VARCHAR(10),

L\_name VARCHAR(20), Gender VARCHAR(10),

Conatct\_no VARCHAR(20), Rating integer,

Age integer,

PRIMARY KEY (Driver\_id)

);

CREATE TABLE TRIP\_DETAILS (

Trip\_id integer NOT NULL, Trip\_date DATE,

Trip\_amt decimal(10,2), Driver\_id integer, Usr\_id integer, Taxi\_id integer, Strt\_time TIMESTAMP, End\_time TIMESTAMP, PRIMARY KEY (Trip\_id),

FOREIGN KEY (Driver\_id) REFERENCES DRIVER(Driver\_id) ON DELETE CASCADE,

FOREIGN KEY (Usr\_id) REFERENCES USER\_TBL(Usr\_id) ON DELETE CASCADE,

FOREIGN KEY (Taxi\_id) REFERENCES TAXI(Taxi\_id) ON DELETE CASCADE

);

CREATE TABLE BILL\_DETAILS (

Bill\_no integer NOT NULL, Bill\_date DATE, Advance\_amt decimal(10,2), Discount\_amt decimal(10,2), Total\_amt decimal(10,2), Usr\_id integer,

Trip\_id integer, PRIMARY KEY (Bill\_no),

FOREIGN KEY (Trip\_id) REFERENCES TRIP\_DETAILS(Trip\_id) ON DELETE CASCADE,

FOREIGN KEY (Usr\_id) REFERENCES USER\_TBL(Usr\_id) ON DELETE CASCADE

);

CREATE TABLE CUSTOMER\_SERVICE (

Emp\_id integer NOT NULL, F\_name VARCHAR(20), L\_name VARCHAR(20), PRIMARY KEY (Emp\_id)

);

CREATE TABLE FEEDBACK (

Fbk\_id integer NOT NULL, Message VARCHAR(140), Email VARCHAR(50),

Emp\_id integer, Usr\_id integer, Trip\_id integer, PRIMARY KEY (Fbk\_id),

FOREIGN KEY (Emp\_id) REFERENCES CUSTOMER\_SERVICE(Emp\_id) ON DELETE CASCADE,

FOREIGN KEY (Trip\_id) REFERENCES TRIP\_DETAILS(Trip\_id) ON DELETE CASCADE,

FOREIGN KEY (Usr\_id) REFERENCES USER\_TBL(Usr\_id) ON DELETE CASCADE

);

CREATE TABLE OWNS (

Owner\_id integer NOT NULL, No\_Cars integer,

PRIMARY KEY (Owner\_id)

);

CREATE TABLE OWNER\_TAXI (

Owner\_id integer NOT NULL, Taxi\_id integer,

PRIMARY KEY (Owner\_id, Taxi\_id),

FOREIGN KEY (Taxi\_id) REFERENCES TAXI(Taxi\_id) ON DELETE CASCADE,

FOREIGN KEY (Owner\_id) REFERENCES OWNS(Owner\_id) ON DELETE CASCADE

);

CREATE TABLE INDIVIDUAL (

Ssn integer NOT NULL, Name VARCHAR(20),

Owner\_id integer, PRIMARY KEY (Ssn),

FOREIGN KEY (Owner\_id) REFERENCES OWNS(Owner\_id) ON DELETE CASCADE

);

CREATE TABLE TAXI\_SERVICE\_COMPANY (

Tsc\_id integer NOT NULL, Tsc\_name VARCHAR(20), Owner\_id integer, PRIMARY KEY (Tsc\_id),

FOREIGN KEY (Owner\_id) REFERENCES OWNS(Owner\_id) ON DELETE CASCADE

);

# SQL STATEMENS FOR INSERT COMMANDS

-- Insert Commands

INSERT INTO TAXI VALUES(1,'KA-15R-3367','BENZE300','2017-01-01','Economy','Available',1);

INSERT INTO TAXI VALUES(2,'KA-15R-3368','BENZE300','2017-01-02','Standard','Not Available',2);

INSERT INTO TAXI VALUES(3,'KA-15R-3369','BENZE300','2017-01-03','SUV','Not Available',3);

INSERT INTO TAXI VALUES(4,'KA-15R-3370','BENZE300','2017-01-04','Premium','Available',4);

INSERT INTO TAXI VALUES(5,'KA-15R-3371','BENZE300','2017-01-05','SUV','Not Available',5);

INSERT INTO TAXI VALUES(6,'KA-15R-33675','BENZE300','2017-01-06','SUV','Available',6);

INSERT INTO TAXI VALUES(7,'KA-15R-3337','BENZE300','2017-01-07','Economy','Available',7);

INSERT INTO TAXI VALUES(8,'KA-15R-33627','BENZE300','2017-01-08','SUV','Available',8);

INSERT INTO TAXI VALUES(9,'KA-15R-33673','BENZE300','2017-01-09','Premium','Not Available',9);

INSERT INTO TAXI VALUES(10,'KA-15R-33671','BENZE300','2017-01-10','Standard','Available',10);

INSERT INTO DRIVER VALUES(2,'Nandini','Setia','Female','4693805870',5,25);

INSERT INTO DRIVER VALUES(3,'Martin','Kaushal','Male','4693805871',3,23);

INSERT INTO DRIVER VALUES(4,'Mudrika','Jain','Female','4693805872',2,20);

INSERT INTO DRIVER VALUES(5,'Sudhit','Soni','Male','4693805873',1,25);

INSERT INTO DRIVER VALUES(6,'Gaurav','Vats','Male','4693805874',2,26);

INSERT INTO DRIVER VALUES(7,'Sehroop','Kaur','Female','4693805875',2,27);

INSERT INTO DRIVER VALUES(8,'Sakshi','Kumar','Female','4693805876',5,21);

INSERT INTO DRIVER VALUES(9,'Anuj','Ahlawat','Male','4693805877',5,20);

INSERT INTO DRIVER VALUES(10,'Manan','Vig','Male','4693805878',3,24);

INSERT INTO DRIVER VALUES(11,'Shantam','Anand','Male','4693805879',2,25);

INSERT INTO USER\_TBL VALUES(2,'Ram','Aggarwal','123456','Male','Abohar','2');

INSERT INTO USER\_TBL VALUES(3,'Chintu','Goel','123457','Male','Sonipat','3');

INSERT INTO USER\_TBL VALUES(4,'Pintu','sharma','123458','Male','Patiala','4');

INSERT INTO USER\_TBL VALUES(5,'Pinki','Khurana','123459','Female','Nabha','5');

INSERT INTO USER\_TBL VALUES(6,'Khali','Jain','123456','Male','Malout','6');

INSERT INTO USER\_TBL VALUES(7,'Khyatii','Jain','223456','Male','Amritsar','7');

INSERT INTO USER\_TBL VALUES(8,'shyam','setia','123556','Male','Goa','8');

INSERT INTO USER\_TBL VALUES(9,'Mina','gupta','123356','Female','Shimla','9');

INSERT INTO USER\_TBL VALUES(10,'Tina','Mishra','133456','Female','Panipat','10');

INSERT INTO TRIP\_DETAILS VALUES(1,'2017-01-01',123,1,1,1,'2017-01-01 06:14:00','2017-01-01 08:14:00');

INSERT INTO TRIP\_DETAILS VALUES(2,'2017-01-02',124,2,12,2,'2017-01-02 06:16:00','2017-01-02 09:14:00');

INSERT INTO BILL\_DETAILS VALUES(1,'2017-01-01',1000.10,20.11,null,1,1);

INSERT INTO BILL\_DETAILS VALUES(2,'2017-01-02',10530.10,20.11,null,2,2);

INSERT INTO BILL\_DETAILS VALUES(3,'2017-01-03',1530.10,20.11,null,3,3);

INSERT INTO BILL\_DETAILS VALUES(4,'2017-01-04',1040.10,20.11,null,4,4);

INSERT INTO BILL\_DETAILS VALUES(5,'2017-01-05',10420.10,20.11,null,5,5);

INSERT INTO BILL\_DETAILS VALUES(6,'2017-01-06',140.10,20.11,null,6,6);

INSERT INTO BILL\_DETAILS VALUES(7,'2017-01-07',200.10,20.11,null,7,7);

INSERT INTO BILL\_DETAILS VALUES(8,'2017-01-08',150.10,20.11,null,8,8);

INSERT INTO BILL\_DETAILS VALUES(9,'2017-01-09',2400.10,20.11,null,9,9);

INSERT INTO BILL\_DETAILS VALUES(10,'2017-01-10',5300.10,20.11,null,10,10);

INSERT INTO CUSTOMER\_SERVICE VALUES(2,'Aditya','Pathak');

INSERT INTO CUSTOMER\_SERVICE VALUES(3,'Avneet','Kaur');

INSERT INTO CUSTOMER\_SERVICE VALUES(4,'Avnoor','Singh');

INSERT INTO CUSTOMER\_SERVICE VALUES(5,'Twesha','Arvind');

INSERT INTO CUSTOMER\_SERVICE VALUES(6,'AAkash','goel');

INSERT INTO CUSTOMER\_SERVICE VALUES(7,'Sushant','Malik');

INSERT INTO CUSTOMER\_SERVICE VALUES(8,'Gurkirat','Singh');

INSERT INTO CUSTOMER\_SERVICE VALUES(9,'Samar','Deol');

INSERT INTO CUSTOMER\_SERVICE VALUES(10,'Piyush','gumber');

INSERT INTO CUSTOMER\_SERVICE VALUES(11,'nitish','garg');

INSERT INTO FEEDBACK VALUES(2,'good','abhi@gmail.com',1,1,6);

INSERT INTO FEEDBACK VALUES(3,'nice','abcd@gmail.com',2,5,10);

INSERT INTO FEEDBACK VALUES(4,'good experience','ijjbhjv@gmail.com',5,2,2);

INSERT INTO FEEDBACK VALUES(5,'not so good','cjce@gmail.com',8,4,7);

INSERT INTO FEEDBACK VALUES(6,'not so bad','jbjer@gmail.com',3,7,5);

INSERT INTO FEEDBACK VALUES(7,'so good','vjuwhsi@gmail.com',9,3,8);

INSERT INTO FEEDBACK VALUES(8,'disappointing','ankji@gmail.com',10,8,1);

INSERT INTO FEEDBACK VALUES(9,'not so good','akjhcai@gmail.com',2,10,3);

INSERT INTO FEEDBACK VALUES(10,'very good','sdfw@gmail.com',10,9,1);

INSERT INTO OWNS VALUES(1,1);

INSERT INTO OWNS VALUES(2,1);

INSERT INTO OWNS VALUES(3,1);

INSERT INTO OWNS VALUES(4,1);

INSERT INTO OWNS VALUES(5,1);

INSERT INTO OWNS VALUES(6,1);

INSERT INTO OWNS VALUES(7,1);

INSERT INTO OWNS VALUES(8,1);

INSERT INTO OWNS VALUES(9,1);

INSERT INTO OWNS VALUES(10,1);

INSERT INTO OWNER\_TAXI VALUES(1,2);

INSERT INTO OWNER\_TAXI VALUES(2,7);

INSERT INTO OWNER\_TAXI VALUES(3,4);

INSERT INTO OWNER\_TAXI VALUES(4,1);

INSERT INTO OWNER\_TAXI VALUES(5,6);

INSERT INTO OWNER\_TAXI VALUES(6,5);

INSERT INTO OWNER\_TAXI VALUES(7,3);

INSERT INTO OWNER\_TAXI VALUES(8,9);

INSERT INTO OWNER\_TAXI VALUES(9,10);

INSERT INTO OWNER\_TAXI VALUES(10,8);

INSERT INTO INDIVIDUAL VALUES(100,"Rakesh",1);

INSERT INTO INDIVIDUAL VALUES(101,"Radha",1);

INSERT INTO INDIVIDUAL VALUES(102,"Rinki",1);

INSERT INTO INDIVIDUAL VALUES(103,"Mahesh",1);

INSERT INTO INDIVIDUAL VALUES(104,"Varesh",1);

INSERT INTO INDIVIDUAL VALUES(105,"Rashi",1);

INSERT INTO INDIVIDUAL VALUES(106,"Kavya",1);

INSERT INTO INDIVIDUAL VALUES(107,"Yukti",1);

INSERT INTO INDIVIDUAL VALUES(108,"Fatima",1);

INSERT INTO INDIVIDUAL VALUES(109,"Pushpa",1);

INSERT INTO TAXI\_SERVICE\_COMPANY VALUES (1,'Akshit',2);

INSERT INTO TAXI\_SERVICE\_COMPANY VALUES (2,'Amit',3);

INSERT INTO TAXI\_SERVICE\_COMPANY VALUES (3,'Akshima',1);

INSERT INTO TAXI\_SERVICE\_COMPANY VALUES (4,'Arti',1);

INSERT INTO TAXI\_SERVICE\_COMPANY VALUES (5,'Radhika',2);

INSERT INTO TAXI\_SERVICE\_COMPANY VALUES (6,'Vicky',3);

INSERT INTO TAXI\_SERVICE\_COMPANY VALUES (7,'Lakshay',1);

INSERT INTO TAXI\_SERVICE\_COMPANY VALUES (8,'Arzoo',2);

INSERT INTO TAXI\_SERVICE\_COMPANY VALUES (9,'Mohit',1);

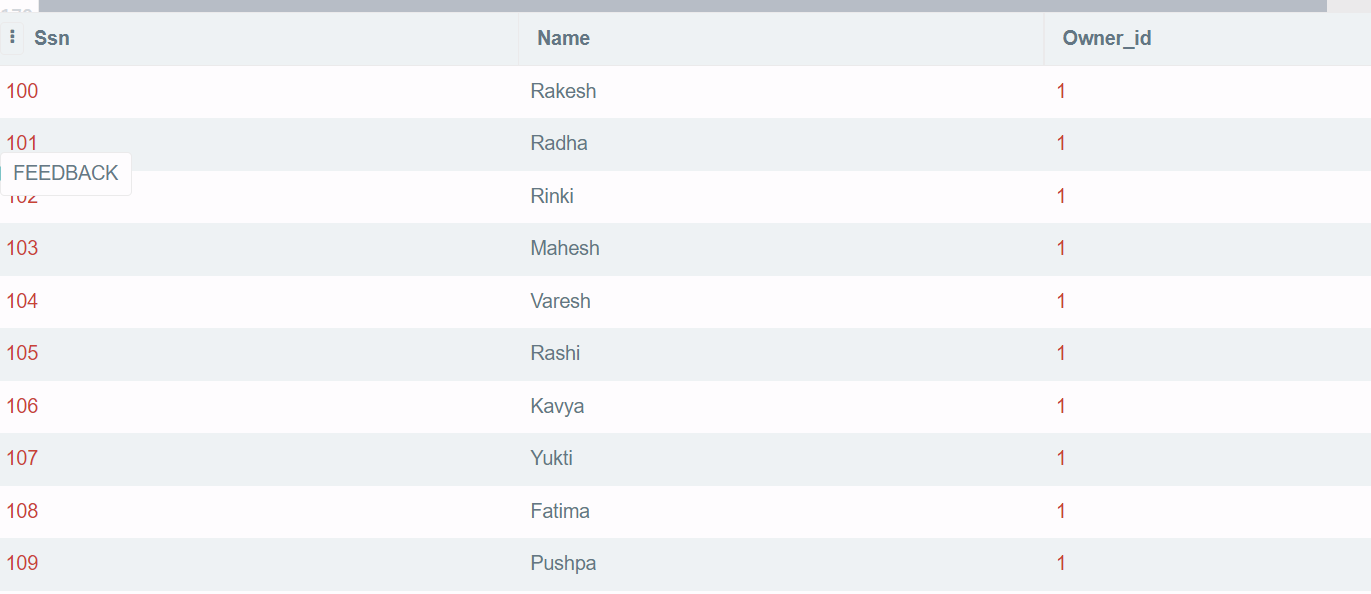
INSERT INTO TAXI\_SERVICE\_COMPANY VALUES (10,'Sid',1);

OUTPUT SCREENSHOTS









# NORMALIZATION OF RELATIONAL SCHEMA

#### TAXI

#### {Taxi\_id  Registration\_no, Taxi\_Model, Taxi\_Year, Taxi\_type, Status}

#### USER

#### {Usr\_id  F\_name, L\_name, Contat\_no, Gender, Address, Taxi\_id}

#### DRIVER

#### {Driver\_id  F\_name, L\_name, Gender, Conatct\_no, Rating, Age}

#### TRIP\_DETAILS

#### {Trip\_id  Trip\_date, Trip\_amt, Driver\_id, Usr\_id, Taxi\_id, Strt\_time, End\_time}

#### BILL\_DETAILS

#### {Bill\_no  Bill\_date, Advance\_amt, Discount\_amt, Total\_amt, Usr\_id, Trip\_id}

#### CUSTOMER\_SERVICE

#### {Emp\_id  F\_name, L\_name}

#### FEEDBACK

#### {Fbk\_id  Message, Email, Emp\_id, Usr\_id, Trip\_id}

#### OWNER\_TAXI

#### {Owner\_id  Taxi\_id}

#### OWNS

#### {Owner\_id  No\_Cars}

#### INDIVIDUAL

#### {Ssn  Name, Owner\_id}

#### TAXI\_SERVICE\_COMPANY

#### {Tsc\_id  Tsc\_name, Owner\_id}