



**Project:**

**CS-241**

**Spring 2024**

**Submitted to:**

Dr Naveed anwer butt

**Roll No:**

22024119-014

## **Contents:**

- Introduction
- Requirements
- ER Diagram
- Creating database
- Creating table

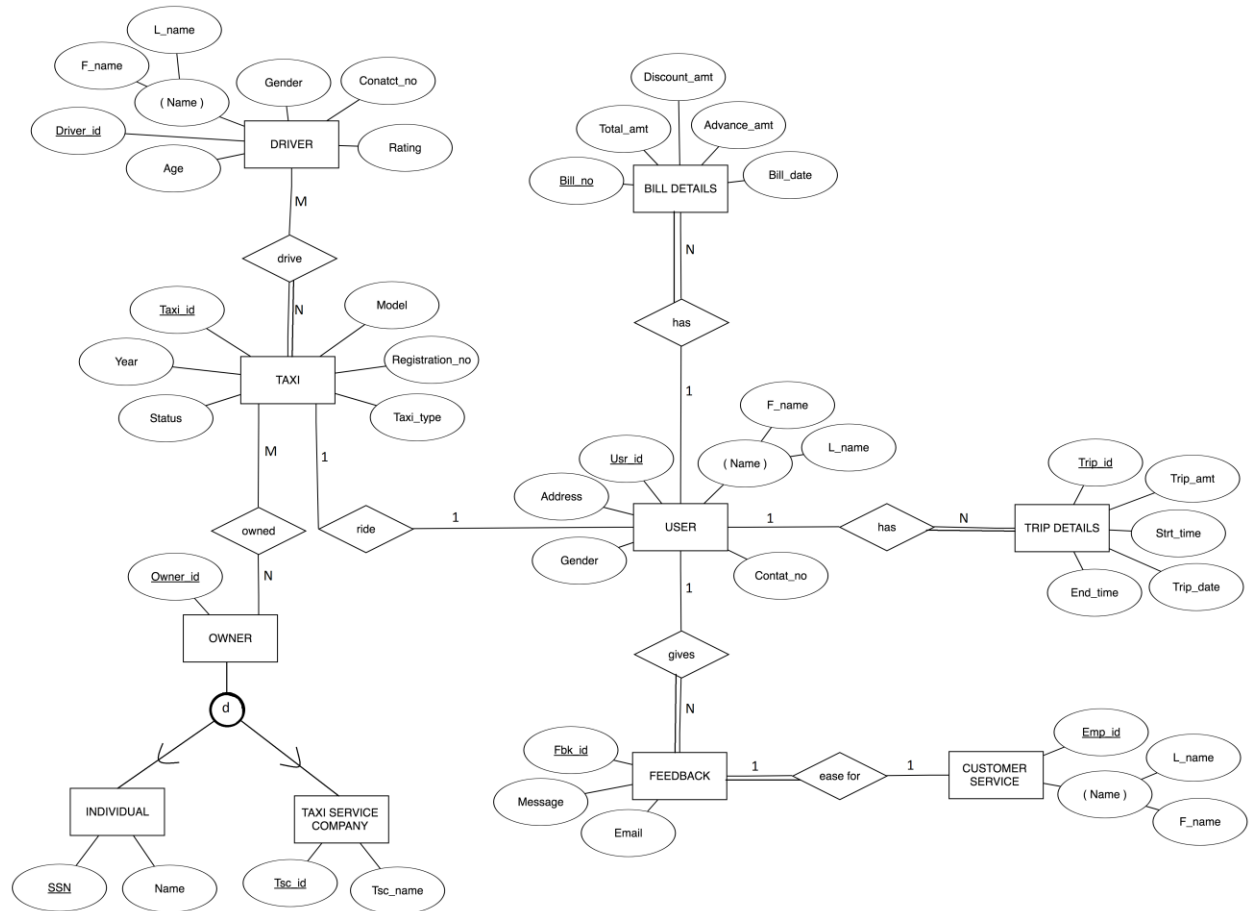
## Introduction:

A cab management software solution streamlines the taxicab business management processes and improves the overall experience of users. The primary requirement of every taxi business owner is to provide a streamlined cab booking solution and manage their fleet of cars through their mobile or tablets.

## 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.
- 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.
- 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 registered user will be provided with a login id and password. A customer can save his credit/debit card details for future payment.
- 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.
- Feedback can be taking by the customer service center representative. They have the information like emp\_id, name and email.

## ER Diagram:



## Creating database:

```
CREATE DATABASE TAXI_MANAGMENT_SYSTEM
```

## Creating tables:

```
-- 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)  
);
```

```
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)  
);
```

```
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)  
);
```

```
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),
    UNIQUE (Trip_id)
);

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),
    UNIQUE (Emp_id)
);

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)
);

CREATE TABLE INDIVIDUAL (
    Ssn integer NOT NULL,
    Name VARCHAR(20),
    Owner_id integer,
    PRIMARY KEY (Ssn)
);

CREATE TABLE TAXI_SERVICE_COMPANY (
    Tsc_id integer NOT NULL,
    Tsc_name VARCHAR(20),
    Owner_id integer,
    PRIMARY KEY (Tsc_id)
);

```

```

-----
-- Foreign key creation
-----

```

```

ALTER TABLE TAXI ADD CONSTRAINT fketadr FOREIGN KEY (Driver_id) REFERENCES
DRIVER(Driver_id) ON DELETE CASCADE;
ALTER TABLE USER_TBL ADD CONSTRAINT fkusta FOREIGN KEY (Taxi_id) REFERENCES TAXI(Taxi_id)
ON DELETE CASCADE;
ALTER TABLE TRIP_DETAILS ADD CONSTRAINT fktddr FOREIGN KEY (Driver_id) REFERENCES
DRIVER(Driver_id) ON DELETE CASCADE;
ALTER TABLE TRIP_DETAILS ADD CONSTRAINT fktdustr FOREIGN KEY (Usr_id) REFERENCES
USER_TBL(Usr_id) ON DELETE CASCADE;
ALTER TABLE TRIP_DETAILS ADD CONSTRAINT fktntax FOREIGN KEY (Taxi_id) REFERENCES
TAXI(Taxi_id) ON DELETE CASCADE;
ALTER TABLE BILL_DETAILS ADD CONSTRAINT fkbtd FOREIGN KEY (Trip_id) REFERENCES
TRIP_DETAILS(Trip_id) ON DELETE CASCADE;
ALTER TABLE BILL_DETAILS ADD CONSTRAINT fkbdustr FOREIGN KEY (Usr_id) REFERENCES
USER_TBL(Usr_id) ON DELETE CASCADE;
ALTER TABLE FEEDBACK ADD CONSTRAINT fkfbemp FOREIGN KEY (Emp_id) REFERENCES
CUSTOMER_SERVICE(Emp_id) ON DELETE CASCADE;
ALTER TABLE FEEDBACK ADD CONSTRAINT fkfbtd FOREIGN KEY (Trip_id) REFERENCES
TRIP_DETAILS(Trip_id) ON DELETE CASCADE;
ALTER TABLE FEEDBACK ADD CONSTRAINT fkfbusr FOREIGN KEY (Usr_id) REFERENCES
USER_TBL(Usr_id) ON DELETE CASCADE;
ALTER TABLE OWNER_TAXI ADD CONSTRAINT fkeowtax FOREIGN KEY (Taxi_id) REFERENCES
TAXI(Taxi_id) ON DELETE CASCADE;
ALTER TABLE OWNER_TAXI ADD CONSTRAINT fkeowowns FOREIGN KEY (Owner_id) REFERENCES
OWNS(Owner_id) ON DELETE CASCADE;
ALTER TABLE INDIVIDUAL ADD CONSTRAINT fkeinowns FOREIGN KEY (Owner_id) REFERENCES
OWNS(Owner_id) ON DELETE CASCADE;
ALTER TABLE TAXI_SERVICE_COMPANY ADD CONSTRAINT fketscowns FOREIGN KEY (Owner_id)
REFERENCES OWNS(Owner_id) ON DELETE CASCADE;

```

```

-----
-- Insert Commands
-----

```

```

INSERT INTO TAXI VALUES(1,'KA-15R-3367','BENZE
300',to_date('01/01/2017','mm/dd/yyyy'),'SUV','Available',1)
INSERT INTO DRIVER VALUES(1,'Abhi','Gowda','Male','4693805870',5,25);
INSERT INTO USER_TBL VALUES(1,'USER1','LNAME','123456','Male','MCCallum','1');
INSERT INTO TRIP_DETAILS
VALUES(1,to_date('01/01/2017','mm/dd/yyyy'),123,1,1,1,TO_TIMESTAMP('2017-01-01 06:14:00',
'YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2017-01-01 08:14:00','YYYY-MM-DD HH24:MI:SS'));
INSERT INTO BILL_DETAILS
VALUES(1,to_date('01/01/2017','mm/dd/yyyy'),1000.10,20.11,null,1,1);
INSERT INTO CUSTOMER_SERVICE VALUES(1,'prashuk','ajmera');
INSERT INTO CUSTOMER_SERVICE VALUES(1,'abhi','gowda');
INSERT INTO FEEDBACK VALUES(1,'good','prashuk.ajmera@gmail.com',1,1,1);
INSERT INTO FEEDBACK VALUES(1,'not so good','abhi@gmail.com',1,1,1);
INSERT INTO OWNS VALUES(1,1);
INSERT INTO OWNS VALUES(2,1);
INSERT INTO OWNER_TAXI (1,1);
INSERT INTO INDIVIDUAL VALUES(123,'abhi owner ind',1);
INSERT INTO TAXI_SERVICE_COMPANY VALUES (1,'abhi taxi comp',2);

```

```

-----
-- Procedure Creation 1
-----

```

```

CREATE OR REPLACE PROCEDURE BOOK_TAXI
( Name IN VARCHAR2
, v_Address IN VARCHAR2
, v_Contact IN VARCHAR2
, Taxi_Model IN VARCHAR2
, v_Gender IN VARCHAR2
, Advance IN decimal
)
AS
BEGIN
DECLARE
v_usr_id INT := -1;
v_Trip_id INT := -1;
v_Bill_no INT := -1;
v_Taxi_id INT := -1;
v_Driver_id INT := 1;
BEGIN
select MAX(Usr_id)+1 into v_usr_id from USER_TBL ;
select MAX(Trip_id)+1 into v_Trip_id from TRIP_DETAILS ;
select MAX(Bill_no)+1 into v_Bill_no from BILL_DETAILS ;
select taxi_id, Driver_id into v_Taxi_id,v_Driver_id from TAXI where Status =
'Available' and Taxi_Model = Taxi_Model;

INSERT INTO USER_TBL values(v_usr_id, SUBSTR (Name, 1, INSTR(Name, ' ',1)),SUBSTR (Name,
INSTR(Name, ' ',1)+1,LENGTH(Name)),v_Contact,v_Gender,v_Address,v_Taxi_id);
INSERT INTO TRIP_DETAILS values(v_Trip_id,sysdate,
50,v_Driver_id,v_usr_id,v_Taxi_id,sysdate,null);
INSERT INTO BILL_DETAILS values(v_Bill_no,null,Advance,null,null,v_usr_id,v_Trip_id);

END ;
END;
/

```

```

-----
-- Procedure Creation 2
-----

```

```

CREATE OR REPLACE PROCEDURE TRIP_END(v_trip IN INT , v_discount IN Decimal )
AS
BEGIN
DECLARE
v_total_time INT := -1;
v_bill_no INT := -1;
BEGIN
select extract(day from (sysdate - Strt_time))*24 + extract(hour from (sysdate -
Strt_time)) into v_total_time from TRIP_DETAILS where Trip_id = v_trip;

update TRIP_DETAILS set End_time = sysdate where Trip_id = Trip_id ;
update BILL_DETAILS set Bill_date = sysdate , Discount_amt = v_discount ,Total_amt =
(v_total_time * 15) - v_discount where Trip_id = v_trip ;
END ;
END ;
/

```

-- Trigger Creation 1

```
-----  
CREATE OR REPLACE TRIGGER UPDATE_DRIVER_RATING  
AFTER INSERT ON FEEDBACK  
FOR EACH ROW  
WHEN (NEW.Message like '%Bad Driver%' )  
DECLARE  
    v_driver_id INT;  
BEGIN  
    select driver_id into v_driver_id from TRIP_DETAILS where trip_id = :NEW.Trip_id;  
  
    update DRIVER set Rating = Rating -1 where driver_id = v_driver_id;  
END;  
/
```

-- Trigger Creation 2

```
-----  
CREATE OR REPLACE TRIGGER ADD_NO_OF_CARS  
BEFORE INSERT OR UPDATE ON OWNS  
FOR EACH ROW  
DECLARE  
    v_no_of_cars INT;  
BEGIN  
    select count(Taxi_id) into v_no_of_cars from OWNER_TAXI where Owner_id = :NEW.Owner_id  
group by Owner_id;  
    :NEW.No_Cars := v_no_of_cars;  
END;
```