

DATABASE SYSTEM FOR TAXI SERVICE

GROUP PROJECT

ANUPRIYA LATHEY -- 102103373

NITLEEN KAUR -- 102103377

PIA GUPTA -- 102103394

INDEX

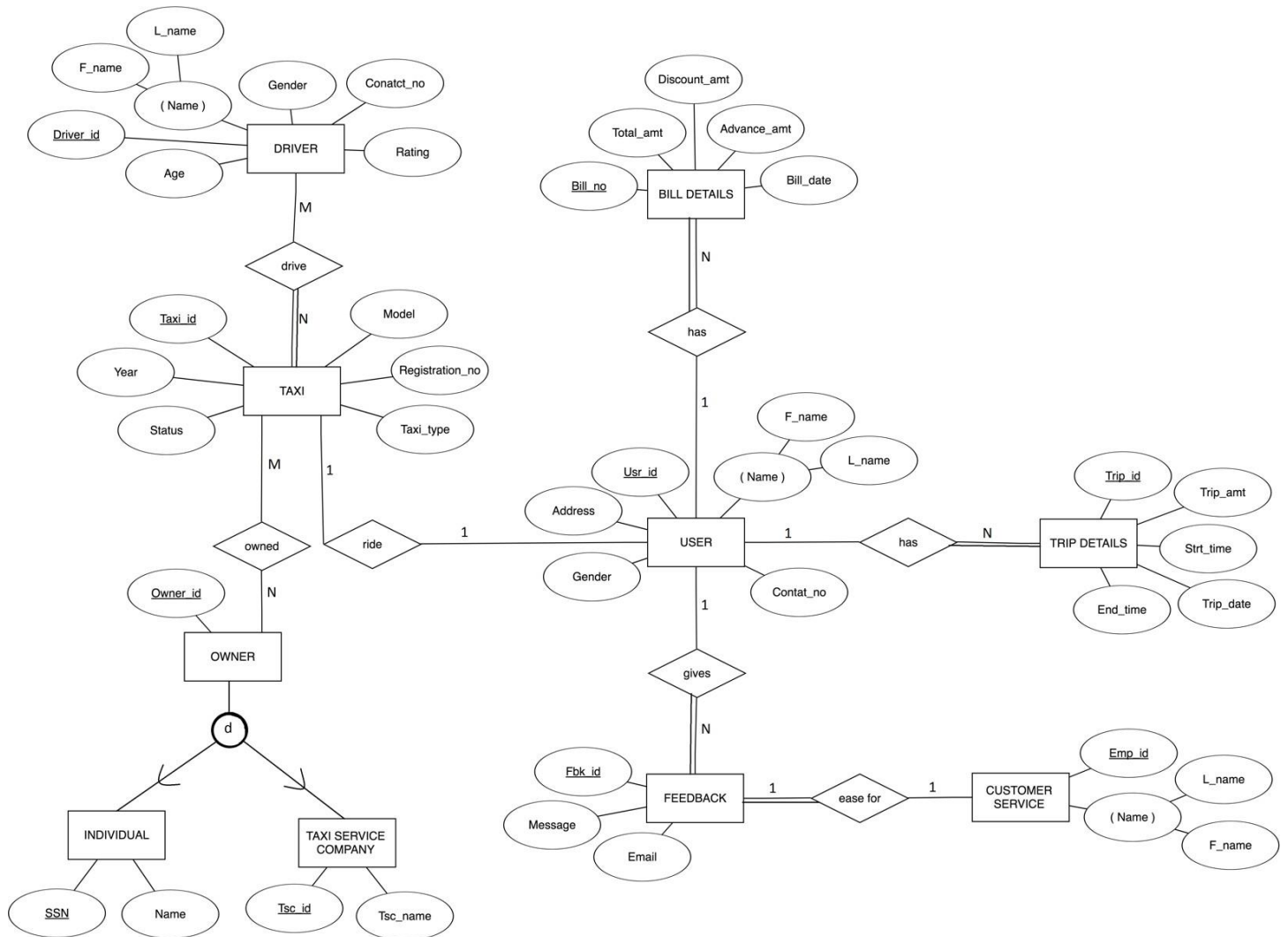
- ***INTRODUCTION & REQUIREMENT ANALYSIS***
- ***MODELLING OF REQUIREMENTS AS ER-DIAGRAM***
- ***MAPPING OF ERD IN RELATIONAL SCHEMA***
- ***SQL STATEMENTS FOR TABLE CREATION***
- ***SQL STATEMENTS FOR FOREIGN KEY CREATION***
- ***SQL STATEMENTS FOR INSERT COMMANDS***
- ***PL/SQL – PROCEDURES***
- ***PL/SQL – TRIGGERS***
- ***NORMALIZATION OF RELATIONAL SCHEMA***
- ***CONCLUSION***
- ***REFERENCES***

INTRODUCTION & REQUIREMENT ANALYSIS

The Taxi Service Database is based on inter-city travel facility provided by taxi company **RideX**. It revolves around **user information**, **drivers' information** and **trip details**. The 11 entities created below involve corresponding entries as explained –

- 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.
- Partial booking payment is made at the time of booking and the balance must be paid by the user at the end of the trip.
- 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.

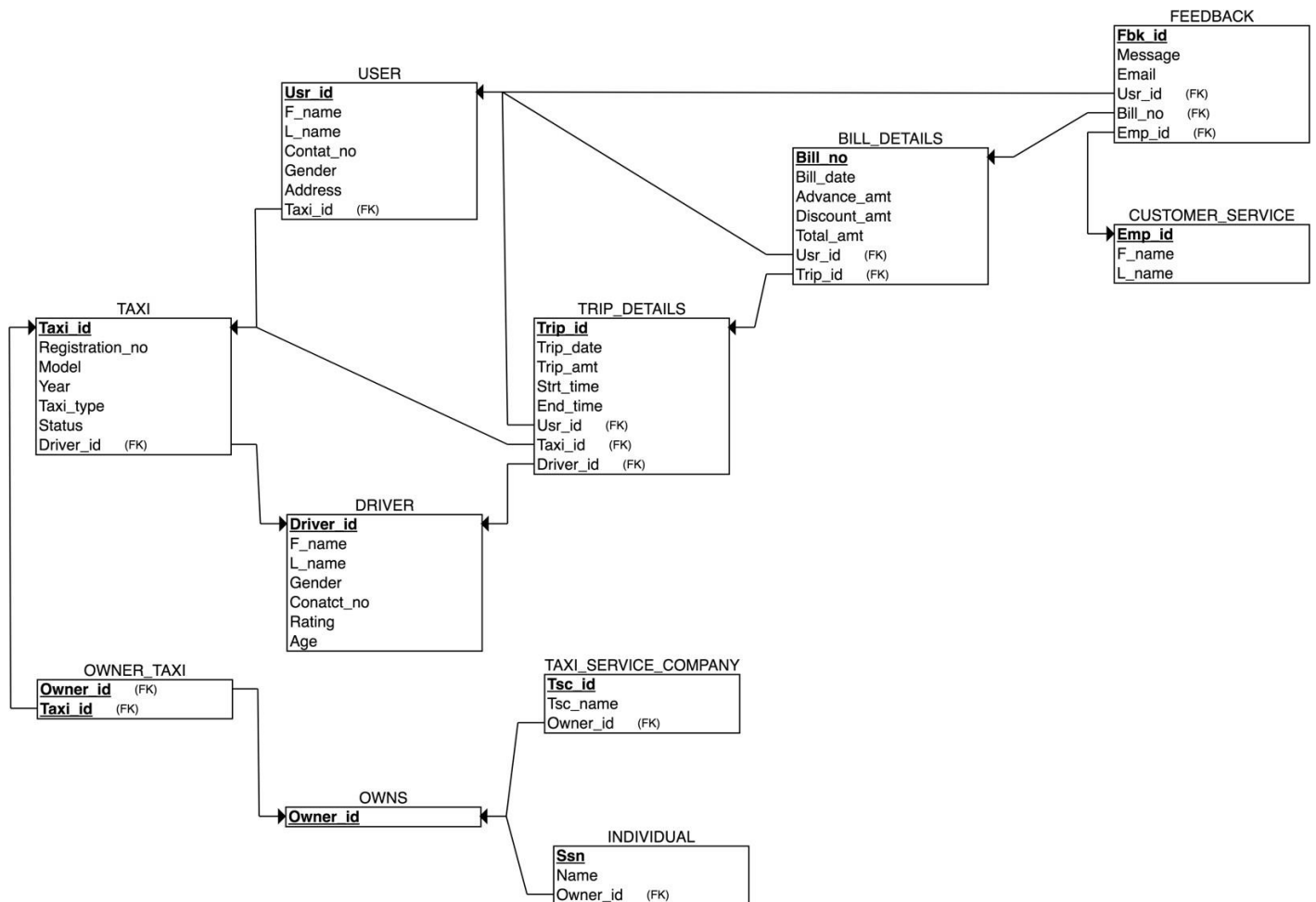
MODELLING OF REQUIREMENTS AS ER-DIAGRAM



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 ERD IN RELATIONAL SCHEMA



TAXI

Taxi_id	Registration_no	Taxi_Model	Taxi_Year	Taxi_type	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

Owner_id	Taxi_id
----------	---------

- Primary Key: Owner_id, Taxi_id
- Foreign Keys: Owner_id, Taxi_id

OWNS

Owner_id	No_Cars
----------	---------

- 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

NORMALIZATION OF RELATIONAL SCHEMA

- TAXI
{Taxi_id → Registration_no, Taxi_Model, Taxi_Year, Taxi_type, Status}
- USER
{Usr_id → F_name, L_name, Contact_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}

SQL STATEMENTS FOR 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)  
);
```

SQL STATEMENTS FOR FOREIGN KEY CREATION

```
ALTER TABLE TAXI ADD CONSTRAINT fk1 FOREIGN KEY (Driver_id) REFERENCES  
DRIVER(Driver_id) ON DELETE CASCADE;
```

```
ALTER TABLE USER_TBL ADD CONSTRAINT fk2 FOREIGN KEY (Taxi_id) REFERENCES  
TAXI(Taxi_id) ON DELETE CASCADE;
```

```
ALTER TABLE TRIP_DETAILS ADD CONSTRAINT fk3 FOREIGN KEY (Driver_id)  
REFERENCES DRIVER(Driver_id) ON DELETE CASCADE;
```

```
ALTER TABLE TRIP_DETAILS ADD CONSTRAINT fk4 FOREIGN KEY (Usr_id) REFERENCES  
USER_TBL(Usr_id) ON DELETE CASCADE;
```

```
ALTER TABLE TRIP_DETAILS ADD CONSTRAINT fk5 FOREIGN KEY (Taxi_id) REFERENCES  
TAXI(Taxi_id) ON DELETE CASCADE;
```

```
ALTER TABLE BILL_DETAILS ADD CONSTRAINT fk6 FOREIGN KEY (Trip_id) REFERENCES  
TRIP_DETAILS(Trip_id) ON DELETE CASCADE;
```

```
ALTER TABLE BILL_DETAILS ADD CONSTRAINT fk7 FOREIGN KEY (Usr_id) REFERENCES  
USER_TBL(Usr_id) ON DELETE CASCADE;
```

```
ALTER TABLE FEEDBACK ADD CONSTRAINT fk8 FOREIGN KEY (Emp_id) REFERENCES  
CUSTOMER_SERVICE(Emp_id) ON DELETE CASCADE;
```

```
ALTER TABLE FEEDBACK ADD CONSTRAINT fk9 FOREIGN KEY (Trip_id) REFERENCES  
TRIP_DETAILS(Trip_id) ON DELETE CASCADE;
```

```
ALTER TABLE FEEDBACK ADD CONSTRAINT fk10 FOREIGN KEY (Usr_id) REFERENCES  
USER_TBL(Usr_id) ON DELETE CASCADE;
```

```
ALTER TABLE OWNER_TAXI ADD CONSTRAINT fk11 FOREIGN KEY (Taxi_id) REFERENCES  
TAXI(Taxi_id) ON DELETE CASCADE;
```

```
ALTER TABLE OWNER_TAXI ADD CONSTRAINT fk12 FOREIGN KEY (Owner_id) REFERENCES  
OWNS(Owner_id) ON DELETE CASCADE;
```

```
ALTER TABLE INDIVIDUAL ADD CONSTRAINT fk13 FOREIGN KEY (Owner_id) REFERENCES  
OWNS(Owner_id) ON DELETE CASCADE;
```

```
ALTER TABLE TAXI_SERVICE_COMPANY ADD CONSTRAINT fk14 FOREIGN KEY (Owner_id)  
REFERENCES OWNS(Owner_id) ON DELETE CASCADE;
```

SQL STATEMENTS FOR INSERT COMMANDS

```
INSERT INTO DRIVER VALUES(1,'Amit','Sharma','Male','9693805870',5,23);
INSERT INTO DRIVER VALUES(2,'Raghav','Goel','Male','8693665854',4,27);
INSERT INTO DRIVER VALUES(3,'Mahesh','Mishra','Male','8773805888',2,35);
INSERT INTO DRIVER VALUES(4,'Gaurav','Singh','Male','3453805870',3,29);
INSERT INTO DRIVER VALUES(5,'Smriti','Gupta','Female','4693805870',5,45);
INSERT INTO DRIVER VALUES(6,'Mehak','Sinha','Female','8693877822',5,28);
INSERT INTO DRIVER VALUES(7,'Prem','Malik','Male','7693805113',4,25);
INSERT INTO DRIVER VALUES(8,'Priya','Sharma','Female','2693805891',3,28);
INSERT INTO DRIVER VALUES(9,'Abhishek','Aggarwal','Male','6622215870',2,23);
INSERT INTO DRIVER VALUES(10,'Meera','Bhalla','Female','1233805866',5,33);
```

```
INSERT INTO TAXI VALUES(101,'T0501','BENZE
300',to_date('01/01/2017','dd/mm/yyyy'),'SUV','Available',1);
INSERT INTO TAXI VALUES(102,'T0502','MACRO
500',to_date('01/01/2016','dd/mm/yyyy'),'Standard','Not Available',2);
INSERT INTO TAXI VALUES(103,'T0503','MINI
400',to_date('01/01/2009','dd/mm/yyyy'),'Economy','Not Available',3);
INSERT INTO TAXI VALUES(104,'T0504','XUV
300',to_date('01/01/2010','dd/mm/yyyy'),'SUV','Available',4);
INSERT INTO TAXI VALUES(105,'T0505','BREZZA
300',to_date('01/01/2019','dd/mm/yyyy'),'Premium','Available',5);
INSERT INTO TAXI VALUES(106,'T0506','BENZE
900',to_date('01/01/2019','dd/mm/yyyy'),'SUV','Not Available',6);
INSERT INTO TAXI VALUES(107,'T0507','SWIFT
500',to_date('01/01/2017','dd/mm/yyyy'),'Standard','Available',7);
INSERT INTO TAXI VALUES(108,'T0508','XUV
700',to_date('01/01/2019','dd/mm/yyyy'),'SUV','Not Available',8);
INSERT INTO TAXI VALUES(109,'T0509','MINI
300',to_date('01/01/2020','dd/mm/yyyy'),'Minivan','Available',9);
INSERT INTO TAXI VALUES(110,'T0510','MACRO
900',to_date('01/01/2019','dd/mm/yyyy'),'Premium','Available',10);
INSERT INTO TAXI VALUES(111,'T0511','WAGON
300',to_date('01/01/2018','dd/mm/yyyy'),'Economy','Not Available',1);
INSERT INTO TAXI VALUES(112,'T0512','MAGIC
300',to_date('01/01/2011','dd/mm/yyyy'),'Minivan','Available',2);
```

```
INSERT INTO USER_TBL
VALUES(201,'Raghav','Nayak','9451277009','Male','Bengaluru','105');
INSERT INTO USER_TBL
VALUES(202,'Aryan','Sinha','9876543212','Male','Gurgaon','107');
INSERT INTO USER_TBL
VALUES(203,'Ananya','Pathak','8761122345','Female','Delhi','109');
INSERT INTO USER_TBL
VALUES(204,'Devansh','Merchant','7896544490','Male','Jalandhar','109');
INSERT INTO USER_TBL
VALUES(205,'Akash','Mani','8866459012','Male','Patiala','112');
INSERT INTO USER_TBL
VALUES(206,'Seema','Goel','6578990123','Female','Amritsar','101');
```

```

INSERT INTO USER_TBL
VALUES(207,'Vivek','Diwan','7658994321','Male','Dehradun','112');
INSERT INTO USER_TBL
VALUES(208,'Akash','Jha','9866459012','Male','Mumbai','110');
INSERT INTO USER_TBL
VALUES(209,'Pragya','Sharma','9800678443','Female','Chennai','104');
INSERT INTO USER_TBL
VALUES(210,'Aneesha','Sachdev','9993345678','Female','Varanasi','105');

```

```

INSERT INTO TRIP_DETAILS
VALUES(301,to_date('21/01/2023','dd/mm/yyyy'),1100,1,206,101,TO_TIMESTAMP('20
23-01-21 06:14:00','YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2023-01-21
08:14:00','YYYY-MM-DD HH24:MI:SS'));
INSERT INTO TRIP_DETAILS
VALUES(302,to_date('28/02/2023','dd/mm/yyyy'),500,5,201,105,TO_TIMESTAMP('202
3-02-28 07:00:00','YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2023-02-28
08:30:00','YYYY-MM-DD HH24:MI:SS'));
INSERT INTO TRIP_DETAILS
VALUES(303,to_date('01/03/2023','dd/mm/yyyy'),180,7,202,107,TO_TIMESTAMP('202
3-03-01 08:00:00','YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2023-03-01
10:25:00','YYYY-MM-DD HH24:MI:SS'));
INSERT INTO TRIP_DETAILS
VALUES(304,to_date('05/04/2023','dd/mm/yyyy'),250,4,209,104,TO_TIMESTAMP('202
3-04-05 09:00:00','YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2023-04-05
12:00:00','YYYY-MM-DD HH24:MI:SS'));
INSERT INTO TRIP_DETAILS
VALUES(305,to_date('15/03/2023','dd/mm/yyyy'),490,9,204,109,TO_TIMESTAMP('202
3-03-15 10:00:00','YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2023-03-15
13:30:00','YYYY-MM-DD HH24:MI:SS'));
INSERT INTO TRIP_DETAILS
VALUES(306,to_date('17/09/2022','dd/mm/yyyy'),360,1,206,101,TO_TIMESTAMP('202
2-09-17 10:30:00','YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2022-09-17
12:24:00','YYYY-MM-DD HH24:MI:SS'));
INSERT INTO TRIP_DETAILS
VALUES(307,to_date('27/12/2022','dd/mm/yyyy'),256,10,208,110,TO_TIMESTAMP('20
22-12-27 09:30:00','YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2022-12-27
11:30:00','YYYY-MM-DD HH24:MI:SS'));
INSERT INTO TRIP_DETAILS
VALUES(308,to_date('29/11/2022','dd/mm/yyyy'),2500,7,202,107,TO_TIMESTAMP('20
22-11-29 08:30:00','YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2022-11-29
10:20:00','YYYY-MM-DD HH24:MI:SS'));
INSERT INTO TRIP_DETAILS
VALUES(309,to_date('31/07/2022','dd/mm/yyyy'),790,2,205,112,TO_TIMESTAMP('202
2-07-31 07:30:00','YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2022-07-31
11:02:00','YYYY-MM-DD HH24:MI:SS'));
INSERT INTO TRIP_DETAILS
VALUES(310,to_date('30/05/2022','dd/mm/yyyy'),650,2,207,112,TO_TIMESTAMP('202
2-05-30 11:00:00','YYYY-MM-DD HH24:MI:SS'),TO_TIMESTAMP('2022-05-30
15:00:00','YYYY-MM-DD HH24:MI:SS'));

```

```

INSERT INTO BILL_DETAILS
VALUES(401,to_date('21/01/2023','dd/mm/yyyy'),200,100,1100,201,301);
INSERT INTO BILL_DETAILS
VALUES(402,to_date('28/02/2023','dd/mm/yyyy'),100,50,500,203,302);
INSERT INTO BILL_DETAILS
VALUES(403,to_date('01/03/2023','dd/mm/yyyy'),100,50,180,205,303);
INSERT INTO BILL_DETAILS
VALUES(404,to_date('05/04/2023','dd/mm/yyyy'),150,20,250,207,304);
INSERT INTO BILL_DETAILS
VALUES(405,to_date('15/03/2023','dd/mm/yyyy'),100,50,490,209,305);
INSERT INTO BILL_DETAILS
VALUES(406,to_date('17/09/2022','dd/mm/yyyy'),100,50,360,201,306);
INSERT INTO BILL_DETAILS
VALUES(407,to_date('27/12/2022','dd/mm/yyyy'),100,50,256,202,307);
INSERT INTO BILL_DETAILS
VALUES(408,to_date('29/11/2022','dd/mm/yyyy'),1000,500,2500,203,308);
INSERT INTO BILL_DETAILS
VALUES(409,to_date('31/07/2022','dd/mm/yyyy'),300,100,790,204,309);
INSERT INTO BILL_DETAILS
VALUES(410,to_date('30/05/2022','dd/mm/yyyy'),200,100,650,206,310);

```

```

INSERT INTO CUSTOMER_SERVICE VALUES(501,'Sara','Maheshwari');
INSERT INTO CUSTOMER_SERVICE VALUES(502,'Mohak','Gowda');

```

```

INSERT INTO FEEDBACK VALUES(601,'bad','seema_goel@gmail.com',501,206,301);
INSERT INTO FEEDBACK VALUES(602,'good','raghav@gmail.com',502,201,302);

```

```

INSERT INTO OWNS VALUES(701,7);
INSERT INTO OWNS VALUES(702,5);

```

```

INSERT INTO OWNER_TAXI VALUES(701,101);
INSERT INTO OWNER_TAXI VALUES(701,102);
INSERT INTO OWNER_TAXI VALUES(701,103);
INSERT INTO OWNER_TAXI VALUES(701,104);
INSERT INTO OWNER_TAXI VALUES(701,105);
INSERT INTO OWNER_TAXI VALUES(701,106);
INSERT INTO OWNER_TAXI VALUES(701,107);
INSERT INTO OWNER_TAXI VALUES(702,108);
INSERT INTO OWNER_TAXI VALUES(702,109);
INSERT INTO OWNER_TAXI VALUES(702,110);
INSERT INTO OWNER_TAXI VALUES(702,111);
INSERT INTO OWNER_TAXI VALUES(702,112);

```

```

INSERT INTO INDIVIDUAL VALUES(783,'Mahesh',702);
INSERT INTO TAXI_SERVICE_COMPANY VALUES (981,'ITC Car Company',701);

```

PL/SQL – PROCEDURES

Procedure Code 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 Code 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 ;
```

PL/SQL – TRIGGERS

Trigger 1

```
-----
-- Trigger Creation
-- this trigger is called when inserted(After) to the feedback
table
-- the trigger will decrease the driver rating by 1 if user feed
back is bad for a driver
-----

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 2

```
-----
-- Trigger Creation
-- this trigger is called before the INSERT OR UPDATE ON OWNS
table
-- the trigger will calculate the number of cars owned by the
owner and updates the no_of_cars columns in the OWNS table
-----

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;
```


PL/SQL – CURSOR

```
DECLARE
    CURSOR C1 IS SELECT * FROM BILL_DETAILS;
    Dnew NUMBER;
BEGIN
    FOR REC IN C1 LOOP
        IF REC.DISCOUNT_AMT=20 THEN
            Dnew:=30;
        else
            Dnew:=REC.DISCOUNT_AMT;
        END IF;
        Update BILL_DETAILS SET DISCOUNT_AMT=Dnew WHERE
BILL_NO=REC.BILL_NO;
    END LOOP;

END;
/
```

PL/SQL – FUNCTION

```
CREATE OR REPLACE FUNCTION TOTAL_SUM (U_id NUMBER)
RETURN NUMBER IS
    final_sum NUMBER;
BEGIN
    select sum(Total_amt) into final_sum from BILL_DETAILS where Usr_id=U_id;
    RETURN(final_sum);
END;
/

DECLARE
    U NUMBER;
    FINAL_SUM NUMBER;
BEGIN
    U:=201;    --U:=&Usr_id;
    FINAL_SUM:= TOTAL_SUM(U);
    DBMS_OUTPUT.PUT_LINE('Total amount paid by user with user id ' || U || ' is
    ' || FINAL_SUM);
END;
```

SNAPSHOTS

Live SQL

FeedbackHelpnkaur1_be21@thapar.edu

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

SQL Worksheet

ClearFindActionsSaveRun

TAXI_ID	REGISTRATION_NO	TAXI_MODEL	TAXI_YEAR	TAXI_TYPE	STATUS	DRIVER_ID
101	T0501	BENZE 300	01-JAN-17	SUV	Available	1
102	T0502	MACRO 500	01-JAN-16	Standard	Not Available	2
103	T0503	MINI 400	01-JAN-09	Economy	Not Available	3
104	T0504	XUV 300	01-JAN-10	SUV	Available	4
105	T0505	BREZZA 300	01-JAN-19	Premium	Available	5
106	T0506	BENZE 900	01-JAN-19	SUV	Not Available	6
107	T0507	SWIFT 500	01-JAN-17	Standard	Available	7
108	T0508	XUV 700	01-JAN-19	SUV	Not Available	8
109	T0509	MINI 300	01-JAN-20	Minivan	Available	9
110	T0510	MACRO 900	01-JAN-19	Premium	Available	10
111	T0511	WAGON 300	01-JAN-18	Economy	Not Available	1
112	T0512	MAGIC 300	01-JAN-11	Minivan	Available	2

2023 Oracle · Live SQL 23.1.5, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with using Oracle APEX · Privacy · Terms of Use

Live SQL

FeedbackHelpnkaur1_be21@thapar.edu

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

SQL Worksheet

ClearFindActionsSaveRun

USR_ID	F_NAME	L_NAME	CONTAT_NO	GENDER	ADDRESS	TAXI_ID
201	Raghav	Nayak	9451277009	Male	Bengaluru	105
202	Aryan	Sinha	9876543212	Male	Gurgaon	107
203	Ananya	Pathak	8761122345	Female	Delhi	109
204	Devansh	Merchant	7896544490	Male	Jalandhar	109
205	Akash	Mani	8866459012	Male	Patiala	112
206	Seema	Goel	6578990123	Female	Amritsar	101
207	Vivek	Diwan	7658994321	Male	Dehradun	112
208	Akash	Jha	9866459012	Male	Mumbai	110
209	Pragya	Sharma	9800678443	Female	Chennai	104
210	Aneesha	Sachdev	9993345678	Female	Varanasi	105

Download CSV

2023 Oracle · Live SQL 23.1.5, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with using Oracle APEX · Privacy · Terms of Use

Live SQL

FeedbackHelpnkaur1_be21@thapar.edu

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

SQL Worksheet

ClearFindActionsSaveRun

DRIVER_ID	F_NAME	L_NAME	GENDER	CONATCT_NO	RATING	AGE
1	Amit	Sharma	Male	9693805870	5	23
2	Raghav	Goel	Male	8693665854	4	27
3	Mahesh	Mishra	Male	8773805888	2	35
4	Gaurav	Singh	Male	3453805870	3	29
5	Smriti	Gupta	Female	4693805870	5	45
6	Mehak	Sinha	Female	8693877822	5	28
7	Prem	Malik	Male	7693805113	4	25
8	Priya	Sharma	Female	2693805891	3	28
9	Abhishek	Aggarwal	Male	6622215870	2	23
10	Meera	Bhalla	Female	1233805866	5	33

Download CSV

Live SQL

FeedbackHelpnkaur1_be21@thapar.edu

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

SQL Worksheet

ClearFindActionsSaveRun

TRIP_ID	TRIP_DATE	TRIP_AMT	DRIVER_ID	USR_ID	TAXI_ID	STRT_TIME	END_TIME
301	21-JAN-23	1100	1	206	101	21-JAN-23 06.14.00.000000 AM	21-JAN-23 08.14.00.000000 AM
302	28-FEB-23	500	5	201	105	28-FEB-23 07.00.00.000000 AM	28-FEB-23 08.30.00.000000 AM
303	01-MAR-23	180	7	202	107	01-MAR-23 08.00.00.000000 AM	01-MAR-23 10.25.00.000000 AM
304	05-APR-23	250	4	209	104	05-APR-23 09.00.00.000000 AM	05-APR-23 12.00.00.000000 PM
305	15-MAR-23	490	9	204	109	15-MAR-23 10.00.00.000000 AM	15-MAR-23 01.30.00.000000 PM
306	17-SEP-22	360	1	206	101	17-SEP-22 10.30.00.000000 AM	17-SEP-22 12.24.00.000000 PM
307	27-DEC-22	256	10	208	110	27-DEC-22 09.30.00.000000 AM	27-DEC-22 11.30.00.000000 AM
308	29-NOV-22	2500	7	202	107	29-NOV-22 08.30.00.000000 AM	29-NOV-22 10.20.00.000000 AM
309	31-JUL-22	790	2	205	112	31-JUL-22 07.30.00.000000 AM	31-JUL-22 11.02.00.000000 AM
310	30-MAY-22	650	2	207	112	30-MAY-22 11.00.00.000000 AM	30-MAY-22 03.00.00.000000 PM

Download CSV

HomeSQL WorksheetMy SessionSchemaQuick SQLMy ScriptsMy TutorialsCode Library

SQL Worksheet

ClearFindActionsSaveRun

BILL_NO	BILL_DATE	ADVANCE_AMT	DISCOUNT_AMT	TOTAL_AMT	USR_ID	TRIP_ID
401	21-JAN-23	200	100	1100	201	301
402	28-FEB-23	100	50	500	203	302
403	01-MAR-23	100	50	180	205	303
404	05-APR-23	150	20	250	207	304
405	15-MAR-23	100	50	490	209	305
406	17-SEP-22	100	50	360	201	306
407	27-DEC-22	100	50	256	202	307
408	29-NOV-22	1000	500	2500	203	308
409	31-JUL-22	300	100	790	204	309
410	30-MAY-22	200	100	650	206	310

Download CSV

HomeSQL WorksheetMy SessionSchemaQuick SQLMy ScriptsMy TutorialsCode Library

SQL Worksheet

ClearFindActionsSaveRun

Download CSV

10 rows selected.

EMP_ID	F_NAME	L_NAME
501	Sara	Maheshwari
502	Mohak	Gowda

Download CSV

2 rows selected.

FBK_ID	MESSAGE	EMAIL	EMP_ID	USR_ID	TRIP_ID
601	bad service	seema_goel@gmail.com	501	206	301
602	good	raghav@gmail.com	502	201	302

Download CSV

2 rows selected.

≡

Live SQL

FeedbackHelpnkaur1_be21@thapar.edu

🌙

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

ClearFindActionsSaveRun

601	bad service	seema_goel@gmail.com	501	206	301
602	good	raghav@gmail.com	502	201	302

Download CSV

2 rows selected.

OWNER_ID	NO_CARS
701	7
702	5

Download CSV

2 rows selected.

OWNER_ID	TAXI_ID
701	101
701	102

≡

Live SQL

FeedbackHelpnkaur1_be21@thapar.edu

🌙

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

ClearFindActionsSaveRun

OWNER_ID	TAXI_ID
701	101
701	102
701	103
701	104
701	105
701	106
701	107
702	108
702	109
702	110
702	111
702	112

≡

Live SQL

FeedbackHelpnkaur1_be21@thapar.edu

☐🌙

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

SQL Worksheet

ClearFindActionsSaveRun

702	111
702	112

Download CSV

12 rows selected.

SSN	NAME	OWNER_ID
783	Mahesh	702

Download CSV

TSC_ID	TSC_NAME	OWNER_ID
981	ITC Car Company	701

Download CSV

≡

Live SQL

FeedbackHelpnkaur1_be21@thapar.edu

☐🌙

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

SQL Worksheet

ClearFindActionsSaveRun

TSC_ID	TSC_NAME	OWNER_ID
981	ITC Car Company	701

Download CSV

Procedure created.

Procedure created.

Trigger created.

Trigger created.

Statement processed.

BILL_NO	BILL_DATE	ADVANCE_AMT	DISCOUNT_AMT	TOTAL_AMT	USR_ID	TRIP_ID
401	21-JAN-23	200	100	1100	201	301
402	28-FEB-23	100	50	500	203	302

Live SQL

FeedbackHelpnkaurl_be21@thapar.edu

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

SQL Worksheet

ClearFindActionsSaveRun

BILL_NO	BILL_DATE	ADVANCE_AMT	DISCOUNT_AMT	TOTAL_AMT	USR_ID	TRIP_ID
401	21-JAN-23	200	100	1100	201	301
402	28-FEB-23	100	50	500	203	302
403	01-MAR-23	100	50	180	205	303
404	05-APR-23	150	30	250	207	304
405	15-MAR-23	100	50	490	209	305
406	17-SEP-22	100	50	360	201	306
407	27-DEC-22	100	50	256	202	307
408	29-NOV-22	1000	500	2500	203	308
409	31-JUL-22	300	100	790	204	309
410	30-MAY-22	200	100	650	206	310

Download CSV

Live SQL

FeedbackHelpnkaurl_be21@thapar.edu

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

SQL Worksheet

ClearFindActionsSaveRun

```
359 END;
360 /
361
362
363 --PL/SQL BLOCK
364 DECLARE
365 U NUMBER;
366 FINAL_SUM NUMBER;
367
368 BEGIN
369 U:=201; --U:=&usr_id;
370 FINAL_SUM:= TOTAL_SUM(U);
371 DBMS_OUTPUT.PUT_LINE('Total amount paid by user with user id ' || U || ' is ' || FINAL_SUM);
372
373 END;
```

410	30-MAY-22	200	100	650	206	310
-----	-----------	-----	-----	-----	-----	-----

Download CSV

10 rows selected.

Function created.

Statement processed.
Total amount paid by user with user id 201 is 1460

CONCLUSION

The discussed Database Management Project is implemented keeping in mind the travelling and fare inconveniences encountered by students, office-goers, etc. This shall aim at inter-city taxi facility at lowest prices and discounted rates. It can also used as a database management system for any local taxi company.

REFERENCES

- PL/SQL by Prateek Bhatia (YouTube)
- Database Management Systems (Book by Raghu Ramakrishnan & Johannes Gehrke)