Mini project report on

TAXI MANAGEMENT SYSTEM

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

UE20CS301 – DBMS Project

Submitted by:

Sahana Evangeline **PES2UG20CS543**

Under the guidance of

Prof. Nivedita Kasturi

Assistant Professor

Designation

PES University

AUG - DEC 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

CERTIFICATE

This is to certify that the mini project entitled

TAXI MANAGEMENT SYSYEM

is a bonafide work carried out by

SAHANA EVANGELINE **PES2UG20CS543**

In partial fulfilment for the completion of fifth semester DBMS Project (UE20CSS301) in the Program of Study - Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period AUG. 2022 – DEC. 2022. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The project has been approved as it satisfies the 5th semester academic requirements in respect of project work.

Signature
Prof. Nivedita Kasturi
Assistant Professor

DECLARATION

We hereby declare that the DBMS Project entitled **TAXI MANAGEMENT SYSTEM** has been carried out by us under the guidance of **Prof. Nivedita Kasturi, Assistant Professor** and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester AUG – DEC 2022.

SAHANA EVANGELINE PES2UG20CS543 SAHANA

ACKNOWLEDGEMENT

I would like to express my gratitude to Prof. Nivedita Kasturi, Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this UE20CS301 - DBMS Project.

I take this opportunity to thank Dr. Sandesh B J, C, Professor,ChairPerson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this DBMS Project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

This is a project on Taxi management. With emergence of technology, travel within city areas has been made really easy by huge number apps like Ola,Uber,Rapido and all these need a database management management systems maintaining their huge data like rider details,user details,vehicle details etc.

For this project, I've created a number of database entities to represent different objects in the actual world. To keep track of these entities in the Taxi management, I've constructed a database taxis.. For each entity, I have constructed a table to which the user can add tuples of data to be stored. I have implemented various queries (nested,correlated,set operations,aggregate,order) that will that will facilitate users(rider,driver,admin) in knowing vital information about their data.

TABLE OF CONTENTS

1. INTRODUCTION

In the modern era, with the help of technology,transportation companies have shifted online and hence made it so easy for passengers to easily hail rides,book rentals and also for drivers to get customers and charge fares and also get paid. These services also can be used for novel services like ambulance , food delivery etc. We have many apps like Ola,Uber,Rapido etc conquering the market with their seemless services. Hence a huge huge number of users use these apps which therefore leads to creation of thick database demanding a flawless database management system.

The DBMS will have to take of many details like of the user, driver,rentals,taxis,payments etc.
There is also a need for frequent updation of the DB. Hence we come up with a new system called TAXI MANAGEMENT SYSTEM.

This system allows us to keep track of the rides, riders, drivers , vehicles ,payments etc. The user can book a taxi or an auto or bike and the driver is mapped to the user and the vehicle which is taken for the ride. The user details like name,address,phone number is maintained . Along with that and alternate contact details of the user is also maintained for emergency cases and women safety. The driver details likename,phone,address is kept note. The vehicles registration number, type of vehicle is stored.
Billing details is stored is stored in two perspectives , one for the user as biling details and one for the admin as trip details. Both get updated hand in hand. The tables are well normalized and made query friendly.

This system is developed in a manner that it is manageable , time effective , cost effective and flexible.

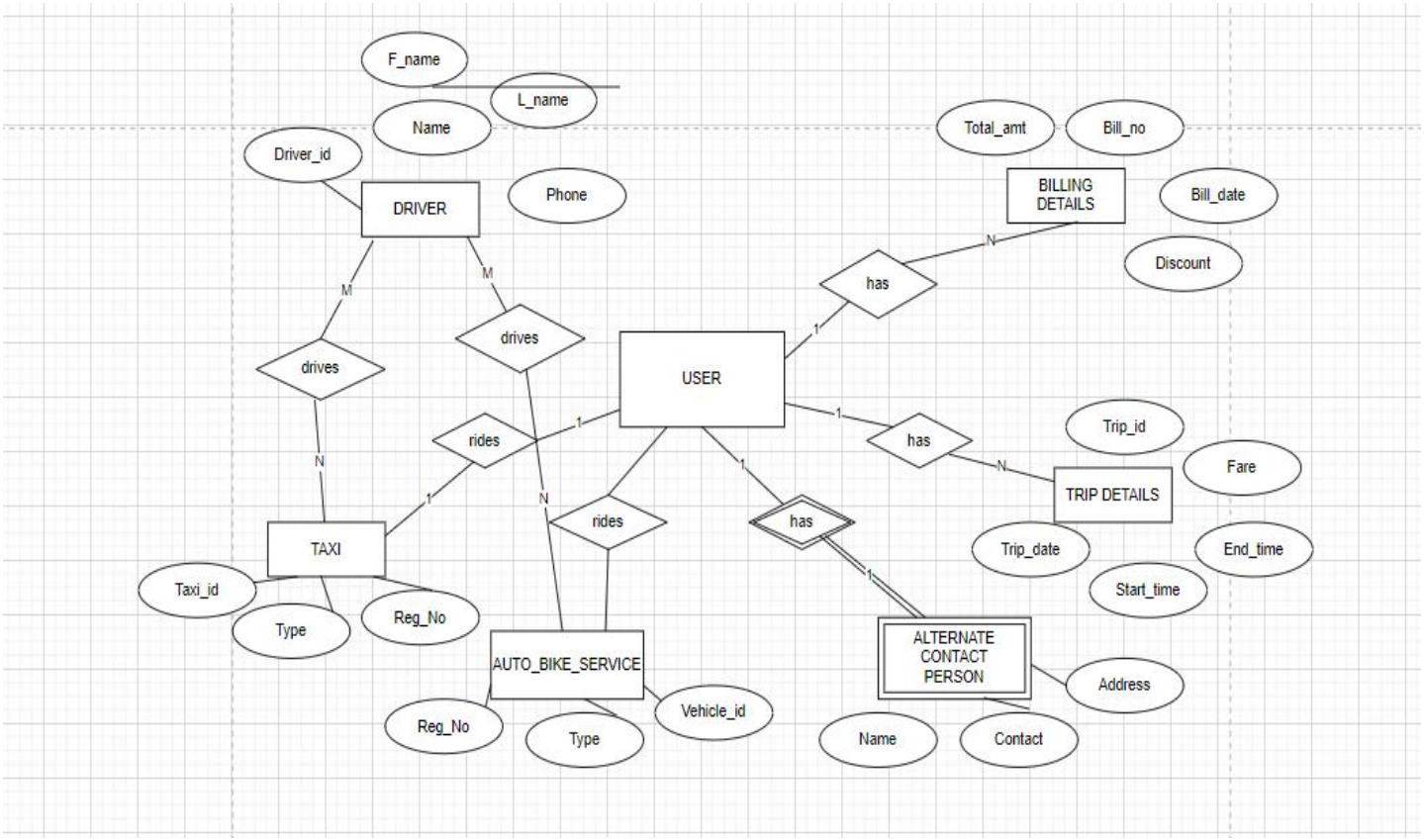
2. PROBLEM DEFINITION

Connecting the digital world with the physical one, we design a database with real world entities like taxi,user,driver,bill,trip etc.We keep track of the rides, riders, drivers , vehicles ,payments etc. The user can book a taxi or an auto or bike and the driver is mapped to the user and the vehicle which is taken for the ride.After the ride,payments are recorded.

The user details like name,address,phone number is maintained . Along with that and alternate contact details of the user is also maintained for emergency cases and women safety. The driver details likename,phone,address is kept note. The vehicles registration number, type of vehicle is stored.

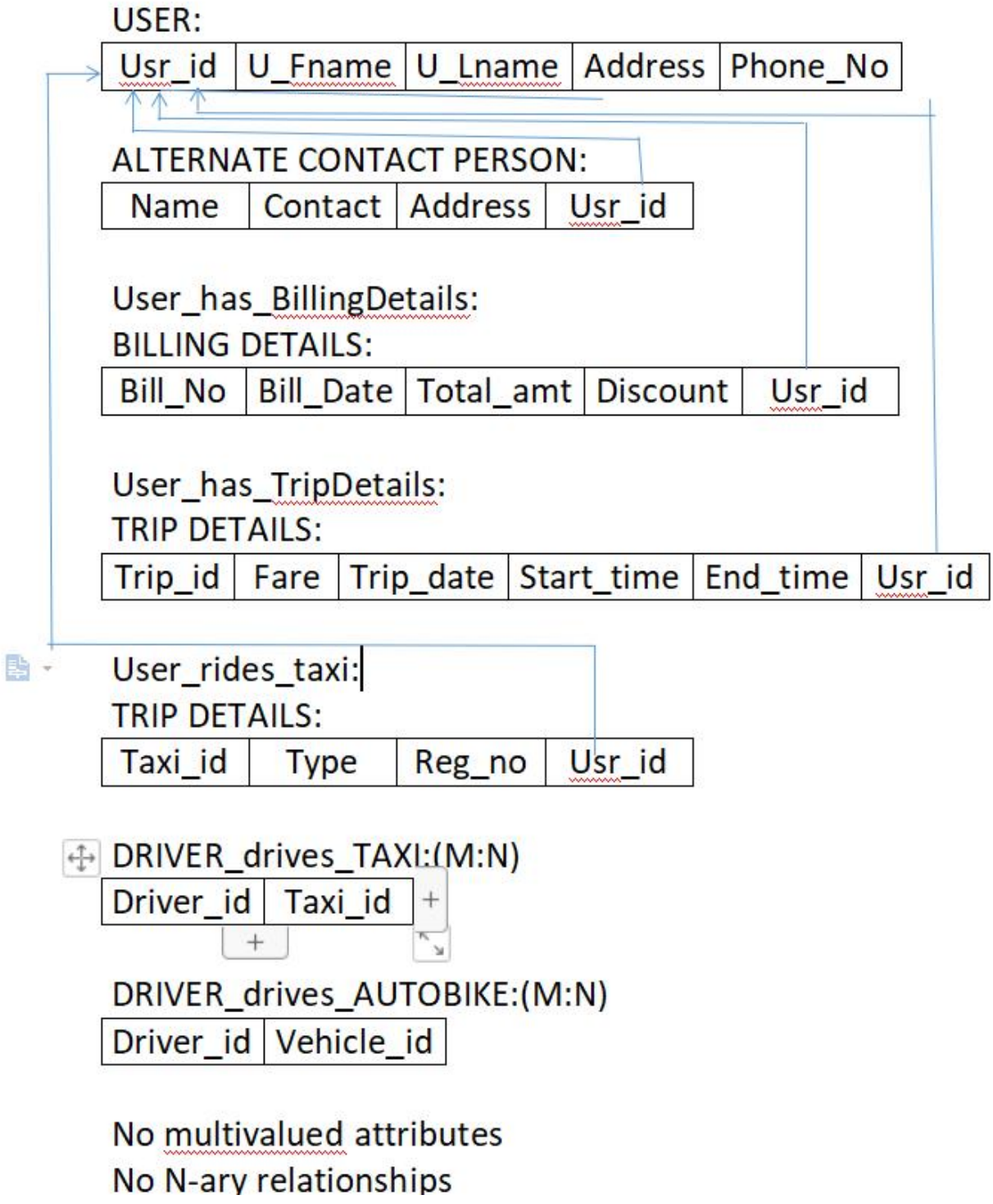
Billing details is stored is stored in two perspectives , one for the user as biling details and one for the admin as trip details. Both get updated hand in hand. The tables are well normalized and made query friendly.

3. ENTITY RELATIONSHIP DIAGRAM



4. ER TO RELATIONAL MAPPING

4.1 COMPLETE DIAGRAM OF RELATIONAL MAPPING



4.2 STEPS OF ALGORITHM FOR CHOOSEN PROBLEM

Step 1: Mapping of Regular Entity Types

Alltables(user,driver,taxi,auto_bike,bill,trip) converted to relation schema

Step 2: Mapping of Weak Entity Types

Alternate_contact_person is a weak entity. It is mapped with user with user_id as foreign key

Step 3: Mapping of Binary 1:1 Relation Types

User_rides_taxi

User_has_alternate_contact_details

Usr_I becomes foreign key in both the tables(taxi,alternate_contact_details)

Step 4: Mapping of Binary 1:N Relationship Types.

Use_has_billingDetails and User_has_tripdetails

Usr_id becomes foreign key in billing_detials and trip_details.

Step 5: Mapping of Binary M:N Relationship Types.

driver_drives_taxi and driver_drives_auto_bike

A separate table is created with the primary keys of both the tables

Step 6: Mapping of Multivalued attributes.

NO multivalued attributes

Step 7: Mapping of N-ary Relationship Types

No N-ary attributes

1) Mapping strong entity into Relation

TAXI

Taxi_id	Reg-No	Type

USER

Usr_id	U_FName	U_LName	Address	Phone_No

DRIVER

Driver_id	FName	LName	Phone_No

TRIP_DETAILS

Trip_id	Fare	Trip_date	Start time	End time

BILLING DETAILS

Bill_no	Bill_date	Total_amt	Discount

2) Mapping weak entity into Relation:

USER

Usr_id	U_FName	L_Name	Addr	Phone_No

ALTERNATE_CONTACT_PERSON

Name	Contact	Address	Usr_id

3) Mapping 1:1 relationship into relation

a) USER_rides_TAXI

USER

Usr_id	U_FName	L_Name	Addr	Phone_No

TAXI

Taxi_id	Type	Reg_No	Usr_id

b) USER - has - ALTERNATE CONTACT PERSON

USER

Usr_id	U_FName	L_Name	Addr	Phone_No

ALT_CONTACT_PERSON

Name	Contact	Addr	Usr_id

4) Mapping 1:N relationship

1) USER - has - BILLINGDETAILS 2) USER - has - TRIPDETAILS

USER

Usr_id	U_FName	U_LName	Addr	Phone_No.

BILLING DETAILS

Bill_no	Bill_date	Total_amt	Discount	Usr_id

TRIP_DETAILS

Trip_id	Fare	Trip_date	Start_time	End_time	Usr_id

5) Mapping M:N relationship:

a) DRIVER_drives_TAXI

drives

Driver_id	Taxi_id

6) Mapping multivalued attribute:

No multivalued attributes

7) There are no N-ary relationships

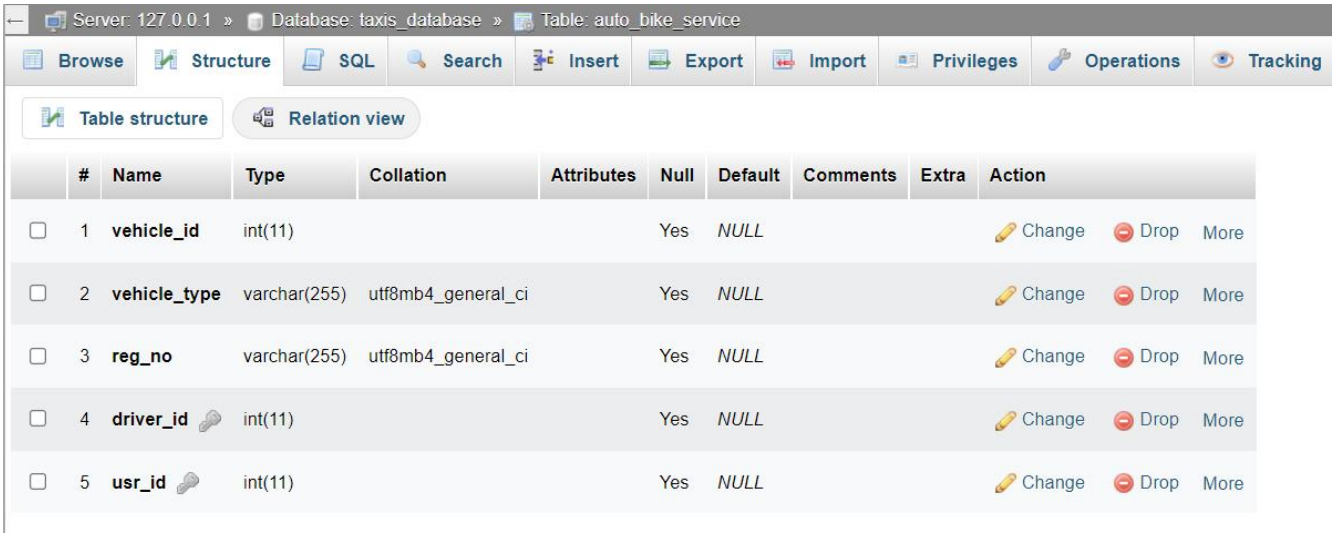
5. DDL STATEMENTS

STATEMENTS WITH SCREEN SHOTS OF THE TABLE CREATION

```
CREATE TABLE `alternate_contact_person_details` (
  `u_name` varchar(255) DEFAULT NULL,
  `u_contact` bigint(11) DEFAULT NULL,
  `u_addr` varchar(255) DEFAULT NULL,
  `usr_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

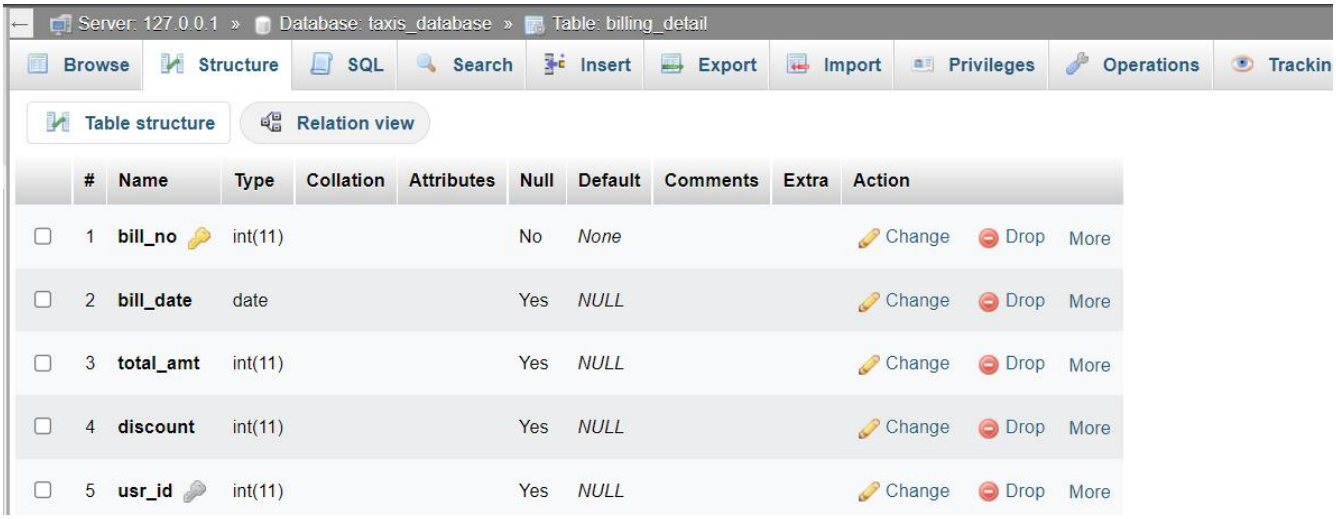
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	u_name	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	2	u_contact	bigint(11)			Yes	NULL			Change Drop More
☐	3	u_addr	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	4	usr_id	int(11)			Yes	NULL			Change Drop More

```
CREATE TABLE `auto_bike_service` (
  `vehicle_id` int(11) DEFAULT NULL,
  `vehicle_type` varchar(255) DEFAULT NULL,
  `reg_no` varchar(255) DEFAULT NULL,
  `driver_id` int(11) DEFAULT NULL,
  `usr_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```



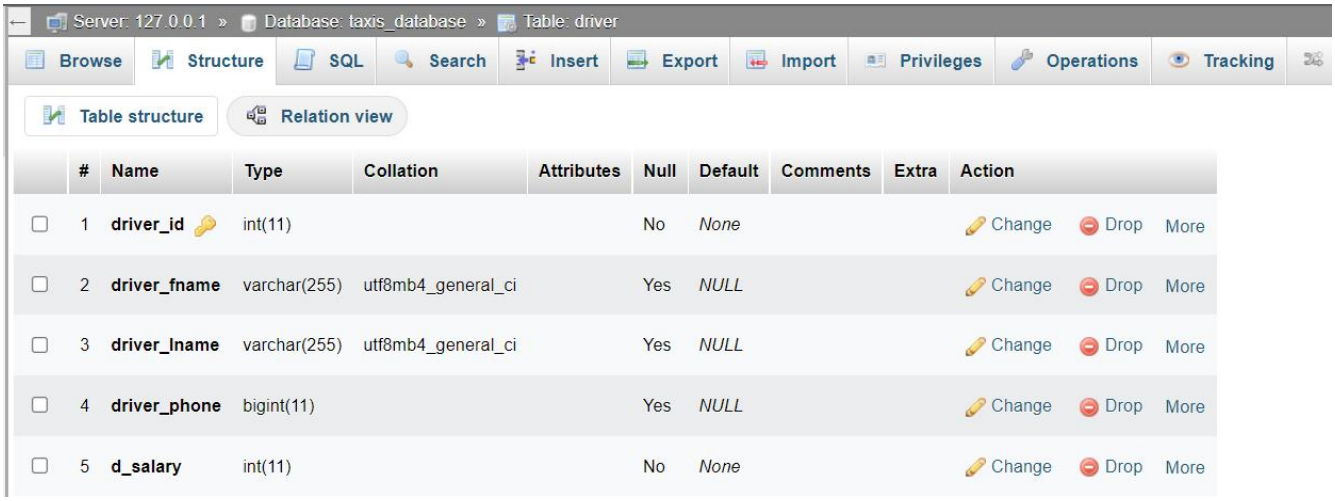
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	vehicle_id	int(11)			Yes	NULL			Change Drop More
☐	2	vehicle_type	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	3	reg_no	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	4	driver_id	int(11)			Yes	NULL			Change Drop More
☐	5	usr_id	int(11)			Yes	NULL			Change Drop More

```
CREATE TABLE `billing_detail` (
  `bill_no` int(11) NOT NULL,
  `bill_date` date DEFAULT NULL,
  `total_amt` int(11) DEFAULT NULL,
  `discount` int(11) DEFAULT NULL,
  `usr_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	bill_no	int(11)			No	None			Change Drop More
☐	2	bill_date	date			Yes	NULL			Change Drop More
☐	3	total_amt	int(11)			Yes	NULL			Change Drop More
☐	4	discount	int(11)			Yes	NULL			Change Drop More
☐	5	usr_id	int(11)			Yes	NULL			Change Drop More

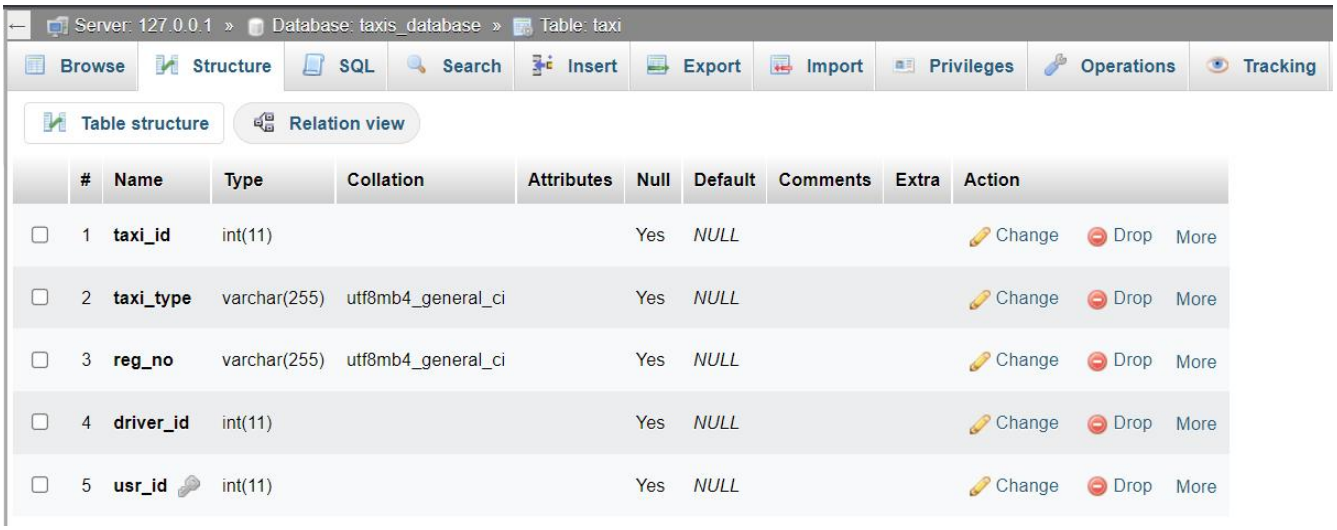
```
CREATE TABLE `driver` (
  `driver_id` int(11) NOT NULL,
  `driver_fname` varchar(255) DEFAULT NULL,
  `driver_lname` varchar(255) DEFAULT NULL,
  `driver_phone` bigint(11) DEFAULT NULL,
  `d_salary` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```




```
CREATE TABLE `driver_bike_auto` (
  `driver_id` int(11) NOT NULL,
  `vehicle_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```


```
CREATE TABLE `driver_taxi` (
  `driver_id` int(11) NOT NULL,
  `taxi_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE `taxi` (
  `taxi_id` int(11) DEFAULT NULL,
  `taxi_type` varchar(255) DEFAULT NULL,
  `reg_no` varchar(255) DEFAULT NULL,
  `driver_id` int(11) DEFAULT NULL,
  `usr_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```




```
CREATE TABLE `trip_detail` (
  `trip_id` int(11) NOT NULL,
  `fare` int(11) DEFAULT NULL,
  `trip_date` date DEFAULT NULL,
  `start_time` datetime DEFAULT NULL,
  `end_time` datetime DEFAULT NULL,
  `usr_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Server: 127.0.0.1 » Database: taxis_database » Table: trip_detail

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	trip_id	int(11)			No	None			Change Drop More
2	fare	int(11)			Yes	NULL			Change Drop More
3	trip_date	date			Yes	NULL			Change Drop More
4	start_time	datetime			Yes	NULL			Change Drop More
5	end_time	datetime			Yes	NULL			Change Drop More
6	usr_id	int(11)			Yes	NULL			Change Drop More

```
CREATE TABLE `users` (
  `usr_id` int(11) NOT NULL,
  `u_fname` varchar(255) DEFAULT NULL,
  `u_lname` varchar(255) DEFAULT NULL,
  `u_phone` bigint(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Server: 127.0.0.1 » Database: taxis_database » Table: users

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	usr_id	int(11)			No	None			Change Drop More
2	u_fname	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	u_lname	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	u_phone	bigint(11)			Yes	NULL			Change Drop More

6. DML STATEMENTS

STATEMENTS WITH SCREEN SHOTS OF THE TABLE WITH INSERTED VALUES

```
INSERT INTO `alternate_contact_person_details` (`u_name`, `u_contact`, `u_addr`, `usr_id`) VALUES
('Harish', 7894561234, 'Koramangala', 1),
('Ravi', 8456123587, 'Jayanagar', 2),
('Tanya', 7596412389, 'Basavanagudi', 3),
('Usha', 7459612358, 'Yeshwantpur', 4),
('Ira', 7456981236, 'JP Nagar', 5),
('Paru', 9564812375, 'Bommanhalli', 6),
('Jay', 9541237845, 'Rajajinagar', 7),
('Nitish', 8451276351, 'Adugodi', 8),
('Banu', 7845692314, 'RR Nagar', 9),
('Kavita', 8456283971, 'Viveknagar', 10);
```

```
SELECT * FROM `alternate_contact_person_details`
```

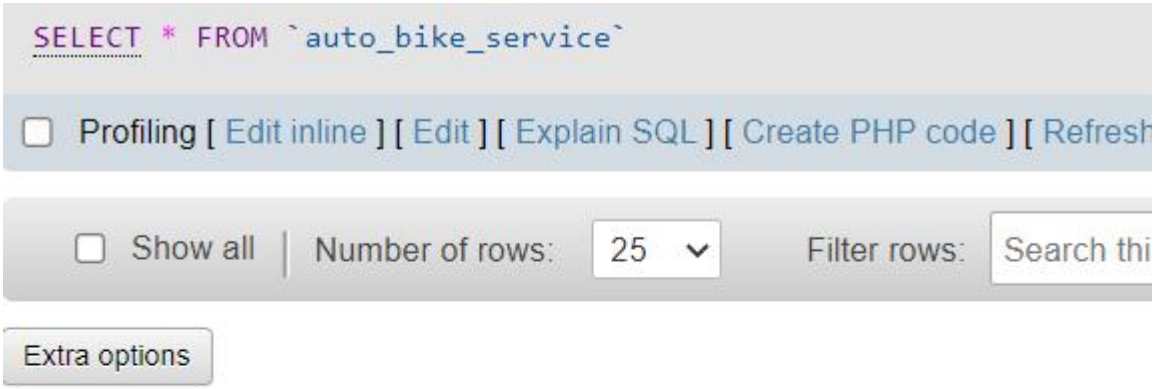
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code]

Show all | Number of rows: 25 Filter rows:

Extra options

u_name	u_contact	u_addr	usr_id
Harish	7894561234	Koramangala	1
Ravi	8456123587	Jayanagar	2
Tanya	7596412389	Basavanagudi	3
Usha	7459612358	Yeshwantpur	4
Ira	7456981236	JP Nagar	5
Paru	9564812375	Bommanhalli	6
Jay	9541237845	Rajajinagar	7
Nitish	8451276351	Adugodi	8
Banu	7845692314	RR Nagar	9
Kavita	8456283971	Viveknagar	10

```
INSERT INTO `auto_bike_service` (`vehicle_id`, `vehicle_type`, `reg_no`, `driver_id`, `usr_id`) VALUES
(201, 'bike_A', 'KA512345', 3, 6),
(202, 'bike_B', 'KA512765', 2, 9),
(203, 'auto', 'KA285362', 4, 5),
(204, 'auto', 'KA561234', 2, 9),
(205, 'bike_C', 'KA457899', 1, 2);
```



vehicle_id	vehicle_type	reg_no	driver_id	usr_id
201	bike_A	KA512345	3	6
202	bike_B	KA512765	2	9
203	auto	KA285362	4	5
204	auto	KA561234	2	9
205	bike_C	KA457899	1	2

```
INSERT INTO `billing_detail` (`bill_no`, `bill_date`, `total_amt`, `discount`, `usr_id`) VALUES
(2022101, '2022-10-09', 203, 32, 1),
(2022102, '2022-10-18', 130, 23, 3),
(2022103, '2022-10-15', 560, 34, 2),
(2022105, '2022-07-10', 193, 5, 5),
(2022109, '2022-08-09', 90, 22, 4),
(2022110, '2022-10-17', 150, 9, 7),
(2022117, '2022-10-09', 293, 28, 6),
(2022118, '2022-10-09', 200, 33, 8);
```

```
SELECT * FROM `billing_detail`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort

Extra options

	bill_no	bill_date	total_amt	discount	usr_id
Edit Copy Delete	202201	2022-10-09	50	48	1
Edit Copy Delete	202202	2022-10-18	100	95	2
Edit Copy Delete	202203	2022-11-01	300	285	3

```
INSERT INTO `driver` (`driver_id`, `driver_fname`, `driver_lname`, `driver_phone`, `d_salary`) VALUES
(1, 'Shiv', 'Kutty', 7528459612, 10000),
(2, 'Ram', 'Jose', 7524459612, 12000),
(3, 'Jeev', 'Jobs', 9875392223, 11000),
(4, 'Steve', 'Jobs', 9875342223, 14000),
(5, 'Bill', 'Gates', 9875342213, 15000);
```

```
SELECT * FROM `driver`
```

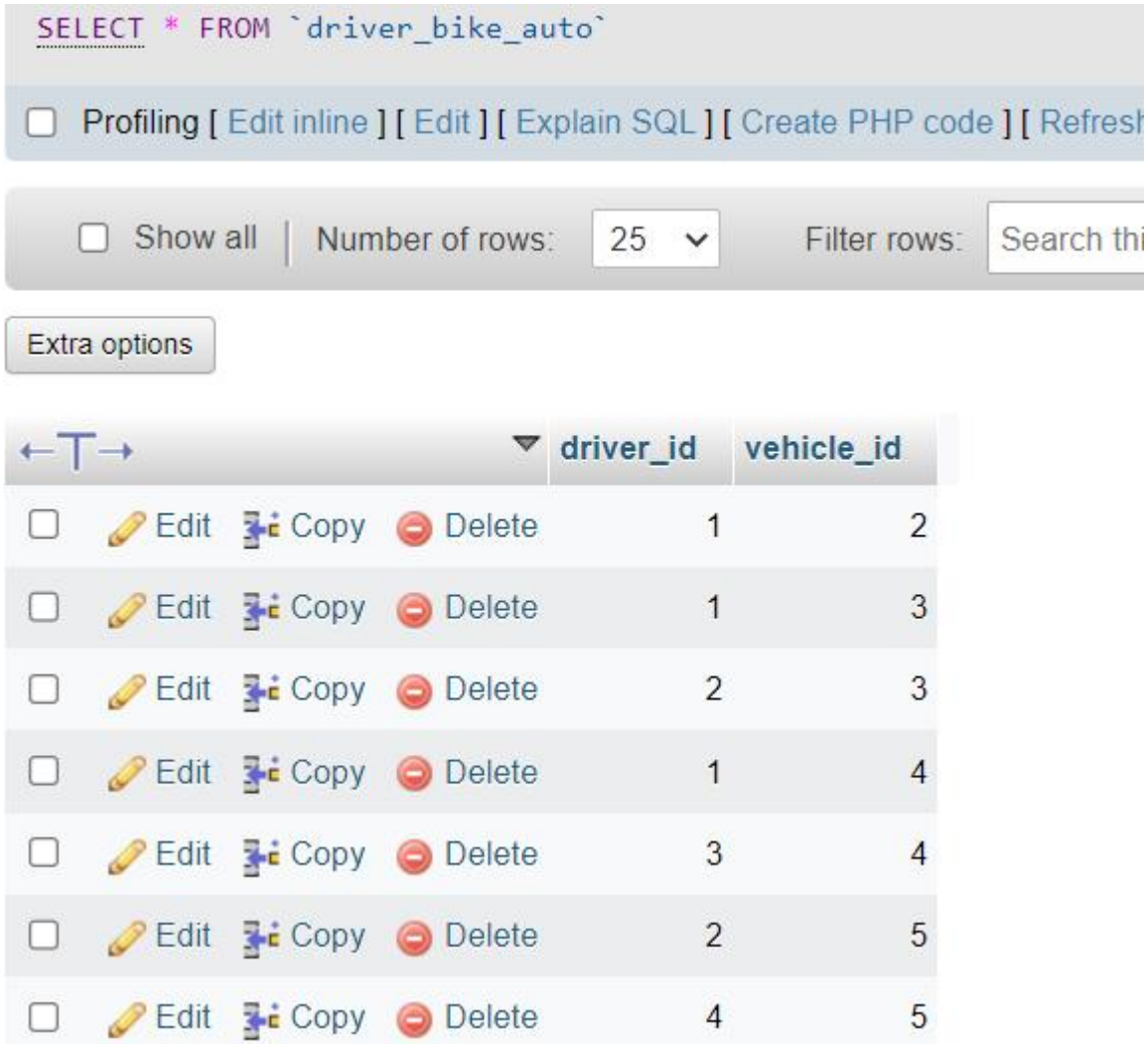
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

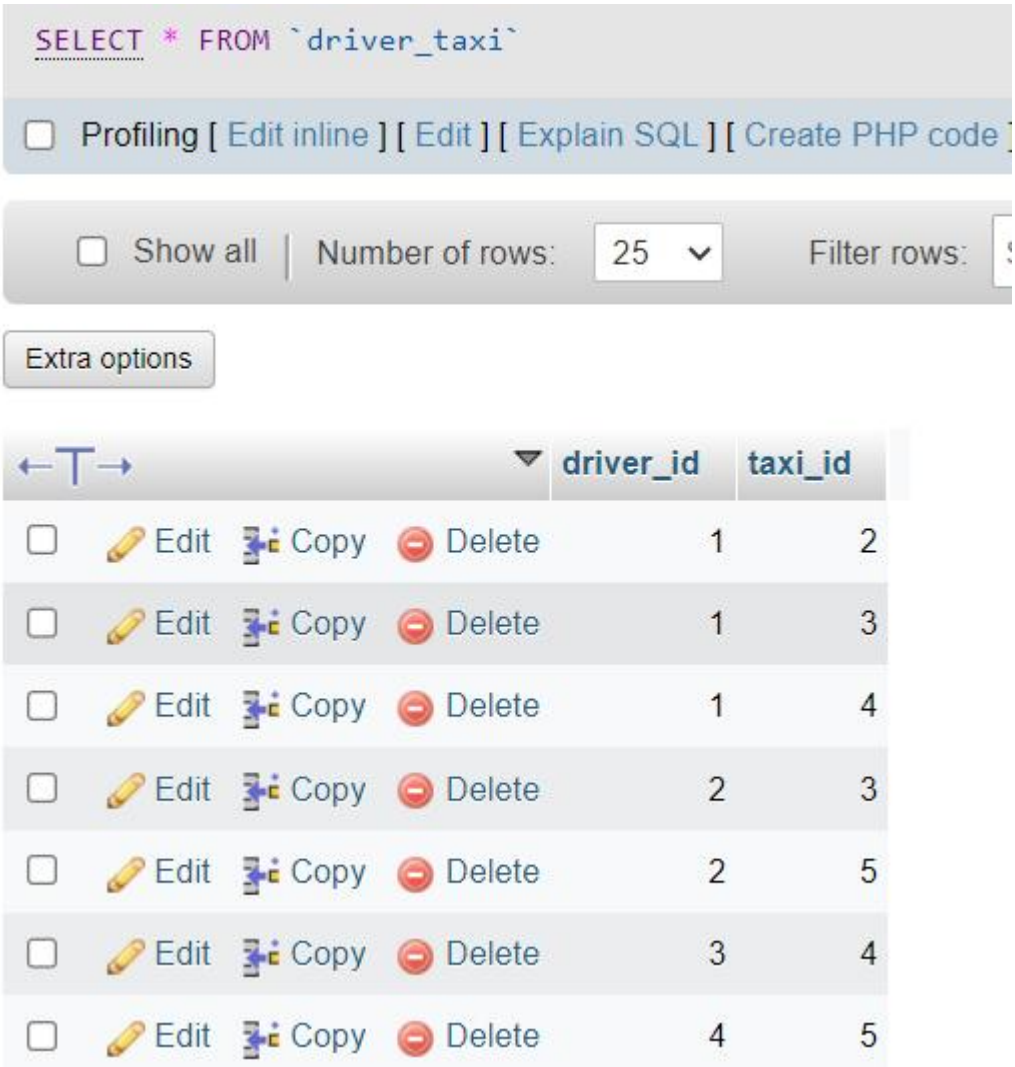
Extra options

	driver_id	driver_fname	driver_lname	driver_phone	d_salary
Edit Copy Delete	1	Shiv	Kutty	7528459612	10000
Edit Copy Delete	2	Ram	Jose	7524459612	12000
Edit Copy Delete	3	Jeev	Jobs	9875392223	11000
Edit Copy Delete	4	Steve	Jobs	9875342223	14000
Edit Copy Delete	5	Bill	Gates	9875342213	15000

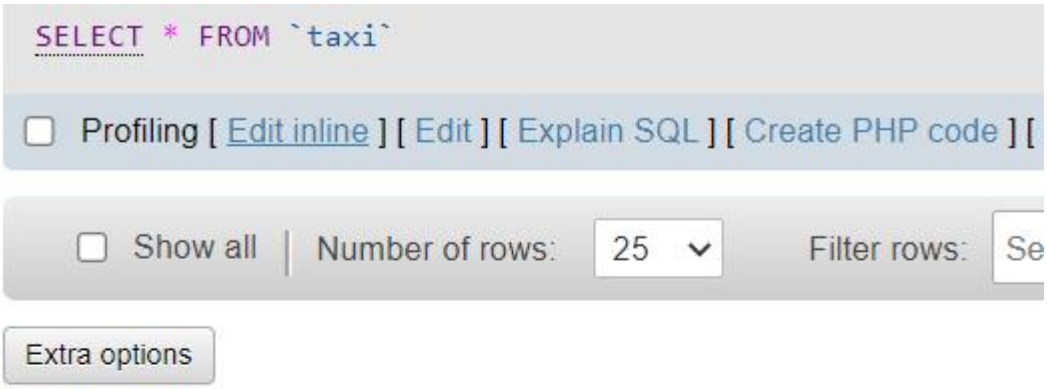
```
INSERT INTO `driver_bike_auto` (`driver_id`, `vehicle_id`) VALUES
(1, 2),
(1, 3),
(2, 3),
(1, 4),
(3, 4),
(2, 5),
(4, 5);
```



```
INSERT INTO `driver_taxi` (`driver_id`, `taxi_id`) VALUES
(1, 2),
(1, 3),
(1, 4),
(2, 3),
(2, 5),
(3, 4),
(4, 5);
```

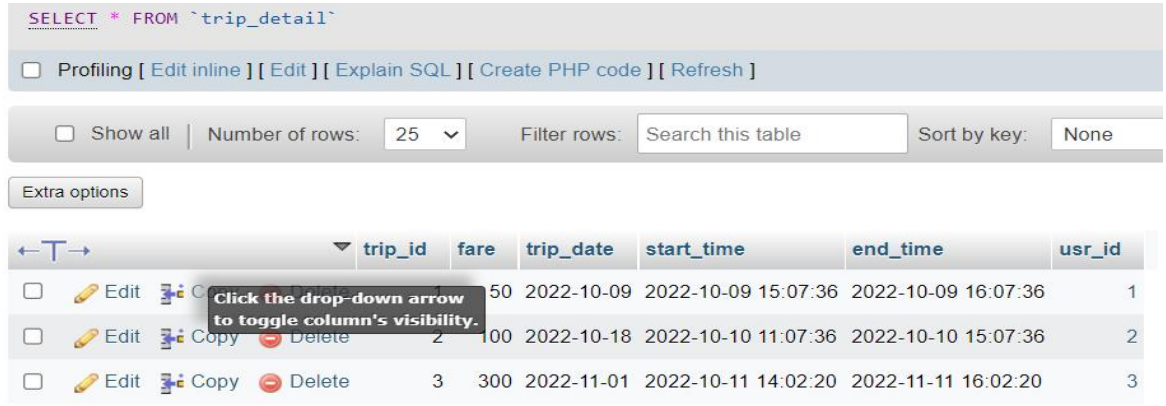


```
INSERT INTO `taxi` (`taxi_id`, `taxi_type`, `reg_no`, `driver_id`, `usr_id`) VALUES
(101, 'premium delux', 'KA011234', 2, 3),
(102, 'XL', 'KA023454', 1, 2),
(103, 'delux', 'KA021234', 3, 5),
(104, 'go sedan', 'KA011233', 1, 4),
(101, 'premium delux', 'KA011234', 5, 6),
(104, 'go sedan', 'KA011233', 2, 7),
(102, 'XL', 'KA023454', 2, 8);
```



taxi_id	taxi_type	reg_no	driver_id	usr_id
101	premium delux	KA011234	2	3
102	XL	KA023454	1	2
103	delux	KA021234	3	5
104	go sedan	KA011233	1	4
101	premium delux	KA011234	5	6
104	go sedan	KA011233	2	7
102	XL	KA023454	2	8

```
INSERT INTO `trip_detail` (`trip_id`, `fare`, `trip_date`, `start_time`, `end_time`, `usr_id`) VALUES
(1, 50, '2022-10-09', '2022-10-09 15:07:36', '2022-10-09 16:07:36', 1),
(2, 100, '2022-10-18', '2022-10-18 11:07:36', '2022-10-18 15:07:36', 2),
(3, 310, '2022-11-01', '2022-11-01 14:02:20', '2022-11-01 16:02:20', 5),
(4, 290, '2022-11-02', '2022-11-02 09:02:20', '2022-11-02 11:02:20', 7),
(5, 100, '2022-11-03', '2022-11-03 14:02:20', '2022-11-01 16:02:20', 5),
(6, 230, '2022-11-03', '2022-11-03 09:02:20', '2022-11-02 11:02:20', 7),
(7, 130, '2022-11-04', '2022-11-04 14:02:20', '2022-11-01 16:02:20', 2),
(8, 250, '2022-11-05', '2022-11-05 09:02:20', '2022-11-02 11:02:20', 7);
```



	trip_id	fare	trip_date	start_time	end_time	usr_id
Edit Copy Delete	1	50	2022-10-09	2022-10-09 15:07:36	2022-10-09 16:07:36	1
Edit Copy Delete	2	100	2022-10-18	2022-10-10 11:07:36	2022-10-10 15:07:36	2
Edit Copy Delete	3	300	2022-11-01	2022-10-11 14:02:20	2022-11-11 16:02:20	3

```
INSERT INTO `users` (`usr_id`, `u_fname`, `u_lname`, `u_phone`) VALUES
(1, 'Aksh', 'Mallo', 7894662336),
(2, 'Kevin', 'Smith', 9784512533),
(3, 'Harry', 'Potter', 7894662336),
(4, 'Ron', 'Weasley', 9784512533),
(5, 'fred', 'george', 7675344223),
(6, 'hermione', 'granger', 7675344223),
(7, 'arthur', 'weasley', 7675344223),
(8, 'deeps', 'geroge', 7675344223),
(9, 'Sahana', 'Evan', 7975391114),
(10, 'Raaghu', 'AK', 87945612345);
```

	usr_id	u_fname	u_lname	u_phone
Edit Copy Delete	1	Aksh	Mallo	7894662336
Edit Copy Delete	2	Kevin	Smith	9784512533
Edit Copy Delete	3	Harry	Potter	7894662336
Edit Copy Delete	4	Ron	Weasley	9784512533
Edit Copy Delete	5	fred	george	7675344223
Edit Copy Delete	6	hermione	granger	7675344223
Edit Copy Delete	7	arthur	weasley	7675344223
Edit Copy Delete	8	deeps	geroge	7675344223
Edit Copy Delete	9	Sahana	Evan	7975391114
Edit Copy Delete	10	Raaghu	AK	87945612345

7. QUERIES

1.**NESTED QUERY**

Q. Select the first and last name of users who travelled on date 09/10/2022

```
SELECT u_fname,u_lname FROM users WHERE usr_id IN (SELECT usr_id FROM billing_detail WHERE bill_date='2022-10-09');
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	u_fname	u_lname
Edit Copy Delete	Aksh	Mallo
Edit Copy Delete	Ron	Weasley
Edit Copy Delete	fred	george

---

2. CORRELATED QUERY

Q.Finds drivers whose salary is greater than the average salary of all other drivers to compare their work and feedback with those paid lesser

```
SELECT driver_fname, d_salary FROM driver d WHERE d_salary > (SELECT AVG(d_salary) FROM driver );
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	driver_fname	d_salary
Edit Copy Delete	Steve	14000
Edit Copy Delete	Bill	15000

---

3. SET OPERATION UNION

Q. Join the taxi and auto and bike service tables into one vehicle table for better reference.

```
SELECT * FROM auto_bike_service UNION SELECT * FROM taxi;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Ref

Show all | Number of rows: 25 Filter rows: Searcl

Extra options

vehicle_id	vehicle_type	reg_no	driver_id	usr_id
201	bike_A	KA512345	3	6
202	bike_B	KA512765	2	9
203	auto	KA285362	4	5
204	auto	KA561234	2	9
205	bike_C	KA457899	1	2
101	premium delux	KA011234	2	3
102	XL	KA023454	1	2
103	delux	KA021234	3	5
104	go sedan	KA011233	1	4
101	premium delux	KA011234	5	6
104	go sedan	KA011233	2	7
102	XL	KA023454	2	8

---

4. SET OPERATION

Q.Check how many drivers know to drive taxi and auto or bike

```
SELECT driver_fname FROM driver WHERE driver_id in ( SELECT driver_id FROM taxi Intersect SELECT driver_id FROM auto_bike_service);
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

←T→	driver_fname
Edit Copy Delete	Shiv
Edit Copy Delete	Ram
Edit Copy Delete	Jeev

5.Two queries for Aggregate functions with group by clause, use Having Clause.

Q. Group the users based on their usage of the company's services and display those users who have spent more than 500 rupees our on taxi services

```
SELECT usr_id ,SUM(fare),COUNT(usr_id) FROM trip_detail td GROUP BY td.usr_id HAVING SUM(fare)>=500;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Extra options

usr_id	SUM(fare)	COUNT(usr_id)
5	1000	1

6.Two queries for Aggregate functions with group by clause, use Having Clause.////

Q. Group based on the billing date and find out on which day how much collection has occurred and display those dates where collection is more than 200

```
SELECT bill_date,usr_id ,SUM(total_amt),COUNT(usr_id) FROM billing_detail GROUP BY bill_date HAVING SUM(total_amt)>=200;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

bill_date	usr_id	SUM(total_amt)	COUNT(usr_id)
2022-10-09	1	1100	3
2022-11-01	3	610	2
2022-11-02	7	290	1

7.ORDER BY CLAUSE

Q. Order the billing table based on bill_date starting from most recent dates.

```
SELECT bill_no, bill_date,u_fname,u_phone,total_amt,discount FROM billing_detail b,users u WHERE u.usr_id=b.usr_id ORDER BY bill_date desc;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

bill_no	bill_date 1	u_fname	u_phone	total_amt	discount
202204	2022-11-02	arthur	7675344223	290	276
202203	2022-11-01	Harry	7894662336	300	285
202207	2022-11-01	hermione	7675344223	310	295
202202	2022-10-18	Kevin	9784512533	100	95
202201	2022-10-09	Aksh	7894662336	50	48
202205	2022-10-09	Ron	9784512533	50	48
202206	2022-10-09	fred	7675344223	1000	950

8. ORDER BY CLAUSE

Q.Order the driver table based on salary in descending order.

```
SELECT driver_fname,driver_lname,d_salary FROM driver ORDER BY d_salary DESC;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

	driver_fname	driver_lname	d_salary 1
Edit Copy Delete	Bill	Gates	15000
Edit Copy Delete	Steve	Jobs	14000
Edit Copy Delete	Ram	Jose	12000
Edit Copy Delete	Jeev	Jobs	11000
Edit Copy Delete	Shiv	Kutty	10000

8. STORED PROCEDURES, FUCNTIONS AND TRIGGERS

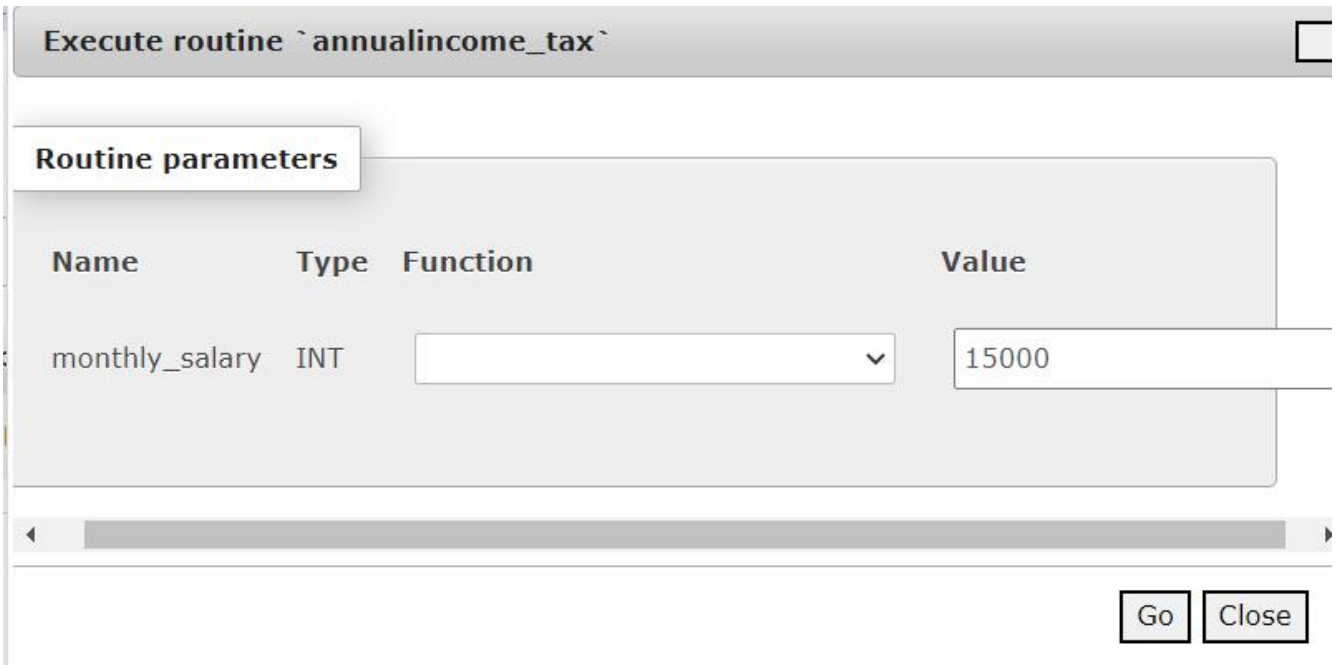
8.1 STORED PROCEDURES OR FUNCTIONS

FUNCTIONS:

A function to check if a driver is taxable of not depending oh his or her salary. Calculate annual salary, if it is greater than 50000 then tax them.

```
CREATE DEFINER=`root`@`localhost` FUNCTION
`annualincome_tax`(`monthly_salary` INT)
RETURNS VARCHAR(20)
 CHARSET utf8mb4 DETERMINISTIC
CONTAINS SQL SQL SECURITY DEFINER
 BEGIN IF monthly_salary*12 > 50000
THEN RETURN ("Taxable");
ELSE RETURN ("Non Taxable");
 END IF;
 END
```

	Name	Type	Returns				
☐	annualincome_tax	FUNCTION	varchar(20)	Edit	Execute	Export	Drop




```
SET @p0='15000'; SELECT `annualincome_tax`(@p0) AS `annualincome_tax`;
```

Execution results of routine `annualincome_tax`

annualincome_tax
Taxable

8.2 TRIGGERS

Write a trigger to insert into billing table everytime the trip_details table has an entry with only the bill_date,total_amt. Aslo calculate 5% discount and show to the user's billing details . Map the user_id also.

```
CREATE DEFINER=`root`@`localhost` TRIGGER `discount`
AFTER INSERT
ON `trip_detail`
FOR EACH ROW
INSERT INTO
billing_detail(`bill_no`,`bill_date`,`total_amt`,`discount`,`usr_id`) VALUES (2022*100+new.trip_id,new.trip_date,new.fare,ne
w.fare-new.fare*0.05,new.usr_id)
```

CREATE A TRIGGER WHICH INSERTS AND UPDATES THE BILLING TABLE EVERYTIME TRIP IS CONFIRMED. The billing table should have unique bill number,bill date , fare , total amt after discount.

	bill_no	bill_date	total_amt	discount	usr_id
Edit Copy Delete	202201	2022-10-09	50	48	1
Edit Copy Delete	202202	2022-10-18	100	95	2
Edit Copy Delete	202203	2022-11-01	300	285	3
Edit Copy Delete	202204	2022-11-02	290	276	7
Edit Copy Delete	202205	2022-10-09	50	48	4
Edit Copy Delete	202206	2022-10-09	1000	950	5
Edit Copy Delete	202207	2022-11-01	310	295	6
Edit Copy Delete	202208	2022-10-09	50	48	8
Edit Copy Delete	202209	2022-10-18	100	95	9
Edit Copy Delete	202210	2022-11-01	310	295	10

	Name	Time	Event			
☐	discount	AFTER	INSERT	Edit	Export	Drop

8. FRONT END DEVELOPMENT

connect.php

```
<?php
    $sname = "localhost";
    $uname = "root";
    $password = "";
    $db_name = "taxis_database";
$conn = mysqli_connect($sname, $uname, $password, $db_name);
if (!$conn) {
    echo "Connection failed!";
    exit();
}
?>
```

index_taxi.php

```
<!--external css-->
<link rel="stylesheet" href="style.css">
<img src="car1.jpg" alt="go die" >
<div class="col-md-2"></div>
            <div class="col-md-8">
            <h1 class="admin_page">TAXI MANAGEMENT SYSTEM</h1>

                  <?php
                    require_once('connect.php'); ?>
                    <h3 style="color:pink">Enter the amount to display the dates  have crossed that
collection</h3>
```

```
                <form method="post">

                <label for="usr_ip">AMOUNT : </label>
                <input type="text" name="usr_ip" id="usr_ip">
                <button type="submit" name="submit">Submit</button>
                <?php
```

```
                if (isset($_POST['submit'])) {?>

                    <table class="table table-bordered table-hover">
            <thead>
                <tr>
                    <th>BILL DATE</th>

                    <th>TOTAL</th>
                    <th>NUMBER OF USERS</th>



                </tr>
            </thead>
            <tbody>
            <?php
                    $ip = $_POST['usr_ip'];
            $query="SELECT  bill_date,usr_id ,SUM(total_amt) AS total,COUNT(usr_id) as no_of_users
               FROM     billing_detail
               GROUP BY bill_date
               HAVING       SUM(total_amt)>='$ip'";

                if($result = mysqli_query($conn, $query)){
                    if($result->num_rows > 0){
                        while($row = $result->fetch_object()){


                ?>
                    <tr>

                            <td><?php echo $row->bill_date; ?></td>

                            <td><?php echo $row->total?></td>
                            <td><?php echo $row->no_of_users?></td>

                    </tr>
```

```
             <?php
                        }
```

```
            }
        }
```

```
                      }


                ?>
                  </form>

              </tbody>
          </table>
          <div class="col-md-2"></div>
```

Styles.css

```
table, th, td ,tr{

    padding: 25px;
    text-align: left;
    color: palevioletred;
    border-bottom: 1px solid #ddd;
    border-color:peachpuff;
    font-size: large;
  }

   h1 {
  font-size: 30px;
  font-weight: 100;
  background-image: linear-gradient(45deg, #553c9a, #ee4b2b);
  color: transparent;
  background-clip: text;
  -webkit-background-clip: text;
  text-align: center;
  background-color:aqua;
  background-color: antiquewhite;
}
img{
    float: right;
    padding-right:15cm ;
    padding-top: 1cm;

}
```

```
input[type=text] {
    border: none;
    border-bottom: 2px solid plum;
  }
```

```
  label{
    color: plum;
    font:italic;
  }
  button{
    border: #553c9a;
    background-color: pink;
    border-radius: 40%;
    color:#553c9a;
  }
```