

DBMS – Mini Project

BUS RESERVATION SYSTEM

SUBMITTED BY-

NAME: DASA A DEEPESH

SRN: PES2UG20CS102

V SEMESTER SECTION B

Short Description and Scope of the Project

This database project helps in booking bus tickets by the new user or a registered user.

It stores all the information of the user and details of the booking in the database.

It has 8 entities which covers all the information needed for the user and for the Admin

It also has special features such as functions and triggers which helps in fast retrieval and easy maintenance of the project

With the help of our project a user can book or cancel tickets for a particular ride

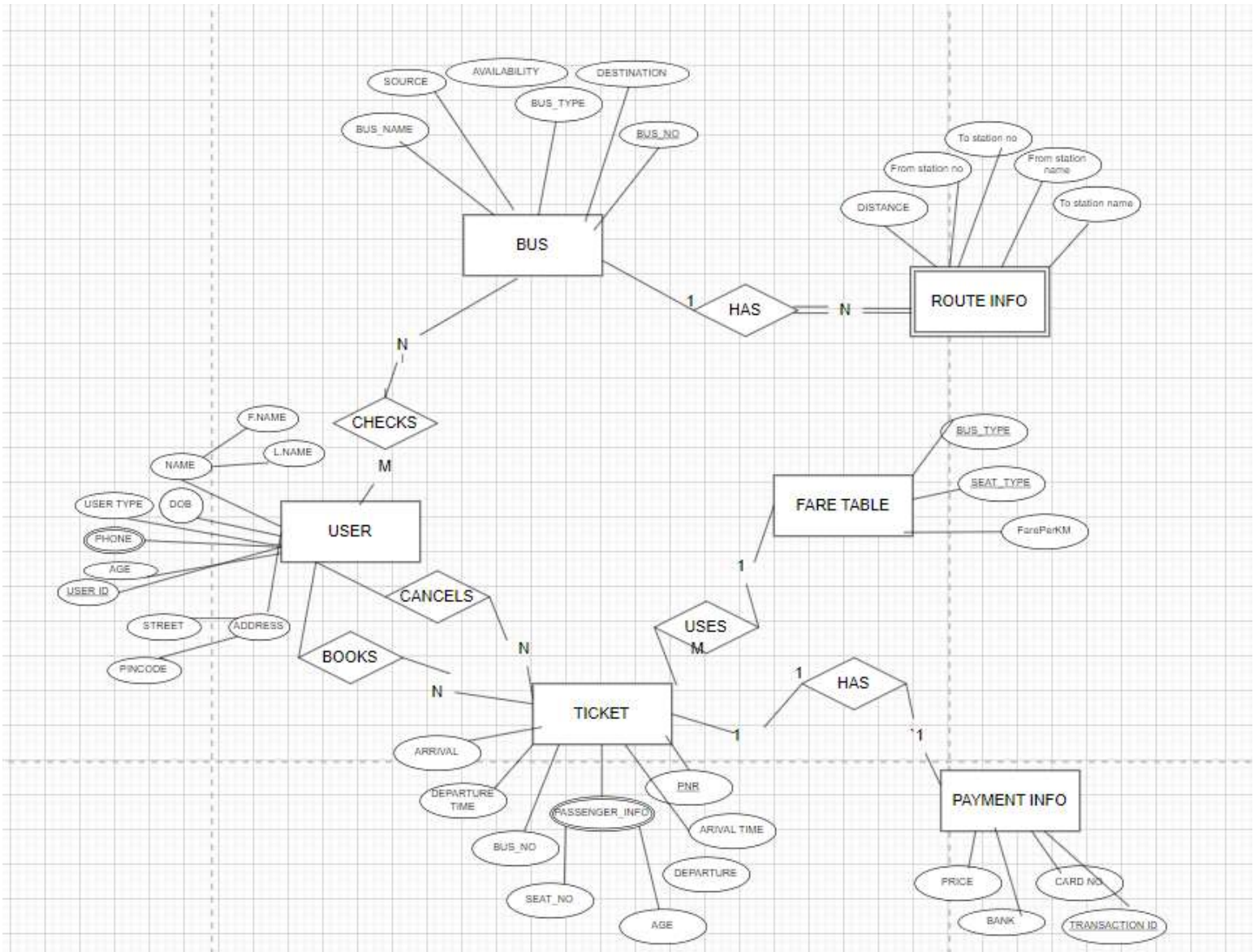
Scope of the project:

It helps the user to know about the bus timings and details by just looking at the database and can make plans.

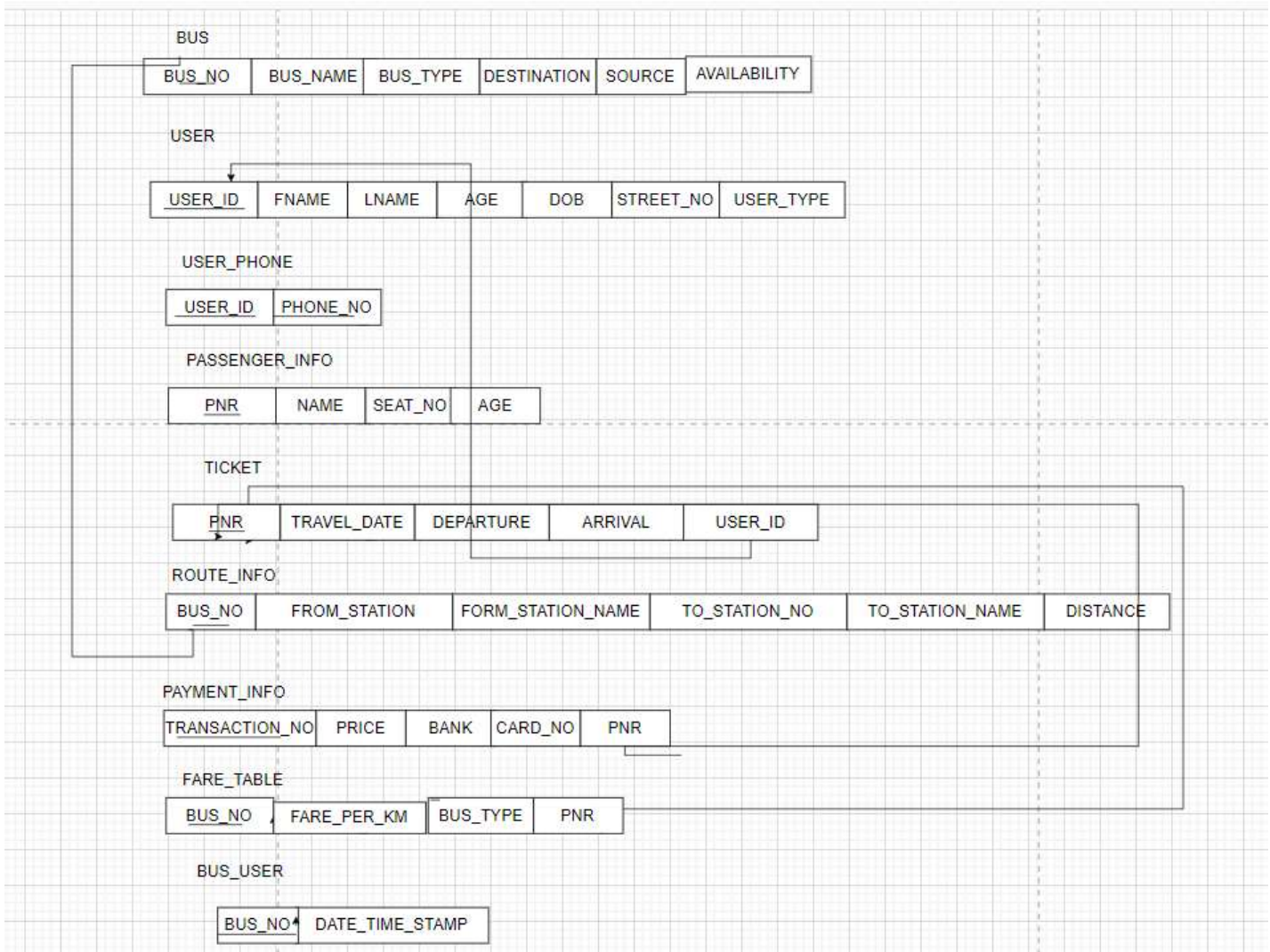
This helps and reduces the time the user spends to bus reservation.

The frontend of this project helps us to create new user to the database or login to the existing user.

ER Diagram



Relational Schema



DDL Statements - Building The Database

CREATE STATEMENTS:

```
--table1
CREATE TABLE bus (
  bus_no INT ,
  bus_name VARCHAR(255),
  bus_type VARCHAR(100) NOT NULL,
  arrival VARCHAR(50),
  destination VARCHAR(50),
  availability VARCHAR(100),
  PRIMARY KEY(bus_no),
  unique(bus_name)
);

DESCRIBE bus;
```

```
--table3
CREATE TABLE USER(
  user_id varchar(20),
  user_type varchar(100),
  Fname varchar(255),
  Lname varchar(255),
  age int(11),
  street_no varchar(100),
  pincode int(11),
  DOB date,
  primary key(user_id)
);

describe user;
```

```
--table2
CREATE TABLE route_info (
  from_no INT(11),
  from_name varchar(100),
  to_no int(11) ,
  to_name varchar(100),
  dist int(11),
  bus_no int,
  primary key(from_no,to_no),
  foreign key(bus_no) REFERENCES bus(bus_no) on delete set NULL
);

describe route info;
```

--table4

```
CREATE TABLE user_bus(  
    user_id varchar(20),  
    bus_no int(29),  
    date_time_stamp datetime,  
    primary key(date_time_stamp),  
    foreign key(user_id) references user(user_id) on delete set NULL,  
    foreign key(bus_no) references bus(bus_no) on delete set NULL  
);  
describe user_bus;
```

--table5

```
CREATE TABLE user_phone(  
    user_id varchar(20),  
    phone_no bigint(20),  
    primary key(phone_no),  
    foreign key(user_id) references user(user_id) on delete set NULL  
);  
describe user_train;
```

--table6

```
CREATE TABLE payment_info(  
    transaction_id int(11),  
    bank varchar(20),  
    card_no bigint(20),  
    price double,  
    PNR varchar(20),  
    primary key(transaction_id),  
    foreign key(PNR) references ticket(PNR) on delete set NULL  
);
```



```
--table7
create table ticket(
  PNR varchar(20),
  bus_no int,
  travel_date date,
  departure varchar(50),
  arrival varchar(50),
  departure_time time,
  arrival_time time,
  user_id varchar(20),
  bus_type varchar(100),
  primary key(PNR),
  foreign key(user_id) references user(user_id) on delete set NULL,
  foreign key(bus_no) references bus(bus_no) on delete set NULL
);
describe ticket;
```

```
--table8
CREATE TABLE ticket_passenger(
  seat_no varchar(20),
  name varchar(255),
  age int(11),
  PNR varchar(20),
  primary key(seat_no),
  foreign key(PNR) references ticket(PNR) on delete set NULL
);
describe ticket_passenger;
```

```
--table9

CREATE TABLE fare_table(
  bus_type varchar(100),
  fareperkm float,
  primary key(bus_type)
);
describe fare_table;
```

Populating the Database

✓ Showing rows 0 - 5 (6 total, Query took 0.0004 seconds.)

```
SELECT * FROM `bus`
```

☐ 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

			bus_no	bus_name	bus_Type	arrival	destination	availability
<input type="checkbox"/>	Edit	Copy	Delete	25260	Chennai Mail	Mail	Mangaluru Chennai	Yes
<input type="checkbox"/>	Edit	Copy	Delete	25261	Managaluru Mail	Mail	Chennai Mangaluru	Yes
<input type="checkbox"/>	Edit	Copy	Delete	58450	MAN-BEN Express	Fast	Mangaluru Bengaluru	yes
<input type="checkbox"/>	Edit	Copy	Delete	58451	BEN-MAN Express	Fast	Bengaluru Mangaluru	yes
<input type="checkbox"/>	Edit	Copy	Delete	62620	CHE-BEN Shatabdi	Superfast	Chennai Bengaluru	No
<input type="checkbox"/>	Edit	Copy	Delete	62621	BEN-CHE Shatabdi	Superfast	Bengaluru Chennai	yes

↑ ☐ Check all | With selected: Edit Copy Delete Export

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

✓ Showing rows 0 - 6 (7 total, Query took 0.0093 seconds.)

```
SELECT * FROM `fare_table`
```

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

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

Extra options

bus_type	fareperkm
Superfast	5
Superfast	1.92
Superfast	1.32
Fast	5
Fast	1.6
Fast	1.1
medium	5

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

Showing rows 0 - 9 (10 total, Query took 0.0006 seconds.)

SELECT * FROM `route_info`

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

<div><div><div></div><div></div><div></div></div></div>				from_no	from_name	to_no	to_name	dist	bus_no
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	Bengaluru	2	Bangarpet	71	62621
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	Bengaluru	3	Katpadi	232	62621
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	Bengaluru	4	Sholigur	272	62621
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	Bengaluru	5	Chennai	362	62621
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	Bangarpet	3	Katpadi	161	62621
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	Bangarpet	4	Sholigur	201	62621
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	Bangarpet	5	Chennai	291	62621
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	Katpadi	4	Sholigur	40	62621
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	Katpadi	5	Chennai	130	62621
<input type="checkbox"/>	 Edit	 Copy	 Delete	4	Sholingur	5	Chennai	90	62621

Check all With selected: Edit Copy Delete Export

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

Showing rows 0 - 11 (12 total, Query took 0.0006 seconds.)

SELECT * FROM `ticket`

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

				PNR	bus_no	travel_date	departure	arrival	departure_time	arrival_time	passenger_id	bus_type
<input type="checkbox"/>	Edit	Copy	Delete	PNR001	62621	2022-08-07	Bengaluru	Chennai	16:00:00	20:30:00	ADM_001	Superfast
<input type="checkbox"/>	Edit	Copy	Delete	PNR002	62620	2022-08-14	Chennai	Bengaluru	08:00:00	12:25:00	ADM_001	Superfast
<input type="checkbox"/>	Edit	Copy	Delete	PNR003	25261	2022-08-01	Chennai	Mangaluru	18:00:00	07:30:00	ADM_003	medium
<input type="checkbox"/>	Edit	Copy	Delete	PNR004	25260	2022-08-15	Mangaluru	Chennai	19:15:00	08:00:00	ADM_003	medium
<input type="checkbox"/>	Edit	Copy	Delete	PNR005	58451	2022-08-03	Bengaluru	Mangaluru	06:00:00	14:00:00	USR_002	Fast
<input type="checkbox"/>	Edit	Copy	Delete	PNR006	58450	2022-08-04	Mangaluru	Bengaluru	08:00:00	10:15:00	USR_002	Fast
<input type="checkbox"/>	Edit	Copy	Delete	PNR007	62621	2022-08-07	Bangarpet	Chennai	18:00:00	20:30:00	ADM_002	Superfast
<input type="checkbox"/>	Edit	Copy	Delete	PNR008	62620	2022-08-02	Chennai	Bangarpet	08:00:00	09:00:00	ADM_002	Superfast
<input type="checkbox"/>	Edit	Copy	Delete	PNR010	25260	2022-08-19	Mangaluru	Kannur	19:15:00	23:15:00	USR_007	medium
<input type="checkbox"/>	Edit	Copy	Delete	PNR011	62621	2022-08-02	Bengaluru	Chennai	19:00:00	20:30:00	USR_008	Superfast
<input type="checkbox"/>	Edit	Copy	Delete	PNR012	58451	2022-08-11	Subramanya	Mangaluru	11:00:00	14:00:00	USR_009	Fast
<input type="checkbox"/>	Edit	Copy	Delete	PNR014	25260	2022-08-17	Kannur	Palakkad	23:15:00	04:30:00	USR_010	medium

Check all With selected: Edit Copy Delete Export

```
SELECT * FROM `ticket_passenger`
```

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all

Number of rows: 25 ▼

Filter rows:

Sort by key: None ▼

Extra options

			seat_no	name	age	PNR
<input type="checkbox"/>	 Edit	 Copy	 Delete	F01-3	Saideepak Reddy	27 PNR001
<input type="checkbox"/>	 Edit	 Copy	 Delete	F01-4	Sheela Reddy	25 PNR001
<input type="checkbox"/>	 Edit	 Copy	 Delete	F01-8	Christon D Souza	27 PNR008
<input type="checkbox"/>	 Edit	 Copy	 Delete	F02-11	SaiDeepak Reddy	27 PNR002
<input type="checkbox"/>	 Edit	 Copy	 Delete	F02-12	Sheela Reddy	25 PNR002
<input type="checkbox"/>	 Edit	 Copy	 Delete	G01-50	Anwar Hussein	32 PNR003
<input type="checkbox"/>	 Edit	 Copy	 Delete	G01-51	Karishma Husein	12 PNR003
<input type="checkbox"/>	 Edit	 Copy	 Delete	G02-31	Anwar Hussein	32 PNR004
<input type="checkbox"/>	 Edit	 Copy	 Delete	G02-32	Karishma Husein	12 PNR004
<input type="checkbox"/>	 Edit	 Copy	 Delete	S01-10	Hema Ullal	27 PNR005
<input type="checkbox"/>	 Edit	 Copy	 Delete	S01-11	Hima Ullal	28 PNR005
<input type="checkbox"/>	 Edit	 Copy	 Delete	S01-12	Asha Ullal	21 PNR005
<input type="checkbox"/>	 Edit	 Copy	 Delete	S01-14	Christon D Souza	27 PNR007
<input type="checkbox"/>	 Edit	 Copy	 Delete	S01-25	Samanta Prabhu	21 PNR010
<input type="checkbox"/>	 Edit	 Copy	 Delete	S01-26	Rakshita Prabhu	21 PNR010
<input type="checkbox"/>	 Edit	 Copy	 Delete	S01-35	Smriti Irani	40 PNR006
<input type="checkbox"/>	 Edit	 Copy	 Delete	S01-9	Ajit Ullal	31 PNR005

✓ Showing rows 0 - 12 (13 total. Query took 0.0006 seconds.)

SELECT * FROM `user`

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: | Sort by key: None

Extra options

				user_id	user_type	Fname	Lname	age	street_no	pincode	DOB
<input type="checkbox"/>				ADM_001	ADMIN	Narendra	Modi	33	Old Jail Road	600047	1989-04-14
<input type="checkbox"/>				ADM_002	ADMIN	Droupadi	Murmu	32	High Street	640012	1990-04-14
<input type="checkbox"/>				ADM_003	ADMIN	Manmohan	Singh	31	Pumpwell Road	575003	1991-04-08
<input type="checkbox"/>				USR_001	PSGR	Ajit	Ullal	20	10, Janpath	560001	2002-08-26
<input type="checkbox"/>				USR_002	PSGR	Muhammed	Ali	22	10, Downing	560003	2000-10-09
<input type="checkbox"/>				USR_003	PSGR	Sai Deepak	Reddy	26	14, Kailsh Marg	600001	1996-12-12
<input type="checkbox"/>				USR_004	PSGR	Rudra	Agarwal	26	5, Lohia Garden	600042	1996-12-13
<input type="checkbox"/>				USR_005	PSGR	Rahul	Khanna	25	1, Pink Street	600067	1997-12-14
<input type="checkbox"/>				USR_006	PSGR	Nirmala	Seturaman	24	3, Blue Avenue	560105	1998-12-15
<input type="checkbox"/>				USR_007	PSGR	Smriti	Irani	23	4, Banyan Avenie	575014	1999-12-16
<input type="checkbox"/>				USR_008	PSGR	Ajit	Sethi	56	6, Poes Garden	575001	1966-04-14
<input type="checkbox"/>				USR_009	PSGR	Arjun	Allu	55	7, Dhamaka street	575020	1967-04-08
<input type="checkbox"/>				USR_010	PSGR	Samanta	Prabhu	54	8, Clone Colony	600042	1968-04-14

☐ Check all | With selected: Edit Copy Delete Export


```
SELECT * FROM `user_bus`
```

☐ 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

				user_id	bus_no	date_time_stamp
<input type="checkbox"/>				USR_007	25260	2022-07-22 08:08:45
<input type="checkbox"/>				ADM_003	58450	2022-07-22 08:08:47
<input type="checkbox"/>				ADM_001	25261	2022-07-22 08:08:48
<input type="checkbox"/>				ADM_002	58450	2022-07-22 08:08:52
<input type="checkbox"/>				ADM_003	25261	2022-07-22 08:08:53
<input type="checkbox"/>				ADM_001	25261	2022-07-22 08:08:54
<input type="checkbox"/>				ADM_002	58450	2022-07-22 08:08:58
<input type="checkbox"/>				ADM_003	25261	2022-07-22 08:08:59
<input type="checkbox"/>				ADM_001	58450	2022-07-22 08:09:00
<input type="checkbox"/>				ADM_002	58450	2022-07-22 08:09:01
<input type="checkbox"/>				ADM_003	58450	2022-07-22 08:09:02
<input type="checkbox"/>				ADM_001	25261	2022-07-22 08:09:09
<input type="checkbox"/>				ADM_002	25261	2022-07-22 08:09:10
<input type="checkbox"/>				ADM_003	58450	2022-07-22 08:09:14
<input type="checkbox"/>				USR_004	58450	2022-07-22 08:09:20
<input type="checkbox"/>				USR_006	58450	2022-07-22 08:09:23
<input type="checkbox"/>				ADM_003	25260	2022-07-22 08:09:38
<input type="checkbox"/>				USR_004	25261	2022-07-22 08:10:15

✓ Showing rows 0 - 9 (10 total, Query took 0.0008 seconds.)

SELECT * FROM `user_phone`































☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all

Number of rows: 25 ▼

Filter rows:

Extra options

				user_id	phone_no
<input type="checkbox"/>	 Edit	 Copy	 Delete	USR_001	9900123456
<input type="checkbox"/>	 Edit	 Copy	 Delete	USR_002	9900123457
<input type="checkbox"/>	 Edit	 Copy	 Delete	USR_003	9900123458
<input type="checkbox"/>	 Edit	 Copy	 Delete	USR_004	9900123459
<input type="checkbox"/>	 Edit	 Copy	 Delete	USR_005	9900123450
<input type="checkbox"/>	 Edit	 Copy	 Delete	USR_006	9900123451
<input type="checkbox"/>	 Edit	 Copy	 Delete	USR_007	9900123452
<input type="checkbox"/>	 Edit	 Copy	 Delete	USR_008	9900123453
<input type="checkbox"/>	 Edit	 Copy	 Delete	USR_009	9900123454
<input type="checkbox"/>	 Edit	 Copy	 Delete	USR_010	9900123455

Join Queries

Showcase at least 4 join queries

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

INNER JOIN:

.Retrieve first and last name of users who have booked a ticket with price greater than 500

```
mysql> select distinct ticket_passenger.name from ticket_passenger join payment_info on payment_info.PNR=ticket_passenger.PNR where payment_info.price>500
-> ;
```

name
Anwar Hussein
Karishma Husein
Samanta Prabhu
Rakshita Prabhu

```
4 rows in set (0.06 sec)
```

LEFT OUTER JOIN

Retrieve the first name, last name, of the Users who have not bought a ticket.

```
mysql> select distinct(user.fname),user.lname from user left outer join ticket on user.user_id!=ticket.user_id
-> ;
```

fname	lname
Narendra	Modi
Droupadi	Murmu
Manmohan	Singh
Ajit	Ullal
Muhammed	Ali
Sai Deepak	Reddy

RIGHT OUTER JOIN

Retrieve the ticket PNR, BUS number, travel date and along with all users first name and last name

```
mysql> select ticket.PNR,ticket.bus_no,ticket.travel_date,user.fname,user.lname from ticket right outer join user on ticket.user_id=user.user_id
-> ;
```

PNR	bus_no	travel_date	fname	lname
PNR012	62620	2022-08-14	Narendra	Modi
PNR014	62621	2022-08-07	Narendra	Modi
PNR013	62620	2022-08-02	Droupadi	Murmu
PNR015	62621	2022-08-07	Droupadi	Murmu
PNR004	25260	2022-08-15	Manmohan	Singh
PNR008	25261	2022-08-01	Manmohan	Singh
NULL	NULL	NULL	Ajit	Ullal
PNR009	58450	2022-08-04	Muhammed	Ali
PNR010	58451	2022-08-03	Muhammed	Ali

NESTED QUERIES:

Retrieve the User ID who has spent more ticket price than the average ticket price.

```
mysql> select distinct(ticket.user_id) from ticket join payment_info on ticket.PNR=payment_info.PNR where payment_info.price >(select avg(price) from payment_info)
-> ;
```

user_id
ADM_003
USR_002

2 rows in set (0.15 sec)

Aggregate Functions

Showcase at least 4 Aggregate function queries

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

Find the average distance between subsequent stations for every bus

```
mysql> select avg(route_info.dist),bus.bus_no from bus join route_info on bus.bus_no=route_info.
bus_no group by bus.arrival
-> ;
```

avg(route_info.dist)	bus_no
278.7500	25260
219.1875	25261
220.5625	58451

3 rows in set (0.19 sec)

Display the number of phone numbers corresponding to the user_id
ADM_001, USR_006, USR_10

```
mysql> select user_id,phone_no from user_phone where user_id in('ADM_001','USR_006','USR_10');
```

user_id	phone_no
USR_006	9801212345
ADM_001	9845012345
ADM_001	9900123456
USR_006	9900123999

4 rows in set (0.31 sec)

Find the average fare per km for each Bus type specified and display the
bus type and corresponding average fare per km in decreasing order

```
mysql> select bus_type,avg(fareperkm) from fare_table group by bus_type order by avg(fareperkm)
desc;
```

bus_type	avg(fareperkm)
Superfast	2.1799999872843423
Fast	1.8166666825612385
medium	1.5333333214124043

3 rows in set (0.08 sec)

Count the number of passengers whose name consists of 'UllaL'

```
mysql> select count(*) from user where fname like "%ulla1%" or lname like "%ulla1%";
+-----+
| count(*) |
+-----+
|          1 |
+-----+
1 row in set (0.01 sec)
```

Set Operations

Showcase at least 4 Set Operations queries

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

List of passengers (user_id, user_type First name and last name) who have travelled from chennai to bengaluru during the month of Oct 2021 and Aug 2022

```
mysql> insert into ticket (PNR,bus_no,travel_date,departure,arrival,departure_time,arrival_time,
user_id,bus_type)VALUES ('PNR061',62621,'2021-11-22','Chennai','Bengaluru','16:00:00','20:30:00'
,'USR_009','medium')
-> ;
Query OK, 1 row affected (0.10 sec)

mysql> update bus set arrival = trim(arrival),destination = trim(destination);
Query OK, 2 rows affected (0.08 sec)
Rows matched: 6  Changed: 2  Warnings: 0
```

List of passengers (user_id, user_type First name and last name) who have travelled from chennai to bengaluru during the month of Oct 2021 and not Aug 2022

```
mysql> select user_id, user_type, fname, lname from user where user_id in (select user_id from t
icket where Departure = 'Chennai' and Arrival = 'Bengaluru' and YEAR(travel_date) = 2022 and MON
TH(travel_date) in (01,02,03,04,05,06,07,08)) UNION select user_id, user_type, fname, lname from
user where user_id in (select user_id from ticket where Departure = 'Chennai' and Arrival = 'Be
ngaluru' and YEAR(Travel_Date) = 2021 and MONTH(Travel_Date) in (10,11,12));
+-----+-----+-----+-----+
| user_id | user_type | fname   | lname  |
+-----+-----+-----+-----+
| ADM_001 | ADMIN    | Narendra | Modi   |
| USR_009 | PSGR     | Arjun   | Allu   |
+-----+-----+-----+-----+
```

List of passengers (user_id, user_type First name and last name) who have travelled from chennai to bengaluru during the month of Oct 2021 and not Aug 2022

```
mysql> select u1.User_ID, u1.User_Type, u1.Fname, u1.Lname from user u1, ticket t1 where u1.User_ID = t1.User_ID and t1.Departure = 'Chennai' and t1.Arrival = 'Bengaluru' and YEAR(t1.Travel_Date) = 2022 and MONTH(t1.Travel_Date) = 08 and NOT EXISTS (select u2.User_ID, u2.User_Type, u2.Fname, u2.Lname from user u2, ticket t2 where u2.User_ID = t2.User_ID and t2.Departure = 'Chennai' and t2.Arrival = 'Bengaluru' and YEAR(t2.Travel_Date) = 2021 and MONTH(t2.Travel_Date) = 11 and u1.User_ID = u2.User_ID);
```

User_ID	User_Type	Fname	Lname
ADM_001	ADMIN	Narendra	Modi

List of passengers (user_id, user_type First name and last name) who have travelled from chennai to bengaluru during the month of Oct 2021 and not Aug 2022

```
mysql> select u1.User_ID, u1.User_Type, u1.Fname, u1.Lname from user u1, ticket t1 where u1.User_ID = t1.User_ID and t1.Departure = 'Chennai' and t1.Arrival = 'Bengaluru' and NOT EXISTS (select u2.User_ID, u2.User_Type, u2.Fname, u2.Lname from user u2, ticket t2 where u2.User_ID = t2.User_ID and t2.Departure = 'Bengaluru' and t2.Arrival = 'Chennai' and u1.User_ID = u2.User_ID);
```

User_ID	User_Type	Fname	Lname
USR_009	PSGR	Arjun	Allu

1 row in set (0.00 sec)

Functions and Procedures

Create a Function and Procedure. State the objective of the function / Procedure. Run and display the results.

Function: number of tickets booked by a customer. If no of tickets is not more than 3 for the current month then display "can buy ticket " else display error "cannot buy tickets , limit is over"

```
mysql> CREATE FUNCTION ticket_no(user_id varchar(20))
-> returns varchar(50)
-> deterministic
-> begin
-> declare result varchar(50);
-> declare ticket_count int;
-> select count(PNR) into ticket_count
-> from ticket
-> where user_id = user_id and
-> month(travel_date) = month(sysdate()) and
-> year(travel_date)=year(sysdate());
-> if ticket_count > 3 then
-> set result = 'cannot buy tickets ,limit is over';
-> else
-> set result = 'can buy tickets';
-> end if;
-> return result;
-> end;
-> $$
Query OK, 0 rows affected (0.38 sec)
```

```
mysql> select user_id, ticket_no(user_id) from user_bus;
+-----+-----+
| user_id | ticket_no(user_id) |
+-----+-----+
| USR_007 | can buy tickets    |
| ADM_002 | can buy tickets    |
| ADM_003 | can buy tickets    |
| ADM_001 | can buy tickets    |
| ADM_002 | can buy tickets    |
| ADM_003 | can buy tickets    |
| ADM_001 | can buy tickets    |
| ADM_002 | can buy tickets    |
| ADM_003 | can buy tickets    |
| ADM_001 | can buy tickets    |
| ADM_002 | can buy tickets    |
```


PROCEDURE: updated age is the stored procedure to calculate the age of the customer when the date of birth is given. Updates the column named age in the customer table.

```
mysql> DELIMITER $$
mysql> create procedure updated_age(IN usr_id varchar(20), in dob date)
    -> begin
    -> declare n_age int;
    -> set n_age =floor(datediff(current_date,dob)/365);
    -> update user set age =n_age where user_id = usr_id;
    -> end;$$
Query OK, 0 rows affected (0.06 sec)

mysql> DELIMITER ;
```

```
mysql> select * from user where user_id = 'ADM_001';
+-----+-----+-----+-----+-----+-----+-----+-----+
| user_id | user_type | Fname   | Lname  | age | street_no       | pincode | DOB       |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ADM_001 | ADMIN     | Narendra | Modi   | 33  | Old Jail Road   | 600047  | 1989-04-14 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.06 sec)
```

```
mysql> call updated_age('ADM_001','1990-08-12');
Query OK, 1 row affected (0.06 sec)
```

```
mysql> select * from user;
+-----+-----+-----+-----+-----+-----+-----+-----+
| user_id | user_type | Fname   | Lname  | age | street_no       | pincode | DOB       |
+-----+-----+-----+-----+-----+-----+-----+-----+
| ADM_001 | ADMIN     | Narendra | Modi   | 32  | Old Jail Road   | 600047  | 1989-04-14 |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Triggers and Cursors

Create a Trigger. State the objective. Run and display the results.

Trigger to add payment information to the backup table named payment backup when we try to delete some information from the ticket table.

Trigger used : before delete

```
ERROR 1306 (HY000): Trigger does not exist
mysql> desc payment_backup
-> ;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| transaction_id | int           | NO   | PRI | NULL    |       |
| bank           | varchar(20)   | YES  |     | NULL    |       |
| card_no        | bigint        | YES  |     | NULL    |       |
| price          | double        | YES  |     | NULL    |       |
| PNR            | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)

mysql> DELIMITER $$
mysql> CREATE TRIGGER payment_backup
-> before delete on ticket for each row
-> begin insert into payment_backup
-> select * from payment_info where PNR=old.PNR;
-> delete from payment_info where PNR =old.PNR;
-> delete from ticket_passenger where PNR = old.PNR;
-> END $$
Query OK, 0 rows affected (0.02 sec)

mysql> delimiter ;
```

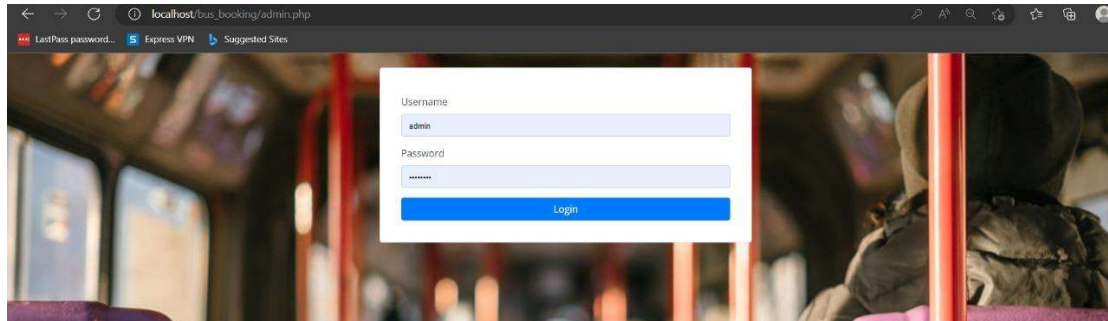
We can see the results from the below screenshots:

```
mysql> delete from ticket WHERE PNR ='PNR005';
Query OK, 1 row affected (0.02 sec)

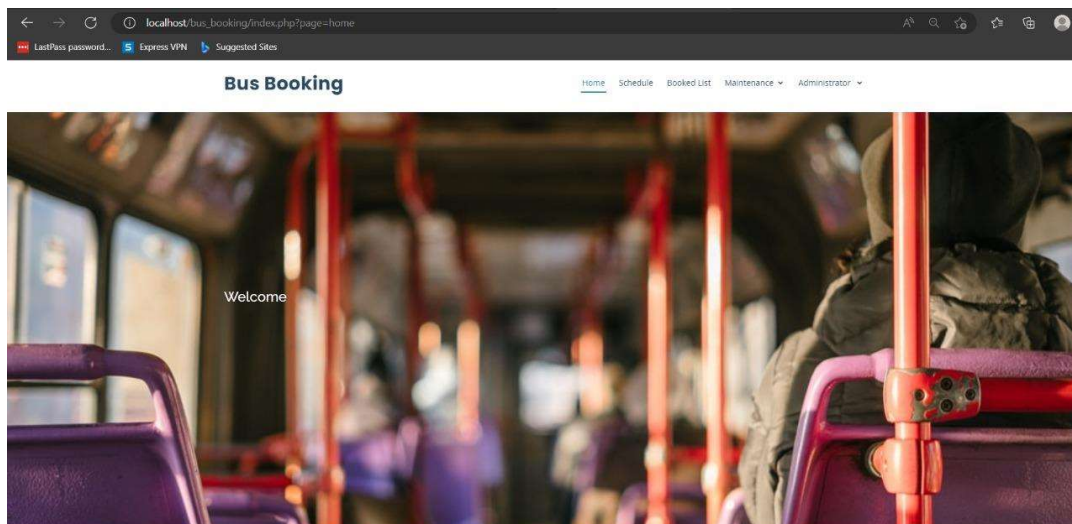
mysql> SELECT * FROM payment_backup;
+-----+-----+-----+-----+-----+
| transaction_id | bank          | card_no          | price          | PNR    |
+-----+-----+-----+-----+-----+
| 978098         | INDIAN BANK  | 9890000000000000 | 497.7999937534332 | PNR005 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Developing a Frontend

Login page



Admin page



User page

