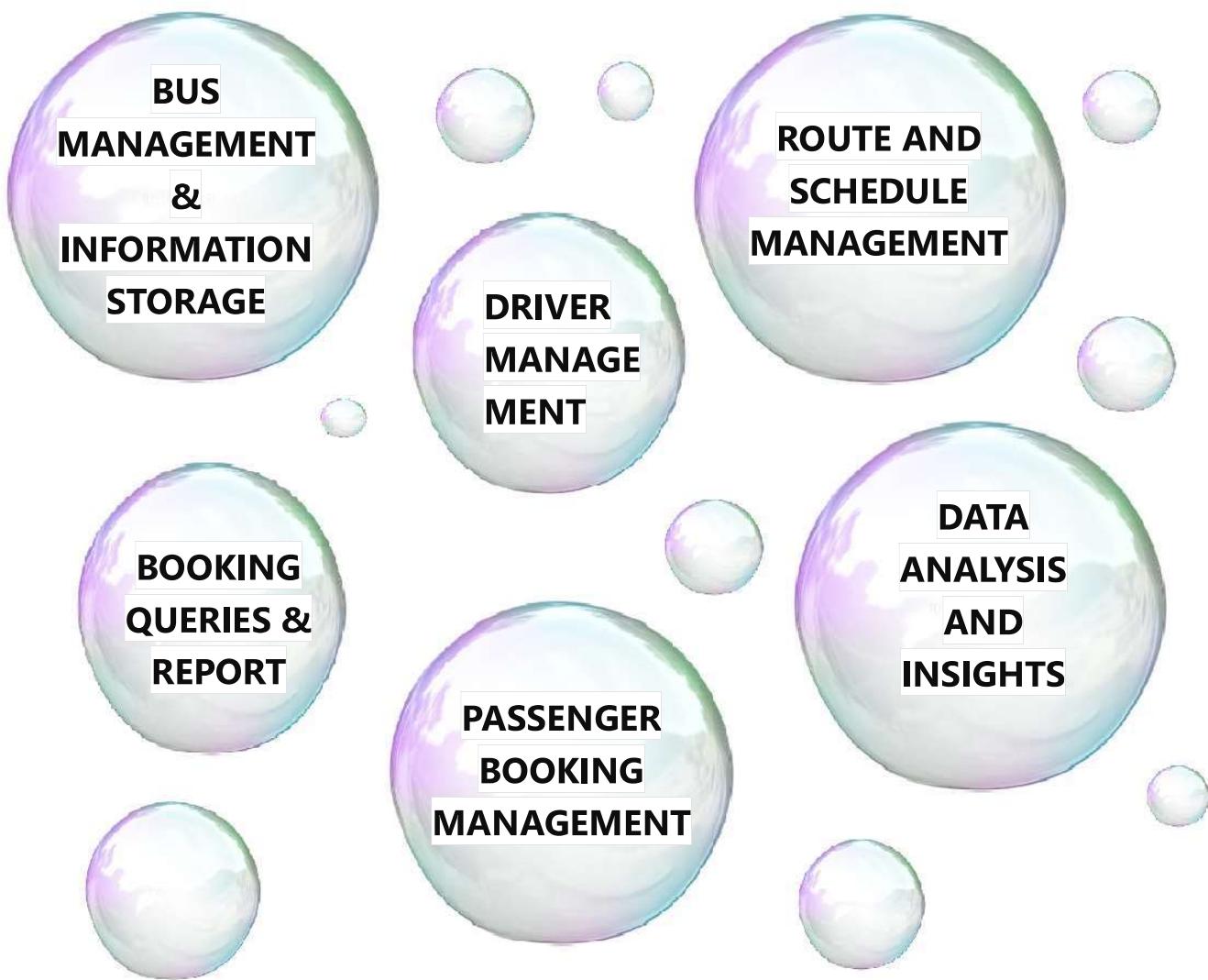


BUS MANAGEMENT SYSTEM



COMMANDS USED:

CREATE (6)	DELETE(2)	AVG() (1)	SUBSTRING() (1)	UNION (1)
INSERT INTO (5)	JOIN (26)	MAX() (3)	ROUND() (1)	INTERSECT (1)
SELECT (49)	ORDER BY (6)	COUNT() (10)	SUM() (1)	EXCEPT (1)
UPDATE(3)	GROUP BY (9)	UPPER() (1)	DISTINCT (1)	VIEW (1)

Commands Used:

DDL	CREATE (6)	Defines and initializes a new database object, such as a table, index, or view.
DML	INSERT INTO (5)	Adds new records (rows) into a specified table in the database.
	SELECT (49)	Retrieves data from one or more tables, possibly with conditions and transformations.
	UPDATE(3)	Modifies existing records in a table based on specified conditions.
	DELETE(2)	Removes records from a table that match a given condition.
Operators and Functions	JOIN (26)	Combines rows from two or more tables based on a related column.
	ORDER BY (6)	Sorts the result set of a query in ascending or descending order.
	GROUP BY (9)	Groups rows that have the same values in specified columns, often used with aggregate functions.
	AVG() (1)	Calculates the average value of a numeric column.
	MAX() (3)	Returns the largest value in a specified column.
	COUNT() (10)	Counts the number of rows or non-NULL values in a column.
	UPPER() (1)	Converts the text in a column or value to uppercase letters.
	SUBSTRING() (1)	Extracts a portion of a string based on specified start and length parameters.
	ROUND() (1)	Rounds a numeric value to a specified number of decimal places.
	SUM() (1)	Computes the total sum of a numeric column.
	DISTINCT (1)	Removes duplicate rows from the result set, returning unique values.
	UNION (1)	Combines result sets of two queries, removing duplicates in the final output.
	INTERSECT (1)	Returns common rows between two result sets.
	EXCEPT (1)	Returns rows from one query that are not in another query's result.
DDE	VIEW (1)	Creates a virtual table based on the result of a SELECT query, simplifying complex queries.

BUS MANAGEMENT SYSTEM

This project focuses on managing a bus system's operations, including bus information, drivers, routes, bookings, and schedules. We will design and implement a relational database with five tables: **Bus**, **Drivers**, **Route**, **Bookings**, and **Schedule**. We will then populate the tables with relevant data and perform different types of SQL queries on them.

1. DATABASE DESIGN

The following is the structure for the **Bus Management System** database.

I. Bus Table

This table stores information about buses.

❖ Creating table Bus:-

```
CREATE TABLE Bus (
bus_id INT PRIMARY KEY,
bus_nameVARCHAR(50),
bus_typeVARCHAR(30),
route_idINT);
```

❖ Inserting data values into the table:-

```
INSERT INTO Bus (bus_id, bus_name, bus_type, capacity, route_id)
VALUES
(1, 'Shivneri Express', 'AC', 40, 1),
(2, 'Golden Travels', 'Non-AC', 50, 2),
(3, 'Bengaluru Express', 'Sleeper', 60, 3),
```

```
(4, 'Kolkata Deluxe', 'AC', 45, 4),
(5, 'Telangana Super', 'Non-AC', 48, 5),
(6, 'Mewar Express', 'Sleeper', 55, 6),
(7, 'MP Travels', 'AC', 40, 7),
(8, 'Patna-Ranchi Express', 'Non-AC', 50, 8),
(9, 'Ganga Travels', 'AC', 35, 9),
(10, 'Surat Express', 'Non-AC', 45, 10);
```

❖ Viewing all the records of the table:-

```
SELECT*FROM Bus;
```

Output:-

Bus				
bus_id	bus_name	bus_type	capacity	route_id
1	Shivneri Express	AC	40	1
2	Golden Travels	Non-AC	50	2
3	Bengaluru Express	Sleeper	60	3
4	Kolkata Deluxe	AC	45	4
5	Telangana Super	Non-AC	48	5
6	Mewar Express	Sleeper	55	6
7	MP Travels	AC	40	7
8	Patna-Ranchi Express	Non-AC	50	8
9	Ganga Travels	AC	35	9
10	Surat Express	Non-AC	45	10

II. Drivers Table

This table stores information about bus drivers.

❖ Creating table Drivers:-

```
CREATE TABLE Drivers (
    driver_id INT PRIMARY KEY,
    driver_name VARCHAR(50),
    contact_number VARCHAR(15),
    license_number VARCHAR(20),
    bus_id INT,
    FOREIGN KEY (bus_id) REFERENCES Bus(bus_id));
```

❖ Inserting data values into the table:-

```
INSERT INTO Drivers (driver_id, driver_name, contact_number,
    license_number, bus_id)
VALUES
    (1, 'Ravi Kumar', '9876543210', 'DL12345', 1),
    (2, 'Rajesh Sharma', '9887654321', 'DL12346', 2),
    (3, 'Suresh Babu', '9976543210', 'KA12345', 3),
    (4, 'Manoj Verma', '9945332170', 'WB12345', 4),
    (5, 'Kiran Reddy', '9912345678', 'TS12345', 5),
    (6, 'Vikas Pandey', '9908765432', 'RJ12345', 6),
    (7, 'Sunil Yadav', '9812345678', 'MP12345', 7),
    (8, 'Prakash Kumar', '9823456789', 'BR12345', 8),
    (9, 'Ajay Verma', '9898765432', 'UP12345', 9),
    (10, 'Vishal Patel', '9807654321', 'GJ12345', 10);
```

❖ Viewing all the records of the table:-

```
SELECT*FROM Drivers;
```

Output:-

Drivers				
driver_id	driver_name	contact_number	license_number	bus_id
1	Ravi Kumar	9876543210	DL12345	1
2	Rajesh Sharma	9887654321	DL12346	2
3	Suresh Babu	9976543210	KAI2345	3
4	Manoj Verma	9945332170	WB12345	4
5	Kiran Reddy	9912345678	TS12345	5
6	Vikas Pandey	9908765432	RJ12345	6
7	Sunil Yadav	9812345678	MP12345	7
8	Prakash Kumar	9823456789	BRI2345	8
9	Ajay Verma	9898765432	UPI2345	9
10	Vishal Patel	9807654321	GJI2345	10

III. Route Table

This table stores route details for the buses.

❖ Creating table Route:-

```
CREATE TABLE Route (
    route_id INT PRIMARY KEY,
    source VARCHAR(50),
    destination VARCHAR(50),
    distance INT
    duration VARCHAR(20)
);
```

❖ Inserting data values into the table:-

```
INSERT INTO Route (route_id, source, destination, distance, duration)
VALUES
(1, 'Mumbai', 'Pune', 150, '3 hours'),
(2, 'Delhi', 'Agra', 230, '5 hours'),
(3, 'Chennai', 'Bengaluru', 350, '7 hours'),
(4, 'Kolkata', 'Durgapur', 170, '4 hours'),
(5, 'Hyderabad', 'Warangal', 150, '3 hours'),
(6, 'Jaipur', 'Udaipur', 400, '8 hours'),
(7, 'Bhopal', 'Indore', 200, '4 hours'),
(8, 'Patna', 'Ranchi', 250, '6 hours'),
(9, 'Lucknow', 'Kanpur', 80, '2 hours'),
(10, 'Ahmedabad', 'Vadodara', 110, '2.5 hours');
```

❖ Viewing all the records of the table:-

```
SELECT*FROM Route;
```

Output:-

Route				
route_id	source	destination	distance	duration
1	Mumbai	Pune	150	3 hours
2	Delhi	Agra	230	5 hours
3	Chennai	Bengaluru	350	7 hours
4	Kolkata	Durgapur	170	4 hours
5	Hyderabad	Warangal	150	3 hours
6	Jaipur	Udaipur	400	8 hours
7	Bhopal	Indore	200	4 hours
8	Patna	Ranchi	250	6 hours
9	Lucknow	Kanpur	80	2 hours
10	Ahmedabad	Vadodara	110	2.5 hours

IV. Bookings Table

This table stores booking information for passengers.

❖ Creating table Bookings:-

```
CREATE TABLE Bookings (
    booking_id INT PRIMARY KEY,
    passenger_name VARCHAR(50),
    bus_id INT,
    booking_date DATE,
    seat_number INT,
    FOREIGN KEY (bus_id) REFERENCES Bus(bus_id)
);
```

❖ Inserting data values into the table:-

```
INSERT INTO Bookings (booking_id, passenger_name, bus_id, booking_date,
seat_number)
VALUES
(1, 'Ramesh Verma', 1, '2024-11-10', 1),
(2, 'Sita Devi', 2, '2024-11-11', 5),
(3, 'Vijay Kumar', 3, '2024-11-12', 10),
(4, 'Nisha Agarwal', 4, '2024-11-13', 2),
(5, 'Ajay Singh', 5, '2024-11-14', 15),
(6, 'Priya Sharma', 6, '2024-11-15', 20),
(7, 'Sunita Patel', 7, '2024-11-16', 25),
(8, 'Deepak Gupta', 8, '2024-11-17', 30),
(9, 'Neha Rai', 9, '2024-11-18', 35),
(10, 'Manoj Mishra', 10, '2024-11-19', 40);
```

❖ Viewing all the records of the table:-

```
SELECT*FROM Bookings;
```

Output:-

Bookings				
booking_id	passenger_name	bus_id	booking_date	seat_number
1	Ramesh Verma	1	2024-11-10	1
2	Sita Devi	2	2024-11-11	5
3	Vijay Kumar	3	2024-11-12	10
4	Nisha Agarwal	4	2024-11-13	2
5	Ajay Singh	5	2024-11-14	15
6	Priya Sharma	6	2024-11-15	20
7	Sunita Patel	7	2024-11-16	25
8	Deepak Gupta	8	2024-11-17	30
9	Neha Rai	9	2024-11-18	35
10	Manoj Mishra	10	2024-11-19	40

V. Schedule Table

This table stores the schedule of buses, including the time of departure.

❖ Creating table Schedule:-

```
CREATE TABLE Schedule (
    schedule_id INT PRIMARY KEY,
    bus_id INT,
    departure_time TIME,
    arrival_time TIME,
    date_of_travel DATE,
    FOREIGN KEY (bus_id) REFERENCES Bus(bus_id)
);
```

❖ Inserting data values into the table:-

```
INSERT INTO Schedule (schedule_id, bus_id, departure_time, arrival_time,
date_of_travel)VALUES
(1, 1, '08:00:00', '11:00:00', '2024-11-10'),
(2, 2, '09:00:00', '14:00:00', '2024-11-11'),
(3, 3, '10:00:00', '17:00:00', '2024-11-12'),
(4, 4, '11:00:00', '15:00:00', '2024-11-13'),
(5, 5, '12:00:00', '15:30:00', '2024-11-14'),
(6, 6, '13:00:00', '18:00:00', '2024-11-15'),
(7, 7, '14:00:00', '18:30:00', '2024-11-16'),
(8, 8, '15:00:00', '21:00:00', '2024-11-17'),
(9, 9, '16:00:00', '18:00:00', '2024-11-18'),
(10, 10, '17:00:00', '19:30:00', '2024-11-19');
```

❖ Viewing all the records of the table:-

```
SELECT*FROM Schedule;
```

Output:-

Schedule				
schedule_id	bus_id	departure_time	arrival_time	date_of_travel
1	1	08:00:00	11:00:00	2024-11-10
2	2	09:00:00	14:00:00	2024-11-11
3	3	10:00:00	17:00:00	2024-11-12
4	4	11:00:00	15:00:00	2024-11-13
5	5	12:00:00	15:30:00	2024-11-14
6	6	13:00:00	18:00:00	2024-11-15
7	7	14:00:00	18:30:00	2024-11-16
8	8	15:00:00	21:00:00	2024-11-17
9	9	16:00:00	18:00:00	2024-11-18
10	10	17:00:00	19:30:00	2024-11-19

2. Performing SQL Queries On These Tables

- ❖ To get the details of buses having capacity greater than 45.

```
SELECT * FROM Bus
```

```
WHERE capacity > 45;
```

Output:-

Output

bus_id	bus_name	bus_type	capacity	route_id
2	Golden Travels	Non-AC	50	2
3	Bengaluru Express	Sleeper	60	3
5	Telangana Super	Non-AC	48	5
6	Mewar Express	Sleeper	55	6
8	Patna-Ranchi Express	Non-AC	50	8

- ❖ To get the list of all bookings made on 2024-11-10.

```
SELECT*FROM Bookings
```

```
WHERE booking_date='2024-11-10';
```

Output:-

Output

booking_id	passenger_name	bus_id	booking_date	seat_number
1	Ramesh Verma	1	2024-11-10	1

- ❖ To get the details of drivers for AC buses only.

```
SELECT D.driver_name, D.contact_number, B.bus_name
FROM Drivers D
JOIN Bus B ON D.bus_id = B.bus_id
WHERE B.bus_type = 'AC';
```

Output:-

Output

driver_name	contact_number	bus_name
Ravi Kumar	9876543210	Shivneri Express
Manoj Verma	9945332170	Kolkata Deluxe
Sunil Yadav	9812345678	MP Travels
Ajay Verma	9898765432	Ganga Travels

- ❖ To get all buses sorted by their capacity (highest to lowest).

```
SELECT * FROM Bus
ORDER BY capacity DESC;
```

Output:-

Output

bus_id	bus_name	bus_type	capacity	route_id
3	Bengaluru Express	Sleeper	60	3
6	Mewar Express	Sleeper	55	6
2	Golden Travels	Non-AC	50	2
8	Patna-Ranchi Express	Non-AC	50	8
5	Telangana Super	Non-AC	48	5
4	Kolkata Deluxe	AC	45	4
10	Surat Express	Non-AC	45	10
1	Shivneri Express	AC	40	1
7	MP Travels	AC	40	7
9	Ganga Travels	AC	35	9

- ❖ To get all the bookings sorted by booking date.

```
SELECT * FROM Bookings
ORDER BY booking_date;
```

Output:-

Output				
booking_id	passenger_name	bus_id	booking_date	seat_number
1	Ramesh Verma	1	2024-11-10	1
2	Sita Devi	2	2024-11-11	5
3	Vijay Kumar	3	2024-11-12	10
4	Nisha Agarwal	4	2024-11-13	2
5	Ajay Singh	5	2024-11-14	15
6	Priya Sharma	6	2024-11-15	20
7	Sunita Patel	7	2024-11-16	25
8	Deepak Gupta	8	2024-11-17	30
9	Neha Rai	9	2024-11-18	35
10	Manoj Mishra	10	2024-11-19	40

- ❖ To get the buses sorted by their route's distance (longest to shortest).

```
SELECT B.bus_name, R.source, R.destination, R.distance
```

```
FROM Bus B
JOIN Route R ON B.route_id = R.route_id
ORDER BY R.distance DESC;
```

Output :-

Output			
bus_name	source	destination	distance
Mewar Express	Jaipur	Udaipur	400
Bengaluru Express	Chennai	Bengaluru	350
Patna-Ranchi Express	Patna	Ranchi	250
Golden Travels	Delhi	Agra	230
MP Travels	Bhopal	Indore	200
Kolkata Deluxe	Kolkata	Durgapur	170
Shivneri Express	Mumbai	Pune	150
Telangana Super	Hyderabad	Warangal	150
Surat Express	Ahmedabad	Vadodara	110
Ganga Travels	Lucknow	Kanpur	80

- ❖ To get the total number of bookings for each bus.

```
SELECT B.bus_name, COUNT(Bo.booking_id) AS  
total_bookings FROM Bus B  
JOIN Bookings Bo ON B.bus_id = Bo.bus_id  
GROUP BY B.bus_name;
```

Output :-

Output	
bus_name	total_bookings
Bengaluru Express	1
Ganga Travels	1
Golden Travels	1
Kolkata Deluxe	1
MP Travels	1
Mewar Express	1
Patna-Ranchi Express	1
Shivneri Express	1
Surat Express	1
Telangana Super	1

- ❖ To get the average distance of all routes .

```
SELECT AVG(distance) AS  
average_route_distance FROM Route;
```

Output :-

Output	
average_route_distance	
209	

- ❖ To get the maximum capacity bus .

```
SELECT bus_name, MAX(capacity) AS max_capacity
FROM Bus
GROUP BY bus_name;
```

Output :-

Output	
bus_name	max_capacity
Bengaluru Express	60
Ganga Travels	35
Golden Travels	50
Kolkata Deluxe	45
MP Travels	40
Mewar Express	55
Patna-Ranchi Express	50
Shivneri Express	40
Surat Express	45
Telangana Super	48

- ❖ To get the number of buses available for each route .

```
SELECT R.source, R.destination, COUNT(B.bus_id) AS
number_of_buses
FROM Route R
JOIN Bus B ON R.route_id = B.route_id
GROUP BY R.source, R.destination;
```

Output :-

Output		
source	destination	number_of_buses
Ahmedabad	Vadodara	1
Bhopal	Indore	1
Chennai	Bengaluru	1
Delhi	Agra	1
Hyderabad	Warangal	1
Jaipur	Udaipur	1
Kolkata	Durgapur	1
Lucknow	Kanpur	1
Mumbai	Pune	1
Patna	Ranchi	1

- ❖ To get the number of bookings for each bus(grouped by bus) .

```
SELECT B.bus_name, COUNT(Bo.booking_id) AS
number_of_bookingsFROM Bus B
LEFT JOIN Bookings Bo ON B.bus_id = Bo.bus_id
GROUP BY B.bus_name;
```

Output :-

Output	
bus_name	number_of_bookings
Bengaluru Express	1
Ganga Travels	1
Golden Travels	1
Kolkata Deluxe	1
MP Travels	1
Mewar Express	1
Patna-Ranchi Express	1
Shivneri Express	1
Surat Express	1
Telangana Super	1

- ❖ To get the total number of bookings made per day.

```
SELECT booking_date, COUNT(booking_id) AS
total_bookings
```

Output :-

Output	
booking_date	total_bookings
2024-11-10	1
2024-11-11	1
2024-11-12	1
2024-11-13	1
2024-11-14	1
2024-11-15	1
2024-11-16	1
2024-11-17	1
2024-11-18	1
2024-11-19	1

- ❖ To get the bus with the highest capacity.

```
SELECT * FROM Bus
WHERE capacity = (SELECT MAX(capacity) FROM
Bus);
```

Output :-

Output

bus_id	bus_name	bus_type	capacity	route_id
3	Bengaluru Express	Sleeper	60	3

- ❖ To get the details of drivers who drive buses that are not of Type 'Non-AC'.

```
SELECT * FROM Drivers WHERE bus_id
IN (SELECT bus_id FROM Bus
WHERE bus_type != 'Non-AC');
```

Output :-

Output

driver_id	driver_name	contact_number	license_number	bus_id
1	Ravi Kumar	9876543210	DLI2345	1
3	Suresh Babu	9976543210	KAI2345	3
4	Manoj Verma	9945332170	WBI2345	4
6	Vikas Pandey	9908765432	RJI2345	6
7	Sunil Yadav	9812345678	MPI2345	7
9	Ajay Verma	9898765432	UPI2345	9

- ❖ To get the booking details where the bus is going from 'Mumbai' to 'Pune' .

```
SELECT * FROM Bookings
WHERE bus_id IN (SELECT bus_id FROM Bus WHERE route_id =
(SELECT route_id FROM Route WHERE source = 'Mumbai' AND
destination = 'Pune'));
```

Output :-

Output				
booking_id	passenger_name	bus_id	booking_date	seat_number
1	Ramesh Verma	1	2024-11-10	1

- ❖ To update the bus capacity to 60 for a specific bus(eg., bus with Bus_id 1).

```
UPDATE BusSET capacity = 60
WHERE bus_id = 1;
```

Output :-

Bus				
bus_id	bus_name	bus_type	capacity	route_id
1	Shivneri Express	AC	60	1
2	Golden Travels	Non-AC	50	2
3	Bengaluru Express	Sleeper	60	3
4	Kolkata Deluxe	AC	45	4
5	Telangana Super	Non-AC	48	5
6	Mewar Express	Sleeper	55	6
7	MP Travels	AC	40	7
8	Patna-Ranchi Express	Non-AC	50	8
9	Ganga Travels	AC	35	9
10	Surat Express	Non-AC	45	10

- ❖ To update a driver's contact number.

```
UPDATE DriversSET
contact_number = '9999999999'
WHERE driver_id = 3;
```

Output :-

Drivers				
driver_id	driver_name	contact_number	license_number	bus_id
1	Ravi Kumar	9876543210	DL12345	1
2	Rajesh Sharma	9887654321	DL12346	2
3	Suresh Babu	9999999999	KA12345	3
4	Manoj Verma	9945332170	WB12345	4
5	Kiran Reddy	9912345678	TS12345	5
6	Vikas Pandey	9908765432	RJ12345	6
7	Sunil Yadav	9812345678	MP12345	7
8	Prakash Kumar	9823456789	BR12345	8
9	Ajay Verma	9898765432	UP12345	9
10	Vishal Patel	9807654321	GJ12345	10

- ❖ To update the departure time for the bus with bus_id 2 to 07:30:00.

```
UPDATE Schedule
SET departure_time = '07:30:00'
WHERE bus_id = 2
```

Output :-

Schedule				
schedule_id	bus_id	departure_time	arrival_time	date_of_travel
1	1	08:00:00	11:00:00	2024-11-10
2	2	07:30:00	14:00:00	2024-11-11
3	3	10:00:00	17:00:00	2024-11-12
4	4	11:00:00	15:00:00	2024-11-13
5	5	12:00:00	15:30:00	2024-11-14
6	6	13:00:00	18:00:00	2024-11-15
7	7	14:00:00	18:30:00	2024-11-16
8	8	15:00:00	21:00:00	2024-11-17
9	9	16:00:00	18:00:00	2024-11-18
10	10	17:00:00	19:30:00	2024-11-19

- ❖ To delete a booking for specific passenger.

```
DELETE FROM Bookings
WHERE passenger_name = 'Ramesh Verma';
```

Output :-

Bookings				
booking_id	passenger_name	bus_id	booking_date	seat_number
2	Sita Devi	2	2024-11-11	5
3	Vijay Kumar	3	2024-11-12	10
4	Nisha Agarwal	4	2024-11-13	2
5	Ajay Singh	5	2024-11-14	15
6	Priya Sharma	6	2024-11-15	20
7	Sunita Patel	7	2024-11-16	25
8	Deepak Gupta	8	2024-11-17	30
9	Neha Rai	9	2024-11-18	35
10	Manoj Mishra	10	2024-11-19	40

- ❖ To delete all schedules for buses with route_id=1.

```
DELETE FROM Schedule WHERE bus_id IN
(SELECT bus_id FROM Bus WHERE route_id = 1);
```

Output :-

Schedule				
schedule_id	bus_id	departure_time	arrival_time	date_of_travel
1	1	08:00:00	11:00:00	2024-11-10
2	2	07:30:00	14:00:00	2024-11-11
3	3	10:00:00	17:00:00	2024-11-12
4	4	11:00:00	15:00:00	2024-11-13
5	5	12:00:00	15:30:00	2024-11-14
6	6	13:00:00	18:00:00	2024-11-15
7	7	14:00:00	18:30:00	2024-11-16
8	8	15:00:00	21:00:00	2024-11-17
9	9	16:00:00	18:00:00	2024-11-18
10	10	17:00:00	19:30:00	2024-11-19

- ❖ To get the list of all passengers with their booking details and the respective route information.

```

SELECT Bo.passenger_name, Bo.booking_date, B.bus_name,
R.source,R.destinationFROM Bookings Bo
JOIN Bus B ON Bo.bus_id = B.bus_id
JOIN Route R ON B.route_id = R.route_id;

```

Output :-

Output				
passenger_name	booking_date	bus_name	source	destination
Sita Devi	2024-11-11	Golden Travels	Delhi	Agra
Vijay Kumar	2024-11-12	Bengaluru Express	Chennai	Bengaluru
Nisha Agarwal	2024-11-13	Kolkata Deluxe	Kolkata	Durgapur
Ajay Singh	2024-11-14	Telangana Super	Hyderabad	Warangal
Priya Sharma	2024-11-15	Mewar Express	Jaipur	Udaipur
Sunita Patel	2024-11-16	MP Travels	Bhopal	Indore
Deepak Gupta	2024-11-17	Patna-Ranchi Express	Patna	Ranchi
Neha Rai	2024-11-18	Ganga Travels	Lucknow	Kanpur
Manoj Mishra	2024-11-19	Surat Express	Ahmedabad	Vadodara

- ❖ To get the list of all buses along with the driver's name, contact, and the route they are assigned to.

```

SELECT B.bus_name, D.driver_name, D.contact_number, R.source,
R.destinationFROM Bus B
JOIN Drivers D ON B.bus_id = D.bus_id
JOIN Route R ON B.route_id = R.route_id;

```

Output :-

Output				
bus_name	driver_name	contact_number	source	destination
Shivneri Express	Ravi Kumar	9876543210	Mumbai	Pune
Golden Travels	Rajesh Sharma	9887654321	Delhi	Agra
Bengaluru Express	Suresh Babu	9999999999	Chennai	Bengaluru
Kolkata Deluxe	Monu Verma	994532170	Kolkata	Durgapur
Telangana Super	Kiran Reddy	9912345678	Hyderabad	Warangal
Mewar Express	Vikas Pandey	9908765432	Jaipur	Udaipur
MP Travels	Sunil Yadav	9812345678	Bhopal	Indore
Patna-Ranchi Express	Prakash Kumar	9823456789	Patna	Ranchi
Ganga Travels	Ajay Verma	9898765432	Lucknow	Kanpur
Surat Express	Vishal Patel	9807654321	Ahmedabad	Vadodara

- ❖ To find all bookings where the passenger's name contains 'Sharma'.

```
SELECT * FROM Bookings
```

```
WHERE passenger_name LIKE '%Sharma%';
```

Output :-

Output

booking_id	passenger_name	bus_id	booking_date	seat_number
6	Priya Sharma	6	2024-11-15	20

- ❖ To get the details of bookings along with bus name, driver name, and route information (source, destination).

```
SELECT Bo.booking_id, Bo.passenger_name, Bo.booking_date, B.bus_name,
D.driver_name, R.source, R.destination
FROM Bookings Bo
JOIN Bus B ON Bo.bus_id = B.bus_id
JOIN Drivers D ON B.bus_id = D.bus_id
JOIN Route R ON B.route_id = R.route_id;
```

Output :-

Output

booking_id	passenger_name	booking_date	bus_name	driver_name	source	destination
2	Sita Devi	2024-11-11	Golden Travels	Rajesh Sharma	Delhi	Agra
3	Vijay Kumar	2024-11-12	Bengaluru Express	Suresh Babu	Chennai	Bengaluru
4	Nisha Agarwal	2024-11-13	Kolkata Deluxe	Manoj Verma	Kolkata	Durgapur
5	Ajay Singh	2024-11-14	Telangana Super	Kiran Reddy	Hyderabad	Warangal
6	Priya Sharma	2024-11-15	Mewar Express	Vikas Pandey	Jaipur	Udaipur
7	Sunita Patel	2024-11-16	MP Travels	Sunil Yadav	Bhopal	Indore
8	Deepak Gupta	2024-11-17	Patna-Ranchi Express	Prakash Kumar	Patna	Ranchi
9	Neha Rai	2024-11-18	Ganga Travels	Ajay Verma	Lucknow	Kanpur
10	Manoj Mishra	2024-11-19	Surat Express	Vishal Patel	Ahmedabad	Vadodara

- ❖ To find buses that have not been booked (i.e., no bookings exist for these buses).

```
SELECT B.bus_name
FROM Bus B
LEFT JOIN Bookings Bo ON B.bus_id = Bo.bus_id
WHERE Bo.booking_id IS NULL;
```

Output :-

Output

bus_name
Shivneri Express

- ❖ To find the top 3 routes with the most number of bookings.

```
SELECT R.source, R.destination, COUNT(Bo.booking_id) AS
total_bookings
FROM Route R
JOIN Bus B ON R.route_id = B.route_id
JOIN Bookings Bo ON B.bus_id = Bo.bus_id
GROUP BY R.source, R.destination
ORDER BY total_bookings DESC
```

LIMIT 3;

Output :-

Output

source	destination	total_bookings
Ahmedabad	Vadodara	1
Bhopal	Indore	1
Chennai	Bengaluru	1

- ❖ To get bus schedules along with bus name and route information (source, destination).

```

SELECT S.schedule_id, B.bus_name, R.source, R.destination, S.departure_time,
S.arrival_time, S.date_of_travel
FROM Schedule S
JOIN Bus B ON S.bus_id = B.bus_id
JOIN Route R ON B.route_id = R.route_id;

```

Output :-

Output						
schedule_id	bus_name	source	destination	departure_time	arrival_time	date_of_travel
2	Golden Travels	Delhi	Agra	07:30:00	14:00:00	2024-11-11
3	Bengaluru Express	Chennai	Bengaluru	10:00:00	17:00:00	2024-11-12
4	Kolkata Deluxe	Kolkata	Durgapur	11:00:00	15:00:00	2024-11-13
5	Telangana Super	Hyderabad	Warangal	12:00:00	15:30:00	2024-11-14
6	Mewar Express	Jaipur	Udaipur	13:00:00	18:00:00	2024-11-15
7	MP Travels	Bhopal	Indore	14:00:00	18:30:00	2024-11-16
8	Patna-Ranchi Express	Patna	Ranchi	15:00:00	21:00:00	2024-11-17
9	Ganga Travels	Lucknow	Kanpur	16:00:00	18:00:00	2024-11-18
10	Surat Express	Ahmedabad	Vadodara	17:00:00	19:30:00	2024-11-19

- ❖ To get the bus with the highest number of schedules.

```

SELECT B.bus_name, COUNT(S.schedule_id) AS schedule_count
FROM Bus B
JOIN Schedule S ON B.bus_id = S.bus_id
GROUP BY B.bus_name
ORDER BY schedule_count DESC
LIMIT 1;

```

Output :-

Output	
bus_name	schedule_count
Telangana Super	1

- ❖ To get the total number of bus schedules for each route (grouped by source and destination).

```
SELECT R.source, R.destination, COUNT(S.schedule_id) AS total_schedules
FROM Schedule S
JOIN Bus B ON S.bus_id = B.bus_id
JOIN Route R ON B.route_id = R.route_id
GROUP BY R.source, R.destination;
```

Output :-

Output		
source	destination	total_schedules
Ahmedabad	Vadodara	1
Bhopal	Indore	1
Chennai	Bengaluru	1
Delhi	Agra	1
Hyderabad	Warangal	1
Jaipur	Udaipur	1
Kolkata	Durgapur	1
Lucknow	Kanpur	1
Patna	Ranchi	1

- ❖ To get the schedules of buses traveling between two cities, e.g., 'Delhi' and 'Agra'.

```
SELECT S.schedule_id, B.bus_name, S.departure_time, S.arrival_time
FROM Schedule S
JOIN Bus B ON S.bus_id = B.bus_id
JOIN Route R ON B.route_id = R.route_id
WHERE R.source = 'Delhi' AND R.destination = 'Agra';
```

Output :-

Output		
source	destination	total_schedules
Ahmedabad	Vadodara	1
Bhopal	Indore	1
Chennai	Bengaluru	1
Delhi	Agra	1
Hyderabad	Warangal	1
Jaipur	Udaipur	1
Kolkata	Durgapur	1
Lucknow	Kanpur	1
Patna	Ranchi	1

- ❖ To get the schedule details for the bus with the latest departure time.

```
SELECT * FROM Schedule
```

```
WHERE departure_time = (SELECT MAX(departure_time) FROM Schedule);
```

Output :-

Output				
schedule_id	bus_id	departure_time	arrival_time	date_of_travel
10	10	17:00:00	19:30:00	2024-11-19

- ❖ To get bus names from both the Bus table and Bookings table (removes duplicates).

```
SELECT bus_name FROM Bus
```

```
UNION
```

```
SELECT DISTINCT bus_name FROM Bookings B
```

```
JOIN Bus B2 ON B.bus_id = B2.bus_id;
```

Output :-

bus_name
Bengaluru Express
Ganga Travels
Golden Travels
Kolkata Deluxe
MP Travels
Mewar Express
Patna-Ranchi Express
Shivneri Express
Surat Express
Telangana Super

❖ To get bus names that has both bookings and schedules.

```
SELECT bus_name FROM Bus  
INTERSECT  
SELECT bus_name FROM Bookings B  
JOIN Bus B2 ON B.bus_id = B2.bus_id;
```

Output :-

bus_name
Bengaluru Express
Ganga Travels
Golden Travels
Kolkata Deluxe
MP Travels
Mewar Express
Patna-Ranchi Express
Surat Express
Telangana Super

- ❖ To get bus names that are available in the Bus table but have no bookings.

```
SELECT bus_name FROM Bus
EXCEPT
SELECT bus_name FROM Bookings B
JOIN Bus B2 ON B.bus_id = B2.bus_id;
```

Output :-

Output	
	bus_name
	Shivneri Express

- ❖ To find out how many buses are available for each route, query the RouteBusAvailabilityView.

This will return a summary of each route with the number of buses available.

```
CREATE VIEW RouteBusAvailabilityView AS
SELECT
R.source,
R.destination,
COUNT(B.bus_id) AS number_of_buses
FROM Route R
JOIN Bus B ON R.route_id = B.route_id
GROUP BY R.source, R.destination;
```

```
SELECT * FROM RouteBusAvailabilityView;
```

Output :-

Output		
source	destination	number_of_buses
Ahmedabad	Vadodara	1
Bhopal	Indore	1
Chennai	Bengaluru	1
Delhi	Agra	1
Hyderabad	Warangal	1
Jaipur	Udaipur	1
Kolkata	Durgapur	1
Lucknow	Kanpur	1
Mumbai	Pune	1
Patna	Ranchi	1

❖ To convert bus names to uppercase.

```
SELECT bus_name, UPPER(bus_name) AS uppercase_bus_name  
FROM Bus;
```

Output :-

Output	
bus_name	uppercase_bus_name
Shivneri Express	SHIVNERI EXPRESS
Golden Travels	GOLDEN TRAVELS
Bengaluru Express	BENGALURU EXPRESS
Kolkata Deluxe	KOLKATA DELUXE
Telangana Super	TELANGANA SUPER
Mewar Express	MEWAR EXPRESS
MP Travels	MP TRAVELS
Patna-Ranchi Express	PATNA-RANCHI EXPRESS
Ganga Travels	GANGA TRAVELS
Surat Express	SURAT EXPRESS

- ❖ To get the first 5 characters of a driver's name.

```
SELECT driver_name, SUBSTRING(driver_name, 1, 5) AS
short_driver_nameFROM Drivers;
```

Output :-

Output	
driver_name	short_driver_name
Ravi Kumar	Ravi
Rajesh Sharma	Rajes
Suresh Babu	Sures
Manoj Verma	Manoj
Kiran Reddy	Kiran
Vikas Pandey	Vikas
Sunil Yadav	Sunil
Prakash Kumar	Praka
Ajay Verma	Ajay
Vishal Patel	Visha

- ❖ To round the distance of a route to the nearest whole number.

```
SELECT source, destination, distance, ROUND(distance) AS rounded_distanceFROM
Route;
```

Output :-

Output			
source	destination	distance	rounded_distance
Mumbai	Pune	150	150
Delhi	Agra	230	230
Chennai	Bengaluru	350	350
Kolkata	Durgapur	170	170
Hyderabad	Warangal	150	150
Jaipur	Udaipur	400	400
Bhopal	Indore	200	200
Patna	Ranchi	250	250
Lucknow	Kanpur	80	80
Ahmedabad	Vadodara	110	110

❖ To get the total capacity of all buses.

```
SELECT SUM(capacity) AS total_capacity  
FROM Bus;
```

Output :-

Output

total_capacity
488