SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)
(Established under section 3 of the UGC Act 1956)
Re - accredited by NAAC with 'A' Grade
Founder: Prof. Dr. S. B. Mujumdar, M.Sc.,Ph.D. (Awarded Padma Bhushan and Padma Shri by President of India)

Database Management Systems Lab

Project Phase: 2

Group Members:

Sr.No.	Name	PRN	Branch	Division
1	Jay Gondaliya	22070122088	CS	B1
2	K.Rishik Kumar Reddy	22070122091	CS	B1
3	Kush Deo Shukla	22070122103	CS	B1
4	Manas Dhanpawde	22070122108	CS	B1

Title of the project:

Online Travel Agency Management

About the project:

Online travel agencies (OTAs) play a vital role in the modern travel industry, offering travelers a convenient platform to book flights, hotels, and activities. Effective management of these platforms is crucial for providing seamless booking experiences and maximizing revenue.

Our project focuses on developing a comprehensive management system for an online travel agency. We aim to streamline reservation processes, manage inventory efficiently, integrate secure payment gateways, enhance customer relationship management, and optimize user interface and experience.

Effective management of online travel agencies is essential for meeting the demands of modern travelers. Our project aims to provide a comprehensive solution that optimizes every aspect of OTA management, from booking to analytics, ultimately improving the travel experience for users and boosting business growth.

Addressed paint points:

Addressing pain points for managing an online travel agency involves understanding the challenges faced by both the agency and its customers. Here are some common pain points and how to address them:

1. **Complex Booking Process**: Customers often find the booking process complicated and time-consuming.
- Address by simplifying the booking process with an intuitive user interface.
- Implement a streamlined booking system with clear steps and minimal form fields.
- Offer a guest checkout option for quick bookings without registration.
1. **Limited Search Options:** Users may struggle to find relevant travel options due to limited search filters.
- Provide advanced search options such as flexible dates, budget ranges, and specific amenities.
- Implement robust filtering options for destinations, accommodation types, transportation preferences, etc.
- Include a map-based search feature for visual exploration of destinations and nearby attractions.
2. **Information Overload**: Customers can feel overwhelmed with excessive information about destinations, accommodations, and activities.
- Offer personalized recommendations based on customer preferences and past bookings.
- Provide concise, relevant information with clear descriptions, high-quality images, and user reviews.
- Utilize chatbots or live chat support to assist customers in finding the information they need quickly.
3. **Payment Security Concerns**: Security concerns about online payments can deter customers from booking.
- Ensure the website is secure with SSL encryption and PCI compliance for handling payment information.
- Offer multiple payment options including credit/debit cards, PayPal, and secure online payment gateways.
- Clearly communicate the security measures taken to protect customer data.
4. **Lack of Trust:** Customers may be hesitant to book through a new or lesser-known online travel agency.
- Build trust through transparent pricing, detailed terms and conditions, and clear refund policies.
- Showcase customer testimonials and reviews to demonstrate positive experiences.
- Partner with reputable travel providers and display relevant certifications and affiliations.
5. **Inadequate Customer Support:** Poor customer support can lead to dissatisfaction and lost business.
- Provide multiple channels for customer support, including phone, email, and live chat.
- Offer 24/7 customer support, especially for urgent issues or bookings.

- Invest in training for support staff to ensure knowledgeable and helpful assistance.

6. **Mobile Optimization**: Inadequate mobile experience can lead to frustration for users trying to book on smartphones or tablets.

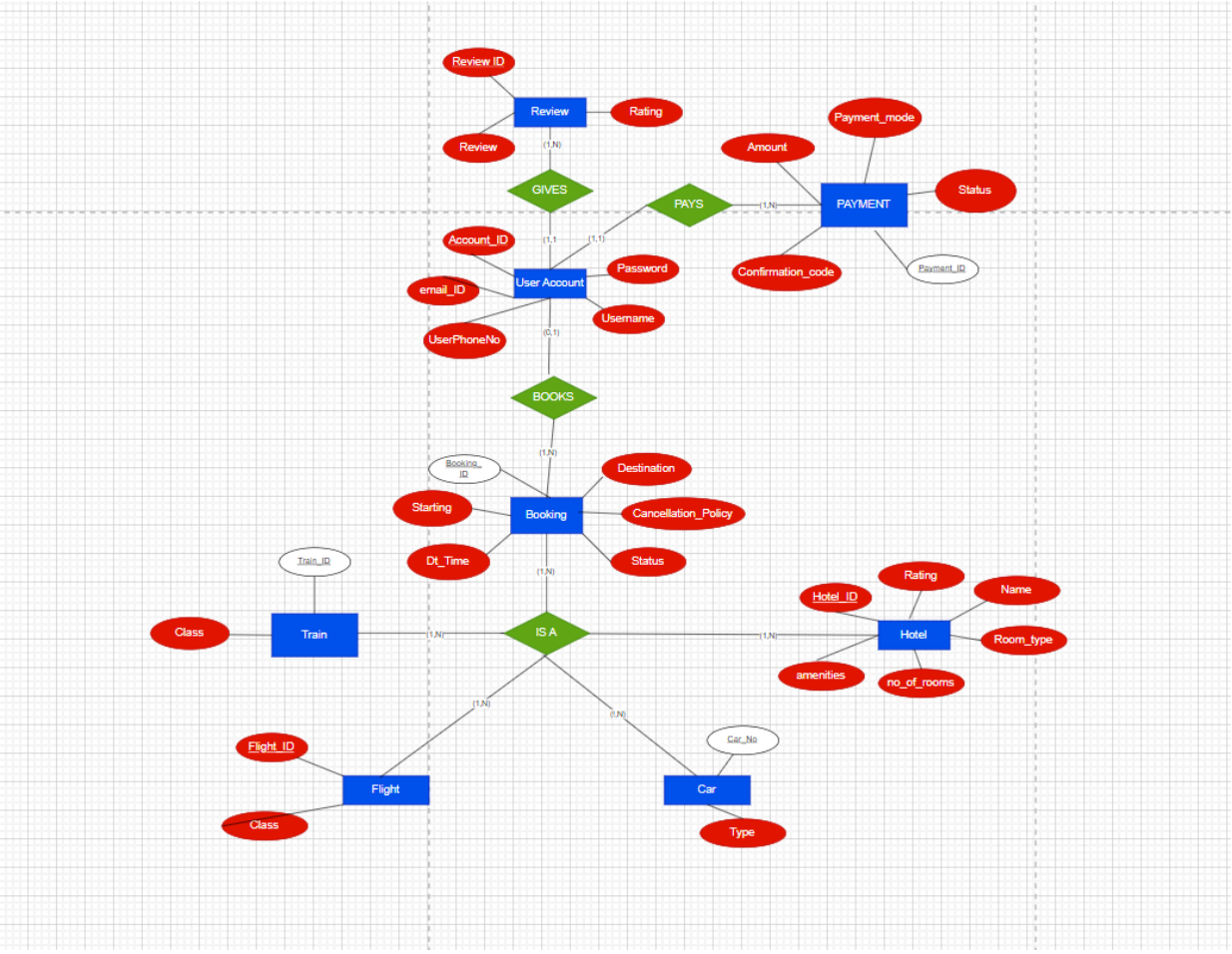
- Develop a mobile-responsive website or a dedicated mobile app for easy booking on all devices.
- Optimize page load times and simplify navigation for mobile users.
- Implement mobile-specific features like one-click booking and push notifications for updates.

7. **Inventory Management:** Managing inventory across various destinations and accommodations can be challenging.

- Utilize a robust inventory management system to track availability in real-time.
- Implement automated alerts for low inventory or sold-out items.
- Integrate with property management systems (PMS) and channel managers for seamless updates across platforms.

By addressing these pain points, an online travel agency can improve user experience, increase customer satisfaction, and ultimately drive more bookings.

ER diagram:


Review ID
Review
Rating
Review
(1,N)
GIVES
PAYS
(1,N)
Payment_mode
Amount
PAYMENT
Status
Account_ID
(1,1)
(1,1)
Password
User Account
email_ID
Username
UserPhoneNo
(0,1)
Confirmation_code
Payment_ID
BOOKS
(1,N)
Booking_ID
Destination
Starting
Booking
Cancellation_Policy
Dt_Time
Status
(1,N)
Train_ID
Class
Train
(1,N)
IS A
(1,N)
Hotel_ID
Rating
Name
Hotel
Room_type
amenities
no_of_rooms
(1,N)
(1,N)
Flight_ID
Flight
Class
Car_No
Car
Type

Anomalies:
A database anomaly is a fault within a database, which can occur because of poor planning or when everything is stored in a flat database. These anomalies naturally occur and result in data that does not match the real-world the database purports to represent.

Anomalies occurring in our code:

1.Delete Anomaly:

Similarly, if a user decides to delete their account from the User_Account table, there's a risk of orphaned records in other tables. For instance, if the user has made hotel bookings or car bookings, and those records are not automatically deleted when the user account is deleted, it could lead to data inconsistencies and orphaned records in the HotelBookings and CarBookings tables.

Account_ID	Username	Email_id	Phone_number	Password
1	user1	user1@gmail.com	1234567890	password1
2	user2	user2@gmail.com	9876543210	password2
3	user3	user3@gmail.com	5551234567	password3
4	user4	user4@gmail.com	9998887776	password4
5	user5	user5@gmail.com	5555555555	password5
6	user6	user6@gmail.com	6666666666	password6
7	user7	user7@gmail.com	7777777777	password7
8	user8	user8@gmail.com	8888888888	password8
9	rishik	rishik@gmail.com	1234567890	123
10	boy	boy2004@gmail.com	0000000001	Boy2004
11	gal	gal2343@gmail.com	9191919191	Gal2343
12	Rishik ku...	reddyrishik2004@g...	9347994965	Reddyrish...
13	Kush	kush2004@gmail.com	8989898970	Kush2004
14	man	man2004@gmail.com	9837847839	Man2004
15	reddy	reddy123@gmail.com	8989898989	Reddy2004
NULL	NULL	NULL	NULL	NULL

2. Update Anomaly:

The name of the hotel with Hotel_id 101 is updated from "Tranquil Haven Hotel" to "Serene Haven Hotel". If there are hotel bookings stored in the HotelBookings table referencing the hotel by its name rather than its Hotel_id, those bookings won't automatically reflect the name change. As a result, the information in the HotelBookings table would become inconsistent with the updated name in the Hotel table.

Hotel_id	Name	Rating	Amenities	Location	Price	Num_rooms_available	Type
101	Tranquil Haven Hotel	4.2	WiFi, Breakfast	Pune	500	47	AC
202	Serenity Lodge	4.5	Pool, Gym	Anand	500	60	Non-AC
303	Blissful Retreat Inn	4.8	Spa, Room Service	Hyderabad	500	68	AC
404	Oasis Heights Resort	3.9	Parking	Lucknow	500	40	AC
505	Peaceful Valley Villa	4.3	WiFi, Breakfast	Mumbai	600	45	AC
606	Tranquility Suites	4.7	Gym, Pool	Chennai	550	55	Non-AC
707	Harmony Inn	4.1	Room Service	Delhi	450	65	Non-AC
808	Sunny Shores Hotel	4.4	WiFi, Parking	Bangalore	700	30	AC
909	Calm Breeze Resort	4.6	Spa, Gym	Jaipur	650	35	AC
1010	Tranquil Oaks Lodge	4	Pool	Kolkata	550	50	Non-AC
1111	Soothing Sands Resort	4.9	WiFi, Room Service	Goa	750	25	AC
1212	Blissful Bay Hotel	4.2	Gym	Ahmedabad	600	40	Non-AC
1313	Tranquility Retreat	4.5	WiFi, Breakfast	Surat	550	55	AC

Functional dependencies:
Functional dependency refers to a relationship that is present between attributes of any table that are dependent on each other. It helps in avoiding data redundancy and getting to know more about bad designs.

Functional dependencies of schema occurring in our code:

1. Institute Library:
Entities (Lid, Lname, city, area, phone)
Lid -> Lname, City, Area, Phone
Lname ->Area, Phone

2. Employee:
Entities (Eid, ename, email, salary, phone)
Eid -> ename, email, salary, Phone
3. Members:
Entities (Mid, mname, phone, address)
Mid -> mname, address, Phone
4. Books:

Entities (ISBN, Bname, Noof_copies, Availibility)
ISBN -> Bname, Noof_copies, Availibility

5. Author:
Entities (Aid, Aname, email)
Aid -> Aname, email

6. Publisher:
Entities (Pid, Pname, email)
Pid -> Pname, email

7. Sellers:
Entities (Sid, Sname, City)
Sid -> Sname, City

8. Sells:
Entities (Ssid, sid, pid)
Ssid -> Sid, Pid

9. Issues:
Entities (Id, mid, isbn, issue_date, return_date)
ID -> mid, isbn, issue_date, return_date

Normalization

A normalization procedure, which combines and splits tables, is usually sufficient to remove this. By normalizing the database, we reduce the likelihood of creating tables that generate anomalies. Normalization is necessary if you do not do it then the overall integrity of the data stored in the database will eventually degrade. Specifically, this is due to data anomalies.

Types of normalization

1. **Unnormalized form or UNF:**
   It is the simplest database model also known as non-first normal form (NF2).A UNF model will suffer problems like data redundancy thus it lacks the efficiency of database normalization.

2. **First Normal Form or 1NF:**
   A relation is in first normal form only if the relational table doesn't contain any multivalued attribute but contains only single-valued attributes.
3. **Second Normal Form or 2NF:**
   A relation is in second normal form if:
   • It is in first normal form or 1NF
   • It doesn't contain any partial dependencies. (It shouldn't have any non-prime attribute which is functionally dependent on any proper subset of the candidate key of the relation.).
4. **Third Normal Form or 3NF:**
   Let R be the relational schema, X->Y any non-trivial functional dependency over R is in 3NF if:

- R should be in 2NF
- X should be candidate key or superkey, or
- Y should be prime attribute

Normalization occurring in our code:

1. First, using *1NF* we will have only one data per cell. By doing this we achieve atomicity. Like the below table:

Train_id	Name	starting_location	starting_date	starting_time	ending_location	ending_date	ending_time	Cancellation_Policy	sleeper_price	ac_2_tier	ac_first_	available_seats_sleepe
12345	Rajdhani Exp...	Mumbai	2024-04-18	08:00:00	Delhi	2024-04-19	17:00:00	Flexible	500	1000	1500	50
13457	Jan Shatabdi	Lucknow	2024-04-29	19:00:00	Patna	2024-04-30	06:00:00	Strict	470	970	1470	55
13579	Garib Rath	Chennai	2024-04-22	12:00:00	Mumbai	2024-04-23	12:00:00	Strict	450	900	1400	60
24680	Humsafar Ex...	Delhi	2024-04-23	13:00:00	Chennai	2024-04-25	13:00:00	Flexible	550	1050	1550	45
36987	Kavi Guru Ex...	Patna	2024-05-02	22:00:00	Jaipur	2024-05-03	10:00:00	Strict	490	990	1490	53
42068	Express Special	Jaipur	2024-04-30	20:00:00	Chandigarh	2024-05-01	06:00:00	Moderate	500	1000	1500	50
54321	Shatabdi Exp...	Pune	2024-04-19	09:00:00	Hyderabad	2024-04-19	20:00:00	Strict	500	1000	1500	50
67890	Duronto Expr...	Bangalore	2024-04-20	10:00:00	Kolkata	2024-04-21	06:00:00	Flexible	500	1000	1500	50
75319	Mahamana E...	Varanasi	2024-04-26	16:00:00	New Delhi	2024-04-27	09:00:00	Flexible	550	950	1450	65
78695	Superfast Ex...	Bhopal	2024-05-01	21:00:00	Lucknow	2024-05-02	09:00:00	Flexible	480	980	1480	58
86420	Tejas Express	Goa	2024-04-24	14:00:00	Mumbai	2024-04-24	20:00:00	Moderate	600	1100	1600	55
86459	Sampark Kranti	Hyderabad	2024-04-27	17:00:00	Jaipur	2024-04-28	17:00:00	Moderate	525	1025	1525	48

2. 2nd normal form :
Now we'll check for 2NF.
The non key attributes for table HotelBooking are -
Name,Rating,Amenities,Location,Price,num_of_rooms_booked,type,Check_in,Check_out, Username,Email_id,Phone_number,Password,Hotel_name,Hotel_rating,Hotel_location, Hotel_price,Hotel_num_rooms_available,Hotel_type,Hotel_Check_in Hotel_Check_out are fully dependent on BookingID.
Hotel-related details are repeated for each booking. It also leads to anomalies like update anomalies, where if a hotel's information changes, we have to update multiple records in this table.To resolve this we can create separate tables for users and hotels, and establish relationships using foreign keys, Hence it does not fulfill 2nd Normal Form.

To do so, you will create 2 tables like this:

1.Separate HotelBookings and User_Account tables.
2.Create a foreign key relationship between HotelBookings and Hotel tables using Hotel_id.
3.Create a Hotel table to store hotel-related information.
4.Move hotel-related information (Hotel_name, Hotel_rating, Hotel_location, Hotel_price, Hotel_num_rooms_available, Hotel_type, Hotel_Check_in, Hotel_Check_out) to the Hotel table.

1) *User Account Table and Entries:*

```sql
CREATE TABLE User_Account (
    Account_ID INT PRIMARY KEY auto_increment ,
    Username VARCHAR(255),
    Email_id VARCHAR(255),
    Phone_number VARCHAR(20),
    Password VARCHAR(255)
);

select * from User_Account;
```

Result Grid | Filter Rows: | Edit: | Export

Account_ID	Username	Email_id	Phone_number	Password
1	user1	user1@gmail.com	1234567890	password1
2	user2	user2@gmail.com	9876543210	password2
3	user3	user3@gmail.com	5551234567	password3
4	user4	user4@gmail.com	9998887776	password4
5	user5	user5@gmail.com	5555555555	password5

```sql
CREATE TABLE Hotel (
    Hotel_id INT PRIMARY KEY,
    Name VARCHAR(255),
    Rating FLOAT,
    Amenities TEXT,
    Location VARCHAR(255),
    Price INT,
    Num_rooms_available INT,
    Type VARCHAR(20),
    Check_in TIME,
    Check_out TIME
);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Hotel_id	Name	Rating	Amenities	Location	Price	Num_rooms_available	Type	Check_in	Check_out
101	Tranquil Haven Hotel	4.2	WiFi, Breakfast	Pune	500	47	AC	NULL	NULL
202	Serenity Lodge	4.5	Pool, Gym	Anand	500	60	Non-AC	NULL	NULL
303	Blissful Retreat Inn	4.8	Spa, Room Service	Hyderabad	500	68	AC	NULL	NULL
404	Oasis Heights Resort	3.9	Parking	Lucknow	500	40	AC	NULL	NULL
505	Peaceful Valley Villa	4.3	WiFi, Breakfast	Mumbai	600	45	AC	NULL	NULL

```sql
CREATE TABLE HotelBookings (
    HotelBooking_id INT AUTO_INCREMENT PRIMARY KEY,
    Hotel_id int,
    FOREIGN KEY (Hotel_id) REFERENCES Hotel(Hotel_id),
    Name VARCHAR(255) NOT NULL,
    Rating DECIMAL(2,1) NOT NULL,
    Amenities TEXT,
    Location VARCHAR(255) NOT NULL,
    Price DECIMAL(10, 2) NOT NULL,
    num_of_rooms_boked INT,
    type VARCHAR(20),
    Check_in TIME,
    Check_out TIME
);
select * from HotelBookings;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

HotelBooking_id	Hotel_id	Name	Rating	Amenities	Location	Price	num_of_rooms_boked	type	Check_in	Check_out
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3. For *3NF*, we first check for transitive functional dependencies. There should be no transitive dependency for non-prime attributes, which indicates that non-prime attributes (which are not a part of the candidate key) should not depend on other non-prime attributes in a table. In both the tables, none of the non-prime attributes depend on each other, and only the primary key. Hence it is already in 3NF.

Thus, we have normalized this table.

What is MySQL:
MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL).

Implementation of the tables in MySQL:

```
create database travel_agency;
use travel_agency;
```

```
CREATE TABLE User_Account (
   Account_ID INT PRIMARY KEY auto_increment ,
   Username VARCHAR(255),
   Email_id VARCHAR(255),
   Phone_number VARCHAR(20),
   Password VARCHAR(255)
);
```

```
INSERT INTO User_Account (account_id, Username, Email_id, Phone_number, Password)
VALUES
(1, 'user1', 'user1@gmail.com', '1234567890', 'password1'),
(2, 'user2', 'user2@gmail.com', '9876543210', 'password2'),
(3, 'user3', 'user3@gmail.com', '5551234567', 'password3'),
(4, 'user4', 'user4@gmail.com', '9998887776', 'password4'),
(5, 'user5', 'user5@gmail.com', '5555555555', 'password5'),
(6, 'user6', 'user6@gmail.com', '6666666666', 'password6'),
(7, 'user7', 'user7@gmail.com', '7777777777', 'password7'),
(8, 'user8', 'user8@gmail.com', '8888888888', 'password8');
```

```
CREATE TABLE flight (
   flight_id INT PRIMARY KEY,
   Airline VARCHAR(255) NOT NULL,
   Starting_Location VARCHAR(255) NOT NULL,
   Destination VARCHAR(255) NOT NULL,
   Cancellation_Policy VARCHAR(255),
   Starting_Dt DATE,
   starting_time time,
   Ending_dt DATE,
   ending_time time,
   economy_price int,
```

```
  business_price int,
   first_class_price int,
   available_seats_economy int,
   available_seats_business int,
   available_seats_first_class int
);
INSERT INTO Flight (flight_id, Airline, Starting_Location, Destination, Cancellation_Policy,
Starting_Dt, starting_time, ending_dt, ending_time, economy_price, business_price,
first_class_price, available_seats_economy, available_seats_business,
available_seats_first_class)
VALUES
(1, 'Air India', 'Delhi', 'Mumbai', 'Flexible', '2024-03-15', '08:00:00', '2024-03-15', '12:00:00',
500, 1000, 1500, 25, 15, 10),
(2, 'IndiGo', 'Ahmedabad', 'Pune', 'Strict', '2024-03-20', '10:00:00', '2024-03-20', '12:00:00',
500, 1000, 1500, 25, 15, 10),
(3, 'SpiceJet', 'Bangalore', 'Chandigarh', 'Flexible', '2024-03-25', '14:00:00', '2024-03-25',
'17:00:00', 500, 1000, 1500, 25, 15, 10),
(4, 'GoAir', 'Guwahati', 'Delhi', 'Moderate', '2024-03-30', '12:00:00', '2024-03-30', '16:00:00',
500, 1000, 1500, 25, 15, 10),
(5, 'Vistara', 'Mumbai', 'Kolkata', 'Moderate', '2024-04-02', '09:00:00', '2024-04-02', '12:00:00',
600, 1100, 1600, 30, 20, 12),
(6, 'AirAsia', 'Chennai', 'Delhi', 'Flexible', '2024-04-05', '07:00:00', '2024-04-05', '10:00:00',
550, 1050, 1550, 28, 16, 11),
(7, 'Air India', 'Delhi', 'Bangalore', 'Strict', '2024-04-10', '17:00:00', '2024-04-10', '20:00:00',
580, 1150, 1650, 26, 18, 9),
(8, 'IndiGo', 'Mumbai', 'Jaipur', 'Moderate', '2024-04-15', '11:00:00', '2024-04-15', '13:00:00',
570, 1070, 1570, 24, 14, 13),
(9, 'SpiceJet', 'Hyderabad', 'Mumbai', 'Flexible', '2024-04-18', '16:00:00', '2024-04-18',
'18:30:00', 540, 1040, 1540, 27, 17, 14),
(10, 'GoAir', 'Kochi', 'Delhi', 'Strict', '2024-04-22', '18:00:00', '2024-04-22', '21:00:00', 520,
1020, 1520, 23, 13, 15),
(11, 'Vistara', 'Jaipur', 'Chennai', 'Moderate', '2024-04-25', '20:00:00', '2024-04-25', '23:00:00',
510, 1010, 1510, 22, 12, 11),
(12, 'AirAsia', 'Delhi', 'Goa', 'Flexible', '2024-04-28', '06:00:00', '2024-04-28', '08:30:00', 600,
1200, 1700, 20, 10, 12),
(13, 'Air India', 'Mumbai', 'Chennai', 'Strict', '2024-05-01', '13:00:00', '2024-05-01', '15:30:00',
550, 1100, 1600, 29, 19, 10),
(14, 'IndiGo', 'Kolkata', 'Mumbai', 'Moderate', '2024-05-05', '15:00:00', '2024-05-05',
'18:00:00', 530, 1030, 1530, 21, 11, 9),
(15, 'SpiceJet', 'Delhi', 'Ahmedabad', 'Flexible', '2024-05-08', '12:00:00', '2024-05-08',
'14:00:00', 560, 1060, 1560, 32, 22, 13);


CREATE TABLE car (
```

```
 car_no VARCHAR(255) PRIMARY KEY,
   seater INT,
   driver_name VARCHAR(255),
   starting_location VARCHAR(255),
   destination VARCHAR(255),
   driver_status VARCHAR(255),
   price varchar(255)
);

INSERT INTO car (car_no, seater, driver_name, starting_location, destination, driver_status,
price)
VALUES
('ABC123', 4, 'Manas Dhanpawde', 'Mumbai', 'Pune', 'available', 500),
('DEF456', 5, 'Rishik reddy', 'Bangalore', 'Hyderabad', 'available', 500),
('GHI789', 7, 'Jay Gondaliya', 'Chennai', 'Anand', 'available', 500),
('JKL012', 4, 'Kushdeo Shukla', 'Kolkata', 'Lucknow', 'available', 500),
('MNO345', 6, 'Kushay reddypawde', 'Delhi', 'Ahmedabad', 'available', 500),
('PQR678', 4, 'Rohan Sharma', 'Jaipur', 'Indore', 'available', 550),
('STU901', 5, 'Shubham Singh', 'Bhopal', 'Patna', 'available', 600),
('VWX234', 7, 'Varun Khanna', 'Nagpur', 'Bhubaneswar', 'available', 700),
('YZA567', 4, 'Yash Gupta', 'Indore', 'Jaipur', 'available', 550),
('BCD890', 6, 'Bhavya Patel', 'Surat', 'Ahmedabad', 'available', 600),
('MNO678', 6, 'Kushay reddy', 'Ahmedabad', 'Delhi', 'available', 500),
('PQR910', 4, 'Rohit Sharma', 'Indore', 'Jaipur', 'available', 550),
('STU111', 5, 'Shubham Patel', 'Patna', 'Bhopal', 'available', 600),
('VWX121', 7, 'Varun Gupta', 'Bhubaneswar', 'Nagpur', 'available', 700),
('YZA314', 4, 'Yash Singh', 'Jaipur', 'Indore', 'available', 550),
('BCD516', 6, 'Bhavya Khanna', 'Ahmedabad', 'Surat', 'available', 600);

CREATE TABLE Hotel (
   Hotel_id INT PRIMARY KEY,
   Name VARCHAR(255),
   Rating FLOAT,
   Amenities TEXT,
   Location VARCHAR(255),
   Price INT,
   Num_rooms_available INT,
   Type VARCHAR(20),
   Check_in TIME,
   Check_out TIME
);


INSERT INTO Hotel (Hotel_id, Name, Rating, Amenities, location, price,
```

```
num_rooms_available, type)
VALUES
(101, 'Tranquil Haven Hotel', 4.2, 'WiFi, Breakfast', 'Pune', 500, 50, 'AC'),
(202, 'Serenity Lodge', 4.5, 'Pool, Gym', 'Anand', 500, 60, 'Non-AC'),
(303, 'Blissful Retreat Inn', 4.8, 'Spa, Room Service', 'Hyderabad', 500, 70, 'AC'),
(404, 'Oasis Heights Resort', 3.9, 'Parking', 'Lucknow', 500, 40, 'AC'),
(505, 'Peaceful Valley Villa', 4.3, 'WiFi, Breakfast', 'Mumbai', 600, 45, 'AC'),
(606, 'Tranquility Suites', 4.7, 'Gym, Pool', 'Chennai', 550, 55, 'Non-AC'),
(707, 'Harmony Inn', 4.1, 'Room Service', 'Delhi', 450, 65, 'Non-AC'),
(808, 'Sunny Shores Hotel', 4.4, 'WiFi, Parking', 'Bangalore', 700, 30, 'AC'),
(909, 'Calm Breeze Resort', 4.6, 'Spa, Gym', 'Jaipur', 650, 35, 'AC'),
(1010, 'Tranquil Oaks Lodge', 4.0, 'Pool', 'Kolkata', 550, 50, 'Non-AC'),
(1111, 'Soothing Sands Resort', 4.9, 'WiFi, Room Service', 'Goa', 750, 25, 'AC'),
(1212, 'Blissful Bay Hotel', 4.2, 'Gym', 'Ahmedabad', 600, 40, 'Non-AC'),
(1313, 'Tranquility Retreat', 4.5, 'WiFi, Breakfast', 'Surat', 550, 55, 'AC'),
(1414, 'Calm Waters Inn', 3.8, 'Room Service, Parking', 'Nagpur', 500, 60, 'Non-AC'),
(1515, 'Serenity Suites', 4.7, 'Spa', 'Bhopal', 650, 35, 'AC'),
(1616, 'Tranquil Garden Resort', 4.3, 'WiFi, Gym', 'Indore', 600, 40, 'Non-AC'),
(1717, 'Quiet Haven Hotel', 4.6, 'Breakfast', 'Gurgaon', 700, 30, 'AC'),
(1818, 'Calm Retreat Inn', 4.4, 'Pool, WiFi', 'Chandigarh', 750, 25, 'Non-AC');
```

```
CREATE TABLE HotelBookings (
   HotelBooking_id INT AUTO_INCREMENT PRIMARY KEY,
   Hotel_id int,
   FOREIGN KEY (Hotel_id) REFERENCES Hotel(Hotel_id),
   Name VARCHAR(255) NOT NULL,
   Rating DECIMAL(2,1) NOT NULL,
   Amenities TEXT,
   Location VARCHAR(255) NOT NULL,
   Price DECIMAL(10, 2) NOT NULL,
   num_of_rooms_boked INT,
   type VARCHAR(20),
   Check_in TIME,
   Check_out TIME

);
```

```
CREATE TABLE CarBookings (
   carbookingId int  AUTO_INCREMENT PRIMARY KEY,
   car_no VARCHAR(255),
   seater INT NOT NULL,
   driver_name VARCHAR(100),
   starting_location VARCHAR(255),
   destination VARCHAR(255),
   driver_status VARCHAR(50),
   price varchar(255)

);

CREATE TABLE train (
   Train_id INT PRIMARY KEY ,
   Name VARCHAR(255) NOT NULL,
   starting_location varchar(100),
   starting_date varchar(100),
   starting_time varchar(100),
   ending_location varchar(100),
   ending_date varchar(100),
   ending_time varchar(100),
   Cancellation_Policy VARCHAR(255),

   sleeper_price int,
   ac_2_tier_price int,
   ac_first_class_price int,
   available_seats_sleeper int,
   available_seats_ac_2_tier int,
   available_seats_ac_first_class int
);


INSERT INTO Train (Train_id, Name, starting_location, starting_date, starting_time,
ending_location, ending_date, ending_time, Cancellation_Policy, sleeper_price,
ac_2_tier_price, ac_first_class_price, available_seats_sleeper, available_seats_ac_2_tier,
available_seats_ac_first_class)
VALUES
(12345, 'Rajdhani Express', 'Mumbai', '2024-04-18', '08:00:00', 'Delhi', '2024-04-19',
'17:00:00', 'Flexible', 500, 1000, 1500, 50, 25, 25),
(54321, 'Shatabdi Express', 'Pune', '2024-04-19', '09:00:00', 'Hyderabad', '2024-04-19',
'20:00:00', 'Strict', 500, 1000, 1500, 50, 25, 25),
(67890, 'Duronto Express', 'Bangalore', '2024-04-20', '10:00:00', 'Kolkata', '2024-04-21',
'06:00:00', 'Flexible', 500, 1000, 1500, 50, 25, 25),
(98765, 'Gatimaan Express', 'Kolkata', '2024-04-21', '11:00:00', 'Bangalore', '2024-04-22',
```

'05:00:00', 'Moderate', 500, 1000, 1500, 50, 25, 25),
(13579, 'Garib Rath', 'Chennai', '2024-04-22', '12:00:00', 'Mumbai', '2024-04-23', '12:00:00', 'Strict', 450, 900, 1400, 60, 30, 20),
(24680, 'Humsafar Express', 'Delhi', '2024-04-23', '13:00:00', 'Chennai', '2024-04-25', '13:00:00', 'Flexible', 550, 1050, 1550, 45, 20, 15),
(86420, 'Tejas Express', 'Goa', '2024-04-24', '14:00:00', 'Mumbai', '2024-04-24', '20:00:00', 'Moderate', 600, 1100, 1600, 55, 25, 10),
(97531, 'Vande Bharat', 'Ahmedabad', '2024-04-25', '15:00:00', 'Delhi', '2024-04-25', '21:00:00', 'Strict', 700, 1200, 1700, 40, 15, 5),
(75319, 'Mahamana Express', 'Varanasi', '2024-04-26', '16:00:00', 'New Delhi', '2024-04-27', '09:00:00', 'Flexible', 550, 950, 1450, 65, 35, 30),
(86459, 'Sampark Kranti', 'Hyderabad', '2024-04-27', '17:00:00', 'Jaipur', '2024-04-28', '17:00:00', 'Moderate', 525, 1025, 1525, 48, 24, 12),
(95175, 'Antyodaya Express', 'Mumbai', '2024-04-28', '18:00:00', 'Kolkata', '2024-04-30', '18:00:00', 'Flexible', 400, 800, 1300, 70, 40, 35),
(13457, 'Jan Shatabdi', 'Lucknow', '2024-04-29', '19:00:00', 'Patna', '2024-04-30', '06:00:00', 'Strict', 470, 970, 1470, 55, 27, 13),
(42068, 'Express Special', 'Jaipur', '2024-04-30', '20:00:00', 'Chandigarh', '2024-05-01', '06:00:00', 'Moderate', 500, 1000, 1500, 50, 25, 25),
(78695, 'Superfast Express', 'Bhopal', '2024-05-01', '21:00:00', 'Lucknow', '2024-05-02', '09:00:00', 'Flexible', 480, 980, 1480, 58, 29, 14),
(36987, 'Kavi Guru Express', 'Patna', '2024-05-02', '22:00:00', 'Jaipur', '2024-05-03', '10:00:00', 'Strict', 490, 990, 1490, 53, 28, 16);

Table outputs:

User Account:

Account_ID	Username	Email_id	Phone_number	Password
1	user1	user1@gmail.com	1234567890	password1
2	user2	user2@gmail.com	9876543210	password2
3	user3	user3@gmail.com	5551234567	password3
4	user4	user4@gmail.com	9998887776	password4
5	user5	user5@gmail.com	5555555555	password5
6	user6	user6@gmail.com	6666666666	password6
7	user7	user7@gmail.com	7777777777	password7
8	user8	user8@gmail.com	8888888888	password8
9	rishik	rishik@gmail.com	1234567890	123
10	boy	boy2004@gmail.com	0000000001	Boy2004
11	gal	gal2343@gmail.com	9191919191	Gal2343
12	Rishik ku...	reddyrishik2004@g...	9347994965	Reddyrish...
13	Kush	kush2004@gmail.com	8989898970	Kush2004
14	man	man2004@gmail.com	9837847839	Man2004
15	reddy	reddy123@gmail.com	8989898989	Reddy2004
NULL	NULL	NULL	NULL	NULL

Flight Table:

flight_i	Airline	Starting_Loca	Destination	Cancellation_Po	Starting_Dt	starting_time	Ending_dt	ending_time	economy_	business_pri	first_cla	available_sea	available	available_seats_
1	Air India	Delhi	Mumbai	Flexible	2024-03-15	08:00:00	2024-03-15	12:00:00	500	1000	1500	25	15	10
2	IndiGo	Ahmedabad	Pune	Strict	2024-03-20	10:00:00	2024-03-20	12:00:00	500	1000	1500	25	15	10
3	SpiceJet	Bangalore	Chandigarh	Flexible	2024-03-25	14:00:00	2024-03-25	17:00:00	500	1000	1500	25	15	10
4	GoAir	Guwahati	Delhi	Moderate	2024-03-30	12:00:00	2024-03-30	16:00:00	500	1000	1500	25	15	10
5	Vistara	Mumbai	Kolkata	Moderate	2024-04-02	09:00:00	2024-04-02	12:00:00	600	1100	1600	30	20	12
6	AirAsia	Chennai	Delhi	Flexible	2024-04-05	07:00:00	2024-04-05	10:00:00	550	1050	1550	28	16	11
7	Air India	Delhi	Bangalore	Strict	2024-04-10	17:00:00	2024-04-10	20:00:00	580	1150	1650	26	18	9
8	IndiGo	Mumbai	Jaipur	Moderate	2024-04-15	11:00:00	2024-04-15	13:00:00	570	1070	1570	24	14	13
9	SpiceJet	Hyderabad	Mumbai	Flexible	2024-04-18	16:00:00	2024-04-18	18:30:00	540	1040	1540	27	17	14
10	GoAir	Kochi	Delhi	Strict	2024-04-22	18:00:00	2024-04-22	21:00:00	520	1020	1520	23	13	15
11	Vistara	Jaipur	Chennai	Moderate	2024-04-25	20:00:00	2024-04-25	23:00:00	510	1010	1510	22	12	11
12	AirAsia	Delhi	Goa	Flexible	2024-04-28	06:00:00	2024-04-28	08:30:00	600	1200	1700	20	10	12
13	Air India	Mumbai	Chennai	Strict	2024-05-01	13:00:00	2024-05-01	15:30:00	550	1100	1600	29	19	10
14	IndiGo	Kolkata	Mumbai	Moderate	2024-05-05	15:00:00	2024-05-05	18:00:00	530	1030	1530	21	11	9
15	SpiceJet	Delhi	Ahmedabad	Flexible	2024-05-08	12:00:00	2024-05-08	14:00:00	560	1060	1560	32	22	13
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Car Table:

car_no	seater	driver_name	starting_location	destination	driver_status	price
ABC123	4	Manas Dhanpawde	Mumbai	Pune	unavailable	500
BCD516	6	Bhavya Khanna	Ahmedabad	Surat	available	600
BCD890	6	Bhavya Patel	Surat	Ahmedabad	available	600
DEF456	5	Rishik reddy	Bangalore	Hyderabad	available	500
GHI789	7	Jay Gondaliya	Chennai	Anand	available	500
JKL012	4	Kushdeo Shukla	Kolkata	Lucknow	available	500
MNO345	6	Kushay reddypawde	Delhi	Ahmedabad	available	500
MNO678	6	Kushay reddy	Ahmedabad	Delhi	available	500
PQR678	4	Rohan Sharma	Jaipur	Indore	available	550
PQR910	4	Rohit Sharma	Indore	Jaipur	unavailable	550
STU111	5	Shubham Patel	Patna	Bhopal	available	600
STU901	5	Shubham Singh	Bhopal	Patna	available	600
VWX121	7	Varun Gupta	Bhubaneswar	Nagpur	available	700
VWX234	7	Varun Khanna	Nagpur	Bhubaneswar	available	700
YZA314	4	Yash Singh	Jaipur	Indore	available	550
YZA567	4	Yash Gupta	Indore	Jaipur	available	550
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Hotel Table:

Hotel_id	Name	Rating	Amenities	Location	Price	Num_rooms_available	Type	Check_in	Check_out
101	Tranquil Haven Hotel	4.2	WiFi, Breakfast	Pune	500	47	AC	NULL	NULL
202	Serenity Lodge	4.5	Pool, Gym	Anand	500	60	Non-AC	NULL	NULL
303	Blissful Retreat Inn	4.8	Spa, Room Service	Hyderabad	500	68	AC	NULL	NULL
404	Oasis Heights Resort	3.9	Parking	Lucknow	500	40	AC	NULL	NULL
505	Peaceful Valley Villa	4.3	WiFi, Breakfast	Mumbai	600	45	AC	NULL	NULL
606	Tranquility Suites	4.7	Gym, Pool	Chennai	550	55	Non-AC	NULL	NULL
707	Harmony Inn	4.1	Room Service	Delhi	450	65	Non-AC	NULL	NULL
808	Sunny Shores Hotel	4.4	WiFi, Parking	Bangalore	700	30	AC	NULL	NULL
909	Calm Breeze Resort	4.6	Spa, Gym	Jaipur	650	35	AC	NULL	NULL
1010	Tranquil Oaks Lodge	4	Pool	Kolkata	550	50	Non-AC	NULL	NULL
1111	Soothing Sands Resort	4.9	WiFi, Room Service	Goa	750	25	AC	NULL	NULL
1212	Blissful Bay Hotel	4.2	Gym	Ahmedabad	600	40	Non-AC	NULL	NULL
1313	Tranquility Retreat	4.5	WiFi, Breakfast	Surat	550	55	AC	NULL	NULL
1414	Calm Waters Inn	3.8	Room Service, Pa...	Nagpur	500	60	Non-AC	NULL	NULL
1515	Serenity Suites	4.7	Spa	Bhopal	650	35	AC	NULL	NULL
1616	Tranquil Garden Res...	4.3	WiFi, Gym	Indore	600	40	Non-AC	NULL	NULL
1717	Quiet Haven Hotel	4.6	Breakfast	Gurgaon	700	30	AC	NULL	NULL
1818	Calm Retreat Inn	4.4	Pool, WiFi	Chandigarh	750	25	Non-AC	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Train Table:

Train_id	Name	starting_location	starting_date	starting_time	ending_location	ending_date	ending_time	Cancellation_Policy	sleeper_price	ac_2_tier	ac_first_	available_seats_sleepe
12345	Rajdhani Exp...	Mumbai	2024-04-18	08:00:00	Delhi	2024-04-19	17:00:00	Flexible	500	1000	1500	50
13457	Jan Shatabdi	Lucknow	2024-04-29	19:00:00	Patna	2024-04-30	06:00:00	Strict	470	970	1470	55
13579	Garib Rath	Chennai	2024-04-22	12:00:00	Mumbai	2024-04-23	12:00:00	Strict	450	900	1400	60
24680	Humsafar Ex...	Delhi	2024-04-23	13:00:00	Chennai	2024-04-25	13:00:00	Flexible	550	1050	1550	45
36987	Kavi Guru Ex...	Patna	2024-05-02	22:00:00	Jaipur	2024-05-03	10:00:00	Strict	490	990	1490	53
42068	Express Special	Jaipur	2024-04-30	20:00:00	Chandigarh	2024-05-01	06:00:00	Moderate	500	1000	1500	50
54321	Shatabdi Exp...	Pune	2024-04-19	09:00:00	Hyderabad	2024-04-19	20:00:00	Strict	500	1000	1500	50
67890	Duronto Expr...	Bangalore	2024-04-20	10:00:00	Kolkata	2024-04-21	06:00:00	Flexible	500	1000	1500	50
75319	Mahamana E...	Varanasi	2024-04-26	16:00:00	New Delhi	2024-04-27	09:00:00	Flexible	550	950	1450	65
78695	Superfast Ex...	Bhopal	2024-05-01	21:00:00	Lucknow	2024-05-02	09:00:00	Flexible	480	980	1480	58
86420	Tejas Express	Goa	2024-04-24	14:00:00	Mumbai	2024-04-24	20:00:00	Moderate	600	1100	1600	55
86459	Sampark Kranti	Hyderabad	2024-04-27	17:00:00	Jaipur	2024-04-28	17:00:00	Moderate	525	1025	1525	48

Hotel Bookings Table:

	HotelBooking_id	Hotel_id	Name	Rating	Amenities	Location	Price	num_of_rooms_boked	type	Check_in	Check_out
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Car Bookings Table:

	carbookingId	car_no	seater	driver_name	starting_location	destination	driver_status	price
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Queries implemented:

Query 1: Number of flights with economy price more than 500

```
select * from flight where economy_price>500;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

flight_id	Airline	Starting_Location	Destination	Cancellation_Policy	Starting_Dt	starting_time	Ending_dt	ending_time	economy_pr
5	Vistara	Mumbai	Kolkata	Moderate	2024-04-02	09:00:00	2024-04-02	12:00:00	600
6	AirAsia	Chennai	Delhi	Flexible	2024-04-05	07:00:00	2024-04-05	10:00:00	550
7	Air India	Delhi	Bangalore	Strict	2024-04-10	17:00:00	2024-04-10	20:00:00	580
8	IndiGo	Mumbai	Jaipur	Moderate	2024-04-15	11:00:00	2024-04-15	13:00:00	570
9	SpiceJet	Hyderabad	Mumbai	Flexible	2024-04-18	16:00:00	2024-04-18	18:30:00	540
10	GoAir	Kochi	Delhi	Strict	2024-04-22	18:00:00	2024-04-22	21:00:00	520
11	Vistara	Jaipur	Chennai	Moderate	2024-04-25	20:00:00	2024-04-25	23:00:00	510
12	AirAsia	Delhi	Goa	Flexible	2024-04-28	06:00:00	2024-04-28	08:30:00	600
13	Air India	Mumbai	Chennai	Strict	2024-05-01	13:00:00	2024-05-01	15:30:00	550

flight 1 | Apply | Revert

Query 2: Rating of hotels in descending order

```
select * from hotel order by rating desc;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Hotel_id	Name	Rating	Amenities	Location	Price	Num_rooms_available	Type	Check_in	Check_out
1111	Soothing Sands Resort	4.9	WiFi, Room Service	Goa	750	25	AC	NULL	NULL
303	Blissful Retreat Inn	4.8	Spa, Room Service	Hyderabad	500	70	AC	NULL	NULL
606	Tranquility Suites	4.7	4.8 , Pool	Chennai	550	55	Non-AC	NULL	NULL
1515	Serenity Suites	4.7	Spa	Bhopal	650	35	AC	NULL	NULL
909	Calm Breeze Resort	4.6	Spa, Gym	Jaipur	650	35	AC	NULL	NULL
1717	Quiet Haven Hotel	4.6	Breakfast	Gurgaon	700	30	AC	NULL	NULL
202	Serenity Lodge	4.5	Pool, Gym	Anand	500	60	Non-AC	NULL	NULL
1313	Tranquility Retreat	4.5	WiFi, Breakfast	Surat	550	55	AC	NULL	NULL
808	Sunny Shores Hotel	4.4	WiFi, Parking	Bangalore	700	30	AC	NULL	NULL
1818	Calm Retreat Inn	4.4	Pool, WiFi	Chandigarh	750	25	Non-AC	NULL	NULL

hotel 2 | Apply | Revert

Query 3: Average sleeper price in trains grouped by cancellation policy

```
select avg(sleeper_price),cancellation_policy from train group by cancellation_policy;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avg(sleeper_price)	cancellation_policy
496.6667	Flexible
522.0000	Strict
531.2500	Moderate

Result 3 × Read Only

Query 4: display the hotels which have bookings in decreasing order of number of rooms booked

```
SELECT h.Name AS HotelName,h.Location,hb.num_of_rooms_boked AS RoomsBooked,hb.Price AS BookingPrice,
h.Amenities FROM HotelBookings hb inner JOIN Hotel h ON hb.Hotel_id = h.Hotel_id
WHERE hb.num_of_rooms_boked > 0 ORDER BY hb.num_of_rooms_boked DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

HotelName	Location	RoomsBooked	BookingPrice	Amenities
Blissful Bay Hotel	Ahmedabad	4	600.00	Gym
Tranquility Retreat	Surat	4	550.00	WiFi, Breakfast
Blissful Retreat Inn	Hyderabad	3	500.00	Spa, Room Service
Calm Breeze Resort	Jaipur	3	650.00	Spa, Gym
Tranquil Haven Hotel	Pune	2	500.00	WiFi, Breakfast
Oasis Heights Resort	Lucknow	2	500.00	Parking

Result 6 × Read Only

Query 5: Display the details of booked cars

```
SELECT cb.carbookingId,c.car_no,c.driver_name,c.starting_location,c.destination,
c.price,c.seater FROM CarBookings cb JOIN car c ON cb.car_no = c.car_no;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

carbookingId	car_no	driver_name	starting_location	destination	price	seater
1	ABC123	Manas Dhanpawde	Mumbai	Pune	500	4
10	BCD890	Bhavya Patel	Surat	Ahmedabad	600	6
2	DEF456	Rishik reddy	Bangalore	Hyderabad	500	5
3	GHI789	Jay Gondaliya	Chennai	Anand	500	7
4	JKL012	Kushdeo Shukla	Kolkata	Lucknow	500	4
5	MNO345	Kushay reddypawde	Delhi	Ahmedabad	500	6

We have covered select, order by, group by and join commands in our queries.

Function:
A function is a set of SQL statements, that you can pass parameters into, and they perform some operation and return a single value.

Functions:

1. The 'CalculateHotelBookingTotalPrice' function computes the total price for booking a specific number of rooms at a hotel based on the hotel ID and the number of rooms booked.

```
DELIMITER //
CREATE FUNCTION CalculateHotelBookingTotalPrice(hotel_id INT, num_of_rooms INT)
RETURNS DECIMAL(10, 2)
DETERMINISTIC
BEGIN
    DECLARE total_price DECIMAL(10, 2);
    SELECT SUM(Price * num_of_rooms)
    INTO total_price
    FROM Hotel
    WHERE Hotel_id = hotel_id;
    RETURN total_price;
END//
DELIMITER ;
SELECT CalculateHotelBookingTotalPrice(101, 3);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CalculateHotelBookingTotalPrice(101, 3)
31800.00

2. Function to check if a car is available for booking

```
DELIMITER //
CREATE FUNCTION IsCarAvailableForBooking(car_no VARCHAR(255), start_date DATE, end_date DATE)
RETURNS VARCHAR(20)
DETERMINISTIC
BEGIN
    DECLARE availability_status VARCHAR(20);
    SELECT CASE WHEN COUNT(*) > 0 THEN 'Available' ELSE 'Not Available' END
    INTO availability_status
    FROM CarBookings
    WHERE car_no = car_no
    AND ((start_date BETWEEN start_date AND end_date) OR (end_date BETWEEN start_date AND end_date));
    RETURN availability_status;
END//
DELIMITER ;
SELECT IsCarAvailableForBooking('ABC123', '2024-04-15', '2024-04-18');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

IsCarAvailableForBooking('ABC123', '2024-04-15', '2024-04-18')
Not Available

Stored Procedure:
A stored procedure is a set of pre-compiled Structured Query Languages (SQL), so it can be reused and shared by multiple programs. It can access or modify data in a database.

Procedure:

1.Procedure to retrieve user information based on username:

```
DELIMITER //
CREATE PROCEDURE GetUserDetailsByUsername(IN uname VARCHAR(255))
BEGIN
    SELECT Username, Email_id, Phone_number
    FROM User_Account
    WHERE Username = uname;
END //
DELIMITER ;
CALL GetUserDetailsByUsername('user1');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Username	Email_id	Phone_number
user1	user1@gmail.com	1234567890

2. Procedure to check the availability of a car for booking:

```
DELIMITER //
CREATE PROCEDURE CheckCarAvailability(IN car_num VARCHAR(255), IN start_date DATE, IN end_date DATE)
BEGIN
    SELECT
        CASE
            WHEN EXISTS (
                SELECT *
                FROM CarBookings
                WHERE car_no = car_num
                AND (start_date BETWEEN start_date AND end_date OR end_date BETWEEN start_date AND end_date)
            ) THEN 'Not Available'
            ELSE 'Available'
        END AS Availability;
END //
DELIMITER ;
CALL CheckCarAvailability('ABC123', '2024-04-15', '2024-04-18');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Availability
Available

Result Grid

Trigger:
A trigger is also a set of SQL statements in the database which automatically execute whenever any special event occurs in the database, like insert, delete, update, etc.

Triggers:

1. Trigger for updating available rooms in Hotel table after insertion into HotelBookings table

```
DELIMITER //
CREATE TRIGGER after_hotel_booking_insert
AFTER INSERT ON HotelBookings
FOR EACH ROW
BEGIN
    UPDATE Hotel
    SET Num_rooms_available = Num_rooms_available - NEW.num_of_rooms_boked
    WHERE Hotel_id = NEW.Hotel_id;
END;
//
DELIMITER ;
```

Output

Action Output

#	Time	Action	Message
4	11:46:50	drop trigger after_hotel_booking_insert	0 row(s) affected
5	11:47:02	CREATE TRIGGER after_hotel_booking_insert AFTER INSERT ON HotelBookings ...	0 row(s) affected

2. Trigger for updating available cars in car table after insertion into CarBookings table:

```
DELIMITER //
CREATE TRIGGER after_car_booking_insert
AFTER INSERT ON CarBookings
FOR EACH ROW
BEGIN
    UPDATE car
    SET driver_status = 'booked'
    WHERE car_no = NEW.car_no;
END;
//
DELIMITER ;
```

Output

Action Output

#	Time	Action	Message
1	11:57:03	use travel_agency	0 row(s) affected
2	11:57:09	CREATE TRIGGER after_car_booking_insert AFTER INSERT ON CarBookings FOR EACH R...	0 row(s) affected

---

References:

Tools:

- https://scholar.google.com/
- https://www.canva.com/

Research:

- https://deliverypdf.ssrn.com/delivery.php?ID=045009089115009002113104120090124088033043069002065054095096003092123074031012001123043028032104020043034086117076086076006087022034064058004081074030087000020110124037042054008102074077007112093087126001015126121124007065015100024025029081111065029073&EXT=pdf&INDEX=TRUE
- https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/2500/1386
- http://103.133.167.5:8080/bitstream/handle/123456789/2055/Hasib_Ahmed_Abir.pdf?sequence=1&isAllowed=y