SQL PROJECT :

Airline Reservation System

Name : Vikrant Jagannath Chavan

INTODUCTION

The Airline Reservation System is an SQL-based project designed to manage and streamline the process of booking and managing airline flights. This system focuses on storing, retrieving, and processing essential information related to passengers, flights, bookings, payments, schedules, and crew assignments. It simulates the core functionalities of a real-world airline database, ensuring efficient data management for both passengers and airlines.

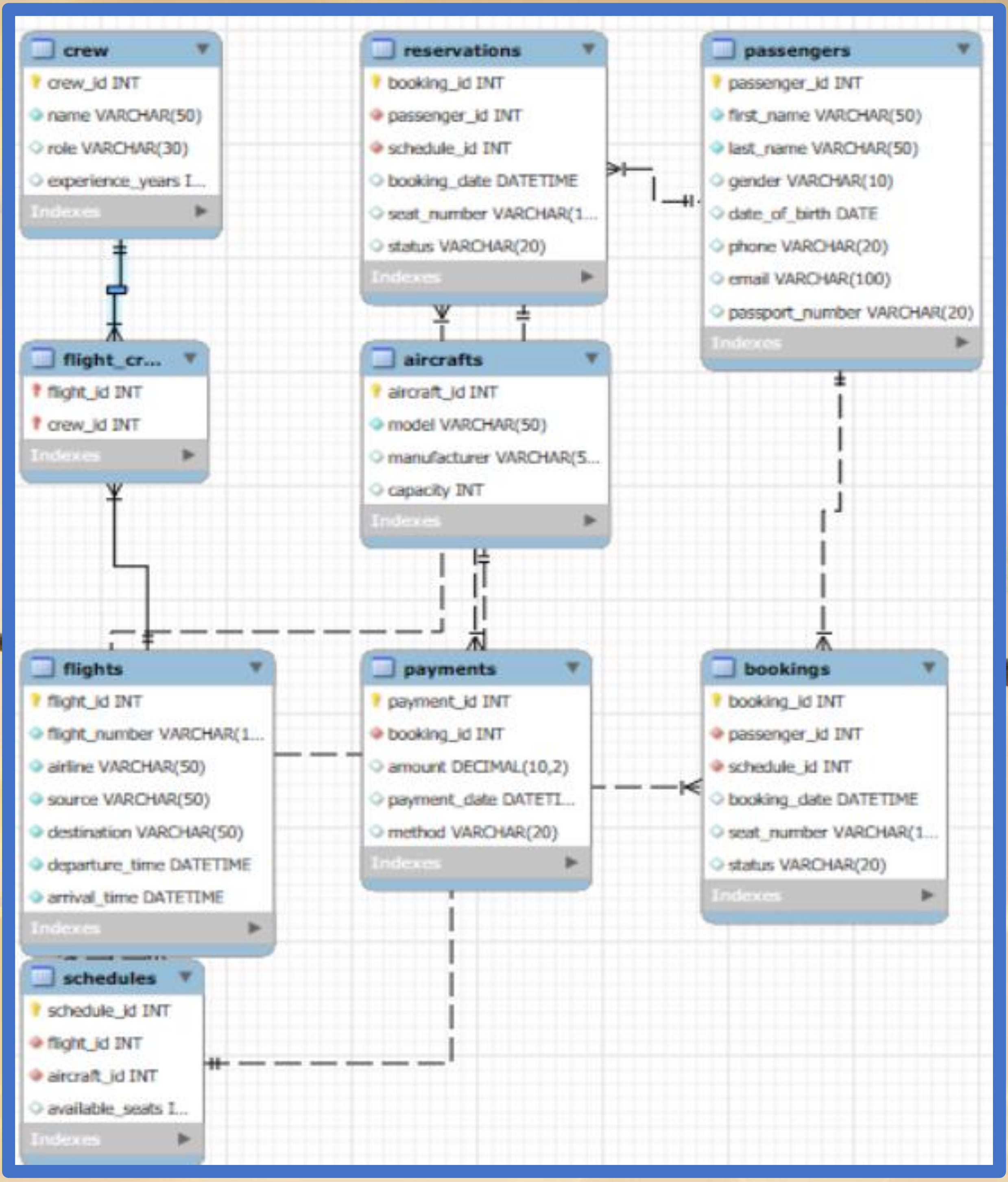
The primary goal of this system is to provide quick access to critical flight and passenger information, allowing airlines to manage reservations effectively and passengers to book tickets with ease. By maintaining well-structured records of flights, available seats, passenger details, and payment transactions, the system ensures reliability, accuracy, and convenience in the airline reservation process.

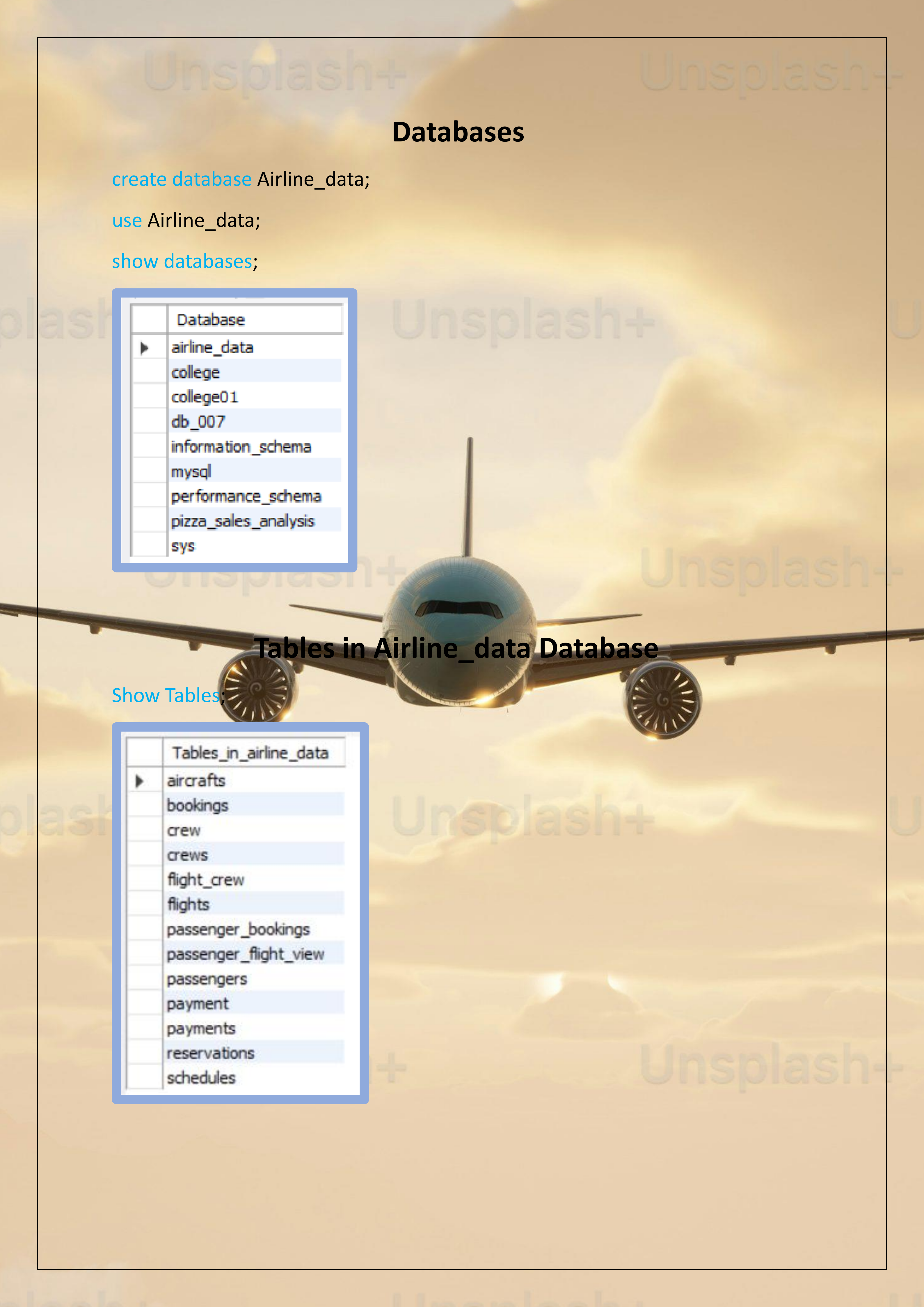
From a technical perspective, the project utilizes relational database concepts such as table creation, constraints, relationships, and normalization to organize data efficiently. It also employs SQL operations like insertion, updates, filtering, aggregation, joins, subqueries, and views to support powerful queries and analysis.

The project demonstrates how databases can be used in real-world scenarios to manage complex information systems. It can be extended further to include advanced features such as frequent flyer programs, ticket cancellations, real-time seat availability, and dynamic pricing.

By working on this project, one can strengthen their skills in SQL, database design, and data analysis while understanding how technology powers the aviation industry's core operations.

ER Diagram


crew
crew_id INT
name VARCHAR(50)
role VARCHAR(30)
experience_years I...
Indexes
reservations
booking_id INT
passenger_id INT
schedule_id INT
booking_date DATETIME
seat_number VARCHAR(1...
status VARCHAR(20)
Indexes
passengers
passenger_id INT
first_name VARCHAR(50)
last_name VARCHAR(50)
gender VARCHAR(10)
date_of_birth DATE
phone VARCHAR(20)
email VARCHAR(100)
passport_number VARCHAR(20)
Indexes
flight_cr...
flight_id INT
crew_id INT
Indexes
aircrafts
aircraft_id INT
model VARCHAR(50)
manufacturer VARCHAR(5...
capacity INT
Indexes
flights
flight_id INT
flight_number VARCHAR(1...
airline VARCHAR(50)
source VARCHAR(50)
destination VARCHAR(50)
departure_time DATETIME
arrival_time DATETIME
Indexes
payments
payment_id INT
booking_id INT
amount DECIMAL(10,2)
payment_date DATETI...
method VARCHAR(20)
Indexes
bookings
booking_id INT
passenger_id INT
schedule_id INT
booking_date DATETIME
seat_number VARCHAR(1...
status VARCHAR(20)
Indexes
schedules
schedule_id INT
flight_id INT
aircraft_id INT
available_seats I...
Indexes

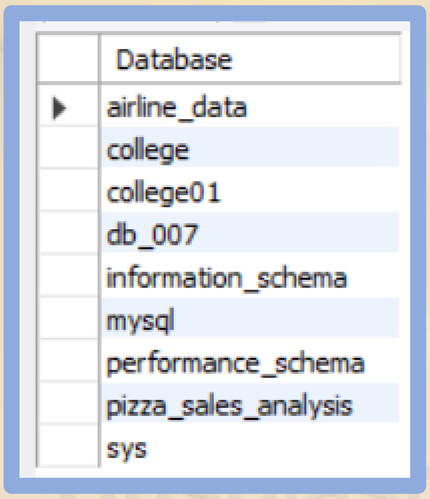


Databases

create database Airline_data;

use Airline_data;

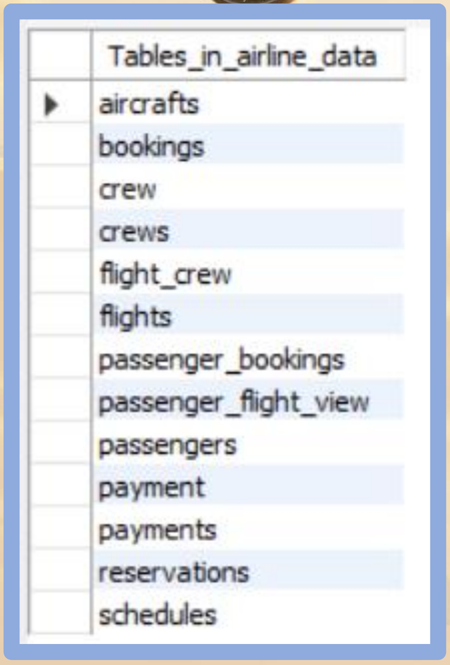
show databases;



```
Database
airline_data
college
college01
db_007
information_schema
mysql
performance_schema
pizza_sales_analysis
sys
```

Tables in Airline_data Database

Show Tables;



```
Tables_in_airline_data
aircrafts
bookings
crew
crews
flight_crew
flights
passenger_bookings
passenger_flight_view
passengers
payment
payments
reservations
schedules
```

Data Definition Language(DDL)

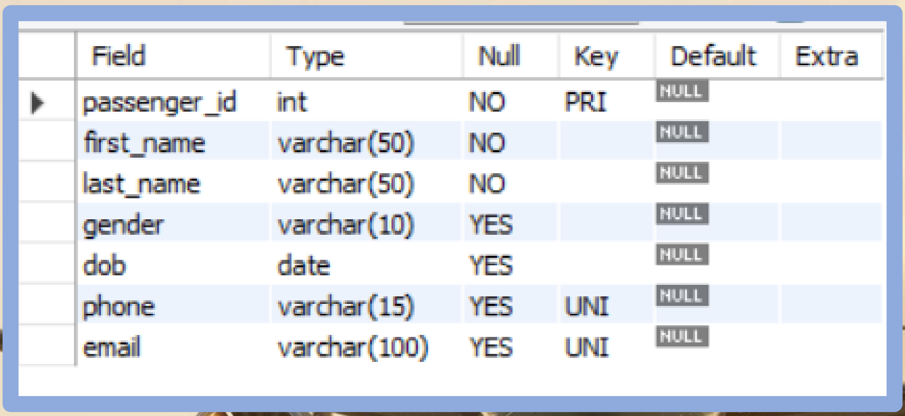
1.Creating Tables

A.Passengers

Stores passengers details.

```
Create Table Passengers (passenger_id int primary key, first_name varchar(50) not null, last_name varchar(50) not null, gender varchar(10), dob date, phone varchar(15) unique, email varchar(100) unique);
```

```
Desc Passengers;
```



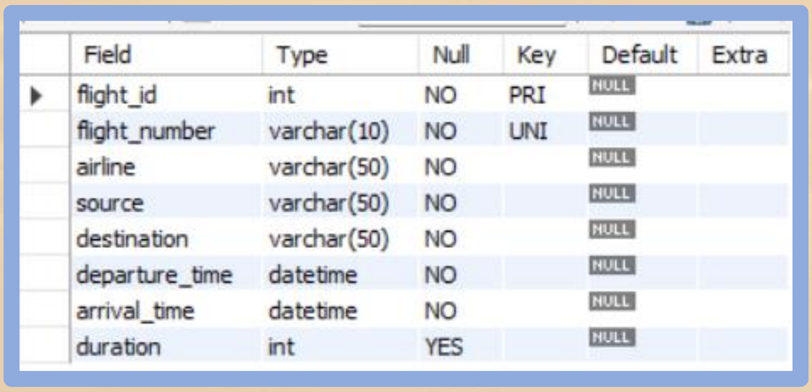
Field	Type	Null	Key	Default	Extra
passenger_id	int	NO	PRI	NULL	
first_name	varchar(50)	NO		NULL	
last_name	varchar(50)	NO		NULL	
gender	varchar(10)	YES		NULL	
dob	date	YES		NULL	
phone	varchar(15)	YES	UNI	NULL	
email	varchar(100)	YES	UNI	NULL	

B.Flights

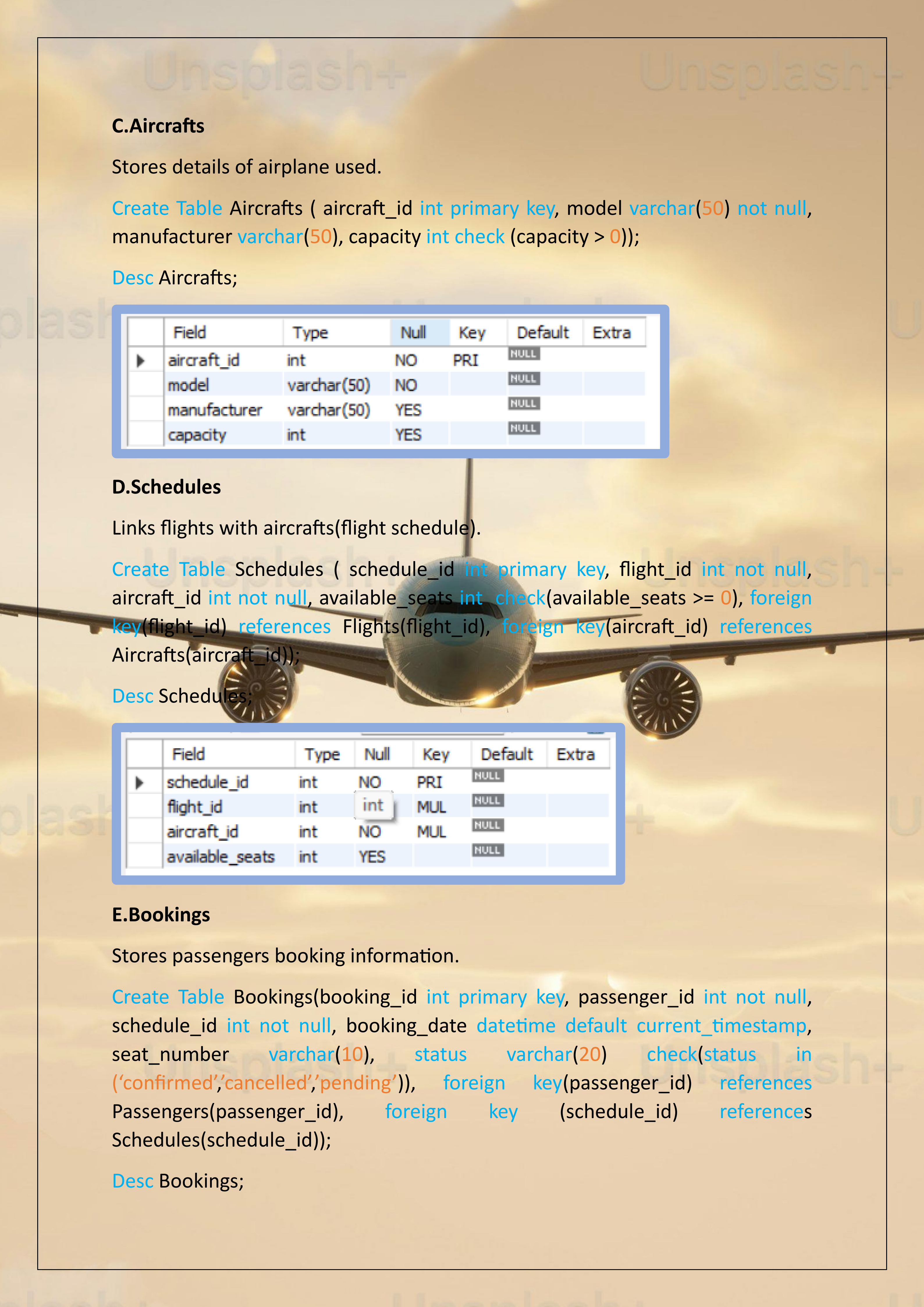
Stores flight details.

```
Create Table Flights ( flight_id int primary key, flight_number varchar(10) unique not null, airline varchar(50) not null, source varchar(50) not null, destination varchar(50) not null, departure_time datetime not null, arrival_time datetime not null, duration int check (duration > 0));
```

```
Desc Flights;
```



Field	Type	Null	Key	Default	Extra
flight_id	int	NO	PRI	NULL	
flight_number	varchar(10)	NO	UNI	NULL	
airline	varchar(50)	NO		NULL	
source	varchar(50)	NO		NULL	
destination	varchar(50)	NO		NULL	
departure_time	datetime	NO		NULL	
arrival_time	datetime	NO		NULL	
duration	int	YES		NULL	

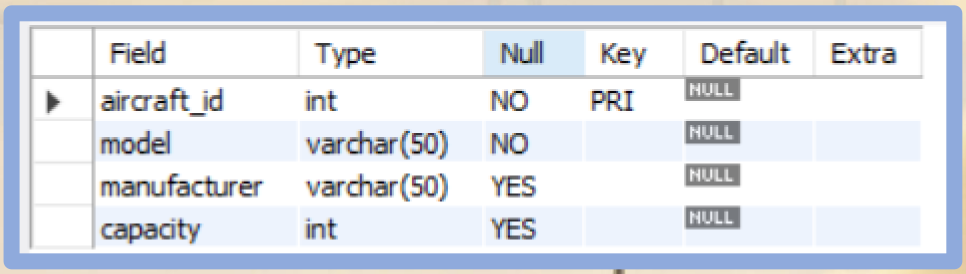


C.Aircrafts

Stores details of airplane used.

Create Table Aircrafts (aircraft_id int primary key, model varchar(50) not null, manufacturer varchar(50), capacity int check (capacity > 0));

Desc Aircrafts;



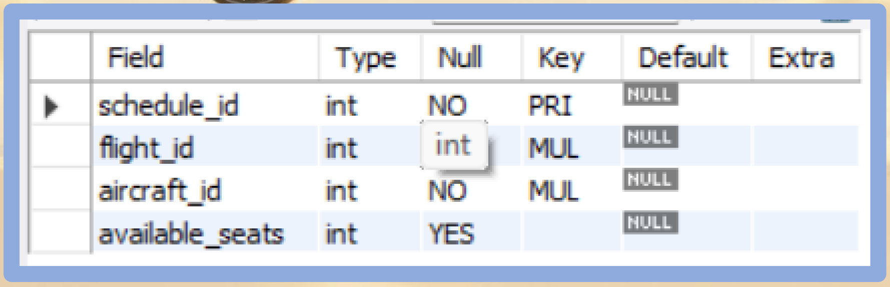
Field	Type	Null	Key	Default	Extra
aircraft_id	int	NO	PRI	NULL	
model	varchar(50)	NO		NULL	
manufacturer	varchar(50)	YES		NULL	
capacity	int	YES		NULL	

D.Schedules

Links flights with aircrafts(flight schedule).

Create Table Schedules (schedule_id int primary key, flight_id int not null, aircraft_id int not null, available_seats int check(available_seats >= 0), foreign key(flight_id) references Flights(flight_id), foreign key(aircraft_id) references Aircrafts(aircraft_id));

Desc Schedules;



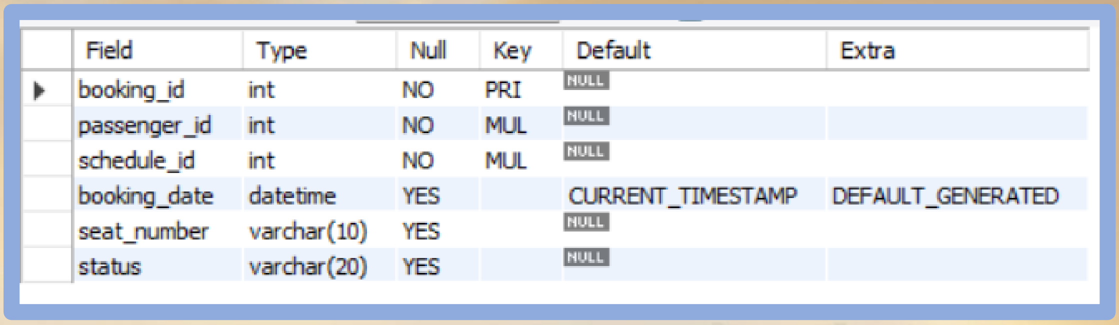
Field	Type	Null	Key	Default	Extra
schedule_id	int	NO	PRI	NULL	
flight_id	int	int	MUL	NULL	
aircraft_id	int	NO	MUL	NULL	
available_seats	int	YES		NULL	

E.Bookings

Stores passengers booking information.

Create Table Bookings(booking_id int primary key, passenger_id int not null, schedule_id int not null, booking_date datetime default current_timestamp, seat_number varchar(10), status varchar(20) check(status in (‘confirmed’,’cancelled’,’pending’)), foreign key(passenger_id) references Passengers(passenger_id), foreign key (schedule_id) references Schedules(schedule_id));

Desc Bookings;



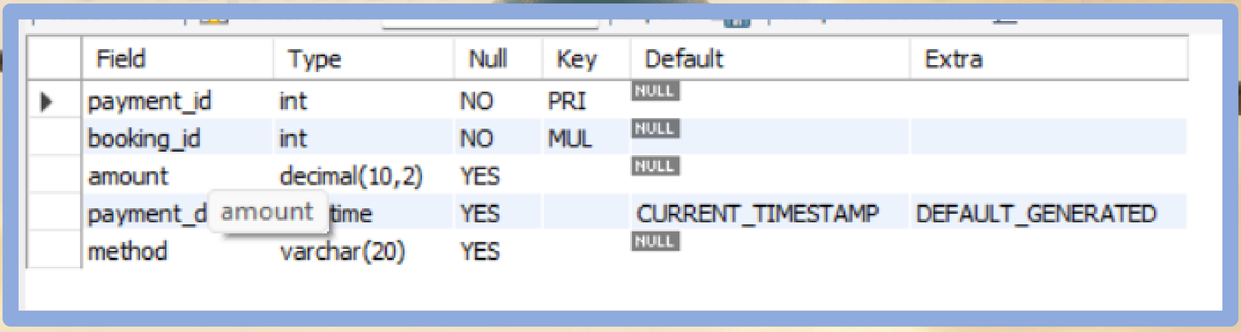
	Field	Type	Null	Key	Default	Extra
▶	booking_id	int	NO	PRI	NULL	
	passenger_id	int	NO	MUL	NULL	
	schedule_id	int	NO	MUL	NULL	
	booking_date	datetime	YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
	seat_number	varchar(10)	YES		NULL	
	status	varchar(20)	YES		NULL	

F.Payment

Stores payments details for bookings.

Create Table Payment (payment_id int primary key, booking_id int not null, amount decimal(10,2) check(amount > 0), payment_date datetime default current_timestamp,method varchar(20) check(method in ('Credit Card','Debit Card','UPI','Net Banking','Cash')), foreign key (booking_id) references Bookings(booking_id));

Desc Payment;



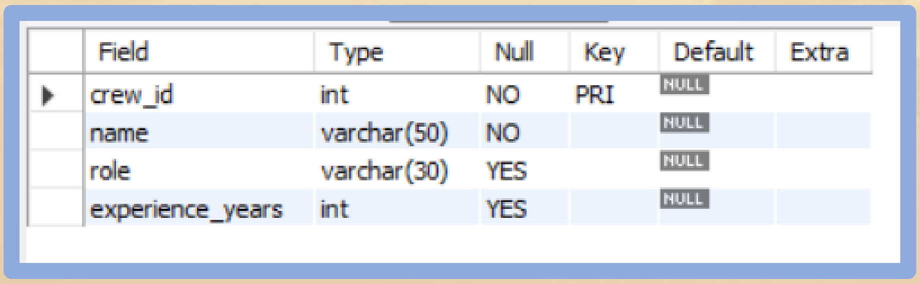
	Field	Type	Null	Key	Default	Extra
▶	payment_id	int	NO	PRI	NULL	
	booking_id	int	NO	MUL	NULL	
	amount	decimal(10,2)	YES		NULL	
	payment_d amount time		YES		CURRENT_TIMESTAMP	DEFAULT_GENERATED
	method	varchar(20)	YES		NULL	

G.Crews

Stores flight crew details.

Create Table Crews (crew_id int primary key, name varchar(50) not null, role varchar(30) check (role in ('Pilot','Co-Pilot','Cabin Crew','Engineer')), experience_years int check (experience_years >= 0));

Desc Crews



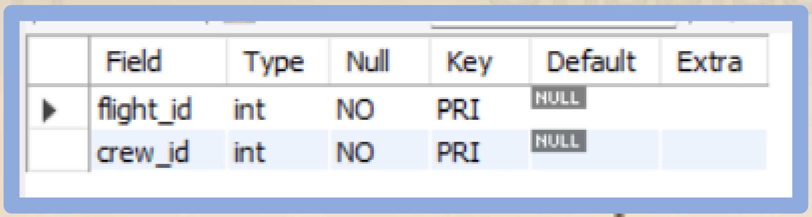
	Field	Type	Null	Key	Default	Extra
▶	crew_id	int	NO	PRI	NULL	
	name	varchar(50)	NO		NULL	
	role	varchar(30)	YES		NULL	
	experience_years	int	YES		NULL	

H.Flight Crew

Mapping table for many-to-many relation between flights and crew.

```
Create Table Flight_Crew (flight_id int not null, crew_id int not null, primary key (flight_id, crew_id), foreign key (flight_id) references Flights(flight_id), foreign key (crew_id) references Crew(crew_id));
```

```
Desc Flight_Crew;
```

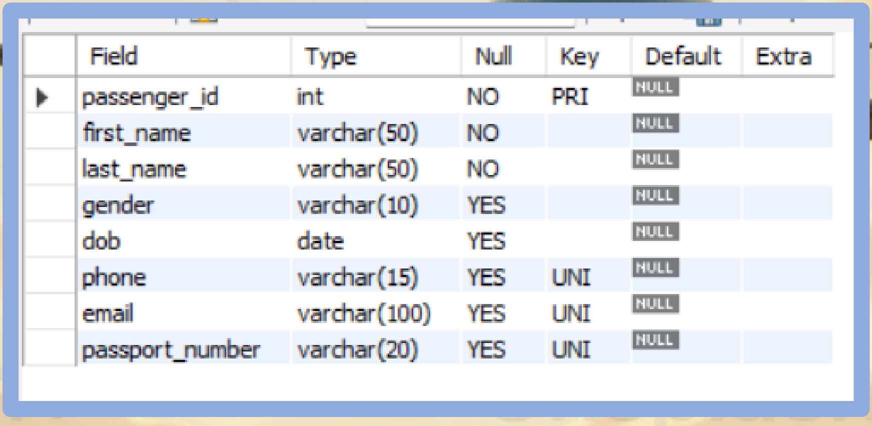


Field	Type	Null	Key	Default	Extra
flight_id	int	NO	PRI	NULL	
crew_id	int	NO	PRI	NULL	

2.Alter Tables

A. Add New Column

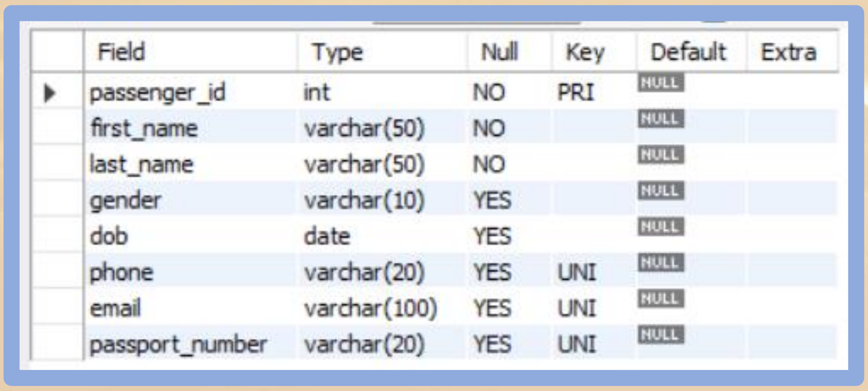
```
Alter table Passengers add passport_number varchar(20) unique;
```



Field	Type	Null	Key	Default	Extra
passenger_id	int	NO	PRI	NULL	
first_name	varchar(50)	NO		NULL	
last_name	varchar(50)	NO		NULL	
gender	varchar(10)	YES		NULL	
dob	date	YES		NULL	
phone	varchar(15)	YES	UNI	NULL	
email	varchar(100)	YES	UNI	NULL	
passport_number	varchar(20)	YES	UNI	NULL	

B. Modify Column

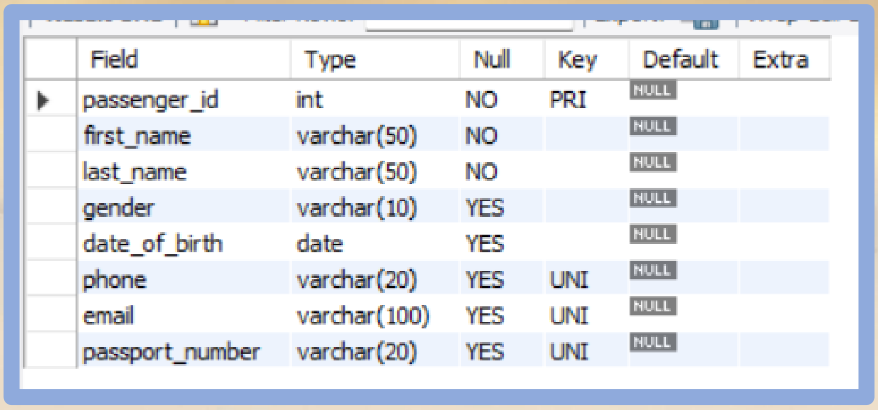
```
Alter table Passengers modify phone varchar(20);
```



Field	Type	Null	Key	Default	Extra
passenger_id	int	NO	PRI	NULL	
first_name	varchar(50)	NO		NULL	
last_name	varchar(50)	NO		NULL	
gender	varchar(10)	YES		NULL	
dob	date	YES		NULL	
phone	varchar(20)	YES	UNI	NULL	
email	varchar(100)	YES	UNI	NULL	
passport_number	varchar(20)	YES	UNI	NULL	

C. Rename Column

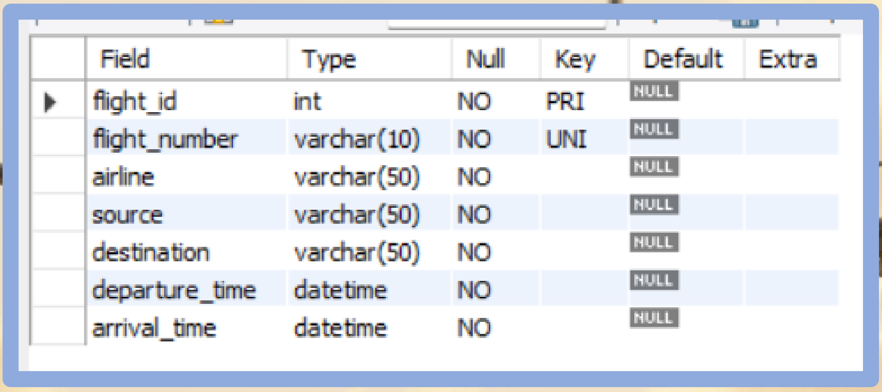
Alter table Passengers change dob date_of_birth date;



Field	Type	Null	Key	Default	Extra
passenger_id	int	NO	PRI	NULL	
first_name	varchar(50)	NO		NULL	
last_name	varchar(50)	NO		NULL	
gender	varchar(10)	YES		NULL	
date_of_birth	date	YES		NULL	
phone	varchar(20)	YES	UNI	NULL	
email	varchar(100)	YES	UNI	NULL	
passport_number	varchar(20)	YES	UNI	NULL	

D. Drop Column

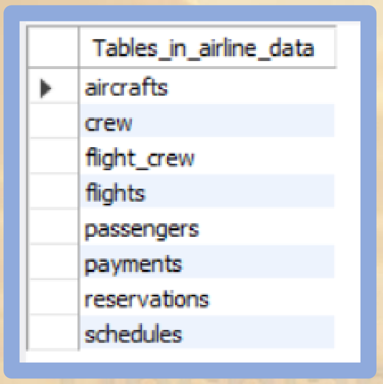
Alter table Flights drop duration;



Field	Type	Null	Key	Default	Extra
flight_id	int	NO	PRI	NULL	
flight_number	varchar(10)	NO	UNI	NULL	
airline	varchar(50)	NO		NULL	
source	varchar(50)	NO		NULL	
destination	varchar(50)	NO		NULL	
departure_time	datetime	NO		NULL	
arrival_time	datetime	NO		NULL	

E. Rename table

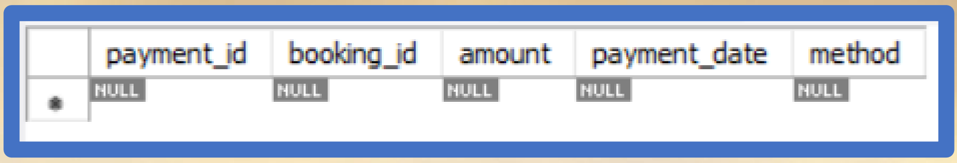
Alter table Bookings rename Reservations;



Tables_in_airline_data
aircrafts
crew
flight_crew
flights
passengers
payments
reservations
schedules

3. Truncate Table

Truncate table payment;



	payment_id	booking_id	amount	payment_date	method
*	NULL	NULL	NULL	NULL	NULL

4. Drop Table

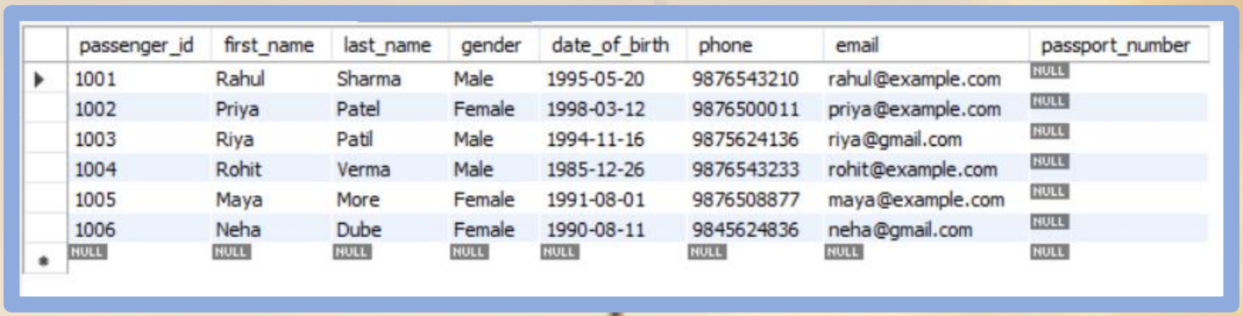
Drop table payment;

Data Manipulation Language (DML)

1. Insert into Table

Insert into Passengers(passenger_id,first_name, last_name, gender, date_of_birth, phone, email) values (1001,'Rahul', 'Sharma', 'Male', '1995-05-20', 9876543210, 'rahul@example.com');

select * from passengers;

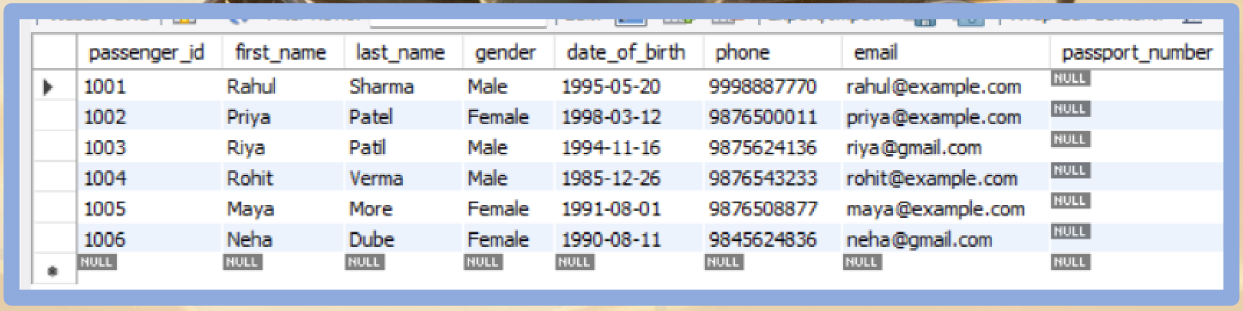


passenger_id	first_name	last_name	gender	date_of_birth	phone	email	passport_number
1001	Rahul	Sharma	Male	1995-05-20	9876543210	rahul@example.com	NULL
1002	Priya	Patel	Female	1998-03-12	9876500011	priya@example.com	NULL
1003	Riya	Patil	Male	1994-11-16	9875624136	riya@gmail.com	NULL
1004	Rohit	Verma	Male	1985-12-26	9876543233	rohit@example.com	NULL
1005	Maya	More	Female	1991-08-01	9876508877	maya@example.com	NULL
1006	Neha	Dube	Female	1990-08-11	9845624836	neha@gmail.com	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. Update into Table

Q. Update passengers phone number.

Update Passengers set phone = 9998887770 where passenger_id = 1001;

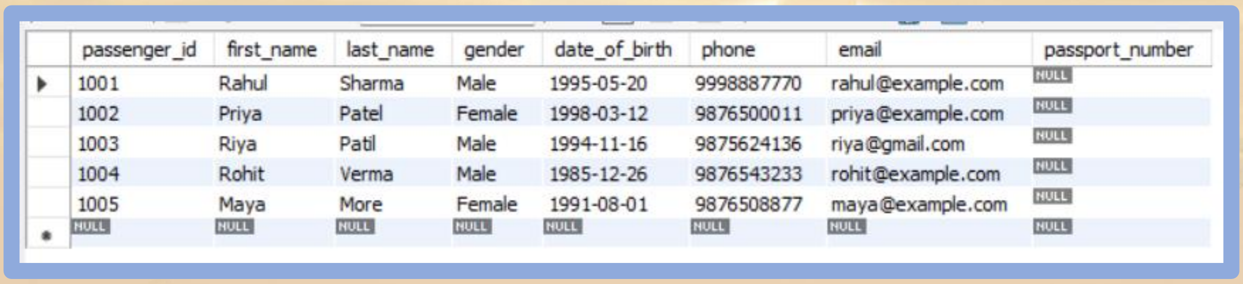


passenger_id	first_name	last_name	gender	date_of_birth	phone	email	passport_number
1001	Rahul	Sharma	Male	1995-05-20	9998887770	rahul@example.com	NULL
1002	Priya	Patel	Female	1998-03-12	9876500011	priya@example.com	NULL
1003	Riya	Patil	Male	1994-11-16	9875624136	riya@gmail.com	NULL
1004	Rohit	Verma	Male	1985-12-26	9876543233	rohit@example.com	NULL
1005	Maya	More	Female	1991-08-01	9876508877	maya@example.com	NULL
1006	Neha	Dube	Female	1990-08-11	9845624836	neha@gmail.com	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3.Delete From Table

Q. Delete passenger record.

Delete from Passengers where passenger_id =1006;



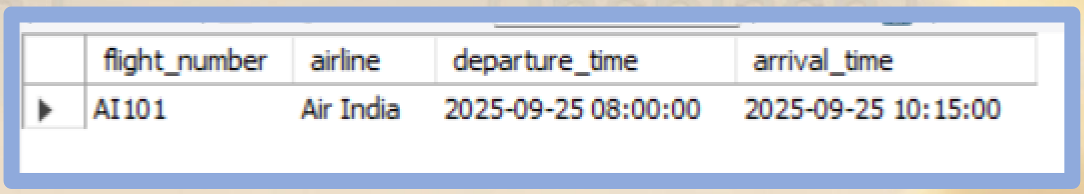
passenger_id	first_name	last_name	gender	date_of_birth	phone	email	passport_number
1001	Rahul	Sharma	Male	1995-05-20	9998887770	rahul@example.com	NULL
1002	Priya	Patel	Female	1998-03-12	9876500011	priya@example.com	NULL
1003	Riya	Patil	Male	1994-11-16	9875624136	riya@gmail.com	NULL
1004	Rohit	Verma	Male	1985-12-26	9876543233	rohit@example.com	NULL
1005	Maya	More	Female	1991-08-01	9876508877	maya@example.com	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Data Query Language (DQL)

1.Select

Q. Display all flights between Delhi and Mumbai

Select flight_number, airline, departure_time, arrival_time from Flights where source = 'Delhi' and destination = 'Mumbai';

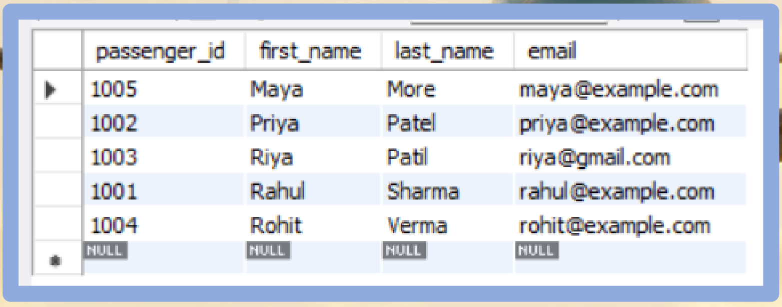


flight_number	airline	departure_time	arrival_time
AI101	Air India	2025-09-25 08:00:00	2025-09-25 10:15:00

2. Order by

Q. Show all passengers sorted by last name in alphabetical order.

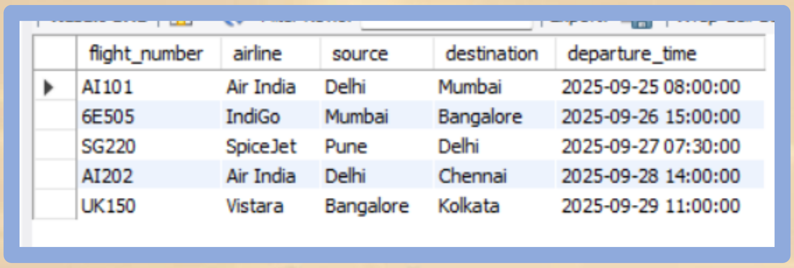
Select passenger_id, first_name, last_name, email from Passengers order by last_name asc;



passenger_id	first_name	last_name	email
1005	Maya	More	maya@example.com
1002	Priya	Patel	priya@example.com
1003	Riya	Patil	riya@gmail.com
1001	Rahul	Sharma	rahul@example.com
1004	Rohit	Verma	rohit@example.com
NULL	NULL	NULL	NULL

Q. List all flights sorted by departure time (earliest first).

Select flight_number, airline, source, destination, departure_time from Flights order by departure_time asc;

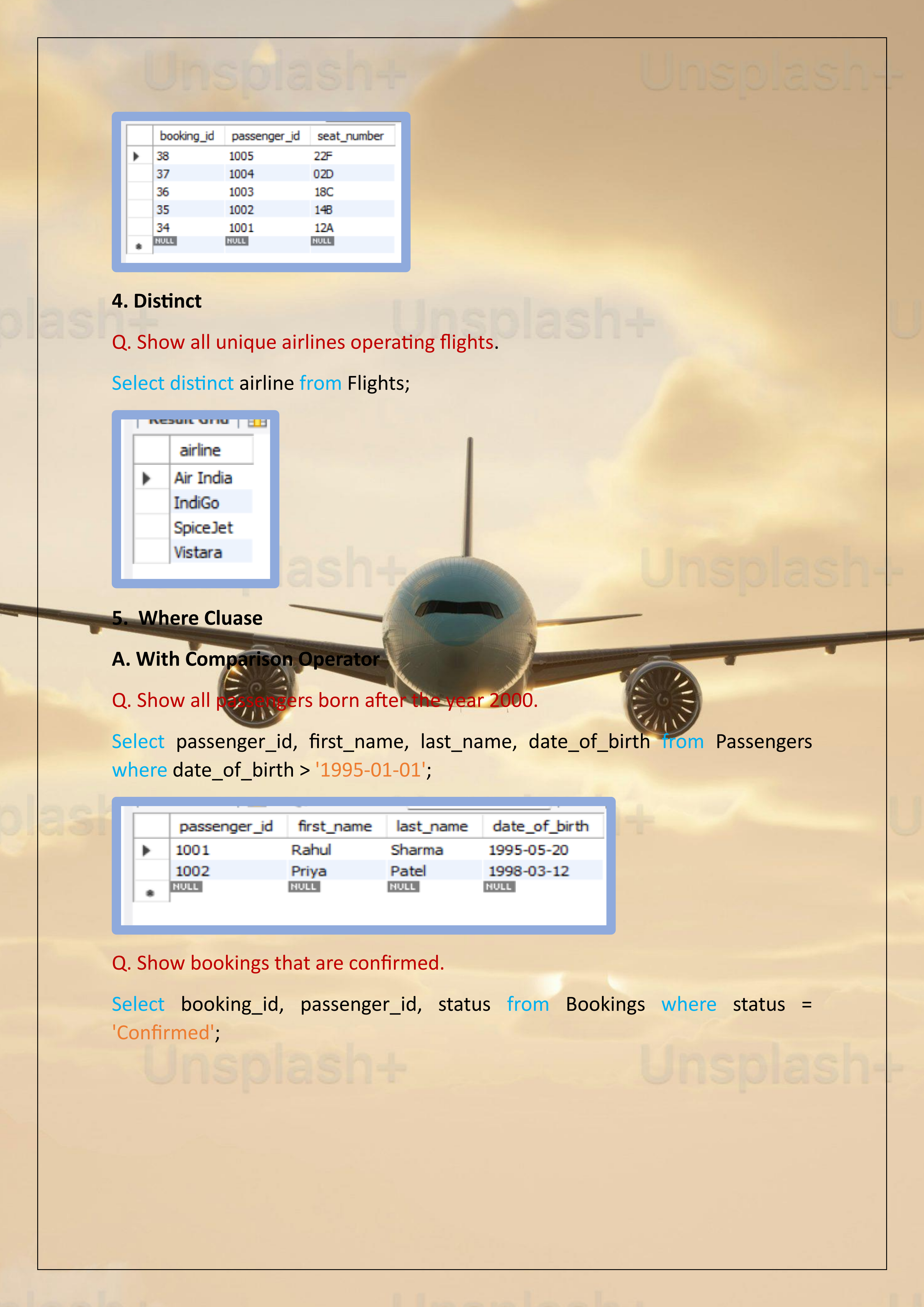


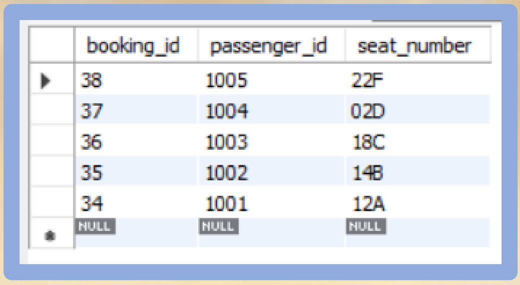
flight_number	airline	source	destination	departure_time
AI101	Air India	Delhi	Mumbai	2025-09-25 08:00:00
6E505	IndiGo	Mumbai	Bangalore	2025-09-26 15:00:00
SG220	SpiceJet	Pune	Delhi	2025-09-27 07:30:00
AI202	Air India	Delhi	Chennai	2025-09-28 14:00:00
UK150	Vistara	Bangalore	Kolkata	2025-09-29 11:00:00

3. Limit

Q. Top 5 most expensive bookings

Select booking_id, passenger_id, seat_number from Bookings order by booking_date dsc limit 5;



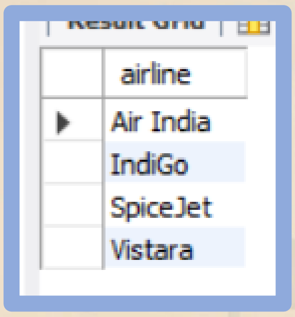


booking_id	passenger_id	seat_number
38	1005	22F
37	1004	02D
36	1003	18C
35	1002	14B
34	1001	12A
NULL	NULL	NULL

4. Distinct

Q. Show all unique airlines operating flights.

Select distinct airline from Flights;



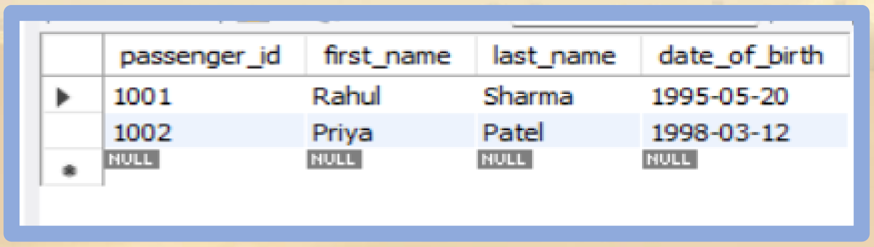
airline
Air India
IndiGo
SpiceJet
Vistara

5. Where Cluase

A. With Comparison Operator

Q. Show all passengers born after the year 2000.

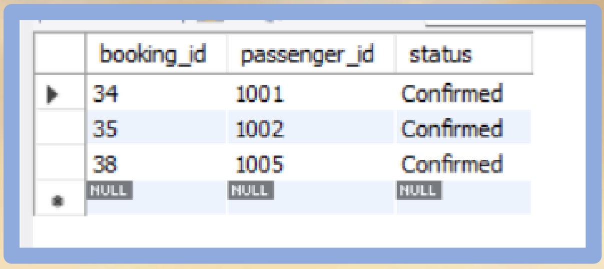
Select passenger_id, first_name, last_name, date_of_birth from Passengers where date_of_birth > '1995-01-01';



passenger_id	first_name	last_name	date_of_birth
1001	Rahul	Sharma	1995-05-20
1002	Priya	Patel	1998-03-12
NULL	NULL	NULL	NULL

Q. Show bookings that are confirmed.

Select booking_id, passenger_id, status from Bookings where status = 'Confirmed';



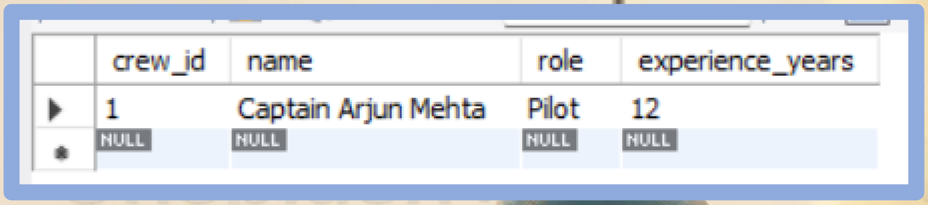
	booking_id	passenger_id	status
▶	34	1001	Confirmed
	35	1002	Confirmed
	38	1005	Confirmed
*	NULL	NULL	NULL

B. With Logical Operators

- Using AND Operator

Q. Find all crew members who are Pilots with more than 10 years of experience.

Select crew_id, name, role, experience_years from Crews where role = 'Pilot' and experience_years > 10;

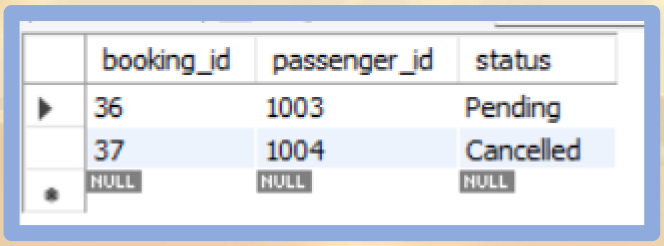


	crew_id	name	role	experience_years
▶	1	Captain Arjun Mehta	Pilot	12
*	NULL	NULL	NULL	NULL

- Using OR Operator

Q. List bookings that are either Cancelled or Pending.

Select booking_id, passenger_id, status from Bookings where status = 'Cancelled' OR status = 'Pending';

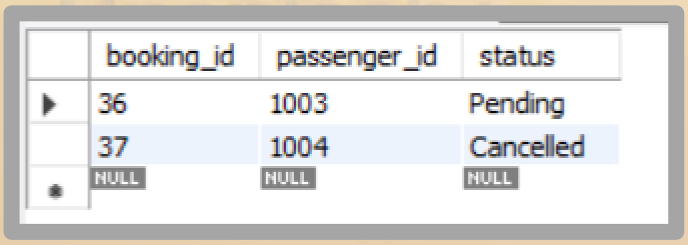


	booking_id	passenger_id	status
▶	36	1003	Pending
	37	1004	Cancelled
*	NULL	NULL	NULL

- Using NOT Operator

Q. Show all bookings that are not confirmed.

Select booking_id, passenger_id, status from Bookings where not status = 'Confirmed';

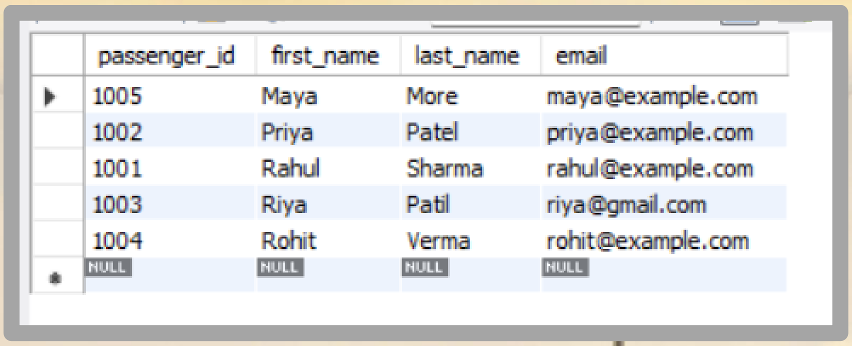


	booking_id	passenger_id	status
▶	36	1003	Pending
	37	1004	Cancelled
*	NULL	NULL	NULL

- Using NOT NULL Operator

Q. Show all passengers who have registered an email.

Select passenger_id, first_name, last_name, email from Passengers where email is not null;

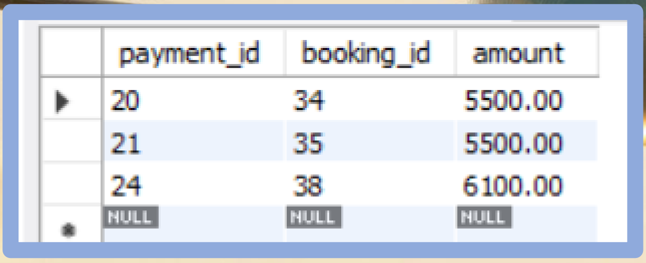


	passenger_id	first_name	last_name	email
▶	1005	Maya	More	maya@example.com
	1002	Priya	Patel	priya@example.com
	1001	Rahul	Sharma	rahul@example.com
	1003	Riya	Patil	riya@gmail.com
	1004	Rohit	Verma	rohit@example.com
*	NULL	NULL	NULL	NULL

- Using BETWEEN Operator

Q. Find all payments made between ₹5000 and ₹10000.

Select payment_id, booking_id, amount from Payment where amount between 5000 AND 8000;

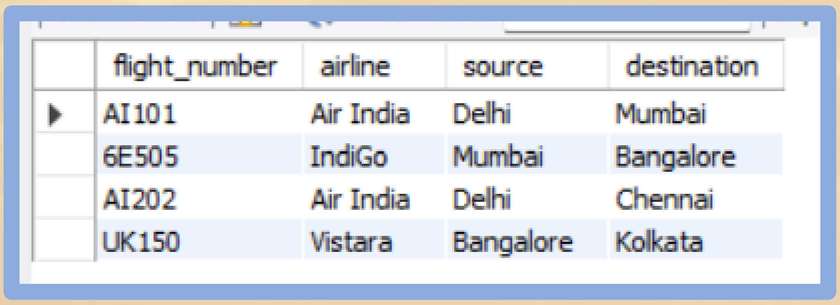


	payment_id	booking_id	amount
▶	20	34	5500.00
	21	35	5500.00
	24	38	6100.00
*	NULL	NULL	NULL

- Using IN Operator

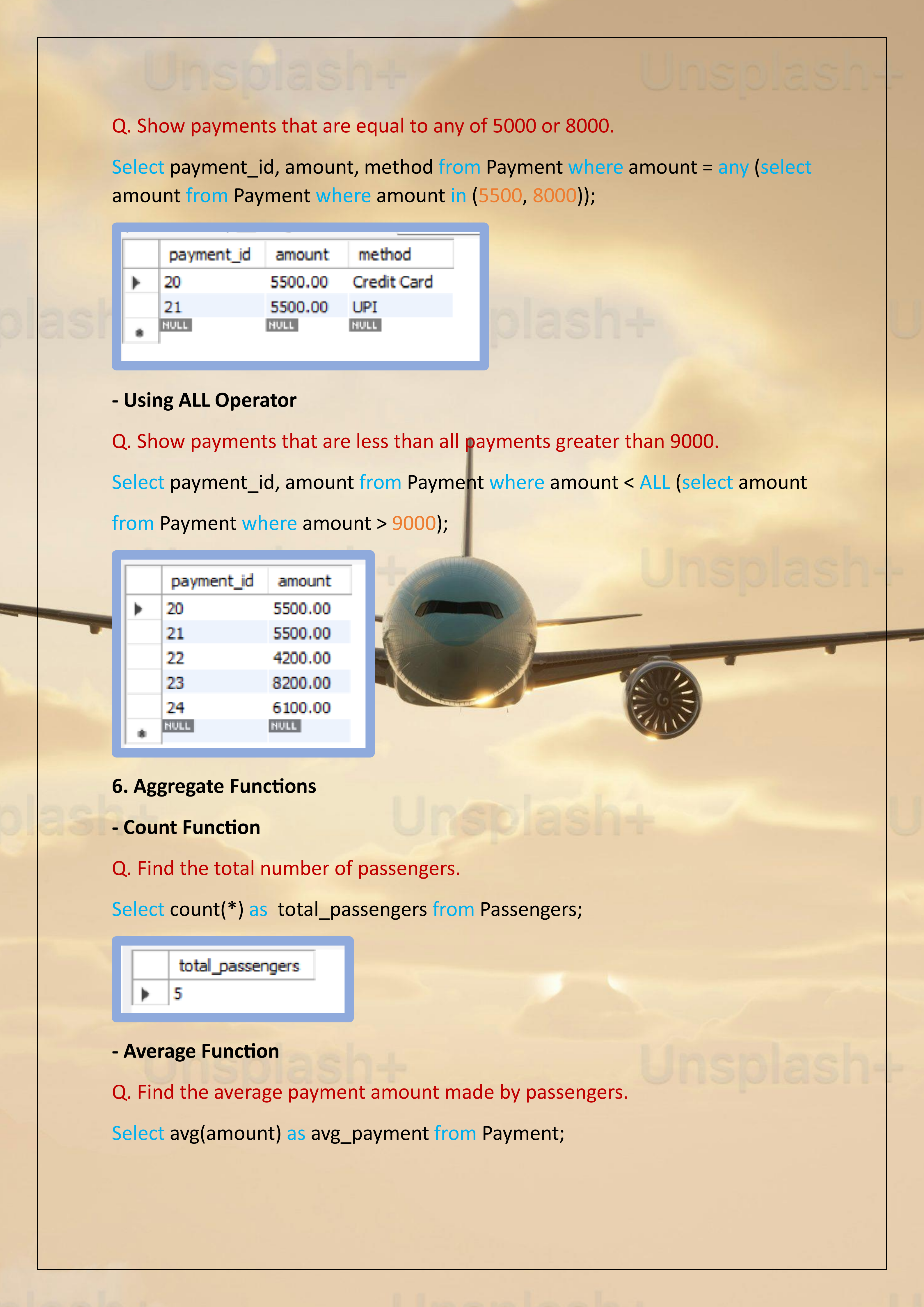
Q. Find flights that operate from Delhi, Mumbai, or Bangalore.

Select flight_number, airline, source, destination from Flights where source in ('Delhi', 'Mumbai', 'Bangalore');



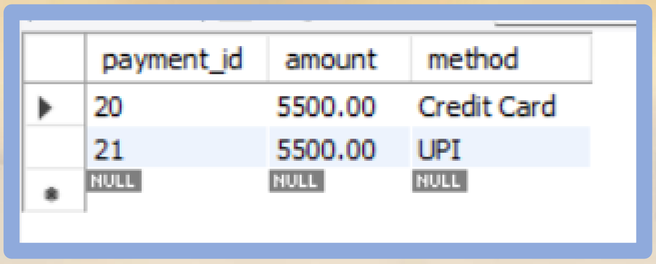
	flight_number	airline	source	destination
▶	AI101	Air India	Delhi	Mumbai
	6E505	IndiGo	Mumbai	Bangalore
	AI202	Air India	Delhi	Chennai
	UK150	Vistara	Bangalore	Kolkata

- Using ANY Operator



Q. Show payments that are equal to any of 5000 or 8000.

Select payment_id, amount, method from Payment where amount = any (select amount from Payment where amount in (5500, 8000));

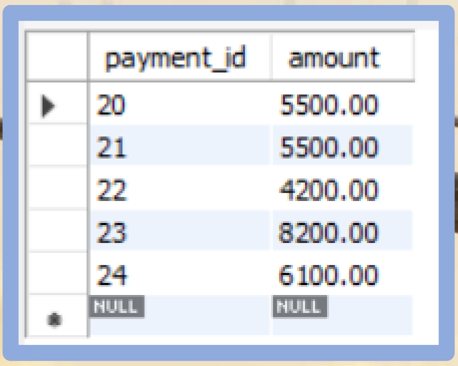


	payment_id	amount	method
▶	20	5500.00	Credit Card
	21	5500.00	UPI
*	NULL	NULL	NULL

- Using ALL Operator

Q. Show payments that are less than all payments greater than 9000.

Select payment_id, amount from Payment where amount < ALL (select amount from Payment where amount > 9000);



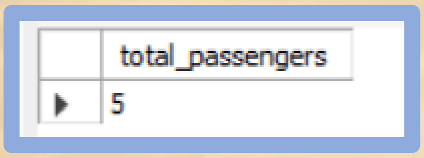
	payment_id	amount
▶	20	5500.00
	21	5500.00
	22	4200.00
	23	8200.00
	24	6100.00
*	NULL	NULL

6. Aggregate Functions

- Count Function

Q. Find the total number of passengers.

Select count(*) as total_passengers from Passengers;

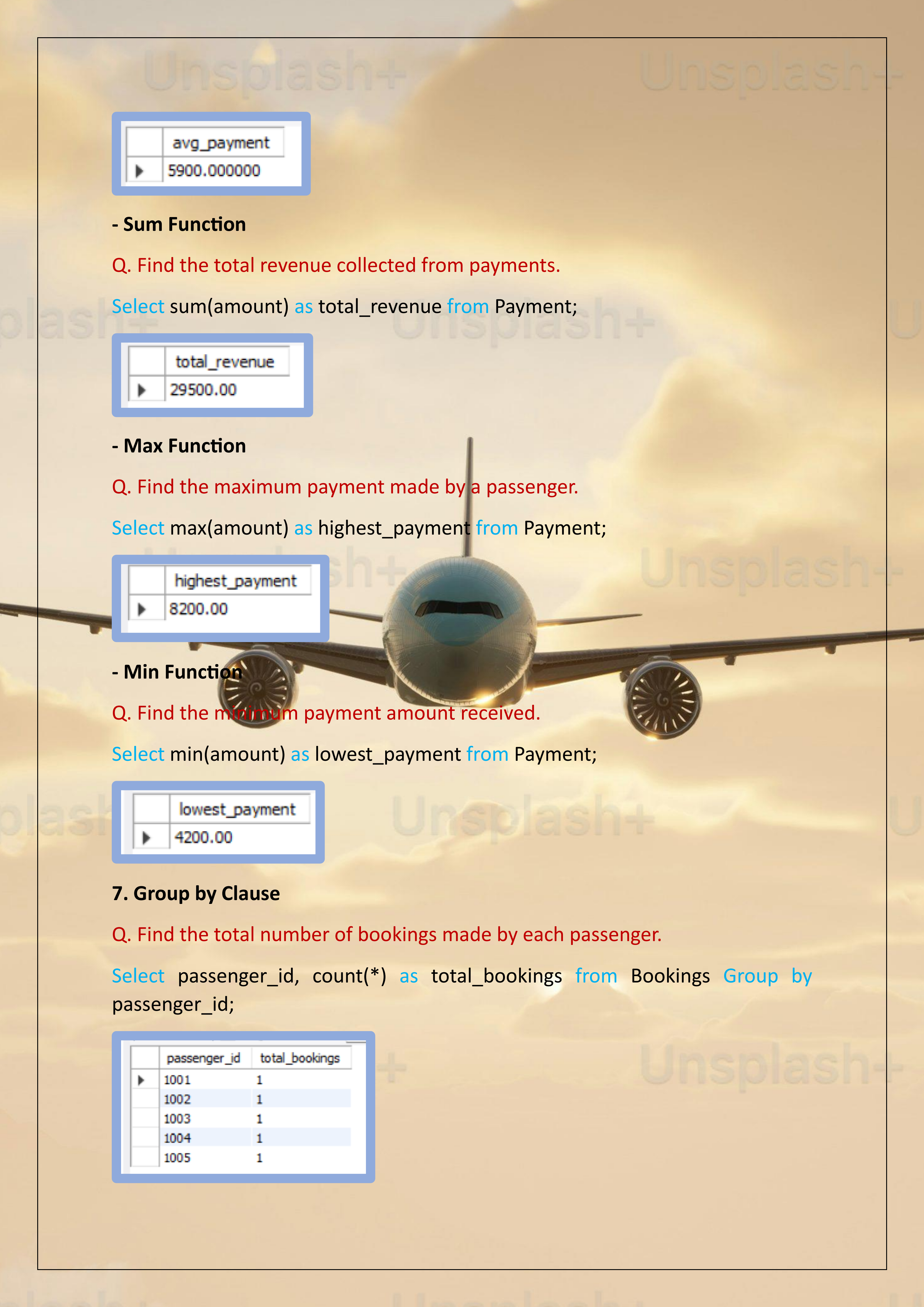


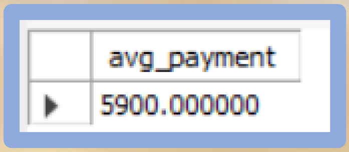
	total_passengers
▶	5

- Average Function

Q. Find the average payment amount made by passengers.

Select avg(amount) as avg_payment from Payment;



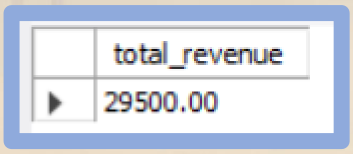


	avg_payment
▶	5900.000000

- Sum Function

Q. Find the total revenue collected from payments.

Select sum(amount) as total_revenue from Payment;

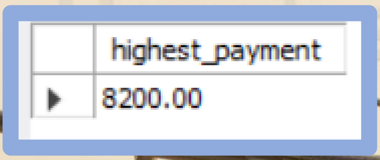


	total_revenue
▶	29500.00

- Max Function

Q. Find the maximum payment made by a passenger.

Select max(amount) as highest_payment from Payment;

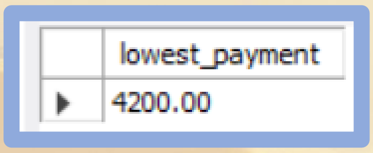


	highest_payment
▶	8200.00

- Min Function

Q. Find the minimum payment amount received.

Select min(amount) as lowest_payment from Payment;

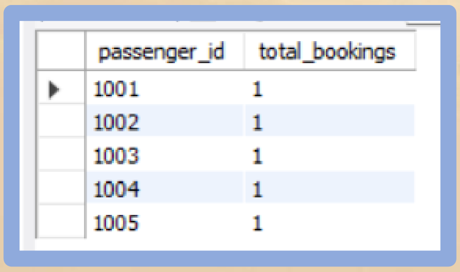


	lowest_payment
▶	4200.00

7. Group by Clause

Q. Find the total number of bookings made by each passenger.

Select passenger_id, count(*) as total_bookings from Bookings Group by passenger_id;

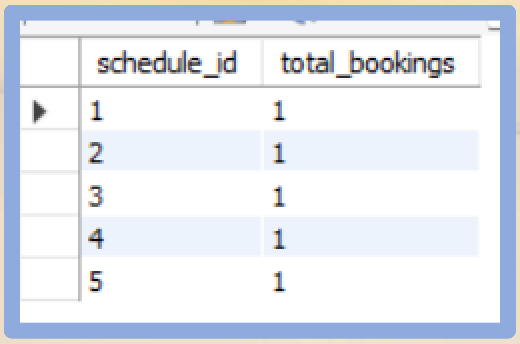


	passenger_id	total_bookings
▶	1001	1
	1002	1
	1003	1
	1004	1
	1005	1

8. Having Clause

Q. Find schedules that have more than 5 bookings.

```
Select schedule_id, count(*) as total_bookings from Bookings Group by schedule_id Having count(*) > 0;
```

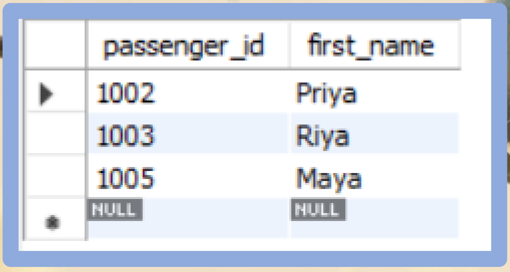


schedule_id	total_bookings
1	1
2	1
3	1
4	1
5	1

9. LIKE Operator

Q. Find passengers whose name ends with "n".

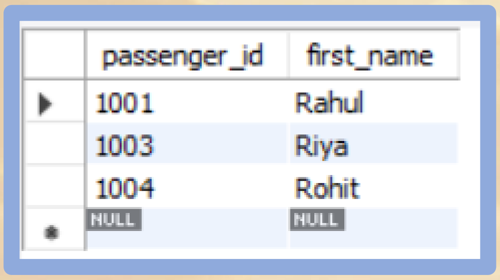
```
Select passenger_id, first_name from Passengers where first_name like '%n';
```



passenger_id	first_name
1002	Priya
1003	Riya
1005	Maya
NULL	NULL

Q. Find passengers whose name start with "r".

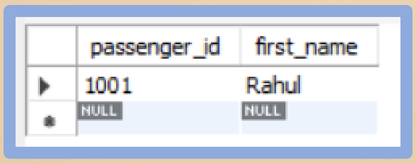
```
Select passenger_id, first_name from Passengers where first_name like 'r%';
```



passenger_id	first_name
1001	Rahul
1003	Riya
1004	Rohit
NULL	NULL

Q. Find passengers whose name contains "ra".

```
Select passenger_id, first_name from Passengers where first_name like '%ra%';
```

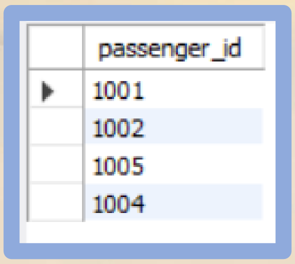


passenger_id	first_name
1001	Rahul
NULL	NULL

10. UNION

Q. Display a list of passenger IDs who booked either confirmed or cancelled bookings.

Select passenger_id from Bookings Where status = 'Confirmed' UNION select passenger_id from Bookings where status = 'Cancelled';



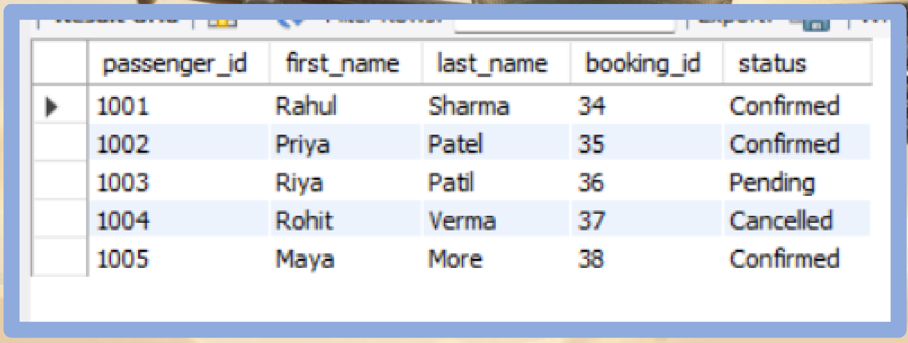
passenger_id
1001
1002
1005
1004

11. Joins

-Inner Join

Q. Show passenger details with their bookings.

Select p.passenger_id, p.first_name, p.last_name, b.booking_id, b.status from passengers p inner join bookings b on p.passenger_id = b.passenger_id;

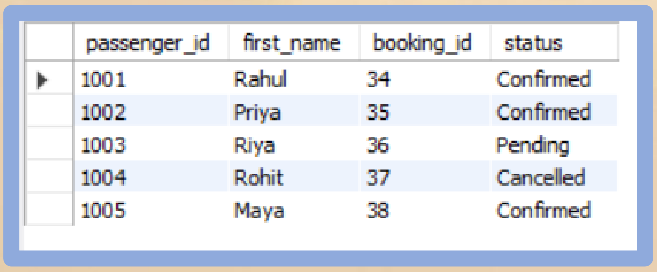


passenger_id	first_name	last_name	booking_id	status
1001	Rahul	Sharma	34	Confirmed
1002	Priya	Patel	35	Confirmed
1003	Riya	Patil	36	Pending
1004	Rohit	Verma	37	Cancelled
1005	Maya	More	38	Confirmed

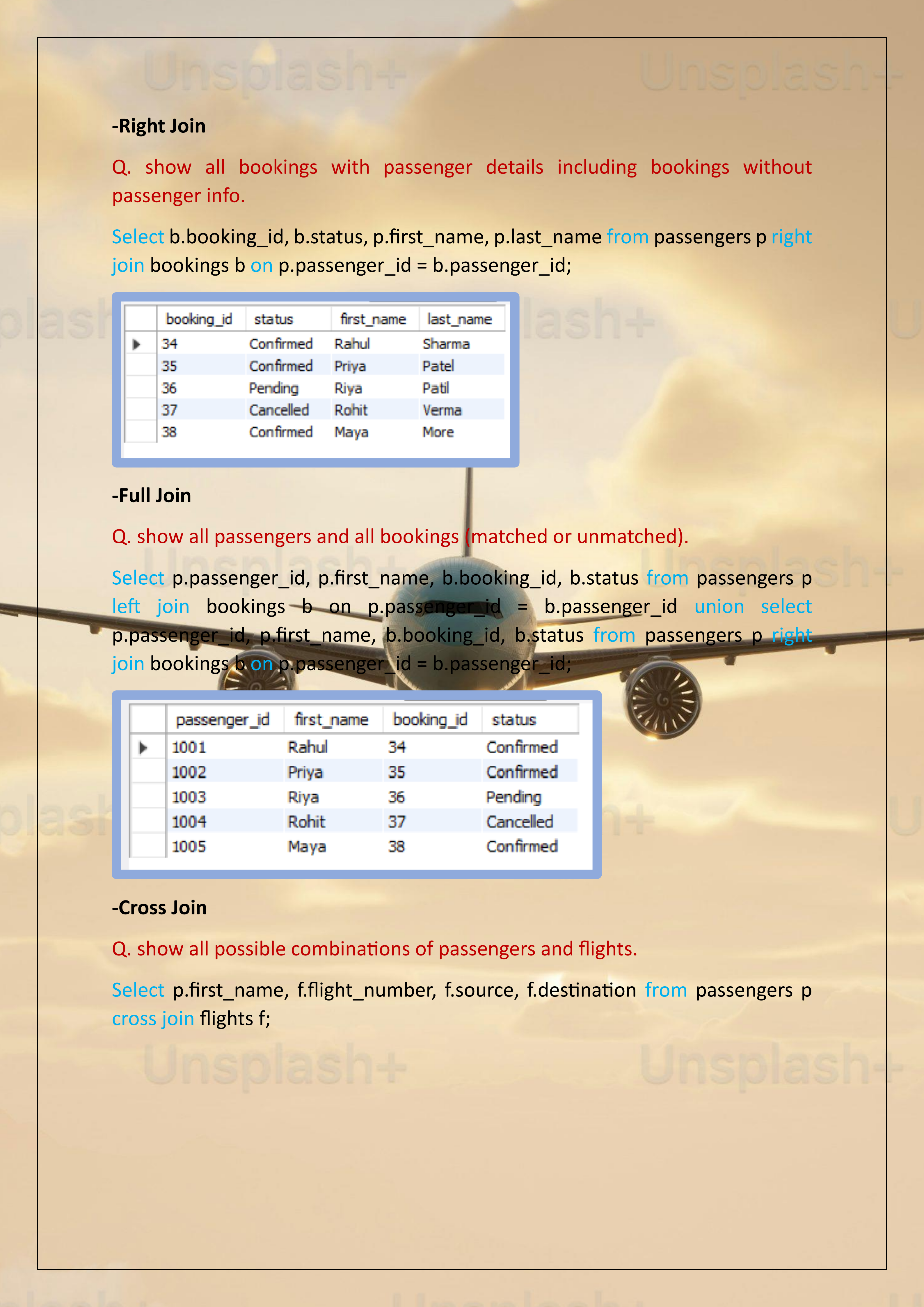
-Left Join

Q. show all passengers and their bookings including passengers who have not booked yet.

Select p.passenger_id, p.first_name, b.booking_id, b.status from passengers p left join bookings b on p.passenger_id = b.passenger_id;



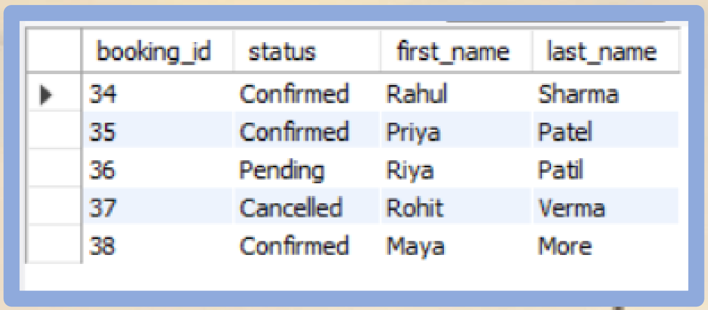
passenger_id	first_name	booking_id	status
1001	Rahul	34	Confirmed
1002	Priya	35	Confirmed
1003	Riya	36	Pending
1004	Rohit	37	Cancelled
1005	Maya	38	Confirmed



-Right Join

Q. show all bookings with passenger details including bookings without passenger info.

Select b.booking_id, b.status, p.first_name, p.last_name from passengers p right join bookings b on p.passenger_id = b.passenger_id;

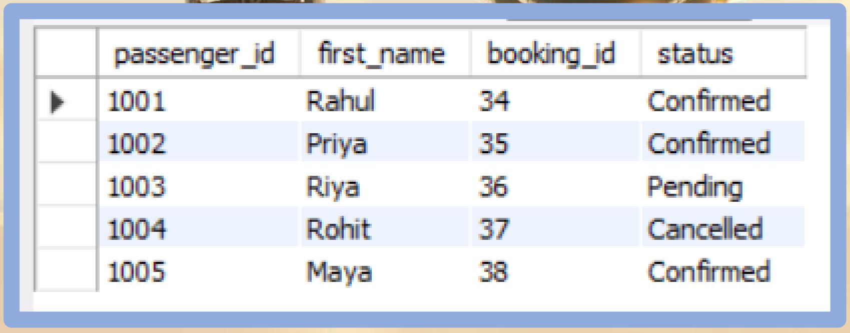


booking_id	status	first_name	last_name
34	Confirmed	Rahul	Sharma
35	Confirmed	Priya	Patel
36	Pending	Riya	Patil
37	Cancelled	Rohit	Verma
38	Confirmed	Maya	More

-Full Join

Q. show all passengers and all bookings (matched or unmatched).

Select p.passenger_id, p.first_name, b.booking_id, b.status from passengers p left join bookings b on p.passenger_id = b.passenger_id union select p.passenger_id, p.first_name, b.booking_id, b.status from passengers p right join bookings b on p.passenger_id = b.passenger_id;

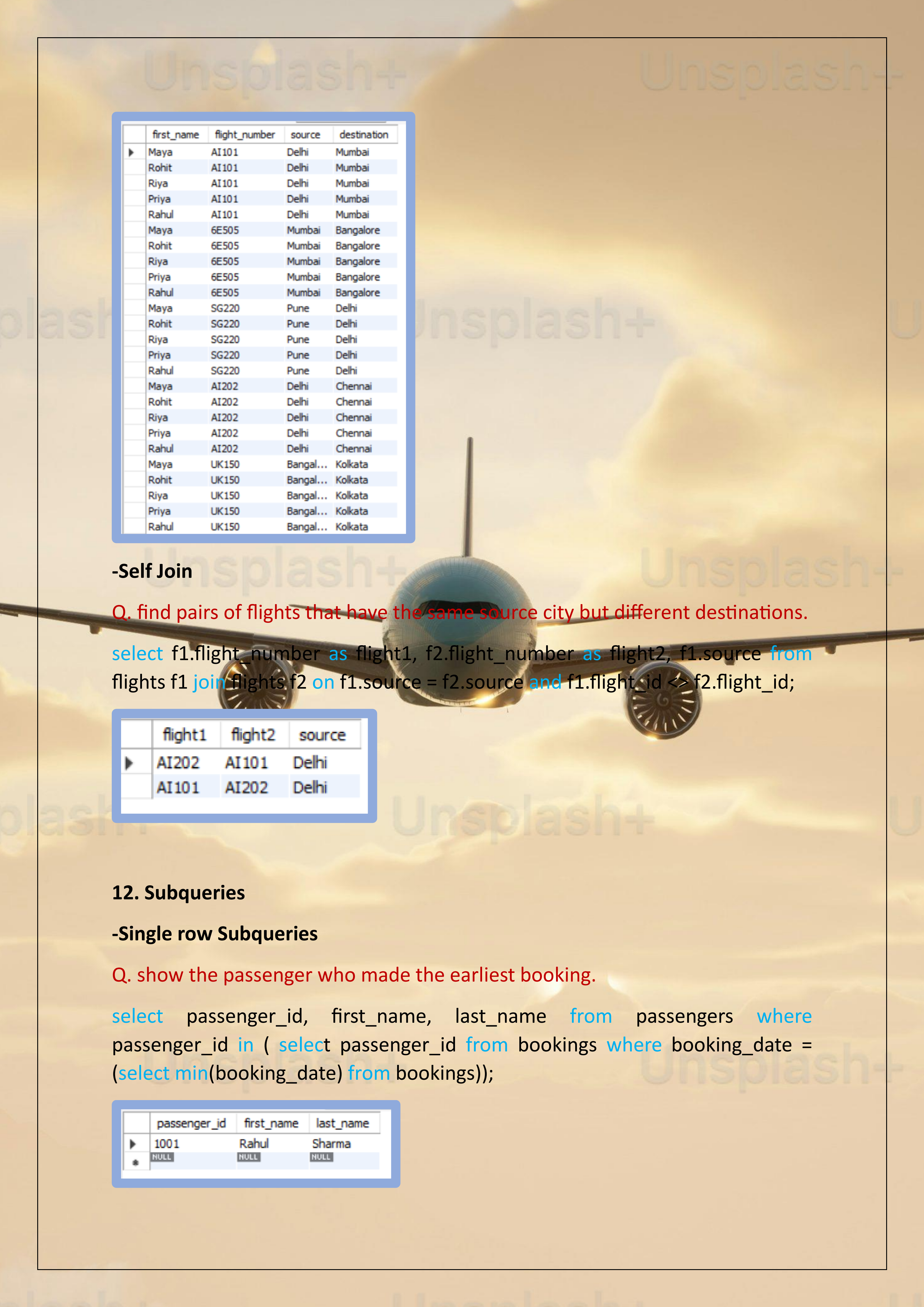


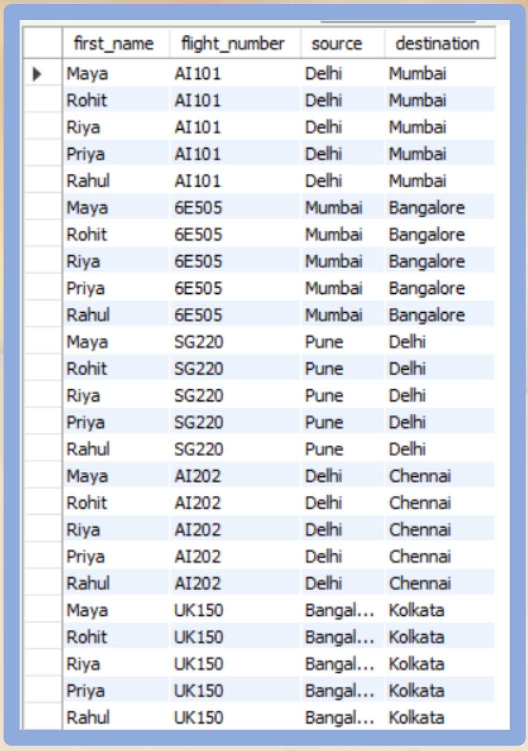
passenger_id	first_name	booking_id	status
1001	Rahul	34	Confirmed
1002	Priya	35	Confirmed
1003	Riya	36	Pending
1004	Rohit	37	Cancelled
1005	Maya	38	Confirmed

-Cross Join

Q. show all possible combinations of passengers and flights.

Select p.first_name, f.flight_number, f.source, f.destination from passengers p cross join flights f;



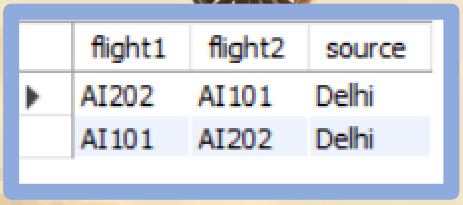


first_name	flight_number	source	destination
Maya	AI101	Delhi	Mumbai
Rohit	AI101	Delhi	Mumbai
Riya	AI101	Delhi	Mumbai
Priya	AI101	Delhi	Mumbai
Rahul	AI101	Delhi	Mumbai
Maya	6E505	Mumbai	Bangalore
Rohit	6E505	Mumbai	Bangalore
Riya	6E505	Mumbai	Bangalore
Priya	6E505	Mumbai	Bangalore
Rahul	6E505	Mumbai	Bangalore
Maya	SG220	Pune	Delhi
Rohit	SG220	Pune	Delhi
Riya	SG220	Pune	Delhi
Priya	SG220	Pune	Delhi
Rahul	SG220	Pune	Delhi
Maya	AI202	Delhi	Chennai
Rohit	AI202	Delhi	Chennai
Riya	AI202	Delhi	Chennai
Priya	AI202	Delhi	Chennai
Rahul	AI202	Delhi	Chennai
Maya	UK150	Bangal...	Kolkata
Rohit	UK150	Bangal...	Kolkata
Riya	UK150	Bangal...	Kolkata
Priya	UK150	Bangal...	Kolkata
Rahul	UK150	Bangal...	Kolkata

-Self Join

Q. find pairs of flights that have the same source city but different destinations.

select f1.flight_number as flight1, f2.flight_number as flight2, f1.source from flights f1 join flights f2 on f1.source = f2.source and f1.flight_id <> f2.flight_id;



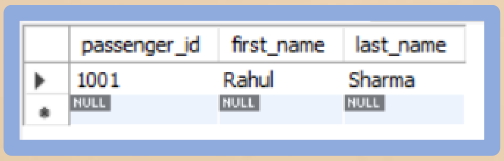
flight1	flight2	source
AI202	AI101	Delhi
AI101	AI202	Delhi

12. Subqueries

-Single row Subqueries

Q. show the passenger who made the earliest booking.

select passenger_id, first_name, last_name from passengers where passenger_id in (select passenger_id from bookings where booking_date = (select min(booking_date) from bookings));

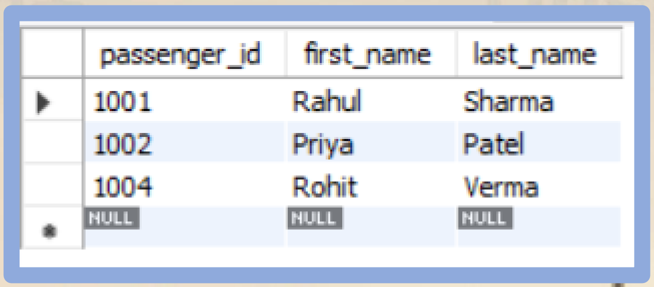


passenger_id	first_name	last_name
1001	Rahul	Sharma
NULL	NULL	NULL

-Multiple row Subqueries

Q. show passengers who booked flights from delhi or mumbai.

Select passenger_id, first_name, last_name from passengers where passenger_id in (select passenger_id from bookings b join schedules s on b.schedule_id = s.schedule_id join flights f on s.flight_id = f.flight_id where f.source in ('delhi', 'mumbai'));

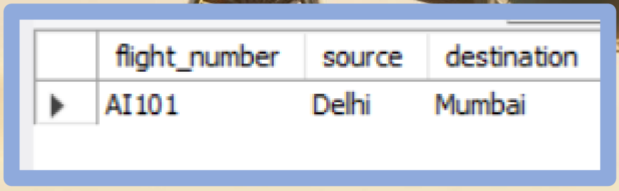


	passenger_id	first_name	last_name
▶	1001	Rahul	Sharma
	1002	Priya	Patel
	1004	Rohit	Verma
*	NULL	NULL	NULL

- Multiple column Subqueries

Q. find flights that have the same source and destination as flight 'ai101'.

Select flight_number, source, destination from flights where (source, destination) = (select source, destination from flights where flight_number = 'ai101');



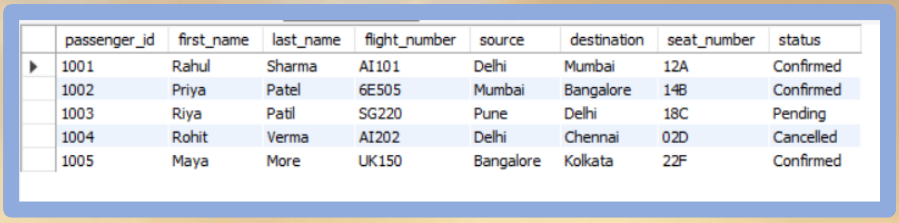
	flight_number	source	destination
▶	AI101	Delhi	Mumbai

13. View

Q. Create a view to show passenger booking details with flight information.

create view passenger_bookings as select p.passenger_id, p.first_name, p.last_name, f.flight_number, f.source, f.destination, b.seat_number, b.status from passengers p join bookings b on p.passenger_id = b.passenger_id join schedules s on b.schedule_id = s.schedule_id join flights f on s.flight_id = f.flight_id;

select * from passenger_bookings;



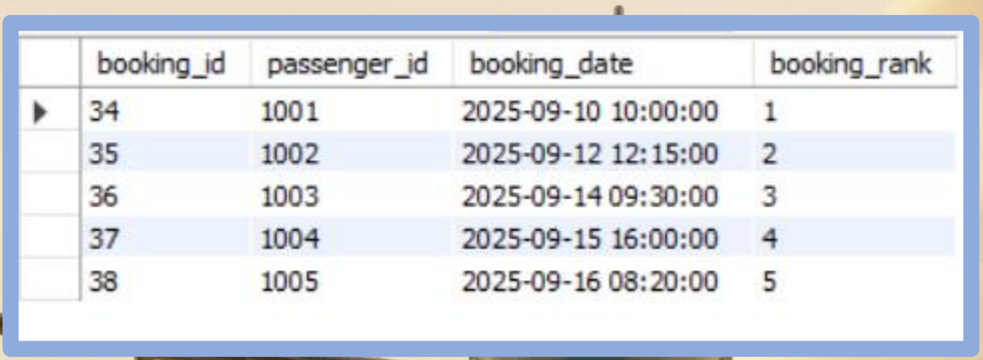
	passenger_id	first_name	last_name	flight_number	source	destination	seat_number	status
▶	1001	Rahul	Sharma	AI101	Delhi	Mumbai	12A	Confirmed
	1002	Priya	Patel	6E505	Mumbai	Bangalore	14B	Confirmed
	1003	Riya	Patil	SG220	Pune	Delhi	18C	Pending
	1004	Rohit	Verma	AI202	Delhi	Chennai	02D	Cancelled
	1005	Maya	More	UK150	Bangalore	Kolkata	22F	Confirmed

14. Windows Functions

1. Row Number()

Q. assign a unique rank to each booking ordered by booking_date.

Select booking_id, passenger_id, booking_date, row_number() over (order by booking_date) as booking_rank from bookings;

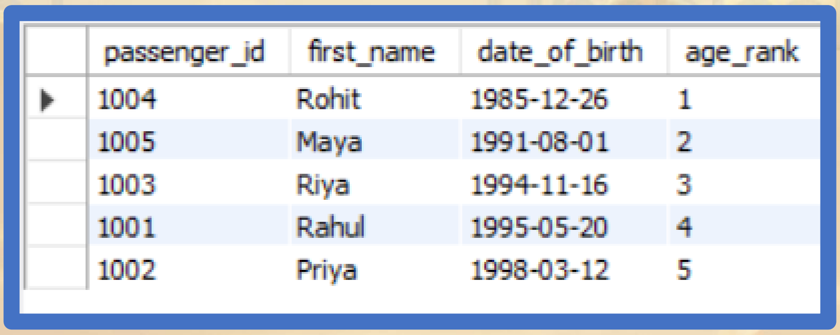


	booking_id	passenger_id	booking_date	booking_rank
▶	34	1001	2025-09-10 10:00:00	1
	35	1002	2025-09-12 12:15:00	2
	36	1003	2025-09-14 09:30:00	3
	37	1004	2025-09-15 16:00:00	4
	38	1005	2025-09-16 08:20:00	5

2. Rank()

Q. Rank passengers based on their date of birth (oldest first).

select passenger_id, first_name, date_of_birth,rank() over (order by date_of_birth asc) as age_rank from passengers;

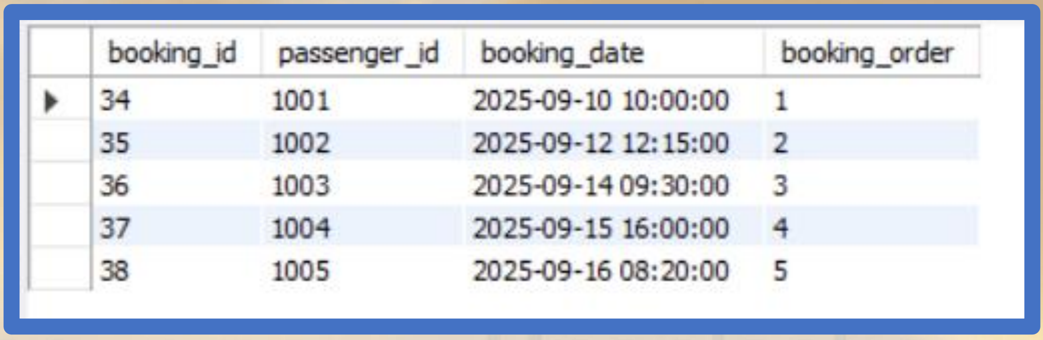


	passenger_id	first_name	date_of_birth	age_rank
▶	1004	Rohit	1985-12-26	1
	1005	Maya	1991-08-01	2
	1003	Riya	1994-11-16	3
	1001	Rahul	1995-05-20	4
	1002	Priya	1998-03-12	5

3. dense rank()

Q. give dense rank to bookings based on booking date (earliest first).

select booking_id, passenger_id, booking_date,dense_rank() over (order by booking_date asc) as booking_order from bookings;

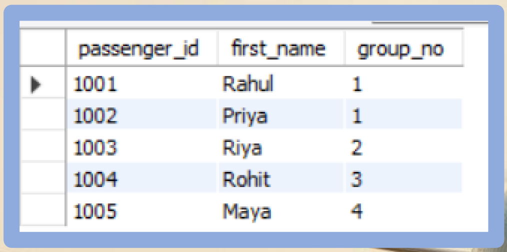


booking_id	passenger_id	booking_date	booking_order
34	1001	2025-09-10 10:00:00	1
35	1002	2025-09-12 12:15:00	2
36	1003	2025-09-14 09:30:00	3
37	1004	2025-09-15 16:00:00	4
38	1005	2025-09-16 08:20:00	5

4. ntile(n)

Q. divide passengers into 4 groups based on their passenger_id.

Select passenger_id, first_name, ntile(4) over (order by passenger_id) as group_no from passengers;

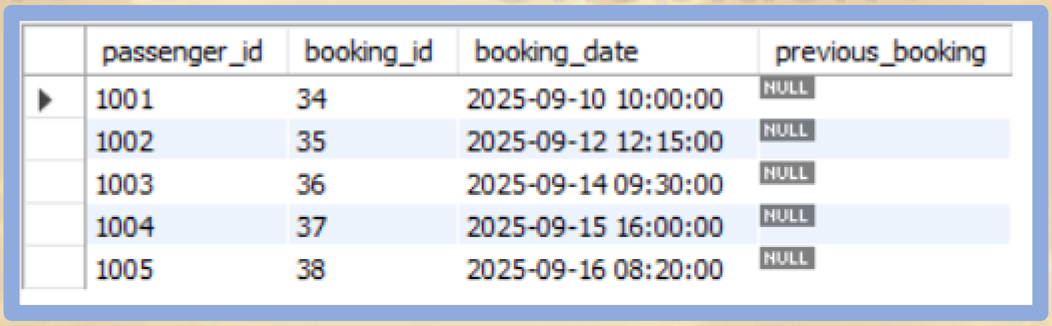


passenger_id	first_name	group_no
1001	Rahul	1
1002	Priya	1
1003	Riya	2
1004	Rohit	3
1005	Maya	4

5. Lag()

Q. show each booking date and the previous booking date of the same passenger.

Select passenger_id, booking_id, booking_date, lag(booking_date) over (partition by passenger_id order by booking_date) as previous_booking from bookings;

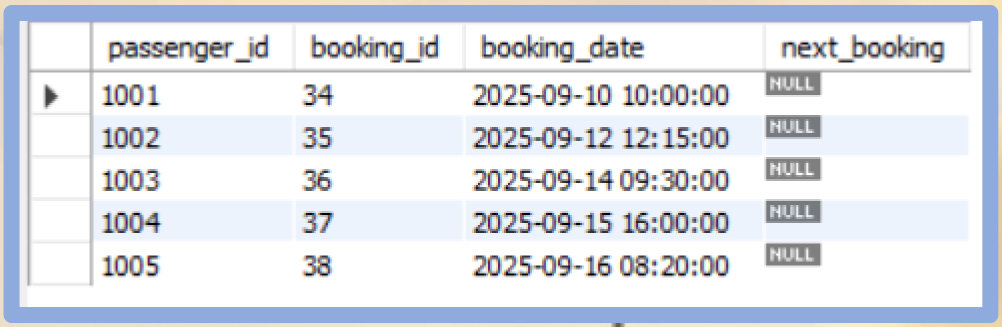


passenger_id	booking_id	booking_date	previous_booking
1001	34	2025-09-10 10:00:00	NULL
1002	35	2025-09-12 12:15:00	NULL
1003	36	2025-09-14 09:30:00	NULL
1004	37	2025-09-15 16:00:00	NULL
1005	38	2025-09-16 08:20:00	NULL

6. Lead()

Q. show each booking date and the next booking date of the same passenger.

```
Select passenger_id, booking_id, booking_date,lead(booking_date) over (partition by passenger_id order by booking_date) as next_booking from bookings;
```

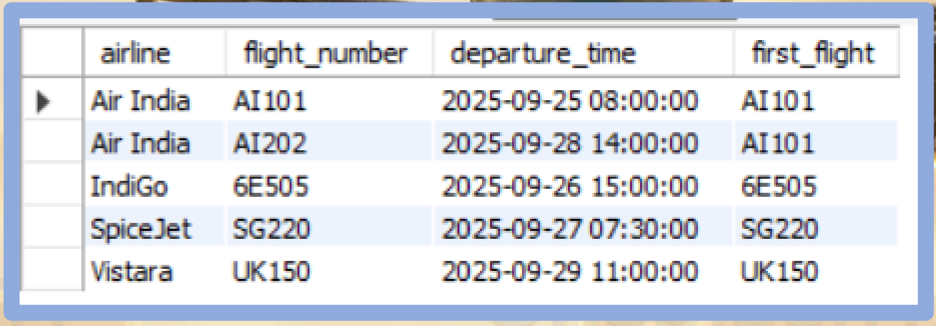


	passenger_id	booking_id	booking_date	next_booking
▶	1001	34	2025-09-10 10:00:00	NULL
	1002	35	2025-09-12 12:15:00	NULL
	1003	36	2025-09-14 09:30:00	NULL
	1004	37	2025-09-15 16:00:00	NULL
	1005	38	2025-09-16 08:20:00	NULL

7. first_value()

Q. find the first flight scheduled for each airline.

```
Select airline, flight_number, departure_time,first_value(flight_number) over (partition by airline order by departure_time) as first_flight from flights;
```

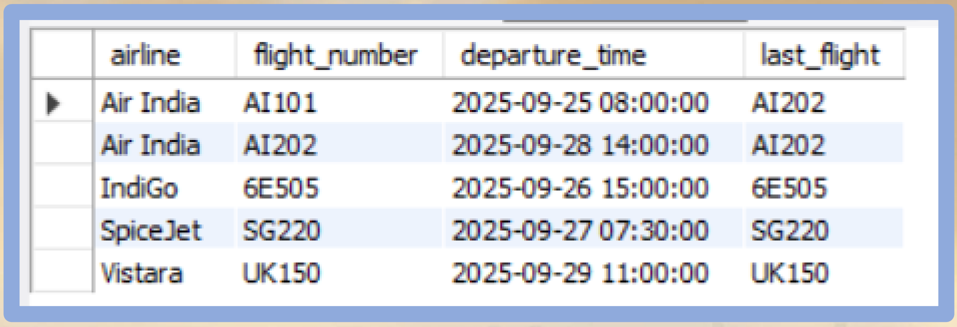


	airline	flight_number	departure_time	first_flight
▶	Air India	AI101	2025-09-25 08:00:00	AI101
	Air India	AI202	2025-09-28 14:00:00	AI101
	IndiGo	6E505	2025-09-26 15:00:00	6E505
	SpiceJet	SG220	2025-09-27 07:30:00	SG220
	Vistara	UK150	2025-09-29 11:00:00	UK150

8. Last value()

Q. find the last flight scheduled for each airline.

```
Select airline, flight_number, departure_time,last_value(flight_number) over (partition by airline order by departure_time rows between unbounded preceding and unbounded following) as last_flight from flights;
```

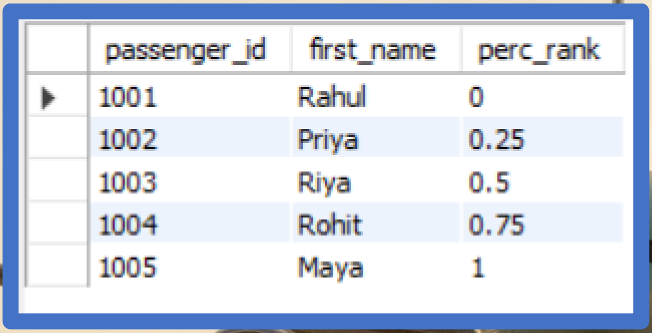


airline	flight_number	departure_time	last_flight
Air India	AI101	2025-09-25 08:00:00	AI202
Air India	AI202	2025-09-28 14:00:00	AI202
IndiGo	6E505	2025-09-26 15:00:00	6E505
SpiceJet	SG220	2025-09-27 07:30:00	SG220
Vistara	UK150	2025-09-29 11:00:00	UK150

9. Percent rank()

Q. calculate percent rank of passengers based on passenger_id.

select passenger_id, first_name, percent_rank() over (order by passenger_id) as perc_rank from passengers;



passenger_id	first_name	perc_rank
1001	Rahul	0
1002	Priya	0.25
1003	Riya	0.5
1004	Rohit	0.75
1005	Maya	1