

SQL PROJECT

INDIAN AIRLINE SYSTEM

ABSTRACT-

The Indian Airline System is a comprehensive software solution designed to streamline and enhance the operations of airlines operating within India. The Indian Airline system is designed to enhance the overall travel experience for passengers, optimize operational efficiency for airlines, and improve coordination among airport authorities .it reflect a commitment to providing a safe, efficient, and passengers-centric air travel experience within India.



Aim Of the Project-

The aim of the Indian Airline System Project in SQL is to provide a robust and efficient database management system that supports the daily operations of the Airlines, enhance the passenger experience and aids in the efficient management of flight schedules, reservations, and overall airline operations.

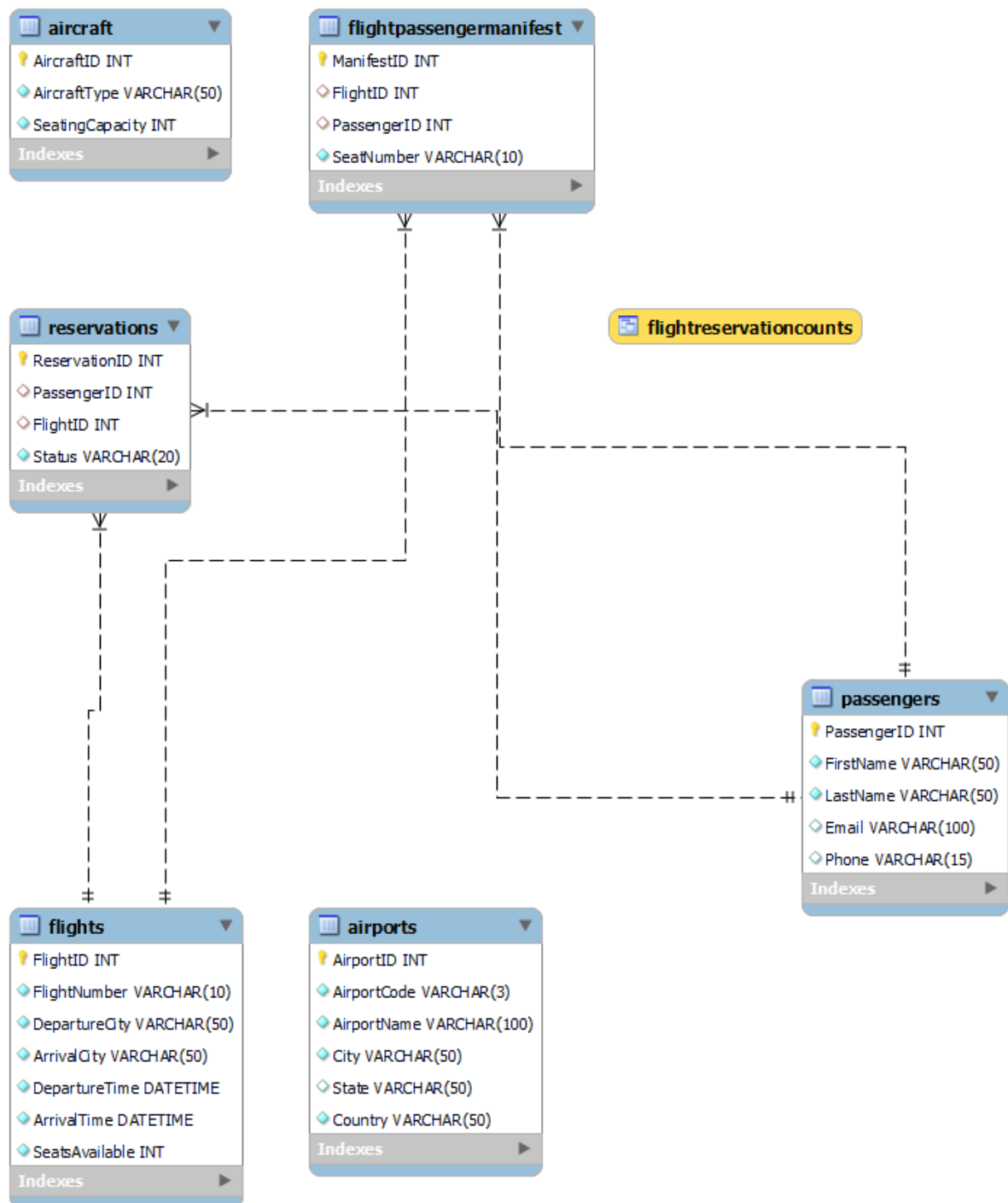
Objective of the project –

The Objective of this project for Indian Airline System can be multifaceted and can encompass several key goals and objectives. below are some common objectives that a SQL project for an Indian Airline System might aim to achieve.

Introduction-

The Indian Airline System project is an all- encompassing solution that addresses the unique needs and challenges of the airline industry in India. It streamlines airline operations. Enhance passenger services, and empowers the airline to make data- driven decision, all while ensuring security and compliance with industry standard.

ER Diagram-



STRUCTURE OF TABLES=

Flights-

13 • DESC Flights;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Field	Type	Null	Key	Default	Extra
▶	FlightID	int	NO	PRI	NULL	
	FlightNumber	varchar(10)	NO		NULL	
	DepartureCity	varchar(50)	NO		NULL	
	ArrivalCity	varchar(50)	NO		NULL	
	DepartureTime	datetime	NO		NULL	
	ArrivalTime	datetime	NO		NULL	
	SeatsAvailable	int	NO		NULL	

Passengers-

32 • desc passengers;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Field	Type	Null	Key	Default	Extra
▶	PassengerID	int	NO	PRI	NULL	
	FirstName	varchar(50)	NO		NULL	
	LastName	varchar(50)	NO		NULL	
	Email	varchar(100)	YES		NULL	
	Phone	varchar(15)	YES		NULL	

Result 12 x

Reservations-

53 • desc reservations;

54

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

IA

	Field	Type	Null	Key	Default	Extra
▶	ReservationID	int	NO	PRI	NULL	
	PassengerID	int	YES	MUL	NULL	
	FlightID	int	YES	MUL	NULL	
	Status	varchar(20)	NO		NULL	

Result 14 x

Flight Passenger Manifest-

```
110 • desc FlightPassengerManifest;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content: [IA](#)

	Field	Type	Null	Key	Default	Extra
▶	ManifestID	int	NO	PRI	NULL	
	FlightID	int	YES	MUL	NULL	
	PassengerID	int	YES	MUL	NULL	
	SeatNumber	varchar(10)	NO		NULL	


Result 18


×


Airports-

```
74 • desc Airports;
```

Result Grid


Filter Rows:

Export:


Wrap Cell Content:


	Field	Type	Null	Key	Default	Extra
▶	AirportID	int	NO	PRI	NULL	
	AirportCode	varchar(3)	NO		NULL	
	AirportName	varchar(100)	NO		NULL	
	City	varchar(50)	NO		NULL	
	State	varchar(50)	YES		NULL	
	Country	varchar(50)	NO		NULL	

Result 15

×

Aircrafts-

```
91 • desc Aircraft;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Field	Type	Null	Key	Default	Extra
▶	AircraftID	int	NO	PRI	NULL	
	AircraftType	varchar(50)	NO		NULL	
	SeatingCapacity	int	NO		NULL	

Result 17
×

Content of table-

Flights-

```
23 • select * from flights;
```

```
24
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

IA

	FlightID	FlightNumber	DepartureCity	ArrivalCity	DepartureTime	ArrivalTime	SeatsAvailable
▶	1	AI101	Mumbai	Delhi	2023-10-15 08:00:00	2023-10-15 10:30:00	150
	2	SG202	Delhi	Bangalore	2023-10-15 12:00:00	2023-10-15 14:30:00	120
	3	6E303	Bangalore	Chennai	2023-10-15 16:00:00	2023-10-15 18:30:00	100
	4	AI404	Chennai	Kolkata	2023-10-15 20:00:00	2023-10-15 22:30:00	130
	5	SG505	Kolkata	Mumbai	2023-10-16 08:00:00	2023-10-16 10:30:00	110
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

flights 21 ×

Passengers-

```
43 • select * from Passengers;
```

PassengerID	FirstName	LastName	Email	Phone
1	John	Doe	john.doe@example.com	+1-555-123-4567
2	Jane	Smith	jane.smith@example.com	+1-555-987-6543
3	Robert	Johnson	robert.j@example.com	+1-555-777-8888
4	Emily	Brown	emily.b@example.com	+1-555-444-5555
5	David	Wilson	david.w@example.com	+1-555-999-1111
6	Sarah	Davis	sarah.d@example.com	+1-555-333-7777
*	NULL	NULL	NULL	NULL

Passengers 22 × Apply

Reservations-

```
64 • select * from reservations;
```

```
65
```

ReservationID	PassengerID	FlightID	Status
1	1	1	Confirmed
2	2	3	Confirmed
3	3	2	Confirmed
4	4	4	Confirmed
5	5	5	Confirmed
6	6	1	Confirmed
*	NULL	NULL	NULL

reservations 23 ×

Airports-

84 • `select * from Airports;`

85

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	AirportID	AirportCode	AirportName	City	State	Country
▶	1	BOM	Chhatrapati Shivaji International Airport	Mumbai	Maharashtra	India
	2	DEL	Indira Gandhi International Airport	Delhi	Delhi	India
	3	BLR	Kempegowda International Airport	Bangalore	Karnataka	India
	4	MAA	Chennai International Airport	Chennai	Tamil Nadu	India
	5	CCU	Netaji Subhas Chandra Bose International Airport	Kolkata	West Bengal	India
	6	HYD	Rajiv Gandhi International Airport	Hyderabad	Telangana	India
•	NULL	NULL	NULL	NULL	NULL	NULL

Airports 24

×

Apply

Aircraft-

100 • `select * from Aircraft;`

101

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	AircraftID	AircraftType	SeatingCapacity
▶	1	Boeing 737	150
	2	Airbus A320	180
	3	Boeing 777	300
	4	Airbus A350	250
	5	Embraer E190	100
*	NULL	NULL	NULL

Aircraft 25

×

Apply

Flight Passenger Manifest-

120 • `select * from FlightPassengerManifest;`

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	ManifestID	FlightID	PassengerID	SeatNumber
▶	1	1	1	1A
	2	1	2	1B
	3	2	3	2A
	4	2	4	2B
	5	3	5	3A
	6	3	6	3B
✱	NULL	NULL	NULL	NULL




FlightPassengerManifest 26 ✕

Apply




JOIN-

Inner join-

```
121      -- joins--
122      -- inner joins
123      -- Retrieve Passenger and Flight Information for a Reservation
124      • SELECT
125          P.FirstName,
126          P.LastName,
127          F.FlightNumber,
128          F.DepartureCity,
129          F.ArrivalCity
130      FROM Passengers AS P
131      INNER JOIN Reservations AS R ON P.PassengerID = R.PassengerID
132      INNER JOIN Flights AS F ON R.FlightID = F.FlightID
133      WHERE R.ReservationID = 1; -- Replace 1 with the desired ReservationID
134
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 					
	FirstName	LastName	FlightNumber	DepartureCity	ArrivalCity
▶	John	Doe	AI101	Mumbai	Delhi

```
135      -- List All Passengers on a Specific Flight
136      -- Replace 'AI101' with the desired FlightNumber
137      • SELECT
138          P.FirstName,
139          P.LastName
140      FROM Passengers AS P
141      INNER JOIN Reservations AS R ON P.PassengerID = R.PassengerID
142      INNER JOIN Flights AS F ON R.FlightID = F.FlightID
143      WHERE F.FlightNumber = 'AI101';
144
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 		
	FirstName	LastName
▶	John	Doe
	Sarah	Davis

Left Join

```
144
145      -- left join
146      -- Retrieve Passengers with Their Reservations (Including Unreserved Passengers)
147 •   SELECT
148         P.FirstName,
149         P.LastName,
150         R.ReservationID
151   FROM Passengers AS P
152  LEFT JOIN Reservations AS R ON P.PassengerID = R.PassengerID;
153
```

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

	FirstName	LastName	ReservationID
▶	John	Doe	1
	Jane	Smith	2
	Robert	Johnson	3
	Emily	Brown	4
	David	Wilson	5
	Sarah	Davis	6

Result 7 ×

```
154      -- List All Flights and Their Reservations (Including Flights with No Reservations)
155 •   SELECT
156         F.FlightNumber,
157         F.DepartureCity,
158         F.ArrivalCity,
159         R.ReservationID
160   FROM Flights AS F
161  LEFT JOIN Reservations AS R ON F.FlightID = R.FlightID;
162
```

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

	FlightNumber	DepartureCity	ArrivalCity	ReservationID
▶	AI101	Mumbai	Delhi	1
	AI101	Mumbai	Delhi	6
	SG202	Delhi	Bangalore	3
	6E303	Bangalore	Chennai	2
	AI404	Chennai	Kolkata	4
	SG505	Kolkata	Mumbai	5

Result 8 ×

Right Join-

```
163      -- right join
164      -- Retrieve Flights and Their Reservations (Including Flights with No Reservations)
165 •    SELECT
166          F.FlightNumber,
167          F.DepartureCity,
168          F.ArrivalCity,
169          R.ReservationID
170      FROM Flights AS F
171      RIGHT JOIN Reservations AS R ON F.FlightID = R.FlightID;
172
```

Result Grid				
Filter Rows: <input type="text"/> Export: Wrap Cell Content:				
	FlightNumber	DepartureCity	ArrivalCity	ReservationID
▶	AI101	Mumbai	Delhi	1
	AI101	Mumbai	Delhi	6
	SG202	Delhi	Bangalore	3
	6E303	Bangalore	Chennai	2
	AI404	Chennai	Kolkata	4
	SG505	Kolkata	Mumbai	5

Result 9 x

```
173      -- List All Reservations and Their Associated Passengers (Including Passengers with No Reservations)
174 •    SELECT
175          R.ReservationID,
176          P.FirstName,
177          P.LastName
178      FROM Reservations AS R
179      RIGHT JOIN Passengers AS P ON R.PassengerID = P.PassengerID;
```

Result Grid			
Filter Rows: <input type="text"/> Export: Wrap Cell Content:			
	ReservationID	FirstName	LastName
▶	1	John	Doe
	2	Jane	Smith
	3	Robert	Johnson
	4	Emily	Brown
	5	David	Wilson
	6	Sarah	Davis

Result 10 x

Read Only

Cross Join-

```
182 -- cross join
183 -- Combining All Flights and Passengers
184 • SELECT F.FlightNumber, P.FirstName
185 FROM Flights F
186 CROSS JOIN Passengers P;
```

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

	FlightNumber	FirstName
▶	SG505	John
	AI404	John
	6E303	John
	SG202	John
	AI101	John

Result 11 ×

```
188 -- Combining All Flights and Airports
189 • SELECT F.FlightNumber, A.AirportName
190 FROM Flights F
191 CROSS JOIN Airports A;
192
193
---
```

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

	FlightNumber	AirportName
▶	SG505	Chhatrapati Shivaji International Airport
	AI404	Chhatrapati Shivaji International Airport
	6E303	Chhatrapati Shivaji International Airport
	SG202	Chhatrapati Shivaji International Airport
	AI101	Chhatrapati Shivaji International Airport
	SG505	Indira Gandhi International Airport

Result 12 ×

SUB-QUERY-comparison operator-

```
188      -- Combining All Flights and Airports
189  •    SELECT F.FlightNumber, A.AirportName
190      FROM Flights F
191      CROSS JOIN Airports A;
192
193
---
```



Result Grid  Filter Rows: Export:  Wrap Cell Content: 

	FlightNumber	AirportName
▶	SG505	Chhatrapati Shivaji International Airport
	AI404	Chhatrapati Shivaji International Airport
	6E303	Chhatrapati Shivaji International Airport
	SG202	Chhatrapati Shivaji International Airport
	AI101	Chhatrapati Shivaji International Airport
	SG505	Indira Gandhi International Airport

Result 12 x

SUB -QUERY-In Operator-

```
200
201      -- Using IN Subquery
202  •    SELECT FirstName, LastName
203      FROM Passengers
204      WHERE PassengerID IN (SELECT PassengerID FROM Reservations WHERE FlightID IN (1, 2));
205
---
```




Result Grid  Filter Rows: Export:  Wrap Cell Content: 

	FirstName	LastName
▶	John	Doe
	Robert	Johnson
	Sarah	Davis

Passengers 14 x

SUB -QUERY Exist Operator-

```
205
206  -- Using EXISTS Subquery
207 • SELECT FlightNumber, DepartureCity, ArrivalCity
208    FROM Flights
209    WHERE EXISTS (SELECT 1 FROM Reservations
210                  WHERE Reservations.FlightID = Flights.FlightID);
---
```



Result Grid  Filter Rows: Export:  Wrap Cell Content: 

	FlightNumber	DepartureCity	ArrivalCity
▶	AI101	Mumbai	Delhi
	SG202	Delhi	Bangalore
	6E303	Bangalore	Chennai
	AI404	Chennai	Kolkata
	SG505	Kolkata	Mumbai

Flights 15 x

View-Create View-

```
212  -- view create view
213  -- Create a view to show flight information and reservation counts
214 • CREATE VIEW FlightReservationCounts AS
215    SELECT
216        F.FlightID,
217        F.FlightNumber,
218        F.DepartureCity,
219        F.ArrivalCity,
220        COUNT(R.ReservationID) AS ReservationCount
221    FROM Flights F
222    LEFT JOIN Reservations R ON F.FlightID = R.FlightID
223    GROUP BY F.FlightID, F.FlightNumber, F.DepartureCity, F.ArrivalCity;
---
```





Result Grid  Filter Rows: Export:  Wrap Cell Content: 

	FlightID	FlightNumber	DepartureCity	ArrivalCity	ReservationCount
▶	1	AI101	Mumbai	Delhi	2
	2	SG202	Delhi	Bangalore	1
	3	6E303	Bangalore	Chennai	1
	4	AI404	Chennai	Kolkata	1
	5	SG505	Kolkata	Mumbai	1

FlightReservationCounts 16 x

View -Update View-

```
225 -- view update view
226 -- Update the number of reservations for a specific flight
227 • UPDATE Reservations
228 SET ReservationID = ReservationID
229 WHERE FlightID = 1;
230
231
232 • SELECT * FROM FlightReservationCounts;
233
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	FlightID	FlightNumber	DepartureCity	ArrivalCity	ReservationCount
▶	1	AI101	Mumbai	Delhi	2
	2	SG202	Delhi	Bangalore	1
	3	6E303	Bangalore	Chennai	1
	4	AI404	Chennai	Kolkata	1
	5	SG505	Kolkata	Mumbai	1

FlightReservationCounts 17 ×

DQL QUERIES-

Where clause-

```
235 -- Retrieve Passengers with a Specific Last Name
236
237 -- Retrieve Passengers with a Specific Last Name
238 • SELECT FirstName, LastName, Email
239 FROM Passengers
240 WHERE LastName = 'Smith';
241
242
```

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

	FirstName	LastName	Email
▶	Jane	Smith	jane.smith@example.com

Passengers 18 x

Output

Range Operator-

```
241
242 -- range operator
243 -- Retrieve Flights within a Date Range??
244
245 • SELECT FlightNumber, DepartureCity, ArrivalCity, DepartureTime
246 FROM Flights
247 WHERE DepartureTime BETWEEN '2023-10-15 08:00:00' AND '2023-10-16 08:00:00';
248
249
```

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

	FlightNumber	DepartureCity	ArrivalCity	DepartureTime
▶	AI101	Mumbai	Delhi	2023-10-15 08:00:00
	SG202	Delhi	Bangalore	2023-10-15 12:00:00
	6E303	Bangalore	Chennai	2023-10-15 16:00:00
	AI404	Chennai	Kolkata	2023-10-15 20:00:00
	SG505	Kolkata	Mumbai	2023-10-16 08:00:00

Flights 1 x

Read Only

List Operator-

```
249 -- list Operator
250 -- Find Flights from a List of Cities
251 • SELECT FlightNumber, DepartureCity, ArrivalCity
252 FROM Flights
253 WHERE DepartureCity IN ('Mumbai', 'Delhi', 'Bangalore');
```

Result Grid			
Filter Rows: <input type="text"/>			
Export: Wrap Cell Content:			
	FlightNumber	DepartureCity	ArrivalCity
▶	AI101	Mumbai	Delhi
	SG202	Delhi	Bangalore
	6E303	Bangalore	Chennai

Flights 2 x

Like Operator-

```
255 -- like operator
256
257 -- Find passengers with first names starting with 'A'
258 • SELECT FirstName, LastName, Email
259 FROM Passengers
260 WHERE FirstName LIKE 'A%';
```

Result Grid			
Filter Rows: <input type="text"/>			
Export: Wrap Cell Content:			
	FirstName	LastName	Email

Passengers 3 x

```
262
263 -- Find flights with numbers containing '123'
264 • SELECT FlightNumber, DepartureCity, ArrivalCity
265 FROM Flights
266 WHERE FlightNumber LIKE '%123%';
267
```

Result Grid			
Filter Rows: <input type="text"/>			
Export: Wrap Cell Content:			
	FlightNumber	DepartureCity	ArrivalCity

Flights 4 x


```

268 -- Find passengers with email addresses from a specific domain
269 • SELECT FirstName, LastName, Email
270 FROM Passengers
271 WHERE Email LIKE '%@example.com';
272
273

```

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

	FirstName	LastName	Email
▶	John	Doe	john.doe@example.com
	Jane	Smith	jane.smith@example.com
	Robert	Johnson	robert.j@example.com
	Emily	Brown	emily.b@example.com
	David	Wilson	david.w@example.com
	Sarah	Davis	sarah.d@example.com

Passengers 5 x

Order By-

```

272
273 -- Orderby
274 -- Sort Flights by Departure City in Ascending Order
275 • SELECT FlightNumber, DepartureCity, ArrivalCity
276 FROM Flights
277 ORDER BY DepartureCity;
278

```

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

	FlightNumber	DepartureCity	ArrivalCity
▶	6E303	Bangalore	Chennai
	AI404	Chennai	Kolkata
	SG202	Delhi	Bangalore
	SG505	Kolkata	Mumbai
	AI101	Mumbai	Delhi

Flights 6 x

```

278
279 -- Sort Airports by City in Ascending Order and State in Descending Order
280 • SELECT AirportCode, AirportName, City, State
281 FROM Airports
282 ORDER BY City ASC, State DESC;
283

```


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

	AirportCode	AirportName	City	State
▶	BLR	Kempegowda International Airport	Bangalore	Karnataka
	MAA	Chennai International Airport	Chennai	Tamil Nadu
	DEL	Indira Gandhi International Airport	Delhi	Delhi
	HYD	Rajiv Gandhi International Airport	Hyderabad	Telangana
	CCU	Netaji Subhas Chandra Bose International Airport	Kolkata	West Bengal
	BOM	Chhatrapati Shivaji International Airport	Mumbai	Maharashtra

Airports 7 x

Distinct-

```
284 -- Distinct
285 -- Retrieve Distinct Departure Cities
286 • SELECT DISTINCT DepartureCity
287 FROM Flights;
288
```

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

	DepartureCity
▶	Mumbai
	Delhi
	Bangalore
	Chennai
	Kolkata

Flights 8 ×

```
288
289 -- Find Distinct Aircraft Types
290 • SELECT DISTINCT AircraftType
291 FROM Aircraft;
292
```




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

	AircraftType
▶	Boeing 737
	Airbus A320
	Boeing 777
	Airbus A350
	Embraer E190

Aircraft 9 ×

Is Null-

```
293 -- is null
294 -- Identify Passengers with Missing Phone Numbers
295 • SELECT FirstName, LastName, Email, Phone
296 FROM Passengers
297 WHERE Phone IS NULL;
```

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

	FirstName	LastName	Email	Phone
--	-----------	----------	-------	-------

Passengers 10 ×

```

299 -- Retrieve Airports with Undefined State Information
300 • SELECT AirportCode, AirportName, City, State
301 FROM Airports
302 WHERE State IS NULL;
303
304

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
AirportCode	AirportName	City	State

Airports 11 x

Is Not Null-

```

303
304 -- is not null
305 -- Identify Passengers with Provided Phone Numbers
306 • SELECT FirstName, LastName, Email, Phone
307 FROM Passengers
308 WHERE Phone IS NOT NULL;

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
FirstName	LastName	Email	Phone
John	Doe	john.doe@example.com	+1-555-123-4567
Jane	Smith	jane.smith@example.com	+1-555-987-6543
Robert	Johnson	robert.j@example.com	+1-555-777-8888
Emily	Brown	emily.b@example.com	+1-555-444-5555
David	Wilson	david.w@example.com	+1-555-999-1111
Sarah	Davis	sarah.d@example.com	+1-555-333-7777

Passengers 12 x

```

309
310 -- Retrieve Airports with State Information Defined
311 • SELECT AirportCode, AirportName, City, State
312 FROM Airports
313 WHERE State IS NOT NULL;
314

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
AirportCode	AirportName	City	State
BOM	Chhatrapati Shivaji International Airport	Mumbai	Maharashtra
DEL	Indira Gandhi International Airport	Delhi	Delhi
BLR	Kempegowda International Airport	Bangalore	Karnataka
MAA	Chennai International Airport	Chennai	Tamil Nadu
CCU	Netaji Subhas Chandra Bose International Airport	Kolkata	West Bengal
HYD	Rajiv Gandhi International Airport	Hyderabad	Telangana

Airports 13 x

Built In -String Function-

```
315 -- Built In String Function
316 • select lower("INDIAN AIRLINE SYSTEM");
317
```

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

lower("INDIAN AIRLINE SYSTEM")
▶ indian airline system

Result 14 x



```
317 • select upper("indian airline system");
318
```

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

upper("indian airline system")
▶ INDIAN AIRLINE SYSTEM

Result 15 x





```
318 • select substr("indian airline system",2,6);
319
```

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

substr("indian airline system",2,6)
▶ ndian

Result 17 x

```
319 • select replace("indian airline system", "india", "America");
```

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

replace("indian airline system", "india", "America")
▶ American airline system

Result 18 x

Group By Clause-

```
321 -- group by clause
322 -- Find the Total Number of Flights Departing from Each City
323 • SELECT DepartureCity, COUNT(*) AS NumberOfFlights
324 FROM Flights
325 GROUP BY DepartureCity;
326
327
328
329
```

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

	DepartureCity	NumberOfFlights
▶	Mumbai	1
	Delhi	1
	Bangalore	1
	Chennai	1
	Kolkata	1

Result 20 x

Aggregation Function-

Count –

```
327
328 -- Aggregation Function
329 -- count
330 -- Count the number of flights departing from each city.
331 • SELECT DepartureCity, COUNT(*) AS NumberOfFlights
332 FROM Flights
333 GROUP BY DepartureCity;
334
```

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

	DepartureCity	NumberOfFlights
▶	Mumbai	1
	Delhi	1
	Bangalore	1
	Chennai	1
	Kolkata	1

Result 21 x

SUM-

```
335 -- SUM
336
337 -- Calculate the Total Seating Capacity for All Aircraft Types
338 • SELECT SUM(SeatingCapacity) AS TotalSeatingCapacity
339 FROM Aircraft;
```

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

TotalSeatingCapacity
980

Result 22 x

AVG-

```
341 -- AVG
342 -- Calculate the Average Seating Capacity for All Aircraft Types
343 • SELECT AVG(SeatingCapacity) AS AverageSeatingCapacity
344 FROM Aircraft;
```

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

AverageSeatingCapacity
196.0000

Result 24 x

MAX-

```
346 -- max
347 -- Find the Highest Passenger ID
348 • SELECT MAX(PassengerID) AS MaxPassengerID
349 FROM Passengers;
```

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

MaxPassengerID
6

Result 25 x

MIN-

```
350
351 -- MIN
352 -- Find the Lowest Passenger ID
353 • SELECT MIN(PassengerID) AS MinPassengerID
354 FROM Passengers;
355
```





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

	MinPassengerID
▶	1

Result 27 x

CONCAT-

```
356 -- concat
357 -- List all flight numbers departing from a specific city.
358 • SELECT DepartureCity, GROUP_CONCAT(FlightNumber) AS FlightNumbers
359 FROM Flights
360 WHERE DepartureCity = 'Delhi'
361 GROUP BY DepartureCity;
362
```

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

	DepartureCity	FlightNumbers
▶	Delhi	SG202

Result 28 x