aircargo* SQL File 7*

Limit to 1000 rows

```sql
use aircargo;

describe aircargo.routes;

CREATE TABLE IF NOT EXISTS aircargo.routes_details (
route_id INT NOT NULL UNIQUE,
flight_num INT NOT NULL,
origin_airport VARCHAR (100) NOT NULL,
destination_airport VARCHAR (100) NOT NULL,
aircraft_id VARCHAR (100) NOT NULL,
distance_miles INT NOT NULL CHECK (distance_miles > 0),
CONSTRAINT flight_num CHECK (flight_num > 0)
);
```

aircargo* | SQL File 7*

Limit to 1000 rows

```sql
/*Write a query to display all the passengers (customers) who have travelled in routes 01 to 25. Take data  from the passengers_on_flights table.*/

SELECT*FROM aircargo.passengers_on_flights
WHERE route_id BETWEEN '01' AND '25';
```

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

customer_id	aircraft_id	route_id	depart	arrival	seat_num	class_id	travel_date	flight_num
2	767-301ER	4	JFK	LAX	01E	Economy	02-09-2018	1114
1	ERJ142	9	DEN	LAX	01EP	Economy Plus	26-12-2019	1119
5	767-301ER	12	ABI	ADK	02B	Bussiness	02-07-2018	1122
5	ERJ142	18	ANI	BGR	02E	Economy	06-05-2020	1128
4	767-301ER	5	LAX	JFX	02FC	First Class	06-04-2020	1115
7	767-301ER	20	AVL	BOI	03B	Bussiness	08-07-2020	1130
5	ERJ142	22	BGR	BJI	03E	Economy	31-05-2020	1132
4	767-301ER	4	JFK	LAX	03FC	First Class	30-04-2020	1114
11	767-301ER	5	LAX	JFX	04B	Bussiness	12-11-2020	1115
17	A321	13	ABI	ADK	04EP	Economy Plus	03-06-2019	1123
9	767-301ER	15	CAK	ANI	04FC	First Class	10-09-2020	1125
11	767-301ER	4	JFK	LAX	05B	Bussiness	09-11-2020	1114
10	A321	10	HNL	DEN	05E	Economy	11-10-2020	1120
15	A321	14	BON	CAK	06B	Bussiness	02-11-2018	1124

passengers_on_flights 2

Read Only

aircargo* SQL File 7*

Limit to 1000 rows

```sql
/*Write a query to identify the number of passengers and total revenue in business class from the ticket_details table*/

SELECT COUNT(no_of_tickets) AS total_passengers, SUM((no_of_tickets)*(Price_per_ticket)) AS business_class_revenue
FROM aircargo.ticket_details
WHERE class_id = 'Bussiness';
```

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

total_passengers	business_class_revenue
13	6034

Result 3

Read Only

aircargo* SQL File 7*

Limit to 1000 rows

```sql
/*Write a query to display the full name of the customer by extracting the first name and last name from the customer table.*/

SELECT CONCAT(first_name, ' ', last_name) AS full_name
FROM aircargo.customer;
```

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

full_name
Julie Sam
Steve Ryan
Morris Lois
Cathenna Emily
Aaron Kim
Alexander Scot
Anderson Stewart
Floyd Ted
Leo Travis
Melvin Tracy
Roger Walson
Shirley Wally
Solomon Walter
Carol Vernon

Result 4

Read Only

aircargo* SQL File 7*

Limit to 1000 rows

```sql
/*Write a query to extract the customers who have registered and booked a ticket. Use data from the customer and ticket_details tables.*/

SELECT DISTINCT(customer_id), first_name, last_name
FROM aircargo.customer
INNER JOIN ticket_details USING (customer_id);
```

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

customer_id	first_name	last_name
27	Cherly	Vernon
22	Pheny	Eri
21	Chirsty	Josh
4	Cathenna	Emily
5	Aaron	Kim
7	Anderson	Stewart
8	Floyd	Ted
9	Leo	Travis
10	Melvin	Tracy
11	Roger	Walson
19	Joyce	Paul
13	Solomon	Walter
14	Carol	Vernon
25	Moss	Morris

Result 5

Read Only

aircargo* SQL File 7*

Limit to 1000 rows

```sql
/*Write a query to identify the customer's first name and last name based on their customer ID and brand (Emirates) from the ticket_details table.*/

SELECT first_name, last_name
FROM aircargo.customer
INNER JOIN ticket_details USING (customer_id)
WHERE brand = 'Emirates';
```

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

first_name	last_name
Steve	Ryan
Cathenna	Emily
Cathenna	Emily
Aaron	Kim
Anderson	Stewart
Leo	Travis
Roger	Walson
Roger	Walson
Carol	Vernon
Gloria	Richie
Gloria	Richie
Joyce	Paul
Moss	Morris
Moss	Morris

Result 6

Read Only

aircargo* SQL File 7*

Limit to 1000 rows

```sql
/*Write a query to identify the customers who have travelled by Economy Plus class using Group By and Having clause on the passengers_on_flights table*/

SELECT customer_id, route_id, class_id
FROM aircargo.passengers_on_flights
GROUP BY customer_id, route_id, class_id
HAVING class_id = 'Economy Plus';
```

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

customer_id	route_id	class_id
1	9	Economy Plus
8	38	Economy Plus
11	31	Economy Plus
17	13	Economy Plus
19	47	Economy Plus
19	30	Economy Plus
22	22	Economy Plus
32	31	Economy Plus
47	33	Economy Plus
50	21	Economy Plus

passengers_on_flights 7

Read Only

aircargo* SQL File 7*

Limit to 1000 rows

```sql
/*Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket_details table.*/

select sum(Price_per_ticket) as Total_Revenue,
if (sum(Price_per_ticket) >10000, "YES", "NO") as Reveune_Crossed from ticket_details;
```

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

Total_Revenue	Reveune_Crossed
15369	YES

Result 8

Read Only

aircargo* SQL File 7*

Limit to 1000 rows

```sql
/*Write a query to create and grant access to a new user to perform operations on a database.*/

create user 'username'@'localhost' identified by 'password';
grant permission_type on aircargo  to 'username'@'localhost';
```

aircargo* SQL File 7*

```sql
/*Write a query to find the maximum ticket price for each class using window functions on the ticket_details table.*/

SELECT p_date, customer_id, aircraft_id, class_id, no_of_tickets, a_code, brand, MAX(Price_per_ticket) OVER (PARTITION BY brand)
AS Max_Ticket_Class_Price
FROM aircargo.ticket_details;
```

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

p_date	customer_id	aircraft_id	class_id	no_of_tickets	a_code	brand	Max_Ticket_Class_Price
03-03-2020	21	CRJ900	Bussiness	1	BOH	Bristish Airways	490
12-12-2020	19	CRJ900	Economy Plus	1	DEN	Bristish Airways	490
04-04-2019	16	CRJ900	First Class	1	YVR	Bristish Airways	490
09-08-2019	20	CRJ900	First Class	1	MCO	Bristish Airways	490
01-06-2018	20	CRJ900	First Class	1	PEK	Bristish Airways	490
01-10-2018	1	CRJ900	First Class	1	DEN	Bristish Airways	490
12-03-2020	33	CRJ900	Bussiness	1	BOH	Bristish Airways	490
09-12-2020	47	CRJ900	Economy Plus	1	DEN	Bristish Airways	490
13-12-2020	19	CRJ900	Economy Plus	1	DEN	Bristish Airways	490
01-01-2018	9	CRJ900	First Class	1	AGB	British Airways	390
26-12-2018	27	767-301ER	Economy	1	DAL	Emirates	499
04-04-2020	4	767-301ER	First Class	1	AGB	Emirates	499
07-07-2020	7	767-301ER	Bussiness	1	BFS	Emirates	499

Result 9

Read Only

aircargo* SQL File 7*

Limit to 1000 rows

```sql
/*Write a query to extract the passengers whose route ID is 4 by improving the speed and performance of the passengers_on_flights table.*/

SELECT*FROM aircargo.passengers_on_flights
WHERE route_id = '4';

CREATE INDEX route_id_4 ON passengers_on_flights(route_id);
SELECT*FROM aircargo.passengers_on_flights
WHERE route_id = '4';
```

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

customer_id	aircraft_id	route_id	depart	arrival	seat_num	class_id	travel_date	flight_num
2	767-301ER	4	JFK	LAX	01E	Economy	02-09-2018	1114
4	767-301ER	4	JFK	LAX	03FC	First Class	30-04-2020	1114
11	767-301ER	4	JFK	LAX	05B	Bussiness	09-11-2020	1114

aircargo* SQL File 7*

Limit to 1000 rows

```sql
/* For the route ID 4, write a query to view the execution plan of the passengers_on_flights table.*/

SELECT*FROM aircargo.passengers_on_flights
WHERE route_id = '4';
```

Result Grid Filter Rows: Export: Wrap Cell Content:

customer_id	aircraft_id	route_id	depart	arrival	seat_num	class_id	travel_date	flight_num
2	767-301ER	4	JFK	LAX	01E	Economy	02-09-2018	1114
4	767-301ER	4	JFK	LAX	03FC	First Class	30-04-2020	1114
11	767-301ER	4	JFK	LAX	05B	Bussiness	09-11-2020	1114

passengers_on_flights 11

Read Only

aircargo* | SQL File 7*

Limit to 1000 rows

```sql
/*Write a query to calculate the total price of all tickets booked by a customer across different aircraft IDs using rollup function.*/

SELECT customer_id, SUM(price_per_ticket) as total_price, count(aircraft_id)
FROM aircargo.ticket_details
GROUP BY customer_id WITH ROLLUP;
```

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

customer_id	total_price	count(aircraft_id)
1	570	2
2	635	2
4	780	2
5	670	3
7	430	1
8	465	2
9	770	2
10	135	1
11	1225	3
13	395	1
14	290	2
15	430	1
16	395	1

Result 12 | Read Only

aircargo* SQL File 7*

Limit to 1000 rows

```sql
/*Write a query to create a view with only business class customers along with the brand of airlines.*/
CREATE VIEW bussiness_class_view AS
SELECT customer_id, class_id, brand
FROM aircargo.ticket_details
WHERE class_id = 'Bussiness'
ORDER BY brand;

SELECT*FROM bussiness_class_view;
```

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

customer_id	class_id	brand
21	Bussiness	Bristish Airways
33	Bussiness	Bristish Airways
7	Bussiness	Emirates
11	Bussiness	Emirates
25	Bussiness	Emirates
5	Bussiness	Emirates
49	Bussiness	Emirates
11	Bussiness	Emirates
29	Bussiness	Jet Airways
24	Bussiness	Qatar Airways

bussiness_class_view 13

Read Only

aircargo* | SQL File 7*

Limit to 1000 rows

Find | Done

```
1   /*Write a query to create a stored procedure to get the details of all passengers flying between a range of routes defined in run time. Also, return an
2
3   delimiter //
4   create procedure route_range(in rangefrom int, in rangeto int)
5   begin
6   select pass.customer_id, concat(cust.first_name,' ',cust.last_name) as cust_name, pass.class_id,
7   pass.route_id, pass.aircraft_id
8   from customer cust
9   join passengers_on_flights pass
10  where pass.route_id between rangefrom and rangeto;
11  end //
12  delimiter ;
13  call route_range(1,100);
```

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

customer_id	cust_name	class_id	route_id	aircraft_id
2	Rose Arthur	Bussiness	34	A321
2	Russell Peter	Bussiness	34	A321
2	Wayne Noah	Bussiness	34	A321
2	Sophia Carl	Bussiness	34	A321
2	Louis Douglas	Bussiness	34	A321
2	Doris Walter	Bussiness	34	A321
2	Bily Brian	Bussiness	34	A321
2	Joe Daniel	Bussiness	34	A321
2	Roger Mattew	Bussiness	34	A321
2	Kyle Mark	Bussiness	34	A321

Result 19 | Read Only

aircargo* | SQL File 7*

Limit to 1000 rows

```sql
/*Write a query to create a stored procedure that extracts all the details from the routes table where the travelled distance is more than 2000 miles.*

DELIMITER //
CREATE PROCEDURE travelled_more_than_2000_miles()
BEGIN
Select*FROM aircargo.routes
WHERE distance_miles > 2000;
END //

CALL travelled_more_than_2000_miles();
```

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

route_id	flight_num	origin_airport	destination_airport	aircraft_id	distance_miles
1	1111	EWR	HNL	767-301ER	4962
2	1112	HNL	EWR	767-301ER	4962
3	1113	EWR	LHR	A321	3466
4	1114	JFK	LAX	767-301ER	2475
5	1115	LAX	JFK	767-301ER	2475
6	1116	HNL	LAX	767-301ER	2556
10	1120	HNL	DEN	A321	3365
12	1122	ABI	ADK	767-301ER	4300
13	1123	ADK	BQN	A321	2232
14	1124	BQN	CAK	A321	2445

Result 15 | Read Only

aircargo* | SQL File 7*

Limit to 1000 rows

Find | Done

```sql
/*Write a query to create a stored procedure that groups the distance travelled by each flight into three categories.
The categories are, short distance travel (SDT) for >=0 AND <= 2000 miles, intermediate distance travel (IDT) for >2000 AND <=6500, and long-distance travel (LDT) for >6500.*/DELIMI
CREATE FUNCTION travelled_distance(distance_miles INT)
RETURNS VARCHAR(100) deterministic
BEGIN
DECLARE travelled_distance VARCHAR (100);
IF distance_miles BETWEEN 0 AND 2000 THEN SET travelled_distance = 'Short Distance Travel (SDT)';
ELSEIF distance_miles BETWEEN 2000 AND 6500 THEN SET travelled_distance = 'Intermediate Distance Travel (IDT)';
ELSEIF distance_miles > 6500 THEN SET travelled_distance = 'Long-Distance Travel (LDT)';
END IF;
RETURN (travelled_distance);
END //
DELIMITER //
SELECT route_id, flight_num, origin_airport, destination_airport, aircraft_id, travelled_distance (distance_miles) AS distance_miles
FROM aircargo.routes
ORDER BY distance_miles;
```

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

route_id	flight_num	origin_airport	destination_airport	aircraft_id	distance_miles
1	1111	EWR	HNL	767-301ER	Intermediate Distance Travel (IDT)
2	1112	HNL	EWR	767-301ER	Intermediate Distance Travel (IDT)
3	1113	EWR	LHR	A321	Intermediate Distance Travel (IDT)
4	1114	JFK	LAX	767-301ER	Intermediate Distance Travel (IDT)

Result Grid

Result 16

Read Only

aircargo* | SQL File 7*

Limit to 1000 rows

Find | Done

```
using a stored function in stored procedure on the ticket_details table.
Condition: If the class is Business and Economy Plus, then complimentary services are given as Yes, else it is No.*/
DELIMITER //
CREATE FUNCTION complimentary_services (class_id VARCHAR (100))
RETURNS VARCHAR(100) deterministic
BEGIN
DECLARE complimentary_services VARCHAR(100);
IF class_id = 'Business' THEN SET complimentary_services = 'Yes';
ELSEIF class_id = 'Economy Plus' THEN SET complimentary_services = 'Yes';
ELSEIF class_id = 'Economy' THEN SET complimentary_services = 'No';
ELSEIF class_id = 'First Class' THEN SET complimentary_services = 'No';
END IF;
RETURN (complimentary_services);
END //
DELIMITER //
SELECT p_date, customer_id, class_id, complimentary_services(class_id) AS complimentary_services
FROM aircargo.ticket_details;
```

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

p_date	customer_id	class_id	complimentary_services
26-12-2018	27	Economy	No
02-02-2020	22	Economy Plus	Yes
03-03-2020	21	Bussiness	NULL
04-04-2020	4	First Class	No
05-05-2020	5	Economy	No

Result 17

Result Grid

Read Only

aircargo* SQL File 7*

Limit to 1000 rows

Find Done

```sql
/*Write a query to extract the first record of the customer whose last name ends with Scott using a cursor from the customer table.*/
DELIMITER //
CREATE PROCEDURE my_cursor ()
BEGIN
DECLARE a VARCHAR (100);
DECLARE b VARCHAR (100);
DECLARE my_cursor CURSOR FOR SELECT last_name, first_name FROM aircargo.customer
WHERE last_name = 'Scott';
OPEN my_cursor;
REPEAT FETCH my_cursor INTO a,b;
UNTIL b = 0 END REPEAT;
SELECT a AS last_name, b AS first_name;
CLOSE my_cursor; END;
// DELIMITER ;
CALL my_cursor();
```

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

last_name	first_name
Scott	Samuel

Result 18

Read Only