Air Cargo Analysis

PROBLEM DESCRIPTION

Air Cargo is an aviation company that provides air transportation services for passengers and freight. Air Cargo uses its aircraft to provide different services with the help of partnerships or alliances with other airlines. The company wants to prepare reports on regular passengers, busiest routes, ticket sales details, and other scenarios to improve the ease of travel and booking for customers.

TASKS TO BE PERFORMED:

1. Create an ER diagram for the given airlines database.
2. Write a query to create route_details table using suitable data types for the fields, such as route_id, flight_num, origin_airport, destination_airport, aircraft_id, and distance_miles. Implement the check constraint for the flight number and unique constraint for the route_id fields. Also, make sure that the distance miles field is greater than 0.
3. 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.
4. Write a query to identify the number of passengers and total revenue in business class from the ticket_details table.
5. Write a query to display the full name of the customer by extracting the first name and last name from the customer table.
6. Write a query to extract the customers who have registered and booked a ticket. Use data from the customer and ticket_details tables.
7. 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.
8. 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.
9. Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket_details table.
10. Write a query to create and grant access to a new user to perform operations on a database.
11. Write a query to find the maximum ticket price for each class using window functions on the ticket_details table.
12. Write a query to extract the passengers whose route ID is 4 by improving the speed and performance of the passengers_on_flights table.
13. For the route ID 4, write a query to view the execution plan of the passengers_on_flights table.

14. Write a query to calculate the total price of all tickets booked by a customer across different aircraft IDs using rollup function.
15. Write a query to create a view with only business class customers along with the brand of airlines.
16. 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 error message if the table doesn't exist.
17. 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.
18. 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.
19. Write a query to extract ticket purchase date, customer ID, class ID and specify if the complimentary services are provided for the specific class using a stored function in stored procedure on the ticket_details table.

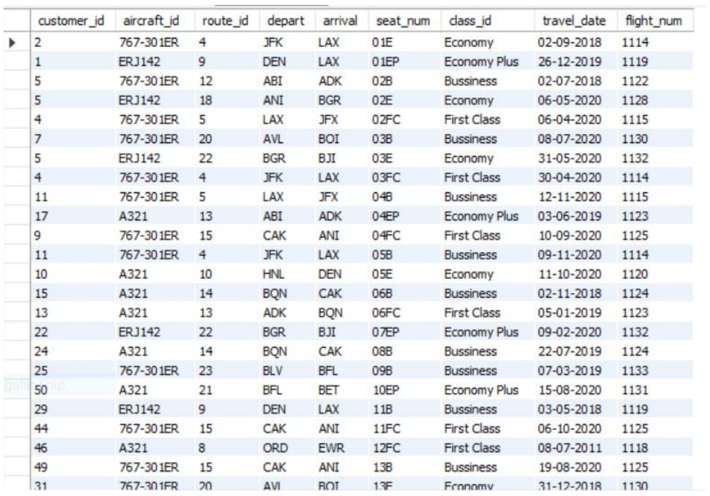
Condition:

- If the class is *Business* and *Economy Plus,* then complimentary services are given as *Yes,* else it is *No*

20. Write a query to extract the first record of the customer whose last name ends with Scott using a cursor from the customer table.

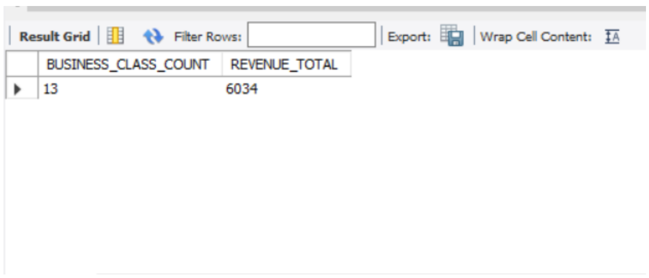
SOLUTION

```
CREATE DATABASE Air_Cargo_Analysis;

USE Air_Cargo_Analysis;

SET sql_mode=(SELECT REPLACE(@@sql_mode,'ONLY_FULL_GROUP_BY',''));


CREATE TABLE route_details (route_id int NOT NULL,

flight_num int NOT NULL,

origin_airport varchar(20),

destination_airport varchar(20),

aircraft_id varchar(10),

distance int NOT NULL,

UNIQUE(route_id), CHECK (distance>0));


SELECT*FROM passengers_on_flights

WHERE route_id BETWEEN 1 AND 25;
```



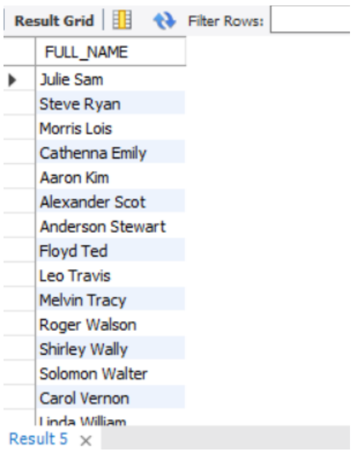
customer_id	aircraft_id	route_id	depart	arrival	seat_num	class_id	travel_date	flight_num
2	767-301ER	4	JFK	LAX	01E	Economy	02-09-2018	1114
1	ERJ142	9	DEN	LAX	01EP	Economy Plus	26-12-2019	1119
5	767-301ER	12	ABI	ADK	02B	Bussiness	02-07-2018	1122
5	ERJ142	18	ANI	BGR	02E	Economy	06-05-2020	1128
4	767-301ER	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	BQN	CAK	06B	Bussiness	02-11-2018	1124
13	A321	13	ADK	BQN	06FC	First Class	05-01-2019	1123
22	ERJ142	22	BGR	BJI	07EP	Economy Plus	09-02-2020	1132
24	A321	14	BQN	CAK	08B	Bussiness	22-07-2019	1124
25	767-301ER	23	BLV	BFL	09B	Bussiness	07-03-2019	1133
50	A321	21	BFL	BET	10EP	Economy Plus	15-08-2020	1131
29	ERJ142	9	DEN	LAX	11B	Bussiness	03-05-2018	1119
44	767-301ER	15	CAK	ANI	11FC	First Class	06-10-2020	1125
46	A321	8	ORD	EWR	12FC	First Class	08-07-2011	1118
49	767-301ER	15	CAK	ANI	13B	Bussiness	19-08-2020	1125
31	767-301ER	20	AVL	BOI	13E	Economy	31-12-2018	1130

```
SELECT COUNT(class_id = 'Business') AS BUSINESS_CLASS_COUNT,

SUM(no_of_tickets*price_per_ticket) AS REVENUE_TOTAL  FROM ticket_details

WHERE class_id = 'Bussiness';
```



BUSINESS_CLASS_COUNT	REVENUE_TOTAL
13	6034

SELECT CONCAT(first_name, " ", last_name) AS FULL_NAME FROM customer;



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
Linda William

SELECT customer_id, CONCAT(first_name, " " , last_name) AS NAME,

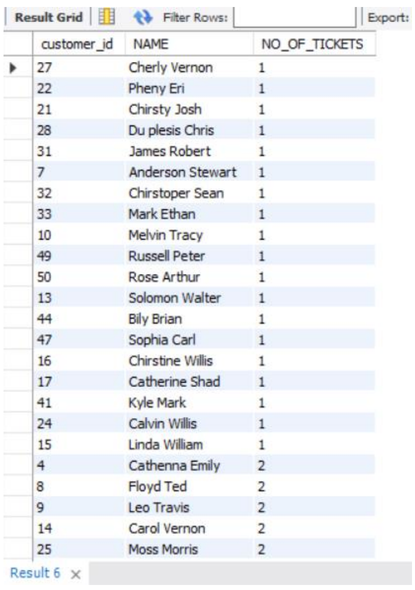
COUNT(no_of_tickets) AS NO_OF_TICKETS

FROM customer

JOIN ticket_details USING (customer_id)

GROUP BY customer_id, NAME

ORDER BY NO_OF_TICKETS;

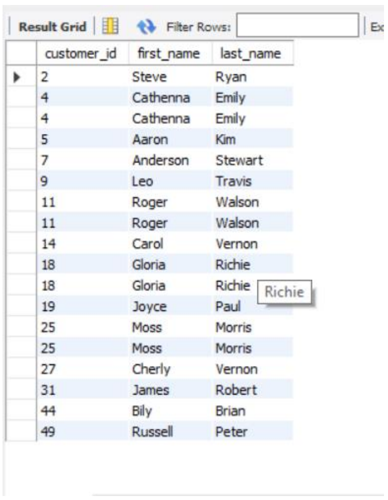


customer_id	NAME	NO_OF_TICKETS
27	Cherly Vernon	1
22	Pheny Eri	1
21	Chirsty Josh	1
28	Du plesis Chris	1
31	James Robert	1
7	Anderson Stewart	1
32	Chirstoper Sean	1
33	Mark Ethan	1
10	Melvin Tracy	1
49	Russell Peter	1
50	Rose Arthur	1
13	Solomon Walter	1
44	Bily Brian	1
47	Sophia Carl	1
16	Chirstine Willis	1
17	Catherine Shad	1
41	Kyle Mark	1
24	Calvin Willis	1
15	Linda William	1
4	Cathenna Emily	2
8	Floyd Ted	2
9	Leo Travis	2
14	Carol Vernon	2
25	Moss Morris	2

SELECT customer_id, first_name, last_name FROM customer

JOIN ticket_details USING(customer_id)

WHERE brand = 'Emirates';


Result Grid | Filter Rows: | Ex

customer_id	first_name	last_name
2	Steve	Ryan
4	Cathenna	Emily
4	Cathenna	Emily
5	Aaron	Kim
7	Anderson	Stewart
9	Leo	Travis
11	Roger	Walson
11	Roger	Walson
14	Carol	Vernon
18	Gloria	Richie
18	Gloria	Richie
19	Joyce	Paul
25	Moss	Morris
25	Moss	Morris
27	Cherly	Vernon
31	James	Robert
44	Bily	Brian
49	Russell	Peter

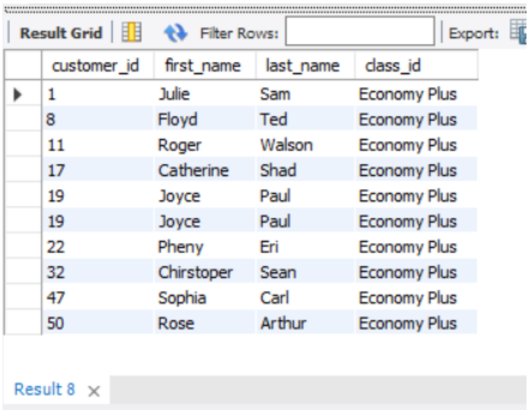
Richie

SELECT customer.customer_id, customer.first_name, customer.last_name, passengers_on_flights.class_id

FROM customer

JOIN passengers_on_flights ON customer.customer_id = passengers_on_flights.customer_id

WHERE passengers_on_flights.class_id = 'Economy Plus';


Result Grid | Filter Rows: | Export:

customer_id	first_name	last_name	class_id
1	Julie	Sam	Economy Plus
8	Floyd	Ted	Economy Plus
11	Roger	Walson	Economy Plus
17	Catherine	Shad	Economy Plus
19	Joyce	Paul	Economy Plus
19	Joyce	Paul	Economy Plus
22	Pheny	Eri	Economy Plus
32	Chirstoper	Sean	Economy Plus
47	Sophia	Carl	Economy Plus
50	Rose	Arthur	Economy Plus

Result 8 ×

SELECT customer.customer_id, customer.first_name, customer.last_name, passengers_on_flights.class_id

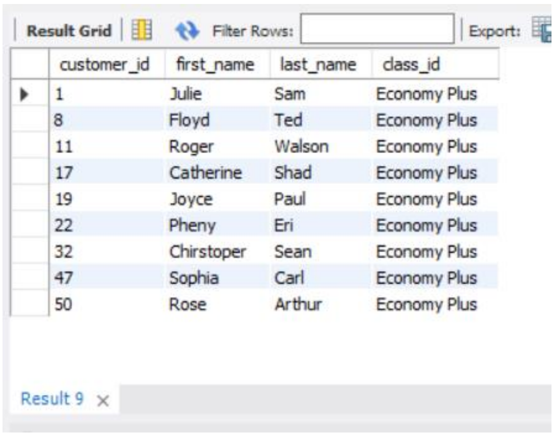
FROM customer

JOIN passengers_on_flights ON customer.customer_id = passengers_on_flights.customer_id

GROUP BY customer_id

HAVING class_id = 'Economy Plus'

ORDER BY customer_id;



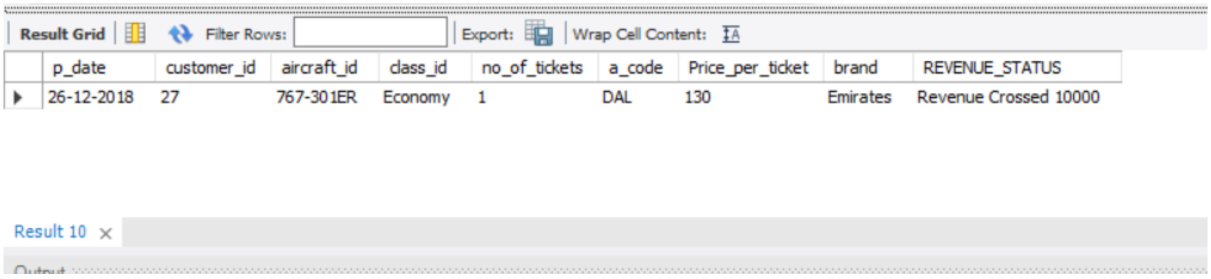
Result Grid | Filter Rows: | Export:

customer_id	first_name	last_name	class_id
1	Julie	Sam	Economy Plus
8	Floyd	Ted	Economy Plus
11	Roger	Walson	Economy Plus
17	Catherine	Shad	Economy Plus
19	Joyce	Paul	Economy Plus
22	Pheny	Eri	Economy Plus
32	Chirstoper	Sean	Economy Plus
47	Sophia	Carl	Economy Plus
50	Rose	Arthur	Economy Plus

Result 9

SELECT * , IF(SUM(no_of_tickets * Price_per_ticket)>10000,'Revenue Crossed 10000','Revenue Less Than 10000')

AS REVENUE_STATUS FROM ticket_details;



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

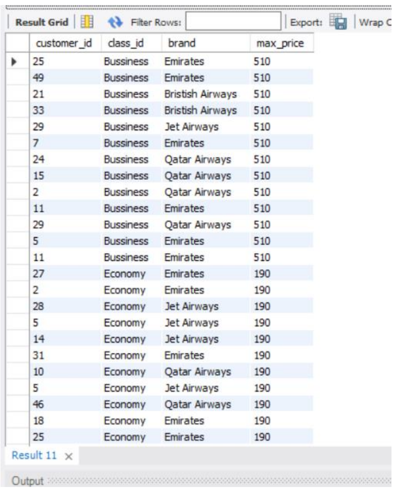
p_date	customer_id	aircraft_id	class_id	no_of_tickets	a_code	Price_per_ticket	brand	REVENUE_STATUS
26-12-2018	27	767-301ER	Economy	1	DAL	130	Emirates	Revenue Crossed 10000

Result 10

CREATE USER 'new_user'@'localhost' IDENTIFIED BY 'new_password';

GRANT ALL ON Air_Cargo_Analysis.* TO 'new_user'@'localhost';

SELECT customer_id,class_id,brand , MAX(Price_per_ticket) OVER (PARTITION BY class_id) AS max_price FROM ticket_details;



Result Grid | Filter Rows: | Export: | Wrap C

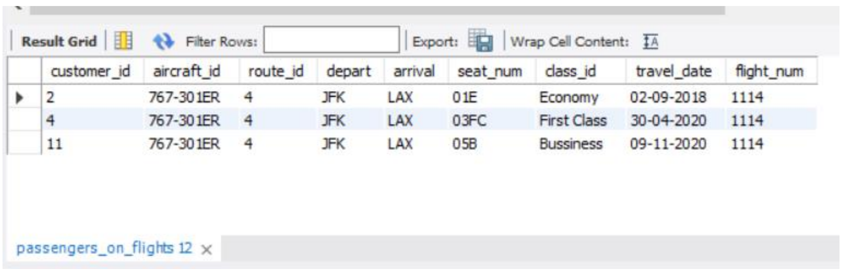
customer_id	class_id	brand	max_price
25	Bussiness	Emirates	510
49	Bussiness	Emirates	510
21	Bussiness	Bristish Airways	510
33	Bussiness	Bristish Airways	510
29	Bussiness	Jet Airways	510
7	Bussiness	Emirates	510
24	Bussiness	Qatar Airways	510
15	Bussiness	Qatar Airways	510
2	Bussiness	Qatar Airways	510
11	Bussiness	Emirates	510
29	Bussiness	Qatar Airways	510
5	Bussiness	Emirates	510
11	Bussiness	Emirates	510
27	Economy	Emirates	190
2	Economy	Emirates	190
28	Economy	Jet Airways	190
5	Economy	Jet Airways	190
14	Economy	Jet Airways	190
31	Economy	Emirates	190
10	Economy	Qatar Airways	190
5	Economy	Jet Airways	190
46	Economy	Qatar Airways	190
18	Economy	Emirates	190
25	Economy	Emirates	190

Result 11

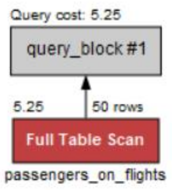
Output

SELECT * FROM passengers_on_flights WHERE route_id = 4 ;

SELECT * FROM passengers_on_flights HAVING route_id = 4 ;



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

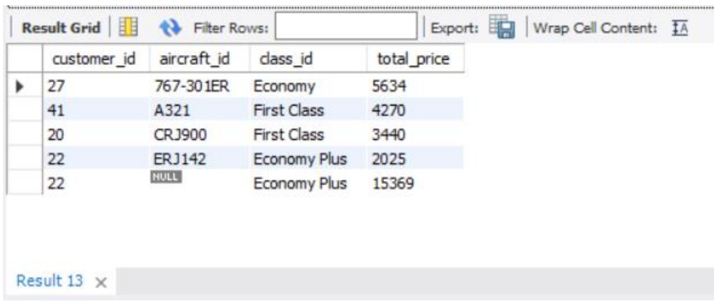


SELECT customer_id, aircraft_id , class_id , sum(no_of_tickets * Price_per_ticket)

AS total_price

FROM ticket_details

GROUP BY aircraft_id WITH ROLLUP ;



customer_id	aircraft_id	class_id	total_price
27	767-301ER	Economy	5634
41	A321	First Class	4270
20	CRJ900	First Class	3440
22	ERJ142	Economy Plus	2025
22	NULL	Economy Plus	15369

DROP VIEW business_class ;

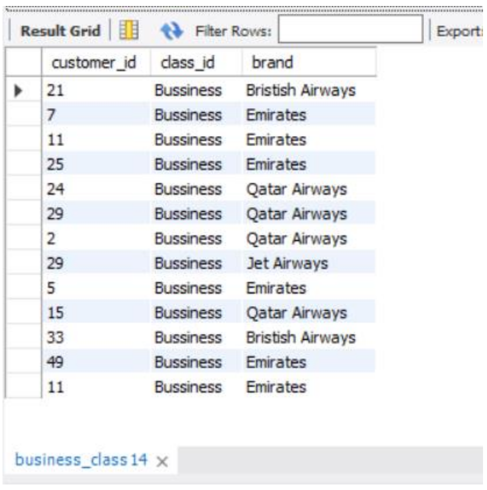
CREATE VIEW business_class AS

SELECT customer_id, class_id, brand

FROM ticket_details

WHERE class_id = 'bussiness' ;

SELECT * FROM business_class ;


Result Grid | Filter Rows: | Export:

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

business_class 14 ×

DROP PROCEDURE passenger_details;

DELIMITER &&

CREATE PROCEDURE passenger_details(route_id INT)

BEGIN

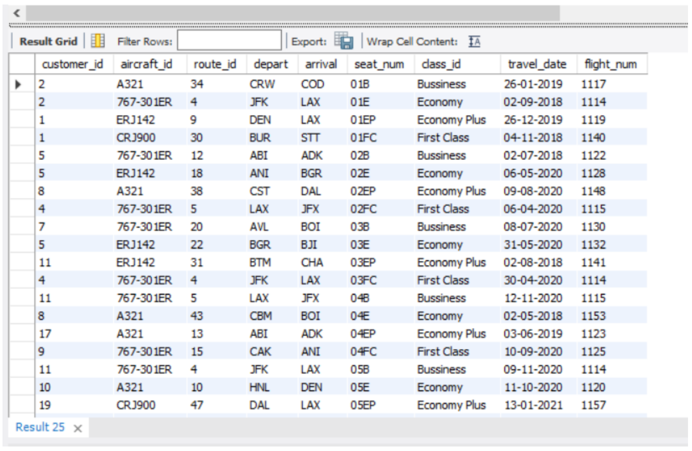
SELECT * FROM passengers_on_flights

 WHERE route_id BETWEEN

 1 AND 50 ORDER BY route_id ;

 END &&

 CALL passenger_details() ;


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	A321	34	CRW	COD	01B	Bussiness	26-01-2019	1117
	2	767-301ER	4	JFK	LAX	01E	Economy	02-09-2018	1114
	1	ERJ142	9	DEN	LAX	01EP	Economy Plus	26-12-2019	1119
	1	CRJ900	30	BUR	STT	01FC	First Class	04-11-2018	1140
	5	767-301ER	12	ABI	ADK	02B	Bussiness	02-07-2018	1122
	5	ERJ142	18	ANI	BGR	02E	Economy	06-05-2020	1128
	8	A321	38	CST	DAL	02EP	Economy Plus	09-08-2020	1148
	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
	11	ERJ142	31	BTM	CHA	03EP	Economy Plus	02-08-2018	1141
	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
	8	A321	43	CBM	BOI	04E	Economy	02-05-2018	1153
	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
	19	CRJ900	47	DAL	LAX	05EP	Economy Plus	13-01-2021	1157

Result 25 ×

DROP PROCEDURE travelled_distance() IF EXIST ;

DELIMITER //

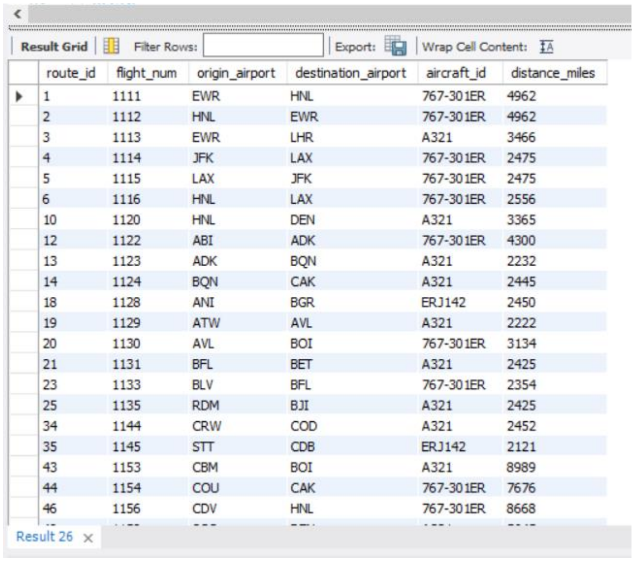
CREATE PROCEDURE travelled_distance()

BEGIN

SELECT * FROM routes WHERE distance_miles > 2000 ;

END //

CALL travelled_distance() ;



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
18	1128	ANI	BGR	ERJ142	2450
19	1129	ATW	AVL	A321	2222
20	1130	AVL	BOI	767-301ER	3134
21	1131	BFL	BET	A321	2425
23	1133	BLV	BFL	767-301ER	2354
25	1135	RDM	BJI	A321	2425
34	1144	CRW	COD	A321	2452
35	1145	STT	CDB	ERJ142	2121
43	1153	CBM	BOI	A321	8989
44	1154	COU	CAK	767-301ER	7676
46	1156	CDV	HNL	767-301ER	8668

DROP PROCEDURE travel_category;

DELIMITER //

CREATE PROCEDURE travel_category(IN distance int , OUT category VARCHAR(40))

BEGIN

SELECT distance_miles INTO distance

 FROM routes

 WHERE routes.distance_miles = distance;

IF distance >= 0 AND distance <=2000 THEN

            SET category = 'SHORT DISTANCE TRAVEL';

ELSEIF distance >= 2000 AND distance <=6500 THEN

            SET category = 'INTERMEDIATE DISTANCE TRAVEL';

ELSEIF distance > 6000 THEN

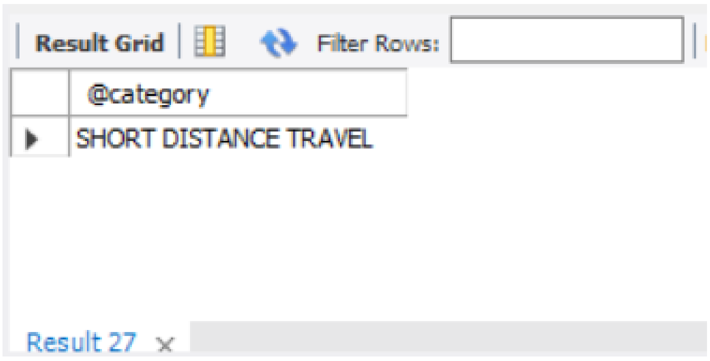
            SET category = 'LONG DISTANCE TRAVEL';

END IF ;

END //

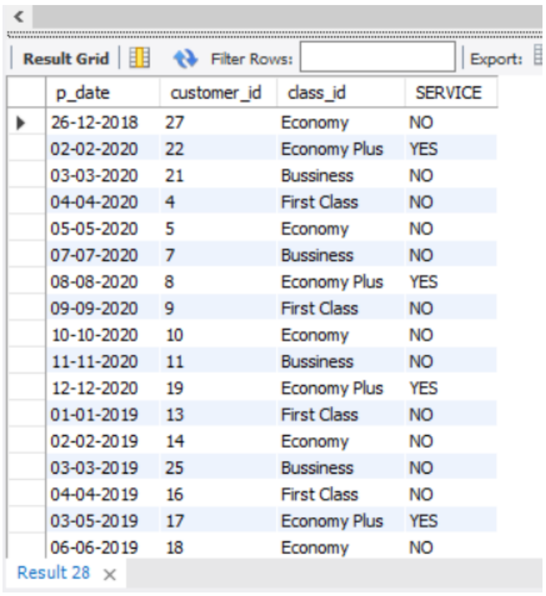
CALL travel_category(1523 , @category) ;

SELECT @category;



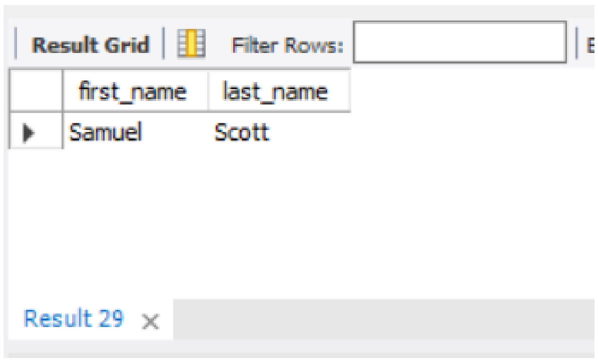
@category
SHORT DISTANCE TRAVEL

```
DROP FUNCTION IF EXISTS Complementary_Services

DELIMITER //

CREATE FUNCTION Complementary_Services(class_id VARCHAR(40))

RETURNS VARCHAR(10) DETERMINISTIC

BEGIN

DECLARE SERVICE VARCHAR(20);

IF class_id = 'Economy Plus' OR 'Business' THEN

SET SERVICE  = 'YES';

ELSE SET SERVICE = 'NO';

END IF;

RETURN SERVICE;

END //

SELECT p_date,customer_id,class_id, Complementary_Services(class_id) AS SERVICE FROM
ticket_details;
```



p_date	customer_id	class_id	SERVICE
26-12-2018	27	Economy	NO
02-02-2020	22	Economy Plus	YES
03-03-2020	21	Bussiness	NO
04-04-2020	4	First Class	NO
05-05-2020	5	Economy	NO
07-07-2020	7	Bussiness	NO
08-08-2020	8	Economy Plus	YES
09-09-2020	9	First Class	NO
10-10-2020	10	Economy	NO
11-11-2020	11	Bussiness	NO
12-12-2020	19	Economy Plus	YES
01-01-2019	13	First Class	NO
02-02-2019	14	Economy	NO
03-03-2019	25	Bussiness	NO
04-04-2019	16	First Class	NO
03-05-2019	17	Economy Plus	YES
06-06-2019	18	Economy	NO

```
DROP PROCEDURE IF EXISTS First_Scott

DELIMITER //

CREATE PROCEDURE First_Scott()

BEGIN

DECLARE a VARCHAR(50);

DECLARE b VARCHAR(50);

DECLARE cursor_1 CURSOR FOR SELECT first_name, last_name FROM customer

WHERE last_name = 'Scott';

OPEN cursor_1;

REPEAT FETCH cursor_1 INTO a,b;

UNTIL b=0

END REPEAT;

SELECT a AS first_name , b as last_name;

CLOSE cursor_1;

END //

CALL First_Scott()
```



first_name	last_name
Samuel	Scott

***