Course-End Project

Problem Statement Scenario:

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.

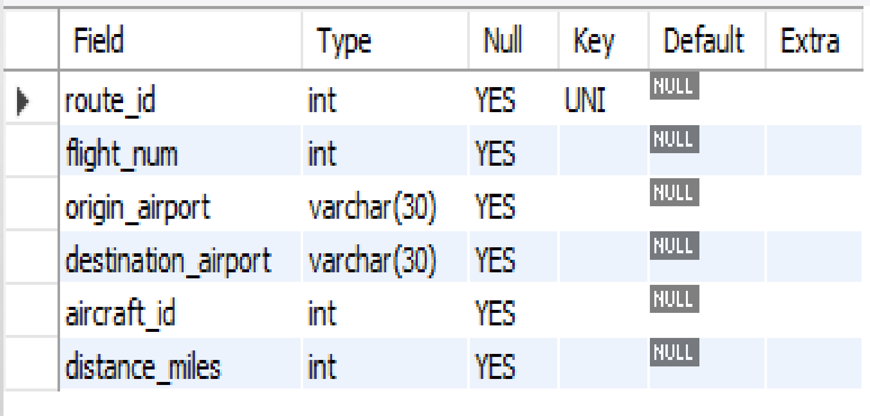
Following operations should be performed:

1. Write a query to create a 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.

➢ **SOLUTION:**

```
create table route_details(
 route_id int UNIQUE,flight_num int check(flight_num>=1111),
 origin_airport varchar(30),
 destination_airport varchar(30),
 aircraft_id int,distance_miles int check(distance_miles>0) );
 select*from route_details;
```

OUTPUT:



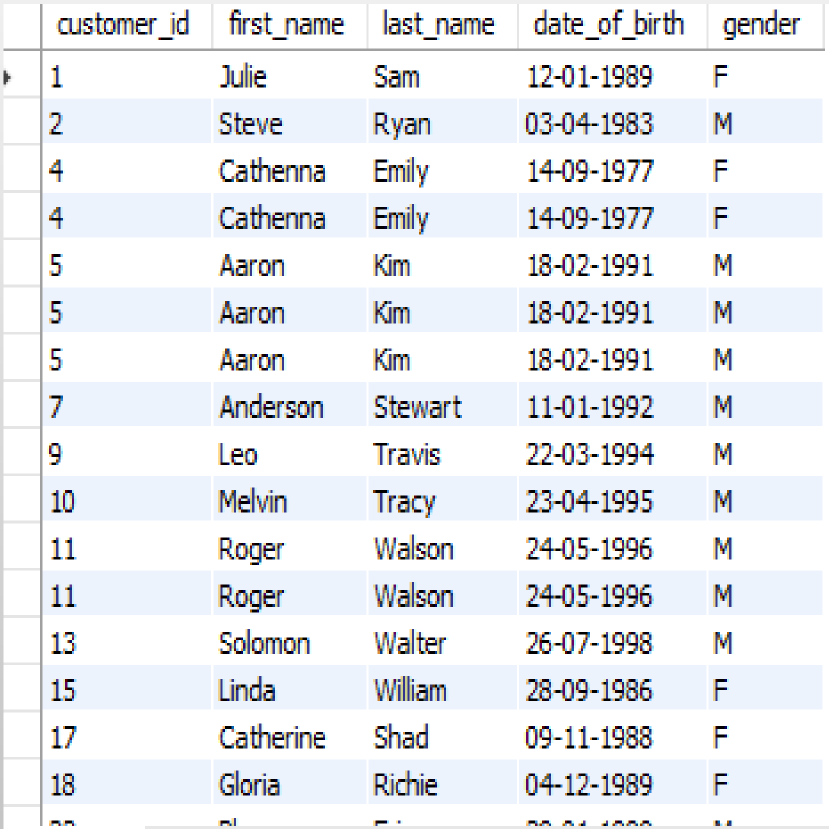
	Field	Type	Null	Key	Default	Extra
▶	route_id	int	YES	UNI	NULL	
	flight_num	int	YES		NULL	
	origin_airport	varchar(30)	YES		NULL	
	destination_airport	varchar(30)	YES		NULL	
	aircraft_id	int	YES		NULL	
	distance_miles	int	YES		NULL	

2. 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.

- **SOLUTION:**

SELECT customer.customer_id, customer.first_name, customer.last_name, customer.date_of_birth, customer.genderFROM passengers_on_flightsJOIN customer ON customer.customer_id = passengers_on_flights.customer_idWHERE passengers_on_flights.route_id BETWEEN 1 AND 25;

OUTPUT:



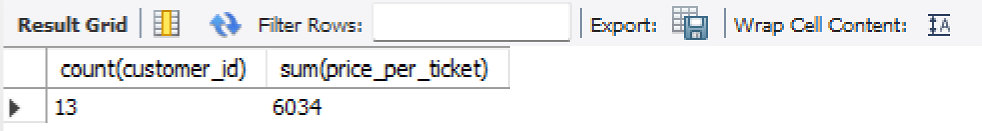
customer_id	first_name	last_name	date_of_birth	gender
1	Julie	Sam	12-01-1989	F
2	Steve	Ryan	03-04-1983	M
4	Cathenna	Emily	14-09-1977	F
4	Cathenna	Emily	14-09-1977	F
5	Aaron	Kim	18-02-1991	M
5	Aaron	Kim	18-02-1991	M
5	Aaron	Kim	18-02-1991	M
7	Anderson	Stewart	11-01-1992	M
9	Leo	Travis	22-03-1994	M
10	Melvin	Tracy	23-04-1995	M
11	Roger	Walson	24-05-1996	M
11	Roger	Walson	24-05-1996	M
13	Solomon	Walter	26-07-1998	M
15	Linda	William	28-09-1986	F
17	Catherine	Shad	09-11-1988	F
18	Gloria	Richie	04-12-1989	F

3. Write a query to identify the number of passengers and total revenue in business class from the ticket_details table.

➢ **SOLUTION:**

```
select count(customer_id),sum(price_per_ticket)
from ticket_details where class_id="Bussiness";
```

OUTPUT:



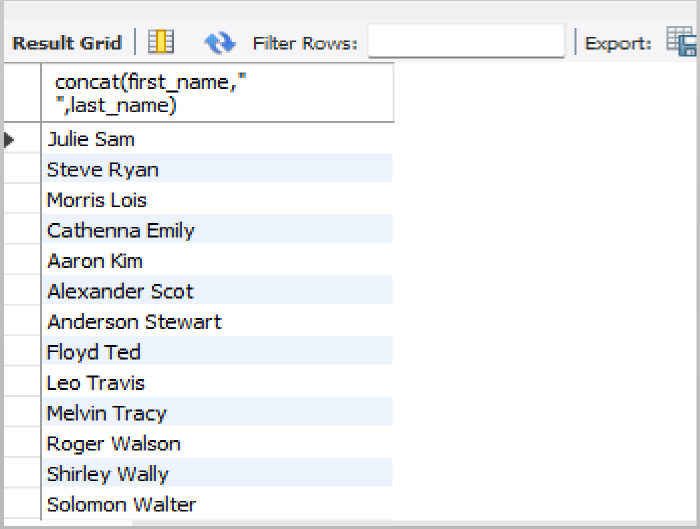
count(customer_id)	sum(price_per_ticket)
13	6034

4. Write a query to display the full name of the customer by extracting the first name and last name from the customer table.

➢ **SOLUTION:**

```
select concat(first_name," ",last_name) from customer;
```

OUTPUT:



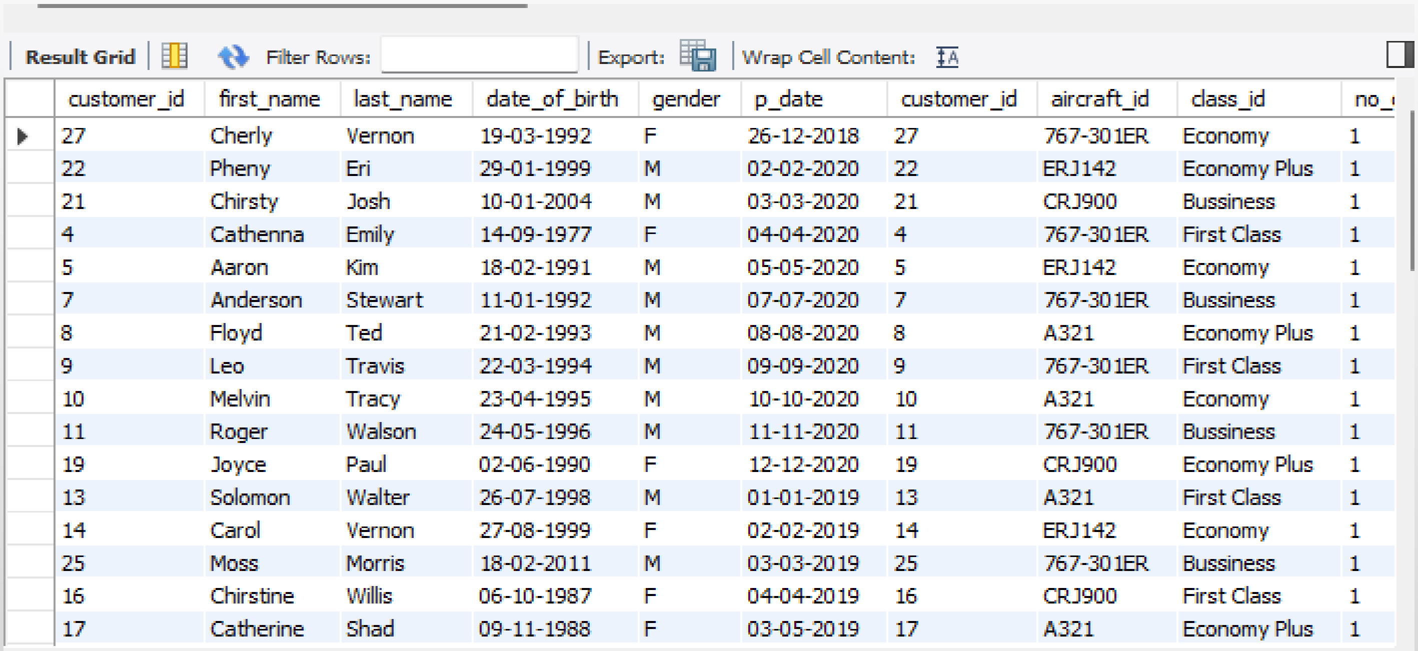
concat(first_name," ",last_name)
Julie Sam
Steve Ryan
Morris Lois
Cathenna Emily
Aaron Kim
Alexander Scot
Anderson Stewart
Floyd Ted
Leo Travis
Melvin Tracy
Roger Walson
Shirley Wally
Solomon Walter

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

➢ **SOLUTION:**

```
select *from customer join ticket_details on customer.customer_id=ticket_details.customer_id;
```

OUTPUT:-



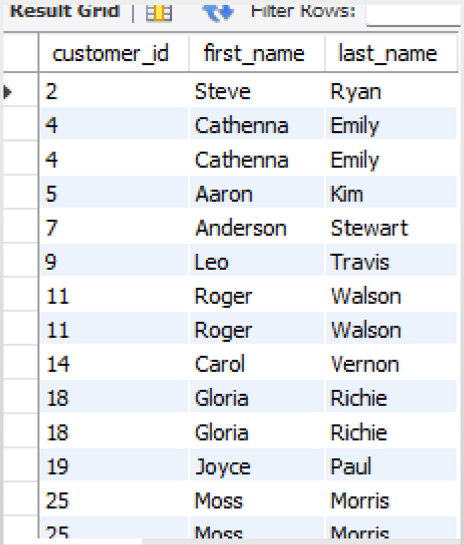
customer_id	first_name	last_name	date_of_birth	gender	p_date	customer_id	aircraft_id	class_id	no_
27	Cherly	Vernon	19-03-1992	F	26-12-2018	27	767-301ER	Economy	1
22	Pheny	Eri	29-01-1999	M	02-02-2020	22	ERJ142	Economy Plus	1
21	Chirsty	Josh	10-01-2004	M	03-03-2020	21	CRJ900	Bussiness	1
4	Cathenna	Emily	14-09-1977	F	04-04-2020	4	767-301ER	First Class	1
5	Aaron	Kim	18-02-1991	M	05-05-2020	5	ERJ142	Economy	1
7	Anderson	Stewart	11-01-1992	M	07-07-2020	7	767-301ER	Bussiness	1
8	Floyd	Ted	21-02-1993	M	08-08-2020	8	A321	Economy Plus	1
9	Leo	Travis	22-03-1994	M	09-09-2020	9	767-301ER	First Class	1
10	Melvin	Tracy	23-04-1995	M	10-10-2020	10	A321	Economy	1
11	Roger	Walson	24-05-1996	M	11-11-2020	11	767-301ER	Bussiness	1
19	Joyce	Paul	02-06-1990	F	12-12-2020	19	CRJ900	Economy Plus	1
13	Solomon	Walter	26-07-1998	M	01-01-2019	13	A321	First Class	1
14	Carol	Vernon	27-08-1999	F	02-02-2019	14	ERJ142	Economy	1
25	Moss	Morris	18-02-2011	M	03-03-2019	25	767-301ER	Bussiness	1
16	Chirstine	Willis	06-10-1987	F	04-04-2019	16	CRJ900	First Class	1
17	Catherine	Shad	09-11-1988	F	03-05-2019	17	A321	Economy Plus	1

6. 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.

- **SOLUTION:**

```
SELECT c.customer_id,
             c.first_name,
             c.last_name
FROM customer c
JOIN ticket_details t ON c.customer_id = t.customer_id
     WHERE   t.brand = 'Emirates';
```

OUTPUT:



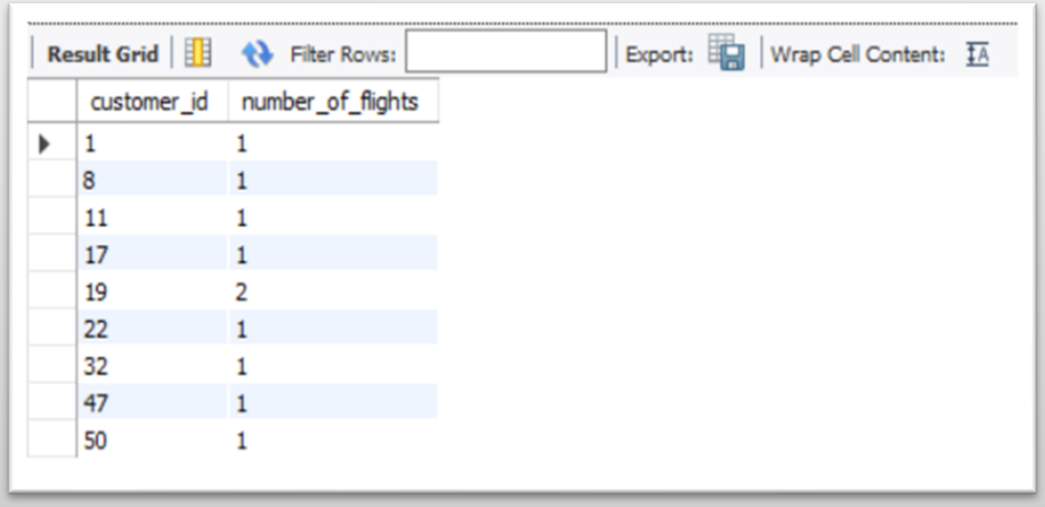
customer_id	first_name	last_name
2	Steve	Ryan
4	Cathenna	Emily
4	Cathenna	Emily
5	Aaron	Kim
7	Anderson	Stewart
9	Leo	Travis
11	Roger	Walson
11	Roger	Walson
14	Carol	Vernon
18	Gloria	Richie
18	Gloria	Richie
19	Joyce	Paul
25	Moss	Morris
25	Moss	Morris

7. 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.

- **SOLUTION:**

select customer_id,count(seat_num) from passengers_on_flights where class_id="Economy Plus"group by(customer_id) having count(*)>0;

OUTPUT:



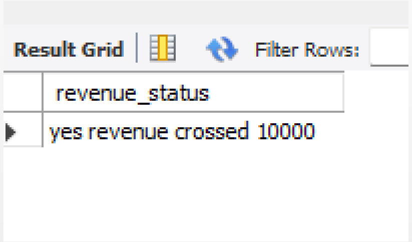
customer_id	number_of_flights
1	1
8	1
11	1
17	1
19	2
22	1
32	1
47	1
50	1

8. Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket_details table

- **SOLUTION:**

select if(sum(price_per_ticket)>10000,"yes revenue crossed 10000","no revenue less than 10000")AS revenue_status from ticket_details;

OUTPUT:



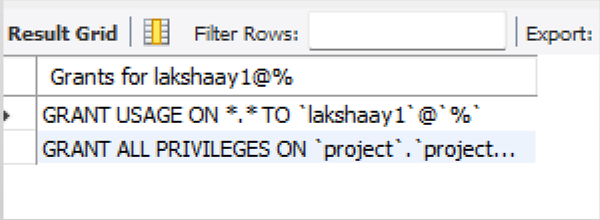
revenue_status
yes revenue crossed 10000

9. Write a query to create and grant access to a new user to perform operations on a database

- **SOLUTION:**

```
USE PROJECT;
create user lakshaay1;
GRANT ALL PRIVILEGES ON project TO 'lakshaay1’;
show grants for lakshaay1;
```

OUTPUT:



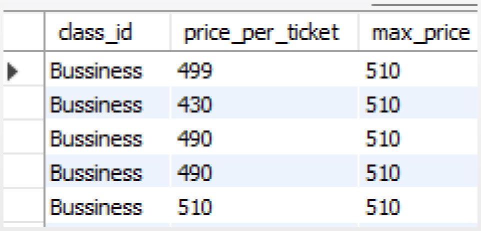
Grants for lakshaay1@%
GRANT USAGE ON *.* TO `lakshaay1`@`%`
GRANT ALL PRIVILEGES ON `project`.`project...

10. Write a query to find the maximum ticket price for each class using window functions on the ticket_details table.

- **SOLUTION:**

```
      select class_id, price_per_ticket, max(price_per_ticket)
    over (partition by class_id) As max_price from ticket_details;
```

OUTPUT:



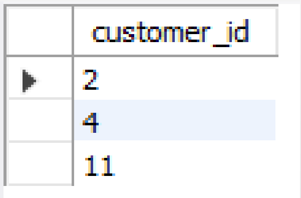
class_id	price_per_ticket	max_price
Bussiness	499	510
Bussiness	430	510
Bussiness	490	510
Bussiness	490	510
Bussiness	510	510

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

- **SOLUTION:**

```
 create index lakshay on
passengers_on_flights(route
_id); select customer_id
from passengers_on_flights
where route_id=4;
```

OUTPUT:



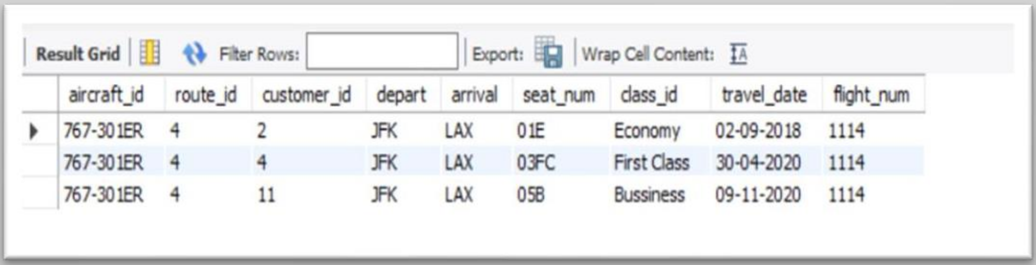
customer_id
2
4
11

12. For the route ID 4, write a query to view the execution plan of the passengers_on_flights table.

- **SOLUTION:**

```
select*from passengers_on_flights where route_id=4 ;
```

OUTPUT:



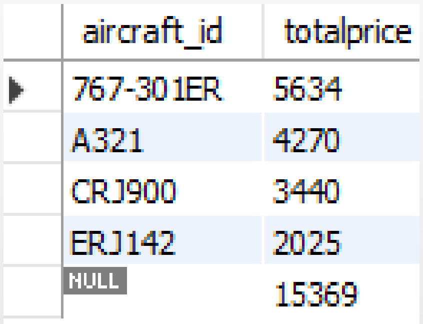
aircraft_id	route_id	customer_id	depart	arrival	seat_num	class_id	travel_date	flight_num
767-301ER	4	2	JFK	LAX	01E	Economy	02-09-2018	1114
767-301ER	4	4	JFK	LAX	03FC	First Class	30-04-2020	1114
767-301ER	4	11	JFK	LAX	05B	Bussiness	09-11-2020	1114

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

- **SOLUTION:**

```
select price_per_ticket from ticket_details;select aircraft_id,sum(price_per_ticket*no_of_tickets) as
totalpricefrom ticket_details group by aircraft_id with rollup;
```

OUTPUT:



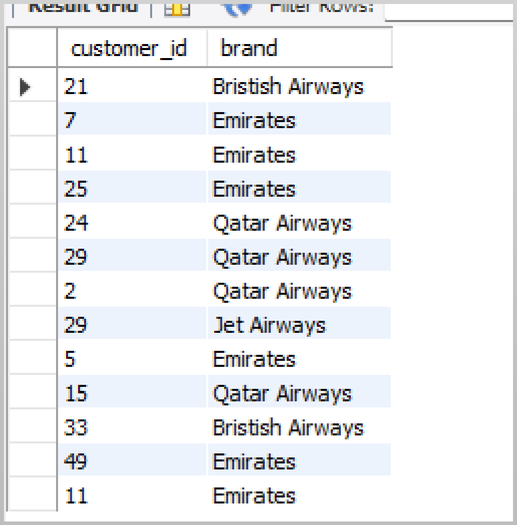
aircraft_id	totalprice
767-301ER	5634
A321	4270
CRJ900	3440
ERJ142	2025
NULL	15369

14. Write a query to create a view with only business class customers along with the brand of airlines.

- **SOLUTION:**

```
create view projectview As select customer_id
,brand from ticket_details where
class_id="Bussiness";select*from projectview;
```

OUTPUT:



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

15. 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.

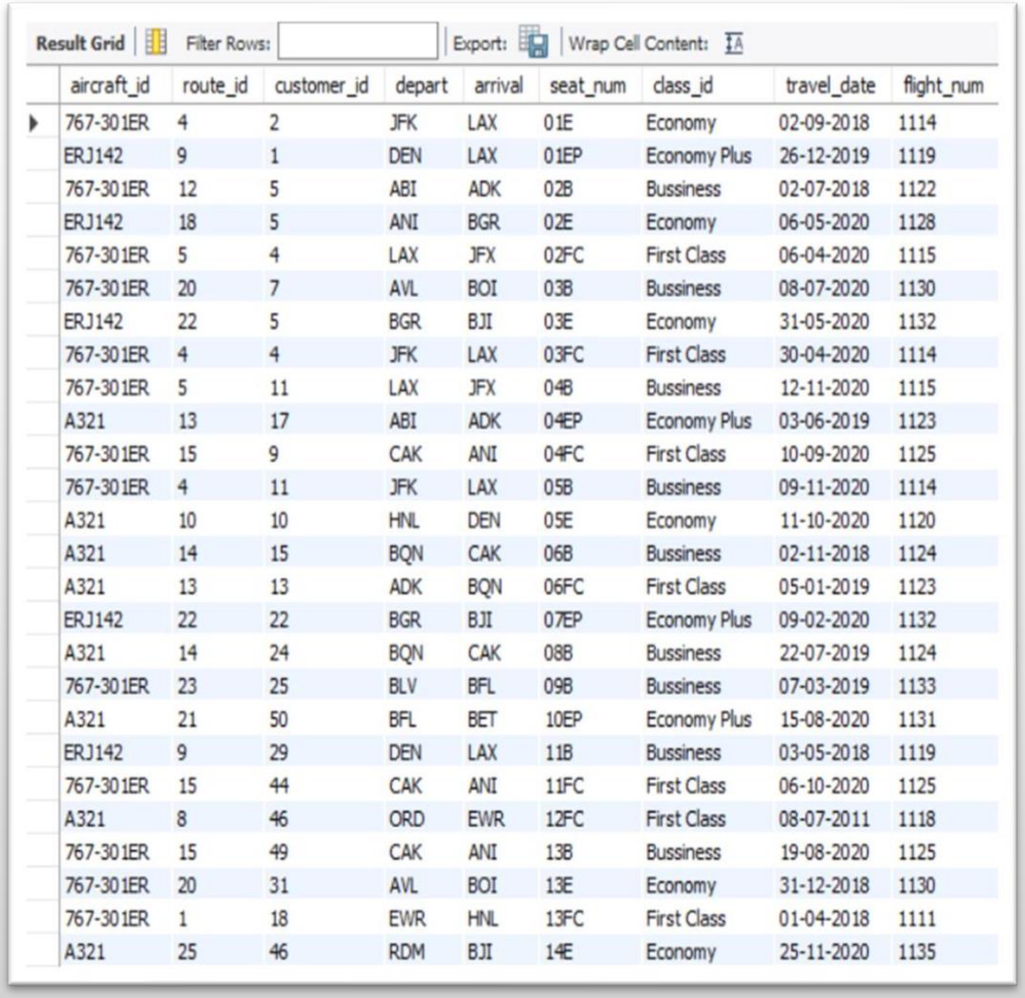
- **SOLUTION:**

```
CREATE PROCEDURE GetPassengers ( IN
    start_route_id INT,
    IN end_route_id INT
)
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        IF SQLSTATE = '42S02' THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Error: Table Passengers_on_flights does not exist';
        ELSE
          RESIGNAL;
        END IF;
    END;
    SELECT
        aircraft_id, route_id, customer_id, depart, arrival, seat_num, class_id, travel_date, flight_num
    FROM
        Passengers_on_flights
    WHERE
        route_id BETWEEN start_route_id AND end_route_id;

END
```

```
CALL GetPassengers (1, 25);
```

OUTPUT:



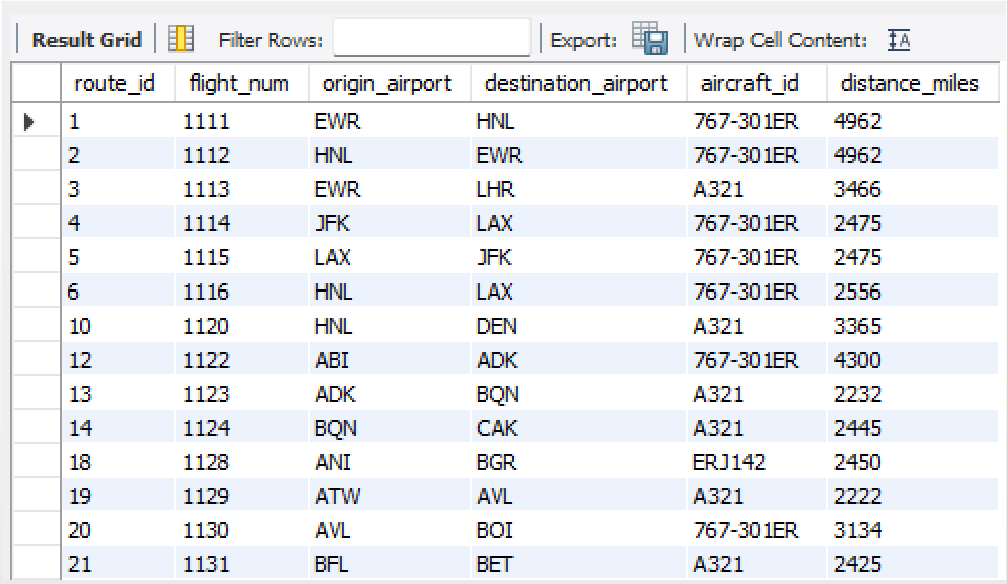
aircraft_id	route_id	customer_id	depart	arrival	seat_num	class_id	travel_date	flight_num
767-301ER	4	2	JFK	LAX	01E	Economy	02-09-2018	1114
ERJ142	9	1	DEN	LAX	01EP	Economy Plus	26-12-2019	1119
767-301ER	12	5	ABI	ADK	02B	Bussiness	02-07-2018	1122
ERJ142	18	5	ANI	BGR	02E	Economy	06-05-2020	1128
767-301ER	5	4	LAX	JFX	02FC	First Class	06-04-2020	1115
767-301ER	20	7	AVL	BOI	03B	Bussiness	08-07-2020	1130
ERJ142	22	5	BGR	BJI	03E	Economy	31-05-2020	1132
767-301ER	4	4	JFK	LAX	03FC	First Class	30-04-2020	1114
767-301ER	5	11	LAX	JFX	04B	Bussiness	12-11-2020	1115
A321	13	17	ABI	ADK	04EP	Economy Plus	03-06-2019	1123
767-301ER	15	9	CAK	ANI	04FC	First Class	10-09-2020	1125
767-301ER	4	11	JFK	LAX	05B	Bussiness	09-11-2020	1114
A321	10	10	HNL	DEN	05E	Economy	11-10-2020	1120
A321	14	15	BQN	CAK	06B	Bussiness	02-11-2018	1124
A321	13	13	ADK	BQN	06FC	First Class	05-01-2019	1123
ERJ142	22	22	BGR	BJI	07EP	Economy Plus	09-02-2020	1132
A321	14	24	BQN	CAK	08B	Bussiness	22-07-2019	1124
767-301ER	23	25	BLV	BFL	09B	Bussiness	07-03-2019	1133
A321	21	50	BFL	BET	10EP	Economy Plus	15-08-2020	1131
ERJ142	9	29	DEN	LAX	11B	Bussiness	03-05-2018	1119
767-301ER	15	44	CAK	ANI	11FC	First Class	06-10-2020	1125
A321	8	46	ORD	EWR	12FC	First Class	08-07-2011	1118
767-301ER	15	49	CAK	ANI	13B	Bussiness	19-08-2020	1125
767-301ER	20	31	AVL	BOI	13E	Economy	31-12-2018	1130
767-301ER	1	18	EWR	HNL	13FC	First Class	01-04-2018	1111
A321	25	46	RDM	BJI	14E	Economy	25-11-2020	1135

16. 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.

- **SOLUTION:**

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `lakshayy`()
BEGIN
select*from route_detail where distance_miles>2000;
END
call lakshayy;
```

OUTPUT:



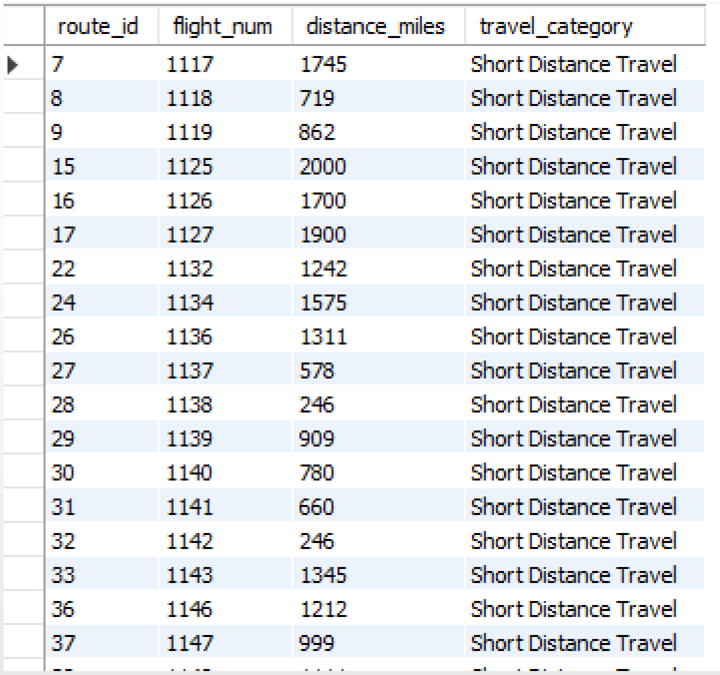
route_id	flight_num	origin_airport	destination_airport	aircraft_id	distance_miles
1	1111	EWR	HNL	767-301ER	4962
2	1112	HNL	EWR	767-301ER	4962
3	1113	EWR	LHR	A321	3466
4	1114	JFK	LAX	767-301ER	2475
5	1115	LAX	JFK	767-301ER	2475
6	1116	HNL	LAX	767-301ER	2556
10	1120	HNL	DEN	A321	3365
12	1122	ABI	ADK	767-301ER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

17. 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.

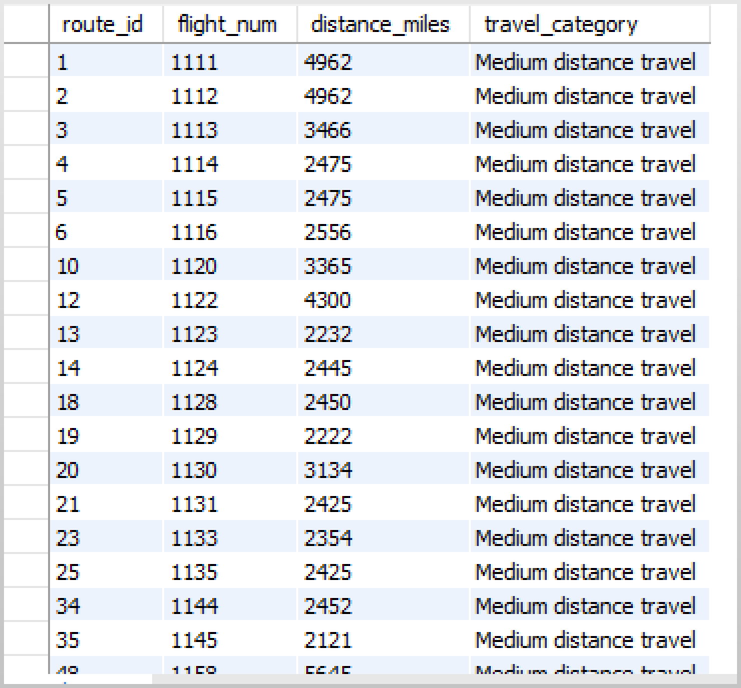
- **SOLUTION:**

```
CREATE
DEFINER=`root`@`localhost`PROCEDURE`helllo`()
begin
SELECT route_id,       flight_num,       distance_miles,
'Short Distance Travel' AS travel_category
FROM route_detail    WHERE distance_miles >= 0
AND distance_miles <= 2000
UNION ALL
 SELECT route_id,flight_num,distance_miles,'Medium
distance travel' As    travel_category
   FROM route_detail
   WHERE distance_miles > 2000 AND distance_miles
<= 6500
   UNION ALL   SELECT
route_id,flight_num,distance_miles,'Long distance
travel' As   travel_category
  FROM route_detail WHERE distance_miles > 6500;
end
CALL hello();
```

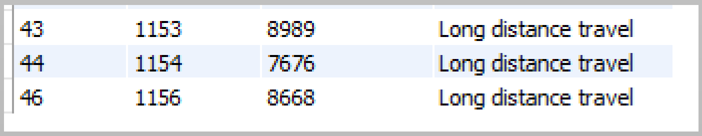
OUTPUT:



route_id	flight_num	distance_miles	travel_category
7	1117	1745	Short Distance Travel
8	1118	719	Short Distance Travel
9	1119	862	Short Distance Travel
15	1125	2000	Short Distance Travel
16	1126	1700	Short Distance Travel
17	1127	1900	Short Distance Travel
22	1132	1242	Short Distance Travel
24	1134	1575	Short Distance Travel
26	1136	1311	Short Distance Travel
27	1137	578	Short Distance Travel
28	1138	246	Short Distance Travel
29	1139	909	Short Distance Travel
30	1140	780	Short Distance Travel
31	1141	660	Short Distance Travel
32	1142	246	Short Distance Travel
33	1143	1345	Short Distance Travel
36	1146	1212	Short Distance Travel
37	1147	999	Short Distance Travel



route_id	flight_num	distance_miles	travel_category
1	1111	4962	Medium distance travel
2	1112	4962	Medium distance travel
3	1113	3466	Medium distance travel
4	1114	2475	Medium distance travel
5	1115	2475	Medium distance travel
6	1116	2556	Medium distance travel
10	1120	3365	Medium distance travel
12	1122	4300	Medium distance travel
13	1123	2232	Medium distance travel
14	1124	2445	Medium distance travel
18	1128	2450	Medium distance travel
19	1129	2222	Medium distance travel
20	1130	3134	Medium distance travel
21	1131	2425	Medium distance travel
23	1133	2354	Medium distance travel
25	1135	2425	Medium distance travel
34	1144	2452	Medium distance travel
35	1145	2121	Medium distance travel



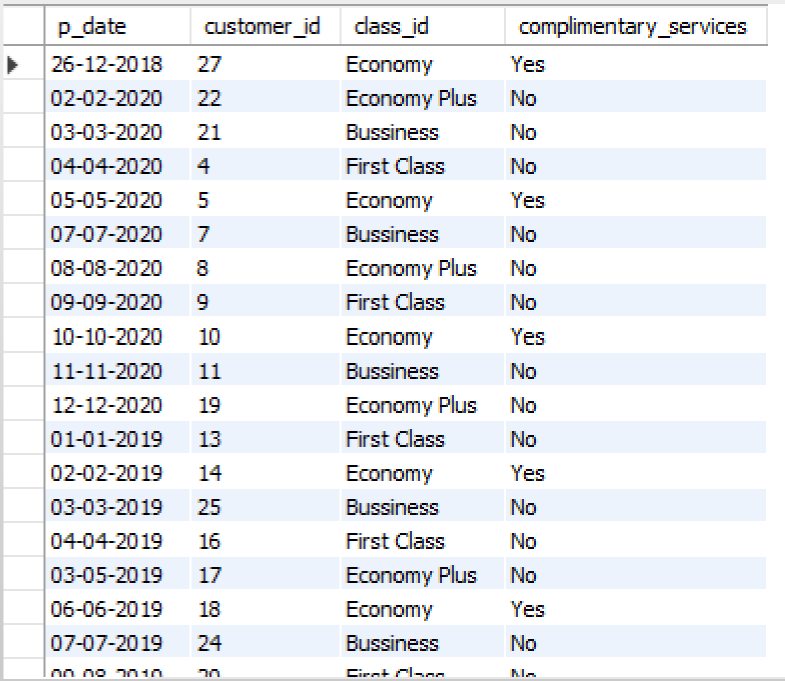
43	1153	8989	Long distance travel
44	1154	7676	Long distance travel
46	1156	8668	Long distance travel

18. 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

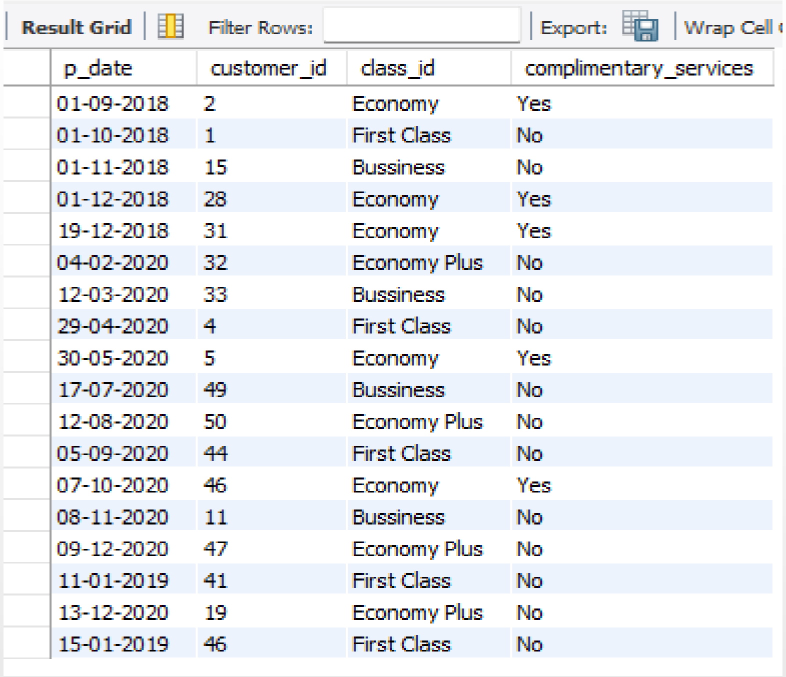
- **SOLUTION:**

```
CREATE DEFINER=`root`@`localhost` PROCEDURE`eighteen`()
BEGIN
SELECT     p_date,    customer_id,     class_id,
CASE   WHEN class_id = 'Business ' THEN  'Yes'
 WHEN class_id = 'Economy' THEN  'Yes'
 ELSE 'No'    END
AS complimentary_services
FROM     ticket_details;
END
CALL eighteen();
```

OUTPUT:



p_date	customer_id	class_id	complimentary_services
26-12-2018	27	Economy	Yes
02-02-2020	22	Economy Plus	No
03-03-2020	21	Bussiness	No
04-04-2020	4	First Class	No
05-05-2020	5	Economy	Yes
07-07-2020	7	Bussiness	No
08-08-2020	8	Economy Plus	No
09-09-2020	9	First Class	No
10-10-2020	10	Economy	Yes
11-11-2020	11	Bussiness	No
12-12-2020	19	Economy Plus	No
01-01-2019	13	First Class	No
02-02-2019	14	Economy	Yes
03-03-2019	25	Bussiness	No
04-04-2019	16	First Class	No
03-05-2019	17	Economy Plus	No
06-06-2019	18	Economy	Yes
07-07-2019	24	Bussiness	No
09-08-2019	20	First Class	No



p_date	customer_id	class_id	complimentary_services
01-09-2018	2	Economy	Yes
01-10-2018	1	First Class	No
01-11-2018	15	Bussiness	No
01-12-2018	28	Economy	Yes
19-12-2018	31	Economy	Yes
04-02-2020	32	Economy Plus	No
12-03-2020	33	Bussiness	No
29-04-2020	4	First Class	No
30-05-2020	5	Economy	Yes
17-07-2020	49	Bussiness	No
12-08-2020	50	Economy Plus	No
05-09-2020	44	First Class	No
07-10-2020	46	Economy	Yes
08-11-2020	11	Bussiness	No
09-12-2020	47	Economy Plus	No
11-01-2019	41	First Class	No
13-12-2020	19	Economy Plus	No
15-01-2019	46	First Class	No

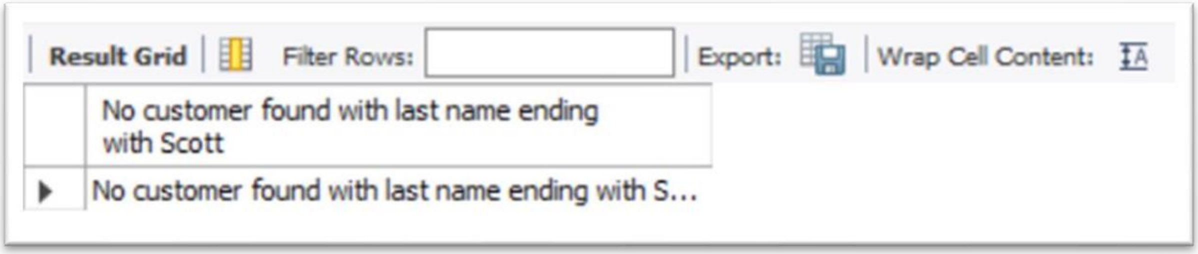
19. Write a query to extract the first record of the customer whose last name ends with Scott using a cursor from the customer table.

- **SOLUTION:**

```
CREATE PROCEDURE GetCustomerByLastName()
BEGIN
   DECLARE done INT DEFAULT FALSE;
   DECLARE customer_id INT;
   DECLARE first_name VARCHAR(50);
   DECLARE last_name VARCHAR(50);
   DECLARE date_of_birth DATE;
   DECLARE gender CHAR(1);
   DECLARE customer_cursor CURSOR FOR
      SELECT customer_id, first_name, last_name, date_of_birth, gender
      FROM customer
      WHERE last_name LIKE '%Scott'
      LIMIT 1;
   DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
   OPEN customer_cursor;
   FETCH customer_cursor INTO customer_id, first_name, last_name, date_of_birth, gender;
   IF NOT done THEN
      SELECT customer_id, first_name, last_name, date_of_birth, gender;
   ELSE
      SELECT 'No customer found with last name ending with Scott';
   END IF;
   CLOSE customer_cursor;
END

CALL  GetCustomerByLastName  ();
```

OUTPUT:



```
No customer found with last name ending
with Scott
No customer found with last name ending with S...
```

THANK YOU!

- **SUBMITTED TO :**

  Mr. Veera

- **SUBMITTED BY :**

  NAME:-LAKSHAY DADHICH

  CONTACT:- 6367035116

  COURSE:-SQL