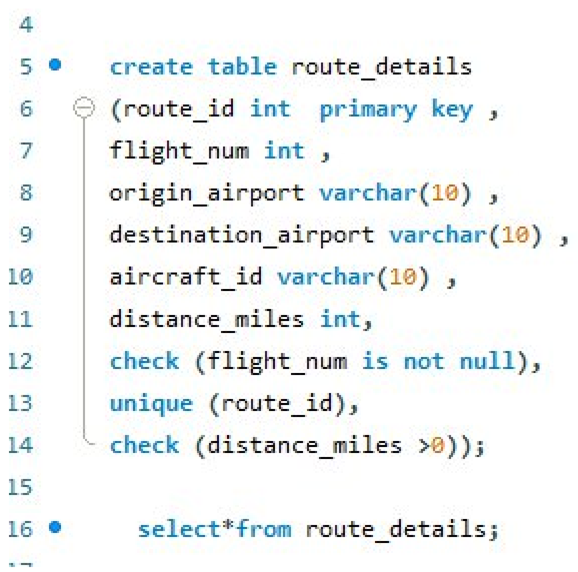
SQL PROJECT 2

AIR CARGO ANALYSIS

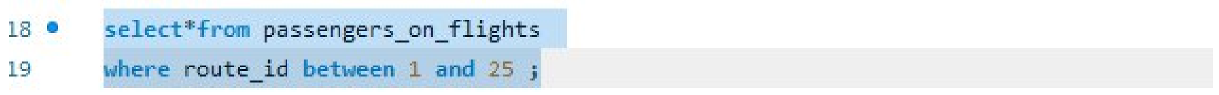
1. Write a query to create route_details table using suitable data types for the fields, such as route_id, flight_num, origin_airport, destination_airport, aircraft_id, and distance_miles. Implement the check constraint for the flight number and unique constraint for the route_id fields. Also, make sure that the distance miles field is greater than 0.



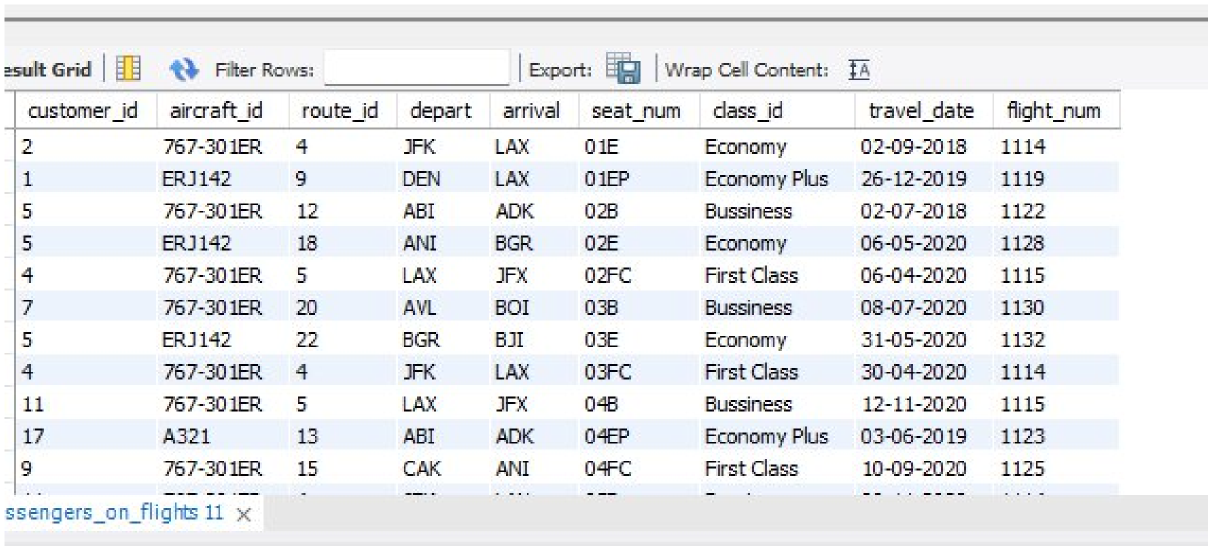
```
create table route_details
(route_id int  primary key ,
flight_num int ,
origin_airport varchar(10) ,
destination_airport varchar(10) ,
aircraft_id varchar(10) ,
distance_miles int,
check (flight_num is not null),
unique (route_id),
check (distance_miles >0));

  select*from route_details;
```

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.



```
select*from passengers_on_flights
where route_id between 1 and 25 ;
```

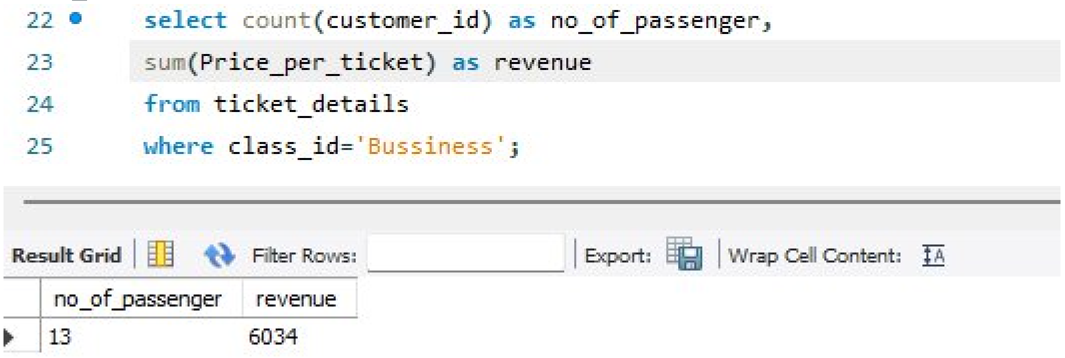


esult Grid | Filter Rows: | Export: | Wrap Cell Content:

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

ssengers_on_flights 11 ×

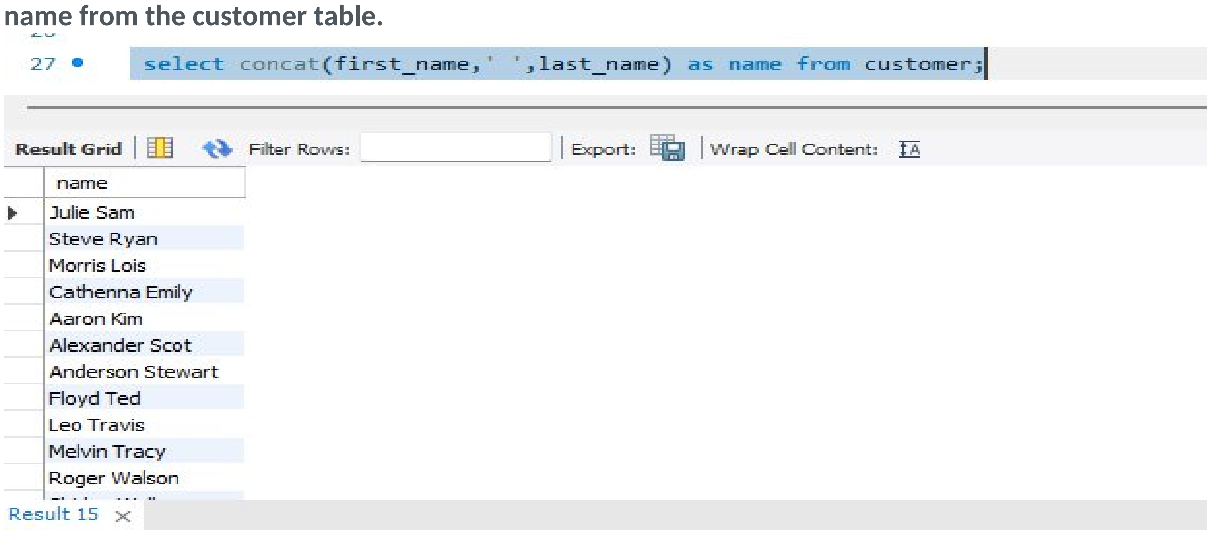
3.Write a query to identify the number of passengers and total revenue in business class from the ticket_details table.



```sql
select count(customer_id) as no_of_passenger,
sum(Price_per_ticket) as revenue
from ticket_details
where class_id='Bussiness';
```

no_of_passenger	revenue
13	6034

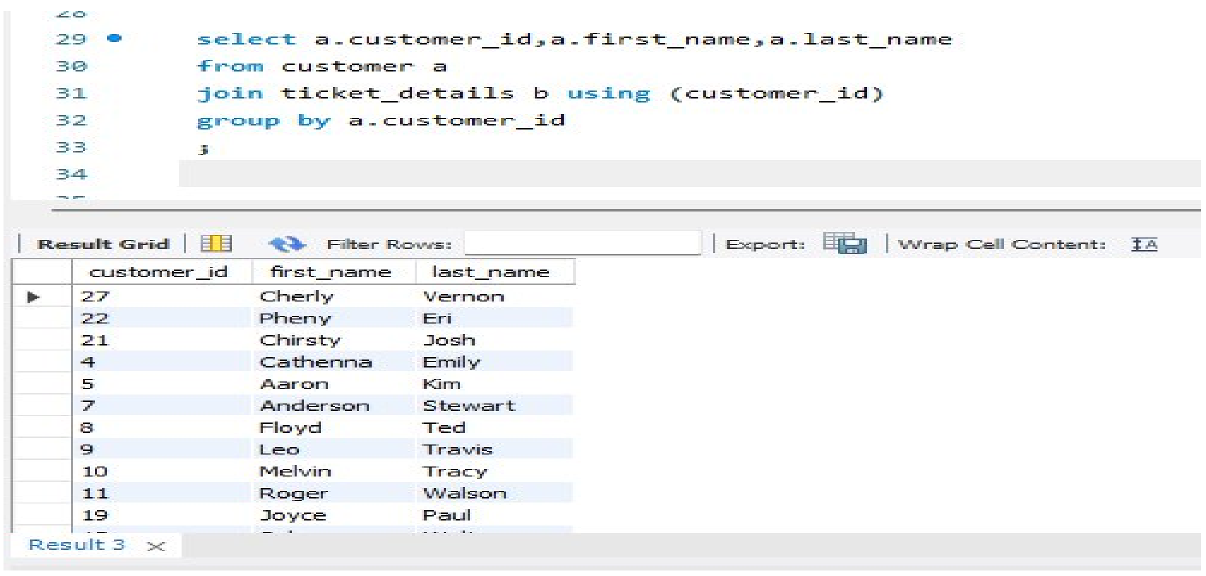
4. Write a query to display the full name of the customer by extracting the first name and last name from the customer table.



```sql
select concat(first_name,' ',last_name) as name from customer;
```

name
Julie Sam
Steve Ryan
Morris Lois
Cathenna Emily
Aaron Kim
Alexander Scot
Anderson Stewart
Floyd Ted
Leo Travis
Melvin Tracy
Roger Walson

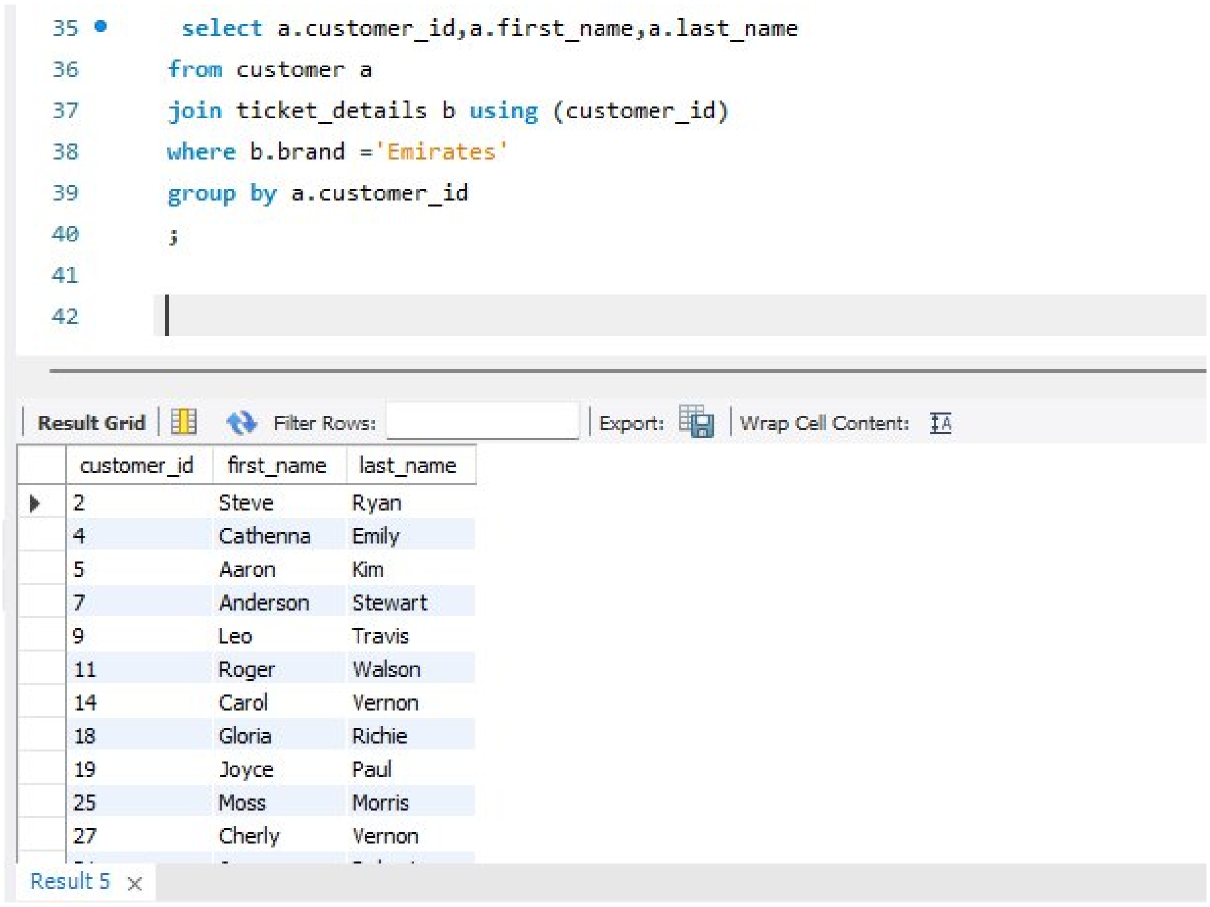
5.Write a query to extract the customers who have registered and booked a ticket. Use data from the customer and ticket_details tables.



```sql
select a.customer_id,a.first_name,a.last_name
from customer a
join ticket_details b using (customer_id)
group by a.customer_id
;
```

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

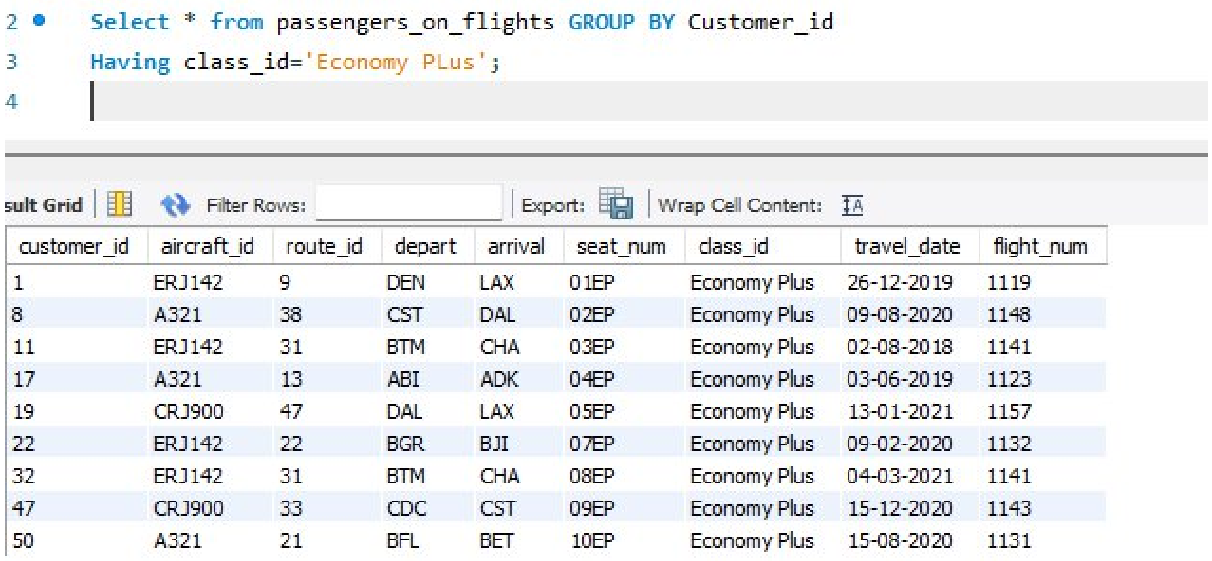
6. Write a query to identify the customer's first name and last name based on their customer ID and brand (Emirates) from the ticket_details table



```sql
 select a.customer_id,a.first_name,a.last_name
from customer a
join ticket_details b using (customer_id)
where b.brand ='Emirates'
group by a.customer_id
;
```

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

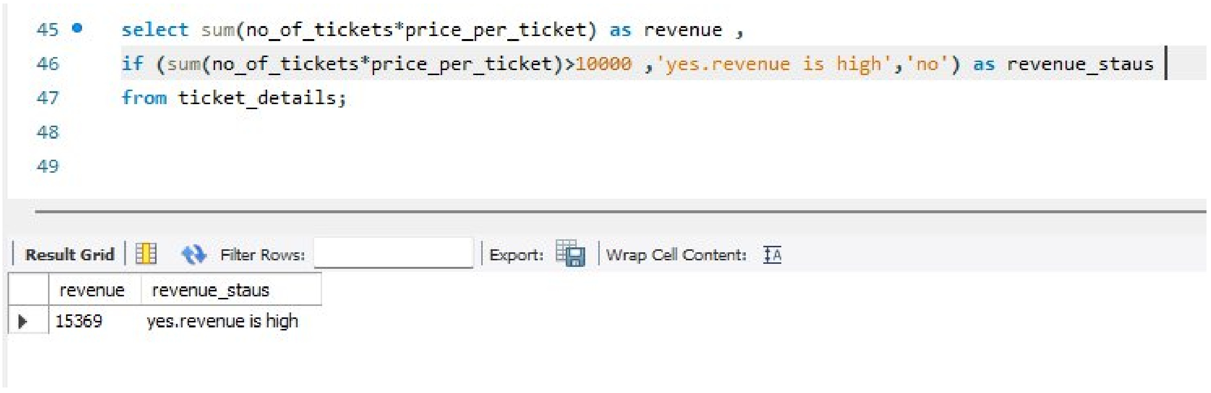
7. Write a query to identify the customers who have travelled by *Economy Plus* class using Group By and Having clause on the passengers_on_flights table.



```sql
Select * from passengers_on_flights GROUP BY Customer_id
Having class_id='Economy PLus';
```

customer_id	aircraft_id	route_id	depart	arrival	seat_num	class_id	travel_date	flight_num
1	ERJ142	9	DEN	LAX	01EP	Economy Plus	26-12-2019	1119
8	A321	38	CST	DAL	02EP	Economy Plus	09-08-2020	1148
11	ERJ142	31	BTM	CHA	03EP	Economy Plus	02-08-2018	1141
17	A321	13	ABI	ADK	04EP	Economy Plus	03-06-2019	1123
19	CRJ900	47	DAL	LAX	05EP	Economy Plus	13-01-2021	1157
22	ERJ142	22	BGR	BJI	07EP	Economy Plus	09-02-2020	1132
32	ERJ142	31	BTM	CHA	08EP	Economy Plus	04-03-2021	1141
47	CRJ900	33	CDC	CST	09EP	Economy Plus	15-12-2020	1143
50	A321	21	BFL	BET	10EP	Economy Plus	15-08-2020	1131

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

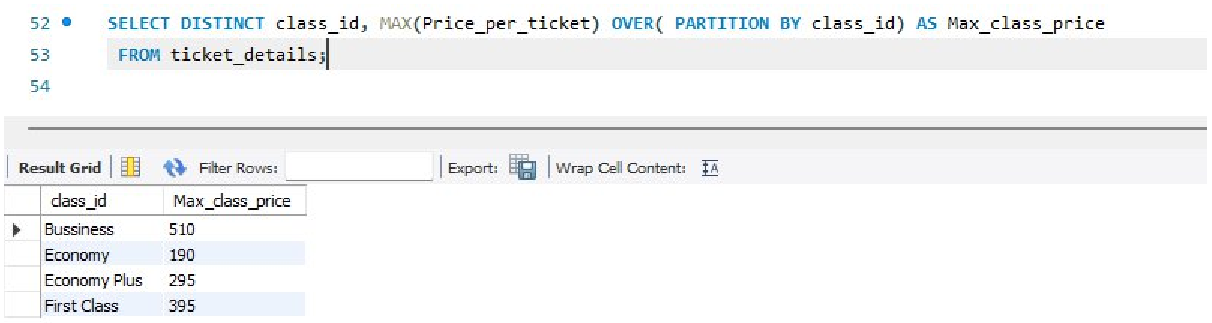


```
select sum(no_of_tickets*price_per_ticket) as revenue ,
if (sum(no_of_tickets*price_per_ticket)>10000 ,'yes.revenue is high','no') as revenue_staus
from ticket_details;
```

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

revenue	revenue_staus
15369	yes.revenue is high

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

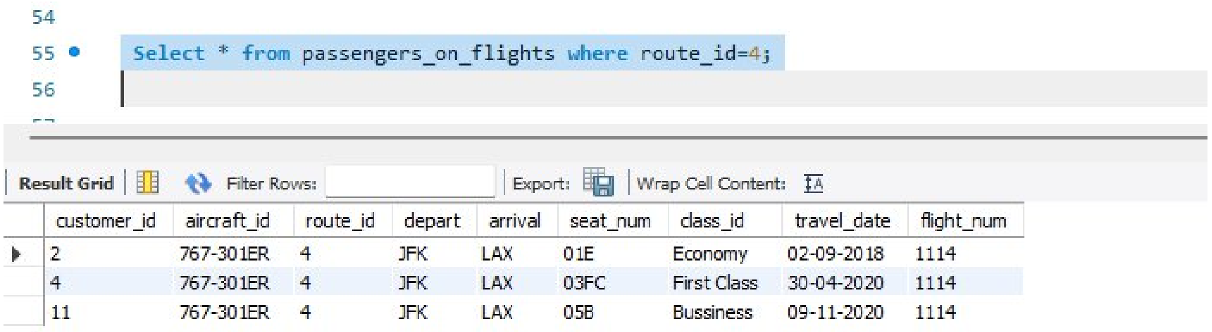


```
SELECT DISTINCT class_id, MAX(Price_per_ticket) OVER( PARTITION BY class_id) AS Max_class_price
 FROM ticket_details;
```

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

class_id	Max_class_price
Bussiness	510
Economy	190
Economy Plus	295
First Class	395

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

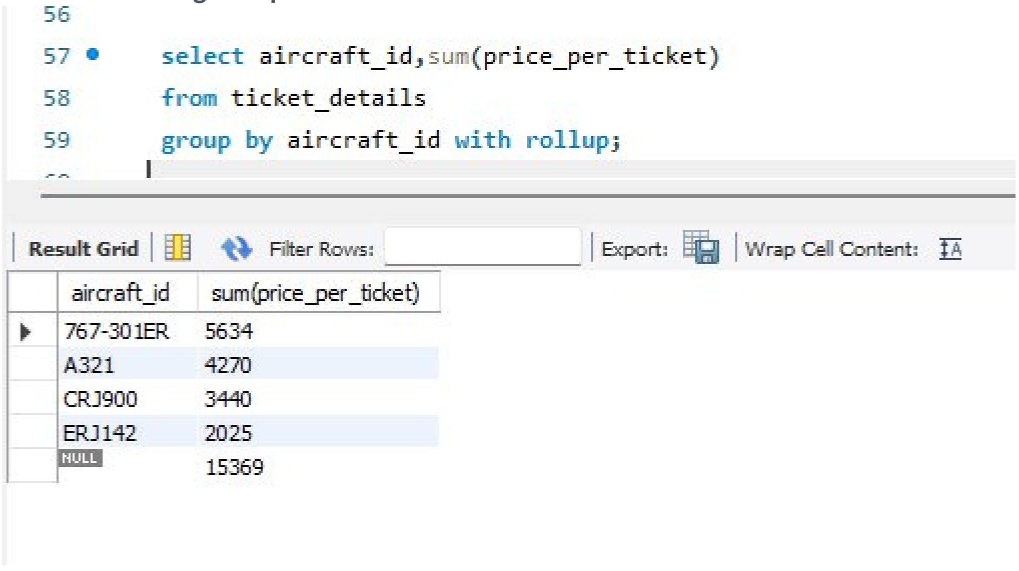


```
Select * from passengers_on_flights where route_id=4;
```

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

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

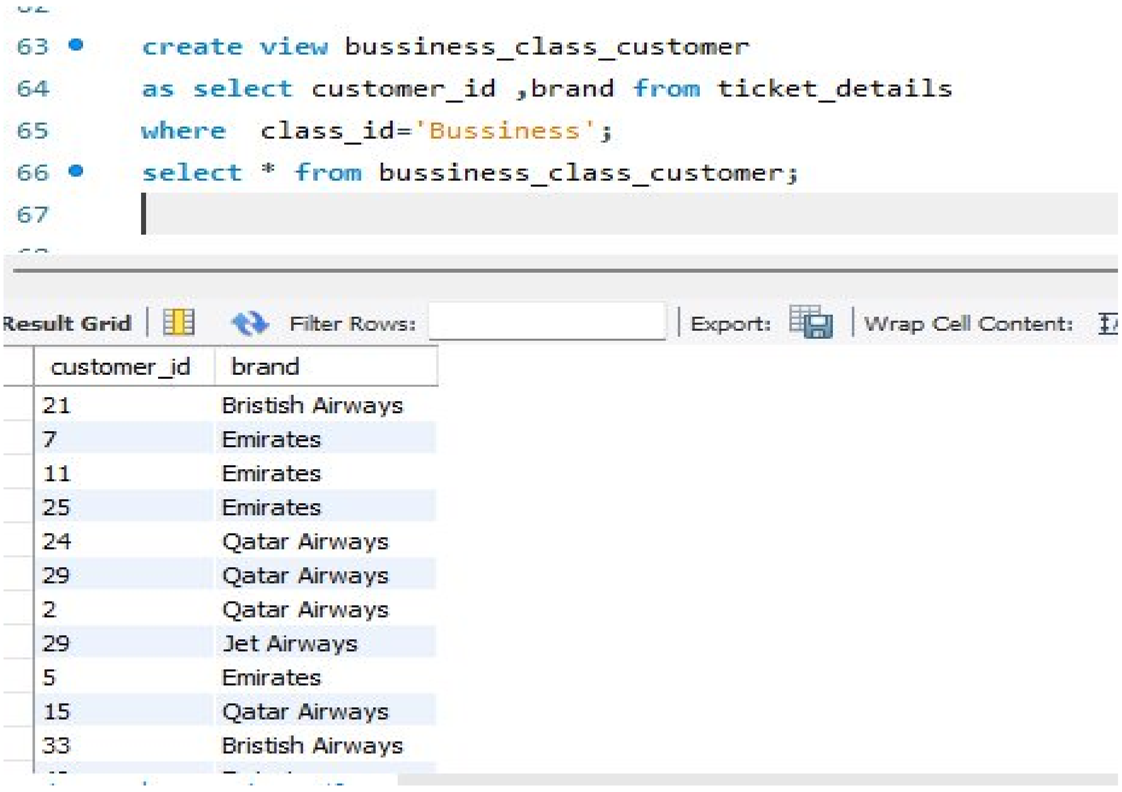


```
select aircraft_id,sum(price_per_ticket)
from ticket_details
group by aircraft_id with rollup;
```

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

aircraft_id	sum(price_per_ticket)
767-301ER	5634
A321	4270
CRJ900	3440
ERJ142	2025
NULL	15369

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

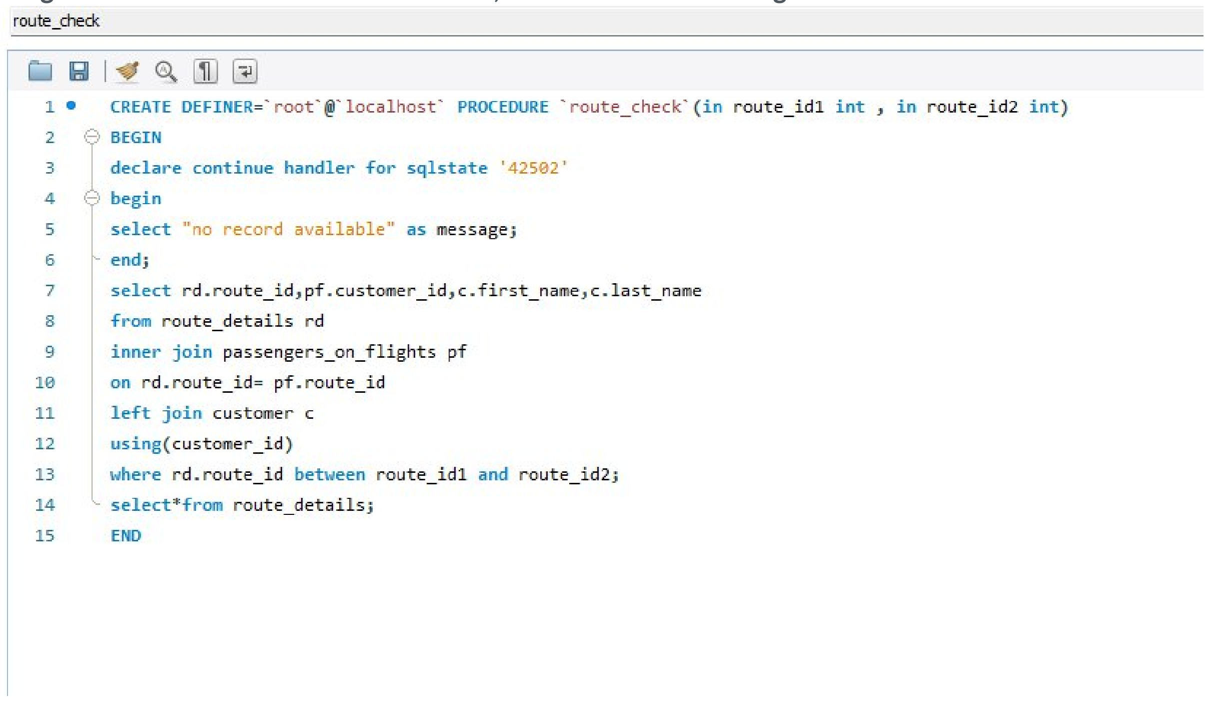


```
create view bussiness_class_customer
as select customer_id ,brand from ticket_details
where  class_id='Bussiness';
select * from bussiness_class_customer;
```

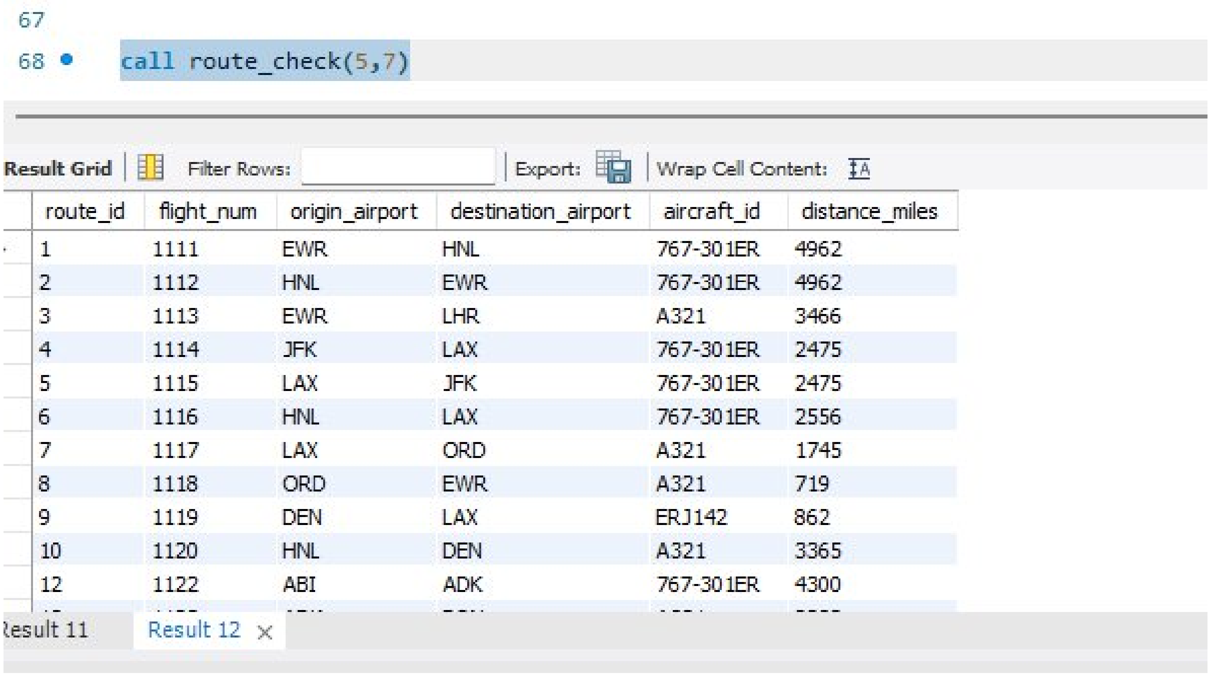
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

13. **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.**


route_check

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `route_check`(in route_id1 int , in route_id2 int)
BEGIN
declare continue handler for sqlstate '42502'
begin
select "no record available" as message;
end;
select rd.route_id,pf.customer_id,c.first_name,c.last_name
from route_details rd
inner join passengers_on_flights pf
on rd.route_id= pf.route_id
left join customer c
using(customer_id)
where rd.route_id between route_id1 and route_id2;
select*from route_details;
END
```

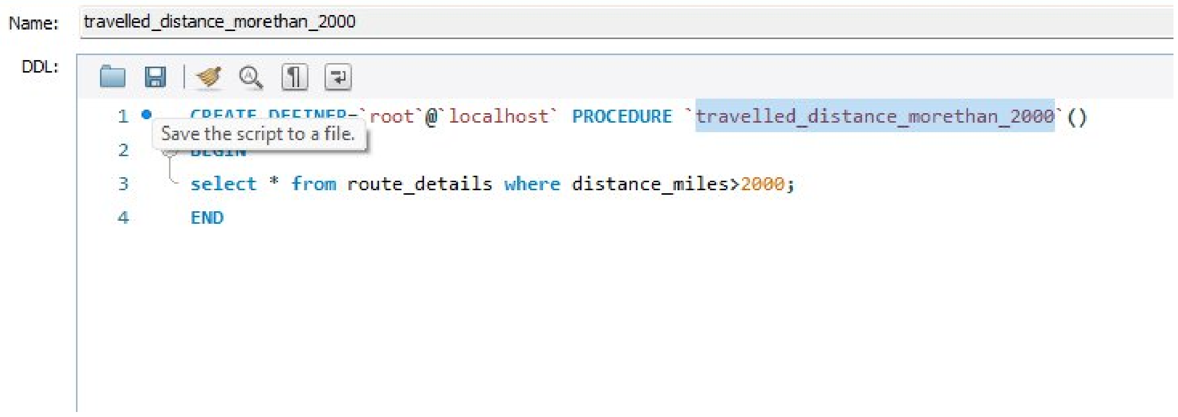


```
call route_check(5,7)
```

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
7	1117	LAX	ORD	A321	1745
8	1118	ORD	EWR	A321	719
9	1119	DEN	LAX	ERJ142	862
10	1120	HNL	DEN	A321	3365
12	1122	ABI	ADK	767-301ER	4300

esult 11 Result 12

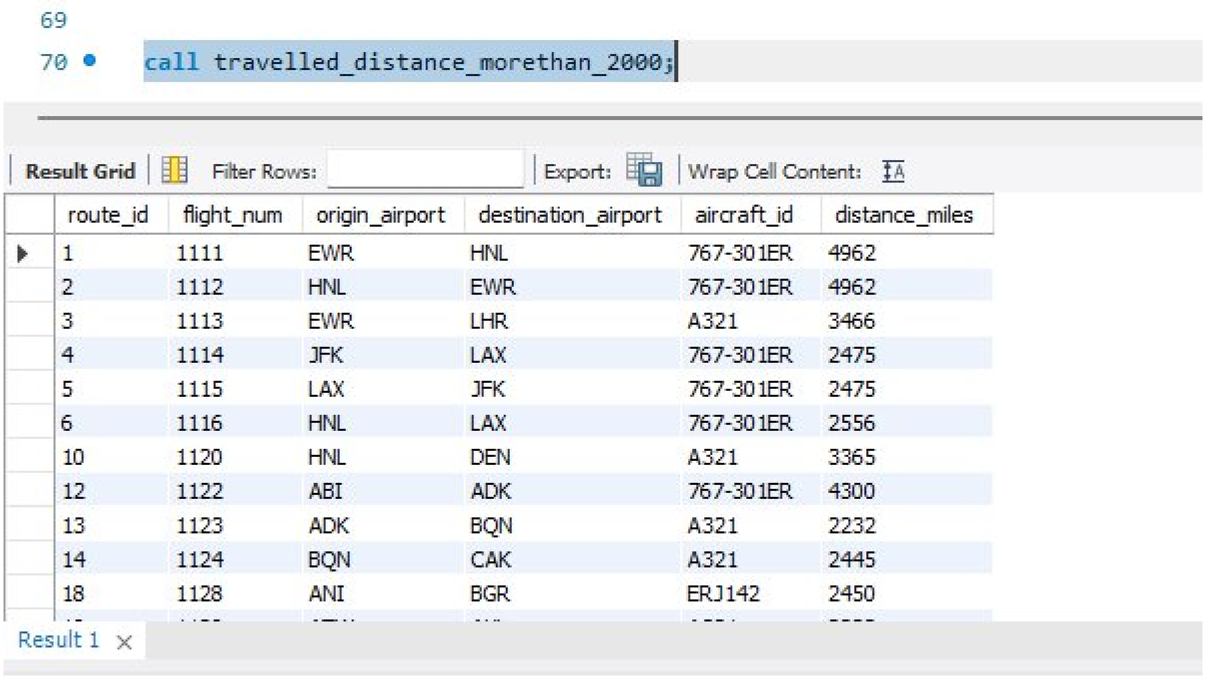
14. **Write a query to create a stored procedure that extracts all the details from the routes table where the travelled distance is more than 2000 miles.**



Name: travelled_distance_morethan_2000

DDL:

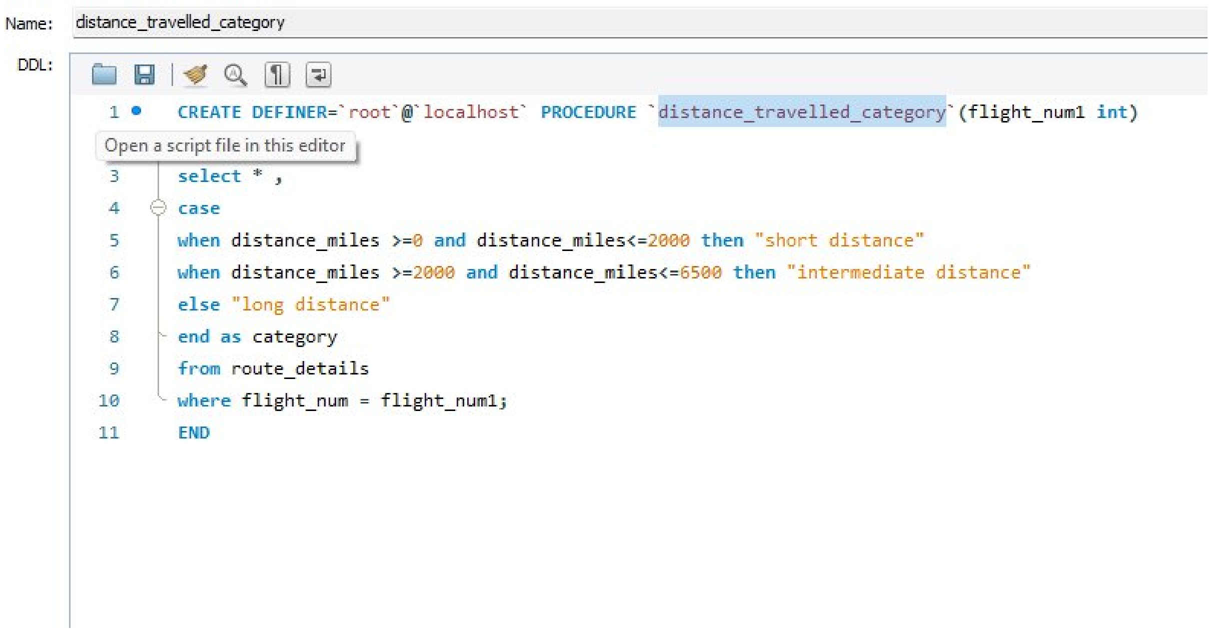
```
CREATE DEFINER=`root`@`localhost` PROCEDURE `travelled_distance_morethan_2000`()
BEGIN
select * from route_details where distance_miles>2000;
END
```



```
call travelled_distance_morethan_2000;
```

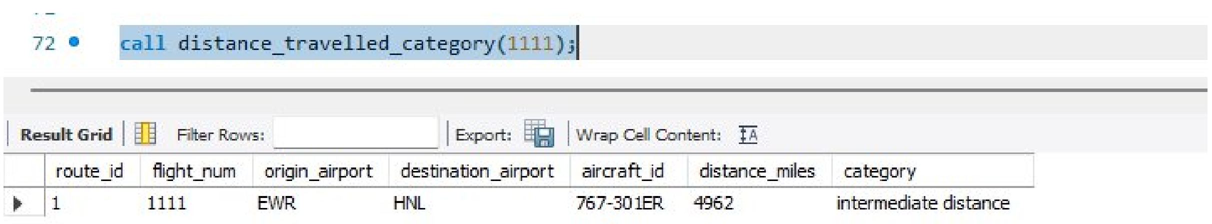
route_id	flight_num	origin_airport	destination_airport	aircraft_id	distance_miles
1	1111	EWR	HNL	767-301ER	4962
2	1112	HNL	EWR	767-301ER	4962
3	1113	EWR	LHR	A321	3466
4	1114	JFK	LAX	767-301ER	2475
5	1115	LAX	JFK	767-301ER	2475
6	1116	HNL	LAX	767-301ER	2556
10	1120	HNL	DEN	A321	3365
12	1122	ABI	ADK	767-301ER	4300
13	1123	ADK	BQN	A321	2232
14	1124	BQN	CAK	A321	2445
18	1128	ANI	BGR	ERJ142	2450

15. 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.


Name: distance_travelled_category
DDL:
Open a script file in this editor

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `distance_travelled_category`(flight_num1 int)

select * ,
case
when distance_miles >=0 and distance_miles<=2000 then "short distance"
when distance_miles >=2000 and distance_miles<=6500 then "intermediate distance"
else "long distance"
end as category
from route_details
where flight_num = flight_num1;
END
```



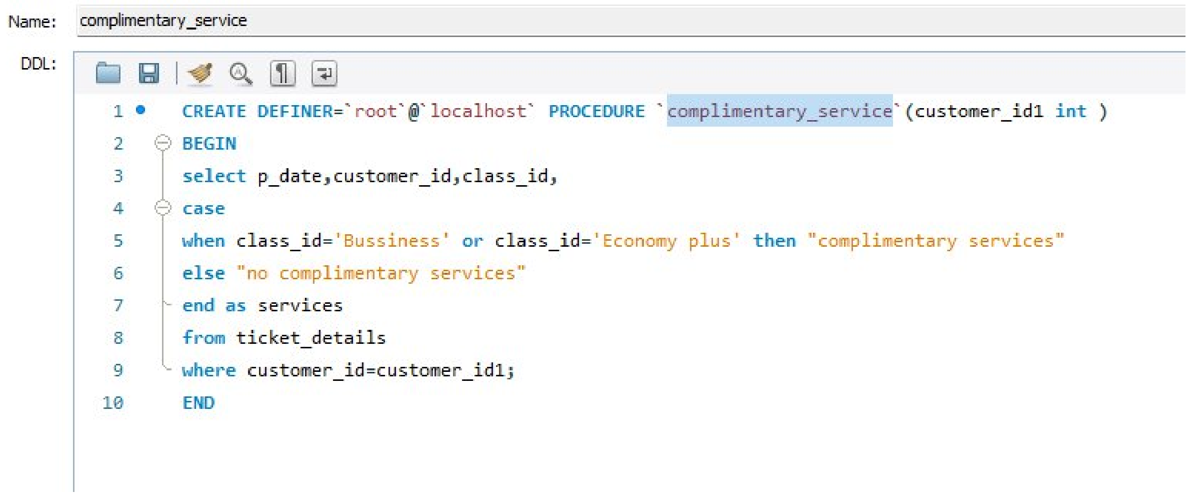
```
call distance_travelled_category(1111);
```

	route_id	flight_num	origin_airport	destination_airport	aircraft_id	distance_miles	category
▶	1	1111	EWR	HNL	767-301ER	4962	intermediate distance

16. 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:

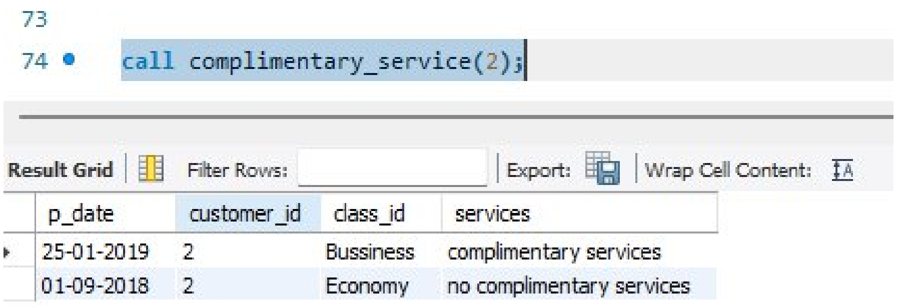
- If the class is *Business* and *Economy Plus*, then complimentary services are given as *Yes*, else it is *No*



Name: complimentary_service

DDL:

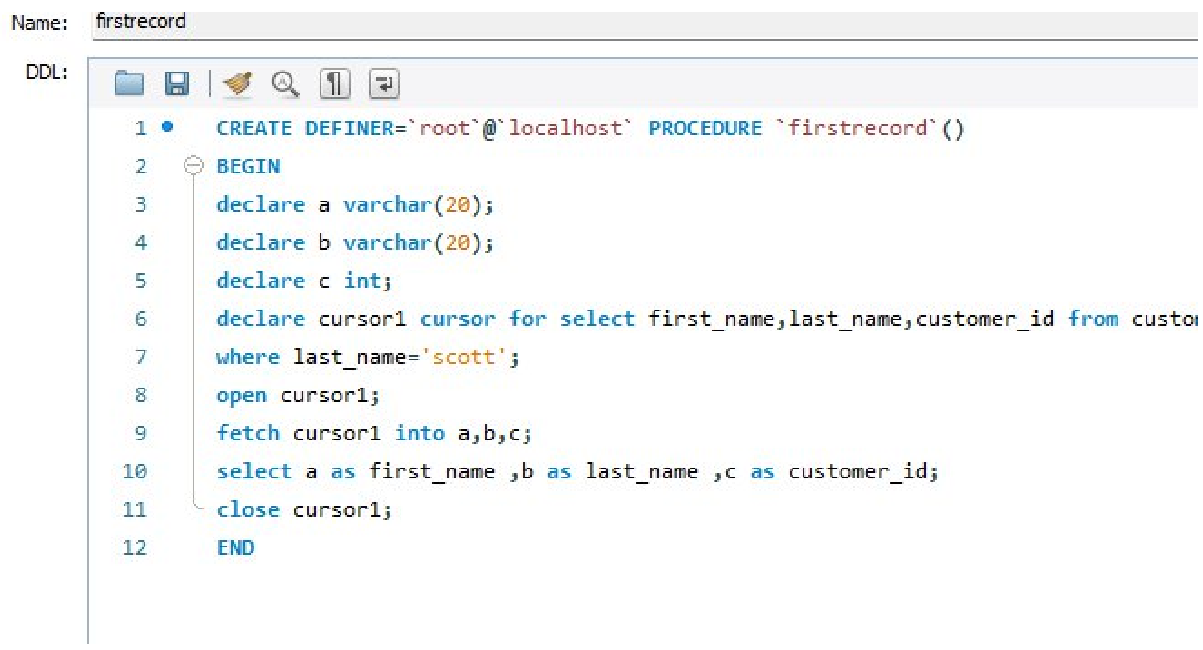
```sql
CREATE DEFINER=`root`@`localhost` PROCEDURE `complimentary_service`(customer_id1 int )
BEGIN
select p_date,customer_id,class_id,
case
when class_id='Bussiness' or class_id='Economy plus' then "complimentary services"
else "no complimentary services"
end as services
from ticket_details
where customer_id=customer_id1;
END
```



```sql
call complimentary_service(2);
```

p_date	customer_id	class_id	services
25-01-2019	2	Bussiness	complimentary services
01-09-2018	2	Economy	no complimentary services

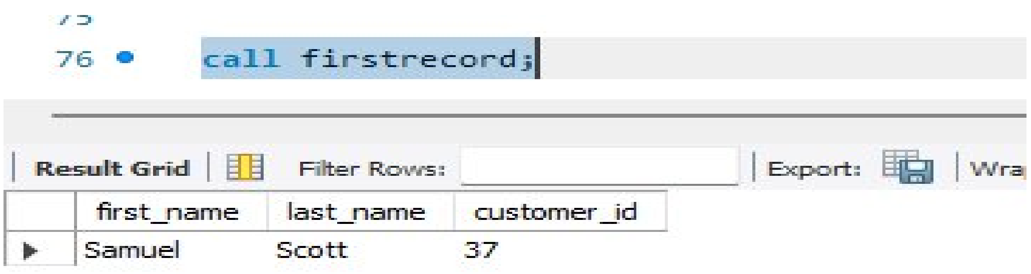
17. Write a query to extract the first record of the customer whose last name ends with Scott using a cursor from the customer table.



Name: firstrecord

DDL:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `firstrecord`()
BEGIN
declare a varchar(20);
declare b varchar(20);
declare c int;
declare cursor1 cursor for select first_name,last_name,customer_id from custo
where last_name='scott';
open cursor1;
fetch cursor1 into a,b,c;
select a as first_name ,b as last_name ,c as customer_id;
close cursor1;
END
```



```
call firstrecord;
```

```
first_name   last_name   customer_id
Samuel       Scott       37
```