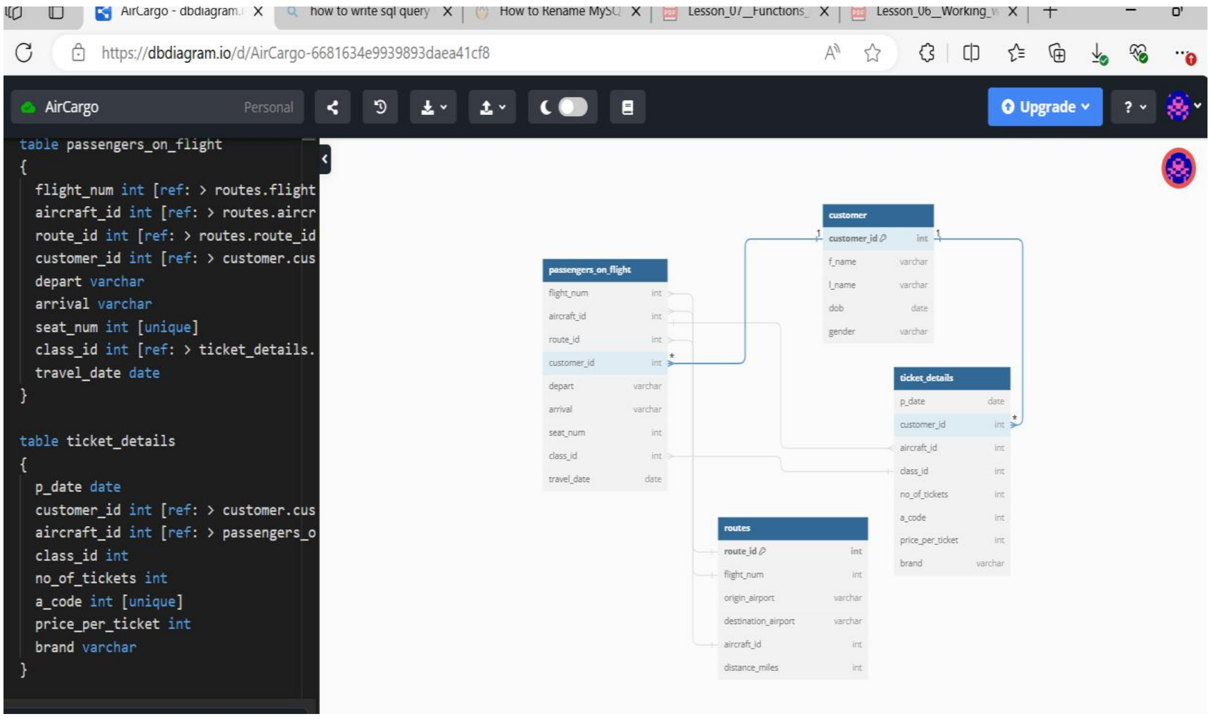
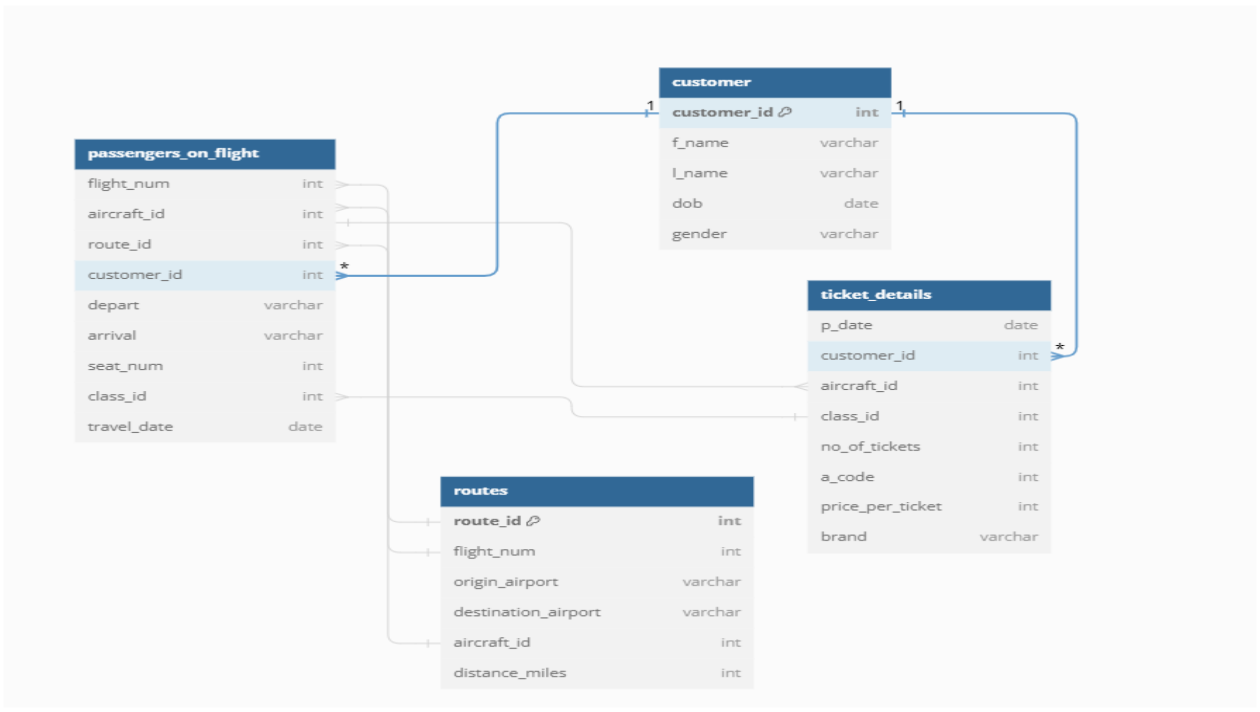
Air Cargo Analysis

Course-end Project 2 – Solution

1. Create an ER diagram for the given airlines database.



```
table passengers_on_flight
{
  flight_num int [ref: > routes.flight
  aircraft_id int [ref: > routes.aircr
  route_id int [ref: > routes.route_id
  customer_id int [ref: > customer.cus
  depart varchar
  arrival varchar
  seat_num int [unique]
  class_id int [ref: > ticket_details.
  travel_date date
}

table ticket_details
{
  p_date date
  customer_id int [ref: > customer.cus
  aircraft_id int [ref: > passengers_o
  class_id int
  no_of_tickets int
  a_code int [unique]
  price_per_ticket int
  brand varchar
}
```

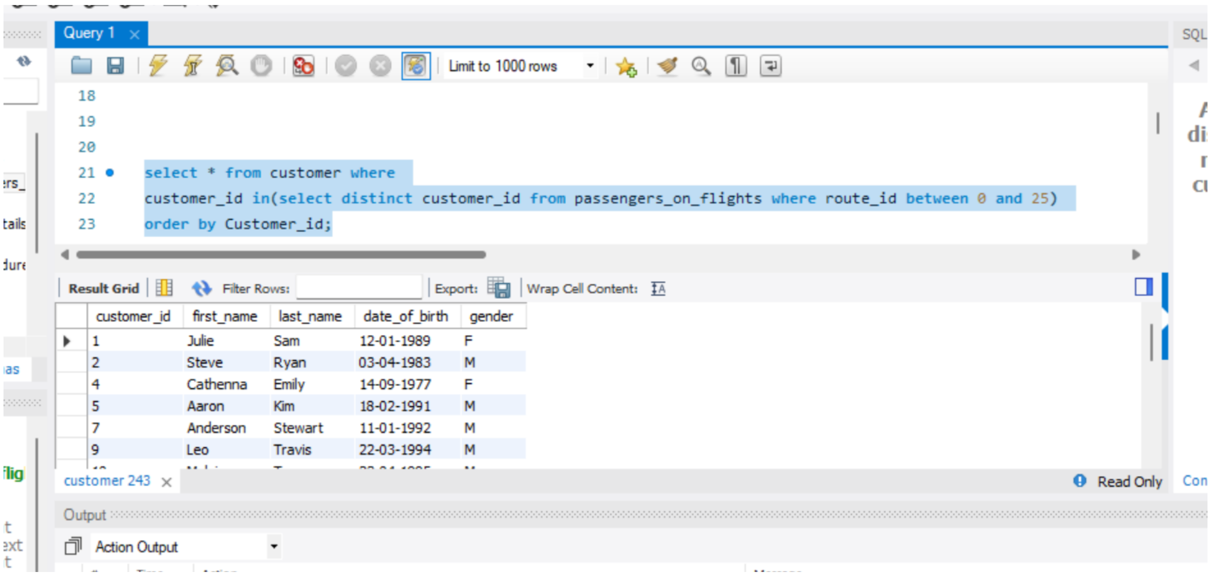


2. **Write a query to create route_details table using suitable data types for the fields, such as route_id, flight_num, origin_airport, destination_airport, aircraft_id, and distance_miles. Implement the check constraint for the flight number and unique constraint for the route_id fields. Also, make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

create table routes
(
route_id int NOT NULL UNIQUE,
flight_num int NOT NULL
origin_airport varchar(50)
destination_airport(50)
aircraft_id int
distance_miles int CHECK(miles >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.**

select * from customer where customer_id in(select distinct customer_id from passengers_on_flights where route_id between 0 and 25) order by Customer_id;

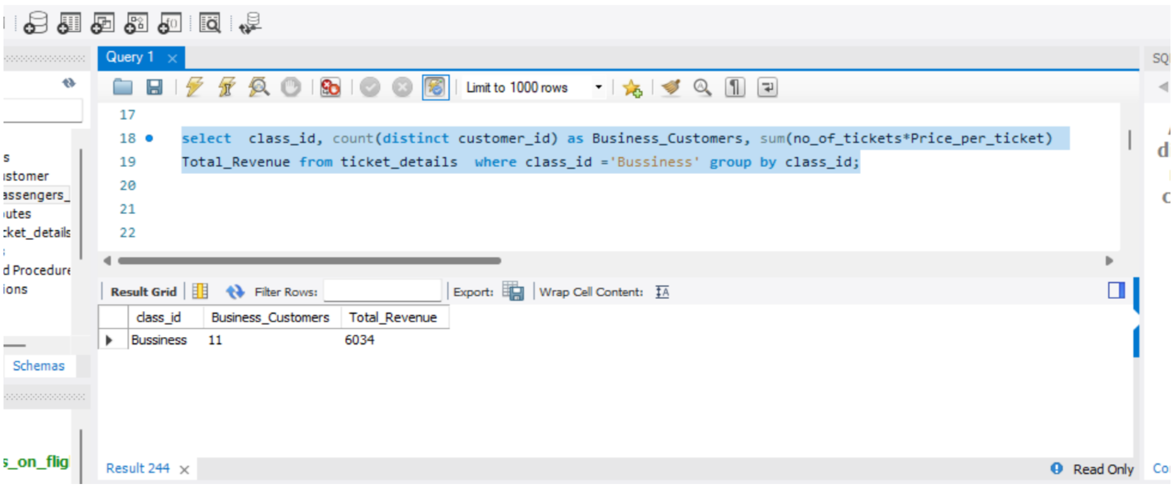


```
select * from customer where
customer_id in(select distinct customer_id from passengers_on_flights where route_id between 0 and 25)
order by Customer_id;
```

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
5	Aaron	Kim	18-02-1991	M
7	Anderson	Stewart	11-01-1992	M
9	Leo	Travis	22-03-1994	M

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

select class_id, count(distinct customer_id) as Business_Customers, sum(no_of_tickets*Price_per_ticket) Total_Revenue from ticket_details where class_id ='Bussiness' group by class_id;

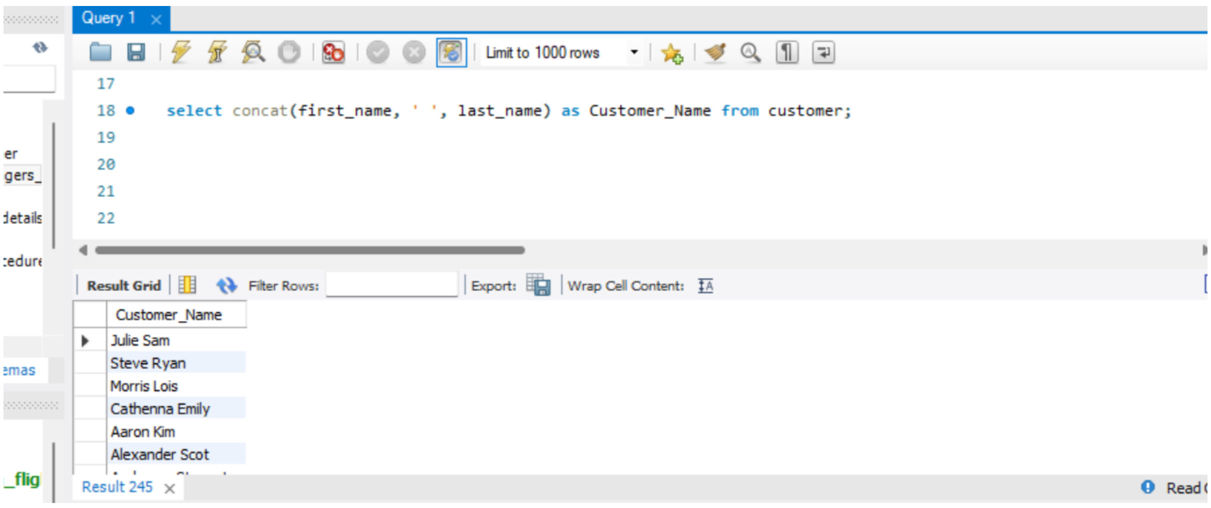


```
select  class_id, count(distinct customer_id) as Business_Customers, sum(no_of_tickets*Price_per_ticket)
Total_Revenue from ticket_details  where class_id ='Bussiness' group by class_id;
```

class_id	Business_Customers	Total_Revenue
Bussiness	11	6034

5. **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 Customer_Name from customer;

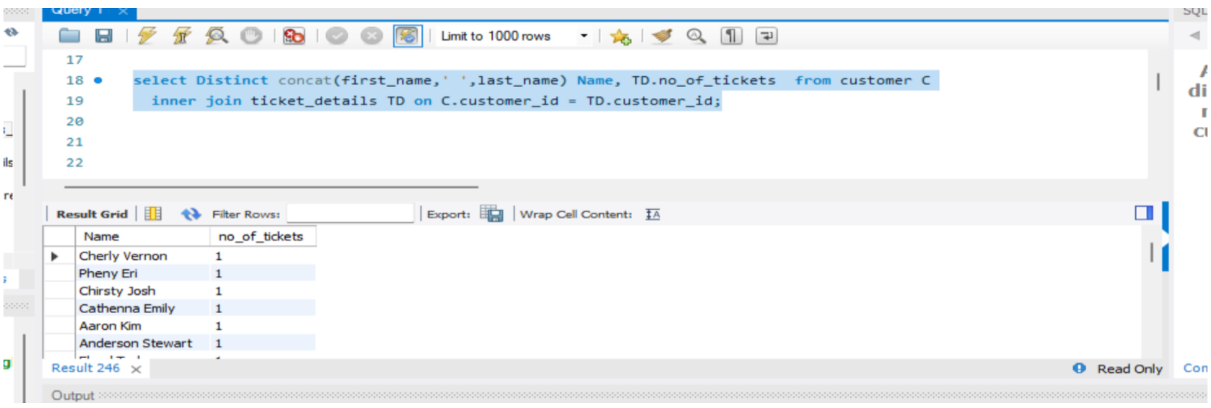


```
select concat(first_name, ' ', last_name) as Customer_Name from customer;
```

Customer_Name
Julie Sam
Steve Ryan
Morris Lois
Cathenna Emily
Aaron Kim
Alexander Scot

6. **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 concat(first_name,' ',last_name) Name, TD.no_of_tickets from customer C
  inner join ticket_details TD on C.customer_id = TD.customer_id;

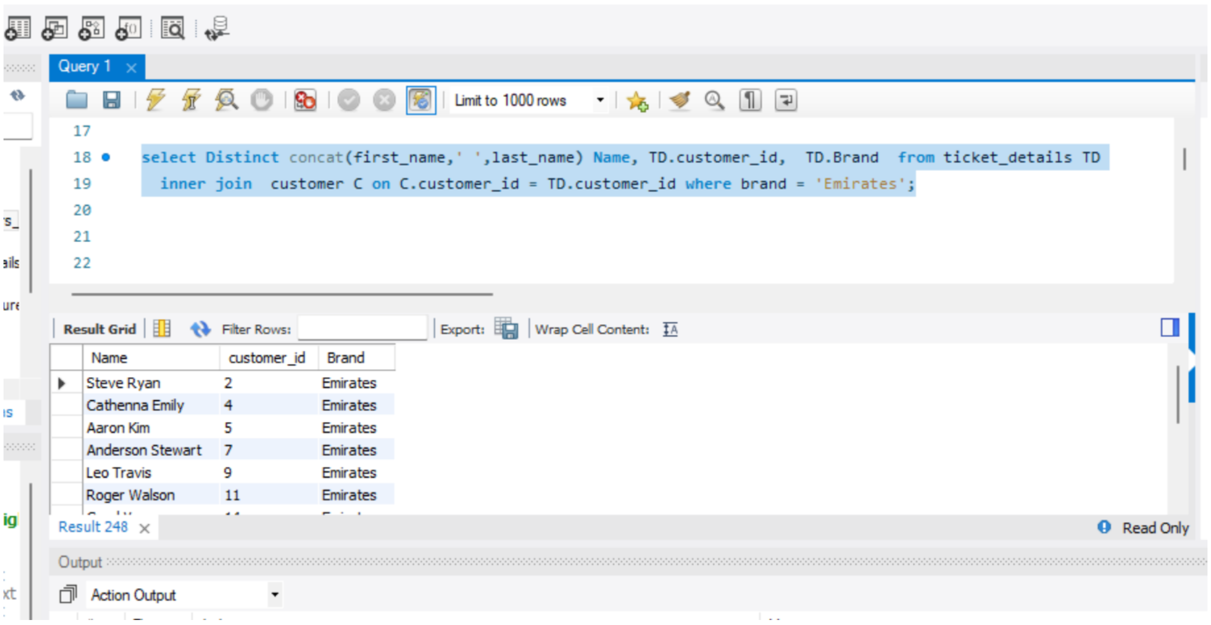


```
select Distinct concat(first_name,' ',last_name) Name, TD.no_of_tickets  from customer C
  inner join ticket_details TD on C.customer_id = TD.customer_id;
```

Name	no_of_tickets
Cherly Vernon	1
Pheny Eri	1
Chirsty Josh	1
Cathenna Emily	1
Aaron Kim	1
Anderson Stewart	1

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.**

select Distinct concat(first_name,' ',last_name) Name, TD.customer_id, TD.Brand from ticket_details TD
inner join customer C on C.customer_id = TD.customer_id where brand = 'Emirates';



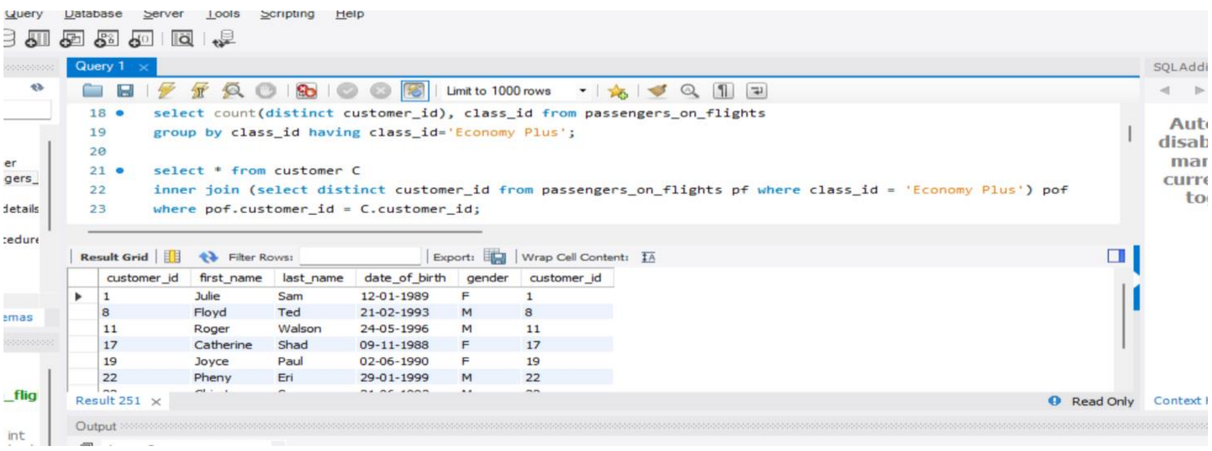
```
select Distinct concat(first_name,' ',last_name) Name, TD.customer_id,  TD.Brand  from ticket_details TD
  inner join  customer C on C.customer_id = TD.customer_id where brand = 'Emirates';
```

Name	customer_id	Brand
Steve Ryan	2	Emirates
Cathenna Emily	4	Emirates
Aaron Kim	5	Emirates
Anderson Stewart	7	Emirates
Leo Travis	9	Emirates
Roger Walson	11	Emirates

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.**

select count(distinct customer_id), class_id from passengers_on_flights group by class_id having class_id='Economy Plus';

select * from customer C
inner join (select distinct customer_id from passengers_on_flights pf where class_id = 'Economy Plus') pof
where pof.customer_id = C.customer_id;

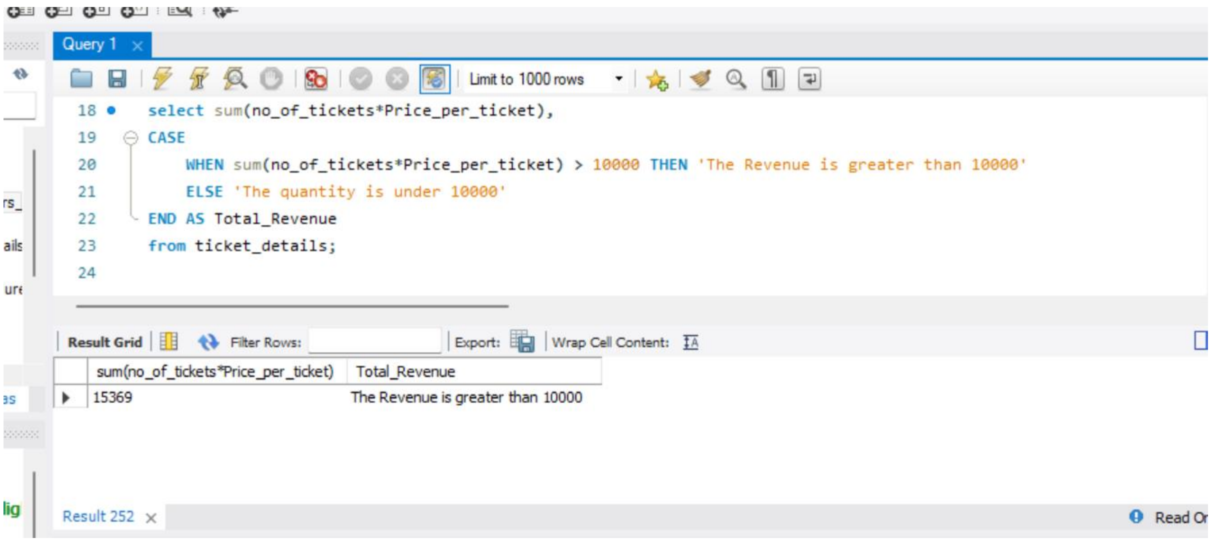


```
select count(distinct customer_id), class_id from passengers_on_flights
group by class_id having class_id='Economy Plus';

select * from customer C
inner join (select distinct customer_id from passengers_on_flights pf where class_id = 'Economy Plus') pof
where pof.customer_id = C.customer_id;
```

customer_id	first_name	last_name	date_of_birth	gender	customer_id
1	Julie	Sam	12-01-1989	F	1
8	Floyd	Ted	21-02-1993	M	8
11	Roger	Walson	24-05-1996	M	11
17	Catherine	Shad	09-11-1988	F	17
19	Joyce	Paul	02-06-1990	F	19
22	Pheny	Eri	29-01-1999	M	22

9. 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),
CASE
    WHEN sum(no_of_tickets*Price_per_ticket) > 10000 THEN 'The Revenue is greater than 10000'
    ELSE 'The quantity is under 10000'
END AS Total_Revenue
from ticket_details;



```
select sum(no_of_tickets*Price_per_ticket),
CASE
    WHEN sum(no_of_tickets*Price_per_ticket) > 10000 THEN 'The Revenue is greater than 10000'
    ELSE 'The quantity is under 10000'
END AS Total_Revenue
from ticket_details;
```

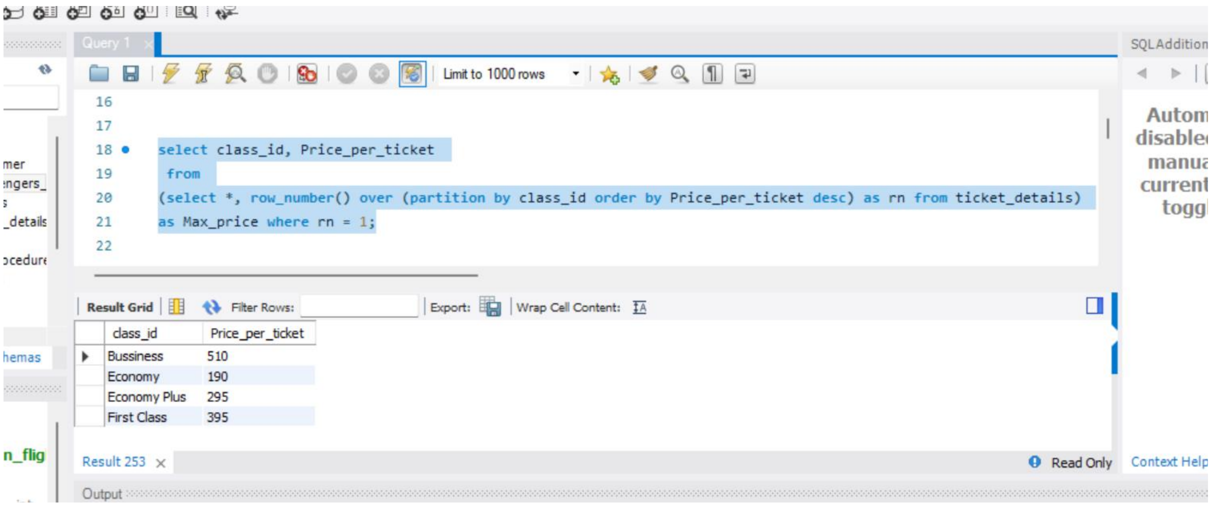
sum(no_of_tickets*Price_per_ticket)	Total_Revenue
15369	The Revenue is greater than 10000

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

create user `Sariya` identified by '123456';
grant select on aircargo.* to 'Sariya';
grant insert on aircargo.* to 'Sariya';
grant update on aircargo.* to 'Sariya';
grant delete on aircargo.* to 'Sariya';
grant execute on aircargo.* to 'Sariya';

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

select class_id, Price_per_ticket
 from
(select *, row_number() over (partition by class_id order by Price_per_ticket desc) as rn from ticket_details)
as Max_price where rn = 1;



```
select class_id, Price_per_ticket
 from
(select *, row_number() over (partition by class_id order by Price_per_ticket desc) as rn from ticket_details)
as Max_price where rn = 1;
```

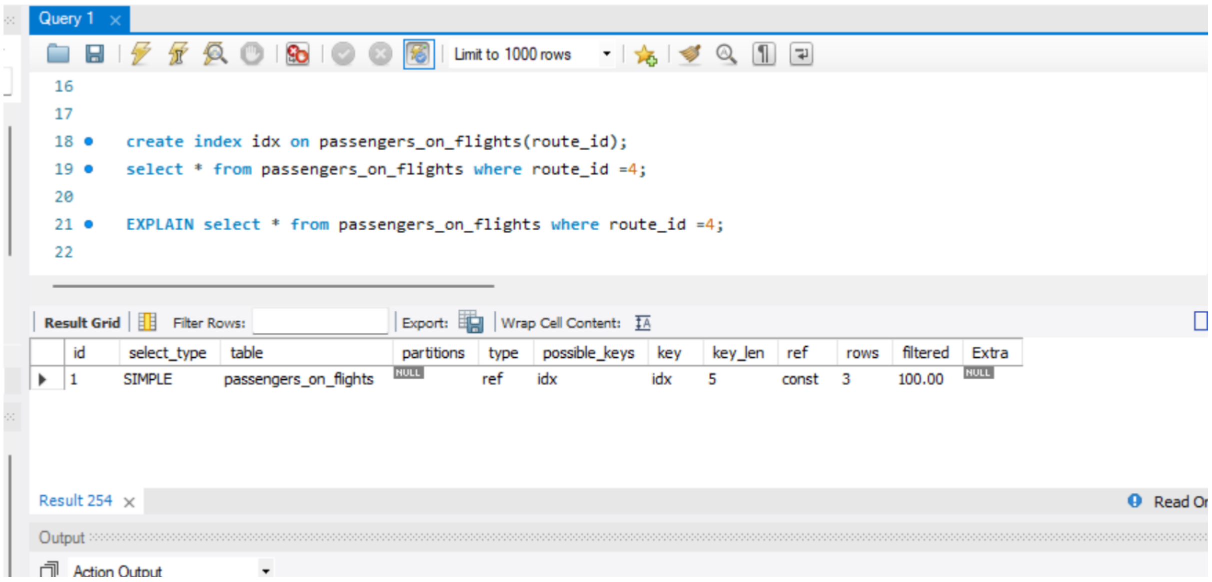
class_id	Price_per_ticket
Bussiness	510
Economy	190
Economy Plus	295
First Class	395

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.

create index idx on passengers_on_flights(route_id);
select * from passengers_on_flights where route_id =4;

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

EXPLAIN select * from passengers_on_flights where route_id =4;



```
create index idx on passengers_on_flights(route_id);
select * from passengers_on_flights where route_id =4;

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

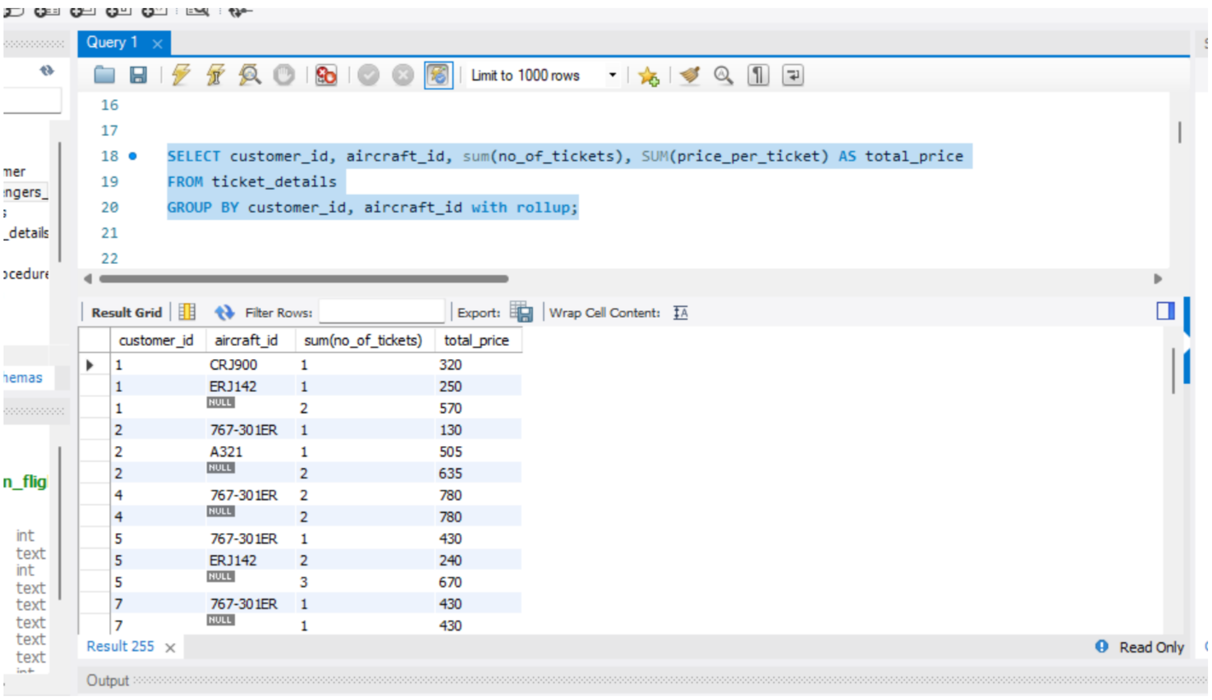
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	passengers_on_flights	NULL	ref	idx	idx	5	const	3	100.00	NULL

14. **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, aircraft_id, sum(no_of_tickets), SUM(price_per_ticket) AS total_price

FROM ticket_details

GROUP BY customer_id, aircraft_id with rollup;



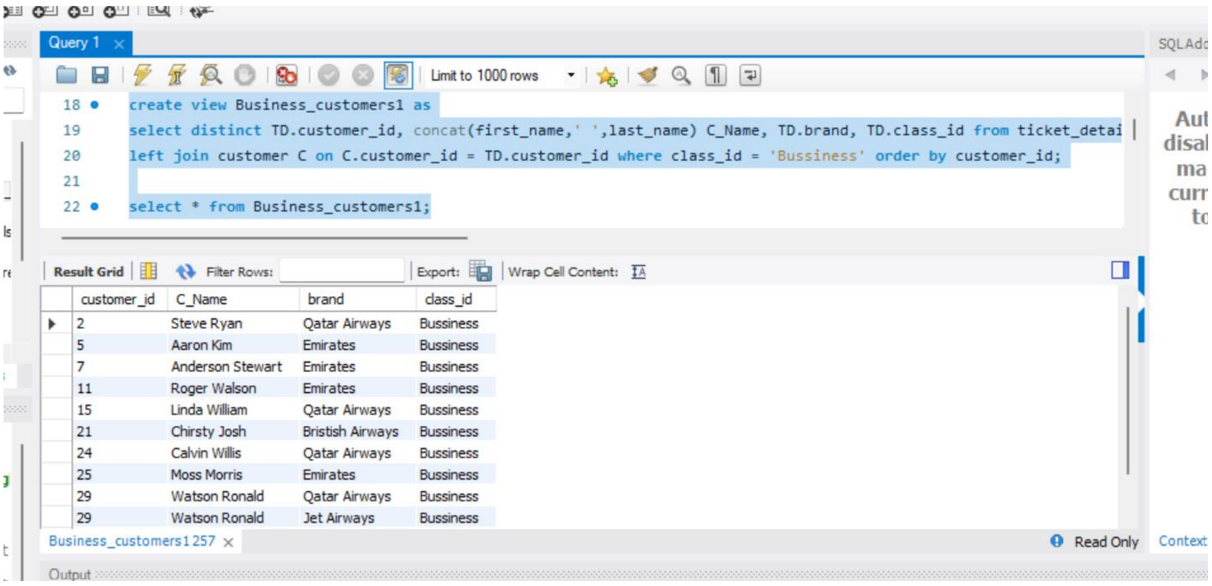
```
SELECT customer_id, aircraft_id, sum(no_of_tickets), SUM(price_per_ticket) AS total_price
FROM ticket_details
GROUP BY customer_id, aircraft_id with rollup;
```

customer_id	aircraft_id	sum(no_of_tickets)	total_price
1	CRJ900	1	320
1	ERJ142	1	250
1	NULL	2	570
2	767-301ER	1	130
2	A321	1	505
2	NULL	2	635
4	767-301ER	2	780
4	NULL	2	780
5	767-301ER	1	430
5	ERJ142	2	240
5	NULL	3	670
7	767-301ER	1	430
7	NULL	1	430

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

create view Business_customers as
select Distinct TD.customer_id, concat(first_name,' ',last_name) C_Name, TD.brand, TD.class_id from ticket_details TD
left join customer C on C.customer_id = TD.customer_id where class_id = 'Bussiness'
order by customer_id;

select * from Business_customers;



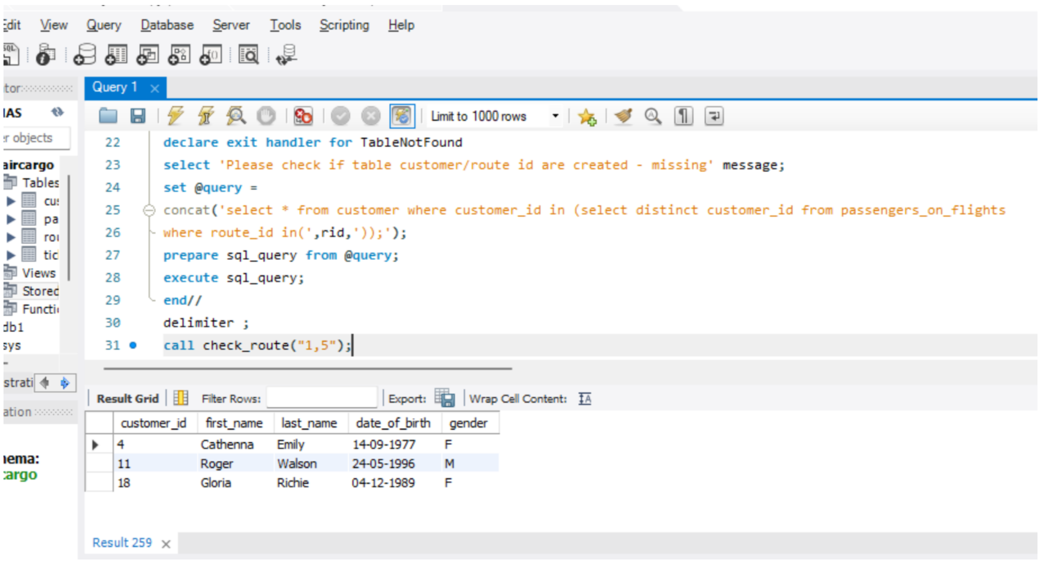
```
create view Business_customers1 as
select distinct TD.customer_id, concat(first_name,' ',last_name) C_Name, TD.brand, TD.class_id from ticket_detai
left join customer C on C.customer_id = TD.customer_id where class_id = 'Bussiness' order by customer_id;

select * from Business_customers1;
```

customer_id	C_Name	brand	class_id
2	Steve Ryan	Qatar Airways	Bussiness
5	Aaron Kim	Emirates	Bussiness
7	Anderson Stewart	Emirates	Bussiness
11	Roger Walson	Emirates	Bussiness
15	Linda William	Qatar Airways	Bussiness
21	Chirsty Josh	Bristish Airways	Bussiness
24	Calvin Willis	Qatar Airways	Bussiness
25	Moss Morris	Emirates	Bussiness
29	Watson Ronald	Qatar Airways	Bussiness
29	Watson Ronald	Jet Airways	Bussiness

Business_customers1 257

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.**

delimiter //
create procedure check_route(in rid varchar(255))
begin
declare TableNotFound condition for 1146;
declare exit handler for TableNotFound
select 'Please check if table customer/route id are created - missing' message;
set @query =
concat('select * from customer where customer_id in (select distinct customer_id from passengers_on_flights
where route_id in(',rid,'));');
prepare sql_query from @query;
execute sql_query;
end//
delimiter ;
call check_route("1,5");



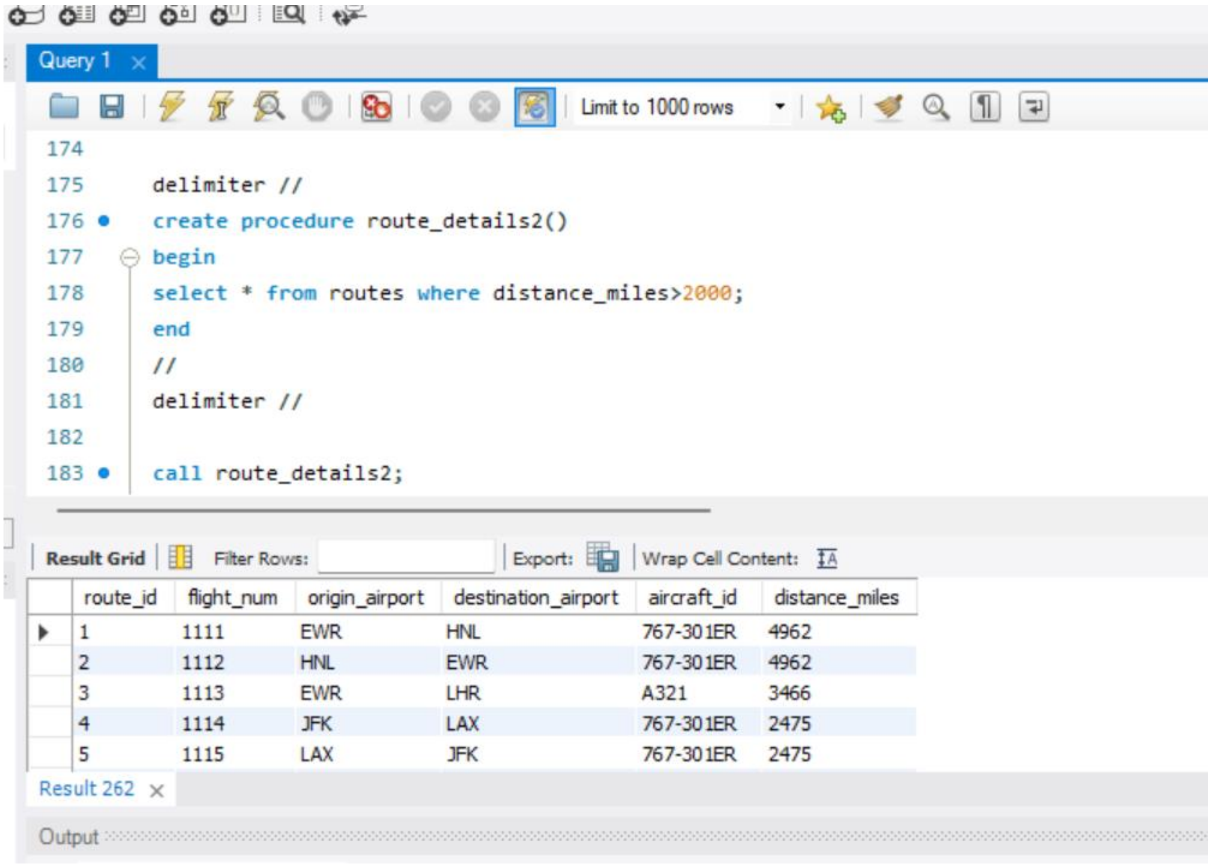
```
declare exit handler for TableNotFound
select 'Please check if table customer/route id are created - missing' message;
set @query =
concat('select * from customer where customer_id in (select distinct customer_id from passengers_on_flights
where route_id in(',rid,'));');
prepare sql_query from @query;
execute sql_query;
end//
delimiter ;
call check_route("1,5");
```

customer_id	first_name	last_name	date_of_birth	gender
4	Cathenna	Emily	14-09-1977	F
11	Roger	Walson	24-05-1996	M
18	Gloria	Richie	04-12-1989	F

Result 259

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.

```
delimiter //
create procedure route_details()
begin select * from routes where distance_miles>2000;
end
//
call route_details;
```

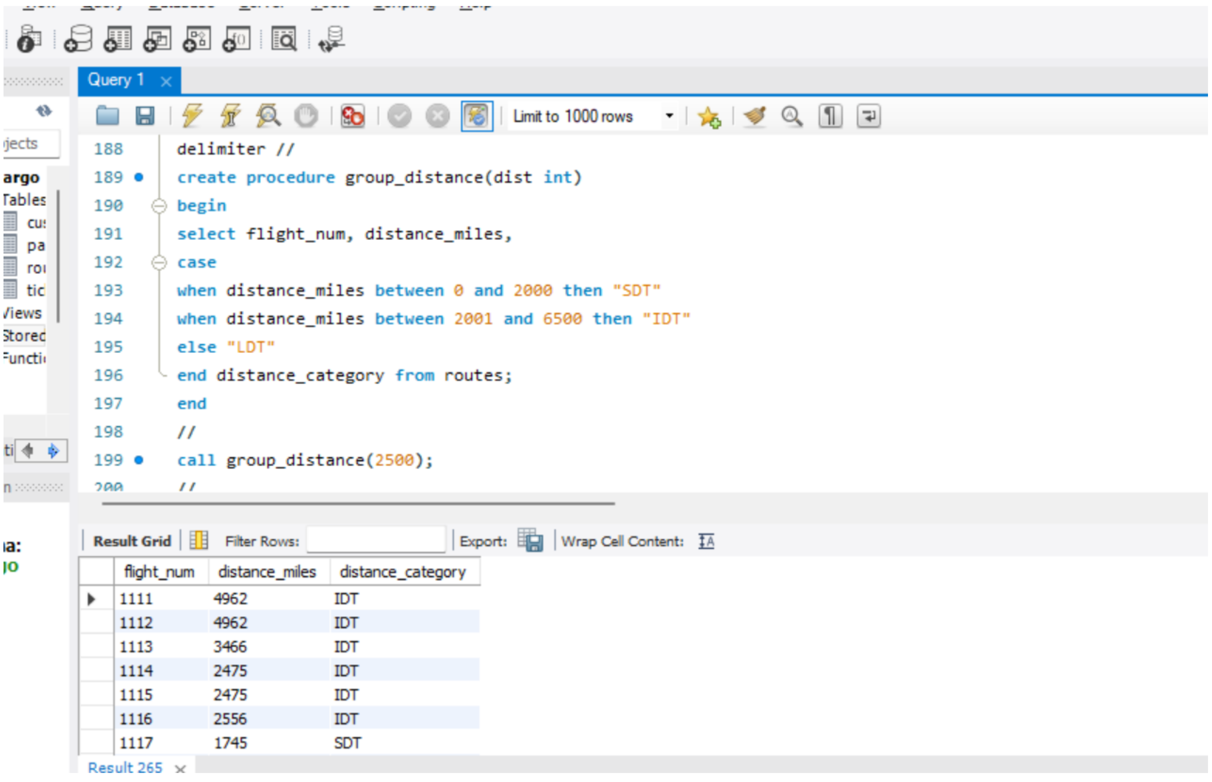


```
delimiter //
create procedure route_details2()
begin
select * from routes where distance_miles>2000;
end
//
delimiter //

call route_details2;
```

	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

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.

```
delimiter //
create procedure group_distance(dist int)
begin
select flight_num, distance_miles,
case
when distance_miles between 0 and 2000 then "SDT"
when distance_miles between 2001 and 6500 then "IDT"
else "LDT"
end distance_category from routes;
end
//
```



```
delimiter //
create procedure group_distance(dist int)
begin
select flight_num, distance_miles,
case
when distance_miles between 0 and 2000 then "SDT"
when distance_miles between 2001 and 6500 then "IDT"
else "LDT"
end distance_category from routes;
end
//
call group_distance(2500);
//
```

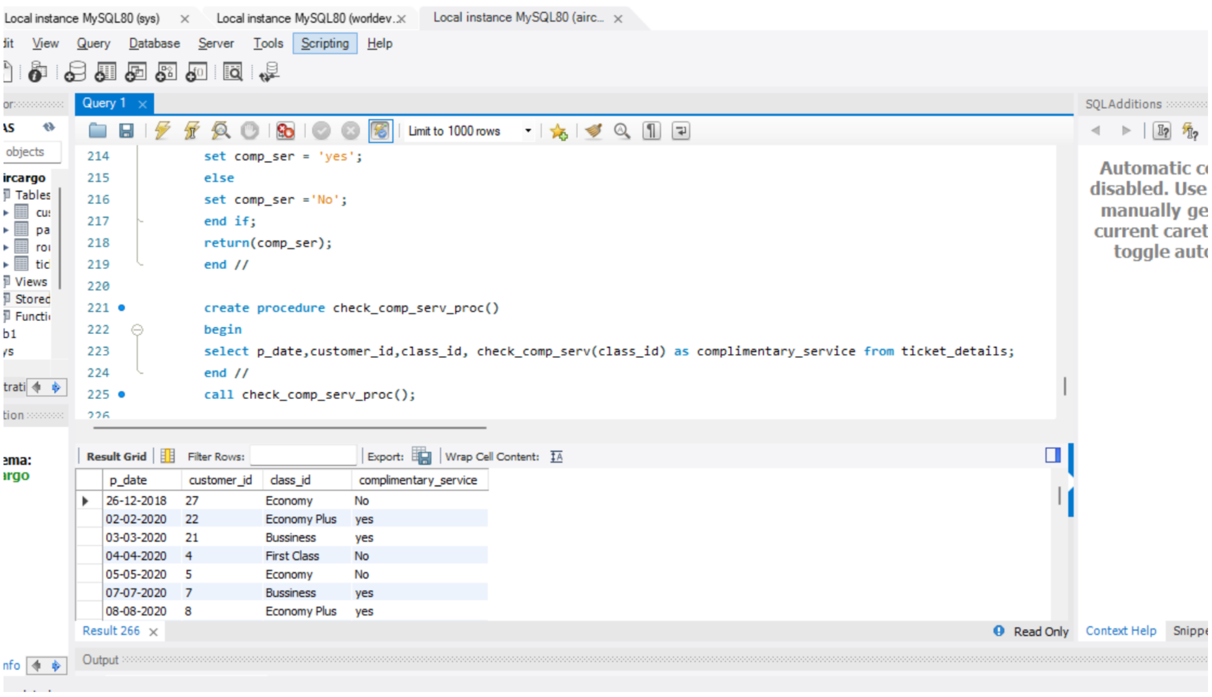
flight_num	distance_miles	distance_category
1111	4962	IDT
1112	4962	IDT
1113	3466	IDT
1114	2475	IDT
1115	2475	IDT
1116	2556	IDT
1117	1745	SDT

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*

```
delimiter//
        create function check_comp_serv( cls varchar(15))
    returns char(3)
    deterministic
    begin
    declare comp_ser char(3);
    if cls in ('Bussiness','Economy Plus') then
     set comp_ser = 'yes';
     else
     set comp_ser ='No';
     end if;
     return(comp_ser);
     end //

     create procedure check_comp_serv_proc()
     begin
     select p_date,customer_id,class_id, check_comp_serv(class_id) as
complimentary_service from ticket_details;
     end //
     call check_comp_serv_proc();
```

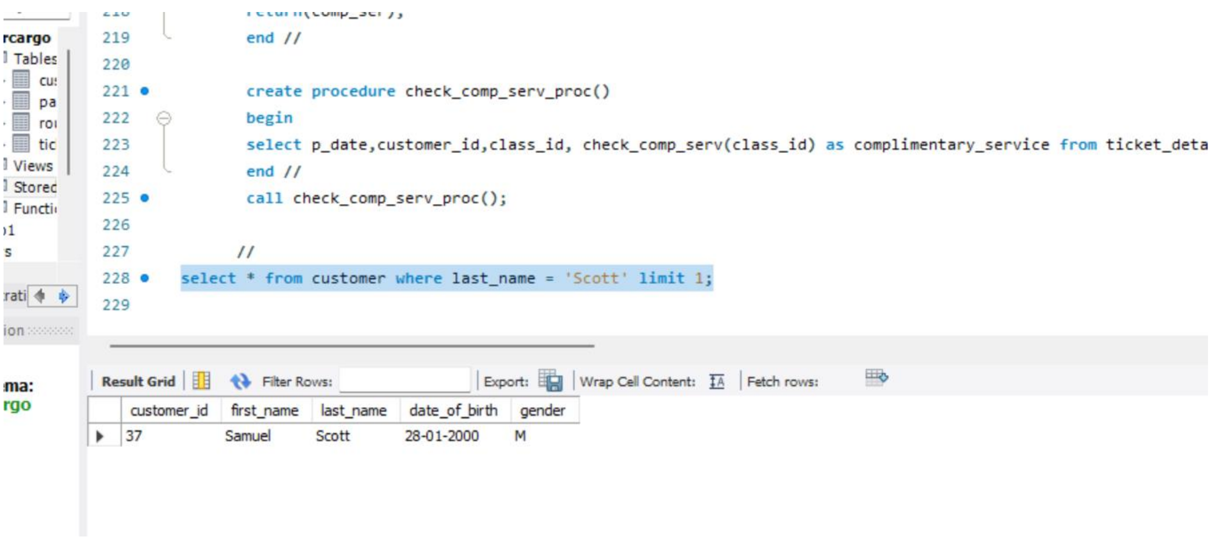

Local instance MySQL80 (sys)
Local instance MySQL80 (worldev...
Local instance MySQL80 (airc...
View Query Database Server Tools Scripting Help
Query 1
Limit to 1000 rows
SQLAdditions
Automatic c disabled. Use manually ge current caret toggle auto
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Result 266
Read Only
Context Help
Output

```
        set comp_ser = 'yes';
        else
        set comp_ser ='No';
        end if;
        return(comp_ser);
        end //

        create procedure check_comp_serv_proc()
        begin
        select p_date,customer_id,class_id, check_comp_serv(class_id) as complimentary_service from ticket_details;
        end //
        call check_comp_serv_proc();
```

p_date	customer_id	class_id	complimentary_service
26-12-2018	27	Economy	No
02-02-2020	22	Economy Plus	yes
03-03-2020	21	Bussiness	yes
04-04-2020	4	First Class	No
05-05-2020	5	Economy	No
07-07-2020	7	Bussiness	yes
08-08-2020	8	Economy Plus	yes

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.

select * from customer where last_name = 'Scott' limit 1;



```
        end //

        create procedure check_comp_serv_proc()
        begin
        select p_date,customer_id,class_id, check_comp_serv(class_id) as complimentary_service from ticket_deta
        end //
        call check_comp_serv_proc();

       //
select * from customer where last_name = 'Scott' limit 1;
```

customer_id	first_name	last_name	date_of_birth	gender
37	Samuel	Scott	28-01-2000	M

-Done by
Ulma Sariya