**SQL**

Course-End Project Problem Statement

****

**Air Cargo Analysis.**

Course-end Project 2

**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.

**Project Objective:**

You, as a DBA expert, need to focus on identifying the regular customers to provide offers, analyze the busiest route which helps to increase the number of aircraft required and prepare an analysis to determine the ticket sales details. This will ensure that the company improves its operability and becomes more customer-centric and a favorable choice for air travel.

**Dataset description:**

**Customer:** Contains the information of customers

* customer\_id – ID of the customer
* first\_name – First name of the customer
* last\_name – Last name of the customer
* date\_of\_birth – Date of birth of the customer
* gender – Gender of the customer

**passengers\_on\_flights:** Contains information about the travel details

* aircraft\_id – ID of each aircraft in a brand
* route\_id – Route ID of from and to location
* customer\_id – ID of the customer
* depart – Departure place from the airport
* arrival – Arrival place in the airport
* seat\_num – Unique seat number for each passenger
* class\_id – ID of travel class
* travel\_date – Travel date of each passenger
* flight\_num – Specific flight number for each route

**ticket\_details:** Contains information about the ticket details

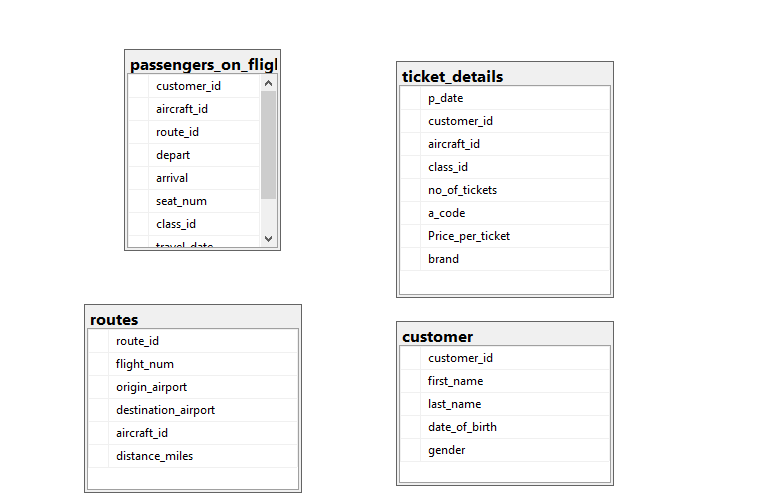
* p\_date – Ticket purchase date
* customer\_id – ID of the customer
* aircraft\_id – ID of each aircraft in a brand
* class\_id – ID of travel class
* no\_of\_tickets – Number of tickets purchased
* a\_code – Code of each airport
* price\_per\_ticket – Price of a ticket
* brand – Aviation service provider for each aircraft

**routes:** Contains information about the route details

* Route\_id – Route ID of from and to location
* Flight\_num – Specific fight number for each route
* Origin\_airport – Departure location
* Destination\_airport – Arrival location
* Aircraft\_id – ID of each aircraft in a brand
* Distance\_miles – Distance between departure and arrival location

**Solution**

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



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 UNIQUE,

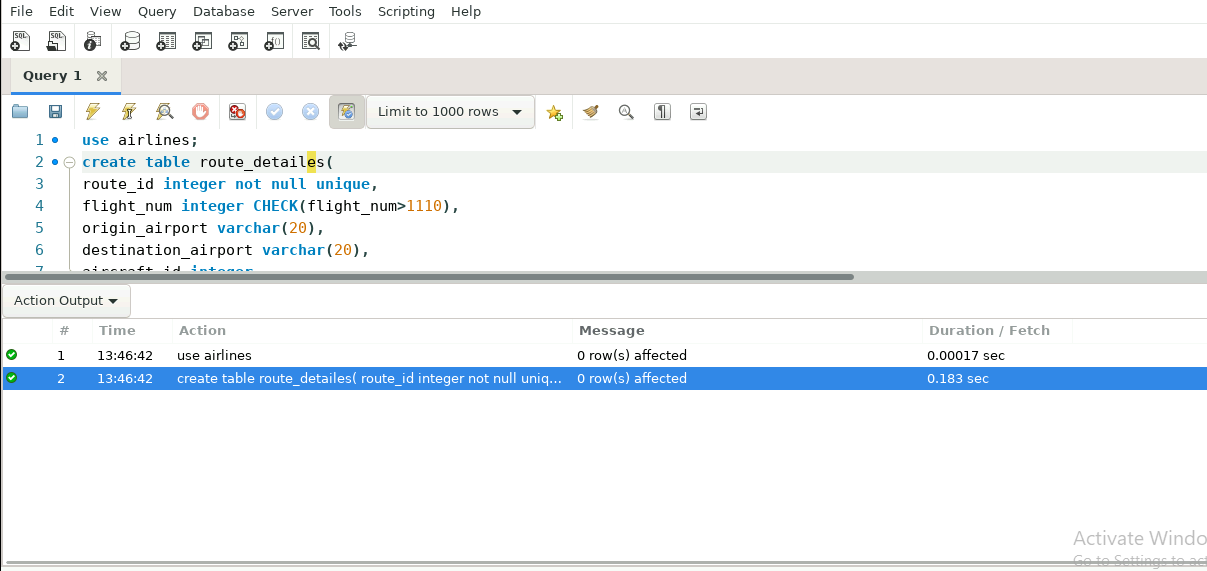
flight\_num int CHECK(flight\_num>1110),

origin\_airport varchar(20),

destination\_airport vrchar(20),

aircraft\_id varchar(30),

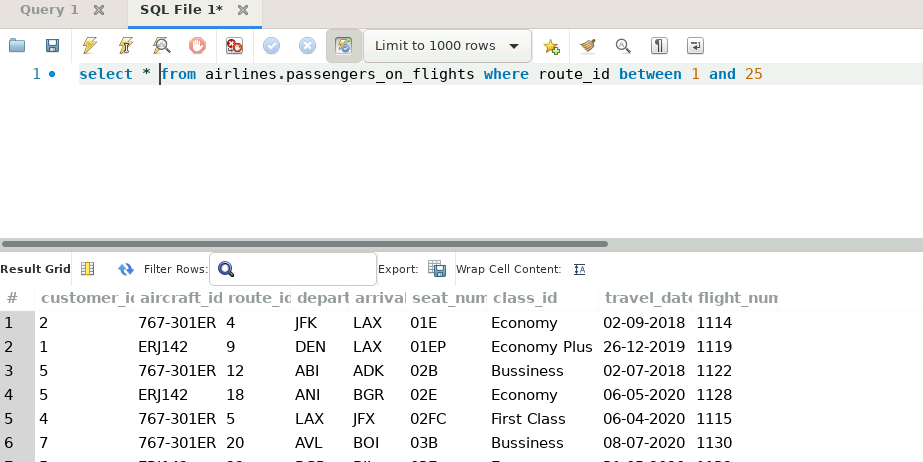
distance\_miles int CHECK(distance\_miles>0);)



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.

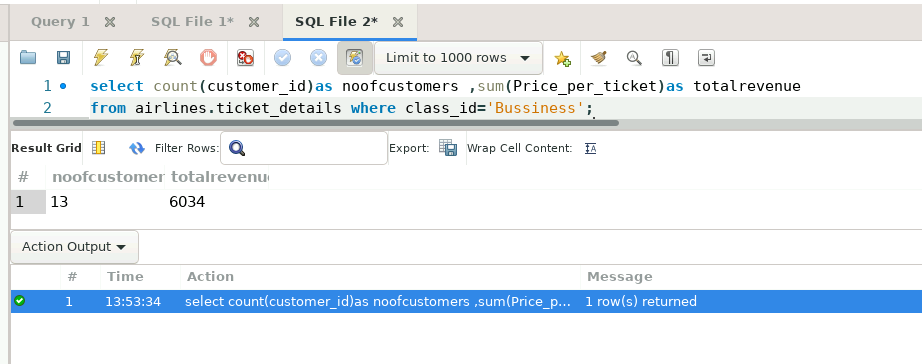
SELECT \* FROM AIRLINES.DBO.passengers\_on\_flights

WHERE ROUTE\_ID BETWEEN 1 AND 25



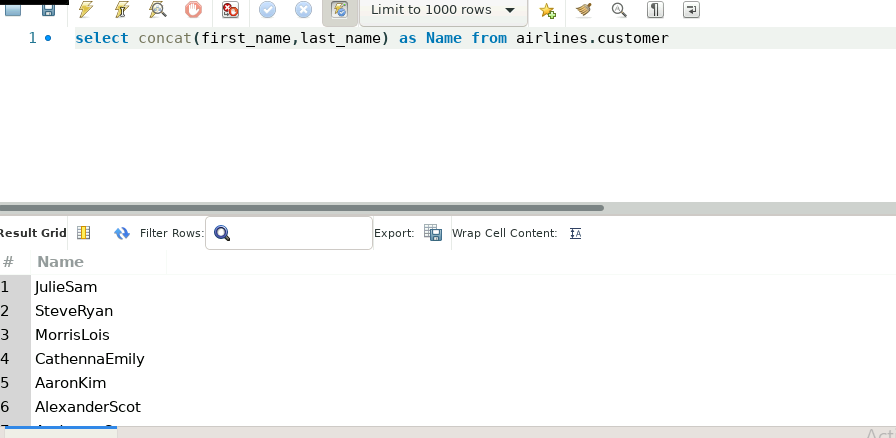
4.Write a query to identify the number of passengers and total revenue in business class from the ticket\_details table.

select count(customer\_id)as noofcustomers,sum(Price\_per\_ticket) as totalrevenue from airlines.dbo.ticket\_details where class\_id='Bussiness'



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 first\_name+last\_name as name from airlines.dbo.customer



Like this we have 50 customers.

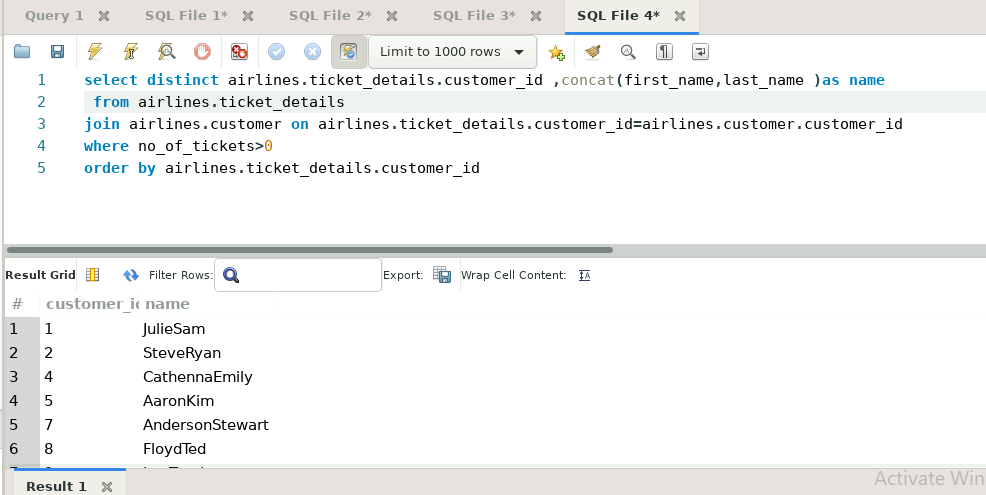
6.Write a query to extract the customers who have registered and booked a ticket. Use data from the customer and ticket\_details tables.

Select distinct airlines.dbo.ticket\_details.customer\_id ,first\_name+last\_name as name from airlines.dbo.ticket\_details

join airlines.dbo.customer on airlines.dbo.ticket\_details.customer\_id=airlines.dbo.customer.customer\_id

where no\_of\_tickets>0

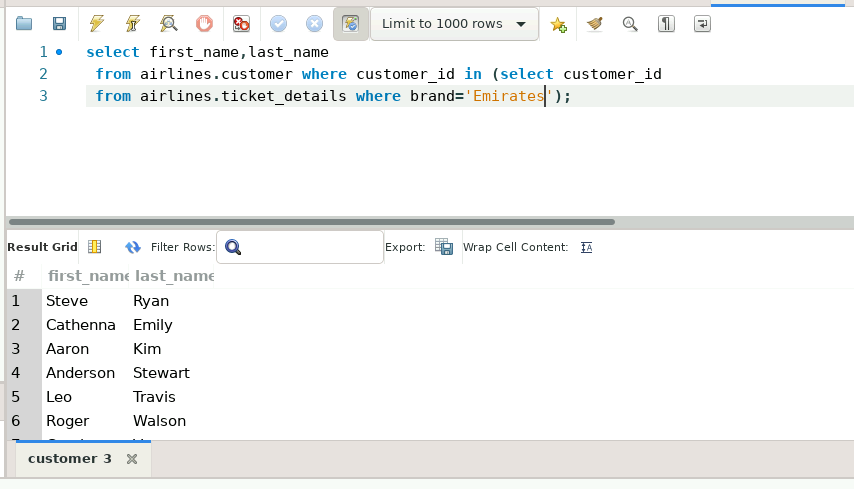
order by airlines.dbo.ticket\_details.customer\_id



Like this we have all 50 customers booked a ticket.

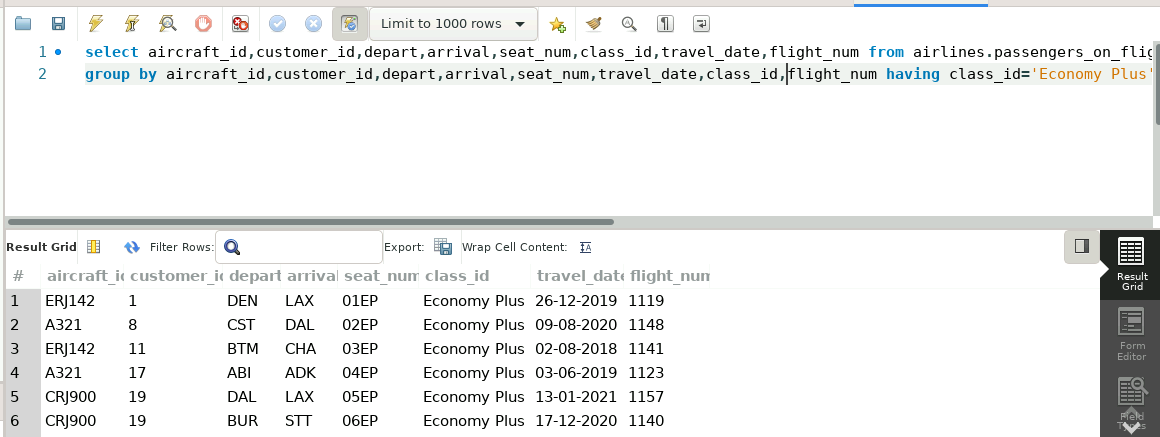
7. Write a query to identify the customer’s first name and last name based on their customer ID and brand (Emirates) from the ticket\_details table.

select first\_name,last\_name from airlines.dbo.customer where customer\_id in (select customer\_id from airlines.dbo.ticket\_details where brand='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 aircraft\_id ,customer\_id,depart, arrival,seat\_num ,class\_id ,travel\_date ,flight\_num from airlines.dbo.passengers\_on\_flights group by aircraft\_id ,customer\_id,depart, arrival,seat\_num ,class\_id ,travel\_date ,flight\_num having class\_id='Economy Plus' ;



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

ticket\_details table.

declare @revenue int;

select @revenue=sum(price\_per\_ticket) from airlines.dbo.ticket\_details

If @revenue>10000

select 'the revenue crossed 10000'

Else

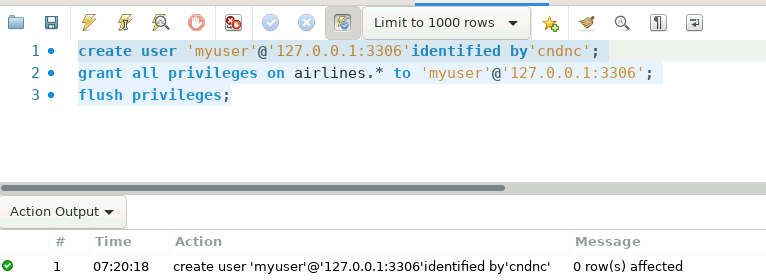
select'the revenue not crossed 10000'

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

create user 'myuser'@'127.0.0.1:3306'identified by'cndnc';

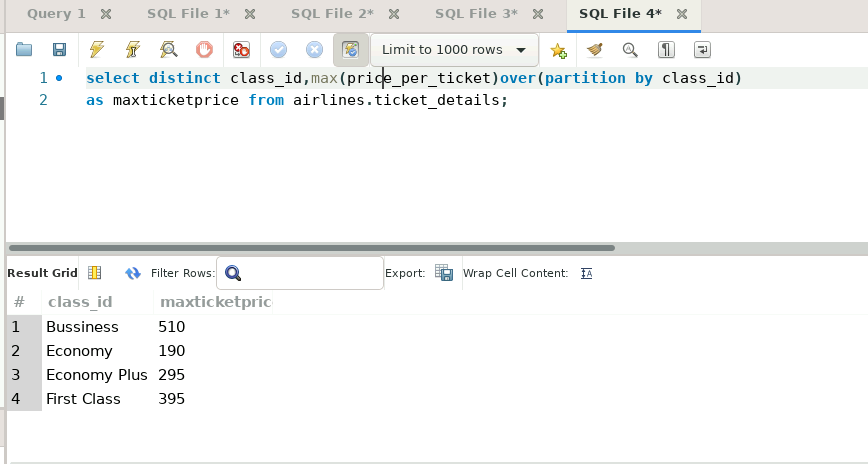
grant all privileges on airlines.\* to 'myuser'@'127.0.0.1:3306';

flush privileges;



11.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 MAXTICKETPRICE from airlines.dbo.ticket\_details ;

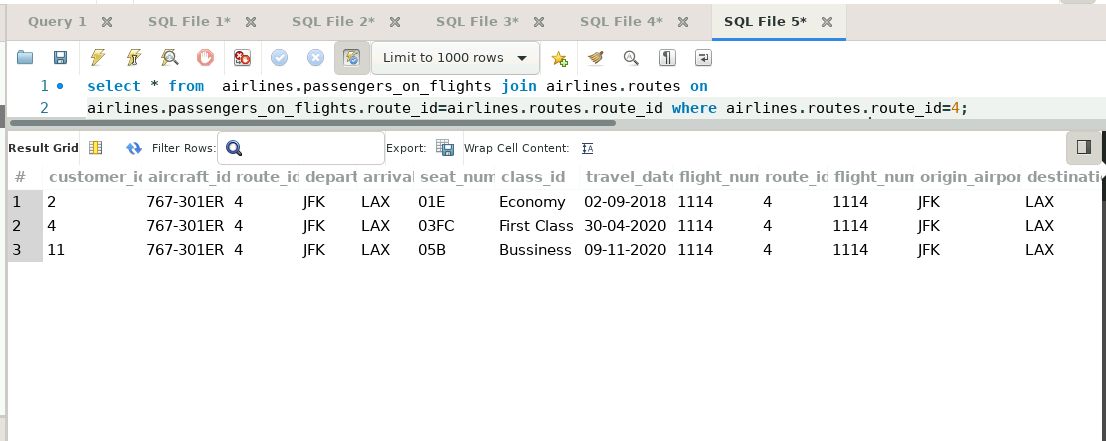


12. Write a query to extract the passengers whose route ID is 4 by improving the speed and performance of the passengers\_on\_flights table.

Select \* from airlines.dbo.passengers\_on\_flights join

airlines.dbo.routes on airlines.dbo.routes .route\_id= airlines.dbo.passengers\_on\_flights .route\_id

where airlines.dbo.routes.route\_id=4;



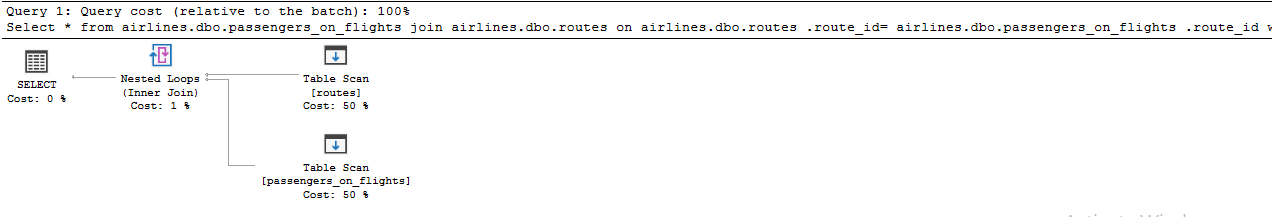
13. For the route ID 4, write a query to view the execution plan of the passengers\_on\_flights table.

Select \* from airlines.dbo.passengers\_on\_flights join

airlines.dbo.routes on airlines.dbo.routes .route\_id= airlines.dbo.passengers\_on\_flights .route\_id

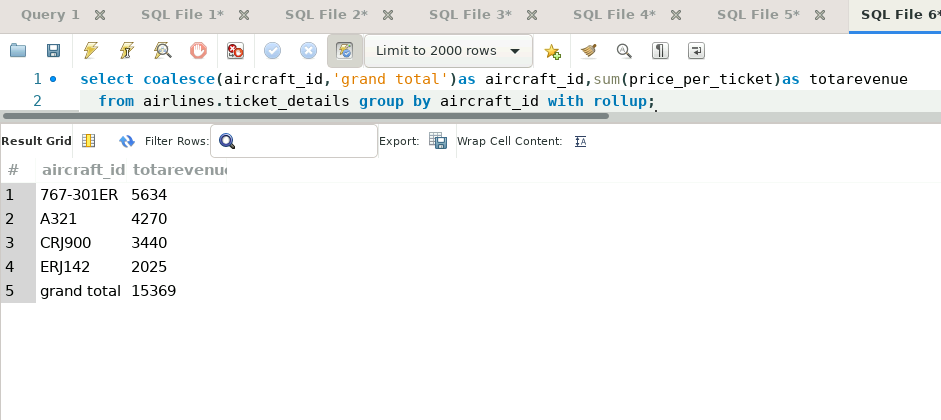
where airlines.dbo.routes.route\_id=4

to see execution plan select display estimated execution plan option to visual explain instead of tabular explain with in the query results tab



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

select coalesce(aircraft\_id,'GRAND TOTAL')as aircraft\_id,sum(price\_per\_ticket)as totalrevenueperaircraft from airlines.dbo.ticket\_details group by aircraft\_id with rollup;



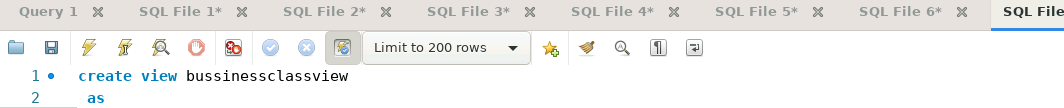
15.Write a query to create a view with only business class customers along with the brand of airlines.

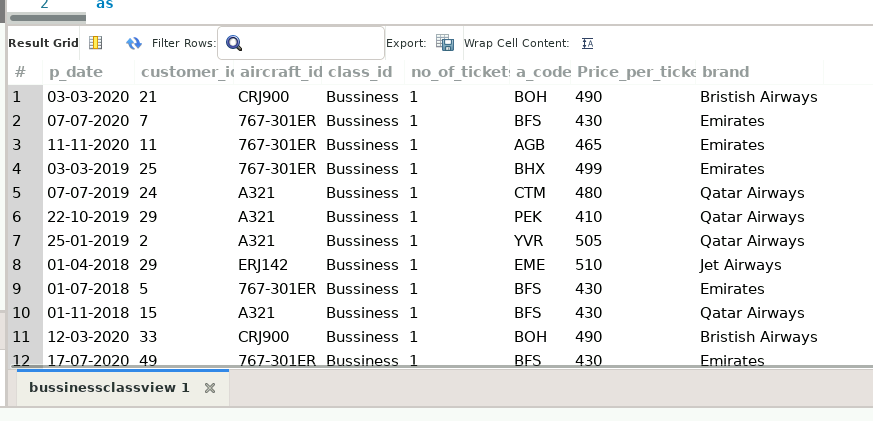
create view bussinessclassview as

select \* from airlines.dbo.ticket\_details

where class\_id='Bussiness'

select \* from bussinessclassview





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 passengerroutes1(in rid1 int,in rid2 int)**

**begin**

**declare table\_exists int default 0;**

**select count(\*) into table\_exists from information\_schema.tables where (table\_schema = database() )AND (table\_name = 'Passengers\_on\_flights');**

**if table\_exists =0 then**

**select 'Error:table passengeres\_on\_flights table doesnot exists 'as message;**

**else**

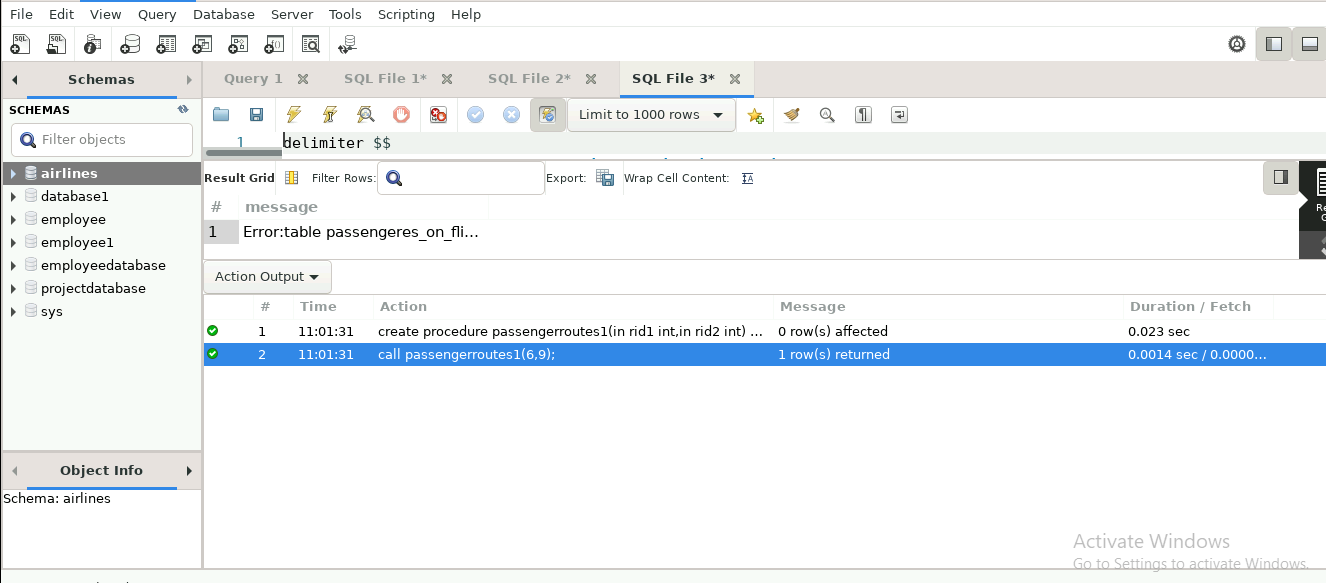
**select c.first\_name,c.last\_name from customer c join passengers\_on\_flights pof on c.customer\_id=pof.customer\_id**

**join route\_detailes rd on rd.route\_id=pof.route\_id where rd.origin\_airport=rid1 and rd.destination\_airport=rid2 ;**

**end if;**

**end $$**

**call passengerroutes1(6,9);**



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.

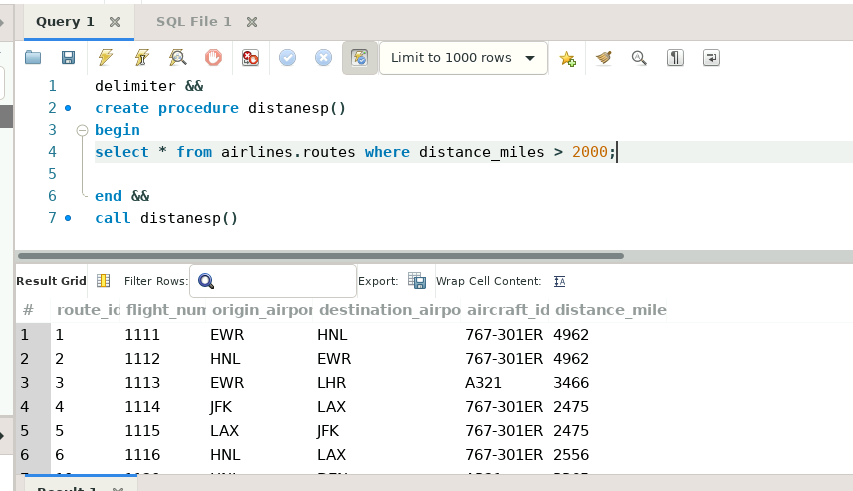
Create procedure distance()

As

Begin

Select \* from airlines.routes where distance\_miles>2000

End



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 function distanceSETTING( distance\_miles int)

returns varchar(50)

deterministic

begin

declare distancegroup varchar(50);

if distance\_miles between 0 and 2000 then

set distancegroup='short distance travel';

elseif distance\_miles between 2000 and 6500 then

set distancegroup='intermediate distance travel';

elseif distance\_miles > 6500

then

set distancegroup='long distance travel';

END IF;

RETURN(distancegroup);

END &&

delimiter &&

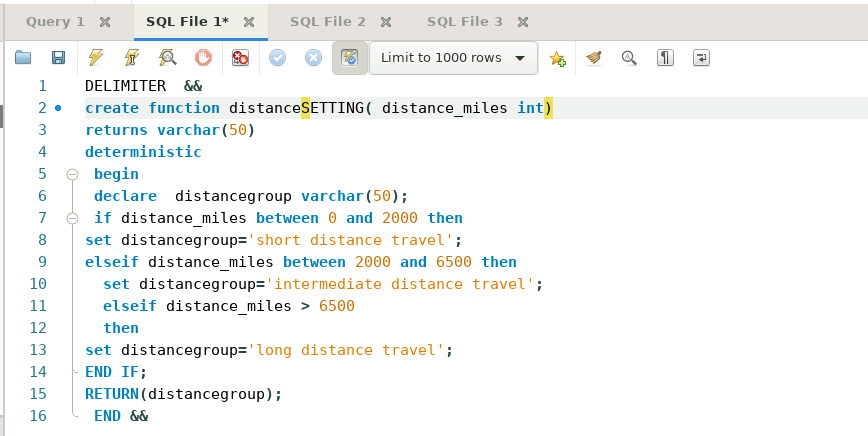
create procedure distanceset1()

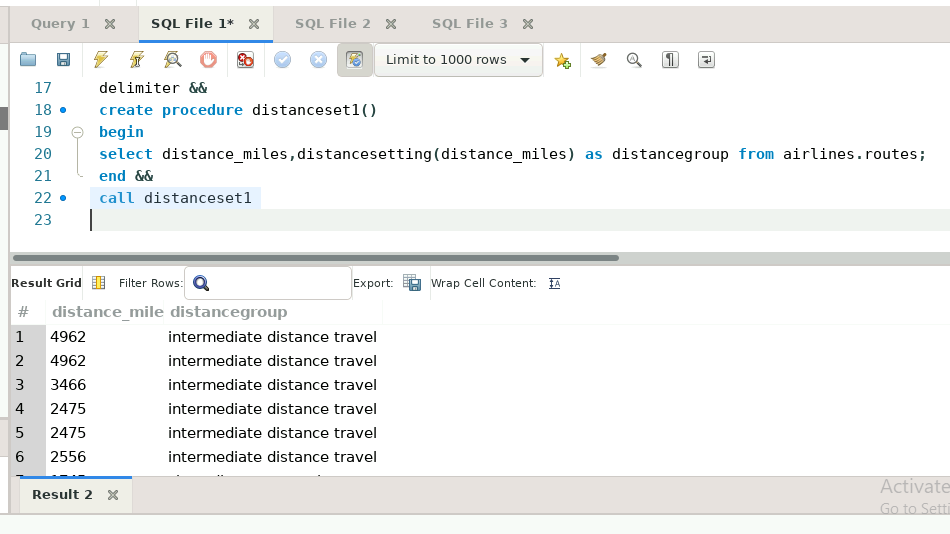
begin

select distance\_miles,distancesetting(distance\_miles) as distancegroup from airlines.routes;

end &&

call distanceset1





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 storedfu(class varchar(20))

returns varchar(20)

deterministic

begin

declare complimentary varchar(20);

if class='Bussiness' or "Economy Plus"then

set complimentary='Yes';

else

set complimentary='NO';

end if;

return(complimentary);

end &&

delimiter &&

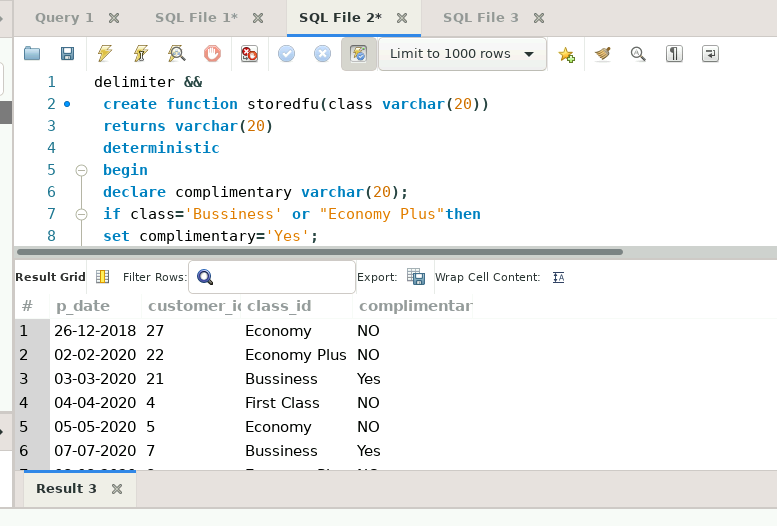
create procedure spc()

begin

select p\_date,customer\_id,class\_id,airlines.storedfu(class\_id)as complimentary from airlines.ticket\_details;

end &&

call spc;



    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.

delimiter &&

create procedure cursorsp()

begin

declare id int;

declare fname varchar(20);

declare lname varchar(30);

declare gender varchar(10);

declare c1 cursor for select Customer\_id,first\_name,last\_name,gender from airlines.customer where last\_name like '%scott';

open c1;

myloop:loop

fetch c1 into id,fname,lname,gender;

select id,fname,lname,gender;

leave myloop;

end loop myloop;

close c1;

end &&

call cursorsp();

