**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.

**Note:** You must download the dataset from the course resource section in the LMS and create the tables to perform the above objective.

**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

**Following operations should 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.

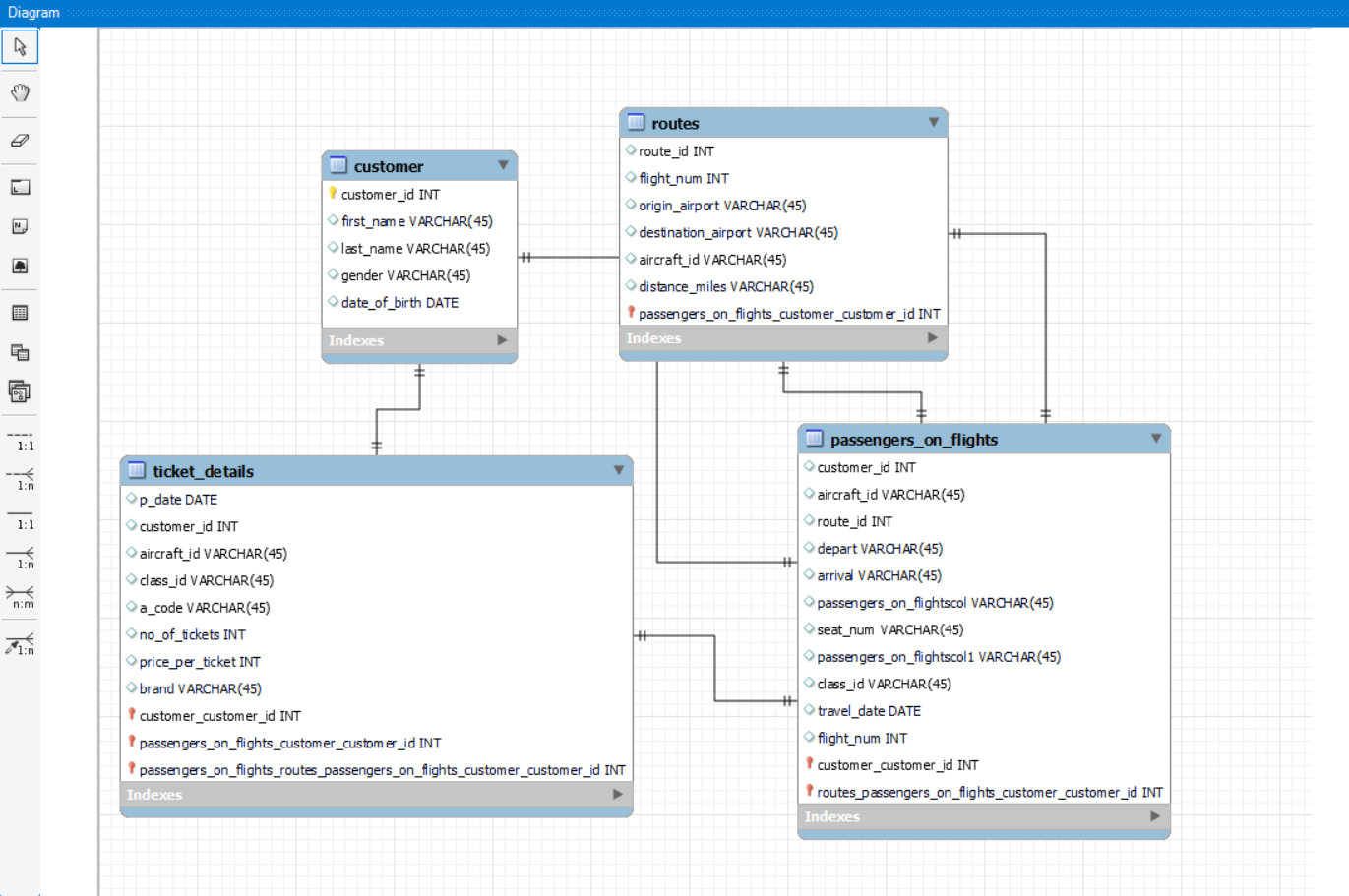
Condition:

* 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.

Answers :

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 `aircargo`.`route\_details`(

`route\_id` int,

`flight\_num` int,

`origin\_airport` varchar(10),

`destination\_airport` varchar(10),

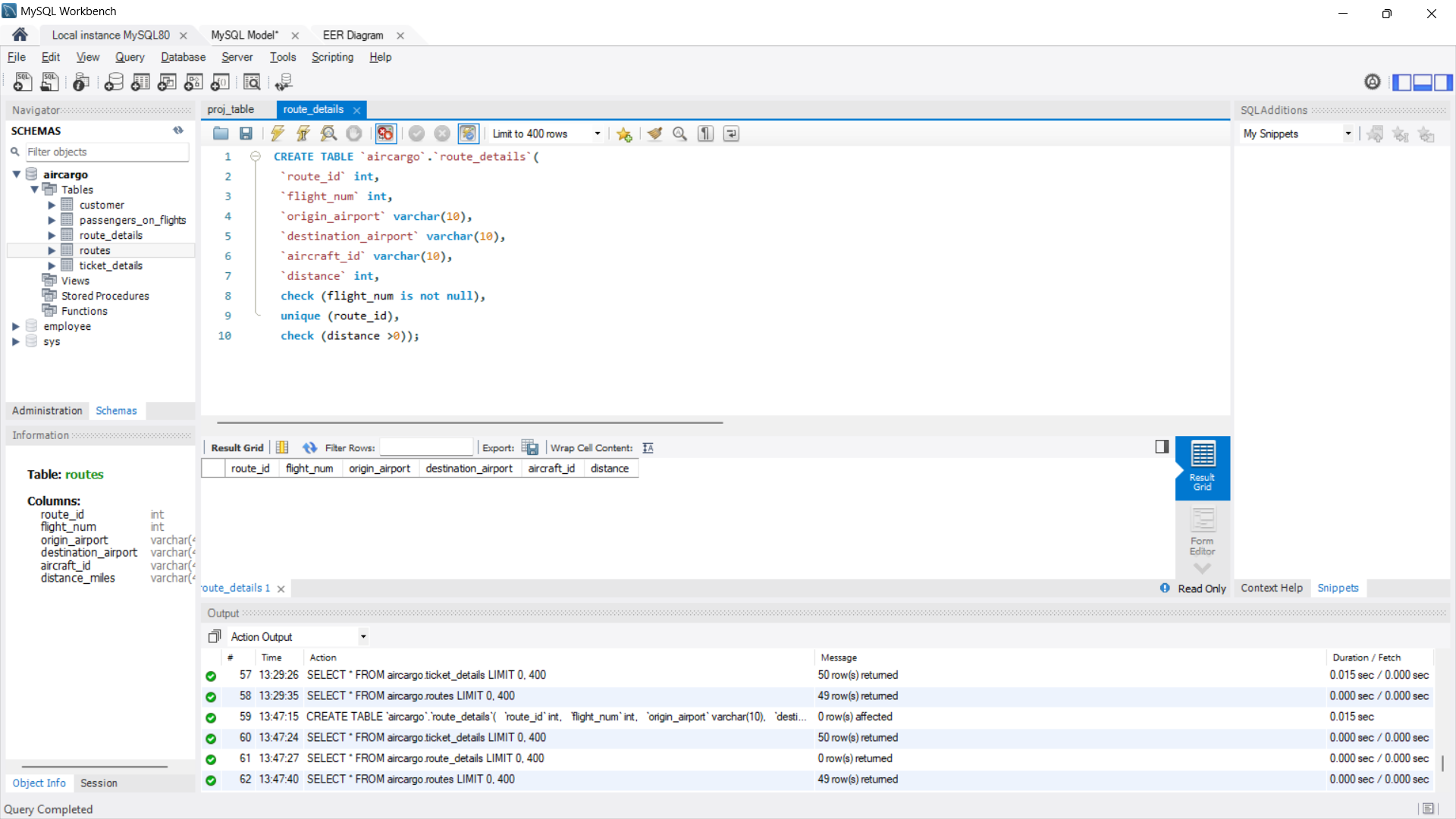
`aircraft\_id` varchar(10),

`distance` int,

check (flight\_num is not null),

unique (route\_id),

check (distance >0));

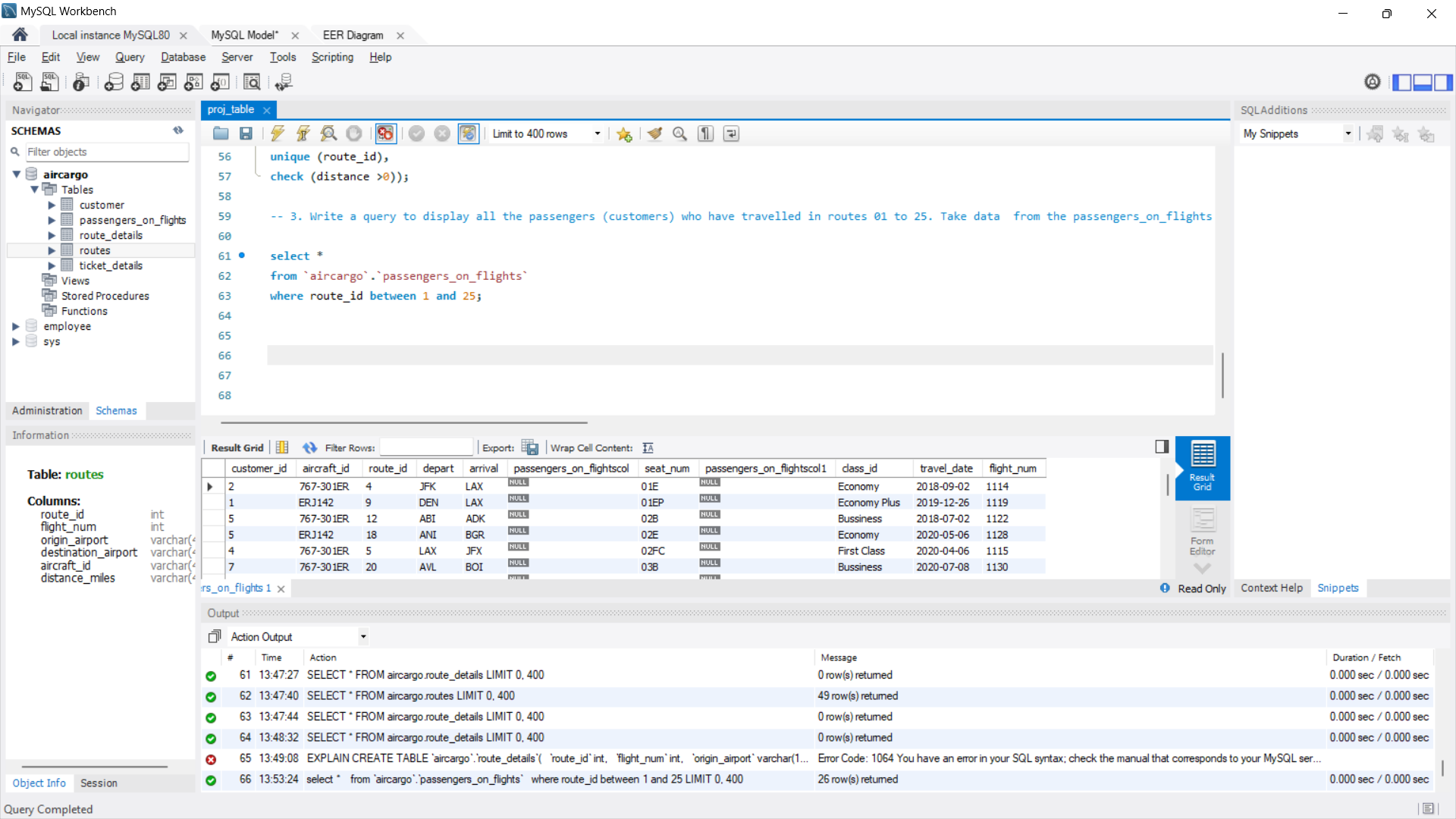


-- 3. Write a query to display all the passengers (customers) who have travelled in routes 01 to 25. Take data from the passengers\_on\_flights table.

select \*

from `aircargo`.`passengers\_on\_flights`

where route\_id between 1 and 25;

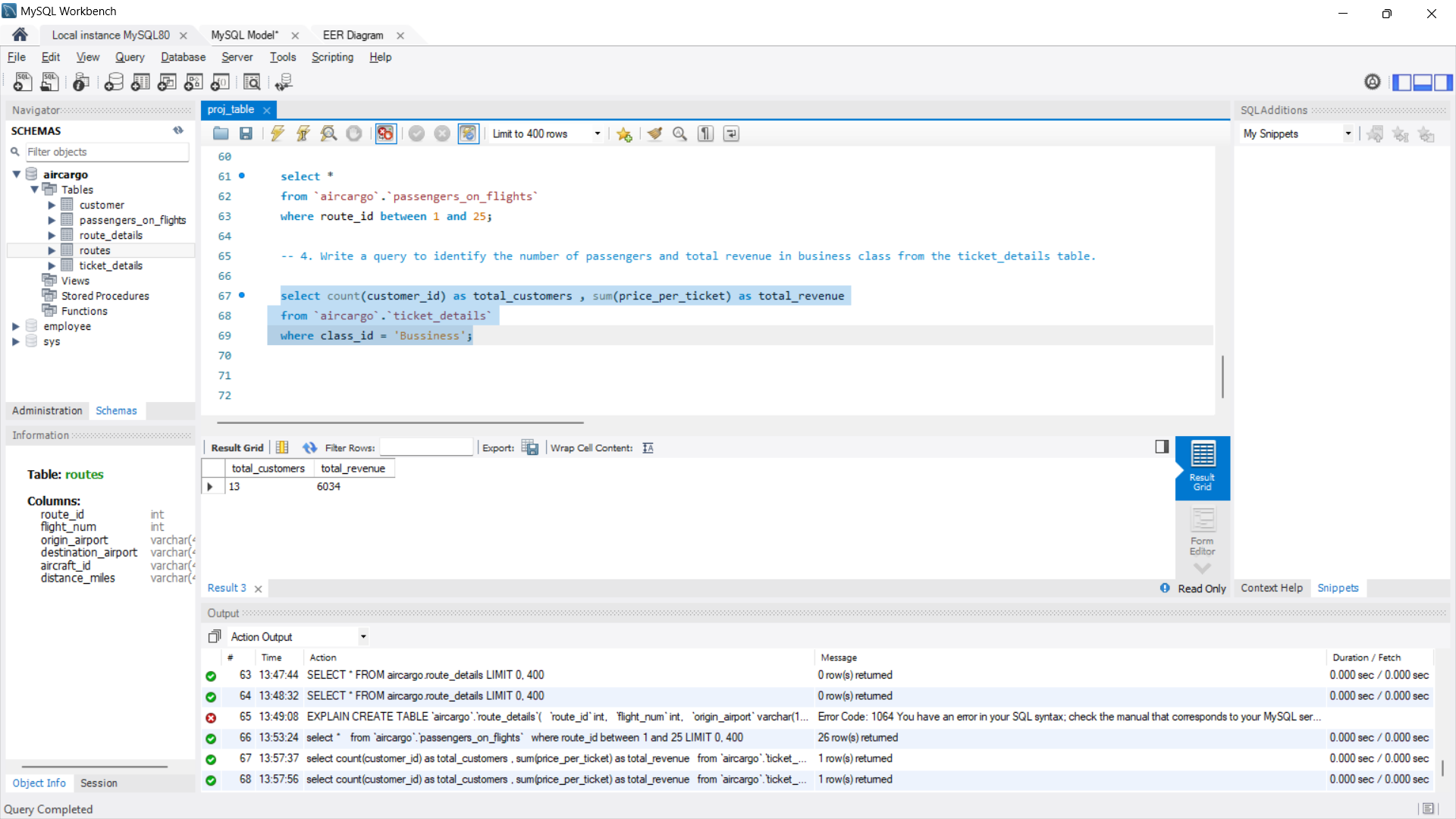


-- 4. Write a query to identify the number of passengers and total revenue in business class from the ticket\_details table.

select count(customer\_id) as total\_customers , sum(price\_per\_ticket) as total\_revenue

from `aircargo`.`ticket\_details`

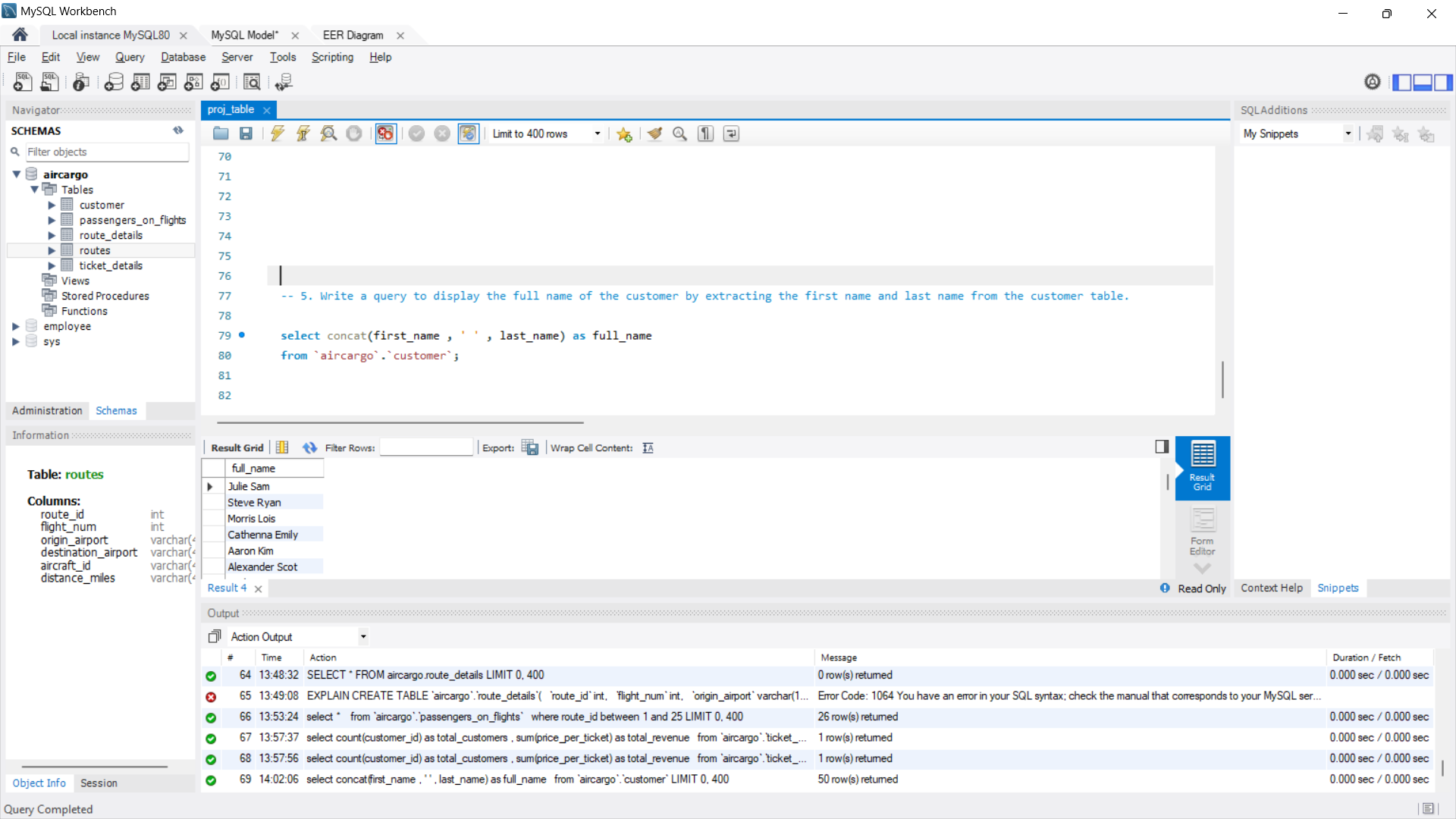
where class\_id = 'Bussiness';



-- 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 concat(first\_name , ' ' , last\_name) as full\_name

from `aircargo`.`customer`;



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

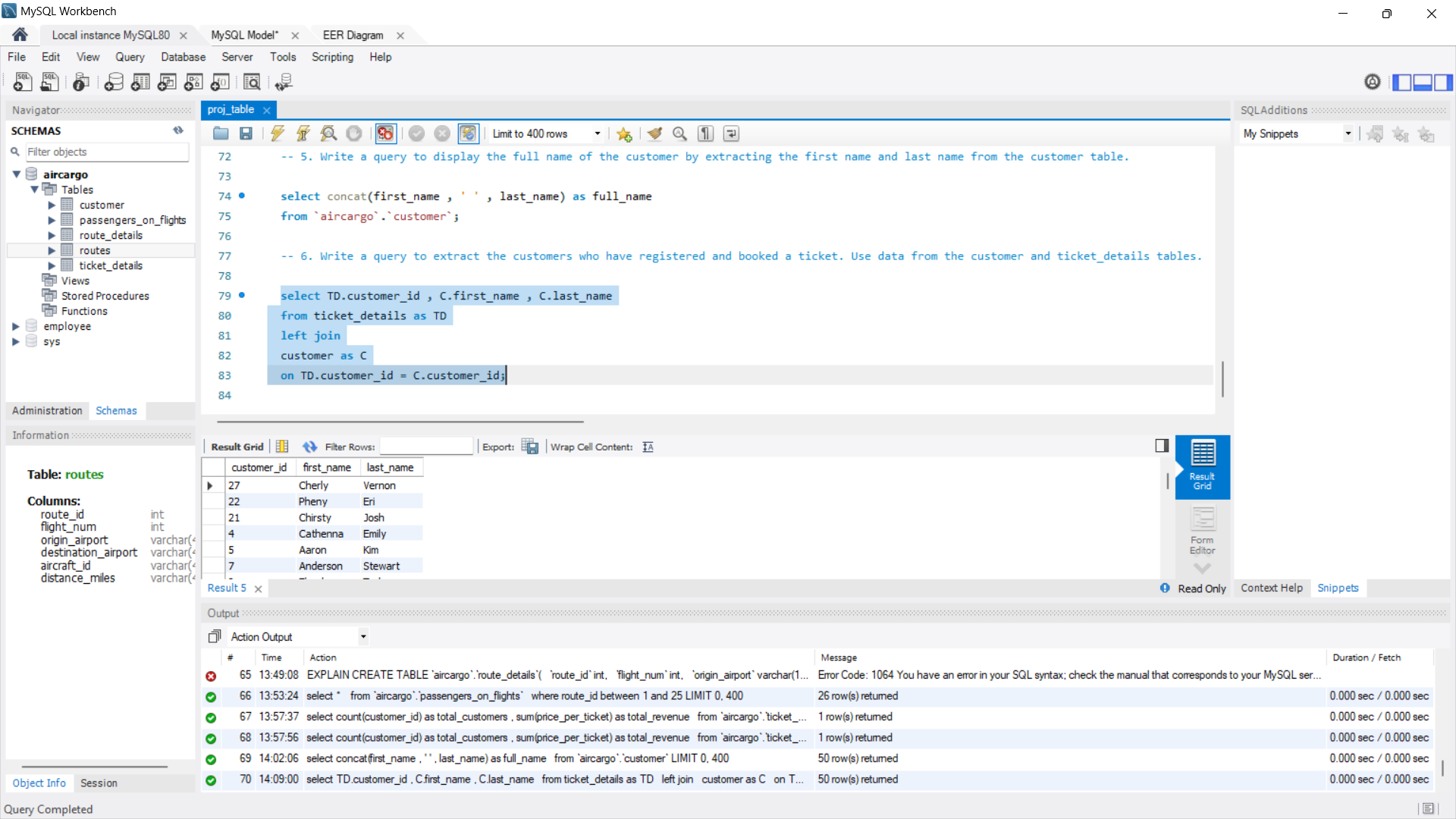
select TD.customer\_id , C.first\_name , C.last\_name

from ticket\_details as TD

left join

customer as C

on TD.customer\_id = C.customer\_id;



-- 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.

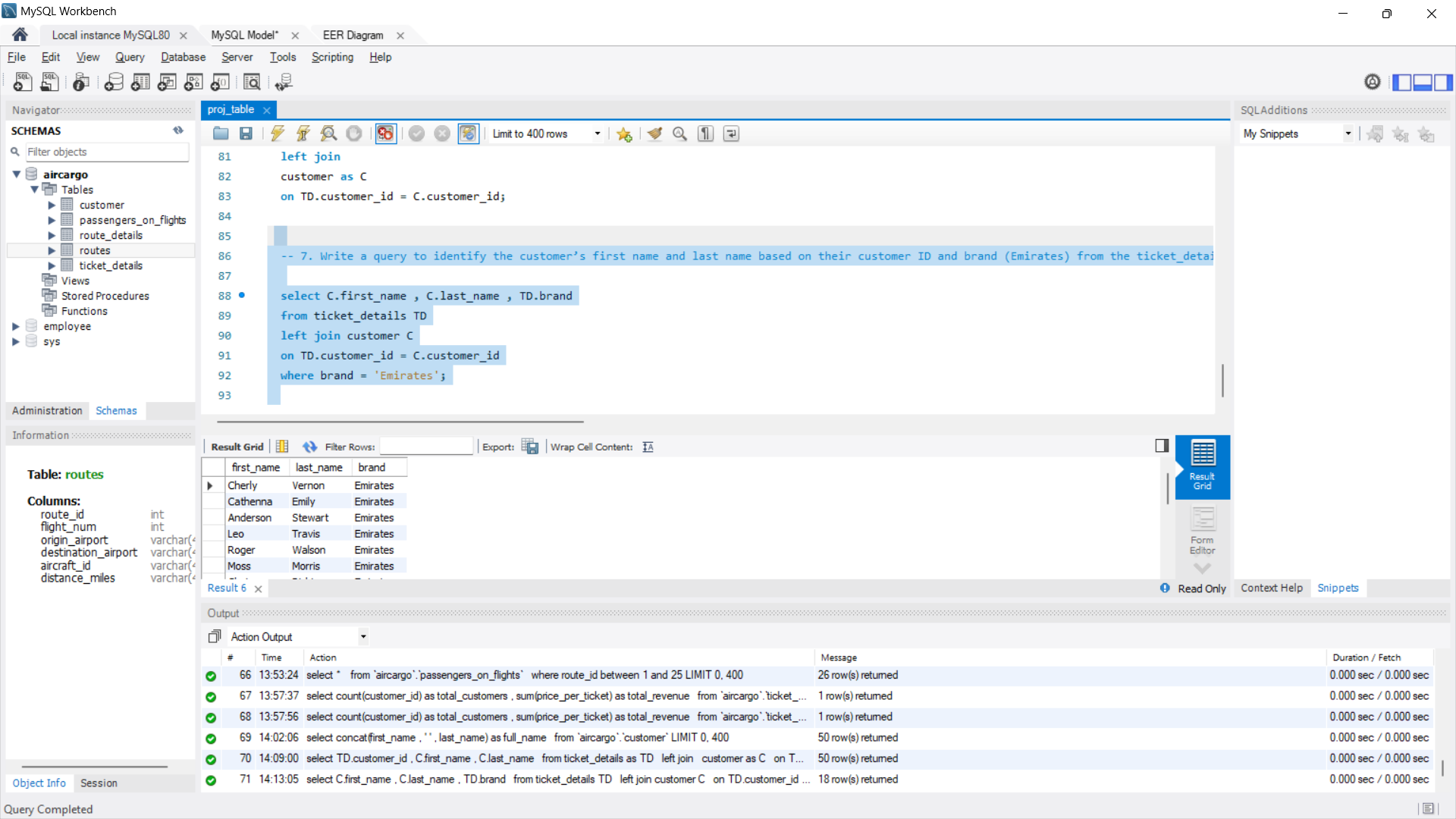
select C.first\_name , C.last\_name , TD.brand

from ticket\_details TD

left join customer C

on TD.customer\_id = C.customer\_id

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 C.first\_name , C.last\_name , PF.customer\_id , PF.class\_id

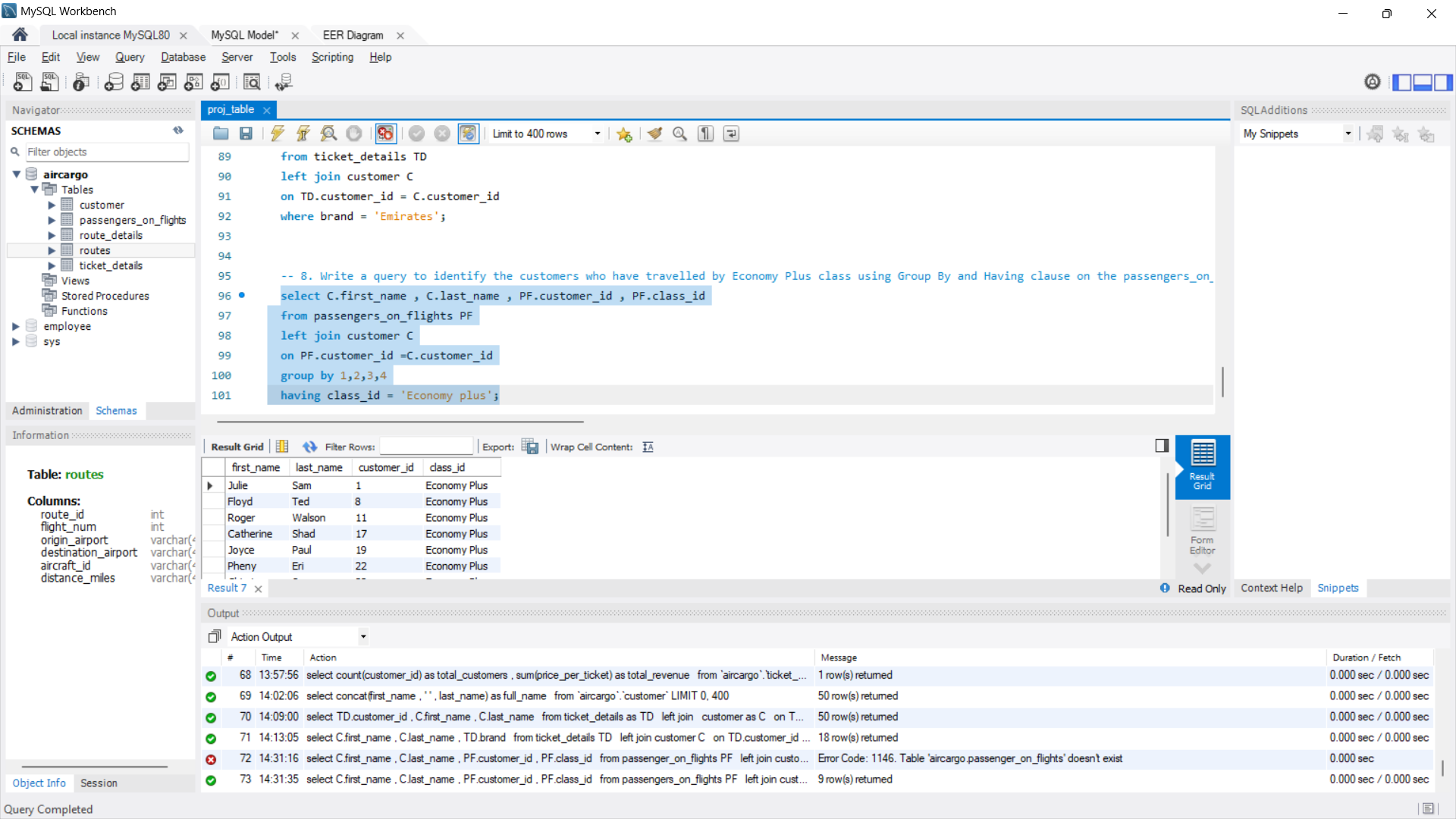
from passengers\_on\_flights PF

left join customer C

on PF.customer\_id =C.customer\_id

group by 1,2,3,4

having class\_id = 'Economy plus';

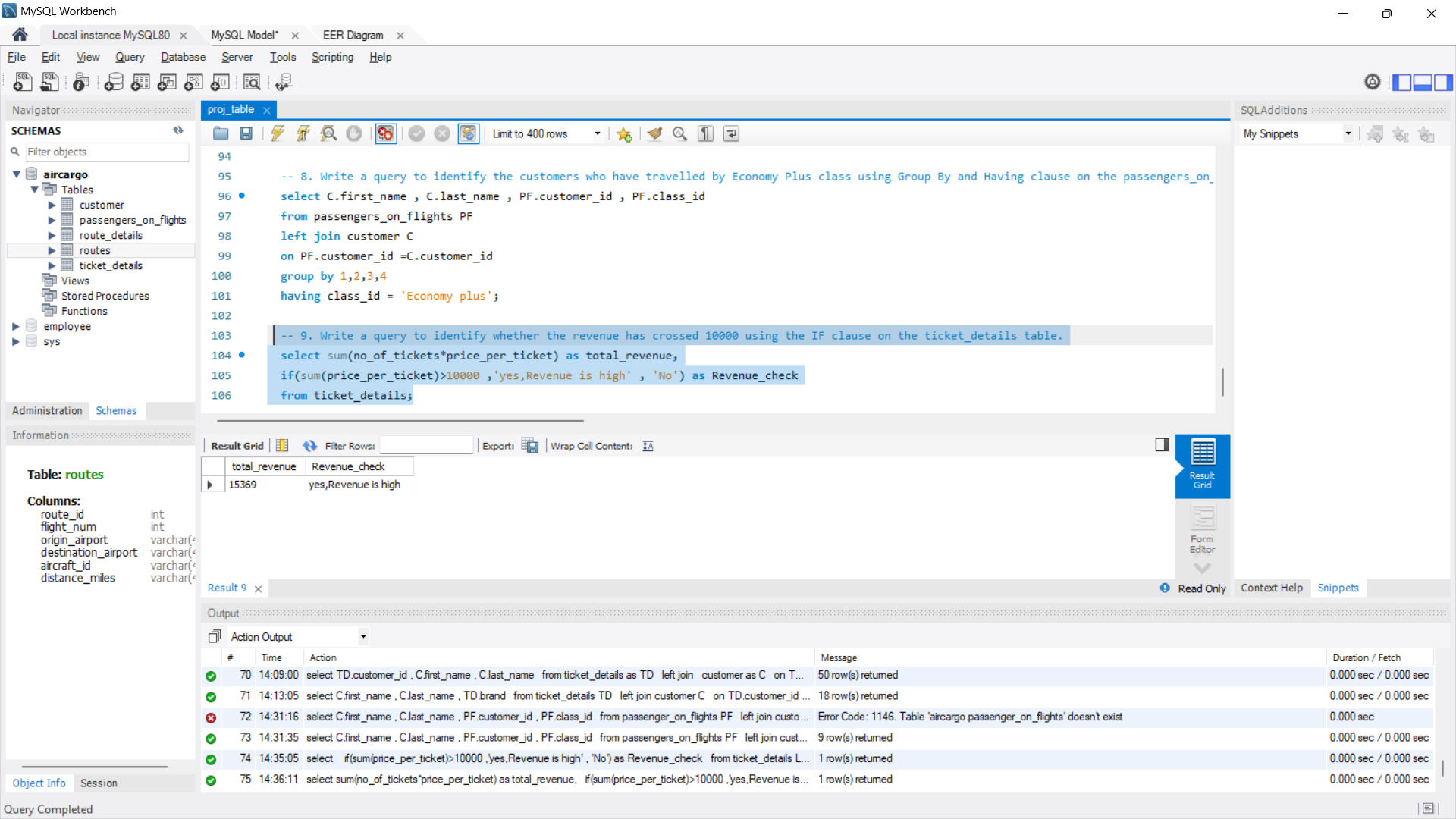


-- 9. Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket\_details table.

select sum(no\_of\_tickets\*price\_per\_ticket) as total\_revenue,

if(sum(price\_per\_ticket)>10000 ,'yes,Revenue is high' , 'No') as Revenue\_check

from ticket\_details;



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

select user , host from mysql.user;

create user paraspahwa@localhost;

show grants for paraspahwa@localhost;

grant all on aircargo to paraspahwa@localhost;

show grants for paraspahwa@localhost;



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

with pahwa as (

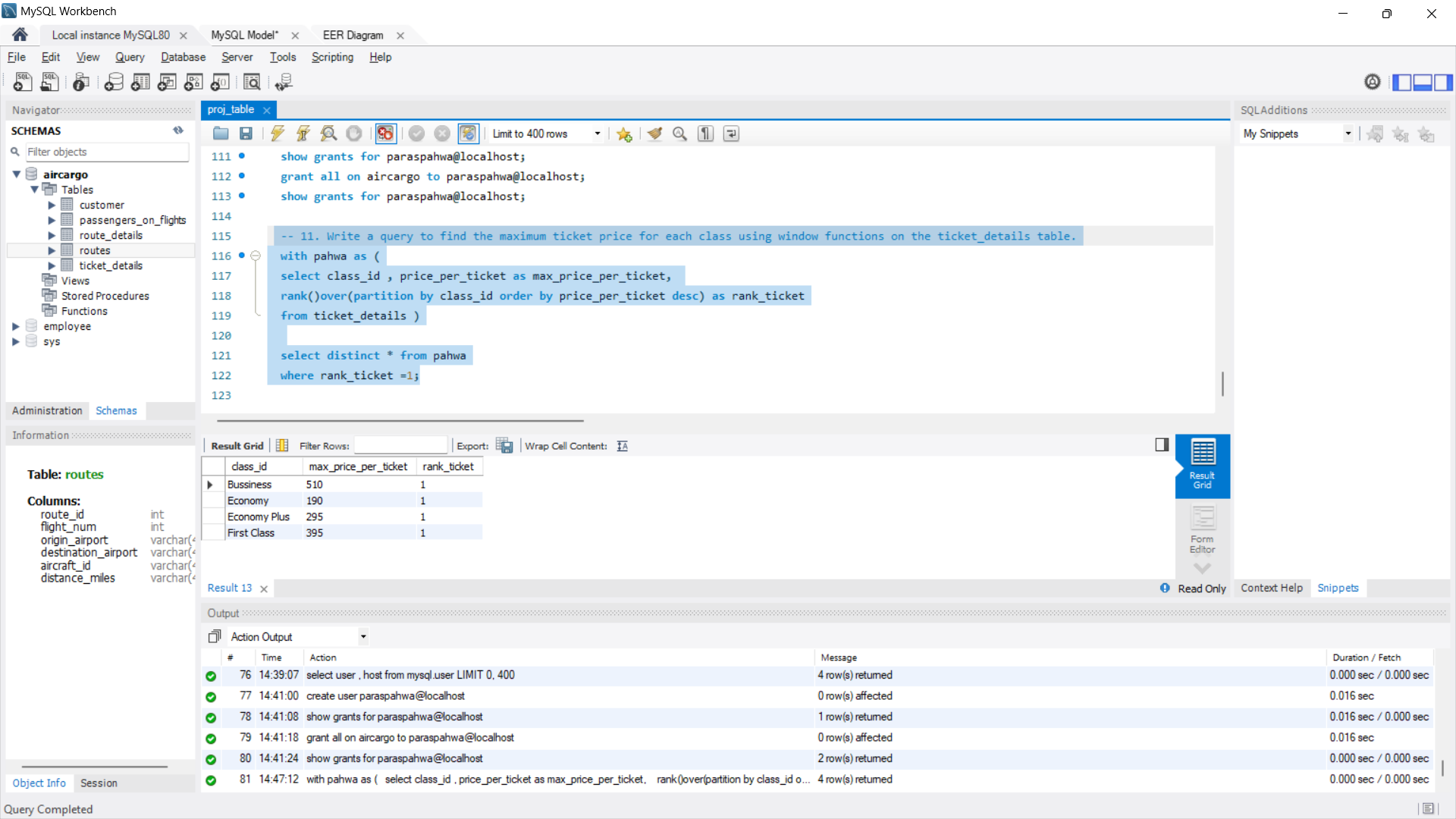
select class\_id , price\_per\_ticket as max\_price\_per\_ticket,

rank()over(partition by class\_id order by price\_per\_ticket desc) as rank\_ticket

from ticket\_details )

select distinct \* from pahwa

where rank\_ticket =1;



-- 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.

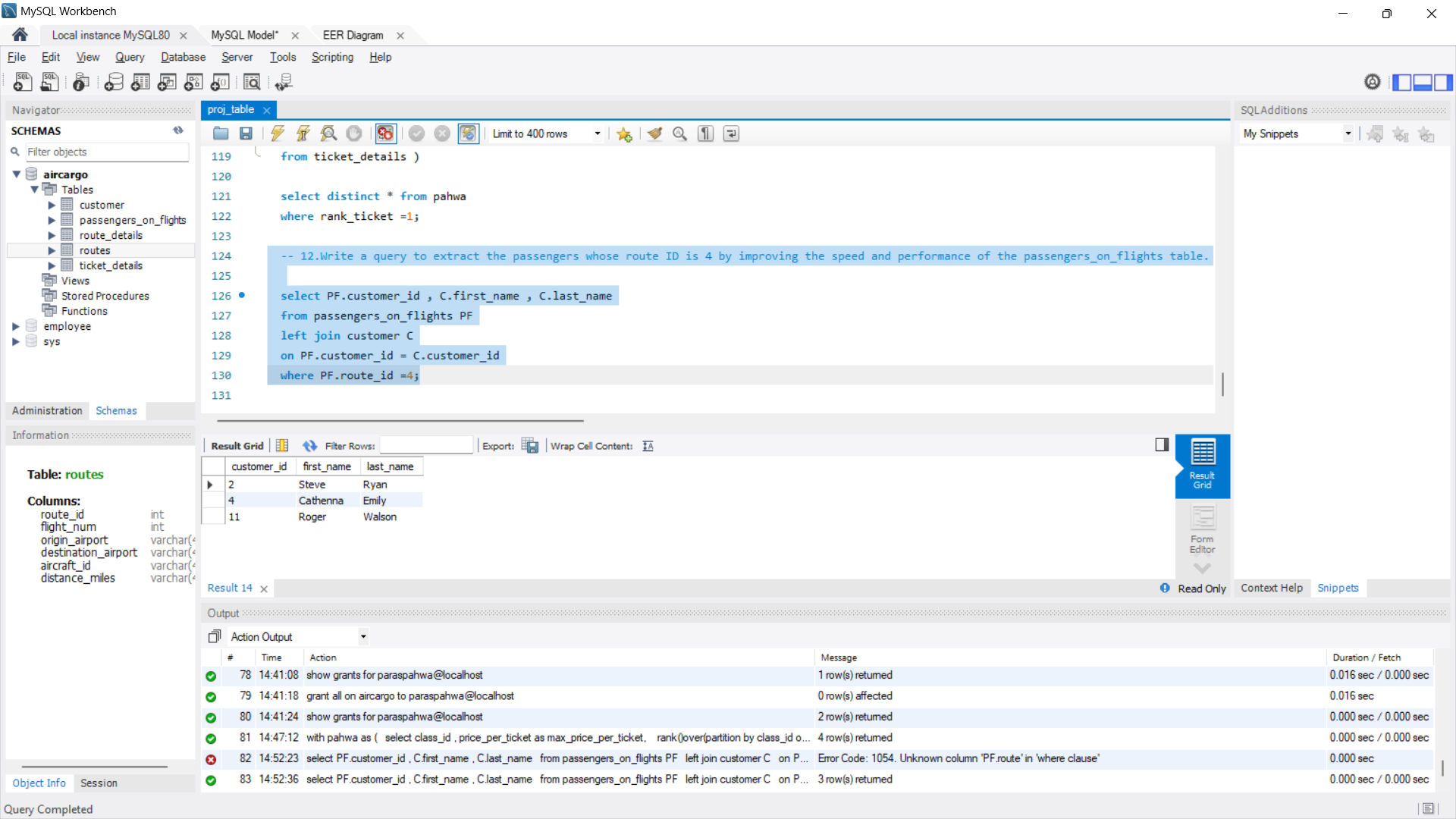
select PF.customer\_id , C.first\_name , C.last\_name

from passengers\_on\_flights PF

left join customer C

on PF.customer\_id = C.customer\_id

where PF.route\_id =4;

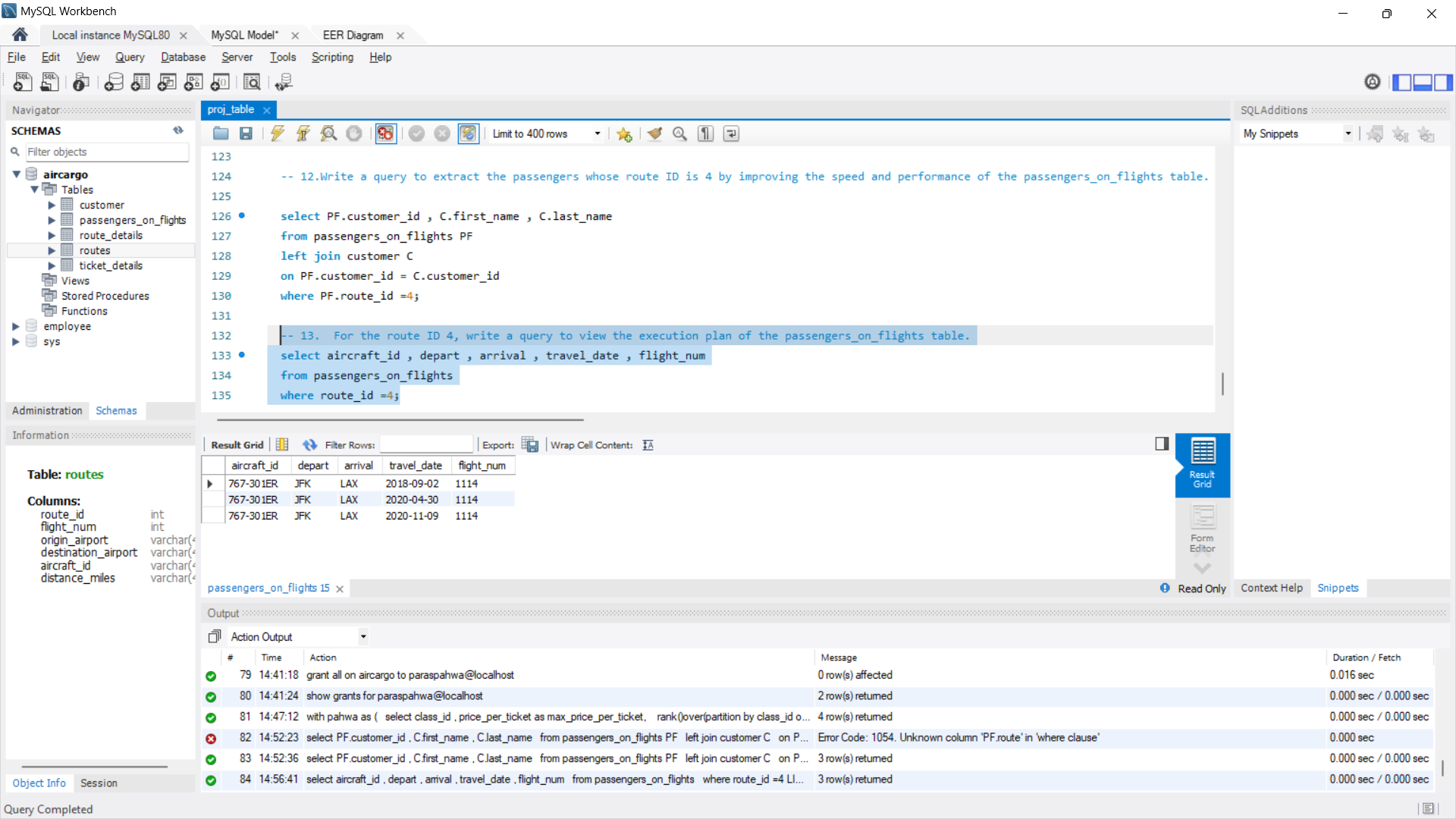


-- 13. For the route ID 4, write a query to view the execution plan of the passengers\_on\_flights table.

select aircraft\_id , depart , arrival , travel\_date , flight\_num

from passengers\_on\_flights

where route\_id =4;

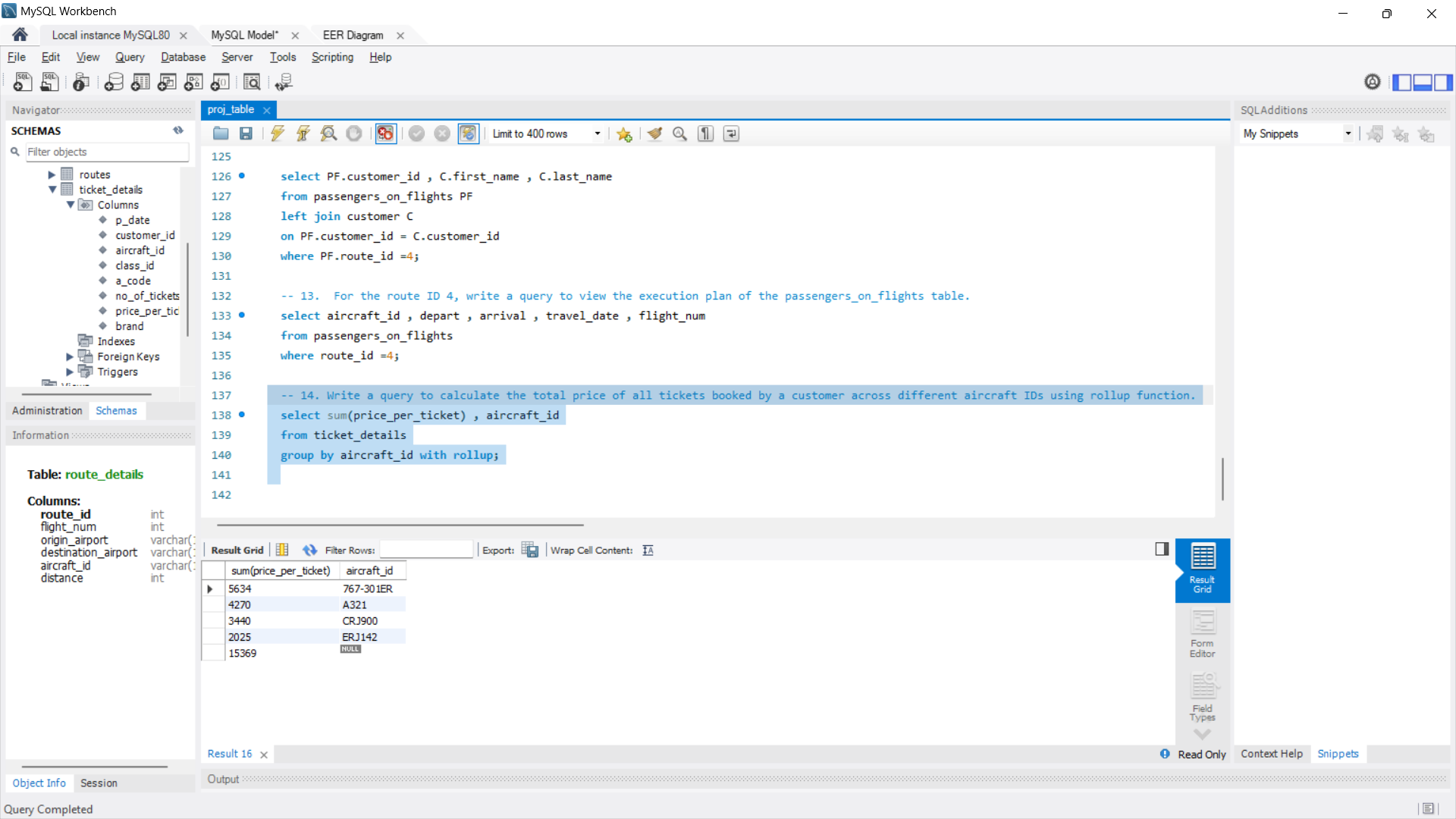


-- 14. Write a query to calculate the total price of all tickets booked by a customer across different aircraft IDs using rollup function.

select sum(price\_per\_ticket) , aircraft\_id

from 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 brand

as

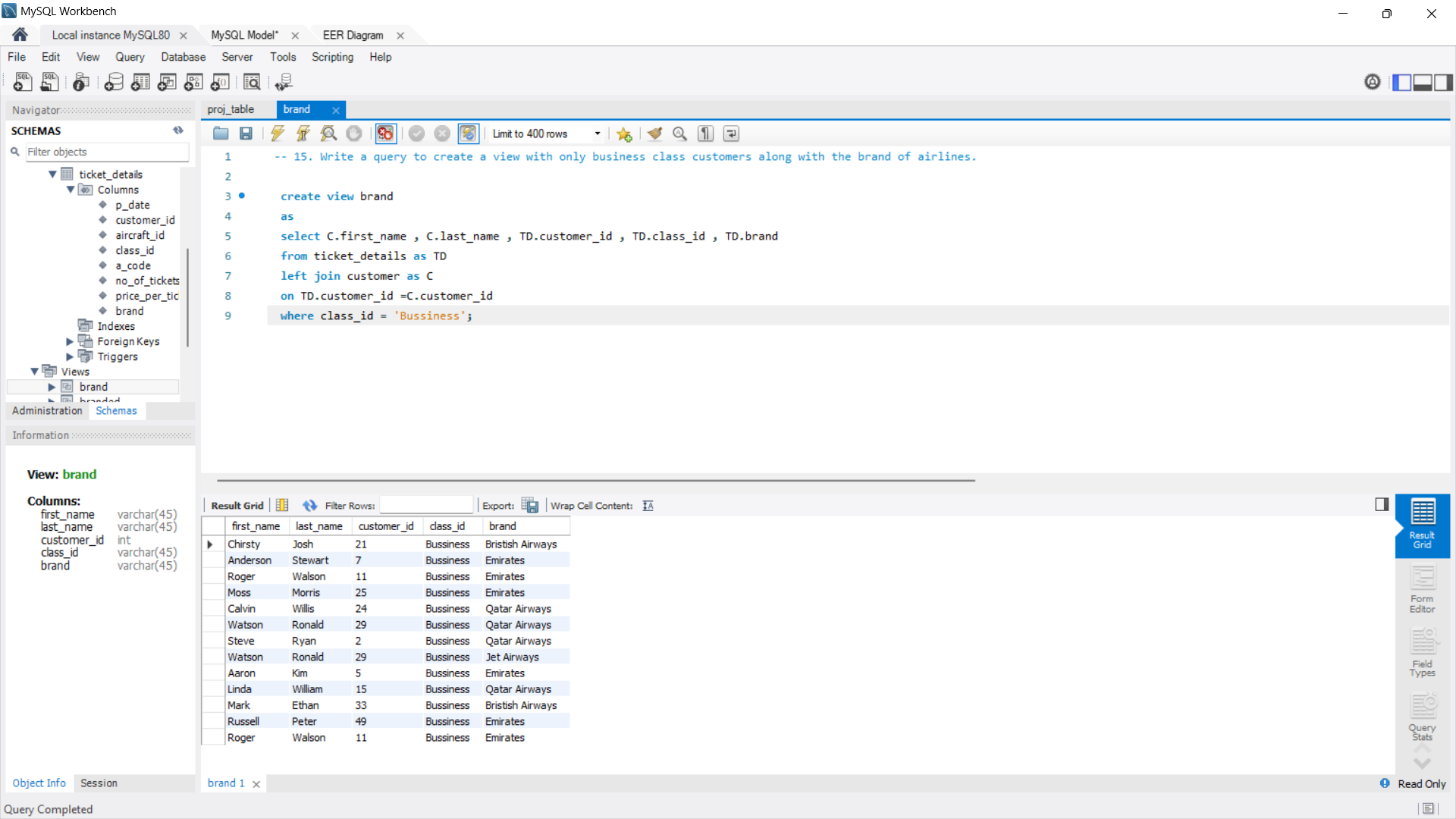
select C.first\_name , C.last\_name , TD.customer\_id , TD.class\_id , TD.brand

from ticket\_details as TD

left join customer as C

on TD.customer\_id =C.customer\_id

where class\_id = 'Bussiness';



-- 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 route2(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 routes RD

inner join passengers\_on\_flights PF

on RD.route\_id =PF.route\_id

left join customer C

using (customer\_id)

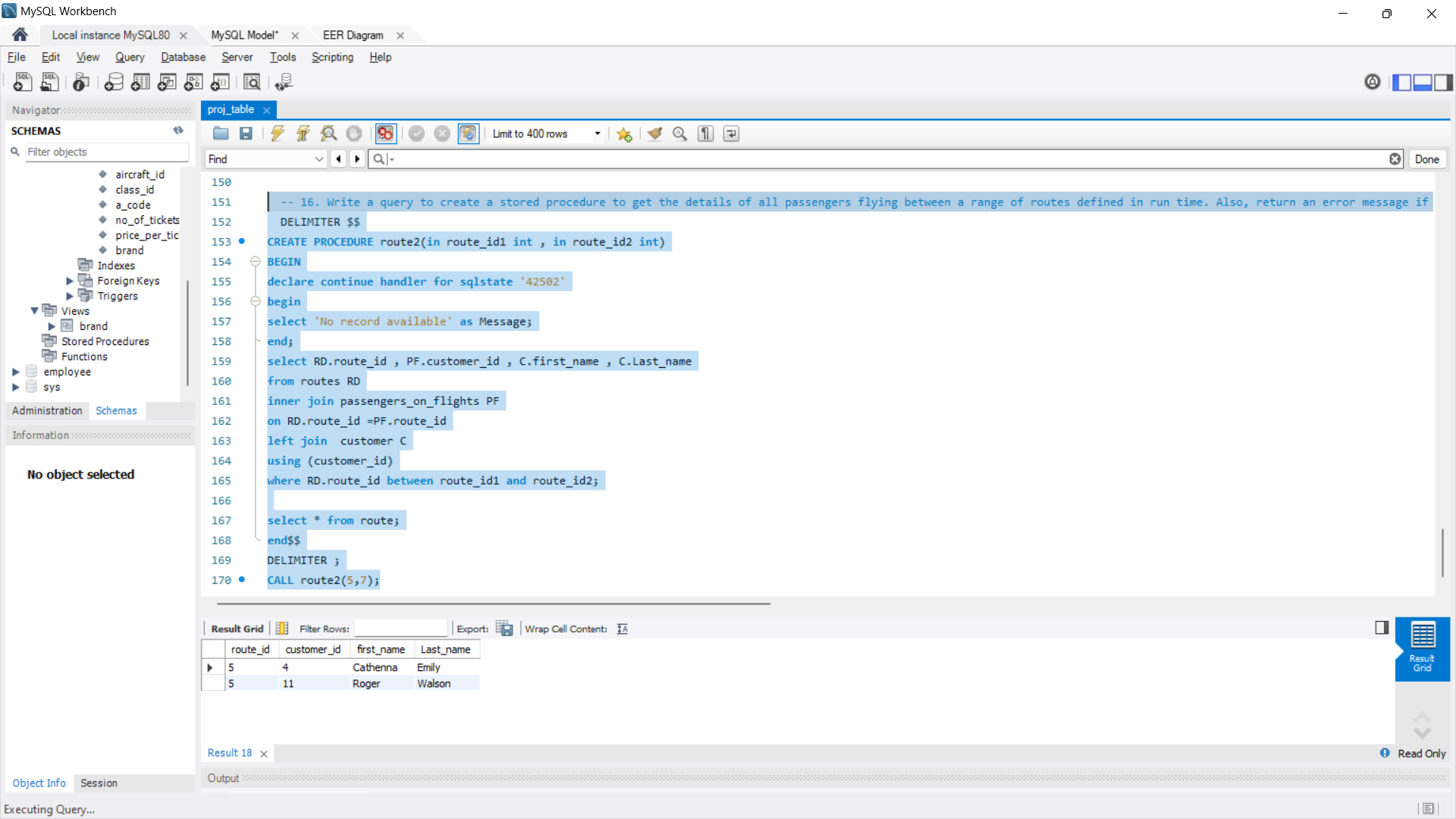
where RD.route\_id between route\_id1 and route\_id2;

select \* from route;

end$$

DELIMITER ;

CALL route2(5,7);



-- 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 distance()

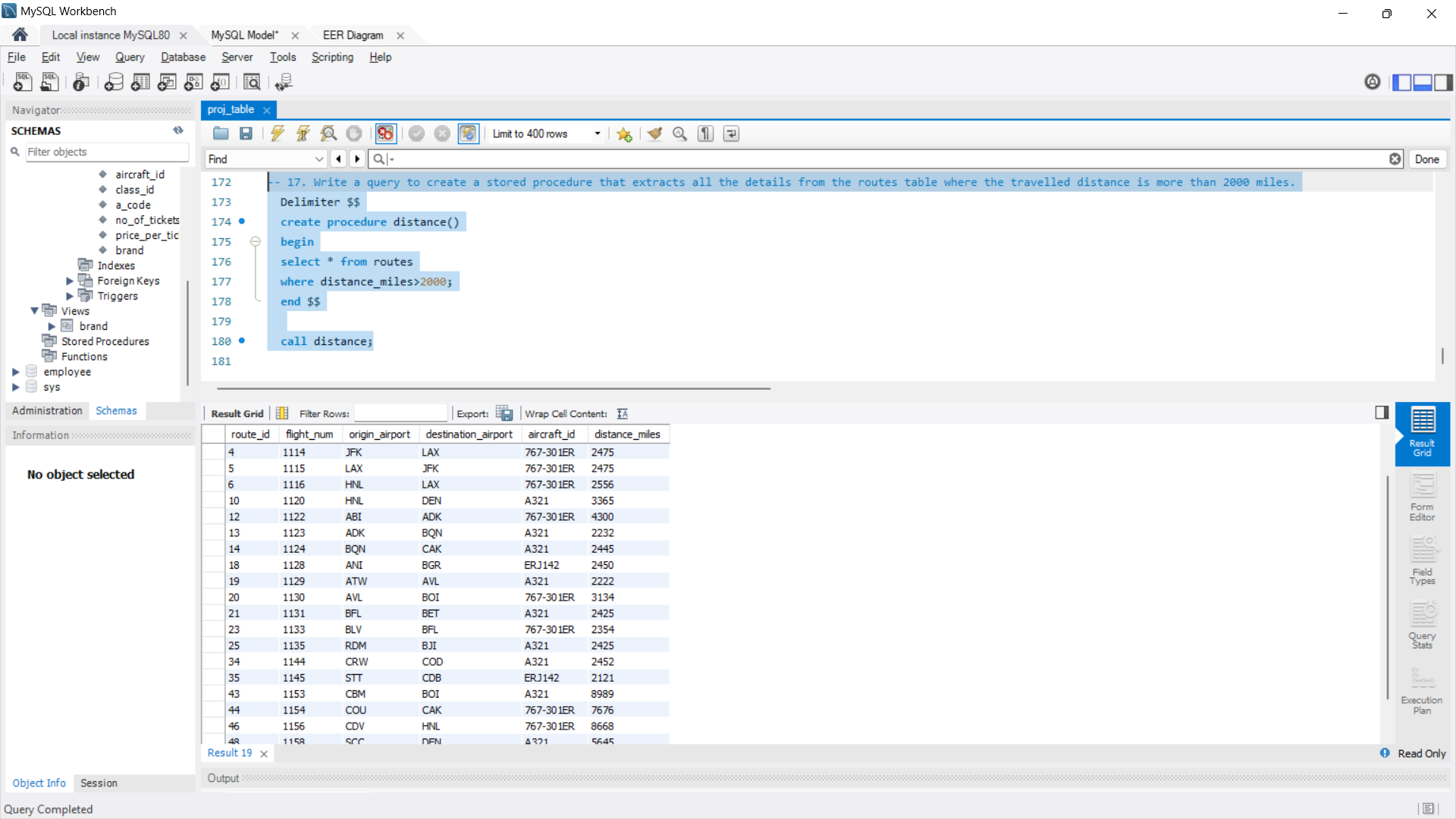
begin

select \* from routes

where distance\_miles>2000;

end $$

call distance;



* 1. 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.

CREATE DEFINER=`root`@`localhost` PROCEDURE `distancegrouping`(In flight\_num1 int)

BEGIN

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 routes

where flight\_num= flight\_num1;

END

call aircargo.distancegrouping(1111);



-- 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 $$

USE `aircargo`$$

CREATE PROCEDURE `complementary\_service` (In 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 complementary service'

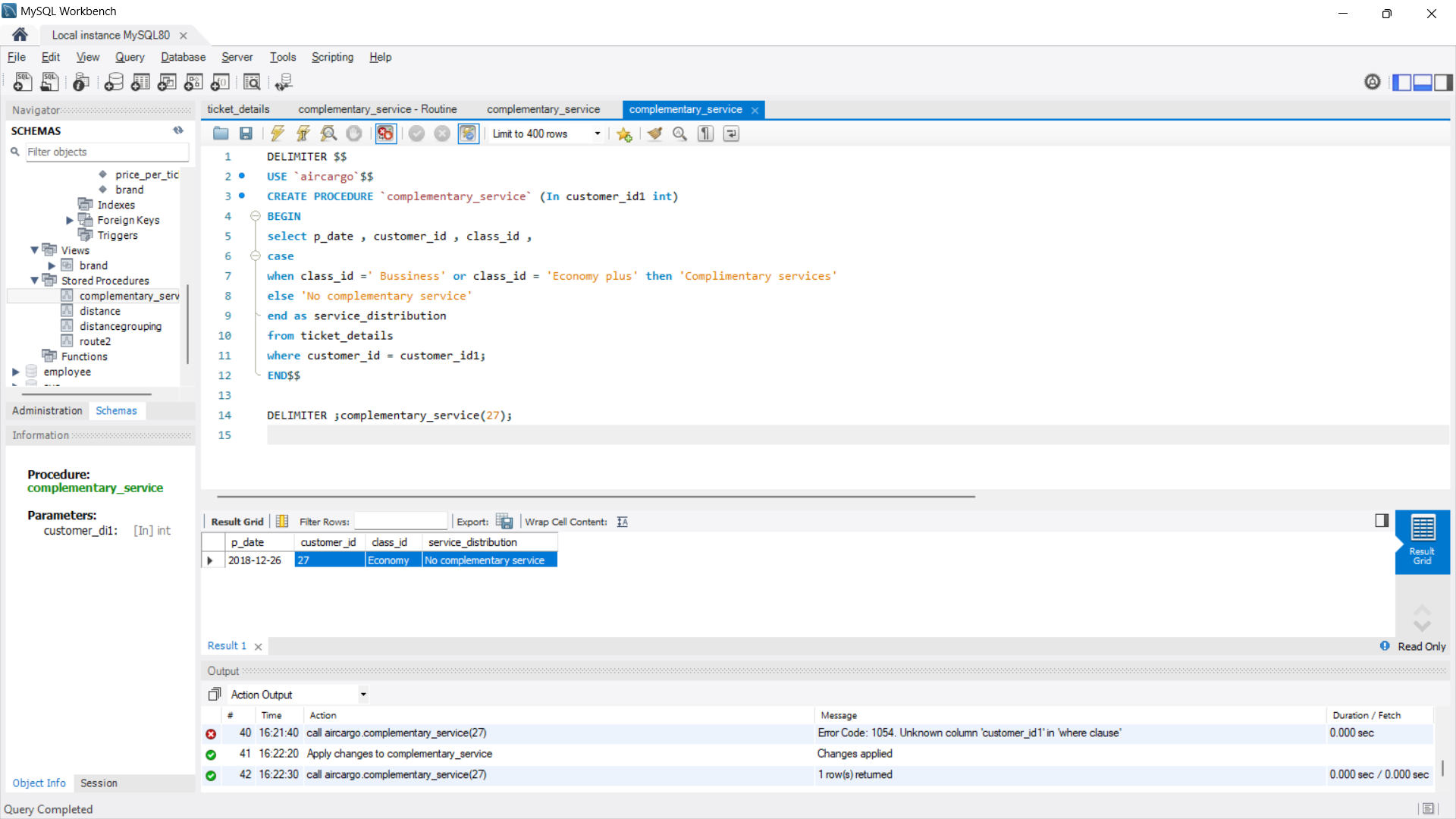
end as service\_distribution

from ticket\_details

where customer\_id = customer\_id1;

END$$

DELIMITER ;complementary\_service(27);



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.

USE `aircargo`;

DROP procedure IF EXISTS `firstrecord`;

DELIMITER $$

USE `aircargo`$$

CREATE PROCEDURE `firstrecord`()

BEGIN declare a varchar(20);

DECLARE b varchar(20);

declare c int;

declare cursor\_1 cursor for select first\_name , last\_name , customer\_id from customer

where last\_name = 'scott';

open cursor\_1;

fetch cursor\_1 into a, b, c;

select a as first\_name , b as last\_name , c as customer\_id;

close cursor\_1;

END$$

call aircargo.firstrecord();

