**Project: Air Cargo Analysis**

**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.

**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

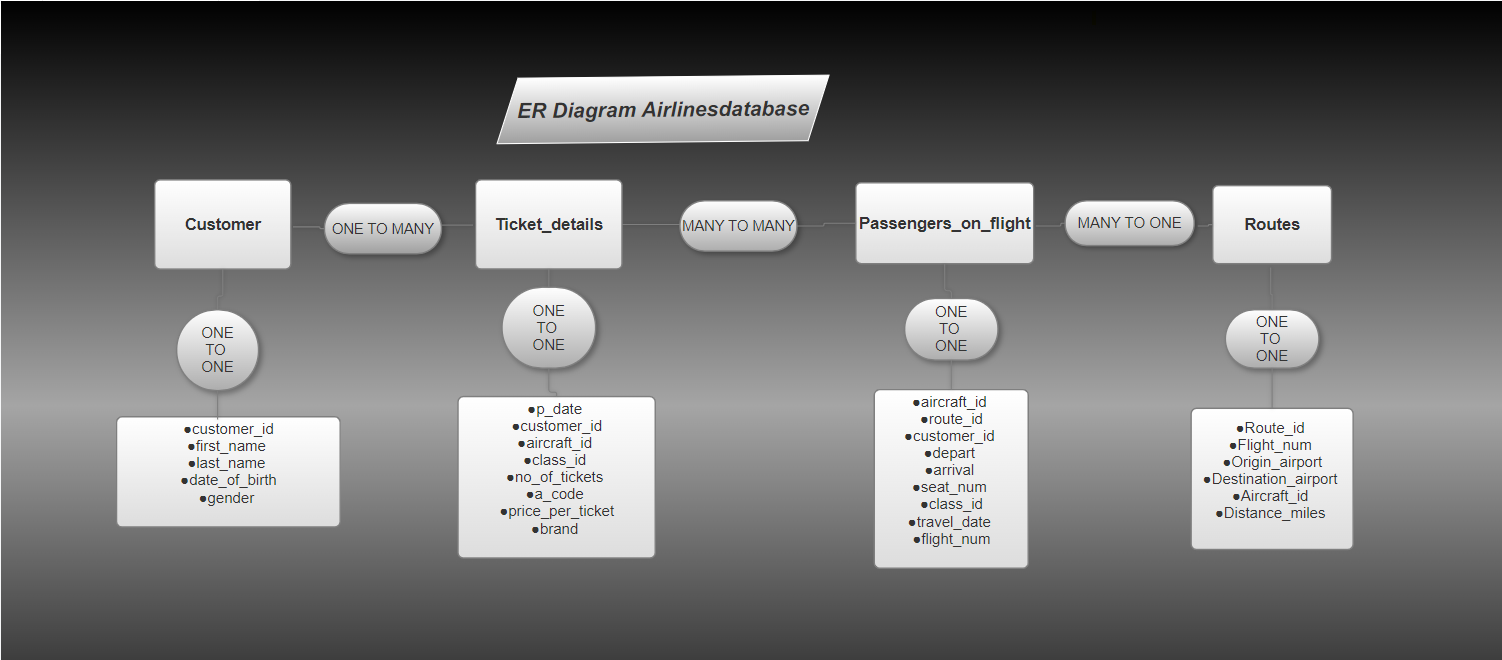
* p\_date – Ticket purchase date
* customer\_id – ID of the customer
* aircraft\_id – ID of each aircraft in a brand
* class\_id – ID of travel class
* no\_of\_tickets – Number of tickets purchased
* a\_code – Code of each airport
* price\_per\_ticket – Price of a ticket
* brand – Aviation service provider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

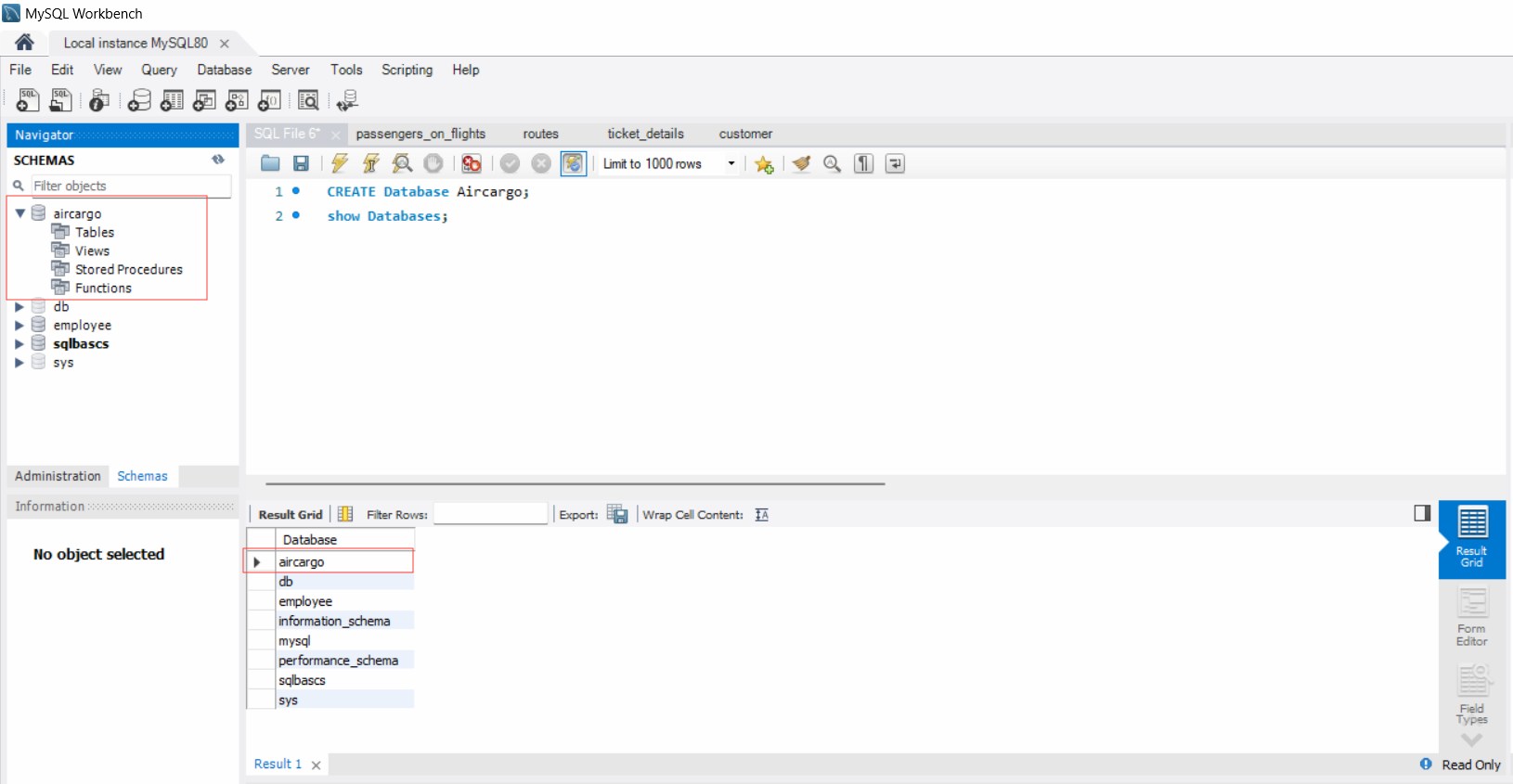
1. Create an ER diagram for the given airlines database.



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.

CREATE Database Aircargo;

show Databases;



**customer**

1. use Aircargo;

create table customer(

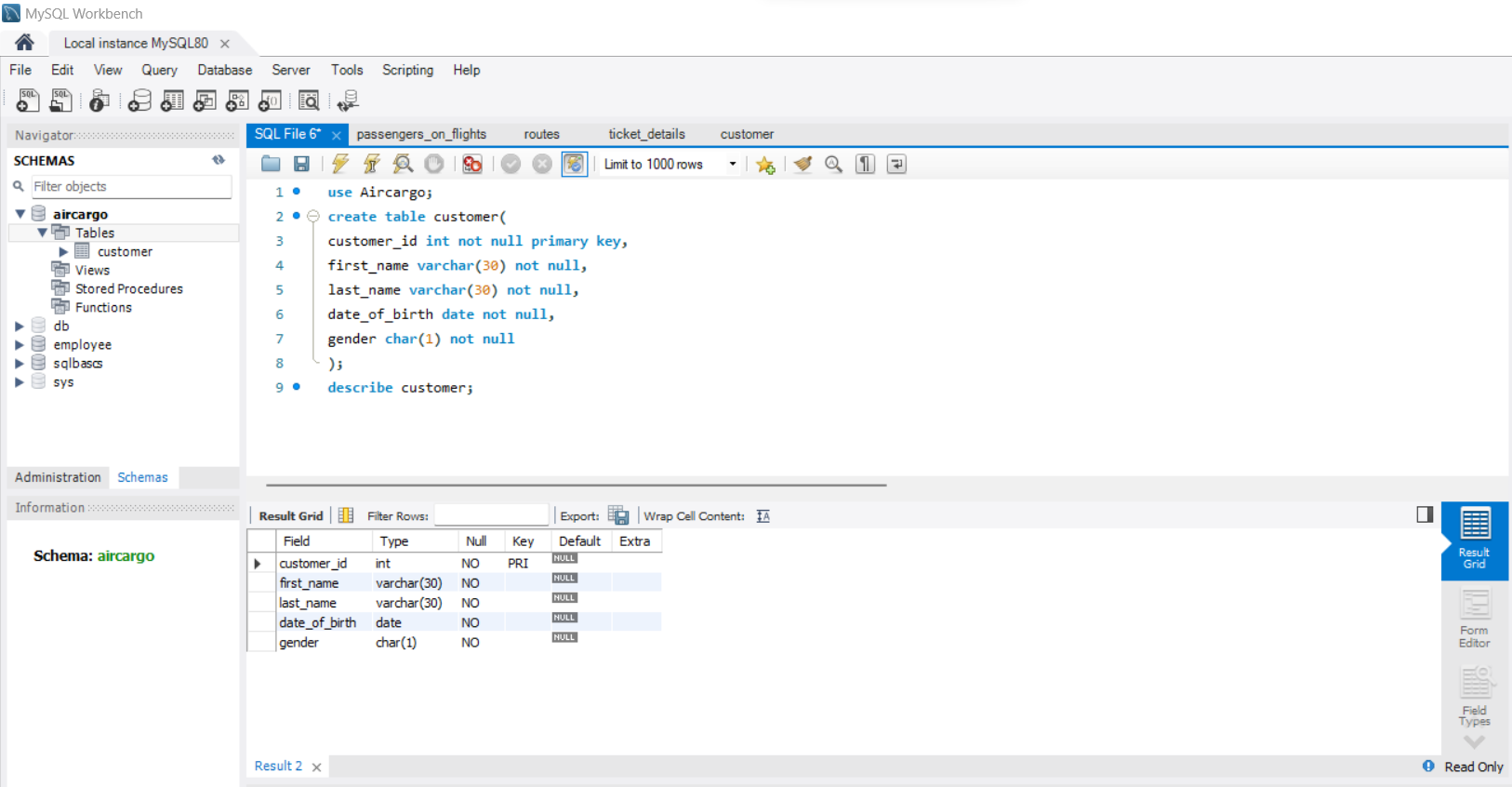
customer\_id int not null primary key,

first\_name varchar(30) not null,

last\_name varchar(30) not null,

date\_of\_birth date not null,

gender char(1) not null);describe customer;

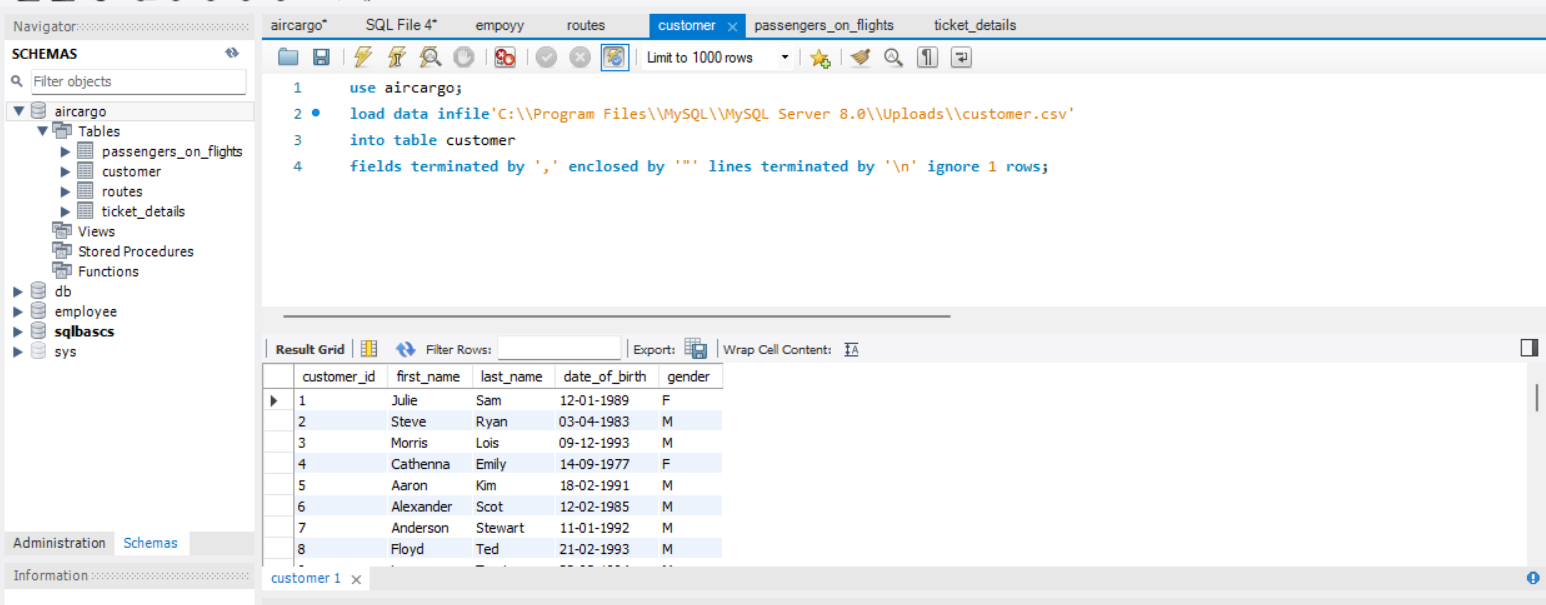


use aircargo;

load data infile'C:\\Program Files\\MySQL\\MySQL Server 8.0\\Uploads\\customer.csv'

into table customer

fields terminated by ',' enclosed by '"' lines terminated by '\n' ignore 1 rows;



**Tickets\_detail**

use Aircargo;

create table tickets\_detail(

tkt\_id int auto\_increment primary key,

p\_date date not null,

customer\_id int not null,

aircraft\_id varchar(10)not null,

class\_id varchar(30) not null,

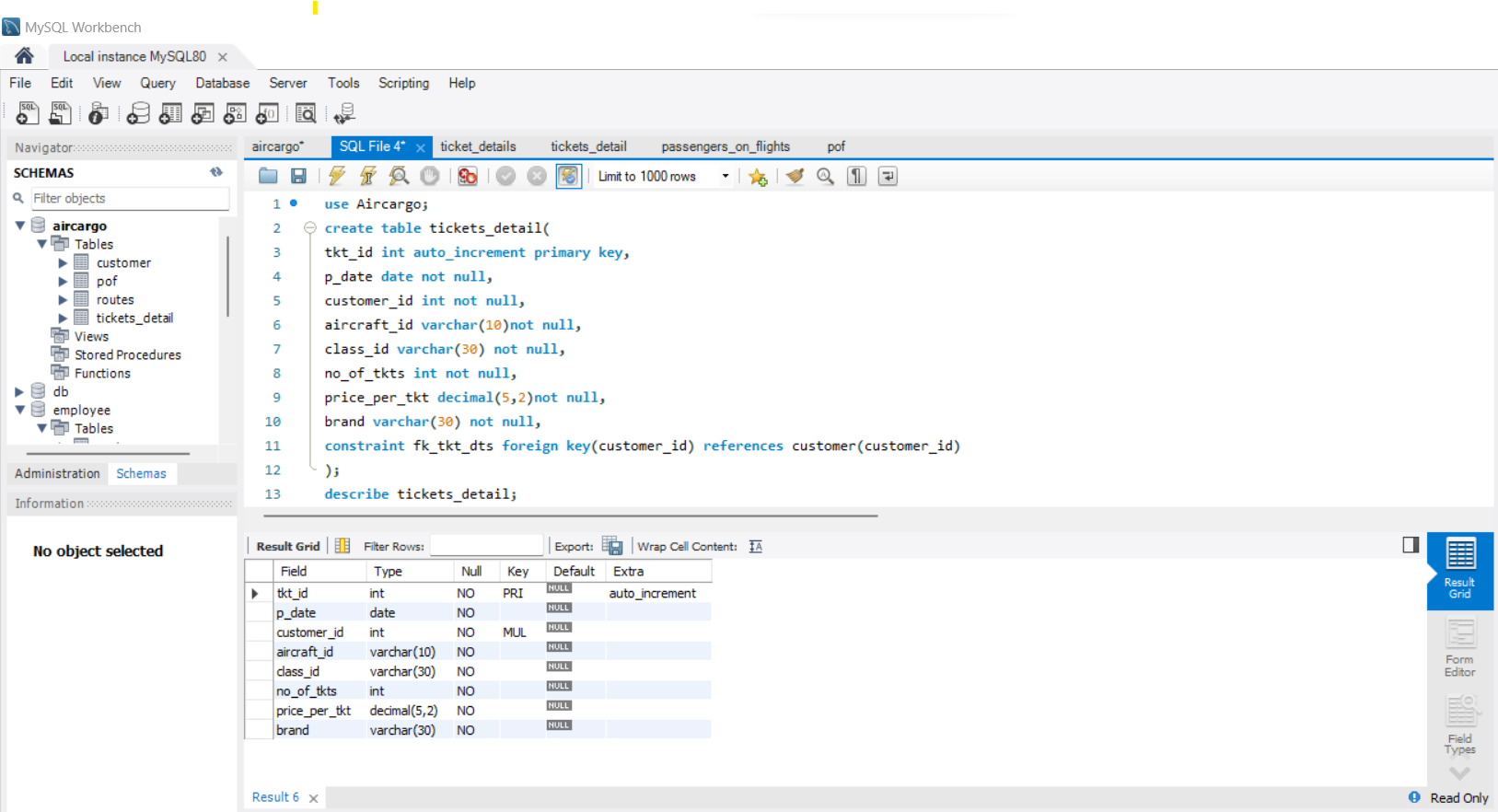
no\_of\_tkts int not null,

price\_per\_tkt decimal(5,2)not null,

brand varchar(30) not null,

constraint fk\_tkt\_dts foreign key(customer\_id) references customer(customer\_id)

);

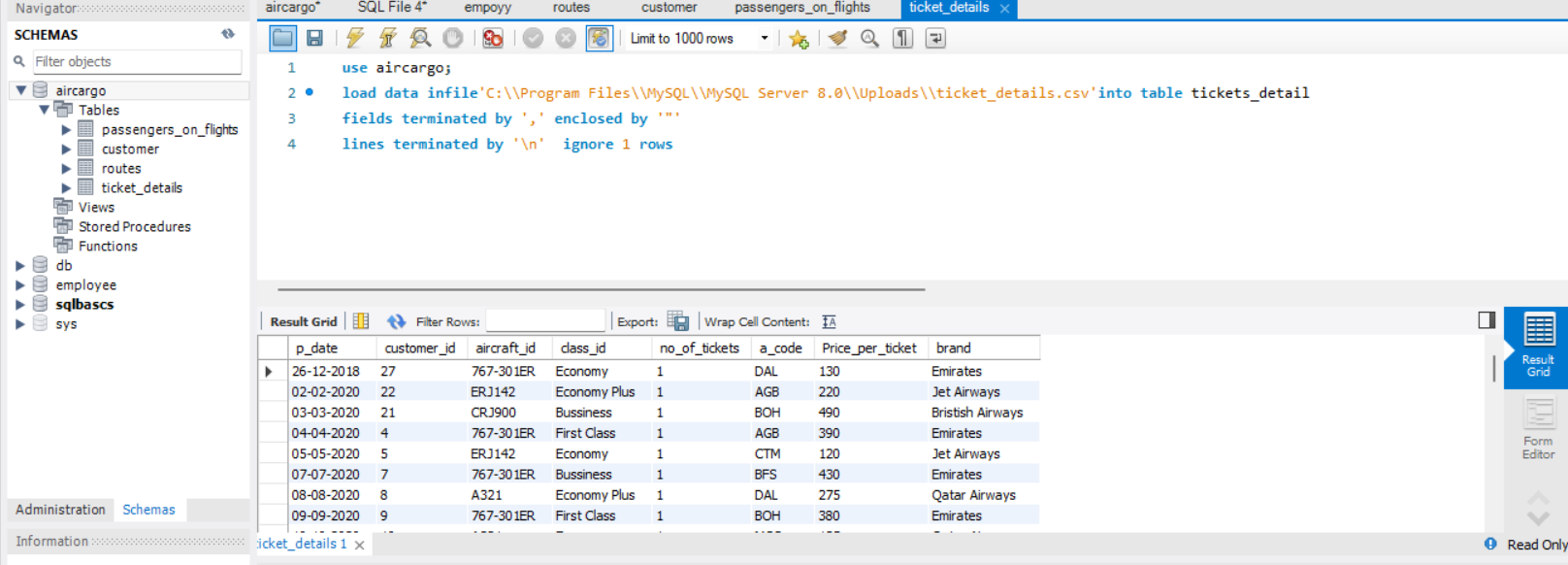
Describeaircargo.tickets\_detail;

use aircargo;

load data infile'C:\\Program Files\\MySQL\\MySQL Server 8.0\\Uploads\\ticket\_details.csv'into table tickets\_detail

fields terminated by ',' enclosed by '"'

lines terminated by '\n' ignore 1 rows



**Routes**

use Aircargo;

create table routes(

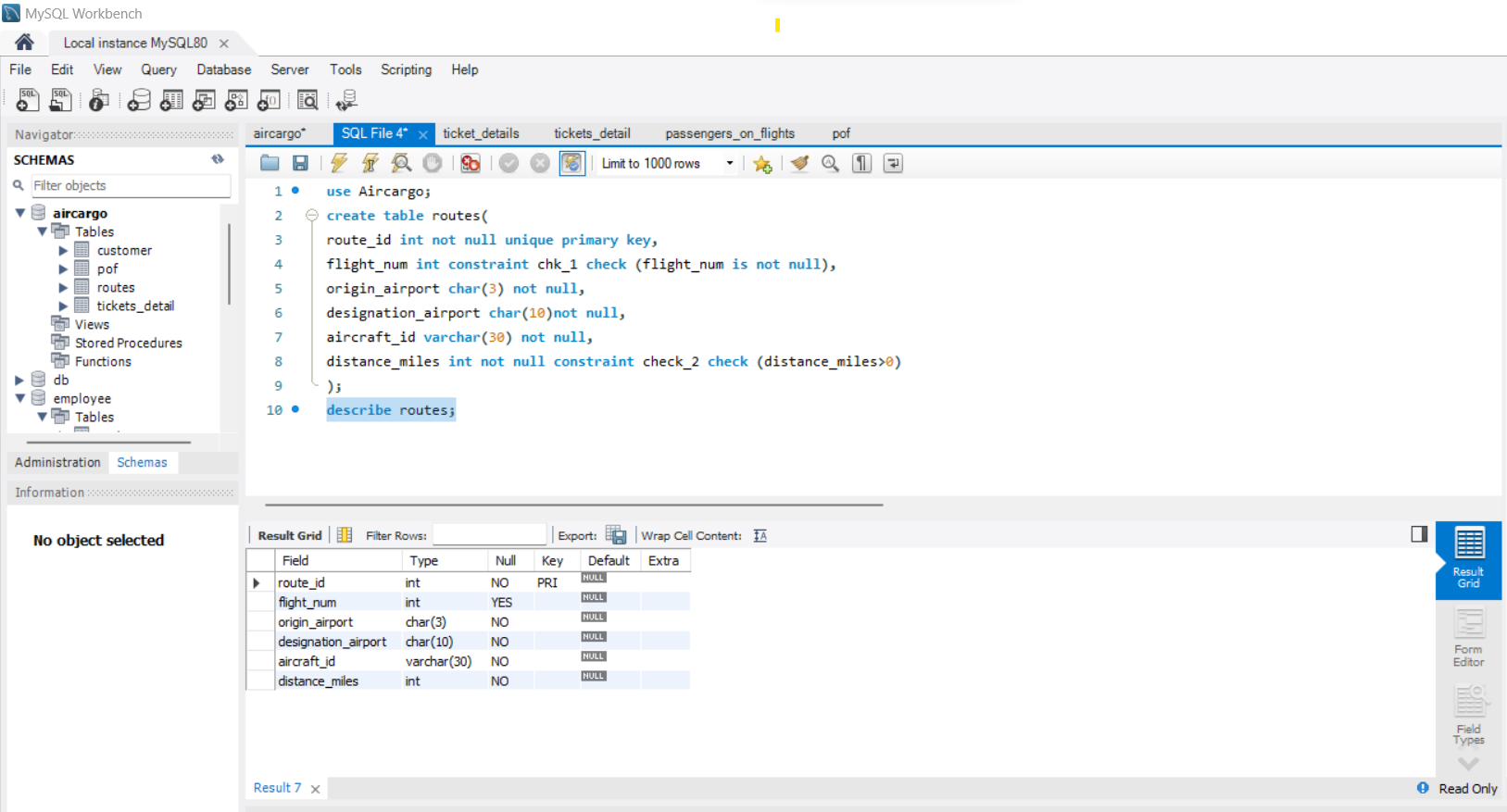
route\_id int not null unique primary key,

flight\_num int constraint chk\_1 check (flight\_num is not null),

origin\_airport char(3) not null,

designation\_airport char(10)not null,

aircraft\_id varchar(30) not null,distance\_miles int not null constraint check\_2 check (distance\_miles>0) Describe routes;



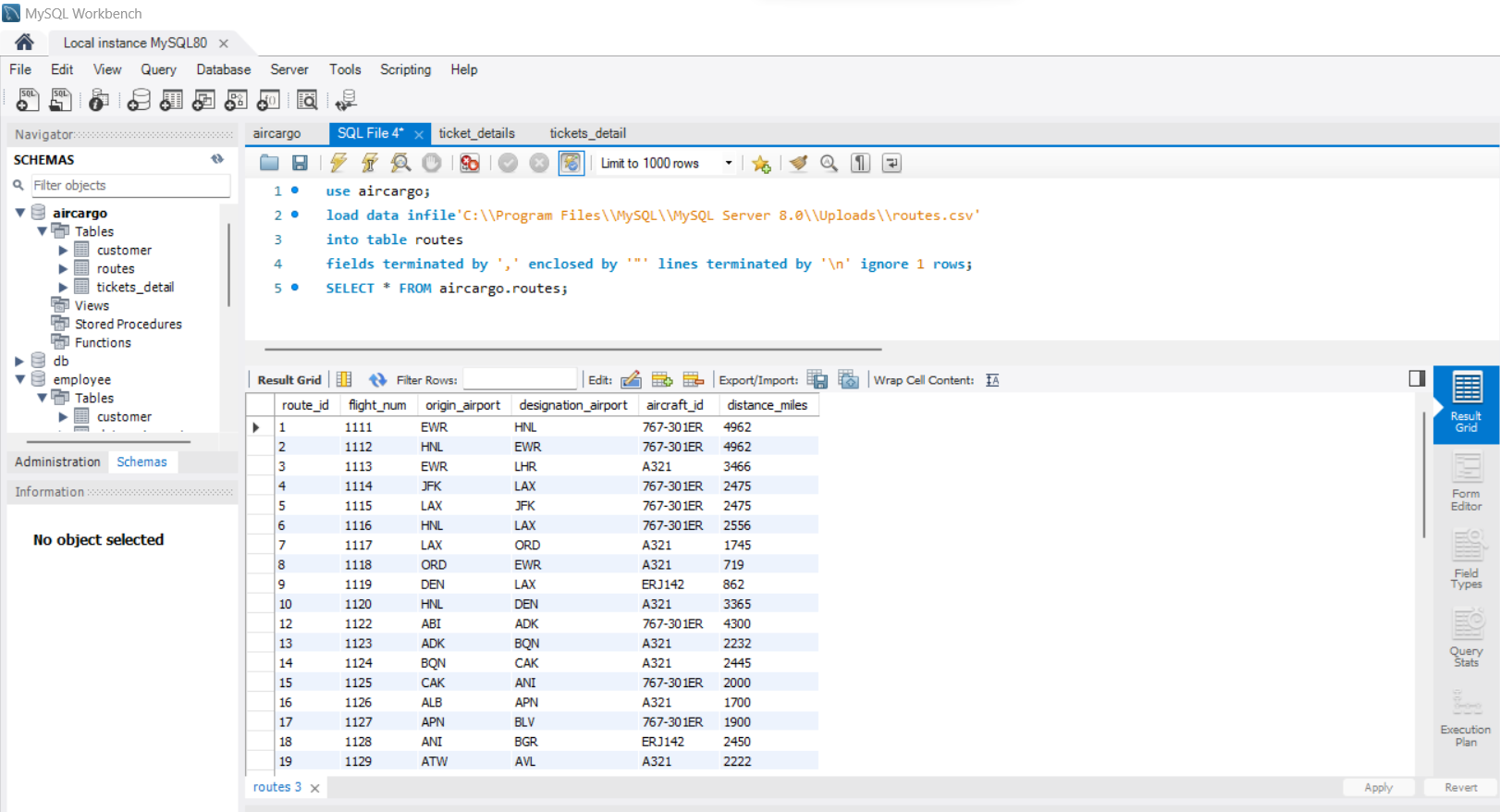
use aircargo;

loaddata infile'C:\\Program Files\\MySQL\\MySQL Server 8.0\\Uploads\\routes.csv'

into table routes

fields terminated by ',' enclosed by '"' lines terminated by '\n' ignore 1 rows;

SELECT \* FROM aircargo.routes;



**Passengers\_on\_flights**

use Aircargo;

create table pof(

pof\_id int auto\_increment primary key,

customer\_id int not null,

aircraft\_id varchar(10) not null,

route\_id int not null,

depart char(10)not null,

arrival char(3) not null,

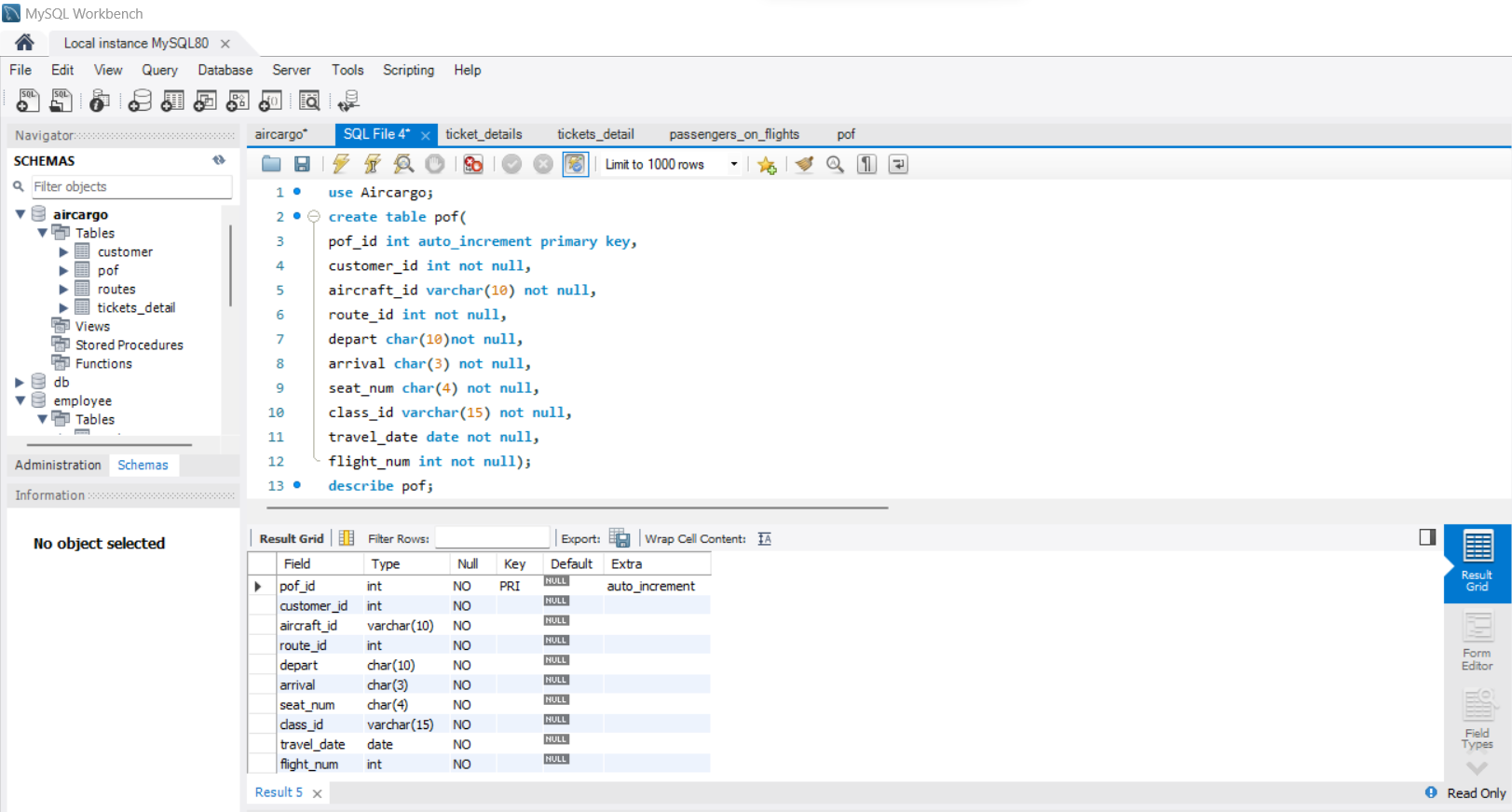
seat\_num char(4) not null,

class\_id varchar(15) not null,

travel\_date date not null,

flight\_num int not null);

describe pof;

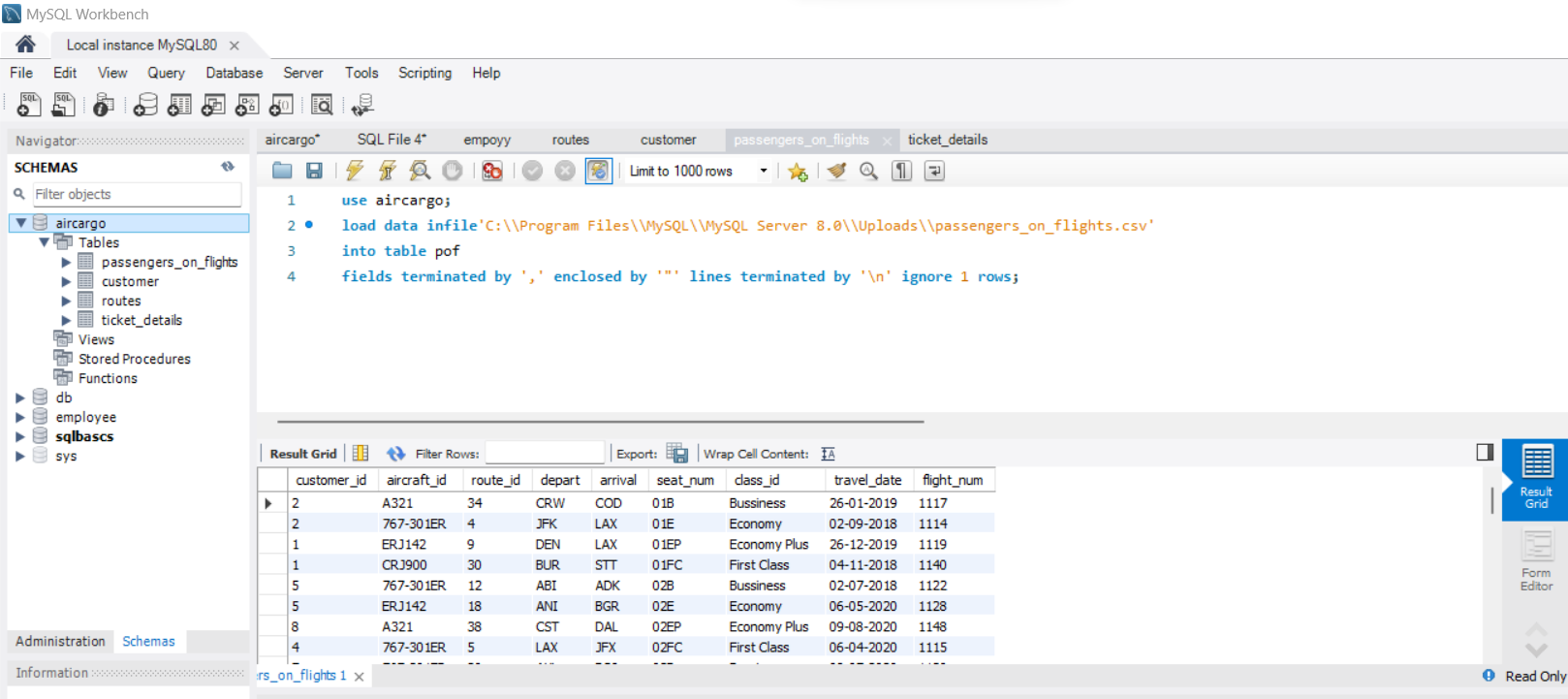
****

use aircargo;

load data infile'C:\\Program Files\\MySQL\\MySQL Server 8.0\\Uploads\\passengers\_on\_flights.csv'

into table pof

fields terminated by ',' enclosed by '"' lines terminated by '\n' ignore 1 rows;

****

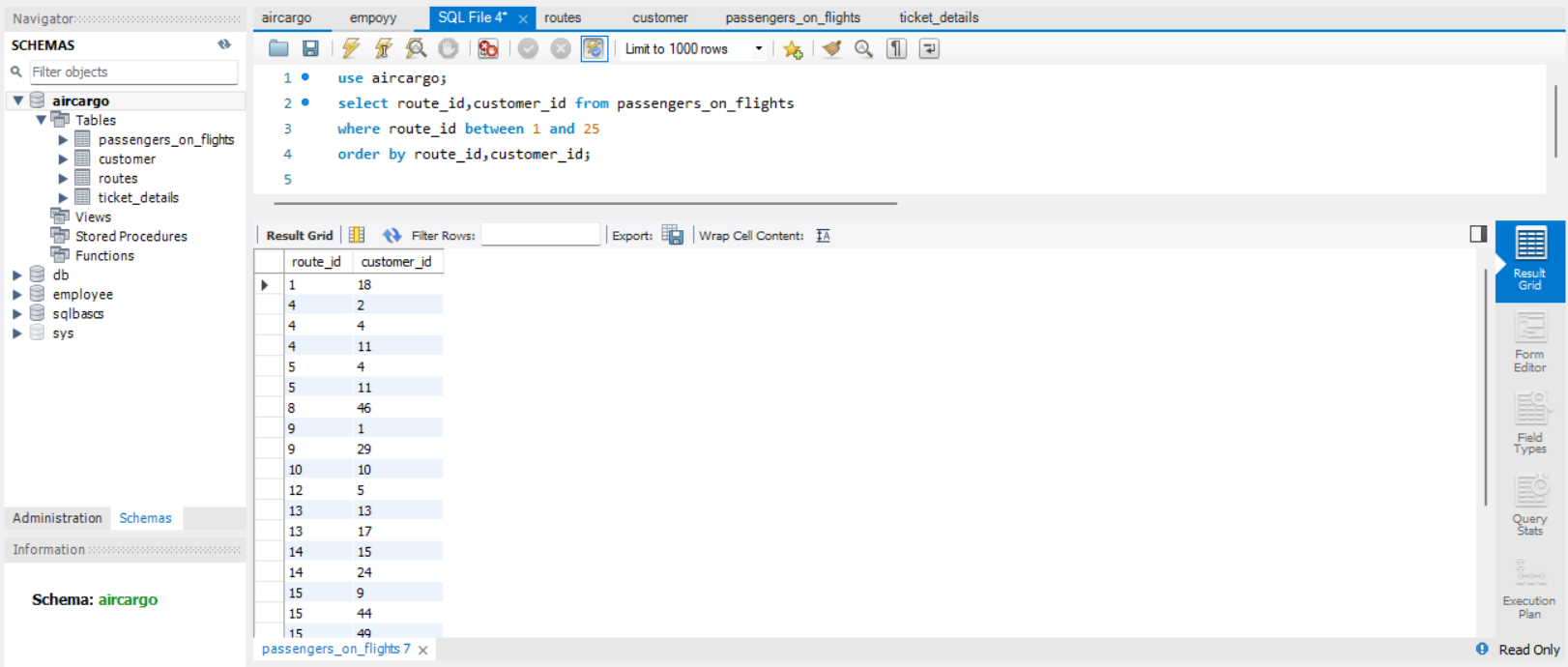
1. Write a query to display all the passengers (customers) who have travelled in routes 01 to 25. Take data from the passengers\_on\_flights table.

use aircargo;

select route\_id,customer\_id from passengers\_on\_flights

where route\_id between 1 and 25

order by route\_id,customer\_id;

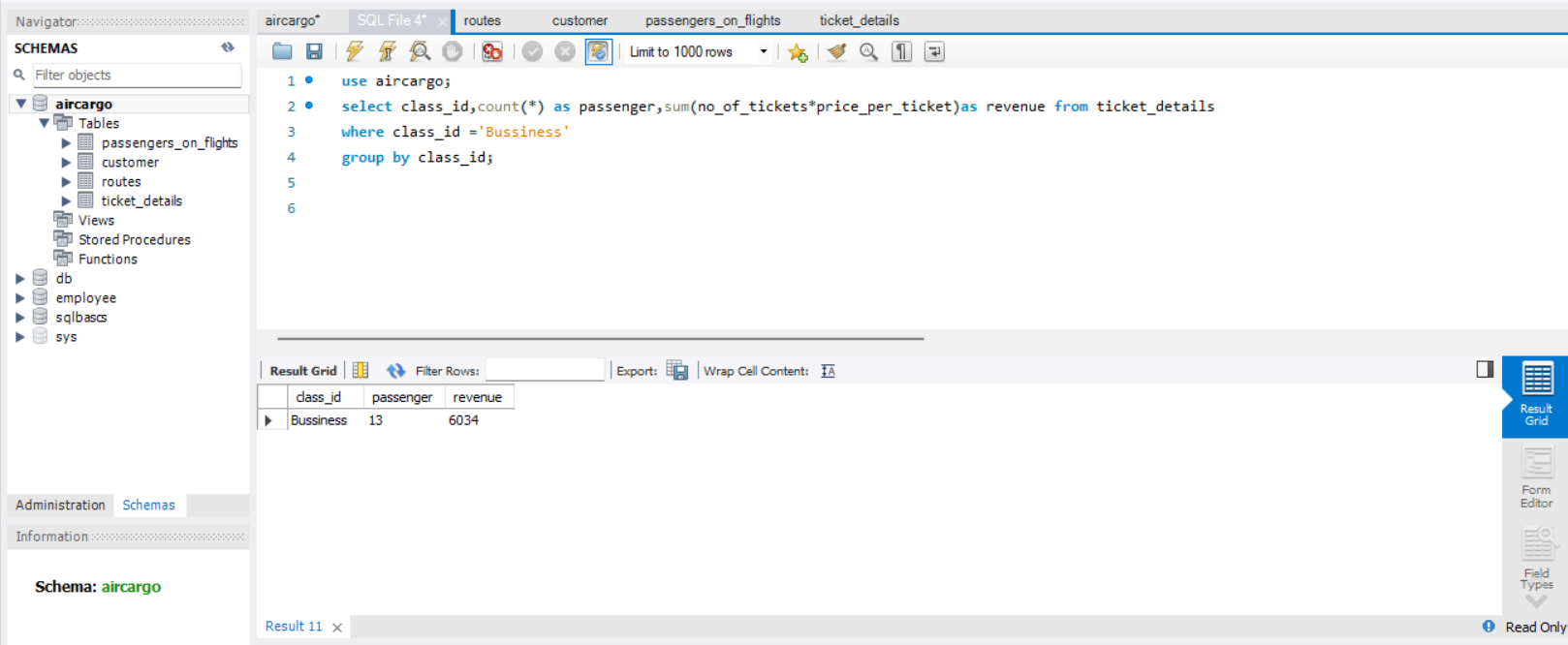


1. Write a query to identify the number of passengers and total revenue in business class from the ticket\_details table.

use aircargo;

select class\_id,count(\*) as passenger,sum(no\_of\_tickets\*price\_per\_ticket)as revenue from ticket\_details where class\_id ='Bussiness'

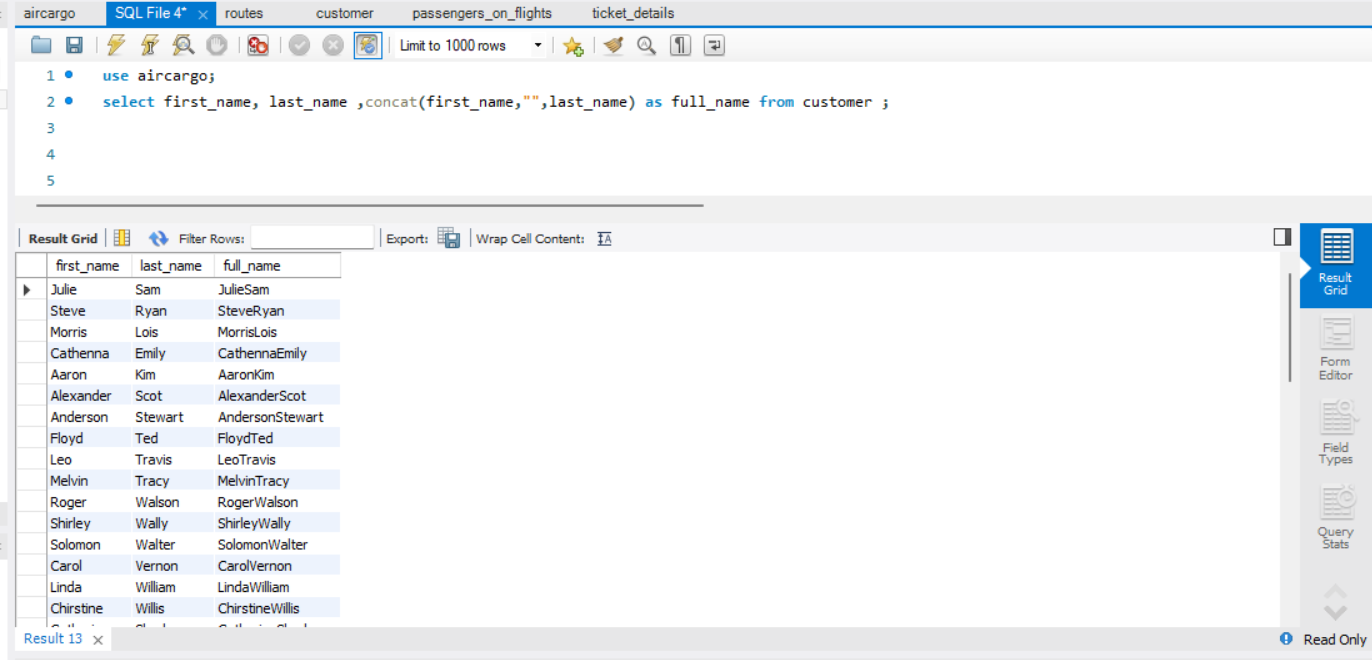
group by class\_id;



1. 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 ,concat(first\_name,"",last\_name) as full\_name

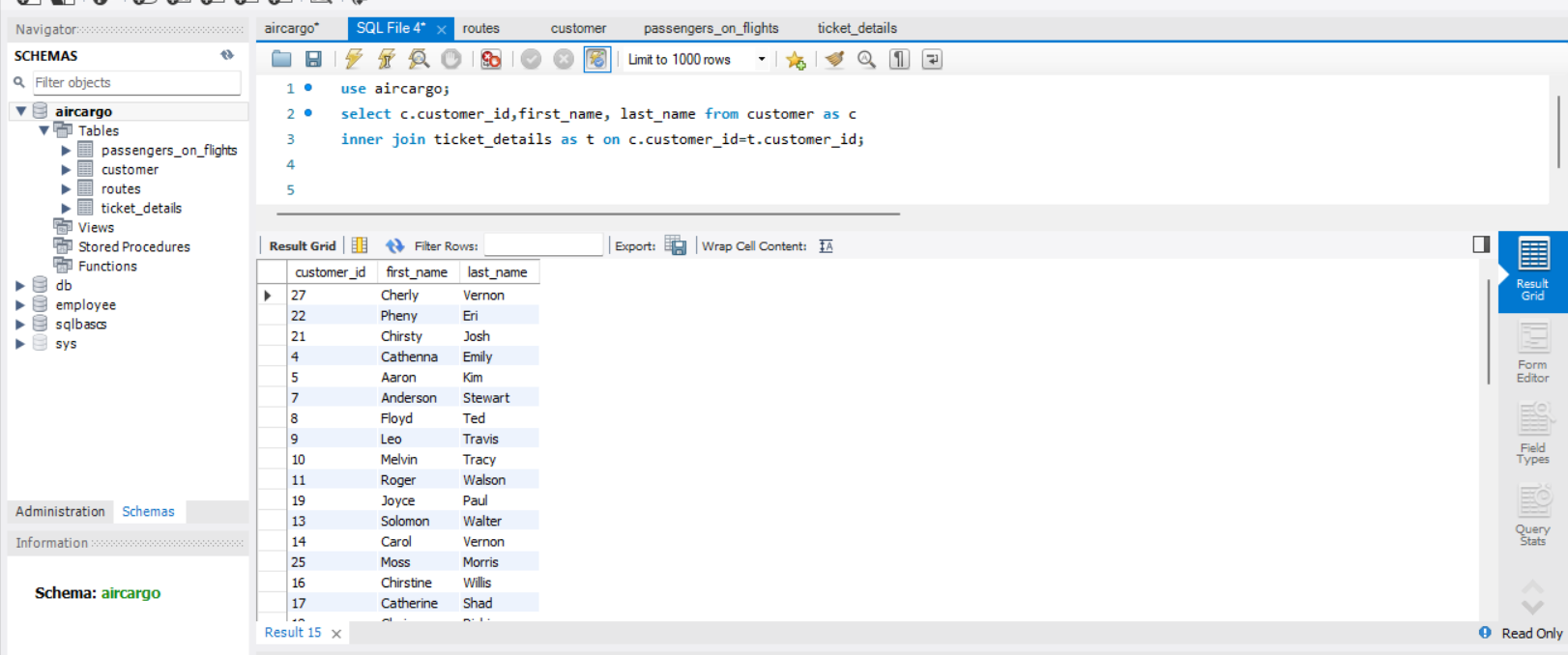
from customer ;



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

select c.customer\_id,first\_name, last\_name from customer as c

inner join ticket\_details as t on c.customer\_id=t.customer\_id;

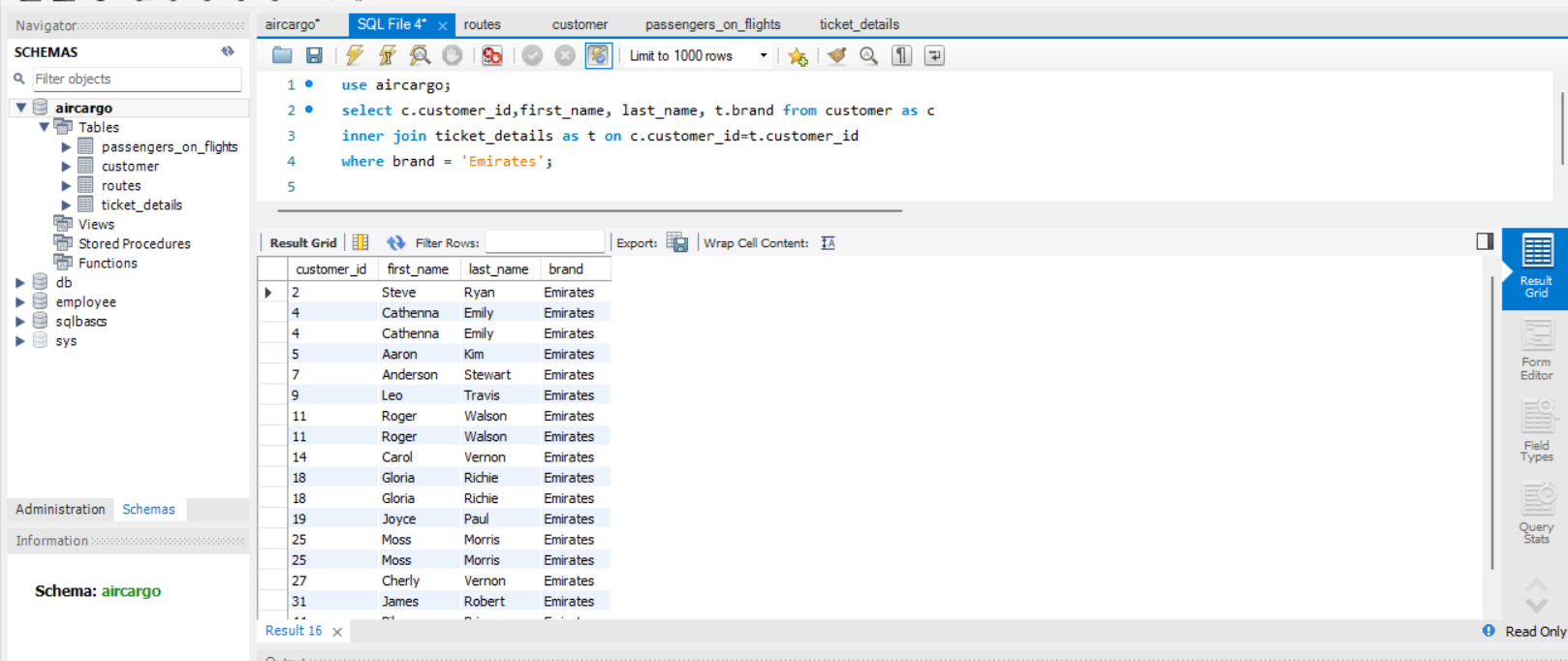


1. Write a query to identify the customer’s first name and last name based on their customer ID and brand (Emirates) from the ticket\_details table.

select c.customer\_id,first\_name, last\_name, t.brand from customer as c

inner join ticket\_details as t on c.customer\_id=t.customer\_id

where brand = 'Emirates';



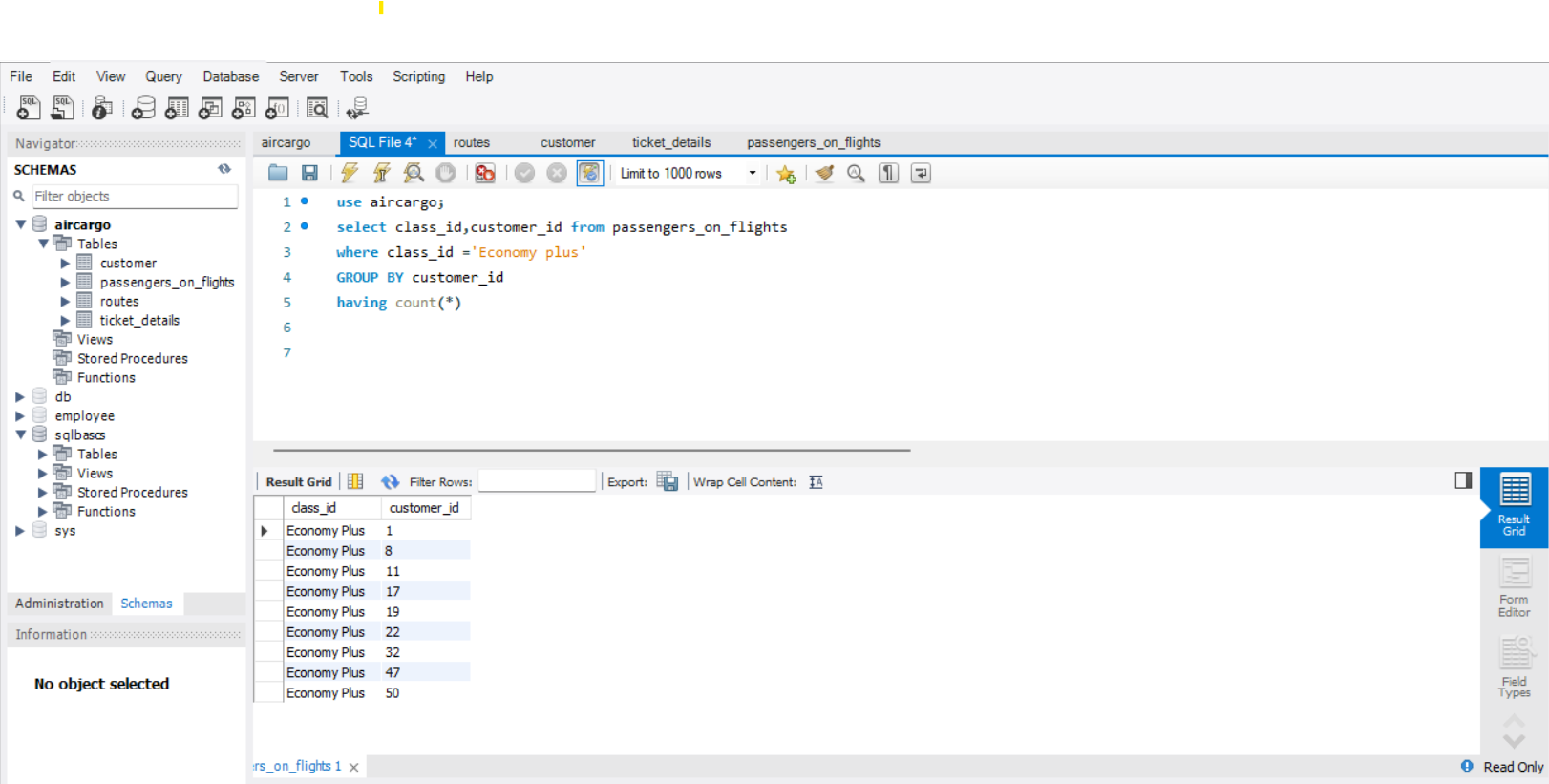
1. Write a query to identify the customers who have travelled by *Economy Plus* class using Group By and Having clause on the passengers\_on\_flights table.

use aircargo;

select class\_id,customer\_id from passengers\_on\_flights

where class\_id ='Economy plus'

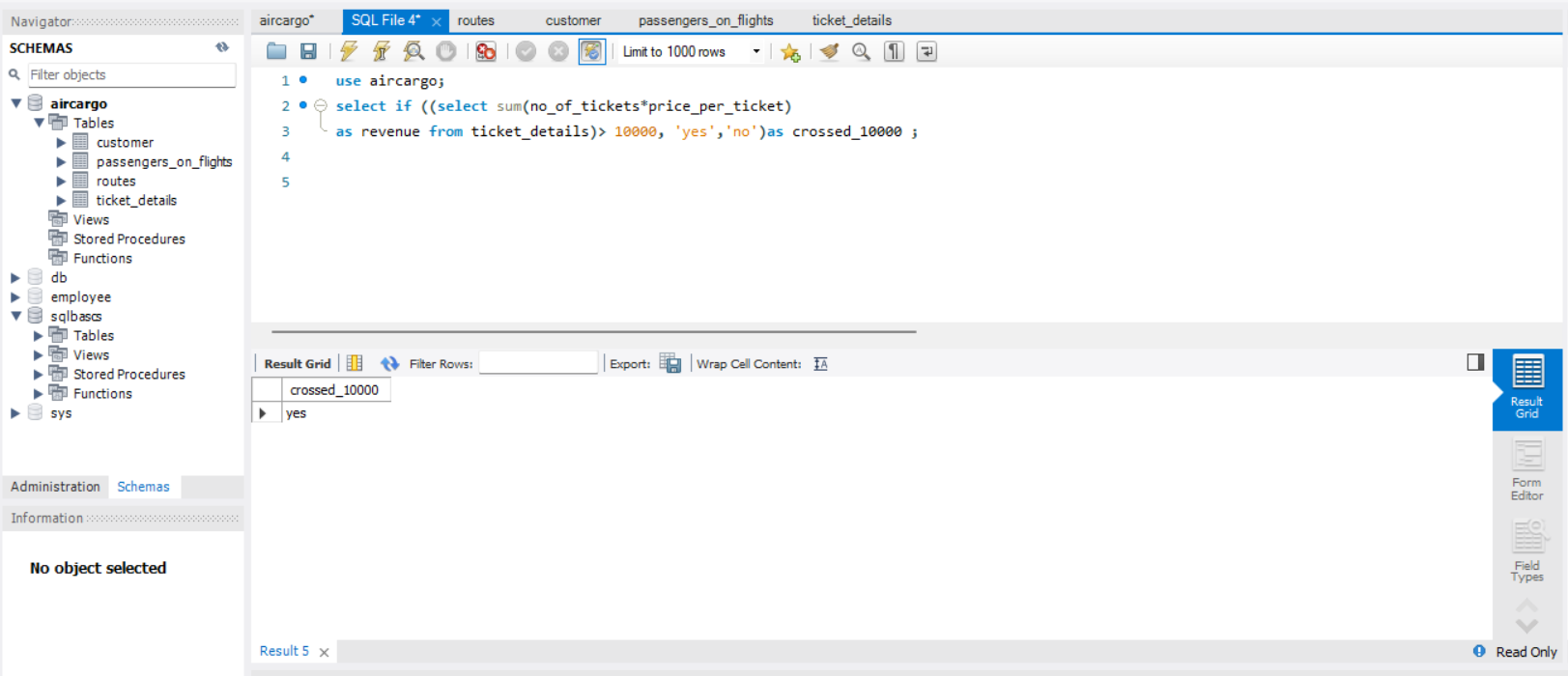
GROUP BY customer\_id having count(\*)



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

select if ((select sum(no\_of\_tickets\*price\_per\_ticket)

as revenue from ticket\_details)> 10000, 'yes','no')as crossed\_10000 ;

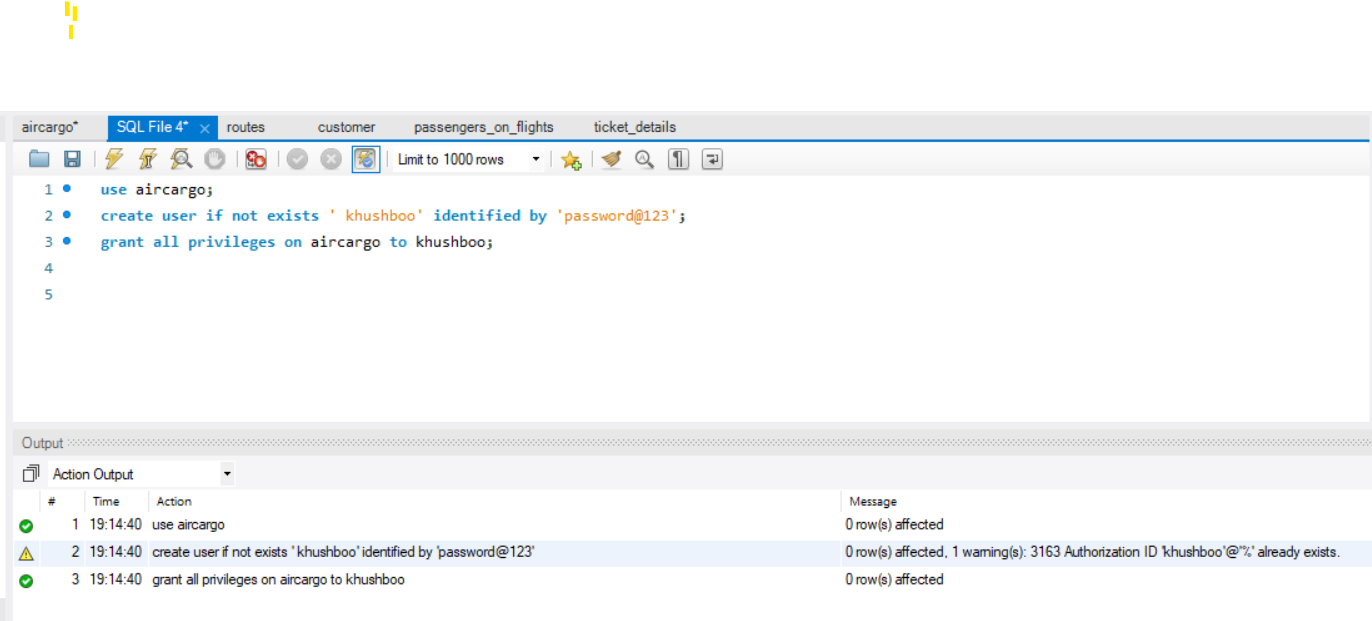


1. Write a query to create and grant access to a new user to perform operations on a database.

use aircargo;

create user if not exists ' khushboo' identified by 'password@123';

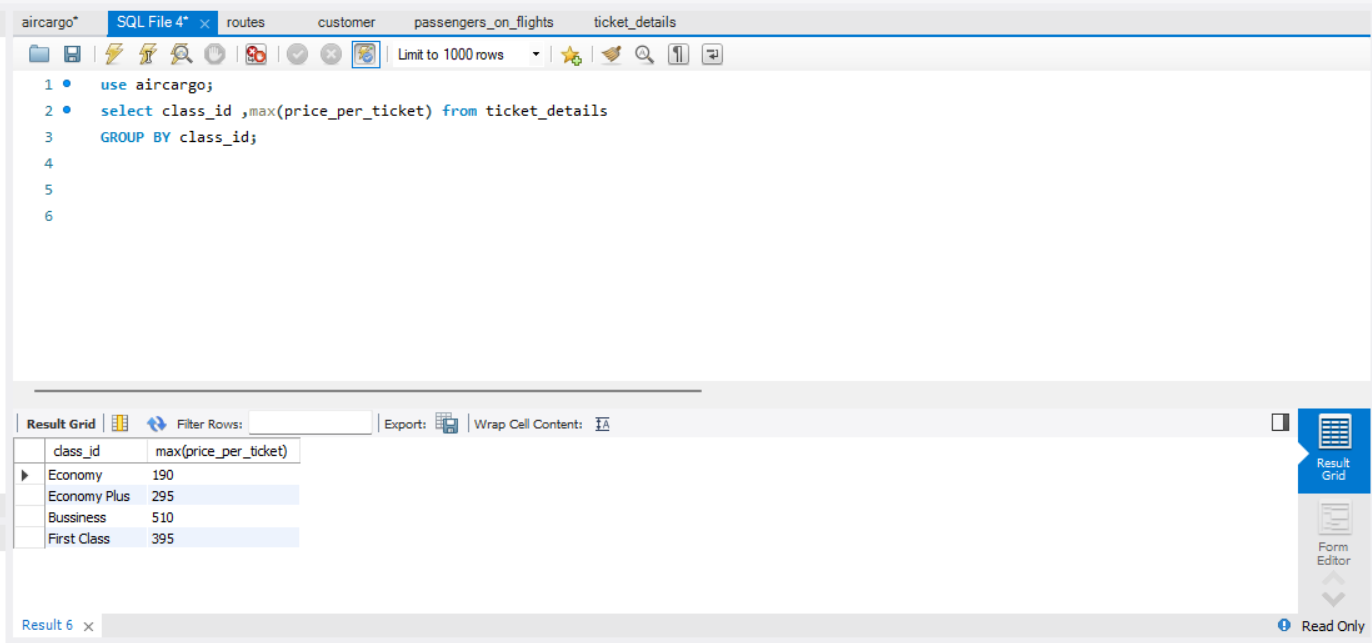
grant all privileges on aircargo to khushboo;



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

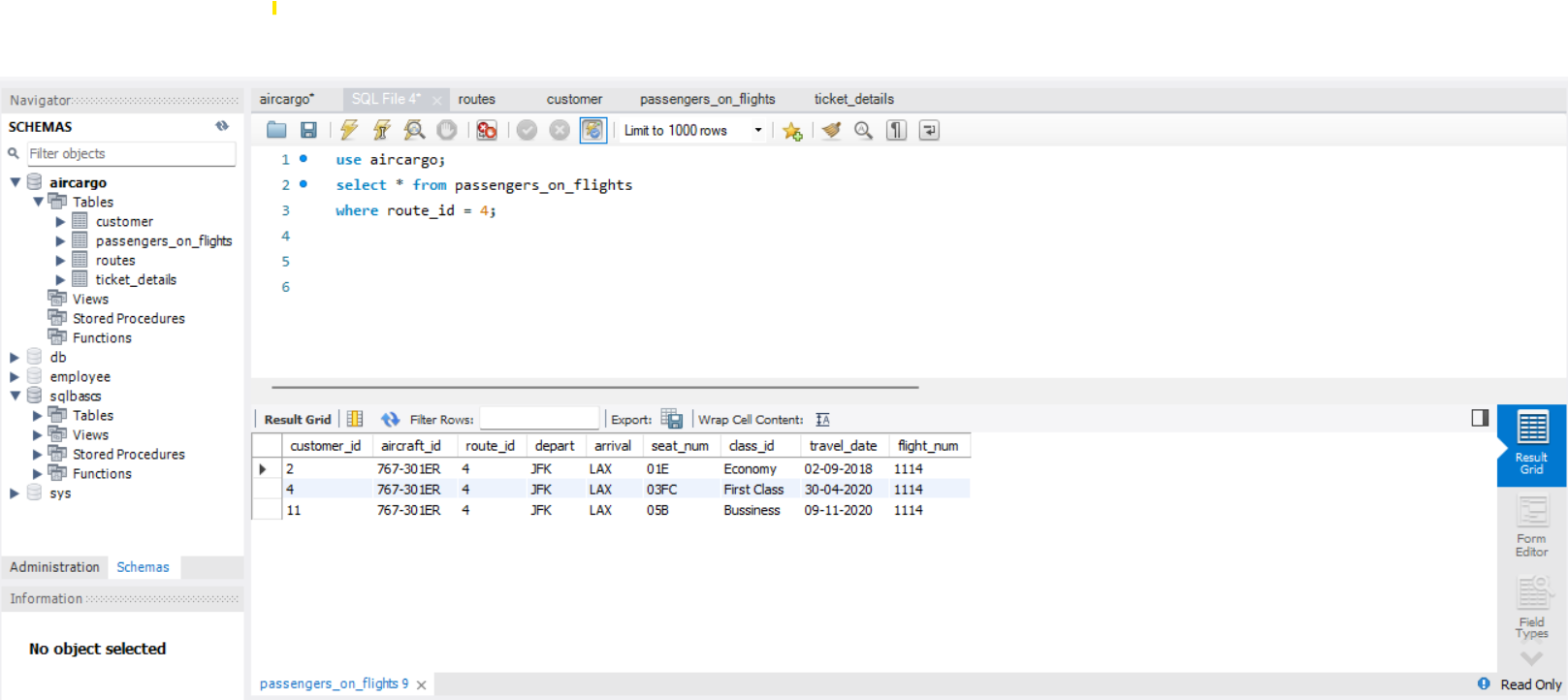
use aircargo;

select class\_id ,max(price\_per\_ticket) from ticket\_details

GROUP BY class\_id; 

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

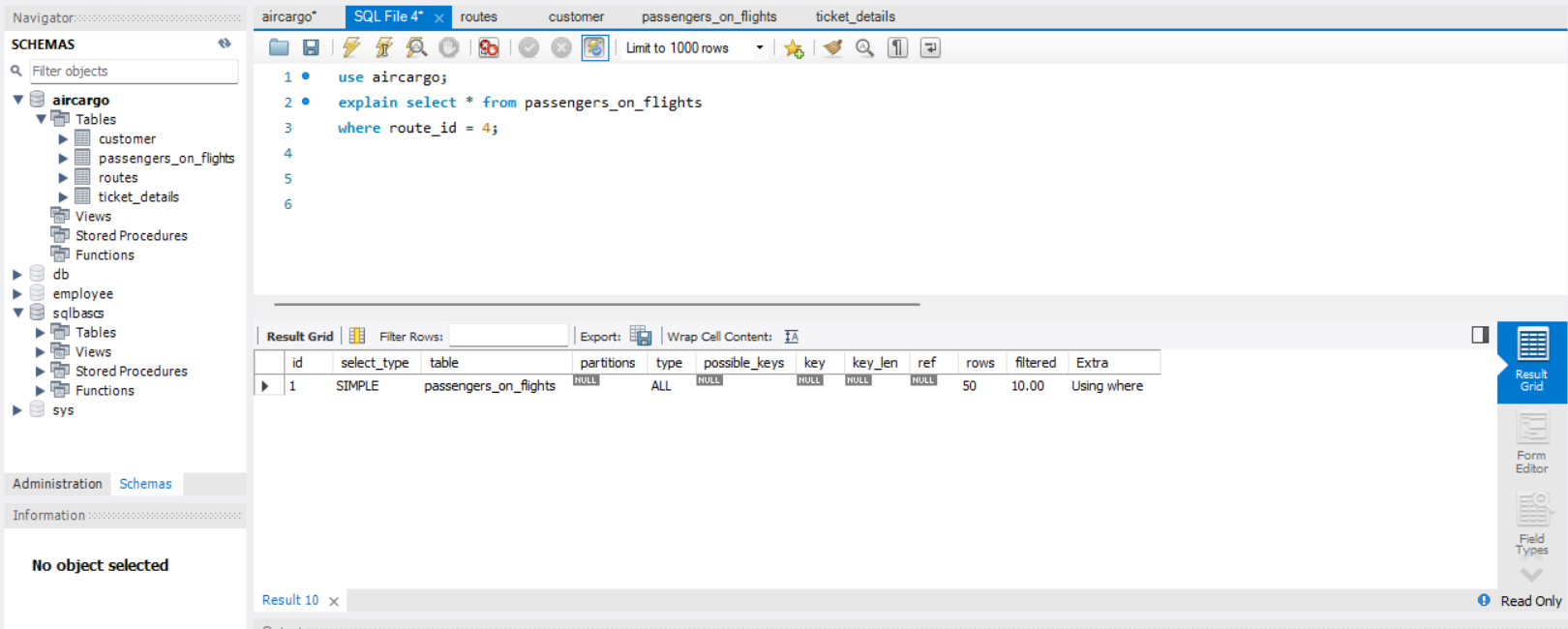
select \* from passengers\_on\_flights

where route\_id = 4; 

1. 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;

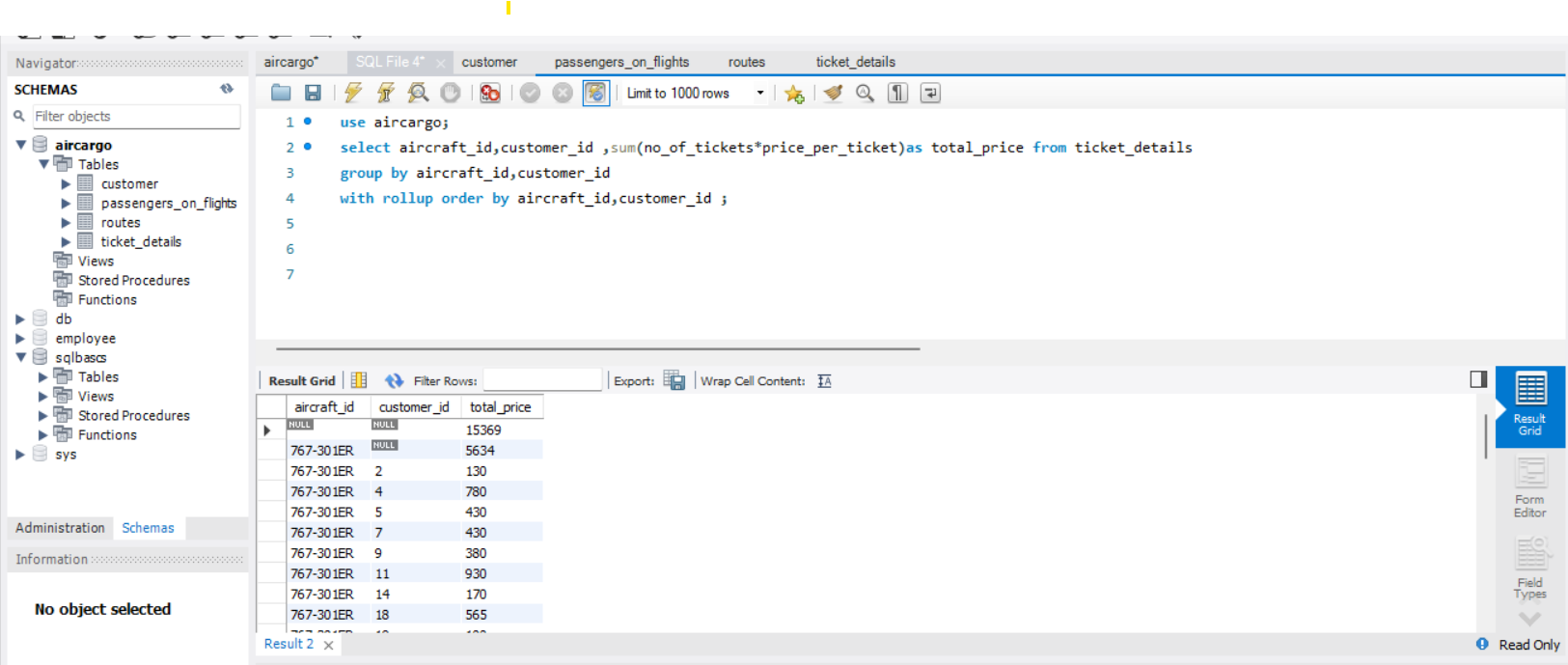


1. Write a query to calculate the total price of all tickets booked by a customer across different aircraft IDs using rollup function.

use aircargo;

select aircraft\_id,customer\_id ,sum(no\_of\_tickets\*price\_per\_ticket)as total\_price from ticket\_details

group by aircraft\_id,customer\_id

with rollup order by aircraft\_id,customer\_id ; 

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

use aircargo;

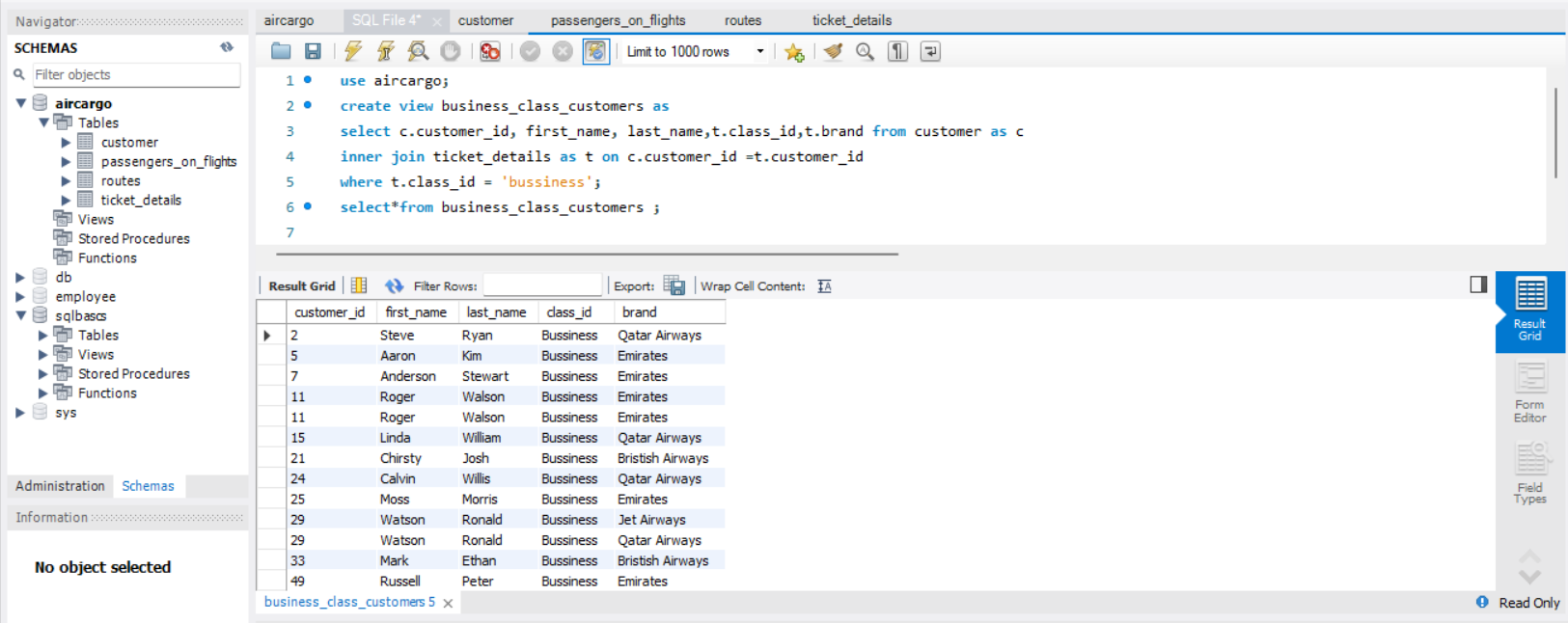
create view business\_class\_customers as

select c.customer\_id, first\_name, last\_name,t.class\_id,t.brand from customer as c

inner join ticket\_details as t on c.customer\_id =t.customer\_id

where t.class\_id = 'bussiness';

select\*from business\_class\_customers ;



1. 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.

use aircargo;

delimiter &&

create procedure get\_pasenger\_by\_route\_range(

IN start\_route int,

IN end\_route int)

Begin declare error\_msg varchar(260);

if not exists (select 1 from information\_schema.tables where table\_name = 'passengers\_on\_flights') then

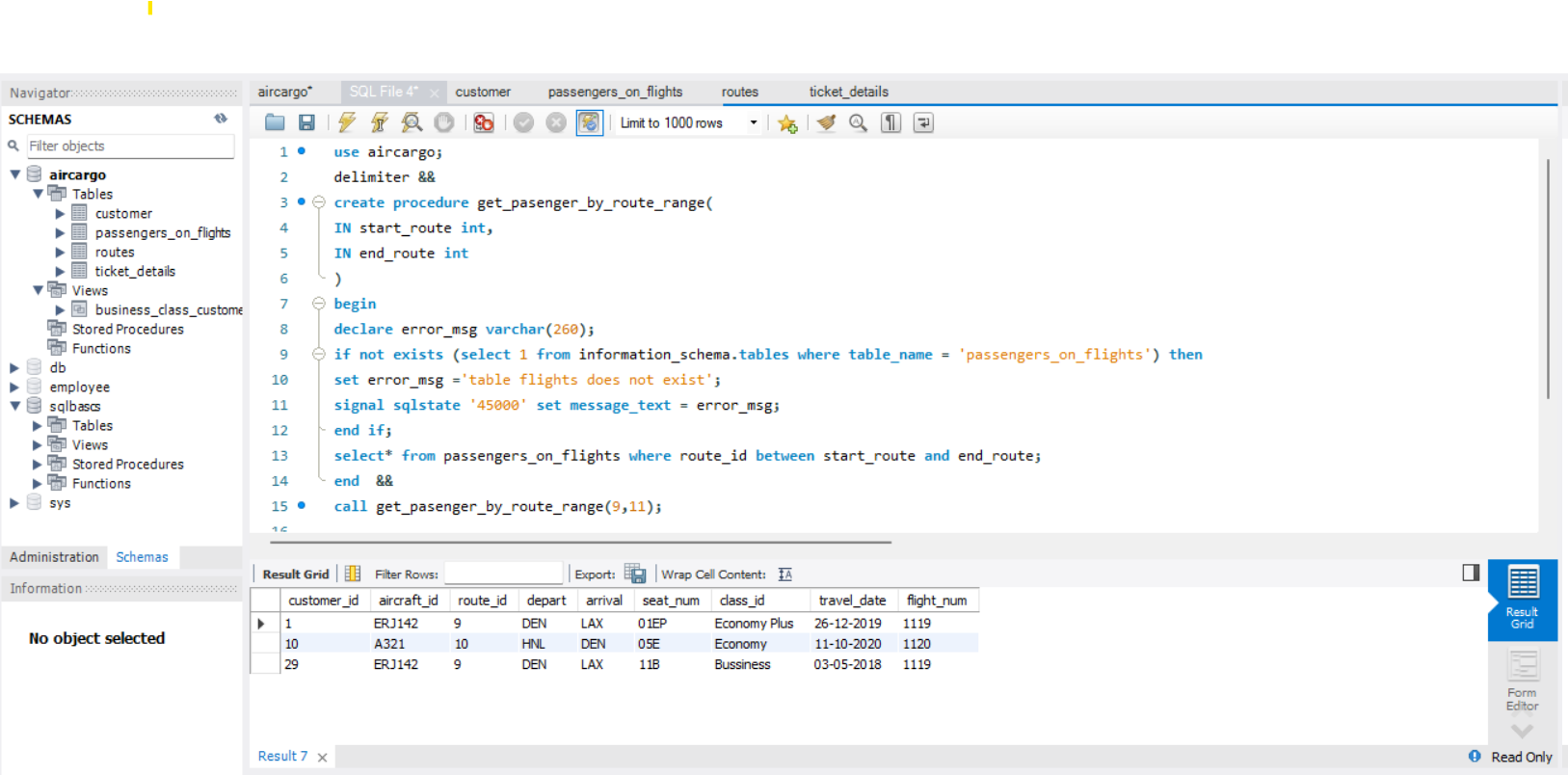
set error\_msg ='table flights does not exist';

signal sqlstate '45000' set message\_text = error\_msg;

end if;select\* from passengers\_on\_flights where route\_id between start\_route and end\_route;

end &&

call get\_pasenger\_by\_route\_range(9,11);



1. 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.

use aircargo;

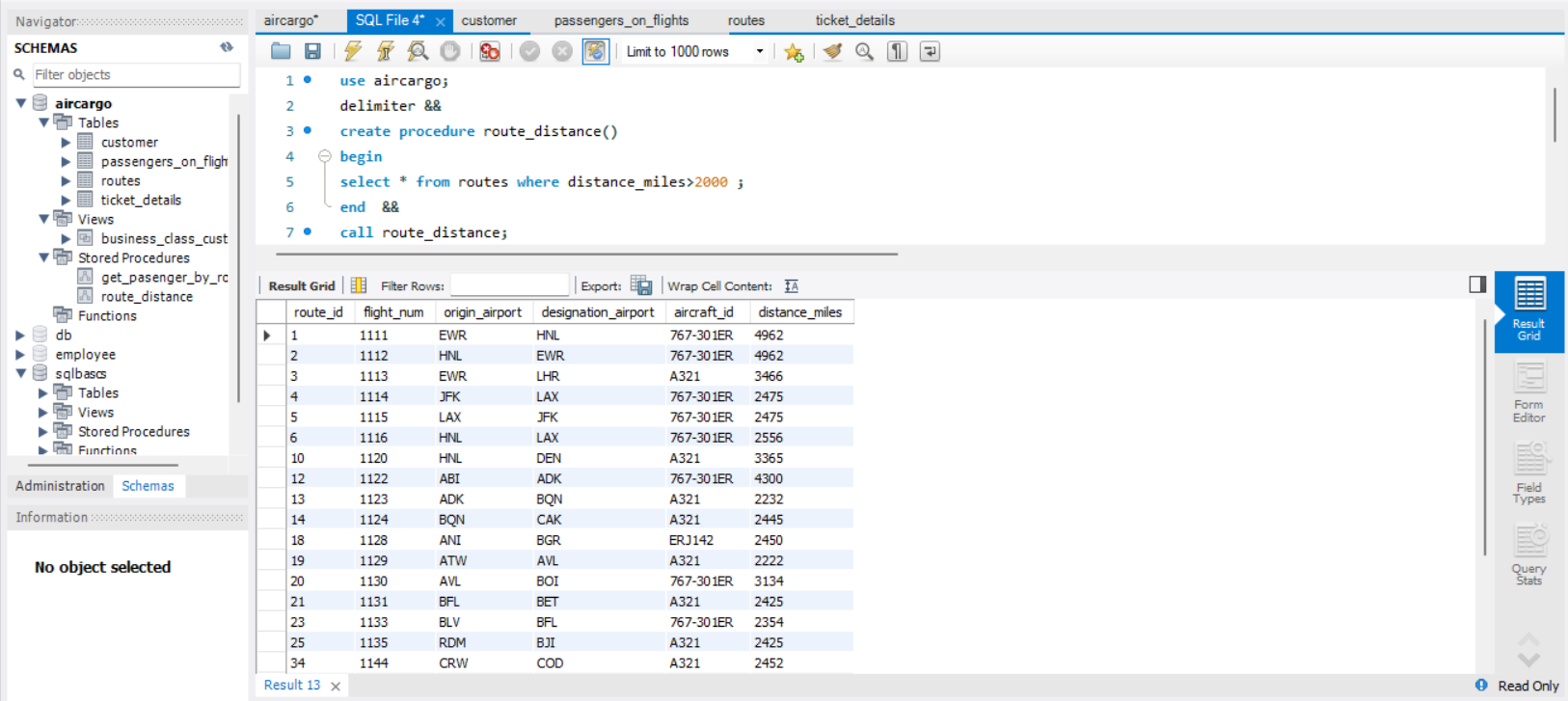
delimiter &&

create procedure route\_distance()

begin select \* from routes where distance\_miles>2000 ;

end &&

call route\_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.

delimiter &&

create procedure fight\_distance\_category()

begin

declare distance\_category varchar(20);

select flight\_num,distance\_miles,

case

when distance\_miles between 0 and 2000 then "SDT"

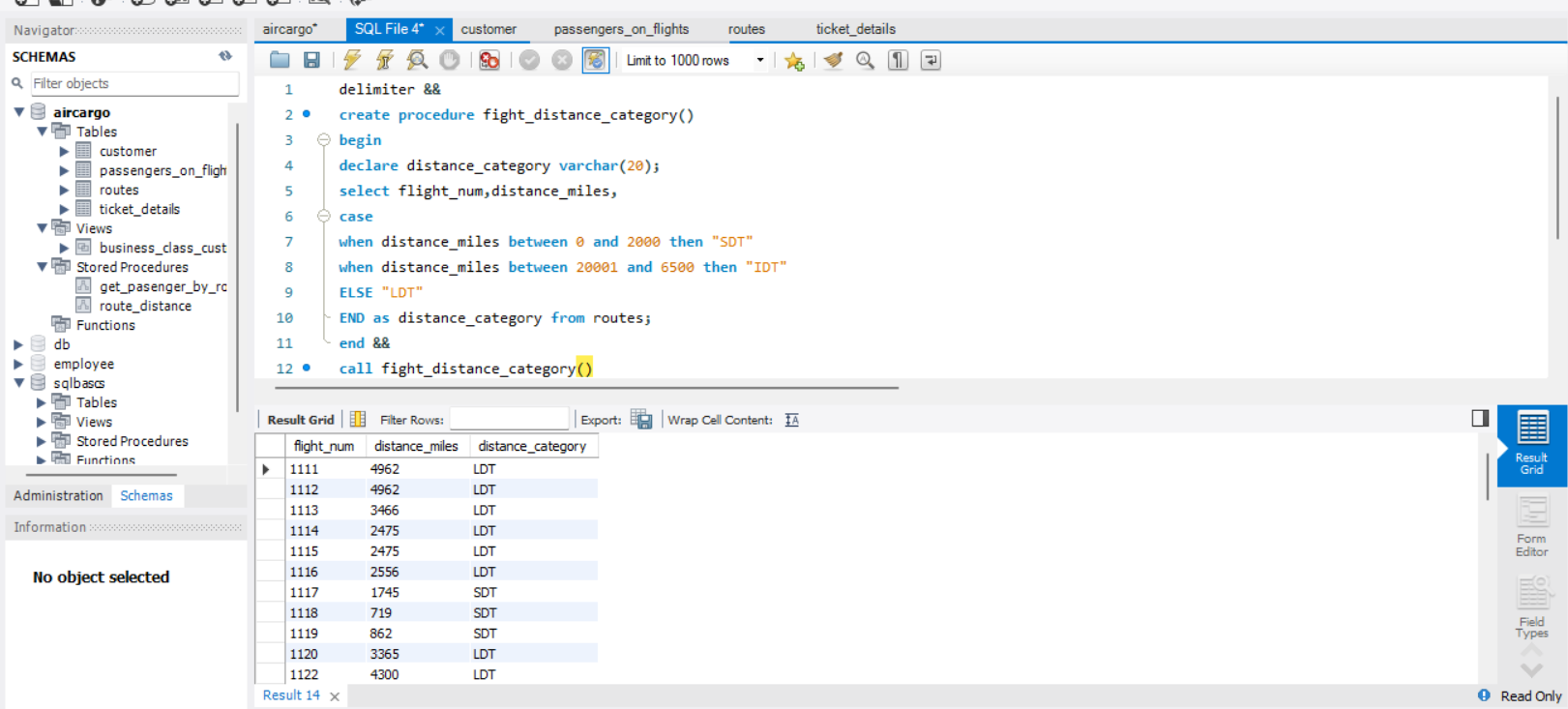
when distance\_miles between 20001 and 6500 then "IDT"

ELSE "LDT"

END as distance\_category from routes;

end &&

call fight\_distance\_category()



1. 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

use aircargo;

delimiter &&

create procedure complimentary\_services ()

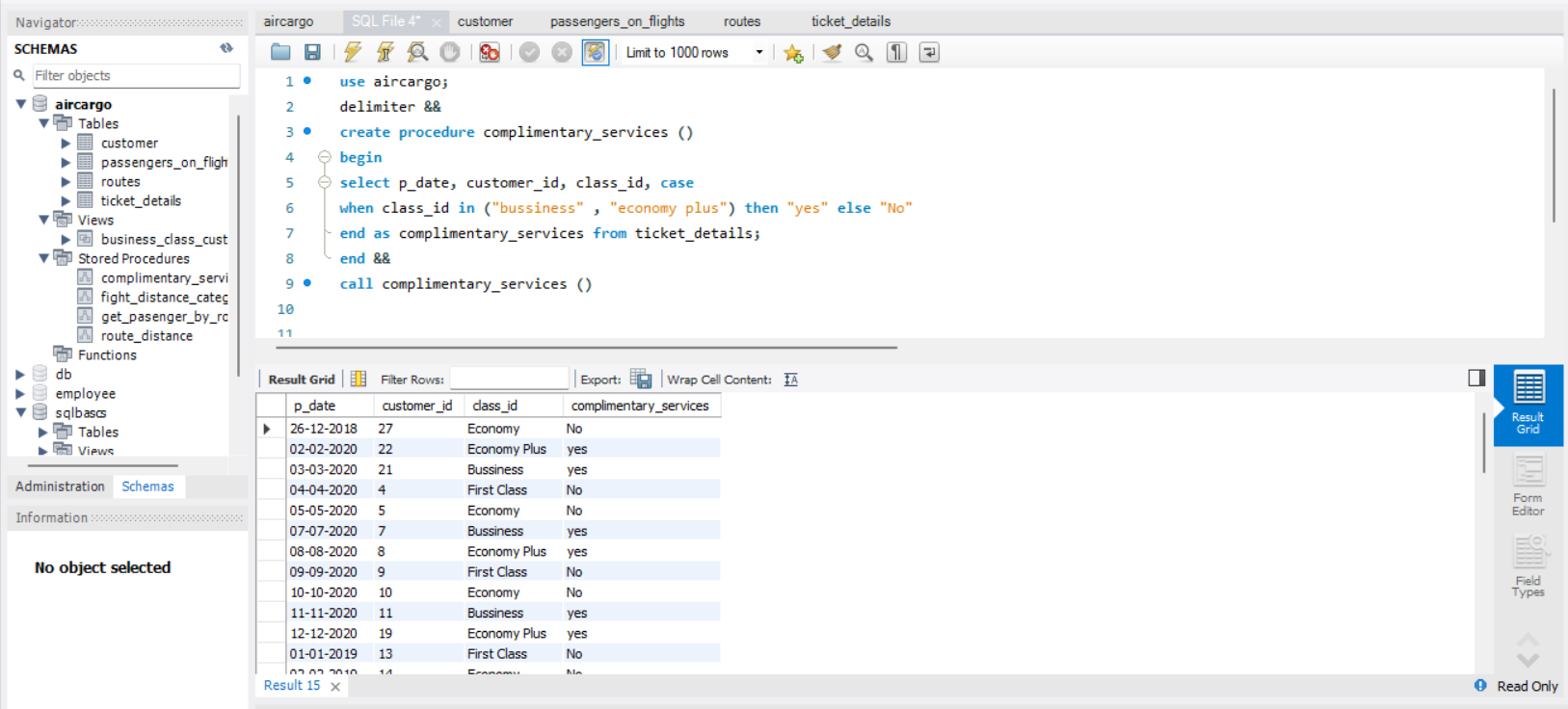
begin select p\_date, customer\_id, class\_id, case

when class\_id in ("bussiness" , "economy plus") then "yes" else "No"

end as complimentary\_services from ticket\_details;

end &&

call complimentary\_services ()

**

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

