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

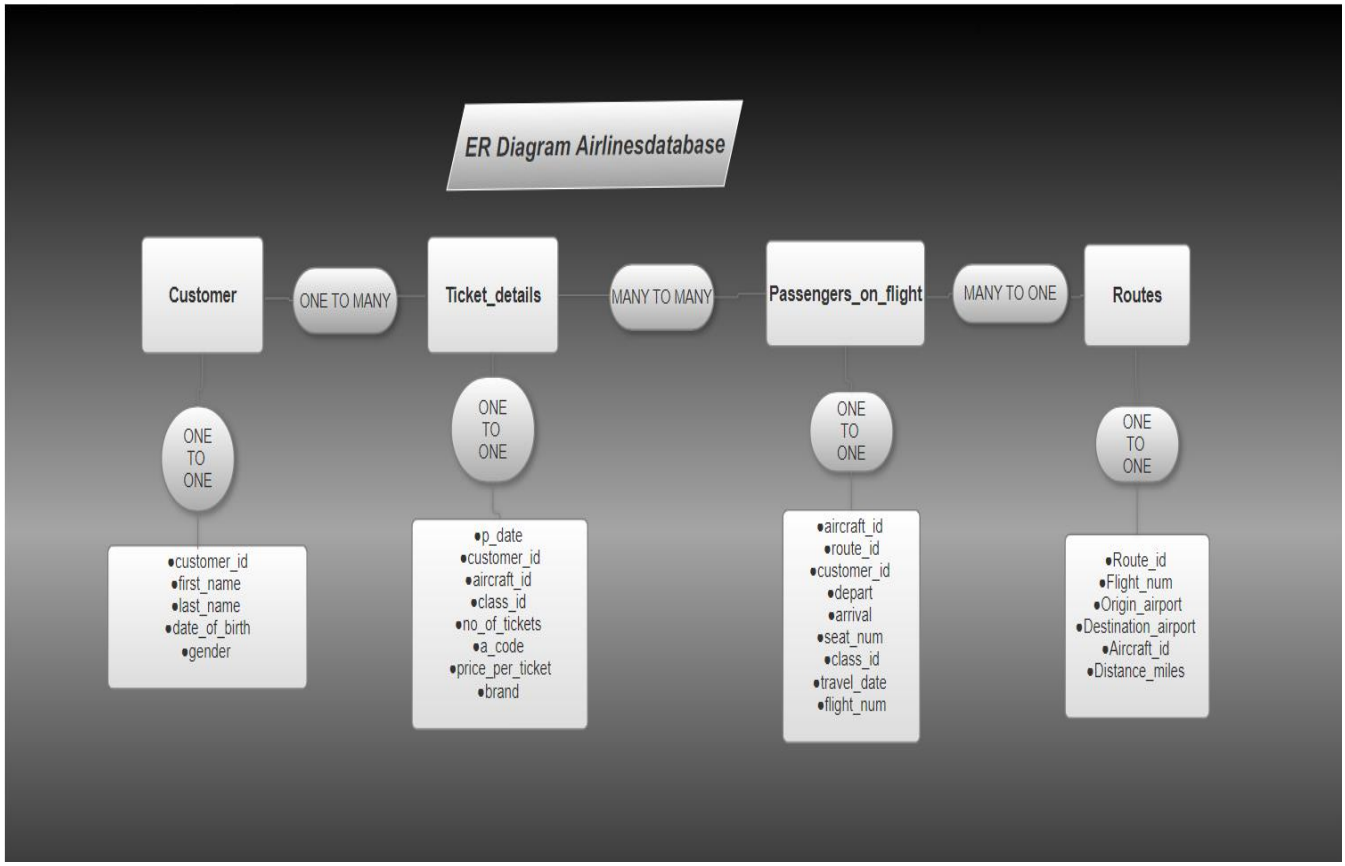
- p_date – Ticket purchase date
- customer_id – ID of the customer
- aircraft_id – ID of each aircraft in a brand
- class_id – ID of travel class
- no_of_tickets – Number of tickets purchased
- a_code – Code of each airport
- price_per_ticket – Price of a ticket
- brand – Aviation service provider for 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:

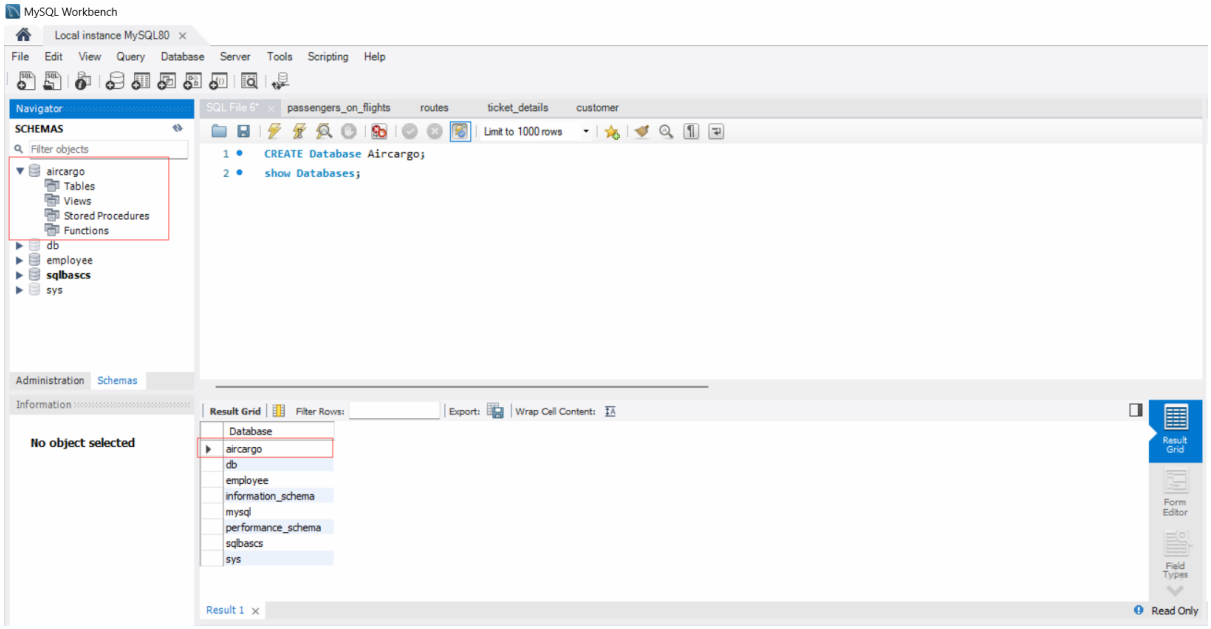
1. Create an ER diagram for the given airlines database.



2. 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,

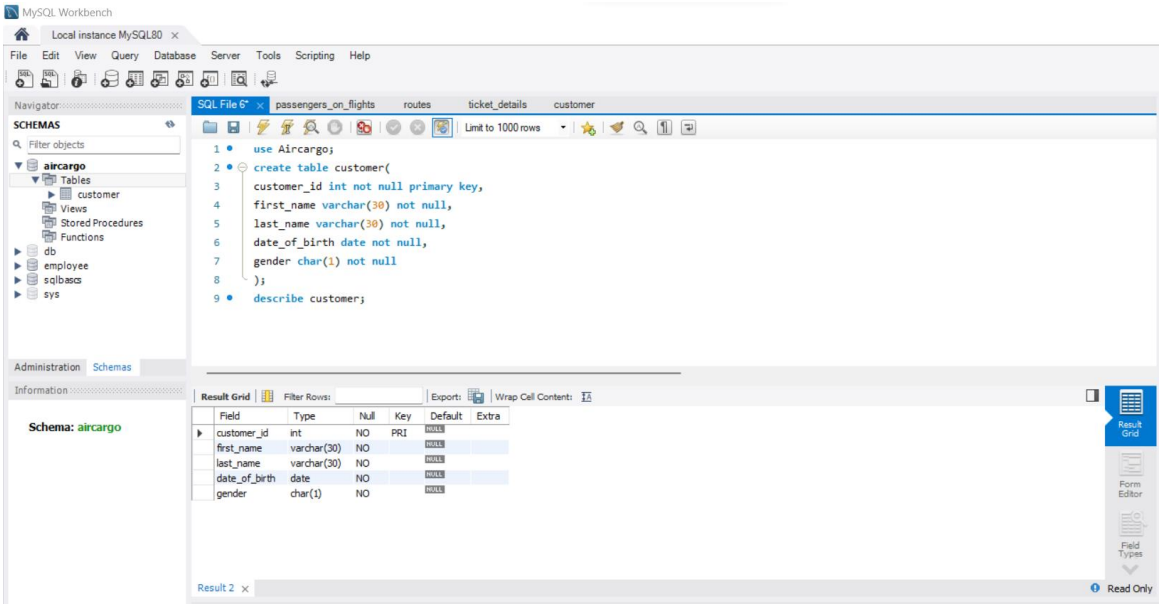
aircraft_id, and distance_miles. Implement the check constraint for the flight number and unique constraint for the route_id fields. Also, make sure that the distance miles field is greater than 0.

CREATE Database Aircargo;
show Databases;

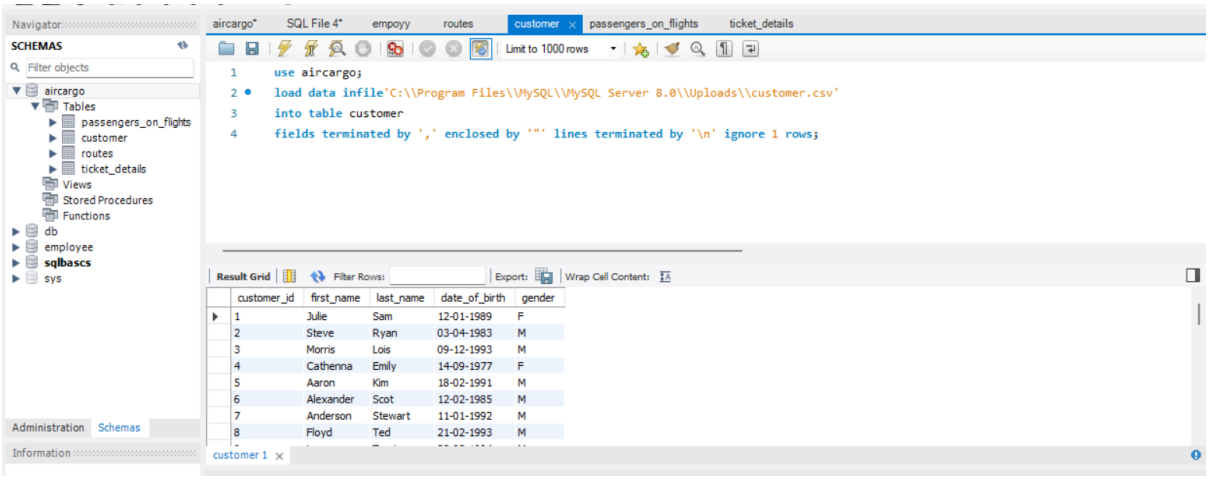


customer

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;

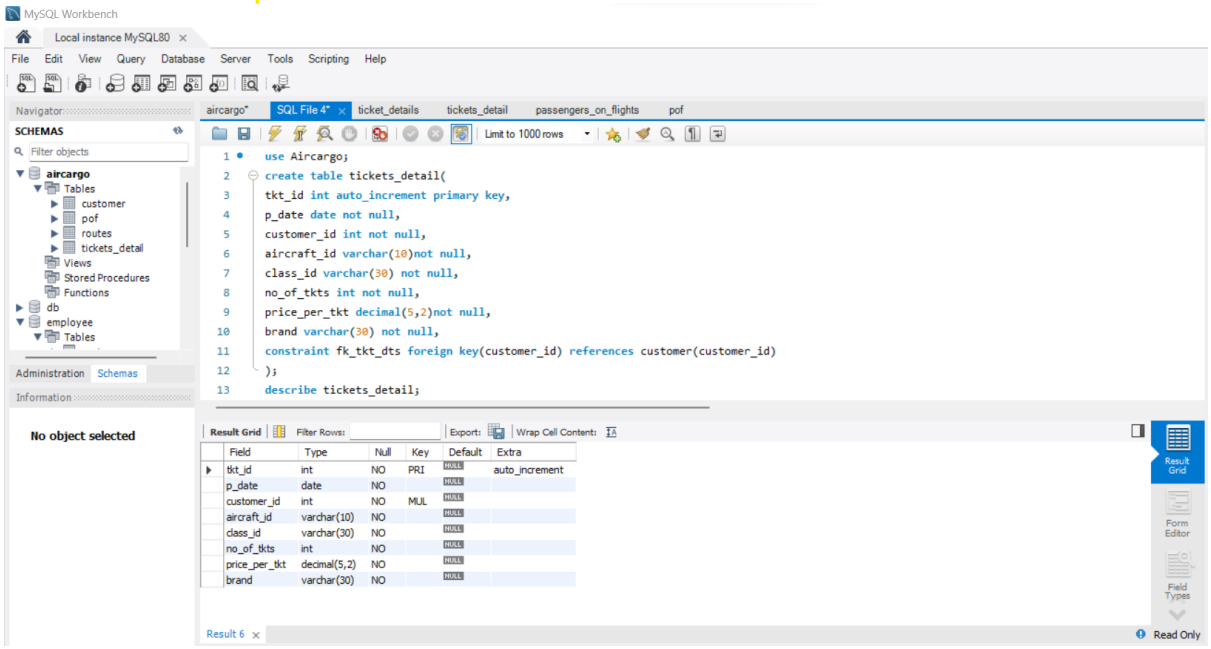


use aircargo;
load data infile'C:\\Program Files\\MySQL\\MySQL Server 8.0\\Uploads\\customer.csv'
into table customer
fields terminated by ',' enclosed by '"' lines terminated by '\n' ignore 1 rows;

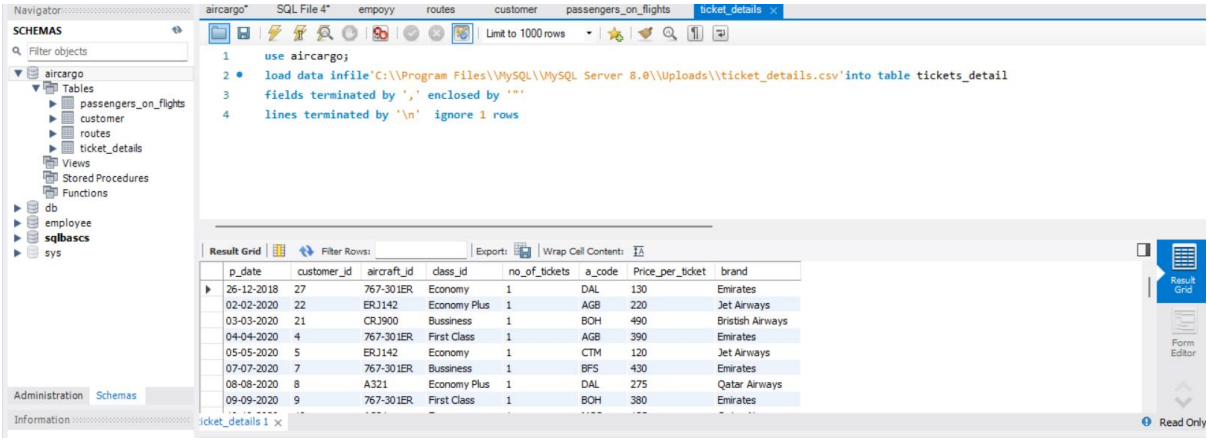


Tickets_detail

```
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;
```

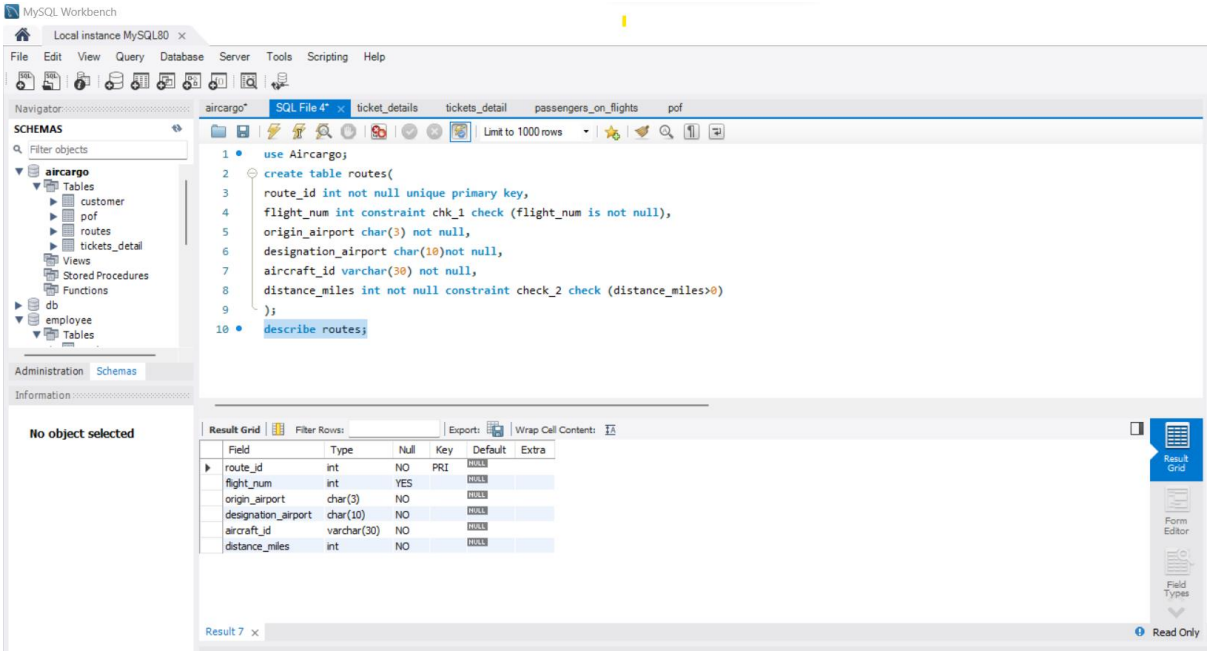



```
use aircargo;
load data infile'C:\\Program Files\\MySQL\\MySQL Server 8.0\\Uploads\\ticket_details.csv'into table tickets_detail
fields terminated by ',' enclosed by '"'
lines terminated by '\n'  ignore 1 rows
```

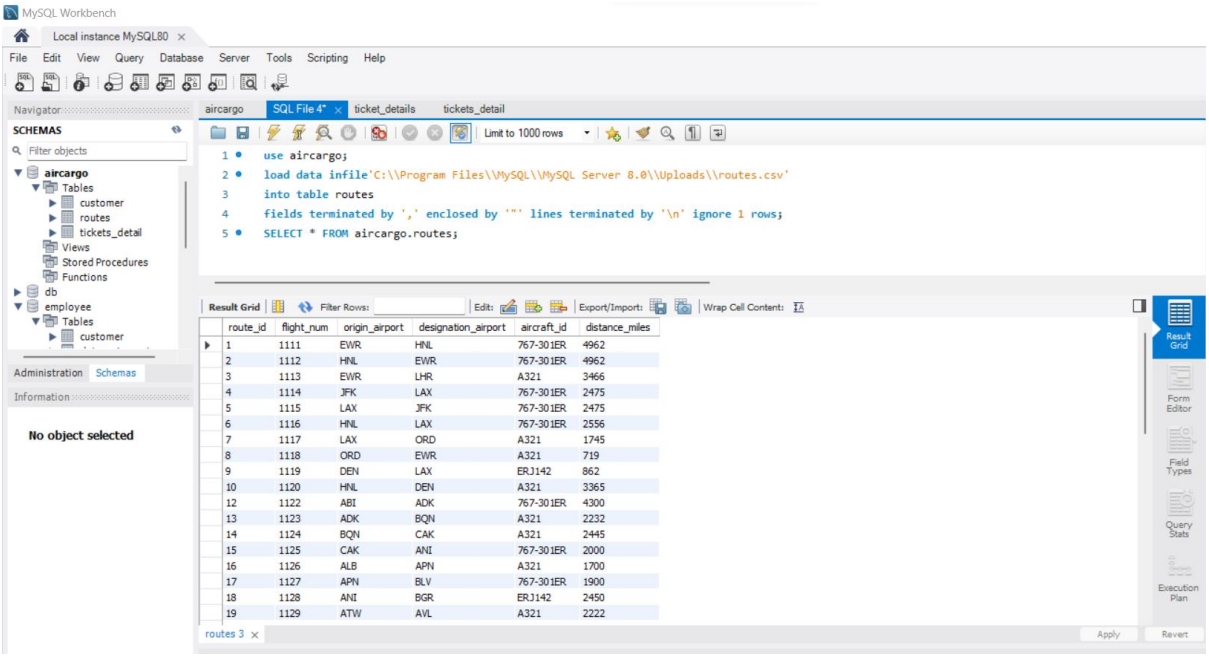


Routes

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;

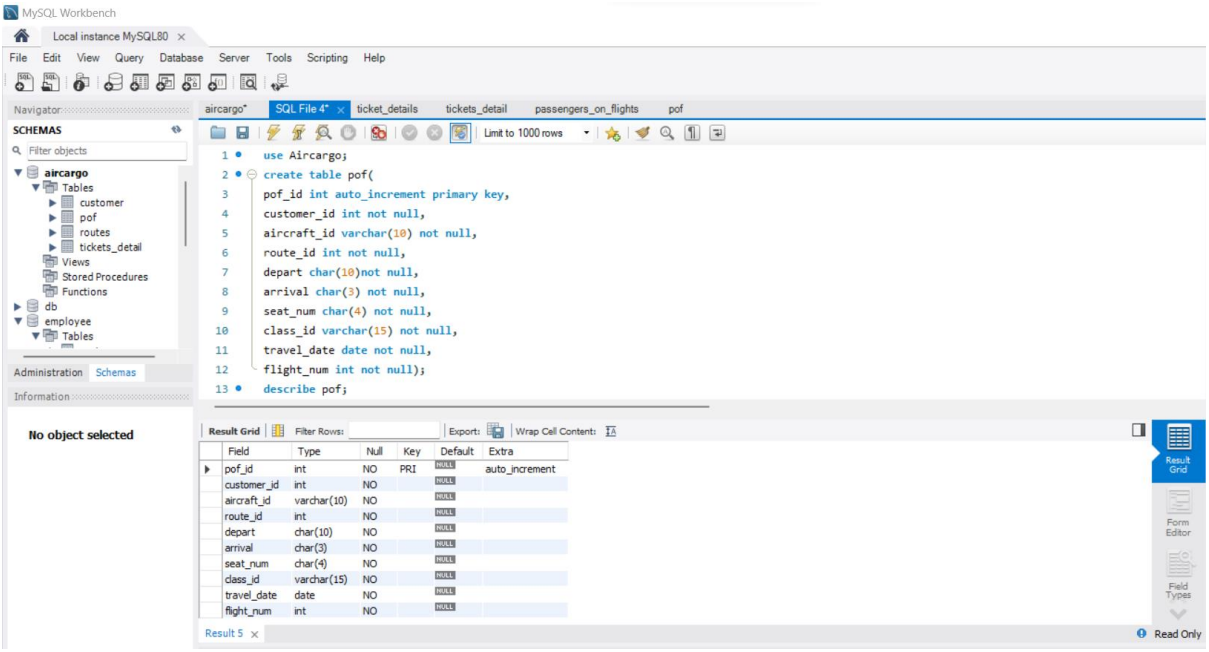


use aircargo;
loaddata infile'C:\\Program Files\\MySQL\\MySQL Server 8.0\\Uploads\\routes.csv'
into table routes
fields terminated by ',' enclosed by '"' lines terminated by '\n' ignore 1 rows;
SELECT * FROM aircargo.routes;



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;

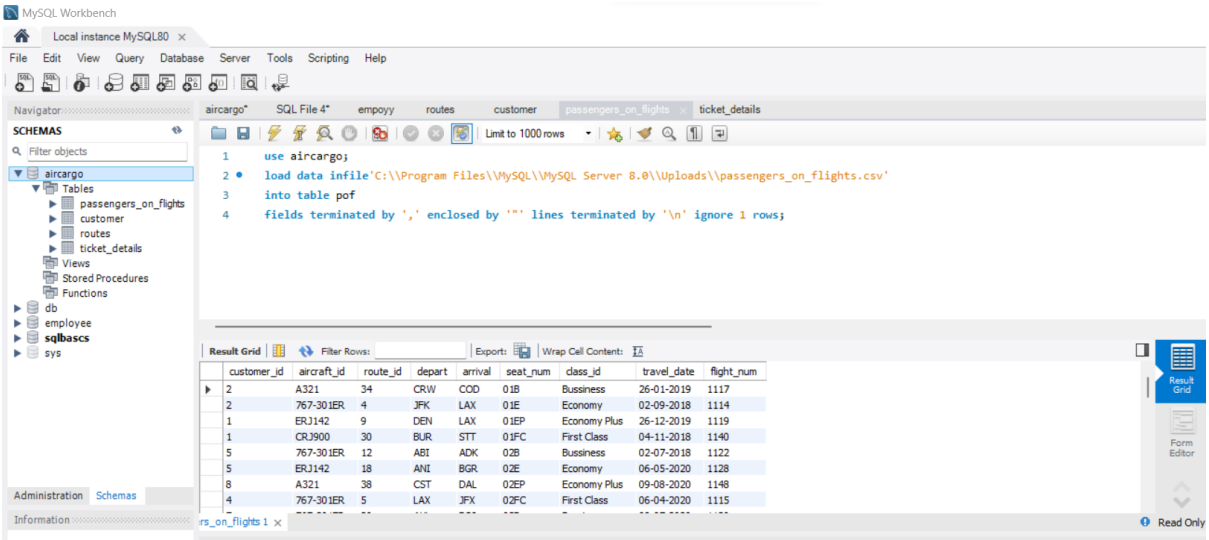


use aircargo;

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

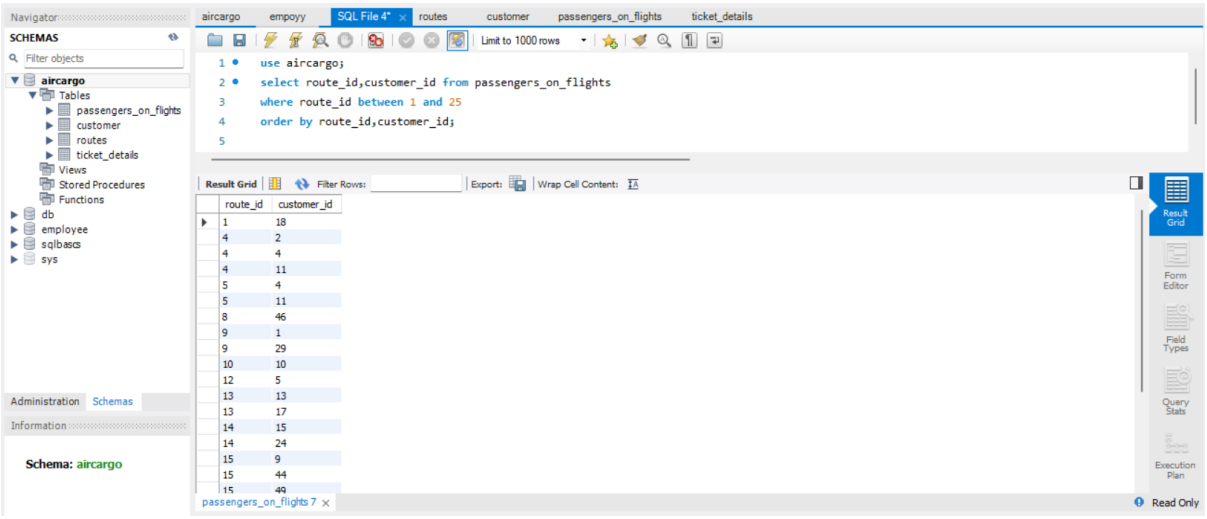
into table pof

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



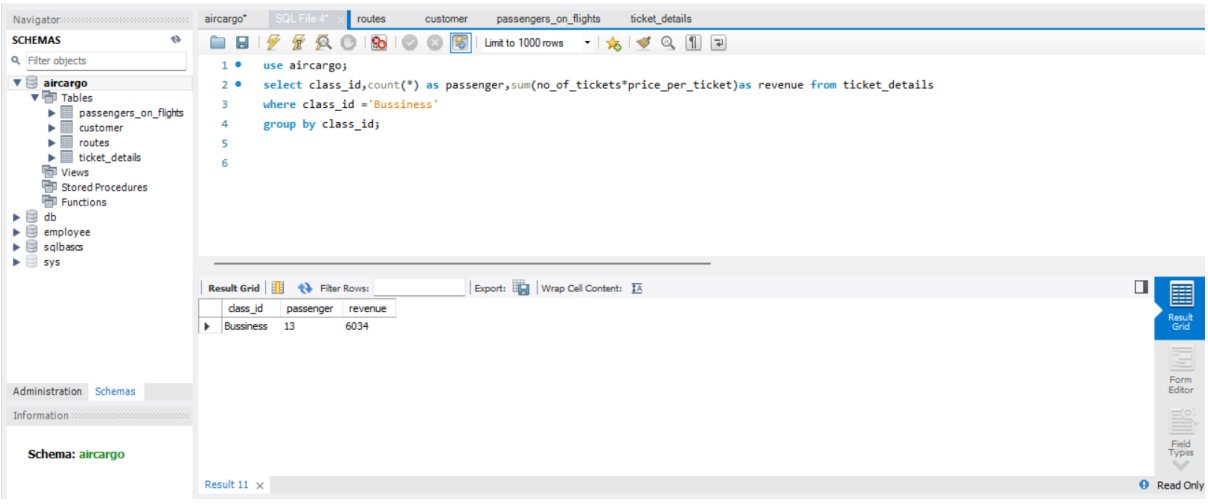
3. Write a query to display all the passengers (customers) who have travelled in routes 01 to 25. Take data from the passengers_on_flights table.

```
use aircargo;
select route_id,customer_id from passengers_on_flights
where route_id between 1 and 25
order by route_id,customer_id;
```



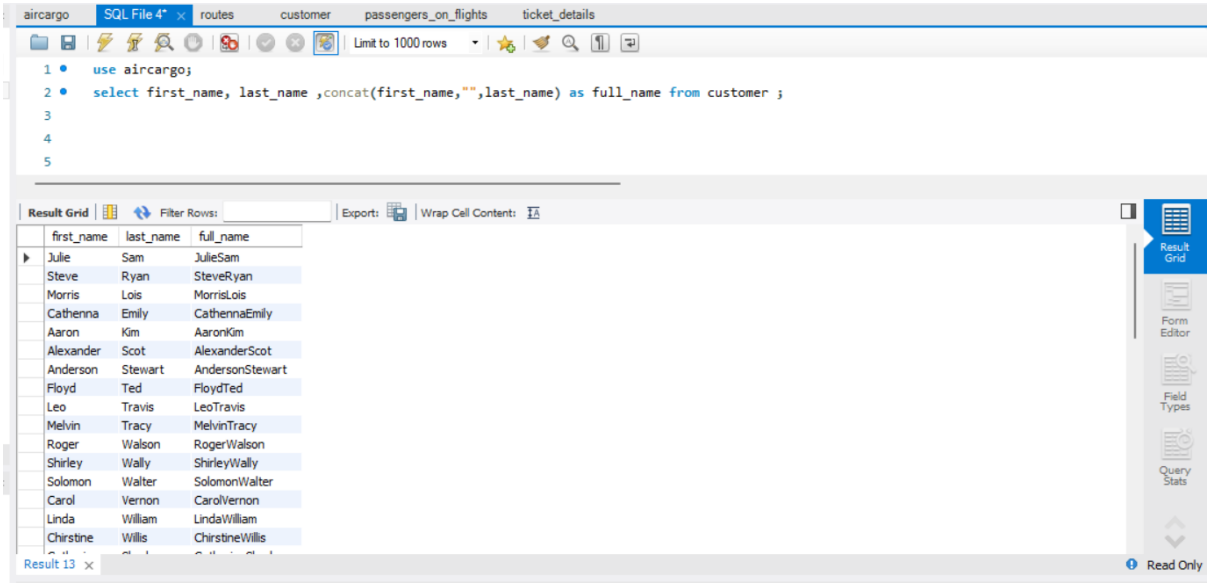
4. Write a query to identify the number of passengers and total revenue in business class from the ticket_details table.

```
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;
```



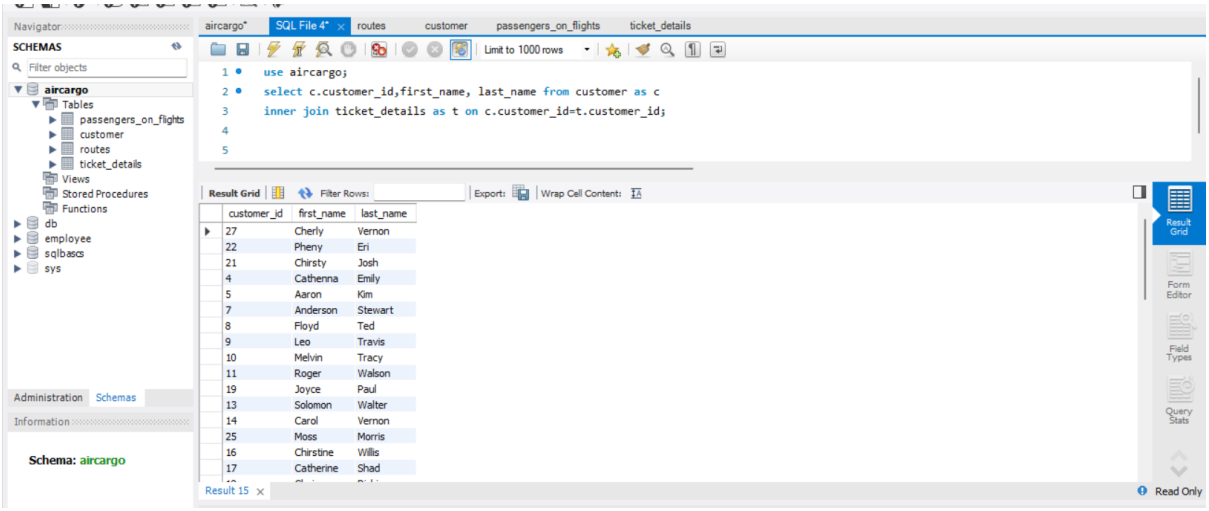
5. Write a query to display the full name of the customer by extracting the first name and last name from the customer table.

```
select first_name, last_name ,concat(first_name,"",last_name) as full_name
from customer ;
```



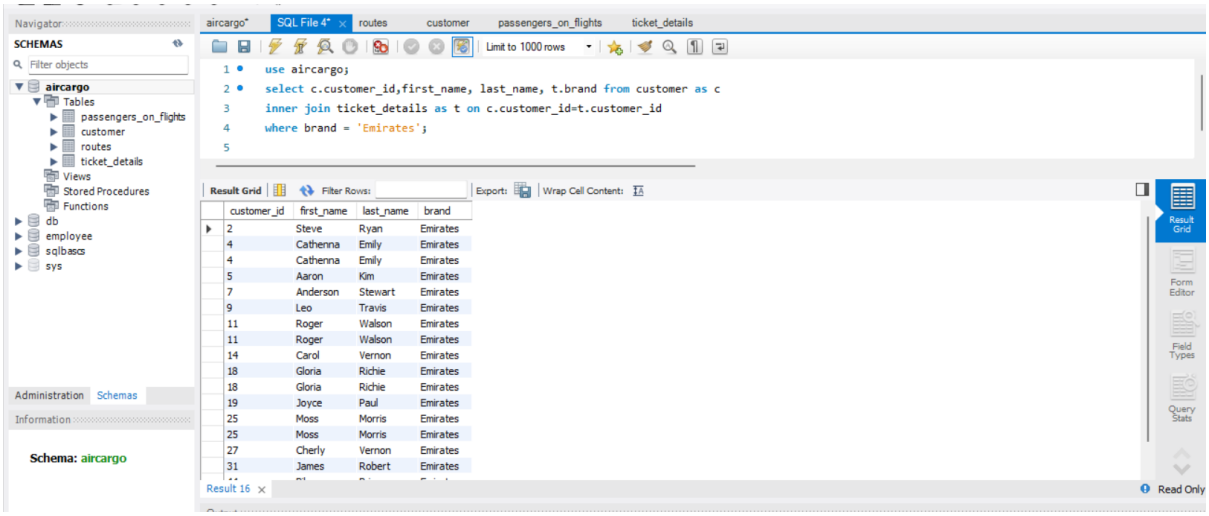
6. Write a query to extract the customers who have registered and booked a ticket. Use data from the customer and ticket_details tables.

```
select c.customer_id,first_name, last_name from customer as c
inner join ticket_details as t on c.customer_id=t.customer_id;
```



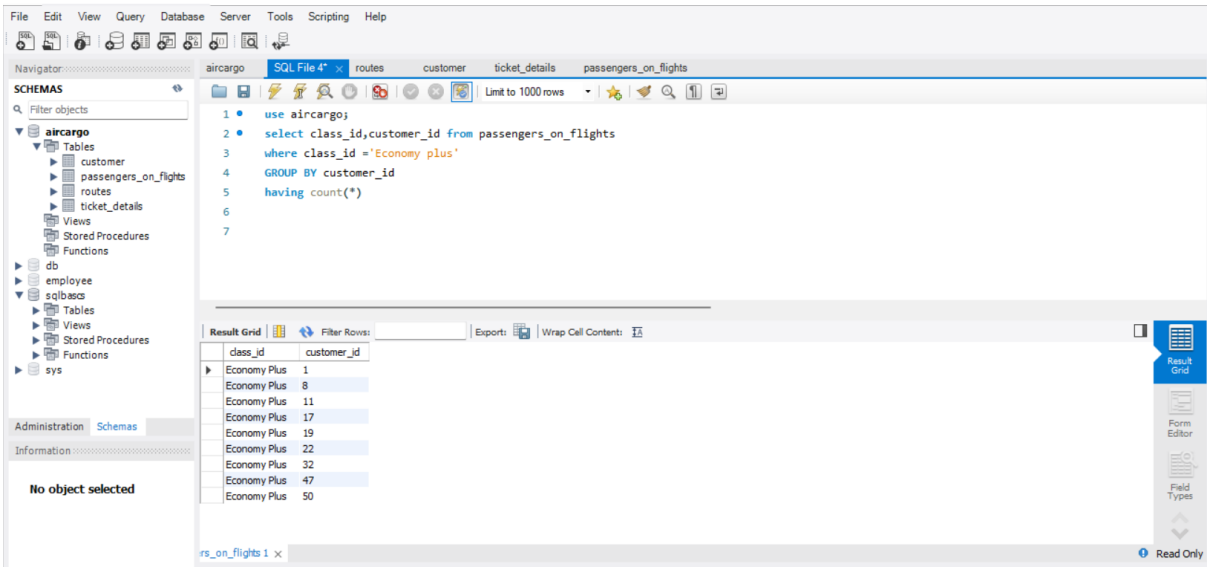
7. Write a query to identify the customer's first name and last name based on their customer ID and brand (Emirates) from the ticket_details table.

```
select c.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';
```



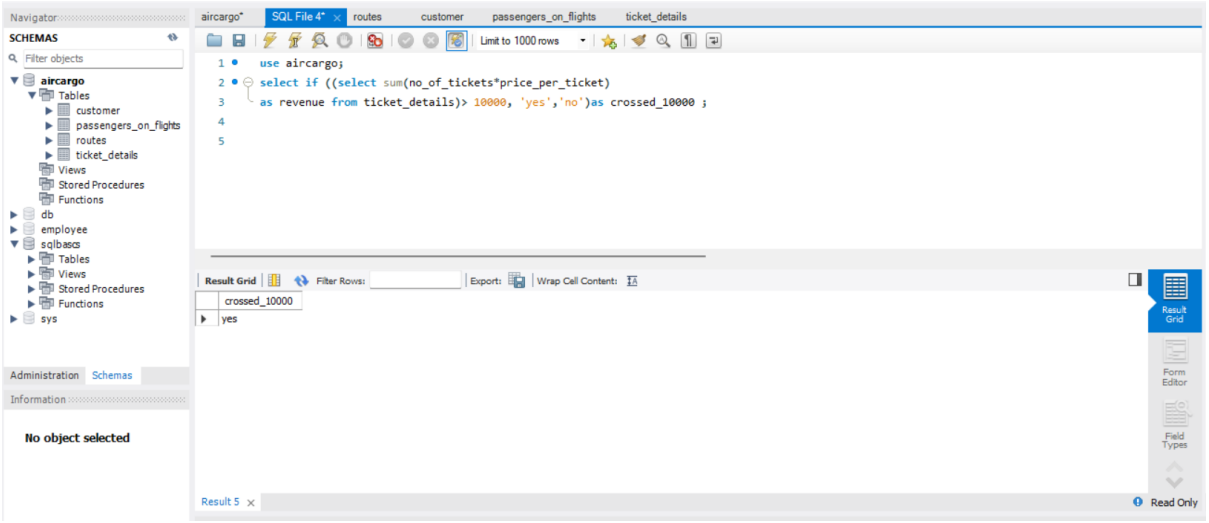
8. Write a query to identify the customers who have travelled by *Economy Plus* class using Group By and Having clause on the passengers_on_flights table.

```
use aircargo;
select class_id,customer_id from passengers_on_flights
where class_id ='Economy plus'
GROUP BY customer_id having count(*)
```



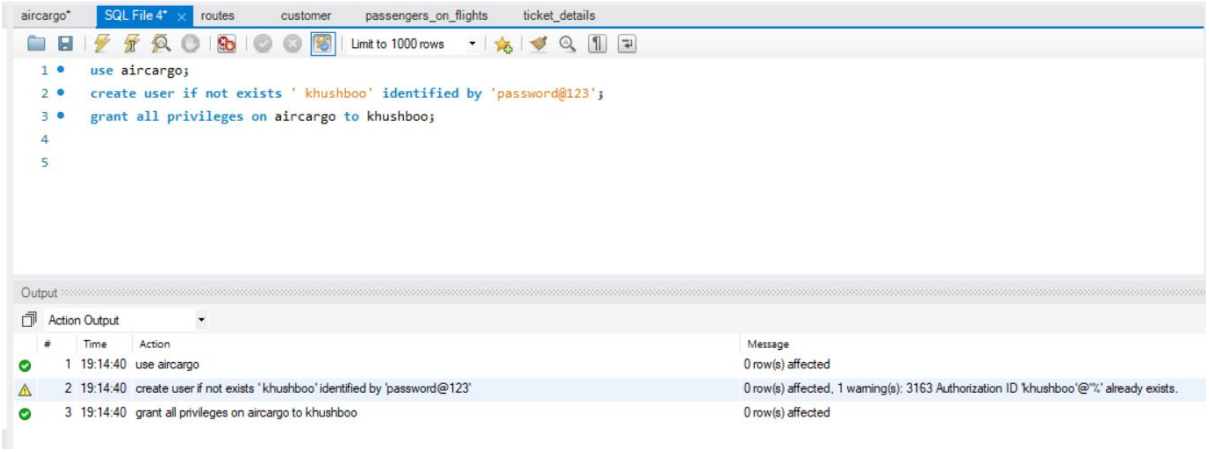
9. Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket_details table.

   select if ((select sum(no_of_tickets*price_per_ticket)
   as revenue from ticket_details)> 10000, 'yes','no')as crossed_10000 ;



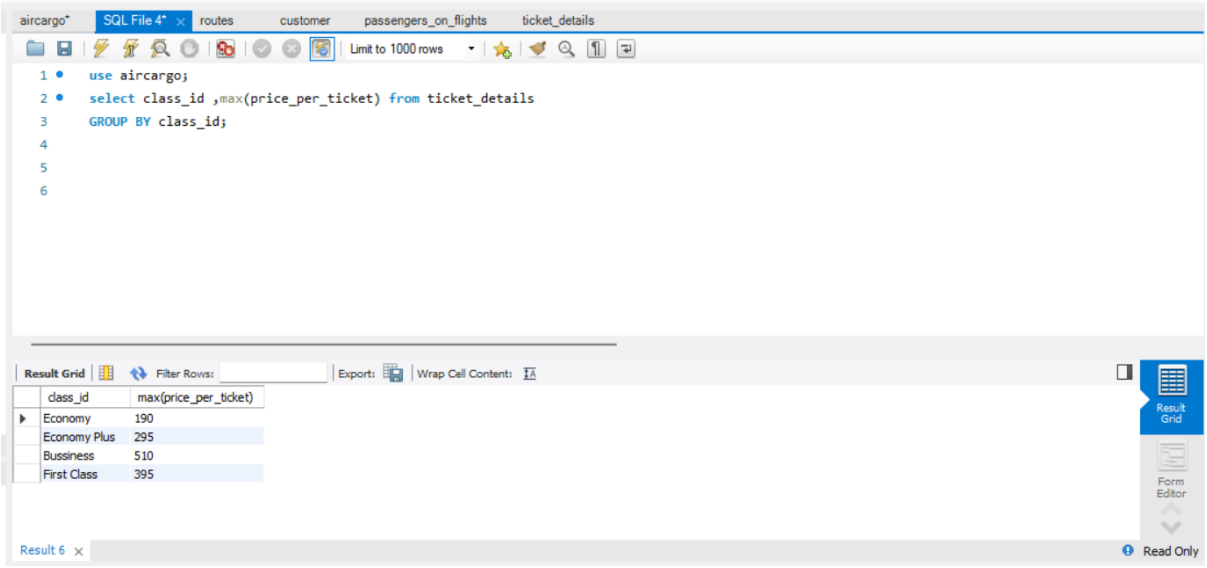
10. Write a query to create and grant access to a new user to perform operations on a database.

use aircargo;
create user if not exists ' khushboo' identified by 'password@123';
grant all privileges on aircargo to khushboo;



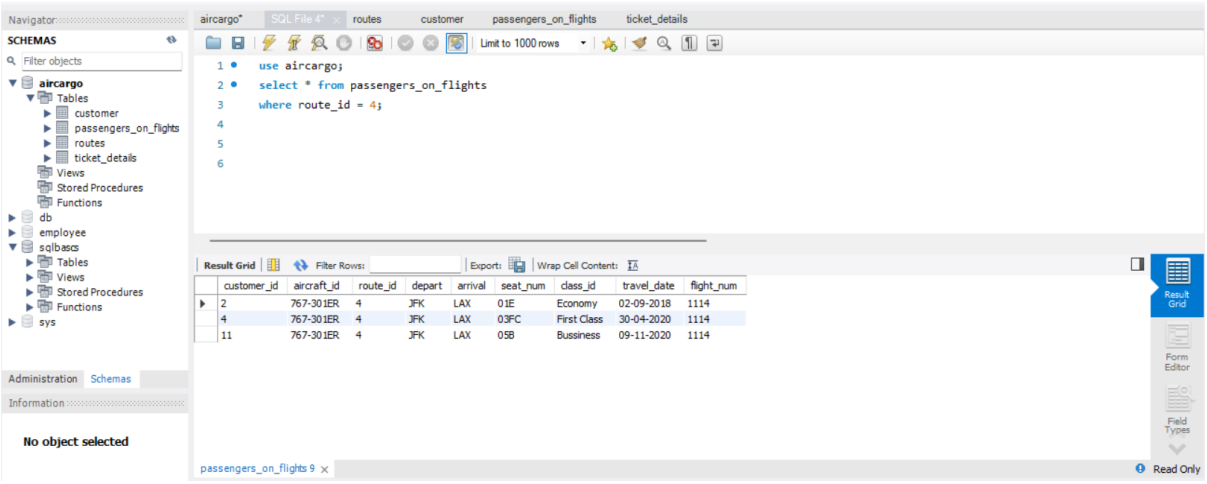
11. Write a query to find the maximum ticket price for each class using window functions on the ticket_details table.

```
use aircargo;
select class_id ,max(price_per_ticket) from ticket_details
GROUP BY class_id;
```



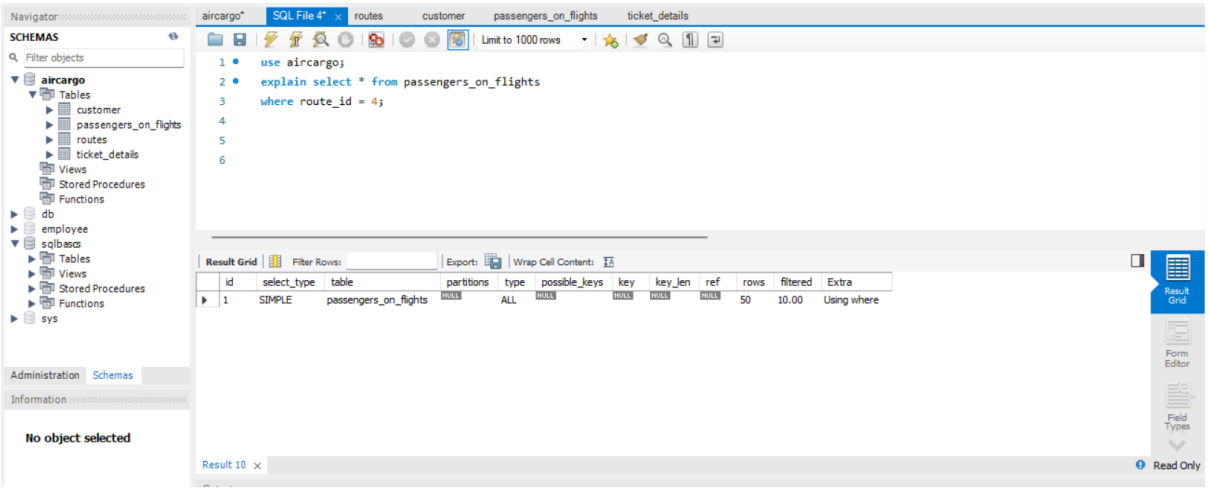
12. Write a query to extract the passengers whose route ID is 4 by improving the speed and performance of the passengers_on_flights table.

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



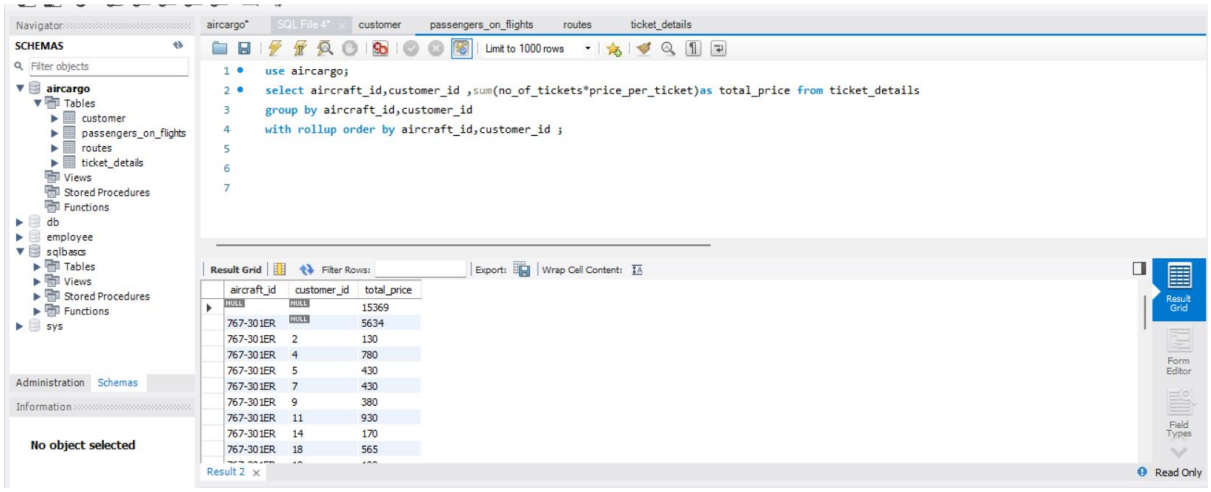
13. For the route ID 4, write a query to view the execution plan of the passengers_on_flights table.

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



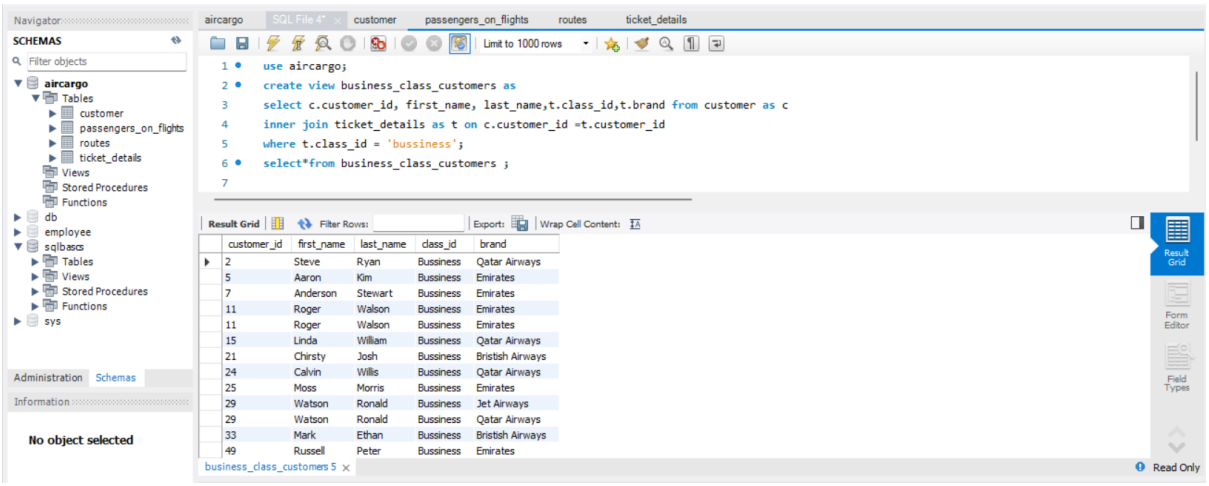
14. Write a query to calculate the total price of all tickets booked by a customer across different aircraft IDs using rollup function.

```
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 ;
```



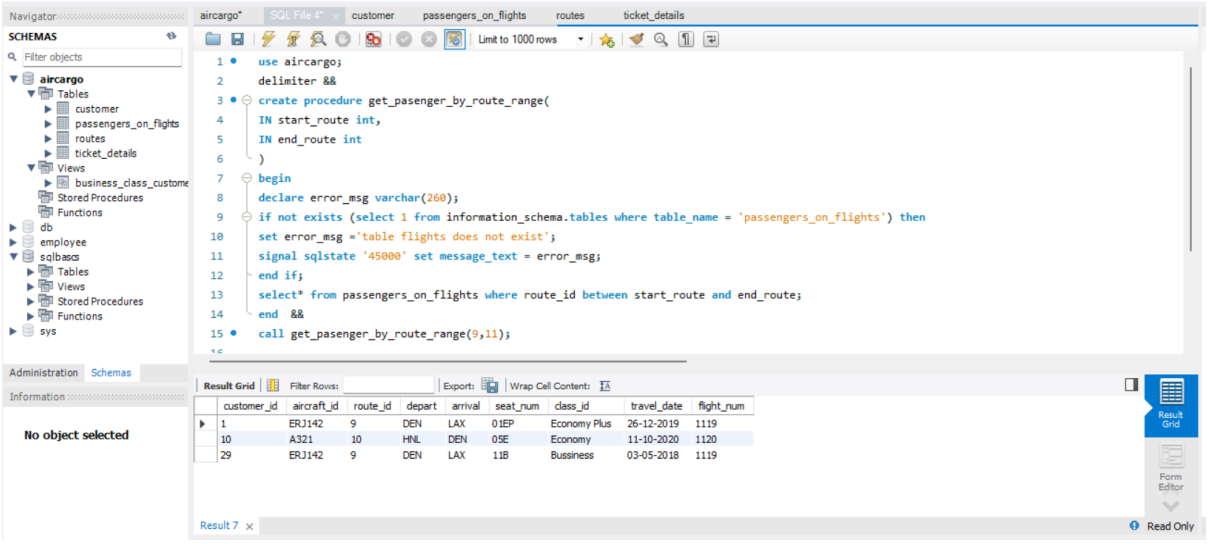
15. Write a query to create a view with only business class customers along with the brand of airlines.

```
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 ;
```



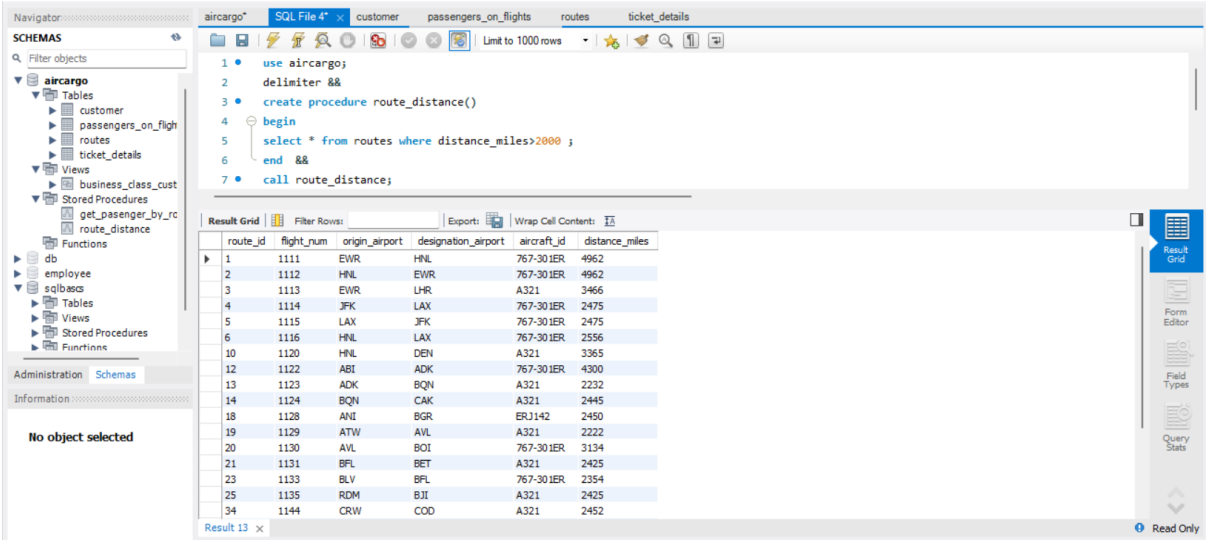
16. Write a query to create a stored procedure to get the details of all passengers flying between a range of routes defined in run time. Also, return an error message if the table doesn't exist.

```
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);
```



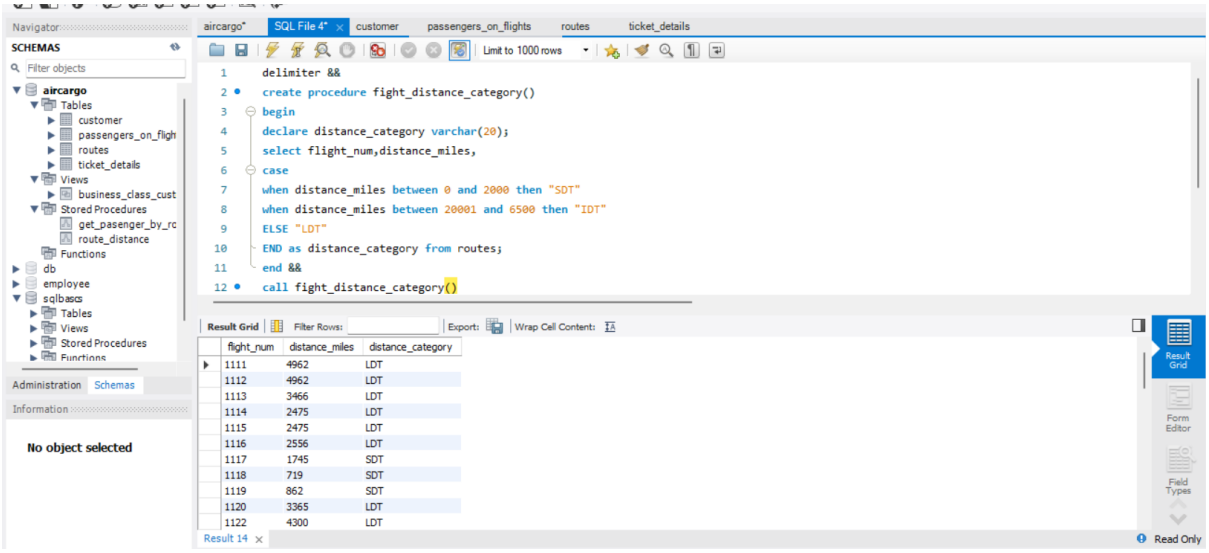
17. Write a query to create a stored procedure that extracts all the details from the routes table where the travelled distance is more than 2000 miles.

```
use aircargo;
delimiter &&
create procedure route_distance()
begin select * from routes where distance_miles>2000 ;
end  &&
call route_distance;
```



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 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()
```

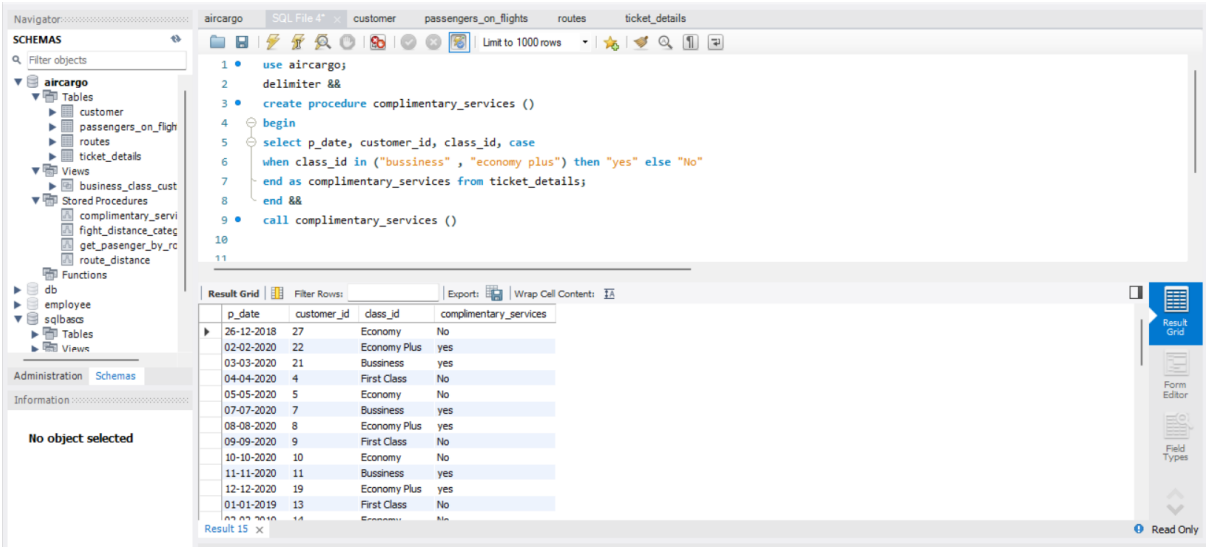


19. Write a query to extract ticket purchase date, customer ID, class ID and specify if the complimentary services are provided for the specific class using a stored function in stored procedure on the ticket_details table.

    Condition:

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

```
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 ()
```



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.