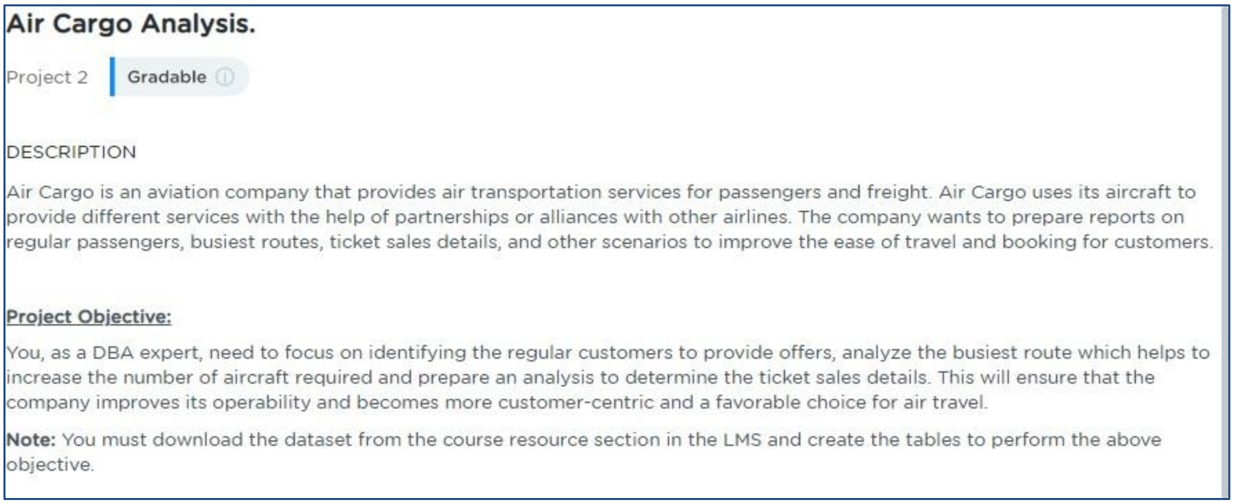
# Air Cargo Analysis



## Air Cargo Analysis.

Project 2 | Gradable ⓘ

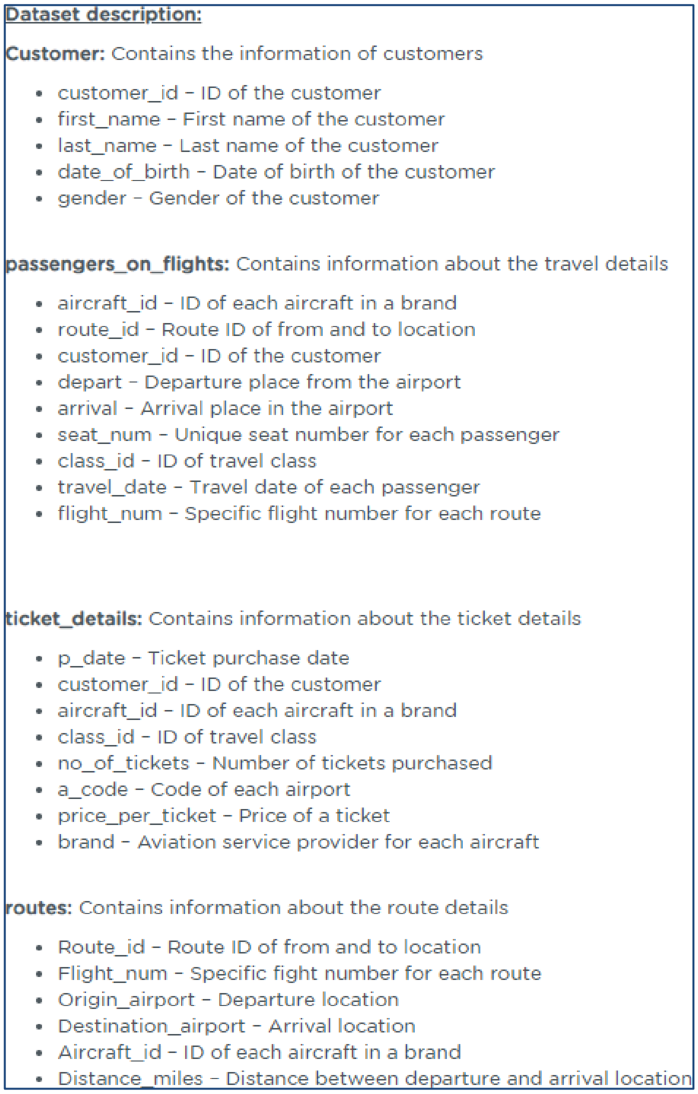
DESCRIPTION

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

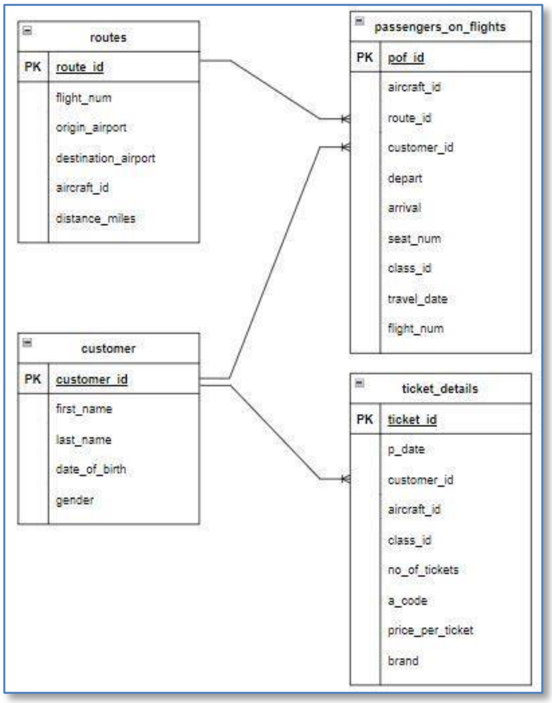
**routes:** Contains information about the route details

- 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

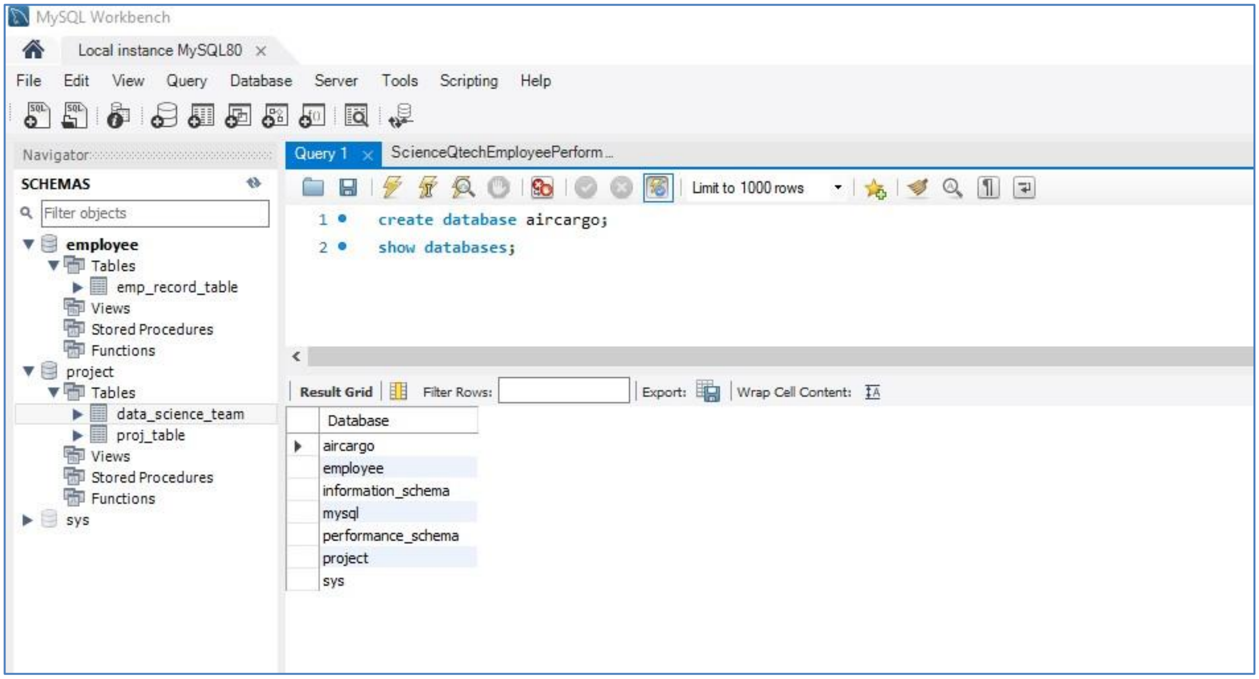
# Air Cargo Analysis

## Following operations should be performed: SQL Code and Output Screenshots

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



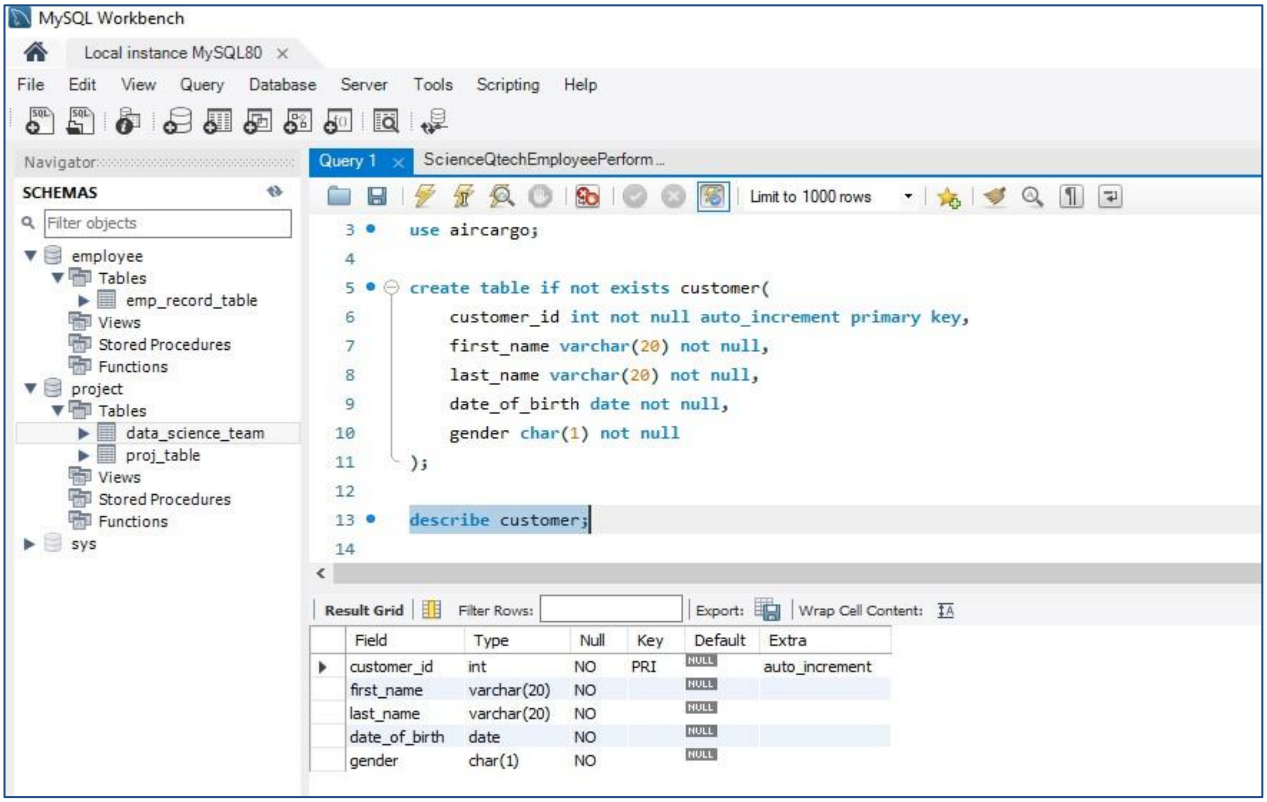
2. Write a query to create route_details table using suitable data types for the fields, such as route_id, flight_num, origin_airport, destination_airport, aircraft_id, and distance_miles. Implement the check constraint for the flight number and unique constraint for the route_id fields. Also, make surethat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



```
create database aircargo;
show databases;
```

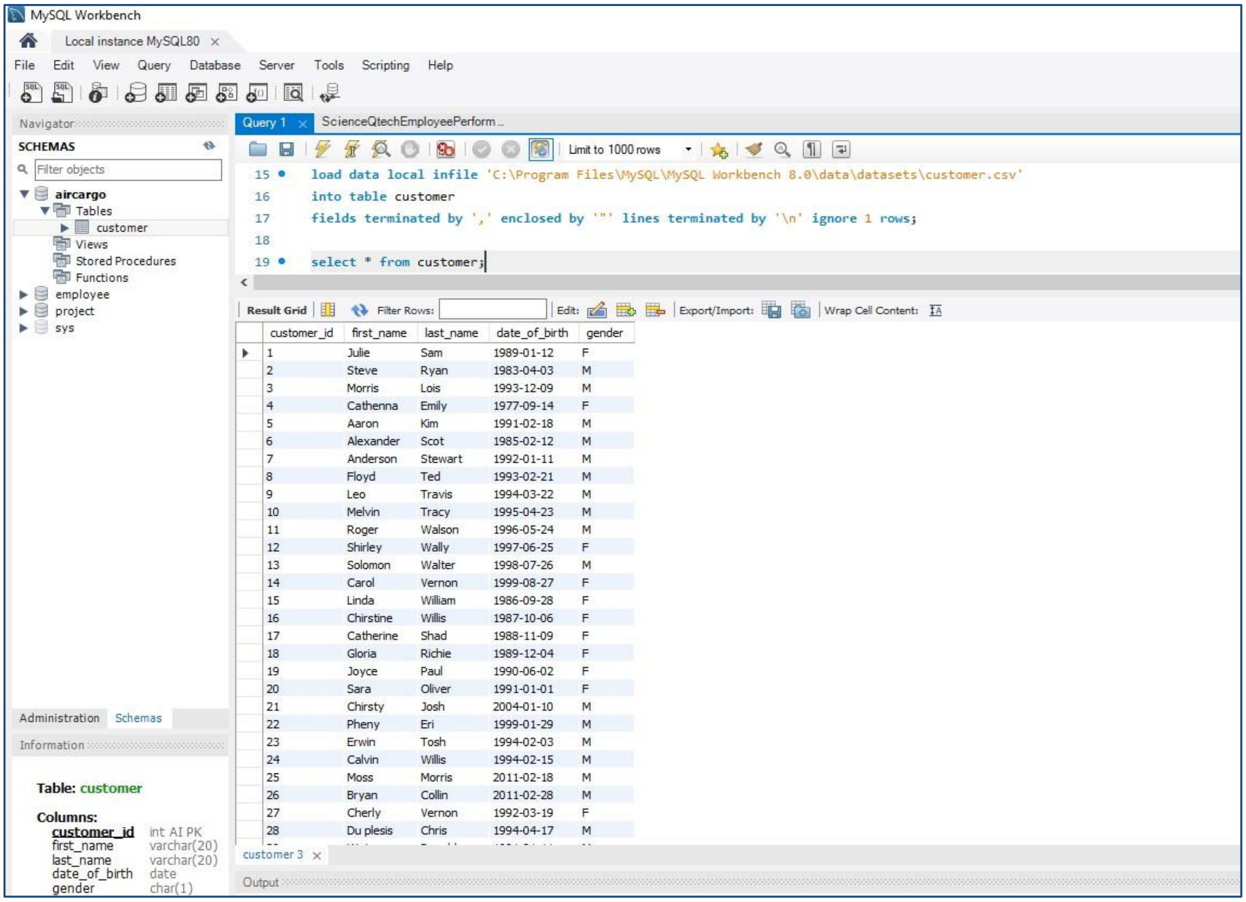
| Database |
|---|
| aircargo |
| employee |
| information_schema |
| mysql |
| performance_schema |
| project |
| sys |

# Air Cargo Analysis



```
use aircargo;

create table if not exists customer(
    customer_id int not null auto_increment primary key,
    first_name varchar(20) not null,
    last_name varchar(20) not null,
    date_of_birth date not null,
    gender char(1) not null
);

describe customer;
```

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| customer_id | int | NO | PRI | NULL | auto_increment |
| first_name | varchar(20) | NO | | NULL | |
| last_name | varchar(20) | NO | | NULL | |
| date_of_birth | date | NO | | NULL | |
| gender | char(1) | NO | | NULL | |



```
load data local infile 'C:\Program Files\MySQL\MySQL Workbench 8.0\data\datasets\customer.csv'
into table customer
fields terminated by ',' enclosed by '"' lines terminated by '\n' ignore 1 rows;

select * from customer;
```

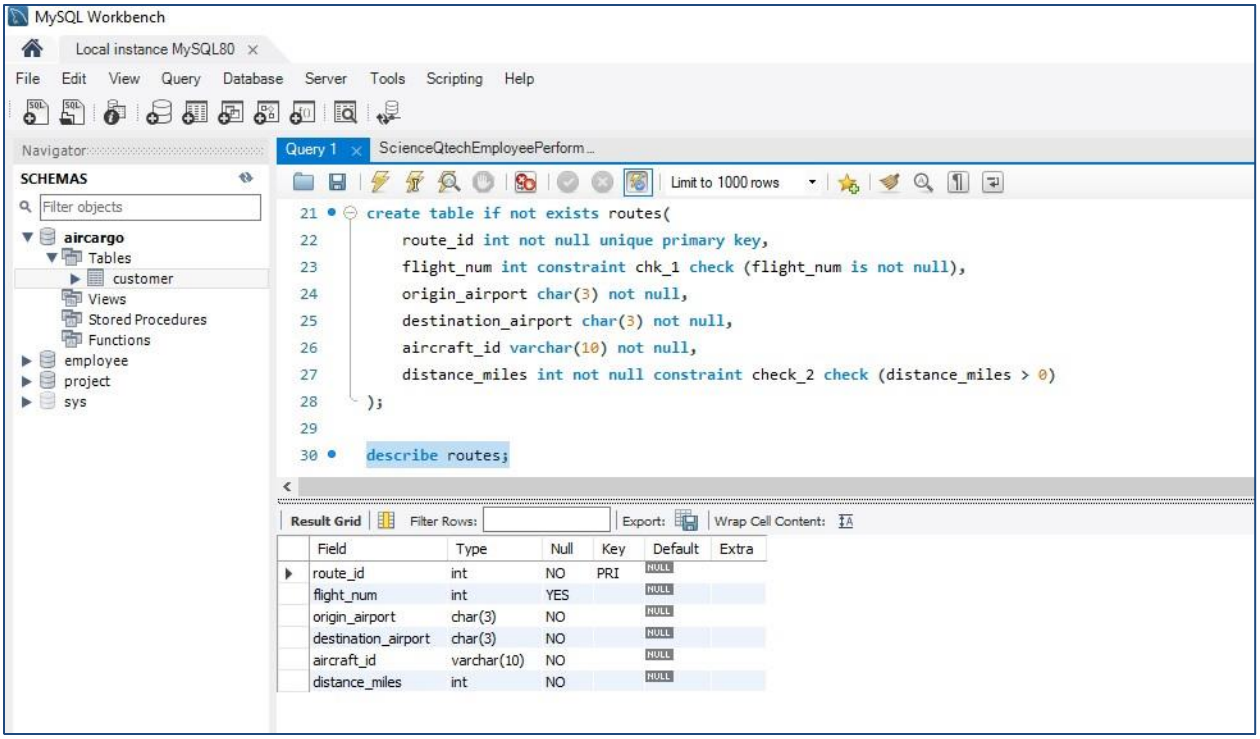
| customer_id | first_name | last_name | date_of_birth | gender |
|---|---|---|---|---|
| 1 | Julie | Sam | 1989-01-12 | F |
| 2 | Steve | Ryan | 1983-04-03 | M |
| 3 | Morris | Lois | 1993-12-09 | M |
| 4 | Cathenna | Emily | 1977-09-14 | F |
| 5 | Aaron | Kim | 1991-02-18 | M |
| 6 | Alexander | Scot | 1985-02-12 | M |
| 7 | Anderson | Stewart | 1992-01-11 | M |
| 8 | Floyd | Ted | 1993-02-21 | M |
| 9 | Leo | Travis | 1994-03-22 | M |
| 10 | Melvin | Tracy | 1995-04-23 | M |
| 11 | Roger | Walson | 1996-05-24 | M |
| 12 | Shirley | Wally | 1997-06-25 | F |
| 13 | Solomon | Walter | 1998-07-26 | M |
| 14 | Carol | Vernon | 1999-08-27 | F |
| 15 | Linda | William | 1986-09-28 | F |
| 16 | Chirstine | Willis | 1987-10-06 | F |
| 17 | Catherine | Shad | 1988-11-09 | F |
| 18 | Gloria | Richie | 1989-12-04 | F |
| 19 | Joyce | Paul | 1990-06-02 | F |
| 20 | Sara | Oliver | 1991-01-01 | F |
| 21 | Chirsty | Josh | 2004-01-10 | M |
| 22 | Pheny | Eri | 1999-01-29 | M |
| 23 | Erwin | Tosh | 1994-02-03 | M |
| 24 | Calvin | Willis | 1994-02-15 | M |
| 25 | Moss | Morris | 2011-02-18 | M |
| 26 | Bryan | Collin | 2011-02-28 | M |
| 27 | Cherly | Vernon | 1992-03-19 | F |
| 28 | Du plesis | Chris | 1994-04-17 | M |

Table: customer

Columns:

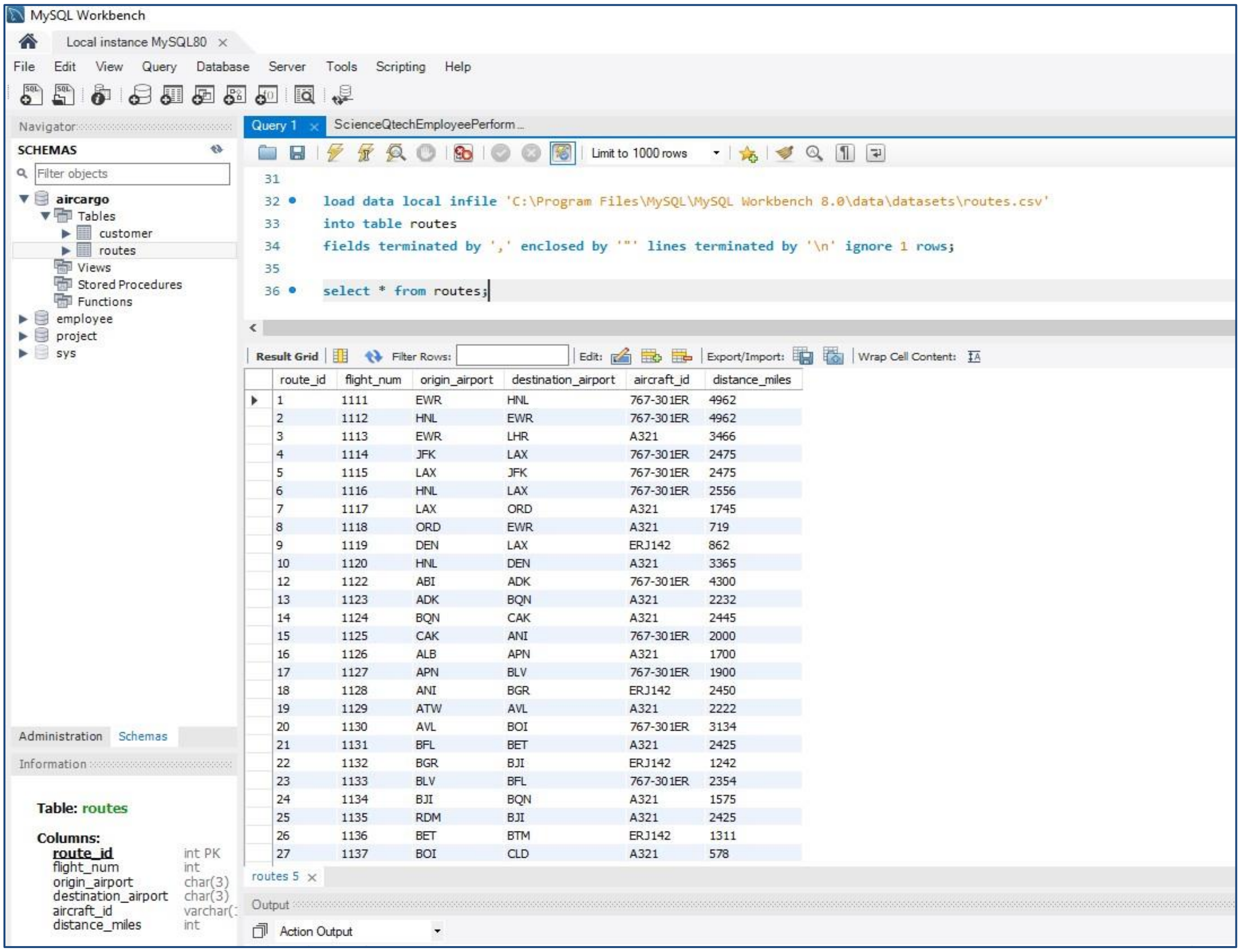
| | |
|---|---|
| customer_id | int AI PK |
| first_name | varchar(20) |
| last_name | varchar(20) |
| date_of_birth | date |
| gender | char(1) |

# Air Cargo Analysis



```
create table if not exists 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,
    destination_airport char(3) not null,
    aircraft_id varchar(10) not null,
    distance_miles int not null constraint check_2 check (distance_miles > 0)
);

describe routes;
```

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| route_id | int | NO | PRI | NULL | |
| flight_num | int | YES | | NULL | |
| origin_airport | char(3) | NO | | NULL | |
| destination_airport | char(3) | NO | | NULL | |
| aircraft_id | varchar(10) | NO | | NULL | |
| distance_miles | int | NO | | NULL | |



```

load data local infile 'C:\Program Files\MySQL\MySQL Workbench 8.0\data\datasets\routes.csv'
into table routes
fields terminated by ',' enclosed by '"' lines terminated by '\n' ignore 1 rows;

select * from routes;
```

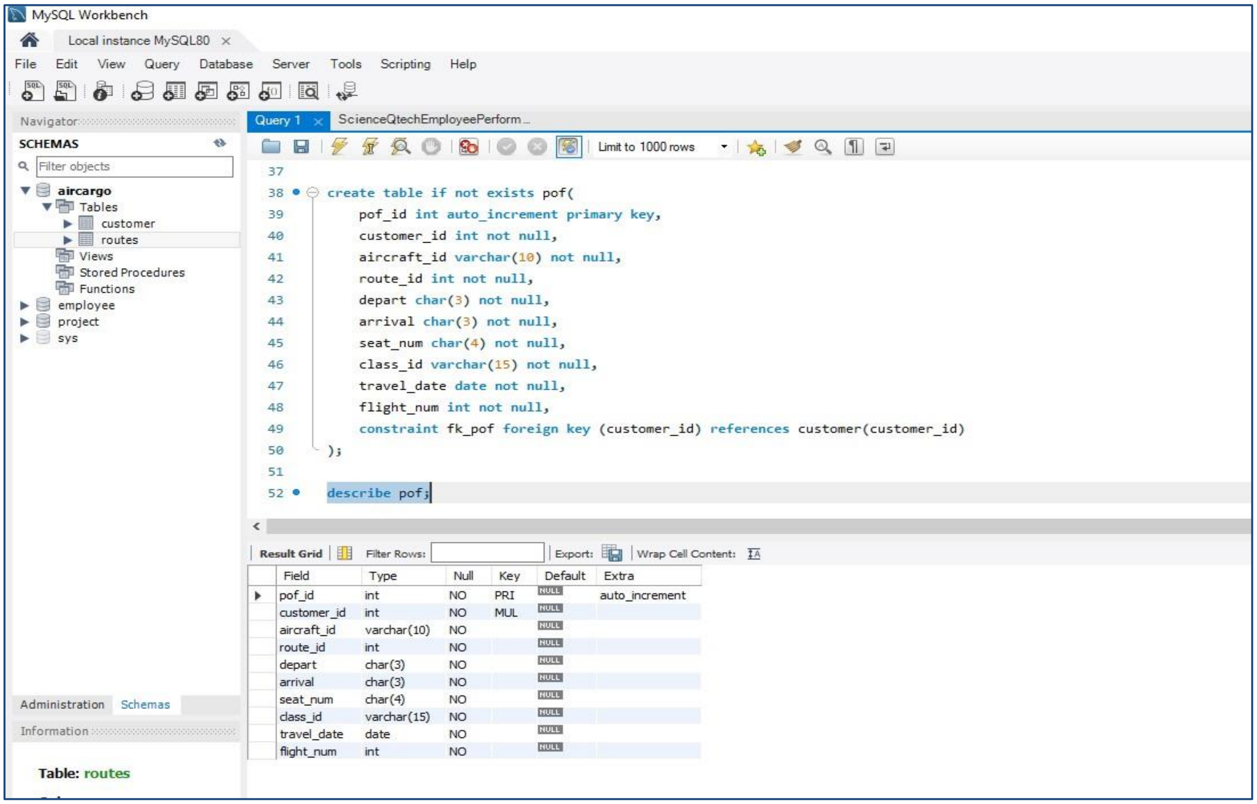
| route_id | flight_num | origin_airport | destination_airport | aircraft_id | distance_miles |
|---|---|---|---|---|---|
| 1 | 1111 | EWR | HNL | 767-301ER | 4962 |
| 2 | 1112 | HNL | EWR | 767-301ER | 4962 |
| 3 | 1113 | EWR | LHR | A321 | 3466 |
| 4 | 1114 | JFK | LAX | 767-301ER | 2475 |
| 5 | 1115 | LAX | JFK | 767-301ER | 2475 |
| 6 | 1116 | HNL | LAX | 767-301ER | 2556 |
| 7 | 1117 | LAX | ORD | A321 | 1745 |
| 8 | 1118 | ORD | EWR | A321 | 719 |
| 9 | 1119 | DEN | LAX | ERJ142 | 862 |
| 10 | 1120 | HNL | DEN | A321 | 3365 |
| 12 | 1122 | ABI | ADK | 767-301ER | 4300 |
| 13 | 1123 | ADK | BQN | A321 | 2232 |
| 14 | 1124 | BQN | CAK | A321 | 2445 |
| 15 | 1125 | CAK | ANI | 767-301ER | 2000 |
| 16 | 1126 | ALB | APN | A321 | 1700 |
| 17 | 1127 | APN | BLV | 767-301ER | 1900 |
| 18 | 1128 | ANI | BGR | ERJ142 | 2450 |
| 19 | 1129 | ATW | AVL | A321 | 2222 |
| 20 | 1130 | AVL | BOI | 767-301ER | 3134 |
| 21 | 1131 | BFL | BET | A321 | 2425 |
| 22 | 1132 | BGR | BJI | ERJ142 | 1242 |
| 23 | 1133 | BLV | BFL | 767-301ER | 2354 |
| 24 | 1134 | BJI | BQN | A321 | 1575 |
| 25 | 1135 | RDM | BJI | A321 | 2425 |
| 26 | 1136 | BET | BTM | ERJ142 | 1311 |
| 27 | 1137 | BOI | CLD | A321 | 578 |

Table: routes

Columns:

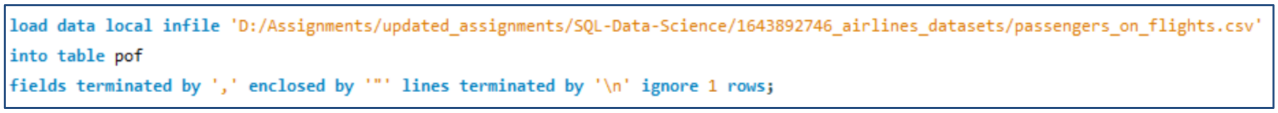
| | |
|---|---|
| route_id | int PK |
| flight_num | int |
| origin_airport | char(3) |
| destination_airport | char(3) |
| aircraft_id | varchar(: |
| distance_miles | int |

# Air Cargo Analysis

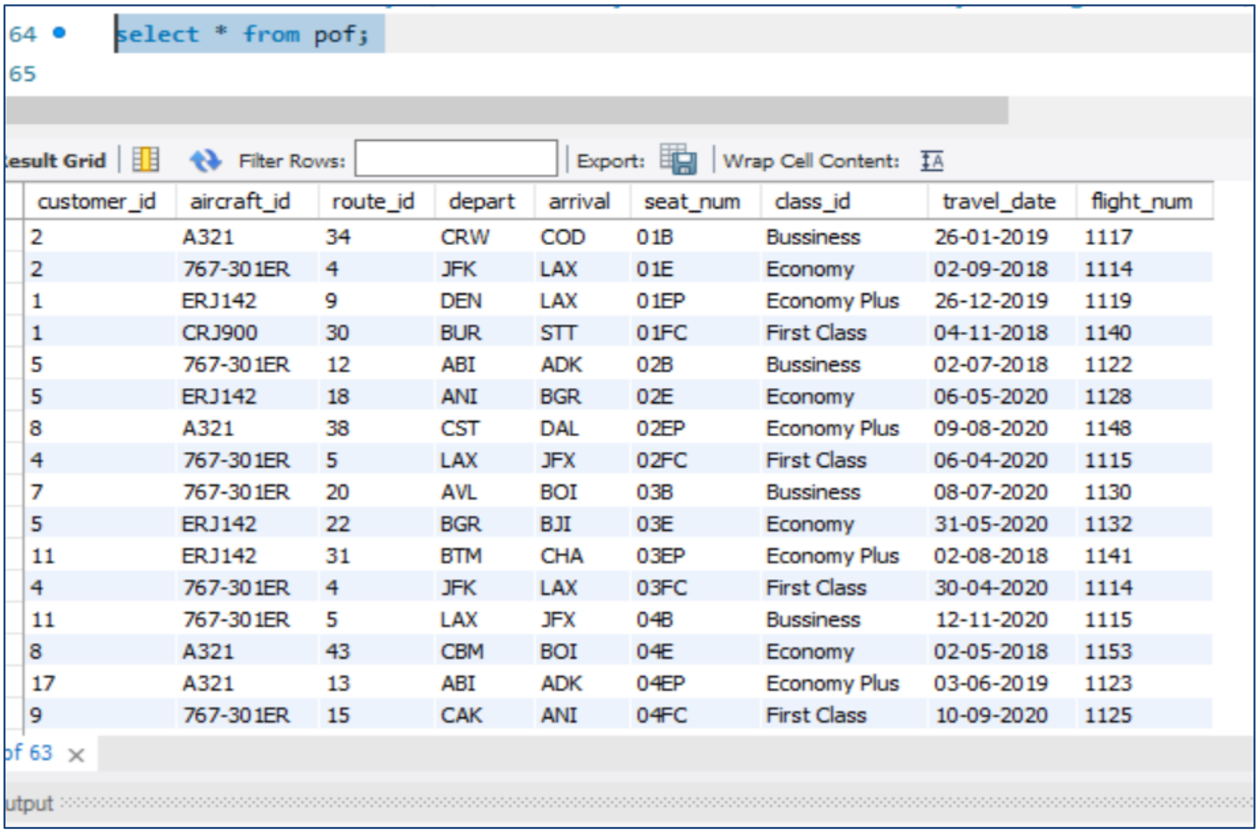


```sql

create table if not exists 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(3) 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,
    constraint fk_pof foreign key (customer_id) references customer(customer_id)
);

describe pof;
```

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| pof_id | int | NO | PRI | NULL | auto_increment |
| customer_id | int | NO | MUL | NULL | |
| aircraft_id | varchar(10) | NO | | NULL | |
| route_id | int | NO | | NULL | |
| depart | char(3) | NO | | NULL | |
| arrival | char(3) | NO | | NULL | |
| seat_num | char(4) | NO | | NULL | |
| class_id | varchar(15) | NO | | NULL | |
| travel_date | date | NO | | NULL | |
| flight_num | int | NO | | NULL | |



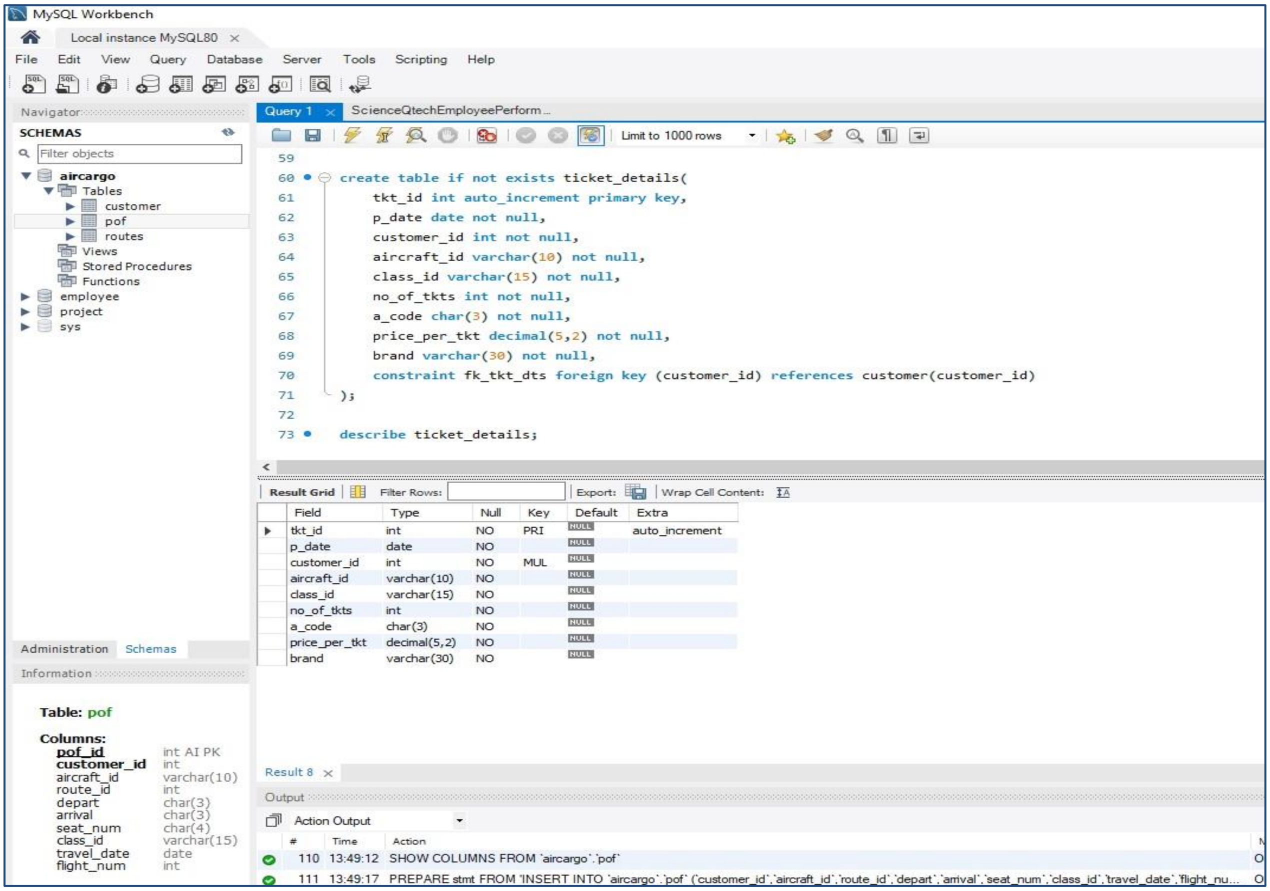
```sql
load data local infile 'D:/Assignments/updated_assignments/SQL-Data-Science/1643892746_airlines_datasets/passengers_on_flights.csv'
into table pof
fields terminated by ',' enclosed by '"' lines terminated by '\n' ignore 1 rows;
```



```sql
select * from pof;
```

| customer_id | aircraft_id | route_id | depart | arrival | seat_num | class_id | travel_date | flight_num |
|---|---|---|---|---|---|---|---|---|
| 2 | A321 | 34 | CRW | COD | 01B | Bussiness | 26-01-2019 | 1117 |
| 2 | 767-301ER | 4 | JFK | LAX | 01E | Economy | 02-09-2018 | 1114 |
| 1 | ERJ142 | 9 | DEN | LAX | 01EP | Economy Plus | 26-12-2019 | 1119 |
| 1 | CRJ900 | 30 | BUR | STT | 01FC | First Class | 04-11-2018 | 1140 |
| 5 | 767-301ER | 12 | ABI | ADK | 02B | Bussiness | 02-07-2018 | 1122 |
| 5 | ERJ142 | 18 | ANI | BGR | 02E | Economy | 06-05-2020 | 1128 |
| 8 | A321 | 38 | CST | DAL | 02EP | Economy Plus | 09-08-2020 | 1148 |
| 4 | 767-301ER | 5 | LAX | JFX | 02FC | First Class | 06-04-2020 | 1115 |
| 7 | 767-301ER | 20 | AVL | BOI | 03B | Bussiness | 08-07-2020 | 1130 |
| 5 | ERJ142 | 22 | BGR | BJI | 03E | Economy | 31-05-2020 | 1132 |
| 11 | ERJ142 | 31 | BTM | CHA | 03EP | Economy Plus | 02-08-2018 | 1141 |
| 4 | 767-301ER | 4 | JFK | LAX | 03FC | First Class | 30-04-2020 | 1114 |
| 11 | 767-301ER | 5 | LAX | JFX | 04B | Bussiness | 12-11-2020 | 1115 |
| 8 | A321 | 43 | CBM | BOI | 04E | Economy | 02-05-2018 | 1153 |
| 17 | A321 | 13 | ABI | ADK | 04EP | Economy Plus | 03-06-2019 | 1123 |
| 9 | 767-301ER | 15 | CAK | ANI | 04FC | First Class | 10-09-2020 | 1125 |

# Air Cargo Analysis



```sql
create table if not exists ticket_details(
    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(15) not null,
    no_of_tkts int not null,
    a_code char(3) 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)
);

describe ticket_details;
```

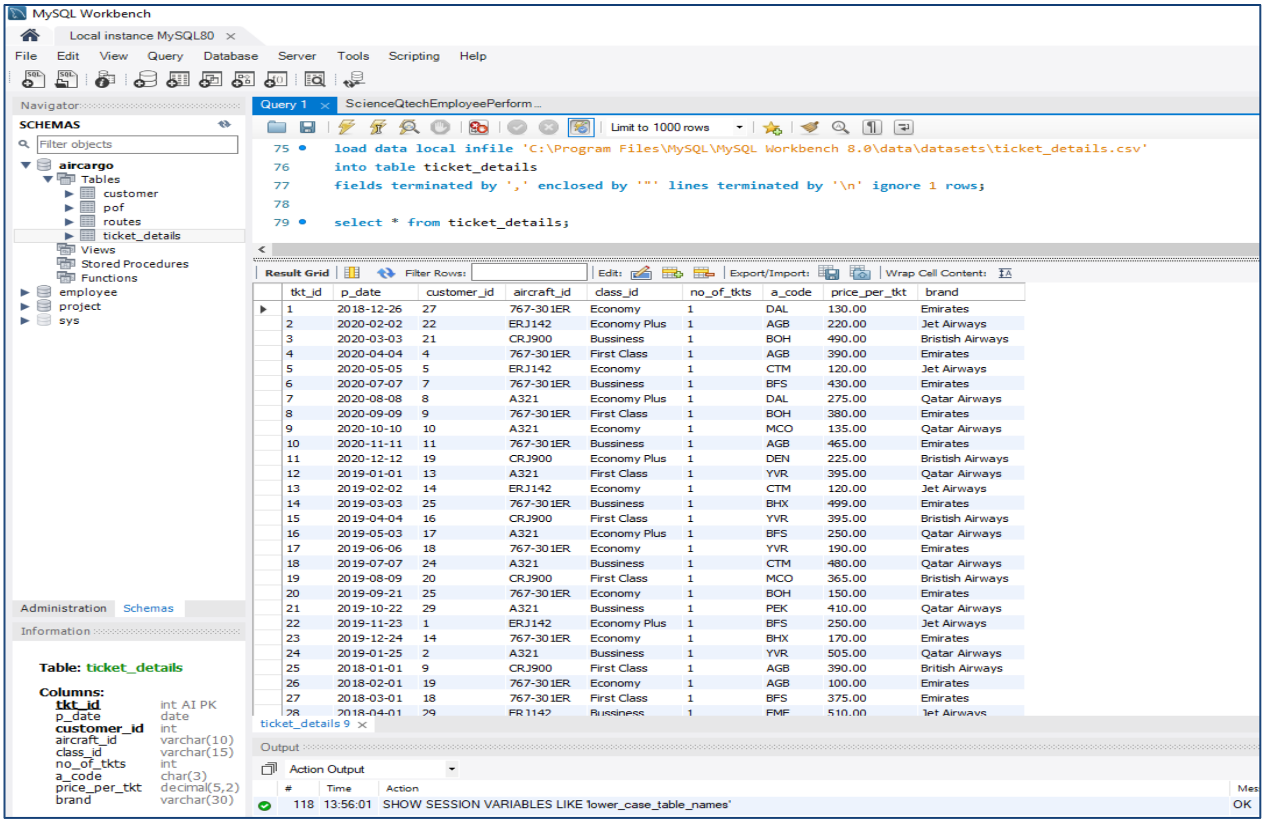
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| tkt_id | int | NO | PRI | NULL | auto_increment |
| p_date | date | NO | | NULL | |
| customer_id | int | NO | MUL | NULL | |
| aircraft_id | varchar(10) | NO | | NULL | |
| class_id | varchar(15) | NO | | NULL | |
| no_of_tkts | int | NO | | NULL | |
| a_code | char(3) | NO | | NULL | |
| price_per_tkt | decimal(5,2) | NO | | NULL | |
| brand | varchar(30) | NO | | NULL | |

Table: pof

Columns:

| | |
|---|---|
| pof_id | int AI PK |
| customer_id | int |
| aircraft_id | varchar(10) |
| route_id | int |
| depart | char(3) |
| arrival | char(3) |
| seat_num | char(4) |
| class_id | varchar(15) |
| travel_date | date |
| flight_num | int |

```
#    Time      Action
110  13:49:12  SHOW COLUMNS FROM `aircargo`.`pof`
111  13:49:17  PREPARE stmt FROM 'INSERT INTO `aircargo`.`pof` (`customer_id`,`aircraft_id`,`route_id`,`depart`,`arrival`,`seat_num`,`class_id`,`travel_date`,`flight_nu...
```



```sql
load data local infile 'C:\Program Files\MySQL\MySQL Workbench 8.0\data\datasets\ticket_details.csv'
into table ticket_details
fields terminated by ',' enclosed by '"' lines terminated by '\n' ignore 1 rows;

select * from ticket_details;
```

| tkt_id | p_date | customer_id | aircraft_id | class_id | no_of_tkts | a_code | price_per_tkt | brand |
|---|---|---|---|---|---|---|---|---|
| 1 | 2018-12-26 | 27 | 767-301ER | Economy | 1 | DAL | 130.00 | Emirates |
| 2 | 2020-02-02 | 22 | ERJ142 | Economy Plus | 1 | AGB | 220.00 | Jet Airways |
| 3 | 2020-03-03 | 21 | CRJ900 | Bussiness | 1 | BOH | 490.00 | Bristish Airways |
| 4 | 2020-04-04 | 4 | 767-301ER | First Class | 1 | AGB | 390.00 | Emirates |
| 5 | 2020-05-05 | 5 | ERJ142 | Economy | 1 | CTM | 120.00 | Jet Airways |
| 6 | 2020-07-07 | 7 | 767-301ER | Bussiness | 1 | BFS | 430.00 | Emirates |
| 7 | 2020-08-08 | 8 | A321 | Economy Plus | 1 | DAL | 275.00 | Qatar Airways |
| 8 | 2020-09-09 | 9 | 767-301ER | First Class | 1 | BOH | 380.00 | Emirates |
| 9 | 2020-10-10 | 10 | A321 | Economy | 1 | MCO | 135.00 | Qatar Airways |
| 10 | 2020-11-11 | 11 | 767-301ER | Bussiness | 1 | AGB | 465.00 | Emirates |
| 11 | 2020-12-12 | 19 | CRJ900 | Economy Plus | 1 | DEN | 225.00 | Bristish Airways |
| 12 | 2019-01-01 | 13 | A321 | First Class | 1 | YVR | 395.00 | Qatar Airways |
| 13 | 2019-02-02 | 14 | ERJ142 | Economy | 1 | CTM | 120.00 | Jet Airways |
| 14 | 2019-03-03 | 25 | 767-301ER | Bussiness | 1 | BHX | 499.00 | Emirates |
| 15 | 2019-04-04 | 16 | CRJ900 | First Class | 1 | YVR | 395.00 | Bristish Airways |
| 16 | 2019-05-03 | 17 | A321 | Economy Plus | 1 | BFS | 250.00 | Qatar Airways |
| 17 | 2019-06-06 | 18 | 767-301ER | Economy | 1 | YVR | 190.00 | Emirates |
| 18 | 2019-07-07 | 24 | A321 | Bussiness | 1 | CTM | 480.00 | Qatar Airways |
| 19 | 2019-08-09 | 20 | CRJ900 | First Class | 1 | MCO | 365.00 | Bristish Airways |
| 20 | 2019-09-21 | 25 | 767-301ER | Economy | 1 | BOH | 150.00 | Emirates |
| 21 | 2019-10-22 | 29 | A321 | Bussiness | 1 | PEK | 410.00 | Qatar Airways |
| 22 | 2019-11-23 | 1 | ERJ142 | Economy Plus | 1 | BFS | 250.00 | Jet Airways |
| 23 | 2019-12-24 | 14 | 767-301ER | Economy | 1 | BHX | 170.00 | Emirates |
| 24 | 2019-01-25 | 2 | A321 | Bussiness | 1 | YVR | 505.00 | Qatar Airways |
| 25 | 2018-01-01 | 9 | CRJ900 | First Class | 1 | AGB | 390.00 | British Airways |
| 26 | 2018-02-01 | 19 | 767-301ER | Economy | 1 | AGB | 100.00 | Emirates |
| 27 | 2018-03-01 | 18 | 767-301ER | First Class | 1 | BFS | 375.00 | Emirates |
| 28 | 2018-04-01 | 29 | ERJ142 | Bussiness | 1 | EMF | 510.00 | Jet Airways |

Table: ticket_details

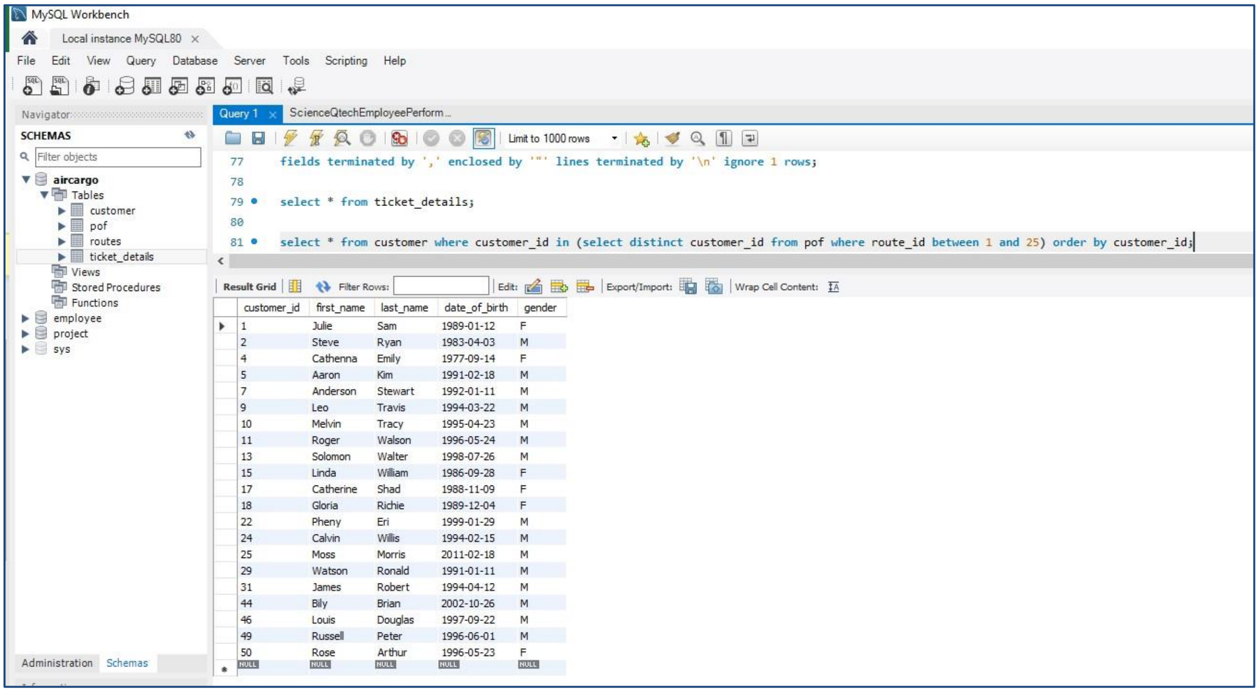
Columns:

| | |
|---|---|
| tkt_id | int AI PK |
| p_date | date |
| customer_id | int |
| aircraft_id | varchar(10) |
| class_id | varchar(15) |
| no_of_tkts | int |
| a_code | char(3) |
| price_per_tkt | decimal(5,2) |
| brand | varchar(30) |

```
#    Time      Action                                          Message
118  13:56:01  SHOW SESSION VARIABLES LIKE 'lower_case_table_names'  OK
```

# Air Cargo Analysis

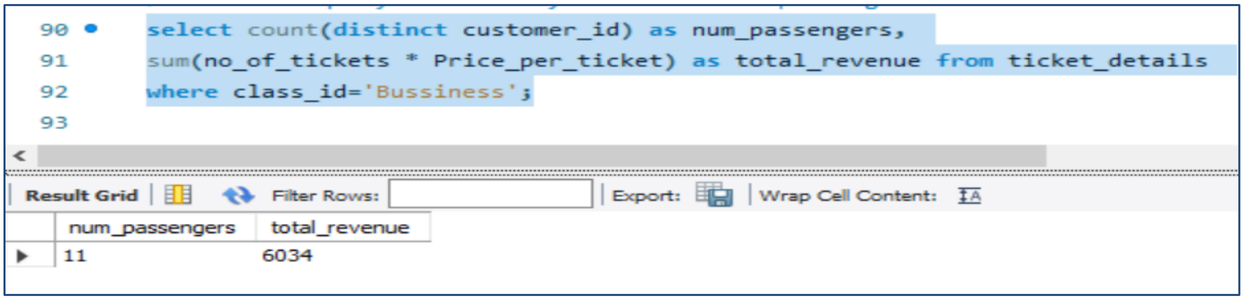
3. Write a query to display all the passengers (customers) who have travelled in routes 01 to 25. Take data from the passengers_on_flights table.



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

select * from ticket_details;

select * from customer where customer_id in (select distinct customer_id from pof where route_id between 1 and 25) order by customer_id;
```

| customer_id | first_name | last_name | date_of_birth | gender |
|---|---|---|---|---|
| 1 | Julie | Sam | 1989-01-12 | F |
| 2 | Steve | Ryan | 1983-04-03 | M |
| 4 | Cathenna | Emily | 1977-09-14 | F |
| 5 | Aaron | Kim | 1991-02-18 | M |
| 7 | Anderson | Stewart | 1992-01-11 | M |
| 9 | Leo | Travis | 1994-03-22 | M |
| 10 | Melvin | Tracy | 1995-04-23 | M |
| 11 | Roger | Walson | 1996-05-24 | M |
| 13 | Solomon | Walter | 1998-07-26 | M |
| 15 | Linda | William | 1986-09-28 | F |
| 17 | Catherine | Shad | 1988-11-09 | F |
| 18 | Gloria | Richie | 1989-12-04 | F |
| 22 | Pheny | Eri | 1999-01-29 | M |
| 24 | Calvin | Willis | 1994-02-15 | M |
| 25 | Moss | Morris | 2011-02-18 | M |
| 29 | Watson | Ronald | 1991-01-11 | M |
| 31 | James | Robert | 1994-04-12 | M |
| 44 | Bily | Brian | 2002-10-26 | M |
| 46 | Louis | Douglas | 1997-09-22 | M |
| 49 | Russell | Peter | 1996-06-01 | M |
| 50 | Rose | Arthur | 1996-05-23 | F |
| NULL | NULL | NULL | NULL | NULL |

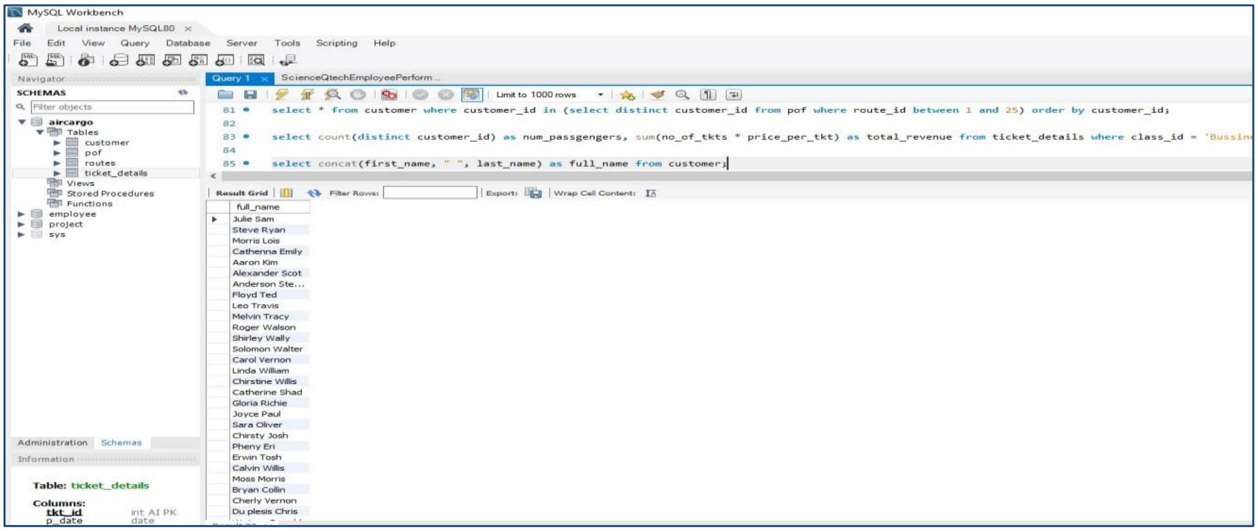
4. Write a query to identify the number of passengers and total revenue in business class from the ticket_details table.



```sql
select count(distinct customer_id) as num_passengers,
sum(no_of_tickets * Price_per_ticket) as total_revenue from ticket_details
where class_id='Bussiness';
```

| num_passengers | total_revenue |
|---|---|
| 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.

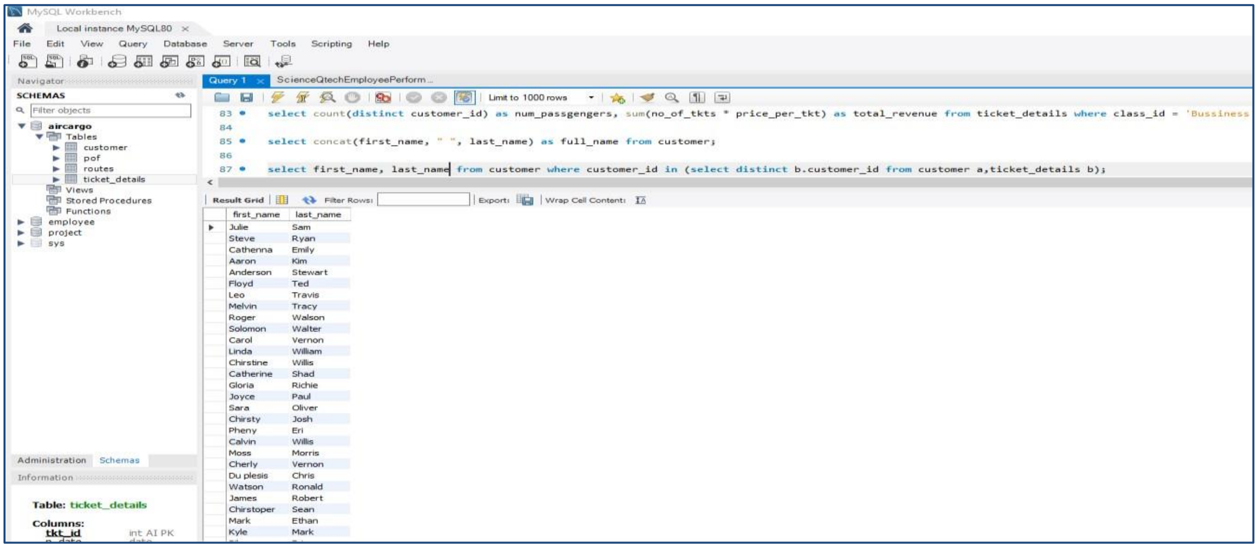


```sql
select * from customer where customer_id in (select distinct customer_id from pof where route_id between 1 and 25) order by customer_id;

select count(distinct customer_id) as num_passgengers, sum(no_of_tkts * price_per_tkt) as total_revenue from ticket_details where class_id = 'Bussin

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

| full_name |
|---|
| Julie Sam |
| Steve Ryan |
| Morris Lois |
| Cathenna Emily |
| Aaron Kim |
| Alexander Scot |
| Anderson Ste... |
| Floyd Ted |
| Leo Travis |
| Melvin Tracy |
| Roger Walson |
| Shirley Wally |
| Solomon Walter |
| Carol Vernon |
| Linda William |
| Chirstine Willis |
| Catherine Shad |
| Gloria Richie |
| Joyce Paul |
| Sara Oliver |
| Chirsty Josh |
| Pheny Eri |
| Erwin Tosh |
| Calvin Willis |
| Moss Morris |
| Bryan Collin |
| Cherly Vernon |
| Du plesis Chris |

# Air Cargo Analysis

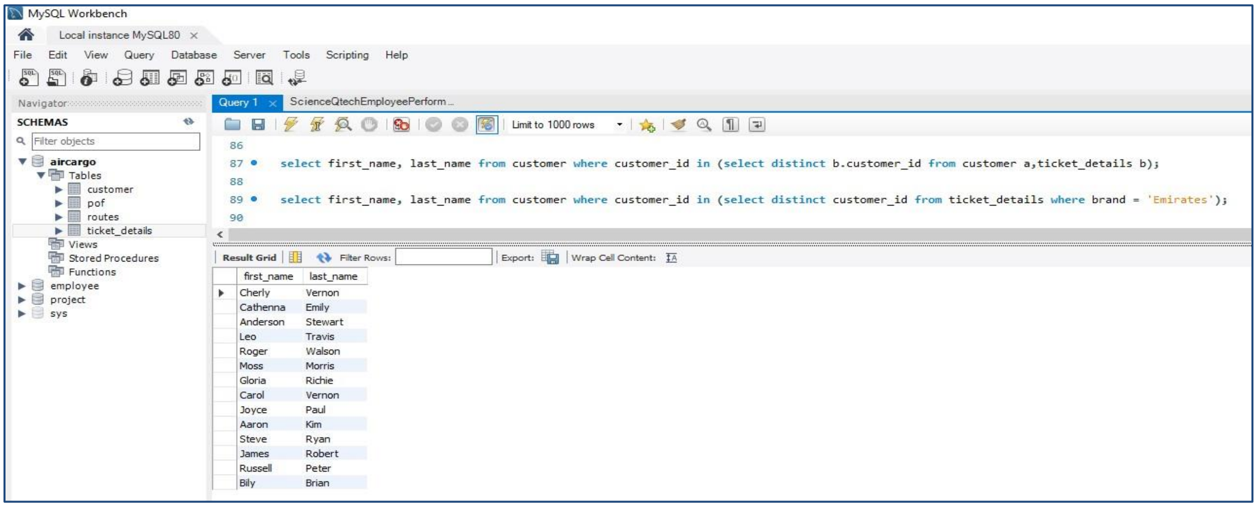
6. Write a query to extract the customers who have registered and booked a ticket. Use data fromthe customer and ticket_details tables.



```sql
select count(distinct customer_id) as num_passgengers, sum(no_of_tkts * price_per_tkt) as total_revenue from ticket_details where class_id = 'Bussiness

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

select first_name, last_name from customer where customer_id in (select distinct b.customer_id from customer a,ticket_details b);
```

| first_name | last_name |
|---|---|
| Julie | Sam |
| Steve | Ryan |
| Cathenna | Emily |
| Aaron | Kim |
| Anderson | Stewart |
| Floyd | Ted |
| Leo | Travis |
| Melvin | Tracy |
| Roger | Walson |
| Solomon | Walter |
| Carol | Vernon |
| Linda | William |
| Chirstine | Willis |
| Catherine | Shad |
| Gloria | Richie |
| Joyce | Paul |
| Sara | Oliver |
| Chirsty | Josh |
| Pheny | Eri |
| Calvin | Willis |
| Moss | Morris |
| Cherly | Vernon |
| Du plesis | Chris |
| Watson | Ronald |
| James | Robert |
| Chirstoper | Sean |
| Mark | Ethan |
| Kyle | Mark |

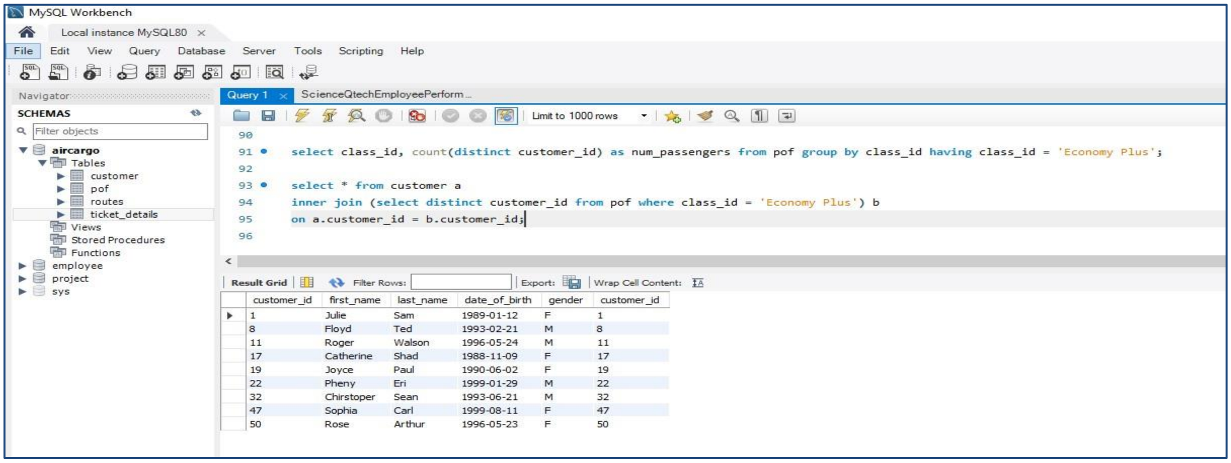
7. Write a query to identify the customer's first name and last name based on their customer ID and brand (Emirates) from the ticket_details table.



```sql
select first_name, last_name from customer where customer_id in (select distinct b.customer_id from customer a,ticket_details b);

select first_name, last_name from customer where customer_id in (select distinct customer_id from ticket_details where brand = 'Emirates');
```

| first_name | last_name |
|---|---|
| Cherly | Vernon |
| Cathenna | Emily |
| Anderson | Stewart |
| Leo | Travis |
| Roger | Walson |
| Moss | Morris |
| Gloria | Richie |
| Carol | Vernon |
| Joyce | Paul |
| Aaron | Kim |
| Steve | Ryan |
| James | Robert |
| Russell | Peter |
| Bily | Brian |

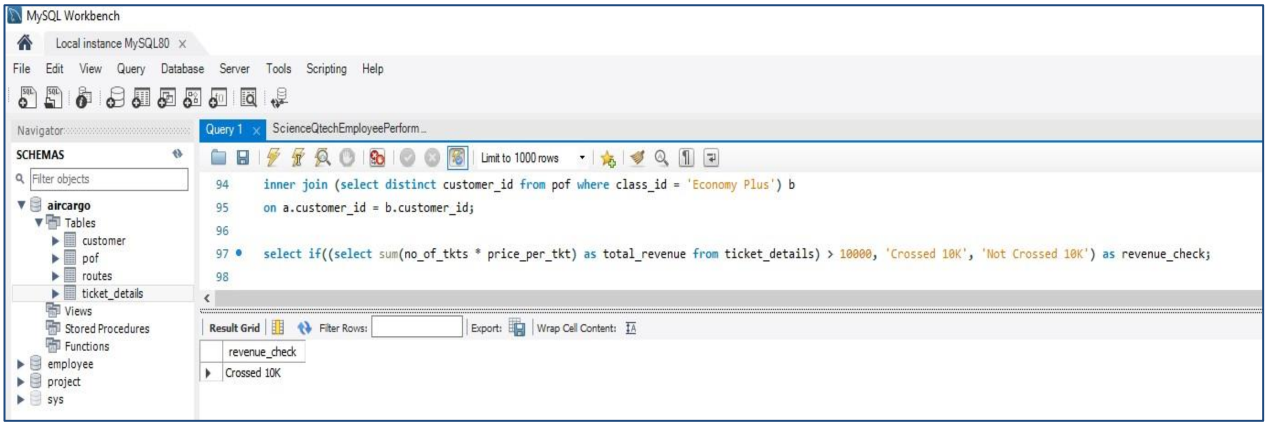
8. Write a query to identify the customers who have travelled by *Economy Plus* class using GroupBy and Having clause on the passengers_on_flights table.



```sql
select class_id, count(distinct customer_id) as num_passengers from pof group by class_id having class_id = 'Economy Plus';

select * from customer a
inner join (select distinct customer_id from pof where class_id = 'Economy Plus') b
on a.customer_id = b.customer_id;
```

| customer_id | first_name | last_name | date_of_birth | gender | customer_id |
|---|---|---|---|---|---|
| 1 | Julie | Sam | 1989-01-12 | F | 1 |
| 8 | Floyd | Ted | 1993-02-21 | M | 8 |
| 11 | Roger | Walson | 1996-05-24 | M | 11 |
| 17 | Catherine | Shad | 1988-11-09 | F | 17 |
| 19 | Joyce | Paul | 1990-06-02 | F | 19 |
| 22 | Pheny | Eri | 1999-01-29 | M | 22 |
| 32 | Chirstoper | Sean | 1993-06-21 | M | 32 |
| 47 | Sophia | Carl | 1999-08-11 | F | 47 |
| 50 | Rose | Arthur | 1996-05-23 | F | 50 |

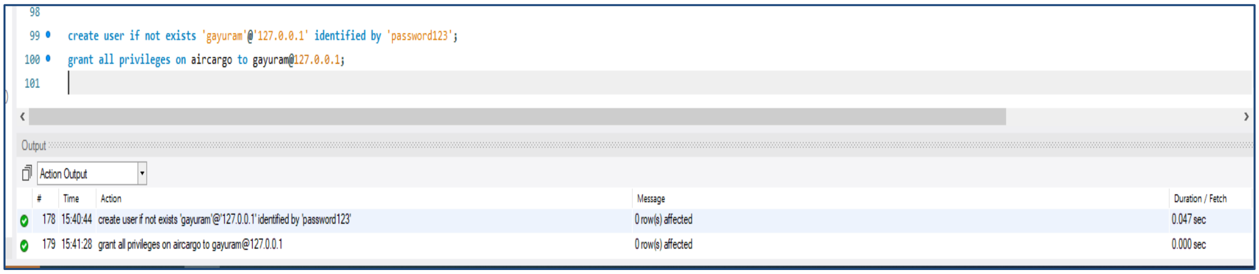
9. Write a query to identify whether the revenue has crossed 10000 using the IF clause on the ticket_details table.



```sql
inner join (select distinct customer_id from pof where class_id = 'Economy Plus') b
on a.customer_id = b.customer_id;

select if((select sum(no_of_tkts * price_per_tkt) as total_revenue from ticket_details) > 10000, 'Crossed 10K', 'Not Crossed 10K') as revenue_check;
```

| revenue_check |
|---|
| Crossed 10K |

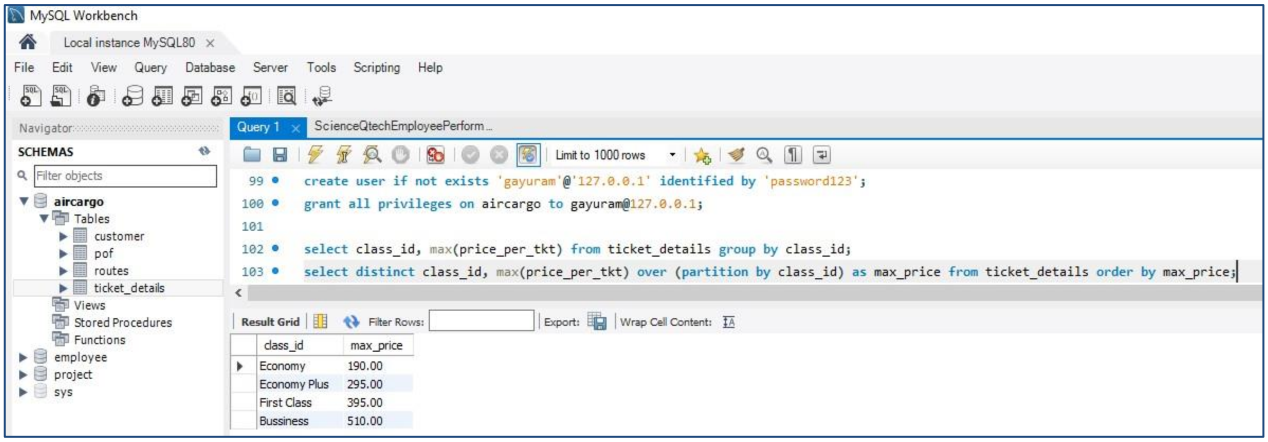
10. Write a query to create and grant access to a new user to perform operations on a database.



```sql
create user if not exists 'gayuram'@'127.0.0.1' identified by 'password123';
grant all privileges on aircargo to gayuram@127.0.0.1;
```

```
Output
Action Output
#    Time      Action                                                                  Message               Duration / Fetch
178  15:40:44  create user if not exists 'gayuram'@'127.0.0.1' identified by 'password123'  0 row(s) affected  0.047 sec
179  15:41:28  grant all privileges on aircargo to gayuram@127.0.0.1                       0 row(s) affected  0.000 sec
```

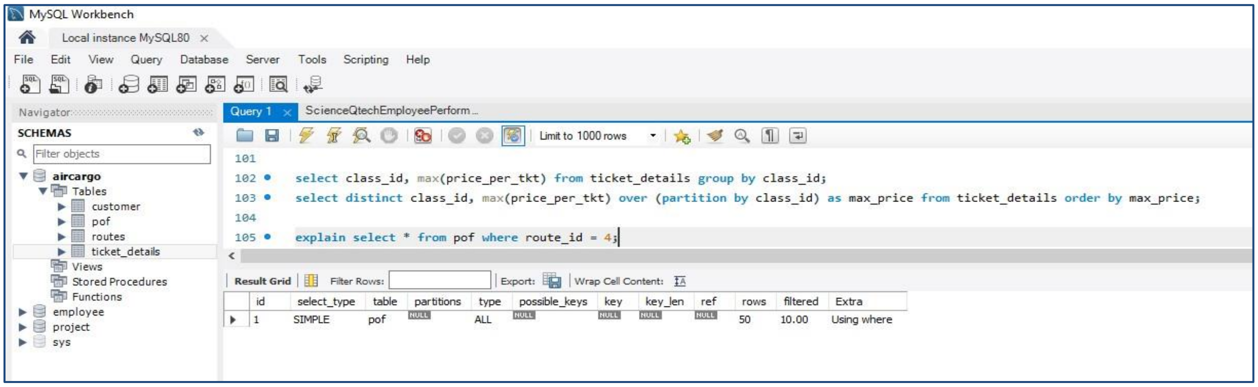
11. Write a query to find the maximum ticket price for each class using window functions on the ticket_details table.



```sql
create user if not exists 'gayuram'@'127.0.0.1' identified by 'password123';
grant all privileges on aircargo to gayuram@127.0.0.1;

select class_id, max(price_per_tkt) from ticket_details group by class_id;
select distinct class_id, max(price_per_tkt) over (partition by class_id) as max_price from ticket_details order by max_price;
```

| class_id | max_price |
|---|---|
| Economy | 190.00 |
| Economy Plus | 295.00 |
| First Class | 395.00 |
| Bussiness | 510.00 |

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

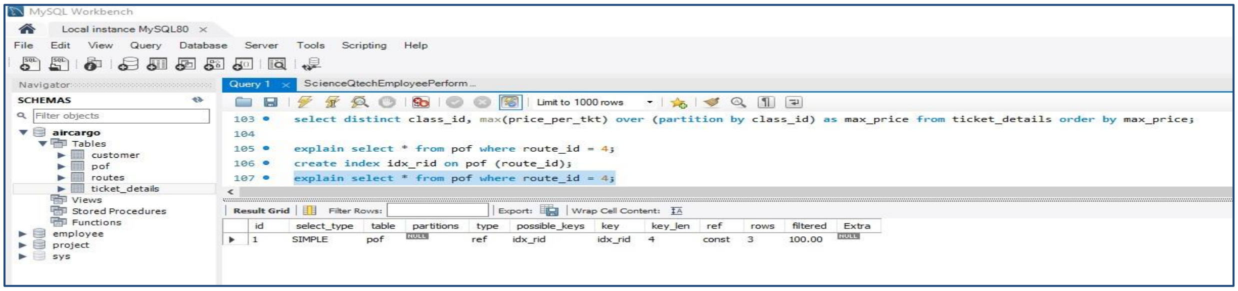


```sql
select class_id, max(price_per_tkt) from ticket_details group by class_id;
select distinct class_id, max(price_per_tkt) over (partition by class_id) as max_price from ticket_details order by max_price;

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

| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | SIMPLE | pof | NULL | ALL | NULL | NULL | NULL | NULL | 50 | 10.00 | Using where |

# Air Cargo Analysis

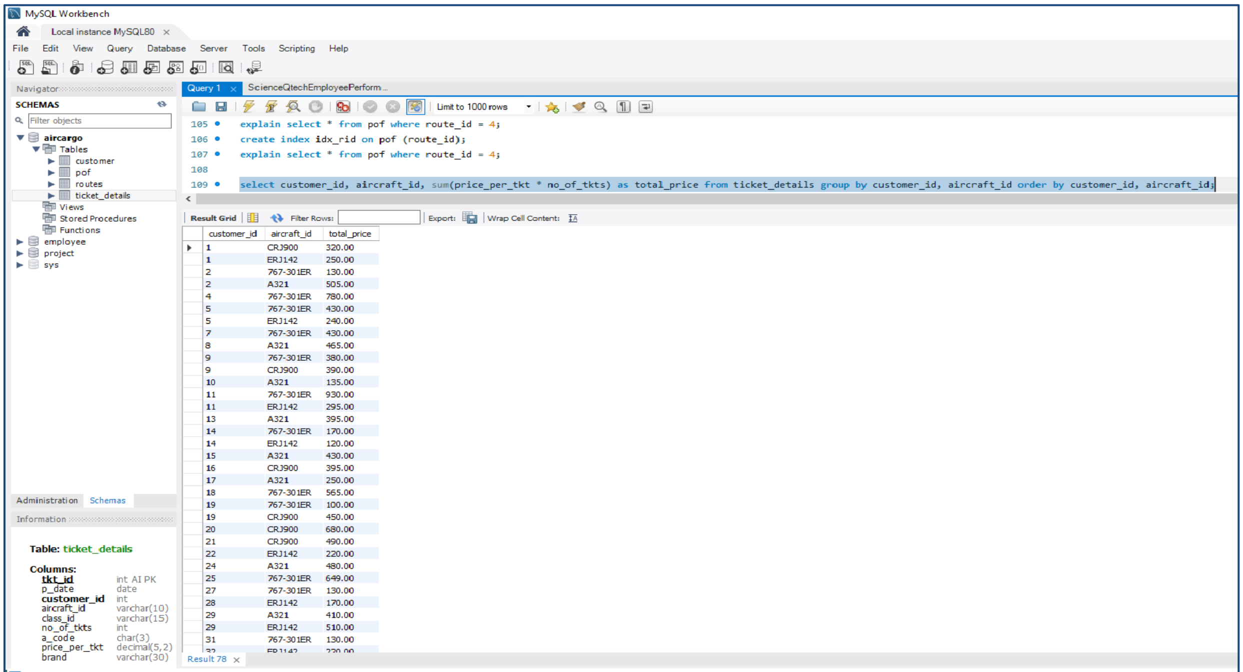
13. Write a query to extract the passengers whose route ID is 4 by improving the speed and performance of the passengers_on_flights table.



```
select distinct class_id, max(price_per_tkt) over (partition by class_id) as max_price from ticket_details order by max_price;

explain select * from pof where route_id = 4;
create index idx_rid on pof (route_id);
explain select * from pof where route_id = 4;
```

| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | SIMPLE | pof | NULL | ref | idx_rid | idx_rid | 4 | 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.



```
explain select * from pof where route_id = 4;
create index idx_rid on pof (route_id);
explain select * from pof where route_id = 4;

select customer_id, aircraft_id, sum(price_per_tkt * no_of_tkts) as total_price from ticket_details group by customer_id, aircraft_id order by customer_id, aircraft_id;
```

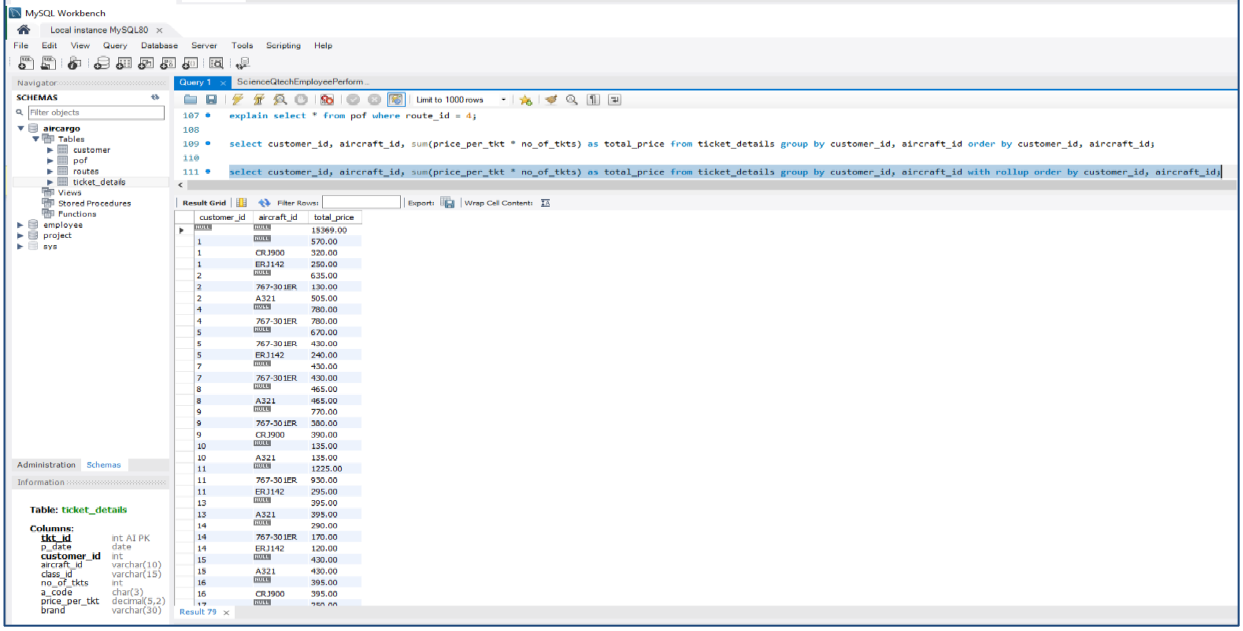
| customer_id | aircraft_id | total_price |
|---|---|---|
| 1 | CRJ900 | 320.00 |
| 1 | ERJ142 | 250.00 |
| 2 | 767-301ER | 130.00 |
| 2 | A321 | 505.00 |
| 4 | 767-301ER | 780.00 |
| 5 | 767-301ER | 430.00 |
| 5 | ERJ142 | 240.00 |
| 7 | 767-301ER | 430.00 |
| 8 | A321 | 465.00 |
| 9 | 767-301ER | 380.00 |
| 9 | CRJ900 | 390.00 |
| 10 | A321 | 135.00 |
| 11 | 767-301ER | 930.00 |
| 11 | ERJ142 | 295.00 |
| 13 | A321 | 395.00 |
| 14 | 767-301ER | 170.00 |
| 14 | ERJ142 | 120.00 |
| 15 | A321 | 430.00 |
| 16 | CRJ900 | 395.00 |
| 17 | A321 | 250.00 |
| 18 | 767-301ER | 565.00 |
| 19 | 767-301ER | 100.00 |
| 19 | CRJ900 | 450.00 |
| 20 | CRJ900 | 680.00 |
| 21 | CRJ900 | 490.00 |
| 22 | ERJ142 | 220.00 |
| 24 | A321 | 480.00 |
| 25 | 767-301ER | 649.00 |
| 27 | 767-301ER | 130.00 |
| 28 | ERJ142 | 170.00 |
| 29 | A321 | 410.00 |
| 29 | ERJ142 | 510.00 |
| 31 | 767-301ER | 130.00 |
| 32 | ERJ142 | 220.00 |

Result 78

Table: ticket_details

Columns:

| | |
|---|---|
| tkt_id | int AI PK |
| p_date | date |
| customer_id | int |
| aircraft_id | varchar(10) |
| class_id | varchar(15) |
| no_of_tkts | int |
| a_code | char(3) |
| price_per_tkt | decimal(5,2) |
| brand | varchar(30) |



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

select customer_id, aircraft_id, sum(price_per_tkt * no_of_tkts) as total_price from ticket_details group by customer_id, aircraft_id order by customer_id, aircraft_id;

select customer_id, aircraft_id, sum(price_per_tkt * no_of_tkts) as total_price from ticket_details group by customer_id, aircraft_id with rollup order by customer_id, aircraft_id;
```

| customer_id | aircraft_id | total_price |
|---|---|---|
| NULL | NULL | 15369.00 |
| 1 | NULL | 570.00 |
| 1 | CRJ900 | 320.00 |
| 1 | ERJ142 | 250.00 |
| 2 | NULL | 635.00 |
| 2 | 767-301ER | 130.00 |
| 2 | A321 | 505.00 |
| 4 | NULL | 780.00 |
| 4 | 767-301ER | 780.00 |
| 5 | NULL | 670.00 |
| 5 | 767-301ER | 430.00 |
| 5 | ERJ142 | 240.00 |
| 7 | NULL | 430.00 |
| 7 | 767-301ER | 430.00 |
| 8 | NULL | 465.00 |
| 8 | A321 | 465.00 |
| 9 | NULL | 770.00 |
| 9 | 767-301ER | 380.00 |
| 9 | CRJ900 | 390.00 |
| 10 | NULL | 135.00 |
| 10 | A321 | 135.00 |
| 11 | NULL | 1225.00 |
| 11 | 767-301ER | 930.00 |
| 11 | ERJ142 | 295.00 |
| 13 | NULL | 395.00 |
| 13 | A321 | 395.00 |
| 14 | NULL | 290.00 |
| 14 | 767-301ER | 170.00 |
| 14 | ERJ142 | 120.00 |
| 15 | NULL | 430.00 |
| 15 | A321 | 430.00 |
| 16 | NULL | 395.00 |
| 16 | CRJ900 | 395.00 |
| 17 | NULL | 250.00 |

Result 79

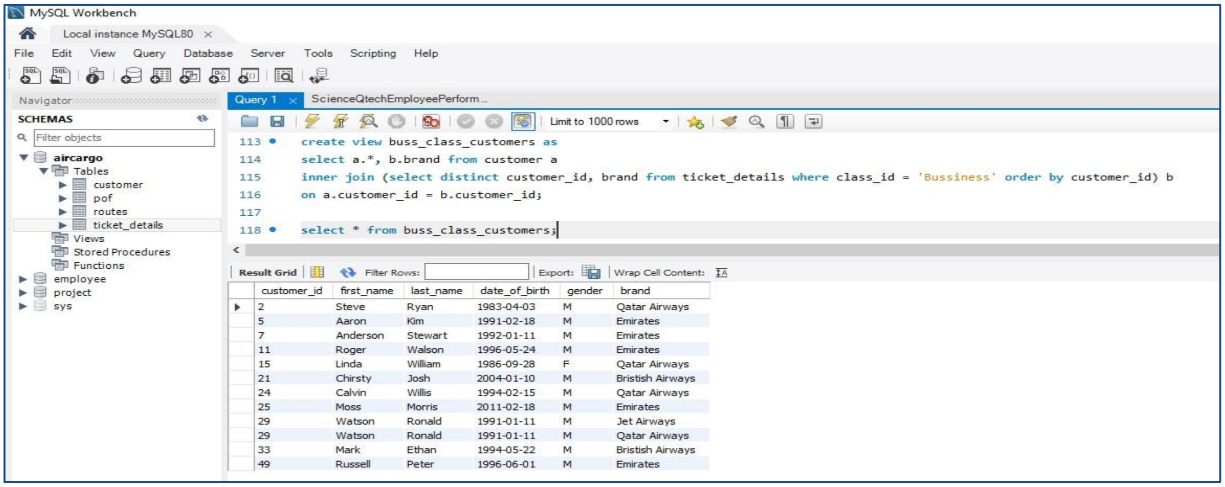
Table: ticket_details

Columns:

| | |
|---|---|
| tkt_id | int AI PK |
| p_date | date |
| customer_id | int |
| aircraft_id | varchar(10) |
| class_id | varchar(15) |
| no_of_tkts | int |
| a_code | char(3) |
| price_per_tkt | decimal(5,2) |
| brand | varchar(30) |

# Air Cargo Analysis

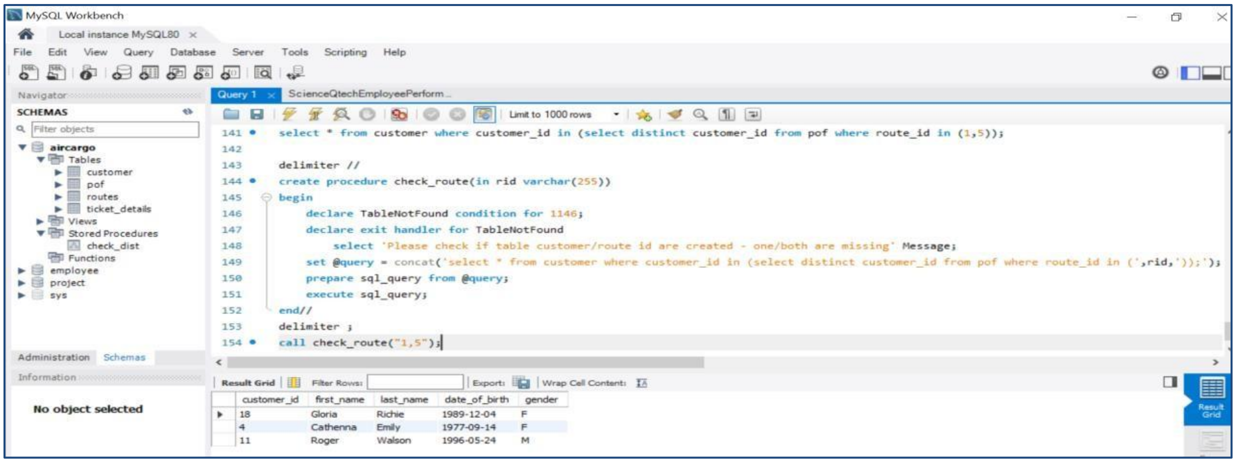
15. Write a query to create a view with only business class customers along with the brand of airlines.



```
create view buss_class_customers as
select a.*, b.brand from customer a
inner join (select distinct customer_id, brand from ticket_details where class_id = 'Bussiness' order by customer_id) b
on a.customer_id = b.customer_id;

select * from buss_class_customers;
```

| customer_id | first_name | last_name | date_of_birth | gender | brand |
|---|---|---|---|---|---|
| 2 | Steve | Ryan | 1983-04-03 | M | Qatar Airways |
| 5 | Aaron | Kim | 1991-02-18 | M | Emirates |
| 7 | Anderson | Stewart | 1992-01-11 | M | Emirates |
| 11 | Roger | Walson | 1996-05-24 | M | Emirates |
| 15 | Linda | William | 1986-09-28 | F | Qatar Airways |
| 21 | Chirsty | Josh | 2004-01-10 | M | Bristish Airways |
| 24 | Calvin | Willis | 1994-02-15 | M | Qatar Airways |
| 25 | Moss | Morris | 2011-02-18 | M | Emirates |
| 29 | Watson | Ronald | 1991-01-11 | M | Jet Airways |
| 29 | Watson | Ronald | 1991-01-11 | M | Qatar Airways |
| 33 | Mark | Ethan | 1994-05-22 | M | Bristish Airways |
| 49 | Russell | Peter | 1996-06-01 | M | Emirates |

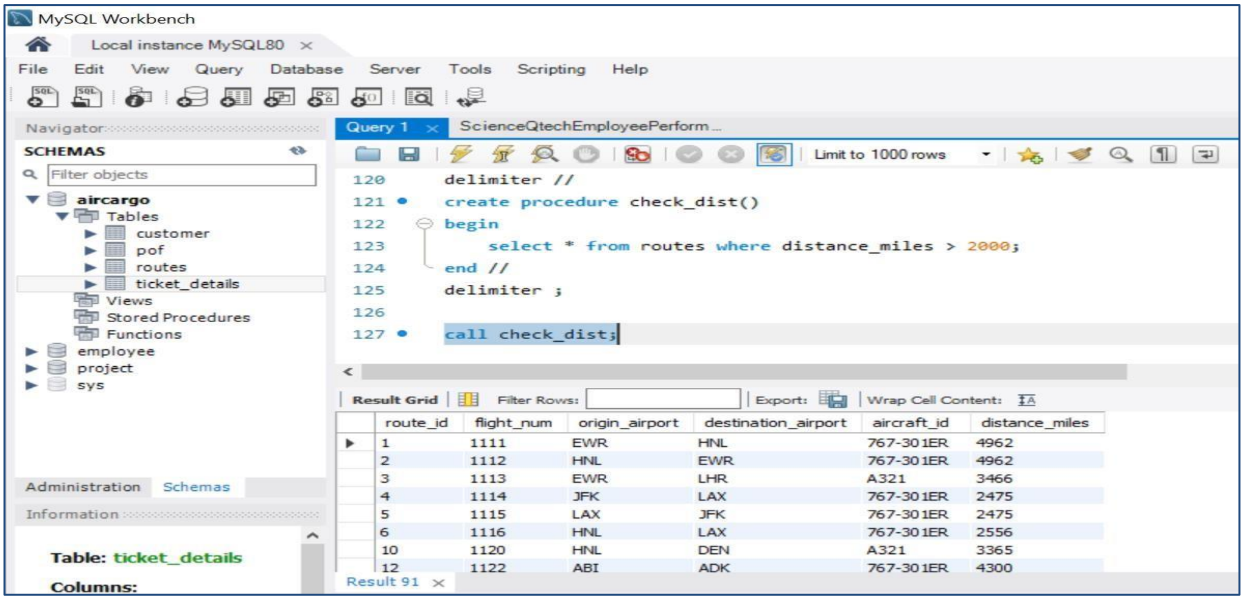
16. Write a query to create a stored procedure to get the details of all passengers flying between a range of routes defined in run time. Also, return an error message if the table doesn't exist.



```
select * from customer where customer_id in (select distinct customer_id from pof where route_id in (1,5));

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 - one/both are missing' Message;
    set @query = concat('select * from customer where customer_id in (select distinct customer_id from pof 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 |
|---|---|---|---|---|
| 18 | Gloria | Richie | 1989-12-04 | F |
| 4 | Cathenna | Emily | 1977-09-14 | F |
| 11 | Roger | Walson | 1996-05-24 | M |

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

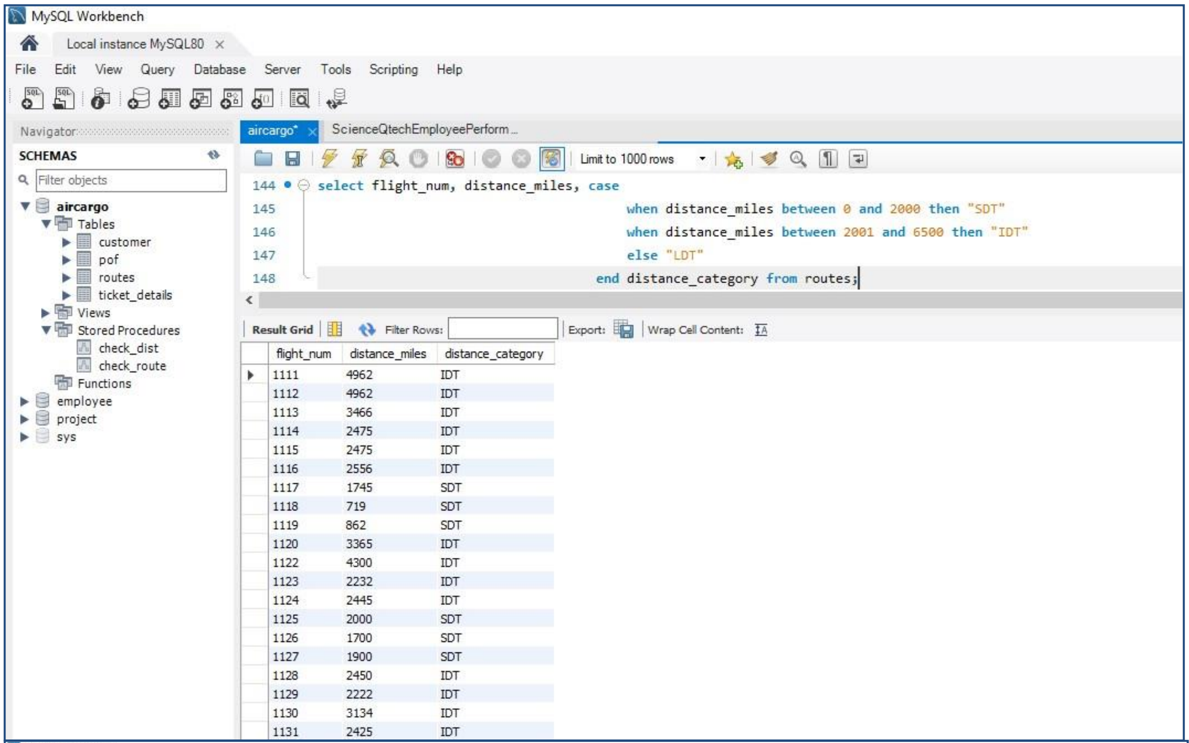


```
delimiter //
create procedure check_dist()
begin
    select * from routes where distance_miles > 2000;
end //
delimiter ;

call check_dist;
```

| 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 |
| 6 | 1116 | HNL | LAX | 767-301ER | 2556 |
| 10 | 1120 | HNL | DEN | A321 | 3365 |
| 12 | 1122 | ABI | ADK | 767-301ER | 4300 |

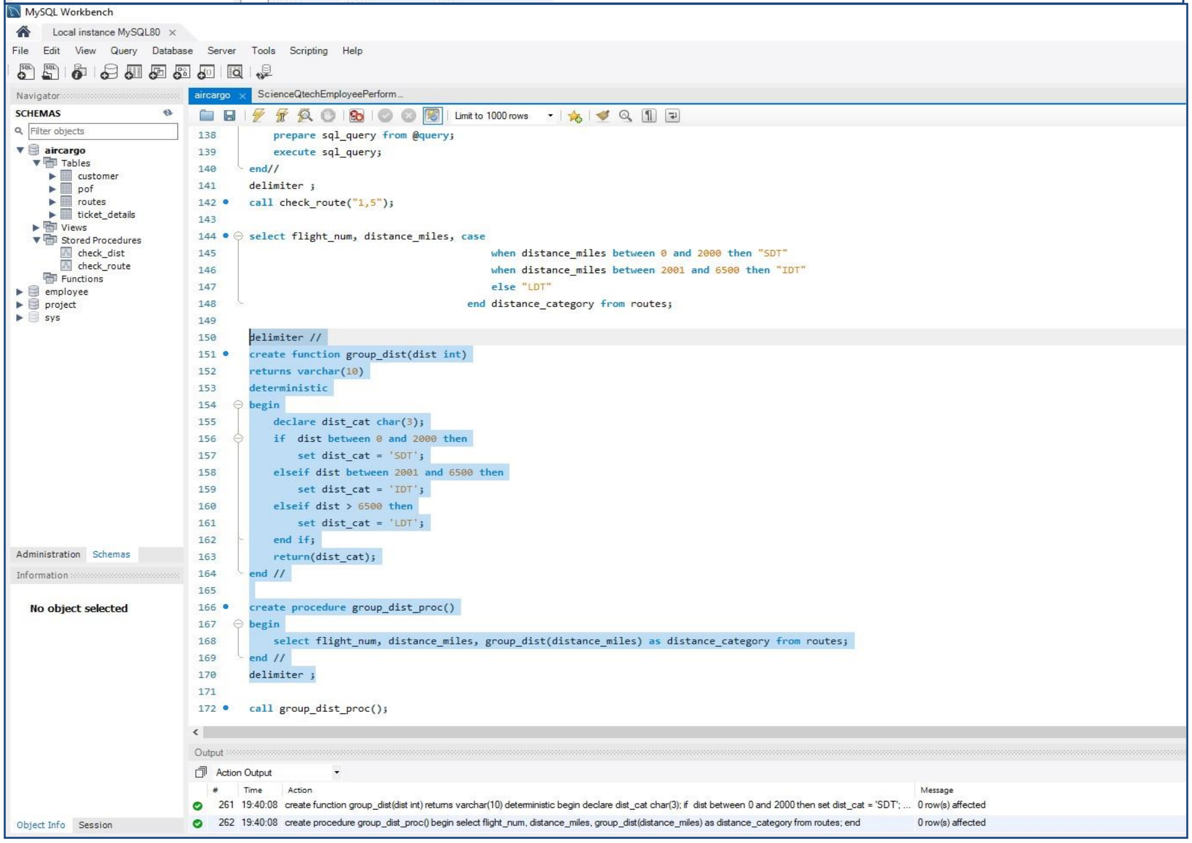
# Air Cargo Analysis

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.



```sql
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;
```

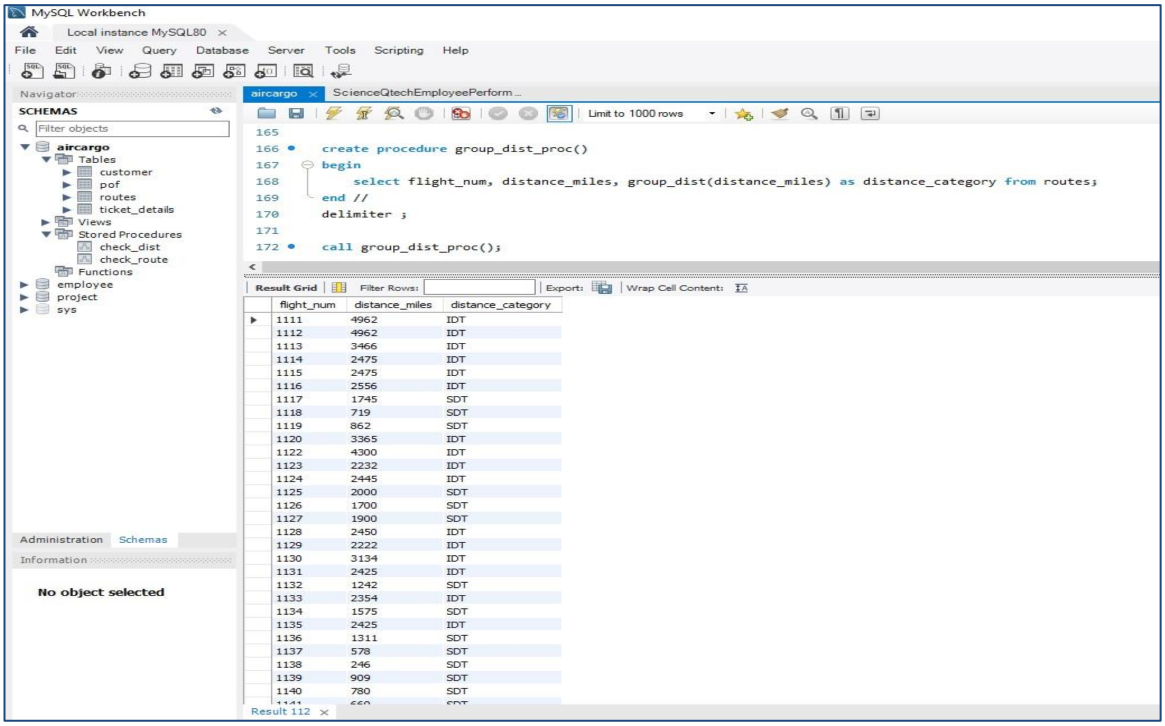
| flight_num | distance_miles | distance_category |
|---|---|---|
| 1111 | 4962 | IDT |
| 1112 | 4962 | IDT |
| 1113 | 3466 | IDT |
| 1114 | 2475 | IDT |
| 1115 | 2475 | IDT |
| 1116 | 2556 | IDT |
| 1117 | 1745 | SDT |
| 1118 | 719 | SDT |
| 1119 | 862 | SDT |
| 1120 | 3365 | IDT |
| 1122 | 4300 | IDT |
| 1123 | 2232 | IDT |
| 1124 | 2445 | IDT |
| 1125 | 2000 | SDT |
| 1126 | 1700 | SDT |
| 1127 | 1900 | SDT |
| 1128 | 2450 | IDT |
| 1129 | 2222 | IDT |
| 1130 | 3134 | IDT |
| 1131 | 2425 | IDT |



```sql
    prepare sql_query from @query;
    execute sql_query;
end//
delimiter ;
call check_route("1,5");

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;

delimiter //
create function group_dist(dist int)
returns varchar(10)
deterministic
begin
    declare dist_cat char(3);
    if  dist between 0 and 2000 then
        set dist_cat = 'SDT';
    elseif dist between 2001 and 6500 then
        set dist_cat = 'IDT';
    elseif dist > 6500 then
        set dist_cat = 'LDT';
    end if;
    return(dist_cat);
end //

create procedure group_dist_proc()
begin
    select flight_num, distance_miles, group_dist(distance_miles) as distance_category from routes;
end //
delimiter ;

call group_dist_proc();
```

| # | Time | Action | Message |
|---|---|---|---|
| 261 | 19:40:08 | create function group_dist(dist int) returns varchar(10) deterministic begin declare dist_cat char(3); if  dist between 0 and 2000 then set dist_cat = 'SDT'; ... | 0 row(s) affected |
| 262 | 19:40:08 | create procedure group_dist_proc() begin select flight_num, distance_miles, group_dist(distance_miles) as distance_category from routes; end | 0 row(s) affected |

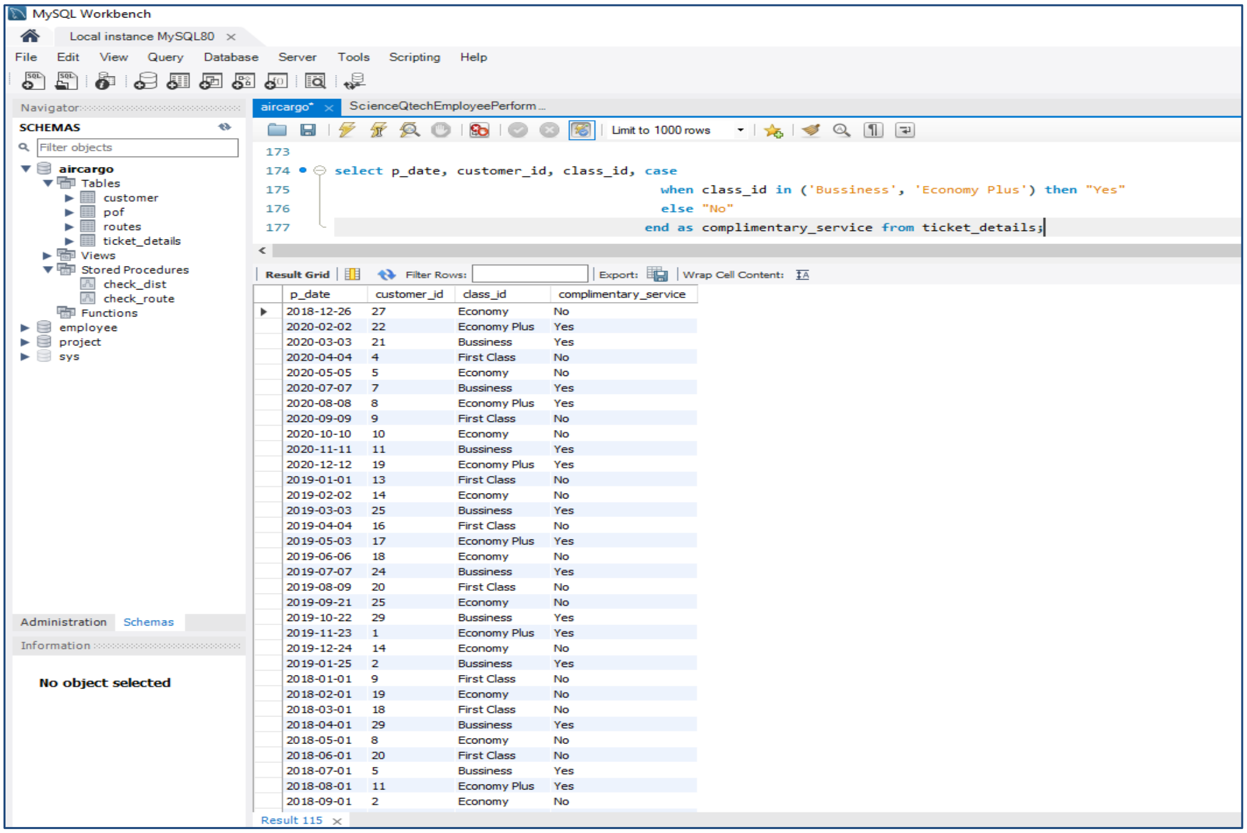
# Air Cargo Analysis



```sql
create procedure group_dist_proc()
begin
    select flight_num, distance_miles, group_dist(distance_miles) as distance_category from routes;
end //
delimiter ;

call group_dist_proc();
```

| 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 |
| 1118 | 719 | SDT |
| 1119 | 862 | SDT |
| 1120 | 3365 | IDT |
| 1122 | 4300 | IDT |
| 1123 | 2232 | IDT |
| 1124 | 2445 | IDT |
| 1125 | 2000 | SDT |
| 1126 | 1700 | SDT |
| 1127 | 1900 | SDT |
| 1128 | 2450 | IDT |
| 1129 | 2222 | IDT |
| 1130 | 3134 | IDT |
| 1131 | 2425 | IDT |
| 1132 | 1242 | SDT |
| 1133 | 2354 | IDT |
| 1134 | 1575 | SDT |
| 1135 | 2425 | IDT |
| 1136 | 1311 | SDT |
| 1137 | 578 | SDT |
| 1138 | 246 | SDT |
| 1139 | 909 | SDT |
| 1140 | 780 | SDT |

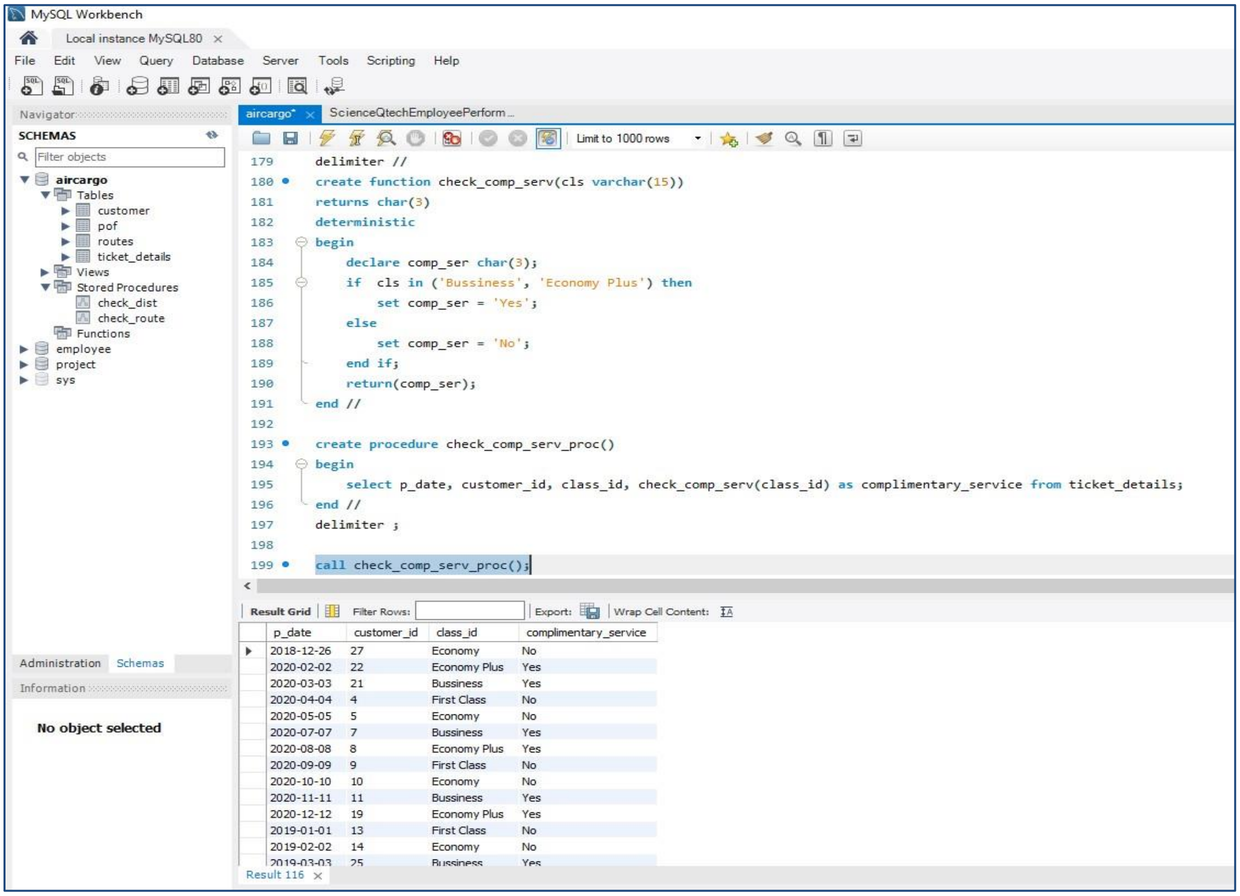
19. Write a query to extract ticket purchase date, customer ID, class ID and specify if the complimentary services are provided for the specific class using a stored function in stored procedure on the ticket_details table. Condition: If the class is *Business* and *Economy Plus,* thencomplimentary services are given as *Yes,* else it is *No*



```sql
select p_date, customer_id, class_id, case
                                        when class_id in ('Bussiness', 'Economy Plus') then "Yes"
                                        else "No"
                                      end as complimentary_service from ticket_details;
```

| p_date | customer_id | class_id | complimentary_service |
|---|---|---|---|
| 2018-12-26 | 27 | Economy | No |
| 2020-02-02 | 22 | Economy Plus | Yes |
| 2020-03-03 | 21 | Bussiness | Yes |
| 2020-04-04 | 4 | First Class | No |
| 2020-05-05 | 5 | Economy | No |
| 2020-07-07 | 7 | Bussiness | Yes |
| 2020-08-08 | 8 | Economy Plus | Yes |
| 2020-09-09 | 9 | First Class | No |
| 2020-10-10 | 10 | Economy | No |
| 2020-11-11 | 11 | Bussiness | Yes |
| 2020-12-12 | 19 | Economy Plus | Yes |
| 2019-01-01 | 13 | First Class | No |
| 2019-02-02 | 14 | Economy | No |
| 2019-03-03 | 25 | Bussiness | Yes |
| 2019-04-04 | 16 | First Class | No |
| 2019-05-03 | 17 | Economy Plus | Yes |
| 2019-06-06 | 18 | Economy | No |
| 2019-07-07 | 24 | Bussiness | Yes |
| 2019-08-09 | 20 | First Class | No |
| 2019-09-21 | 25 | Economy | No |
| 2019-10-22 | 29 | Bussiness | Yes |
| 2019-11-23 | 1 | Economy Plus | Yes |
| 2019-12-24 | 14 | Economy | No |
| 2019-01-25 | 2 | Bussiness | Yes |
| 2018-01-01 | 9 | First Class | No |
| 2018-02-01 | 19 | Economy | No |
| 2018-03-01 | 18 | First Class | No |
| 2018-04-01 | 29 | Bussiness | Yes |
| 2018-05-01 | 8 | Economy | No |
| 2018-06-01 | 20 | First Class | No |
| 2018-07-01 | 5 | Bussiness | Yes |
| 2018-08-01 | 11 | Economy Plus | Yes |
| 2018-09-01 | 2 | Economy | No |

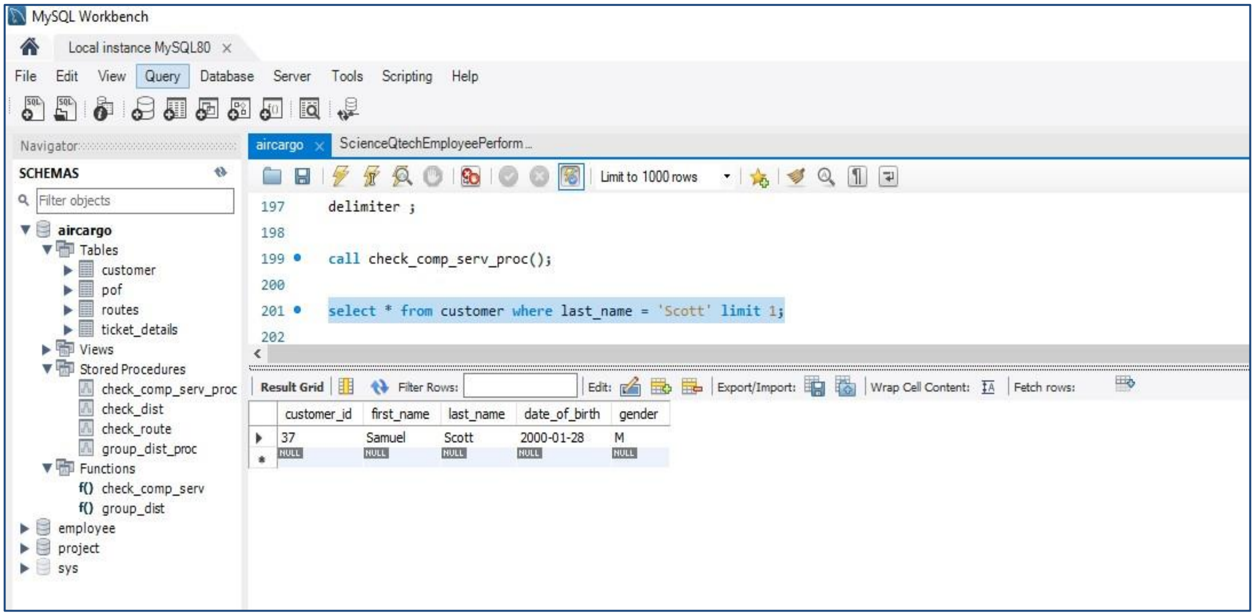
# Air Cargo Analysis



```
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 //
delimiter ;

call check_comp_serv_proc();
```

| p_date | customer_id | class_id | complimentary_service |
|---|---|---|---|
| 2018-12-26 | 27 | Economy | No |
| 2020-02-02 | 22 | Economy Plus | Yes |
| 2020-03-03 | 21 | Bussiness | Yes |
| 2020-04-04 | 4 | First Class | No |
| 2020-05-05 | 5 | Economy | No |
| 2020-07-07 | 7 | Bussiness | Yes |
| 2020-08-08 | 8 | Economy Plus | Yes |
| 2020-09-09 | 9 | First Class | No |
| 2020-10-10 | 10 | Economy | No |
| 2020-11-11 | 11 | Bussiness | Yes |
| 2020-12-12 | 19 | Economy Plus | Yes |
| 2019-01-01 | 13 | First Class | No |
| 2019-02-02 | 14 | Economy | No |
| 2019-03-03 | 25 | Bussiness | Yes |

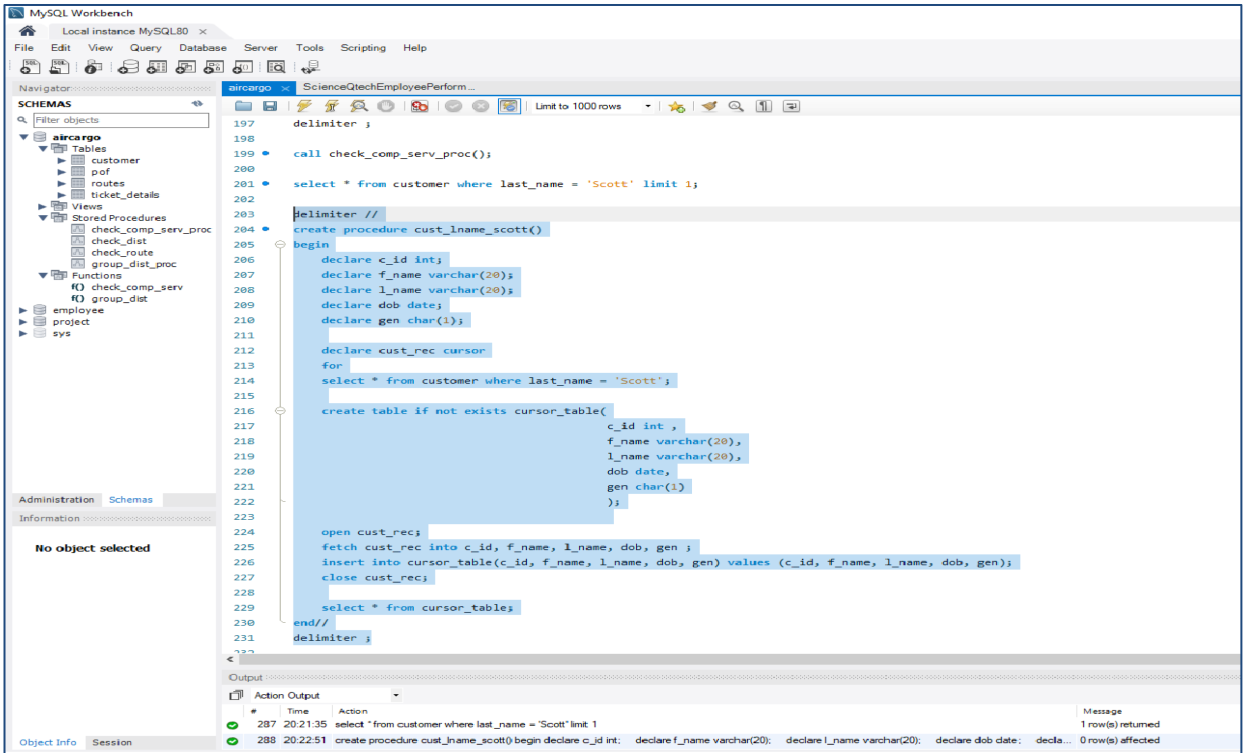
20. Write a query to extract the first record of the customer whose last name ends with Scott using a cursor from the customer table.



```
delimiter ;

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 | 2000-01-28 | M |
| NULL | NULL | NULL | NULL | NULL |

# Air Cargo Analysis

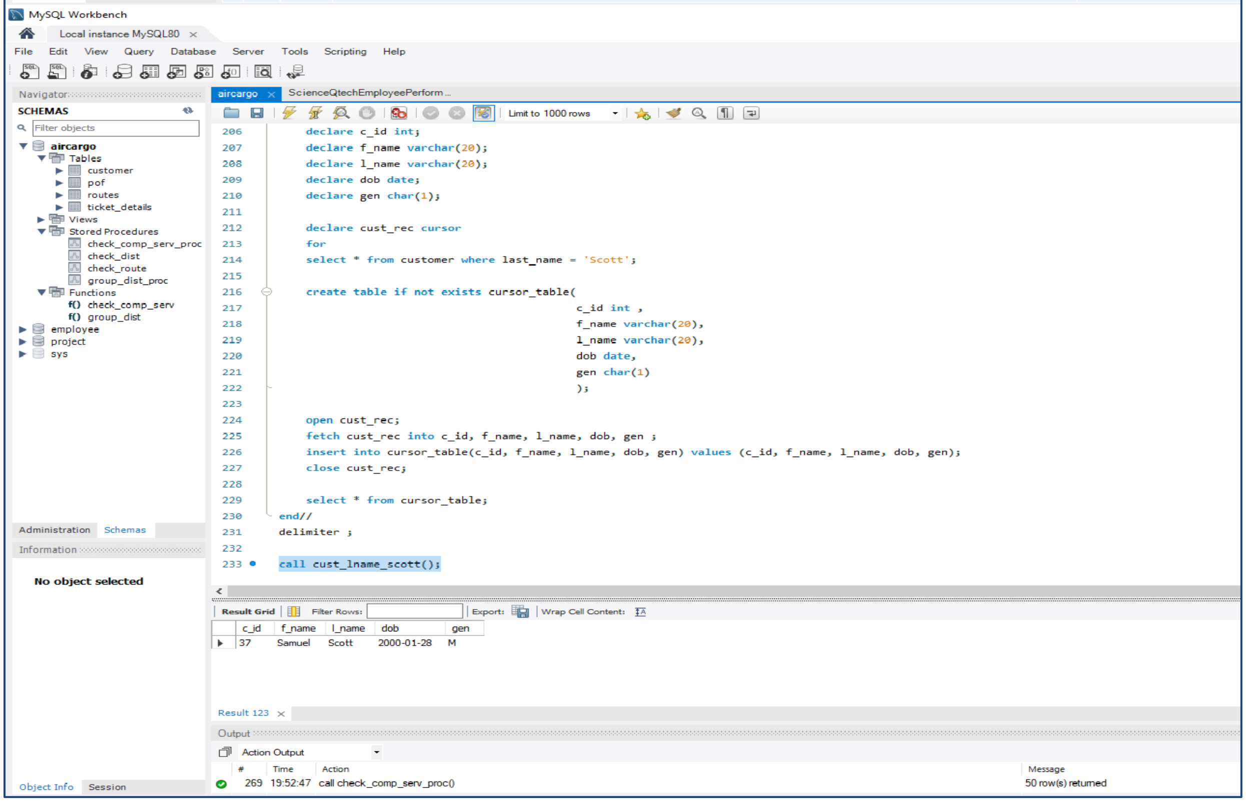


```sql
delimiter ;

call check_comp_serv_proc();

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

delimiter //
create procedure cust_lname_scott()
begin
    declare c_id int;
    declare f_name varchar(20);
    declare l_name varchar(20);
    declare dob date;
    declare gen char(1);

    declare cust_rec cursor
    for
    select * from customer where last_name = 'Scott';

    create table if not exists cursor_table(
                                            c_id int ,
                                            f_name varchar(20),
                                            l_name varchar(20),
                                            dob date,
                                            gen char(1)
                                            );

    open cust_rec;
    fetch cust_rec into c_id, f_name, l_name, dob, gen ;
    insert into cursor_table(c_id, f_name, l_name, dob, gen) values (c_id, f_name, l_name, dob, gen);
    close cust_rec;

    select * from cursor_table;
end//
delimiter ;
```

Output

Action Output

```
#    Time      Action                                                                                                                                          Message
287  20:21:35  select * from customer where last_name = 'Scott' limit 1                                                                                         1 row(s) returned
288  20:22:51  create procedure cust_lname_scott() begin declare c_id int;    declare f_name varchar(20);    declare l_name varchar(20);    declare dob date;    decla...  0 row(s) affected
```



```sql
    declare c_id int;
    declare f_name varchar(20);
    declare l_name varchar(20);
    declare dob date;
    declare gen char(1);

    declare cust_rec cursor
    for
    select * from customer where last_name = 'Scott';

    create table if not exists cursor_table(
                                            c_id int ,
                                            f_name varchar(20),
                                            l_name varchar(20),
                                            dob date,
                                            gen char(1)
                                            );

    open cust_rec;
    fetch cust_rec into c_id, f_name, l_name, dob, gen ;
    insert into cursor_table(c_id, f_name, l_name, dob, gen) values (c_id, f_name, l_name, dob, gen);
    close cust_rec;

    select * from cursor_table;
end//
delimiter ;

call cust_lname_scott();
```

| c_id | f_name | l_name | dob | gen |
|---|---|---|---|---|
| 37 | Samuel | Scott | 2000-01-28 | M |

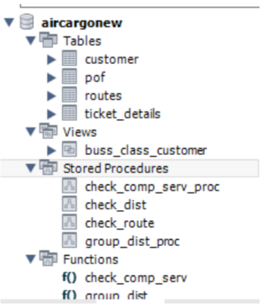
Output

Action Output

```
#    Time      Action                       Message
269  19:52:47  call check_comp_serv_proc()  50 row(s) returned
```

# Air Cargo Analysis

So, finally Schema Screen Shot is given below



```
▼ aircargonew
  ▼ Tables
    ▶ customer
    ▶ pof
    ▶ routes
    ▶ ticket_details
  ▼ Views
    ▶ buss_class_customer
  ▼ Stored Procedures
      check_comp_serv_proc
      check_dist
      check_route
      group_dist_proc
  ▼ Functions
      f() check_comp_serv
      f() group_dist
```

Done By,

**Kaushik Prasad Dey**