

1) Number of passenger who will be flying from airport id 13 :- 2

```
select count(booking_passenger.passenger_id),from_airport from booking_passenger natural join
```

```
booking join ticket on booking_passenger.ticket_id = ticket.ticket_id
```

```
group by from_airport having from_airport = 13
```

The screenshot shows a database query interface with a top navigation bar containing 'Dashboard', 'Properties', 'SQL', 'Statistics', 'Dependencies', 'Dependents', and 'Processes'. The current user is '202101418/202101418@Yash*'. Below the navigation bar is a toolbar with icons for file operations, filters, and execution. The 'Query' tab is active, displaying the following SQL query:

```
1 select count(booking_passenger.passenger_id),from_airport from booking_passenger natural join
2 booking join ticket on booking_passenger.ticket_id = ticket.ticket_id
3 group by from_airport having from_airport = 13
4
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query in a table format:

	count bigint	from_airport integer
1	2	13

2) Total Passengers whose check-in status was invalid

```
Select distinct passenger_name,passenger_contact from(checkin natural join ticket) as k
natural join passenger where status = 'NOT OK';
```

The screenshot shows a database query interface with a toolbar at the top. Below the toolbar, the results of a query are displayed in a table format:

	passenger_name character varying (20)	passenger_contact bigint
1	Alma	6841814182
2	Austin	9106306628
3	Cathrin	1851069732
4	Chancey	3788692103
5	Dacia	9965965982
6	Gery	9897248596
7	Horatius	6068105436
8	Jorgan	9702383724
9	Kinnie	2098846336
10	Louis	2984650608
11	Maddalena	8081150019
12	Madonna	9693745250
13	Mariann	9147657334
14	Martie	3707069534
15	Nolly	6264660814
16	Sara	6939007576
17	Valle	2787724446
18	Wayland	4955850400
19	Zackariah	2367307228

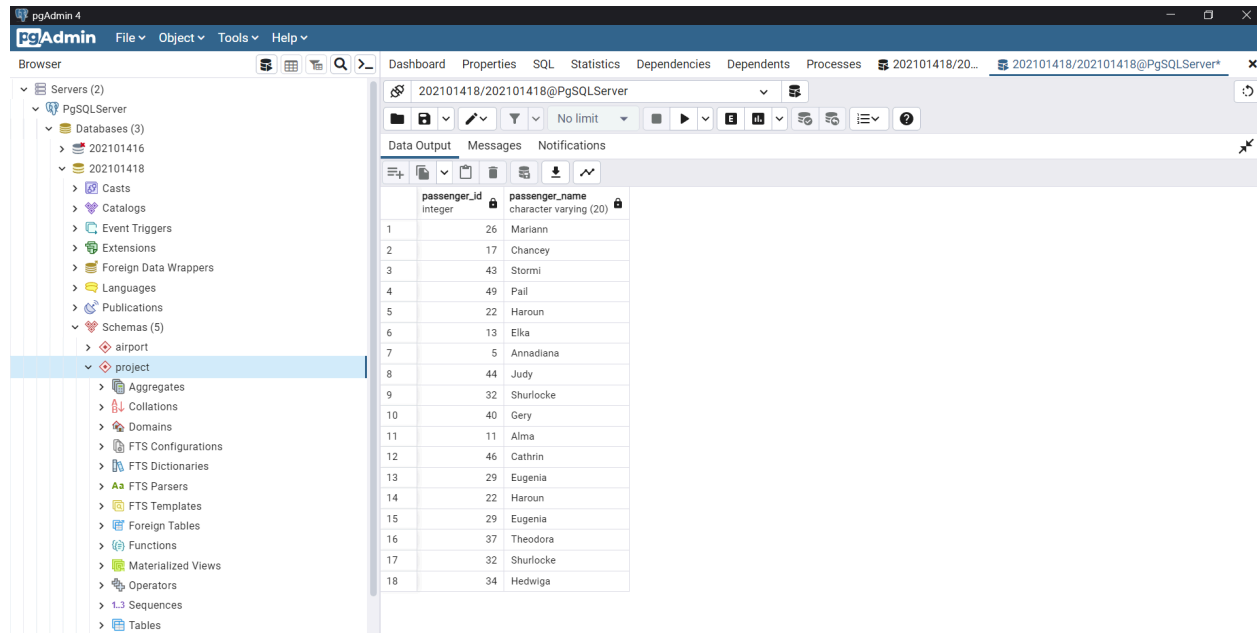
3)

People who missed their flight

select passenger_id,passenger_name from (((select ticket_id from ticket where status='confirmed'

Except

select ticket_id from checkin) as p natural join ticket) as k natural join passenger);



The screenshot shows the pgAdmin 4 interface. On the left, the 'Browsers' pane is expanded to show the 'project' schema under the 'airport' database. The main pane displays a query result table with two columns: 'passenger_id' (integer) and 'passenger_name' (character varying (20)). The table contains 18 rows of data.

passenger_id	passenger_name
1	26 Mariann
2	17 Chancey
3	43 Stormi
4	49 Pail
5	22 Haroun
6	13 Elka
7	5 Annadiana
8	44 Judy
9	32 Shurlocke
10	40 Gery
11	11 Alma
12	46 Cathrin
13	29 Eugenia
14	22 Haroun
15	29 Eugenia
16	37 Theodora
17	32 Shurlocke
18	34 Hedwiga

4) Project the flight_id and it's airlines name which got delayed on 20th April 2023

Select s.flight_id,k.airline_name

from ((flight natural join flight_status)k join schedule s on s.flight_id=k.flight_id)

where k.actual_arr_time>s.arrival_time and s.date_flight='2023-04-20';

The screenshot shows the pgAdmin 4 interface. On the left, the 'Columns (5)' folder is expanded under the 'flight' table, showing columns: departure_time, arrival_time, date_flight, flight_id, and airport_id. The main query editor contains the following SQL query:

```

1  Select s.flight_id,k.airline_name
2  from ((flight natural join flight_status)k join schedule s on s.flight_id=k.flight_id)
3  where k.actual_arr_time>s.arrival_time and s.date_flight='2023-04-20';

```

The 'Data Output' tab shows the results of the query:

flight_id	airline_name
486k457e578o	Akasa

At the bottom, it indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.098'.

5)Project the number of employees in airport with airport_id = 12

```

select count(staff_id)
from (staff natural join airport)
where airport_id=12;

```

The screenshot shows the pgAdmin 4 interface. On the left, the 'Columns (5)' folder is expanded under the 'staff' table, showing columns: departure_time, arrival_time, date_flight, flight_id, and airport_id. The main query editor contains the following SQL query:

```

1  select count(staff_id)
2  from (staff natural join airport)
3  where airport_id=12;

```

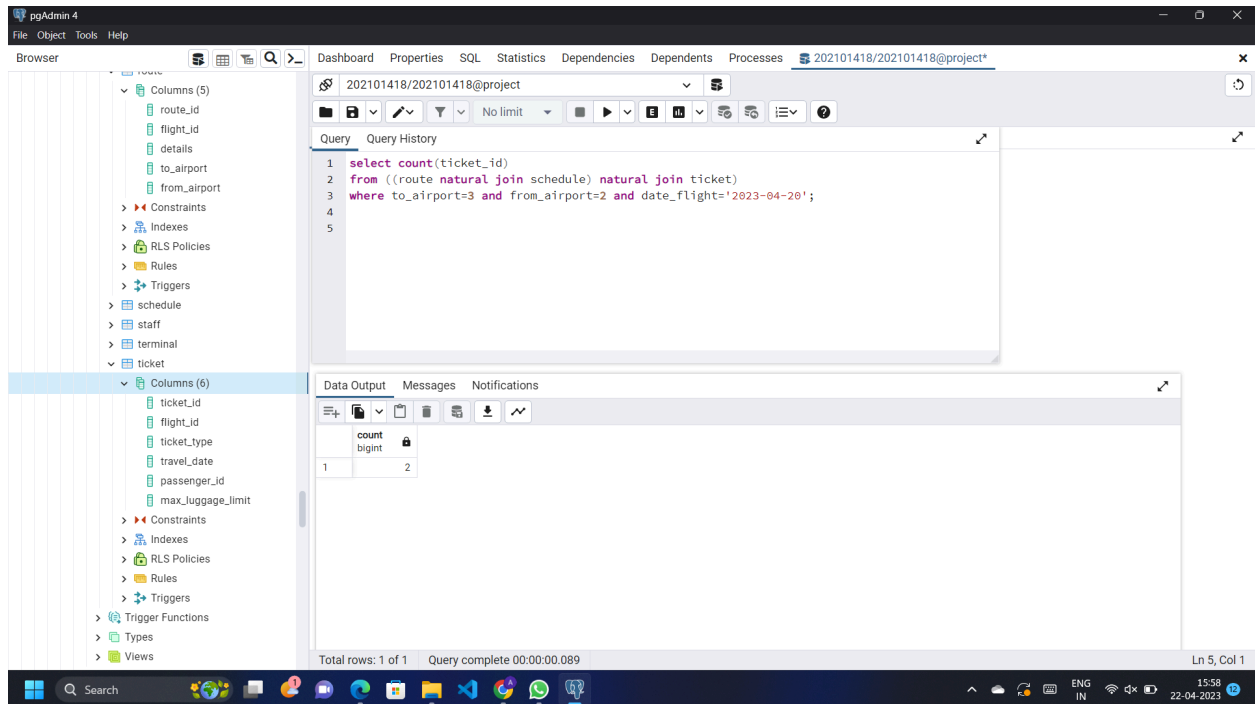
The 'Data Output' tab shows the results of the query:

count
19

At the bottom, it indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.093'.

6) Project the number of tickets for a route from airport id = 2 and airport id = 3 on 20th April 2023

```
select count(ticket_id)
from ((route natural join schedule) natural join ticket)
where to_airport=3 and from_airport=2 and date_flight='2023-04-20';
```



7)
List the people whose tickets are yet to be confirmed

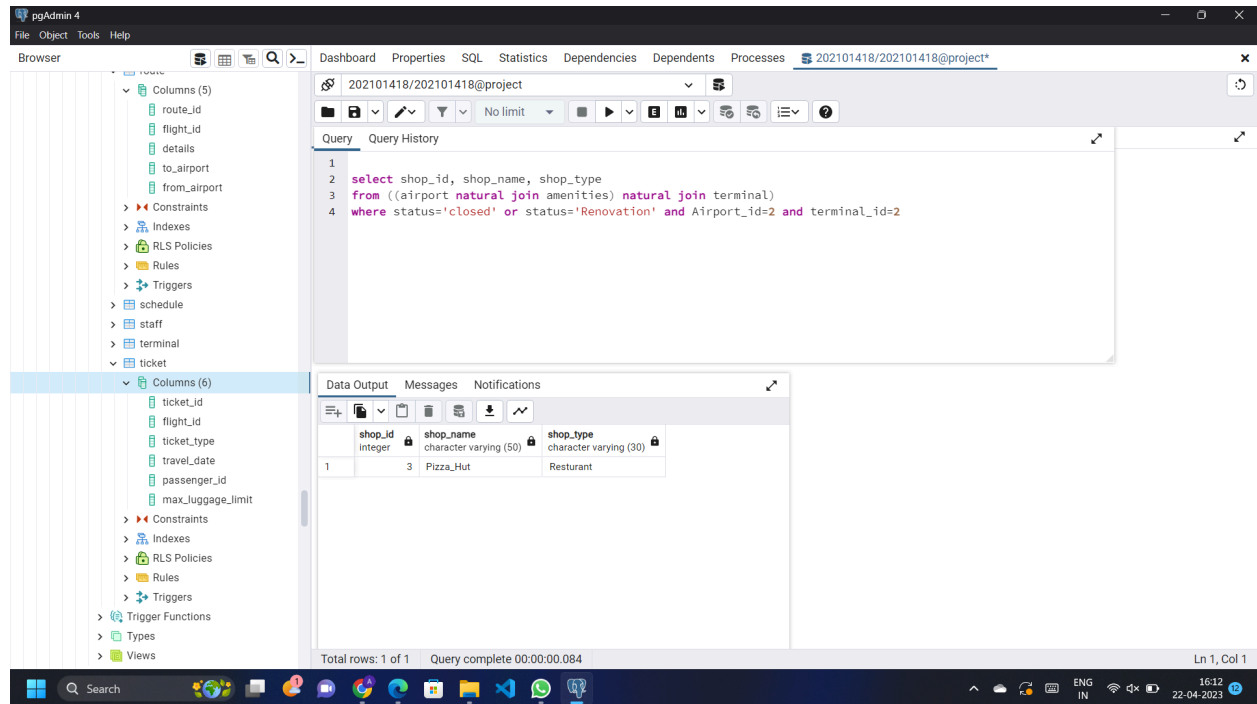
```
select distinct passenger_id, passenger_name, and passenger_contact from ticket as c natural
join booking_passenger natural join passenger where
c.status = 'Not-Confirmed';
```

The screenshot shows the pgAdmin 4 web interface. On the left, the 'Servers' tree is expanded to 'project'. The main pane displays a table with 7 rows and 3 columns. The columns are 'passenger_id' (integer), 'passenger_name' (character varying (20)), and 'passenger_contact' (bigint). The data is as follows:

passenger_id	passenger_name	passenger_contact
1	Alma	6841814182
2	Elka	6097218345
3	Haroun	5763615783
4	Dacia	9965965982
5	Maddalena	8081150019
6	Hedwiga	6119369626
7	Ingmar	5564276460



8) List the shop id, shop name and shop type which are closed or are under renovation in airport_id=2 and in the terminal id = 2.

```
select shop_id, shop_name, shop_type
from ((airport natural join amenities) natural join terminal)
where status='closed' or status='Renovation' and Airport_id=2 and terminal_id=2
```

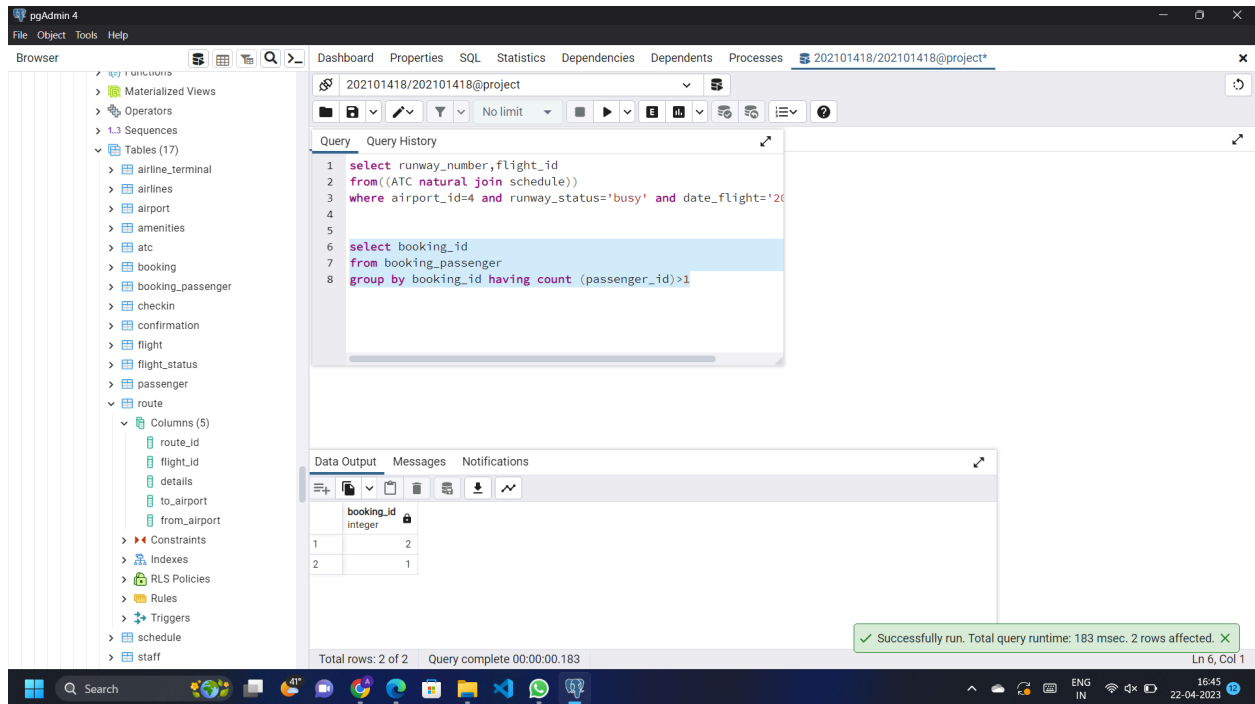


9) count airlines on particular airport

select count(airline_name),airport_id from airline_terminal group by airport_id;

	count bigint		airport_id integer	
1	6		8	
2	3		11	
3	4		19	
4	5		4	
5	4		14	
6	4		3	
7	3		17	
8	4		20	
9	3		13	
10	4		10	
11	5		7	
12	5		9	
13	2		1	
14	2		5	
15	3		18	
16	4		2	
17	3		15	
18	4		16	
19	2		6	
20	1		12	

10) List the booking id of tickets having same booking id
select booking_id
from booking_passenger
group by booking_id having count (passenger_id)>1

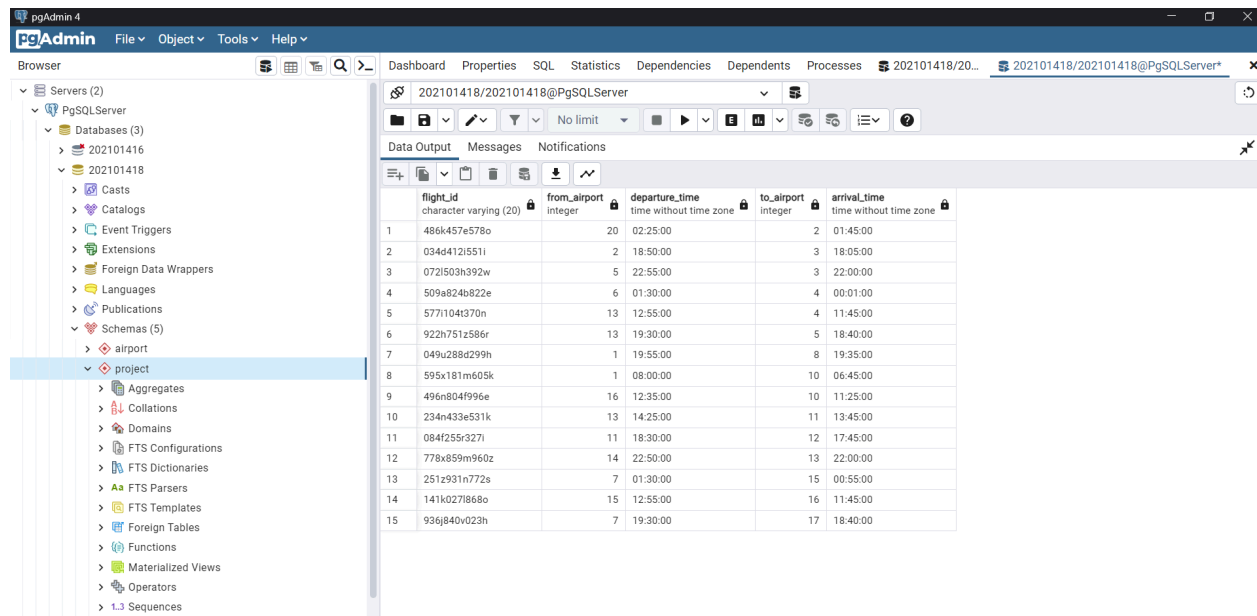


11) List flight details of flights arriving on a particular date at a particular airport
 select flight_id,airline_name,arrival_time,departure_time from(schedule natural join flight) where airport_id=2 and date_flight='2023-04-20';

	flight_id character varying (20)	airline_name character varying (20)	arrival_time time without time zone	departure_time time without time zone
1	072l503h392w	Green Inc	22:00:00	22:55:00
2	509a824b822e	Indigo	00:01:00	01:30:00

12) Give Schedule (Departure Time, Arrival Time, Source Airport, Destination Airport) Of A Flight (doubt - why only 15 flights and also how to write airport name instead of id)

```
select flight_id, r.from_airport, departure_time, r.to_airport, arrival_time
from (schedule natural join (airport natural join route as r));
```



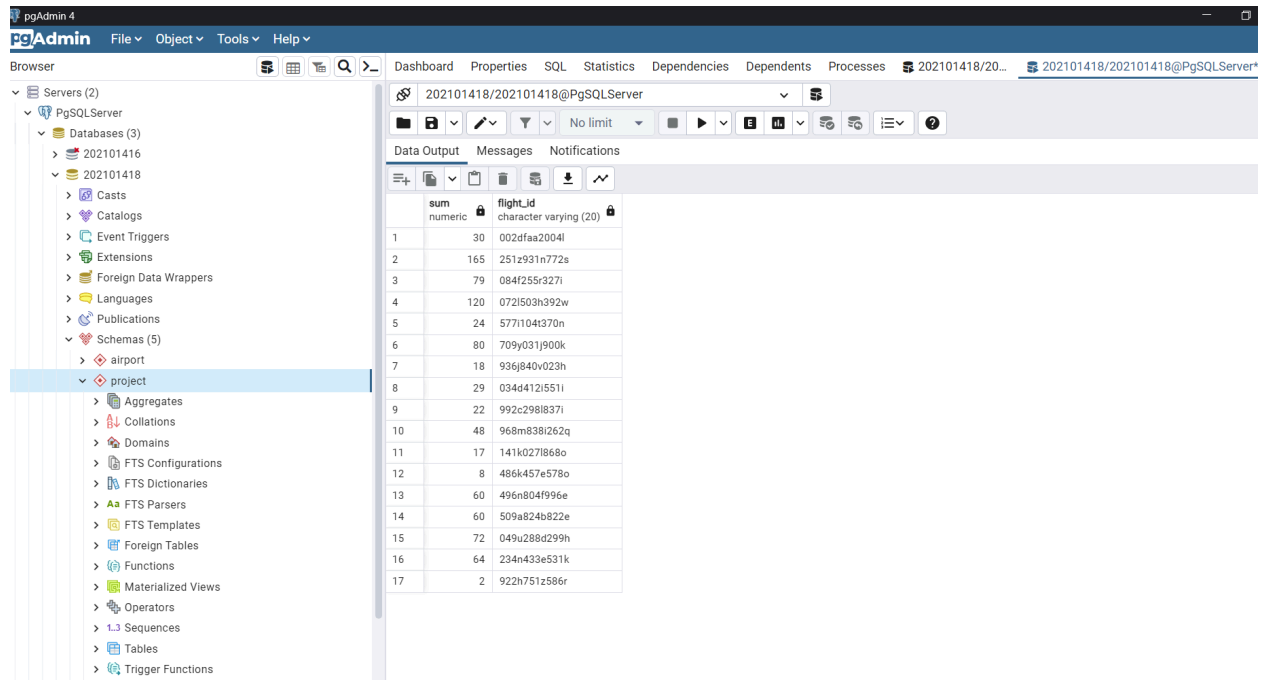
The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays a tree structure of the database. The 'project' schema is selected, showing various database objects. The main pane displays the results of a query, showing a table with 15 rows of flight data. The table has columns: flight_id, from_airport, departure_time, to_airport, and arrival_time. The data is as follows:

flight_id	from_airport	departure_time	to_airport	arrival_time
1	486k457e578o	20 02:25:00	2	01:45:00
2	034d412i551i	2 18:50:00	3	18:05:00
3	072i503h392w	5 22:55:00	3	22:00:00
4	509a824b822e	6 01:30:00	4	00:01:00
5	577i104i370n	13 12:55:00	4	11:45:00
6	922h751z586r	13 19:30:00	5	18:40:00
7	049u288d299h	1 19:55:00	8	19:35:00
8	595x181m605k	1 08:00:00	10	06:45:00
9	496n804f996e	16 12:35:00	10	11:25:00
10	234n433e531k	13 14:25:00	11	13:45:00
11	084f255r327i	11 18:30:00	12	17:45:00
12	778x859m960z	14 22:50:00	13	22:00:00
13	251z931n772s	7 01:30:00	15	00:55:00
14	141k027868o	15 12:55:00	16	11:45:00
15	936j840v023h	7 19:30:00	17	18:40:00

13)

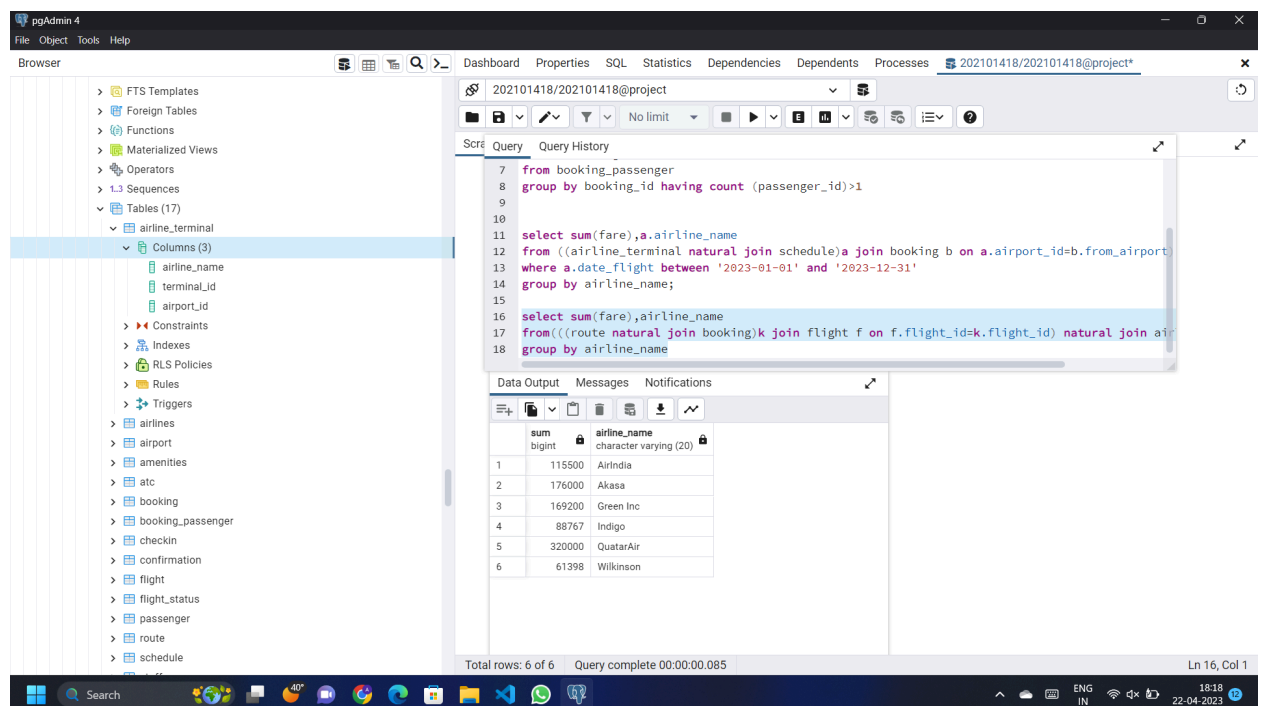
Total weight carried on a particular flight

```
select sum(weight), flight_id from checkin natural join ticket group by flight_id;
```



14) List the total revenue of each airline

```
select sum(fare),airline_name
from(((route natural join booking)p natural join route_flight k join flight f on f.flight_id=k.flight_id)
natural join airline_terminal)
group by airline_name
```



15) list passengers arriving on particular airport on a particular day

```
select airport_city,count(passenger_id) from (((route join airport on to_airport=airport_id)as k
natural join route_flight) natural join ticket)as p join schedule on p.flight_id=schedule.flight_id)
group by airport_city,date_flight having date_flight='2023-04-20';
```

	airport_city character varying (20) 🔒	count bigint 🔒
1	Ahmedabad	5
2	Delhi	1
3	Hyderabad	1
4	Kolkata	6
5	London	4
6	Paris	1

16) List flights having time interval greater than 6 hours

```
select * from flight_status as f1 natural join route join flight_status as f2
on f2.airport_id = route.to_airport and route.flight_id = f2.flight_id
where f2.actual_arr_time - f1.actual_dep_time > '6:00:00';
```

Query Editor

```

1 set search_path to project;
2
3 select * from flight_status as f1 natural join route join flight_status as f2
4 on f2.airport_id = route.to_airport and route.flight_id = f2.flight_id
5 where f2.actual_arr_time - f1.actual_dep_time > '6:00:00';

```

Data Output Messages Notifications

flight_id	actual_dep_time	actual_arr_time	date_flight	terminal_id	airport_id	route_id	details	to_airport
character varying (20)	time without time zone	time without time zone	date	integer	integer	integer	character varying (200)	integer

17) Find passenger who flew in business class from airport 6 to airport 4

```

select
booking_passenger.booking_id,booking_passenger.passenger_id,ticket.ticket_type,booking.fare
,ticket.travel_date
from booking_passenger natural join booking
join ticket on booking_passenger.ticket_id = ticket.ticket_id
where to_airport=4 and from_airport=6 and ticket_type='BUSINESS';

```

Dashboard Properties SQL Statistics Dependencies Dependents Processes 202101418/202101418@Yash

Query Editor

```

1 select booking_passenger.booking_id,booking_passenger.passenger_id,ticket.ticket_type,booking.fare,ticket.travel_date
2 from booking_passenger natural join booking
3 join ticket on booking_passenger.ticket_id = ticket.ticket_id
4 where to_airport=4 and from_airport=6 and ticket_type='BUSINESS';

```

Data Output Messages Notifications

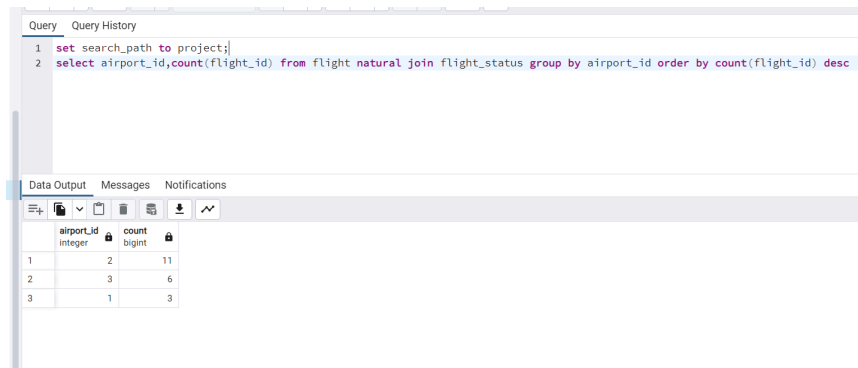
	booking_id	passenger_id	ticket_type	fare	travel_date
	integer	integer	character varying (20)	integer	date
1	13	41	BUSINESS	4500	2023-04-20
2	13	42	BUSINESS	4500	2023-04-20
3	13	43	BUSINESS	4500	2023-04-20
4	14	44	BUSINESS	2563	2023-04-20
5	14	45	BUSINESS	2563	2023-04-20
6	14	46	BUSINESS	2563	2023-04-20
7	15	47	BUSINESS	2800	2023-04-20
8	15	48	BUSINESS	2800	2023-04-20

18)Write the order of busiest airports based on number of passengers

```
select airport_id,count(passenger_id) from ticket natural join flight natural join flight_status group
by airport_id order by count(passenger_id) desc
```

19)Busiest Airport in terms of flights

```
select airport_id,count(flight_id) from flight natural join flight_status group by airport_id order by
count(flight_id) desc
```



The screenshot shows a database query interface. The top section is titled 'Query' and contains two lines of SQL code:
1 set search_path to project;
2 select airport_id,count(flight_id) from flight natural join flight_status group by airport_id order by count(flight_id) desc
The bottom section is titled 'Data Output' and displays the results of the query in a table. The table has two columns: 'airport_id' (integer) and 'count' (bigint). The results are as follows:

airport_id	count
1	11
2	6
3	3

20)Descending order of capacity - count going flight's route on 20th april

```
select * from (select
flight.flight_id,capacity,airline_name,actual_dep_time,actual_arr_time,date_flight,airport_id,count
(ticket_id),date_flight
from flight natural join flight_status left outer join ticket on flight.flight_id = ticket.flight_id
group by
flight.flight_id,capacity,airline_name,actual_dep_time,actual_arr_time,date_flight,airport_id
having date_flight='2023-04-20') as p join route on p.flight_id = route.flight_id order by
p.capacity - p.count desc;
```

Dashboard Properties SQL Statistics Dependencies Dependents Processes 202101418/202101418@Yash*

202101418/202101418@Yash

No limit

Query Query History

```

1 select * from (select flight.flight_id,capacity,airline_name,actual_dep_time,actual_arr_time,date_flight,airport_id,count(ticket_id),d
2 from flight natural join flight_status left outer join ticket on flight.flight_id = ticket.flight_id
3 group by flight.flight_id,capacity,airline_name,actual_dep_time,actual_arr_time,date_flight,airport_id
4 having date_flight='2023-04-20') as p join route on p.flight_id = route.flight_id order by p.capacity - p.count desc;
5

```

Data Output Messages Notifications

	flight_id character varying (20)	capacity integer	airline_name character varying (20)	actual_dep_time time without time zone	actual_arr_time time without time zone	date_flight date	airport_id integer	count bigint	date_flight date	route_id integer	flight char
1	072f503h392w	965	Green Inc	22:55:00	22:00:00	2023-04-20	2	0	2023-04-20	42	072
2	509a824b822e	905	Indigo	01:30:00	00:01:00	2023-04-20	2	3	2023-04-20	62	509
3	049u288d299h	824	AirIndia	19:55:00	19:35:00	2023-04-20	2	3	2023-04-20	134	049
4	486k457e578o	618	Akasa	14:25:00	13:45:00	2023-04-20	1	0	2023-04-20	38	486
5	992c298l837i	483	Akasa	04:45:00	03:55:00	2023-04-20	2	0	2023-04-20	350	992
6	002dfaa2004l	458	AirIndia	12:45:00	12:00:00	2023-04-20	1	4	2023-04-20	3	002
7	595x181m605k	398	QuatarAir	08:00:00	06:45:00	2023-04-20	2	0	2023-04-20	172	595
8	577f104t370n	399	Wilkinson	12:55:00	11:45:00	2023-04-20	2	3	2023-04-20	69	577
9	922h751z586r	279	Ali baba	19:30:00	18:40:00	2023-04-20	2	5	2023-04-20	88	922
10	034d412f551i	199	QuatarAir	18:50:00	18:05:00	2023-04-20	1	3	2023-04-20	40	034

Total rows: 10 of 10 Query complete 00:00:00.073 In 5 Col 1

21) People affected by delay in flight schedule -

Select distinct s.flight_id,p.passenger_name,p.passenger_id
 from (((flight natural join flight_status)k natural join schedule s) natural join ticket) natural
 join(select ticket_id from checkin where status = 'confirmed') natural join booking_passenger
 natural join passenger p
 where k.actual_arr_time>s.arrival_time and s.date_flight='2023-04-20';

pgAdmin 4

File Object Tools Help

Browser

Servers (2)

PgSQL Server

Databases (3)

202101416

202101418

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas (5)

airport

project

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Sequences

Tables

Trigger Functions

202101418/202101418@PgSQLServer

No limit

Data Output Messages Notifications

	flight_id	airline_name	passenger_name	passenger_id
	character varying (20)	character varying (20)	character varying (20)	integer
1	486k457e578o	Akasa	Elka	13
2	486k457e578o	Akasa	Valle	12
3	486k457e578o	Akasa	Pail	49
4	486k457e578o	Akasa	Dacia	24
5	486k457e578o	Akasa	Eugenia	29
6	486k457e578o	Akasa	Haroun	22
7	486k457e578o	Akasa	Harland	31
8	486k457e578o	Akasa	Hedwiga	34
9	486k457e578o	Akasa	Mariann	26
10	486k457e578o	Akasa	Chancey	17
11	486k457e578o	Akasa	Ingmar	35
12	486k457e578o	Akasa	Alma	11
13	486k457e578o	Akasa	Maddalena	27
14	486k457e578o	Akasa	Leola	23
15	486k457e578o	Akasa	Flora	30