DML – Data Manipulation Language

Update the airline country "SkyHigh" to "Spain Remove the airline named " KKG" from the airline table.

Add three airlines at once: "AirNorth" in "Canada", "AirAstana" in "Kazakhstan" and "EastFly" in "China"

Обновить страну авиакомпании "SkyHigh" на "Испанию"

Удалить авиакомпанию с названием "ККG" из таблицы авиакомпаний. Добавить сразу три авиакомпании: "AirNorth" в "Канаде", "AirAstana" в "Казахстане" и "EastFly" в "Китае".

```
UPDATE airline
SET airline_country = 'Spain'
WHERE airline_name = 'SkyHigh';
DELETE FROM airline WHERE airline_name = 'KKG';
INSERT INTO airline (airline_id, airline_code, airline_name, airline_country, created_at, update_at)
VALUES
    (52, 'AN', 'AirNorth', 'Canada', '2024-04-21', '2024-04-21'),
    (53, 'AA', 'AirAstana', 'Kazakhstan', '2024-04-21', '2024-04-21'),
    (54, 'EF', 'EastFly', 'China', '2024-04-21', '2024-04-21');
```

Delete all flights whose arrival in 2023 year

Удалить все рейсы, прибывающие в 2023 году.

```
DELETE FROM flights
WHERE actual_arrival >= '2023.01.01'
AND actual_arrival < '2024.01.01';</pre>
```

Add a new flight for SkyHigh to destination Sangzhou with three flights at different dates.

Добавить новый рейс для "SkyHigh" в пункт назначения "Sangzhou" с тремя вылетами в разные даты.

Update the departure date of all flights heading to Hilotongan, moving them forward one day

Перенести дату вылета всех рейсов, направляющихся в "Hilotongan", на один день вперед.

```
UPDATE flights
SET scheduled_departure = scheduled_departure + INTERVAL '1 DAY'
WHERE arrival_airport_id IN
(
    SELECT airport_id
    FROM airport
    WHERE airport_name = 'Hilotongan'
);
```

Check in new passenger "John Smith" for the flight to "Sangzhou"

Зарегистрировать нового пассажира "John Smith" на рейс в "Sangzhou".

```
INSERT INTO passengers (passenger_id, first_name, last_name, date_of_birth, gender, country_of_citizenship, cot
VALUES (201, 'John', 'Smith', '1990-01-01', 'Male', 'USA', 'USA', 'ABC123456', '2024-04-21', '2024-04-22');

INSERT INTO booking (booking_id, passenger_id, booking_platform, created_at, update_at, status, price)
VALUES (501, 201, 'Barrows Group', '2024-04-19', '2024-04-21', 'Male', 4569.00);

INSERT INTO booking_flight (booking_flight_id, booking_id, flight_id, created_at, update_at)
VALUES (1001, 501, 1001, '2024-04-20', '2024-04-21');
```

Increase the price of all tickets for flights to Wewit by 10% Увеличить цену всех билетов на рейсы в "Wewit" на 10%.

```
UPDATE booking
SET price = price * 1.1
WHERE booking_id IN (
    SELECT booking_id
    FROM booking_flight
    WHERE flight_id IN (
        SELECT flight_id
        FROM flights
        WHERE arrival_airport_id = (
            SELECT airport_id
            FROM airport
            WHERE city = 'Wewit'
        )
    )
)
);
```

Delete all tickets whose price is less than 1000 units

Удалить все билеты, цена которых ниже 1000 единиц.

```
DELETE FROM booking
WHERE price < 1000;</pre>
```

JOIN operations

Write a query that displays all flights of a specific airline

Написать запрос, который отображает все рейсы конкретной авиакомпании.

```
SELECT flights.*
FROM flights
JOIN airline
ON flights.airline_id = airline.airline_id
WHERE airline.airline_name = 'IPC';
```

Compose a query to obtain a list of all flights with the names of departure airports

Составить запрос для получения списка всех рейсов с названиями аэропортов отправления.

```
SELECT flights.*, airport.airport_name AS departure_airport_name
FROM flights
JOIN airport ON flights.departure_airport_id = airport.airport_id
```

Create a query that finds all airlines that have no flights scheduled for the next month

Создать запрос для поиска всех авиакомпаний, у которых нет запланированных рейсов на следующий месяц.

```
SELECT airline.*
FROM airline
LEFT JOIN flights ON airline.airline_id = flights.airline_id
WHERE EXTRACT(MONTH FROM flights.scheduled_departure) != EXTRACT(MONTH FROM CURRENT_DATE) +
1 OR flights.flight_id IS NULL;
```

Create a query to display a list of passengers on a specific flight.

Создать запрос для отображения списка пассажиров на конкретном рейсе.

```
SELECT passengers.*
FROM passengers
JOIN booking ON passengers.passenger_id = booking.passenger_id
JOIN booking_flight ON booking.booking_id = booking_flight.booking_id
JOIN flights ON booking_flight.flight_id = flights.flight_id
WHERE flights.flight_id = 224;
```

Write a query that calculates the average, total, maximum and minimum price of tickets for each flight.

Написать запрос, который вычисляет среднюю, общую, максимальную и минимальную стоимость билетов для каждого рейса.

```
SELECT
    f.flight_id,
    AVG(b.price) AS average_price,
    SUM(b.price) AS total_price,
    MAX(b.price) AS max_price,
    MIN(b.price) AS min_price
FROM
    flights f

JOIN
    booking_flight bf ON f.flight_id = bf.flight_id

JOIN
    booking b ON bf.booking_id = b.booking_id

GROUP BY
    f.flight_id;
```

Create a query that shows all flights flying to a specific country by combining flights, airports and airline, and using the condition on the country name.

Создать запрос, который показывает все рейсы в определенную страну, объединяя данные о рейсах, аэропортах и авиакомпаниях, используя условие по названию страны.

Display a list of minor passengers and their destination.

Отобразить список несовершеннолетних пассажиров и их пункта назначения.

```
SELECT p.first_name, p.last_name, p.date_of_birth, p.country_of_citizenship, f.scheduled_arrival
FROM passengers p
JOIN booking b ON p.passenger_id = b.passenger_id
JOIN booking_flight bf ON b.booking_id = bf.booking_id
JOIN flights f ON bf.flight_id = f.flight_id
WHERE EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM p.date_of_birth) < 18;</pre>
```

The phone was found at the "RAS^B" location, display the passenger's full name, passport number, and the passenger's current time of arrival at the destination.

Телефон был найден в местоположении "RAS^B". Отобразить полное имя пассажира, номер паспорта и текущее время прибытия пассажира в пункт назначения.

```
SELECT passengers.first_name, passengers.last_name, passengers.passport_number, flights.actual_arrival
FROM passengers
JOIN booking ON passengers.passenger_id = booking.passenger_id
JOIN booking_flight ON booking.booking_id = booking_flight.booking_id
JOIN flights ON booking_flight.flight_id = flights.flight_id
JOIN airport ON flights.arrival_airport_id = airport.airport_id
WHERE airport.city = 'RAS^B';
```

Print a list of flights where the airline's home country and origin country are the same. Sort them by the airport country

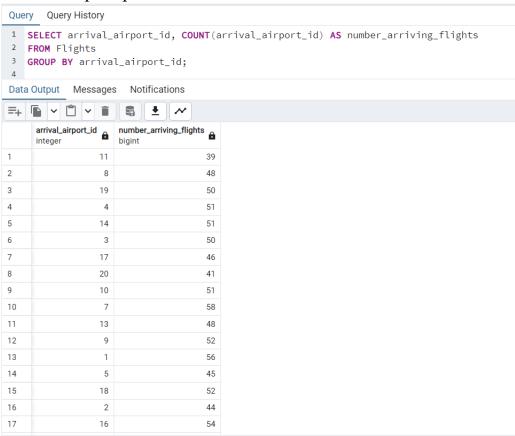
Вывести список рейсов, где страна авиакомпании и страна отправления совпадают. Отсортировать по стране аэропорта.

```
SELECT flights.*, airport.country AS departure_country
FROM flights
JOIN airport ON flights.departure_airport_id = airport.airport_id
JOIN airline ON flights.airline_id = airline.airline_id
WHERE airline.airline_country = airport.country
ORDER BY airport.country;
```

Grouping Data, Subquery

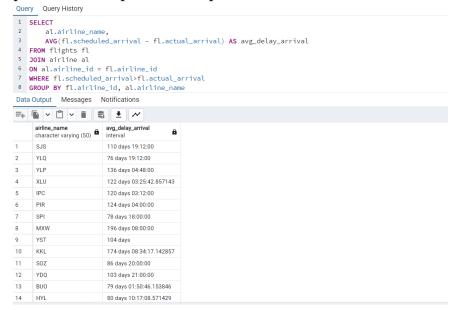
1) Write a query to find the number of flights arriving at each airport.

Написать запрос для определения количества рейсов, прибывающих в каждый аэропорт.



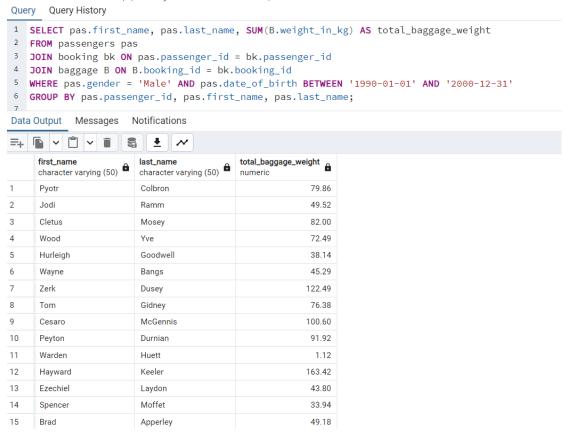
2) What is the average flight delay time for each airline in the arrived schedule?

Определить среднее время задержки рейсов для каждой авиакомпании в расписании прибывших рейсов.



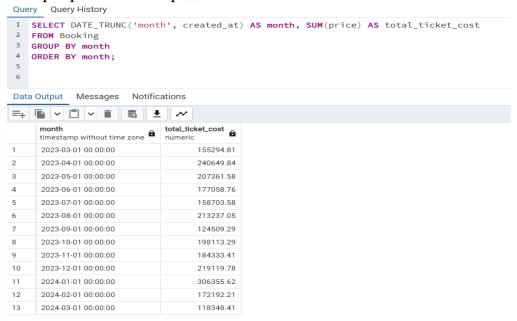
3) Fine the first name and last name of each male passenger born between 1990 and 2000, along with the total baggage weight for each passenger.

Найти имя и фамилию каждого пассажира-мужчины, родившегося между 1990 и 2000 годами, а также общий вес его багажа.



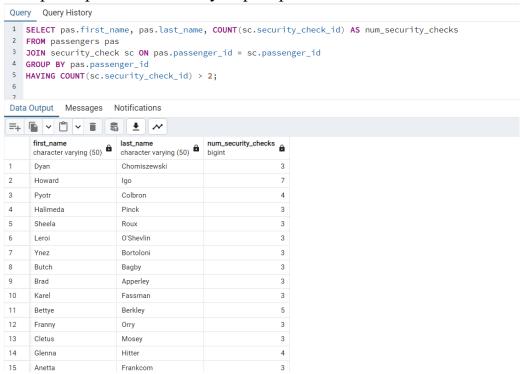
4) What is the total cost of tickets sold for each month in sorted way?

Определить общую стоимость проданных билетов за каждый месяц в отсортированном порядке.



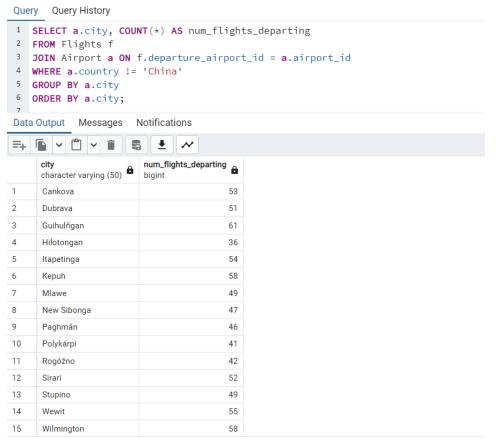
5) Find the first name, last name, and number of security checks for passengers who have undergone more than two security checks.

Найти имя, фамилию и количество проверок безопасности для пассажиров, которые прошли более двух проверок.



6) What is the number of flights departing from each city excluding Chinese cities in the alphabetic order?

Определить количество рейсов, отправляющихся из каждого города, за исключением китайских городов, в алфавитном порядке



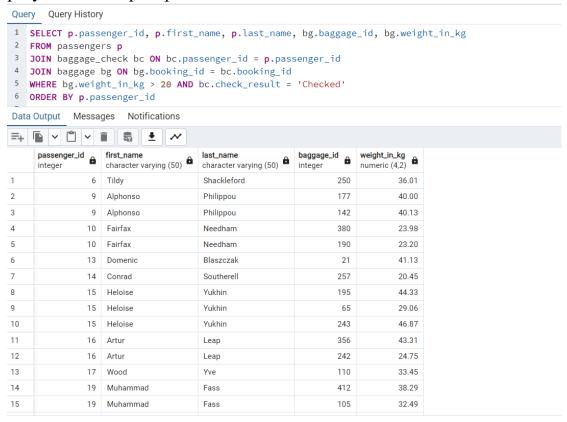
7) How many passengers have booked flights departing from airports located in Poland?

Определить количество пассажиров, забронировавших рейсы, отправляющиеся из аэропортов, расположенных в Польше

```
Query History
Query
   SELECT COUNT(DISTINCT bk.passenger_id) as passenger_count
2
   FROM airport a
   JOIN flights f ON f.departure_airport_id = a.airport_id
   JOIN booking_flight bf ON bf.flight_id = f.flight_id
5
   JOIN booking bk ON bk.booking_id = bf.booking_id
6
   WHERE a.country = 'Poland'
7
                       Notifications
Data Output
            Messages
=+
     passenger_count
     bigint
1
                 35
```

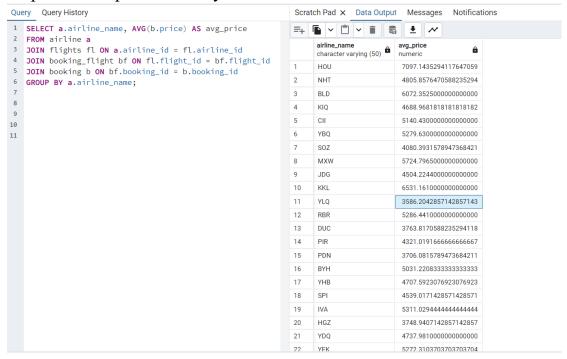
8) Find all overweight baggage (greater than 20 kg) for a passenger, along with the particular security check results "Checked".

Найти весь перевес багажа (свыше 20 кг) для каждого пассажира вместе с результатами проверки безопасности "Checked".



9) What is the average ticket price for each airline?

Определить среднюю цену билетов для каждой авиакомпании.



10) Show the amount of checked baggage, total baggage and what percentage of baggage has passed security checks for each flight.

Вывести количество зарегистрированного багажа, общий вес багажа и процент багажа, прошедшего проверку безопасности, для каждого рейса.

```
SELECT F.ftight_no,
SUM(CASE WHEN BC.check_result = 'Checked' THEN B.weight_in_kg ELSE 0 END) AS checked_baggage_weight,
SUM(B.weight_in_kg) AS total_baggage_weight,
(SUM(CASE WHEN BC.check_result = 'Checked' THEN 1 ELSE 0 END):: numeric / COUNT(*):: numeric) * 100 :: numeric
AS percentage_passed_security
FROM Flights F
JOIN Booking_flight BF ON F.flight_id = BF.flight_id
JOIN Baggage B ON BF.booking_id = B.booking_id
LEFT JOIN Baggage_check BC ON B.booking_id = BC.booking_id
GROUP BY F.flight_no;
```

| | flight_no character varying (50) | checked_baggage_weight numeric | total_baggage_weight numeric | percentage_passed_security numeric |
|----|----------------------------------|--------------------------------|------------------------------|---|
| 1 | FR-O | 0 | 4.47 | 0.0000000000000000000000000000000000000 |
| 2 | ZA-NL | 53.15 | 53.15 | 100.00000000000000000000000000000000000 |
| 3 | US-NV | 0 | 57.04 | 0.0000000000000000000000000000000000000 |
| 4 | CA-AB | 0 | 55.10 | 0.0000000000000000000000000000000000000 |
| 5 | BR-PA | 0 | 50.85 | 0.0000000000000000000000000000000000000 |
| 6 | UA-32 | 60.81 | 72.93 | 66.666666666666666700 |
| 7 | RU-SAK | 130.26 | 214.29 | 62.500000000000000000000 |
| 8 | AU-WA | 104.75 | 375.52 | 23.07692307692307692300 |
| 9 | UA-46 | 0 | 131.06 | 0.0000000000000000000000000000000000000 |
| 10 | AZ-BA | 0 | 16.12 | 0.0000000000000000000000000000000000000 |
| 11 | IN-WB | 47.27 | 47.27 | 100.00000000000000000000000000000000000 |
| 12 | DZ-33 | 0 | 36.28 | 0.0000000000000000000000000000000000000 |
| 13 | DE-SH | 0 | 9.17 | 0.0000000000000000000000000000000000000 |
| 14 | GB-NIR | 0 | 79.86 | 0.0000000000000000000000000000000000000 |
| 15 | AU-TAS | 0 | 24.69 | 0.0000000000000000000000000000000000000 |
| 16 | US-MT | 0 | 133.57 | 0.0000000000000000000000000000000000000 |
| 17 | DE-NW | 36.73 | 87.10 | 25.000000000000000000000 |
| 18 | VC-U-A | 0 | 78.90 | 0.0000000000000000000000000000000000000 |
| 19 | DK-84 | 49.06 | 111.76 | 33.333333333333333333 |
| 20 | NP-SA | 0 | 57.17 | 0.0000000000000000000000000000000000000 |
| 21 | MZ-P | 0 | 109.55 | 0.0000000000000000000000000000000000000 |
| 22 | IIS-WA | n | กว จก | 0 0000000000000000000000000000000000000 |

11) Write query Identify the top 5 busiest airports based on the total number of bookings.

Определить 5 самых загруженных аэропортов на основе общего количества бронирований.

| | airport_name character varying (50) | num_bookings bigint |
|---|-------------------------------------|------------------------|
| 1 | Industrial Airpark | 80 |
| 2 | Elorza Airport | 71 |
| 3 | Hana Airport | 59 |
| 4 | Figari Sud-Corse Airport | 59 |
| 5 | Henri Coandă International Airport | 57 |

```
SELECT A.airport_name, COUNT(*) AS num_bookings
FROM Booking_flight BF

JOIN Flights F ON BF.flight_id = F.flight_id

JOIN Airport A ON F.departure_airport_id = A.airport_id

GROUP BY A.airport_name

ORDER BY num_bookings DESC

LIMIT 5;
```

12) Find the average weight of checked baggage for each flight. Group the results by the flight number

Найти средний вес зарегистрированного багажа для каждого рейса. Группировать результаты по номеру рейса.

```
SELECT f.flight_no, AVG(b.weight_in_kg) AS avg_weight_checked_baggage FROM Flights f

JOIN Booking_flight bf ON f.flight_id = bf.flight_id

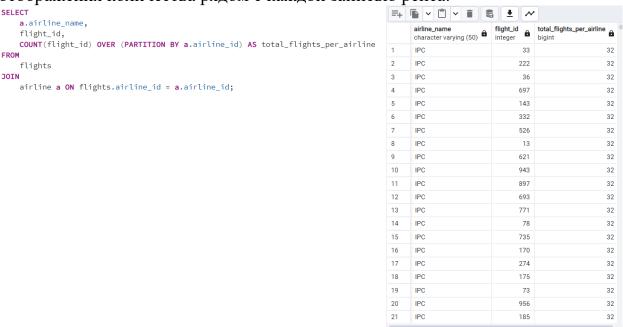
JOIN Baggage b ON bf.booking_id = b.booking_id

GROUP BY f.flight_no;
```

| | flight_no character varying (50) | avg_weight_checked_baggage numeric |
|----|----------------------------------|------------------------------------|
| 1 | FR-O | 4.4700000000000000 |
| 2 | ZA-NL | 26.5750000000000000 |
| 3 | US-NV | 28.5200000000000000 |
| 4 | CA-AB | 11.02000000000000000 |
| 5 | BR-PA | 25.4250000000000000 |
| 6 | UA-32 | 24.3100000000000000 |
| 7 | RU-SAK | 25.1250000000000000 |
| 8 | AU-WA | 28.8861538461538462 |
| 9 | UA-46 | 42.6700000000000000 |
| 10 | AZ-BA | 8.0600000000000000 |
| 11 | IN-WB | 23.6350000000000000 |
| 12 | DZ-33 | 36.2800000000000000 |
| 13 | DE-SH | 9.1700000000000000 |
| 14 | GB-NIR | 39.9300000000000000 |
| 15 | AU-TAS | 24.6900000000000000 |
| 16 | US-MT | 33.3925000000000000 |
| 17 | DE-NW | 16.7900000000000000 |
| 18 | VC-U-A | 26.3000000000000000 |
| 19 | DK-84 | 15.6750000000000000 |
| 20 | NP-SA | 19.056666666666667 |
| 21 | MZ-P | 27.3875000000000000 |
| 22 | LIC WA | 27 0022222222222 |

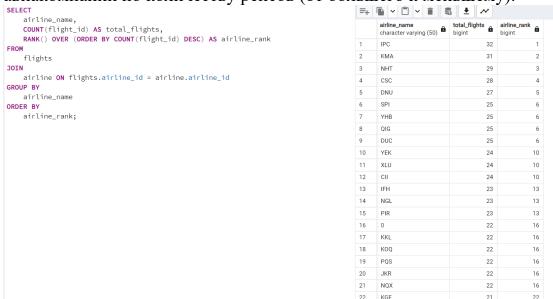
Aggregate Functions. Window Functions

1. Write a SQL query to list the total number of flights per airline. Use the COUNT() window function to show the count next to each flight entry. Написать SQL-запрос для вывода общего количества рейсов на каждую авиакомпанию, используя оконную функцию COUNT() для отображения количества рядом с каждой записью рейса.



2. Use the RANK() window function to rank airlines based on the number of flights they have from highest to lowest.

Использовать оконную функцию RANK() для ранжирования авиакомпаний по количеству рейсов (от большего к меньшему).



3. Employ the DENSE_RANK() window function to rank the airports based on the number of incoming flights without gaps in rank values.

Использовать оконную функцию DENSE_RANK() для ранжирования аэропортов по количеству прибывающих рейсов без пропусков в

рангах

```
SELECT

airport_name,

COUNT(f.arrival_airport_id) AS incoming_flights,

DENSE_RANK() OVER (

ORDER BY COUNT(f.arrival_airport_id) DESC
)

AS airport_rank

FROM

flights f

JOIN

airport a ON f.arrival_airport_id = a.airport_id

GROUP BY

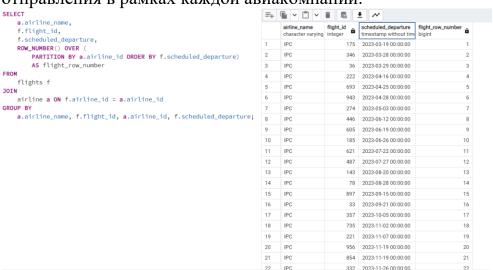
airport_name

ORDER BY

airport_rank;
```



4. Utilize the ROW_NUMBER() function to assign a unique row number to each flight sorted by departure time within each airline.
Применить функцию ROW_NUMBER() для присвоения уникального номера каждой строке рейсов, отсортированных по времени отправления в рамках каждой авиакомпании.



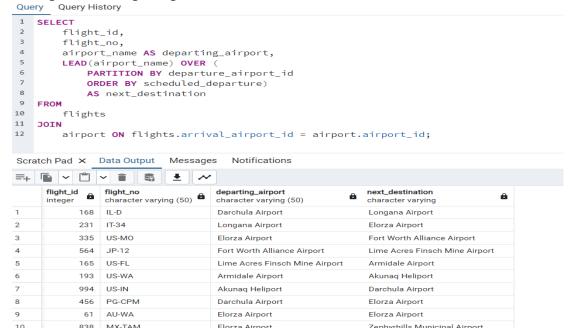
5. Write a query using the LAG() window function to find the time difference in minutes between consecutive flights for each airline.

Использовать функцию LAG() для нахождения разницы во времени (в минутах) между последовательными рейсами каждой авиакомпании.

```
SELECT
    airline id.
    flight_id,
    flight no.
    scheduled departure.
    LAG(scheduled_arrival) OVER (PARTITION BY airline_id ORDER BY scheduled_departure) AS previous_arrival,
    (scheduled_departure - LAG(scheduled_arrival) OVER (
         PARTITION BY airline_id
         ORDER BY scheduled_departure)) / 60
    AS time_diff_minutes
FROM
    flights
    airline_id, scheduled_departure;
tch Pad x Data Output Messages Notifications
...griCld flight_no character
 airline_id
                          flight_no character varying (50) scheduled_departure timestamp without time zone previous_arrival timestamp without PA-8 2023-03-19 00:00:00 [null]
                                                                                                     time_diff_minutes
                                                                          timestamp without time zone
 integer
                                                                                                     interval
                     175 PA-8
                     346 US-MS
                                               2023-03-28 00:00:00
                                                                          2023-08-21 00:00:00
                                                                                                     -2 days -10:24:00
                                              2023-03-29 00:00:00
                                                                         2023-10-25 00:00:00
                                                                                                    -3 days -12:00:00
                                   2023-04-16 00:00:00
                                                                 2023-05-06 00:00:00
                                                                                                    -08:00:00
                     222 RU-CU
                                       2023-04-25 00:00:00
                                                                 2023-09-15 00:00:00
                                                                                                   -2 days -09:12:00
                     693 US-LA
                                               2023-04-28 00:00:00
2023-05-03 00:00:00
                     943 CA-BC
                                                                          2023-08-04 00:00:00
                                                                                                     -1 days -15:12:00
                     274 BE-VAN
                                                                          2023-06-28 00:00:00
                                                                                                     -22:24:00
                                               2023-06-12 00:00:00
                                                                                                     04:24:00
                     446 KH-9
                                                                          2023-06-01 00:00:00
                                               2023-06-19 00:00:00
                                                                          2024-01-26 00:00:00
                                                                                                     -3 days -16:24:00
```

6. Use the LEAD() window function to predict the next destination airport for each flight departing from a specific airport.

Использовать функцию LEAD() для предсказания следующего аэропорта назначения для каждого рейса, отправляющегося из конкретного аэропорта.



7. Apply the FIRST_VALUE() window function to identify the first flight of the day for each airline.

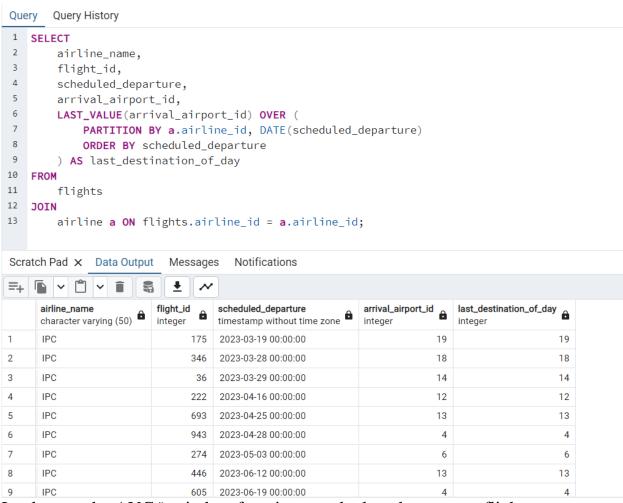
Применить функцию FIRST_VALUE() для определения первого рейса дня для каждой авиакомпании.

| SELECT | | | | | |
|---|----|------------------------|-----------|-----------------------------|--|
| airline_name, | | airline name | flight_id | scheduled departure | |
| flight_id, | | character varying (50) | integer | timestamp without time zone | |
| scheduled_departure ROM (| | IPC | 175 | 2023-03-19 00:00:00 | |
| SELECT | 2 | IPC | 346 | 2023-03-28 00:00:00 | |
| airline_name, | 3 | IPC | 36 | 2023-03-29 00:00:00 | |
| flight_id, | 4 | IPC | 222 | 2023-04-16 00:00:00 | |
| scheduled_departure, | 5 | IPC | 693 | 2023-04-25 00:00:00 | |
| <pre>FIRST_VALUE(scheduled_departure) OVER (PARTITION BY a.airline_id, DATE(scheduled_departure)</pre> | 6 | IPC | 943 | 2023-04-28 00:00:00 | |
| ORDER BY scheduled departure) | 7 | IPC | 274 | 2023-05-03 00:00:00 | |
| AS first_departure_of_day | 8 | IPC | 446 | 2023-06-12 00:00:00 | |
| FROM | 9 | IPC | 605 | 2023-06-19 00:00:00 | |
| flights | | | | | |
| JOIN | 10 | IPC | 185 | 2023-06-26 00:00:00 | |
| airline a ON flights.airline_id = a.airline_id | 11 | IPC | 621 | 2023-07-22 00:00:00 | |
| AS first_flights | | IPC | 487 | 2023-07-27 00:00:00 | |
| <pre>wHERE scheduled_departure = first_departure_of_day;</pre> | 13 | IPC | 143 | 2023-08-20 00:00:00 | |
| scheduced_departare = Trist_departare_or_day; | 14 | IPC | 78 | 2023-08-28 00:00:00 | |
| | 15 | IPC | 897 | 2023-09-15 00:00:00 | |
| | 16 | IPC | 33 | 2023-09-21 00:00:00 | |
| | | IPC | 357 | 2023-10-05 00:00:00 | |
| | 18 | IPC | 735 | 2023-11-02 00:00:00 | |
| | 19 | IPC | 221 | 2023-11-07 00:00:00 | |
| | 20 | IPC | 854 | 2023-11-19 00:00:00 | |
| | 21 | IPC | 956 | 2023-11-19 00:00:00 | |
| | 22 | IPC | 332 | 2023-11-26 00:00:00 | |

8. Use the LAST_VALUE() window function (with the appropriate frame clause) to determine the last destination of the day for each airline. Использовать функцию LAST_VALUE() (с соответствующим определением окна) для определения последнего пункта назначения дня для каждой авиакомпании.

 □
 ∨
 □
 √
 □
 √
 □
 ✓
 □
 ✓
 □
 ✓
 □
 ✓
 □
 ✓
 □
 ✓
 □
 ✓
 □
 ✓
 □
 ✓
 □
 ✓
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○
 ○</ _id airline_name a.airline_name, f.scheduled_departure::date AS flight_date, 1 IPC 2023-03-19 19 Longana Airport LAST_VALUE(f.arrival_airport_id) OVER (PARTITION BY f.airline_id, f.scheduled_departure::date 2 2023-03-28 18 Darchula Airport ORDER BY f.scheduled_departure 1 IPC 2023-03-29 14 Industrial Airpark) AS last_destination_airport_id, 1 IPC 2023-04-16 12 Elorza Airport ap.airport_name AS last_destination_airport_name 2023-04-25 13 Figari Sud-Corse Airport 2023-04-28 flights f 1 IPC 4 Garbaharey Airport 2023-05-03 6 Hana Airport 2023-06-12 airline a ON f.airline_id = a.airline_id 1 IPC 13 Figari Sud-Corse Airport 2023-06-19 4 Garbaharev Airport airport ap ON f.arrival_airport_id = ap.airport_id 5 Delta County Airport ORDER BY 2023-07-22 2023-07-27 6 Hana Airport
5 Delta County Airport f.airline_id, flight_date; 1 IPC 1 IPC 2023-08-20 9 Pitalito Airport 2023-08-28 14 11 Bermuda Dunes Airport 12 Elorza Airport 1 IPC 2023-09-21 19 Longana Airport 17 2023-10-05 6 Hana Airport 18 2023-11-02 19 Longana Airport 2023-11-07 19 1 IPC 18 Darchula Airport 5 Delta County Airport 2023-11-19

9. Create a report using SUM() window function to calculate the cumulative number of passengers for flights throughout a day for each airline. Создать отчет с помощью функции SUM(), вычисляющий накопительное количество пассажиров для рейсов в течение дня для каждой авиакомпании.



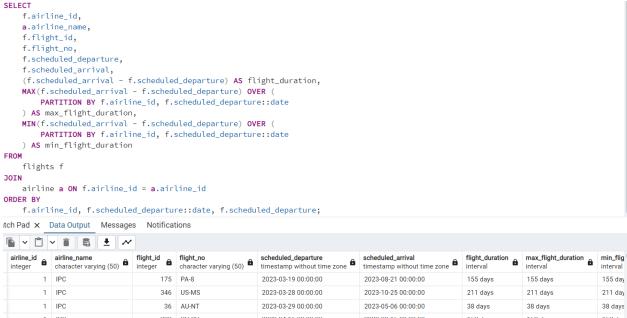
10.Implement the AVG() window function to calculate the average flight duration for each airline, displayed next to each flight.

Применить функцию AVG() для расчета средней продолжительности полета каждой авиакомпании, отображая её рядом с каждым рейсом.

```
SELECT
     f.airline_id,
     a.airline_name
     f.flight_id,
     f.flight_no,
     f.scheduled departure.
     f.scheduled arrival.
     (f.scheduled_arrival - f.scheduled_departure) AS flight_duration,
     AVG(f.scheduled_arrival - f.scheduled_departure) OVER (
         PARTITION BY f.airline_id)
         AS avg_flight_duration
FROM
     flights f
JOIN
    airline a ON f.airline_id = a.airline_id
ORDER BY
     f.airline id, f.scheduled departure:
itch Pad X Data Output Messages Notifications
 flight_no
character varying (50) scheduled_departure
timestamp without time zone scheduled_arrival
timestamp without time zone
                                                                                                                       flight_duration avg_flight_duration
             character varying (50)
                                                                                                                       interval
                                                                                                                                       interval
                                                                                                                        155 days
          1 IPC
                                        175 PA-8
                                                                  2023-03-19 00:00:00
                                                                                             2023-08-21 00:00:00
                                                                                                                                       2 days 23:15:00
                                        346 US-MS
                                                                  2023-03-28 00:00:00
                                                                                             2023-10-25 00:00:00
                                                                                                                        211 days
                                                                                                                                       2 days 23:15:00
                                                                                                                                       2 days 23:15:00
          1 IPC
                                        222 RU-CH
                                                                  2023-04-16 00:00:00
                                                                                             2023-09-15 00:00:00
                                                                                                                        152 days
                                                                                                                                       2 days 23:15:00
          1 IPC
                                        693 US-LA
                                                                  2023-04-25 00:00:00
                                                                                             2023-08-04 00:00:00
                                                                                                                        101 days
                                                                                                                                       2 days 23:15:00
          1 IPC
                                        943 CA-BC
                                                                  2023-04-28 00:00:00
                                                                                             2023-06-28 00:00:00
                                                                                                                        61 days
                                                                                                                                       2 days 23:15:00
```

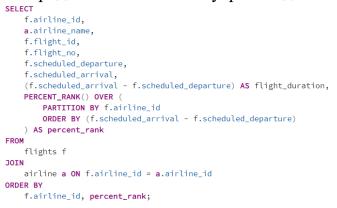
11.Utilize window functions to display each flight along with the maximum and minimum flight durations for that airline on the same day.

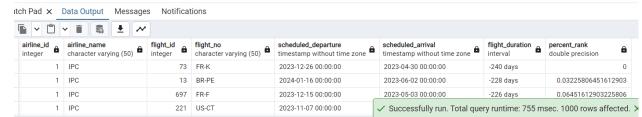
Использовать оконные функции для отображения каждого рейса вместе с максимальной и минимальной продолжительностью рейсов для данной авиакомпании за тот же день.



12. Using the PERCENT_RANK() function, rank the flights based on their duration within each airline.

Использовать функцию PERCENT_RANK() для ранжирования рейсов по продолжительности внутри каждой авиакомпании.





13. Write a SQL query that uses the NTILE(4) function to divide all flights into four quartiles based on their duration for each airline.

Написать SQL-запрос, использующий функцию NTILE(4) для разделения всех рейсов на четыре квартиля по продолжительности для каждой авиакомпании.

```
f.flight_id,
     f.flight_no,
     f.scheduled departure.
     f.scheduled_arrival.
     (f.scheduled_arrival - f.scheduled_departure) AS flight_duration,
     NTILE(4) OVER (
          PARTITION BY f.airline_id
          ORDER BY (f.scheduled_arrival - f.scheduled_departure)
     ) AS quartile
FROM
     flights f
JOIN
    airline a ON f.airline_id = a.airline_id
     f.airline_id, quartile;
itch Pad X Data Output Messages Notifications
 airline_id airline_name
             airline_name character varying (50) a flight_ld integer b flight_no character varying (50) a scheduled_departure timestamp without time zone a scheduled_arrival timestamp without time zone a
                                                                                                                          flight_duration quartile
                                                                    2023-12-26 00:00:00
                                                                                               2023-04-30 00:00:00
                                                                                                                           -240 days
          1 IPC
                                          13 BR-PE
                                                                    2024-01-16 00:00:00
                                                                                               2023-06-02 00:00:00
                                                                                                                           -228 days
          1 IPC
                                         697 FR-F
                                                                    2023-12-15 00:00:00
                                                                                                2023-05-03 00:00:00
                                                                                                                           -226 days
```

✓ Successfully run. Total query runtime: 737 msec. 1000 rows affected. X

14.Employ a combination of COUNT() to list each flight and the total number of flights by the same aircraft type.

2023-11-07 00:00:00

221 US-CT

SELECT

f.airline_id,
a.airline_name

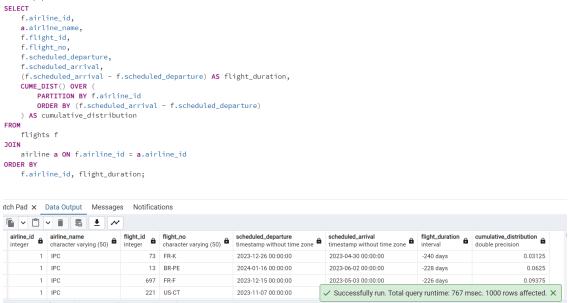
1 IPC

Использовать комбинацию функции COUNT() для вывода списка каждого рейса и общего количества рейсов для одного и того же типа самолета.

```
SELECT
         f.airline_id,
         a.airline_name,
         d.airport_name AS destination,
         COUNT(*) AS num_flights,
         RANK() OVER (PARTITION BY f.airline_id ORDER BY COUNT(*) DESC) AS desti
    FROM
         flights f
    INNER JOIN
         airport d ON f.arrival_airport_id = d.airport_id
         airline a ON f.airline_id = a.airline_id
    GROUP BY
         f.airline_id,
         a.airline_name,
         d.airport_name
) AS FlightCounts
VHERE
    destination_rank = 1;
ch Pad X
           Data Output
                        Messages
                                    Notifications
    ~
 airline_id
                                                              num_flights
            airline_name
                                 destination
 integer
            character varying (50)
                                 character varying (50)
                                                              bigint
            IPC
                                                                        5
         1
                                  Delta County Airport
            PDN
                                                                        3
                                 Hana Airport
         2
            PDN
                                 Fort Worth Alliance Airport
                                                                        3
         3
            KLF
                                  Armidale Airport
                                                                        2
```

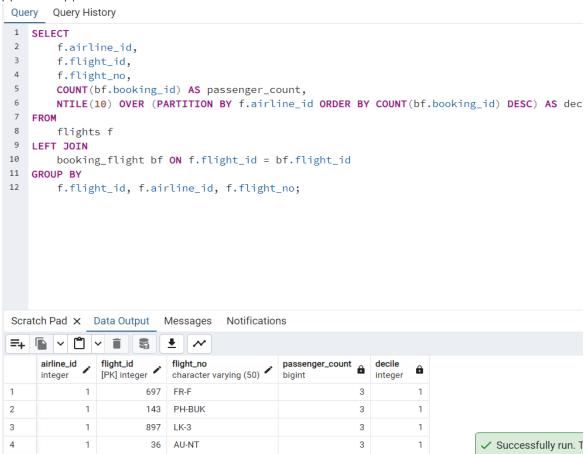
15.Use the CUME_DIST() window function to find the cumulative distribution of flights by duration for each airline.

Использовать оконную функцию CUME_DIST() для нахождения кумулятивного распределения рейсов по их продолжительности для каждой авиакомпании.



16. Write a query using the NTILE(10) function to divide flights by expected passenger load into ten deciles for each airline.

Написать запрос с использованием функции NTILE(10) для разделения рейсов по ожидаемой загруженности пассажирами на десять децилей для каждой авиакомпании.



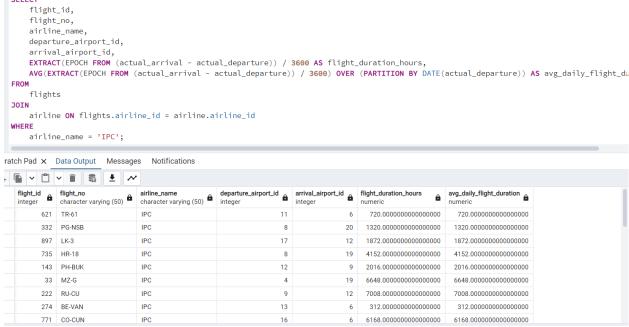
17. Using window functions, calculate the moving average of the number of passengers for the last three flights for each airline.

Использовать оконные функции для расчета скользящего среднего количества пассажиров за последние три рейса для каждой авиакомпании.

```
SELECT

    →
    □
    ✓
    □
    □
    ✓
    □
    ✓
    □
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓
    ✓</
       airline_id,
                                                                                                                                               airline_id moving_average_passengers
       AVG(passengers_count) AS moving_average_passengers
                                                                                                                                                                              1.6666666666666667
       SELECT
                                                                                                                                                                             1 3333333333333333
               f.airline id.
               f.flight id.
                                                                                                                                                                       1 000000000000000000000
               ROW_NUMBER() OVER (PARTITION BY f.airline_id ORDER BY f.ac 4
                                                                                                                                                                            1.33333333333333333
               COUNT(p.passenger_id) OVER (PARTITION BY f.airline_id, f.f
                                                                                                                                                                             1 33333333333333333
                                                                                                                                                                    1.33333333333333333
               booking_flight bf ON f.flight_id = bf.flight_id
                                                                                                                                                                            1.333333333333333333
                                                                                                                                                                      1.0000000000000000000000
               booking b ON bf.booking_id = b.booking_id
                                                                                                                                    10
                                                                                                                                                          10
                                                                                                                                                                            1.33333333333333333
                                                                                                                                    11
               passengers p ON b.passenger_id = p.passenger_id
                                                                                                                                                          11
                                                                                                                                                                      2.00000000000000000
) AS LastThreeFlights
                                                                                                                                    12
                                                                                                                                                          12
WHERE
                                                                                                                                    13
                                                                                                                                                           13
                                                                                                                                                                             1.6666666666666667
                                                                                                                                                                              1.6666666666666667
GROUP BY
       airline_id
                                                                                                                                    15
                                                                                                                                                           15
                                                                                                                                                                             2.00000000000000000
HAVING
                                                                                                                                                                              1.33333333333333333
       MAX(rn) = 3;
                                                                                                                                    17
                                                                                                                                                           17
                                                                                                                                                                              1.33333333333333333
                                                                                                                                    18
                                                                                                                                                           18
                                                                                                                                                                       19
                                                                                                                                                           19
                                                                                                                                                                       1.0000000000000000000000
                                                                                                                                    20
                                                                                                                                                          20
                                                                                                                                                                             1.33333333333333333
                                                                                                                                    21
                                                                                                                                                                              1.33333333333333333
```

18. Create a SQL query with window functions to compare the flight duration and the average flight duration of each day for a specific airline. Создать SQL-запрос с оконными функциями для сравнения продолжительности полета и средней продолжительности полетов за день для конкретной авиакомпании.



19.Implement the RANK() function to list the top 5 busiest airports based on the number of incoming and outgoing flights, using window functions to dynamically calculate rankings.

Применить функцию RANK() для вывода списка 5 самых загруженных аэропортов на основе количества прибывающих и отправляющихся рейсов, используя оконные функции для динамического расчета рангов.

