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 "KKG" 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';
```

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.

```
INSERT INTO flights (flight_id, flight_no, scheduled_departure, scheduled_arrival, departure_airport_id, arriva
VALUES
    (1001, 'SH-1', '2024-04-20', '2024-04-21', 18, 20, '335', '154', 51, 'Delayed', '2024-04-20', '2024-04-20',
    (1002, 'SH-1', '2024-04-26', '2024-04-26', 19, 20, '335', '154', 51, 'Delayed', '2024-04-26', '2024-04-26',
    (1003, 'SH-1', '2024-05-10', '2024-05-11', 17, 20, '335', '154', 51, 'Delayed', '2024-05-10', '2024-05-11',
```

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.

```sql
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".

```sql
INSERT INTO passengers (passenger_id, first_name, last_name, date_of_birth, gender, country_of_citizenship, cou
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%.

```sql
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.

```sql
DELETE FROM booking
WHERE price < 1000;
```

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.

```sql
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.

```sql
SELECT flights.*,
       airlines.airline_name,
       departure_airport.airport_name AS departure_airport,
       arrival_airport.airport_name AS arrival_airport
FROM flights
JOIN airport AS departure_airport ON flights.departure_airport_id = departure_airport.airport_id
JOIN airport AS arrival_airport ON flights.arrival_airport_id = arrival_airport.airport_id
JOIN airline AS airlines ON flights.airline_id = airlines.airline_id
WHERE arrival_airport.country = 'China';
```

Display a list of minor passengers and their destination.

Отобразить список несовершеннолетних пассажиров и их пункта назначения.

```sql
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;
```

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.

```sql
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.

```sql
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.

Query | Query History

```
SELECT arrival_airport_id, COUNT(arrival_airport_id) AS number_arriving_flights
FROM Flights
GROUP BY arrival_airport_id;
```

Data Output | Messages | Notifications

	arrival_airport_id integer	number_arriving_flights bigint
1	11	39
2	8	48
3	19	50
4	4	51
5	14	51
6	3	50
7	17	46
8	20	41
9	10	51
10	7	58
11	13	48
12	9	52
13	1	56
14	5	45
15	18	52
16	2	44
17	16	54

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.

Query | Query History

```
SELECT
    al.airline_name,
    AVG(fl.scheduled_arrival - fl.actual_arrival) AS avg_delay_arrival
FROM flights fl
JOIN airline al
ON al.airline_id = fl.airline_id
WHERE fl.scheduled_arrival>fl.actual_arrival
GROUP BY fl.airline_id, al.airline_name
```

Data Output | Messages | Notifications

	airline_name character varying (50)	avg_delay_arrival interval
1	SJS	110 days 19:12:00
2	YLQ	76 days 19:12:00
3	YLP	136 days 04:48:00
4	XLU	122 days 03:25:42.857143
5	IPC	120 days 03:12:00
6	PIR	124 days 04:00:00
7	SPI	78 days 18:00:00
8	MXW	196 days 08:00:00
9	YST	104 days
10	KKL	174 days 08:34:17.142857
11	SOZ	86 days 20:00:00
12	YDQ	103 days 21:00:00
13	BUO	79 days 01:50:46.153846
14	HYL	80 days 10:17:08.571429

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.

Query | Query History

```
SELECT pas.first_name, pas.last_name, SUM(B.weight_in_kg) AS total_baggage_weight
FROM passengers pas
JOIN booking bk ON pas.passenger_id = bk.passenger_id
JOIN baggage B ON B.booking_id = bk.booking_id
WHERE pas.gender = 'Male' AND pas.date_of_birth BETWEEN '1990-01-01' AND '2000-12-31'
GROUP BY pas.passenger_id, pas.first_name, pas.last_name;
```

Data Output | Messages | Notifications

	first_name character varying (50)	last_name character varying (50)	total_baggage_weight numeric
1	Pyotr	Colbron	79.86
2	Jodi	Ramm	49.52
3	Cletus	Mosey	82.00
4	Wood	Yve	72.49
5	Hurleigh	Goodwell	38.14
6	Wayne	Bangs	45.29
7	Zerk	Dusey	122.49
8	Tom	Gidney	76.38
9	Cesaro	McGennis	100.60
10	Peyton	Durnian	91.92
11	Warden	Huett	1.12
12	Hayward	Keeler	163.42
13	Ezechiel	Laydon	43.80
14	Spencer	Moffet	33.94
15	Brad	Apperley	49.18

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.

Query | Query History

```
SELECT DATE_TRUNC('month', created_at) AS month, SUM(price) AS total_ticket_cost
FROM Booking
GROUP BY month
ORDER BY month;
```

Data Output | Messages | Notifications

	month timestamp without time zone	total_ticket_cost numeric
1	2023-03-01 00:00:00	155294.81
2	2023-04-01 00:00:00	240649.84
3	2023-05-01 00:00:00	207361.58
4	2023-06-01 00:00:00	177058.76
5	2023-07-01 00:00:00	158703.58
6	2023-08-01 00:00:00	213237.05
7	2023-09-01 00:00:00	124509.29
8	2023-10-01 00:00:00	198113.29
9	2023-11-01 00:00:00	184333.41
10	2023-12-01 00:00:00	219119.78
11	2024-01-01 00:00:00	306355.62
12	2024-02-01 00:00:00	172192.21
13	2024-03-01 00:00:00	118348.41

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.

Query | Query History

```
1  SELECT pas.first_name, pas.last_name, COUNT(sc.security_check_id) AS num_security_checks
2  FROM passengers pas
3  JOIN security_check sc ON pas.passenger_id = sc.passenger_id
4  GROUP BY pas.passenger_id
5  HAVING COUNT(sc.security_check_id) > 2;
6
7
```

Data Output | Messages | Notifications

	first_name character varying (50)	last_name character varying (50)	num_security_checks bigint
1	Dyan	Chomiszewski	3
2	Howard	Igo	7
3	Pyotr	Colbron	4
4	Halimeda	Pinck	3
5	Sheela	Roux	3
6	Leroi	O'Shevlin	3
7	Ynez	Bortoloni	3
8	Butch	Bagby	3
9	Brad	Apperley	3
10	Karel	Fassman	3
11	Bettye	Berkley	5
12	Franny	Orry	3
13	Cletus	Mosey	3
14	Glenna	Hitter	4
15	Anetta	Frankcom	3

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

Query Query History

```
SELECT a.city, COUNT(*) AS num_flights_departing
FROM Flights f
JOIN Airport a ON f.departure_airport_id = a.airport_id
WHERE a.country != 'China'
GROUP BY a.city
ORDER BY a.city;
```

Data Output Messages Notifications

	city character varying (50)	num_flights_departing bigint
1	Cankova	53
2	Dubrava	51
3	Guihulñgan	61
4	Hilotongan	36
5	Itapetinga	54
6	Kepuh	58
7	Mlawe	49
8	New Sibonga	47
9	Paghmān	46
10	Polykárpi	41
11	Rogóźno	42
12	Sirari	52
13	Stupino	49
14	Wewit	55
15	Wilmington	58

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 Query History

```
SELECT COUNT(DISTINCT bk.passenger_id) as passenger_count
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
JOIN booking bk ON bk.booking_id = bf.booking_id
WHERE a.country = 'Poland'
```

Data Output Messages Notifications

	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".

Query | Query History

```
SELECT p.passenger_id, p.first_name, p.last_name, bg.baggage_id, bg.weight_in_kg
FROM passengers p
JOIN baggage_check bc ON bc.passenger_id = p.passenger_id
JOIN baggage bg ON bg.booking_id = bc.booking_id
WHERE bg.weight_in_kg > 20 AND bc.check_result = 'Checked'
ORDER BY p.passenger_id
```

Data Output | Messages | Notifications

	passenger_id integer	first_name character varying (50)	last_name character varying (50)	baggage_id integer	weight_in_kg numeric (4,2)
1	6	Tildy	Shackleford	250	36.01
2	9	Alphonso	Philippou	177	40.00
3	9	Alphonso	Philippou	142	40.13
4	10	Fairfax	Needham	380	23.98
5	10	Fairfax	Needham	190	23.20
6	13	Domenic	Blaszczak	21	41.13
7	14	Conrad	Southerell	257	20.45
8	15	Heloise	Yukhin	195	44.33
9	15	Heloise	Yukhin	65	29.06
10	15	Heloise	Yukhin	243	46.87
11	16	Artur	Leap	356	43.31
12	16	Artur	Leap	242	24.75
13	17	Wood	Yve	110	33.45
14	19	Muhammad	Fass	412	38.29
15	19	Muhammad	Fass	105	32.49

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

Определить среднюю цену билетов для каждой авиакомпании.

Query | Query History

```
SELECT a.airline_name, AVG(b.price) AS avg_price
FROM airline a
JOIN flights fl ON a.airline_id = fl.airline_id
JOIN booking_flight bf ON fl.flight_id = bf.flight_id
JOIN booking b ON bf.booking_id = b.booking_id
GROUP BY a.airline_name;
```

Scratch Pad | Data Output | Messages | Notifications

	airline_name character varying (50)	avg_price numeric
1	HOU	7097.1435294117647059
2	NHT	4805.8576470588235294
3	BLD	6072.3525000000000000
4	KIQ	4688.9681818181818182
5	CII	5140.4300000000000000
6	YBQ	5279.6300000000000000
7	SOZ	4080.3931578947368421
8	MXW	5724.7965000000000000
9	JDG	4504.2244000000000000
10	KKL	6531.1610000000000000
11	YLQ	3586.2042857142857143
12	RBR	5286.4410000000000000
13	DUC	3763.8170588235294118
14	PIR	4321.0191666666666667
15	PDN	3706.0815789473684211
16	BYH	5031.2208333333333333
17	YHB	4707.5923076923076923
18	SPI	4539.0171428571428571
19	IVA	5311.0294444444444444
20	HGZ	3748.9407142857142857
21	YDQ	4737.9810000000000000
22	YFK	5272.3103703703703704

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.flight_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	0.00000000000000000000
2	ZA-NL	53.15	53.15	100.00000000000000000000
3	US-NV	0	57.04	0.00000000000000000000
4	CA-AB	0	55.10	0.00000000000000000000
5	BR-PA	0	50.85	0.00000000000000000000
6	UA-32	60.81	72.93	66.66666666666666666700
7	RU-SAK	130.26	214.29	62.50000000000000000000
8	AU-WA	104.75	375.52	23.07692307692307692300
9	UA-46	0	131.06	0.00000000000000000000
10	AZ-BA	0	16.12	0.00000000000000000000
11	IN-WB	47.27	47.27	100.00000000000000000000
12	DZ-33	0	36.28	0.00000000000000000000
13	DE-SH	0	9.17	0.00000000000000000000
14	GB-NIR	0	79.86	0.00000000000000000000
15	AU-TAS	0	24.69	0.00000000000000000000
16	US-MT	0	133.57	0.00000000000000000000
17	DE-NW	36.73	87.10	25.00000000000000000000
18	VC-U-A	0	78.90	0.00000000000000000000
19	DK-84	49.06	111.76	33.33333333333333333300
20	NP-SA	0	57.17	0.00000000000000000000
21	MZ-P	0	109.55	0.00000000000000000000
22	US-WA	0	83.90	0.00000000000000000000

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

```sql
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.

```sql
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	11.0200000000000000
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	19.0566666666666667
21	MZ-P	27.3875000000000000
22	US-WA	27.0933333333333333

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.

```sql
SELECT
    a.airline_name,
    flight_id,
    COUNT(flight_id) OVER (PARTITION BY a.airline_id) AS total_flights_per_airline
FROM
    flights
JOIN
    airline a ON flights.airline_id = a.airline_id;
```

	airline_name character varying (50)	flight_id integer	total_flights_per_airline bigint
1	IPC	33	32
2	IPC	222	32
3	IPC	36	32
4	IPC	697	32
5	IPC	143	32
6	IPC	332	32
7	IPC	526	32
8	IPC	13	32
9	IPC	621	32
10	IPC	943	32
11	IPC	897	32
12	IPC	693	32
13	IPC	771	32
14	IPC	78	32
15	IPC	735	32
16	IPC	170	32
17	IPC	274	32
18	IPC	175	32
19	IPC	73	32
20	IPC	956	32
21	IPC	185	32

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).

```sql
SELECT
    airline_name,
    COUNT(flight_id) AS total_flights,
    RANK() OVER (ORDER BY COUNT(flight_id) DESC) AS airline_rank
FROM
    flights
JOIN
    airline ON flights.airline_id = airline.airline_id
GROUP BY
    airline_name
ORDER BY
    airline_rank;
```

	airline_name character varying (50)	total_flights bigint	airline_rank bigint
1	IPC	32	1
2	KMA	31	2
3	NHT	29	3
4	CSC	28	4
5	DNU	27	5
6	SPI	25	6
7	YHB	25	6
8	QIG	25	6
9	DUC	25	6
10	YEK	24	10
11	XLU	24	10
12	CII	24	10
13	IFH	23	13
14	NGL	23	13
15	PIR	23	13
16	0	22	16
17	KKL	22	16
18	KOQ	22	16
19	PQS	22	16
20	JKR	22	16
21	NQX	22	16
22	KGF	21	22

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

```sql
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;
```

	airport_name character varying (50)	incoming_flights bigint	airport_rank bigint
1	Hana Airport	61	1
2	Armidale Airport	58	2
3	Akunaq Heliport	56	3
4	Elorza Airport	56	3
5	Zephyrhills Municipal Airport	54	4
6	Pitalito Airport	52	5
7	Darchula Airport	52	5
8	Industrial Airpark	51	6
9	Henri Coandă International Airport	51	6
10	Garbaharey Airport	51	6
11	Longana Airport	50	7
12	Lime Acres Finsch Mine Airport	50	7
13	Figari Sud-Corse Airport	48	8
14	Fort Worth Alliance Airport	48	8
15	Ocean Falls Seaplane Base	47	9
16	Tom Price Airport	46	10
17	Delta County Airport	45	11
18	Alert Bay Airport	44	12
19	Melilla Airport	41	13
20	Bermuda Dunes Airport	39	14

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.

```sql
SELECT
    a.airline_name,
    f.flight_id,
    f.scheduled_departure,
    ROW_NUMBER() OVER (
        PARTITION BY a.airline_id ORDER BY f.scheduled_departure)
        AS flight_row_number
FROM
    flights f
JOIN
    airline a ON f.airline_id = a.airline_id
GROUP BY
    a.airline_name, f.flight_id, a.airline_id, f.scheduled_departure;
```

	airline_name character varying	flight_id integer	scheduled_departure timestamp without tim	flight_row_number bigint
1	IPC	175	2023-03-19 00:00:00	1
2	IPC	346	2023-03-28 00:00:00	2
3	IPC	36	2023-03-29 00:00:00	3
4	IPC	222	2023-04-16 00:00:00	4
5	IPC	693	2023-04-25 00:00:00	5
6	IPC	943	2023-04-28 00:00:00	6
7	IPC	274	2023-05-03 00:00:00	7
8	IPC	446	2023-06-12 00:00:00	8
9	IPC	605	2023-06-19 00:00:00	9
10	IPC	185	2023-06-26 00:00:00	10
11	IPC	621	2023-07-22 00:00:00	11
12	IPC	487	2023-07-27 00:00:00	12
13	IPC	143	2023-08-20 00:00:00	13
14	IPC	78	2023-08-28 00:00:00	14
15	IPC	897	2023-09-15 00:00:00	15
16	IPC	33	2023-09-21 00:00:00	16
17	IPC	357	2023-10-05 00:00:00	17
18	IPC	735	2023-11-02 00:00:00	18
19	IPC	221	2023-11-07 00:00:00	19
20	IPC	956	2023-11-19 00:00:00	20
21	IPC	854	2023-11-19 00:00:00	21
22	IPC	332	2023-11-26 00:00:00	22

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.

```sql
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
ORDER BY
    airline_id, scheduled_departure;
```

tch Pad | Data Output | Messages | Notifications

airline_id integer	flight_id [PK] integer	flight_no character varying (50)	scheduled_departure timestamp without time zone	previous_arrival timestamp without time zone	time_diff_minutes interval
1	175	PA-8	2023-03-19 00:00:00	[null]	[null]
1	346	US-MS	2023-03-28 00:00:00	2023-08-21 00:00:00	-2 days -10:24:00
1	36	AU-NT	2023-03-29 00:00:00	2023-10-25 00:00:00	-3 days -12:00:00
1	222	RU-CU	2023-04-16 00:00:00	2023-05-06 00:00:00	-08:00:00
1	693	US-LA	2023-04-25 00:00:00	2023-09-15 00:00:00	-2 days -09:12:00
1	943	CA-BC	2023-04-28 00:00:00	2023-08-04 00:00:00	-1 days -15:12:00
1	274	BE-VAN	2023-05-03 00:00:00	2023-06-28 00:00:00	-22:24:00
1	446	KH-9	2023-06-12 00:00:00	2023-06-01 00:00:00	04:24:00
1	605	AF-AZ	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.

Query | Query History

```sql
SELECT
    flight_id,
    flight_no,
    airport_name AS departing_airport,
    LEAD(airport_name) OVER (
        PARTITION BY departure_airport_id
        ORDER BY scheduled_departure)
        AS next_destination
FROM
    flights
JOIN
    airport ON flights.arrival_airport_id = airport.airport_id;
```

Scratch Pad | Data Output | Messages | Notifications

	flight_id integer	flight_no character varying (50)	departing_airport character varying (50)	next_destination character varying
1	168	IL-D	Darchula Airport	Longana Airport
2	231	IT-34	Longana Airport	Elorza Airport
3	335	US-MO	Elorza Airport	Fort Worth Alliance Airport
4	564	JP-12	Fort Worth Alliance Airport	Lime Acres Finsch Mine Airport
5	165	US-FL	Lime Acres Finsch Mine Airport	Armidale Airport
6	193	US-WA	Armidale Airport	Akunaq Heliport
7	994	US-IN	Akunaq Heliport	Darchula Airport
8	456	PG-CPM	Darchula Airport	Elorza Airport
9	61	AU-WA	Elorza Airport	Elorza Airport
10	838	MX-TAM	Elorza Airport	Zephyrhills Municipal Airport

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.

```sql
SELECT
    airline_name,
    flight_id,
    scheduled_departure
FROM (
    SELECT
        airline_name,
        flight_id,
        scheduled_departure,
        FIRST_VALUE(scheduled_departure) OVER (
            PARTITION BY a.airline_id, DATE(scheduled_departure)
            ORDER BY scheduled_departure)
        AS first_departure_of_day
    FROM
        flights
    JOIN
        airline a ON flights.airline_id = a.airline_id
) AS first_flights
WHERE
    scheduled_departure = first_departure_of_day;
```

	airline_name character varying (50)	flight_id integer	scheduled_departure timestamp without time zone
1	IPC	175	2023-03-19 00:00:00
2	IPC	346	2023-03-28 00:00:00
3	IPC	36	2023-03-29 00:00:00
4	IPC	222	2023-04-16 00:00:00
5	IPC	693	2023-04-25 00:00:00
6	IPC	943	2023-04-28 00:00:00
7	IPC	274	2023-05-03 00:00:00
8	IPC	446	2023-06-12 00:00:00
9	IPC	605	2023-06-19 00:00:00
10	IPC	185	2023-06-26 00:00:00
11	IPC	621	2023-07-22 00:00:00
12	IPC	487	2023-07-27 00:00:00
13	IPC	143	2023-08-20 00:00:00
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
17	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.

```sql
SELECT
    f.airline_id,
    a.airline_name,
    f.scheduled_departure::date AS flight_date,
    LAST_VALUE(f.arrival_airport_id) OVER (
        PARTITION BY f.airline_id, f.scheduled_departure::date
        ORDER BY f.scheduled_departure
    ) AS last_destination_airport_id,
    ap.airport_name AS last_destination_airport_name
FROM
    flights f
JOIN
    airline a ON f.airline_id = a.airline_id
JOIN
    airport ap ON f.arrival_airport_id = ap.airport_id
ORDER BY
    f.airline_id, flight_date;
```

	_id integer	airline_name character varying (50)	flight_date date	last_destinati integer	last_destination_airport_name character varying (50)
1	1	IPC	2023-03-19	19	Longana Airport
2	1	IPC	2023-03-28	18	Darchula Airport
3	1	IPC	2023-03-29	14	Industrial Airpark
4	1	IPC	2023-04-16	12	Elorza Airport
5	1	IPC	2023-04-25	13	Figari Sud-Corse Airport
6	1	IPC	2023-04-28	4	Garbaharey Airport
7	1	IPC	2023-05-03	6	Hana Airport
8	1	IPC	2023-06-12	13	Figari Sud-Corse Airport
9	1	IPC	2023-06-19	4	Garbaharey Airport
10	1	IPC	2023-06-26	5	Delta County Airport
11	1	IPC	2023-07-22	6	Hana Airport
12	1	IPC	2023-07-27	5	Delta County Airport
13	1	IPC	2023-08-20	9	Pitalito Airport
14	1	IPC	2023-08-28	11	Bermuda Dunes Airport
15	1	IPC	2023-09-15	12	Elorza Airport
16	1	IPC	2023-09-21	19	Longana Airport
17	1	IPC	2023-10-05	6	Hana Airport
18	1	IPC	2023-11-02	19	Longana Airport
19	1	IPC	2023-11-07	18	Darchula Airport
20	1	IPC	2023-11-19	5	Delta County Airport
21	1	IPC	2023-11-19	5	Delta County Airport

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.

Query | Query History

```sql
SELECT
    airline_name,
    flight_id,
    scheduled_departure,
    arrival_airport_id,
    LAST_VALUE(arrival_airport_id) OVER (
        PARTITION BY a.airline_id, DATE(scheduled_departure)
        ORDER BY scheduled_departure
    ) AS last_destination_of_day
FROM
    flights
JOIN
    airline a ON flights.airline_id = a.airline_id;
```

Scratch Pad | Data Output | Messages | Notifications

	airline_name character varying (50)	flight_id integer	scheduled_departure timestamp without time zone	arrival_airport_id integer	last_destination_of_day integer
1	IPC	175	2023-03-19 00:00:00	19	19
2	IPC	346	2023-03-28 00:00:00	18	18
3	IPC	36	2023-03-29 00:00:00	14	14
4	IPC	222	2023-04-16 00:00:00	12	12
5	IPC	693	2023-04-25 00:00:00	13	13
6	IPC	943	2023-04-28 00:00:00	4	4
7	IPC	274	2023-05-03 00:00:00	6	6
8	IPC	446	2023-06-12 00:00:00	13	13
9	IPC	605	2023-06-19 00:00:00	4	4

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.

```sql
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;
```

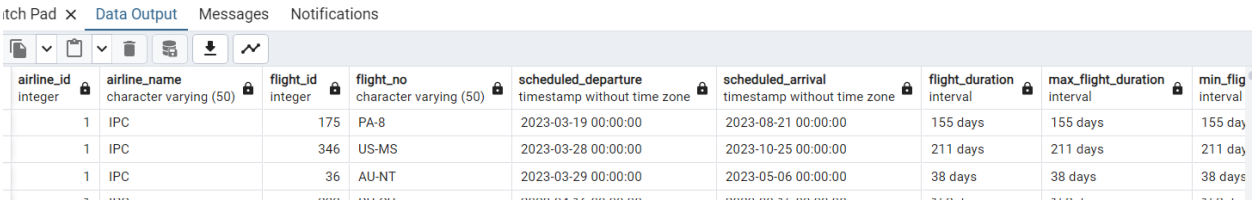
tch Pad | Data Output | Messages | Notifications

airline_id integer	airline_name character varying (50)	flight_id integer	flight_no character varying (50)	scheduled_departure timestamp without time zone	scheduled_arrival timestamp without time zone	flight_duration interval	avg_flight_duration interval
1	IPC	175	PA-8	2023-03-19 00:00:00	2023-08-21 00:00:00	155 days	2 days 23:15:00
1	IPC	346	US-MS	2023-03-28 00:00:00	2023-10-25 00:00:00	211 days	2 days 23:15:00
1	IPC	36	AU-NT	2023-03-29 00:00:00	2023-05-06 00:00:00	38 days	2 days 23:15:00
1	IPC	222	RU-CU	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
1	IPC	274	BE-VAN	2023-05-03 00:00:00	2023-06-01 00:00:00	29 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.

```
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,
    MAX(f.scheduled_arrival - f.scheduled_departure) OVER (
        PARTITION BY f.airline_id, f.scheduled_departure::date
    ) AS max_flight_duration,
    MIN(f.scheduled_arrival - f.scheduled_departure) OVER (
        PARTITION BY f.airline_id, f.scheduled_departure::date
    ) AS min_flight_duration
FROM
    flights f
JOIN
    airline a ON f.airline_id = a.airline_id
ORDER BY
    f.airline_id, f.scheduled_departure::date, f.scheduled_departure;
```

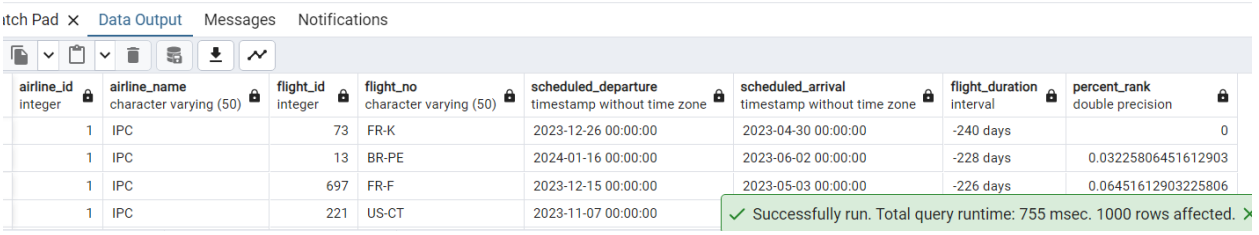


Scratch Pad | Data Output | Messages | Notifications

airline_id integer	airline_name character varying (50)	flight_id integer	flight_no character varying (50)	scheduled_departure timestamp without time zone	scheduled_arrival timestamp without time zone	flight_duration interval	max_flight_duration interval	min_flig interval
1	IPC	175	PA-8	2023-03-19 00:00:00	2023-08-21 00:00:00	155 days	155 days	155 day
1	IPC	346	US-MS	2023-03-28 00:00:00	2023-10-25 00:00:00	211 days	211 days	211 day
1	IPC	36	AU-NT	2023-03-29 00:00:00	2023-05-06 00:00:00	38 days	38 days	38 days

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.

```
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,
    PERCENT_RANK() OVER (
        PARTITION BY f.airline_id
        ORDER BY (f.scheduled_arrival - f.scheduled_departure)
    ) AS percent_rank
FROM
    flights f
JOIN
    airline a ON f.airline_id = a.airline_id
ORDER BY
    f.airline_id, percent_rank;
```



Scratch Pad | Data Output | Messages | Notifications

airline_id integer	airline_name character varying (50)	flight_id integer	flight_no character varying (50)	scheduled_departure timestamp without time zone	scheduled_arrival timestamp without time zone	flight_duration interval	percent_rank double precision
1	IPC	73	FR-K	2023-12-26 00:00:00	2023-04-30 00:00:00	-240 days	0
1	IPC	13	BR-PE	2024-01-16 00:00:00	2023-06-02 00:00:00	-228 days	0.0322580645161290 3
1	IPC	697	FR-F	2023-12-15 00:00:00	2023-05-03 00:00:00	-226 days	0.0645161290322580 6
1	IPC	221	US-CT	2023-11-07 00:00:00			

Successfully run. Total query runtime: 755 msec. 1000 rows affected.

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.

```sql
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,
    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
ORDER BY
    f.airline_id, quartile;
```

tch Pad × Data Output Messages Notifications

airline_id integer	airline_name character varying (50)	flight_id integer	flight_no character varying (50)	scheduled_departure timestamp without time zone	scheduled_arrival timestamp without time zone	flight_duration interval	quartile integer
1	IPC	73	FR-K	2023-12-26 00:00:00	2023-04-30 00:00:00	-240 days	1
1	IPC	13	BR-PE	2024-01-16 00:00:00	2023-06-02 00:00:00	-228 days	1
1	IPC	697	FR-F	2023-12-15 00:00:00	2023-05-03 00:00:00	-226 days	1
1	IPC	221	US-CT	2023-11-07 00:00:00			

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

14. Employ a combination of COUNT() to list each flight and the total number of flights by the same aircraft type.
Использовать комбинацию функции COUNT() для вывода списка каждого рейса и общего количества рейсов для одного и того же типа самолета.

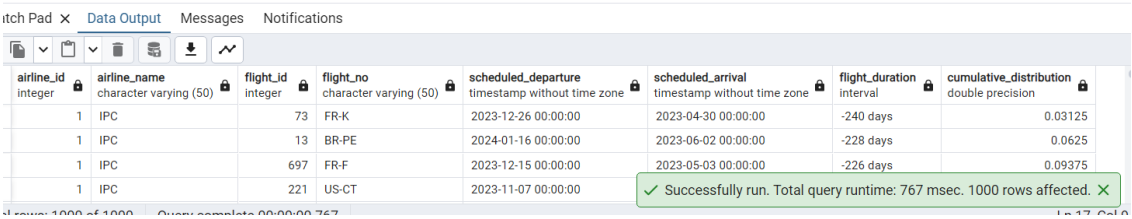
```sql
SELECT
        f.airline_id,
        a.airline_name,
        d.airport_name AS destination,
        COUNT(*) AS num_flights,
        RANK() OVER (PARTITION BY f.airline_id ORDER BY COUNT(*) DESC) AS desti
    FROM
        flights f
    INNER JOIN
        airport d ON f.arrival_airport_id = d.airport_id
    INNER JOIN
        airline a ON f.airline_id = a.airline_id
    GROUP BY
        f.airline_id,
        a.airline_name,
        d.airport_name
) AS FlightCounts
WHERE
    destination_rank = 1;
```

tch Pad × Data Output Messages Notifications

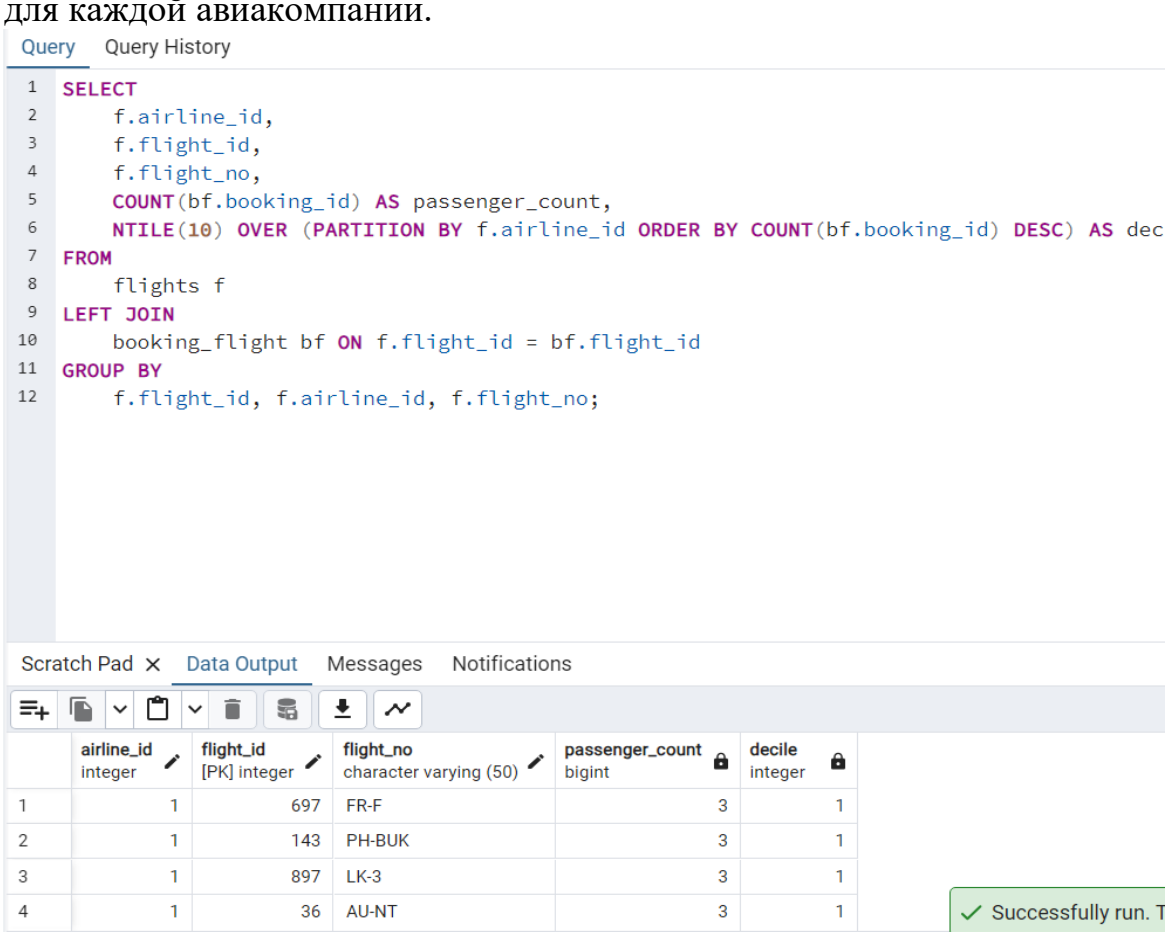
airline_id integer	airline_name character varying (50)	destination character varying (50)	num_flights bigint
1	IPC	Delta County Airport	5
2	PDN	Hana Airport	3
2	PDN	Fort Worth Alliance Airport	3
3	KLE	Armidale Airport	2

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.

```
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,
    CUME_DIST() OVER (
        PARTITION BY f.airline_id
        ORDER BY (f.scheduled_arrival - f.scheduled_departure)
    ) AS cumulative_distribution
FROM
    flights f
JOIN
    airline a ON f.airline_id = a.airline_id
ORDER BY
    f.airline_id, flight_duration;
```



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.

```sql
SELECT
    airline_id,
    AVG(passengers_count) AS moving_average_passengers
FROM (
    SELECT
        f.airline_id,
        f.flight_id,
        ROW_NUMBER() OVER (PARTITION BY f.airline_id ORDER BY f.ac
        COUNT(p.passenger_id) OVER (PARTITION BY f.airline_id, f.f
    FROM
        flights f
    JOIN
        booking_flight bf ON f.flight_id = bf.flight_id
    JOIN
        booking b ON bf.booking_id = b.booking_id
    JOIN
        passengers p ON b.passenger_id = p.passenger_id
) AS LastThreeFlights
WHERE
    rn <= 3
GROUP BY
    airline_id
HAVING
    MAX(rn) = 3;
```

	airline_id integer	moving_average_passengers numeric
1	1	1.6666666666666667
2	2	1.3333333333333333
3	3	1.0000000000000000000
4	4	1.3333333333333333
5	5	1.3333333333333333
6	6	1.3333333333333333
7	7	1.3333333333333333
8	8	1.3333333333333333
9	9	1.0000000000000000000
10	10	1.3333333333333333
11	11	1.0000000000000000000
12	12	2.0000000000000000
13	13	1.6666666666666667
14	14	1.6666666666666667
15	15	2.0000000000000000
16	16	1.3333333333333333
17	17	1.3333333333333333
18	18	1.0000000000000000000
19	19	1.0000000000000000000
20	20	1.3333333333333333
21	21	1.3333333333333333
22	22	1.3333333333333333

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.

```sql
SELECT
    flight_id,
    flight_no,
    airline_name,
    departure_airport_id,
    arrival_airport_id,
    EXTRACT(EPOCH FROM (actual_arrival - actual_departure)) / 3600 AS flight_duration_hours,
    AVG(EXTRACT(EPOCH FROM (actual_arrival - actual_departure)) / 3600) OVER (PARTITION BY DATE(actual_departure)) AS avg_daily_flight_du
FROM
    flights
JOIN
    airline ON flights.airline_id = airline.airline_id
WHERE
    airline_name = 'IPC';
```

ratch Pad × Data Output Messages Notifications

flight_id integer	flight_no character varying (50)	airline_name character varying (50)	departure_airport_id integer	arrival_airport_id integer	flight_duration_hours numeric	avg_daily_flight_duration numeric
621	TR-61	IPC	11	6	720.0000000000000000	720.0000000000000000
332	PG-NSB	IPC	8	20	1320.0000000000000000	1320.0000000000000000
897	LK-3	IPC	17	12	1872.0000000000000000	1872.0000000000000000
735	HR-18	IPC	8	19	4152.0000000000000000	4152.0000000000000000
143	PH-BUK	IPC	12	9	2016.0000000000000000	2016.0000000000000000
33	MZ-G	IPC	4	19	6648.0000000000000000	6648.0000000000000000
222	RU-CU	IPC	9	12	7008.0000000000000000	7008.0000000000000000
274	BE-VAN	IPC	13	6	312.0000000000000000	312.0000000000000000
771	CO-CUN	IPC	16	6	6168.0000000000000000	6168.0000000000000000

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.

```sql
SELECT
    airport_id,
    airport_name,
    outgoing_flights,
    incoming_flights,
    RANK() OVER (ORDER BY (outgoing_flights + incoming_flights)
                 DESC) AS airport_rank
FROM (
    SELECT
        a.airport_id,
        a.airport_name,
        a.country,
        a.state,
        a.city,
        COUNT(f1.flight_id)
        AS outgoing_flights,
        (SELECT COUNT(f2.flight_id)
         FROM flights f2
         WHERE f2.arrival_airport_id = a.airport_id)
        AS incoming_flights
    FROM
        airport a
    LEFT JOIN
        flights f1 ON a.airport_id = f1.departure_airport_id
    GROUP BY
        a.airport_id, a.airport_name, a.country, a.state, a.city
) AS airport_flights
ORDER BY
    airport_rank
```

	airport_id [PK] integer	airport_name character varying (50)	outgoing_flights bigint	incoming_flights bigint
1	12	Elorza Airport	63	56
2	6	Hana Airport	58	61
3	14	Industrial Airpark	61	51
4	7	Armidale Airport	52	58
5	10	Henri Coandă International Airport	58	51