

Control work

1.

```
SELECT airport_name FROM airports_data  
EXCEPT  
SELECT city FROM airports_data  
ORDER BY airport_name ASC;
```

The screenshot shows a SQL IDE interface. The top panel displays a query editor with the following SQL code:

```
--  
SELECT airport_name FROM airports_data  
EXCEPT  
SELECT city FROM airports_data  
ORDER BY airport_name ASC;
```

The bottom panel shows the output of the query, which is a table with 21 rows. The table has a single column named 'airport_name'. The data is as follows:

airport_name
{ "en": "Abakan Airport", "ru": "Абакан" }
{ "en": "Anapa Vityazevo Airport", "ru": "Витязево" }
{ "en": "Astrakhan Airport", "ru": "Астрахань" }
{ "en": "Barnaul Airport", "ru": "Барнаул" }
{ "en": "Begishevo Airport", "ru": "Бегишево" }
{ "en": "Belgorod International Airport", "ru": "Белгород" }
{ "en": "Beloyarskiy Airport", "ru": "Белоярский" }
{ "en": "Beslan Airport", "ru": "Беслан" }
{ "en": "Bogashevo Airport", "ru": "Богашёво" }
{ "en": "Bolshoye Savino Airport", "ru": "Пермь" }
{ "en": "Bratsk Airport", "ru": "Братск" }
{ "en": "Bryansk Airport", "ru": "Брянск" }
{ "en": "Bugulma Airport", "ru": "Бугульма" }
{ "en": "Cheboksary Airport", "ru": "Чебоксары" }
{ "en": "Chelyabinsk Balandino Airport", "ru": "Челябинск" }
{ "en": "Cherepovets Airport", "ru": "Череповец" }
{ "en": "Chita-Kadala Airport", "ru": "Чита" }
{ "en": "Chulman Airport", "ru": "Чульман" }
{ "en": "Domodedovo International Airport", "ru": "Домодедово" }
{ "en": "Donskoye Airport", "ru": "Донское" }
{ "en": "Elista Airport", "ru": "Элиста" }

2.

```
SELECT airport_name FROM airports_data
UNION
SELECT city FROM airports_data
ORDER BY airport_name ASC;
```

The screenshot shows a SQL playground interface with a dark theme. The top bar contains tabs for various database tables: console_5, aircrafts_data, airports_data, aircrafts_data, boarding_passes, bookings, flights, and seats. Below the tabs is a toolbar with icons for running queries, saving, and other functions. The main area displays a SQL query in a text editor, with line numbers 36 through 50 on the left. The query is as follows:

```
--
36 --
37
38
39 SELECT airport_name FROM airports_data
40 EXCEPT
41 SELECT city FROM airports_data
42 ORDER BY airport_name ASC;
43
44 ✓ SELECT airport_name FROM airports_data
45 UNION
46 SELECT city FROM airports_data
47 ORDER BY airport_name ASC;
48
49
50
```

Below the query editor, the 'Output' tab is selected, showing the results of the query. The results are displayed in a table with 205 rows. The first column is labeled 'airport_name'. The data is as follows:

airport_name
{ "en": "Abakan", "ru": "Абакан" }
{ "en": "Abakan Airport", "ru": "Абакан" }
{ "en": "Anadyr", "ru": "Анадырь" }
{ "en": "Anapa", "ru": "Анапа" }
{ "en": "Anapa Vityazevo Airport", "ru": "Витязево" }
{ "en": "Arkhangelsk", "ru": "Архангельск" }
{ "en": "Astrakhan", "ru": "Астрахань" }
{ "en": "Astrakhan Airport", "ru": "Астрахань" }
{ "en": "Barnaul", "ru": "Барнаул" }
{ "en": "Barnaul Airport", "ru": "Барнаул" }
{ "en": "Begishevo Airport", "ru": "Бегишево" }
{ "en": "Belgorod", "ru": "Белгород" }
{ "en": "Belgorod International Airport", "ru": "Белгород" }
{ "en": "Belgorod Airport", "ru": "Белгород" }

3.

```
SELECT aircraft_code, COUNT(*) AS flight_count
FROM flights
WHERE departure_airport = 'KZN'
      AND EXTRACT(MONTH FROM scheduled_departure) = 8
      AND EXTRACT(YEAR FROM scheduled_departure) = 2017
GROUP BY aircraft_code
HAVING COUNT(*) > 50
ORDER BY flight_count DESC, aircraft_code ASC;
```

The screenshot shows a SQL IDE interface with a dark theme. The top panel displays a list of database tables: console_5, aircrafts_data, airports_data, aircrafts_data, boarding_passes, and bookings. The main editor area contains a SQL query, with line 50 marked with a green checkmark. The query is as follows:

```
39 SELECT airport_name FROM airports_data
40 EXCEPT
41 SELECT city FROM airports_data
42 ORDER BY airport_name ASC;
43
44 SELECT airport_name FROM airports_data
45 UNION
46 SELECT city FROM airports_data
47 ORDER BY airport_name ASC;
48
49
50 ✓ SELECT aircraft_code, COUNT(*) AS flight_count
51 FROM flights
52 WHERE departure_airport = 'KZN'
53       AND EXTRACT(MONTH FROM scheduled_departure) = 8
54       AND EXTRACT(YEAR FROM scheduled_departure) = 2017
55 GROUP BY aircraft_code
56 HAVING COUNT(*) > 50
57 ORDER BY flight_count DESC, aircraft_code ASC;
58
```

The bottom panel shows the 'Result 10' tab, indicating 3 rows of data. The results are displayed in a table with two columns: aircraft_code and flight_count.

	aircraft_code	flight_count
1	CN1	62
2	SU9	62
3	CR2	54

4. -
5. -
6. -
7. -
8. -

SELECT

```
    aircrafts_data.model AS aircraft_type,  
    SUM(CASE WHEN EXTRACT(MONTH FROM  
flights.actual_departure) = 8 THEN ticket_flights.amount ELSE 0  
END) AS august_earnings,  
    SUM(CASE WHEN EXTRACT(MONTH FROM  
flights.actual_departure) = 9 THEN ticket_flights.amount ELSE 0  
END) AS september_earnings,  
    COUNT(CASE WHEN EXTRACT(MONTH FROM  
flights.actual_departure) = 8 THEN 1 END) AS  
august_passenger_count,  
    COUNT(CASE WHEN EXTRACT(MONTH FROM  
flights.actual_departure) = 9 THEN 1 END) AS  
september_passenger_count  
FROM  
    flights  
JOIN  
    aircrafts_data ON flights.aircraft_code =  
aircrafts_data.aircraft_code  
JOIN  
    ticket_flights ON flights.flight_id = ticket_flights.flight_id  
WHERE  
    EXTRACT(YEAR FROM flights.actual_departure) = 2017  
    AND EXTRACT(MONTH FROM flights.actual_departure) IN  
(8, 9)  
GROUP BY  
    aircrafts_data.model  
ORDER BY  
    aircrafts_data.model;  
;
```

119	
120	✓ SELECT
121	aircrafts_data.model AS aircraft_type,
122	SUM(CASE WHEN EXTRACT(MONTH FROM flights.actual_departure) = 8 THEN ticket_flights.amount ELSE 0 END) AS august_earnings,
123	SUM(CASE WHEN EXTRACT(MONTH FROM flights.actual_departure) = 9 THEN ticket_flights.amount ELSE 0 END) AS september_earnings,
124	COUNT(CASE WHEN EXTRACT(MONTH FROM flights.actual_departure) = 8 THEN 1 END) AS august_passenger_count,
125	COUNT(CASE WHEN EXTRACT(MONTH FROM flights.actual_departure) = 9 THEN 1 END) AS september_passenger_count
126	FROM
127	flights
128	JOIN
129	aircrafts_data ON flights.aircraft_code = aircrafts_data.aircraft_code
130	JOIN
131	ticket_flights ON flights.flight_id = ticket_flights.flight_id
132	WHERE
133	EXTRACT(YEAR FROM flights.actual_departure) = 2017
134	AND EXTRACT(MONTH FROM flights.actual_departure) IN (8, 9)
135	GROUP BY
136	aircrafts_data.model
137	ORDER BY
138	aircrafts_data.model;
139	

Output	Result 24
--------	-----------

1	{ "en": "Airbus A319-100", "ru": "Аэробус А319-100" }	858386700	0	16718
2	{ "en": "Airbus A321-200", "ru": "Аэробус А321-200" }	501645800	0	31900
3	{ "en": "Boeing 737-300", "ru": "Боинг 737-300" }	444710000	0	26809
4	{ "en": "Boeing 767-300", "ru": "Боинг 767-300" }	1335702700	0	38471
5	{ "en": "Boeing 777-300", "ru": "Боинг 777-300" }	1090045900	0	45569
6	{ "en": "Bombardier CRJ-200", "ru": "Бомбардье CRJ-200" }	633677900	0	47784
7	{ "en": "Cessna 208 Caravan", "ru": "Сессна 208 Караван" }	30473600	0	4611
8	{ "en": "Sukhoi SuperJet-100", "ru": "Сухой Суперджет-100" }	1616560500	0	114306