

1)

The screenshot shows a SQL console interface with a dark theme. The top bar indicates the current session is 'console_2'. The main editor area contains a SQL query. The query is as follows:

```
1. Write a query that displays all flights of a specific airline.

SELECT f.flight_id,
       f.sch_departure_time,
       f.sch_arrival_time,
       a.airline_name
FROM Flights f
JOIN Airline a ON f.airline_id = a.airline_id
WHERE a.airline_name = 'Airline 1';
```

Below the editor, the 'Output' tab is active, showing 'Result 40'. The column headers for the result are: flight_id, sch_departure_time, sch_arrival_time, and airline_name. The result area currently shows '0 rows'.

2)

The screenshot shows the same SQL console interface. The main editor area contains a new SQL query. The query is as follows:

```
2. Compose a query to obtain a list of all flights with the names of the airports.

SELECT
    f.flight_id,
    f.sch_departure_time,
    a.airport_name AS departing_airport
FROM Flights f
JOIN Airport a
    ON f.departing_airport_id = a.airport_id;
```

Below the editor, the 'Output' tab is active, showing 'Result 42'. The column headers for the result are: flight_id, sch_departure_time, and departing_airport. The result area currently shows '0 rows'.

4)

```
console console_2 x
Tx: Auto Playground
1 4. Create a query to display a list of passengers on a specific flight.
2
3 ✓ SELECT
4     p.passenger_id,
5     p.first_name,
6     p.last_name,
7     p.gender,
8     p.country_of_citizenship
9 FROM Booking b
10 JOIN Passengers p
11     ON b.passenger_id = p.passenger_id
12 WHERE b.flight_id = 123;
13
14
```

Output slabwork.public.passengers x

passenger_id	first_name	last_name	gender	country_of_citizenship
--------------	------------	-----------	--------	------------------------

3)

```
console console_2 x
Tx: Auto Playground
1 3. Create a query that finds all airlines that have no flights scheduled
2
3 ✓ SELECT a.airline_id,
4     a.airline_name,
5     a.airline_code,
6     a.airline_country
7 FROM Airline a
8 LEFT JOIN Flights f
9     ON f.airline_id = a.airline_id
10    AND f.sch_departure_time >= CURRENT_DATE
11    AND f.sch_departure_time < (CURRENT_DATE + INTERVAL '1 month')
12 WHERE f.flight_id IS NULL;
13
14
```

Output slabwork.public.airline x

airline_id	airline_name	airline_code	airline_country
2	Airline 2	AL02	USA
5	Airline 5	AL05	Kazakhstan
8	FlyHigh	BR001	Brazil
6	KazAir	121	Kazakhstan
4	Airline 4	AL04	UAE
1	Airline 1	AL01	Germany
3	Airline 3	AL03	y

9 rows

5)

5. Write a query that calculates the average, total, maximum and minimum

```
SELECT
    flight_id,
    AVG(ticket_price) AS avg_price,
    SUM(ticket_price) AS total_price,
    MAX(ticket_price) AS max_price,
    MIN(ticket_price) AS min_price
FROM Booking
GROUP BY flight_id
ORDER BY flight_id;
```

[42703] ОШИБКА: столбец "price" не существует
Позиция: 33

Output Result 46

flight_id	avg_price	total_price	max_price	min_price
-----------	-----------	-------------	-----------	-----------

0 rows

6)

6. Create a query that shows all flights flying to a specific country by

```
SELECT
    f.flight_id,
    a.airline_name,
    ap.airport_name AS arrival_airport,
    ap.country AS arrival_country,
    f.sch_departure_time,
    f.sch_arrival_time
FROM Flights f
JOIN Airline a
    ON f.airline_id = a.airline_id
JOIN Airport ap
    ON f.arriving_airport_id = ap.airport_id
WHERE ap.country = 'Germany';
```

Output Result 59 Result 58

flight_id	airline_name	arrival_airport	arrival_country	sch_departure_time
2	Airline 2	Frankfurt Airport	Germany	2025-11-28 08:55:25.253983
3	Airline 3	Munich Airport	Germany	2025-12-18 21:55:25.253983
12	Airline 5	Frankfurt Airport	Germany	2025-12-19 06:55:25.253983
13	KazAir	Munich Airport	Germany	2025-12-12 17:55:25.253983

40 rows

7)

```

1 ✓ SELECT p.passenger_id,
2       p.first_name,
3       p.last_name,
4       p.date_of_birth,
5       ap.airport_name AS arrival_airport,
6       ap.city          AS arrival_city,
7       ap.country       AS arrival_country,
8       f.flight_id
9 FROM Passengers p
10      JOIN Booking b
11          ON p.passenger_id = b.passenger_id
12      JOIN Flights f
13          ON b.flight_id = f.flight_id
14      JOIN Airport ap
15          ON f.arriving_airport_id = ap.airport_id
16 WHERE p.date_of_birth > CURRENT_DATE - INTERVAL '18 years';

```

Output: Result 61

passenger_id | first_name | last_name | date_of_birth | arrival_airport | arrival_city | arrival_country

0 rows

8)

```

SELECT
    p.first_name || ' ' || p.last_name AS full_name,
    p.passport_number,
    f.act_arrival_time AS arrival_time,
    ap.airport_name    AS arrival_airport,
    ap.city            AS arrival_city,
    ap.country         AS arrival_country
FROM Passengers p
JOIN Booking b
    ON p.passenger_id = b.passenger_id
JOIN Flights f
    ON b.flight_id = f.flight_id
JOIN Airport ap
    ON f.arriving_airport_id = ap.airport_id;

```

Output: slabwork.public.booking Result 64

	full_name	passport_number	arrival_time	arrival_airport	arrival_city
1	Lillian Rivera	P000000058	2025-11-26 22:01:25.253983	Nursultan Nazarbayev Airport	Astana
2	Lillian Rivera	P000000058	2025-11-26 22:01:25.253983	Nursultan Nazarbayev Airport	Astana
3	Lillian Rivera	P000000058	2025-11-26 22:01:25.253983	Nursultan Nazarbayev Airport	Astana

200 rows

9)

The screenshot shows a SQL playground interface with a dark theme. At the top, there are tabs for 'console' and 'console_2'. Below the tabs is a toolbar with icons for running, refreshing, and other actions. The main area contains a SQL query. A green checkmark is next to the first line of the query. Below the query, there is a table with 9 rows and 3 columns: 'origin_country', 'flight_id', and 'airline_name'. The table is titled 'Result 67' and 'Result 66'. The bottom status bar shows 'localhost > console_2'.

```
1  9. Print a list of flights where the home of airline country
2
3  ✓ SELECT
4      ap.country AS origin_country,
5      f.flight_id,
6      a.airline_name
7  FROM Flights f
8  JOIN Airline a
9      ON f.airline_id = a.airline_id
10 JOIN Airport ap
11     ON f.departing_airport_id = ap.airport_id
12 WHERE a.airline_country = ap.country
13 ORDER BY ap.country, f.flight_id;
```

	origin_country	flight_id	airline_name
1	France	56	AirEasy
2	France	126	AirEasy
3	France	196	AirEasy
4	Germany	43	Airline 1
5	Germany	64	Airline 1
6	Germany	113	Airline 1
7	Germany	134	Airline 1
8	Germany	183	Airline 1
9	Kazakhstan	12	Airline 5

localhost > console_2