

1)

The screenshot shows a PostgreSQL database interface. In the Database Explorer, the 'flights' table under the 'airline' schema is selected. A query is run in the console:

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

The results are displayed in a table titled 'Output' for the 'postgres.public.flights' service. The table contains 32 rows of flight information, including flight ID, number, departure and arrival dates, and airport IDs.

| flight_id | flight_no | scheduled_departure | scheduled_arrival | departure_airport_id | arrival_airport_id | depart |
|-----------|------------|---------------------|-------------------|----------------------|--------------------|--------|
| 1 | 13 BR-PE | 2024-01-16 | 2023-06-02 | 13 | 20 | 1891 |
| 2 | 33 MZ-G | 2023-09-21 | 2023-11-29 | 4 | 19 | <null> |
| 3 | 36 AU-NT | 2023-03-29 | 2023-05-06 | 12 | 14 | <null> |
| 4 | 73 FR-K | 2023-12-26 | 2023-04-30 | 18 | 9 | 144 |
| 5 | 78 US-VT | 2023-08-28 | 2023-08-02 | 20 | 11 | 335 |
| 6 | 143 PH-BUK | 2023-08-20 | 2023-10-24 | 12 | 9 | 919 |
| 7 | 170 SD-01 | 2023-12-02 | 2023-32 rows | 3 | 20 | 897 |
| 8 | 185 PA-0 | 2023-07-10 | 2023-09-09 | 10 | 10 | 100 |

2)

The screenshot shows a PostgreSQL database interface. In the Database Explorer, the 'flights' table under the 'flights' schema is selected. A query is run in the console:

```
SELECT f.flight_no, a.airport_name FROM flights f JOIN airport a ON f.departure_airport_id = a.airport_id;
```

The results are displayed in a table titled 'Result 1' for the 'postgres.public.flights' service. The table lists 501 flights and their corresponding departure airports.

| flight_no | airport_name |
|-----------|--------------------------------|
| US-CT | Elorza Airport |
| US-NM | Figari Sud-Corse Airport |
| FI-OL | Darchula Airport |
| RU-KR | Lime Acres Finsch Mine Airport |
| RO-DJ | Hana Airport |
| CA-SK | Darchula Airport |
| AU-TAS | Ocean Falls Seaplane Base |
| US-AZ | Figari Sud-Corse Airport |

3)

Database Explorer Tx: Auto playground

```
1 ✓ SELECT a.airline_name FROM airline a LEFT JOIN flights f ON a.airline_id = f.flight_id  
2 AND f.scheduled_departure BETWEEN current_date and current_date + interval '1 month' WHERE f.flight_id IS NULL;
```

Services

Tx > Output postgres.public.airline Result 1

| | airline_name |
|---|--------------|
| 1 | IPC |
| 2 | PDN |
| 3 | KLE |
| 4 | KHS |
| 5 | YLQ |
| 6 | NGL |
| 7 | 0 |
| 8 | QIG |

Database Consoles > database > console

2:112 CRLF UTF-8 4 spaces

This screenshot shows a PostgreSQL database interface. In the top navigation bar, there are tabs for 'Databases', 'Version control', and several icons. Below the navigation bar is the 'Database Explorer' sidebar, which lists databases like 'information_schema', 'pg_catalog', and 'public', and tables such as 'airline', 'airport', 'baggage', 'baggage_check', 'boarding_pass', and 'booking'. The main area is a 'console' tab where a query has been run. The query selects airline names from the 'airline' table, joining it with the 'flights' table on the airline ID. It filters for flights scheduled between the current date and one month from the current date, where the flight ID is null. The results show eight entries: IPC, PDN, KLE, KHS, YLQ, NGL, 0, and QIG. At the bottom right, there are status indicators for 'CRLF', 'UTF-8', '4 spaces', and a refresh icon.

4)

Database Explorer Tx: Auto playground

```
1 ✓ SELECT p.first_name, p.last_name FROM passengers p JOIN booking b 1<->1..n ON p.passenger_id = b.passenger_id  
2 JOIN booking_flight bf 1<->1..n: ON b.booking_id = bf.booking_id  
3 JOIN flights f 1..n->1: ON bf.flight_id = f.flight_id WHERE f.flight_no = 'US-CT';
```

Services

Tx > Output postgres.public.airline postgres.public.passengers

| first_name | last_name |
|------------|-----------|
| Alphonso | Philippou |
| Elfrida | Schukert |

Database Consoles > database > console

1:34 CRLF UTF-8 4 spaces

This screenshot shows a PostgreSQL database interface. The top navigation bar and sidebar are similar to the previous screenshot. The main console area contains a more complex query involving multiple joins between 'passenger', 'booking', 'booking_flight', and 'flights' tables. It filters for flights with flight number 'US-CT'. The results are displayed in a table with two rows, showing passenger names Alphonso Philippou and Elfrida Schukert. The bottom right corner shows status indicators for 'CRLF', 'UTF-8', '4 spaces', and a refresh icon.

5)

Database Explorer console flights

```

SELECT
    f.flight_id,
    f.flight_no,
    AVG(b.price) as average_price,
    SUM(b.price) as total_revenue,
    MAX(b.price) as maximum_price,
    MIN(b.price) as minimum_price,
    COUNT(b.booking_id) as tickets_sold
FROM flights f
JOIN booking_flight bf 1<->1: ON f.flight_id = bf.flight_id
JOIN booking b 1..n->1: ON bf.booking_id = b.booking_id
GROUP BY f.flight_id, f.flight_no;

```

Services

Tx > Output Result 7 > postgres.public.passengers

| | flight_id | flight_no | average_price | total_revenue | maximum_price | minimum_price | tickets_sold |
|---|-----------|-----------|-----------------------|------------------------|---------------|---------------|--------------|
| 1 | 273 | AU-WA | 8929.2 | 8929.2 | 8929.2 | 8929.2 | 8929.2 |
| 2 | 951 | P6-MPL | 3034.54 | 6069.08 | 5888.12 | 180.96 | |
| 3 | 70 | MY-12 | 683.27 | 683.27 | 683.27 | 683.27 | |
| 4 | 350 | CN-41 | 5936.768 | 29683.84 | 9792.28 | 2520.04 | |
| 5 | 539 | FR-F | 5874.09 | 17622.27 | 7927.75 | 3116.63 | |
| 6 | 874 | CA-MB | 4981.8866666666666667 | 14705.66 | 7822.69 | 1839.8 | |
| 7 | 176 | AZ-YE | 6895 | 1500 ~ of 501+ < > : | 9849.54 | 3942.36 | |

Database Consoles > database > console

5:17 CRLF UTF-8 4 spaces ⌂

6)

Database Explorer console airport flights

```

SELECT
    f.flight_id,
    f.flight_no,
    a.airline_name,
    dep.airport_name as departure_airport,
    dep.city as departure_city,
    arr.airport_name as arrival_airport,
    arr.city as arrival_city,
    arr.country as arrival_country
FROM flights f
JOIN airline a 1..n->1: ON f.airline_id = a.airline_id
JOIN airport dep 1..n->1: ON f.departure_airport_id = dep.airport_id
JOIN airport arr 1..n->1: ON f.arrival_airport_id = arr.airport_id
WHERE arr.country = 'China';

```

Services

Tx > Output Result 9 > Result 8

| | flight_id | flight_no | airline_name | departure_airport | departure_city | arrival_airport | arrival_country |
|---|-----------|-----------|--------------|-------------------|--------------------------|-----------------|--------------------------------|
| 1 | | 3 | FI-OL | SMM | Darchula Airport | Hilotongan | Elorza Airport |
| 2 | | 9 | IN-OR | IFH | Hana Airport | Wilmington | Pitalito Airport |
| 3 | | 13 | BR-PE | IPC | Figari Sud-Corse Airport | Itapetinga | Linfen |
| 4 | | 14 | US-IN | QIG | Darchula Airport | Hilotongan | Melilla Airport |
| 5 | | 23 | CN-65 | YLP | Alert Bay Airport | Dubrava | Sangzhou |
| 6 | | 24 | BR-T0 | CSC | Lime Acres | Tielu | Lime Acres Finsch Mine Airport |
| | | | | 238 rows | Report | Tielu | Pitalito Airport |

Database Consoles > database > console

14:29 CRLF UTF-8 4 spaces ⌂

7)

The screenshot shows a PostgreSQL database interface. In the Database Explorer, the 'airport' table is selected. A query is being typed into the 'console' tab:

```

SELECT
    p.first_name,
    p.last_name,
    arr.airport_name as arrival_airport,
    arr.country
FROM passengers p
JOIN booking b 1<->1..n ON p.passenger_id = b.passenger_id
JOIN booking_flight bf 1<->1..n ON b.booking_id = bf.booking_id
JOIN flights f 1..n<->1..n ON bf.flight_id = f.flight_id
JOIN airport arr 1..n<->1..n ON f.arrival_airport_id = arr.airport_id
WHERE EXTRACT(YEAR FROM AGE(CURRENT_DATE, p.date_of_birth)) < 18;

```

The results pane below shows four columns: first_name, last_name, arrival_airport, and country. The status bar indicates 0 rows.

8)

The screenshot shows a PostgreSQL database interface. In the Database Explorer, the 'airport' table is selected. A query is being typed into the 'console' tab:

```

SELECT
    p.first_name || ' ' || p.last_name as passenger_name,
    p.passport_number,
    f.scheduled_arrival
FROM passengers p
JOIN booking b 1<->1..n ON p.passenger_id = b.passenger_id
JOIN booking_flight bf 1<->1..n ON b.booking_id = bf.booking_id
JOIN flights f 1..n<->1..n ON bf.flight_id = f.flight_id;

```

The results pane below displays a table with three columns: passenger_name, passport_number, and scheduled_arrival. The data is as follows:

| | passenger_name | passport_number | scheduled_arrival |
|---|-----------------|-----------------|-------------------|
| 1 | Muhammad Fass | 109932770-9 | 2023-10-18 |
| 2 | Trevor Broun | 788025864-7 | 2023-05-08 |
| 3 | Auria Breffit | 570537341-4 | 2023-11-03 |
| 4 | Archie Toffel | 677556708-1 | 2023-05-18 |
| 5 | Sanders Biddles | 514546405-3 | 2023-05-24 |
| 6 | Sanders Biddles | 514546405-3 | 2023-05-24 |

The status bar indicates 8:43 CRLF UTF-8 4 spaces.

9)

The screenshot shows a PostgreSQL database interface with the following details:

Database Explorer sidebar:

- database
- > pg_catalog
- > public
 - tables 10
 - > airline
 - > airport
 - columns 7
 - keys 2
 - indexes 2
 - checks 1
 - > baggage
 - > baggage_check

console tab (selected):

```
1 ✓  SELECT a.country,COUNT(*) as flight_count
2   FROM flights f JOIN airline al [1..n<->1] ON f.airline_id = al.airline_id
3   JOIN airport a [1..n<->1] ON f.departure_airport_id = a.airport_id
4 WHERE al.airline_country = a.country GROUP BY a.country;
```

Services tab:

| country | flight_count |
|-----------|--------------|
| Indonesia | 6 |
| Slovenia | 2 |
| Greece | 1 |
| Russia | 5 |
| China | 58 |
| Brazil | 5 |
| ... | ... |

Bottom status bar:

Database Consoles > database > console

4:30 CRLF UTF-8 4 spaces