

L2 llab-2 Version control

Database Explorer

postgres 1 of 4

postres 1 of 3

public

tables 10

airline

airport

baggage

baggage_check

boarding_pass

booking

booking_flight

flights

console Tx: Auto

airport flights airline

Playground

postgres.public

```
1 ✓ SELECT f.flight_id,
2           a.airline_name,
3           dep.country AS origin_country
4   FROM flights f
5   JOIN airline a 1..n->1 ON f.airline_id = a.airline_id
6   JOIN airport dep 1..n->1 ON f.departure_airport_id = dep.airport_id
7   WHERE a.airline_country = dep.country
8   GROUP BY f.flight_id, a.airline_name, dep.country
9   ORDER BY dep.country;
```

Services

Tx Output Result 19

Database

postgres

console 3

airline 413

airport 45

flights 15

flight_id 141 333 579 783 948 1 37 39 48

airline_name PDN RBR YLP YLP YLP GYA KIQ SJS JKR

origin_country Brazil Brazil Brazil Brazil Brazil China China China China

74 rows

The screenshot shows a PostgreSQL database interface in a code editor-like environment. The top bar includes tabs for 'console' (which is active), 'flights', and 'booking'. The left sidebar shows a tree view of database objects under 'postgres': baggage, baggage_check, boarding_pass, booking (selected), booking_flight, flights, passengers, security_check, Database Objects, and Server Objects. The main area displays a SQL query:

```
1 ✓  SELECT
2     p.last_name,
3         DATE_PART('year', AGE(CURRENT_DATE, p.date_of_birth)) AS age,
4         arr.airport_name AS arrival_airport,
5         arr.country AS arrival_country
6     FROM passengers p
7     JOIN booking b 1<->1..n ON p.passenger_id = b.passenger_id
8     JOIN booking_flight bf 1<->1..n ON b.booking_id = bf.booking_id
9     JOIN flights f 1..n<->1..n ON bf.flight_id = f.flight_id
10    JOIN airport arr 1..n<->1..n ON f.arrival_airport_id = arr.airport_id
11    WHERE DATE_PART('year', AGE(CURRENT_DATE, p.date_of_birth)) < 18;
12
```

Below the query, the 'Services' section shows a transaction (Tx) and an output tab labeled 'Result 18'. The result grid has columns for last_name, age, arrival_airport, and arrival_country. A Snipping Tool window is open in the bottom right corner, stating 'Screenshot copied to clipboard'.

llab-2 Version control

Database Explorer

postgres

- > baggage
- > baggage_check
- > boarding_pass
- > booking
- > booking_flight
- > flights
- > passengers
- > security_check

flights

booking

Playground

postgres.public

```
1 ✓  SELECT f.flight_id,
          a.airline_name,
          dep.airport_name AS departure_airport,
          arr.airport_name AS arrival_airport,
          arr.country AS destination_country
     FROM Flights f
   JOIN airline a [Line 1] ON f.airline_id = a.airline_id
   JOIN airport dep [Line 2] ON f.departure_airport_id = dep.airport_id
   JOIN airport arr [Line 3] ON f.arrival_airport_id = arr.airport_id
 WHERE arr.country = 'China';
```

Services

Tx +, Output Result 13 ×

Database

postgres

- console
- booking
- flights

	flight_id	airline_name	departure_airport	arrival_airport	destination_country
1	3	SMM	Darchula Airport	Elorza Airport	China
2	9	IPH	Hana Airport	Pitalito Airport	China
3	13	IPC	Figari Sud-Corse Airport	Melilla Airport	China
4	14	QIG	Darchula Airport	Lime Acres Finsch Mine Airport	China
5	23	YLP	Alert Bay Airport	Lime Acres Finsch Mine Airport	China
6	24	CSC	Lime Acres Finsch Mine Airport	Pitalito Airport	China
7	25	PQS	Ocean Falls Seaplane Base	Elorza Airport	China
8	34	ONU	Bermuda Dunes Airp[238rows]	Lime Acres Finsch Mine Airport	China

Database Consoles > postgres > console

10:27 CRLF UTF-8 4 spaces

llab-2 Version control

Database Explorer

postgres

- > baggage
- > baggage_check
- > boarding_pass
- > booking
- > booking_flight
- > flights
- > passengers
- > security_check

Database Objects

Server Objects

console flights booking

Tx Auto Tx Auto

Playground

postgres.public

```
1 ✓ SELECT f.booking_id,
          ROUND(AVG(f.price), 2) AS avg_price,
          SUM(f.price) AS total_price,
          MAX(f.price) AS max_price,
          MIN(f.price) AS min_price
     FROM booking f
    GROUP BY f.booking_id
   ORDER BY f.booking_id;
```

Services

Tx Output Result 11

Database

postgres

booking_id	avg_price	total_price	max_price	min_price
1	7462.13	7462.13	7462.13	7462.13
2	4216.6	4216.6	4216.6	4216.6
3	5782.37	5782.37	5782.37	5782.37
4	102.96	102.96	102.96	102.96
5	6711.84	6711.84	6711.84	6711.84
6	7813.55	7813.55	7813.55	7813.55
7	1346.71	1346.71	1346.71	1346.71
8	5711.67	5711.67	5711.67	5711.67
9	7015.82	7015.82	7015.82	7015.82

Database Consoles > postgres > console

1:20 CRLF UTF-8 4 spaces

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

- Database Explorer:** Shows the `postgres` database with tables: `baggage`, `baggage_check`, `boarding_pass`, `booking`, `booking_flight`, `flights`, `passengers`, and `security_check`. It also lists `Database Objects` and `Server Objects`.
- Console:** A query window titled `console` containing the following SQL code:

```
1 ✓  SELECT p.first_name, p.passport_number
2   FROM passengers p
3  JOIN booking b  1<->1..n ON p.passenger_id = b.passenger_id
4  JOIN booking_flight bf  1<->1..n ON b.booking_id = bf.booking_id
5  JOIN flights f  1..n<->1..n ON bf.flight_id = f.flight_id
6 WHERE f.flight_id = 123;
```
- Services:** A table viewer window titled `Output` showing the results of the query:

first_name	passport_number
Gui	097333085-6

The table has 2 rows.
- Bottom Status Bar:** Shows the path `Database Consoles > postgres > console`, and system information like `1:20 CRLF UTF-8 4 spaces`.

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

Database Explorer (Left Panel):

- Connected to **postgres** database.
- Tables listed: airline, airport, baggage, baggage_check, boarding_pass, booking, booking_flight, flights, passengers, security_check.

Services (Bottom Left):

- Transactions (Tx): +, -.
- Database: postgres, console (selected).
- Table: flights (540 rows).

Console Tab (Top Right):

```
1 ✓ select a.airline_name
  2   from airline a
  3   where a.airline_id not in (
  4     select f.airline_id
  5       from flights f
  6       where f.sch_departure_time between current_date and (current_date + interval '1 month')
  7   );
```

Output Tab (Bottom Right):

airline_name
IPC
PDN
KLE
KHS
YLQ
NGL
B
QIG
NQX

Bottom Status Bar:

Database Consoles > postgres > console 6:30 CRLF UTF-8 4 spaces

The screenshot shows a PostgreSQL database interface in a code editor. The top section displays a query in the 'console' tab:

```
1 ✓ SELECT f.flight_id, a.airport_name AS departure_airport
FROM Flights f
JOIN airport a ON f.departure_airport_id = a.airport_id;
```

The 'Database Explorer' sidebar on the left lists tables under the 'airline' schema. The 'Services' panel at the bottom shows the results of the executed query, listing 501 rows of flight IDs and their corresponding departure airports.

flight_id	departure_airport
1	Elorza Airport
2	Figari Sud-Corse Airport
3	Darchula Airport
4	Lime Acres Finsch Mine Airport
5	Hana Airport
6	Darchula Airport
7	Ocean Falls Seaplane Base
8	Figari Sud-Corse Airport
9	Hana Airport

At the bottom of the screen, the taskbar shows the weather (6°C, Light rain), a search bar ('Поиск'), and system icons.

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

Database Explorer (Left Panel):

- Selected database: `postgres`
- Tables under `public`: `airline`, `airport`, `baggage`, `baggage_check`, `boarding_pass`, `booking`, `booking_flight`, `flights`.

console Tab (Top Right):

```
1 ✓  select *  
2   from flights  
3   where airline_id = (select airline_id from airline where airline_country = 'Senegal');
```

Services Tab (Bottom Right):

flight_id	sch_departure_time	sch_arrival_time	departing_gate	arriving_gate	airline_id	cre
11	<null>	<null>	1280	3288	46	2023-1
28	<null>	<null>	56	436	46	2023-1
88	<null>	<null>	137	123	46	2023-1
95	<null>	<null>	123	37	46	2023-1
126	<null>	<null>	91	10	46	2024-1
173	<null>	<null>	295	1437	46	2023-1
309	<null>	<null>	7285	162	46	2023-1
319	<null>	<null>	33	33	46	2023-1
...

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

- Database Explorer:** Shows the `postgres` database with 1 of 4 tables. The `public` schema contains the following tables:
 - `airline`
 - `airport`
 - `baggage`
 - `baggage_check`
 - `boarding_pass`
 - `booking`
 - `booking_flight`
 - `flights`
- Services:** Shows a transaction (`Tx`) and an output tab for `postgres.public.flights`. The output tab displays the structure of the `flights` table with columns: `flight_id`, `sch_departure_time`, `sch_arrival_time`, `departing_gate`, `arriving_gate`, `airline_id`, and `create`.
- Console:** Displays a query in the `console` tab:

```
1 ✓  select *  
2   from flights  
3   where airline_id = (select airline_id from airline where airline_name = 'Emirates');
```