

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres... × postgres/postgres... × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

Query Query History Scratch Pad

```
1 SELECT UPPER(airline_name) FROM airline;
```

Data Output Messages Notifications

Showing rows: 1 to 210 Page No: 1 of 1

	upper text
1	KAZAIR
2	AIREASY
3	FLYHIGH
4	FLYFLY
5	AIR_4CBC79
6	AIR_044C66
7	AIR_271BE5
8	AIR_857DF2
9	AIR_3C6463
10	AIR_2B2662
11	AIR_9B67CE
12	AIR_7CBBD3
13	AIR_B7E9AF
14	AIR_E28CBC

✓ Successfully run. Total query runtime: 129 msec. 210 rows affected.

Total rows: 210 Query complete 00:00:00.129 CRLF Ln 1, Col 19

Retrieve all airline names in uppercase.

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres... × postgres/postgres... × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

Query Query History

```
1 SELECT REPLACE(airline_name, 'Air', 'Aero') FROM airline;
```

Data Output Messages Notifications

	replace text	
1	KazAero	
2	AeroEasy	
3	FlyHigh	
4	FlyFly	
5	Aero_4cbc79	
6	Aero_044c66	
7	Aero_271be5	
8	Aero_857df2	
9	Aero_3c6463	
10	Aero_2b2662	
11	Aero_9b67ce	
12	Aero_7cbbd3	
13	Aero_b7e9af	
14	Aero_e28cbc	

Total rows: 210 Query complete 00:00:00.151

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres... × postgres/postgres... × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

Query Query History

```
1 UPDATE airlinE SET airline_name = REPLACE(airline_name, 'Air', 'Aero');
```

Data Output Messages Notifications

UPDATE 210

Query returned successfully in 138 msec.

Total rows: Query complete 00:00:00.138

Replace any occurrence of the word "Air" in airline names with "Aero".

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres... × postgres/postgres... × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

Query Query History Scratch Pad

```
1 SELECT flight_id FROM flights WHERE airline_id = '150'
2 INTERSECT
3 SELECT flight_id FROM flights WHERE airline_id = '150';
```

Data Output Messages Notifications

Showing rows: 1 to 200 Page No: 1 of 1

	flight_id integer
1	58
2	8
3	184
4	116
5	87
6	71
7	68
8	51
9	146
10	80
11	70
12	52
13	190
14	162

✓ Successfully run. Total query runtime: 148 msec. 200 rows affected. ✕

Total rows: 200 Query complete 00:00:00.148 CRLF Ln 3, Col 53

Find all flight numbers that coordinates with both airline 1 and airline 2.

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres... × postgres/postgres... × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

Query Query History Scratch Pad

```
1 SELECT airport_name FROM airport WHERE airport_name LIKE '%Airport%' UNION ALL
2 SELECT airport_name FROM airport WHERE airport_name LIKE '%Regional%';
```

Data Output Messages Notifications

Showing rows: 1 to 217 Page No: 1 of 1

	airport_name character varying (50)
1	Airport_445ca0
2	Airport_02a8d5
3	Airport_951c54
4	Airport_ae9978
5	Airport_c3fa42
6	Airport_17e48f
7	Airport_cdaf00
8	Airport_a1bb9d
9	Airport_59baff
10	Airport_583379
11	Airport_e77887
12	Airport_af6a76
13	Airport_cbf123
14	Airport_6a9a1b

✓ Successfully run. Total query runtime: 171 msec. 217 rows affected. ✕

Total rows: 217 Query complete 00:00:00.171 CRLF Ln 1, Col 79

Retrieve airports that contain the word "Reginal" and "Air" in their names.

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres... × postgres/postgres... × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

Query Query History Scratch Pad

```
1 SELECT first_name || ' ' || last_name AS passenger_name,
2    TO_CHAR(date_of_birth, 'Month DD, YYYY') AS formatted_birth_date
3 FROM passengers;
```

Data Output Messages Notifications

Showing rows: 1 to 221 Page No: 1 of 1

	passenger_name text	formatted_birth_date text
1	FN_9cffbe LN_fe800d	February 23, 1984
2	FN_158208 LN_a33...	August 29, 1996
3	FN_d26842 LN_529f...	September 02, 2003
4	FN_203e3d LN_e099...	June 11, 2000
5	FN_15bde0 LN_be7e...	January 23, 1977
6	FN_b3799d LN_26d...	February 26, 1976
7	FN_8d0c25 LN_5afd...	August 13, 2005
8	FN_edd5cf LN_1384...	February 17, 1983
9	FN_e54072 LN_d80...	December 22, 1995
10	FN_167f8b LN_af90...	April 11, 1984
11	FN_86b522 LN_365...	March 08, 2015
12	FN_1b3bbd LN_6b0...	March 02, 1981
13	FN_ce1678 LN_8e5e...	June 08, 2016
14	FN_b4cf35 LN_4edf...	November 02, 1977

Successfully run. Total query runtime: 157 msec. 221 rows affected.

Total rows: 221 Query complete 00:00:00.157 CRLF Ln 1, Col 29

Retrieve passenger names and format their birth dates as 'Month DD, YYYY'..o

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres... × postgres/postgres... × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

Query Query History Scratch Pad

```
1 SELECT flight_id, act_arrival_time - sch_arrival_time AS arrival_delay
2 FROM flights WHERE act_arrival_time > sch_arrival_time AND act_arrival_time IS NOT NULL;
```

Data Output Messages Notifications

Showing rows: 1 to 95 Page No: 1 of 1

	flight_id [PK] integer	arrival_delay interval
1	1	33 days 04:02:08.901154
2	8	9 days 02:12:53.60535
3	11	14 days 15:22:19.865783
4	13	23 days 04:23:34.495959
5	14	28 days 21:30:26.032038
6	18	21 days 08:15:26.756619
7	19	22 days 16:05:58.34506
8	22	14 days 05:36:35.672931
9	24	14 days 15:29:51.996244
10	25	27 days 21:37:59.808459
11	26	7 days 23:39:10.770775
12	27	20 days 20:16:17.465863
13	28	17 days 11:19:01.358084
14	29	12 days 12:37:53.887304

Total rows: 95 Query complete 00:00:00.310

✓ Successfully run. Total query runtime: 310 msec. 95 rows affected. ✕

CRLF Ln 2, Col 89

Find flight numbers that have been delayed based on the actual arrival time.

pgAdmin 4

File Object Tools Edit View Window Help

Welcome X postgres/postgres... X postgres/postgres... X postgres/postgres@PostgreSQL 17\* X

postgres/postgres@PostgreSQL 17

Query Query History Scratch Pad X

```
1 SELECT first_name, last_name, date_of_birth, EXTRACT(YEAR FROM AGE(NOW(), date_of_birth)) AS current_age,
2 CASE
3     WHEN EXTRACT(YEAR FROM AGE(NOW(), date_of_birth)) BETWEEN 0 AND 18 THEN 'Child'
4     WHEN EXTRACT(YEAR FROM AGE(NOW(), date_of_birth)) BETWEEN 18 AND 35 THEN 'Young'
5     WHEN EXTRACT(YEAR FROM AGE(NOW(), date_of_birth)) BETWEEN 36 AND 55 THEN 'Adult'
6 END AS age_group FROM Passengers ORDER BY current_age;
```

Data Output Messages Notifications

Showing rows: 1 to 221 Page No: 1 of 1

	first_name character varying (50)	last_name character varying (50)	date_of_birth date	current_age numeric	age_group text
1	Damir	Samatov	2020-05-15	5	Child
2	FN_5eebbc	LN_f9e7e5	2019-01-15	6	Child
3	FN_3df6aa	LN_902523	2019-04-06	6	Child
4	FN_aa3ed9	LN_e26c30	2018-01-11	7	Child
5	FN_2f81a3	LN_042662	2017-12-09	7	Child
6	FN_8f6025	LN_ea4b8f	2017-10-14	7	Child
7	FN_40a341	LN_cd4c65	2017-11-26	7	Child
8	FN_606bf7	LN_ca1aab	2017-05-11	8	Child
9	FN_14f5c0	LN_db99f4	2017-07-06	8	Child
10	FN_6ce138	LN_ea0d9b	2017-05-09	8	Child
11	FN_1bf2f9	LN_75c3c9	2016-12-22	8	Child
12	FN_fc9d4d	LN_23838b	2017-06-14	8	Child
13	FN_541171	LN_c23d66	2016-02-28	9	Child
14	FN_ce1678	LN_8e5e7a	2016-06-08	9	Child

✓ Successfully run. Total query runtime: 272 msec. 221 rows affected. X

Total rows: 221 Query complete 00:00:00.272 CRLF Ln 6, Col 43

Create a query that divides passengers into age groups like 'Young' and 'Adult' based on their birth date. Young passengers age between 18 and 35, Adult passengers age between 36 and 55.



pgAdmin 4

File Object Tools Edit View Window Help

Welcome x postgres/postgres... x postgres/postgres... x postgres/postgres@PostgreSQL 17\* x

postgres/postgres@PostgreSQL 17

Query Query History

```
1 SELECT booking_id, ticket_price,
2 CASE
3     WHEN ticket_price <= 700.00 THEN 'Cheap'
4     WHEN ticket_price > 700.00 AND ticket_price <= 1000.00 THEN 'Medium'
5     WHEN ticket_price > 1000.00 THEN 'Expensive'
6 END AS price_category FROM Booking;
```

Scratch Pad

Data Output Messages Notifications

Showing rows: 1 to 133 Page No: 1 of 1

	booking_id [PK] integer	ticket_price numeric (7,2)	price_category text
1	1	1087.16	Expensive
2	3	1025.80	Expensive
3	4	864.70	Medium
4	5	1109.78	Expensive
5	6	567.46	Cheap
6	9	762.06	Medium
7	10	970.63	Medium
8	11	856.55	Medium
9	12	1035.02	Expensive
10	14	782.82	Medium
11	16	956.43	Medium
12	18	812.36	Medium
13	19	1105.62	Expensive
14	22	848.26	Medium

Successfully run. Total query runtime: 108 msec. 133 rows affected.

Total rows: 133 Query complete 00:00:00.108 CRLF Ln 5, Col 26

Create a query that categorizes ticket prices based on their price as "Cheap," "Medium" or "Expensive."



pgAdmin 4

File Object Tools Edit View Window Help

Welcome x postgres/postgres... x postgres/postgres... x postgres/postgres@PostgreSQL 17\* x

postgres/postgres@PostgreSQL 17

Query Query History Scratch Pad x

```
1 SELECT airline_country, COUNT(airline_name) AS number_of_airlines
2 FROM Airline GROUP BY airline_country;
```

Data Output Messages Notifications

Showing rows: 1 to 207 Page No: 1 of 1

	airline_country character varying (50)	number_of_airlines bigint
1	Country_450da4	1
2	Poland	2
3	Country_06764b	1
4	Country_c89d21	1
5	Country_d92efa	1
6	Country_ee5e46	1
7	Country_6f61b4	1
8	Country_d0d5fa	1
9	Country_eca5f0	1
10	Country_366508	1
11	Country_8d4f33	1
12	Country_c63c27	1
13	Country_0b6c7a	1
14	Country_709736	1

Total rows: 207 Query complete 00:00:00.310

Successfully run. Total query runtime: 310 msec. 207 rows affected.

CRLF Ln 2, Col 14

Find number of airline names in each airline country.

pgAdmin 4

File Object Tools Edit View Window Help

Welcome x postgres/postgres... x postgres/postgres... x postgres/postgres@PostgreSQL 17\* x

postgres/postgres@PostgreSQL 17

Query Query History Scratch Pad

```
1 SELECT flight_id, sch_arrival_time, act_arrival_time, (act_arrival_time - sch_arrival_time) AS arrival_delay_duration
2 FROM flights
3 WHERE act_arrival_time > sch_arrival_time AND act_arrival_time IS NOT NULL;
```

Data Output Messages Notifications

Showing rows: 1 to 95 Page No: 1 of 1

	flight_id [PK] integer	sch_arrival_time timestamp without time zone	act_arrival_time timestamp without time zone	arrival_delay_duration interval
1	1	2025-09-28 10:01:06.71777	2025-10-31 14:03:15.618924	33 days 04:02:08.901154
2	8	2025-09-26 16:35:26.557003	2025-10-05 18:48:20.162353	9 days 02:12:53.60535
3	11	2025-09-28 09:48:35.548966	2025-10-13 01:10:55.414749	14 days 15:22:19.865783
4	13	2025-10-03 16:54:29.610692	2025-10-26 21:18:04.106651	23 days 04:23:34.495959
5	14	2025-09-28 03:27:12.608151	2025-10-27 00:57:38.640189	28 days 21:30:26.032038
6	18	2025-10-06 08:24:25.553887	2025-10-27 16:39:52.310506	21 days 08:15:26.756619
7	19	2025-09-26 18:59:05.501014	2025-10-19 11:05:03.846074	22 days 16:05:58.34506
8	22	2025-10-18 08:43:53.772478	2025-11-01 14:20:29.445409	14 days 05:36:35.672931
9	24	2025-10-01 04:37:09.965107	2025-10-15 20:07:01.961351	14 days 15:29:51.996244
10	25	2025-09-27 14:18:23.781128	2025-10-25 11:56:23.589587	27 days 21:37:59.808459
11	26	2025-10-21 12:35:33.591675	2025-10-29 12:14:44.36245	7 days 23:39:10.770775
12	27	2025-09-29 02:40:34.555327	2025-10-19 22:56:52.02119	20 days 20:16:17.465863
13	28	2025-10-04 02:41:44.981974	2025-10-21 14:00:46.340058	17 days 11:19:01.358084
14	29	2025-10-17 11:19:34.139033	2025-10-29 23:57:28.026337	12 days 12:37:53.887304

✓ Successfully run. Total query runtime: 269 msec. 95 rows affected. ✕

Total rows: 95 Query complete 00:00:00.272 CRLF Ln 3, Col 42

Find flights that arrived late according to their actual arrival time compared to the scheduled arrival time.