

# LAB8 Abdykamat Adilet. VIEW

## 1. Create a view to show details of all flights that are departing on a specific date.

```
CREATE VIEW flights_on_date AS SELECT * FROM flights  
WHERE DATE(sch_departure_time) = '2025-01-15';
```



The screenshot displays a SQL query editor interface. The top section, titled 'Query', shows the following SQL code:

```
1 CREATE VIEW flights_on_date AS SELECT * FROM flights  
2 WHERE DATE(sch_departure_time) = '2025-01-15';
```

The bottom section, titled 'Messages', shows the execution result:

```
CREATE VIEW  
  
Query returned successfully in 101 msec.
```

A green status bar at the bottom right indicates: '✓ Query returned successfully in 101 msec. ✕'. The bottom left corner shows 'Total rows: 0' and 'Query complete 00:00:00.101'. The bottom right corner shows 'CRLF Ln 2, Col 47'.

## 2. Create a view that shows bookings for flights scheduled to depart within the next week.

```
CREATE VIEW ext_week AS  
  
SELECT b.booking_id, bf.flight_id, f.sch_departure_time FROM booking b  
JOIN booking_flight bf on b.booking_id = bf.booking_id  
JOIN flights f on f.flight_id = bf.flight_id  
WHERE f.sch_departure_time BETWEEN now() AND now() + INTERVAL '7 days';
```

Query History

```
1 CREATE VIEW ext_week AS
2 SELECT b.booking_id, bf.flight_id, f.sch_departure_time FROM booking b
3 JOIN booking_flight bf on b.booking_id = bf.booking_id
4 JOIN flights f on f.flight_id = bf.flight_id
5 WHERE f.sch_departure_time BETWEEN now() AND now() + INTERVAL '7 days';
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 104 msec.

✓ Query returned successfully in 104 msec. ✕

Total rows: Query complete 00:00:00.104 CRLF Ln 5 Col 69

### 3. Create a view to show the top 5 most popular flight routes based on the number of bookings.

```
CREATE VIEW top5_routes AS
```

```
SELECT
```

```
    dep.airport_name AS from_airport,
```

```
    arr.airport_name AS to_airport,
```

```
    COUNT(*) AS total_bookings
```

```
FROM booking_flight bf
```

```
JOIN flights f ON bf.flight_id = f.flight_id
```

```
JOIN airport dep ON f.departing_airport_id = dep.airport_id
```

```
JOIN airport arr ON f.arriving_airport_id = arr.airport_id
```

```
GROUP BY dep.airport_name, arr.airport_name
```

```
ORDER BY total_bookings DESC
```

```
LIMIT 5;
```

Query

Query History

1

2

3

4

5

6

7

8

9

10

11

12

```
CREATE VIEW top5_routes AS
SELECT
    dep.airport_name AS from_airport,
    arr.airport_name AS to_airport,
    COUNT(*) AS total_bookings
FROM booking_flight bf
JOIN flights f ON bf.flight_id = f.flight_id
JOIN airport dep ON f.departing_airport_id = dep.airport_id
JOIN airport arr ON f.arriving_airport_id = arr.airport_id
GROUP BY dep.airport_name, arr.airport_name
ORDER BY total_bookings DESC
LIMIT 5;
```

Data Output

Messages

Notifications

CREATE VIEW

Query returned successfully in 113 msec.

✓ Query returned successfully in 113 msec. ✕

Total rows: Query complete 00:00:00.113 CRLF Ln 12 Col 9

#### 4. Create a view that lists all flights for a specific airline.

```
CREATE VIEW flights_by_airline AS
```

```
SELECT f.flight_id, a.airline_name, f.sch_departure_time, f.sch_arrival_time
```

FROM flights f

JOIN airline a on f.airline\_id = a.airline\_id

```
WHERE a.airline_name = 'Air Astana';
```

Query

Query History

1

2

3

4

5

▼

```
CREATE VIEW flights_by_airline AS
SELECT f.flight_id, a.airline_name, f.sch_departure_time, f.sch_arrival_time
FROM flights f
JOIN airline a on f.airline_id = a.airline_id
WHERE a.airline_name = 'Air Astana';
```

Data Output

Messages

Notifications

CREATE VIEW

Query returned successfully in 108 msec.

✓ Query returned successfully in 108 msec. ✕

**5. Modify the view created in task 4 to show only flights departing within the next 7 days for a specific airline.**

```
CREATE OR REPLACE VIEW flights_by_airline AS
```

```
SELECT f.flight_id, a.airline_name,
```

```
       f.sch_departure_time, f.sch_arrival_time
```

```
FROM flights f
```

```
JOIN airline a ON f.airline_id = a.airline_id
```

```
WHERE a.airline_name = 'Air Astana' AND f.sch_departure_time
```

```
BETWEEN now() AND now() + INTERVAL '7 days';
```

The screenshot shows a SQL query editor interface. The top tab is 'Query' and the bottom tab is 'Messages'. The 'Query' tab displays the following SQL code:

```
1 CREATE OR REPLACE VIEW flights_by_airline AS
2 SELECT f.flight_id, a.airline_name,
3        f.sch_departure_time, f.sch_arrival_time
4 FROM flights f
5 JOIN airline a ON f.airline_id = a.airline_id
6 WHERE a.airline_name = 'Air Astana' AND f.sch_departure_time
7 BETWEEN now() AND now() + INTERVAL '7 days';
```

The 'Messages' tab shows the following output:

```
CREATE VIEW
Query returned successfully in 93 msec.
```

At the bottom right, there is a green status bar with a checkmark icon and the text: 'Query returned successfully in 93 msec. X'.

**6. Create a view to show flights that are delayed by more than 24 hours.**

```
CREATE VIEW delayed_24h AS
```

```
SELECT flight_id, sch_departure_time, act_departure_time,
```

```
       act_departure_time - sch_departure_time FROM flights
```

```
WHERE act_departure_time - sch_departure_time > INTERVAL '24 hours';
```

Query
Query History

1 **CREATE VIEW** delayed\_24h **AS**  
2 **SELECT** flight\_id, sch\_departure\_time, act\_departure\_time,  
3     act\_departure\_time - sch\_departure\_time **FROM** flights  
4 **WHERE** act\_departure\_time - sch\_departure\_time > INTERVAL '24 hours';

Data Output
Messages
Notifications

CREATE VIEW  
  
Query returned successfully in 94 msec.

✓ Query returned successfully in 94 msec. ✕

**7. Create a view in which you can display the full name and country of origin of passengers who made bookings on Leffler-Thompson platform. Then show the list of that passengers.**

CREATE VIEW leffler\_passengers AS

SELECT p.first\_name || ' ' || p.last\_name as full\_name, p.country\_of\_citizenship FROM passengers p

JOIN tickets t on t.passenger\_id = p.passenger\_id

JOIN booking b on b.passenger\_id = p.passenger\_id

WHERE b.booking\_platform = 'Leffler-Thompson';

Query
Query History

1 **CREATE VIEW** leffler\_passengers **AS**  
2 **SELECT** p.first\_name || ' ' || p.last\_name **as** full\_name,  
3     p.country\_of\_citizenship **FROM** passengers p  
4 **JOIN** tickets t **on** t.passenger\_id = p.passenger\_id  
5 **JOIN** booking b **on** b.passenger\_id = p.passenger\_id  
6 **WHERE** b.booking\_platform = 'Leffler-Thompson';

Data Output
Messages
Notifications

CREATE VIEW  
  
Query returned successfully in 106 msec.

✓ Query returned successfully in 106 msec. ✕

## 8. Create a view that shows top 10 most visited countries.

```
CREATE VIEW top10_countries AS
```

```
SELECT arr.country, COUNT(*) AS visits FROM booking_flight bf
```

```
JOIN flights f ON bf.flight_id = f.flight_id
```

```
JOIN airport arr ON arr.airport_id = f.arriving_airport_id
```

```
GROUP BY arr.country
```

```
ORDER BY visits DESC
```

```
limit 10;
```

The screenshot displays a database query editor interface. The top section, titled 'Query History', contains a list of queries. The first query is selected and its SQL code is shown in the main editor area. The code is as follows:

```
1 CREATE VIEW top10_countries AS
2 SELECT arr.country, COUNT(*) AS visits FROM booking_flight bf
3 JOIN flights f ON bf.flight_id = f.flight_id
4 JOIN airport arr ON arr.airport_id = f.arriving_airport_id
5 GROUP BY arr.country
6 ORDER BY visits DESC
7 limit 10;
```

Below the query editor, there is a 'Data Output' section with tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is active, showing a confirmation message: 'CREATE VIEW' and 'Query returned successfully in 147 msec.'.

At the bottom right of the interface, a green status bar indicates: '✓ Query returned successfully in 147 msec. ✕'.

**9. Update any of the created views by adding new information in the view table.  
Show results.**

```
CREATE OR REPLACE VIEW flights_on_date AS
```

```
SELECT f.*, dep.city AS departure_city FROM flights f
```

```
JOIN airport dep on dep.airport_id = f.departing_airport_id
```

```
WHERE date(f.sch_departure_time) = '2025-01-15';
```

The screenshot shows a database query editor interface. The top tab is 'Query', and the bottom tab is 'Messages'. The query text is as follows:

```
1 CREATE OR REPLACE VIEW flights_on_date AS
2 SELECT f.*, dep.city AS departure_city FROM flights f
3 JOIN airport dep on dep.airport_id = f.departing_airport_id
4 WHERE date(f.sch_departure_time) = '2025-01-15';
```

The 'Messages' tab shows the following output:

```
CREATE VIEW
Query returned successfully in 70 msec.
```

A green notification box at the bottom right states: 'Query returned successfully in 70 msec. X'.

**RESULT:**

The screenshot shows a database query editor interface. The top tab is 'Query', and the bottom tab is 'Messages'. The query text is as follows:

```
1 SELECT * FROM flights_on_date
```

The 'Messages' tab shows the following output:

```
Data Output Messages Notifications
```

A toolbar at the bottom contains icons for various database operations, including a table icon, a dropdown menu, a trash icon, a download icon, a refresh icon, and an 'SQL' button.

flight_id	sch_departure_time	sch_arrival_time	departing_airport_id	arriving_airport_id	departing_gate	arriving_gate	airline
integer	timestamp without time zone	timestamp without time zone	integer	integer	text	character varying (50)	integ

# 10. Drop all existing views.

Query

Query History

1

DROP VIEW flights\_on\_date CASCADE;

2

DROP VIEW ext\_week CASCADE;

3

DROP VIEW top5\_routes CASCADE;

4

DROP VIEW flights\_by\_airline CASCADE;

5

DROP VIEW delayed\_24h CASCADE;

6

DROP VIEW leffler\_passengers CASCADE;

7

DROP VIEW top10\_countries CASCADE;

Data Output

Messages

Notifications

DROP VIEW

Query returned successfully in 120 msec.

Total rows: 0

Query complete 00:00:00.120

✓ Query returned successfully in 120 msec. ✕

CRI F Ln 2 Col 11