

דוח פרויקט שלב ב' - מאגר נתונים תעופה

Project Report Phase B - Aviation Database

מבוא כללי / General Introduction

פרויקט זה מתמקד בפיתוח מאגר נתונים מקיף עבור מערכת ניהול תעופה, הכולל מטוסים, טייסים, מפעילים, האנגרים ומוקדי תעופה. במסגרת שלב ב' ביצענו שאילתות מורכבות, עדכוני נתונים, מחיקות ובדיקות אילוצי מאגר הנתונים.

This project focuses on developing a comprehensive database for an aviation management system, including planes, pilots, operators, hangars, and aviation hubs. In Phase B, we performed complex queries, data updates, deletions, and database constraint testing.

8 SELECT מורכבות / Complex SELECT Queries

שאילתה 1: מידע מקיף על מטוסים עם ניתוח גיל

Query 1: Comprehensive Plane Information with Age Analysis

תיאור בעברית: שאילתה המציגה פרטים מלאים על כל המטוסים במערכת, כולל פרטי המפעיל, ההאנגר והיצרן. השאילתה מבצעת חישוב גיל המטוס בשנים ומפרקת את תאריך הייצור לשנה, חודש ויום. זהו מידע חיוני עבור ממשק המשתמש במערכת לניהול צי המטוסים.

קוד השאילתה:

```
SELECT
  p.Plane_id,
  p.Model,
  p.Status,
  p.Capacity as Plane_Capacity,
  EXTRACT(YEAR FROM p.ProductionDate) as Production_Year,
  EXTRACT(MONTH FROM p.ProductionDate) as Production_Month,
  EXTRACT(DAY FROM p.ProductionDate) as Production_Day,
  ROUND((CURRENT_DATE - p.ProductionDate) / 365.0, 1) as Age_Years,
```

o.Name as Operator_Name,
o.Type as Operator_Type,
h.Name as Hangar_Name,
h.Location as Hangar_Location,
pr.Pname as Producer_Name,
pr.Owner as Producer_Owner

FROM Plane p

JOIN Operator o ON p.Operator_id = o.Operator_id

JOIN Hangar h ON p.Hangar_id = h.Hangar_id

JOIN Producer pr ON p.Producer_id = pr.Producer_id

ORDER BY p.ProductionDate DESC;

The screenshot shows the pgAdmin 4 web interface in a browser. The left sidebar displays the database structure, including the 'public' schema with tables like 'hangar', 'hub', 'operator', and 'pilot'. The main pane shows a SQL query in the 'Query' tab. The query is a SELECT statement with various fields and joins. Below the query, the 'Data Output' tab shows an error message: 'ERROR: function year(date) does not exist LINE 6: YEAR(p.ProductionDate) as Production_Year, ^'. The status bar at the bottom indicates 'Total rows: 1' and 'Query complete 00:00:00.157'.

```
1 SELECT
2   p.Plane_id,
3   p.Model,
4   p.Status,
5   p.Capacity as Plane_Capacity,
6   EXTRACT(YEAR FROM p.ProductionDate) as Production_Year,
7   EXTRACT(MONTH FROM p.ProductionDate) as Production_Month,
8   EXTRACT(DAY FROM p.ProductionDate) as Production_Day,
9   ROUND((CURRENT_DATE - p.ProductionDate) / 365.0, 1) as Age_Years,
10  o.Name as Operator_Name,
11  o.Type as Operator_Type,
12  h.Name as Hangar_Name,
13  h.Location as Hangar_Location,
14  pr.Pname as Producer_Name,
15  pr.Owner as Producer_Owner
16 FROM Plane p
17 JOIN Operator o ON p.Operator_id = o.Operator_id
18 JOIN Hangar h ON p.Hangar_id = h.Hangar_id
19 JOIN Producer pr ON p.Producer_id = pr.Producer_id
20 ORDER BY p.ProductionDate DESC;
```

ERROR: function year(date) does not exist
LINE 6: YEAR(p.ProductionDate) as Production_Year,
^

Total rows: 1 Query complete 00:00:00.157

The screenshot shows the pgAdmin 4 web interface in a browser. The left sidebar displays the database structure, including schemas like 'public' and 'hangar'. The main pane shows a SQL query in the 'Query' tab. The query selects various attributes from the 'planes' and 'operators' tables, including plane details, production dates, and operator information. Below the query, the 'Data Output' tab shows the results of the query, displaying a table with columns for plane_id, model, status, plane_capacity, production_year, production_month, production_day, age_years, operator_name, operator_type, and operator_status. The table contains 6 rows of data.

plane_id	model	status	plane_capacity	production_year	production_month	production_day	age_years	operator_name	operator_type	operator_status
1	33 Beechcraft King Air	Grounded	371	2023	12	12	1.5	American Star 33	Private	Private
2	11 Embraer E190	Standby	161	2023	12	1	1.5	United Star 255	Carg	Carg
3	328 Airbus A350	Grounded	103	2023	10	20	1.6	Asian Flights 313	Exec	Exec
4	364 Airbus A380	Testing	371	2023	9	3	1.8	African Flights 37	Private	Private
5	6 Antonov An-124	Training	237	2023	8	7	1.8	National Airways 282	Com	Com
6	89 Canadair Regional Jet	Emergency Ready	298	2023	5	23	2.1	Atlantic Sky 239	Carg	Carg

שאלתה 2: סטטיסטיקות צי מטוסים לפי מפעיל עם פרטי מוקד

Query 2: Fleet Statistics by Operator with Hub Details

תיאור בעברית: שאלתה המרכזת מידע על המפעילים השונים במערכת יחד עם פרטי המוקדים שלהם וסטטיסטיקות מפורטות על צי המטוסים. השאלתה מחשבת את מספר המטוסים הפעיל, קיבולת ממוצעת וקיבולת כוללת. מידע זה חיוני להערכת יכולות התפעול של כל מפעיל.

קוד השאלתה:

```
SELECT
    o.Operator_id,
    o.Name as Operator_Name,
    o.Type,
    o.Fleet_Size,
    hub.Name as Hub_Name,
    hub.Location as Hub_Location,
    hub.IATA_code,
    hub.Capacity as Hub_Capacity,
    COUNT(p.Plane_id) as Actual_Planes_Count,
    ROUND(AVG(p.Capacity), 2) as Avg_Plane_Capacity,
    SUM(p.Capacity) as Total_Passenger_Capacity
FROM Operator o
JOIN Hub hub ON o.Hub_id = hub.Hub_id
```

```
LEFT JOIN Plane p ON o.Operator_id = p.Operator_id
GROUP BY o.Operator_id, o.Name, o.Type, o.Fleet_Size, hub.Name, hub.Location,
hub.IATA_code, hub.Capacity
ORDER BY Total_Passenger_Capacity DESC NULLS LAST;
```

The screenshot displays the pgAdmin 4 web interface. The top navigation bar includes 'File', 'Object', 'Tools', and 'Help' menus, along with a user profile 'raphster321@gmail.com (internal)'. The left sidebar shows a tree view with 'Servers (2)' expanded. The main pane is titled 'postgres/myuser@postgresql serverrrr' and contains a SQL query editor. The query is as follows:

```
1 SELECT
2     o.Operator_id,
3     o.Name as Operator_Name,
4     o.Type,
5     o.Fleet_Size,
6     hub.Name as Hub_Name,
7     hub.Location as Hub_Location,
8     hub.IATA_code,
9     hub.Capacity as Hub_Capacity,
10    COUNT(p.Plane_id) as Actual_Planes_Count,
11    ROUND(AVG(p.Capacity), 2) as Avg_Plane_Capacity,
12    SUM(p.Capacity) as Total_Passenger_Capacity
13 FROM Operator o
14 JOIN Hub hub ON o.Hub_id = hub.Hub_id
15 LEFT JOIN Plane p ON o.Operator_id = p.Operator_id
16 GROUP BY o.Operator_id, o.Name, o.Type, o.Fleet_Size, hub
17 ORDER BY Total_Passenger_Capacity DESC NULLS LAST;
18
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The status bar at the bottom indicates 'Total rows: 400', 'Query complete 00:00:00.081', 'CRLF', and 'Ln 18, Col 1'.

	operator_id integer	operator_name character varying (50)	type character varying (30)	fleet_size integer
1	149	Pacific Airways 149	Regional	85
2	381	Continental Sky 381	Cargo	120
3	349	African Sky 349	Government	476
4	343	United Aviation 343	Private	179
5	26	African Star 26	International	384
6	33	American Star 33	Private	452

שאלת 3: השמות טייסים עם ניתוח ניסיון מפורט

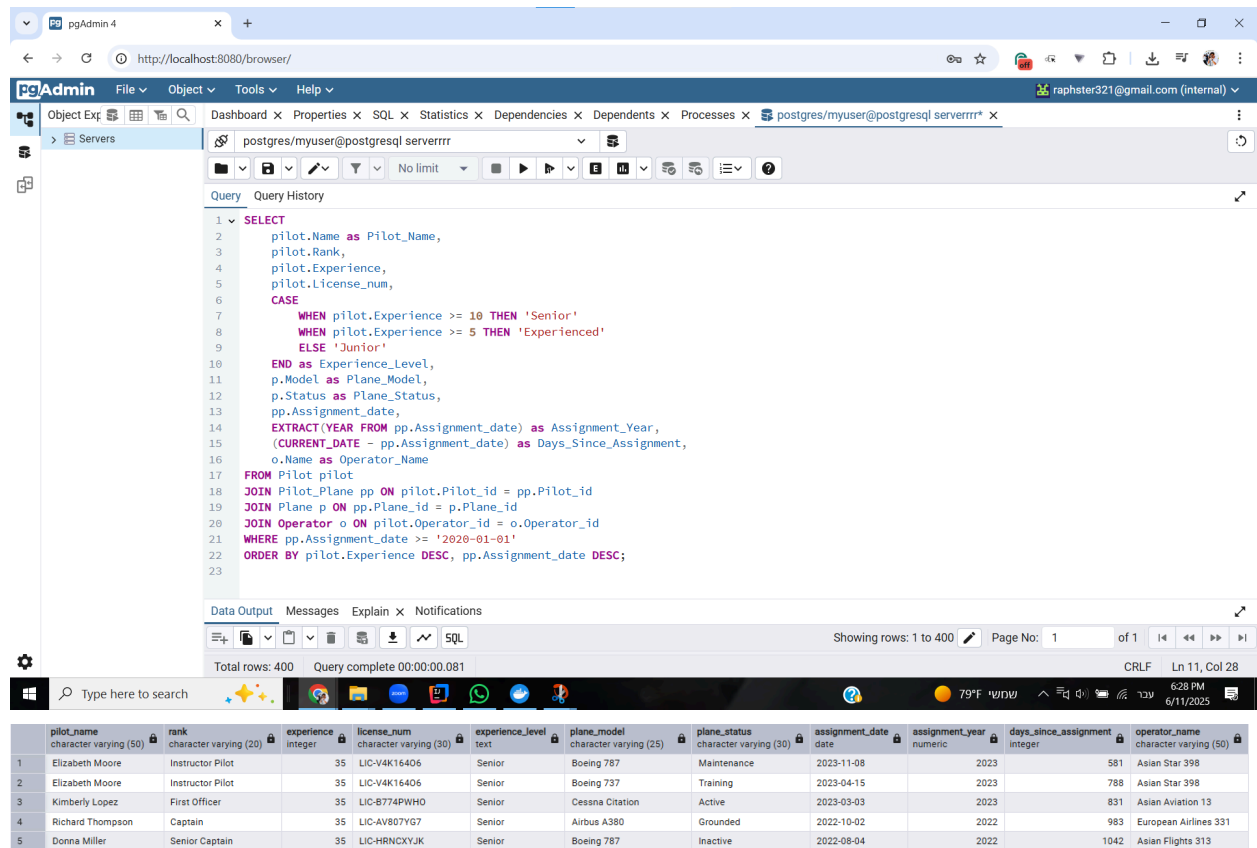
Query 3: Pilot Assignments with Detailed Experience Analysis

תיאור בעברית: שאלתה המציגה את כל השמות הטייסים במערכת עם פירוט רמת הניסיון (זוטר/מנוסה/בכיר), פרטי המטוסים המוקצים ומשך זמן ההשמה. השאלתה כוללת חישוב מספר הימים מאז ההשמה וסיווג הטייסים לרמות ניסיון שונות.

קוד השאלתה:

```
SELECT
  pilot.Name as Pilot_Name,
  pilot.Rank,
  pilot.Experience,
  pilot.License_num,
  CASE
    WHEN pilot.Experience >= 10 THEN 'Senior'
    WHEN pilot.Experience >= 5 THEN 'Experienced'
    ELSE 'Junior'
  END as Experience_Level,
  p.Model as Plane_Model,
  p.Status as Plane_Status,
  pp.Assignment_date,
  EXTRACT(YEAR FROM pp.Assignment_date) as Assignment_Year,
  (CURRENT_DATE - pp.Assignment_date) as Days_Since_Assignment,
  o.Name as Operator_Name
FROM Pilot pilot
JOIN Pilot_Plane pp ON pilot.Pilot_id = pp.Pilot_id
JOIN Plane p ON pp.Plane_id = p.Plane_id
JOIN Operator o ON pilot.Operator_id = o.Operator_id
```

WHERE pp.Assignment_date >= '2020-01-01'
ORDER BY pilot.Experience DESC, pp.Assignment_date DESC;



The screenshot shows the pgAdmin 4 web interface. The top navigation bar includes 'Object Explorer', 'Tools', and 'Help'. The main pane displays a SQL query in the 'Query' tab. The query is a complex SELECT statement joining pilot, plane, and operator tables, filtering by assignment date, and ordering by pilot experience and assignment date. The bottom pane shows the 'Data Output' tab with 400 rows of results. The results table has columns for pilot details, plane details, assignment details, and operator details.

pilot_name	rank	experience	license_num	experience_level	plane_model	plane_status	assignment_date	assignment_year	days_since_assignment	operator_name
Elizabeth Moore	Instructor Pilot	35	LIC-V4K16406	Senior	Boeing 787	Maintenance	2023-11-08	2023	581	Asian Star 398
Elizabeth Moore	Instructor Pilot	35	LIC-V4K16406	Senior	Boeing 787	Training	2023-04-15	2023	788	Asian Star 398
Kimberly Lopez	First Officer	35	LIC-8774PWHO	Senior	Cessna Citation	Active	2023-03-03	2023	831	Asian Aviation 13
Richard Thompson	Captain	35	LIC-AV807YG7	Senior	Airbus A380	Grounded	2022-10-02	2022	983	European Airlines 331
Donna Miller	Senior Captain	35	LIC-HRNCXYJK	Senior	Boeing 787	Inactive	2022-08-04	2022	1042	Asian Flights 313

שאלת 4: ניתוח מטוסים לפי יצרן עם סטטיסטיקות גיל וסטטוס

Query 4: Planes Analysis by Producer with Age and Status Statistics

תיאור בעברית: שאלת המציגה פילוח מטוסים לפי יצרנים, כולל ספירת מטוסים תפעוליים ובתחזוקה, חישוב גיל ממוצע של הצי, וסטטיסטיקות ביצועים. השאלתה מספקת תמונה מקיפה על איכות ומצב הצי של כל יצרן.

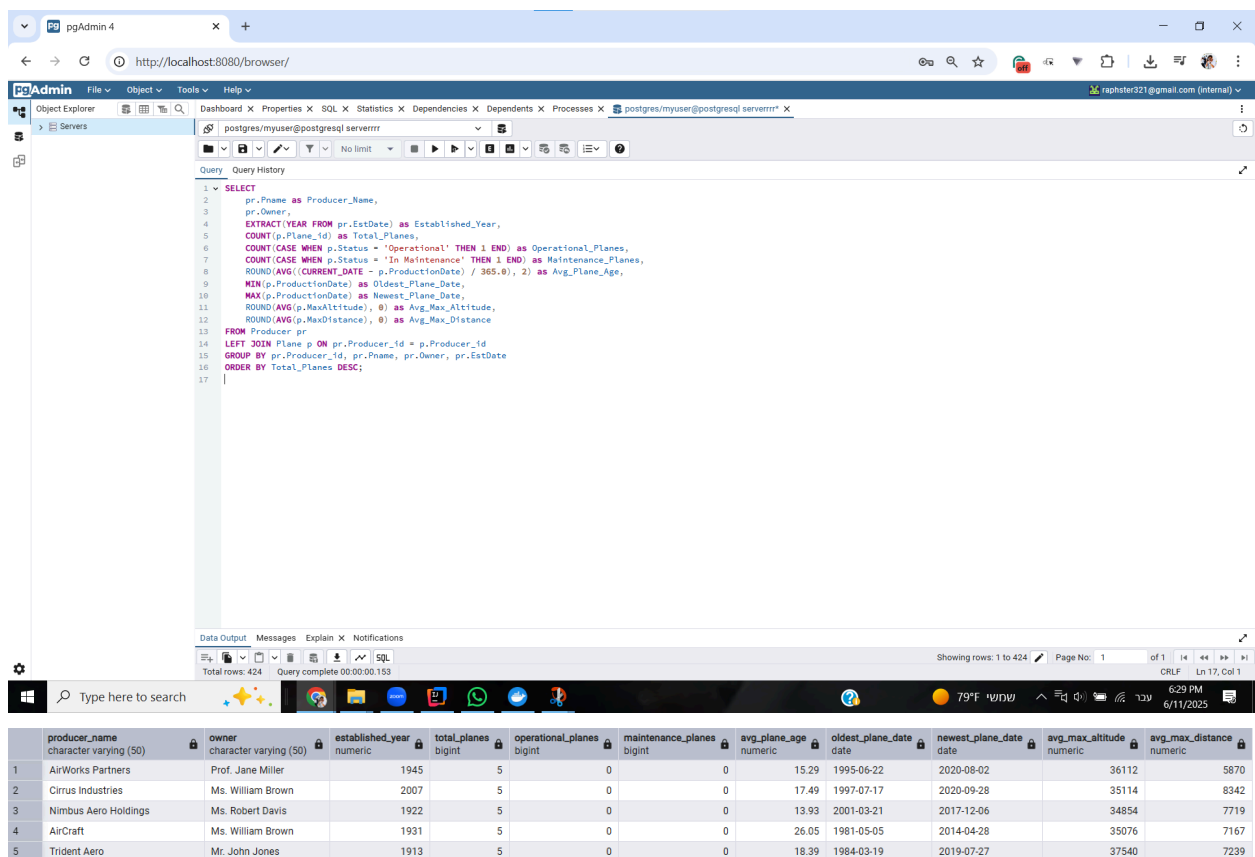
קוד השאלתה:

```
SELECT
    pr.PName as Producer_Name,
    pr.Owner,
    EXTRACT(YEAR FROM pr.EstDate) as Established_Year,
    COUNT(p.Plane_id) as Total_Planes,
    COUNT(CASE WHEN p.Status = 'Operational' THEN 1 END) as Operational_Planes,
    COUNT(CASE WHEN p.Status = 'In Maintenance' THEN 1 END) as Maintenance_Planes,
```

```

ROUND(AVG((CURRENT_DATE - p.ProductionDate) / 365.0), 2) as Avg_Plane_Age,
MIN(p.ProductionDate) as Oldest_Plane_Date,
MAX(p.ProductionDate) as Newest_Plane_Date,
ROUND(AVG(p.MaxAltitude), 0) as Avg_Max_Altitude,
ROUND(AVG(p.MaxDistance), 0) as Avg_Max_Distance
FROM Producer pr
LEFT JOIN Plane p ON pr.Producer_id = p.Producer_id
GROUP BY pr.Producer_id, pr.Pname, pr.Owner, pr.EstDate
ORDER BY Total_Planes DESC;

```



The screenshot shows the pgAdmin 4 interface with a SQL query executed. The query calculates various statistics for aircraft producers. The results table shows data for five producers: AirWorks Partners, Cirrus Industries, Nimbus Aero Holdings, AirCraft, and Trident Aero.

producer_name	owner	established_year	total_planes	operational_planes	maintenance_planes	avg_plane_age	oldest_plane_date	newest_plane_date	avg_max_altitude	avg_max_distance
1 AirWorks Partners	Prof. Jane Miller	1945	5	0	0	15.29	1995-06-22	2020-08-02	36112	5870
2 Cirrus Industries	Ms. William Brown	2007	5	0	0	17.49	1997-07-17	2020-09-28	35114	8342
3 Nimbus Aero Holdings	Ms. Robert Davis	1922	5	0	0	13.93	2001-03-21	2017-12-06	34854	7719
4 AirCraft	Ms. William Brown	1931	5	0	0	26.05	1981-05-05	2014-04-28	35076	7167
5 Trident Aero	Mr. John Jones	1913	5	0	0	18.39	1984-03-19	2019-07-27	37540	7239

שאלת 5: ניתוח ייצור חודשי עם מגמות עונתיות

Query 5: Monthly Production Analysis with Seasonal Trends

תיאור בעברית: שאלתה המנתחת את ייצור המטוסים לפי חודשים ושנים, כולל זיהוי מגמות עונתיות. השאלתה מציגה את כמות המטוסים שיוצרו בכל חודש, קיבולת ממוצעת, ורשימת היצרנים והדגמים שיוצרו באותה תקופה.

קוד השאלתה:

SELECT

```
EXTRACT(YEAR FROM p.ProductionDate) as Production_Year,  
EXTRACT(MONTH FROM p.ProductionDate) as Production_Month,  
TO_CHAR(p.ProductionDate, 'Month') as Month_Name,  
COUNT(*) as Planes_Produced,  
ROUND(AVG(p.Capacity), 2) as Avg_Capacity,  
STRING_AGG(DISTINCT pr.Pname, ', ' ORDER BY pr.Pname) as Producers,  
STRING_AGG(p.Model, ', ' ORDER BY p.Model) as Models_Produced
```

FROM Plane p

JOIN Producer pr ON p.Producer_id = pr.Producer_id

GROUP BY EXTRACT(YEAR FROM p.ProductionDate), EXTRACT(MONTH FROM
p.ProductionDate), TO_CHAR(p.ProductionDate, 'Month')

HAVING COUNT(*) >= 1

ORDER BY Production_Year DESC, Production_Month DESC;

The screenshot shows the pgAdmin 4 web interface in a browser. The SQL query is entered in the query editor and has been executed. The results are displayed in a table with 14 columns: production_year, production_month, month_name, planes_produced, avg_capacity, producers, and models_produced. The table shows 5 rows of data, sorted by production_year descending and production_month descending. The interface includes a sidebar with 'Servers' and 'Query History' tabs, and a bottom status bar showing 'Total rows: 400' and 'Query complete 00:00:00.209'.

production_year	production_month	month_name	planes_produced	avg_capacity	producers	models_produced
2023	12	December	2	266.00	AeroDynamics Enterprises, Pinnacle Aerospace	Beechcraft King Air, Embraer E190
2023	10	October	1	103.00	WingWorks	Airbus A350
2023	9	September	1	371.00	Liberty Aircraft Incorporated	Airbus A380
2023	8	August	1	237.00	AirWorks	Antonov An-124
2023	5	May	3	217.67	Global/Craft, Pegasus Aircraft LLC, SkyWorks	Bombardier CRJ, Canadair Regional Jet, Cessna Citation

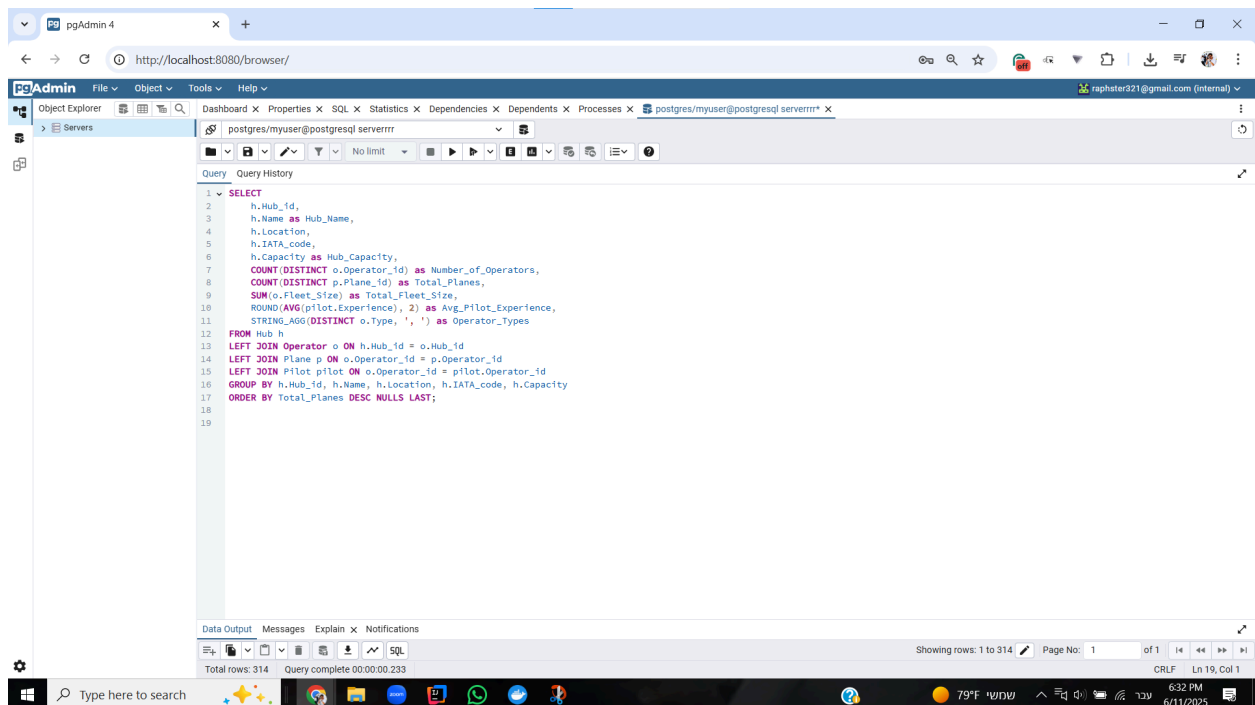
שאלתה 6: ניתוח ניצול מוקדים והתפלגות מפעילים

Query 6: Hub Utilization and Operator Distribution Analysis

תיאור בעברית: שאילתה המנתחת את רמת הניצול של מוקדי התעופה השונים, כולל מספר המפעילים הפועלים מכל מוקד, סה"כ מטוסים, ורמת ניסיון ממוצעת של הטייסים. מידע זה חיוני לתכנון קיבולת ואופטימיזציה של המוקדים.

קוד השאילתה:

```
SELECT
  h.Hub_id,
  h.Name as Hub_Name,
  h.Location,
  h.IATA_code,
  h.Capacity as Hub_Capacity,
  COUNT(DISTINCT o.Operator_id) as Number_of_Operators,
  COUNT(DISTINCT p.Plane_id) as Total_Planes,
  SUM(o.Fleet_Size) as Total_Fleet_Size,
  ROUND(AVG(pilot.Experience), 2) as Avg_Pilot_Experience,
  STRING_AGG(DISTINCT o.Type, ' ') as Operator_Types
FROM Hub h
LEFT JOIN Operator o ON h.Hub_id = o.Hub_id
LEFT JOIN Plane p ON o.Operator_id = p.Operator_id
LEFT JOIN Pilot pilot ON o.Operator_id = pilot.Operator_id
GROUP BY h.Hub_id, h.Name, h.Location, h.IATA_code, h.Capacity
ORDER BY Total_Planes DESC NULLS LAST;
```



	hub_id [PK] integer	hub_name character varying (50)	location character varying (80)	iata_code character varying (10)	hub_capacity integer	number_of_operators bigint	total_planes bigint	total_fleet_size bigint	avg_pilot_experience numeric	operator_types text
1	208	Continental Charlotte Airport	Charlotte	OMR	20220	5	10	5976	18.21	Commercial, Domestic, International, Mi...
2	245	United Minneapolis Terminal	Minneapolis	QSB	80197	3	9	3332	10.50	Cargo, Commercial
3	206	Northern Long Beach Airfield	Long Beach	WJR	41113	4	9	7732	18.80	International, Military, Specialized
4	349	Capital Mesa Aerodrome	Mesa	GWC	22857	3	8	2100	21.17	Cargo, Specialized
5	91	Municipal Albuquerque Aerodrome	Albuquerque	CHK	47856	4	7	1063	18.17	Domestic, Executive, Private

שאלתה 7: היסטוריית השמות מורכבת עם ניתוח זמני

Query 7: Complex Assignment History with Time-based Analysis

תיאור בעברית: שאלתה המנתחת את היסטוריית השמות הטיסים לאורך זמן, כולל פילוח לפי שנים ורבעונים. השאלתה מספקת סטטיסטיקות על מספר ההשמות, טייסים ומטוסים ייחודיים, ורמת ניסיון ממוצעת לתקופה.

קוד השאלתה:

```
SELECT
    EXTRACT(YEAR FROM pp.Assignment_date) as Assignment_Year,
    EXTRACT(QUARTER FROM pp.Assignment_date) as Assignment_Quarter,
    COUNT(*) as Total_Assignments,
    COUNT(DISTINCT pp.Pilot_id) as Unique_Pilots,
    COUNT(DISTINCT pp.Plane_id) as Unique_Planes,
    ROUND(AVG(pilot.Experience), 2) as Avg_Pilot_Experience,
    STRING_AGG(DISTINCT pilot.Rank, ', ') as Pilot_Ranks,
    STRING_AGG(DISTINCT p.Status, ', ') as Plane_Statuses
FROM Pilot_Plane pp
JOIN Pilot pilot ON pp.Pilot_id = pilot.Pilot_id
JOIN Plane p ON pp.Plane_id = p.Plane_id
WHERE pp.Assignment_date BETWEEN '2020-01-01' AND CURRENT_DATE
GROUP BY EXTRACT(YEAR FROM pp.Assignment_date), EXTRACT(QUARTER FROM
pp.Assignment_date)
ORDER BY Assignment_Year DESC, Assignment_Quarter DESC;
```

pgAdmin 4 interface showing a SQL query and its results.

Query:

```
SELECT
  EXTRACT(YEAR FROM pp.Assignment_date) as Assignment_Year,
  EXTRACT(QUARTER FROM pp.Assignment_date) as Assignment_Quarter,
  COUNT(*) as Total_Assignments,
  COUNT(DISTINCT pp.Pilot_id) as Unique_Pilots,
  COUNT(DISTINCT pp.Plane_id) as Unique_Planes,
  ROUND(AVG(pilot.Experience), 2) as Avg_Pilot_Experience,
  STRING_AGG(DISTINCT pilot.Rank, ' ') as Pilot_Ranks,
  STRING_AGG(DISTINCT p.Status, ' ') as Plane_Statuses
FROM Pilot_Plane pp
JOIN Pilot pilot ON pp.Pilot_id = pilot.Pilot_id
JOIN Plane p ON pp.Plane_id = p.Plane_id
WHERE pp.Assignment_date BETWEEN '2020-01-01' AND CURRENT_DATE
GROUP BY EXTRACT(YEAR FROM pp.Assignment_date), EXTRACT(QUARTER FROM pp.Assignment_date)
ORDER BY Assignment_Year DESC, Assignment_Quarter DESC;
```

Data Output:

hub_id	hub_name	location	data_code	hub_capacity	number_of_operators	total_planes	total_fleet_size	avg_pilot_experience	operator_types
1	Hub 1	Location 1	Code 1	100	5	10	100	10.00	Active, Emergency Ready, Grounded, Inactive, In Service, Leased, Maintenance, Reserved, Scheduled, Standby, Training
2	Hub 2	Location 2	Code 2	150	7	15	150	12.00	Active, Emergency Ready, Grounded, Inactive, In Service, Leased, Maintenance, Reserved, Scheduled, Standby, Training
3	Hub 3	Location 3	Code 3	200	10	20	200	15.00	Active, Emergency Ready, Grounded, Inactive, In Service, Leased, Maintenance, Reserved, Scheduled, Standby, Training
4	Hub 4	Location 4	Code 4	250	12	25	250	18.00	Active, Emergency Ready, Grounded, Inactive, In Service, Leased, Maintenance, Reserved, Scheduled, Standby, Training
5	Hub 5	Location 5	Code 5	300	15	30	300	20.00	Active, Emergency Ready, Grounded, Inactive, In Service, Leased, Maintenance, Reserved, Scheduled, Standby, Training

שאלתה 8: ניתוח צי מקיף לפי סוג מפעיל

Query 8: Comprehensive Fleet Analysis by Operator Type

תיאור בעברית: שאלתה מקיפה המנתחת את הצי לפי סוגי המפעילים השונים (מסחרי, מטען, וכו'). השאלתה מספקת תמונה כוללת על ביצועי הצי, גיל ממוצע, וסטטיסטיקות תפעוליות עבור כל סוג מפעיל.

קוד השאלתה:

```
SELECT
  o.Type as Operator_Type,
  COUNT(DISTINCT o.Operator_id) as Number_of_Operators,
  COUNT(DISTINCT p.Plane_id) as Total_Planes,
  COUNT(DISTINCT pilot.Pilot_id) as Total_Pilots,
  ROUND(AVG(p.Capacity), 2) as Avg_Plane_Capacity,
  ROUND(AVG(p.MaxAltitude), 0) as Avg_Max_Altitude,
  ROUND(AVG(pilot.Experience), 2) as Avg_Pilot_Experience,
  SUM(CASE WHEN p.Status = 'Operational' THEN 1 ELSE 0 END) as Operational_Count,
  STRING_AGG(DISTINCT h.IATA_code, ' ') as Hub_Codes,
  ROUND(AVG((CURRENT_DATE - p.ProductionDate) / 365.0), 2) as Avg_Fleet_Age
```

```

FROM Operator o
LEFT JOIN Plane p ON o.Operator_id = p.Operator_id
LEFT JOIN Pilot pilot ON o.Operator_id = pilot.Operator_id
LEFT JOIN Hub h ON o.Hub_id = h.Hub_id
GROUP BY o.Type
ORDER BY Total_Planes DESC NULLS LAST;

```

The screenshot shows the pgAdmin 4 web interface in a browser. The SQL query editor contains the following query:

```

1 SELECT
2   o.Type as Operator_Type,
3   COUNT(DISTINCT o.Operator_id) as Number_of_Operators,
4   COUNT(DISTINCT p.Plane_id) as Total_Planes,
5   COUNT(DISTINCT pilot.Pilot_id) as Total_Pilots,
6   ROUND(AVG(p.Capacity), 2) as Avg_Plane_Capacity,
7   ROUND(AVG(p.MaxAltitude), 0) as Avg_Max_Altitude,
8   ROUND(AVG(pilot.Experience), 2) as Avg_Pilot_Experience,
9   SUM(CASE WHEN p.Status = 'Operational' THEN 1 ELSE 0 END) as Operational_Count,
10  STRING_AGG(DISTINCT h.Hub_code, ',' ) as Hub_Codes,
11  ROUND(AVG(CURRENT_DATE - p.ProductionDate) / 365.0, 2) as Avg_Fleet_Age
12 FROM Operator o
13 LEFT JOIN Plane p ON o.Operator_id = p.Operator_id
14 LEFT JOIN Pilot pilot ON o.Operator_id = pilot.Operator_id
15 LEFT JOIN Hub h ON o.Hub_id = h.Hub_id
16 GROUP BY o.Type
17 ORDER BY Total_Planes DESC NULLS LAST;
18

```

The Data Output tab shows the results of the query. The first row is highlighted:

year	assignment_quarter	total_assignments	unique_pilots	unique_planes	avg_pilot_experience	pilot_ranks	plane_statuses
2023	4	20	19	20	17.60	Cadet, Chief Pilot, Commander, First Officer, Flight Engineer, Instructor Pilot, Second Officer, Senior Captain, Trainee	Active, Emergency Ready, Grounded

Total rows: 16. Query complete 00:00:00.099.

operator_type	number_of_operators	total_planes	total_pilots	avg_plane_capacity	avg_max_altitude	avg_pilot_experience	operational_count	hub_codes
Specialized	39	56	77	248.65	35166	17.33	0	BBW, BPH, BVI, CHP, DSE, ECC, EGO, EGG, ELR, GVN, GWC, IOJ, JTI, KAG, KHV, LSB, LYJ, MMM, MUR, ...
Commercial	38	53	79	180.65	32336	19.55	0	ATK, BOQ, BRK, DOQ, DVW, EVQ, FJO, HCW, HPP, JFM, JXD, MKY, MTK, OMR, OXJ, OXY, PAX, PIJ, PTS, ...
Executive	39	53	86	205.40	35506	19.86	0	BGF, BNM, CHK, COH, CRU, DRF, DYS, EGO, GEF, GVN, HOS, HOV, IKV, JBS, JNY, LDL, LIR, LKD, LOI, LVE, ...
Military	42	50	92	210.93	33272	18.78	0	ANM, AOF, BNG, BNM, BOA, BWZ, CVH, CYK, DSE, EKZ, FBL, FQZ, FWY, GEF, GJS, GYJ, HDM, HEJ, HQF, ...
Private	37	49	74	227.51	34458	16.94	0	ADE, AQL, CHK, DBG, FHI, GNL, GVN, JNV, KBA, KFS, KOE, LYJ, MKY, MTK, NOH, NUS, NWE, QEH, QVG, ...

3 DELETE / 3 DELETE Queries

שאלת מחיקה 1: הסרת השמות טייסים ישנות

Delete Query 1: Remove Old Pilot Assignments

תיאור בעברית: מחיקת השמות טייסים שבוצעו לפני יותר מ-3 שנים, כדי לשמור על נתונים לרונטיים בלבד ולשפר ביצועי המערכת.

קוד השאלתה:

```
DELETE FROM Pilot_Plane  
WHERE Assignment_date < CURRENT_DATE - INTERVAL '3 years';
```

The screenshot shows a SQL query editor with a toolbar at the top containing icons for file operations, filters, and execution. Below the toolbar, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying the following SQL code:

```
1  DELETE FROM Pilot_Plane  
2  WHERE Assignment_date < CURRENT_DATE - INTERVAL '3 years';  
3  
4
```

Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is active, showing the following output:

```
DELETE 1443  
  
Query returned successfully in 65 msec.
```

✓ Query returned successfully in

Query Query History

1 **SELECT** * **FROM** pilot_PLANE

Data Output Messages Notifications



Showing rows: 1 to 157 Page No: 1 of 1

	assignment_date date	plane_id [PK] integer	pilot_id [PK] integer
1	2023-09-20	325	740
2	2023-11-22	380	617
3	2023-03-18	230	763
4	2023-04-28	52	626
5	2023-12-20	384	756
6	2022-08-26	117	464
7	2023-01-14	175	141
8	2022-09-13	355	799
9	2023-09-15	388	536
10	2022-12-13	320	237
11	2023-08-08	383	60
12	2022-08-30		
13	2023-03-29		

✓ Successfully run. Total q
affected.

שאלת מחיקה 2: הסרת מטוסים ישנים בתחזוקה

Delete Query 2: Remove Old Planes in Maintenance

תיאור בעברית: מחיקת מטוסים שנמצאים בתחזוקה ובני יותר מ-10 שנים, כחלק מתהליך פרישה של ציוד ישן.

Query

Query History

1 SELECT * FROM PLANE

Data Output

Messages

Notifications

SQL

Showing rows: 1 to 500 Page No: 1 of 1

	er varying (25) <div></div>	productiondate date <div></div>	capacity integer <div></div>	maxaltitude integer <div></div>	maxdistance integer <div></div>	status character varying (30) <div></div>
1	/ An-124	2007-05-07	199	38816	7700	Reserved
2	am G650	1980-10-22	101	28297	3858	In Service
3	747	1983-11-14	397	34283	3059	Maintenance
4	raft King Air	2005-07-29	122	34491	7908	Scheduled
5	100	2023-04-15	108	37968	5983	Standby
6	/ An-124	2023-08-07	237	26981	2581	Training
7	raft King Air	1982-04-30	76	34684	1720	Testing
8	747	1992-09-30	239	33745	7216	Testing
9	raft King Air	1982-02-08	152	27280	1748	Inactive
10	747	2003-12-05	321	26401	2859	Inactive
11	r E190	2023-12-01	161	31936	9008	Standby
12	100	1994-09-19	346	37612	7538	Training

קוד השאילתה:

```
DELETE FROM Plane
WHERE Status = 'In Maintenance'
AND (CURRENT_DATE - ProductionDate) / 365.0 > 10;
```

Query Query History

Execute script

F5

```
1  ✓ DELETE FROM Plane
2  WHERE Status = 'In Maintenance'
3  AND (CURRENT_DATE - ProductionDate) / 365.0 > 10;
4
```

Data Output Messages Notifications

DELETE 0

Query returned successfully in 79 msec.

SQL X Statistics X Dependents X Processes X postgres/myuser@postgres server 3*

postgres/myuser@postgres server 3

No limit

Query Query History

1 SELECT * FROM PLANE

Data Output Messages Notifications

Showing rows: 1 to 500 Page No: 1 of 1

	iondate	capacity integer	maxaltitude integer	maxdistance integer	status character varying (30)	producer_id integer	hangar integer
1	5-07	199	38816	7700	Reserved	246	
2	0-22	101	28297	3858	In Service	180	
3	1-14	397	34283	3059	Maintenance	25	
4	7-29	122	34491	7908	Scheduled	169	
5	4-15	108	37968	5983	Standby	10	
6	8-07	237	26981	2581	Training	386	
7	4-30	76	34684	1720	Testing	100	
8	9-30	239	33745	7216	Testing	126	
9	2-08	152	27280	1748	Inactive	287	
10	2-05	321	26401	2859	Inactive	324	
11	2-01	161	31936	9008	Standby	264	
12	9-19	346	37612	7538	Training	300	

שאלת מחיקה 3: הסרת מפעילים ללא מטוסים

Delete Query 3: Remove Operators with No Planes

תיאור בעברית: מחיקת מפעילים שאין להם מטוסים מוקצים, כדי לנקות את מאגר הנתונים ממפעילים לא פעילים.


```
FROM Plane
WHERE Operator_id IS NOT NULL
);
\update or delete on table "operator" violates foreign key constraint
"pilot_operator_id_fkey" on table "pilot"
Key (operator_id)=(5) is still referenced from table "pilot".

SQL state: 23503
Detail: Key (operator_id)=(5) is still referenced from table "pilot".
```

3 שאלות UPDATE / 3 UPDATE Queries

שאלת עדכון 1: עדכון סטטוס מטוסים לפי גיל

Update Query 1: Update Plane Status Based on Age

כחלק מתהליך ניהול, "תיאור בעברית: עדכון סטטוס מטוסים לכל מטוס בן יותר מ-15 שנים כ"פרוש מחזור החיים של הצי.

postgres/myuser@postgres server 3

Query Query History

```
1 select * from plane
```

Data Output Messages Notifications

Showing rows: 1 to 500 of 1

	plane_id [PK] integer	model character varying (25)	productiondate date	capacity integer	maxaltitude integer	maxdistance integer	status
1	1	Antonov An-124	2007-05-07	199	38816	7700	R
2	2	Gulfstream G650	1980-10-22	101	28297	3858	li
3	3	Boeing 747	1983-11-14	397	34283	3059	M
4	4	Beechcraft King Air	2005-07-29	122	34491	7908	S
5	5	Fokker 100	2023-04-15	108	37968	5983	S
6	6	Antonov An-124	2023-08-07	237	26981	2581	T
7	7	Beechcraft King Air	1982-04-30	76	34684	1720	T
8	8	Boeing 747	1992-09-30	239	33745	7216	T
9	9	Beechcraft King Air	1982-02-08	152	27280	1748	li
10	10	Boeing 747	2003-12-05	321	26401	2859	li
11	11	Embraer E190	2023-12-01	161	31936	9008	S
12	12	Fokker 100	1994-09-19	346	37612	7538	T

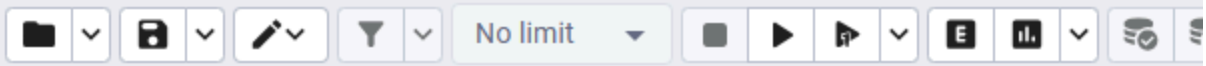
Total rows: 500 Query complete 00:00:00.078 CRLF Ln 1, Col 20

11:16 PM 6/24/2025

קוד השאילתה:

UPDATE Plane

```
SET Status = 'Retired'  
WHERE (CURRENT_DATE - ProductionDate) / 365.0 > 15  
AND Status != 'Retired';
```



Query Query History

```
1  UPDATE Plane
2  SET Status = 'Retired'
3  WHERE (CURRENT_DATE - ProductionDate) / 365.0 > 15
4  AND Status != 'Retired';
5
```

Data Output Messages Notifications

UPDATE 340

Query returned successfully in 71 msec.

תיאור בעברית: הגדלת שנת ניסיון לכל טייס שקיבל השמה בשנה האחרונה, כדי לעדכן את רמת הניסיון בהתאם לפעילות המתמשכת.

Insert screenshot of Pilot table before update] לפני העדכון Pilot **צילום מסך לפני העדכון:** [הכנס צילום מסך של טבלת

קוד השאילתה:

```
UPDATE Pilot
SET Experience = Experience + 1
WHERE Pilot_id IN (
    SELECT DISTINCT Pilot_id
    FROM Pilot_Plane
    WHERE Assignment_date >= CURRENT_DATE - INTERVAL '1 year'
);
```

Insert screenshot of execution message] / **צילום מסך של הרצת השאילתה:** [הכנס צילום מסך של הודעת ההצלחה

Insert screenshot of Pilot table after update] אחרי העדכון Pilot **צילום מסך אחרי העדכון:** [הכנס צילום מסך של טבלת

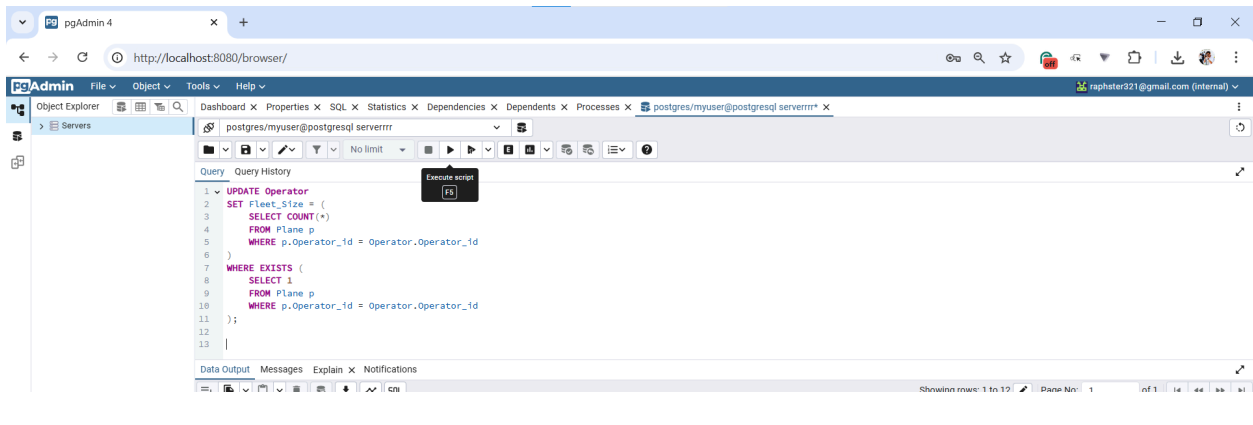
שאילתת עדכון 3: עדכון גודל צי לפי מספר מטוסים פעיל

Update Query 3: Update Fleet Size Based on Actual Plane Count

תיאור בעברית: עדכון גודל הצי של כל מפעיל בהתאם למספר המטוסים הפעיל שמוקצה לו, כדי לשמור על דיוק הנתונים.

קוד השאילתה:

```
UPDATE Operator
SET Fleet_Size = (
    SELECT COUNT(*)
    FROM Plane p
    WHERE p.Operator_id = Operator.Operator_id
)
WHERE EXISTS (
    SELECT 1
    FROM Plane p
    WHERE p.Operator_id = Operator.Operator_id
);
```



Database Constraints / אילוצי מאגר נתונים

אילוח 1: בדיקת קיבולת מטוס חוקית

Constraint 1: Valid Plane Capacity Check

להבטחת שקיבולת המטוס היא חיובית ומתחת ל-1000 Plane על טבלת CHECK **תיאור השינוי**: הוספת אילוח נוסעים, כדי למנוע ערכים לא הגיוניים.

ALTER TABLE פקודת:

```
ALTER TABLE Plane  
ADD CONSTRAINT chk_plane_capacity  
CHECK (Capacity > 0 AND Capacity <= 1000);
```

ניסיון הפרת האילוח:

```
INSERT INTO Plane VALUES (100, 'Test Model', '2023-01-01', -50, 35000, 3000, 'Operational',  
1, 1, 1);
```

SQL X Statistics X Dependents X Processes X postgres/myuser@postgres server 3 X

postgres/myuser@postgres server 3

No limit

Query Query History

```
1 INSERT INTO Plane VALUES (100, 'Test Model', '2023-01-01', -50, 35000, 3000, 'Op
2
```

Data Output Messages Notifications

ERROR: new row for relation "plane" violates check constraint "chk_plane_capacity"
Failing row contains (100, Test Model, 2023-01-01, -50, 35000, 3000, Operational, 1, 1, 1).

SQL state: 23514
Detail: Failing row contains (100, Test Model, 2023-01-01, -50, 35000, 3000, Operational, 1, 1, 1).

אילוח 2: בדיקת ניסיון טייס חוקי

Constraint 2: Valid Pilot Experience Check

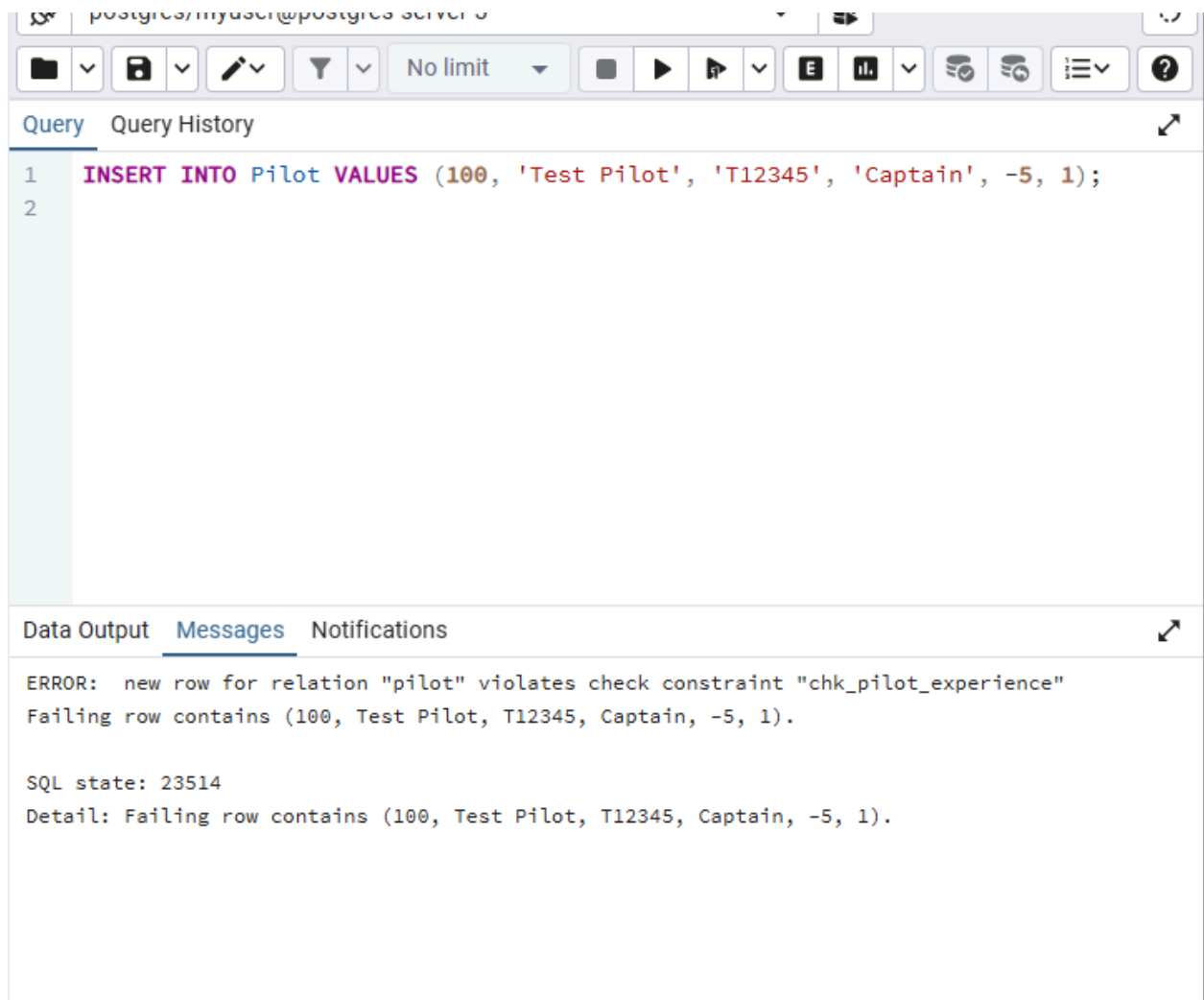
להבטחת שניסיון הטייס הוא בין 0 ל-50 שנים, כדי למנוע Pilot על טבלת CHECK **תיאור השינוי**: הוספת אילוץ ערכים שליליים או לא סבירים.

פקודת ALTER TABLE:

```
ALTER TABLE Pilot
ADD CONSTRAINT chk_pilot_experience
CHECK (Experience >= 0 AND Experience <= 50);
```

ניסיון הפרת האילוץ:

```
INSERT INTO Pilot VALUES (100, 'Test Pilot', 'T12345', 'Captain', -5, 1);
```



The screenshot shows a PostgreSQL query editor interface. The top toolbar includes icons for file operations, query execution, and settings. The 'Query' tab is active, displaying the following SQL statement:

```
1 INSERT INTO Pilot VALUES (100, 'Test Pilot', 'T12345', 'Captain', -5, 1);
2
```

Below the query editor, the 'Messages' tab is selected, showing an error message:

```
ERROR:  new row for relation "pilot" violates check constraint "chk_pilot_experience"
Failing row contains (100, Test Pilot, T12345, Captain, -5, 1).

SQL state: 23514
Detail: Failing row contains (100, Test Pilot, T12345, Captain, -5, 1).
```

אילוץ 3: ערך ברירת מחדל לסטטוס מטוס

Constraint 3: Default Value for Plane Status

כך שכל מטוס, Plane, בטבלת Status לעמודת "In Maintenance" תימלא אוטומטית. הוספת ערך ברירת מחדל
חדש יוכנס אוטומטית למצב תחזוקה עד לבדיקה.

פקודת ALTER TABLE:

```
ALTER TABLE Plane  
ALTER COLUMN Status SET DEFAULT 'In Maintenance';
```

בדיקת ערך ברירת מחדל:

```
INSERT INTO Plane (Plane_id, Model, ProductionDate, Capacity, MaxAltitude, MaxDistance,  
Producer_id, Hangar_id, Operator_id)  
VALUES (3200, 'Test Default', '2023-01-01', 23, 35000, 3000, 1, 1, 1);
```

```
SELECT * FROM Plane WHERE Plane_id = 200;
```

The screenshot shows a database management interface. At the top, there's a toolbar with various icons for file operations, filters, and execution. Below the toolbar, the 'Query' tab is active, displaying a SQL query: `SELECT * FROM Plane WHERE Plane_id = 2300;`. The 'Data Output' tab is also visible, showing the results of the query in a table format. The table has 8 columns: plane_id [PK] integer, model character varying (25), productiondate date, capacity integer, maxaltitude integer, maxdistance integer, and status charac. The first row of data shows plane_id 2300, model 'Test Default', productiondate '2023-01-01', capacity 23, maxaltitude 35000, maxdistance 3000, and status 'In Maintenance'.

	plane_id [PK] integer	model character varying (25)	productiondate date	capacity integer	maxaltitude integer	maxdistance integer	status charac
1	2300	Test Default	2023-01-01	23	35000	3000	In Maintenance

ROLLBACK ו-COMMIT / ROLLBACK and COMMIT

Examples

ביטול שינויים - ROLLBACK דוגמת

:שלב 1 - מצב מאגר הנתונים לפני עדכון

```
SELECT * FROM Pilot;
```

```
SELECT Plane_id, Model, Status FROM Plane;
```

Query Query History

```
1 SELECT * FROM Pilot;
2 SELECT Plane_id, Model, Status FROM Plane;
3
```

Data Output Messages Notifications

Showing rows: 1 to 501 Page No: 1 of 1

	plane_id [PK] integer	model character varying (25)	status character varying (30)
1	2300	Test Default	In Maintenance
2	5	Fokker 100	Standby
3	6	Antonov An-124	Training
4	11	Embraer E190	Standby
5	13	Airbus A380	Grounded
6	14	Beechcraft King Air	Scheduled
7	16	Boeing 737	Scheduled
8	17	Cessna Citation	Reserved
9	19	Boeing 777	Scheduled
10	23	Boeing 747	Testing
11	31	Dash 8	Active
12	33	Beechcraft Ki	

Successfully run. Total query runtime: 74. שלב 2 - ביצוע

שינויים בטרנזקציה:

```
BEGIN;
UPDATE Pilot SET Experience = Experience + 5 WHERE Pilot_id = 1;
UPDATE Plane SET Status = 'Retired' WHERE Plane_id = 1;
INSERT INTO Pilot VALUES (10, 'Test Pilot Rollback', 'TR12345', 'Captain', 15, 1);
```

```
Query History
BEGIN;
UPDATE Pilot SET Experience = Experience + 5 WHERE Pil
UPDATE Plane SET Status = 'Retired' WHERE Plane_id = 1
INSERT INTO Pilot VALUES (1000, 'Test Pilot Rollback',

Data Output Messages Notifications
INSERT 0 1

Query returned successfully in 53 msec.
```

שלב 4 - ביצוע

ROLLBACK:

ROLLBACK;

Query Query History

1 **ROLLBACK;**

Data Output **Messages** Notifications

ROLLBACK

Query returned successfully in 43 msec.

אישור שינויים - COMMIT דוגמת

שלב 1 - מצב מאגר הנתונים לפני עדכון

```
SELECT * FROM Hub;  
SELECT Operator_id, Name, Fleet_Size FROM Operator;
```

```
SELECT * FROM Hub;
SELECT Operator_id, Name, Fleet_Size FROM Operator;
```

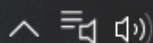


ing rows: 1 to 400 Page No: 1 of 1

operator_id [PK] integer	name character varying (50)	fleet_size integer
5	American Sky 5	199
6	Pacific Airways 6	435
10	Continental Airways 10	199
13	Asian Aviation 13	157
15	Global Star 15	318
18	European Star 18	422
19	United Sky 19	348
23	National Airways 23	13
25	American Sky 25	31
27	United Aviation 27	131
31	Atlantic Star 31	105
32	Pacific Aviat	
38	African Sky 3	

✓ Successfully run. Total query runtime: 57 msec. 400 rows affected.

al rows: 400 Query complete 00:00:00.057 CRLF Ln 3, Col 1



ENG

11:24 PM
6/24/2025



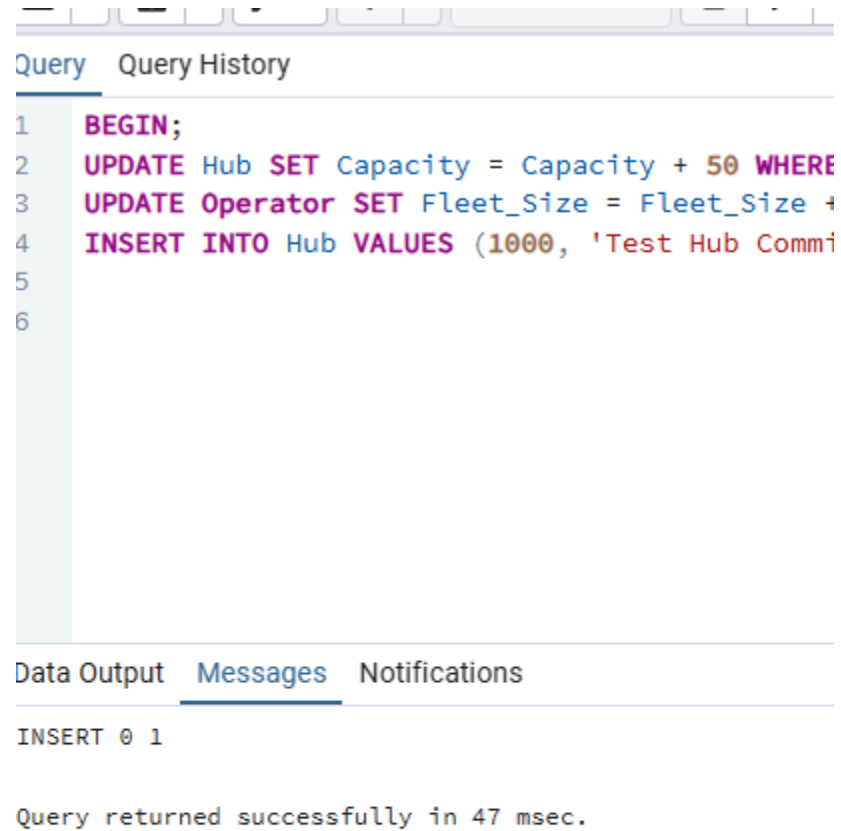
שלב 2 - ביצוע שינויים בטרנזקציה

BEGIN;

UPDATE Hub SET Capacity = Capacity + 50 WHERE Hub_id = 1;

UPDATE Operator SET Fleet_Size = Fleet_Size + 10 WHERE Operator_id = 1;

INSERT INTO Hub VALUES (10, 'Test Hub Commit', 'Test Location', 'TST', 150);



The screenshot shows a SQL query editor with a tab labeled 'Query' and 'Query History'. The query is as follows:

```
1 BEGIN;  
2 UPDATE Hub SET Capacity = Capacity + 50 WHERE  
3 UPDATE Operator SET Fleet_Size = Fleet_Size +  
4 INSERT INTO Hub VALUES (1000, 'Test Hub Commi  
5  
6
```

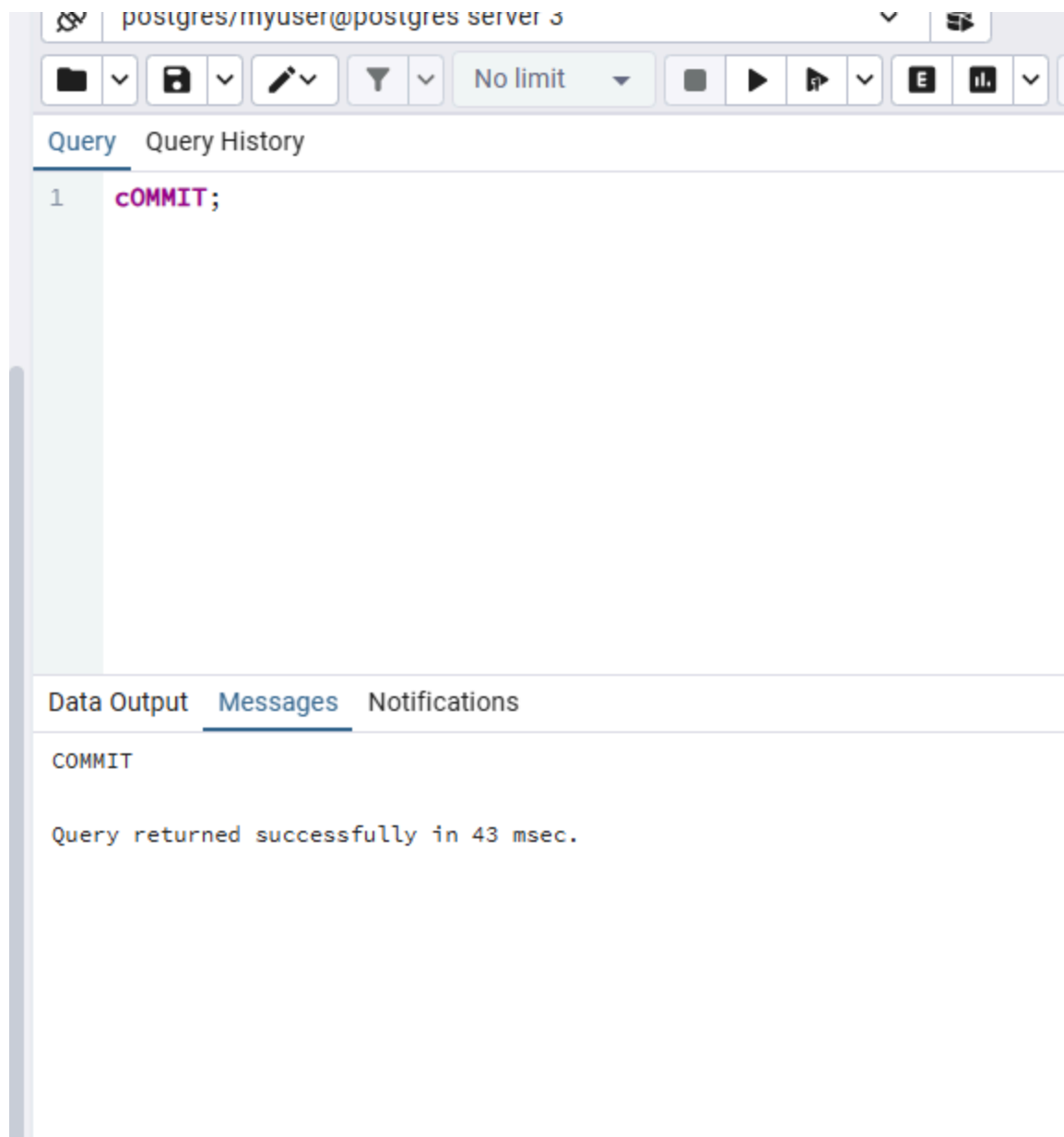
Below the query editor, there are three tabs: 'Data Output', 'Messages', and 'Notifications'. The 'Messages' tab is selected, showing the following output:

```
INSERT 0 1
```

Below the messages, it says: 'Query returned successfully in 47 msec.'

COMMIT: שלב 4 - ביצוע

COMMIT;



סיכום והתרשמות / Summary and Conclusions

השגים עיקריים / Main Achievements:

✓ פונקציות, JOINS, GROUP BY, מורכבות: פותחו 8 שאלות מתקדמות המשלבות SELECT שאלות תאריך ואגרגציה לצורך ניתוח מעמיק של נתוני התעופה.

✓ לתחזוקה ועדכון מאגר הנתונים UPDATE ו-3 שאלות DELETE ניהול נתונים: מומשו 3 שאלות.

✓ להבטחת איכות הנתונים ומניעת (CHECK, DEFAULT) **אילוץ אינטגרטה**: הוספו 3 סוגי אילוצים שונים ערכים לא חוקיים.

✓ להבטחת עקביות הנתונים ובקרה על שינויים COMMIT-ו ROLLBACK **בקרת טרנזקציות**: הודגמו פעולות.

Business Value / ערך עסקי:

השאלות שפותחו מספקות תובנות חיוניות לניהול מערכת התעופה, כולל:

- ניתוח ביצועי צי מטוסים
- מעקב אחר השמות וניסיון טייסים
- אופטימיזציה של ניצול מוקדי תעופה
- ניהול מחזור חיי הציוד

Technologies Used / טכנולוגיות בשימוש:

- **PostgreSQL** - מערכת ניהול מאגר נתונים
- **מתקדם** - שאלות מורכבות עם פונקציות תאריך ואגרציה **SQL**
- **Transaction Control** - ניהול עקביות נתונים
- **Data Integrity** - אילוץ מאגר נתונים

המערכת מוכנה לשילוב בממשק משתמש גרפי ומספקת בסיס איתן לפיתוח יישום ניהול תעופה מקיף.
