**פרויקט- אנליזה של ביג דאטה**

מאגר מידע: <https://www.kaggle.com/datasets/dilwong/flightprices/data>

מאגר המידע מכיל 27 עמודות, כ82 מיליון שורות בנפח כולל של כ31GB.  
כל שורה במאגר מתאר חיפוש טיסה באתר Expedia, עבור כל טיסה מתועד ID יחודי לטיסה, תאריך חיפוש, תאריך הטיסה, מוצא, יעד, זמן טיסה, מרחק טיסה, ימי טיסה, מידע אודות תעריף ותוספות לכרטיס, האם הטיסה ישירה, תעריף בסיס, מחיר מלא, מס' המושבים שנותרו בטיסה זו, חברת תעופה, מחלקה, סוג מטוס ועוד.

על מנת להכניס את הדאטה לDUCKDB השתמשנו ביכולות של DUCKDB, מפאת גודל הדאטה, הקובץ נוצר רק אם הוא לא קיים, שכן מדובר בהמון זמן ריצה.

הקוד להלן:

if not file\_exists("database.duckdb"):  
 duckdb\_conn = duckdb.connect("database.duckdb")  
 duckdb\_conn.execute("CREATE TABLE main AS SELECT \* FROM read\_csv\_auto('itineraries.csv')")  
else:  
 duckdb\_conn = duckdb.connect("database.duckdb")

השאלות שנשאל על הדאטה והשאילתות SQL שנבצע עבורן:

1. כיצד משתנים מחירי הטיסות ביחס לזמן החיפוש מול מועד הטיסה?

duckdb\_conn.execute("""  
 CREATE TABLE query1 AS  
 WITH time\_ranges AS (  
 SELECT \*,  
 (flightDate - searchDate) as days\_until\_flight  
 FROM main  
 )  
 SELECT   
 days\_until\_flight as days\_before\_flight,  
 ROUND(AVG(totalFare), 2) as avg\_fare,  
 COUNT(\*) as number\_of\_searches,  
 ROUND(MIN(totalFare), 2) as min\_fare,  
 ROUND(MAX(totalFare), 2) as max\_fare  
 FROM time\_ranges  
 GROUP BY days\_until\_flight  
 ORDER BY days\_until\_flight  
""")

1. האם משתלם יותר לטוס בטיסה ישירה או בטיסה עם עצירות (קונקשן) בהתחשב במרחקי טיסה שונים?

duckdb\_conn.execute("""  
 CREATE TABLE query2 AS  
 WITH flight\_categories AS (  
 SELECT \*,  
 CASE   
 WHEN totalTravelDistance <= 500 THEN 1  
 WHEN totalTravelDistance <= 1000 THEN 2  
 WHEN totalTravelDistance <= 1500 THEN 3  
 WHEN totalTravelDistance <= 2000 THEN 4  
 ELSE 5  
 END as distance\_category  
 FROM main  
 WHERE totalTravelDistance IS NOT NULL  
 )  
 SELECT   
 CASE distance\_category  
 WHEN 1 THEN 'Short (0-500 miles)'  
 WHEN 2 THEN 'Medium (501-1000 miles)'  
 WHEN 3 THEN 'Long (1001-1500 miles)'  
 WHEN 4 THEN 'Very Long (1501-2000 miles)'  
 ELSE 'Ultra Long (2000+ miles)'  
 END as flight\_distance,  
 isNonStop,  
 ROUND(AVG(totalFare), 2) as avg\_fare,  
 ROUND(AVG(travelDuration), 0) as avg\_duration\_minutes,  
 COUNT(\*) as number\_of\_flights  
 FROM flight\_categories  
 GROUP BY distance\_category, isNonStop  
 ORDER BY distance\_category, isNonStop  
""")

1. איך שעת ההמראה משפיעה על המחיר, הזמינות ואופי הטיסה (ישירה\לא ישירה)?

duckdb\_conn.execute("""  
 CREATE TABLE query3 AS  
 WITH flight\_times AS (  
 SELECT \*,  
 CAST(SUBSTR(segmentsDepartureTimeRaw, 12, 2) AS INTEGER) as departure\_hour  
 FROM main  
 WHERE segmentsDepartureTimeRaw IS NOT NULL  
 )  
 SELECT   
 departure\_hour,  
 ROUND(AVG(totalFare), 2) as avg\_fare,  
 ROUND(AVG(seatsRemaining), 1) as avg\_seats\_remaining,  
 COUNT(\*) as number\_of\_flights,  
 ROUND(MIN(totalFare), 2) as min\_fare,  
 ROUND(MAX(totalFare), 2) as max\_fare,  
 SUM(isNonStop::INTEGER) \* 100.0 / COUNT(\*) as nonstop\_percentage  
 FROM flight\_times  
 GROUP BY departure\_hour  
 ORDER BY departure\_hour  
""")

1. מהם ימי השבוע היקרים והזולים ביותר לטיסה והאם יש הבדל בין טיסות ישירות לטיסות עם עצירות?

duckdb\_conn.execute("""  
 CREATE TABLE query4 AS  
 WITH day\_info AS (  
 SELECT   
 totalFare,  
 TO\_CHAR(flightDate, 'Day') as flight\_day,  
 isNonStop  
 FROM main  
 )  
 SELECT   
 TRIM(flight\_day) as flight\_day,  
 ROUND(AVG(totalFare), 2) as avg\_fare,  
 ROUND(AVG(CASE WHEN isNonStop THEN totalFare END), 2) as avg\_nonstop\_fare,  
 ROUND(AVG(CASE WHEN NOT isNonStop THEN totalFare END), 2) as avg\_connection\_fare,  
 COUNT(\*) as number\_of\_flights  
 FROM day\_info  
 GROUP BY flight\_day  
 ORDER BY EXTRACT(DOW FROM DATE\_TRUNC('day', MIN(flightDate)))  
""")

1. מהם הנתיבים (מוצא ויעד) של טיסות להם היו שינויי מחיר קיצוניים (מעל 20%) כלפי מעלה וכלפי מטה?

duckdb\_conn.execute("""  
 CREATE TABLE query5 AS  
 WITH price\_volatility AS (  
 SELECT   
 startingAirport,  
 destinationAirport,  
 flightDate,  
 AVG(totalFare) as avg\_fare,  
 LAG(AVG(totalFare)) OVER (  
 PARTITION BY startingAirport, destinationAirport  
 ORDER BY flightDate  
 ) as prev\_day\_fare  
 FROM main  
 GROUP BY startingAirport, destinationAirport, flightDate  
 )  
 SELECT   
 flightDate,  
 startingAirport,  
 destinationAirport,  
 avg\_fare,  
 prev\_day\_fare,  
 ((avg\_fare - prev\_day\_fare) / prev\_day\_fare \* 100) as daily\_change\_percent  
 FROM price\_volatility  
 WHERE ((avg\_fare - prev\_day\_fare) / prev\_day\_fare \* 100) > 20  
 ORDER BY daily\_change\_percent DESC;  
""")

בכדי לייצא את הדאטה מduckDB לsqlite חיברנו את הsqlite לduckDB, פתחנו קובץ חדש בפורמט המתאים ואליו העתקנו את הטבלאות אחת אחת

duckdb\_conn.execute("INSTALL sqlite;")  
duckdb\_conn.execute("LOAD sqlite;")  
duckdb\_conn.execute("ATTACH 'database.sqlite' AS sqliteDB (TYPE SQLITE);")  
  
duckdb\_conn.execute("CREATE TABLE sqliteDB.sample AS SELECT \* FROM sample")  
duckdb\_conn.execute("CREATE TABLE sqliteDB.query1 AS SELECT \* FROM query1")  
duckdb\_conn.execute("CREATE TABLE sqliteDB.query2 AS SELECT \* FROM query2")  
duckdb\_conn.execute("CREATE TABLE sqliteDB.query3 AS SELECT \* FROM query3")  
duckdb\_conn.execute("CREATE TABLE sqliteDB.query4 AS SELECT \* FROM query4")  
duckdb\_conn.execute("CREATE TABLE sqliteDB.query5 AS SELECT \* FROM query5")