SQL Querying

Authors: Ayman EL ALASS & Abderaouf KHELFAOUI

Objective: In this notebook, we perform analytical queries on our relational database project_database.db . We focus on **raw data extraction and aggregation** to derive relevent insights directly from the query results.

Exemples

1. **Basic Retrieval:** Extracting specific high-delay incidents.
2. **Aggregation:** Ranking airlines by punctuality and reliability.
3. **Complex Analysis:** Identifying the "Black List" of flight routes using double joins.
4. **Cross-Domain Analysis:** Correlating weather conditions with flight cancellations.
5. **Database Evolution:** Modifying the schema to permanently store delay categories.
6. **Global Statistics:** Analyzing total flight volume and the busiest day of the year.
7. **Geographic Insights:** Identifying busiest airports and state-level traffic peaks.
8. **Delay & Cancellation Deep Dive:** Pinpointing the worst airline/airport combinations and cancellation hotspots.

In [1]:
```python
import sqlite3
import pandas as pd

# Connect to the database generated by main.py
db_name = "project_database.db"
conn = sqlite3.connect(db_name)
print(f"Connected to database: {db_name}")

# Helper function to execute SQL and return a readable DataFrame
def run_query(query):
    try:
        df = pd.read_sql(query, conn)
        return df
    except Exception as e:
        print(f"SQL Error: {e}")
```

Connected to database: project_database.db

1. Basic Data Retrieval (Sanity Check)

Scenario: A specific request from the Operations Center. We need to list details of flights operated by 'American Airlines Inc.' that faced extreme delays (> 4 hours) to identify patterns in specific airports.

In [2]:
```python
query_1 = """
SELECT
    f.flight_date,
    f.flight_number,
    f.origin_airport,
    f.dest_airport,
    f.dep_delay || ' min' as delay_minutes -- Formatting for readability
FROM FLIGHTS f
JOIN AIRLINES a ON f.airline_code = a.airline_code
WHERE a.airline_name = 'American Airlines Inc.'
  AND f.dep_delay > 240
ORDER BY f.dep_delay DESC
LIMIT 10;
"""

print(">>> Extreme Delays Report (American Airlines):")
display(run_query(query_1))
```

	flight_date	flight_number	origin_airport	dest_airport	delay_minutes
0	2015-01-18	224	LAS	LAX	1604 min
1	2015-03-10	1594	SAT	DFW	1557 min
2	2015-11-15	290	MCO	JFK	1536 min
3	2015-09-26	1293	STX	MIA	1471 min
4	2015-02-22	1080	EGE	ORD	1457 min
5	2015-02-19	1564	PHX	ORD	1367 min
6	2015-01-04	1279	OMA	DFW	1255 min
7	2015-05-03	2263	CMH	DFW	1181 min
8	2015-03-02	162	HNL	LAX	1142 min
9	2015-04-28	395	SJU	ORD	1120 min

2. Airline Performance Audit

Exemple Question: Which airlines are the most reliable? We calculate three key KPIs per airline:

1. **Volume:** Total flights.
2. **Punctuality:** Average departure delay.
3. **Reliability:** Cancellation Rate (%).

In [3]:
```
query_2 = """
SELECT
    a.airline_name,
    COUNT(f.flight_id) as total_flights,
    ROUND(AVG(f.dep_delay), 2) as avg_delay_min,
    SUM(CASE WHEN f.cancelled = 1 THEN 1 ELSE 0 END) as cancelled_count,
    ROUND(
        (CAST(SUM(CASE WHEN f.cancelled = 1 THEN 1 ELSE 0 END) as FLOAT) / COUNT(f.flight_id)) * 100,
    2) || '%' as cancellation_rate
FROM FLIGHTS f
JOIN AIRLINES a ON f.airline_code = a.airline_code
GROUP BY a.airline_name
HAVING total_flights > 500 -- Filter to keep only major airlines
ORDER BY avg_delay_min ASC;
"""

print(">>> Airline Performance Matrix (Ranked by Punctuality):")
display(run_query(query_2))
```

	airline_name	total_flights	avg_delay_min	cancelled_count	cancellation_rate
0	Hawaiian Airlines Inc.	7042	0.20	12	0.17%
1	Alaska Airlines Inc.	15725	1.91	47	0.3%
2	US Airways Inc.	19894	6.22	408	2.05%
3	Delta Air Lines Inc.	80221	7.77	373	0.46%
4	Skywest Airlines Inc.	54214	8.25	936	1.73%
5	Atlantic Southeast Airlines	52500	8.93	1476	2.81%
6	American Airlines Inc.	64636	8.97	997	1.54%
7	Virgin America	5448	9.24	47	0.86%
8	American Eagle Airlines Inc.	27223	10.44	1445	5.31%
9	Southwest Airlines Co.	115771	10.91	1590	1.37%
10	JetBlue Airways	24230	11.66	408	1.68%
11	Frontier Airlines Inc.	8339	14.78	59	0.71%
12	United Air Lines Inc.	47131	15.02	688	1.46%
13	Spirit Air Lines	10674	16.72	211	1.98%

3. Route Analysis (Double Join)

Context: We want to identify the specific City-to-City connections that suffer from the worst delays.

Technique: We perform a **Double Join** on the AIRPORTS table (aliased as origin and dest) to retrieve readable city names instead of IATA codes.

In [4]:
```
query_3 = """
SELECT
    origin.city || ' -> ' || dest.city AS Route,
    COUNT(*) as Flight_Count,
    ROUND(AVG(f.dep_delay), 2) as Avg_Delay_Min,
    MAX(f.dep_delay) as Max_Delay_Min
FROM FLIGHTS f
JOIN AIRPORTS origin ON f.origin_airport = origin.iata_code
JOIN AIRPORTS dest ON f.dest_airport = dest.iata_code
GROUP BY f.origin_airport, f.dest_airport
HAVING Flight_Count > 20 -- Ignore rare charter routes
ORDER BY Avg_Delay_Min DESC
LIMIT 10;
"""

print(">>> Top 10 Routes with Highest Average Delays:")
display(run_query(query_3))
```

	Route	Flight_Count	Avg_Delay_Min	Max_Delay_Min
0	Myrtle Beach -> Philadelphia	23	47.22	276
1	Newark -> Portland	34	47.00	218
2	Chicago -> Honolulu	42	45.29	602
3	Denver -> Honolulu	22	45.18	350
4	Agana -> Honolulu	28	43.29	994
5	Honolulu -> New York	33	43.27	1433
6	Charlottesville -> Chicago	77	41.17	576
7	Dallas-Fort Worth -> Kahului	47	40.28	573
8	Charlotte Amalie -> Philadelphia	28	39.39	493
9	Tampa -> Cleveland	52	39.04	548

4. Exemple of Weather Impact Querying

Context: For instance we can assume that to avoid heavy processing times, we analyze the correlation between wind and delays for a **single specific day** (January 1st, 2015).

Technique: We aggregate the average wind speed and average delay per airport for that day.

In [5]:
```
query_4 = """
WITH DailyWeather AS (
    SELECT
        airport_code,
        AVG(wind_speed) as avg_wind_speed
    FROM WEATHER
    WHERE date(reading_time) = '2015-01-01'
    GROUP BY airport_code
),
DailyFlights AS (
    SELECT
        origin_airport,
        AVG(dep_delay) as avg_dep_delay
    FROM FLIGHTS
    WHERE date(flight_date) = '2015-01-01'
    GROUP BY origin_airport
)
SELECT
    f.origin_airport,
    f.avg_dep_delay,
    w.avg_wind_speed
FROM DailyFlights f
JOIN DailyWeather w ON f.origin_airport = w.airport_code
ORDER BY f.avg_dep_delay DESC;
"""

df_result = run_query(query_4)
display(df_result)
```

	origin_airport	avg_dep_delay	avg_wind_speed
0	DEN	23.526316	1.375000
1	DFW	23.271605	2.416667
2	MIA	15.315789	2.041667
3	SFO	15.038462	2.708333
4	LAX	14.058824	0.916667
5	SEA	13.500000	1.291667
6	IAH	13.243902	3.291667
7	ORD	11.827586	8.833333
8	PHX	10.314286	1.333333
9	BOS	7.133333	5.666667
10	JFK	7.076923	3.166667
11	MSP	4.500000	4.791667
12	ATL	3.000000	1.041667
13	DTW	2.318182	7.625000
14	PHL	-0.666667	2.666667

5. Database Evolution

Requirement: To optimize future reporting, we need to persist the "Delay Category" directly in the database table, rather than calculating it every time.

Actions:

1. **ALTER TABLE:** Add a new column `delay_category` .
2. **UPDATE:** Populate this column based on the `dep_delay` value.

In [6]:

```python
# 1. Add the column structure
try:
    conn.execute("ALTER TABLE FLIGHTS ADD COLUMN delay_category VARCHAR(20)")
    print("Schema Altered: Column 'delay_category' added.")
except sqlite3.OperationalError:
    print("Column 'delay_category' already exists.")

# 2. Populate the data
update_query = """
UPDATE FLIGHTS
SET delay_category = CASE
    WHEN dep_delay <= 0 THEN 'On Time / Early'
    WHEN dep_delay > 0 AND dep_delay <= 15 THEN 'Small Delay'
    WHEN dep_delay > 15 AND dep_delay <= 45 THEN 'Medium Delay'
    ELSE 'Major Delay (>45m)'
END;
"""
conn.execute(update_query)
conn.commit()
print("Data Updated: Categories populated.")

# 3. Verification Query
query_check = """
SELECT
    delay_category,
    COUNT(*) as flight_count,
    ROUND((CAST(COUNT(*) as FLOAT) / (SELECT COUNT(*) FROM FLIGHTS)) * 100, 1) || '%' as proportion
FROM FLIGHTS
GROUP BY delay_category
ORDER BY flight_count DESC;
"""

print(">>> Verification: Distribution of new categories:")
display(run_query(query_check))
```

```
Column 'delay_category' already exists.
Data Updated: Categories populated.
>>> Verification: Distribution of new categories:
```

	delay_category	flight_count	proportion
0	On Time / Early	334940	62.8%
1	Small Delay	101428	19.0%
2	Medium Delay	54293	10.2%
3	Major Delay (>45m)	42387	8.0%

6. Global Statistics

In [7]:
```python
# Total flights in 2015
query_total = "SELECT COUNT(flight_id) as 'Total Flights 2015' FROM flights;"
display(run_query(query_total))

# Busiest day of the year
query_busiest_day = """
SELECT flight_date, COUNT(flight_id) as number_of_flights
FROM flights
WHERE strftime('%Y', flight_date) = '2015'
GROUP BY flight_date
ORDER BY number_of_flights DESC
LIMIT 1;
"""
print(">>> Busiest day of 2015:")
display(run_query(query_busiest_day))
```

	Total Flights 2015
0	533048

>>> Busiest day of 2015:

	flight_date	number_of_flights
0	2015-07-16	1822

7. Airport & Location Analysis

In [8]:
```python
# Busiest Airport (Origin)
query_busiest_airport = """
SELECT f.origin_airport, a.airport_name, COUNT(f.flight_id) AS number_of_flights
FROM flights f
JOIN airports a ON f.origin_airport = a.iata_code
WHERE strftime('%Y', f.flight_date) = '2015'
GROUP BY f.origin_airport
ORDER BY number_of_flights DESC
LIMIT 1;
"""

print(">>> Busiest Airport:")
display(run_query(query_busiest_airport))


# Peak traffic day per State (Complex Sub-query)
query_state_peak = """
WITH DailyStats AS (
    SELECT
        a.state,
        f.origin_airport,
        a.airport_name,
        f.flight_date,
        COUNT(f.flight_id) AS number_of_flights,
        RANK() OVER (
            PARTITION BY a.state
            ORDER BY COUNT(f.flight_id) DESC
        ) as rang
    FROM flights f
    JOIN airports a ON f.origin_airport = a.iata_code
    WHERE f.flight_date BETWEEN '2015-01-01' AND '2015-12-31'
    GROUP BY a.state, f.origin_airport, a.airport_name, f.flight_date
)
SELECT
    state,
    origin_airport,
    airport_name,
    flight_date,
    number_of_flights
FROM DailyStats
WHERE rang = 1
ORDER BY state;
"""

print(">>> Peak Traffic Day per State (Optimized):")
display(run_query(query_state_peak))
```

>>> Busiest Airport:

	origin_airport	airport_name	number_of_flights
0	ATL	Hartsfield-Jackson Atlanta International Airport	34594

>>> Peak Traffic Day per State (Optimized):

	state	origin_airport	airport_name	flight_date	number_of_flights
0	AK	ANC	Ted Stevens Anchorage International Airport	2015-07-22	12
1	AK	ANC	Ted Stevens Anchorage International Airport	2015-08-06	12
2	AL	BHM	Birmingham-Shuttlesworth International Airport	2015-11-25	11
3	AL	BHM	Birmingham-Shuttlesworth International Airport	2015-12-27	11
4	AR	LIT	Bill and Hillary Clinton National Airport (Ada...	2015-08-21	9
...	...	...	...	...	...
133	WY	JAC	Jackson Hole Airport	2015-07-15	4
134	WY	JAC	Jackson Hole Airport	2015-08-06	4
135	WY	JAC	Jackson Hole Airport	2015-12-20	4
136	WY	JAC	Jackson Hole Airport	2015-12-22	4
137	WY	JAC	Jackson Hole Airport	2015-12-29	4

138 rows × 5 columns

8. Delays and Cancellations

In [9]:
```python
# Average Delay per Airline per Airport
query_delay_airline_airport = """
SELECT ar.airline_name, f.origin_airport, ROUND(AVG(f.dep_delay), 2) AS avg_dep_delay
FROM flights f
JOIN airports ap ON f.origin_airport = ap.iata_code
JOIN airlines ar ON f.airline_code = ar.airline_code
WHERE strftime('%Y', f.flight_date) = '2015'
GROUP BY ar.airline_name, f.origin_airport
ORDER BY avg_dep_delay DESC
LIMIT 10;
"""
print(">>> Top 10 Highest Average Delays (Airline/Airport):")
display(run_query(query_delay_airline_airport))

# Top 3 Airports with most cancellations
query_cancelled_airports = """
SELECT origin_airport, airport_name, COUNT(cancelled) AS cancelled_count
FROM airports ap
JOIN flights f ON ap.iata_code = f.origin_airport
WHERE cancelled = 1
GROUP BY origin_airport, airport_name
ORDER BY cancelled_count DESC
LIMIT 3;
"""
print(">>> Top 3 Airports for Cancellations:")
display(run_query(query_cancelled_airports))
```

```
>>> Top 10 Highest Average Delays (Airline/Airport):
```

	airline_name	origin_airport	avg_dep_delay
0	United Air Lines Inc.	CVG	905.00
1	Skywest Airlines Inc.	MQT	118.50
2	Frontier Airlines Inc.	MDW	101.00
3	Skywest Airlines Inc.	JNU	94.30
4	US Airways Inc.	TUS	81.00
5	Frontier Airlines Inc.	BMI	72.00
6	United Air Lines Inc.	FAI	70.11
7	Skywest Airlines Inc.	DAL	68.00
8	Alaska Airlines Inc.	GST	63.33
9	Alaska Airlines Inc.	JFK	62.69

```
>>> Top 3 Airports for Cancellations:
```

	origin_airport	airport_name	cancelled_count
0	ORD	Chicago O'Hare International Airport	825
1	DFW	Dallas/Fort Worth International Airport	613
2	LGA	LaGuardia Airport (Marine Air Terminal)	480