**Goal:**  
Monitor a real-time stream of **parking slot sensors** in an **airport**.  
For every **terminal (A, B, C)** and **zone (P1–P5)**, sensors stream **slot usage** every second.  
You must:

1. Detect **zone congestion** (if >80% slots are occupied in a zone).
2. Track **average occupancy** over a 1-minute window.
3. Trigger **alerts** when a terminal hit >85% utilization continuously for 15 seconds.

**Stream Schema Design**

Simulated sensor event will include:

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| timestamp | timestamp | Sensor timestamp |
| terminal | string | 'A', 'B', 'C' |
| zone | string | 'P1' to 'P5' |
| slot\_id | integer | Unique slot ID |
| status | string | 'Occupied' or 'Available' |

parking\_df = stream\_df.withColumn("terminal", expr("CASE WHEN value % 3 = 0 THEN 'A' WHEN value % 3 = 1 THEN 'B' ELSE 'C' END")) \

.withColumn("zone", expr("concat('P', (value % 5) + 1)")) \

.withColumn("slot\_id", (col("value") % 500) + 1) \

.withColumn("status", expr("CASE WHEN value % 4 = 0 THEN 'Occupied' ELSE 'Available' END")) \

.withColumn("event\_time", from\_unixtime(col("timestamp").cast("long")).cast("timestamp"))