SQL SCRIPT

Step 1: Aggregate DB1B by route and quarter

```
CREATE TABLE db1b_grouped AS
SELECT
  YEAR,
  QUARTER,
  ORIGIN,
  DEST,
  AVG(MARKET_FARE) AS AVG_MARKET_FARE
FROM db1b_market
GROUP BY YEAR, QUARTER, ORIGIN, DEST;
```

Step 2: Aggregate T-100 data by route and quarter

```
CREATE TABLE t100_grouped AS
SELECT
  YEAR,
  QUARTER,
  ORIGIN,
  DEST,
  UNIQUE_CARRIER,
  SUM(TOTAL_PASSENGERS) AS TOTAL_PASSENGERS,
  SUM(TOTAL_SEATS) AS TOTAL_SEATS,
  SUM(DEPARTURES) AS DEPARTURES,
  AVG(DISTANCE) AS DISTANCE
FROM t100_segment
WHERE NOT (YEAR = 2025 AND QUARTER = 2) -- Exclude Q2 2025
```

```
GROUP BY YEAR, QUARTER, ORIGIN, DEST, UNIQUE_CARRIER;
```

Step 3: Join aggregated T-100 with DB1B fare data and filter for Avelo (XP)

```
CREATE TABLE avelo_route_performance AS
SELECT
   t.YEAR,
   t.QUARTER,
   t.ORIGIN,
   t.DEST,
   t.UNIQUE_CARRIER,
   t.TOTAL_PASSENGERS,
   t.TOTAL_SEATS,
   t.DEPARTURES,
   t.DISTANCE,
   d.AVG_MARKET_FARE
FROM t100_grouped t
LEFT JOIN db1b_grouped d
  ON t.YEAR = d.YEAR
  AND t.QUARTER = d.QUARTER
  AND t.ORIGIN = d.ORIGIN
  AND t.DEST = d.DEST
WHERE t.UNIQUE_CARRIER = 'XP';
```

Step 4: Remove irrelevant or null rows

```
DELETE FROM avelo_route_performance
WHERE TOTAL_PASSENGERS = 0;
```

```
DELETE FROM avelo_route_performance
WHERE AVG_MARKET_FARE IS NULL;
```

Step 5: Add and populate route column (for convenience)

```
ALTER TABLE avelo_route_performance
ADD COLUMN ROUTE TEXT;

UPDATE avelo_route_performance
SET ROUTE = ORIGIN || '-' || DEST;
```

Step 6: Count directionless unique routes in 2024 (sanity check)

```
SELECT COUNT(DISTINCT
  CASE
    WHEN ORIGIN < DEST THEN ORIGIN || '-' || DEST
    ELSE DEST || '-' || ORIGIN
  END
) AS unique_directionless_routes_2024
FROM avelo_route_performance
WHERE YEAR = 2024
 AND DEPARTURES > 5;
```