Programme: Bachelor of Computer Science (Honours) in Data Science

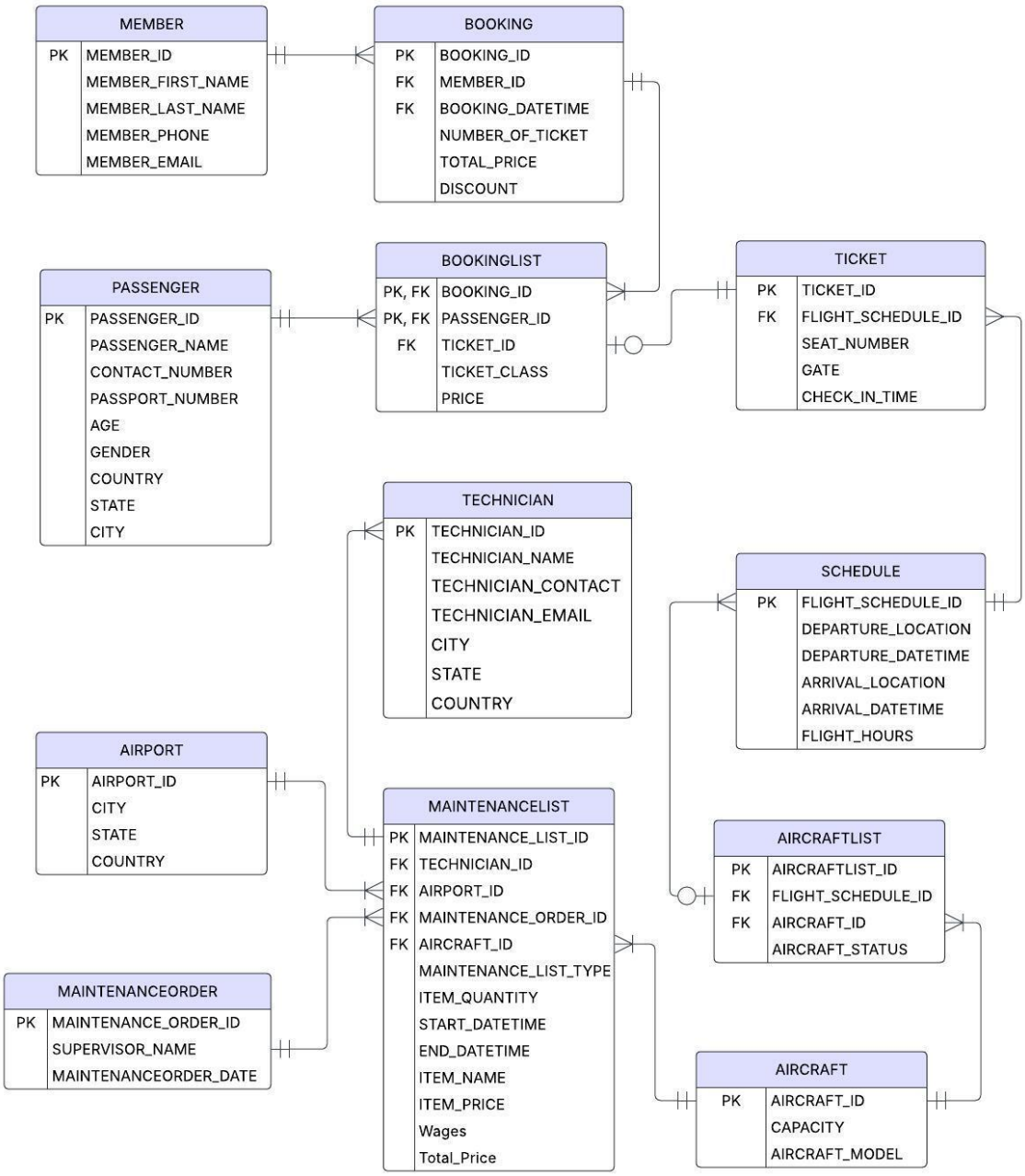
Subject: BAIT 3003 Data Warehouse Technology (202501)

Table of Contents

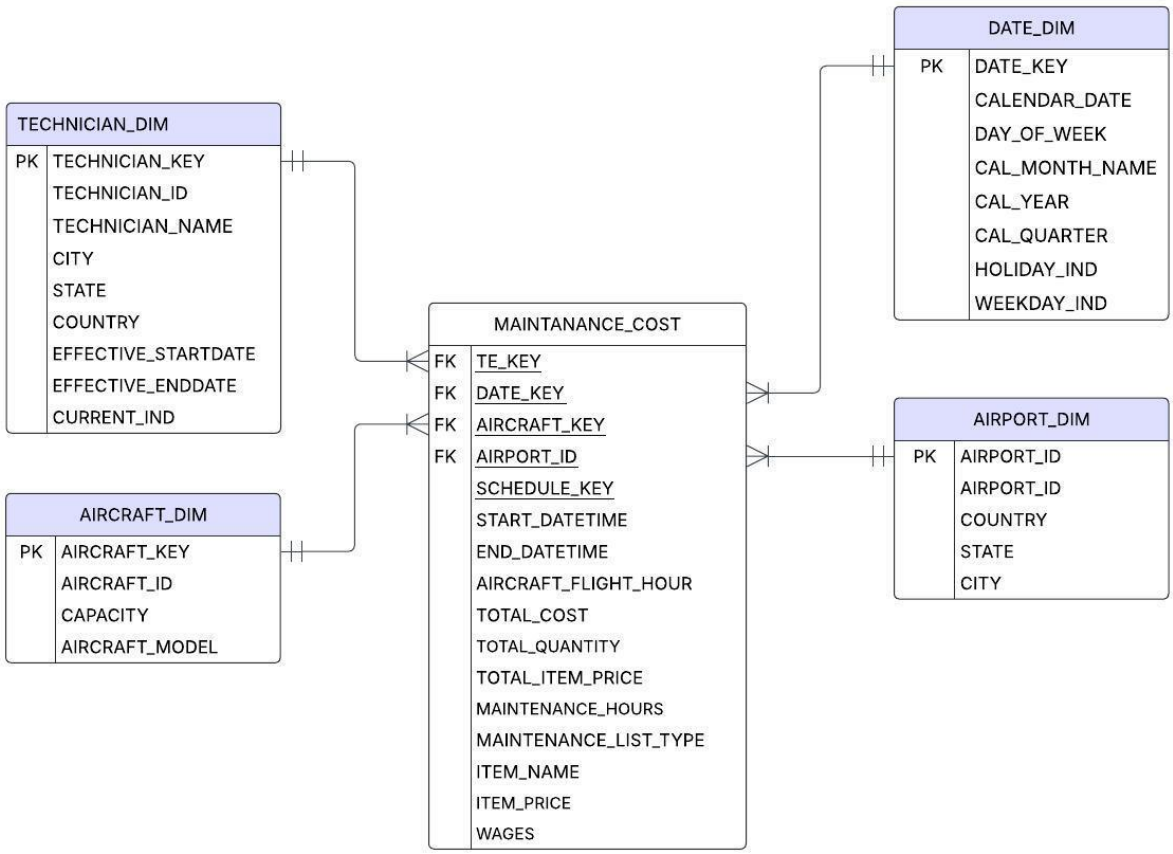
Chapter 1 Design of Data Warehouse

1.1 Logical Design

1.1.1 Original Database



1.1.2 Star Schema Dimension and Fact Tables



1.2 Physical Design

1.2.1 Dimension Tables

Date Dimension Table

```
CREATE TABLE DATE_DIM (
  DATE_KEY           NUMBER NOT NULL,
  CALENDAR_DATE      DATE NOT NULL,
  DAY_OF_WEEK        NUMBER(1) NOT NULL,
  CAL_MONTH_NAME     VARCHAR2(10) NOT NULL,
  CAL_YEAR           NUMBER(4) NOT NULL,
  CAL_QUARTER        CHAR(2) NOT NULL,--Q1
  HOLIDAY_IND        CHAR(1) NOT NULL,--Y
  WEEKDAY_IND        NUMBER(1) NOT NULL,
  constraint DATE_KEY_PK PRIMARY KEY(DATE_KEY)
);
```

Technician Dimension Table

```
CREATE TABLE TECHNICIAN_DIM (
  TECHNICIAN_KEY     NUMBER NOT NULL,
  TECHNICIAN_ID      VARCHAR2(6) NOT NULL,
  TECHNICIAN_NAME    VARCHAR2(50) NOT NULL,
  CITY VARCHAR(40) NOT NULL,
  STATE VARCHAR(40) NOT NULL,
  COUNTRY VARCHAR(40) NOT NULL,
  EFFECTIVE_STARTDATE DATE NOT NULL,
  EFFECTIVE_ENDDATE DATE NOT NULL,
  CURRENT_IND CHAR(1) NOT NULL,
  constraint TECHNICIAN_DIM_PK PRIMARY KEY(TECHNICIAN_KEY)
);
```

Aircraft Dimension Table

```
CREATE TABLE AIRCRAFT_DIM (
  AIRCRAFT_KEY     NUMBER NOT NULL,
```

```
  AIRCRAFT_ID      VARCHAR(6) NOT NULL,
  CAPACITY         NUMBER NOT NULL,
  AIRCRAFT_MODEL   VARCHAR2(50) NOT NULL,
  constraint AIRCRAFT_DIM_PK PRIMARY KEY(AIRCRAFT_KEY)
);
```

Airport Dimension Table

```
CREATE TABLE AIRPORT_DIM (
  AIRPORT_KEY NUMBER      NOT NULL,
  AIRPORT_ID  VARCHAR2(4) NOT NULL,
  COUNTRY     VARCHAR2(50) NOT NULL,
  STATE       VARCHAR2(50) NOT NULL,
  CITY        VARCHAR2(50) NOT NULL,
  constraint AIRPORT_DIM_PK PRIMARY KEY(AIRPORT_KEY)
);
```

1.2.2 Fact Table

```
CREATE TABLE MAINTENANCE_COST (
  TE_KEY              NUMBER NOT NULL,
  DATE_KEY            NUMBER NOT NULL,
  AIRCRAFT_KEY        NUMBER NOT NULL,
  AIRPORT_KEY         NUMBER NOT NULL,
  START_DATETIME      TIMESTAMP(0) NOT NULL,
  END_DATETIME            TIMESTAMP(0) NOT NULL,
  AIRCRAFT_FLIGHT_HOUR NUMBER(6,2) NOT NULL,
  TOTAL_COST          NUMBER(10,2) NOT NULL,
  TOTAL_QUANTITY      NUMBER(4),
  TOTAL_ITEMPRICE         NUMBER(10,2),
  MAINTENANCE_HOURS   NUMBER(6,2) NOT NULL,
  MAINTENANCE_LIST_TYPE VARCHAR2(30) NOT NULL,
  ITEM_NAME           VARCHAR2(30),
  ITEM_PRICE       NUMBER,
  WAGES               NUMBER(8,2) NOT NULL,
  constraint MC_TE_PK FOREIGN KEY (TE_KEY)       REFERENCES
TECHNICIAN_DIM(TECHNICIAN_KEY),
  constraint MC_DATE_PK FOREIGN KEY (DATE_KEY)     REFERENCES DATE_DIM(DATE_KEY),
  constraint MC_AIRCRAFT_PK FOREIGN KEY (AIRCRAFT_KEY) REFERENCES
AIRCRAFT_DIM(AIRCRAFT_KEY),
  constraint MC_AIRPORT_PK FOREIGN KEY (AIRPORT_KEY)   REFERENCES
AIRPORT_DIM(AIRPORT_KEY)
);
```

Chapter 2 Extract, Transform, Load Process

2.1 Script for initial loading

Date Dimension Table

```
DROP SEQUENCE date_seq;
CREATE SEQUENCE date_seq START WITH 1001 INCREMENT BY 1;

CREATE OR REPLACE VIEW VW_DATE_DIM AS
SELECT * FROM DATE_DIM;

CREATE OR REPLACE PROCEDURE Load_Date_Dim_IL IS
  every_date DATE := TO_DATE('2020-01-01','YYYY-MM-DD');
  end_date   DATE := TRUNC(SYSDATE);
  v_day_of_week NUMBER(1);
  v_month_name VARCHAR2(10);
  v_year NUMBER(4);
  v_quarter CHAR(2);
  v_weekday_ind NUMBER(1);
BEGIN
  WHILE every_date <= end_date LOOP
    v_day_of_week := TO_NUMBER(TO_CHAR(every_date, 'D'));
    v_month_name := INITCAP(TO_CHAR(every_date, 'Month'));
    v_year := TO_NUMBER(TO_CHAR(every_date, 'YYYY'));

    CASE
      WHEN EXTRACT(MONTH FROM every_date) <= 3 THEN v_quarter := 'Q1';
      WHEN EXTRACT(MONTH FROM every_date) <= 6 THEN v_quarter := 'Q2';
      WHEN EXTRACT(MONTH FROM every_date) <= 9 THEN v_quarter := 'Q3';
      ELSE v_quarter := 'Q4';
    END CASE;

    IF v_day_of_week IN (1, 7) THEN
      v_weekday_ind := 0;
    ELSE
      v_weekday_ind := 1;
    END IF;

    INSERT INTO DATE_DIM
    VALUES (
      date_seq.NEXTVAL,
      every_date,
      v_day_of_week,
      TRIM(v_month_name),
      v_year,
      v_quarter,
      'N',
      v_weekday_ind
    );
    every_date := every_date + 1;
```

```
  END LOOP;
  COMMIT;
END;
/
```

Technician Dimension Table

```
DROP SEQUENCE tech_dim_seq;

CREATE SEQUENCE tech_dim_seq START WITH 20001 INCREMENT BY 1;

-- Create or replace the technician dimension view
CREATE OR REPLACE VIEW VW_TECHNICIAN_DIM AS
SELECT * FROM TECHNICIAN_DIM;

CREATE OR REPLACE PROCEDURE Load_Technician_Dim_IL IS
BEGIN
  -- Insert data into TECHNICIAN_DIM from TECHNICIAN table
  INSERT INTO TECHNICIAN_DIM (
    TECHNICIAN_KEY,
    TECHNICIAN_ID,
    TECHNICIAN_NAME,
    CITY,
    STATE,
    COUNTRY,
    EFFECTIVE_STARTDATE,
    EFFECTIVE_ENDDATE,
    CURRENT_IND
  )
  SELECT
    tech_dim_seq.NEXTVAL,  -- Generate TECHNICIAN_KEY using sequence
    TECHNICIAN_ID,         -- TECHNICIAN_ID from source table
    TECHNICIAN_NAME,       -- TECHNICIAN_NAME from source table
    CITY,                  -- CITY from source table
    STATE,                 -- STATE from source table
    COUNTRY,               -- COUNTRY from source table
    -- Generate random start date between 01/01/2020 and current system date
    TO_DATE('01/01/2020', 'DD/MM/YYYY') +
      ROUND(DBMS_RANDOM.VALUE(0, (SYSDATE - TO_DATE('01/01/2020', 'DD/MM/YYYY')))),
-- Random start date
    TO_DATE('31/12/9999', 'DD/MM/YYYY'), -- Fixed end date
    'Y'                    -- Set CURRENT_IND as 'Y'
  FROM TECHNICIAN;

  -- Commit the transaction
  COMMIT;
END;
/
```

Aircraft Dimension Table

```
DROP SEQUENCE aircraft_dim_seq;
CREATE SEQUENCE aircraft_dim_seq START WITH 30001 INCREMENT BY 1;

CREATE OR REPLACE VIEW VW_AIRCRAFT_DIM AS
SELECT * FROM AIRCRAFT_DIM;

CREATE OR REPLACE PROCEDURE Load_Aircraft_Dim_IL IS
BEGIN
  INSERT INTO AIRCRAFT_DIM
  SELECT aircraft_dim_seq.NEXTVAL,
         AIRCRAFT_ID, -- No SUBSTR, use full AIRCRAFT_ID (5 characters)
         CAPACITY,
         AIRCRAFT_MODEL
  FROM AIRCRAFT;
  COMMIT;
END;
/
```

Airport Dimension Table

```
DROP SEQUENCE airport_dim_seq;
CREATE SEQUENCE airport_dim_seq START WITH 40001 INCREMENT BY 1;

CREATE OR REPLACE VIEW VW_AIRPORT_DIM AS
SELECT * FROM AIRPORT_DIM;

CREATE OR REPLACE PROCEDURE Load_Airport_Dim_IL IS
BEGIN
  INSERT INTO AIRPORT_DIM
  SELECT airport_dim_seq.NEXTVAL,
         AIRPORT_ID,
         COUNTRY,
         STATE,
         CITY
  FROM AIRPORT;
  COMMIT;
END;
/
```

Fact Table

```
CREATE OR REPLACE VIEW VW_MAINTENANCE_COST AS
SELECT * FROM MAINTENANCE_COST;

CREATE OR REPLACE PROCEDURE Load_MaintenanceCost_Fact_IL IS
BEGIN
  INSERT INTO MAINTENANCE_COST (
    TE_KEY, DATE_KEY, AIRCRAFT_KEY, AIRPORT_KEY,
    START_DATETIME, END_DATETIME, AIRCRAFT_FLIGHT_HOUR, TOTAL_COST,
    MAINTENANCE_HOURS, TOTAL_QUANTITY, ITEM_PRICE, WAGES, TOTAL_ITEMPRICE,
    MAINTENANCE_LIST_TYPE, ITEM_NAME
  )
  SELECT
    T.TECHNICIAN_KEY,
    D.DATE_KEY,
    A.AIRCRAFT_KEY,
    P.AIRPORT_KEY,
    ML.START_DATETIME,
    ML.END_DATETIME,
    (EXTRACT(DAY FROM (ML.END_DATETIME - ML.START_DATETIME)) * 24 +
     EXTRACT(HOUR FROM (ML.END_DATETIME - ML.START_DATETIME)) +
     EXTRACT(MINUTE FROM (ML.END_DATETIME - ML.START_DATETIME)) / 60) AS
AIRCRAFT_FLIGHT_HOUR,
    ML.TOTAL_PRICE AS TOTAL_COST,
    (EXTRACT(DAY FROM (ML.END_DATETIME - ML.START_DATETIME)) * 24 +
     EXTRACT(HOUR FROM (ML.END_DATETIME - ML.START_DATETIME)) +
     EXTRACT(MINUTE FROM (ML.END_DATETIME - ML.START_DATETIME)) / 60) AS
MAINTENANCE_HOURS,
    ML.ITEM_QUANTITY,
    ML.ITEM_PRICE,
    ML.WAGES,
    NVL(ML.ITEM_PRICE * ML.ITEM_QUANTITY, 0) AS TOTAL_ITEMPRICE,
    ML.MAINTENANCE_LIST_TYPE,
    ML.ITEM_NAME
  FROM MAINTENANCELIST ML
  JOIN TECHNICIAN_DIM T ON ML.TECHNICIAN_ID = T.TECHNICIAN_ID
  JOIN AIRCRAFT_DIM A ON ML.AIRCRAFT_ID = A.AIRCRAFT_ID -- No SUBSTR
  JOIN AIRPORT_DIM P ON ML.AIRPORT_ID = P.AIRPORT_ID
  JOIN MAINTENANCEORDER MO ON ML.MAINTENANCE_ORDER_ID = MO.MAINTENANCE_ORDER_ID
  JOIN DATE_DIM D ON TRUNC(MO.MAINTENANCEORDER_DATE) = D.CALENDAR_DATE;
  COMMIT;
END;
/

-- Execute All Load
BEGIN
  Load_Date_Dim_IL;
  Load_Technician_Dim_IL;
  Load_Aircraft_Dim_IL;
  Load_Airport_Dim_IL;
  Load_MaintenanceCost_Fact_IL;
END;
/
```

2.2 Script for subsequent loading

Date Dimension Table

```
CREATE OR REPLACE PROCEDURE Load_Date_Dim_Subsequent IS
  v_max_date DATE;
  every_date  DATE;
  end_date    DATE := TRUNC(SYSDATE);
  v_day_of_week NUMBER(1);
  v_month_name VARCHAR2(10);
  v_year        NUMBER(4);
  v_quarter     CHAR(2);
  v_weekday_ind NUMBER(1);
BEGIN
  SELECT MAX(CALENDAR_DATE) INTO v_max_date FROM DATE_DIM;
  every_date := NVL(v_max_date + 1, TO_DATE('2020-01-01', 'YYYY-MM-DD'));

  WHILE every_date <= end_date LOOP
    v_day_of_week := TO_NUMBER(TO_CHAR(every_date, 'D'));
    v_month_name := INITCAP(TO_CHAR(every_date, 'Month'));
    v_year := TO_NUMBER(TO_CHAR(every_date, 'YYYY'));

    CASE
      WHEN EXTRACT(MONTH FROM every_date) <= 3 THEN v_quarter := 'Q1';
      WHEN EXTRACT(MONTH FROM every_date) <= 6 THEN v_quarter := 'Q2';
      WHEN EXTRACT(MONTH FROM every_date) <= 9 THEN v_quarter := 'Q3';
      ELSE v_quarter := 'Q4';
    END CASE;

    IF v_day_of_week IN (1, 7) THEN
      v_weekday_ind := 0;
    ELSE
      v_weekday_ind := 1;
    END IF;

    INSERT INTO DATE_DIM
    VALUES (
      date_seq.NEXTVAL,
      every_date,
      v_day_of_week,
      TRIM(v_month_name),
      v_year,
      v_quarter,
      'N',
      v_weekday_ind
    );
    every_date := every_date + 1;
  END LOOP;
  COMMIT;
END;
/
```

Technician Dimension Table

```
CREATE OR REPLACE PROCEDURE Load_Technician_Dim_Subsequent IS
BEGIN
  INSERT INTO TECHNICIAN_DIM (
    TECHNICIAN_KEY,         -- Primary Key
    TECHNICIAN_ID,          -- Technician ID
    TECHNICIAN_NAME,        -- Technician Name
    CITY,                   -- City
    STATE,                  -- State
    COUNTRY,                -- Country
    EFFECTIVE_STARTDATE,    -- Effective start date
    EFFECTIVE_ENDDATE,      -- Effective end date
    CURRENT_IND             -- Current indicator
  )
  SELECT
    tech_dim_seq.NEXTVAL,                 -- Generate TECHNICIAN_KEY using sequence
    t.TECHNICIAN_ID,                      -- TECHNICIAN_ID from source table
    TRIM(t.TECHNICIAN_NAME),              -- Trim whitespace from TECHNICIAN_NAME
    TRIM(t.CITY),                         -- Trim whitespace from CITY
    TRIM(t.STATE),                        -- Trim whitespace from STATE
    TRIM(t.COUNTRY),                      -- Trim whitespace from COUNTRY
    SYSDATE,                              -- Set current date as the effective
start date
    TO_DATE('31/12/9999', 'DD/MM/YYYY'),   -- Set distant future date as the
effective end date
    'Y'                                   -- Set CURRENT_IND as 'Y' (indicating
active record)
  FROM
    TECHNICIAN t
  WHERE
    NOT EXISTS (SELECT 1 FROM TECHNICIAN_DIM td WHERE t.TECHNICIAN_ID =
td.TECHNICIAN_ID);  -- Avoid inserting duplicates
    COMMIT;
END;
/
```

Aircraft Dimension Table

```
CREATE OR REPLACE PROCEDURE Load_Aircraft_Dim_Subsequent IS
BEGIN
  INSERT INTO AIRCRAFT_DIM
  SELECT aircraft_dim_seq.NEXTVAL,
         SUBSTR(a.AIRCRAFT_ID, 1, 5), -- Corrected SUBSTR length to 5 to match
initial load
         a.CAPACITY,
         TRIM(a.AIRCRAFT_MODEL)
  FROM AIRCRAFT a
  WHERE NOT EXISTS (SELECT 1 FROM AIRCRAFT_DIM ad WHERE SUBSTR(a.AIRCRAFT_ID, 1, 5)
= ad.AIRCRAFT_ID); -- Corrected SUBSTR length
```

```
  COMMIT;
END;
/
```

Airport Dimension Table

```
CREATE OR REPLACE PROCEDURE Load_Airport_Dim_Subsequent IS
BEGIN
  INSERT INTO AIRPORT_DIM
  SELECT airport_dim_seq.NEXTVAL,
         ap.AIRPORT_ID,
         TRIM(ap.COUNTRY),
         TRIM(ap.STATE),
         TRIM(ap.CITY)
  FROM AIRPORT ap
  WHERE NOT EXISTS (SELECT 1 FROM AIRPORT_DIM ad WHERE ap.AIRPORT_ID =
ad.AIRPORT_ID);
  COMMIT;
END;
/
```

Fact Table

```
-- 6. MAINTENANCE_COST FACT ETL Subsequent Load
CREATE OR REPLACE PROCEDURE Load_MaintenanceCost_Fact_S IS
BEGIN
  INSERT INTO MAINTENANCE_COST (
    MAINTENANCE_KEY, TE_KEY, DATE_KEY, AIRCRAFT_KEY, AIRPORT_KEY,
    START_DATETIME, END_DATETIME, AIRCRAFT_FLIGHT_HOUR, TOTAL_COST,
    MAINTENANCE_HOURS, TOTAL_QUANTITY, ITEM_PRICE, WAGES, TOTAL_ITEMPRICE
  )
  SELECT
    mld.MAINTENANCE_KEY,
    td.TECHNICIAN_KEY,
    dd.DATE_KEY,
    ad.AIRCRAFT_KEY,
    apd.AIRPORT_KEY,
    ml.START_DATETIME,
    ml.END_DATETIME,
    ml.END_DATETIME, -- Placeholder
    ml.TOTAL_PRICE,
    (CAST(ml.END_DATETIME AS DATE) - CAST(ml.START_DATETIME AS DATE)) * 24,
    ml.ITEM_QUANTITY,
    ml.ITEM_PRICE,
    ml.WAGES,
    NVL(ml.ITEM_PRICE * ml.ITEM_QUANTITY, 0)
  FROM MAINTENANCELIST ml
  JOIN MAINTENANCELIST_DIM mld ON ml.MAINTENANCE_LIST_ID = mld.MAINTENANCE_LIST_ID
  JOIN TECHNICIAN_DIM td ON ml.TECHNICIAN_ID = td.TECHNICIAN_ID
  JOIN AIRCRAFT_DIM ad ON SUBSTR(ml.AIRCRAFT_ID, 1, 6) = ad.AIRCRAFT_ID
  JOIN AIRPORT_DIM apd ON ml.AIRPORT_ID = apd.AIRPORT_ID
  JOIN MAINTENANCEORDER mo ON ml.MAINTENANCE_ORDER_ID = mo.MAINTENANCE_ORDER_ID
  JOIN DATE_DIM dd ON TRUNC(mo.MAINTENANCEORDER_DATE) = dd.CALENDAR_DATE
  WHERE NOT EXISTS (SELECT 1
                    FROM MAINTENANCE_COST mc
```

```
                      WHERE mld.MAINTENANCE_KEY = mc.MAINTENANCE_KEY
                        AND td.TECHNICIAN_KEY = mc.TE_KEY
                        AND dd.DATE_KEY = mc.DATE_KEY
                        AND ad.AIRCRAFT_KEY = mc.AIRCRAFT_KEY
                        AND apd.AIRPORT_KEY = mc.AIRPORT_KEY);
  COMMIT;
END;
/

-- Execute All Subsequent Loads
BEGIN
  Load_Date_Dim_Subsequent;
  Load_MaintList_Dim_Subsequent;
  Load_Technician_Dim_Subsequent;
  Load_Aircraft_Dim_Subsequent;
  Load_Airport_Dim_Subsequent;
  Load_MaintenanceCost_Fact_S;
END;
/
```

2.3 Script for SCD Type 2

```
CREATE OR REPLACE PROCEDURE PROC_UPDATE_TECHNICIAN_TYPE2 (
    v_technician_id IN VARCHAR2,  -- Technician ID to update
    v_category IN VARCHAR2,       -- Category to update: 'NAME', 'CITY', 'STATE', or 'COUNTRY'
    v_new_value IN VARCHAR2       -- New value for the update
) IS
    -- Declare variables for the current record in TECHNICIAN_DIM
    v_technician_name TECHNICIAN_DIM.TECHNICIAN_NAME%TYPE;
    v_city TECHNICIAN_DIM.CITY%TYPE;
    v_state TECHNICIAN_DIM.STATE%TYPE;
    v_country TECHNICIAN_DIM.COUNTRY%TYPE;
    v_effective_date DATE;        -- Will be set to previous end date + 1
    v_expiration_date DATE := TO_DATE('31/12/9999', 'DD/MM/YYYY');  -- Set expiration date to a
distant future
    v_current_indicator CHAR(1) := 'Y';  -- New records will have 'Y' as current
    v_previous_end_date DATE;

    CURSOR dim_cursor IS
        SELECT TECHNICIAN_NAME, CITY, STATE, COUNTRY, EFFECTIVE_ENDDATE, CURRENT_IND
        FROM TECHNICIAN_DIM
        WHERE TECHNICIAN_ID = v_technician_id AND CURRENT_IND = 'Y';

BEGIN
    -- Open the cursor to fetch the current record from the technician dimension
    OPEN dim_cursor;
    FETCH dim_cursor INTO v_technician_name, v_city, v_state, v_country, v_previous_end_date,
v_current_indicator;

    -- Check if record is found
    IF dim_cursor%NOTFOUND THEN
        DBMS_OUTPUT.PUT_LINE('Technician ID ' || v_technician_id || ' does not exist in
TECHNICIAN_DIM.');
    ELSE
        -- Close the current active record in TECHNICIAN_DIM
        UPDATE TECHNICIAN_DIM
        SET EFFECTIVE_ENDDATE = SYSDATE, CURRENT_IND = 'N'
        WHERE TECHNICIAN_ID = v_technician_id AND CURRENT_IND = 'Y';

        -- Set the effective date to previous end date + 1 day
        v_effective_date := SYSDATE + 1;

        -- Insert new record with updated information into TECHNICIAN_DIM
        INSERT INTO TECHNICIAN_DIM (
            TECHNICIAN_KEY,
            TECHNICIAN_ID,
            TECHNICIAN_NAME,
            CITY,
            STATE,
            COUNTRY,
            EFFECTIVE_STARTDATE,
            EFFECTIVE_ENDDATE,
            CURRENT_IND
        )
        VALUES (
            tech_dim_seq.NEXTVAL,                   -- Generate new key using the sequence
            v_technician_id,                        -- Technician ID
            CASE WHEN UPPER(v_category) = 'NAME' THEN v_new_value ELSE v_technician_name END,
-- Update based on category
            CASE WHEN UPPER(v_category) = 'CITY' THEN v_new_value ELSE v_city END,           --
Update based on category
            CASE WHEN UPPER(v_category) = 'STATE' THEN v_new_value ELSE v_state END,         --
Update based on category
            CASE WHEN UPPER(v_category) = 'COUNTRY' THEN v_new_value ELSE v_country END,     --
Update based on category
            v_effective_date,                       -- Set effective date to previous end date
+ 1
            v_expiration_date,                      -- Set expiration date to distant future
            v_current_indicator                     -- Set as current (active) record
```

```
        );

        DBMS_OUTPUT.PUT_LINE('Technician ' || v_technician_id || ' updated ' || v_category || '
from ' ||
                             CASE
                                 WHEN UPPER(v_category) = 'NAME' THEN v_technician_name
                                 WHEN UPPER(v_category) = 'CITY' THEN v_city
                                 WHEN UPPER(v_category) = 'STATE' THEN v_state
                                 WHEN UPPER(v_category) = 'COUNTRY' THEN v_country
                                 ELSE 'N/A'
                             END || ' to ' || v_new_value);
        DBMS_OUTPUT.PUT_LINE('TECHNICIAN_DIM table updated. New record inserted.');
    END IF;

    -- Close the cursor
    CLOSE dim_cursor;
END;
/
```

Chapter 3 Business Analytics Reports

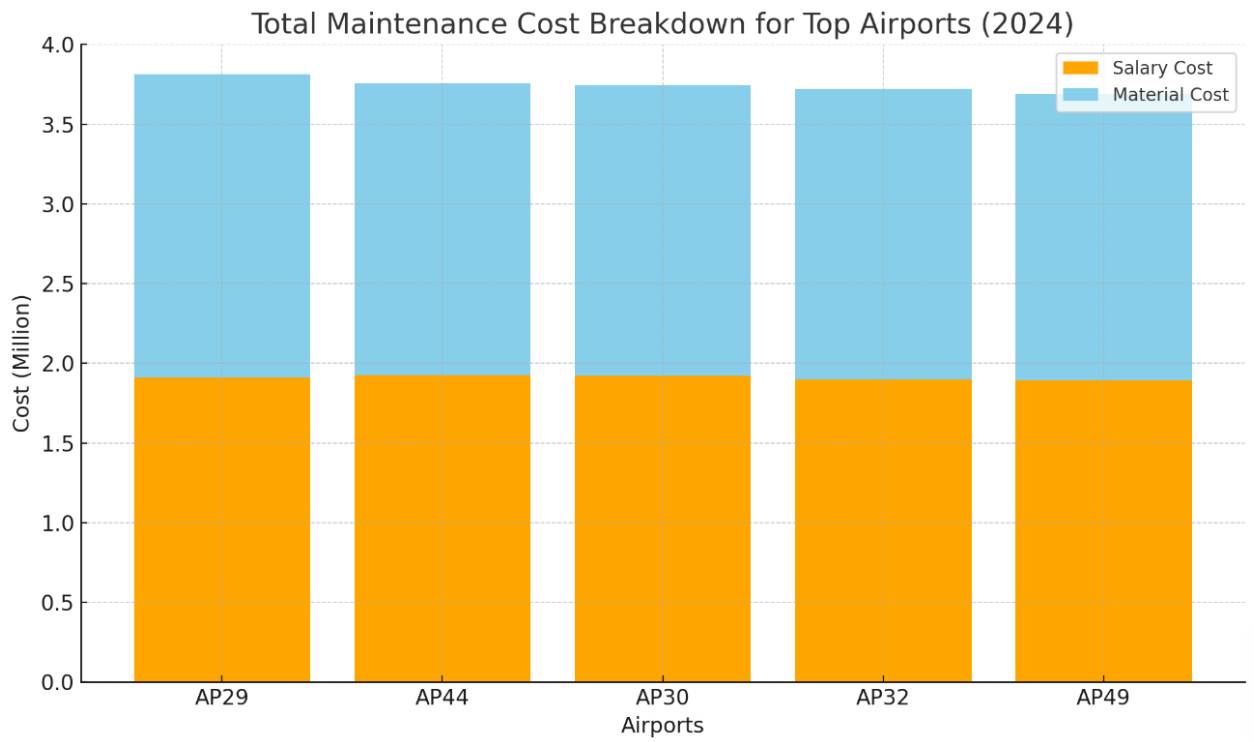
3.1 Tan Rou Ming

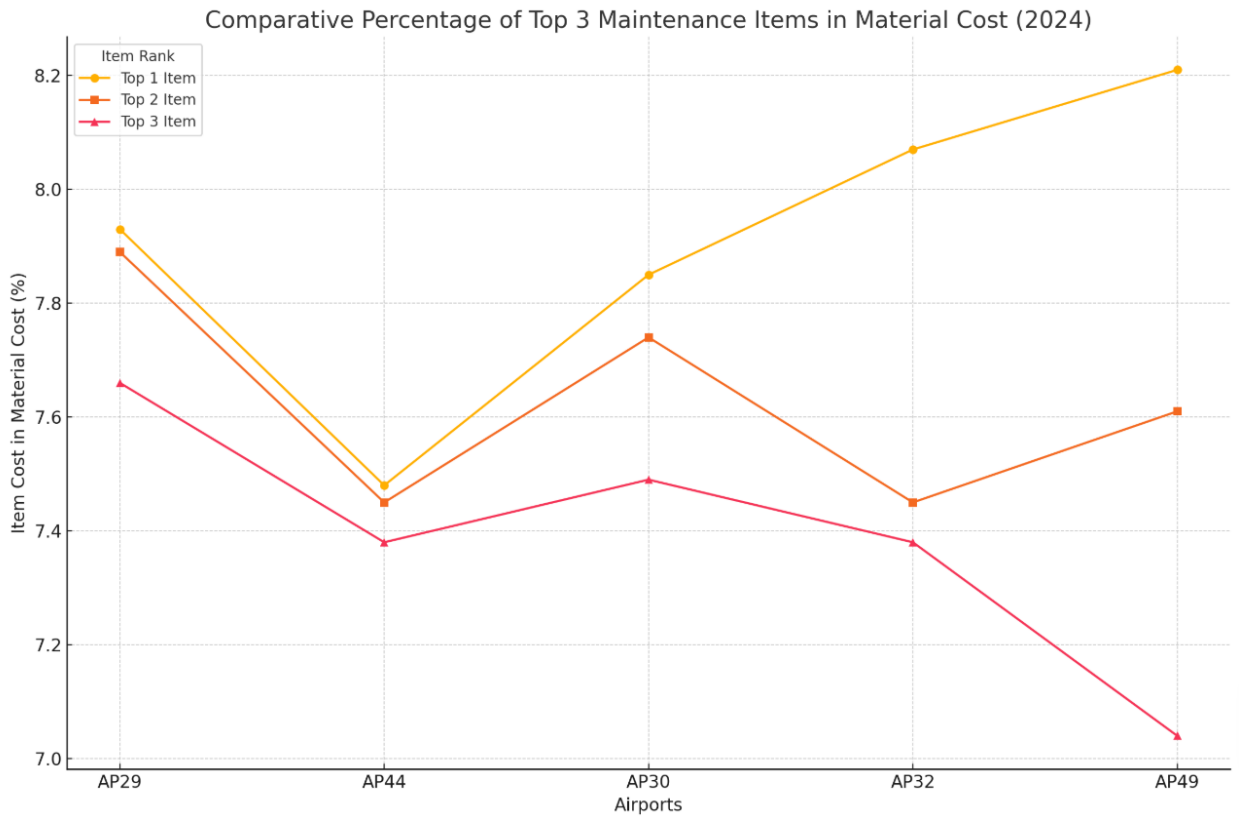
3.1.1 Top 5 Airports by Maintenance Cost and Their Top 3 Items

```
ACCEPT rpt_year PROMPT 'Enter Year (YYYY): '

SET LINESIZE  132
SET PAGESIZE  50

-- Column formatting
COLUMN airport_id    FORMAT A8               HEADING 'Airport'
COLUMN total_cost    FORMAT $999,999,999.99  HEADING 'Total Cost'
COLUMN material_cost FORMAT $999,999,999.99  HEADING 'Material Cost'
COLUMN salary_cost   FORMAT $999,999,999.99  HEADING 'Salary Cost'
COLUMN pct_material  FORMAT  999.99           HEADING '% Material'
COLUMN pct_salary    FORMAT  999.99           HEADING '% Salary'
COLUMN item_name     FORMAT A20              HEADING 'Top Item'
COLUMN item_cost     FORMAT $999,999,999.99  HEADING 'Item Cost'
COLUMN item_pct      FORMAT  9999.99           HEADING 'Item (%MC)'

BREAK ON airport_id SKIP 1 ON total_cost ON material_cost ON salary_cost ON
pct_material ON pct_salary

COMPUTE SUM LABEL 'Total: ' OF "Item Cost" ON airport_id
COMPUTE SUM LABEL 'Total: ' OF "Item (%MC)" ON airport_id

TTITLE CENTER 'Top 5 Airports by Maintenance Cost and Their Top 3 Items (&rpt_year)'
-
       SKIP 2 -

WITH all_summary AS (
  SELECT
    ad.airport_id,
    SUM(mc.total_cost)                     AS total_cost,
    SUM(mc.item_price * mc.total_quantity) AS material_cost,
    SUM(mc.wages * mc.maintenance_hours)   AS salary_cost
  FROM maintenance_cost mc
  JOIN airport_dim ad ON mc.airport_key = ad.airport_key
  JOIN date_dim    dd ON mc.date_key    = dd.date_key
  WHERE dd.cal_year = &rpt_year
  GROUP BY ad.airport_id
),
ranked_airports AS (
  SELECT
    airport_id,
    total_cost,
    material_cost,
    salary_cost,
    ROUND(material_cost/NULLIF(total_cost,0)*100,2) AS pct_material,
    ROUND(salary_cost  /NULLIF(total_cost,0)*100,2) AS pct_salary,
    ROW_NUMBER() OVER (ORDER BY total_cost DESC)    AS airport_rank
  FROM all_summary
),
top_items AS (
  SELECT
    ad.airport_id,
```

```
    mc.item_name,
    SUM(mc.item_price * mc.total_quantity) AS item_cost,
    ROW_NUMBER() OVER (
      PARTITION BY ad.airport_id
      ORDER BY SUM(mc.item_price * mc.total_quantity) DESC
    ) AS item_rank
  FROM maintenance_cost mc
  JOIN airport_dim ad ON mc.airport_key = ad.airport_key
  JOIN date_dim    dd ON mc.date_key    = dd.date_key
  WHERE dd.cal_year = &rpt_year
  GROUP BY ad.airport_id, mc.item_name
),
combined AS (
  -- only top 3 non-null items for top 10 airports
  SELECT
    ra.airport_id,
    ra.total_cost,
    ra.material_cost,
    ra.salary_cost,
    ra.pct_material,
    ra.pct_salary,
    ti.item_name,
    ti.item_cost,
    ROUND(ti.item_cost/NULLIF(ra.material_cost,0)*100,2) AS item_pct,
    ti.item_rank
  FROM ranked_airports ra
  JOIN top_items ti ON ra.airport_id = ti.airport_id
  WHERE ra.airport_rank <= 5
    AND ti.item_rank <= 3
    AND ti.item_name IS NOT NULL
)
SELECT
  airport_id,
  total_cost,
  material_cost,
  salary_cost,
  pct_material,
  pct_salary,
  item_name   AS "Top Item",
  item_cost   AS "Item Cost",
  item_pct    AS "Item (%MC)"
FROM combined
ORDER BY total_cost DESC,
         airport_id,
         item_rank;

TTITLE OFF
```

```
SQL> start c:\q3.txt;
Enter Year (YYYY): 2024

                                Top 5 Airports by Maintenance Cost and Their Top 3 Items (2024)

Airport        Total Cost    Material Cost      Salary Cost % Material % Salary Top Item                        Item Cost Item (%MC)
-------- ---------------- ---------------- ---------------- ---------- -------- ------------------------------ ---------- ----------
AP29        $3,812,506.06    $1,904,366.01    $1,908,191.40      49.95    50.05 Bearing                       151110.16       7.93
                                                                                 Landing Gear Bolt               150306.9       7.89
                                                                                 Brake Pad                      145852.97       7.66
******** ***************** ***************** ***************** ********** ******** ------------------------------ ---------- ----------
Total:                                                                                                          447270.03      23.48

AP44        $3,759,887.86    $1,836,337.60    $1,923,473.02      48.84    51.16 Landing Gear Bolt             137367.51       7.48
                                                                                 Pressure Gauge                 136816.07       7.45
                                                                                 Bearing                        135592.49       7.38
******** ***************** ***************** ***************** ********** ******** ------------------------------ ---------- ----------
Total:                                                                                                          409776.07      22.31

AP30        $3,747,160.74    $1,825,199.03    $1,921,785.05      48.71    51.29 Bearing                       143346.39       7.85
                                                                                 Spark Plug                     141243.51       7.74
                                                                                 O-ring                            136689       7.49
******** ***************** ***************** ***************** ********** ******** ------------------------------ ---------- ----------
Total:                                                                                                          421278.9       23.08

AP32        $3,722,313.17    $1,823,376.87    $1,898,922.34      48.99    51.01 Fuel Pump                     147118.1       8.07
                                                                                 Brake Pad                      135899.51       7.45
                                                                                 Gasket                         134502.99       7.38
******** ***************** ***************** ***************** ********** ******** ------------------------------ ---------- ----------
Total:                                                                                                          417520.6       22.9

AP49        $3,690,298.90    $1,796,769.67    $1,893,376.60      48.69    51.31 Pressure Gauge                147557.52       8.21
                                                                                 Filter Kit                     136798.98       7.61
                                                                                 O-ring                         126521.19       7.04
******** ***************** ***************** ***************** ********** ******** ------------------------------ ---------- ----------
Total:                                                                                                          410877.69      22.86
```





In 2024, maintenance costs across the top five airports are nearly evenly split between salary and material expenses, each around 1.9 million. Since salary accounts for 50% of total costs, airports should focus on improving workforce efficiency—for example, by optimizing shift scheduling and reducing overtime. This could yield up to 5% in savings per site without reducing headcount.

The material cost analysis reveals a significant insight: just three items account for 22%–23% of total material expenditure per airport. AP49's pressure gauge alone makes up 8.21%, while AP29 spends nearly 460k across its top three items. Since these components (e.g., bearings, O-rings, brake pads) are commonly used across all airports, airports should adopt centralized, bulk purchasing contracts. Locking in firm-volume, multi-year agreements at current rates would protect against future price increases and improve supplier terms.

Moreover, the second chart highlights that high-cost items are consistent across locations. This makes a strong case for standardizing parts where possible to reduce complexity and enable inter-airport inventory sharing.

Collectively, these strategies can drive down operational costs, improve part availability, and boost long-term cost stability.

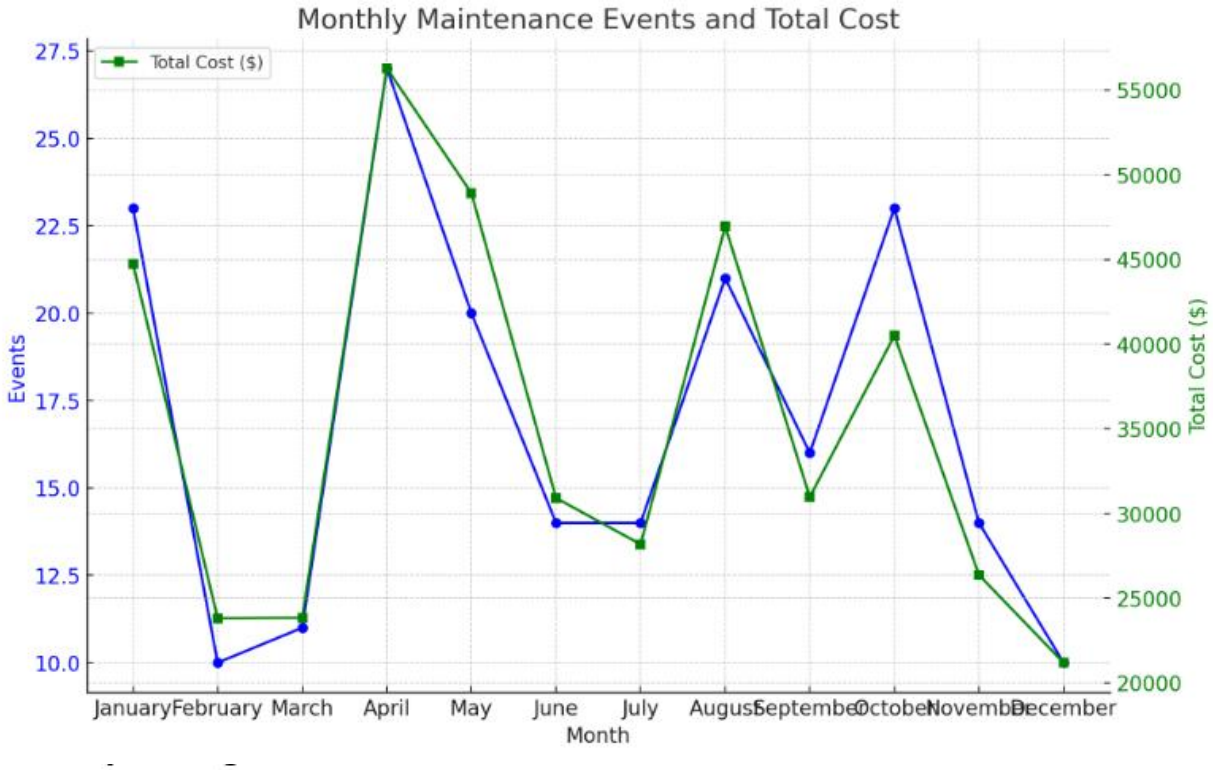
3.1.2 Maintenance Cost and Event Report by Yearly

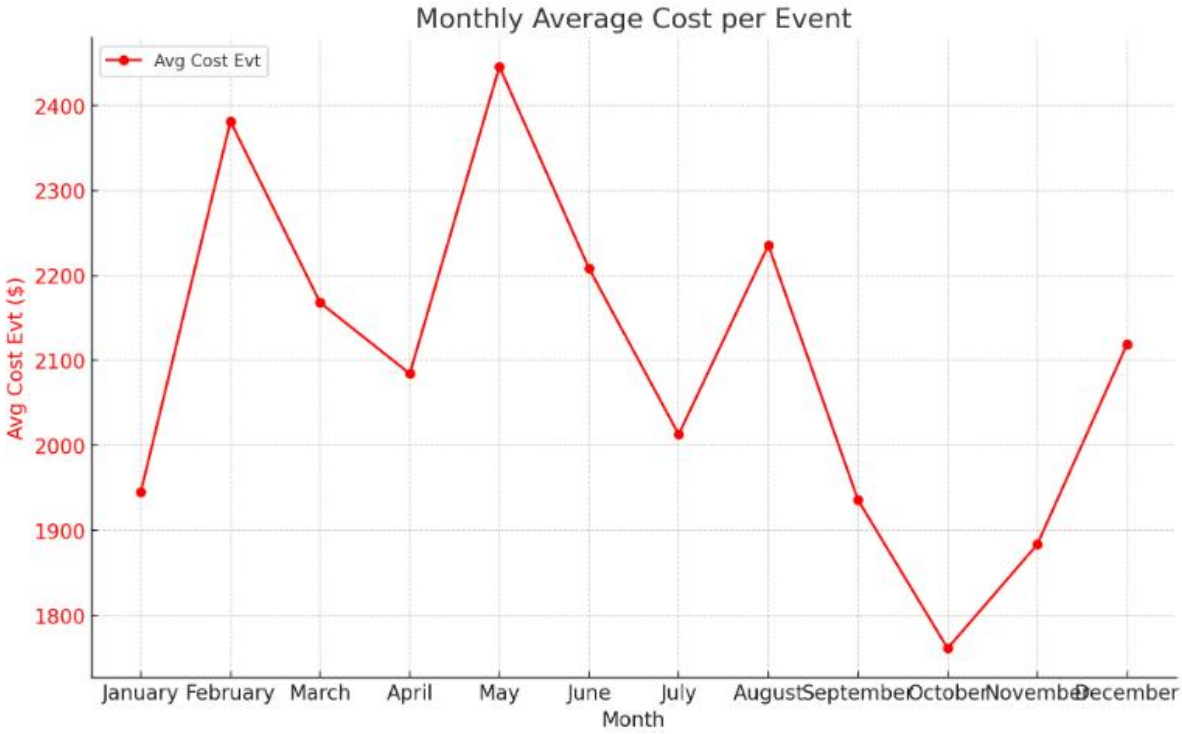
```
SET LINESIZE 85
SET PAGESIZE 50
CLEAR COLUMNS
-- Define column formats
COLUMN aircraft_id FORMAT A8 HEADING 'Aircraft'
COLUMN maint_type FORMAT A15 HEADING 'Maint Type'
COLUMN CAL_MONTH_NAME FORMAT A15 HEADING 'Mon'
COLUMN total_events FORMAT 999 HEADING 'Events'
COLUMN total_cost FORMAT $999,999.99 HEADING 'Total Cost'
COLUMN total_hours FORMAT 999.99 HEADING 'Maint Hrs'
COLUMN avg_cost_per_event FORMAT $999,999.99 HEADING 'Avg Cost'
COLUMN events_change FORMAT 999 HEADING 'Evt+/-'
COLUMN cost_change_pct FORMAT 990.9 HEADING 'Cost+/- %'
-- Prompt for inputs
ACCEPT rpt_year NUMBER PROMPT 'Enter targeted year (YYYY): '
ACCEPT rpt_aircraft CHAR PROMPT 'Enter targeted Aircraft ID: '
ACCEPT rpt_maint_type CHAR PROMPT 'Enter targeted Maintenance Type: '
-- Report title
TTITLE CENTER 'Monthly Maintenance Cost And Events Report' SKIP 1 -
       CENTER 'Year: &rpt_year' SKIP 1 -
       CENTER 'Aircraft: &rpt_aircraft   |   Type: &rpt_maint_type' SKIP 2
-- Create CTE for monthly maintenance data
WITH monthly_data AS (
  SELECT
    ad.aircraft_id,
    mc.maintenance_list_type AS maint_type,
    TO_NUMBER(SUBSTR(dd.date_key, 5, 2)) AS month_no,
    dd.CAL_MONTH_NAME,
    COUNT(*) AS total_events,
    SUM(mc.total_cost) AS total_cost,
    SUM(mc.maintenance_hours) AS total_hours
  FROM maintenance_cost mc
  JOIN aircraft_dim ad ON mc.aircraft_key = ad.aircraft_key
  JOIN date_dim dd ON mc.date_key = dd.date_key
  WHERE dd.cal_year = &rpt_year
    AND UPPER(ad.aircraft_id) = UPPER('&rpt_aircraft')
    AND UPPER(mc.maintenance_list_type) = UPPER('&rpt_maint_type')
  GROUP BY
    ad.aircraft_id,
    mc.maintenance_list_type,
    TO_NUMBER(SUBSTR(dd.date_key, 5, 2)),
    dd.CAL_MONTH_NAME
)
SELECT
  md.CAL_MONTH_NAME,
  md.total_events,
  md.total_cost,
  md.total_hours,
  CASE
    WHEN md.total_events = 0 THEN NULL
    ELSE md.total_cost / md.total_events
  END AS avg_cost_per_event,
  -- Calculate events change (current month - previous month)
  md.total_events - LAG(md.total_events, 1, 0) OVER (
    ORDER BY
    CASE md.CAL_MONTH_NAME
      WHEN 'January' THEN 1
      WHEN 'February' THEN 2
      WHEN 'March' THEN 3
```

```
    WHEN 'April' THEN 4
    WHEN 'May' THEN 5
    WHEN 'June' THEN 6
    WHEN 'July' THEN 7
    WHEN 'August' THEN 8
    WHEN 'September' THEN 9
    WHEN 'October' THEN 10
    WHEN 'November' THEN 11
    WHEN 'December' THEN 12
  END
) AS events_change,
-- Calculate cost change percentage
CASE
  WHEN LAG(md.total_cost, 1, NULL) OVER (
    ORDER BY
    CASE md.CAL_MONTH_NAME
      WHEN 'January' THEN 1
      WHEN 'February' THEN 2
      WHEN 'March' THEN 3
      WHEN 'April' THEN 4
      WHEN 'May' THEN 5
      WHEN 'June' THEN 6
      WHEN 'July' THEN 7
      WHEN 'August' THEN 8
      WHEN 'September' THEN 9
      WHEN 'October' THEN 10
      WHEN 'November' THEN 11
      WHEN 'December' THEN 12
    END
  ) IS NULL
      OR LAG(md.total_cost, 1, 0) OVER (
        ORDER BY
        CASE md.CAL_MONTH_NAME
          WHEN 'January' THEN 1
          WHEN 'February' THEN 2
          WHEN 'March' THEN 3
          WHEN 'April' THEN 4
          WHEN 'May' THEN 5
          WHEN 'June' THEN 6
          WHEN 'July' THEN 7
          WHEN 'August' THEN 8
          WHEN 'September' THEN 9
          WHEN 'October' THEN 10
          WHEN 'November' THEN 11
          WHEN 'December' THEN 12
        END
      ) = 0 THEN NULL
  ELSE ROUND(((md.total_cost - LAG(md.total_cost, 1, 0) OVER (
        ORDER BY
        CASE md.CAL_MONTH_NAME
          WHEN 'January' THEN 1
          WHEN 'February' THEN 2
          WHEN 'March' THEN 3
          WHEN 'April' THEN 4
          WHEN 'May' THEN 5
          WHEN 'June' THEN 6
          WHEN 'July' THEN 7
          WHEN 'August' THEN 8
          WHEN 'September' THEN 9
          WHEN 'October' THEN 10
          WHEN 'November' THEN 11
          WHEN 'December' THEN 12
```

```
          END
        ))
        / NULLIF(LAG(md.total_cost, 1, 0) OVER (
          ORDER BY
          CASE md.CAL_MONTH_NAME
            WHEN 'January' THEN 1
            WHEN 'February' THEN 2
            WHEN 'March' THEN 3
            WHEN 'April' THEN 4
            WHEN 'May' THEN 5
            WHEN 'June' THEN 6
            WHEN 'July' THEN 7
            WHEN 'August' THEN 8
            WHEN 'September' THEN 9
            WHEN 'October' THEN 10
            WHEN 'November' THEN 11
            WHEN 'December' THEN 12
          END
        ), 0)) * 100, 1)
  END AS cost_change_pct
FROM monthly_data md
ORDER BY
  CASE md.CAL_MONTH_NAME
    WHEN 'January' THEN 1
    WHEN 'February' THEN 2
    WHEN 'March' THEN 3
    WHEN 'April' THEN 4
    WHEN 'May' THEN 5
    WHEN 'June' THEN 6
    WHEN 'July' THEN 7
    WHEN 'August' THEN 8
    WHEN 'September' THEN 9
    WHEN 'October' THEN 10
    WHEN 'November' THEN 11
    WHEN 'December' THEN 12
  END;
TTITLE OFF;
```

```
SQL> start c:\q2.txt;
Enter targeted year (YYYY): 2024
Enter targeted Aircraft ID: AC001
Enter targeted Maintenance Type: heavy maintenance
old  13:   WHERE dd.cal_year = &rpt_year
new  13:   WHERE dd.cal_year =       2024
old  14:     AND UPPER(ad.aircraft_id) = UPPER('&rpt_aircraft')
new  14:     AND UPPER(ad.aircraft_id) = UPPER('AC001')
old  15:     AND UPPER(mc.maintenance_list_type) = UPPER('&rpt_maint_type')
new  15:     AND UPPER(mc.maintenance_list_type) = UPPER('heavy maintenance')

                    Monthly Maintenance Cost And Events Report
                                  Year:       2024
                  Aircraft: AC001   |   Type: heavy maintenance

Mon             Events   Total Cost Maint Hrs     Avg Cost Evt+/- Cost+/- %
--------------- ------ ------------ --------- ------------ ------ ---------
January             23   $44,740.56     26.37    $1,945.24     23
February            10   $23,812.87     10.33    $2,381.29    -13     -46.8
March               11   $23,848.72     14.03    $2,168.07      1       0.2
April               27   $56,277.69     29.31    $2,084.36     16     136.0
May                 20   $48,915.52     21.59    $2,445.78     -7     -13.1
June                14   $30,913.66     15.65    $2,208.12     -6     -36.8
July                14   $28,180.84     12.14    $2,012.92      0      -8.8
August              21   $46,948.23     21.66    $2,235.63      7      66.6
September           16   $30,970.88     18.99    $1,935.68     -5     -34.0
October             23   $40,504.66     21.02    $1,761.07      7      30.8
November            14   $26,367.41     10.16    $1,883.39     -9     -34.9
December            10   $21,188.73     12.32    $2,118.87     -4     -19.6

12 rows selected.
```





Optimize Cost Management for Complex Maintenance: February saw fewer events (10) but higher costs, with an average cost per event of $2,381.29. This likely reflects more complex tasks or higher material costs. A detailed review of these events can help identify areas for cost savings, such as renegotiating

supplier contracts, sourcing less expensive parts, or finding more efficient methods to complete tasks. This will help reduce the impact of complex maintenance on the overall budget.

Streamline Efficiency in High-Cost Months: August experienced a significant spike in cost per event, increasing by 66.6% to $2,235.63. This indicates that the complexity of tasks during this month drove up costs. To improve efficiency, it is recommended to analyze these tasks closely and identify opportunities for streamlining maintenance processes, using more cost-effective materials, or optimizing workflows to reduce unnecessary costs.

Implement Predictive Maintenance: To better manage future workloads, predictive maintenance strategies should be adopted. By analyzing historical data, such as spikes in events and costs in April and January, the team can forecast high-demand periods and plan resource allocation more effectively. This proactive approach will help prevent unplanned downtime and ensure that the necessary resources are in place before peak periods occur.

3.1.3 Maintenance Type Analysis In Three Year

```
SET LINESIZE 132
SET PAGESIZE 50

-- Column formatting for costs, YOY changes, and percentage of total
COLUMN "Maintenance Type"      FORMAT A20
COLUMN "2022"                  FORMAT $99,999,999.99
COLUMN "2023"                  FORMAT $99,999,999.99
COLUMN "2024"                  FORMAT $99,999,999.99
COLUMN "Change 22-23 (%)"      FORMAT 9,999.99
COLUMN "Change 23-24 (%)"      FORMAT 9,999.99
COLUMN "2022 (%)"              FORMAT 9,999.99
COLUMN "2023 (%)"              FORMAT 9,999.99
COLUMN "2024 (%)"              FORMAT 9,999.99

-- Header
TTITLE CENTER 'Airport Maintenance Report: Costs, YOY Change, and % of Total for
Three Year' -
       SKIP 1

WITH
  annual AS (
    SELECT
      mc.maintenance_list_type AS maintenance_type,
      dd.cal_year              AS yr,
      SUM(mc.total_cost)       AS cost
    FROM maintenance_cost mc
    JOIN date_dim       dd ON mc.date_key = dd.date_key
    -- include all years for comparison
    GROUP BY mc.maintenance_list_type, dd.cal_year
  ),
  year_totals AS (
    SELECT
      dd.cal_year        AS yr,
      SUM(mc.total_cost) AS total_cost
    FROM maintenance_cost mc
    JOIN date_dim       dd ON mc.date_key = dd.date_key
    GROUP BY dd.cal_year
  ),
  pct_data AS (
    SELECT
      a.maintenance_type,
      a.yr,
      ROUND(a.cost / NULLIF(y.total_cost,0) * 100, 2) AS pct
    FROM annual a
    JOIN year_totals y ON a.yr = y.yr
  ),
  cost_pivot AS (
    SELECT * FROM annual
    PIVOT ( SUM(cost) FOR yr IN (2022 AS "2022", 2023 AS "2023", 2024 AS "2024") )
  ),
  pct_pivot AS (
    SELECT * FROM pct_data
    PIVOT ( MAX(pct) FOR yr IN (2022 AS "2022 (%)", 2023 AS "2023 (%)", 2024 AS
"2024 (%)") )
  ),
  combined AS (
    SELECT
      c.maintenance_type,
      c."2022", c."2023", c."2024",
```

```
      p."2022 (%)" AS pct22,
      p."2023 (%)" AS pct23,
      p."2024 (%)" AS pct24
    FROM cost_pivot c
    JOIN pct_pivot p ON c.maintenance_type = p.maintenance_type
  )
SELECT *
FROM (
  -- Detailed rows
  SELECT
    maintenance_type      AS "Maintenance Type",
    TO_CHAR("2022", '$999,999,999.99')     AS "2022",
    TO_CHAR("2023", '$999,999,999.99')     AS "2023",
    TO_CHAR("2024", '$999,999,999.99')     AS "2024",
    TO_CHAR(ROUND(("2023"-"2022")/NULLIF("2022",0)*100,2),'999.99') AS "Change 22-23
(%)",
    TO_CHAR(ROUND(("2024"-"2023")/NULLIF("2023",0)*100,2),'999.99') AS "Change 23-24
(%)",
    TO_CHAR(pct22,'999.99')                AS "2022 (%)",
    TO_CHAR(pct23,'999.99')                AS "2023 (%)",
    TO_CHAR(pct24,'999.99')                AS "2024 (%)"
  FROM combined
  UNION ALL
  -- Separator line above total
  SELECT
    '--------------------'              AS "Maintenance Type",
    '-----------------'                  AS "2022",
    '-----------------'                  AS "2023",
    '-----------------'                  AS "2024",
    '-----------------'                  AS "Change 22-23 (%)",
    '-----------------'                  AS "Change 23-24 (%)",
    '-------'                             AS "2022 (%)",
    '-------'                             AS "2023 (%)",
    '-------'                             AS "2024 (%)"
  FROM dual
  UNION ALL
  -- Total row by year
  SELECT
    'Total'                             AS "Maintenance Type",
    TO_CHAR(SUM("2022"), '$999,999,999.99')  AS "2022",
    TO_CHAR(SUM("2023"), '$999,999,999.99')  AS "2023",
    TO_CHAR(SUM("2024"), '$999,999,999.99')  AS "2024",
    NULL                                 AS "Change 22-23 (%)",
    NULL                                 AS "Change 23-24 (%)",
    TO_CHAR(100,'999.99')               AS "2022 (%)",
    TO_CHAR(100,'999.99')               AS "2023 (%)",
    TO_CHAR(100,'999.99')               AS "2024 (%)"
  FROM combined
)
ORDER BY
  CASE WHEN "Maintenance Type" = '--------------------' THEN 1
       WHEN "Maintenance Type" = 'Total' THEN 2
       ELSE 0 END,
  "Maintenance Type";

TTITLE OFF
```

```
SQL> start c:\q1.txt;

                                Airport Maintenance Report: Costs, YOY Change, and % of Total for Three Year
Maintenance Type     2022             2023             2024             Change 22-23 (%) Change 23-24 (%) 2022 (% 2023 (% 2024 (%
-------------------- ---------------- ---------------- ---------------- ---------------- ---------------- ------- ------- -------
heavy maintenance      $90,685,341.52   $91,305,599.05   $90,653,596.90      .68              -.71            50.27   50.65   50.26
line maintenance       $24,798,287.65   $24,731,275.52   $24,884,255.00     -.27               .62            13.75   13.72   13.80
part of maintenance    $40,516,212.56   $40,086,566.86   $40,260,226.37    -1.06               .43            22.46   22.24   22.32
repair maintenance     $24,395,224.88   $24,139,922.87   $24,559,500.11    -1.05              1.74            13.52   13.39   13.62
-------------------- ---------------- ---------------- ---------------- ---------------- ---------------- ------- ------- -------
Total                $180,395,066.61  $180,263,364.30  $180,357,578.38                                     100.00  100.00  100.00

6 rows selected.
```

Based on the graph, heavy maintenance remains the dominant cost driver from 2022 to 2024, consistently consuming over 50% of the total annual budget (around 90 million USD). Though slightly reduced in 2024 by 0.71%, this figure still indicates a significant opportunity for cost optimization. Locking in a firm-volume, multi-year contract at today's rates could cap future price increases and stabilize what is now over half of your annual maintenance outlay. Additionally, implementing predictive maintenance technologies and scheduling major checks during off-peak seasons can reduce unplanned downtime and lower long-term costs. Meanwhile, repair maintenance, although the smallest category, rose by 1.74% in 2024—the largest year-over-year growth. This suggests increasing equipment failures, which may result from deferring preventive work. Airports should consider strengthening their inspection routines and technician training to prevent costly reactive repairs. Line and part maintenance costs remain stable, which is positive, but still require continuous monitoring. Standardizing frequently used parts and consolidating procurement could drive further savings.

3.2 Wong Kiong Wei

3.2.1 Comparison of Maintenance Items: 2023 vs 2024

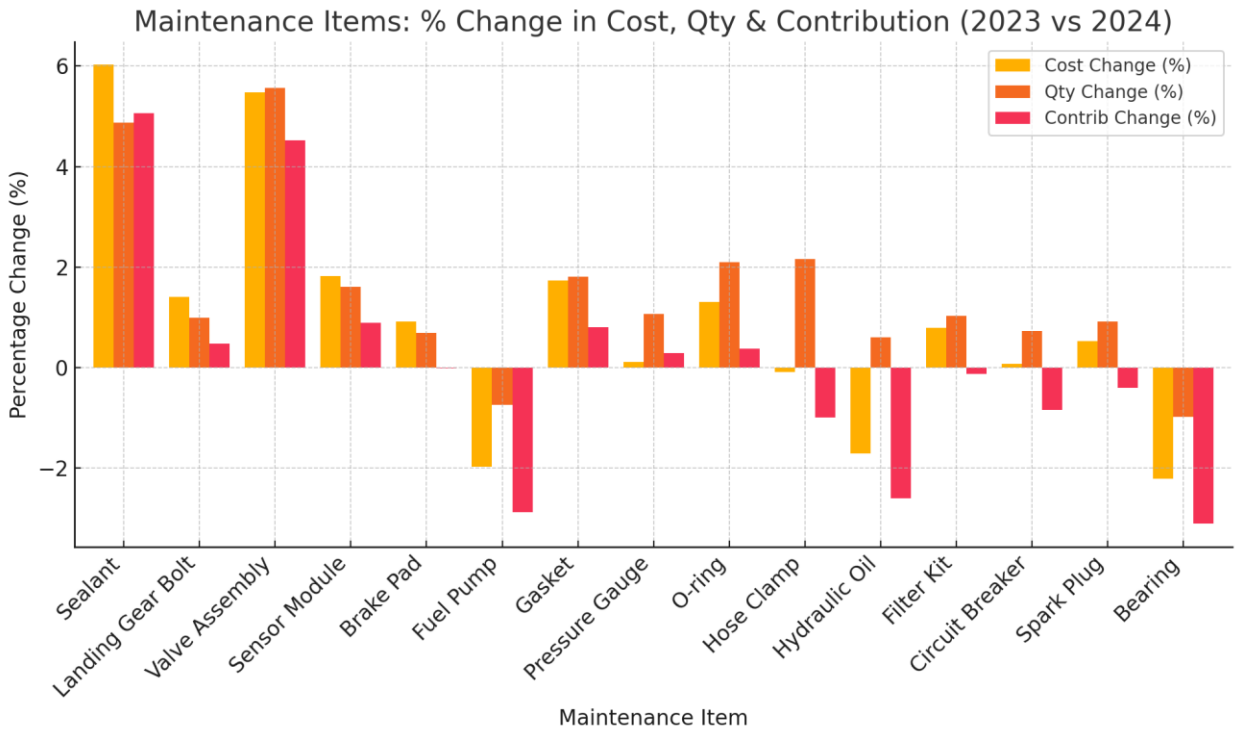
```
SET PAGESIZE 50
SET LINESIZE 132
SET NUMWIDTH 12
SET VERIFY OFF
CLEAR COLUMNS

-- Prompt for years
ACCEPT P_START_YEAR CHAR FORMAT '9999' PROMPT 'Enter Start Year (YYYY): '
ACCEPT P_END_YEAR   CHAR FORMAT '9999' PROMPT 'Enter End Year   (YYYY): '

COLUMN "Item Names"             FORMAT A17
COLUMN "Cost &P_START_YEAR"     FORMAT 999,999,999.99 HEADING "&P_START_YEAR Cost"
COLUMN "Qty &P_START_YEAR"      FORMAT 999,999        HEADING "&P_START_YEAR Qty"
COLUMN "Avg &P_START_YEAR"      FORMAT 999,999.99     HEADING "&P_START_YEAR Avg"
COLUMN "Cost &P_END_YEAR"       FORMAT 999,999,999.99 HEADING "&P_END_YEAR Cost"
COLUMN "Qty &P_END_YEAR"        FORMAT 999,999        HEADING "&P_END_YEAR Qty"
COLUMN "Avg &P_END_YEAR"        FORMAT 999,999.99     HEADING "&P_END_YEAR Avg"
COLUMN "Cost Change(%)"         FORMAT 999.99              HEADING "Cost |
Change(%)"
COLUMN "Qty Change(%)"          FORMAT 999.99         HEADING "Qty | Change(%)"
COLUMN "Contr Change(%)"        FORMAT 999.99         HEADING "CONTRIB | Change(%)"

TTITLE CENTER 'Maintenance Cost Analysis Report' SKIP 1 -
       CENTER 'Comparison of Maintenance Items: &P_START_YEAR vs &P_END_YEAR' SKIP 1
-
       CENTER 'All Airports | Years: &P_START_YEAR vs &P_END_YEAR' SKIP 2

WITH data_start AS (
  SELECT
    MC.ITEM_NAME,
    COALESCE(SUM(MC.TOTAL_ITEMPRICE), 0) AS Price_start,
    COALESCE(SUM(MC.TOTAL_QUANTITY),  0) AS Qty_start
  FROM MAINTENANCE_COST MC
  JOIN DATE_DIM D ON MC.DATE_KEY = D.DATE_KEY
  WHERE D.CAL_YEAR = &P_START_YEAR
  GROUP BY MC.ITEM_NAME
),
data_end AS (
  SELECT
    MC.ITEM_NAME,
    COALESCE(SUM(MC.TOTAL_ITEMPRICE), 0) AS Price_end,
    COALESCE(SUM(MC.TOTAL_QUANTITY),  0) AS Qty_end
  FROM MAINTENANCE_COST MC
  JOIN DATE_DIM D ON MC.DATE_KEY = D.DATE_KEY
  WHERE D.CAL_YEAR = &P_END_YEAR
  GROUP BY MC.ITEM_NAME
),
total_start AS (
  SELECT COALESCE(SUM(MC.TOTAL_ITEMPRICE),0) AS Total_start
  FROM MAINTENANCE_COST MC
  JOIN DATE_DIM D ON MC.DATE_KEY = D.DATE_KEY
  WHERE D.CAL_YEAR = &P_START_YEAR
),
total_end AS (
  SELECT COALESCE(SUM(MC.TOTAL_ITEMPRICE),0) AS Total_end
  FROM MAINTENANCE_COST MC
```

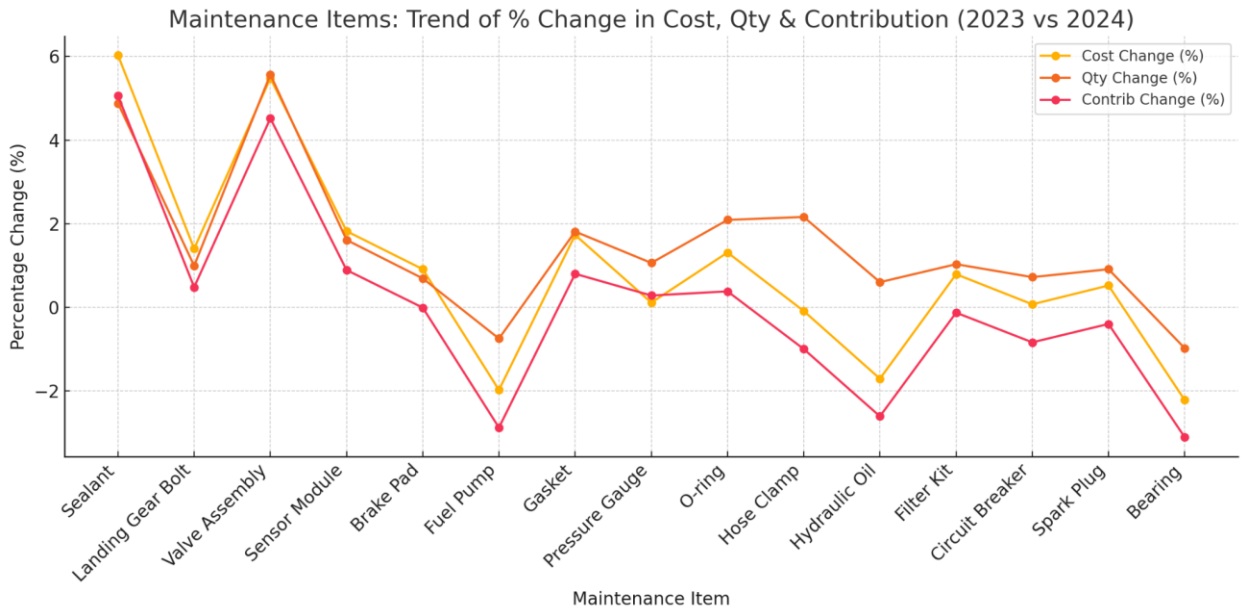
```
  JOIN DATE_DIM D ON MC.DATE_KEY = D.DATE_KEY
  WHERE D.CAL_YEAR = &P_END_YEAR
),
combined AS (
  SELECT
    COALESCE(ds.ITEM_NAME, de.ITEM_NAME)                      AS ITEM_NAME,
    NVL(ds.Price_start,0)                                      AS Price_start,
    NVL(ds.Qty_start,0)                                        AS Qty_start,
    CASE WHEN NVL(ds.Qty_start,0)=0 THEN 0
         ELSE ROUND(NVL(ds.Price_start,0)/ds.Qty_start,2) END AS Avg_start,
    NVL(de.Price_end,0)                                        AS Price_end,
    NVL(de.Qty_end,0)                                          AS Qty_end,
    CASE WHEN NVL(de.Qty_end,0)=0 THEN 0
         ELSE ROUND(NVL(de.Price_end,0)/de.Qty_end,2)   END AS Avg_end,
    ts.Total_start,
    te.Total_end
  FROM data_start ds
  FULL OUTER JOIN data_end de ON ds.ITEM_NAME = de.ITEM_NAME
  CROSS JOIN total_start ts
  CROSS JOIN total_end   te
  WHERE NVL(ds.Price_start,0)>0 OR NVL(de.Price_end,0)>0
),
rolled AS (
  SELECT
    CASE WHEN GROUPING(ITEM_NAME)=1 THEN 'GRAND TOTAL' ELSE ITEM_NAME END AS
ITEM_NAME,
    SUM(Price_start)                                           AS Price_start,
    SUM(Qty_start)                                             AS Qty_start,
    CASE WHEN SUM(Qty_start)=0 THEN 0
         ELSE ROUND(SUM(Price_start)/SUM(Qty_start),2) END      AS Avg_start,
    SUM(Price_end)                                             AS Price_end,
    SUM(Qty_end)                                               AS Qty_end,
    CASE WHEN SUM(Qty_end)=0 THEN 0
         ELSE ROUND(SUM(Price_end)/SUM(Qty_end),2)   END       AS Avg_end,
    CASE WHEN SUM(Price_start)=0 THEN NULL
         ELSE ROUND(((SUM(Price_end)-SUM(Price_start))
                /SUM(Price_start))*100,2) END                  AS Cost_Change,
    CASE WHEN SUM(Qty_start)=0 THEN NULL
         ELSE ROUND(((SUM(Qty_end)-SUM(Qty_start))
                /SUM(Qty_start))*100,2) END                    AS Qty_Change,
    CASE WHEN GROUPING(ITEM_NAME)=1 THEN 0
         ELSE ROUND(
           (
             (SUM(Price_end)/MAX(Total_end))
             - (SUM(Price_start)/MAX(Total_start))
           )
           / (SUM(Price_start)/MAX(Total_start))
           * 100
         ,2) END                                               AS Contr_Change,
    GROUPING(ITEM_NAME)                                        AS Is_Total
  FROM combined
  GROUP BY ROLLUP(ITEM_NAME)
)
SELECT
  ITEM_NAME               AS "Item Names",
  Price_start             AS "Cost &P_START_YEAR",
  Qty_start               AS "Qty &P_START_YEAR",
  Avg_start               AS "Avg &P_START_YEAR",
  Price_end               AS "Cost &P_END_YEAR",
  Qty_end                 AS "Qty &P_END_YEAR",
  Avg_end                 AS "Avg &P_END_YEAR",
  Cost_Change             AS "Cost Change(%)",
```

```
  Qty_Change                AS "Qty Change(%)",
  Contr_Change              AS "Contr Change(%)"
FROM rolled
ORDER BY Is_Total, Price_end DESC;
```

Maintenance Cost Analysis Report
Comparison of Maintenance Items: 2023 vs 2024
All Airports | Years: 2023 vs 2024

Item Names	2023 Cost	2023 Qty	2023 Avg	2024 Cost	2024 Qty	2024 Avg	Cost Change(%)	Qty Change(%)	CONTRIB Change(%)
Sealant	5,776,518.43	22,727	254.17	6,124,760.69	23,835	256.96	6.03	4.88	5.06
Landing Gear Bolt	5,903,482.74	23,016	256.49	5,986,166.42	23,244	257.54	1.40	.99	.48
Valve Assembly	5,636,288.00	22,203	253.85	5,945,437.89	23,440	253.64	5.48	5.57	4.52
Sensor Module	5,832,645.32	22,926	254.41	5,938,608.29	23,295	254.93	1.82	1.61	.89
Brake Pad	5,853,804.36	23,010	254.40	5,907,199.84	23,169	254.96	.91	.69	-.01
Fuel Pump	6,011,953.42	23,452	256.35	5,892,691.52	23,276	253.17	-1.98	-.75	-2.88
Gasket	5,775,861.05	22,558	256.04	5,875,634.13	22,967	255.83	1.73	1.81	.80
Pressure Gauge	5,860,356.85	22,983	254.99	5,866,738.15	23,227	252.58	.11	1.06	-.80
O-ring	5,776,013.73	22,482	256.92	5,851,476.94	22,951	254.96	1.31	2.09	.38
Hose Clamp	5,849,577.86	22,637	258.41	5,844,277.74	23,127	252.70	-.09	2.16	-1.00
Hydraulic Oil	5,918,755.59	23,063	256.63	5,817,404.02	23,201	250.74	-1.71	.60	-2.61
Filter Kit	5,752,004.27	22,601	254.50	5,797,603.61	22,833	253.91	.79	1.03	-.13
Circuit Breaker	5,784,875.34	22,897	252.65	5,788,966.45	23,063	251.01	.07	.72	-.84
Spark Plug	5,757,410.19	22,589	254.88	5,787,236.19	22,794	253.89	.52	.91	-.40
Bearing	5,867,670.83	22,730	258.15	5,737,494.41	22,508	254.91	-2.22	-.98	-3.11
GRAND TOTAL	87,357,217.98	341,874	255.52	88,161,696.29	346,930	254.12	.92	1.48	.00





From the bar chart and line chart, it can be seen that the cost of Sealant and Valve Assembly increased by approximately 6.0% and 5.5% respectively, while their quantities also increased by 4.9% and 5.6%, indicating that the market demand and procurement cost for these two components are rising simultaneously. In the next budget cycle, special attention should be paid to negotiating the unit purchase price. The quantity increases of Filter Kit and Circuit Breaker (1.0% and 0.7%) exceed their cost increases (0.8% and 0.1%), suggesting that inventory occupation may be increasing. Based on this, the safety stock level should be adjusted to avoid long-term capital lock-up. Although the costs of Fuel Pump and Hydraulic Oil dropped by 2.0% and 1.7% respectively, their quantity changes show divergent trends, with one decreasing (–0.8%) and the other increasing (0.6%). It is recommended to continue with the current contracts but strengthen demand forecasting to prevent oversupply or shortage. Lastly, both the cost and quantity of Bearing declined (–2.2% and –1.0%), and the supplier's price reduction measures and delivery quality should be strictly reviewed in the next quarter.

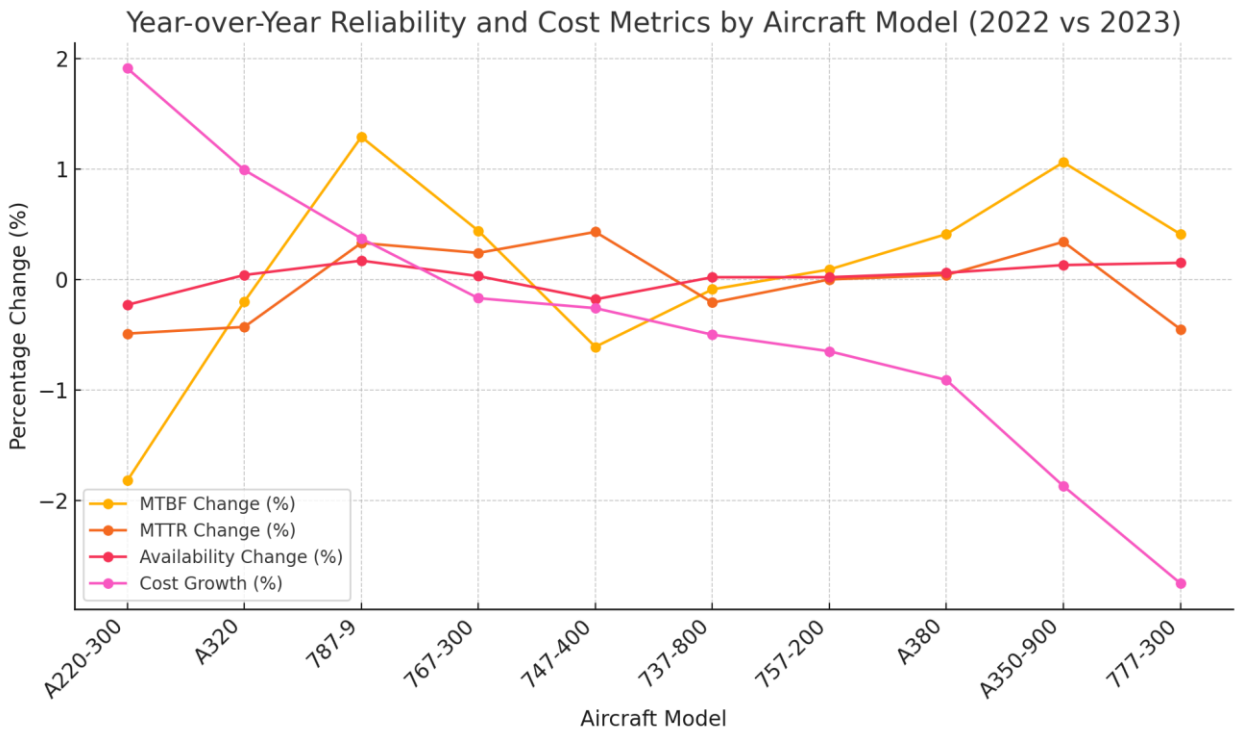
3.2.2 MTBF (Mean Time Between Failures), MTTR (Mean Time To Repair), Availability, and Maintenance Cost Trends by Aircraft Model (2022 vs 2023).

```
SET PAGESIZE 50
SET LINESIZE 132
SET NUMWIDTH 12
SET VERIFY OFF
CLEAR COLUMNS

-- Prompt for start and end years
ACCEPT P_START_YEAR CHAR FORMAT '9999' PROMPT 'Enter Start Year (YYYY): '
ACCEPT P_END_YEAR   CHAR FORMAT '9999' PROMPT 'Enter End Year   (YYYY): '

COLUMN "Aircraft Model"            FORMAT A15 HEADING "Model"
COLUMN "MTBF &P_START_YEAR"        FORMAT 9.99 HEADING "&P_START_YEAR|MTBF"
COLUMN "MTBF &P_END_YEAR"          FORMAT 9.99 HEADING "&P_END_YEAR|MTBF"
COLUMN "MTTR &P_START_YEAR"        FORMAT 999.99 HEADING "&P_START_YEAR|MTTR"
COLUMN "MTTR &P_END_YEAR"          FORMAT 999.99 HEADING "&P_END_YEAR|MTTR"
COLUMN "Avail(%) &P_START_YEAR"    FORMAT 999.99 HEADING "&P_START_YEAR|Avail(%)"
COLUMN "Avail(%) &P_END_YEAR"      FORMAT 999.99 HEADING "&P_END_YEAR|Avail(%)"
COLUMN "Cost &P_START_YEAR"        FORMAT 99,999,999 HEADING "&P_START_YEAR|Total
Cost"
COLUMN "Cost &P_END_YEAR"          FORMAT 99,999,999 HEADING "&P_END_YEAR|Total
Cost"
COLUMN "MTBF Change(%)"            FORMAT 999.99 HEADING "MTBF|Change(%)"
COLUMN "MTTR Change(%)"            FORMAT 999.99 HEADING "MTTR|Change(%)"
COLUMN "Avail Change(%)"           FORMAT 999.99 HEADING "Avail|Change(%)"
COLUMN "Cost Growth(%)"            FORMAT 999.99 HEADING "Cost|Growth(%)"

TTITLE CENTER 'Aircraft Model Reliability and Cost Metrics Comparison' SKIP 1 -
       CENTER 'Year-over-Year Analysis: &P_START_YEAR vs &P_END_YEAR' SKIP 1 -
       CENTER 'MTBF, MTTR, Availability, and Maintenance Cost Trends by Aircraft
Model' SKIP 2

WITH time_intervals AS (
  SELECT
    a.AIRCRAFT_MODEL,
    a.AIRCRAFT_ID,
    d.CAL_YEAR                         AS year,
    mc.START_DATETIME,
    mc.END_DATETIME,
    mc.MAINTENANCE_HOURS               AS maintenance_time,
    mc.AIRCRAFT_FLIGHT_HOUR            AS flight_hours,
    mc.TOTAL_COST                      AS event_cost,
    -- time between current event and previous event, in hours
    (EXTRACT(DAY    FROM (mc.START_DATETIME - LAG(mc.END_DATETIME) OVER
         (PARTITION BY a.AIRCRAFT_ID, d.CAL_YEAR ORDER BY mc.START_DATETIME))) * 24
     + EXTRACT(HOUR  FROM (mc.START_DATETIME - LAG(mc.END_DATETIME) OVER
         (PARTITION BY a.AIRCRAFT_ID, d.CAL_YEAR ORDER BY mc.START_DATETIME)))
     + EXTRACT(MINUTE FROM (mc.START_DATETIME - LAG(mc.END_DATETIME) OVER
         (PARTITION BY a.AIRCRAFT_ID, d.CAL_YEAR ORDER BY mc.START_DATETIME))) / 60
    ) AS time_between_events
  FROM MAINTENANCE_COST mc
  JOIN AIRCRAFT_DIM   a ON mc.AIRCRAFT_KEY = a.AIRCRAFT_KEY
  JOIN DATE_DIM       d ON mc.DATE_KEY     = d.DATE_KEY
  WHERE d.CAL_YEAR IN (&P_START_YEAR, &P_END_YEAR)
),
yearly_metrics AS (
  SELECT
    AIRCRAFT_MODEL,
```

```
    year,
    COUNT(DISTINCT AIRCRAFT_ID)                                          AS aircraft_count,
    COUNT(*)                                                              AS total_events,
    AVG(NVL(time_between_events, 0))                                      AS MTBF,
    AVG(maintenance_time)                                                 AS MTTR,
    (SUM(NVL(time_between_events, 0)) /
     NULLIF(SUM(NVL(time_between_events, 0)) + SUM(maintenance_time), 0)
    ) * 100                                                               AS availability,
    SUM(event_cost)                                                       AS total_cost
  FROM time_intervals
  GROUP BY AIRCRAFT_MODEL, year
),
metrics_start AS (
  SELECT * FROM yearly_metrics WHERE year = &P_START_YEAR
),
metrics_end AS (
  SELECT * FROM yearly_metrics WHERE year = &P_END_YEAR
)
SELECT
  COALESCE(ms.AIRCRAFT_MODEL, me.AIRCRAFT_MODEL)              AS "Aircraft Model",
  ROUND(NVL(ms.MTBF,        0), 2)                           AS "MTBF
&P_START_YEAR",
  ROUND(NVL(ms.MTTR,        0), 2)                           AS "MTTR
&P_START_YEAR",
  ROUND(NVL(ms.availability,0), 2)                           AS "Avail(%)
&P_START_YEAR",
  NVL(ms.total_cost,        0)                               AS "Cost
&P_START_YEAR",
  ROUND(NVL(me.MTBF,        0), 2)                           AS "MTBF &P_END_YEAR",
  ROUND(NVL(me.MTTR,        0), 2)                           AS "MTTR &P_END_YEAR",
  ROUND(NVL(me.availability,0), 2)                           AS "Avail(%)
&P_END_YEAR",
  NVL(me.total_cost,        0)                               AS "Cost &P_END_YEAR",
  CASE WHEN NVL(ms.MTBF,0)=0 THEN NULL
       ELSE ROUND(((NVL(me.MTBF,0)-NVL(ms.MTBF,0))/NVL(ms.MTBF,0))*100,2)
  END                                                         AS "MTBF Change(%)",
  CASE WHEN NVL(ms.MTTR,0)=0 THEN NULL
       ELSE ROUND(((NVL(me.MTTR,0)-NVL(ms.MTTR,0))/NVL(ms.MTTR,0))*100,2)
  END                                                         AS "MTTR Change(%)",
  CASE WHEN NVL(ms.availability,0)=0 THEN NULL
       ELSE ROUND(((NVL(me.availability,0)-
NVL(ms.availability,0))/NVL(ms.availability,0))*100,2)
  END                                                         AS "Avail Change(%)",
  CASE WHEN NVL(ms.total_cost,0)=0 THEN NULL
       ELSE ROUND(((NVL(me.total_cost,0)-
NVL(ms.total_cost,0))/NVL(ms.total_cost,0))*100,2)
  END                                                         AS "Cost Growth(%)"
FROM   metrics_start ms
FULL   JOIN metrics_end   me ON ms.AIRCRAFT_MODEL = me.AIRCRAFT_MODEL
ORDER  BY "Cost Growth(%)" DESC, "Avail Change(%)" ASC;
```

Aircraft Model Reliability and Cost Metrics Comparison
Year-over-Year Analysis: 2022 vs 2023
MTBF, MTTR, Availability, and Maintenance Cost Trends by Aircraft Model

Model	2022 MTBF	2022 MTTR	2022 Avail(%)	2022 Total Cost	2023 MTBF	2023 MTTR	2023 Avail(%)	2023 Total Cost	MTBF Change(%)	MTTR Change(%)	Avail Change(%)	Cost Growth(%)
Airbus A220-300	5.18	1.09	82.66	18,839,070	5.09	1.08	82.47	19,198,097	-1.82	-.49	-.23	1.91
Airbus A320	5.12	1.08	82.51	23,268,761	5.11	1.08	82.54	23,499,593	-.20	-.43	.04	.99
Boeing 787-9	5.09	1.09	82.42	11,675,731	5.16	1.09	82.55	11,719,321	1.29	.33	.17	.37
Boeing 767-300	5.14	1.08	82.62	19,579,558	5.17	1.09	82.64	19,545,658	.44	.24	.03	-.17
Boeing 747-400	5.13	1.08	82.57	23,408,023	5.10	1.09	82.42	23,347,320	-.61	.43	-.18	-.26
Boeing 737-800	5.17	1.08	82.70	16,193,770	5.16	1.08	82.71	16,113,084	-.09	-.21	.02	-.50
Boeing 757-200	5.12	1.08	82.53	18,961,992	5.13	1.08	82.54	18,837,914	.09	.00	.02	-.65
Airbus A380	5.14	1.08	82.62	15,287,125	5.16	1.08	82.68	15,147,578	.41	.04	.06	-.91
Airbus A350-900	5.07	1.08	82.43	14,440,707	5.13	1.08	82.53	14,170,865	1.06	.34	.13	-1.87
Boeing 777-300	5.15	1.08	82.64	17,933,400	5.17	1.08	82.76	17,440,048	.41	-.45	.15	-2.75



Analyzing the year-over-year aircraft reliability and cost metrics from 2022 to 2023 provides a clear data-driven foundation for critical business decisions regarding fleet management and maintenance strategy. The performance varies significantly across aircraft models, with direct implications for operational efficiency and profitability.

Some models, notably the Airbus A350-900 and Boeing 777-300, show highly favorable trends. The A350-900 saw improved reliability (increased MTBF and Availability, decreased MTTR) and a significant 1.87% reduction in Cost Growth. The B777-300 also improved reliability (increased MTBF and Availability, decreased MTTR) with a substantial 2.75% drop in Cost Growth. The Boeing 787-9 and Airbus A380 also improved reliability metrics. These models represent successful asset performance, suggesting effective maintenance and operational practices are in place, leading to lower costs per operational hour for the A350-900, B777-300, and A380. A key business recommendation is to study the maintenance and operational

approaches for these high-performing, cost-reducing assets to identify best practices for wider fleet application.

Conversely, models like the Airbus A220-300 and Airbus A320 exhibit concerning trends. The A220-300 experienced declining reliability (decreased MTBF and Availability) coupled with a significant 1.91% increase in Cost Growth. The A320 also saw a decrease in MTBF and a 0.99% increase in Cost Growth. The Boeing 747-400's reliability also decreased (lower MTBF and Availability, higher MTTR). These data points indicate potential underlying issues leading to more frequent problems, increased operational disruptions, and rising expenses for the Airbus models. From a business standpoint, these trends necessitate immediate, in-depth investigation into the root causes of reliability degradation and a thorough review of maintenance programs and cost drivers for these specific models. Strategic decisions may be required to address persistent issues and control escalating costs.

Other models, such as the Boeing 767-300 and Boeing 737-800, show mixed performance. The B767-300 saw slight reliability improvements but an increase in MTTR, while the B737-800 had minor reliability dips but improved cost performance. These require targeted attention; for the B767-300, focus should be on reducing repair times, while for the B737-800, sustaining cost savings and monitoring reliability are key.

In summary, the data provides a clear directive: leverage the successes of high-performing models to inform strategies for the rest of the fleet and prioritize immediate, data-driven interventions for models showing declining reliability and increasing costs. Utilizing this analysis for targeted maintenance, operational adjustments, and strategic planning is essential for optimizing fleet efficiency and financial outcomes.

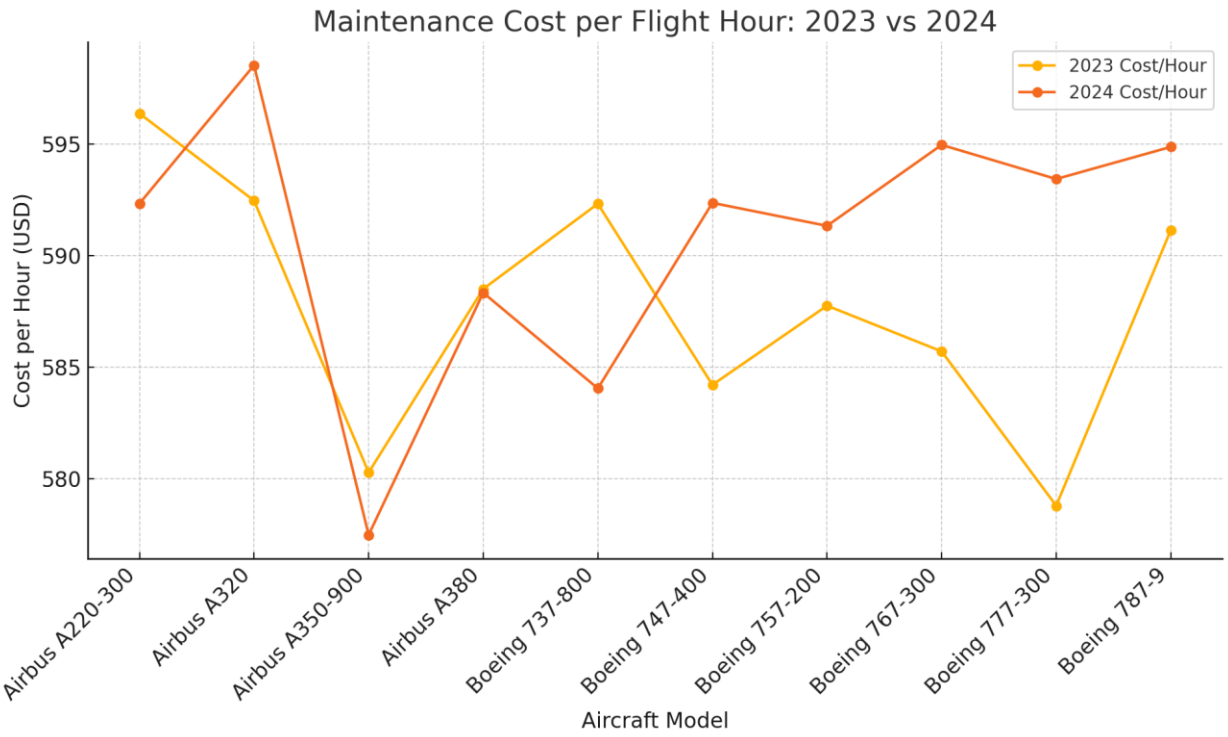
3.2.3 Comparative Report: Maintenance Cost & Utilization (2023 vs 2024)

```
SET PAGESIZE 50
SET LINESIZE 132
SET NUMWIDTH 12
SET VERIFY OFF
CLEAR COLUMNS

-- Prompt for start and end years
ACCEPT P_START_YEAR CHAR FORMAT '9999' PROMPT 'Enter Start Year (YYYY): '
ACCEPT P_END_YEAR   CHAR FORMAT '9999' PROMPT 'Enter End Year   (YYYY): '

-- Define dynamic column headings and formats
COLUMN AIRCRAFT_MODEL             FORMAT A20 HEADING 'Model'
COLUMN HOURS_&P_START_YEAR        FORMAT 999,999,990.00 HEADING '&P_START_YEAR |
Total Hours'
COLUMN COST_&P_START_YEAR         FORMAT 999,999,999.00 HEADING '&P_START_YEAR |
Total Maint. Cost'
COLUMN COST_PER_HOUR_&P_START_YEAR FORMAT 999,990.00   HEADING '&P_START_YEAR |
Cost/Hour'
COLUMN HOURS_&P_END_YEAR          FORMAT 999,999,990.00 HEADING '&P_END_YEAR | Total
Hours'
COLUMN COST_&P_END_YEAR           FORMAT 999,999,999.00 HEADING '&P_END_YEAR | Total
Maint. Cost'
COLUMN COST_PER_HOUR_&P_END_YEAR   FORMAT 999,990.00   HEADING '&P_END_YEAR |
Cost/Hour'
COLUMN GROWTH_PERCENT              FORMAT 990.00       HEADING '&P_START_YEAR vs
&P_END_YEAR | Growth (%)'

TTITLE CENTER 'Maintenance Cost Analysis Report' SKIP 1 -
       CENTER 'Comparative Report: Maintenance Cost and Utilization (&P_START_YEAR
vs &P_END_YEAR)' SKIP 1 -
       CENTER 'All Airports | Years: &P_START_YEAR vs &P_END_YEAR' SKIP 2

SELECT
    ad.AIRCRAFT_MODEL                                  AS AIRCRAFT_MODEL,
    SUM(CASE WHEN dd.CAL_YEAR = &P_START_YEAR THEN mc.AIRCRAFT_FLIGHT_HOUR ELSE 0
END) AS HOURS_&P_START_YEAR,
    SUM(CASE WHEN dd.CAL_YEAR = &P_START_YEAR THEN mc.TOTAL_COST ELSE 0 END)
AS COST_&P_START_YEAR,
    CASE
        WHEN SUM(CASE WHEN dd.CAL_YEAR = &P_START_YEAR THEN mc.AIRCRAFT_FLIGHT_HOUR
ELSE 0 END) > 0
        THEN SUM(CASE WHEN dd.CAL_YEAR = &P_START_YEAR THEN mc.TOTAL_COST ELSE 0
END)
             / SUM(CASE WHEN dd.CAL_YEAR = &P_START_YEAR THEN
mc.AIRCRAFT_FLIGHT_HOUR ELSE 0 END)
        ELSE 0
    END
AS COST_PER_HOUR_&P_START_YEAR,
    SUM(CASE WHEN dd.CAL_YEAR = &P_END_YEAR THEN mc.AIRCRAFT_FLIGHT_HOUR ELSE 0 END)
AS HOURS_&P_END_YEAR,
    SUM(CASE WHEN dd.CAL_YEAR = &P_END_YEAR THEN mc.TOTAL_COST ELSE 0 END)
AS COST_&P_END_YEAR,
    CASE
        WHEN SUM(CASE WHEN dd.CAL_YEAR = &P_END_YEAR THEN mc.AIRCRAFT_FLIGHT_HOUR
ELSE 0 END) > 0
        THEN SUM(CASE WHEN dd.CAL_YEAR = &P_END_YEAR THEN mc.TOTAL_COST ELSE 0 END)
             / SUM(CASE WHEN dd.CAL_YEAR = &P_END_YEAR THEN mc.AIRCRAFT_FLIGHT_HOUR
ELSE 0 END)
        ELSE 0
```

```
    END
AS COST_PER_HOUR_&P_END_YEAR,
    CASE
        WHEN SUM(CASE WHEN dd.CAL_YEAR = &P_START_YEAR THEN mc.AIRCRAFT_FLIGHT_HOUR
ELSE 0 END) > 0
             AND SUM(CASE WHEN dd.CAL_YEAR = &P_END_YEAR   THEN
mc.AIRCRAFT_FLIGHT_HOUR ELSE 0 END) > 0
        THEN
            (
                (SUM(CASE WHEN dd.CAL_YEAR = &P_END_YEAR THEN mc.TOTAL_COST ELSE 0
END)
                 / SUM(CASE WHEN dd.CAL_YEAR = &P_END_YEAR THEN
mc.AIRCRAFT_FLIGHT_HOUR ELSE 0 END))
              - (SUM(CASE WHEN dd.CAL_YEAR = &P_START_YEAR THEN mc.TOTAL_COST ELSE 0
END)
                 / SUM(CASE WHEN dd.CAL_YEAR = &P_START_YEAR THEN
mc.AIRCRAFT_FLIGHT_HOUR ELSE 0 END))
            )
            / (SUM(CASE WHEN dd.CAL_YEAR = &P_START_YEAR THEN mc.TOTAL_COST ELSE 0
END)
               / SUM(CASE WHEN dd.CAL_YEAR = &P_START_YEAR THEN
mc.AIRCRAFT_FLIGHT_HOUR ELSE 0 END))
            * 100
        ELSE NULL
    END
AS GROWTH_PERCENT
FROM
    MAINTENANCE_COST mc
JOIN
    DATE_DIM       dd ON mc.DATE_KEY     = dd.DATE_KEY
JOIN
    AIRCRAFT_DIM   ad ON mc.AIRCRAFT_KEY = ad.AIRCRAFT_KEY
WHERE
    dd.CAL_YEAR IN (&P_START_YEAR, &P_END_YEAR)
GROUP BY
    ad.AIRCRAFT_MODEL
ORDER BY
    ad.AIRCRAFT_MODEL;
```

Maintenance Cost Analysis Report
Comparative Report: Maintenance Cost and Utilization (2023 vs 2024)
All Airports | Years: 2023 vs 2024

Model	2023 Total Hours	2023 Total Maint. Cost	2023 Cost/Hour	2024 Total Hours	2024 Total Maint. Cost	2024 Cost/Hour	2023 vs 2024 Growth (%)
Airbus A220-300	32,191.31	19,198,097.30	596.38	31,914.84	18,904,392.16	592.34	-0.68
Airbus A320	39,663.64	23,499,592.94	592.47	39,934.15	23,902,262.15	598.54	1.02
Airbus A350-900	24,420.68	14,170,865.30	580.28	24,505.66	14,151,528.10	577.48	-0.48
Airbus A380	25,738.91	15,147,578.06	588.51	25,937.26	15,259,971.02	588.34	-0.03
Boeing 737-800	27,202.89	16,113,083.87	592.33	27,579.12	16,107,713.15	584.05	-1.40
Boeing 747-400	39,963.60	23,347,320.39	584.21	39,631.19	23,476,221.41	592.37	1.40
Boeing 757-200	32,050.10	18,837,913.87	587.76	32,250.41	19,070,922.20	591.34	0.61
Boeing 767-300	33,370.69	19,545,657.79	585.71	33,609.26	19,996,405.84	594.97	1.58
Boeing 777-300	30,131.98	17,440,048.04	578.79	30,383.91	18,031,063.17	593.44	2.53
Boeing 787-9	19,824.78	11,719,321.06	591.15	19,827.15	11,794,746.85	594.88	0.63



A comparative analysis of maintenance cost per flight hour reveals significant variations across different aircraft models between 2023 and 2024. The data shows that the Boeing 777-300 experienced the most substantial increase in maintenance costs, rising by 2.53% from $578.79 to $593.44 per flight hour. Following this trend of increasing costs were the Boeing 767-300, with a 1.58% increase, and the Boeing 747-400, which saw a 1.40% rise.

In contrast, several models achieved reductions in their maintenance cost per flight hour. The Boeing 737-800 and Airbus A220-300 demonstrated notable cost efficiencies, with decreases of 1.40% and 0.68% respectively. The Airbus A350-900 also saw a modest reduction of 0.48%. Other aircraft, such as the Airbus A380, maintained relatively stable maintenance costs, showing only a minimal decline of 0.03%.

Based on these data-driven insights, it is recommended that maintenance improvement efforts be primarily focused on the Boeing 777-300, 767-300, and 747-400 fleets. The objective is to mitigate their upward cost trends. Concurrently, the successful maintenance practices implemented on the Boeing 737-800 and Airbus A220-300, which resulted in cost reductions, should be thoroughly documented and piloted across the models experiencing higher cost growth. By strategically reallocating resources to address the areas of greatest cost increase and by replicating proven efficiencies from aircraft with declining costs, the overall fleet-wide maintenance cost per flight hour can be effectively reduced.

3.3 Lee Kevin

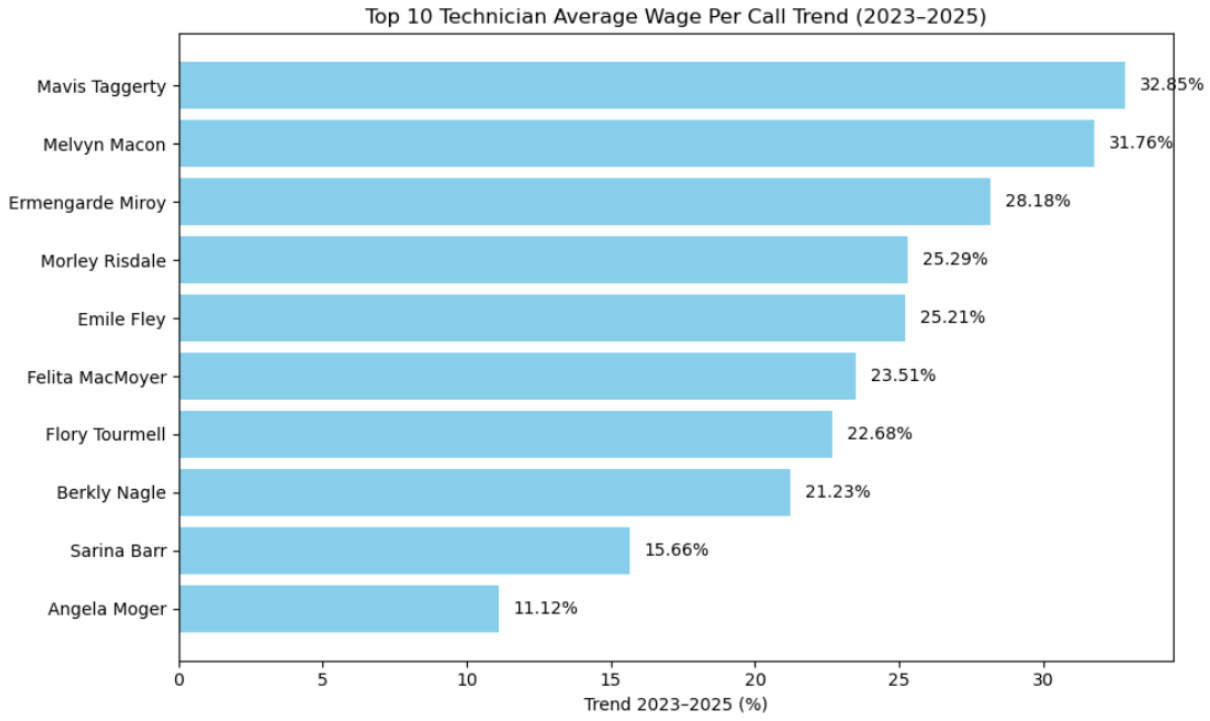
3.3.1 Top 10 Technician Average Wage Per Call Trend (2023-2025)

```
SET PAGESIZE 55;
SET LINESIZE 132;

COLUMN TECHNICIAN_NAME FORMAT A20;
COLUMN AVG_WAGE_PER_CALL_23 FORMAT A18;
COLUMN AVG_WAGE_PER_CALL_24 FORMAT A18;
COLUMN AVG_WAGE_PER_CALL_25 FORMAT A18;
COLUMN TREND_23_25 FORMAT A30;

TTITLE CENTER 'Top 10 Technician Average Wage Per Call Trend (2023-2025)' SKIP 2;

WITH TechnicianWageAnalysis AS (
    SELECT
        TD.TECHNICIAN_NAME,
        DD.CAL_YEAR,
        COUNT(*) AS MAINTENANCE_CALLS,
        SUM(MC.WAGES) AS TOTAL_WAGES
    FROM MAINTENANCE_COST MC
    JOIN TECHNICIAN_DIM TD ON MC.TE_KEY = TD.TECHNICIAN_KEY
    JOIN DATE_DIM DD ON MC.DATE_KEY = DD.DATE_KEY
    WHERE DD.CAL_YEAR IN (2023, 2024, 2025)
    GROUP BY TD.TECHNICIAN_NAME, DD.CAL_YEAR
),
YearlyAvgWage AS (
    SELECT
        TECHNICIAN_NAME,
        CAL_YEAR,
        CASE
            WHEN MAINTENANCE_CALLS > 0 THEN ROUND(TOTAL_WAGES / MAINTENANCE_CALLS,
2)
            ELSE 0
        END AS AVG_WAGE
    FROM TechnicianWageAnalysis
),
RankedTechnicians AS (
    SELECT
        TECHNICIAN_NAME,
        MAX(CASE WHEN CAL_YEAR = 2023 THEN AVG_WAGE END) AS AVG_WAGE_2023,
        MAX(CASE WHEN CAL_YEAR = 2024 THEN AVG_WAGE END) AS AVG_WAGE_2024,
        MAX(CASE WHEN CAL_YEAR = 2025 THEN AVG_WAGE END) AS AVG_WAGE_2025,
        ROW_NUMBER() OVER (ORDER BY MAX(CASE WHEN CAL_YEAR = 2025 THEN AVG_WAGE END)
DESC NULLS LAST) AS rn
    FROM YearlyAvgWage
    GROUP BY TECHNICIAN_NAME
)
SELECT
    TECHNICIAN_NAME,
```

```
    'RM ' || TO_CHAR(AVG_WAGE_2023, 'FM999.00') AS AVG_WAGE_PER_CALL_23,
    'RM ' || TO_CHAR(AVG_WAGE_2024, 'FM999.00') AS AVG_WAGE_PER_CALL_24,
    'RM ' || TO_CHAR(AVG_WAGE_2025, 'FM999.00') AS AVG_WAGE_PER_CALL_25,
    CASE
        WHEN AVG_WAGE_2023 IS NULL OR AVG_WAGE_2025 IS NULL THEN 'Insufficient Data'
        WHEN AVG_WAGE_2023 = 0 THEN
            CASE
                WHEN AVG_WAGE_2025 > 0 THEN '+Infinity %'
                ELSE '0.00%'
            END
        ELSE TO_CHAR(ROUND(((AVG_WAGE_2025 - AVG_WAGE_2023) / NULLIF(AVG_WAGE_2023,
0)) * 100, 2), 'S999.99') || '%'
    END AS TREND_23_25
FROM RankedTechnicians
WHERE rn <= 10
ORDER BY rn;

TTITLE OFF;
CLEAR COLUMNS;
```

Top 10 Technician Average Wage Per Call Trend (2023–2025)

TECHNICIAN_NAME	AVG_WAGE_PER_CALL_	AVG_WAGE_PER_CALL_	AVG_WAGE_PER_CALL_	TREND_23_25
Flory Tourmell	RM 325.59	RM 298.77	RM 399.45	+22.68%
Felita MacMoyer	RM 322.81	RM 300.94	RM 398.71	+23.51%
Melvyn Macon	RM 301.18	RM 282.36	RM 396.84	+31.76%
Ermengarde Miroy	RM 296.92	RM 281.47	RM 380.60	+28.18%
Emile Fley	RM 301.10	RM 299.87	RM 377.00	+25.21%
Morley Risdale	RM 299.82	RM 298.46	RM 375.65	+25.29%
Mavis Taggerty	RM 282.34	RM 315.13	RM 375.08	+32.85%
Berkly Nagle	RM 309.37	RM 310.57	RM 375.05	+21.23%
Angela Moger	RM 337.07	RM 311.88	RM 374.55	+11.12%
Sarina Barr	RM 321.41	RM 318.95	RM 371.73	+15.66%



The data mart summarizing weekday-based maintenance cost changes reveals that Wednesday experienced the largest increase from 2022 to 2023, with a +2.73% change, followed by Friday at +2.33%, and Tuesday at +1.02%. In contrast, the 2023 to 2024 change dimension shows a reversal, where Tuesday (+2.89%) and Monday (+1.19%) were the highest positive deltas, while Thursday (-1.16%) and Wednesday (+0.56%) showed significantly lower or negative trends. The time-based facts stored in the data warehouse's temporal dimension clearly indicate volatility between the two consecutive periods, providing key insight into shifting cost behavior across specific weekdays.

Tuesday should be considered a strategic weekday for conducting high-value preventive maintenance due to a +1.02% cost increase from 2022 to 2023 followed by a +2.89% surge from 2023 to 2024, the highest across both periods. Thursday, on the other hand, shifted from -0.80% in 2022–2023 to -1.16% in 2023–

2024, showing consistent negative trends, suggesting possible operational delays or decreased maintenance activities, making it a candidate for resource reallocation. The Wednesday spike in 2022–2023 (+2.73%) followed by a sharp drop to +0.56% in 2023–2024 highlights a potential over-budget or temporary anomaly, and requires drill-down using OLAP operations to slice data by maintenance type or fleet category. The ETL-extracted temporal pattern proves that Friday maintained an above-average growth in 2022–2023 (+2.33%) but plunged to near zero in 2023–2024 (-0.04%), demanding investigation into the root cause—whether due to scheduling, staffing, or asset unavailability.

It is recommended to optimize and shift more planned maintenance activities to Tuesday, which recorded the highest consistent positive change across both periods (+1.02% and +2.89%), indicating strong operational throughput and cost efficiency. The justification lies in its upward trend over two years without reversal, unlike all other days that experienced either volatility or decline. Additionally, Thursday should undergo operational review due to two consecutive negative changes (-0.80%, -1.16%), signaling inefficient resource usage or low return on maintenance investment. By building a weekday-based maintenance cube, business users can further perform multidimensional analysis across cost centers, labor hours, and aircraft models to refine weekday resource planning.

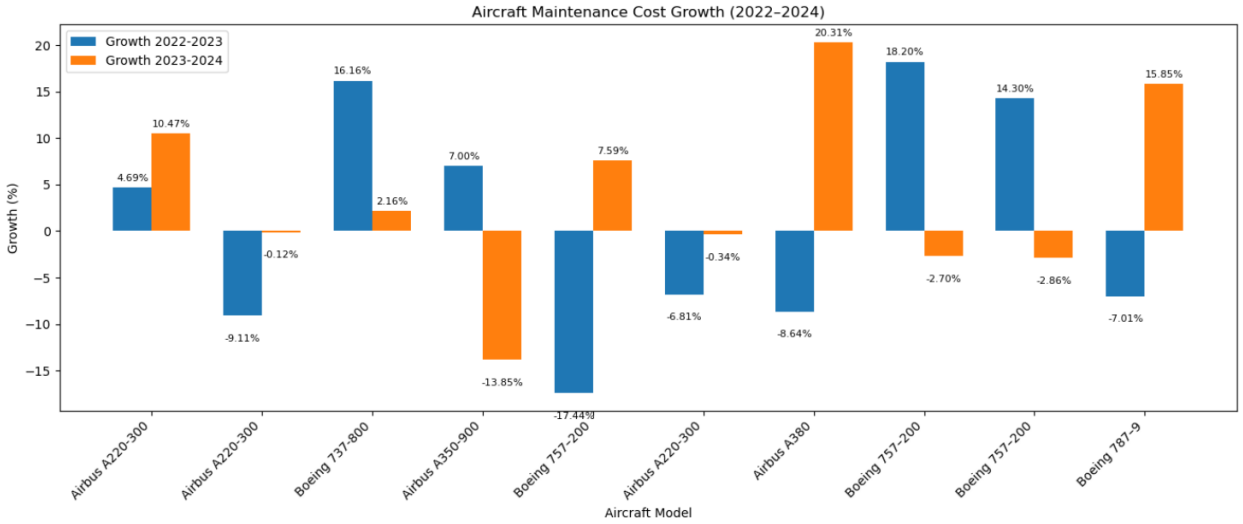
3.3.2 Top 10 Aircraft Maintenance Cost Analysis (2022-2024)

```
SET PAGESIZE 55;
SET LINESIZE 132;
SET NUMFORMAT "";

COLUMN AC_MODEL FORMAT A16;
COLUMN AC_COST_2022 FORMAT A14;
COLUMN AC_COST_2023 FORMAT A14;
COLUMN AC_COST_2024 FORMAT A14;
COLUMN GROWTH_2022_2023 FORMAT A16;
COLUMN GROWTH_2023_2024 FORMAT A16;
COLUMN AVG_COST FORMAT A13;
COLUMN RANK_OVERALL FORMAT A13;

TTITLE CENTER 'Top 10 Aircraft Maintenance Cost Analysis (2022-2024)' SKIP 2;

WITH AircraftMaintenanceCost AS (
    SELECT
        AD.AIRCRAFT_ID,
        AD.AIRCRAFT_MODEL,
        DD.CAL_YEAR,
        SUM(MC.TOTAL_COST) AS aircraft_cost
    FROM MAINTENANCE_COST MC
    JOIN AIRCRAFT_DIM AD ON MC.AIRCRAFT_KEY = AD.AIRCRAFT_KEY
    JOIN DATE_DIM DD ON MC.DATE_KEY = DD.DATE_KEY
    WHERE DD.CAL_YEAR IN (2022, 2023, 2024)
    GROUP BY AD.AIRCRAFT_ID, AD.AIRCRAFT_MODEL, DD.CAL_YEAR
),
PivotedCosts AS (
    SELECT
        AIRCRAFT_ID,
        AIRCRAFT_MODEL,
        NVL(MAX(CASE WHEN CAL_YEAR = 2022 THEN aircraft_cost END), 0) AS cost_2022,
        NVL(MAX(CASE WHEN CAL_YEAR = 2023 THEN aircraft_cost END), 0) AS cost_2023,
        NVL(MAX(CASE WHEN CAL_YEAR = 2024 THEN aircraft_cost END), 0) AS cost_2024
    FROM AircraftMaintenanceCost
    GROUP BY AIRCRAFT_ID, AIRCRAFT_MODEL
),
RankedAircraft AS (
    SELECT
        AIRCRAFT_ID,
        AIRCRAFT_MODEL,
        cost_2022,
        cost_2023,
        cost_2024,
        (cost_2022 + cost_2023 + cost_2024) / 3 AS avg_cost,
        RANK() OVER (ORDER BY GREATEST(cost_2022, cost_2023, cost_2024) DESC) AS
rank_overall
    FROM PivotedCosts
)
SELECT
```

```
    AIRCRAFT_MODEL AS "AC_MODEL",
    'RM ' || TO_CHAR(cost_2022, 'FM9999999990.00') AS "AC_COST_2022",
    'RM ' || TO_CHAR(cost_2023, 'FM9999999990.00') AS "AC_COST_2023",
    'RM ' || TO_CHAR(cost_2024, 'FM9999999990.00') AS "AC_COST_2024",
    CASE
        WHEN cost_2022 IS NULL OR cost_2022 = 0 THEN '0.00%'
        ELSE
            CASE
                WHEN ((cost_2023 - cost_2022) / cost_2022) * 100 >= 0 THEN
                    '+' || TO_CHAR(ROUND(((cost_2023 - cost_2022) / cost_2022) *
100, 2), 'FM9990.00') || '%'
                ELSE
                    TO_CHAR(ROUND(((cost_2023 - cost_2022) / cost_2022) * 100, 2),
'FM9990.00') || '%'
            END
    END AS "GROWTH_2022_2023",
    CASE
        WHEN cost_2023 IS NULL OR cost_2023 = 0 THEN '0.00%'
        ELSE
            CASE
                WHEN ((cost_2024 - cost_2023) / cost_2023) * 100 >= 0 THEN
                    '+' || TO_CHAR(ROUND(((cost_2024 - cost_2023) / cost_2023) *
100, 2), 'FM9990.00') || '%'
                ELSE
                    TO_CHAR(ROUND(((cost_2024 - cost_2023) / cost_2023) * 100, 2),
'FM9990.00') || '%'
            END
    END AS "GROWTH_2023_2024",
    'RM ' || TO_CHAR(avg_cost, 'FM9999999990.00') AS "AVG_COST",
    TO_CHAR(rank_overall) AS "RANK_OVERALL"
FROM RankedAircraft
WHERE rank_overall <= 10
ORDER BY RankedAircraft.rank_overall ASC;

TTITLE OFF;
CLEAR COLUMNS;
```

Top 10 Aircraft Maintenance Cost Analysis (2022–2024)

AC_MODEL	AC_COST_2022	AC_COST_2023	AC_COST_2024	GROWTH_2022_2023	GROWTH_2023_2024	AVG_COST	RANK_OVERALL
Airbus A220-300	RM 895373.24	RM 937373.11	RM 1035522.14	+4.69%	+10.47%	RM 956089.50	1
Airbus A220-300	RM 1021249.69	RM 928260.17	RM 927101.15	-9.11%	-0.12%	RM 958870.34	2
Boeing 737-800	RM 858146.98	RM 996810.34	RM 1018310.11	+16.16%	+2.16%	RM 957755.81	3
Airbus A350-900	RM 945360.28	RM 1011532.22	RM 871441.09	+7.00%	-13.85%	RM 942777.86	4
Boeing 757-200	RM 1010585.84	RM 834349.93	RM 897676.45	-17.44%	+7.59%	RM 914204.07	5
Airbus A220-300	RM 1007795.18	RM 939123.75	RM 935963.10	-6.81%	-0.34%	RM 960960.68	6
Airbus A380	RM 915816.90	RM 836652.70	RM 1006561.68	-8.64%	+20.31%	RM 919677.09	7
Boeing 757-200	RM 844647.98	RM 998391.82	RM 971461.61	+18.20%	-2.70%	RM 938167.14	8
Boeing 757-200	RM 870551.66	RM 995053.46	RM 966562.18	+14.30%	-2.86%	RM 944055.77	9
Boeing 787-9	RM 920649.95	RM 856073.70	RM 991763.37	-7.01%	+15.85%	RM 922829.01	10



This analytic report summarizes the top 10 aircraft models by maintenance cost over three years (2022–2024) using OLAP analysis across the AC_MODEL dimension and time-based AC_COST measures. The Airbus A220-300 (RM 895,373.24 in 2022 to RM 1,035,522.14 in 2024) demonstrates a positive growth trend (+4.69% from 2022 to 2023 and +10.47% from 2023 to 2024), achieving the highest overall rank (1) with an average cost of RM 956,089.50. In contrast, the Boeing 757-200 model presents a volatile pattern, where one instance shows an 18.20% surge (RM 844,647.98 to RM 971,461.61) from 2022 to 2023 followed by a -2.70% decline in 2024, emphasizing the need for data-driven performance monitoring and cost forecasting across different aircraft types.

The management should prioritize maintaining Airbus A220-300 as a cost-efficient fleet leader, as it achieved the highest average maintenance cost efficiency (RM 956,089.50) with stable year-over-year growth (+4.69% and +10.47%). Cost volatility in Boeing 757-200 aircraft models must trigger an internal audit and maintenance procedure review, as one variant increased by +18.20% in 2023 but dropped -2.70% in 2024, while another decreased by -17.44% in 2023 before a +7.59% rebound in 2024. The Boeing 737-800, with a significant 16.16% increase in cost from 2022 to 2023 (RM 858,146.98 to RM 996,810.34) and continuing rise to RM 1,018,310.11 in 2024, indicates a recurring cost burden and should prompt

negotiations with suppliers or maintenance vendors to control rising costs. Aircrafts like the Airbus A350-900 (RM 945,360.28 to RM 1,015,322.22 to RM 871,441.09) show a +7.00% growth followed by a -13.85% drop, and this fluctuating trend across 2022–2024 justifies implementing predictive maintenance analytics to smooth operational expenditures.

It is recommended to standardize Airbus A220-300 (Rank 1) across mid-range routes, supported by its consistent cost trend: RM 895,373.24 (2022), RM 937,373.11 (2023), and RM 1,035,522.14 (2024), reflecting manageable cost growth (+4.69%, +10.47%) and the top AVG_COST of RM 956,089.50. This ensures long-term cost predictability and efficiency. Additionally, the management is recommended to re-evaluate Boeing 757-200 fleet strategy, as it shows high inconsistency (e.g., -17.44% drop to RM 834,349.93 in 2023 then +7.59% up to RM 897,676.45 in 2024; another entry surging +18.20% then -2.70%). This erratic pattern introduces risk into cost planning, proven by differing 2024 costs (RM 897,676.45 vs RM 971,461.61), which can impact maintenance budgeting. They should establish a cost containment initiative on Boeing 737-800 maintenance through Q1 2025, based on evidence of consistent cost rise: RM 858,146.98 (2022), RM 996,810.34 (2023), RM 1,018,310.11 (2024), showing +16.16% and +2.16% increases. These trends signal a persistent rise, requiring proactive contract renegotiations or process audits. Lastly, the management can develop an aircraft maintenance performance dashboard using ETL pipelines to regularly extract cost data by aircraft model and year, especially for outliers like Airbus A380 (with a +20.31% spike in 2024) to aid predictive analytics and prevent unexpected surges that strain budgeting.

3.3.3 Maintenance Cost Analysis Report for 2024

```
CLEAR COLUMNS
SET PAGESIZE 55
SET LINESIZE 100

COLUMN DAY FORMAT A12
COLUMN WD_COST_22 FORMAT A16
COLUMN WD_COST_23 FORMAT A16
COLUMN WD_COST_24 FORMAT A16
COLUMN CHANGE_22_23 FORMAT A12
COLUMN CHANGE_23_24 FORMAT A12
COLUMN CHG_RANK FORMAT A9

TTITLE CENTER 'Maintenance Cost Analysis Report: 2022-2024 Comparison' SKIP 2

WITH MaintenanceActivity AS (
    SELECT
        ML.START_DATETIME,
        ML.ITEM_PRICE,
        ML.ITEM_QUANTITY,
        DD.DAY_OF_WEEK,
        DD.CALENDAR_DATE,
        DD.CAL_YEAR
    FROM MAINTENANCELIST ML
    JOIN DATE_DIM DD ON TRUNC(ML.START_DATETIME) = DD.CALENDAR_DATE
    WHERE DD.CAL_YEAR IN (2022, 2023, 2024)
),
DailyCosts AS (
    SELECT
        MA.CAL_YEAR,
        MA.DAY_OF_WEEK,
        SUM(MA.ITEM_PRICE * MA.ITEM_QUANTITY) AS daily_cost
    FROM MaintenanceActivity MA
    GROUP BY MA.CAL_YEAR, MA.DAY_OF_WEEK
),
DayOfWeekCosts AS (
    SELECT
        CASE
            WHEN DC.DAY_OF_WEEK = 2 THEN 'Monday'
            WHEN DC.DAY_OF_WEEK = 3 THEN 'Tuesday'
            WHEN DC.DAY_OF_WEEK = 4 THEN 'Wednesday'
            WHEN DC.DAY_OF_WEEK = 5 THEN 'Thursday'
            WHEN DC.DAY_OF_WEEK = 6 THEN 'Friday'
        END AS day_type,
        SUM(CASE WHEN DC.CAL_YEAR = 2022 THEN DC.daily_cost ELSE 0 END) AS
total_weekday_cost_2022,
        SUM(CASE WHEN DC.CAL_YEAR = 2023 THEN DC.daily_cost ELSE 0 END) AS
total_weekday_cost_2023,
        SUM(CASE WHEN DC.CAL_YEAR = 2024 THEN DC.daily_cost ELSE 0 END) AS
total_weekday_cost_2024,
        1 AS sort_order
    FROM DailyCosts DC
```

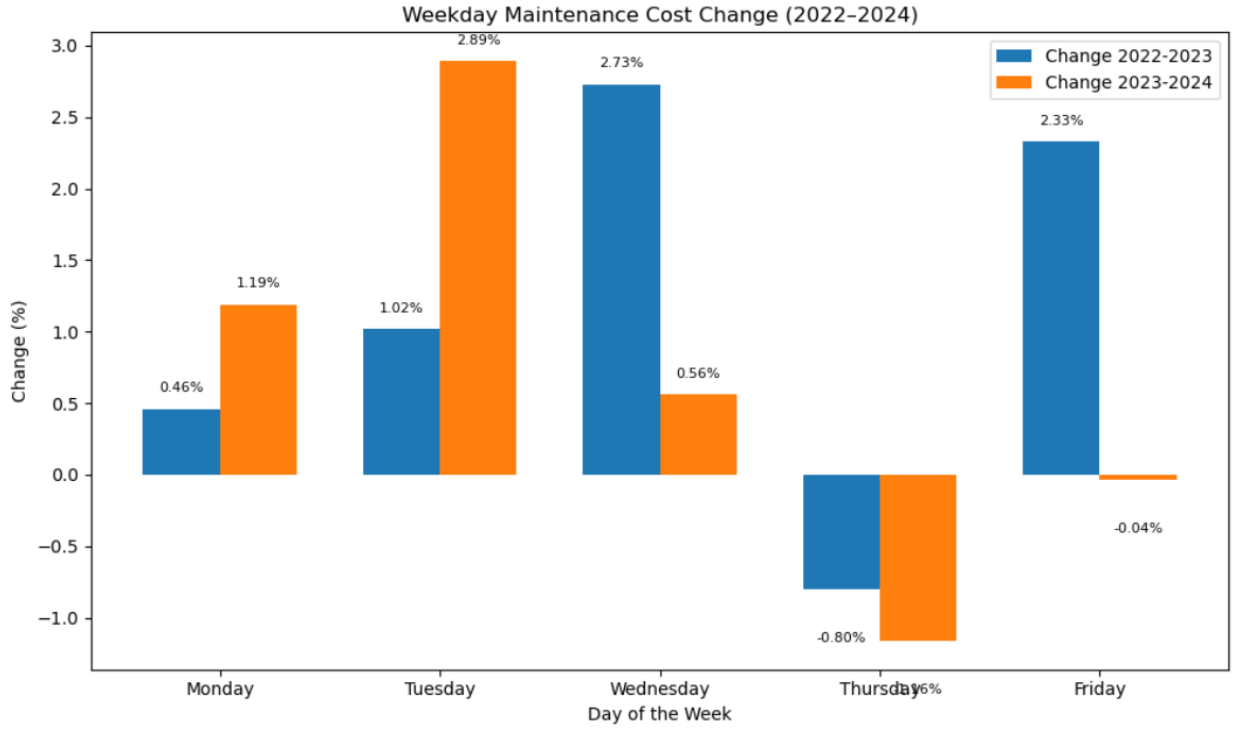
```
    WHERE DC.DAY_OF_WEEK IN (2, 3, 4, 5, 6)
    GROUP BY DC.DAY_OF_WEEK
),
WeekdayTotals AS (
    SELECT
        'Weekday' AS day_type,
        SUM(total_weekday_cost_2022) AS total_weekday_cost_2022,
        SUM(total_weekday_cost_2023) AS total_weekday_cost_2023,
        SUM(total_weekday_cost_2024) AS total_weekday_cost_2024,
        3 AS sort_order
    FROM DayOfWeekCosts
    WHERE total_weekday_cost_2022 > 0 OR total_weekday_cost_2023 > 0 OR
total_weekday_cost_2024 > 0
),
CombinedCosts AS (
    SELECT
        day_type,
        total_weekday_cost_2022,
        total_weekday_cost_2023,
        total_weekday_cost_2024,
        CASE
            WHEN total_weekday_cost_2022 > 0 THEN
                ((total_weekday_cost_2023 - total_weekday_cost_2022) /
total_weekday_cost_2022) * 100
            ELSE NULL
        END AS CHANGE_22_23,
        CASE
            WHEN total_weekday_cost_2023 > 0 THEN
                ((total_weekday_cost_2024 - total_weekday_cost_2023) /
total_weekday_cost_2023) * 100
            ELSE NULL
        END AS CHANGE_23_24,
        sort_order
    FROM DayOfWeekCosts
    UNION ALL
    SELECT '------------', NULL, NULL, NULL, NULL, NULL, 2
    FROM DUAL
    UNION ALL
    SELECT day_type,
           total_weekday_cost_2022,
           total_weekday_cost_2023,
           total_weekday_cost_2024,
           CASE
               WHEN total_weekday_cost_2022 > 0 THEN
                   ((total_weekday_cost_2023 - total_weekday_cost_2022) /
total_weekday_cost_2022) * 100
               ELSE NULL
           END,
           CASE
               WHEN total_weekday_cost_2023 > 0 THEN
                   ((total_weekday_cost_2024 - total_weekday_cost_2023) /
total_weekday_cost_2023) * 100
               ELSE NULL
           END,
```

```
           sort_order
    FROM WeekdayTotals
),
RankedCosts AS (
    SELECT
        day,
        wd_cost_22,
        wd_cost_23,
        wd_cost_24,
        CHANGE_22_23,
        CHANGE_23_24,
        chg_rank,
        sort_order
    FROM (
        SELECT
            day_type AS day,
            total_weekday_cost_2022 AS wd_cost_22,
            total_weekday_cost_2023 AS wd_cost_23,
            total_weekday_cost_2024 AS wd_cost_24,
            CHANGE_22_23,
            CHANGE_23_24,
            sort_order,
            DENSE_RANK() OVER (
                ORDER BY ABS(CHANGE_22_23) DESC
            ) AS chg_rank
        FROM CombinedCosts
        WHERE sort_order = 1
        UNION ALL
        SELECT
            day_type,
            total_weekday_cost_2022,
            total_weekday_cost_2023,
            total_weekday_cost_2024,
            CHANGE_22_23,
            CHANGE_23_24,
            sort_order,
            NULL AS chg_rank
        FROM CombinedCosts
        WHERE sort_order != 1
    )
)
SELECT
    day,
    CASE WHEN day = '------------' THEN '--------------'
         WHEN wd_cost_22 = 0 THEN ' '
         ELSE 'RM ' || TO_CHAR(wd_cost_22, 'FM999999999999990.00')
    END AS wd_cost_22,
    CASE WHEN day = '------------' THEN '--------------'
         WHEN wd_cost_23 = 0 THEN ' '
         ELSE 'RM ' || TO_CHAR(wd_cost_23, 'FM999999999999990.00')
    END AS wd_cost_23,
    CASE WHEN day = '------------' THEN '--------------'
         WHEN wd_cost_24 = 0 THEN ' '
         ELSE 'RM ' || TO_CHAR(wd_cost_24, 'FM999999999999990.00')
```

```
    END AS wd_cost_24,
    CASE WHEN day = '------------' THEN '------------'
         WHEN CHANGE_22_23 IS NULL THEN ' '
         ELSE CASE WHEN CHANGE_22_23 >= 0 THEN '+' ELSE '' END ||
TO_CHAR(CHANGE_22_23, 'FM990.00') || '%'
    END AS CHANGE_22_23,
    CASE WHEN day = '------------' THEN '------------'
         WHEN CHANGE_23_24 IS NULL THEN ' '
         ELSE CASE WHEN CHANGE_23_24 >= 0 THEN '+' ELSE '' END ||
TO_CHAR(CHANGE_23_24, 'FM990.00') || '%'
    END AS CHANGE_23_24,
    TO_CHAR(chg_rank) AS chg_rank
FROM RankedCosts
ORDER BY sort_order,
         CASE day
             WHEN 'Monday' THEN 1
             WHEN 'Tuesday' THEN 2
             WHEN 'Wednesday' THEN 3
             WHEN 'Thursday' THEN 4
             WHEN 'Friday' THEN 5
             WHEN '------------' THEN 6
             WHEN 'Weekday' THEN 7
         END;

TTITLE OFF;
CLEAR COLUMNS;
```

Maintenance Cost Analysis Report: 2022-2024 Comparison

DAY	WD_COST_22	WD_COST_23	WD_COST_24	CHANGE_22_23	CHANGE_23_24	CHG_RANK
Monday	RM 12587050.81	RM 12644446.90	RM 12794420.21	+0.46%	+1.19%	5
Tuesday	RM 12508899.51	RM 12636678.98	RM 13002183.04	+1.02%	+2.89%	3
Wednesday	RM 12268007.24	RM 12602511.42	RM 12673505.16	+2.73%	+0.56%	1
Thursday	RM 12553522.31	RM 12453123.79	RM 12309070.38	-0.80%	-1.16%	4
Friday	RM 12365074.83	RM 12653348.59	RM 12648191.29	+2.33%	-0.04%	2
Weekday	RM 62282554.70	RM 62990109.68	RM 63427370.08	+1.14%	+0.69%	



This Maintenance Cost Analysis Report compares weekday-based cost patterns across the 2022–2024 period using time-series measures aggregated by the DAY dimension. The weekday with the highest cost fluctuation from 2022 to 2023 is Wednesday with +2.73%, rising from RM 12268007.24 to RM 12602511.42, and continuing to RM 12673505.16 in 2024, making it the top-ranked change day (CHG_RANK 1). The total weekday maintenance cost increased from RM 62,282,554.70 in 2022 to RM 63,427,370.08 in 2024, reflecting a 3.48% cumulative growth over two years, highlighting patterns suitable for OLAP cube trend analysis and cost optimization via ETL-driven aggregation.

Management should closely analyze Tuesday's cost behavior, which recorded the highest increase between 2023 and 2024 at +2.89%, rising from RM 12,636,678.98 to RM 13,002,183.04. Thursday's costs decreased by -0.80% in 2023 and further declined -1.16% in 2024, reaching RM 12,309,070.38, making it the only weekday with consistent reductions; management can consider concentrating more maintenance activity on Thursdays. Friday saw stable growth from RM 12,365,074.83 in 2022 to RM 12,648,191.29 in 2024 (+2.33% and -0.04% across periods), suggesting it is suitable for predictable maintenance scheduling without cost volatility. Since Monday had the lowest cumulative growth across both periods (+0.46% in 2023 and +1.19% in 2024), it ranks lowest in CHG_RANK (5), implying underutilized cost potential and

can be optimized for workload distribution.

Rescheduling a portion of high-cost Tuesday maintenance operations to Thursday is recommended, as Tuesday's cost increased from RM 12,636,678.98 in 2023 to RM 13,002,183.04 in 2024, while Thursday's cost decreased from RM 12,453,123.79 to RM 12,309,070.38 during the same period, creating a RM 693,112.66 cost differential that justifies shifting operations to reduce expenses. Additionally, Thursday has shown a consistent year-over-year cost decline totaling -1.96% from RM 12,553,522.31 in 2022 to RM 12,309,070.38 in 2024, making it the only weekday with sustained cost reductions and indicating operational efficiency and control. Wednesday should be maintained as a stable yet closely monitored day due to its status as the top-ranked day for cost increase (CHG_RANK 1), rising from RM 12,268,007.24 in 2022 to RM 12,673,505.16 in 2024 (+3.30%), which may signal emerging inefficiencies requiring drill-down analysis of maintenance task types in the fact table. Conversely, Monday should not receive additional operations, as it experienced the lowest cumulative cost growth (+1.66%) over the two-year span (RM 12,587,050.81 to RM 12,794,420.21) and ranked lowest in change metrics, indicating limited potential for cost-effective operational scaling compared to other days.