

THIRD PROJECT: PHYSICAL DATA WAREHOUSE DESIGN

➤ Proposed SQL queries to retrieve data for the created the queries:

First Query:

```
SELECT t.monthid AS Month, SUM(a.FlightHours) AS FH, SUM(a.FlightCycles) AS FC
FROM AircraftUtilization a, TemporalDimension t, AircraftDimension d
WHERE t.ID = a.timeid AND d.id=a.aircraftid AND d.model='777'
GROUP BY t.monthid;
```

Second Query:

```
SELECT m.y AS Year, SUM(a.scheduledoutofservice) AS ADOSS, SUM(a.unscheduledoutofservice)
AS ADOSU
FROM AircraftUtilization a, TemporalDimension t, Months m
WHERE a.timeid=t.id AND t.monthid=m.id AND a.aircraftid ='XY-WTR'
GROUP BY m.y;
```

Third Query:

```
SELECT l.month AS Month,
1000*SUM(l.pirep+l.marep)/SUM(a.FH) AS RRRh, 100*SUM(l.pirep+l.marep)/SUM(a.FC) AS
RRc,
1000*SUM(l.pirep)/SUM(a.FH) AS PRRh, 1000*SUM(l.marep)/SUM(a.FH) AS MRRh,
100*SUM(l.pirep)/SUM(a.FC) AS PRRc, 100*SUM(l.marep)/SUM(a.FC) AS MRRc
FROM (
    SELECT l.aircraftid AS aircraft, l.monthid AS Month,
    SUM(CASE WHEN p.ROLE = 'P' THEN counter ELSE 0 END) AS PIREP,
    SUM(CASE WHEN p.ROLE = 'M' THEN counter ELSE 0 END) AS MAREP
    FROM LogBookReporting l
    INNER JOIN PeopleDimension p ON l.personid = p.id
    INNER JOIN AircraftDimension d ON l.aircraftid = d.id
    WHERE d.model = '777'
    GROUP BY l.aircraftid, l.monthid
) l
INNER JOIN (
    SELECT a.aircraftid AS aircraft, t.monthid AS Month,
    SUM(a.FlightHours) AS FH, SUM(a.FlightCycles) AS FC
    FROM AircraftUtilization a
    INNER JOIN TemporalDimension t ON a.timeid = t.id
    GROUP BY a.aircraftid, t.monthid
) a ON l.aircraft = a.aircraft AND l.month = a.month
GROUP BY l.month;
```

Fourth query:

```
SELECT d.model AS Model,
1000*SUM(l.marep)/SUM(a.FH) AS MRRh, 100*SUM(l.marep)/SUM(a.FC) AS MRRc
FROM (
    SELECT l.AircraftID AS aircraft, l.monthID AS Month, p.airport AS airport,
    SUM(CASE WHEN p.ROLE = 'M' THEN counter ELSE 0 END) AS MAREP
    FROM LogBookReporting l
    INNER JOIN PeopleDimension p ON l.personid = p.id
```

```

WHERE p.airport = 'KRS'
GROUP BY l.aircraftID, l.monthID, p.airport
)l
INNER JOIN (
SELECT a.aircraftid AS aircraft, t.monthid AS Month,
SUM(a.FlightHours) AS FH, SUM(a.FlightCycles) AS FC
FROM AircraftUtilization a
INNER JOIN TemporalDimension t ON a.timeid = t.id
GROUP BY a.aircraftid, t.monthid
) a ON l.aircraft = a.aircraft AND l.month = a.month
INNER JOIN AircraftDimension d ON l.aircraft = d.id
GROUP BY d.model;

```

➤ Data access structures (i.e., indexes) that optimize the DW:

1. ALTER TABLE AircraftDimension ADD PRIMARY KEY (id) USING INDEX PCTFREE 33;
 2. ALTER TABLE PeopleDimension ADD PRIMARY KEY (id) USING INDEX PCTFREE 33;
 3. CREATE BITMAP INDEX a_model ON AircraftUtilization(AircraftDimension.model) FROM AircraftUtilization, AircraftDimension WHERE AircraftDimension.id = AircraftUtilization.aircraftid PCTFREE 0;
 4. CREATE BITMAP INDEX a_aircraft ON AircraftUtilization(aircraftid) PCTFREE 0;
 5. CREATE BITMAP INDEX l_model ON LogbookReporting(AircraftDimension.model) FROM LogbookReporting, AircraftDimension WHERE AircraftDimension.id = LogbookReporting.aircraftid PCTFREE 0;
 6. CREATE BITMAP INDEX l_airport ON LogbookReporting(PeopleDimension.airport) FROM LogbookReporting, PeopleDimension WHERE PeopleDimension.id = LogbookReporting.personid PCTFREE 0;
- A join algorithm that we used in this project is join-index because in this algorithm we do not need to have access to a table that we defined an index for. We defined bitmap indexes on large tables like; AircraftUtilization and LogbookReporting that have 528 and 1280 blocks respectively (3, 5, 6).
 - Therefore we do not need to go through all these tables (address) and we just go through smaller tables (values) that in this project are AircraftDimension and PeopleDimension that each of them has 8 blocks.
 - We used bitmap because it reduced storage requirements compared to other indexes.