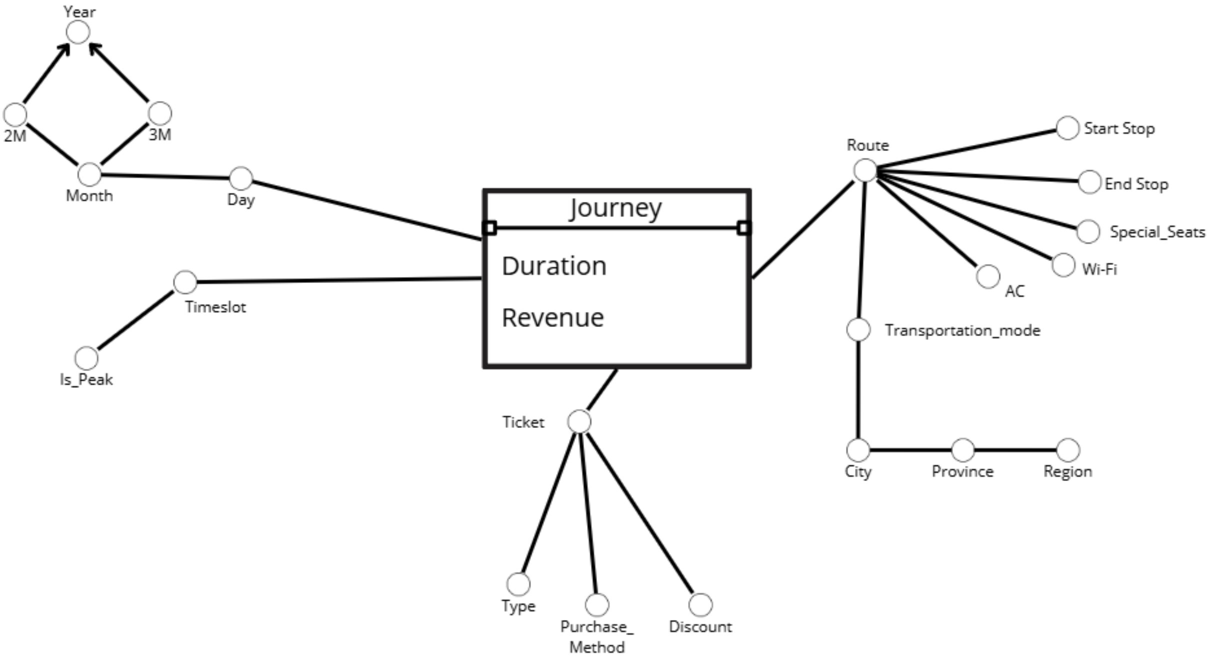
Data Science and Database Technology

Homework #1 - REVALOR RICCARDO - s339423

Conceptual Schema



Logical Schema

TIMESLOT (**TSID**, Is_Peak)

TIME (**TID**, Day, Month, 2M, 3M, Year)

ROUTE (**RID**, Transporation_Mode, City, Province, Region, AC, Wi-Fi, Special_Seats, Start_Stop, End_Stop)

TICKET (**TKID**, Type, Purchase_method, Discount)

JOURNEY (**TSID, TID, RID, TKID**, Duration, Revenue)

Extended-SQL Queries

Query a: *Separately for each transportation mode and for each month of the year, analyze: the average daily number of tickets, the cumulative number of tickets from the beginning of the year, and the percentage of tickets using each transportation mode over the total number of tickets in that month.*

```
SELECT R.Transportation_Mode, T.Year, T.Month,
COUNT(DISTINCT TKID) / COUNT(DISTINCT Day) AS AvgNumTickets,
SUM(COUNT(DISTINCT TKID)) OVER (PARTITION BY Year ORDER BY Month ROWS UNBOUNDED PRECEDIN
G)
AS TrailingYearlyNumTickets,
COUNT(DISTINCT TKID) * 100.0 / SUM(COUNT(DISTINCT TKID)) OVER (PARTITION BY Year, Month)
AS PercentageTickets
FROM ROUTE R, TIME T, JOURNEY J
WHERE J.TID = T.TID AND J.TRID = TR.TRID
GROUP BY TR.Transportation_Mode, T.Year, T.Month
```

Query b: *Considering journeys from 2022, separately for each mode and city, analyze: the average journey duration, the total revenue generated from that city, the percentage of total revenue contributed by each route for the corresponding mode in a city, and assign a rank to each route within its transportation mode based on the total revenue generated in decreasing order.*

```
SELECT SUM(SUM(Duration)) OVER (PARTITION BY Transport_Mode, City) /
SUM(COUNT(*)) OVER (PARTITION BY Transport_Mode, City)
AS AvgDuration,
SUM(SUM(Revenue)) OVER (PARTITION BY City) AS RevByCity,
SUM(Revenue) * 100.0 / SUM(SUM(Revenue)) OVER (PARTITION BY Transport_Mode, City)
AS PercentageOfRevByTransportMode,
RANK() OVER
(PARTITION BY Transportation_Mode, RID ORDER BY SUM(Revenue), DESC)
AS RevRanking
FROM JOURNEY J, ROUTE R, TIME T
WHERE J.TID = T.TID AND J.RID = R.RID
AND Year = 2022
GROUP BY Transportation_Mode, City, RID
```

Materialized View Implementation

Frequent Queries Analysis

A)

Separately for each transportation mode and for each month, analyze the average daily number of tickets.

```
SELECT Transportation_Mode, Year, Month,
COUNT(DISTINCT TKID) / COUNT(DISTINCT Day)
FROM JOURNEY J, TIME T, ROUTE R, TICKET TK
WHERE J.TID = T.TID
AND J.RID = R.RID
GROUP BY Transportation_Mode, Year, Month
```

GROUP BY: Transportation_Mode, Year, Month
WHERE: /
Measures: COUNT(DISTINCT TKID), COUNT(DISTINCT Day)
FROM: JOURNEY, TIME, ROUTE

B) *Separately for each transportation mode and for each month, analyze the cumulative number of tickets from the beginning of the year.*

```
SELECT  Transportation_Mode, Year, Month,
SUM(COUNT(DISTINCT TKID))
OVER (PARTITION BY Year ORDER BY Month ROWS UNBOUNDED PRECEDING)
AS TrailingYearlyNumTickets
FROM JOURNEY J,  TIME T, ROUTE R, TICKET TK
WHERE J.TID = TR.TID
AND J.RID = R.TRID
GROUP BY  Transportation_Mode, Year, Month
```

GROUP BY: Transportation_Mode, Year, Month
WHERE: /
Measures: SUM(COUNT(DISTINCT TKID))
FROM: JOURNEY, TIME, ROUTE

C) *Separately for each transportation mode and for each month, analyze the total number of tickets sold, the total revenue, and the average revenue.*

```
SELECT Transportation_Mode, Year, Month,
COUNT(DISTINCT TKID),
SUM(Revenue),
SUM(Revenue) / COUNT(DISTINCT TKID)
FROM JOURNEY J, ROUTE R, TIME T
WHERE J.TID = T.TID
AND J.RID = R.TRID
GROUP BY  Transportation_Mode, Year, Month
```

GROUP BY: Transportation_Mode, Year, Month
WHERE: /
Measures: COUNT(DISTINCT TKID), SUM(Revenue)
FROM: JOURNEY, TRANSPORTATION, TIME

D) *Separately for each transportation mode and for each month, analyze the total number of tickets sold, the total revenue, and the average revenue for the year 2024.*

```
SELECT Transportation_Mode, Year, Month,
COUNT(DISTINCT TKID),
SUM(Revenue) / COUNT(DISTINCT TKID),
SUM(Revenue)
```

```
FROM JOURNEY J, ROUTE R, TIME T
WHERE J.TID = T.TID
AND J.RID = R.TRID
AND T.Year = 2024
GROUP BY  Transportation_Mode, Year, Month
```

GROUP BY: Transportation_Mode, Year, Month
WHERE: Year
Measures: COUNT(DISTINCT TKID), SUM(Revenue)
FROM: JOURNEY, ROUTE, TIME

E) *Analyze the percentage of tickets related to each transportation mode and month over the total number of tickets of the month for each transportation mode.*

```
SELECT COUNT(DISTINCT TKID) * 100.0 /
SUM(COUNT(DISTINCT TKID))
OVER (PARTITION BY Year, Month)
FROM JOURNEY J, TIME T, ROUTE R
WHERE J.RID =R.TRID
AND J.TID = T.TID
AND J.TKID = TK.TKID
GROUP BY Transportation_Mode, Year, Month
```

GROUP BY: Transporation_Mode, Year, Month
WHERE: /
Measures: COUNT(DISTICT TKID), SUM(COUNT(DISTICT TKID))
FROM JOURNEY, ROUTE, TIME

Materialized View Creation

```
CREATE MATERIALIZED VIEW MV1
BUILD IMMEDIATE
REFRESH FAST ON COMMIT
AS
(SELECT Transportation_Mode, Year, Month,
COUNT(DISTINCT TKID) AS SoldTickets,
COUNT(DISTINCT Day) AS DistinctDays,
SUM(Revenue) AS TotRevenue
FROM JOURNEY J, ROUTE R, TIME T
WHERE J.RID = R.RID
AND J.TID = T.TID
GROUP BY  Transportation_Mode, Year, Month);
```

Materialized View Logs

```
CREATE MATERIALIZED VIEW LOG ON JOURNEY
WITH SEQUENCE, ROWID
(TKID, Revenue)
INCLUDING NEW VALUES;

CREATE MATERIALIZED VIEW LOG ON TIME
WITH SEQUENCE, ROWID
(TID, Day, Year, Month)
INCLUDING NEW VALUES;

CREATE MATERIALIZED VIEW LOG ON ROUTE
WITH SEQUENCE, ROWID
(RID, Transportation_Mode)
INCLUDING NEW VALUES;
```

Operations on the Table JOURNEY influencing the Materialized View

- INSERT: COUNT(DISTINCT TKID) will be potentially increased if the ticket is new, SUM(Revenue) will be increased.
- DELETE: COUNT(DISTINCT TKID) will be potentialy decreased, SUM(Revenue) will be decreased.
- UPDATE: all the measures would be potentially altered some way or another.

Operations on the Table ROUTE influencing the Materialized View

- INSERT: adding a new RID/Transport_Mode alters the grouping of the MV.
- DELETE: deleting a RID/Transport_mode removes a group from the MV and from the output of the queries.
- UPDATE: it changes the grouping of the MV (ex the names of the groups)

Operations on the Table TIME influencing the Materialized View

- INSERT: Adding a new TID for a specific new Day, Month, Year alters the COUNT(Distinct Day) and can alter the groupings by Year and Month.
- DELETE: removing a TID for a specific new Day, Month, YearDays alters the COUNT(Distinct Day) and can alter the groupings by Year and Month.
- UPDATE: it can alters the name of the groups.

Update and management of the view via Trigger assuming that the CREATE MATERIALIZED VIEW command is not available

Table Creation with the CREATE TABLE command and Table pupulation with the already existing records

```
CREATE TABLE MV1 (
Transportation_Mode VARCHAR(15) CHECK (Transportation_Mode IS NOT NULL),
Year NUMBER CHECK (Year IS NOT NULL),
Month NUMBER CHECK (Month IS NOT NULL),
SoldTickets NUMBER CHECK (SoldTickets IS NOT NULL),
DistinctDays NUMBER CHECK (DistincDays IS NOT NULL)
TotRevenue NUMBER CHECK (TotRevenue IS NOT NULL));

INSERT INTO MV1 (Transportation_Mode, Year, Month,
SoldTickets, DistinctDays, TotRevenue)
(SELECT TR.Transportation_Mode, T.Year, T.Month,
COUNT(DISTINCT J.TKID) AS SoldTickets,
COUNT(DISTINCT T.Day) AS DistinctDays,
SUM(J.Revenue) AS TotRevenue
FROM JOURNEY J, TIME T, ROUTE R
WHERE J.TID = T.TID
AND J.RID = R.TRID
GROUP BY R.Transportation_Mode, T.Year, T.Month);
```

PL/SQL Trigger for updating the MV1 Table if a new row has been inserted in the JOURNEY Table

```
CREATE OR REPLACE TRIGGER UpdateMW1Trigger
AFTER INSERT ON JOURNEY
FOR EACH ROW
DECLARE
N NUMBER;
V_DistinctDays NUMBER;
V_SoldTickets NUMBER;
V_T_Mode VARCHAR(15);
V_Year NUMBER;
V_Month NUMBER;
BEGIN

SELECT Transportation_Mode
INTO V_T_Mode
FROM ROUTE R
WHERE :NEW.RID = TR.RID;

SELECT T.Year, T.Month,
INTO V_Year, V_Month
FROM TIME T
WHERE :NEW.TID = T.TID;

SELECT COUNT(*)
INTO N
FROM MV1
WHERE Transportation_Mode = V_T_Mode AND Year = V_Year AND Month = V_Month;

IF (N>0) THEN
```

```
    SELECT COUNT(DISTINCT Day), COUNT(DISTINCT TKID)
    INTO V_DistinctDays, V_SoldTickets
    FROM JOURNEY J, TIME T, ROUTE R
    WHERE J.TID = T.TID AND J.RID = R.TRID
    AND T.Month = V_Month AND T.Year = V_Year
    AND Transportation_Mode = V_T_Mode;

    UPDATE MV1
    SET SoldTickets = SoldTickets + 1,
            TotRevenue = TotRevenue + :NEW.Revenue,
            DistinctDays =  V_DistinctDays,
            SoldTickets = V_SoldTickets
    WHERE Transportation_Mode = V_T_Mode AND Year = V_T_Year AND Month = V_T_Month;
ELSE
    INSERT INTO MV1(Transportation_Mode, Year, Month,
    SoldTickets, DistinctDays, TotRevenue)
    VALUES (V_T_Mode, V_Year, V_Month, 1, 1, :NEW.Revenue);
END IF;

END;
```