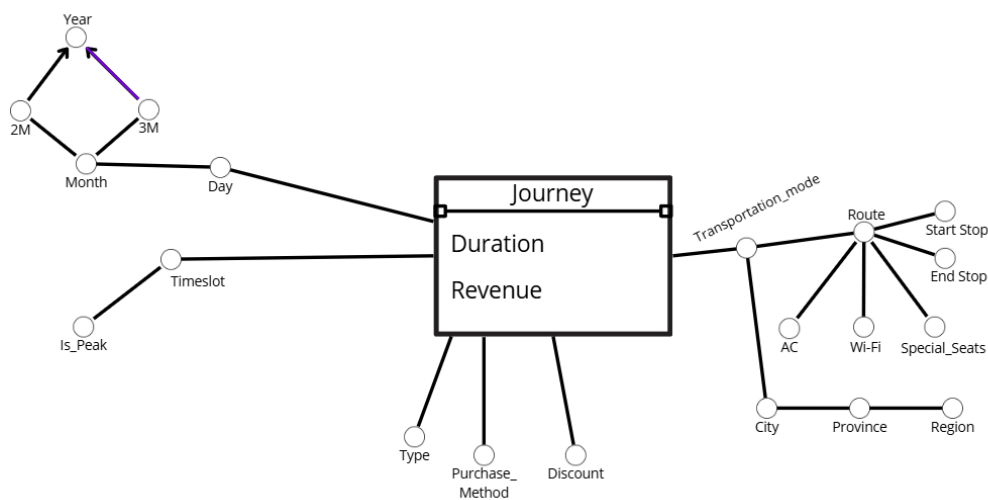


Data Science and Database Technology

Homework #1 - REVALOR RICCARDO - s339423

Conceptual Schema



Logical Schema

TIMESLOT(TSID, Is_Peak)
TIME(TID, Day, Month, 2M, 3M, Year)
TRANSPORTATION(TRID, Transporation_Mode, RID, City, Province, Region, AC, Wi-Fi, Special_Seats, Start_Stop, End_Stop)
JOURNEY(JID, TSID, TID, TRID, Duration, Revenue, Type, Purchase_method, Discount)

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 TR.Transportation_Mode, T.Year, T.Month,
COUNT(JID) / COUNT(DISTINCT Day) AS AvgNumTickets,
SUM(COUNT(JID)) OVER (PARTITION BY Year ORDER BY Month ROWS UNBOUNDED PRECEDING)
AS TrailingYearlyNumTickets,
COUNT(JID) * 100 / SUM(COUNT(JID)) OVER (PARTITION BY Year, Month)
AS PercentageTickets
FROM TRANSPORTATION TR, 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(J.Duration)) OVER (PARTITION BY Transport_Mode, City) / COUNT(JID)
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, Start_Stop, End_Stop ORDER BY SUM(Revenue), DESC)
AS RevRanking
FROM JOURNEY J, TRANSPORTATION TR, TIME T
WHERE J.TID = T.TID AND J.TRID = TR.TRID
AND Year = 2022
GROUP BY Transportation_Mode, City, Start_Stop, End_Stop
```

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(JID) / COUNT(DISTINCT Day)
FROM JOURNEY J, TIME T, TRANSPORTATION TR
WHERE J.TID = T.TID
AND TR.TRID = J.TRID
GROUP BY Transportation_Mode, Year, Month
```

GROUP BY: Transportation_Mode, Year, Month
WHERE:
Measures: COUNT(JID), COUNT(Distinct Day)
FROM: JOURNEY, TIME, TRANSPORTATION

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(JID)) OVER (PARTITION BY Year ORDER BY Month ROWS UNBOUNDED PRECEDING)
AS TrailingYearlyNumTickets
FROM JOURNEY J, TIME T, TRANSPORTATION TR
WHERE J.TID = TR.TID AND J.TRID = TR.TRID
GROUP BY Transportation_Mode, Year, Month
```

GROUP BY: Transportation_Mode, Year, Month
WHERE:
Measures: SUM(COUNT(JID))
FROM: JOURNEY, TIME, TRANSPORTATION

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(JID), AVG(Revenue), SUM(Revenue)
FROM JOURNEY J, TRANSPORTATION TR, TIME T
WHERE J.TID = T.TID AND J.TRID = TR.TRID
GROUP BY Transportation_Mode, Year, Month
```

GROUP BY: Transportation_Mode, Year, Month
WHERE:
Measures: COUNT(JID), SUM(Revenue) / COUNT(JID), 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(JID), SUM(Revenue) / COUNT(JID), SUM(Revenue)
FROM JOURNEY J, TRANSPORTATION TR, TIME T
WHERE J.TID = T.TID AND J.TRID = TR.TRID
AND T.Year = 2024
GROUP BY Transportation_Mode, Year, Month
```

GROUP BY: Transportation_Mode, Year, Month
WHERE: Year
Measures: COUNT(JID), SUM(Revenue)
FROM: JOURNEY, TRANSPORTATION, 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(JID) * 100.0 / SUM(COUNT(JID))
OVER (PARTITION BY Year, Month)
FROM JOURNEY J, TIME T, TRANSPORTATION TR
WHERE J.TRID = TR.TRID AND J.TID = T.TID
GROUP BY Transportation_Mode, Year, Month
```

GROUP BY: Transporation_Mode, Year, Month

WHERE:

Measures: COUNT(JID), SUM(COUNT(JID))

FROM JOURNEY, TRANSPORTATION, TIME

Materialized View Creation

```
CREATE MATERIALIZED VIEW MV1
BUILD IMMEDIATE
REFRESH FAST ON COMMIT
AS
(SELECT Transportation_Mode, Year, Month, COUNT(JID)
AS SoldTickets,
COUNT(Distinct Day) AS DistinctDays, SUM(Revenue)
AS TotRevenue
FROM JOURNEY J, TRANSPORTATION TR, TIME T
WHERE J.TRID = TR.TRID AND J.TID = T.TID
GROUP BY Transportation_Mode, Year, Month);
```

Materialized View Logs

```
CREATE MATERIALIZED VIEW LOG ON JOURNEY
WITH SEQUENCE, ROWID
(JID, 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 TRANSPORTATION
WITH SEQUENCE, ROWID
(TRID, Transportation_Mode)
INCLUDING NEW VALUES;
```

Operations on the Table JOURNEY influencing the Materialized View

- INSERT: COUNT(JID) will be increased, SUM(Revenue) will be increased.
- DELETE: COUNT(JID) will be decreased, SUM(Revenue) will be decreased.
- UPDATE: all the measures would be potentially altered some way or another.

Operations on the Table TRANSPORTATION influencing the Materialized View

- INSERT: adding a new TRID/Transport_Mode alters the grouping of the MW.
- DELETE: deleting a TRID/Transport_mode removes a group from the MW and from the output of the queries.
- UPDATE: it changes the grouping of the MW (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 alter 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 population 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 (DistinctDays 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(J.JID) AS SoldTickets, COUNT(DISTINCT T.Day)  
AS DistinctDays, SUM(J.Revenue) AS TotRevenue  
FROM JOURNEY J, TIME T, TRANSPORTATION TR  
WHERE J.TID = T.TID AND J.TRID = TR.TRID  
GROUP BY TR.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_T_Mode VARCHAR(15);  
  V_Year NUMBER;  
  V_Month NUMBER;  
BEGIN  
  
  SELECT Transportation_Mode  
  INTO V_T_Mode  
  FROM TRANSPORTATION TR  
  WHERE :NEW.TRID = TR.TRID;  
  
  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)  
    INTO V_DistinctDays  
    FROM JOURNEY J, TIME T, TRANSPORTATION TR  
    WHERE J.TID = T.TID AND J.TRID = TR.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  
    WHERE Transportation_Mode = V_T_Mode AND Year = V_Year AND Month = V_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;
```