

Logical Schema:

TIMESLOT(TSID, Is\_Peak)

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

TRANSPORTATION(TRID, Transporation\_Mode, RID, City, Province, Region)

ROUTE(RID, AC, Wi-Fi, Special\_Seats, Start\_Stop, End\_Stop)

JOURNEY(TSID, TID, TRID, JID, Duration, Revenue, Type, Purchase\_method, Discount)

Task 2 Extended SQL Queries

a. 1 journey = 1 ticket

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

b.

WITH RevenuePerRoute AS (  
SELECT TR.Transportation\_Mode, R.RID, SUM(J.Revenue)   
FROM JOURNEY J, ROUTE R, TRANSPORTATION TR, TIME T  
WHERE J.TRID = TR.TRID AND J.RID = R.RID AND T.TID = J.TID  
AND T.Year = 2022  
 TR.Transportation\_Mode, R.RID

),

RevenuePerTRModeAndCity AS (

SELECT TR.Transportation\_Mode, TR.City , SUM(J.Revenue) AS RPTC  
FROM TRANSPORTATION TR, JOURNEY J, TIME T  
WHERE J.TRID = T.TRID AND J.TID = T.TID  
AND T.Year = 2022  
GROUP BY TR.Transportation\_Mode, TR.City  
)

SELECT Transportation\_Mode, City, AVG(DURATION) AS JourneyDuration,   
SUM(REVENUE) OVER (PARTITION BY City) AS RevenuePerCity,  
RevenuePerRoute.RPR \* 100.0 / RevenuePerTRModeAndCity.RPTC AS PercentageRevenuePerRoute,  
RANK() OVER (ORDER BY RevenuePerRoute.RPR, DESC) AS RankRoute  
FROM JOURNEY J , TRANSPORTATION TR, TIME T, RevenuePerRoute RevR, RevenuePerTRModeAndCity RevTr  
WHERE J.TRID = TR.TRID AND J.TID = T.TID  
AND RevR.RID = J.RID  
AND RevTr.Transportation\_Mode = TR.Transportation\_Mode  
AND T.Year = 2022  
GROUP BY Transportation\_Mode, City,

3)

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  
WHERE J.TID = T.TID  
GROUP BY Transportation\_Mode, Year, Month

GROUP BY: Transportation\_Mode, Month  
WHERE:  
Measures: COUNT(JID), COUNT(Distinct Day)  
FROM: JOURNEY, TIME

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  
WHERE J.TID = TR.TID  
GROUP BY Transportation\_Mode, Year, Month

GROUP BY: Transportation\_Mode, Year, Month  
WHERE:   
Measures: SUM(COUNT(JID))  
FROM: JOURNEY, TIME  
  
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), AVG(Revenue), 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), AVG(Revenue), 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, Month  
WHERE: Year  
Measures:COUNT(JID), AVG(Revenue), 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

CREATE MATERIALIZED VIEW MV1  
BUILD IMMEDIATE   
REFRESH FAST ON COMMIT   
AS   
(SELECT Transportation\_Mode, Year, Month, COUNT(JID), SUM(COUNT(JID)), COUNT(Distinct Day), AVG(Revenue), SUM(Revenue)  
FROM JOURNEY J, TRANSPORTATION TR, TIME T  
WHERE J.TRID = TR.TRID AND J.TID = T.TID  
GROUP BY Transportation\_Mode, Year, Month)

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 for Table JOURNEY influencing the Materialized View:

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

Operations for 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 for Table TIME influencing the Materialized View:

* INSERT: Adding a new TID for a specific Day, Month, Year alters the COUNT(Distinct Day).
* DELETE: removing a TID for a specific Day, Month, YearDays alters the COUNT(Distinct Day).
* UPDATE: changing Days values alters COUNT(Distinct Day)

4)  
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),  
CumulativeSoldTickets CHECK (CumulativeSoldTickets IS NOT NULL),  
DistinctDays NUMBER CHECK (DistincDays IS NOT NULL)  
TotRevenue NUMBER CHECK (TotRevenue IS NOT NULL),  
AvgRevenue NUMBER CHECK (AvgRevenue IS NOT NULL));

INSERT INTO MV1 (Transportation\_Mode, Year, Month, SoldTickets, CumulativeSoldTickets, DistinctDays, TotRevenue, AvgRevenue)   
(SELECT TR.Transportation\_Mode, T.Year, T.Month, COUNT(J.JID) AS SoldTickets, SUM(COUNT(J.JID)) OVER (PARTITION BY T.Year, T.Month ORDER BY T.Month ROWS UNBOUNDED PRECEDING) AS CumulativeSoldTickets, COUNT(DISTINCT T.Day) AS DistinctDays, SUM(J.Revenue) AS TotRevenue, AVG(J.Revenue) AS AvgRevenue   
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);

Trigger to update changes after an insertion on the JOURNEY table:  
  
CREATE OR REPLACE TRIGGER UpdateMW1Trigger  
AFTER INSERT ON JOURNEY  
FOR EACH ROW  
DECLARE  
BEGIN  
N NUMBER;  
V\_