**1:**

**Schema concettuale:**

Immagine che contiene diagramma, testo, linea, Piano

Descrizione generata automaticamente

N.B. Si è supposto che le misure Income, TicketsNumber e TravelDuration abbiano subito un processo di ETL per la rappresentazione qua proposta. TravelDuration è la somma dei tempi relativi a quel fatto (tragitto, spostamento).

**Modello logico:**

**Immagine che contiene testo, diagramma, Piano, Disegno tecnico

Descrizione generata automaticamente**

FACTS(LocationId, TypeOfDiscountId, TimeSlotId, TimedimId, RouteId, TicketPurchaseModeId, TypeOfTransportModeId, TypeOfTicketId, DayIncome, DayTicketsNumber, TravelDuration)

TIMEDIM(Id, DayDate, DateMonth, DateThreeMonth, DateTwoMonth, Year)

TIMESLOT(Id, TimeSlot, IsPeakTime)

ROUTE(Id, DepartureStop, ArrivalStop, Wifi, SpecialSiting, AirConditioning)

TYPEOFTRANSPORTMODE(Id, Name)

TYPEOFTICKET(Id, Name)

LOCATION(Id, City, Province, Region)

TYPEOFDISCOUNT(Id, Name)

TICKETPURCHASEMODE(Id, Name)

**2:**

**a):**

SELECT T.Year, T.DateMonth,TTM.Name,

    SUM(SUM(F.DayTicketsNumber)) OVER(PARTITION BY T.DateMonth, TTM.Name) / COUNT(DISTINCT T.DayDate) AS AvgDayTicketsNumber,

    SUM(SUM(F.DayTicketsNumber)) OVER(PARTITION BY T.Year, TTM.Name ORDER BY T.Year, T.DateMonth ROWS UNBOUNDED PRECEDING) AS CumulativeTicketsSumForMonth,

    SUM(F.DayTicketsNumber) / SUM(SUM(F.DayTicketsNumber)) OVER(PARTITION BY T.YEAR, T.DateMonth) \* 100 AS PercentualTicketsForTypeTransportaionModeForMonth

FROM Facts F, TypeOfTransportMode TTM, Timedim T

WHERE F.TypeOfTransportModeId = TTM.Id and F.TimedimId = T.Id

GROUP BY TTM.Name, T.DateMonth, T.Year ORDER BY T.Year, T.DateMonth, TTM.Name

**b):**

SELECT L.City, R.Id AS RouteId, R.DepartureStop, R.ArrivalStop, TTM.Name, AVG(F.TravelDuration) AS AvgTravelDuration,

    SUM(SUM(DayIncome)) OVER(PARTITION BY L.City) AS TotalIncomeForCity,

    SUM(DayIncome) / SUM(SUM(DayIncome)) OVER(PARTITION BY L.City, TTM.Name) \* 100 AS PercIncomeRouteOnCity,

    RANK() OVER(PARTITION BY L.CITY, TTM.Name ORDER BY SUM(DayIncome) DESC) AS RankPercIncome

FROM Facts F, TypeOfTransportMode TTM, Location L, Timedim T, Route R

WHERE F.TypeOfTransportModeId = TTM.Id and F.LocationId = L.Id and F.TimedimId = T.Id and F.RouteId = R.Id and T.Year >= 2022

GROUP BY L.City, R.Id, R.DepartureStop, R.ArrivalStop, TTM.Name ORDER BY L.City, R.Id, R.DepartureStop, R.ArrivalStop, TTM.Name

**3:**

**a):**

CREATE MATERIALIZED VIEW FACTS\_TTM\_DateMonth

BUILD IMMEDIATE

REFRESH FAST ON COMMIT

AS

SELECT TTM.Name, T.DateMonth, T.Year, SUM(F.DayIncome) AS MonthIncome, SUM(DayTicketsNumber) AS MonthTicketsNumber

FROM Facts F, TypeOfTransportMode TTM, Timedim T

WHERE F.TypeOfTransportModeId = TTM.Id and F.TimedimId = T.Id

GROUP BY  TTM.Name, T.DateMonth, T.Year

**b):**

CREATE MATERIALIZED VIEW LOG ON Facts

WITH SEQUENCE, ROWID

(TypeOfTransportModeId, TimedimId, DayIncome, DayTicketsNumber)

INCLUDING NEW VALUES;

CREATE MATERIALIZED VIEW LOG ON TypeOfTransportMode

WITH SEQUENCE, ROWID

(Id, Name)

INCLUDING NEW VALUES;

CREATE MATERIALIZED VIEW LOG ON Timedim

WITH SEQUENCE, ROWID

(Id, DateMonth, Year)

INCLUDING NEW VALUES;

**c):** INSERT INTO Timedim (…) VALUES(…)

INSERT INTO TypeOfTransportMode (…) VALUES(…)

INSERT INTO Facts (…) VALUES(…)

**4:**

**a, b, c):**

CREATE TABLE FACTS\_TTM\_DATEMONTH\_VIEW\_TABLE (

    Name VARCHAR(20),

    DateMonth NUMBER(2),

    Year NUMBER(4),

    MonthIncome NUMBER(10, 2),

    MonthTicketsNumber NUMBER,

    PRIMARY KEY (Name,DateMonth,Year)

)

INSERT INTO FACTS\_TTM\_DATEMONTH\_VIEW\_TABLE (Name, DateMonth, Year, MonthIncome, MonthTicketsNumber)

(

    SELECT TTM.Name, T.DateMonth, T.Year, SUM(F.DayIncome), SUM(DayTicketsNumber)

    FROM Facts F, TypeOfTransportMode TTM, Timedim T

    WHERE F.TypeOfTransportModeId = TTM.Id and F.TimedimId = T.Id

    GROUP BY  TTM.Name, T.DateMonth, T.Year

)

CREATE OR REPLACE TRIGGER TRIGGER\_FACTS\_TTM\_DATEMONTH\_VIEW\_TABLE

AFTER INSERT ON Facts

FOR EACH ROW

DECLARE

    TmpName VARCHAR(20);

    TmpDateMonth NUMBER(2);

    TmpYear NUMBER(4);

    N INTEGER;

BEGIN

        SELECT Name INTO TmpName

        FROM TypeOfTransportMode

        WHERE Id = :NEW.TypeOfTransportModeId;

        SELECT DateMonth, Year INTO TmpDateMonth, TmpYear

        FROM Timedim

        WHERE Id = :NEW.TimedimId;

        SELECT COUNT(\*) INTO N

        FROM FACTS\_TTM\_DATEMONTH\_VIEW\_TABLE

        WHERE Name = TmpName AND DateMonth = TmpDateMonth AND Year = TmpYear;

        IF N > 0 THEN

            UPDATE FACTS\_TTM\_DATEMONTH\_VIEW\_TABLE

            SET MonthTicketsNumber = MonthTicketsNumber + :NEW.DayTicketsNumber,

                MonthIncome = MonthIncome + :NEW.DayIncome

            WHERE Name = TmpName AND DateMonth = TmpDateMonth AND Year = TmpYear;

        ELSE

            INSERT INTO FACTS\_TTM\_DATEMONTH\_VIEW\_TABLE (Name, DateMonth, Year, MonthIncome, MonthTicketsNumber)

            VALUES (TmpName, TmpDateMonth, TmpYear, :NEW.DayIncome, :NEW.DayTicketsNumber);

        END IF;

END;

/

**d):** Solo le INSERT in Facts attivano il trigger sopra mostrato