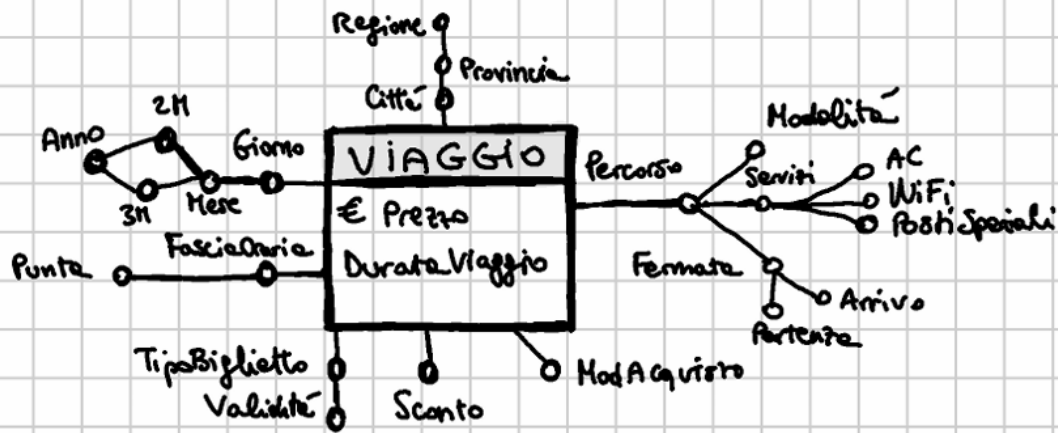


--! ES.1

PROGETTAZIONE CONCETTUALE



/* SCHEMA LOGICO */

TRASPORTO(TrasportoID, Percorso, Modalità, Servizi, FermataPartenza, FermataArrivo)
 TEMPO(TempoID, Giorno, Mese, 2M, 3M, Anno)
 FASCIAORARIA (FasciaOraria, Punta)
 LUOGO(LuogoID, Città, Provincia, Regione)
 BIGLIETTO(TipoBiglietto, Validità)

/* TABELLA DEI FATTI */

VIAGGIO(TrasportoID, TempoID, FasciaOraria, LuogoID, TipoBiglietto, Prezzo, Durata, Sconto, ModAcquisto)

--! ES.2

/* QUERY a */

```

SELECT Modalità, Mese, Anno, COUNT(*)/COUNT(DISTINCT Giorno) AS
BigliettiMediGiornalieri,
SUM(COUNT(*)) OVER(PARTITION BY Anno ROWS UNBOUNDED PRECEDING) AS
BigliettiCumulativiAnno,
100*COUNT(*)/SUM(COUNT(*)) OVER(PARTITION BY Mese) AS PercBigliettiModalitàMese
FROM TRASPORTO TR, VIAGGIO V, TEMPO T
WHERE TR.TrasportoID = V.TrasportoID
AND T.TempoID = V.TempoID
GROUP BY Modalità, Mese, Anno;
  
```

/* QUERY b */

```

SELECT Percorso, Modalità, Città,
SUM(SUM(Durata)) OVER(PARTITION BY Modalità, Città)/SUM(COUNT(*))
OVER(PARTITION BY Modalità, Città) AS DurataMediaCittàModalità,
SUM(SUM(Prezzo)) OVER(PARTITION BY Città) AS RicaviTotCittà,
100*SUM(Prezzo)/SUM(SUM(Prezzo)) OVER(PARTITION BY Modalità, Città) AS
PercRicaviPercorso,
RANK() OVER(PARTITION BY Modalità ORDER BY SUM(Prezzo) DESC) AS RankPercorsi
FROM LUOGO L, TEMPO T, VIAGGIO V, TRASPORTO TR
WHERE L.LuogoID = V.LuogoID
AND T.TempoID = V.TempoID
AND TR.TrasportoID = V.TrasportoID
  
```

```
AND Anno >= 2022
GROUP BY Percorso, Modalità, Città;
```

--! ES.3

```
/* QUERY DI INTERESSE */
SELECT Modalità, Mese, COUNT(*)/COUNT(DISTINCT Giorno) AS BigliettiMediGiornalieri
FROM TRASPORTO TR, VIAGGIO V, TEMPO T
WHERE TR.TrasportoID = V.TrasportoID
AND T.TempoID = V.TempoID
GROUP BY Modalità, Mese;
```

```
SELECT Modalità, Mese, Anno, SUM(COUNT(*)) OVER(PARTITION BY Anno ROWS UNBOUNDED
PRECEDING) AS BigliettiCumulativiAnno
FROM TRASPORTO TR, VIAGGIO V, TEMPO T
WHERE TR.TrasportoID = V.TrasportoID
AND T.TempoID = V.TempoID
GROUP BY Modalità, Mese, Anno;
```

```
SELECT Modalità, Mese, COUNT(*) AS BigliettiTot,
SUM(Prezzo) AS IncassoTot,
AVG(Prezzo) AS IncassoMedio
FROM TRASPORTO TR, VIAGGIO V, TEMPO T
WHERE TR.TrasportoID = V.TrasportoID
AND T.TempoID = V.TempoID
GROUP BY Modalità, Mese;
```

```
SELECT Modalità, Mese, COUNT(*) AS BigliettiTot,
SUM(Prezzo) AS IncassoTot,
AVG(Prezzo) AS IncassoMedio
FROM TRASPORTO TR, VIAGGIO V, TEMPO T
WHERE TR.TrasportoID = V.TrasportoID
AND T.TempoID = V.TempoID
AND Anno = 2024
GROUP BY Modalità, Mese;
```

```
--TODO capire se è sensata... non si capisce la richiesta
SELECT Modalità, Mese, 100*COUNT(*)/SUM(COUNT(*)) OVER(PARTITION BY Mese) AS
PercBigliettiModalitàMese,
100*COUNT(*)/SUM(COUNT(*)) OVER(PARTITION BY Modalità) AS
PercBigliettiMeseModalità
FROM TRASPORTO TR, VIAGGIO V, TEMPO T
WHERE TR.TrasportoID = V.TrasportoID
AND T.TempoID = V.TempoID
GROUP BY Modalità, Mese;
```

```
/* VM */
CREATE MATERIALIZED VIEW VM1
BUILD IMMEDIATE
-- REFRESH FAST ON COMMIT
AS
```

```

SELECT Modalità, Mese, Anno, SUM(Prezzo) AS IncassoTot,
      AVG(Prezzo) AS IncassoMedio
FROM TRASPORTO TR, VIAGGIO V, TEMPO T
WHERE TR.TrasportoID = V.TrasportoID
      AND T.TempoID = V.TempoID
GROUP BY Modalità, Mese, Anno;

```

```
/* LOG */
```

```

CREATE MATERIALIZED VIEW LOG ON VIAGGIO
WITH SEQUENCE, ROWID
(TrasportoID*, TempoID*, FasciaOraria*, LuogoID*, TipoBiglietto*, Prezzo)
INCLUDING NEW VALUES;

```

```

CREATE MATERIALIZED VIEW LOG ON TRASPORTO
WITH SEQUENCE, ROWID
(TrasportoID*, Modalità)
INCLUDING NEW VALUES;

```

```

CREATE MATERIALIZED VIEW LOG ON TEMPO
WITH SEQUENCE, ROWID
(TempoID*, Mese, Anno)
INCLUDING NEW VALUES;

```

```
/* Le operazioni che causano l'aggiornamento della VM sono:
```

```

- INSERT
  -> VIAGGIO
  -> TRASPORTO
  -> TEMPO
- UPDATE
  -> VIAGGIO (modifica su prezzo)
  -> TRASPORTO (modifica su modalità)
  -> TEMPO (modifica su mese o anno)
- DELETE*/

```

--! ES.4

```
/* CREAZIONE DELLA TABELLA VM1 */
```

```

CREATE TABLE VM1(
  Modalità      VARCHAR(20),
  Mese          VARCHAR(20),
  Anno          NUMBER,
  IncassoTot    NUMBER,
  IncassoMedio  NUMBER,
);

```

```
/* POPOLAMENTO INIZIALE DI VM1*/
```

```

INSERT INTO VM1 (Modalità, Mese, Anno, IncassoTot, IncassoMedio) (
  SELECT Modalità, Mese, Anno, SUM(Prezzo) AS IncassoTot,
        AVG(Prezzo) AS IncassoMedio
  FROM TRASPORTO TR, VIAGGIO V, TEMPO T
  WHERE TR.TrasportoID = V.TrasportoID

```

```

        AND T.TempoID = V.TempoID
    GROUP BY Modalità, Mese, Anno
);

/* TRIGGER PER PROPAGARE MODIFICHE DI VM1*/

-- trigger che gestisce:

--! -> insert su VIAGGIO
CREATE OR REPLACE TRIGGER insertViaggio
AFTER INSERT ON VIAGGIO
FOR EACH ROW
DECLARE
    N NUMBER = 0; -- usata per il contatore se trova righe cercate
    VarModalità, VarMese VARCHAR(20);
    VarAnno NUMBER;
    VarCountViaggi NUMBER;

BEGIN
    -- inizializzo variabili
    SELECT Modalità INTO VarModalità
    FROM TRASPORTO
    WHERE TrasportoID = :NEW.TrasportoID;

    SELECT Mese INTO VarMese
    FROM TEMPO
    WHERE TempoID = :NEW.TempoID;

    SELECT Anno INTO VarAnno
    FROM TEMPO
    WHERE TempoID = :NEW.TempoID;

    SELECT COUNT(*) INTO VarCountViaggi -- la uso poi per l'IncassoMedio
    FROM VIAGGIO
    WHERE TrasportoID = :NEW.TrasportoID
        AND TempoID = :NEW.TempoID;

    --- controllo se esiste la riga cercata
    SELECT COUNT(*) INTO N
    FROM VM1
    WHERE Modalità = VarModalità
        AND Mese = VarMese
        AND Anno = VarAnno;

    --- se esiste update
    IF (N > 0) THEN
        UPDATE VM1
        SET IncassoTot = IncassoTot + :NEW.Prezzo,
            IncassoMedio = (IncassoTot + :NEW.Prezzo)/VarCountViaggi
        WHERE Modalità = VarModalità

```

```

        AND Mese = VarMese
        AND Anno = VarAnno;
-- altrimenti insert
ELSE
    INSERT INTO VM1 (Modalità, Mese, Anno, IncassoTot, IncassoMedio) (
        SELECT Modalità, Mese, Anno, :NEW.Prezzo, :NEW.Prezzo
        FROM TRASPORTO TR, TEMPO T
        WHERE TR.TrasportoID = :NEW.TrasportoID
            AND T.TempoID = :NEW.TempoID
    );
END IF;
END;

--! -> update su Prezzo di VIAGGIO
CREATE OR REPLACE TRIGGER updatePrezzoViaggio
AFTER UPDATE OF Prezzo ON VIAGGIO
FOR EACH ROW
DECLARE
    VarModalità, VarMese VARCHAR(20);
    VarAnno NUMBER;
    VarCountViaggi NUMBER;

BEGIN
    -- inizializzo variabili
    SELECT Modalità INTO VarModalità
    FROM TRASPORTO
    WHERE TrasportoID = :NEW.TrasportoID;

    SELECT Mese, Anno INTO VarMese, VarAnno
    FROM TEMPO
    WHERE TempoID = :NEW.TempoID;

    SELECT COUNT(*) INTO VarCountViaggi    -- la uso poi per l'IncassoMedio
    FROM VIAGGIO
    WHERE TrasportoID = :NEW.TrasportoID
        AND TempoID = :NEW.TempoID;

    -- faccio update
    UPDATE VM1
    SET IncassoTot = IncassoTot - :OLD.Prezzo + :NEW.Prezzo,          -- consigliato
    per totali cumulativi (mantiene l'aggregazione, senza la sola assegnazione del
    nuovo valore)
        IncassoMedio = (IncassoTot - :OLD.Prezzo + :NEW.Prezzo)/VarCountViaggi
    WHERE Modalità = VarModalità
        AND Mese = VarMese
        AND Anno = VarAnno;
END;

--! -> delete su VIAGGIO
CREATE OR REPLACE TRIGGER deleteViaggio

```

```

AFTER DELETE ON VIAGGIO
FOR EACH ROW
DECLARE
    VarModalità, VarMese VARCHAR(20);
    VarAnno NUMBER;
    VarCountViaggi NUMBER;

BEGIN
    -- inizializzo variabili
    SELECT Modalità INTO VarModalità
    FROM TRASPORTO
    WHERE TrasportoID = :OLD.TrasportoID;

    SELECT Mese, Anno INTO VarMese, VarAnno
    FROM TEMPO
    WHERE TempoID = :OLD.TempoID;

    SELECT COUNT(*) INTO VarCountViaggi    -- la uso poi per l'IncassoMedio
    FROM VIAGGIO
    WHERE TrasportoID = :OLD.TrasportoID
        AND TempoID = :OLD.TempoID;

    -- update sui valori di VM1
    UPDATE VM1
    SET IncassoTot = IncassoTot - :OLD.Prezzo,
        IncassoMedio = (IncassoTot - :OLD.Prezzo)/VarCountViaggi
    WHERE Modalità = VarModalità
        AND Mese = VarMese
        AND Anno = VarAnno;

    -- delete dalla VM1 se IncassoTot = 0, in quanto è inutile (ragionamento
    opzionale, altrimenti se non si vuole complicare ulteriormente si può togliere
    questa parte)
    DELETE FROM VM1
    WHERE IncassoTot = 0
        AND Modalità = VarModalità
        AND Mese = VarMese
        AND Anno = VarAnno;

END;

/* Le operazioni che attivano il trigger sono:
-> insert su VIAGGIO
-> update su Prezzo di VIAGGIO
-> delete su VIAGGIO */

```