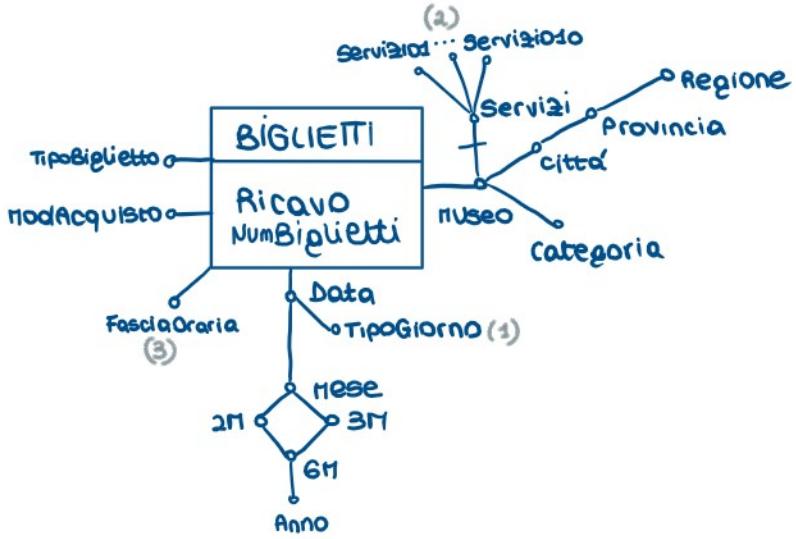
QUADERNO_1

sabato 26 novembre 2022 19:08

1. SCHEMA CONCETTUALE



NOTE

(1) TipoGiorno è un attributo che indica se il giorno è feriale o festivo

(2) Servizi è un attributo configurazione che include 10 tipologie di servizi

(3) La fascia oraria è l'intervallo di validità oraria del biglietto nella data specificata

SCHEMA LOGICO

BIGLIETTI (IdMuseo, IdTempo, IdFasciaOraria, IdModoAcquisto, IdTipoBiglietto, Ricavo, NumBiglietti)

MUSEO (IdMuseo, NomeMuseo, categoria, città, Provincia, Regione, Servizio1, Servizio2, ..., Servizio10)

TEMPO (IdTempo, Data, TipoGiorno, Mese, 2M, 3M, 6M, Anno)

FASCIA_ORARIA (IdFasciaOraria, FasciaOraria)

MOD_ACQUISTO (IdModAcquisto, ModAcquisto)

TIPO_BIGLIETTO (IdTipoBiglietto, TipoBiglietto)

Nota: Le dimensioni degeneri, ossia gli attributi senza gerarchie, possono essere rappresentati nello schema logico o come attributi nella tabella dei fatti oppure come tabelle rappresentate da un id che fa parte della chiave primaria della tabella dei fatti.

2.

a)

```
SELECT TipoBiglietto, Mese,
SUM(Ricavo)/COUNT(DISTINCT Data) AS entrateMedieGiornaliere,
SUM(SUM(Ricavo)) OVER (PARTITION BY TipoBiglietto,Anno
                       ORDER BY Mese
                       ROWS UNBOUNDED PRECEDING)AS entrateCumulative,
100*(SUM(NumBiglietti))/SUM(SUM(NumBiglietti)) OVER (PARTITION BY Mese)AS percentuale
FROM BIGLIETTI B, TIPO_BIGLIETTO TB, TEMPO T
WHERE B.IdTipoBiglietto=TB.IdTipoBiglietto AND B.IdTempo=T.IdTempo
```

```
GROUP BY TipoBiglietto, Mese, Anno
```

b)

```
SELECT B.IdMuseo,TipoBiglietto, SUM(Ricavo)/SUM(NumBiglietti) AS ricavoMedio,
100*SUM(RICAVO)/SUM(SUM(Ricavo)) OVER (PARTITION BY TipoBiglietto)AS percRicavo,
RANK() OVER(PARTITION BY TipoBiglietto ORDER BY (SUM(NumBiglietti)) DESC) as rank
FROM BIGLIETTI B, TIPO_BIGLIETTO TB, TEMPO T, MUSEO M
WHERE B.IdTipoBiglietto=TB.IdTipoBiglietto AND B.IdTempo=T.IdTempo AND B.IdMuseo=M.IdMuseo
AND Anno=2021
GROUP BY B.IdMuseo,TipoBiglietto, Categoria
ORDER BY tipoBiglietto,rank
```

3.

MISURE: SUM(Ricavo), COUNT(DISTINCT Data)
TABELLE: BIGLIETTI, TEMPO, TIPO_BIGLIETTO
GROUP BY: TipoBiglietto, Mese

MISURE: SUM(SUM(Ricavo)) OVER (PARTITION BY TipoBiglietto, Anno
ORDER BY Mese
ROWS UNBOUNDED PRECEDING)
TABELLE: BIGLIETTI, TEMPO, TIPO_BIGLIETTO
GROUP BY: TipoBiglietto, Mese, Anno

MISURE SUM(NumBiglietti), SUM(Ricavo), SUM(Ricavi)/SUM(NumBiglietti)
TABELLE: BIGLIETTI, TEMPO, TIPO_BIGLIETTO
GROUP BY: TipoBiglietto, Mese

MISURE : SUM(NumBiglietti), SUM(Ricavi), SUM(Ricavi)/SUM(NumBiglietti)
TABELLE: BIGLIETTI, TEMPO, TIPO_BIGLIETTO
PREDICATI DI SELEZIONE: Anno=2021
GROUP BY: TipoBiglietto, Mese

MISURE: TipoBiglietto,Mese,100*SUM(NumBiglietti)/SUM(SUM(NumBiglietti))OVER(PARTITION BY Mese)
TABELLE: BIGLIETTI, TEMPO, TIPO_BIGLIETTO
GROUP BY: TipoBiglietto
FROM BIGLIETTI B, TIPO_BIGLIETTO TB, TEMPO_BIGLIETTO T
WHERE B.IdTipoBiglietto=TB.IdTipoBiglietto AND B.IdTempo=T.IdTempo
GROUP BY TipoBiglietto,Mese

3.1)

```
CREATE MATERIALIZED VIEW viewBiglietti
BUILD IMMEDIATE
REFRESH FAST ON COMMIT
AS
SELECT TipoBiglietto,Data, Mese, Anno, SUM(Ricavo) as Ricavo,SUM(NumBiglietti)as NumBiglietti
FROM BIGLIETTI B, TEMPO_BIGLIETTO T, TIPO_BIGLIETTO TB
WHERE B.IdTipoBiglietto=TB.IdTipoBiglietto AND B.IdTempo=T.IdTempo
GROUP BY TipoBiglietto, Data, Mese, Anno;
```

Nota: Per effettuare il 'REFRESH FAST ON COMMIT' i MATERIALIZED VIEW LOG devono essere generati prima della vista.

3.2)

Le tabelle per cui è necessario creare i MATERIALIZED VIEW LOG sono: BIGLIETTI, TEMPO E TIPO_BIGLIETTO.
Gli attributi interessati sono solo quelli presenti nelle query di interesse analizzate precedentemente.

```
CREATE MATERIALIZED VIEW LOG ON BIGLIETTI
WITH SEQUENCE,ROWID
(IdTipoBiglietto,IdTempo,Ricavo,NumBiglietti)
```

```
INCLUDING NEW VALUES;

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

CREATE MATERIALIZED VIEW LOG ON TIPO_BIGLIETTO
WITH SEQUENCE,ROWID
(IdTipoBiglietto,TipoBiglietto)
INCLUDING NEW VALUES;
```

3.3)

Operazioni sulla base dati che causano aggiornamento della materialized view:

- INSERT/UPDATE/DELETE nella TABELLA BIGLIETTI
- INSERT/UPDATE/DELETE nella TABELLA TEMPO
- INSERT/UPDATE/DELETE nella TABELLA TIPO_BIGLIETTO

4.1)

```
CREATE TABLE VM1(
TipoBiglietto VARCHAR(26),
Data DATE CHECK (Data IS NOT NULL),
Mese DATE CHECK (Mese IS NOT NULL),
Anno INTEGER CHECK(Anno IS NOT NULL),
RicaviTot INT CHECK (RicaviTot IS NOT NULL AND RicaviTot>0),
BigliettiTot INT CHECK (BigliettiTot IS NOT NULL AND BigliettiTot>0),
PRIMARY KEY (TipoBiglietto,Data)
)
```

4.2)

```
INSERT INTO VM1(TipoBiglietto,Data,Mese,Anno,RicaviTot,BigliettiTot)(
    SELECT TipoBiglietto,Data, Mese, Anno, SUM(Ricavo) as Ricavi,SUM(NumBiglietti)as NumBiglietti
    FROM BIGLIETTI B, TEMPO_BIGLIETTO T, TIPO_BIGLIETTO TB
    WHERE B.IdTipoBiglietto=TB.IdTipoBiglietto AND B.IdTempo=T.IdTempo
    GROUP BY TipoBiglietto, Data, Mese, Anno
)
```

4.3)

```
CREATE TRIGGER RefreshVM1
        AFTER INSERT ON BIGLIETTI
        FOR EACH ROW
        DECLARE
        varTipoBiglietto VARCHAR2(26);
        varData DATE;
        varMese DATE;
        varAnno INTEGER;
        N INT;

        BEGIN
         -- leggere le tabelle dimensionali per recuperare i valori dell'identificatore della vista materializzata
         -- TipoBiglietto, Data

         SELECT TipoBiglietto INTO varTipoBiglietto
         FROM TIPO_BIGLIETTO
         WHERE IdTipoBiglietto=:NEW.IdTipoBiglietto;

         SELECT Data INTO varData
         FROM TEMPO
         WHERE IdTempo=:NEW.IdTempo;

        SELECT Mese INTO varMese
        FROM TEMPO
        WHERE IdTempo=:NEW.IdTempo;
```

```
SELECT Anno INTO varAnno
FROM TEMPO
WHERE IdTempo=:NEW.IdTempo;

 -- Verifico se esiste una tupla in 'VM1' associata ai valori di

 SELECT COUNT(*) INTO N
 FROM VM1
 WHERE TipoBiglietto=varTipoBiglietto AND Data=varData AND Mese=varMese AND Anno=varAnno;

 IF(N>0) THEN
    UPDATE VM1
     SET RicaviTot=RicaviTot+:NEW.Ricavo,
     BigliettiTot=BigliettiTot+:NEW.NumBiglietti
     WHERE TipoBiglietto=varTipoBiglietto AND Data=varData AND Mese=varMese AND Anno=varAnno;

 ELSE
     INSERT INTO VM1 (TipoBiglietto, Data, Mese, Anno, RicaviTot, BigliettiTot)
     VALUES(varTipoBiglietto, varData, varMese, varAnno, :NEW.Ricavo, :NEW.NumBiglietti);

 END IF;
 END;
```

4.4)

Operazioni sulla base dati che attivano il trigger:

- INSERT nella TABELLA BIGLIETTI

Infatti il trigger è stato definito unicamente per questo scopo. Qualora si volesse che il trigger venisse attivato anche in ulteriori occasioni sarebbe necessario definire ciascun evento che lo scatena.