

Figure 1. Schema ER di una catena di supermercati (schema operativo)

Esercizio 1

Si consideri lo schema operativo di una catena di supermercati in Figura 1. Il significato delle entità, associazioni e attributi sono chiari.

Si chiede di sviluppare i seguenti punti:

1. Individuare eventuali dipendenze tra i dati che possono essere utilizzate per definire gerarchie nelle dimensioni.
2. Identificare quali sono i possibili fatti che l'azienda può essere interessata ad analizzare.
3. Per ognuno dei fatti identificati al punto 2, stabilire le dimensioni di analisi e le misure che possono essere calcolate.
4. Specificare se le misure individuate sono additive, semi-additive, non additive. Per le misure semi-additive e non additive specificare se è possibile (e come) renderle additive.
5. Specificare le gerarchie che possono essere identificate nelle dimensioni individuate.
6. Disegnare gli schemi concettuali dei fatti.
7. Definire il modello logico dei fatti.

Per uno dei fatti individuati:

8. Creare un oggetto dimension di Oracle riferito ad una delle dimensioni individuate, rappresentando i livelli, le gerarchie e gli attributi ai vari livelli.
9. Identificare un'analisi che è possibile effettuare su tale fatto e scrivere un'interrogazione SQL per realizzarla. Sfruttare dove è il caso gli operatori CUBE e ROLLUP.
10. Creare una vista materializzata in Oracle per l'interrogazione al punto precedente. Giustificare le scelte effettuate.

Soluzione

Punto 1

Prima di individuare i possibili fatti che possono essere analizzati a partire da questo schema, cerco di individuare eventuali dipendenze tra i dati che possono poi essere utilizzate per costruire delle gerarchie nelle dimensioni.

1. Nell'entità **città** esiste una dipendenza funzionale **provincia** → **regione**. In una base di dati operativa la presenza di questa dipendenza funzionale denota uno schema ridondante che può portare ad anomalie.
2. In alcune entità dello schema esistono degli attributi che possono essere utilizzati per effettuare delle aggregazioni dei dati. Questi attributi possono essere rappresentati tramite cammini, dall'attributo a granularità fine a quelli a granularità più grossolana:
 - a. Nell'entità **marca**: nome → azienda (considerando che azienda rappresenta l'azienda che produce una certa marca di prodotti).
 - b. Nell'entità **tempo**: giorno → settimana e giorno → mese → anno; si noti che le granularità temporali settimana e mese non sono tra loro compatibili (esistono settimane che cadono fra due mesi).
 - c. Nell'entità **promozione**: nome → data; dal momento che più promozioni possono iniziare lo stesso giorno.
3. Esistono entità legate da associazioni (1:n) che possono essere utilizzate per effettuare aggregazioni sui dati. Questi legami possono essere rappresentati attraverso cammini dall'entità a granularità più fine (cioè quella con cardinalità 1) a quelle a granularità più grossolana (cioè quella con cardinalità n): prodotto → marca, prodotto → categoria, supermercato → città, fornitore → città

Le precedenti considerazioni possono essere combinate insieme per creare dimensioni e gerarchie all'interno delle dimensioni (una volta individuati i fatti).

Punto 2

Nello schema in Fig. 1 si possono identificare 2 associazioni su cui è ragionevole effettuare analisi:

1. **Vendite**. Tale associazione rappresenta le vendite di un certo prodotto, effettuate in un certo magazzino, in un certo giorno e con una certa promozione.
2. **Acquisti**. Tale associazione rappresenta gli acquisti di prodotti che un certo supermercato ha effettuato in un certo istante di tempo da un certo fornitore.

Se l'azienda è di grandi dimensioni queste associazioni possono contenere molti dati. L'azienda può essere interessata a fare analisi su tali dati.

Le altre associazioni dello schema sono di tipo 1:n e non sembrano particolarmente rilevanti per effettuare analisi.

Punto 3 e 4

Per rispondere a questi quesiti, costruisco la Tabella 1 in cui riporto i fatti, le dimensioni lungo cui è possibile analizzare i fatti e le misure relative. Per ogni misura indico infine se si tratta di misura additiva (A), non additiva (NA), oppure semi-additiva (SA).

Tabella 1. Fatti, dimensioni e misure

FATTI	DIMENSIONI	MISURE	TIPO MISURE
VENDITE	PRODOTTO	QTA' VENDUTA	A
	SUPERMERCATO PROMOZIONE TEMPO	RICAVO DI VENDITA	A
ACQUISTI	PRODOTTO	QTA' ACQUISTATA	A
	SUPERMERCATO FORNITORE TEMPO	% SCONTO	NA

Si noti che la misura **%sconto** non è additiva. Se invece della percentuale si considera l'ammontare dello sconto la misura diventa additiva. Quindi, nel passaggio dalla base di dati operativa, alla base di dati riconciliata (e/o al datawarehouse) riporterò l'ammontare dello sconto, invece della percentuale.

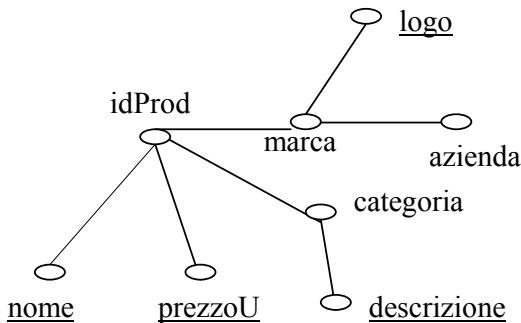
Punto 5

Prendendo in considerazione l'analisi effettuata al punto 1 sulle dipendenze tra attributi, entità e associazioni 1:n, e i fatti individuati al punto 2 su cui è rilevante effettuare un'analisi, disegno per ogni dimensione le possibili gerarchie. Nella costruzione della gerarchia, introduco una chiave artificiale che identifica ogni elemento a granularità minima in quella gerarchia. Tali chiavi artificiali verranno poi utilizzate nella definizione dei fatti nel punto 6. Gli attributi descrittivi di un livello della gerarchia sono stati sottolineati.

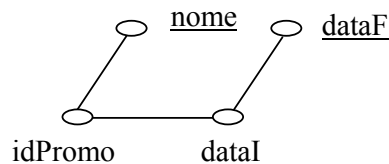
Si noti che la catena di supermercati non è tenuta a mantenere tutti questi livelli di aggregazione e tanto meno è costretta ad avere la granularità più fine uguale a quella della base di dati operativa. Ad esempio, al posto di effettuare analisi per prodotto, l'azienda può effettuare analisi in base alla marca. Quindi nella gerarchia relativa alla dimensione prodotto si può eliminare la radice idProd (e conseguentemente gli attributi che da essa dipendono e la parte di gerarchia relativa a categoria).

In quanto segue, dal momento che il testo non è preciso a riguardo, si considerano tutte le possibili gerarchie che si possono desumere dallo schema della base di dati operativa: sia quelle individuate in una singola entità dello schema che quelle derivate dall'analisi delle associazioni (1:n) che coinvolgono le entità dello schema

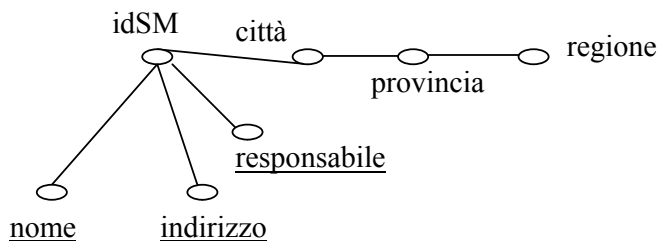
Prodotto



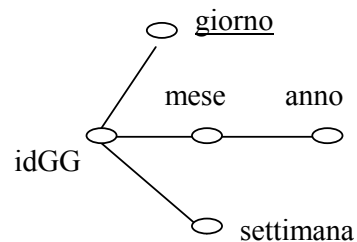
Promozione



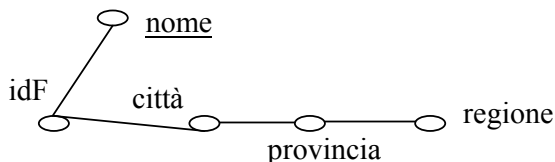
Supermercato



Tempo

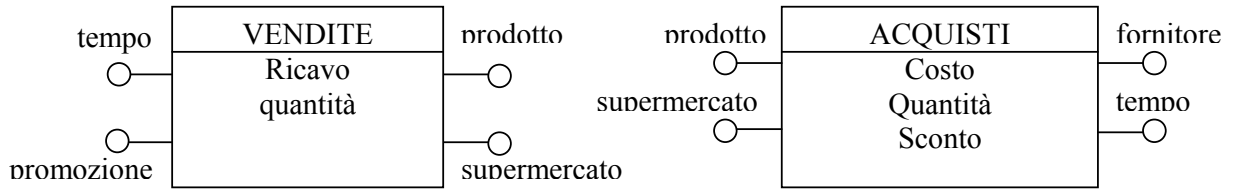


Fornitore



Punto 6

Di seguito presentiamo i fatti con le dimensioni. Le gerarchie delle dimensioni sono state presentate al punto 5



Punto 7

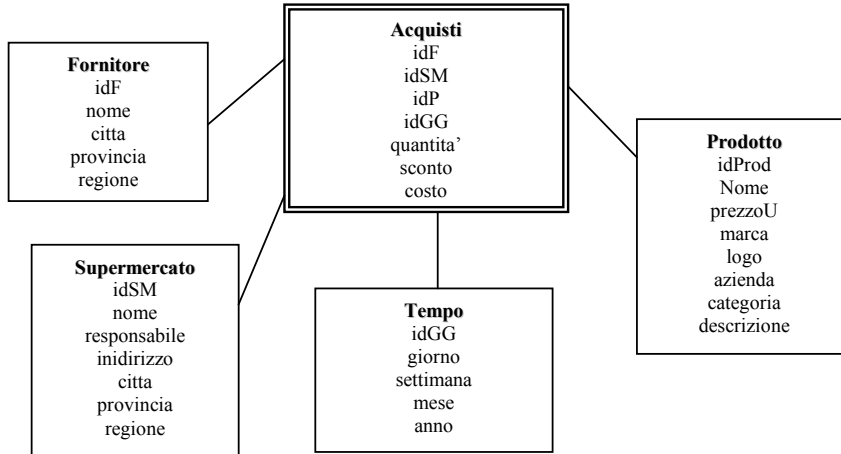


Figura 2. Schema a stella per il fatto acquisti della catena di supermercati

A partire dallo schema concettuale discusso al punto 6 e' possibile individuare due schemi a stella riportati in Figura 2 e 3, rispettivamente.

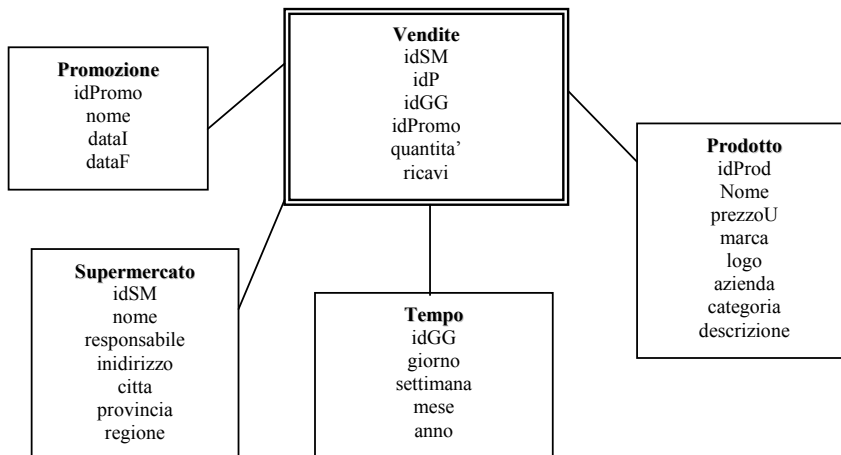


Figura 3. Schema a stella per il fatto vendite della catena di supermercati

Gli schemi a stella in Figura 2 e 3 corrispondono al seguente schema relazionale

Prodotto(idProd, nome, prezzoU, marca, logo, azienda, categoria, descrizione)
 Promozione(idPromo, nome, dataI, dataF)
 Tempo(idGG, giorno, settimana, mese, anno)
 Supermercato(idSM, nome, responsabile, indirizzo, citta, provincia, regione)
 Fornitore(idF, nome, citta, provincia, regione)

} RAPPRESENTAZIONE
DELLE DIMENSIONI

Acquisti(idF, idSM, idP, idGG, quantita, sconto, costo) }
 Vendite(idSM, idP, idGG, idPromo, quantita, ricavo) } FATTI

Il precedente schema non e' normalizzato. Questo non e' un problema per tutti i discorsi fatti a lezione sulla necessita' di effettuare operazioni efficienti e sul tipo di operazioni di modifica che normalmente vengono effettuati sul datawarehouse.

Se si volesse uno schema logico normalizzato allora si dovrebbe realizzare per entrambi i fatti gli schemi a fiocco di neve riportati in Figura 4 e 5.

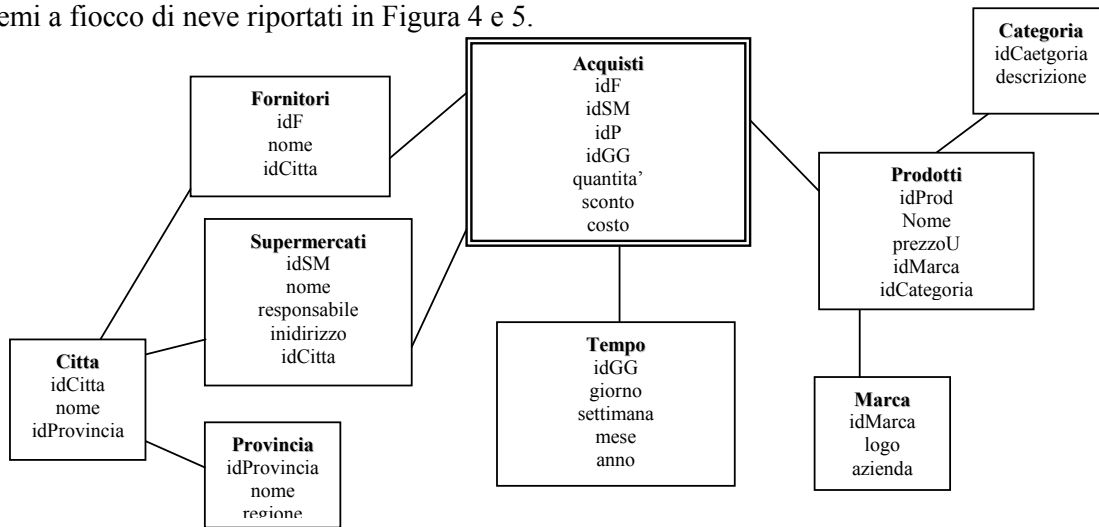


Figura 4 Schema a fiocco di neve per il fatto acquisti della catena di supermercati

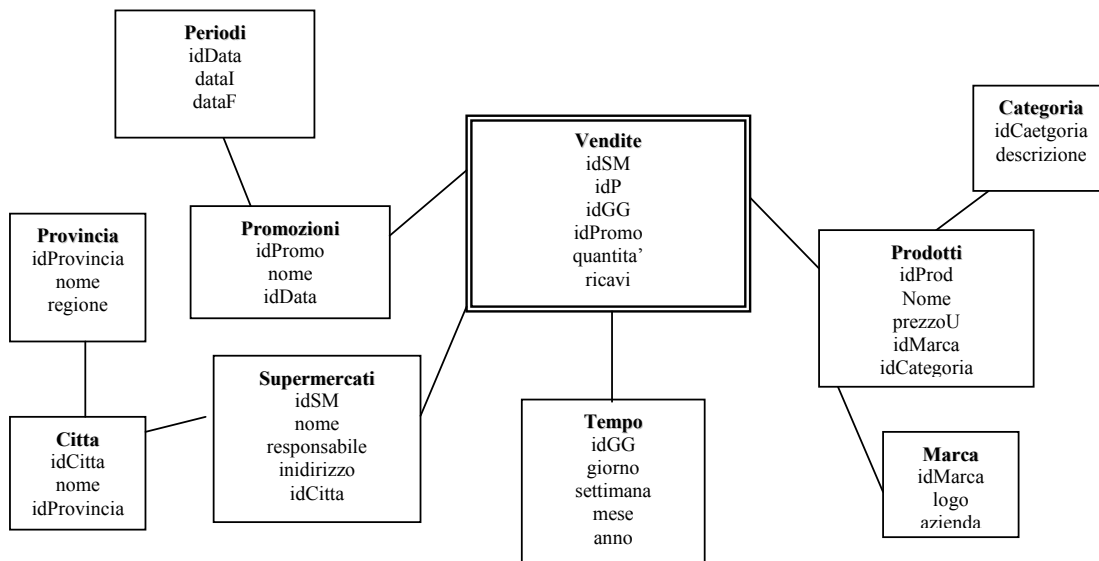
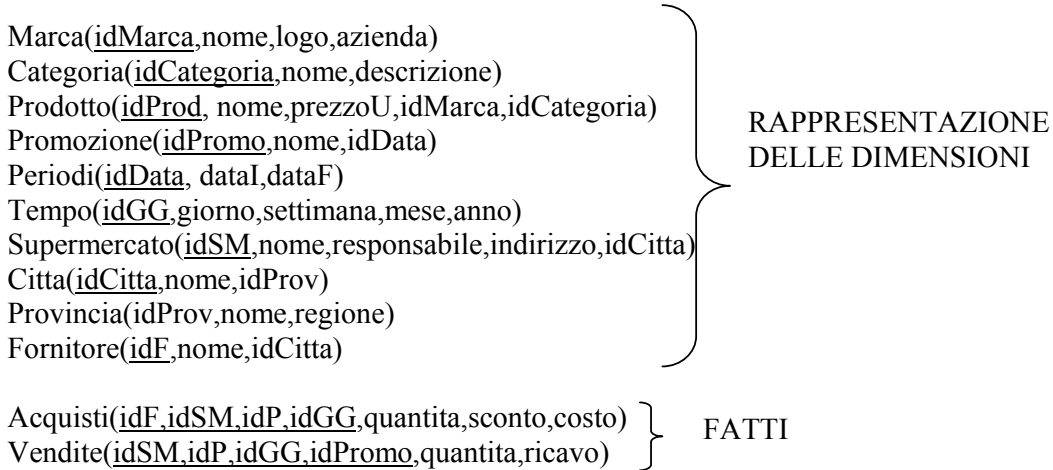


Figura 5. Schema a fiocco di neve per il fatto vendite della catena di supermercati

Gli schemi a fiocco di neve in Figura 4 e 5 corrispondono al seguente schema relazionale



Tale schema e' normalizzato. Si noti però che muoversi lungo le dimensioni per effettuare operazioni di roll-up e drill-down può essere molto costoso (necessità di join).

Si noti infine l'uso di chiavi artificiali sia nel modello a stella che in quello a fiocco di neve. Questa scelta (sempre ritenuta poco appropriata per una base di dati operativa) diventa necessaria nella progettazione di un datawarehouse per rendere piu' efficienti le operazioni di join.

Punto 8

Di seguito vengono presentati gli statement per la creazione di tutte le dimensioni (anche se il testo richiedeva di crearne una sola) qualora venga utilizzato lo schema a stella in Figura 2 e 3.

```

CREATE DIMENSION Prodotto_D
LEVEL prod_1 IS Prodotto.idProd
LEVEL categ_1 IS Prodotto.categoria
LEVEL marca_1 IS Prodotto.marca
LEVEL azienda_1 IS Prodotto.azienda
HIERARCHY categoria_H (
  prod_1 CHILD OF
  categ_1)
HIERARCHY marca_H (
  prod_1 CHILD OF
  marca_1 CHILD OF
  azienda_1)
ATTRIBUTE prod_1 DETERMINES nome
ATTRIBUTE prod_1 DETERMINES prezzoU
ATTRIBUTE categ_1 DETERMINES descrizione
ATTRIBUTE marca_1 DETERMINES logo;

```

```

CREATE DIMENSION Supermercato_D
LEVEL SM_1 IS Supermercato.idSM
LEVEL citta_1 IS Supermercato.citta
LEVEL provincia_1 IS Supermercato.provincia
LEVEL regione_1 IS Supermercato.regione
HIERARCHY supermercato_H (
    SM_1 CHILD OF
    citta_1 CHILD OF
    provincia_1 CHILD OF
    regione_1)
ATTRIBUTE SM_1 DETERMINES Supermercato.nome
ATTRIBUTE SM_1 DETERMINES Supermercato.responsabile
ATTRIBUTE SM_1 DETERMINES Supermercato.indirizzo;

```

```

CREATE DIMENSION Promozione_D
LEVEL promo_1 IS Promozione.idpromo
LEVEL data_1 IS Promozione.dataI
HIERARCHY promozione_H (
    promo_1 CHILD OF
    data_1)
ATTRIBUTE promo_1 DETERMINES Promozione.nome
ATTRIBUTE data_1 DETERMINES Promozione.dataF;

```

```

CREATE DIMENSION Tempo_D
LEVEL giorno_1 IS Tempo.idGG
LEVEL mese_1 IS Tempo.mese
LEVEL anno_1 IS Tempo.anno
LEVEL settimana_1 IS Tempo.settimana
HIERARCHY tempo1_H (
    giorno_1 CHILD OF
    mese_1 CHILD OF
    anno_1)
HIERARCHY tempo2_H (
    giorno_1 CHILD OF
    settimana_1)
ATTRIBUTE giorno_1 DETERMINES Tempo.giorno;

```

```

CREATE DIMENSION Fornitore_D
LEVEL fornitore_1 IS Fornitore.idF
LEVEL citta_1 IS Fornitore.citta
LEVEL provincia_1 IS Fornitore.provincia
LEVEL regione_1 IS Fornitore.regione
HIERARCHY fornitore_H (
    fornitore_1 CHILD OF
    citta_1 CHILD OF
    provincia_1 CHILD OF
    regione_1)
ATTRIBUTE fornitore_1 DETERMINES Fornitore.nome;

```

Di seguito vengono invece presentati gli statement per la creazione degli oggetti dimension di Oracle qualora vengano utilizzati gli schemi a fiocco di neve in Figura 4 e 5. Dal momento che le

dimensioni vengono “spezzettate” in piu’ relazioni occorre specificare come devono essere effettuati i join utilizzando l’operatore JOIN KEY (che non abbiamo visto a lezione). Ogniqualevolta gli attributi che formano i livelli non appartengono alla stessa tabella della base di dati riconciliata e/o del datawarehouse, occorre specificare come deve essere effettuato il join tra le tabelle. Si noti che deve esistere una relazione 1:n tra le tabelle coinvolte.

```
CREATE DIMENSION Prodotto_D
LEVEL prod_1 IS Prodotto.idProd
LEVEL categ_1 IS Categoria.nome
LEVEL marca_1 IS Marca.nome
LEVEL azienda_1 IS Marca.azienda
HIERARCHY categoria_H (
    prod_1 CHILD OF
    categ_1)
HIERARCHY marca_H (
    prod_1 CHILD OF
    marca_1 CHILD OF
    azienda_1)
ATTRIBUTE prod_1 DETERMINES nome
ATTRIBUTE prod_1 DETERMINES prezzoU
ATTRIBUTE categ_1 DETERMINES descrizione
ATTRIBUTE marca_1 DETERMINES logo
JOIN KEY (Prodotto.idCategoria) REFERENCES Categoria
JOIN KEY (Prodotto.idMarca) REFERENCES Marca;
```

```
CREATE DIMENSION Supermercato_D
LEVEL SM_1 IS Supermercato.idSM
LEVEL citta_1 IS Citta.idCitta
LEVEL provincia_1 IS Citta.idProvincia
LEVEL regione_1 IS Provincia.regione
HIERARCHY supermercato_H (
    SM_1 CHILD OF
    citta_1 CHILD OF
    provincia_1 CHILD OF
    regione_1)
ATTRIBUTE SM_1 DETERMINES Supermercato.nome
ATTRIBUTE SM_1 DETERMINES Supermercato.responsabile
ATTRIBUTE SM_1 DETERMINES Supermercato.indirizzo
ATTRIBUTE citta_1 DETERMINES Citta.nome
JOIN KEY (Supermercato.idCitta) REFERENCES Citta
JOIN KEY (Citta.idProvincia) REFERENCES Provincia;
```



```
CREATE DIMENSION Promozione_D
LEVEL promo_1 IS Promozione.idpromo
LEVEL data_1 IS Periodo.dataI
HIERARCHY promozione_H (
    promo_1 CHILD OF
    data_1)
ATTRIBUTE promo_1 DETERMINES Promozione.nome
ATTRIBUTE data_1 DETERMINES Periodo.dataF
JOIN KEY (Promozione.idData) REFERENCES Periodo;
```

```
CREATE DIMENSION Tempo_D
LEVEL giorno_1 IS Tempo.idGG
LEVEL mese_1 IS Tempo.mese
LEVEL anno_1 IS Tempo.anno
LEVEL settimana_1 IS Tempo.settimana
HIERARCHY tempo1_H (
    giorno_1 CHILD OF
    mese_1 CHILD OF
    anno_1)
HIERARCHY tempo2_H (
    giorno_1 CHILD OF
    settimana_1)
ATTRIBUTE giorno_1 DETERMINES Tempo.giorno;
```

```
CREATE DIMENSION Fornitore_D
LEVEL fornitore_1 IS Fornitore.idF
LEVEL citta_1 IS Citta.idCitta
LEVEL provincia_1 IS Provincia.nome
LEVEL regione_1 IS Provincia.regione
HIERARCHY fornitore_H (
    fornitore_1 CHILD OF
    citta_1 CHILD OF
    provincia_1 CHILD OF
    regione_1)
ATTRIBUTE fornitore_1 DETERMINES Fornitore.nome
JOIN KEY (Fornitore.idCitta) REFERENCES Citta
JOIN KEY (Citta.idProvincia) REFERENCES Provincia;
```

Punto 9

Una volta costruito il datawarehouse e popolato con i dati presenti nel database operativo, è possibile fare delle analisi sui dati. Di seguito vengono proposte due tipi di analisi e le corrispondenti interrogazioni SQL.

1. Trovare la media mensile delle vendite per i prodotti di marca Cerruti, venduti in Lombardia nel 2004, per i quali esiste la promozione “saldi fine stagione”.

```
SELECT mese, marca, regione, avg(ricavo)
FROM Vendite, Promozioni, Tempo, Supermercato, Prodotto
WHERE  Vendite.idSM=Supermercato.idSM AND Vendite.idP=Prodotto.idP AND
       Vendite.idPromo = Promozione.idPromo AND Vendite.idGG=Tempo.idGG } condizioni
       AND Supermercato.regione='Lombardia' AND Tempo.anno=2004 AND } di join
       Promozione.nome='saldi fine stagione' AND Prodotto.marca='Cerruti'
GROUP BY CUBE(mese,marca, regione)
```

2. Trovare i costi totali mensili effettivi subiti da supermercati nel triennio 2001-2004 per l'acquisto di prodotti di categoria “vestiario”. Si supponga che si voglia effettuare analisi sia per supermercato che per città, provincia e regione.

```
SELECT mese, idSM, città, provincia, regione, SUM(costo-sconto)
FROM Acquisti, Tempo, Supermercato, Prodotto
WHERE  Acquisti.idSM=Supermercato.idSM AND Acquisti.idP=Prodotto.idP AND
       Acquisti.idGG=Tempo.idGG AND
       Tempo.anno BETWEEN 2001 AND 2004 AND prodotto.categoria='vestiario'
GROUP BY ROLLUP (mese, idSM, città, provincia, regione)
```

Punto 10

Il testo non fornisce indicazioni su come e quando sia meglio creare e aggiornare la vista. Nel definire la vista supponiamo che:

- La vista viene creata la prima volta immediatamente.
- Si vuole utilizzare la vista per migliorare le prestazioni del sistema anche nel caso in cui una interrogazione non sia rivolta esplicitamente sulla vista.
- La vista viene aggiornata in modo incrementale.
- La vista viene aggiornata su richiesta dell'utente.

```
CREATE MATERIALIZED VIEW MediaPerMeseMarcaRegione
BUILD IMMEDIATE
REFRESH FAST ON DEMAND
ENABLE QUERY REWRITE
AS
```

```
SELECT mese, marca, regione, avg(ricavo)
FROM Vendite, Promozioni, Tempo, Supermercato, Prodotto
WHERE  Vendite.idSM=Supermercato.idSM AND Vendite.idP=Prodotto.idP AND
       Vendite.idPromo = Promozione.idPromo AND Vendite.idGG=Tempo.idGG
       AND Supermercato.regione='Lombardia' AND Tempo.anno=2004 AND
       Promozione.nome='saldi fine stagione' AND Prodotto.marca='Cerruti'
GROUP BY CUBE(mese,marca, regione)
```