

VISTE

Una *vista* è una relazione *derivata* a partire dalle relazioni di *base* o di altre viste per mezzo di interrogazioni. Essa consente di mettere a disposizione degli utenti rappresentazioni diverse degli stessi dati. Una vista è

- dotata di uno schema
- utilizzabile nelle interrogazioni alla stregua delle tabelle di base

Due tipologie di viste:

- *virtuali*: prive di estensione (tuple)
- *materializzate*: tabelle derivate il cui contenuto (tuple) viene effettivamente memorizzato nella base di dati (problema: mantenimento dell'allineamento con le tabelle di base)

BD: lo schema logico comprende le seguenti relazioni di base

- **Imp(matr, nome, dip*, stipendio, sup*)**
- **Dip(cod, nome, azienda*, direttore*)**
- **Azienda(cod, nome, città, regione, fatturato)**

A partire dalle suddette relazioni, possiamo definire varie viste, ad esempio, la vista **Info-Dip** che, per ogni dipartimento, riporta codice, nome, stipendio totale, stipendio massimo, numero di impiegati e azienda di appartenenza

```
CREATE VIEW Info-Dip(cod, nome, stip-tot, max-stip, num-imp, azienda) AS
SELECT cod, Dip.nome, sum(stipendio), max(stipendio), count(*), azienda
FROM Imp, Dip
WHERE cod=dip
GROUPBY cod, Dip.nome, azienda
```

Il suddetto costrutto genera una relazione il cui schema è *Info-Dip(cod, nome, stip-tot, max-stip, num-imp, azienda)* e le cui tuple potrebbe, ad esempio, essere le seguenti:

cod	nome	Stip-tot	Max-stip	Num-imp	azienda
d1	ricerca	100.000	20.000	10	A1
d2	vendite	200.000	30.000	15	A1
d3	ricerca	150.000	25.000	12	A2
d4	acquisti	170.000	30.000	20	A2

Lo schema della vista è visibile nello schema logico della BD alla stregua degli schemi delle relazioni di base.

Schema logico della BD esteso con la vista Info-Dip

- **Imp(matr, nome, dip, stipendio, sup*)**
- **Dip(cod, nome, azienda*, direttore*)**
- **Azienda(cod, nome, città regione, fatturato)**
- **Info-Dip(cod, nome, stip-tot, max-stip, num-imp, azienda*)**

Altre viste possono essere definite a partire dalle relazioni (sia di base sia derivate) presenti nel suddetto schema.

Le viste possono essere utilizzate nella formulazione di interrogazioni alla stessa stregua di qualsiasi altra relazione. Di seguito riportiamo alcuni esempi.

- Q1: Stipendio totale e numero di impiegati per tutti i dipartimenti *ricerca* delle varie aziende

```
SELECT azienda, stip-tot, num-imp
FROM Info-Dip
WHERE nome = "ricerca".
```

cod	nome	Stip-tot	Max-stip	Num-imp	azienda
d1	ricerca	100.000	20.000	10	A1
d2	vendite	200.000	30.000	15	A1
d3	ricerca	150.000	25.000	12	A2
d4	acquisti	170.000	30.000	20	A2

Si può notare la semplicità della suddetta query definita sulla vista, soprattutto se paragonata alla complessità della sua formulazione sulle relazioni di base (necessaria in assenza della vista)

```
SELECT azienda, sum(stipendio), count(*)
FROM Imp as I, Dip as D
WHERE D.cod=I.dip and D.nome = "ricerca"
GROUPBY D.cod, azienda
```

- Q2: nome dip col massimo numero di impiegati

```
SELECT nome
FROM Info-Dip
WHERE num-imp = (SELECT Max(num-imp)
                  FROM Info-Dip)
```

NOTA: Q2 si può esprimere sulle tabelle di base usando la clausola Having:

```
SELECT Dip.nome
FROM Imp, Dip
WHERE Imp.dip=Dip.Cod
GROUPBY dip.cod, Dip.nome
HAVING count(*) >= ALL (SELECT count(*)
                        FROM Imp, Dip
                        WHERE Imp.dip=Dip.Cod
                        GROUPBY Dip.cod)
```

POTERE ESPRESSIVO

Ci sono query esprimibili solo grazie alle viste (le sole tabelle di base non sono sufficienti) – quindi le viste incrementano il potere espressivo di SQL. In particolare, si tratta dei seguenti casi

- vi è una applicazione in cascata di funzioni di aggregazione
- query ricorsive

Applicazione di funzioni in cascata. Siccome SQL non supporta l'uso di funzioni che hanno funzioni come argomento (funzioni di funzioni), per poter applicare una funzione sui risultati di un'altra funzione è necessario definire una vista che calcoli i valori della prima funzione.

- Q3: per ogni azienda, numero medio di impiegati nei dipartimenti

Per calcolare il numero medio di impiegati nei dipartimenti di una azienda, bisogna preliminarmente contare il numero di impiegati presenti nei vari dipartimenti. Questa interrogazione ha quindi bisogno della vista *Info-Dip*, in quanto questa fornisce l'attributo *num-imp*

```
SELECT azienda, avg(num-imp)
FROM Info-Dip
GROUPBY azienda
```

cod	nome	Stip-tot	Max-stip	Num-imp	azienda
d1	ricerca	100.000	20.000	10	A1
d2	vendite	200.000	30.000	15	A1
d3	ricerca	150.000	25.000	12	A2
d4	acquisti	170.000	30.000	20	A2

- Q4: per ogni regione, numero medio di imprese per città

Come nel caso precedente, è necessario definire preliminarmente una vista del tipo *Info-Aziende(città, regione, #aziende)*

```
CREATE VIEW Info-Aziende(città, regione, #aziende)
AS SELECT città, regione, count(*)
FROM Azienda
GROUPBY città, regione
```

Su Info-Azienda è quindi possibile formulare la seguente query:

```
SELECT regione, avg(#aziende)
FROM Info-Aziende
GROUPBY regione
```

Query ricorsive- SQL consente di definire viste ricorsive. Ad esempio, possiamo definire una vista sulla relazione

- **Imp(matr, nome, dip*, stipendio, sup*)** – sup è il superiore diretto dell'impiegato con matricola matr

che calcola la gerarchia degli impiegati, attraverso la chiave secondaria sup (diretto superiore). Pertanto, per ogni impiegato, possiamo calcolare il superiore, il superiore del superiore, ecc.

```
CREATE VIEW Superiore(subordinato, superiore) AS
```

```
SELECT matr, sup
FROM Imp
UNION
SELECT X.matr, Y.superiore
FROM Imp as X, Superiore as Y
WHERE X.sup=Y.subordinato
```

La prima sottoquery (quella in alto) calcola il passo base della ricorsione: ogni diretto superiore è un superiore. La relazione Superiore coincide con Imp.

La seconda sottoquery implementa il passo induttivo: se X è una istanza di impiegato (con matricola X.matr e diretto superiore X.sup), e Y è una istanza di Superiore tale che Y.subordinato è uguale a X.sup, allora Y.superiore è un superiore di X.matr. -- cioè, $\langle X.matr, Y.superiore \rangle$ è una istanza di superiore.

ESECUZIONE DI QUERY

- 1) Vista virtuale: trasformazione della query sulla vista in una query equivalente sulle relazioni di base – a tempo di compilazione
- 2) Vista materializzata: quando una query sulla vista viene eseguita per la prima volta, le tuple generate vengono memorizzate come per le relazioni di base. Problema: aggiornamento necessario ogniqualvolta le relazioni di base, in funzione delle quali è definita la vista, vengono aggiornate

VANTAGGI USO VISTE

- Semplicità interrogazioni
- Maggiore potere espressivo - Viste ricorsive e composizione di funzioni
- Protezione dei dati: gruppi di utenti, per motivi di sicurezza/privacy, possono accedere alla BD attraverso opportune viste
- Indipendenze logica: se viene modificato lo schema logico della BD, si può definire un insieme di viste tali da mantenere inalterato lo schema iniziale – per gli utenti non cambia quindi la percezione della BD