SQL – quarta parte

DOCENTE
PROF. ALBERTO BELUSSI

Anno accademico 2018/19

Interrogazione di tipo insiemistico

Consentono di eseguire operazioni insiemistiche sul risultato di due o più interrogazioni SQL espresse come SELECT … FROM … WHERE … [GROUP BY].

Sono disponibili le operazioni di unione, intersezione e differenza con la seguente sintassi:

<selectSQL> { ⟨ **UNION | INTERSECT | EXCEPT** ⟩ **[all]** **<selectSQL>** }

- Se viene precisata la parola chiave **all** allora non vengono eliminati i duplicati
- **INTERSECT** e **EXCEPT** non aggiungono potere espressivo al linguaggio (sono sempre sostituibili con interrogazioni nidificate o join)
- **UNION** invece aggiunge potere espressivo.

Interrogazioni di tipo insiemistico: esempi

CLIENTE(CF, Nome, Cognome, Prof, DataN, Città)

FILIALE(Codice, Nome, Indirizzo, Città)

CONTO(Filiale, Numero, Saldo)

INTESTAZIONE(FilialeCC, NumeroCC, Cliente)

MOVIMENTO(FilialeCC, NumeroCC, Num, Tipo, Data, Imp)

Esempio

Trovare l'elenco complessivo delle città di residenza dei clienti e di ubicazione delle filiali senza duplicati.

```
SELECT Città FROM CLIENTE
UNION
SELECT Città FROM FILIALE
```

Interrogazioni di tipo insiemistico

Vincolo sintattico

Le relazioni prodotte dalle clausole SELECT … FROM … WHERE coinvolte in un operatore insiemistico devono avere lo stesso numero di attributi e attributi di tipo compatibile.

NON è sintatticamente corretta l'interrogazione seguente:

```
SELECT  Cognome FROM CLIENTE
INTERSECT
SELECT Saldo FROM CONTO
```

Interrogazioni di tipo insiemistico: esempi

CLIENTE(CF, Nome, Cognome, Prof, DataN, Città)

FILIALE(Codice, Nome, Indirizzo, Città)

CONTO(Filiale, Numero, Saldo)

INTESTAZIONE(FilialeCC, NumeroCC, Cliente)

MOVIMENTO(FilialeCC, NumeroCC, Num, Tipo, Data, Imp)

Esempio

Trovare il saldo medio dei conti correnti intestati ad avvocati nati dopo il 1/1/1970 e il saldo medio dei conti correnti intestati ad ingegneri nati dopo il 1/1/1980.

Interrogazioni di tipo insiemistico: esempi

CLIENTE(CF, Nome, Cognome, Prof, DataN, Città)

CONTO(Filiale, Numero, Saldo)

INTESTAZIONE(FilialeCC, NumeroCC, Cliente)

```
SELECT 'Avv', AVG(C.saldo) FROM CONTO C, INTESTAZIONE I,
CLIENTE CL
WHERE C.Filiale = I.FilialeCC AND C.Numero=I.NumeroCC
        AND CL.CF = I.Cliente AND CL.Prof = 'Avvocato' AND
        CL.DataN > '1/1/1970'
UNION
SELECT 'Ing', AVG(C.saldo) FROM CONTO C, INTESTAZIONE I,
CLIENTE CL
WHERE C.Filiale = I.FilialeCC AND C.Numero=I.NumeroCC
        AND CL.CF = I.Cliente AND CL.Prof = 'Ingegnere' AND
        CL.DataN > '1/1/1980'
```

Viste in SQL

Consentono di salvare interrogazioni complesse condivise o di definire interfacce esterne verso applicazioni o utenti finali.

Sintassi:

CREATE VIEW <nomeVista> [(<lista attributi>)]

AS <selectSQL>

Si noti che:

- **Le viste aggiungono potere espressivo al linguaggio. Consentono infatti di applicare in cascata due operatori aggregati.**
- **Le viste SQL sono sempre virtuali**

ESEMPIO: CREATE VIEW V1 AS (SELECT DISTINCT Città FROM Filiale)

Viste in SQL

Osservazioni:

- Non è possibile definire viste ricorsive.
- E' possibile definire una vista utilizzando altre viste, ma evitando dipendenze circolari (esempio di dipendenza circolare: V1 definita da V2 definita da V3 definita da V1).
- Le viste non sono in generale aggiornabili (solo in casi particolari i DBMS ammettono l'aggiornamento di viste)

Interrogazioni con viste: esempi

CLIENTE(CF, Nome, Cognome, Prof, DataN, Città)

CONTO(Filiale, Numero, Saldo)

INTESTAZIONE(FilialeCC, NumeroCC, Cliente)

MOVIMENTO(FilialeCC, NumeroCC, Num, Tipo, Data, Imp)

Trovare il nome e il cognome dei clienti con il numero massimo di conti aperti.

```
CREATE VIEW NumConti AS (SELECT Cliente,
    count(*) AS N_Conti FROM INTESTAZIONE I
    GROUP BY Cliente);
SELECT Nome, Cognome FROM CLIENTE
WHERE CF IN (SELECT Cliente FROM NumConti
                    WHERE N_Conti IN (SELECT MAX(N_Conti)
                                      FROM NumConti)
```

Interrogazioni con viste

CLIENTE(CF, Nome, Cognome, Prof, DataN, Città)

FILIALE(Codice, Nome, Indirizzo, Città)

CONTO(Filiale, Numero, Saldo)

INTESTAZIONE(FilialeCC, NumeroCC, Cliente)

Trovare per ogni filiale la città dove risiede il maggior numero di correntisti della filiale

```
CREATE VIEW NumCorr AS (SELECT I.FilialeCC, CL.Città,
    count(DISTINCT I.Cliente) AS NCorr
    FROM INTESTAZIONE I, CLIENTE CL
    WHERE CL.CF = I.Cliente GROUP BY I.FilialeCC, CL.Città);
SELECT FilialeCC, Città FROM NumCorr T
WHERE NCorr IN (SELECT MAX(NCorr)
                FROM NumCorr Tbis
                WHERE Tbis.FilialeCC =T.FilialeCC)
```

Interrogazioni con viste

CLIENTE(CF, Nome, Cognome, Prof, DataN, Città)

FILIALE(Codice, Nome, Indirizzo, Città)

CONTO(Filiale, Numero, Saldo)

INTESTAZIONE(FilialeCC, NumeroCC, Cliente)

Trovare la filiale con il maggior numero di conti aperti, riportando il nome della filiale e il saldo massimo

```
CREATE VIEW NumConti (SELECT Filiale, count(*) as NC
                      FROM CONTO GROUP BY Filiale)
SELECT F.Nome, MAX(Saldo) FROM FILIALE F, CONTO C
WHERE F.Codice=C.Filiale AND
F.Codice IN (SELECT Filiale FROM NumConti
                    WHERE F.NC = (SELECT MAX(NC) FROM NumConti))
GROUP BY F.Codice;
```