

SQL: operazioni sui dati SELECT

Fallucchi Francesca

fallucchi@info.uniroma2.it

<http://art.uniroma2.it/fallucchi/>

a.a. 2010/2011

Select su più tabelle

```
SELECT * FROM FAMIGLIA, MEMBRO
```

- Il risultato è il prodotto cartesiano delle relazioni che partecipano alla select
- Per listare gli attributi è necessario distinguere a quali tabelle appartengono usando la seguente notazione: <tabella>.<colonna>

```
SELECT MEMBRO.Cittadino,  
       FAMIGLIA.CapoFamiglia  
FROM FAMIGLIA, MEMBRO;
```

Join

- Istruzioni SELECT con una sola relazione nella clausola FROM permettono di realizzare: selezioni, proiezioni, ridenominazioni
- Con più relazioni nella FROM si realizzano join (e prodotti cartesiani)
- E' possibile esprimere 'join' mediante opportuni predicati nella clausola WHERE
- L'operatore JOIN può essere usato nella clausola FROM poiché produce relazioni

SQL e algebra relazionale

- $R_1(A_1, A_2) R_2(A_3, A_4)$

```
SELECT DISTINCT R1.A1, R2.A4  
FROM R1, R2  
WHERE R1.A2 = R2.A3
```

- prodotto cartesiano (FROM)
- selezione (WHERE)
- proiezione (SELECT)

SQL e algebra relazionale

- $R_1(A_1, A_2) R_2(A_3, A_4)$

```
SELECT R1.A1, R2.A4  
FROM R1, R2  
WHERE R1.A2 = R2.A3
```

```
PROJA1, A4 (SELA2=A3 (R1 JOIN R2))
```

SQL e algebra relazionale

- possono essere necessarie ridenominazioni nel prodotto cartesiano e nella target list

```
REN B1, B2 A1, A4 (  
  PROJ A1, A4 (SEL A2 = A3 AND A4 = C1(  
    R1 JOIN R2 JOIN REN C1, C2 A1, A2 (R1))))
```

```
SELECT X.A1 AS B1, Y.A4 AS B2  
FROM R1 X, R2 Y, R1 Z  
WHERE X.A2 = Y.A3 AND Y.A4 = Z.A1
```

Maternità	Madre	Figlio	Persone		
			Nome	Età	Reddito
	Luisa	Maria	Andrea	27	21
	Luisa	Luigi	Aldo	25	15
	Anna	Olga	Maria	55	42
	Anna	Filippo	Anna	50	35
	Maria	Andrea	Filippo	26	30
	Maria	Aldo	Luigi	50	40
Paternità	Padre	Figlio	Franco	60	20
	Sergio	Franco	Olga	30	41
	Luigi	Olga	Sergio	85	35
	Luigi	Filippo	Luisa	75	87
	Franco	Andrea			
	Franco	Aldo			

Esempio interrogazione

- I padri di persone che guadagnano più di 20

```
PROJPadre(paternita JOINFiglio =Nome
SELReddito>20(persone))
```

```
SELECT DISTINCT padre
FROM persone, paternita
WHERE figlio = nome and reddito > 20
```

Esempio interrogazione

- Le persone che guadagnano più dei rispettivi padri; mostrare nome, reddito e reddito del padre

```
PROJNome, Reddito, RP(SELReddito>RP
(RENNP,EP,RP ← Nome, Eta, Reddito(persone)
JOINNP=Padre
(paternita JOINFiglio =Nome persone)))
```

```
SELECT f.nome, f.reddito, p.reddito
FROM persone p, paternita, persone f
WHERE p.nome = padre and figlio =
f.nome and f.reddito > p.reddito
```

Ridenominazione del risultato

```
SELECT figlio, f.reddito as reddito,
p.reddito as redditoPadre
FROM persone p, paternita, persone f
WHERE p.nome = padre and
figlio = f.nome and f.reddito > p.reddito
```

Join

- Padre e madre di ogni persona

```
SELECT paternita.figlio, padre, madre
FROM maternita, paternita
WHERE paternita.figlio = maternita.figlio
```

SELECT con join esplicito

- Sintassi

```
SELECT ...
FROM Tabella
{ ... JOIN Tabella ON CondDiJoin }
,...
[ WHERE AltraCondizione ]
```

Join esplicito

- Padre e madre di ogni persona

```
SELECT paternita.figlio, padre, madre
FROM maternita, paternita
WHERE paternita.figlio = maternita.figlio
```

```
SELECT madre, paternita.figlio, padre
FROM maternita JOIN paternita
ON paternita.figlio = maternita.figlio
```

SELECT con join esplicito

- Le persone che guadagnano più dei rispettivi padri; mostrare nome, reddito e reddito del padre

```
SELECT f.nome, f.reddito, p.reddito
FROM persone p, paternita, persone f
WHERE p.nome = padre and figlio = f.nome and
      f.reddito > p.reddito

SELECT f.nome, f.reddito, p.reddito
FROM (persone p join paternita on p.nome = padre)
     join persone f on figlio = f.nome
WHERE f.reddito > p.reddito
```

Join interni

- prodotto cartesiano tra le due tabelle
`<tabella1> CROSS JOIN <tabella2>`
- il join naturale tra le due tabelle che DEVONO avere una o più colonne con lo stesso nome
`<tabella1> NATURAL JOIN <tabella2>`

Join interni

- il join naturale tra le due tabelle in base alle colonne specificate
`<tabella1>[INNER] JOIN <tabella2> USING <lista-colonne>`
- genera il theta-join tra le due tabelle in base alla condizione specificata
`<tabella1>[INNER] JOIN <tabella2> ON <condizione>`

Join interni

- Nome e Cognome dei cittadini capo famiglia
`SELECT CITTADINO.Nome, CITTADINO.Cognome FROM CITTADINO, FAMIGLIA WHERE CITTADINO.CodiceFiscale=Famiglia.CapoFamiglia;`
oppure
`SELECT CITTADINO.Nome, CITTADINO.Cognome FROM CITTADINO INNER JOIN FAMIGLIA ON CITTADINO.CodiceFiscale=Famiglia.CapoFamiglia;`

Join naturale

```
PROJFiglio,Padre,Madre
(paternita JOINFiglio = Nome RENNome=Figlio(maternita))
paternita JOIN maternita
```

```
SELECT madre, paternita.figlio, padre
FROM maternita
JOIN paternita ON paternita.figlio = maternita.figlio
```

```
SELECT madre, figlio, padre
FROM maternita
NATURAL JOIN paternita
```

Join esterni: "outer join"

- Integrano il risultato del JOIN con le tuple escluse completando le colonne mancanti con valori NULL
- Sono definite le seguenti varianti:
completa il risultato del theta-join estendendo le sole righe escluse della tabella a sinistra del join (tabella1)

```
<tabella1> LEFT [OUTER] JOIN  
<tabella2> ON <condizione>
```

Join esterni: "outer join"

completa il risultato del theta-join estendendo le sole righe escluse della tabella a destra del join (tabella2)

```
<tabella1> RIGHT [OUTER] JOIN  
<tabella2> ON <condizione>  
su entrambe le tabelle  
<tabella1> FULL [OUTER] JOIN  
<tabella2> ON <condizione>
```

Join esterno: "outer join"

- Padre e, se nota, madre di ogni persona

```
SELECT paternita.figlio, padre, madre  
FROM paternita
```

```
LEFT OUTER JOIN maternita  
ON paternita.figlio = maternita.figlio
```

L'operatore OUTER è opzionale

```
SELECT paternita.figlio, padre, madre  
FROM paternita
```

```
LEFT JOIN maternita
```

```
ON paternita.figlio = maternita.figlio
```

Outer join: "outer join"

```
SELECT paternita.figlio, padre, madre  
FROM maternita join paternita  
ON maternita.figlio = paternita.figlio
```

```
SELECT paternita.figlio, padre, madre  
FROM maternita left outer join paternita  
ON maternita.figlio = paternita.figlio
```

```
SELECT paternita.figlio, padre, madre  
FROM maternita full outer join paternita  
ON maternita.figlio = paternita.figlio
```