SQL

Laboratorio di Basi di Dati
a.a. 2002/2003

dott.ssa Francesca A. Lisi
lisi@di.uniba.it

Orario di ricevimento: mercoledì ore 10-12

Sommario (III parte)

- Interrogazioni semplici in SQL
  - Proiezioni
  - Selezioni
  - Giunzioni

Riferimenti

- cap. 4, in particolare 4.2.2 di Atzeni et al.
- capp. 3-4 di Pratt

Maternità

Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Paternità

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Persone

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

Interrogazioni

SELECT *ListaAttributi*
FROM *ListaTabelle*
[**WHERE** *Condizione*]

Consente di formulare interrogazioni nelle seguenti tipologie:

- proiezione senza selezione (– clausola `WHERE`)
- selezione (+ clausola `WHERE`)

Selezione con proiezione

Es. Estrarre nome e reddito delle persone con meno di 30 anni

$$PROJ_{Nome,\ Reddito}(SEL_{Eta<30}(Persone))$$

```
SELECT nome, reddito
FROM persone
WHERE eta < 30
```

```
Andrea  | 27
Aldo    | 15
Filippo | 26
```

SELECT, abbreviazioni

```
SELECT nome, reddito
FROM persone
WHERE eta < 30
```

```
SELECT p.nome AS nome,
   p.reddito AS reddito
FROM persone p
WHERE p.eta < 30
```

Selezione senza proiezione

Es. Estrarre nome, età e reddito delle persone con meno di 30 anni

$$SEL_{Eta<30}(Persone)$$

```
SELECT *
FROM persone
WHERE eta < 30
```

```
Andrea  | 27 | 21
Aldo    | 25 | 15
Filippo | 26 | 30
```

Espressioni nella target list

```
SELECT Reddito/12 AS RedditoMensile
FROM Persone
WHERE Nome = 'Luigi’
```

3,33

Condizione complessa

```
SELECT *
FROM persone
WHERE reddito > 25 AND
  (eta < 30 OR eta > 60)
```

```
Filippo | 26 | 30
Sergio  | 85 | 35
Luisa   | 75 | 87
```

Condizione “LIKE”

- Le persone che hanno un nome che inizia per 'A' e ha una 'd' come terza lettera

```
SELECT *
FROM persone
WHERE nome LIKE 'A_d%'
```

```
Andrea  | 27 | 21
```

Impiegati

Matricola	Cognome	Filiale	Età
3787	Rossi	Roma	32
5998	Neri	Milano	45
7309	Bruni	Milano	NULL
9553	Neri	Napoli	50
9570	Rossi	Roma	39

Gestione dei valori nulli

- Gli impiegati la cui età è o potrebbe essere > 40

$$\text{SEL}_{\text{Età} > 40 \text{ OR Età IS NULL}} (\text{Impiegati})$$

```
SELECT *
FROM impiegati
WHERE eta > 40 OR eta IS NULL
```

```
5998 | Neri  | Milano | 45
7309 | Bruni | Milano | null
9553 | Neri  | Napoli | 50
```

Gestione dei duplicati

Es. Estrarre cognome e filiale di tutti gli impiegati

$$PROJ_{Cognome,\ Filiale}(Impiegati)$$

```
SELECT
   cognome, filiale
FROM impiegati
```

```
Rossi | Roma
Neri  | Milano
Bruni | Milano
Neri  | Napoli
Rossi | Roma
```

```
SELECT DISTINCT
   cognome, filiale
FROM impiegati
```

```
Rossi | Roma
Neri  | Milano
Bruni | Milano
Neri  | Napoli
```

Giunzione

- Istruzioni `SELECT` con una sola relazione nella clausola `FROM` permettono di realizzare:
  - selezioni, proiezioni, ridenominazioni
- con più relazioni nella `FROM` si realizzano *giunzioni* (e *prodotti cartesiani*) in maniera:
  - implicita
  - esplicita

Giunzione implicita

- R1(A1,A2) R2(A3,A4)
- **PROJ** $_{A1,A4}$ (**SEL**$_{A2=A3}$ (R1 **JOIN** R2))

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

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

Giunzione implicita (II)

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

```
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
```

$$\text{REN}_{B1,B2 \leftarrow A1,A4} (\text{PROJ}_{A1,A4} (\text{SEL}_{A2 = A3 \text{ AND } A4 = C1} (\text{R1 JOIN R2 JOIN REN}_{C1,C2 \leftarrow A1,A2} (\text{R1}))))$$

Giunzione implicita (III)

Es. Estrarre le persone con i genitori loro associati

$$\text{PROJ}_{\text{Figlio,Padre,Madre}}(\text{paternita JOIN}_{\text{Figlio = Nome}} \text{REN}_{\text{Nome}\leftarrow\text{Figlio}}(\text{maternita}))$$

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

Giunzione implicita (IV)

Es. Estrarre i padri di persone con reddito superiore a venti milioni

$$PROJ_{Padre}(paternita \; JOIN_{Figlio=Nome} \; SEL_{Reddito>20}(persone))$$

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

Giunzione implicita (V)

Es. Estrarre nome e reddito delle persone che guadagnano più dei rispettivi padri, mostrando anche il reddito del padre

$$\text{PROJ}_{\text{Nome, Reddito, RP}}\,(\text{SEL}_{\text{Reddito>RP}}\,(\text{REN}_{\text{NP,EP,RP} \leftarrow \text{Nome,Eta,Reddito}}(\text{persone})\ \text{JOIN}_{\text{NP=Padre}}\ (\text{paternita JOIN}_{\text{Figlio =Nome}}\ \text{persone})))$$

```
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
```

Giunzione esplicita

```
SELECT  …
FROM Tabella1
  {… JOIN Tabella-n ON CondDiJoin }, …
[WHERE AltraCondizione]
```

Es. Estrarre le persone con i genitori loro associati

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

Giunzione esplicita : *join* interno

Es. Estrarre nome e reddito delle persone che guadagnano più dei rispettivi padri, mostrando anche il reddito del padre

```
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
```

Giunzione esplicita: *join* esterno

Es. Estrarre le persone con i genitori loro associati, mantenendo nel risultato anche i casi di madre non nota

```
SELECT p.figlio, p.padre, m.madre
FROM paternita p
  LEFT OUTER JOIN maternita m
  ON p.figlio=m.figlio
```

Giunzione esplicita: *join* esterno (II)

Es. Estrarre le persone con i genitori loro associati, mantenendo nel risultato anche i casi in cui uno dei genitori non è noto

```
SELECT p.figlio, p.padre, m.madre
FROM paternita p
  FULL OUTER JOIN maternita m
  ON p.figlio=m.figlio
```

Ordinamento del risultato

Es. Estrarre nome e reddito delle persone con meno di trenta anni in ordine alfabetico

```
SELECT nome, reddito
FROM persone
WHERE eta < 30
ORDER BY nome
```

```
Aldo    | 15
Andrea  | 27
Filippo | 26
```

SQL: esecuzione delle interrogazioni

- Le espressioni SQL sono dichiarative e noi ne stiamo vedendo la semantica
- In pratica, i DBMS eseguono le operazioni in modo efficiente, ad esempio:
  - eseguono le selezioni al più presto
  - se possibile, eseguono join e non prodotti cartesiani
- La capacità dei DBMS di "ottimizzare" le interrogazioni, rende (di solito) non necessario preoccuparsi dell'efficienza quando si specifica un'interrogazione
- È perciò più importante preoccuparsi della chiarezza (anche perché così è più difficile sbagliare …)

Esercizi

- Esercizi 1-9, 11-13 sul database Prodotti Premiere da cap. 3., pagg. 65-66, di Pratt;
- Esercizi 1-3 sul database Prodotti Premiere da cap. 4., pag. 91 di Pratt;
- Esercizio 4.7 (punti 1-8) da Atzeni et al., pag. 160

IMPORTANTE

Chi è ammesso alle esercitazioni guidate in laboratorio DEVE cimentarsi già a casa nello svolgimento degli esercizi assegnati sul database Prodotti Premiere