

Interrogare una base di dati: algebra relazionale e SQL

Elenco degli argomenti (I)

- Richiamo del concetto di relazione
- Introduzione all'algebra relazionale
- Elenco degli operatori relazionali
- L'operatore di proiezione
- La proiezione senza duplicati
- L'operatore di selezione
- La composizione di operatori
- Gli operatori di intersezione, unione e differenza

Elenco degli argomenti (2)

- Il prodotto cartesiano
- L'operatore di giunzione (join)
- Cenni sull'outer join
- Introduzione al linguaggio SQL
- I sottolinguaggi di SQL
- Il comando SELECT
- Analogie con gli operatori σ e π
- Analogie con il prodotto cartesiano e join
- Il raggruppamento (GROUP BY)
- La clausola HAVING

Richiamo del concetto di Relazione

- Un'istanza di relazione è un insieme di elementi omogenei complessi (tuple)
- Esempio:

R

•(Rossi, 71523, PI, 1980)

•(Bianchi, 67459, LU, 1981)

•(Bianchi, 79856, LI, 1980)

•(Verdi, 75649, PI, 1981)

•(Carli, 66649, PI, 1981)

Algebra Relazionale

- Un insieme di operatori che
 - sono definiti sulle relazioni
 - producono come risultato una relazione
- Gli operatori possono essere combinati per formare espressioni complesse
- La loro esecuzione è di tipo procedurale

Operatori dell'Algebra Relazionale

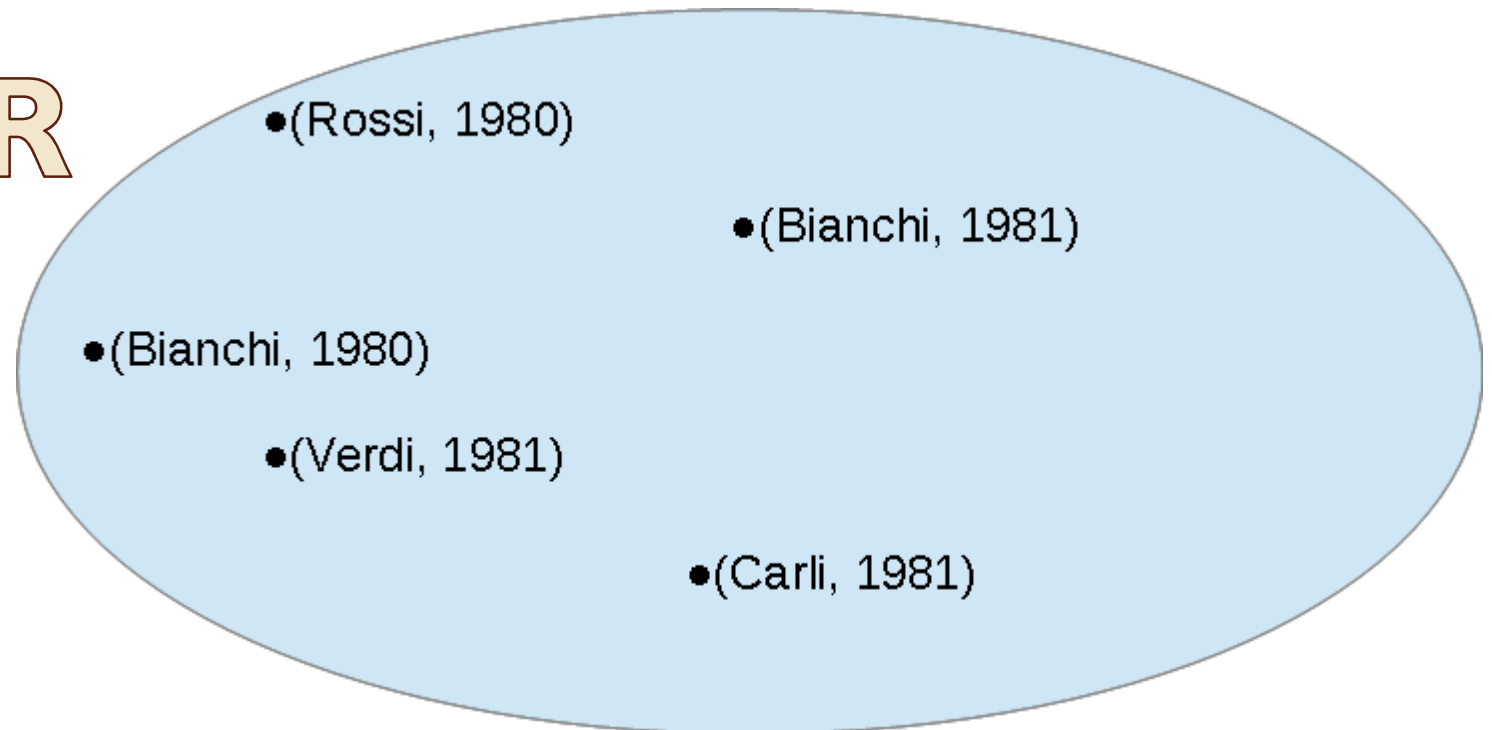
I principali **operatori primitivi** sono:

- Unione;
- Intersezione;
- Differenza;
- Proiezione;
- Selezione;
- Prodotto cartesiano.

Proiezione: l'operatore π

- L'operatore di proiezione applicato ad una relazione R mi restituisce tutti gli elementi di R , considerandone però solo gli attributi specificati da π

$\pi_{\text{nome,nascita}}$ R



Una visione tabellare

R

Nome	Matricola	Provincia	Nascita
Rossi	71523	PI	1980
Bianchi	67459	LU	1981
Bianchi	79856	LI	1980
Verdi	75649	PI	1981
Carli	66649	PI	1981

$\Pi_{\text{nome,nascita}}$ R

Nome	Nascita
Rossi	1980
Bianchi	1981
Bianchi	1980
Verdi	1981
Carli	1981

Proiezione: gestione duplicati

R

Nome	Matricola	Provincia	Nascita
Rossi	71523	PI	1980
Bianchi	67459	LU	1981
Bianchi	79856	LI	1980
Verdi	75649	PI	1981
Carli	66649	PI	1981

Π_{nome} R

Nome
Rossi
Bianchi
Verdi
Carli

Selezione: l'operatore σ

- L'operatore di selezione applicato ad una relazione R mi restituisce solo gli elementi di R che soddisfano la condizione specificata da σ

$\sigma_{\text{nascita}=1981}$ R

•(Verdi, 75649, PI, 1981)

•(Bianchi, 67459, LU, 1981)

•(Carli, 66649, PI, 1981)

Una visione tabellare

R

Nome	Matricola	Provincia	Nascita
Rossi	71523	PI	1980
Bianchi	67459	LU	1981
Bianchi	79856	LI	1980
Verdi	75649	PI	1981
Carli	66649	PI	1981

$\sigma_{\text{nascita}=1981}$ R

Nome	Matricola	Provincia	Nascita
Bianchi	67459	LU	1981
Verdi	75649	PI	1981
Carli	66649	PI	1981

Composizione di operatori

Gli operatori possono essere composti

$\pi_{\text{nome,nascita}} \sigma_{\text{nascita=1981}}$ **R**

•(Verdi, 1981)

•(Bianchi, 1981)

•(Carli, 1981)

Una visione tabellare

R

Nome	Matricola	Provincia	Nascita
Rossi	71523	PI	1980
Bianchi	67459	LU	1981
Bianchi	79856	LI	1980
Verdi	75649	PI	1981
Carli	66649	PI	1981

$\pi_{\text{nome,nascita}} \sigma_{\text{nascita=1981}}$ R

Nome	Nascita
Bianchi	1981
Verdi	1981
Carli	1981

Il linguaggio SQL

- SQL è un linguaggio di interrogazione per database progettato per
 - leggere,
 - modificare
 - gestire **dati** memorizzati in un sistema basato sul **modello relazionale**
 - creare e modificare **schemi** di database
 - creare e gestire strumenti di **controllo** ed **accesso** ai dati.

I linguaggi “dentro” SQL

- **DDL**

- (Data Definition Language, linguaggio di definizione dei dati).
- Consente di descrivere la struttura delle tabelle e di tutti gli elementi di supporto (come indici, vincoli, trigger, viste ecc.).
- Viene utilizzato per realizzare lo schema logico del database.

- **DML**

- (Data Manipulation Language, linguaggio per la manipolazione dei dati).
- Operazioni di inserimento, modifica e cancellazione dei dati.

- **DCL**

- (Data Control Language, linguaggio di controllo dei dati).
- Limiti sui dati (permessi di accesso, vincoli di integrità).

- **QL**

- (Query Language, linguaggio di interrogazione)
- Interrogare il database al fine di individuare i dati che corrispondono ai parametri di ricerca dell'utente.

SELECT

- Per estrarre informazioni dalla base di dati si utilizza l'istruzione SELECT.
- La sintassi completa dell'istruzione SELECT è complessa perché l'istruzione implementa varie funzionalità.

SELECT (proiezione)

SELECT [DISTINCT]

<Campo1> [AS "Alias1"],

<Campo2> [AS "Alias2"],

...

<CampoN> [AS "AliasN"]

FROM <Tabella1>, <Tabella2>, ... <TabellaN>

- DISTINCT - Questa opzione permette di ottenere solo tuple differenti tra loro.
- <Campo> - Elenco dei campi da estrarre.
- <Tabella> - Tabella in cui sono contenuti i campi da estrarre.
- "Alias" - Etichetta da assegnare al campo nella selezione (facoltativa).
- * Sostituendolo ai nomi dei campi implica la selezione di tutti i campi della tabella specificata.

SELECT (restrizione)

- Per estrarre informazioni dal DB, limitate da una condizione:

```
SELECT [DISTINCT]
```

```
<Campo1>, <Campo2>, ... <CampoN>
```

```
FROM <Tabella>
```

```
[WHERE <Condizione>]
```

- <Condizione> - Indica la condizione che devono soddisfare le tuple estratte. All'interno di questa espressione è possibile specificare:
 - nomi dei campi della tabella;
 - operatori di confronto, come =, <>, >, >=, <=, <;
 - operatori logici come NOT, AND, OR;
 - l'operatore LIKE;
 - la parola chiave IS NULL o IS NOT NULL.

Funzioni di aggregazione sui dati

- **COUNT([DISTINCT] <Campo>)**
 - Conta il numero di elementi del campo indicato.
- **MIN(<Campo>)**
 - Restituisce il valore minimo del campo indicato.
- **MAX(<Campo>)**
 - Restituisce il valore massimo del campo indicato.
- **SUM([DISTINCT] <Campo>)**
 - Calcola e restituisce la somma dei valori presenti nel campo indicato.
- **AVG([DISTINCT] <Campo>)**
 - Calcola e restituisce la media aritmetica dei valori presenti nel campo indicato.

Raggruppamento

- Spesso si ha necessità in una query di avere contemporaneamente dati aggregati e dati non aggregati
- **GROUP BY** raggruppa le righe sulla base del valore di uno o più attributi, in genere per effettuare calcoli aggregati su dati omogenei.

Raggruppamento (esempio)

Nome	Matricola	Provincia	Nascita	Materia	Voto
Bianchi	67459	LU	1981	FRA	68
Carli	66649	PI	1981	FRA	94
Verdi	75649	PI	1981	FRA	76
Bianchi	67459	LU	1981	INF	98
Carli	66649	PI	1981	INF	98
Verdi	75649	PI	1981	INF	65
Bianchi	67459	LU	1981	ING	76
Bianchi	67459	LU	1981	ITA	78
Carli	66649	PI	1981	ITA	76
Bianchi	67459	LU	1981	MAT	44
Carli	66649	PI	1981	MAT	56
Verdi	75649	PI	1981	MAT	87

HAVING

- Se ho bisogno di specificare delle condizioni sul risultato delle aggregazioni non posso usare condizioni nel WHERE
- Per restringere il risultato devo specificare una condizione che può considerare sia i campi sia il valore di funzioni di aggregazione: clausola **HAVING**.