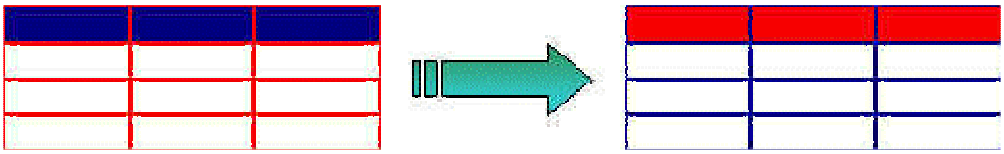
Gli operatori relazionali

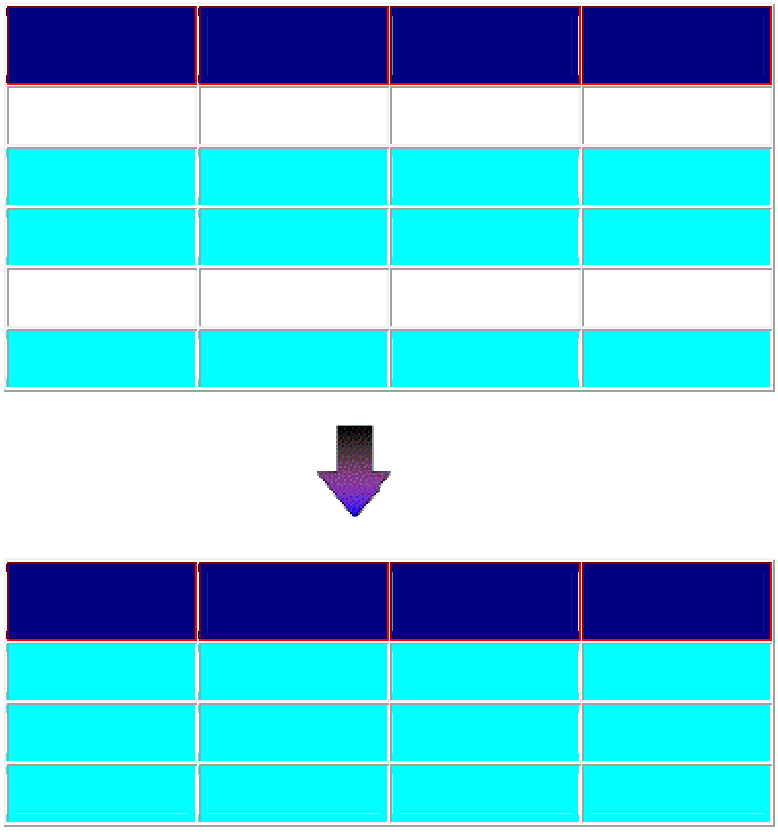
Agiscono su una o più relazioni per ottenere una nuova relazione (servono a realizzare le ***interrogazioni*** sul database)



Ci sono tre operazioni fondamentali per i database relazionali:

- Selezione
- Proiezione
- Congiunzione

L'algebra relazionale: l'operazione di SELEZIONE



La **selezione** (*select*) genera una nuova relazione costituita solo dalle n-tuple della relazione di partenza che soddisfano a una determinata condizione; vengono cioè selezionate le righe con i valori degli attributi corrispondenti alla condizione prefissata.

La relazione ottenuta ha:

- **grado** uguale
- **cardinalità** minore o uguale (di solito minore)

Esempio

Selezione di Studente per Indirizzo = "Torino"

Studente

Matricola	Cognome	Nome	Indirizzo
2345	Rossi	Antonio	Milano
2456	Bianchi	Mario	Torino
5678	Rossi	Luigi	Milano
3341	Verdi	Sara	Torino
2245	Neri	Antonella	Torino
2333	Bruni	Maria	Roma
2222	Viola	Valentina	Perugia

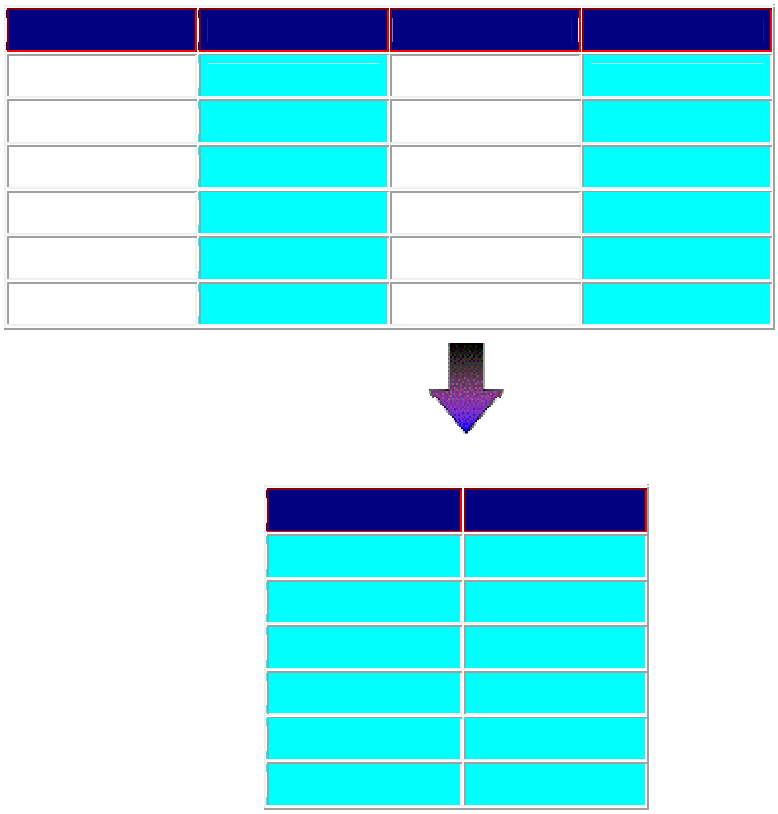
StudenteTorino

Matricola	Cognome	Nome	Indirizzo
2456	Bianchi	Mario	Torino
3341	Verdi	Sara	Torino
2245	Neri	Antonella	Torino

SQL:

```
SELECT * FROM Studente
WHERE Indirizzo=”Torino”;
```

L'algebra relazionale: l'operazione di PROIEZIONE



La **proiezione** (*project*) genera una nuova relazione estraendo dalla tabella iniziale due o più colonne corrispondenti agli attributi prefissati.

La relazione risultante ha:

- **grado** minore o uguale
- **cardinalità** minore o uguale a quella di partenza (perché le righe uguali vengono ridotte a una).

Esempio

Proiezione di Studente su Matricola e Cognome

Studente

Matricola	Cognome	Nome	Indirizzo
2345	Rossi	Antonio	Milano
2456	Bianchi	Mario	Torino
5678	Rossi	Luigi	Milano
3341	Verdi	Sara	Torino
2245	Neri	Antonella	Torino
2333	Bruni	Maria	Roma
2222	Viola	Valentina	Perugia

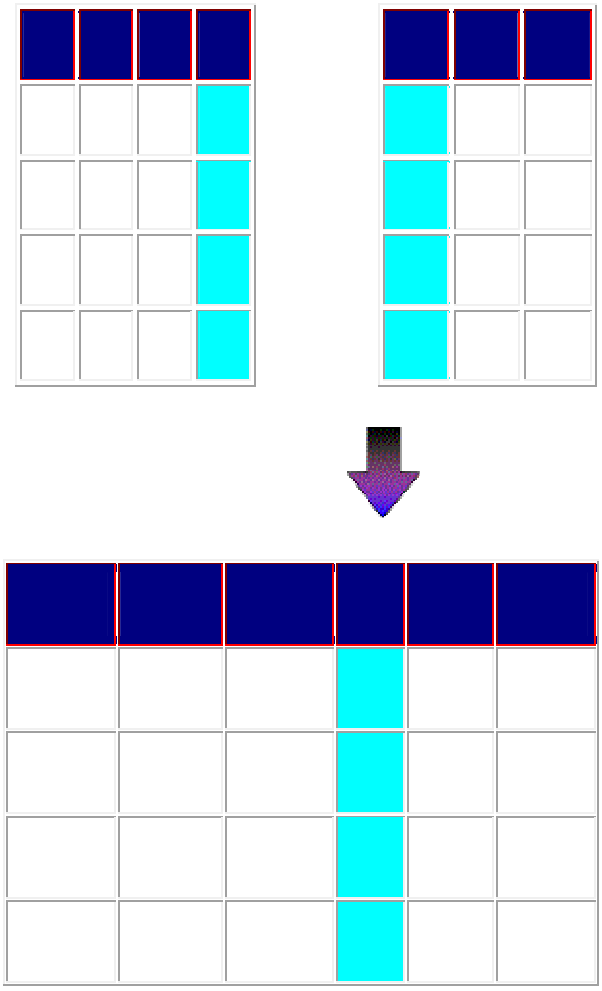
Matr&Cogn

Matricola	Cognome
2345	Rossi
2456	Bianchi
5678	Rossi
3341	Verdi
2245	Neri
2333	Bruni
2222	Viola

SQL:

```
SELECT  Matricola, Cognome
FROM Studente;
```

L’algebra relazionale: l'operazione di CONGIUNZIONE



La **congiunzione** (*join*) serve a combinare due relazioni aventi uno o più attributi in comune, generando una nuova relazione che contiene le righe della prima e della seconda tabella, che possono essere combinate secondo i valori uguali dell'attributo comune.

La relazione risultante ha:

- **grado** uguale alla somma dei gradi delle relazioni di partenza meno uno, perché l'attributo comune compare una sola volta.
- **cardinalità** non prevedibile a priori, in quanto si ottengono solo le righe che possono essere combinate.

Esempio

Congiunzione di Studente su CodFac e di Facoltà su CodFac

Studente

Matricola	Cognome	Nome	Indirizzo	CodFac
2345	Rossi	Antonio	Milano	S23
2456	Bianchi	Mario	Torino	L21
5678	Rossi	Luigi	Milano	T45
3341	Verdi	Sara	Torino	S23
2245	Neri	Antonella	Torino	T45
2333	Bruni	Maria	Roma	L21
2222	Viola	Valentina	Perugia	L21

Facolta

CodFac	Descrizione	Città
S23	Economia	Milano
L21	Lingue	Milano
T45	Ingegneria	Como

Stud&Fac

Matricola	Cognome	Nome	Indirizzo	CodFac	Descrizione	Città
2345	Rossi	Antonio	Milano	S23	Economia	Milano
2456	Bianchi	Mario	Torino	L21	Lingue	Milano
5678	Rossi	Luigi	Milano	T45	Ingegneria	Como
3341	Verdi	Sara	Torino	S23	Economia	Milano
2245	Neri	Antonella	Torino	T45	Ingegneria	Como
2333	Bruni	Maria	Roma	L21	Lingue	Milano
2222	Viola	Valentina	Perugia	L21	Lingue	Milano

SQL:

```
SELECT  *
FROM Studente, Facolta
WHERE  Studente.CodFac = Facolta.CodFac ;
```

Tipi di congiunzione (join)

equi-join : corrispondenza di valori uguali per attributi comuni nelle due tabelle.

join esterno (*outer join*) : restituisce le righe dell'una e dell'altra tabella anche se non sono presenti valori uguali per l'attributo comune; di tutte queste vengono combinate solo le righe per le quali il valore dell'attributo comune nella prima tabella trova un valore uguale nella colonna dell'attributo comune nella seconda tabella.

left join : elenca comunque tutte le righe della prima tabella congiungendo alle righe della seconda solo quelle per le quali si trovano valori corrispondenti per l'attributo comune.

right join : restituisce comunque tutte le righe della seconda tabella e di queste congiunge con le righe della prima tabella solo quelle per le quali si possono trovare valori corrispondenti per l'attributo comune.

self-join : vengono combinate righe di una tabella con le righe della stessa tabella quando sono presenti valori corrispondenti per attributi, cioè due attributi con lo stesso dominio.

Uso di più operatori

Cognome, Nome e Descrizione della Facoltà per gli studenti che risiedono a Milano

1. Selezione di Studente per Indirizzo = "Milano"
2. Congiunzione della tabella ottenuta su CodFac e di Facoltà su CodFac
3. Proiezione della tabella ottenuta su Cognome, Nome, Descrizione

Studente

Matricola	Cognome	Nome	Indirizzo	CodFac
2345	Rossi	Antonio	Milano	S23
2456	Bianchi	Mario	Torino	L21
5678	Rossi	Luigi	Milano	T45
3341	Verdi	Sara	Torino	S23
2245	Neri	Antonella	Torino	T45
2333	Bruni	Maria	Roma	L21
2222	Viola	Valentina	Perugia	L21

StudenteMilano

Matricola	Cognome	Nome	Indirizzo	CodFac
2345	Rossi	Antonio	Milano	S23
5678	Rossi	Luigi	Milano	T45

StudenteMilano

Matricola	Cognome	Nome	Indirizzo	CodFac
2345	Rossi	Antonio	Milano	S23
5678	Rossi	Luigi	Milano	T45

Facoltà

CodFac	Descrizione	Città
S23	Economia	Milano
L21	Lingue	Milano
T45	Ingegneria	Como

Stud&Descr

Cognome	Nome	Descrizione
Rossi	Antonio	Economia
Rossi	Luigi	Ingegneria

SQL:

```
SELECT   Studente.Cognome,  Studente.nome,  Facolta.Descrizione

FROM   Studente, Facolta
WHERE ( (Studente.Indirizzo = “Milano”) AND Studente.CodFac = Facolta.CodFac) ;
```