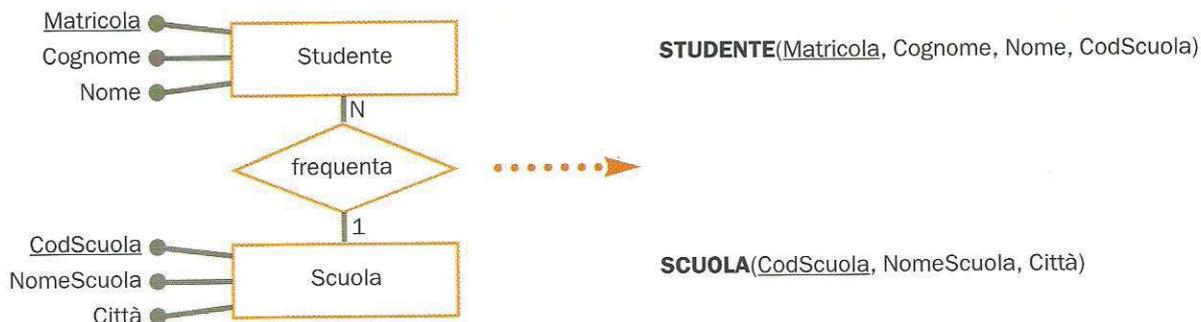




Applichiamo l'algebra relazionale per effettuare semplici interrogazioni sulla nostra base di dati. Vediamo in particolare come si possono interrogare relazioni che provengono dalla traduzione dei diversi tipi di associazione del diagramma ER e come combinare i vari operatori relazionali.

Consideriamo l'esempio dello studente e della scuola:



Per conoscere quali studenti frequentano l'ITIS "Fermi", che ha *CodScuola* pari a "S001", scriveremo:

► $\text{Cognome, Nome}(\sigma_{\text{CodScuola} = \text{"S001"}}(\text{Studente}))$

Si tratta di un'interrogazione composta, formata da due interrogazioni:

► $T = \sigma_{\text{CodScuola} = \text{"S001"}}(\text{Studente})$

Possiamo chiamare questa prima interrogazione **sottointerrogazione**. Essa restituisce come risultato una relazione temporanea (che possiamo chiamare T) sulla quale si esegue un'altra sottointerrogazione:

► $\text{Cognome, Nome}(T)$

Otteniamo così la relazione risultato finale. Il tutto è rappresentato graficamente riassumere nella figura riportata nella pagina successiva.

Supponiamo, ora, di voler conoscere gli studenti che frequentano l'ITIS "Fermi" (con *CodScuola* = S001) e che hanno il cognome Bianchi. L'interrogazione prevede che una condizione composta in effetti da due condizioni. Nell'algebra relazionale avremo:

► $\text{Cognome, Nome}(\sigma_{\text{CodScuola} = \text{"S001"} \text{ AND } \text{Cognome} = \text{"Bianchi"}}(\text{Studente}))$

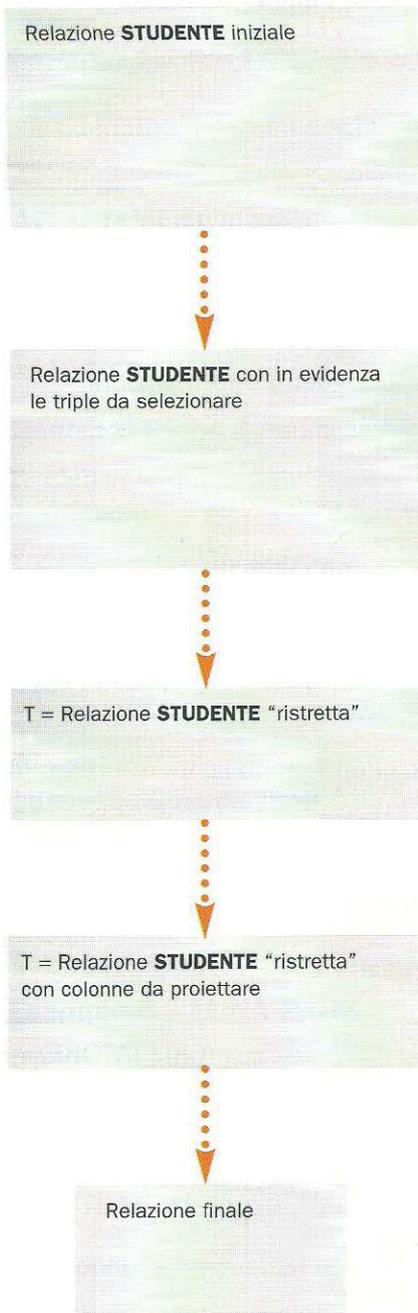
Un utilizzo dell'operazione di join naturale è il seguente:

FILM1(*CodFilm*, Titolo, Lunghezza, NomeStudio)

FILM2(AttorePrincipale, CodiceFilm)

Si deve scrivere un'espressione per rispondere all'interrogazione: "Trovare il protagonista principale dei film che durano almeno 120 minuti".

Occorre mettere in relazione l'attributo *CodFilm* (chiave primaria) di *Film1* con l'attributo *CodiceFilm* (chiave esterna) di *Film2* facendo il join delle due relazioni. Il join naturale accoppia con successo solo quelle t-uple che fanno riferimento allo stesso film. Al join di *Film1* e *Film2* si deve applicare una restrizione che impone la condizione che la lunghezza del film sia almeno di 120 minuti. Successivamente, si effettua la proiezione rispetto all'attributo desiderato, cioè *AttorePrincipale*.



STUDENTE	Matricola	Cognome	Nome	CodScuola
	M001	Rossi	Paolo	S001
	M002	Bianchi	Aldo	S001
	M003	Verdi	Ada	S002
	M004	Neri	Maria	S002

Matricola	Cognome	Nome	CodScuola
M001	Rossi	Paolo	S001
M002	Bianchi	Aldo	S001
M003	Verdi	Ada	S002
M004	Neri	Maria	S002

Matricola	Cognome	Nome	CodScuola
M001	Rossi	Paolo	S001
M002	Bianchi	Aldo	S001

Matricola	Cognome	Nome	CodScuola
M001	Rossi	Paolo	S001
M002	Bianchi	Aldo	S001

Cognome	Nome
Rossi	Paolo
Bianchi	Aldo

La seguente espressione:

$$\pi_{\text{AttorePrincipale}}(\sigma_{\text{Lunghezza} > 120}(\text{Film1} \bowtie_{\text{CodFilm} = \text{CodiceFilm}} \text{Film2}))$$

esprime l'interrogazione desiderata nell'algebra relazionale.

Tutte le volte che eseguiremo interrogazioni utilizzando le espressioni dell'algebra relazionale, faremo sempre ricorso alla composizione di interrogazioni e sottointerrogazioni.



Esempio 1 – PALLONE D'ORO

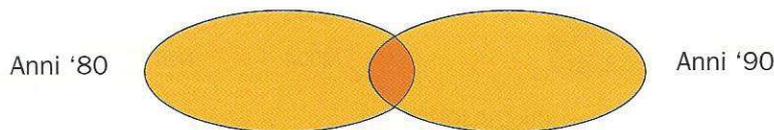
Supponiamo di avere il seguente schema relazionale:

CALCIATORE(CodCalciatore, Nome, Cognome, Anno)

Lo schema è relativo ai calciatori che sono stati premiati con il titolo di "pallone d'oro". L'attributo *Anno* si riferisce, pertanto, all'anno in cui il calciatore è stato nominato "pallone d'oro".

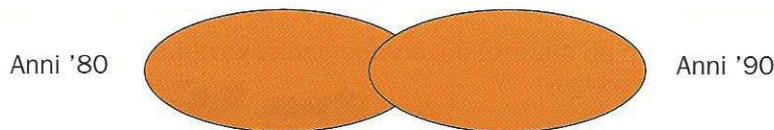
Sfruttando le operazioni relazionali, effettuiamo le seguenti interrogazioni:

1. Visualizza il calciatore che è stato "pallone d'oro" sia negli anni '80 sia negli anni '90.



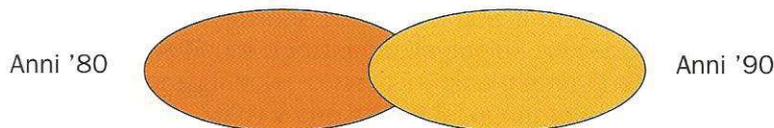
$$\triangleright \delta_{\text{Anno} \geq 1980 \text{ AND } \text{Anno} < 1990}(\text{CALCIATORE}) \cap \delta_{\text{Anno} \geq 1990 \text{ AND } \text{Anno} < 2000}(\text{CALCIATORE})$$

2. Visualizza i calciatori che sono stati "pallone d'oro" negli anni '80 oppure negli anni '90.



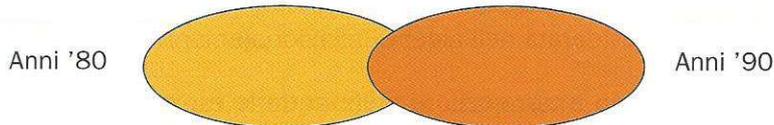
$$\triangleright \delta_{\text{Anno} \geq 1980 \text{ AND } \text{Anno} < 1990}(\text{CALCIATORE}) \cup \delta_{\text{Anno} \geq 1990 \text{ AND } \text{Anno} < 2000}(\text{CALCIATORE})$$

3. Visualizza i calciatori che sono stati "pallone d'oro" negli anni '80 ma non negli anni '90.



$$\triangleright \delta_{\text{Anno} \geq 1980 \text{ AND } \text{Anno} < 1990}(\text{CALCIATORE}) - \delta_{\text{Anno} > 1990 \text{ AND } \text{Anno} < 2000}(\text{CALCIATORE})$$

4. Visualizza i calciatori che sono stati "pallone d'oro" negli anni '90 ma non negli anni '80.



$$\triangleright \delta_{\text{Anno} > 1990 \text{ AND } \text{Anno} < 2000}(\text{CALCIATORE}) - \delta_{\text{Anno} \geq 1980 \text{ AND } \text{Anno} < 1990}(\text{CALCIATORE})$$



Esempio 2 – PALLONE D'ORO

Consideriamo il seguente schema relazionale relativo alla gestione di un condominio:

APPARTAMENTO(CodApp, NumeroVani, Superficie, Indirizzo, Interno, Quota, CodPersona)

PERSONA(CodPersona, Cognome, Nome, Indirizzo, Telefono)

EINQUILINO(CodPersona, CodApp)

VERSA(CodPersona, CodApp, Importo, DataVersamento, Mese)

Sfruttando le operazioni relazionali, effettuiamo le seguenti interrogazioni:

1. Elenca le quote da versare relative agli appartamenti di un determinato proprietario.

▶ $\pi_{\text{CodApp, Indirizzo, Quota}} (\delta_{\text{CodPersona} = \text{"P003"}} (\text{APPARTAMENTO}))$

2. Elenca i proprietari che hanno versato la quota di settembre.

▶ $\pi_{\text{DataVersamento, CodPersona, CodApp}} (\delta_{\text{Mese} = \text{"Settembre"}} (\text{VERSA}))$

In questo modo si ottengono i codici di *Persona* e di *Appartamento*. Se volessimo il nome e cognome relativo ai proprietari scriveremmo:

▶ $\pi_{\text{Cognome, Nome, DataVersamento, CodApp}} (\delta_{\text{Mese} = \text{"Settembre"}} (\text{VERSA} \bowtie \text{PERSONA}))$

Esempio 3 – LUNA PARK

Consideriamo il seguente schema relazionale relativo alla gestione di un luna park:

GESTORE(CodFiscale, Nome, Cognome)

GIOSTRA(CodGiostra, Tipo, Pericolosità, CodFiscale)

Sfruttando le operazioni relazionali, effettuiamo le seguenti interrogazioni:

1. Visualizza tutti i gestori che si chiamano "Paolo Rossi".

▶ $\delta_{\text{Nome} = \text{"Paolo"} \text{ AND } \text{Cognome} = \text{"Rossi"}} (\text{GESTORE})$

2. Visualizza il livello di pericolosità di ogni giostra.

▶ $\pi_{\text{CodGiostra, Pericolosità}} (\text{GIOSTRA})$

3. Visualizza le giostre il cui tipo è "Ruota panoramica".

▶ $\pi_{\text{CodGiostra, Pericolosità}} (\delta_{\text{Tipo} = \text{"Ruota panoramica"}} (\text{GIOSTRA}))$



LEZIONE

26

Il tipo di join finora analizzato è l'**equi-join** (detto anche *join interno*, *inner join* o *join naturale*), in cui la congiunzione viene realizzata solo sulle t-uple di due tabelle distinte che hanno valori uguali per alcuni attributi corrispondenti. Nell'esempio della lezione precedente, la tabella *Cliente* può avere clienti a cui non corrispondono agenti e, viceversa, nella tabella *Agente* possono esserci agenti a cui non corrisponde alcun cliente; in questi casi le t-uple sono ignorate. Cerchiamo di comprendere, ora, in che cosa consiste il **join esterno** (*outer join*). Per poterne esplicitare al meglio le funzionalità e il significato, serviamoci di un esempio basato sulle due relazioni *Persona* e *Automobile*. Supponiamo di avere a disposizione la seguente istanza di *Persona*:

Codice	Nome	Cognome
1	Mario	Rossi
2	Luigi	Bianchi
3	Giuseppe	Neri

Codice è la chiave primaria di questa relazione. La relazione *Automobile* associa una serie di vetture ai corrispondenti proprietari, usando un vincolo di chiave esterna verso la relazione *Persona*. Ad esempio, consideriamo l'istanza visibile a lato:

Targa	Modello	Codice
AASSGG	Tipo1	1
UUJJHH	Tipo2	1
PPLLBB	Tipo3	2
WWYYXX	Tipo4	

Effettuiamo un join naturale sul campo *Codice* comune alle due relazioni. Otteniamo la tabella:

Targa	Modello	Nome	Cognome
AASSGG	Tipo1	Mario	Rossi
UUJJHH	Tipo2	Mario	Rossi
PPLLBB	Tipo3	Luigi	Bianchi

Risulta che *Mario Rossi* possiede un'automobile *Tipo1* targata *AASSGG* e un'automobile *Tipo2* targata *UUJJHH*. *Luigi Bianchi*, invece, è proprietario di un'automobile *Tipo3* targata *PPLLBB*. Le due relazioni, attraverso l'inner join impiegato, sono state fuse in un unico risultato. Questo è stato possibile grazie allo speciale vincolo che unisce il campo *Codice* di *Automobile* al campo *Codice* della relazione *Persona*. Particolare attenzione deve essere rivolta al ruolo che i campi nulli ricoprono negli inner join. L'automobile di targa *WWYYXX*, così come *Giuseppe Neri*, non sono presenti nella tabella risultante dall'inner join. Se avessimo voluto informazioni sui proprietari di tutte le automobili, avremmo cercato il risultato visibile qui a destra.

Targa	Modello	Codice
AASSGG	Tipo1	1
UUJJHH	Tipo2	1
PPLLBB	Tipo3	2
WWYYXX	Tipo4	Null

L'operazione di giunzione naturale esclude automaticamente tutti i record aventi valore nullo lungo la colonna impiegata come elemento di giunzione. Per includere anche tali valori, è stato messo a punto un secondo operatore di giunzione: il join esterno (*outer join*), che esiste in tre varianti:

- **right outer join** (join esterno sinistro);
- **left outer join** (join esterno destro);
- **full outer join** (join esterno completo).

Il right outer join include nel risultato tutti i record della tabella specificata più a destra che l'inner join escluderebbe per mancanza di corrispondenze con la tabella di sinistra, oppure per il riscontro di un valore nullo lungo l'attributo di giunzione.



Utilizzando un *left outer join* fra le tabelle *Automobile* e *Persona* sul comune campo *Codice* si ottiene il risultato visibile a lato:

Targa	Modello	Nome	Cognome
AASSGG	Tipo1	Mario	Rossi
UUJJHH	Tipo2	Mario	Rossi
PPLLBB	Tipo3	Luigi	Bianchi
Null	Null	Giuseppe	Neri

Come è facile osservare, fa parte dell'insieme delle righe restituite anche *Giuseppe Neri*. Il suo record proviene dalla relazione di destra (*right*, appunto). Benché non abbia corrispondenze con la relazione di sinistra, è stato ugualmente incluso. Chiaramente, non è possibile determinare dei valori significativi per i due campi *Targa* e *Modello*, pertanto il DBMS ha completato la riga servendosi di due valori nulli, scelti appositamente per rappresentare l'assenza di informazione.

Il *left outer join*, come è facile intuire, funziona in maniera analoga al *right outer join*, riferendosi però alla tabella di sinistra (*left*) invece che a quella di destra.

Utilizzando un *left outer join* si ottiene il risultato visibile a lato:

Targa	Modello	Nome	Cognome
AASSGG	Tipo1	Mario	Rossi
UUJJHH	Tipo2	Mario	Rossi
PPLLBB	Tipo3	Luigi	Bianchi
WWYYXX	Tipo4	Null	Null

Anche in questo caso, i campi che non hanno corrispondenze sono stati completati con dei valori nulli.

Il *full outer join* applica contemporaneamente sia il *left outer join* sia il *right outer join*.

Utilizzando un *full outer join* si ottiene il risultato visibile a lato:

Targa	Modello	Nome	Cognome
AASSGG	Tipo1	Mario	Rossi
UUJJHH	Tipo2	Mario	Rossi
PPLLBB	Tipo3	Luigi	Bianchi
Null	Null	Giuseppe	Neri
WWYYXX	Tipo4	Null	Null

Un altro tipo di join è il **self join**, una particolare congiunzione nella quale le due tabelle coinvolte nell'operazione coincidono, ossia una tabella è messa in relazione con sé stessa.

Condizione fondamentale affinché il *self join* possa essere eseguito è l'esistenza di una relazione riflessiva, tipicamente di tipo gerarchico. Facciamo un esempio analizzando la seguente relazione:

PERSONA(CodAnagrafico, Cognome, Nome, Sesso, CodConiuge)

di cui una possibile istanza è la seguente:

<u>CodAnagrafico</u>	Cognome	Nome	Sesso	<u>CodConiuge</u>
1	Rossi	Riccardo	M	4
2	Verdi	Anna	F	3
3	Neri	Gianni	M	2
4	Bianchi	Daniela	F	1
5	Gialli	Alessandra	F	

A partire da questa vogliamo conoscere le coppie di persone sposate. L'operazione è realizzabile con una giunzione della relazione con sé stessa. L'operazione è, appunto, il *self join*, il cui risultato è il seguente:

<u>CodAnagrafico</u>	Cognome	Nome	CognomeConiuge	NomeConiuge
1	Rossi	Riccardo	Bianchi	Daniela
2	Verdi	Anna	Neri	Gianni