



INTERROGAZIONI

DML

Linguaggio di Manipolazione dei Dati

30

INTERROGAZIONI – FORMATO BASE

SELECT {C₁, C₂,, C_n | *}

quali attributi si vogliono
tenere (* = tutti)

FROM R₁, R₂,, R_k

clausola FROM
In quali relazioni si cercano i dati

WHERE F;

clausola di qualificazione
Filtro: si prendono solo i dati che soddisfano la
condizione F

SELECT colloc, Video.titolo, anno

FROM Film, Video

WHERE Film.titolo=Video.titolo AND
Film.regista = Video.regista;

CHE COSA RAPPRESENTA?

Video

Colloc	titolo	regista	tipo
1111	Underground	Emir Kusturica	v
1117	Underground	Emir Kusturica	v
1123	Underground	Emir Kusturica	d
1112	Underground	Vincent Sherman	v
1671	Big Fish	Tim Burton	d

Film

titolo	regista	anno	genere	valutaz
Underground	Emir Kusturica	1995	commedia	3,5
Underground	Vincent Sherman	1941	dramma	3,7
Big Fish	Tim Burton	2003	avventura	2,7

```
SELECT colloc, Video.titolo, anno
```

```
FROM Film, Video
```

```
WHERE Film.titolo=Video.titolo AND Film.regista = Video.regista;
```

Colloc	Video.titolo	Video.regista	tipo	Film.titolo	Film.regista	anno	genere	valutaz
1111	Underground	Emir Kusturica	v	Underground	Emir Kusturica	1995	commedia	3,5
1111	Underground	Emir Kusturica	v	Underground	Vincent Sherman	1941	dramma	3,7
1111	Underground	Emir Kusturica	v	Big Fish	Tim Burton	2003	avventura	2,7
1117	Underground	Emir Kusturica	v	Underground	Emir Kusturica	1995	commedia	3,5
1117	Underground	Emir Kusturica	v	Underground	Vincent Sherman	1941	dramma	3,7
1117	Underground	Emir Kusturica	v	Big Fish	Tim Burton	2003	avventura	2,7
1123	Underground	Emir Kusturica	d	Underground	Emir Kusturica	1995	commedia	3,5
1123	Underground	Emir Kusturica	d	Underground	Vincent Sherman	1941	dramma	3,7
1123	Underground	Emir Kusturica	d	Big Fish	Tim Burton	2003	avventura	2,7
1112	Underground	Vincent Sherman	v	Underground	Emir Kusturica	1995	commedia	3,5
1112	Underground	Vincent Sherman	v	Underground	Vincent Sherman	1941	dramma	3,7
1112	Underground	Vincent Sherman	v	Big Fish	Tim Burton	2003	avventura	2,7
1671	Big Fish	Tim Burton	d	Underground	Emir Kusturica	1995	commedia	3,5
1671	Big Fish	Tim Burton	d	Underground	Vincent Sherman	1941	dramma	3,7
1671	Big Fish	Tim Burton	d	Big Fish	Tim Burton	2003	avventura	2,7

RISULTATO FINALE

Video

Colloc	titolo	regista	tipo
1111	Underground	Emir Kusturica	v
1117	Underground	Emir Kusturica	v
1123	Underground	Emir Kusturica	d
1112	Underground	Vincent Sherman	v
1671	Big Fish	Tim Burton	d

Film

titolo	regista	anno	genere	valutaz
Underground	Emir Kusturica	1995	commedia	3,5
Underground	Vincent Sherman	1941	dramma	3,7
Big Fish	Tim Burton	2003	avventura	2,7

SELECT colloc, Video.titolo, anno

FROM Film, Video

WHERE Film.titolo=Video.titolo AND Film.regista = Video.regista;

Colloc	Video.titolo	anno
1111	Underground	1995
1117	Underground	1995
1123	Underground	1995
1112	Underground	1941
1671	Big Fish	2003

ogni C_j è una colonna di R_{ij}

tutte le colonne di tutte le relazioni nella clausola FROM

INTERROGAZIONI – FORMATO BASE

SELECT {[DISTINCT] $R_{i1}.C_1, R_{i2}.C_2, \dots, R_{in}.C_n$ | * }

ogni R_{ij} compare in questa lista

colonne restituite dall'interrogazione

FROM R_1, R_2, \dots, R_k

lista di nomi **distinti** di relazioni oggetto dell'interrogazione

Clausola opzionale

- formula del calcolo dei predicati su AND, OR e NOT
- gli atomi sono confronti fra espressioni che possono usare le colonne delle relazioni

WHERE F;

Se una sola relazione ha una colonna di nome C , si può usare C invece di $R.C$

VALUTAZIONE DELLE INTERROGAZIONI

- In ordine si applicano
 - clausola FROM \Rightarrow costruisco il prodotto cartesiano
 - clausola di qualificazione \Rightarrow tengo solo le tuple che la soddisfano
 - clausola di proiezione \Rightarrow tengo solo le colonne che mi interessano
- l'ordine in cui le colonne appaiono nella relazione risultato
 - coincide con l'ordine in cui le colonne sono elencate nella clausola di proiezione
 - quando si usa *
 - ordine in cui le relazioni compaiono nella clausola FROM
 - per ogni relazione, ordine specificato nella definizione della relazione

CLAUSOLA DI QUALIFICAZIONE

- Può contenere i connettivi booleani AND, OR e NOT con l'usuale precedenza
 - ordini di precedenza diversi possono essere specificati utilizzando le parentesi

- i predicati semplici in SQL hanno la forma

$e \text{ op } e'$

- ***e*** ed ***e'*** sono espressioni che denotano valori
 - semplici: un nome di colonna (il valore denotato è il valore della colonna per la tupla considerata)
 - espressioni che usano funzioni e sotto-interrogazioni (vedi seguito)
- ***op*** è un operatore relazionale di confronto

CLAUSOLA DISTINCT

- La clausola DISTINCT del comando SELECT è opzionale
- se specificata, eventuali tuple duplicate vengono eliminate dal risultato
- se omessa il risultato non è più un insieme
 - **Nota: le tabelle contengono multi-insiemi di tuple, non si ha completa aderenza al modello relazionale**

CLAUSOLA DISTINCT

SELECT **DISTINCT**

genere

FROM Film;

genere

drammatico

fantastico

animazione

fantascienza

horror

commedia

thriller

SELECT

genere

FROM Film;

genere

drammatico

fantastico

animazione

drammatico

fantascienza

horror

fantastico

animazione

fantastico

drammatico

fantascienza

commedia

thriller

thriller

VIDEOTECA ESEMPIO 1

Q1: selezionare i film girati prima del 1999

```
SELECT * FROM Film  
WHERE anno < 1999;
```

Risultato

titolo	regista	anno	genere	valutaz
underground	emir kusturica	1995	drammatico	3.20
edward mani di forbice	tim burton	1990	fantastico	3.60
nightmare before christmas	tim burton	1993	animazione	4.00
ed wood	tim burton	1994	drammatico	4.00
mars attacks	tim burton	1996	fantascienza	3.00
nirvana	gabriele salvatores	1997	fantascienza	3.00
mediterraneo	gabriele salvatores	1991	commedia	3.80
pulp fiction	quentin tarantino	1994	thriller	3.50
le iene	quentin tarantino	1992	thriller	4.00

VIDEOTECA ESEMPIO 2

Serve?
No, perché selezioniamo la chiave della relazione: non ci possono essere duplicazioni di (titolo, regista)

- Q2: selezionare il titolo ed il regista di film di fantascienza girati prima del 1999

```
SELECT DISTINCT titolo, regista
FROM Film
WHERE anno < 1999 AND
       genere = 'fantascienza';
```

Risultato

titolo	regista
mars attacks	tim burton
nirvana	gabriele salvatores

VIDEOTECA ESEMPIO 3

Serve?
No, perché fissato il regista,
non ci possono essere
duplicazioni di titolo
(titolo, regista) chiave di Film

Q3: selezionare il titolo dei film di Tim Burton di
genere horror o fantascienza

```
SELECT DISTINCT titolo
FROM Film
WHERE regista = 'tim burton' AND
       (genere = 'horror' OR genere = 'fantascienza');
```

Risultato

```
titolo
-----
mars attacks
il mistero di sleepy hollow
```

OPERATORI E FUNZIONI

- SQL fornisce numerosi operatori di confronto e funzioni predefinite, da usare in atomi ed espressioni

I più comuni

- operatori di confronto
 - operatori standard matematici $=$, $>$, $<$, $>=$, $<=$, $<>$
 - BETWEEN (due valori min e max) equivale a $_ <= \text{max AND } _ >= \text{min}$
 - IN (insieme $\{v_1, \dots, v_n\}$ equivalente a $_ = v_1 \text{ OR } \dots \text{ OR } _ = v_n$
 - LIKE (pattern) seleziona stringhe che si adeguano al pattern
- funzioni
 - aritmetiche
 - su stringhe
 - su date e tempi

CONDIZIONI SU INTERVALLI DI VALORI: BETWEEN

- Formato: e BETWEEN e_1 AND e_2
 - Equivale a $e \geq e_1$ AND $e \leq e_2$
- Forma negata: e NOT BETWEEN e_1 AND e_2
 - Equivale a $e < e_1$ OR $e > e_2$
- Si può applicare a tutti i tipi per cui si ha un ordinamento totale

- Esempio

```
SELECT * FROM Film
```

```
WHERE anno BETWEEN 1995 AND 2000;
```

Risultato

titolo	regista	anno	genere	valutaz
underground	emir kusturica	1995	drammatico	3.20
mars attacks	tim burton	1996	fantascienza	3.00
il mistero di sleepy hollow	tim burton	1999	horror	3.50
nirvana	gabriele salvatores	1997	fantascienza	3.00

RICERCA DI VALORI IN UN INSIEME: IN

- Formato: $e \text{ IN } (e_1, e_2, \dots, e_n)$
 - Equivale a $e = e_1 \text{ OR } e = e_2 \text{ OR } \dots \text{ OR } e = e_n$
- Forma negata: $e \text{ NOT IN } (e_1, e_2, \dots, e_n)$
 - Equivale a $e \neq e_1 \text{ AND } e \neq e_2 \text{ AND } \dots \text{ AND } e \neq e_n$
- Esempio

```
SELECT * FROM Film
WHERE genere IN ('horror','fantascienza');
```
- Risultato

titolo	regista	anno	genere	valutaz
mars attacks	tim burton	1996	fantascienza	3.00
il mistero di sleepy hollow	tim burton	1999	horror	3.50
nirvana	gabriele salvatores	1997	fantascienza	3.00

CONFRONTO TRA STRINGHE DI CARATTERI: LIKE

- semplici operazioni di *pattern-matching* su valori di tipo stringa
- formato: e [NOT] LIKE *pattern*
- un *pattern* è una stringa di caratteri che può contenere i caratteri speciali % e _
 - il carattere % denota una sequenza di caratteri arbitrari di lunghezza qualsiasi (anche zero)
 - il carattere _ denota esattamente un carattere

CONFRONTO TRA STRINGHE DI CARATTERI: ESEMPIO

- Determinare tutti i film che hanno 'd' come terza lettera del titolo

```
SELECT * FROM Film  
WHERE titolo LIKE '__d%';
```

- Risultato

titolo	regista	anno	genere	valutaz
underground	emir kusturica	1995	drammatico	3.20
mediterraneo	gabriele salvatores	1991	commedia	3.80

ESPRESSIONI E FUNZIONI

- Gli atomi usati nelle interrogazioni sono basati su **espressioni** costruite a partire da
 - nomi di colonna
 - valori costanti
 - funzioni
 - ad esempio aritmetiche, su stringhe, su date e tempi
- Le espressioni possono comparire anche
 - nella clausola di proiezione
 - nelle clausole WHERE
 - nelle espressioni di assegnamento del comando di UPDATE

ESPRESSIONI NELLA CLAUSOLA DI PROIEZIONE

- danno luogo a colonne **virtuali** non presenti nelle relazioni su cui si effettua l'interrogazione
- le colonne virtuali
 - non sono fisicamente memorizzate
 - sono calcolate dinamicamente come risultato dell'esecuzione dell'interrogazione
- è possibile assegnare un nome ad una colonna virtuale con la sintassi

AS <nome colonna>

- la clausola di proiezione è l'ultima ad essere valutata
- ⇒ non è possibile utilizzare il nome della colonna nelle altre clausole
- ⇒ serve solo per lo schema della tabella di output

FUNZIONI ARITMETICHE

Tutti gli operatori comuni sono disponibili

- usuali operatori aritmetici: $+$, $-$, $*$, $/$
- funzione **ABS(n)** che dato un valore numerico n ne calcola il valore assoluto
- funzione **MOD(n,b)** che dati due valori interi n e b calcola il resto della divisione intera di n per b
- funzioni logaritmiche, esponenziali, per il calcolo della radice quadrata, dell'elevamento a potenza, della parte intera superiore ed inferiore...

FUNZIONI PER STRINGHE

- operatore di concatenazione denotato da **||**
- funzioni:
 - **LENGTH(str)** restituisce la lunghezza della stringa str
 - **UPPER (str)** e **LOWER (str)** trasformano la stringa str in caratteri tutti maiuscoli o tutti minuscoli, rispettivamente
 - **SUBSTR (str, m [, n])** estrae dalla stringa str la sottostringa dal carattere di posizione m per una lunghezza n (se n è specificato) oppure fino all'ultimo carattere
 - **TRIM [str₁] FROM str₂** elimina dalla stringa str₂ i caratteri in str₁
 - se str₁ non è specificata elimina gli spazi

FUNZIONI PER DATE E TEMPI

- costanti : **CURRENT_DATE, CURRENT_TIME, CURRENT_TIMESTAMP**
- selettori da valori temporali a interi
 - **EXTRACT (q FROM e)** estrae il campo corrispondente al qualificatore temporale q dall'espressione e
es. **EXTRACT (DAY FROM DATE '08-Ott-1969')**
- funzioni aritmetiche
 - differenza (-) fra valori temporali restituisce un intervallo
 - somma (+) e differenza (-) fra intervalli restituiscono un intervallo
 - somma (+) e differenza (-) fra un valore temporale e un intervallo restituiscono un valore temporale
 - moltiplicazione (*) e divisione (/) di un intervallo per un numero restituiscono un intervallo
 - **ABS** di un intervallo restituisce un intervallo

ESEMPIO DI ESPRESSIONE NELLA CLAUSOLA DI PROIEZIONE

Ritrovare la collocazione del video noleggiato e la durata (in giorni) di ogni noleggio del cliente di codice 6635

```
SELECT colloc, (dataRest - dataNol) DAY  
FROM Noleggio WHERE codCli = 6635;
```

molte implementazioni non lo richiedono

Risultato

colloc	(dataRest - dataNol) DAY
1111	1
1115	1
1117	4
1118	4
1119	2
1120	2
1121	3
1122	3
1113	3
1129	5
1127	?
1125	?

- un valore di tipo intervallo
- bisogna specificarne l'unità di misura per poterlo esprimere come numero

Nome della colonna calcolata attribuito dal sistema usando l'intera espressione

Noleggi ancora aperti hanno dataRest = ?
Il risultato dell'espressione è ?

ESEMPIO DI ESPRESSIONI NELLE CLAUSOLE DI PROIEZIONE E QUALIFICAZIONE

Ritrovare la collocazione del video noleggiato e la durata (in giorni) di ogni noleggio **di durata superiore a due giorni** del cliente di codice 6635

```
SELECT colloc, (dataRest - dataNol) DAY AS durata
FROM Noleggio WHERE codCli = 6635 AND
(dataRest - dataNol) DAY > INTERVAL '2' DAY ;
```

Risultato

colloc	durata
1117	4
1118	4
1121	3
1122	3
1113	3
1129	5

qui non si può usare durata

nel risultato viene associato il nome **durata** alla colonna virtuale

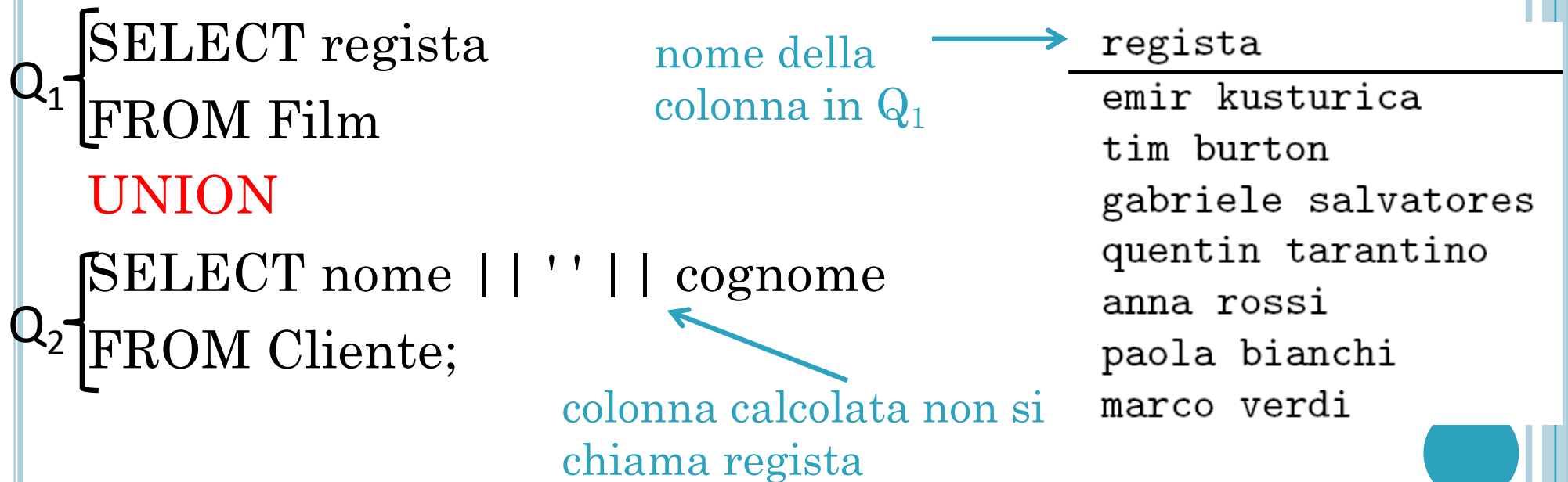
OPERAZIONI INSIEMISTICHE

- In SQL sono disponibili alcune operazioni insiemistiche fra relazioni
 - unione (in sql UNION)
 - differenza (in sql MINUS/EXCEPT sono sinonimi)
 - intersezione (in sql INTERSECT)
- sintassi infissa: $Q_1 \text{ op } Q_2$
 - Q_1 e Q_2 sono tabelle o query
 - op è uno fra UNION , MINUS, EXCEPT, INTERSECT
- Restrizione: Q_1 e Q_2 devono avere lo stesso **numero** di colonne ed i domini corrispondenti devono essere compatibili
 - **non** è richiesto che i nomi delle colonne siano gli stessi
 - in caso siano diversi nel risultato si mantengono i nomi delle colonne nel primo argomento



UNIONE

- $Q_1 \text{ UNION } Q_2$ restituisce tutte le tuple restituite da Q_1 e/o da Q_2 **eliminando eventuali duplicati**
 - variante: UNION ALL non elimina i duplicati
 - da usare ad esempio se si sa priori che non ce ne sono per ottimizzare
- Esempio: determinare i nomi ed i cognomi di registi o di clienti della nostra videoteca



INTERSEZIONE

- $Q_1 \text{ INTERSECT } Q_2$ restituisce tutte le tuple restituite sia da Q_1 che da Q_2 (**senza duplicati**)
- Valgono le stesse condizioni di correttezza statica di union e la stessa convenzione sui nomi delle colonne
- Esempio: determinare gli anni in cui sono usciti sia film di Tim Burton sia film di Quentin Tarantino

Q_1 { SELECT anno FROM Film WHERE regista = 'tim burton'
INTERSECT

Q_2 { SELECT anno FROM Film WHERE regista = 'quentin tarantino';

- il risultato è il solo anno 1994



DIFFERENZA

- $Q_1 \text{ EXCEPT } Q_2$ (sinonimo $Q_1 \text{ MINUS } Q_2$) restituisce tutte le tuple restituite da Q_1 ma non da Q_2 (**senza duplicati**)
- Valgono le stesse condizioni di correttezza statica di union e la stessa convenzione sui nomi delle colonne
- Esempio: determinare gli anni in cui sono usciti film di Tim Burton ma non film di Quentin Tarantino

Q_1 { SELECT anno FROM Film WHERE regista = 'tim burton'
EXCEPT

Q_2 { SELECT anno FROM Film WHERE regista = 'quentin
tarantino';

anno

1990

1993

1996

1999

2003

2005

2005

ORDINAMENTO DEL RISULTATO

- L'ordine delle tuple nel risultato di una interrogazione è determinato dal sistema
 - dipende dalla strategia usata per eseguire l'interrogazione

- Per specificare un particolare ordinamento

ORDER BY <nome colonna [ASC | DESC]>{, <nome colonna [ASC | DESC]>}*

alla fine dell'interrogazione

- Esempio

elencare la collocazione del video, la data di noleggio e la data di restituzione di tutti i noleggi del cliente 6635, **in ordine crescente in base alla data di inizio del noleggio**

```
SELECT colloc, dataNol, dataRest
FROM Noleggio
WHERE codCli = 6635
ORDER BY dataNol;
```

colloc	dataNol	dataRest
1111	01-Mar-2006	02-Mar-2006
1115	01-Mar-2006	02-Mar-2006
1117	02-Mar-2006	06-Mar-2006
1118	02-Mar-2006	06-Mar-2006
1119	08-Mar-2006	10-Mar-2006
1120	08-Mar-2006	10-Mar-2006
1121	15-Mar-2006	18-Mar-2006
1122	15-Mar-2006	18-Mar-2006
1113	15-Mar-2006	18-Mar-2006
1129	15-Mar-2006	20-Mar-2006
1127	22-Mar-2006	?
1125	22-Mar-2006	?

UNIONE – CLAUSOLA DI ORDER BY

- va usata **solo alla fine** dell'interrogazione
- non si possono usare i nomi per specificare le colonne su cui eseguire l'ordinamento
 - potrebbero essere differenti nelle varie relazioni
 - i nomi vengono assegnati come ultimo passo dopo l'ordinamento
- si indicano usandone la **posizione relativa** all'interno della clausola di proiezione

(SELECT regista FROM Film
UNION

SELECT nome || ' ' || cognome FROM Cliente) **ORDER BY 1;**

Senza le tonde

- ha precedenza order by su union
- il parser produce

Q_1 union (Q_2 ORDER BY 1)

⇒ errore perché ORDER BY non è in fondo

OPERAZIONE DI JOIN

- permette di correlare dati memorizzati in relazioni diverse
 - *navigare* le associazioni fra entità
- corrisponde ad un prodotto Cartesiano a cui sono applicati uno o più **predicati di join** per selezionare solo alcune tuple
- vi sono diversi tipi di join utilizzabili nella clausola FROM
 - semplificano la sintassi nei casi più comuni

ESEMPIO DI JOIN

Determinare i titoli dei video noleggiati il 15 Marzo 2006 dal cliente con codice 6635

```
SELECT titolo
FROM Video, Noleggio
WHERE codCli = 6635
      AND dataNol = DATE '15-Mar-2006'
      AND Video.colloc = Noleggio.colloc;
```

Risultato:

<u>titolo</u>
big fish
la sposa cadavere
la fabbrica di cioccolato
le iene

TIPI DI JOIN

<nome relazione> **CROSS JOIN** <nome relazione>

- prodotto Cartesiano puro e semplice
- FROM R CROSS JOIN R' equivale a FROM R, R'

<nome relazione> **JOIN** <nome relazione> **ON** <predicato>

- prodotto Cartesiano filtrato: restano solo le tuple che soddisfano il predicato
- FROM R JOIN R' ON F equivale a FROM R, R' WHERE F

<nome relazione> **NATURAL JOIN** <nome relazione>

- i valori delle colonne con lo stesso nome nelle due relazioni devono essere uguali
- nello schema del risultato le colonne con lo stesso nome nelle due relazioni compaiono una volta sola

<nome relazione> **JOIN** <nome relazione> **USING** (<lista nomi colonne>)

- come il NATURAL JOIN ma solo per le colonne elencate
 - eventuali altre colonne con lo stesso nome nelle due relazioni restano duplicate nel risultato e non soggette a condizioni

VERSIONI EQUIVALENTI DI JOIN

```
SELECT titolo  
FROM Video CROSS JOIN Noleggio  
WHERE codCli = 6635 AND dataNol = DATE '15-Mar-2006'  
      AND Video.colloc = Noleggio.colloc;
```

```
SELECT titolo  
FROM Video JOIN Noleggio ON Video.colloc = Noleggio.colloc  
WHERE codCli = 6635 AND dataNol = DATE '15-Mar-2006';
```

```
SELECT titolo  
FROM Video JOIN Noleggio USING (colloc)  
WHERE codCli = 6635 AND dataNol = DATE '15-Mar-2006';
```

```
SELECT titolo  
FROM Video NATURAL JOIN Noleggio  
WHERE codCli = 6635 AND dataNol = DATE '15-Mar-2006';
```

VERSIONI EQUIVALENTI DI JOIN 2

```
SELECT Video.colloc, titolo  
FROM Video, Noleggio  
WHERE codCli = 6635 AND dataNol = DATE '15-Mar-2006'  
AND Video.colloc = Noleggio.colloc;
```

```
SELECT Video.colloc, titolo FROM Video CROSS JOIN Noleggio  
WHERE codCli = 6635 AND dataNol = DATE '15-Mar-2006'  
AND Video.colloc = Noleggio.colloc;
```

```
SELECT Video.colloc, titolo  
FROM Video JOIN Noleggio ON Video.colloc = Noleggio.colloc  
WHERE codCli = 6635 AND dataNol = DATE '15-Mar-2006';
```

```
SELECT colloc, titolo  
FROM Video JOIN Noleggio USING (colloc)  
WHERE codCli = 6635 AND dataNol = DATE '15-Mar-2006';
```

```
SELECT colloc, titolo  
FROM Video NATURAL JOIN Noleggio  
WHERE codCli = 6635 AND dataNol = DATE '15-Mar-2006';
```

OUTER JOIN

- In $R \text{ JOIN } S$ non si ha traccia delle tuple di R che non corrispondono ad alcuna tupla di S
 - questo non è sempre quello che si desidera
 - esempio per ogni video vogliamo visualizzare la sua collocazione, il titolo del film contenuto ed i codici dei clienti che l'hanno **eventualmente** noleggiato
- l'operatore di OUTER JOIN aggiunge al risultato le tuple di R e S che non hanno partecipato al join, completandole con NULL
- l'operatore di JOIN originario, per contrasto, è anche detto INNER JOIN
- la variante OUTER può essere utilizzata per tutti i tipi di join
 - tranne il cross per cui non ha senso

VARIANTI DI OUTER JOIN

R OUTER JOIN S

- **FULL**: sia le tuple di R che quelle di S che non partecipano al join vengono completate ed inserite nel risultato
- **LEFT**: le tuple di R che non partecipano al join vengono completate ed inserite nel risultato
- **RIGHT**: le tuple di S che non partecipano al join vengono completate ed inserite nel risultato

ESEMPIO DI OUTER JOIN

Per ogni video contenente un film di Tim Burton di genere fantastico vogliamo visualizzare la sua collocazione, il titolo ed i codici dei clienti che l'hanno eventualmente noleggiato

```
SELECT colloc, titolo, codCli
```

```
FROM Film NATURAL JOIN Video
```

```
      NATURAL LEFT OUTER JOIN Noleggio
```

```
WHERE regista = 'tim burton' AND genere = 'fantastico';
```

ESEMPIO DI OUTER JOIN: RISULTATO

colloc	titolo	codCli
1113	big fish	6635
1113	big fish	6642
1114	big fish	6610
1115	edward mani di forbice	6635
1115	edward mani di forbice	6610
1122	la fabbrica di cioccolato	6635
1122	la fabbrica di cioccolato	6642
1123	la fabbrica di cioccolato	?

senza l'utilizzo dell'outer join i video, quali il 1123, che non sono mai stati noleggiati non avrebbero fatto parte del risultato