

SQL: Linguaggio di query

1 Linguaggio set-oriented

SQL è un linguaggio **set-oriented**:

- gli operatori operano su relazioni (che sono insiemi);
- il risultato è sempre una relazione.

2 Formato base di un'interrogazione

Il formato di base di un'interrogazione (query) in SQL è:

```
SELECT [DISTINCT] { $R_{i1}.C_1, R_{i2}.C_2, \dots, R_{in}.C_n$  | *}
FROM  $R_1, R_2, \dots, R_k$ 
[WHERE  $F$ ];
```

- La **clausola di proiezione**, **SELECT**, determina quali colonne devono essere presenti nel risultato:
 - si possono specificare le singole colonne, scrivendo `<nome relazione>.<nome colonna>`, o anche solo `<nome colonna>` se non ci sono ambiguità (cioè se una colonna con tale nome esiste solo in una delle relazioni considerate);
 - si può scrivere semplicemente `*` per includere tutte le colonne di tutte le relazioni considerate.

Si può inoltre specificare la clausola **DISTINCT** per togliere eventuali tuple duplicate dal risultato.

- La clausola **FROM** indica quali sono le relazioni su cui si vuole effettuare l'interrogazione.
- La **clausola di qualificazione**, **WHERE**, può essere opzionalmente usata per specificare un predicato (cioè una condizione) che le tuple cercate devono soddisfare.

2.1 Ordine di valutazione dell'interrogazione

Per valutare un'interrogazione, il DBMS:

1. inizia dalla clausola **FROM**;
2. applica la clausola di qualificazione (se presente) per selezionare le tuple da includere nel risultato;
3. applica la clausola di proiezione per ottenere solo le colonne richieste.

2.2 Ordine dei dati nel risultato

L'ordine delle tuple nella relazione risultato può variare in base a come il DBMS esegue fisicamente la query.

L'ordine delle colonne, invece, è determinato:

- dall'ordine in cui esse sono elencate nella clausola di proiezione;
- quando si usa *****, dall'ordine delle relazioni nella clausola **FROM**, e, per ogni relazione, dall'ordine delle colonne nella definizione di tale relazione (cioè nel comando **CREATE TABLE**).

Per questo, anche se si vogliono ottenere tutte le colonne, può comunque essere utile elencarle esplicitamente (invece di usare *****), per modificare l'ordine in cui appaiono nel risultato.

2.3 Esempi

Selezionare i film girati prima del 1999:

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

Selezionare il titolo e il regista dei film di fantascienza girati prima del 1999:

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

Selezionare i film di Tim Burton di genere horror o fantascienza:

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

Selezionare il titolo di tutti i film della videoteca:

```
SELECT titolo
FROM Film;
```

Selezionare il titolo, il regista, l'anno, il genere, e la valutazione dei film girati dopo il 2000:

```
SELECT titolo, regista, anno, genere, valutaz
FROM Film
WHERE anno > 2000;
-- oppure
SELECT *
FROM Film
WHERE anno > 2000;
```

Trovare tutti i diversi generi dei film offerti dalla videoteca:

```
SELECT DISTINCT genere
FROM Film;
```

3 Operatori

Alcuni operatori che si possono usare nella clausola di qualificazione sono:

- `<nome colonna> BETWEEN <min> AND <max>`: verifica se il valore di una colonna appartiene all'intervallo di valori tra `<min>` e `<max>` (estremi inclusi). Si può applicare solo per i domini sui quali è definito un ordine.
- `<nome colonna> IN (<insieme valori>)`: verifica se il valore di una colonna appartiene all'insieme di possibili valori elencati in `<insieme valori>`. A differenza dell'operatore `BETWEEN`, questo si può applicare a tutti i domini, e l'insieme di valori non deve per forza essere un intervallo.
- `<nome colonna> LIKE <pattern>`: determina se il valore di una colonna, che deve per forza essere una stringa, rispetta un determinato `<pattern>`, definito mediante l'uso di caratteri jolly:
 - `_` rappresenta un singolo carattere qualsiasi (che deve essere obbligatoriamente presente);
 - `%` rappresenta una sequenza di 0 o più caratteri qualsiasi.

3.1 Esempi

Selezionare i film girati tra il 1995 e il 2000:

```
SELECT * FROM Film
WHERE anno BETWEEN 1995 AND 2000;
-- equivale a
SELECT * FROM Film
WHERE anno >= 1995 AND anno <= 2000;
```

Selezionare i film di genere horror o fantascienza:

```
SELECT * FROM Film
WHERE genere IN ('horror', 'fantascienza');
-- equivale a
SELECT * FROM Film
WHERE genere = 'horror' OR genere = 'fantascienza';
```

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

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

Selezionare tutti i film con titoli che finiscono per 'e':

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

Selezionare tutti i film i cui titoli contengono la sequenza di lettere 'is':

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

4 Espressioni e ridenominazione

Oltre a semplici nomi di colonne, nelle clausole di proiezione (SELECT) e di qualificazione (WHERE) sono ammesse anche delle espressioni (es. aritmetiche).

In particolare, le espressioni nella clausola di proiezione “generano” nuove colonne nella relazione risultato. Di default, queste colonne hanno come nomi le espressioni stesse. È però possibile assegnare un nome diverso a una colonna della risultato mediante la parola chiave AS:

```
{<nome colonna> | <espressione>} AS <nuovo nome>
```

4.1 Esempi

Ritrovare il codice di collocazione dei video noleggiati e la *durata* (in giorni) di ogni noleggio del cliente di codice 6635:

```
SELECT colloc, (dataRest - dataNo1) DAY
FROM Noleggio
WHERE codCli = 6635;
```

Calcolare lo stipendio mensile dell'impiegato Bianchi:

```
SELECT Stipendio / 12 AS StipendioMensile
FROM Impiegato
WHERE Cognome = 'Bianchi';
```

Selezionare i codici dei clienti che hanno noleggiato un video più di 5 giorni fa:

```
SELECT codCli
FROM Noleggio
WHERE dataNo1 + 5 < CURRENT_DATE;
```

Nota: La gestione delle date tende a variare tra DBMS diversi.

5 Ordinamento del risultato

Normalmente, l'ordine delle tuple nella relazione risultato di un'interrogazione dipende dalla strategia che il sistema usa per eseguire la query. Se, invece, si vuole ottenere un ordinamento specifico, questo può essere specificato aggiungendo, alla fine dell'interrogazione, la clausola **ORDER BY**, seguita da una lista di nomi di colonne.

Il nome di ogni colonna può opzionalmente essere seguito da:

- ASC per specificare un ordinamento crescente
- DESC per specificare un ordinamento decrescente.

Se non si specifica **ASC** o **DESC**, viene applicato di default l'ordine crescente.

Quando vengono specificate più colonne:

1. le tuple vengono ordinate in base ai valori della prima colonna della lista;
2. se ci sono tuple che hanno valori uguali in tale colonna, queste vengono ordinate in base ai valori nella seconda colonna della lista;
3. così via per tutte le colonne specificate nella clausola **ORDER BY**.

Se ci sono più tuple con valori uguali in tutte le colonne specificate per l'ordinamento, non si ha alcun controllo sul loro ordine, che dipende invece dall'esecuzione della query.

Come caso particolare, i valori NULL vengono sempre considerati più piccoli o più grandi di tutti gli altri, a seconda del DBMS.

5.1 Esecuzione dell'ordinamento

L'ordinamento viene eseguito come ultima fase della query, dopo anche la proiezione. Di conseguenza, gli attributi usati per l'ordinamento devono essere tra quelli specificati nella clausola **SELECT** (esplicitamente, o anche implicitamente mediante *****).

5.2 Esempi

Ottenere tutte le informazioni sui film, ordinati in ordine alfabetico rispetto al titolo:

```
SELECT *  
FROM Film  
ORDER BY titolo;
```

Elencare la collocazione del video, la data del noleggio e la data di restituzione di tutti i noleggi del cliente 6635, ordinando le tuple in base alla data di inizio del noleggio:

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

Elencare la collocazione del video, la data del noleggio e la data di restituzione di tutti i noleggi del cliente 6635, ordinando le tuple in base alla data di inizio del noleggio, in ordine decrescente, e alla collocazione, in ordine crescente.

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