

# **IL LINGUAGGIO SQL**

## Caratteristiche dei linguaggi per Database

I linguaggi per basi di dati relazionali possiedono i comandi per:

- definizione del data base;
- manipolazione dei dati;
- associazione tra tabelle diverse;
- interrogazioni degli utenti.

Ci sono alcune caratteristiche comuni ai diversi linguaggi :

- si basano sulla visione tabellare dei dati;
- non richiedono la specificazione dei percorsi per ritrovare i dati;
- operano su gruppi di righe o sull'intera tabella, anziché su una riga per volta;
- usano interfacce per l'utente (a menu o grafiche).

## Il linguaggio SQL

**SQL (Structured Query Language):**

è di fatto lo standard tra i linguaggi per la gestione di data base relazionali.

- prima versione IBM alla fine degli anni '70 per un prototipo di ricerca (System R)
- negli anni '80 linguaggio per DBMS della IBM (DB2 e SQL/DS)
- standard ANSI (American National Standards Institute) nel 1986
- standard ISO (International Standards Organization) nel 1987
- aggiornamenti degli standard nel 1992 da ANSI (ANSI X3.135) e ISO (ISO 9075).

## Identificatori, dati, costanti e operatori

**Identificatori:** nomi di tabelle e di colonne

Per identificare il nome di una colonna:

NomeTabella.NomeColonna  
(separati dal punto).

*se la parola è unica all'interno del DB, non serve specificare il nome della tabella*

**Tipi standard** per gli attributi:

CHARACTER, DATE, INTEGER, SMALLINT, FLOAT, ecc.

*N.B.: Ci possono essere differenze della dichiarazione dei dati in diverse versioni del linguaggio SQL nei prodotti DBMS (per esempio in Access per Windows).*

Il valore **Null** nelle colonne della tabella indica un valore non disponibile o non definito.

Le **costanti** stringa sono delimitate dai caratteri ' (apice).

*è un semplice voto  
ma due cose diverse*

Si possono usare gli **operatori** NOT, AND e OR nella scrittura delle condizioni.

## La definizione delle tabelle

Il linguaggio SQL possiede i comandi per creare, modificare ed eliminare le tabelle dal database relazionale, cioè le funzioni di linguaggio DDL (*Data Definition Language*):

### CREATE TABLE

seguito dal nome della tabella e dall'elenco degli attributi;

per ogni attributo occorre specificare il nome e il tipo di dato.

### ALTER TABLE

per aggiungere una nuova colonna (**ADD**) a quelle già esistenti;

per togliere una colonna (**DROP**).

### DROP

seguito dal nome della tabella, per eliminare una tabella.

*Nota: nei prodotti DBMS moderni queste operazioni vengono eseguite in modo usando l'interfaccia utente (a menu o grafica).*

## I comandi per la manipolazione dei dati

Il linguaggio SQL possiede i comandi per inserire, modificare ed eliminare le righe di una tabella, cioè le funzioni di linguaggio DML (*Data Manipulation Language*):

### INSERT

inserire nuovi dati nelle righe della tabella

### UPDATE

aggiornare i valori nella tabella

### DELETE

cancellare righe della tabella.

*Nota: nei prodotti DBMS moderni queste operazioni vengono eseguite usando l'interfaccia utente (a menu o grafica).*

## Il comando Select

E' il comando principale di SQL che realizza le **funzioni di linguaggio per le interrogazioni** (Query Language):

- attivare le **interrogazioni sulle relazioni**;
- implementare le **operazioni relazionali** per ottenere nuove tabelle.

Struttura generale del comando *Select*:

**SELECT .....**

**FROM .....**

**WHERE .....**

- dopo *Select*: **nomi delle colonne da elencare** (per indicare tutti gli attributi si scrive l'asterisco \* accanto a *Select*)
- dopo *From*: **il nome o i nomi delle tabelle** *se servono due tabelle, devo specificare la relazione che le lega*
- dopo *Where*: la **condizione da controllare sui valori delle righe** (anche più condizioni combinate con gli operatori AND, OR e NOT).

Con **SELECT DISTINCT...** le **righe duplicate** nella tabella risultante **vengono ridotte a una**.

*Es una persona ha pagato 5 volte una mi intera  
solo sapere chi ha pagato → con select distinct  
la persona in questione viene considerata una volta sola*

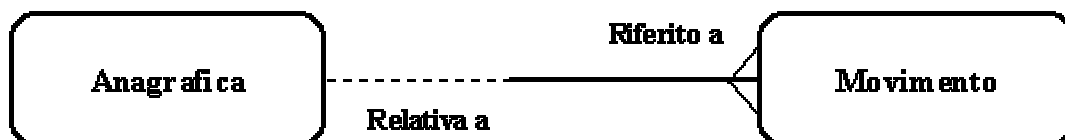
### Modello di database per gli esempi successivi

Viene descritto ora il database che verrà utilizzato per gli esercizi successivi.

E' un database generico con anagrafica e movimenti di uso molto frequente: per esempio, clienti e fatture, studenti ed esami sostenuti, contribuenti e versamenti di imposta, prodotti e movimenti di magazzino, conti e movimenti contabili, ecc.)

Due entità: Anagrafica e Movimento

Associazione: uno a molti



Ogni *Anagrafica* può essere relativa a uno o più *Movimenti*.

Ogni *Movimento* deve essere riferito a una sola *Anagrafica*.

### Tabelle derivate :

Anagrafica (Codice, Nome, Indirizzo)

Movimento (Numero, Descrizione, Data, Importo, *Codice*)

Codice : chiave della tabella Anagrafica

Numero : chiave della tabella Movimento

*Codice* : **chiave esterna della tabella Movimento**

*serve per collegare le righe della  
tabella movimento alle righe  
della tabella "Anagrafica"*

## Le operazioni relazionali nel linguaggio SQL

### Selezione

→ Restituisce tutte le colonne una sola di alcune righe

```
SELECT *  
FROM NomeTabella  
WHERE Condizione
```

Esempio:

Date le tabelle:

Anagrafica (Codice, Nome, Indirizzo)

Movimento (Numero, Descrizione, Data, Importo, Codice)

si vuole ottenere l'elenco delle anagrafiche con Indirizzo = 'Milano'

L'operazione relazionale che consente di ottenere i dati richiesti è:

Selezione di Anagrafica per Indirizzo = 'Milano'

### SQL

Select \* → tutte le colonne  
From Anagrafica  
Where Indirizzo = 'Milano' → solo alcune righe  
↳ di solito i comandi si scrivono in generale: in maiuscolo, ma funzione anche in minuscolo | le variabili si scrivono in minuscolo

### SQL

```
Select *  
From Anagrafica  
Where Indirizzo = [Quale Indirizzo]
```

## Le operazioni relazionali nel linguaggio SQL

### Proiezione

→ - opera su una tabella,  
- restituisce tutte le righe  
- restituisce solo alcune colonne

```
SELECT Colonna1, Colonna2, ...  
FROM NomeTabella
```

Esempio:

Date le tabelle:

Anagrafica (Codice, Nome, Indirizzo)

Movimento (Numero, Descrizione, Data, Importo, Codice)

Elenco dei codici e dei nomi delle anagrafiche

*Operazione relazionale: proiezione di Anagrafica su Codice e Nome*

**SQL**

```
Select Codice, Nome  
From Anagrafica
```

Elenco dei movimenti con data e importo

*Operazione relazionale: proiezione di Movimento su Data e Importo*

**SQL**

```
Select Data, Importo  
From Movimento
```

## **Le operazioni relazionali nel linguaggio SQL**

### **Congiunzione**

```
SELECT *  
  
FROM Tabella1, Tabella2  
  
WHERE Tabella1.Attrib1 = Tabella2.Attrib2
```

→ **abbinare le tabelle**  
- **restituisce tutte le colonne**  
**congiungendo\* le righe di tabelle**  
**diverse**  
- **posso aggiungere condizioni**  
**sugli elementi delle righe**

*Esempio:*

Date le tabelle:

Anagrafica (Codice, Nome, Indirizzo)

Movimento (Numero, Descrizione, Data, Importo, *Codice*)

Elenco dei movimenti con i dati anagrafici ad essi relativi

\* **scelgo una colonna**  
**che deve aver gli**  
**stessi elementi in**  
**tabelle diverse per**  
**congiungere le righe**

*Operazione relazionale: congiunzione di Anagrafica su Codice e di Movimento su Codice*

**SQL**

```
Select *  
From Anagrafica, Movimento  
Where Anagrafica.Codice= Movimento.Codice
```

## Le operazioni relazionali nel linguaggio SQL

### Uso di più operatori

```
SELECT Colonna1, Colonna2, ...  
  
FROM Tabella1, Tabella2  
  
WHERE Tabella1.Attrib1 = Tabella2.Attrib2  
  
AND Condizione
```

Esempio:

Date le tabelle:

Anagrafica (Codice, Nome, Indirizzo)

Movimento (Numero, Descrizione, Data, Importo, *Codice*)

Elenco dei movimenti con Nome e Importo riferiti alle anagrafiche aventi un indirizzo prefissato

Operazioni relazionali:

1. Selezione di Anagrafica per Indirizzo = prefissato
2. Congiunzione della tabella ottenuta su Codice e di Movimento su Codice
3. Proiezione della tabella ottenuta su Nome e Importo

### SQL

```
Select Nome, Importo  
From Anagrafica, Movimento  
Where Anagrafica.Codice= Movimento.Codice  
And Indirizzo = [Quale Indirizzo]
```

*Mostra le colonne nome, importo;  
delle tabelle Anagrafica e Movimento;  
creando r.ghe in base al codice;  
Filtro in base all'indirizzo.*

**Esercizio:** Con riferimento al modello di database con Anagrafica e Movimento, risolvere la seguente interrogazione, specificando le operazioni relazionali e la codifica nel linguaggio SQL.

Elenco dei movimenti relativi a un codice prefissato.

Tabelle derivate:

Anagrafica (Codice, Nome, Indirizzo)

Movimento (Numero, Descrizione, Data, Importo, *Codice*)

Codice: chiave della tabella Anagrafica

Numero: chiave della tabella Movimento

*Codice*: chiave esterna della tabella Movimento

Operazione relazionale:

Selezione di Movimento per Codice = [prefissato]

```
SELECT *  
FROM Movimento  
WHERE Codice = [...]
```

*non serve specificare la  
tabella in quanto devo  
selezionare da una sola  
tabella...*

## SQL

```
Select *  
From Movimento  
Where Codice = [prefissato]
```

**Esercizio:** Con riferimento al modello di database con Anagrafica e Movimento, risolvere la seguente interrogazione, specificando le operazioni relazionali e la codifica nel linguaggio SQL.

Elenco dei movimenti con codice, data e importo.

Tabelle derivate:

Anagrafica (Codice, Nome, Indirizzo)

Movimento (Numero, Descrizione, Data, Importo, *Codice*)

Codice : chiave della tabella Anagrafica

Numero : chiave della tabella Movimento

*Codice* : chiave esterna della tabella Movimento

```
SELECT Codice, Data, Importo  
FROM Movimento
```

Operazione relazionale:

Proiezione di Movimento su Codice, Data, Importo

## SQL

```
Select Codice, Data, Importo  
From Movimento
```

**Esercizio:** Con riferimento al modello di database con Anagrafica e Movimento, risolvere la seguente interrogazione, specificando le operazioni relazionali e la codifica nel linguaggio SQL.

Elenco con Nome anagrafico e Numero di registrazione dei movimenti.

Tabelle derivate:

Anagrafica (Codice, Nome, Indirizzo)

Movimento (Numero, Descrizione, Data, Importo, *Codice*)

Codice: chiave della tabella Anagrafica

Numero: chiave della tabella Movimento

*Codice*: chiave esterna della tabella Movimento

```
SELECT Name, Numero  
FROM Anagrafica, Movimento  
WHERE Anagrafica.Codice =  
Movimento.Codice
```

non specifico perché  
sono unici (in questo  
esempio)

Operazioni relazionali:

Congiunzione di Movimento su Codice e di Anagrafica su Codice

Proiezione della tabella ottenuta su Nome, Numero



## SQL

```
Select Nome, Numero  
From Movimento, Anagrafica  
Where Movimento.Codice = Anagrafica.Codice
```

**Esercizio:** Con riferimento al modello di database con Anagrafica e Movimento, risolvere la seguente interrogazione, specificando le operazioni relazionali e la codifica nel linguaggio SQL.

Data dei movimenti con Indirizzo dell'anagrafica.

Tabelle derivate:

Anagrafica (Codice, Nome, Indirizzo)

Movimento (Numero, Descrizione, Data, Importo, *Codice*)

Codice: chiave della tabella Anagrafica

Numero: chiave della tabella Movimento

*Codice*: chiave esterna della tabella Movimento

```
SELECT Data, Indirizzo  
FROM Anagrafica, Movimento  
WHERE Anag.Codice = Movim.Codice
```

Operazioni relazionali:

Congiunzione di Movimento su Codice e di Anagrafica su Codice

Proiezione della tabella ottenuta su Data, Indirizzo

## SQL

```
Select Data, Indirizzo  
From Movimento, Anagrafica  
Where Movimento.Codice = Anagrafica.Codice
```

**Esercizio:** Con riferimento al modello di database con Anagrafica e Movimento, risolvere la seguente interrogazione, specificando le operazioni relazionali e la codifica nel linguaggio SQL.

Nome anagrafico, Data e Importo dei movimenti riferiti a un indirizzo prefissato

Tabelle derivate:

Anagrafica (Codice, Nome, Indirizzo)

Movimento (Numero, Descrizione, Data, Importo, *Codice*)

Codice: chiave della tabella Anagrafica

Numero: chiave della tabella Movimento

*Codice*: chiave esterna della tabella Movimento

```
SELECT Name, Data, Importo  
FROM Anagrafica, Movimento  
WHERE Anag.Codice = Movim.Codice  
AND Indirizzo = [...]
```

Operazioni relazionali:

Selezione di Anagrafica per Indirizzo = [prefissato]

Congiunzione della tabella ottenuta su Codice e di Movimento su Codice  
Proiezione della tabella ottenuta su Nome, Data, Importo

## SQL

```
Select Nome, Data, Importo
From Movimento, Anagrafica
Where Movimento.Codice = Anagrafica.Codice
And Indirizzo = [prefissato]
```

## Le funzioni di aggregazione *ottenere informazioni aggiuntive*

sono funzioni predefinite che agiscono sui valori contenuti in insiemi di righe della tabella e restituiscono un valore calcolato.

### Funzione COUNT

La funzione **COUNT** restituisce il numero di righe presenti in una tabella.

→ Se nella condizione, conta le righe che rispettano tale condizione

```
Select Count (*)
From NomeTabella
```

*Si tratta di una funzione*

(includere quelle con campi di tipo *Null*)

```
Select Count (NomeAttributo)
From NomeTabella
```

(escluse le righe che hanno valore *Null* nella colonna dell'attributo specificato)

*Esempio:*  
Data la tabella:  
Anagrafica (Codice, Nome, Indirizzo)

```
SELECT COUNT (*)
FROM Anagrafica
WHERE Indirizzo = [...]
```

Numero delle persone registrate nella tabella delle anagrafiche aventi un indirizzo prefissato

## SQL

```
Select Count (*)
From Anagrafica
Where Indirizzo = [Quale Indirizzo]
```

## Le funzioni di aggregazione: Funzione SUM

Restituisce la **somma di tutti i valori contenuti in una colonna specificata** (l'attributo utilizzato nel calcolo deve essere di tipo numerico)

```
Select Sum (NomeAttributo)
```

```
From NomeTabella
```

*Esempio:*

Data la tabella:

Movimento (Numero, Descrizione, Data, Importo, Codice)

```
SELECT SUM (Importo)
FROM Movimento
WHERE Codice = [...]
```

Importo totale dei movimenti riferiti a un codice prefissato

**SQL**

```
Select Sum(Importo)
From Movimento
Where Codice = [Quale Codice]
```

## Le funzioni di aggregazione: Funzione AVG

Calcola la **media (average) dei valori (numerici) contenuti in una determinata colonna di una tabella.**

```
Select Avg (NomeAttributo)
```

```
From NomeTabella
```

(non include nel calcolo i valori di tipo *Null* presenti nella colonna)

*Esempio:*

Data la tabella:

Movimento (Numero, Descrizione, Data, Importo, Codice)

```
SELECT AVG (Importo)
FROM Movimento
```

Importo medio dei movimenti

**SQL**

```
Select Avg(Importo)
From Movimento
```

## ***Le funzioni di aggregazione: Funzioni MIN e MAX***

Restituiscono rispettivamente il valore minimo e il valore massimo tra i valori della colonna specificata come argomento della funzione (anche per campi di tipo carattere).

```
Select Min(NomeAttributo), Max(NomeAttributo)
```

```
From NomeTabella
```

(ignorano i campi con valore *Null*)

*Esempio:*

Date le tabelle:

Anagrafica (Codice, Nome, Indirizzo)

Movimento (Numero, Descrizione, Data, Importo, *Codice*)

Valori minimo e massimo tra gli importi dei movimenti

**SQL**

```
Select Min(Importo), Max(Importo)  
From Movimento
```

Ultimo nome dell'anagrafica

**SQL**

```
Select Max(Nome)  
From Anagrafica
```

## **Ordinamento**

La clausola **ORDER BY** consente di ottenere i risultati di un'interrogazione ordinati secondo i valori contenuti in una o più colonne, tra quelle elencate accanto alla parola *Select*.

```
SELECT Colonna1, Colonna2
```

```
FROM NomeTabella
```

```
ORDER BY Colonna1
```

Ordinamento crescente: **ASC**

stringhe dalla A alla Z, numeri dal minore al maggiore

Ordinamento decrescente: **DESC**

stringhe dalla Z alla A, numeri dal maggiore al minore

*L'ordinamento crescente è quello di default (non occorre specificare ASC).*

*Esempio:*

Data la tabella:

Anagrafica (Codice, Nome, Indirizzo)

Elenco alfabetico delle anagrafiche

## SQL

```
Select Nome, Indirizzo  
From Anagrafica  
Order By Nome
```

## Raggruppamenti

La clausola **GROUP BY** serve per raggruppare un insieme di righe aventi lo stesso valore nelle colonne indicate: produce una riga di risultati per ogni raggruppamento. Viene usata con le funzioni di aggregazione (*Sum, Count...*): per ciascuna riga della tabella risultante viene prodotto un valore di raggruppamento.

```
SELECT Colonna, Funzione  
  
FROM NomeTabella  
  
GROUP BY Colonna
```

*Esempio:*

Data la tabella:

Movimento (Numero, Descrizione, Data, Importo, *Codice*)

Totale degli importi dei movimenti per ciascun codice anagrafico

## SQL

```
Select Codice, Sum(Importo)  
From Movimento  
Group By Codice
```

## Condizioni sui raggruppamenti

L'uso della clausola **HAVING** consente di sottoporre al controllo di una o più condizioni i gruppi creati con la clausola *Group by*.

La condizione scritta dopo *Having* normalmente controlla il valore restituito dalle funzioni di aggregazione (*Count, Sum, Avg, Min, Max*).

```
SELECT Colonna, Funzione  
  
FROM NomeTabella  
  
GROUP BY Colonna  
  
HAVING Condizione
```

*Esempio:*

Data la tabella:

Movimento (Numero, Descrizione, Data, Importo, Codice)

Importo medio dei movimenti per i codici aventi più di 20 movimenti registrati

## SQL

```
Select Codice, Avg(Importo)  
From Movimento  
Group By Codice  
Having Count (*) > 20
```

Attenzione alla differenza tra

**Where** che pone condizioni sulle righe della tabella

**Having** che controlla condizioni su gruppi di righe

## ***Le condizioni di ricerca***

Il linguaggio SQL utilizza operatori e predicati insieme alle clausole *Where* e *Having* per determinare i criteri di selezione rispettivamente delle righe e dei raggruppamenti.

Segni del confronto =, <, >, <>, >=, <=.

Più condizioni legate tra loro con gli operatori **AND** e **OR**, precedute eventualmente dall'operazione **NOT**.

Predicati: **BETWEEN**, **LIKE**, **IN**

## ***Le condizioni di ricerca***

### **BETWEEN**

Controlla se un valore è compreso all'interno di un intervallo di valori, inclusi gli estremi.

*Esempio:*

Data la tabella:

Movimento (Numero, Descrizione, Data, Importo, *Codice*)

Elenco dei movimenti con importo compreso tra 100 e 200

## SQL

```
Select *  
From Movimento  
Where Importo Between 100 And 200
```

## IN

Controlla le righe che hanno i valori di un attributo compresi in una lista di valori indicati dopo la parola *In*.

*Esempio:*

Data la tabella:

Anagrafica (Codice, Nome, Indirizzo)

Elenco delle anagrafiche con indirizzo Milano, Torino o Venezia

## SQL

```
Select *  
From Anagrafica  
Where Indirizzo IN ( 'Milano', 'Torino', 'Venezia' )
```

## ***Le condizioni di ricerca: LIKE***

Il predicato **Like** confronta il valore di un attributo di tipo carattere con un modello di stringa che può contenere caratteri jolly:

**\_** (*underscore*) per indicare un singolo carattere qualsiasi in quella posizione della stringa;

**%** (*percento*) per indicare una sequenza qualsiasi di caratteri in quella posizione della stringa.

*Per esempio:*

LIKE 'xyz%' vengono ricercate tutte le stringhe che iniziano con i caratteri 'xyz' ;

LIKE '%xyz' serve a ricercare tutte le stringhe che finiscono con i caratteri 'xyz' ;

LIKE '%xyz%' per tutte le stringhe che contengono al loro interno i caratteri 'xyz' ;

LIKE '\_xyz' controlla le stringhe di 4 caratteri che finiscono con xyz.

*Esempio:*

Data la tabella:

Anagrafica (Codice, Nome, Indirizzo)

Elenco delle anagrafiche con nome che inizia con 'Ros' (Rossi, Rosi, Rossini,...)

## SQL

```
Select *  
From Anagrafica  
Where Nome Like 'Ros%'
```

**Esercizio:** Con riferimento al database con Anagrafica e Movimento, risolvere la seguente interrogazione in linguaggio SQL.

Calcolare il numero dei movimenti con importo superiore a una cifra prefissata.

Tabelle:

Anagrafica (Codice, Nome, Indirizzo)

Movimento (Numero, Descrizione, Data, Importo, *Codice*)

## SQL

```
Select Count (*)  
From Movimento  
Where Importo > [Quale Importo minimo]
```

**Esercizio:** Con riferimento al database con Anagrafica e Movimento, risolvere la seguente interrogazione in linguaggio SQL.

Calcolare la somma degli importi per i movimenti che si riferiscono alle anagrafiche di un indirizzo prefissato.

Tabelle:

Anagrafica (Codice, Nome, Indirizzo)

Movimento (Numero, Descrizione, Data, Importo, *Codice*)

## SQL

```
Select Sum(Importo)  
From Movimento, Anagrafica  
Where Movimento.Codice = Anagrafica.Codice  
And Indirizzo = [Quale indirizzo]
```



**Esercizio:** Con riferimento al database con Anagrafica e Movimento, risolvere la seguente interrogazione in linguaggio SQL.

Calcolare la media degli importi per i movimenti aventi una descrizione prefissata.

Tabelle:

Anagrafica (Codice, Nome, Indirizzo)

Movimento (Numero, Descrizione, Data, Importo, *Codice*)

**SQL**

```
Select Avg (Importo)
From Movimento
Where Descrizione = [Quale descrizione]
```