

Schema relazionale

Tirocinante (matricola, nome, cognome, classe, voto, idazienda, *idtutor*)

Azienda (id, nome, luogo, settore, tipologia)

Tutor (id, nome, cognome, materia)

SELECT

Serve ad indicare gli attributi da visualizzare.

Ad esempio, per l'istruzione "visualizzare nome e cognome degli studenti che svolgono il tirocinio a Milano, ordinate in ordine crescente per nome" gli attributi da visualizzare saranno **nome** e **cognome** della tabella tirocinante, quindi la prima parte del comando SQL sarà: **SELECT nome, cognome**

Nel SELECT è inoltre possibile indicare delle informazioni aggiuntive:

COUNT (), per effettuare il conteggio di un determinato attributo, ad esempio COUNT(matricola) per contare il numero di matricole.

AVG(), per calcolare il valore medio di un determinato attributo, ad esempio AVG(voto) per calcolare il voto medio dei tirocinanti.

SUM(), per calcolare la somma di un determinato attributo, ad esempio SUM(voto) per calcolare la somma dei voti dei tirocinanti.

FROM

Serve ad indicare le tabelle utilizzate all'interno del comando SQL.

Ad esempio, per l'istruzione "visualizzare nome e cognome degli studenti che svolgono il tirocinio a Milano, ordinate in ordine crescente per nome" le tabelle coinvolte saranno Tirocinante ed Azienda, quindi la seconda parte del comando SQL sarà:

SELECT nome, cognome

FROM Tirocinante, Azienda

WHERE

Serve a specificare le condizioni da inserire nel comando SQL.

Nel caso dell'istruzione "visualizzare nome e cognome degli studenti che svolgono il tirocinio a Milano, ordinate in ordine crescente" le condizioni da inserire saranno due:

- Una condizione per specificare che ci interessano le aziende di Milano
- Una condizione per collegare le tabelle Tirocinante ed Azienda

Il comando SQL sarà:

```
SELECT nome, cognome  
FROM Tirocinante, Azienda  
WHERE Azienda.luogo= "Milano" AND Azienda.id= Tirocinante.IdAzienda
```

AND, OR

Sono degli operatori logici per aggiungere più condizioni all'interno di un singolo WHERE. In particolare AND sta per "e" mentre OR sta per "oppure". Nel nostro caso abbiamo utilizzato AND per aggiungere entrambe le condizioni

ORDER BY

Serve per ordinare il risultato del comando SQL. Si utilizza indicando l'attributo per cui si vuole ordinare e se l'ordinamento deve essere crescente (ASC) o decrescente (DESC).

Nel caso dell'istruzione "visualizzare nome e cognome degli studenti che svolgono il tirocinio a Milano, ordinate in ordine crescente per nome" il comando SQL sarà:

```
SELECT nome, cognome  
FROM Tirocinante, Azienda  
WHERE Azienda.luogo= "Milano" AND Azienda.id= Tirocinante.IdAzienda  
ORDER BY nome ASC
```

GROUP BY

Serve a raggruppare i risultati di una query SQL, aggregandoli in un'unica riga, da cui estrarre delle informazioni.

Ad esempio, considerando la tabella Azienda:

Id	Nome	Luogo	Settore
1	Amazon	Milano	Terziario
2	Reply	Torino	Terziario
3	Barilla	Bologna	Primario
4	Kinder	Milano	Primario
5	Ferrero	Milano	Primario
6	Magneti Marelli	Torino	Secondario
7	Granarolo	Bologna	Primario

Possiamo utilizzare il group by per contare il numero di aziende per ogni settore.

La query SQL sarebbe

```
SELECT Settore, COUNT (settore)  
FROM Azienda  
GROUP BY Settore
```