**Web Developer**

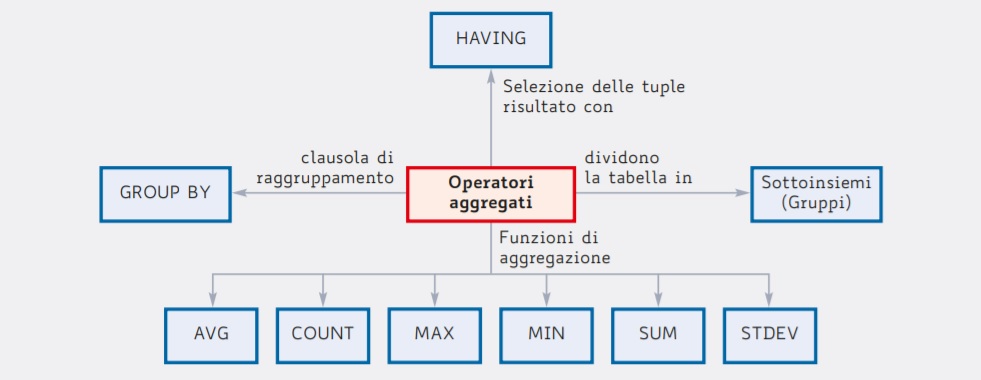
**Unità Formativa (UF): Fondamenti di base di dati**

**Docente: Adelchi Valenti**

**Titolo argomento: Linguaggio SQL – Gli operatori aggregati**

**GLI OPERATORI AGGREGATI**

Le operazioni di aggregazione consentono di raggruppare le tuple per effettuare calcoli specifici (ad es. sommatorie, medie). Possiamo dividere la tabella risultante da una query con operatori aggregati in sottoinsiemi, raggruppando le tuple contenenti gli stessi valori per un insieme di attributi.



Gli operatori di aggregazione (a differenza degli operatori matematici, che agiscono soltanto sulla tupla in corso di elaborazione) possono essere applicati a più tuple della tabella.

Operatori di aggregazione:

* AVG (nome campo) . Calcola la media aritmetica;
* COUNT (espressione | \* ) . Conta le righe;
* MAX (espressione) . Calcola il valore massimo;
* MIN (espressione) . Calcola il valore minimo;
* SUM (nome campo numerico) . Calcola la somma totale;
* STDDEV (nome campo numerico) . Calcola la deviazione standard.

Gli operatori aggregati possono comparire solo dopo SELECT e HAVING.

**Esempio**. La tabella concessionario riporta i modelli ordinabili presso un concessionario



Se volessimo sapere quanti modelli sono ordinabili presso il concessionario, la query sarebbe:

SELECT COUNT(id) AS totale\_vetture\_ordinabili FROM concessionario;

che restituirà:



L’operatore di aggregazione COUNT agisce sull’intera tabella e non solo sulla tupla in corso di elaborazione restituendo un unico risultato.

Se invece volessimo sapere il tempo medio di attesa per la consegna di una qualsiasi auto presso il concessionario, la query sarebbe:

SELECT AVG(giorni\_di\_consegna) AS tempo\_medio FROM concessionario;

e il risultato sarà:



**Regole di visibilità**

I campi presenti nella target list di una query di aggregazione devono sottostare ad alcune regole, chiamate regole di visibilità.

Se volessimo calcolare il prezzo medio delle vetture aventi come tipo ‘SUV’

la seguente query sarebbe sbagliata:

SELECT AVG(prezzo), modello FROM concessionario WHERE Tipo = "SUV";

*Errore: In aggregated query without GROUP BY, expression #2 of SELECT list contains nonaggregated column 'join\_example.concessionario.modello';*

In pratica non posso aggregare per modello dal momento che i modelli delle vetture di tipo ‘SUV’ sono diversi.

**MAX e MIN**

MAX e MIN restituiscono i valori più alti e più bassi contenuti all’interno di una colonna.

Per i campi numerici restituisce il valore numerico, mentre per quelli testuali seleziona il campo che, in ordine alfabetico, è ultimo (max) o primo (min), secondo la tabella dei codici ASCII.

È possibile, inoltre, usare come argomento degli operatori MIN e MAX anche espressioni matematiche.

Esempio:

SELECT MIN(Prezzo) AS prezzo\_minore FROM concessionario;

restituirà il prezzo più basso fra quelli riportati in tabella



**SUM**

L’operatore SUM è rigorosamente applicabile solo a campi numerici e restituisce la somma dei valori contenuti nella colonna prescelta.

Esempio:

SELECT SUM(prezzo) AS somma\_dei\_prezzi FROM concessionario WHERE tipo = "SUV";

restiturà la somma dei prezzi di tutte le vetture aventi come tipo ‘SUV’.



**GROUP BY**

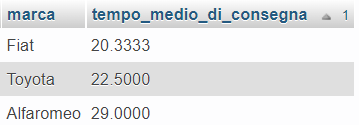
La clausola GROUP BY serve per raggruppare e per elaborare in modo uniforme righe diverse che nella tabella hanno valori uguali in una determinata colonna.

GROUP BY quindi determina la creazione di una singola tupla per tutti i record con il medesimo contenuto nell’attributo (o negli attributi) specificato dopo la parola chiave GROUP BY.

Esempio: se volessi conoscere il tempio medio di attesa di consegna della vettura per ogni brand, la query sarebbe:

SELECT marca, avg(giorni\_di\_consegna) AS tempo\_medio\_di\_consegna FROM concessionario GROUP BY marca ORDER BY tempo\_medio\_di\_consegna;

che restituirà:



Nella clausola GROUP BY vanno inseriti tutti i campi di selezione[[1]](#footnote-2) che non compaiono nell’operatore di aggregazione. Nel caso precedente il campo ‘marca’ compare sia nella SELECT che nella GROUP BY.

Una query del tipo:

SELECT marca, avg(giorni\_di\_consegna) AS tempo\_medio\_di\_consegna FROM concessionario

GROUP BY modello

ORDER BY tempo\_medio\_di\_consegna;

sarebbe sbagliata.

**CLAUSOLA “HAVING”**

Se la clausola GROUP BY viene utilizzata per aggregare i valori, HAVING filtra i record in base a questi valori aggregati.

SELECT marca, avg(giorni\_di\_consegna) AS tempo\_medio\_di\_consegna FROM concessionario

GROUP BY marca

HAVING tempo\_medio\_di\_consegna > 10

ORDER BY tempo\_medio\_di\_consegna;

Non solo raggruppo per ‘marca’ ma filtro anche il risultato per tempo\_medio\_di\_consegna.

1. *MySql ha di default il settaggio ONLY\_FULL\_GROUP\_BY attivo, mentre MariaDB no: questo significa che utilizzando MariaDB non è necessario inserire tutti i campi di selezione (i genere basta la PK o un campo univoco).* [↑](#footnote-ref-2)