

Modello Fisico in MySQL: Join e Aggregazioni

Francesco Gobbi

I.I.S.S. Galileo Galilei, Ostiglia

2 aprile 2025

Introduzione al Modello Fisico

- ▶ Il modello fisico riguarda l'organizzazione reale dei dati all'interno del database.
- ▶ In questa presentazione vedremo come implementare:
 - ▶ Diversi tipi di **JOIN** (LEFT, RIGHT e FULL).
 - ▶ Funzioni di aggregazione: COUNT, MAX, MIN e AVG (per la media).
 - ▶ L'uso di GROUP BY e HAVING per raggruppare e filtrare i dati.

Modello Logico Utilizzato

- ▶ Le entità principali sono:
 - ▶ **Studenti**: include campi come id, nome, cognome e altre informazioni personali.
 - ▶ **Iscrizioni**: rappresenta la relazione tra gli studenti e i corsi a cui si iscrivono.
 - ▶ **Esami**: registra i voti ottenuti dagli studenti negli esami.
 - ▶ **Classi**: raggruppa gli studenti e definisce l'unità didattica.
 - ▶ **Docenti**: identifica gli insegnanti responsabili delle classi.

Modello Fisico

In notazione in riga: nomeTabella(attributi).

Gli attributi chiave primaria sono sottolineati una volta (...) e quelli chiave esterna due volte (...).

- ▶ **Studenti**(id, nome, cognome, classeld)
- ▶ **Docenti**(id, nome, cognome, materia)
- ▶ **Classi**(id, nome, docenteld)
- ▶ **Iscrizioni**(id, studenteld, corso)
- ▶ **Esami**(id, voto, classe)

LEFT JOIN

LEFT JOIN: Restituisce tutte le righe dalla tabella sinistra e solo quelle corrispondenti dalla tabella destra. **Esempio:**

```
1 SELECT s.nome, i.corso
2 FROM studenti AS s
3 LEFT JOIN iscrizioni AS i
4   ON s.id = i.studenteId;
```

Output: Verranno mostrati i nomi degli studenti e il relativo corso. Per gli studenti senza iscrizioni, il campo "corso" apparirà come NULL.

RIGHT JOIN

RIGHT JOIN: Restituisce tutte le righe dalla tabella destra e solo quelle corrispondenti dalla tabella sinistra. **Esempio:**

```
1 SELECT s.nome, i.corso
2 FROM studenti AS s
3 RIGHT JOIN iscrizioni AS i
4   ON s.id = i.studenteId;
```

Output: Verranno mostrate tutte le iscrizioni. Se un'iscrizione non ha uno studente associato, il campo "nome" sarà NULL.

FULL JOIN in MySQL

FULL JOIN: MySQL non supporta direttamente il FULL OUTER JOIN. **Soluzione:** Combinare LEFT JOIN e RIGHT JOIN con UNION. **Esempio:**

```
1 SELECT s.nome, i.corso
2 FROM studenti AS s
3 LEFT JOIN iscrizioni AS i
4   ON s.id = i.studenteId
5 UNION
6 SELECT s.nome, i.corso
7 FROM studenti AS s
8 RIGHT JOIN iscrizioni AS i
9   ON s.id = i.studenteId;
```

Output: Combina i risultati del LEFT e del RIGHT JOIN, mostrando tutte le righe di entrambe le tabelle. I campi mancanti verranno visualizzati come NULL.

Funzioni di Aggregazione

Aggregazioni: Permettono di eseguire calcoli su un insieme di righe.

- ▶ COUNT: Conta il numero di righe.
- ▶ MAX: Restituisce il valore massimo.
- ▶ MIN: Restituisce il valore minimo.
- ▶ AVG: Calcola la media.

Esempio:

```
1 SELECT COUNT(*) AS totale_esami,  
2         MAX(voto) AS voto_massimo,  
3         MIN(voto) AS voto_minimo,  
4         AVG(voto) AS media_voti  
5 FROM esami;
```

Output: La query restituisce:

- ▶ totale_esami: il numero totale di record nella tabella Esami.
- ▶ voto_massimo: il voto più alto registrato.
- ▶ voto_minimo: il voto più basso registrato.
- ▶ media_voti: la media dei voti.

GROUP BY e HAVING

GROUP BY: Raggruppa le righe in base a una o più colonne.

HAVING: Filtra i gruppi creati da GROUP BY. **Esempio:** Calcolare il numero di studenti e la media dei voti per ogni classe, filtrando solo quelle con media maggiore o uguale a 25.

```
1 SELECT classe, COUNT(*) AS num_studenti, AVG(voto) AS
   media_voti
2 FROM esami
3 GROUP BY classe
4 HAVING AVG(voto) >= 25;
```

Output: Per ciascuna classe, la query mostra:

- ▶ num_studenti: il numero di studenti.
- ▶ media_voti: la media dei voti.

Verranno visualizzate solo le classi la cui media è maggiore o uguale a 25.