

## Solo interrogazioni / 11-06

È un'interrogazione dentro un'altra, solitamente posta nel WHERE (HAVING). SQL permette di farlo anche nel FROM  
solo se lo group by

Esempio - Trovare il titolo del film di valutazione più alta

```
SELECT titolo
FROM Film
WHERE valutazione = (SELECT max(valutazione) FROM Film)
```

X WHERE max(valutazione) ottengo un numero, non un booleano

X WHERE valutazione = max(valutazione) voluto tuple per tuple con where, non rispetto tutte le altre

Esempio - Trovare i titoli di film che hanno stessa valutazione di "mediterraneo"

```
SELECT titolo
FROM Film
WHERE valutazione = (SELECT valutazione FROM Film WHERE titolo = "mediterraneo")
```

query **scalare** è una query che restituisce un solo valore  
→ in questo caso ho un **ERRORE** perché le query può restituire più valori, nel caso in cui venisse aggiunto un film con lo stesso titolo  
(più valori = query **table**)

Posso quantificare con **ANY** o **ALL** nel caso il confronto deve essere rispetto ed almeno uno o rispetto tutti

```
SELECT titolo
FROM Film
WHERE valutazione > ANY/ALL (SELECT valore FROM Film WHERE titolo = "mediterr.")
```

WHERE  $A \theta \{ANY/ALL\} \text{ subq}$

sic  $\{v_1, \dots, v_n\}$  l'insieme dei valori restituiti da subq

Si restituiscano le tuple d.c.

ANY:  $\exists i, A \theta v_i \text{ è true}$

ALL:  $\forall i, A \theta v_i \text{ è true}$

" = ANY " cioè un valore è uguale ad almeno uno nell'insieme

" < > ALL " cioè diverso da tutti

sono equivalenti a rispettivamente **IN** e **NOT IN**

Utilizzare le subquery richiede l'ottimizzatore (e l'esecuzione)

Esempio - equivalente di max

```
SELECT titolo
FROM Film
WHERE valore >= ALL (SELECT valore FROM Film)
```

per trovare la  
valutazione più  
alta

se l'attributo non ha valori NULL (altrimenti risultato unknown)

il confronto è corretto ✓

**Sottoquery correlate** : ha dipendenza dalle proprietà della query esterna

Ad esempio determinare i titoli dei film con valutazione più alta della media del loro genere

```
SELECT titolo  
FROM Film  
WHERE valore
```

> (SELECT AVG (valore) FROM Film)

query esterna

↳ il risultato della sottoquery dipende dal film che si sta analizzando nella query esterna

La sua proprietà può assumere valore diverso  
tuple x tuple

Dunque:

```
SELECT titolo  
FROM Film come as (AS) F  
WHERE valore
```

> (SELECT AVG (valore) FROM Film  
WHERE genere = F.genere)

↳ riferimento  
alla sottoquery

La clausola **EXISTS** (subquery) permette di vedere se restituisce almeno una tuple (true) o no (false).

**Esempio** - I registi che hanno girato film di tutti i generi

```
SELECT regista  
FROM Film
```

```
GROUP BY regista
```

```
HAVING COUNT (DISTINCT genere) =
```

```
(SELECT COUNT (genere) FROM Film);
```