

# Appunti Basi di Dati

Luca Seggiani

4 Aprile 2024

## 1 Raggruppamento

Il raggruppamento (o aggregazione) divide in gruppi i record risultanti dalle clausole FROM e WHERE (detti record target), sulla base di un certo attributo particolare in comune, che resta quindi costante in un determinato gruppo. Poniamo di voler calcolare, nello schema clinica, per ogni specializzazione, la parcella media dei medici:

```
1 SELECT Specializzazione , AVG(Parcella) AS ParcellaMedia  
2 FROM Medico  
3 GROUP BY Specializzazione;
```

Dove il select di Specializzazione e AVG(Parcella) ha senso solo a seguito del GROUP BY, con Specializzazione proprio come attributo di raggruppamento (!).

### Condizioni sui gruppi

Le condizioni sui gruppi sono controllate gruppo per gruppo (non record per record) e permettono di scartarne alcuni. Si applicano attraverso l'operatore HAVING. Poniamo di voler indicare le specializzazioni della clinica con più di due medici:

```
1 SELECT Specializzazione  
2 FROM Medico  
3 GROUP BY Specializzazione  
4 HAVING COUNT(*) > 2
```

Si nota che il DISTINCT non è necessario: ogni specializzazione viene presa comunque una volta sola. Vediamo adesso un caso un attimo più complesso: ottenere le specializzazioni con la più alta parcella media. Dovremo allora calcolare tutte le medie delle parcelle dei medici di ciascuna specializzazione, calcolare tra queste la più alta, e selezionare quindi le specializzazioni la cui media corrisponde a questa media massima.

Il massimo si troverà con:

```
1 SELECT MAX(D.MediaParcella)
2 FROM (
3   SELECT M2.Specializzazione, AVG(M2.Parcella) AS
4     MediaParcella
5   FROM Medico M2
6   GROUP BY M2.Specializzazione
7 )
8 AS D;
```

e la query completa sarà:

```
1 SELECT M.Specializzazione
2 FROM Medico M
3 GROUP BY Specializzazione
4 HAVING AVG(Parcella) =
5 (
6   SELECT MAX(D.MediaParcella)
7   FROM (
8     SELECT M2.Specializzazione, AVG(M2.Parcella) AS
9       MediaParcella
10    FROM Medico M2
11   GROUP BY M2.Specializzazione
12 )
13   AS D;
14 )
```

## 2 Subquery correlate

Abbiamo già visto le subquery non correlate (*non-correlated*): il result set di tali subquery viene calcolato una sola volta ed è indipendente dalle specifiche della query esterna (*outer query*). Nelle subquery correlate invece il result set dipende da ciascuna tupla della tupla esterna. L'ordine di esecuzione è: prima si esegue il WHERE della query esterna, e poi la subquery. Poniamo ad esempio di voler indicare matricola e parcella dei medici che hanno visitato per la prima volta almeno un paziente nel mese di ottobre 2013. Avremo bisogno di selezionare pazienti che non erano mai stati visitati da un determinato medico prima di ottobre:

```
1 SELECT DISTINCT V1.Medico
2 FROM Visita V1
3 WHERE YEAR(V1.Data) = 2013
4   AND MONTH (V1.Data) = 10
5   AND V1.Paziente NOT IN (
6     SELECT V2.Paziente
7     FROM Visita V2
```

```
8           WHERE V2.Medico = V1.Medico  
9               AND V2.Data < V1.Data  
10      )
```

Notiamo la particolarità della subquery correlata: la relazione V1, appartenente alla query esterna, viene usata nella clausola WHERE della correlata. Ricordiamo che una subquery correlata viene eseguita nuovamente per ogni tupla, ed è quindi particolarmente inefficiente dal punto di vista della complessità. Notiamo inoltre che una subquery può essere usata anche nella clausola SELECT della query, per calcolare velocemente un valore da inserire nel risultato. Ad esempio, vogliamo trovare tutti i pazienti di sesso maschile, indicandone nome e numero di visite effettuate. Quest'ultima informazione non sarà un dato da proiettare (nel WHERE), ma da calcolare nel SELECT.

```
1 SELECT Nome , (  
2     SELECT COUNT(*)  
3     FROM Visite V  
4     WHERE V.Paziente = P.CodFiscale  
5 ) AS NumeroVisite  
6 FROM Paziente P  
7 WHERE P.Sesso = 'M'
```