

Basi di dati

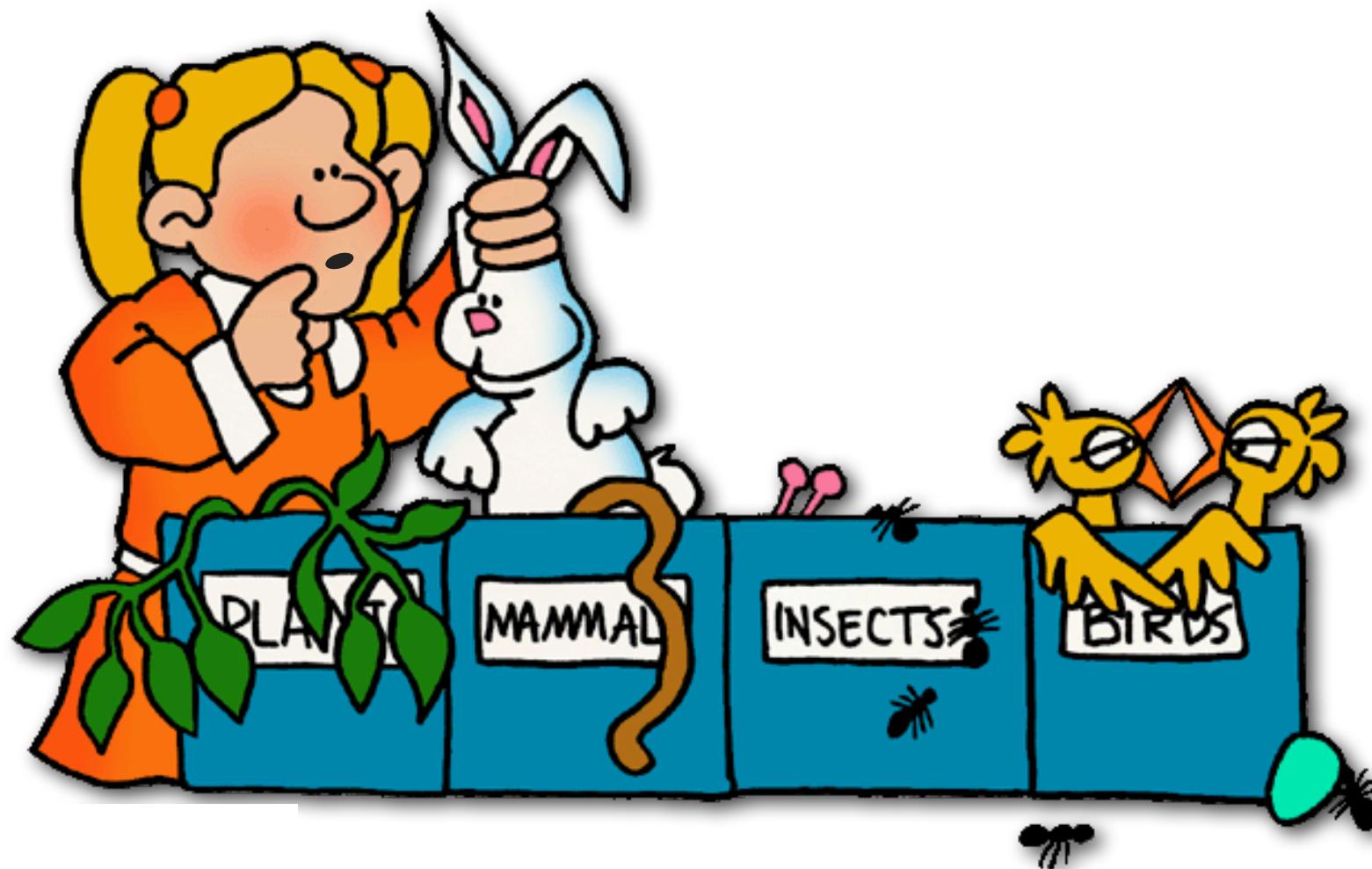
Corso di laurea in Ingegneria Informatica
Scuola di Ingegneria — Università di Pisa

Oracle MySQL
A.A. 2017-2018

Ing. Francesco Pistolesi

Postdoctoral Researcher
Data Science and Engineering Lab
Dipartimento di Ingegneria dell'Informazione
francesco.pistolesi@iet.unipi.it

Raggruppamento



Raggruppamento

la tabella target viene frammentata

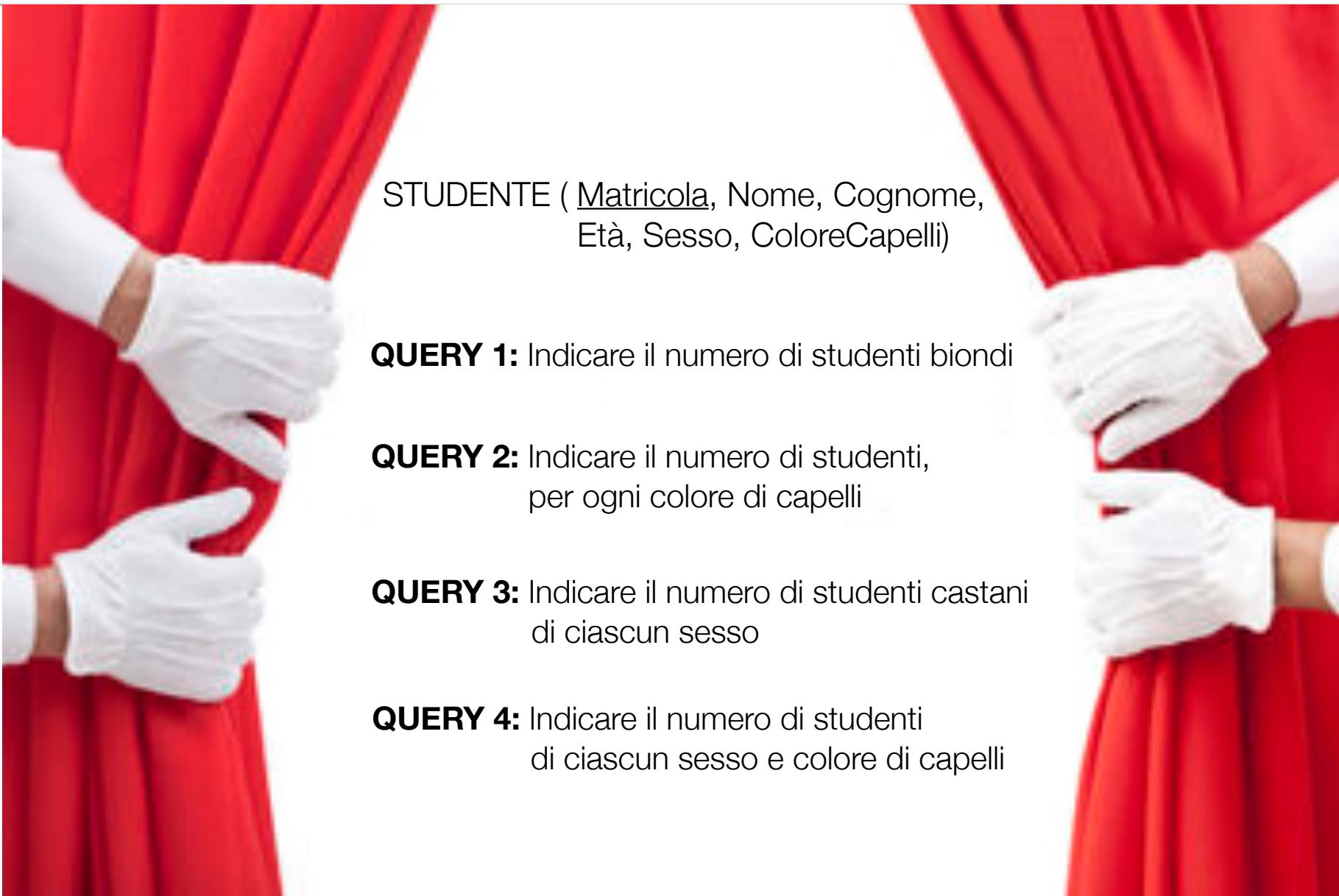


Suddivide un insieme di record in **gruppi di record**, all'interno di ognuno dei quali il valore di uno o più attributi è costante record per record

attributi di raggruppamento



Siparietto



STUDENTE (Matricola, Nome, Cognome,
Età, Sesso, ColoreCapelli)

QUERY 1: Indicare il numero di studenti biondi

QUERY 2: Indicare il numero di studenti,
per ogni colore di capelli

QUERY 3: Indicare il numero di studenti castani
di ciascun sesso

QUERY 4: Indicare il numero di studenti
di ciascun sesso e colore di capelli

Raggruppamento: un semplice esempio

Indicare la parcella media dei medici di ciascuna specializzazione

MEDICO

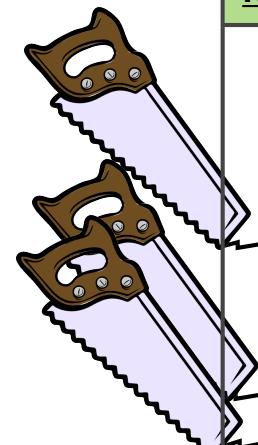
Matricola	Cognome	Nome	Specializzazione	Parcella
014	Indachi	Loredana	Cardiologia	180
003	Gialli	Franco	Otorinolaringoiatria	120
010	Celesti	Clelia	Neurologia	200
006	Bianchi	Ada	Ortopedia	160
002	Verdi	Luigi	Otorinolaringoiatria	150
017	Amaranti	Adevane	Cardiologia	250
001	Rossi	Mario	Otorinolaringoiatria	180
007	Rosi	Alvaro	Ortopedia	180
015	Ciani	Gualtiero	Cardiologia	230
018	Terra di Siena	Bruciata	Cardiologia	260

alcuni attributi sono stati volutamente omessi

Raggruppamento: un semplice esempio

Indicare la parcella media dei medici **di ciascuna specializzazione**

MEDICO



Matricola	Cognome	Nome	Specializzazione	Parcella
014	Indachi	Loredana	Cardiologia	180
003	Amaranti	Adevane	Cardiologia	250
010	Ciani	Gualtiero	Cardiologia	230
006	Terra di Siena	Bruciata	Cardiologia	260
002	Gialli	Franco	Otorinolaringoiatria	120
017	Verdi	Luigi	Otorinolaringoiatria	150
001	Rossi	Mario	Otorinolaringoiatria	180
012	Celesti	Clelia	Neurologia	200
015	Bianchi	Ada	Ortopedia	160
018	Rosi	Alvaro	Ortopedia	180

Raggruppamento: un semplice esempio

Indicare la parcella media dei medici **di ciascuna specializzazione**

Matricola	Cognome	Nome	Specializzazione	Parcella
014	Indachi	Loredana	Cardiologia	180
003	Amaranti	Adevane	Cardiologia	250
010	Ciani	Gualtiero	Cardiologia	230
006	Terra di Siena	Bruciata	Cardiologia	260

Matricola	Cognome	Nome	Specializzazione	Parcella
002	Gialli	Franco	Otorinolaringoiatria	120
017	Verdi	Luigi	Otorinolaringoiatria	150
001	Rossi	Mario	Otorinolaringoiatria	180

Matricola	Cognome	Nome	Specializzazione	Parcella
12	Celesti	Clelia	Neurologia	200

Matricola	Cognome	Nome	Specializzazione	Parcella
015	Bianchi	Ada	Ortopedia	160
018	Rosi	Alvaro	Ortopedia	180

Raggruppamento: un semplice esempio

Matricola	Cognome	Nome	Specializzazione	Parcella
014	Indachi	Loredana	Cardiologia	180
003	Amaranti	Adevane	Cardiologia	250
010	Ciani	Gualtiero	Cardiologia	230
006	Terra di Siena	Bruciata	Cardiologia	260

AVG(Parcella) = 230

Matricola	Cognome	Nome	Specializzazione	Parcella
002	Gialli	Franco	Otorinolaringoiatria	120
017	Verdi	Luigi	Otorinolaringoiatria	150
001	Rossi	Mario	Otorinolaringoiatria	180

AVG(Parcella) = 150

Raggruppamento: un semplice esempio

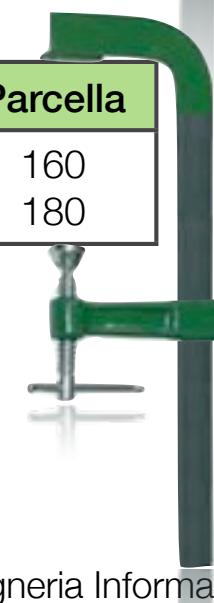
Matricola	Cognome	Nome	Specializzazione	Parcella
12	Celesti	Clelia	Neurologia	200

AVG(Parcella) = 200



Matricola	Cognome	Nome	Specializzazione	Parcella
015	Bianchi	Ada	Ortopedia	160
018	Rosi	Alvaro	Ortopedia	180

AVG(Parcella) = 170



Raggruppamento: un semplice esempio

Indicare la parcella media dei medici **di ciascuna specializzazione**

Specializzazione	ParcellaMedia
Cardiologia	230
Otorinolaringoiatria	150
Neurologia	200
Ortopedia	170



ogni gruppo collassa in un unico record, sempre!



OK, adesso basta con i
disegnini...come cavolo si fa questa
roba in MySQL?????

Raggruppamento in MySQL

Indicare la parcella media dei medici **di ciascuna specializzazione**

l'operatore è applicato gruppo per gruppo
(su un attributo a valori non costanti nel gruppo)

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

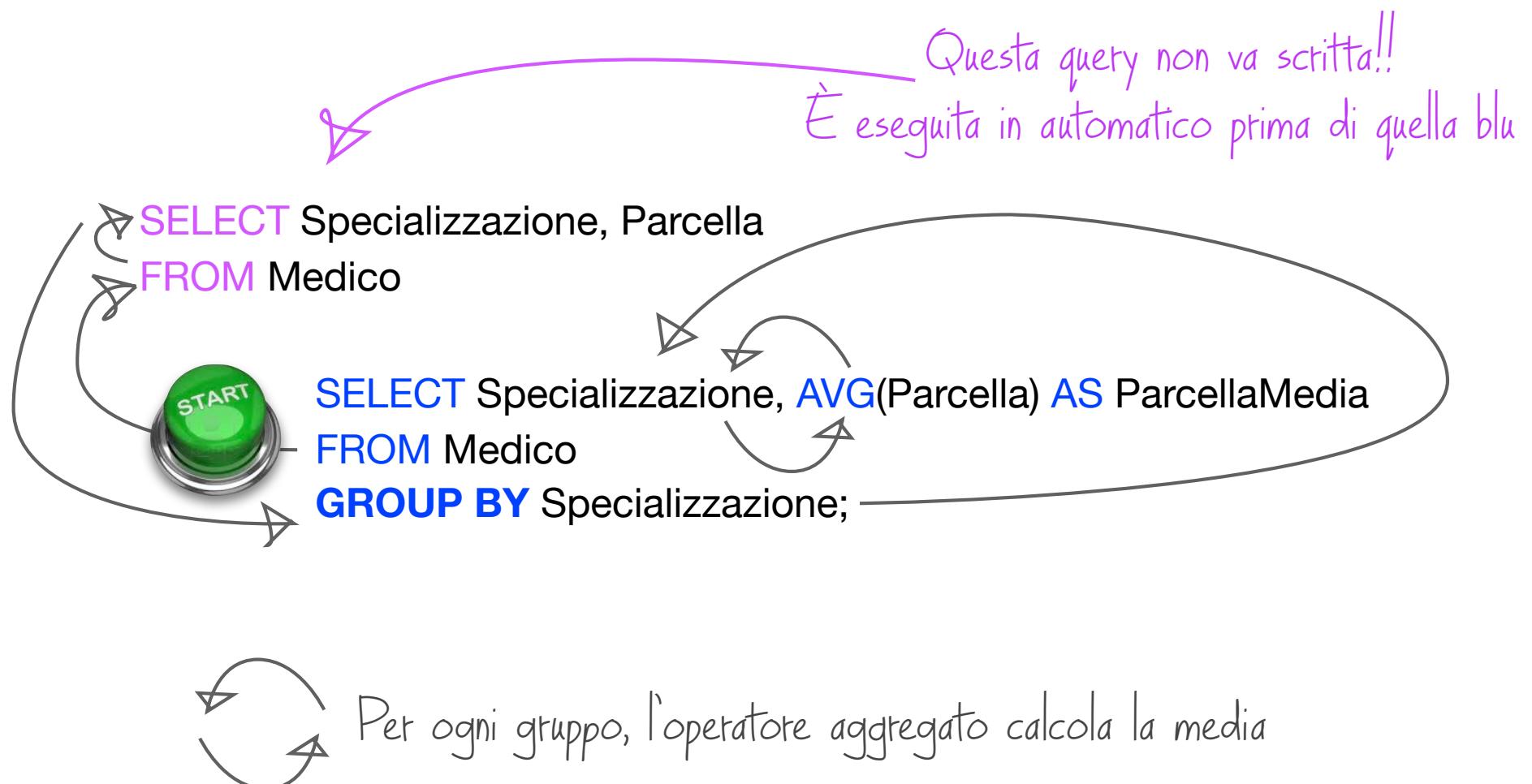
assume lo stesso valore
in un gruppo

assume valori diversi in un gruppo,
non si può proiettare senza aggregazione

la clausola "group by" permette di frammentare la tabella

Raggruppamento in MySQL: processazione

Indicare la parcella media dei medici **di ciascuna specializzazione**



Raggruppamento: regola fondamentale



Raggruppamento e proiezione

Per ogni specializzazione, calcolare la parcella media dei suoi medici

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

L'attributo di raggruppamento
DEVE COMPARIRE nella proiezione.

Il male è sempre in agguato...

Per ogni specializzazione medica, indicarne il nome, la parcella minima
e il cognome del medico a cui appartiene



Guai in vista

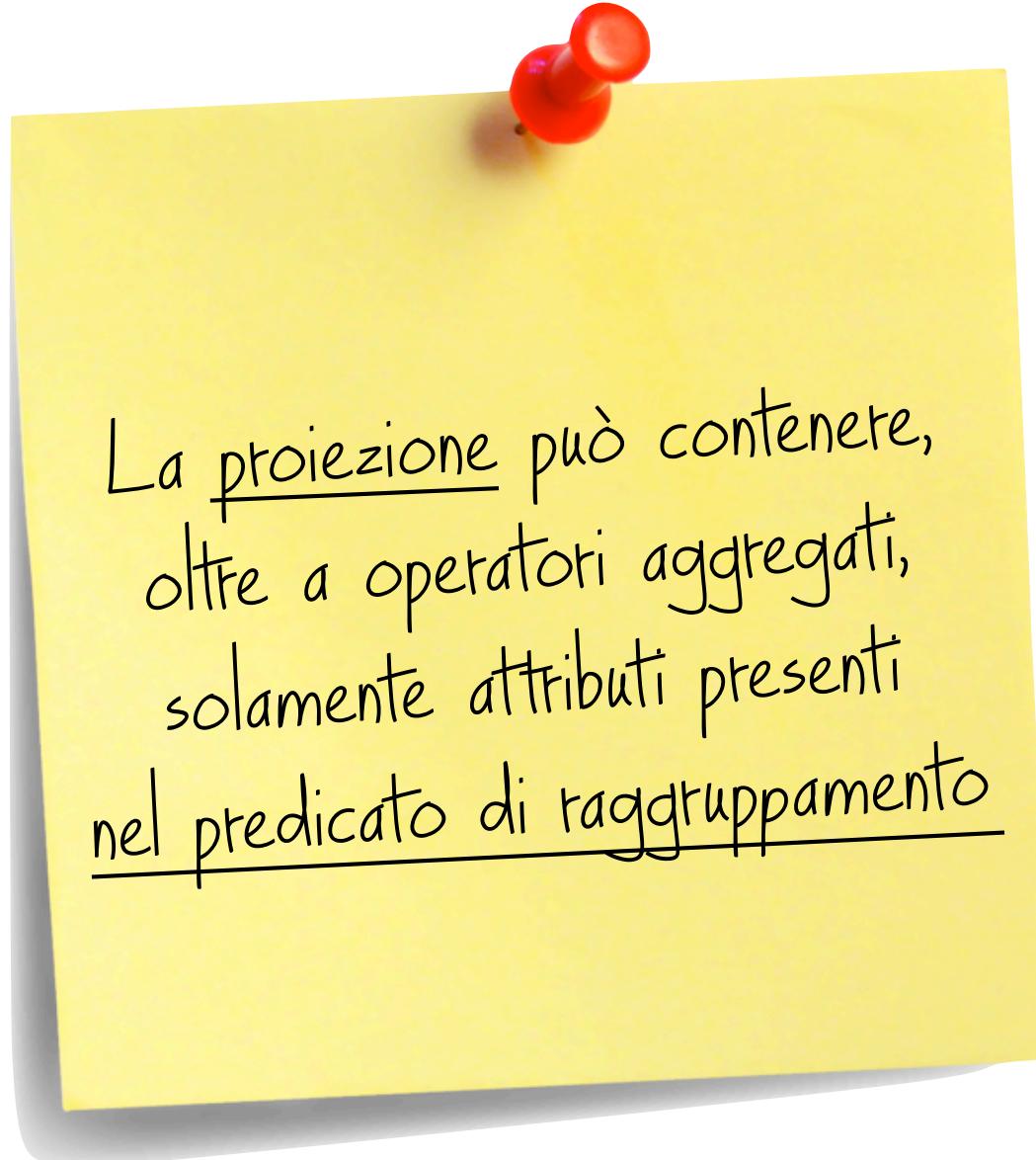


Per ogni specializzazione medica, indicarne il nome, la parcella minima
e il cognome del medico a cui appartiene,

~~SELECT Specializzazione, MIN(Parcella), Cognome
FROM Medico
GROUP BY Specializzazione;~~

Non è connesso al medico
avente parcella minima...
(e non è costante in un gruppo!)

Per non dimenticare...



Sconfiggiamo Malefica!



Sconfiggiamo Malefica!



```
SELECT Specializzazione, MIN(Parcella)  
FROM Medico  
GROUP BY Specializzazione
```



???

Per ogni specializzazione medica, indicarne il nome, la parcella minima
e il cognome del medico a cui appartiene

```
SELECT Cognome  
FROM Medico  
WHERE Parcella = "parcella minima della sua specializzazione"
```



Sconfiggiamo Malefica!



```
SELECT Specializzazione, MIN(Parcella)
      AS ParcellaMin
  FROM Medico
 GROUP BY Specializzazione
```

Specializzazione	ParcellaMin
Cardiologia	180
Otorinolaringoiatria	100
Neurologia	200
Ortopedia	160

```
SELECT Cognome, Specializzazione, Parcella
  FROM Medico
```

Cognome	Specializzazione	Parcella
Indachi	Cardiologia	180
Gialli	Otorinolaringoiatria	200
Celesti	Neurologia	200
Bianchi	Ortopedia	160
Verdi	Otorinolaringoiatria	150
Amaranti	Cardiologia	250
Rossi	Otorinolaringoiatria	100
Rosi	Ortopedia	180

Sconfiggiamo Malefica!



Per ogni specializzazione medica, indicarne il nome, la parcella minima
e il cognome del medico a cui appartiene

Cognome	Specializzazione	Parcella
Indachi	Cardiologia	180
Gialli	Otorinolaringoiatria	200
Celesti	Neurologia	200
Bianchi	Ortopedia	160
Verdi	Otorinolaringoiatria	150
Amaranti	Cardiologia	250
Rossi	Otorinolaringoiatria	100
Rosi	Ortopedia	180



Specializzazione	Parcella Min
Cardiologia	180
Otorinolaringoiatria	100
Neurologia	200
Ortopedia	160

Join naturale

Sconfiggiamo Malefica!



Per ogni specializzazione medica, indicarne il nome, la parcella minima
e il cognome del medico a cui appartiene

Cognome	Specializzazione	Parcella	ParcellaMin
Indachi	Cardiologia	180	180
Gialli	Otorinolaringoiatria	200	100
Celesti	Neurologia	200	200
Bianchi	Ortopedia	160	160
Verdi	Otorinolaringoiatria	150	100
Amaranti	Cardiologia	250	180
Rossi	Otorinolaringoiatria	100	100
Rosi	Ortopedia	180	160

Sconfiggiamo Malefica!



Per ogni specializzazione medica, indicarne il nome, la parcella minima
e il cognome del medico a cui appartiene

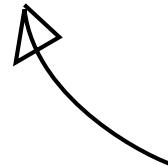
```
SELECT M.Specializzazione,  
       D.ParcellaMinima,  
       M.Cognome  
  FROM Medico M  
    NATURAL JOIN  
    (  
        SELECT Specializzazione, MIN(Parcella) AS ParcellaMinima  
      FROM Medico  
     GROUP BY Specializzazione  
    ) AS D  
 WHERE M.Parcella = D.ParcellaMinima;
```

Condizioni sui gruppi



Condizioni sui gruppi

perché esprimono una caratteristica di un gruppo, cioè globale, sommatoria, riepilogativa degli attributi variabili (non di raggruppamento) dei record che lo compongono



Sono espresse esclusivamente tramite **operatori di aggregazione** e permettono di **scartare gruppi**, qualora non siano soddisfatte



sono controllate gruppo per gruppo,
non record per record

Condizioni sui gruppi: esempio

Indicare le specializzazioni della clinica **con più di due medici**



è evidente che si deve ragionare sul numero
di medici di ciascuna specializzazione

Condizioni sui gruppi: esempio

Indicare le specializzazioni della clinica **con più di due medici**

MEDICO

Cognome	Specializzazione	Città
Indachi	Cardiologia	Pisa
Gialli	Otorinolaringoiatria	Pisa
Celesti	Neurologia	Firenze
Bianchi	Ortopedia	Milano
Verdi	Otorinolaringoiatria	Pisa
Amaranti	Cardiologia	Pisa
Rossi	Otorinolaringoiatria	Pisa
Rosi	Ortopedia	Roma
Terra di Siena	Cardiologia	Siena

Condizioni sui gruppi

Indicare le specializzazioni della clinica **con più di due medici**

Rossi	Otorinolaringoiatria	Pisa
Gialli	Otorinolaringoiatria	Pisa
Verdi	Otorinolaringoiatria	Pisa



COUNT(*) = 3

Bianchi	Ortopedia	Milano
Rosi	Ortopedia	Roma



COUNT(*) = 2

Amaranti	Cardiologia	Pisa
Indachi	Cardiologia	Pisa
Tetra di Siena	Cardiologia	Siena



COUNT(*) = 3

Celesti	Neurologia	Firenze
---------	------------	---------



COUNT(*) = 1

Condizioni sui gruppi

Indicare le specializzazioni della clinica **con più di due medici**

Rossi	Otorinolaringoiatria	Pisa
Gialli	Otorinolaringoiatria	Pisa
Verdi	Otorinolaringoiatria	Pisa

COUNT(*) = 3

Bianchi	Cirurgia	Milano
Zanini	Cirurgia	Roma

COUNT(*) = 2

Amaranti	Cardiologia	Pisa
Indachi	Cardiologia	Pisa
Tetra di Siena	Cardiologia	Siena

COUNT(*) = 3

Levi	Neurologia	Firenze
Castellani	Neurologia	Firenze

COUNT(*) = 1

Condizioni sui gruppi in MySQL

Indicare le specializzazioni della clinica **con più di due medici**

```
SELECT Specializzazione  
FROM Medico  
GROUP BY Specializzazione  
HAVING COUNT(*) > 2;
```

ma serve distinct?/



NO! Specializzazione è l'attributo di raggruppamento. È costante in ciascun gruppo. Che senso ha parlare di duplicati?

Condizioni sui gruppi in SQL: processazione

Indicare le specializzazioni della clinica **con più di due medici**



Si sono alleate contro di noi...

Indicare le specializzazioni con la più alta parcella media



Errore blu

Indicare le specializzazioni con la più alta parcella media

```
SELECT Specializzazione  
FROM Medie  
GROUP BY Specializzazione  
HAVING MAX(AVG(Parcella));
```

Questo è il delitto!!!

Gli operatori di aggregazione non si annidano mai!
La clausola having contiene sempre 2 parti connesse da
un operatore di confronto (=, >, <, >=, <=)



Ragioniamo con calma, cosa ci occorre?



Indicare le specializzazioni con la più alta parcella media

1. Media delle parcelle dei medici di ciascuna specializzazione
2. Calcolo della più alta parcella media
3. Specializzazioni con media parcelle superiore alla media globale delle parcelle



Risoluzione: primo passo

Indicare le specializzazioni con la più alta parcella media

1. Media delle parcelle dei medici di ciascuna specializzazione

```
SELECT Specializzazione, AVG(Parcella)
FROM Medico
GROUP BY Specializzazione;
```



Risoluzione: secondo passo

Indicare le specializzazioni con la più alta parcella media

2. Calcolo della più alta parcella media

```
SELECT MAX(D.MediaParcelle)
FROM
(
    SELECT M2.Specializzazione,
           AVG(M2.Parcella) AS MediaParcelle
    FROM Medico M2
    GROUP BY M2.Specializzazione
)
AS D;
```



Soluzione

Indicare le specializzazioni con la più alta parcella media

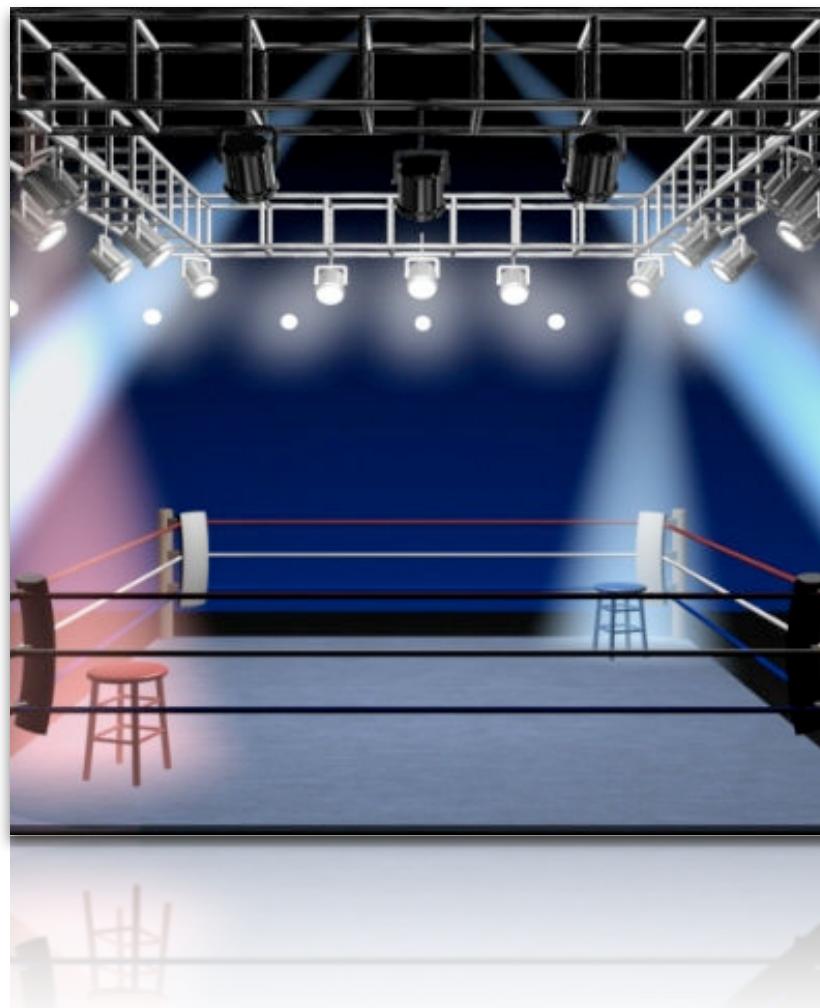
```
SELECT M.Specializzazione ← Proietta tutti i pari merito
FROM Medico M
GROUP BY M.Specializzazione
HAVING AVG(M.Parcella) =
(
    SELECT MAX(D.MediaParcille)
    FROM
        (
            SELECT M2.Specializzazione,
                   AVG(M2.Parcella) AS MediaParcille
            FROM Medico M2
            GROUP BY M2.Specializzazione
        )
    AS D
);
```

The code is annotated with handwritten numbers 1 and 2:

- Annotation 1 is located in the innermost subquery, which calculates the average price for each specialization. It consists of a green rectangular background with the number "1" in the top right corner.
- Annotation 2 is located in the outer query, specifically in the MAX function which retrieves the highest average price from the subquery. It consists of a light green rectangular background with the number "2" in the top right corner.

Condizioni sui gruppi vs. condizioni sui record

H
A
V
I
N
G

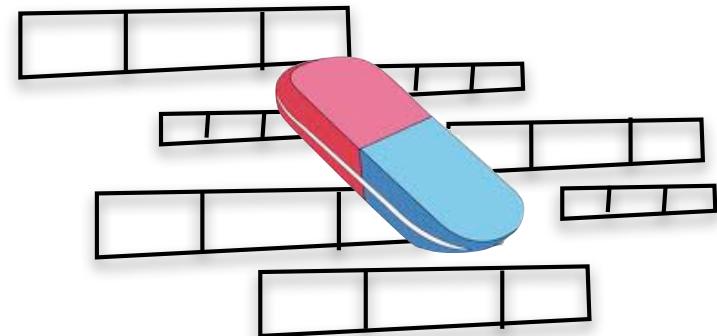


W
H
E
R
E

Condizioni sui gruppi vs. condizioni sui record

le condizioni nel **WHERE** sono applicate
ai record **prima** del raggruppamento

eliminano record



le condizioni nell'**HAVING** sono applicate
ai gruppi, **dopo** il raggruppamento

eliminano gruppi



Condizioni sui gruppi vs. condizioni sulle tuple



Condizioni sui gruppi: regola



Condizioni sui gruppi e sui record: esempio

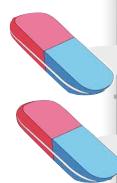


Condizioni sui gruppi e sui record: esempio

Indicare le specializzazioni con **più di due medici di Pisa**.

MEDICO

Cognome	Specializzazione	Città
Indachi	Cardiologia	Pisa
Gialli	Otorinolaringoiatria	Pisa
Verdi	Otorinolaringoiatria	Pisa
Amaranti	Cardiologia	Pisa
Rossi	Otorinolaringoiatria	Pisa
Tutti	Cardiologia	Città



Condizioni sui gruppi e sui record: esempio

Indicare le specializzazioni con **più di due medici** di Pisa.

Indonesia	Catatan	Pisau
Arranji	Catatan	Pisau

COUNT(*) = 2

Rossi	Otorinolaringoiatria	Pisa
Gialli	Otorinolaringoiatria	Pisa
Verdi	Otorinolaringoiatria	Pisa

COUNT(*) = 3

Soluzione

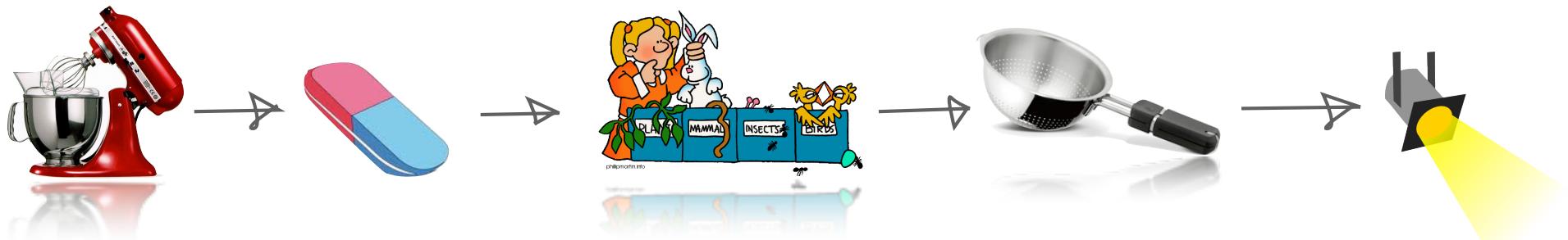
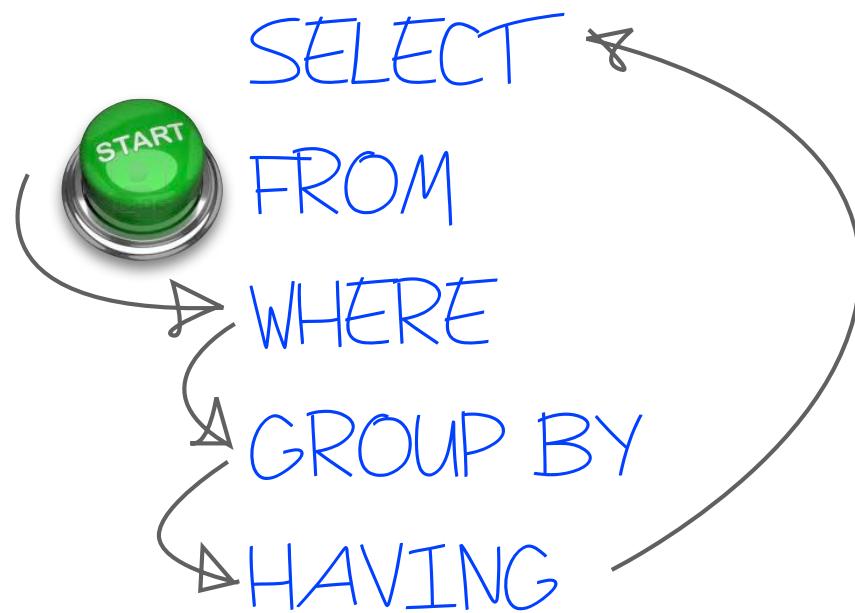
Indicare le specializzazioni con **più di due medici di Pisa**.

```
SELECT Specializzazione  
FROM Medico  
WHERE Citta = 'Pisa'  
GROUP BY Specializzazione  
HAVING COUNT(*) > 2;
```



Si contano i record all'interno
di ciascun gruppo. Ogni record
è relativo a un medico.

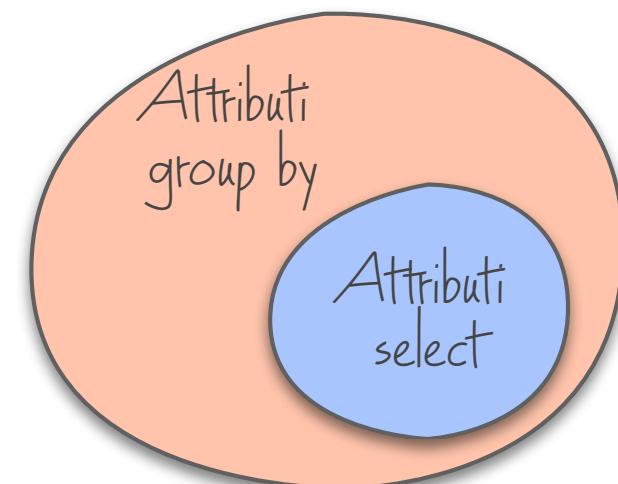
Ordine di esecuzione



Raggruppamento su più attributi

In generale, il raggruppamento può essere fatto **su un insieme di attributi**

Gli attributi non aggregati proiettati devono
OBBLIGATORIAMENTE essere
presenti nel predicato di raggruppamento!!!



Ancora sul raggruppamento: query d'esame

Considerati i soli pazienti di Pisa, indicarne nome e cognome, e la spesa sostenuta per le visite di ciascuna specializzazione, nel triennio 2008-2010

Ancora sul raggruppamento: query d'esame

Considerati i soli pazienti di Pisa, indicarne nome e cognome, e la spesa sostenuta per le visite di ciascuna specializzazione, nel triennio 2008-2010

```
SELECT M.Specializzazione, P.Nome, P.Cognome, SUM(M.Parcella)
FROM Visita V INNER JOIN Paziente P ON V.Paziente = P.CodFiscale
    INNER JOIN Medico M ON V.Medico = M.Matricola
WHERE P.Citta = 'Pisa'
    AND YEAR(V.Data) BETWEEN 2008 AND 2010
GROUP BY M.Specializzazione;
```

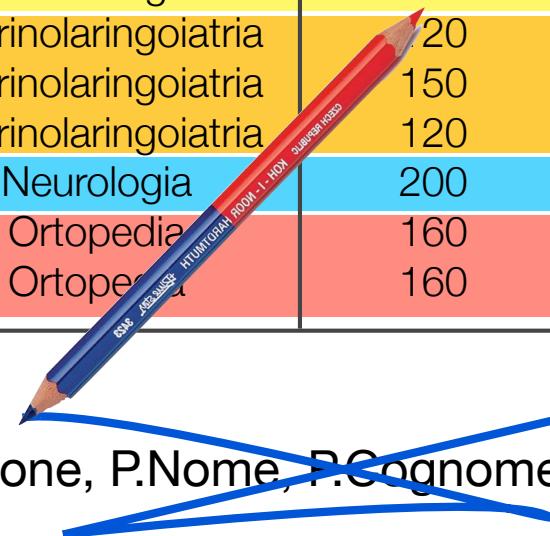
Attenzione, la tentazione è forte...

attributo di raggruppamento

P.Cognome	P.Nome	M.Specializzazione	M.Parcella	V.Data
Nutrie	Lorella	Cardiologia	180	2010-05-21
Bove	Maddalena	Cardiologia	250	2009-07-30
Bove	Maddalena	Cardiologia	180	2009-03-16
Gatti	Tommaso	Cardiologia	180	2008-10-10
Bove	Maddalena	Otorinolaringoiatria	120	2010-04-05
Ratto	Sabina	Otorinolaringoiatria	150	2009-09-13
Lepre	Edoardo	Otorinolaringoiatria	120	2008-02-28
Nutrie	Lorella	Neurologia	200	2010-11-30
Cinghiali	Maria Rita	Ortopedia	160	2010-06-05
Bove	Maddalena	Ortopedia	160	2008-08-28



SELECT M.Specializzazione, P.Nome, ~~P.Cognome~~, SUM(M.Parcella)



Raggruppamento su più attributi

Considerati i soli pazienti di Pisa, indicarne nome e cognome, e la spesa sostenuta per le visite di ciascuna specializzazione, nel triennio 2008-2010

```
SELECT M.Specializzazione, P.Nome, P.Cognome, SUM(M.Parcella)
FROM Visita V INNER JOIN Paziente P ON V.Paziente = P.CodFiscale
    INNER JOIN Medico M ON V.Medico = M.Matricola
WHERE P.Citta = 'Pisa'
    AND YEAR(V.Data) BETWEEN 2008 AND 2010
GROUP BY M.Specializzazione, P.CodFiscale;
```



la proiezione non contiene un sottoinsieme degli attributi di raggruppamento, ma P.Nome e P.Cognome possono essere proiettati perché esiste la dipendenza funzionale CodFiscale \rightarrow Nome, Cognome

Raggruppamento su più attributi: regola

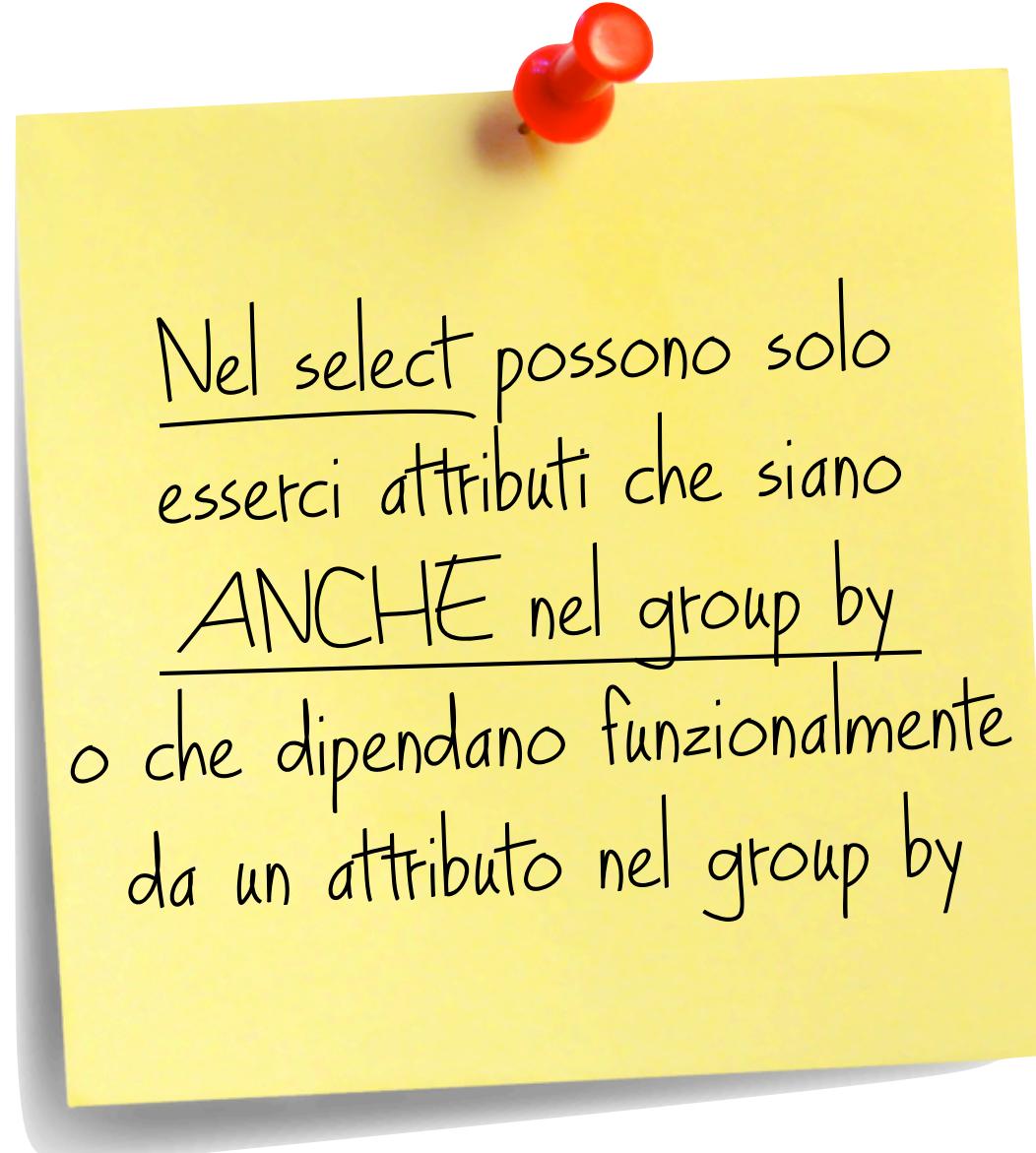
Considerati i soli pazienti di Pisa, indicarne nome e cognome, e la spesa sostenuta per le visite di ciascuna specializzazione, nel triennio 2008-2010

```
SELECT M.Specializzazione, P.Nome, P.Cognome, SUM(M.Parcella)
FROM Visita V INNER JOIN Paziente P ON V.Paziente = P.CodFiscale
    INNER JOIN Medico M ON V.Medico = M.Matricola
WHERE P.Citta = 'Pisa'
    AND YEAR(V.Data) BETWEEN 2008 AND 2010
GROUP BY M.Specializzazione, P.CodFiscale, P.Nome, P.Cognome;
```

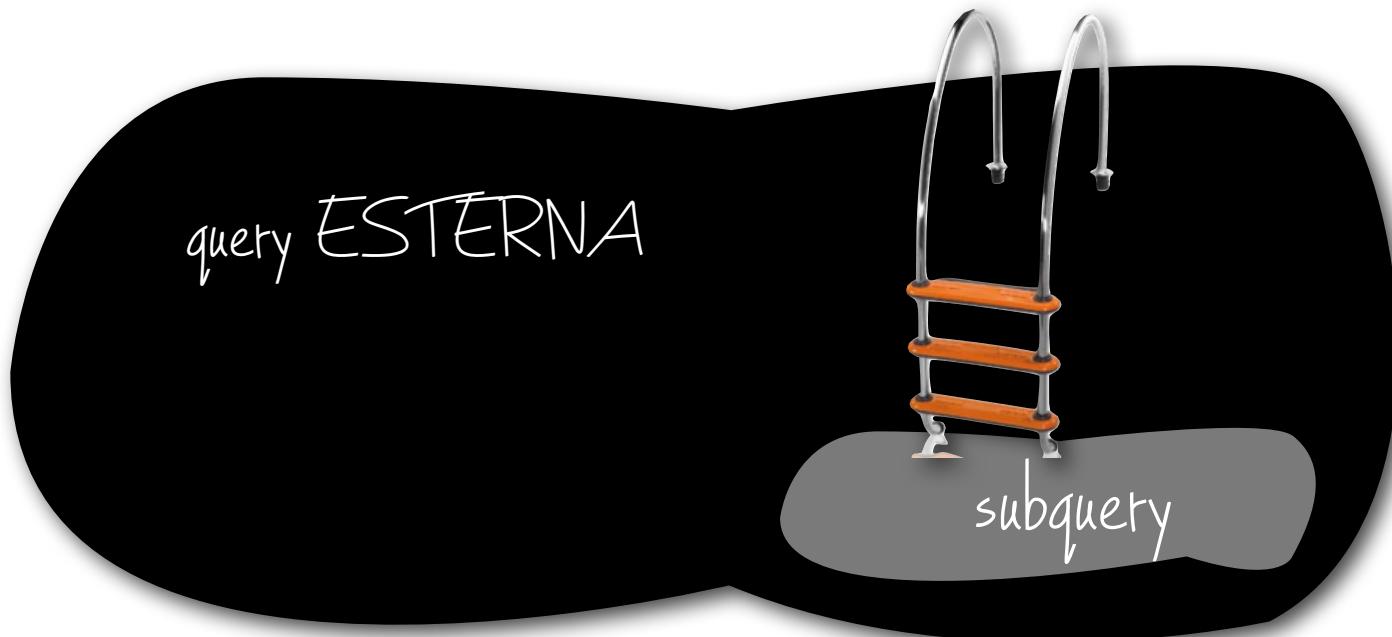


per far capire che siamo consci del fatto che stiamo proiettando coerentemente,
si arricchisce il predicato di raggruppamento con gli attributi da proiettare,
anche se dipendono funzionalmente da CodFiscale

Raggruppamento su più attributi: regola



Correlated subquery



Correlated subquery

Il loro risultato  **dipende** da ciascun record della query esterna

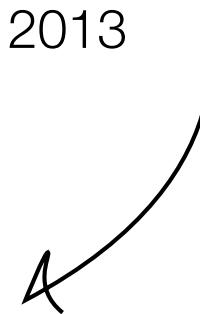
uno per ogni record della query esterna

concetto simile a una chiamata di funzione



Correlated subquery: esempio

Indicare matricola e parcella dei medici che hanno visitato **per la prima volta** almeno un paziente nel mese di Ottobre 2013



il paziente non è **MAI** stato visitato prima da quel medico

Correlated subquery: esempio

Indicare la matricola dei medici che hanno visitato **per la prima volta** almeno un paziente nel mese di Ottobre 2013



Correlated subquery: esempio

Indicare la matricola dei medici che hanno visitato **per la prima volta** almeno un paziente nel mese di Ottobre 2013

VISITA		
Medico	Paziente	Data
35512	GTTFBL	2013-10-20
29858	MNZMBT	2012-11-30
18339	CPRLND	2012-09-28
35512	GTTFBL	2014-01-19
16220	MNZMBT	2013-10-25
35512	LPRNTA	2013-03-01
18339	CPRLND	2013-10-01

è sufficiente la sola tabella Visita!

Correlated subquery: come funzionano

Indicare la matricola dei medici che hanno visitato **per la prima volta** almeno un paziente nel mese di Ottobre 2013

Visite di Ottobre 2013

Medico	Paziente	Data
35512	GTTFBL	2013-10-20
16220	MNZMBT	2013-10-25
18339	CPRLND	2013-10-01

Tutte le visite

Medico	Paziente	Data
35512	GTTFBL	2013-10-20
29858	MNZMBT	2012-11-30
18339	CPRLND	2012-09-28
35512	GTTFBL	2014-01-19
16220	MNZMBT	2013-10-25
35512	LPRNTA	2013-03-01
18339	CPRLND	2013-10-01

Risultato

Medico

Correlated subquery: come funzionano

Indicare la matricola dei medici che hanno visitato **per la prima volta** almeno un paziente nel mese di Ottobre 2013

Visite di Ottobre 2013

VISITA		
Medico	Paziente	Data
35512	GTTFBL	2013-10-20
16220	MNZMBT	2013-10-25
18339	CPRLND	2013-10-01

Tutte le visite

VISITA		
Medico	Paziente	Data
35512	GTTFBL	2013-10-20
29858	MNZMBT	2012-11-30
18339	CPRLND	2012-09-28
35512	GTTFBL	2014-01-19
16220	MNZMBT	2013-10-25
35512	LPRNTA	2013-03-01
18339	CPRLND	2013-10-01

Per ogni visita V di Ottobre 2013 (a sinistra) si cercano le visite a essa precedenti, stesso paziente, stesso medico (a destra). Se non ce ne sono significa che V.Medico ha visitato per la prima volta V.Paziente nella visita V.

Risultato

Medico
35512
16220

Correlated subquery in MySQL

Indicare la matricola dei medici che hanno visitato **per la prima volta** almeno un paziente nel mese di Ottobre 2013

```
SELECT DISTINCT V1.Medico  
FROM Visita V1  
WHERE YEAR(V1.Data) = 2013  
      AND MONTH(V1.Data) = 10  
      AND V1.Paziente NOT IN (
```

query esterna
visite di Ottobre 2013

```
SELECT V2.Paziente  
FROM Visita V2  
WHERE V2.Medico = V1.Medico  
      AND V2.Data < V1.Data
```

);

correlated subquery
eseguita per ogni record della query esterna

Subquery di conteggio con scalari

Indicare nome e cognome dei pazienti che sono stati visitati
due volte dal dottor Paolo Verdi.

Ragionamento risolutivo

Indicare nome e cognome dei pazienti che sono stati visitati dal dottor Paolo Verdi due volte.

SELECT Nome, Cognome
FROM Paziente

SELECT VPaziente
FROM Visita V INNER JOIN Medico M
ON V.Medico = M.Matricola
WHERE M.Nome='Paolo' AND M.Cognome='Verdi'

↗ exattamente due visite!

Ragionamento risolutivo

Indicare nome e cognome dei pazienti che sono stati visitati dal dottor Paolo Verdi due volte.

SELECT P.CodiceFiscale
FROM Paziente P

SELECT COUNT(*) AS VisiteDottorVerdi
FROM Visita V INNER JOIN Medico M
ON V.Medico = M.Matricola
WHERE M.Nome='Paolo' AND M.Cognome='Verdi'
AND V.Paziente=P.CodiceFiscale

WHERE 2 = (

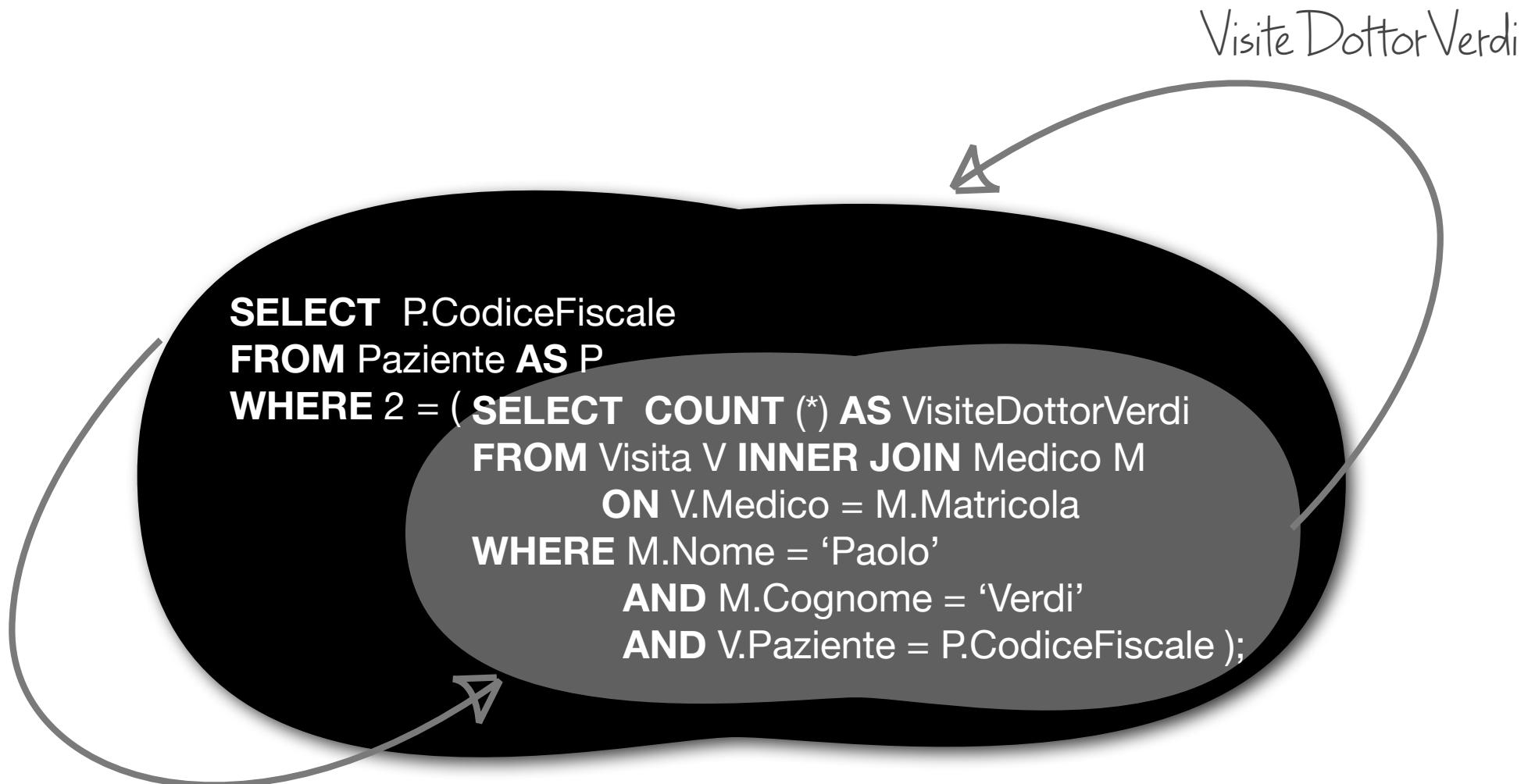
Soluzione

Indicare nome e cognome dei pazienti che sono stati visitati
due volte dal dottor Paolo Verdi.

```
SELECT P.Nome, P.Cognome
FROM Paziente P
WHERE 2 = (
    SELECT COUNT(*) AS VisiteDottorVerdi
    FROM Visita V
        INNER JOIN Medico M ON V.Medico = M.Matricola
    WHERE M.Nome = 'Paolo'
        AND M.Cognome = 'Verdi'
        AND V.Paziente = P.CodiceFiscale
);
```

considerato un paziente P nella query esterna si impone che il numero di volte che è stato visitato dal dottor Paolo Verdi sia esattamente 2

Correlated subquery: processazione



P.CodiceFiscale

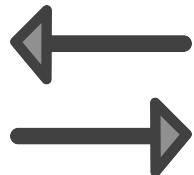
NB: un giro per ogni paziente

Costrutto EXISTS



Costrutto EXISTS

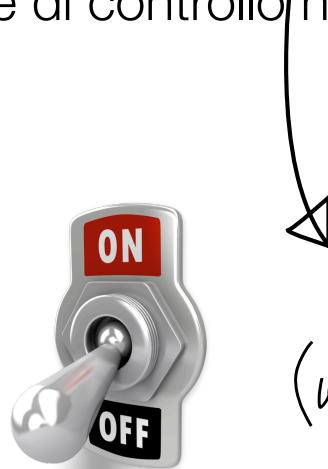
Permette di verificare che il result set di una correlated subquery contenga
almeno un record



la sua negazione controlla che il result set sia vuoto

EXISTS: esempio

Una visita di controllo è una visita in cui un medico visita un paziente **già visitato precedentemente** almeno una volta. Indicare l'incasso ottenuto mediante le visite di controllo nel mese di Gennaio 2016



non interessa sapere quante volte
(una visita precedente o c'è o non c'è)

EXISTS: esempio (cont.)

Una visita di controllo è una visita in cui un medico visita un paziente **già visitato precedentemente** almeno una volta. Indicare medico, paziente e data delle visite di controllo del mese di Gennaio 2016

```
SELECT V1.Medico, V1.Paziente, V1.Data  
FROM Visita V1  
WHERE MONTH(V1.Data) = 1  
      AND YEAR(V1.Data) = 2016  
      AND EXISTS  
(  
    SELECT *  
    FROM Visita V2  
    WHERE V2.Medico = V1.Medico  
          AND V2.Paziente = V1.Paziente  
          AND V2.Data < V1.Data  
);
```



i record del risultato hanno almeno un record nel result set della subquery

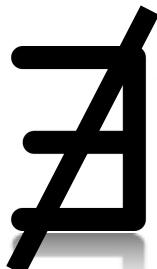
NOT EXISTS (soluzione slide 57)

Indicare la matricola dei medici che hanno visitato **per la prima volta** almeno un paziente nel mese di Ottobre 2013

```
SELECT DISTINCT V1.Medico  
FROM Visita V1  
WHERE YEAR(V1.Data) = 2013  
      AND MONTH(V1.Data) = 10  
      AND NOT EXISTS (
```

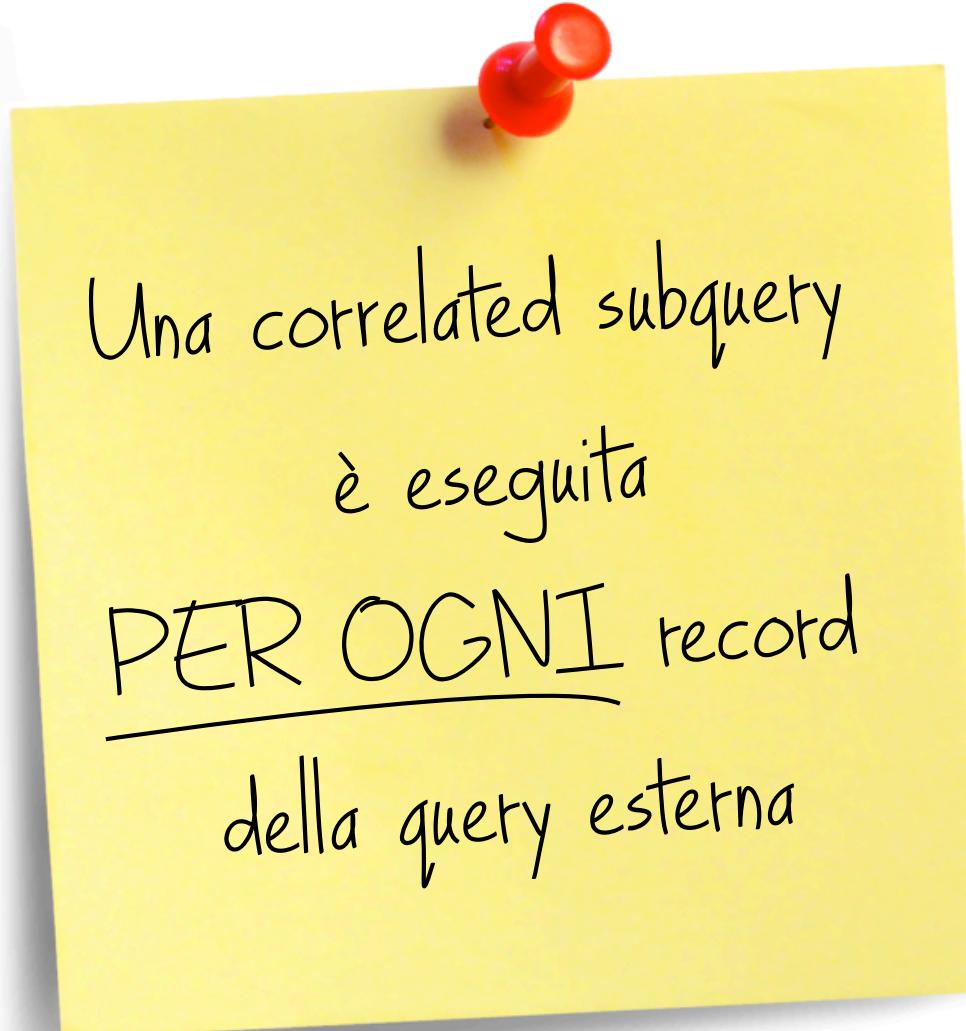
```
        SELECT *  
        FROM Visita V2  
        WHERE V2.Medico = V1.Medico  
              AND V2.Paziente = V1.Paziente  
              AND V2.Data < V1.Data
```

```
);
```



i record del risultato hanno il result set della subquery vuoto

Keep in mind



Correlated subquery nel SELECT

Una correlated subquery può essere inserita **anche nel SELECT** per calcolare un valore da inserire, come attributo, nel risultato



deve essere SCALARE!!!

Esempio

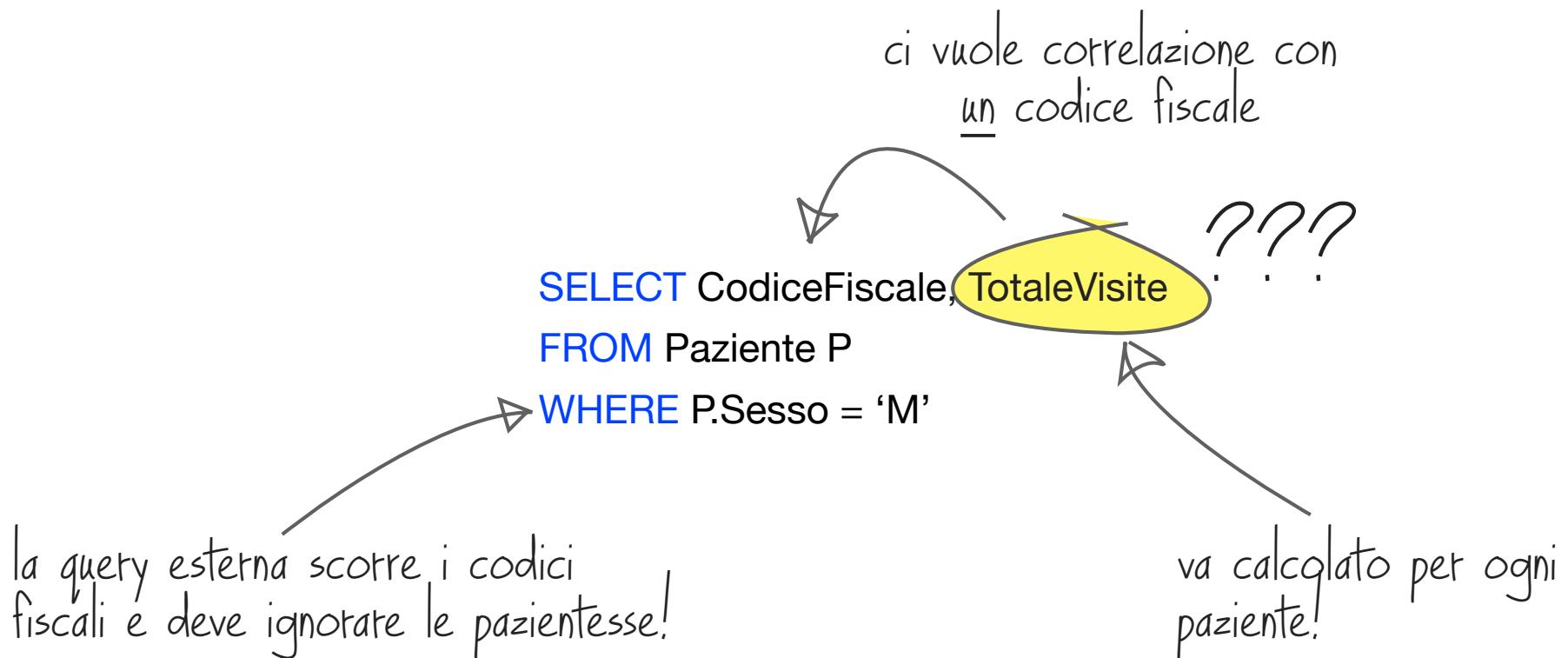
Considerato ciascun paziente di sesso maschile, indicarne il codice fiscale
e il numero di visite effettuate



è un'informazione da proiettare, non una condizione: va nel select!

Esempio

Considerato ciascun paziente di sesso maschile, indicarne il nome e
il numero di visite effettuate



Esempio

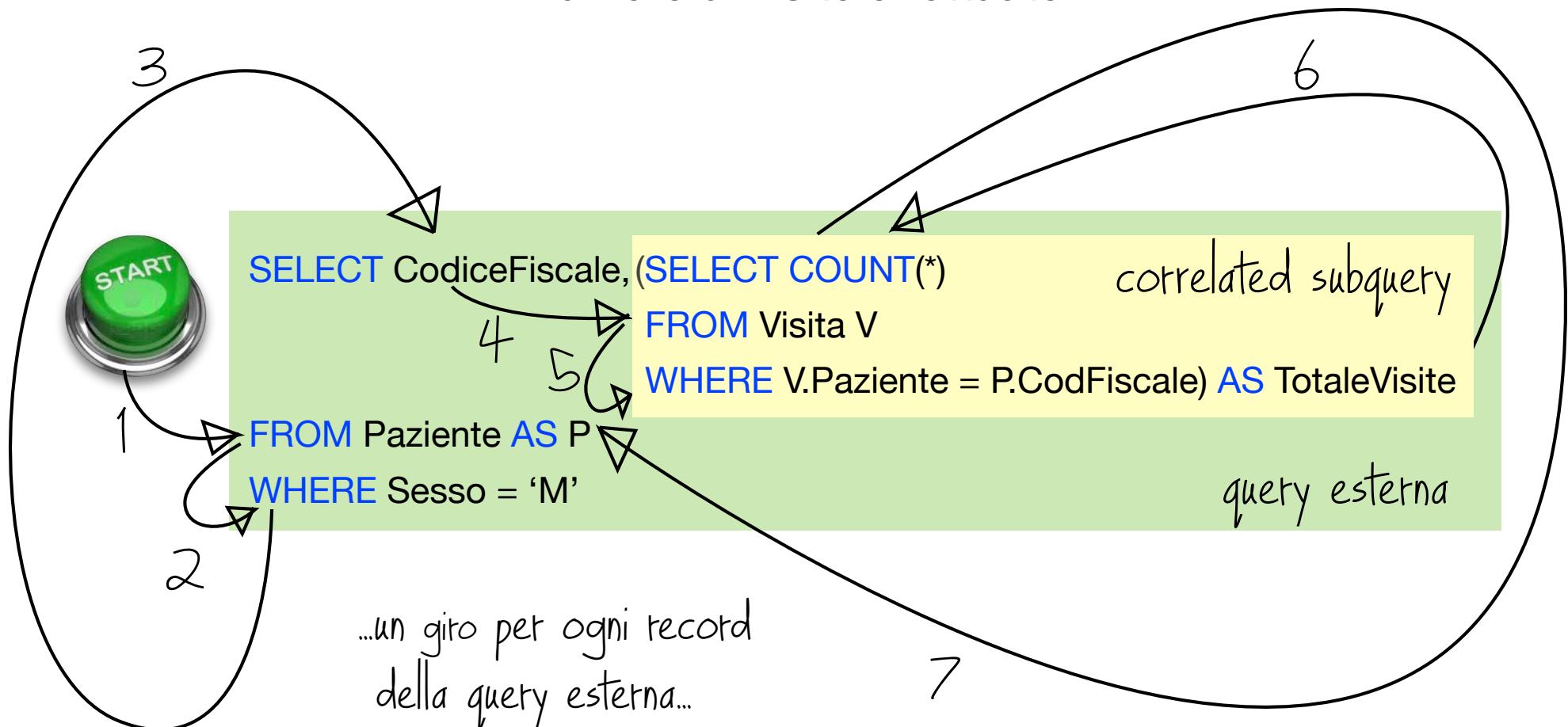
Considerato ciascun paziente di sesso maschile, indicarne il nome e
il numero di visite effettuate

```
SELECT CodiceFiscale, (SELECT COUNT(*)          correlated subquery
                        FROM Visita V
                        WHERE V.Paziente = P.CodFiscale) AS TotaleVisite
  FROM Paziente AS P
 WHERE Sesso = 'M';
```

query esterna

Correlated subquery nel SELECT: esecuzione

Considerato ciascun paziente di sesso maschile, indicarne il nome e
il numero di visite effettuate



Espressioni nel SELECT

Permettono di eseguire **operazioni matematiche** su un attributo numerico contenuto nella select list



possono usare operatori matematici semplici, oppure funzioni

Espressioni nel SELECT: esempio

In visita di controllo un medico visita un paziente che ha già visitato almeno una volta.

Indicare l'incasso ottenuto mediante le visite di controllo nel mese di Gennaio 2016.

Per una visita di controllo, il paziente paga **un terzo della parcella del medico**

```
SELECT SUM( M.Parcella/3 )
FROM Visita V1
    INNER JOIN
        Medico M ON V1.Medico = M.Matricola
WHERE MONTH(V1.Data) = 1
    AND YEAR(V1.Data) = 2016
    AND EXISTS
(
    SELECT *
    FROM Visita V2
    WHERE V2.Medico = V1.Medico
        AND V2.Paziente = V1.Paziente
        AND V2.Data < V1.Data
);
```

per ogni visita, divide la parcella per 3
e incrementa la somma con tale valore

View



View

scomporre una query complessa mascherare parti del database

Definiscono **risultati parziali** o **parti** di informazioni presenti nelle tabelle:
si creano mediante **CREATE OR REPLACE VIEW**

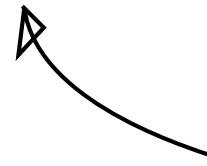


How Does it Work

Creando una vista, il dbms memorizza (senza eseguirlo) il codice di una query, a cui l'utente assegna un identificatore (nome della view). Tale identificatore è inserito poi in altre query (wrapper), nel costrutto from. Durante la processazione di una query contenente una view, il dbms esegue il codice memorizzato ad essa associato, e ne produce il result set (tabella). Questa tabella è quindi utilizzata dalla query wrapper.

View: esempio

vuol dire che si sa già che ne esiste uno solo

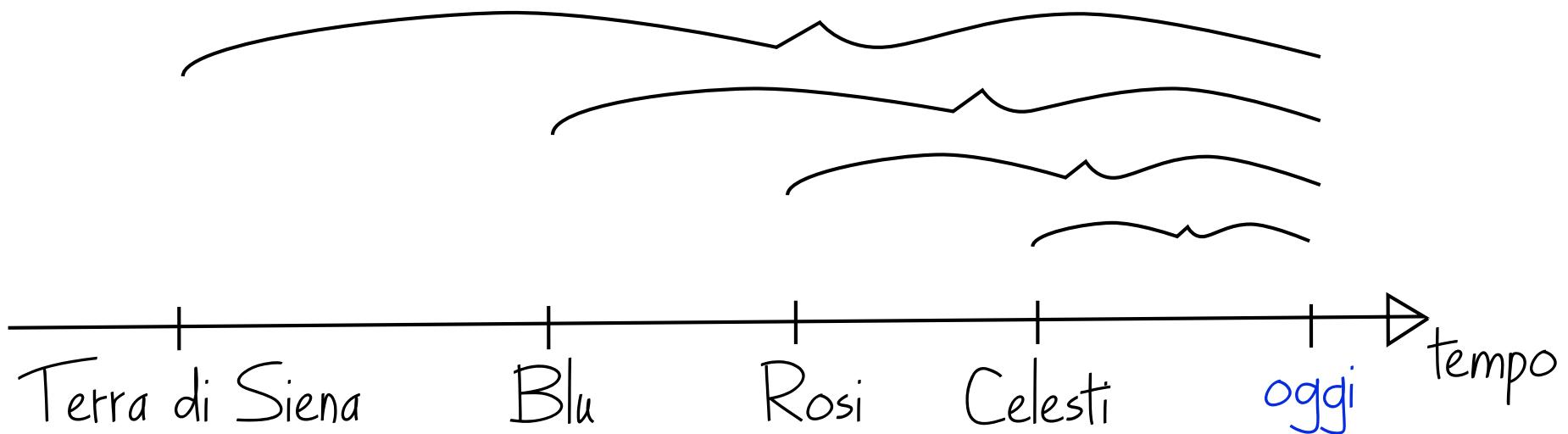


Indicare il numero di pazienti visitati dal medico, supposto unico, che ha iniziato a visitare **da meno tempo** rispetto a tutti gli altri medici.

Schemino

Indicare il numero di pazienti visitati dal medico, supposto unico, che ha iniziato a visitare **da meno tempo** rispetto a tutti gli altri medici.

Prima visita di ogni medico sulla retta temporale



Scomposizione del problema

Indicare il numero di pazienti visitati dal medico, supposto unico, che ha iniziato a visitare **da meno tempo** rispetto a tutti gli altri medici.

- A) individuare la data d_i^{first} della 1^a visita di ogni medico i , cioè $\min_i d_i$. $\forall i$
- B) selezionate il medico i tale che $d_i^{\text{first}} = \max_i d_i^{\text{first}}$

Soluzione

Indicare il numero di pazienti visitati dal medico, supposto unico, che ha iniziato a visitare **da meno tempo** rispetto a tutti gli altri medici.

A

view

```
CREATE OR REPLACE VIEW PrimeVisite AS  
SELECT V1.Medico, V1.Data AS DataPrimaVisita  
FROM Visita V1  
WHERE V1.Data =  
(SELECT MIN(V2.Data)  
FROM Visita V2  
WHERE V2.Medico = V1.Medico);
```



serve il punto e virgola

B

query

```
SELECT COUNT(DISTINCT Paziente) AS PazientiVisitati  
FROM Visita  
WHERE Medico = (SELECT Medico  
FROM PrimeVisite  
WHERE DataPrimaVisita = (SELECT MAX(DataPrimaVisita)  
FROM PrimeVisite))
```

utilizzo della view

Query multi-view

Indicare tutte le specializzazioni il cui **numero di visite** è superiore
alla **media delle visite** fra tutte le specializzazioni

Scomposizione

Indicare tutte le specializzazioni il cui **numero di visite** è superiore
alla **media delle visite** fra tutte le specializzazioni

A) ottenere il numero di visite V_i di ogni specializzazione i

B) ottenere il numero di visite medio \bar{V} fra tutte le specializzazioni

C) selezionare le specializzazioni i tali che $V_i > \bar{V}$

Soluzione

Indicare tutte le specializzazioni il cui **numero di visite** è superiore
alla **media delle visite** fra tutte le specializzazioni

view 1

```
CREATE OR REPLACE VIEW TotaliVisite AS
SELECT M.Specializzazione,
       COUNT(*) AS QuanteVisite
  FROM Medico M
 GROUP BY M.Specializzazione;
```

view 2

```
CREATE OR REPLACE VIEW MediaVisite AS
SELECT AVG(QuanteVisite) AS MediaGlobaleVisite
  FROM TotaliVisite;
```

query

```
SELECT DISTINCT Specializzazione
  FROM TotaliVisite TV
 INNER JOIN
MediaVisite MV ON TV.QuanteVisite > MV.MediaGlobaleVisite;
```

Avvisi importanti

Risolvete gli esercizi che trovate al termine di questo pacchetto di slide **dopo aver studiato gli argomenti della lezione.**

Come di consueto, riguardate e provate e risolvere anche gli **esercizi presentati come esempi** nelle slide.

Prossima lezione:

- **tutoring 3** giovedì **19 Aprile**, aula F9, dalle ore 10:30 alle ore 12:30.
Risolveremo gli esercizi assegnati per casa e chiarirò i dubbi sorti durante lo studio o lo svolgimento degli esercizi.

Esercizi per casa

1. Indicare le specializzazioni che hanno solo medici della stessa città. [Risolvere con e senza subquery].
2. Considerando pazienti della stessa città, indicare il numero di medici di città diversa dalla loro, dai quali sono stati visitati.
3. Indicare la specializzazione più redditizia per la clinica, e il medico che con le sue visite ha contribuito maggiormente agli incassi realizzati da tale specializzazione, nel corso degli ultimi dieci anni. In caso di pari merito, restituire tutti gli ex aequo.
4. Indicare la specializzazione avente meno medici di tutte le altre, e quanti medici ha. In caso di pari merito, restituire tutti gli ex aequo. [Risolvere con e senza subquery].
5. Considerate le sole visite otorinolaringoiatriche, scrivere una query che restituisca il numero di pazienti, ad oggi maggiorenni, che sono stati visitati solo da otorini di Firenze durante il primo trimestre del 2015.