



Interrogazioni nidificate

Operazione di divisione

Operazione di divisione (n.1)

- Trovare il codice dei fornitori che forniscono *tutti* i prodotti

Divisione: definizione

$$R = A / B$$

- La divisione della relazione A per la relazione B genera una relazione R
 - avente come schema *schema(A) - schema(B)*
 - contenente tutte le tuple di A tali che per ogni tupla (Y:y) presente in B esiste una tupla (X:x, Y:y) in A
- La divisione *non gode* né della proprietà commutativa, né della proprietà associativa

Divisione: esempio

➤ *Trovare gli studenti che hanno superato l'esame di **tutti** i corsi del primo anno*

EsamiSuperati

<u>MatrStudente</u>	<u>CodCorso</u>
S1	C1
S1	C2
S1	C3
S1	C4
S1	C5
S1	C6
S2	C1
S2	C2
S3	C2
S4	C2
S4	C4
S4	C5

CorsiPrimoAnno

<u>CodCorso</u>
...
...
...
...

Divisione: esempio (n. 1)

EsamiSuperati

<u>MatrStudente</u>	<u>CodCorso</u>
S1	C1
S1	C2
S1	C3
S1	C4
S1	C5
S1	C6
S2	C1
S2	C2
S3	C2
S4	C2
S4	C4
S4	C5

CorsiPrimoAnno

<u>CodCorso</u>
C1

Divisione: esempio (n. 1)

EsamiSuperati

<u>MatrStudente</u>	<u>CodCorso</u>
<i>S1</i>	<i>C1</i>
S1	C2
S1	C3
S1	C4
S1	C5
S1	C6
<i>S2</i>	<i>C1</i>
S2	C2
S3	C2
S4	C2
S4	C4
S4	C5

CorsiPrimoAnno

<u>CodCorso</u>
C1

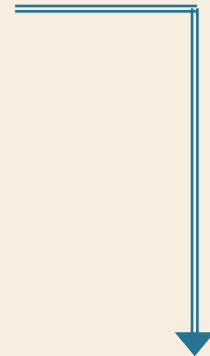
Divisione: esempio

EsamiSuperati

<u>MatrStudente</u>	<u>CodCorso</u>
<i>S1</i>	<i>C1</i>
S1	C2
S1	C3
S1	C4
S1	C5
S1	C6
<i>S2</i>	<i>C1</i>
S2	C2
S3	C2
S4	C2
S4	C4
S4	C5

CorsiPrimoAnno

<u>CodCorso</u>
C1



R

<u>MatrStudente</u>
S1
S2

Divisione: esempio (n. 2)

EsamiSuperati

<u>MatrStudente</u>	<u>CodCorso</u>
S1	C1
S1	C2
S1	C3
S1	C4
S1	C5
S1	C6
S2	C1
S2	C2
S3	C2
S4	C2
S4	C4
S4	C5

CorsiPrimoAnno

<u>CodCorso</u>
C2
C4

Divisione: esempio (n. 2)

EsamiSuperati

<u>MatrStudente</u>	<u>CodCorso</u>
S1	C1
<i>S1</i>	<i>C2</i>
S1	C3
<i>S1</i>	<i>C4</i>
S1	C5
S1	C6
S2	C1
S2	C2
S3	C2
<i>S4</i>	<i>C2</i>
<i>S4</i>	<i>C4</i>
S4	C5

CorsiPrimoAnno

<u>CodCorso</u>
C2
C4

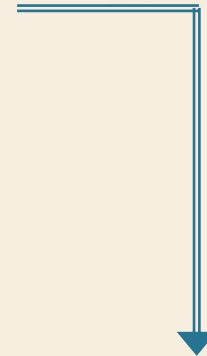
Divisione: esempio (n. 2)

EsamiSuperati

<u>MatrStudente</u>	<u>CodCorso</u>
S1	C1
<i>S1</i>	<i>C2</i>
S1	C3
<i>S1</i>	<i>C4</i>
S1	C5
S1	C6
S2	C1
S2	C2
S3	C2
<i>S4</i>	<i>C2</i>
<i>S4</i>	<i>C4</i>
S4	C5

CorsiPrimoAnno

<u>CodCorso</u>
C2
C4



R

<u>MatrStudente</u>
S1
S4

Divisione: esempio (n. 3)

EsamiSuperati

<u>MatrStudente</u>	<u>CodCorso</u>
S1	C1
S1	C2
S1	C3
S1	C4
S1	C5
S1	C6
S2	C1
S2	C2
S3	C2
S4	C2
S4	C4
S4	C5

CorsiPrimoAnno

<u>CodCorso</u>
C1
C2
C3
C4
C5
C6

Divisione: esempio (n. 3)

EsamiSuperati

<u>MatrStudente</u>	<u>CodCorso</u>
<i>S1</i>	<i>C1</i>
<i>S1</i>	<i>C2</i>
<i>S1</i>	<i>C3</i>
<i>S1</i>	<i>C4</i>
<i>S1</i>	<i>C5</i>
<i>S1</i>	<i>C6</i>
S2	C1
S2	C2
S3	C2
S4	C2
S4	C4
S4	C5

CorsiPrimoAnno

<u>CodCorso</u>
C1
C2
C3
C4
C5
C6

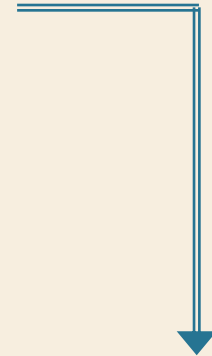
Divisione: esempio (n. 3)

EsamiSuperati

<u>MatrStudente</u>	<u>CodCorso</u>
<i>S1</i>	<i>C1</i>
<i>S1</i>	<i>C2</i>
<i>S1</i>	<i>C3</i>
<i>S1</i>	<i>C4</i>
<i>S1</i>	<i>C5</i>
<i>S1</i>	<i>C6</i>
S2	C1
S2	C2
S3	C2
S4	C2
S4	C4
S4	C5

CorsiPrimoAnno

<u>CodCorso</u>
C1
C2
C3
C4
C5
C6



R

<u>MatrStudente</u>
S1

Divisione in SQL (n.1)

➤ Trovare il codice dei fornitori che forniscono *tutti* i prodotti

➤ Osservazione

- tutti i prodotti che possono essere forniti sono contenuti nella tabella P

Divisione in SQL (n.1)

➤ Trovare il codice dei fornitori che forniscono *tutti* i prodotti

➤ Osservazione

- tutti i prodotti che possono essere forniti sono contenuti nella tabella P



- un fornitore fornisce tutti i prodotti se fornisce un numero di prodotti diversi pari alla cardinalità di P

Divisione in SQL (n.1)

➤ Trovare il codice dei fornitori che forniscono *tutti* i prodotti

Divisione in SQL (n.1)

➤ Trovare il codice dei fornitori che forniscono *tutti* i prodotti

```
SELECT COUNT(*)  
FROM P
```

Divisione in SQL (n.1)

➤ Trovare il codice dei fornitori che forniscono *tutti* i prodotti

```
SELECT COUNT(*)  
FROM P
```

} *Numero
totale di
prodotti*

Divisione in SQL (n.1)

➤ Trovare il codice dei fornitori che forniscono *tutti* i prodotti

```
SELECT CodF  
FROM FP  
GROUP BY CodF
```

...

```
(SELECT COUNT(*)  
FROM P)
```

Divisione in SQL (n.1)

➤ Trovare il codice dei fornitori che forniscono *tutti* i prodotti

```
SELECT CodF
FROM FP
GROUP BY CodF
HAVING COUNT(*)=(SELECT COUNT(*)
                  FROM P);
```

Divisione in SQL (n.2)

- Trovare il codice dei fornitori che forniscono almeno *tutti* i prodotti forniti dal fornitore F2

Divisione in SQL: procedimento (n.2)

- Trovare il codice dei fornitori che forniscono almeno *tutti* i prodotti forniti dal fornitore F2
- Si esegue
 - il conteggio del numero di prodotti forniti da F2

Divisione in SQL: procedimento (n.2)

- Trovare il codice dei fornitori che forniscono almeno *tutti* i prodotti forniti dal fornitore F2
- Si esegue
 - il conteggio del numero di prodotti forniti da F2
 - il conteggio del numero di prodotti forniti da un fornitore arbitrario e anche da F2

Divisione in SQL: procedimento (n.2)

- Trovare il codice dei fornitori che forniscono almeno *tutti* i prodotti forniti dal fornitore F2
- Si esegue
 - il conteggio del numero di prodotti forniti da F2
 - il conteggio del numero di prodotti forniti da un fornitore arbitrario e anche da F2
- I due conteggi devono essere uguali

Divisione in SQL (n.2)

- Trovare il codice dei fornitori che forniscono almeno *tutti* i prodotti forniti dal fornitore F2

Divisione in SQL (n.2)

- Trovare il codice dei fornitori che forniscono almeno *tutti* i prodotti forniti dal fornitore F2

```
SELECT COUNT(*)  
FROM FP  
WHERE CodF='F2'
```

Divisione in SQL (n.2)

- Trovare il codice dei fornitori che forniscono almeno *tutti* i prodotti forniti dal fornitore F2

```
SELECT COUNT(*)  
FROM FP  
WHERE CodF='F2'
```

*Numero
di prodotti
forniti da F2*

Divisione in SQL (n.2)

➤ Trovare il codice dei fornitori che forniscono almeno *tutti* i prodotti forniti dal fornitore F2

```
SELECT CodF
FROM FP
WHERE CodP IN (SELECT CodP
                FROM FP
                WHERE CodF='F2')
GROUP BY CodF
... (SELECT COUNT(*)
      FROM FP
      WHERE CodF='F2')
```

Divisione in SQL (n.2)

➤ Trovare il codice dei fornitori che forniscono almeno *tutti* i prodotti forniti dal fornitore F2

```
SELECT CodF
FROM FP
WHERE CodP IN (SELECT CodP
                FROM FP
                WHERE CodF='F2')
GROUP BY CodF
HAVING COUNT(*)=(SELECT COUNT(*)
                  FROM FP
                  WHERE CodF='F2');
```



Interrogazioni nidificate

Table functions

Schema di esempio

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

CORSO (CodC, NomeC)

Calcolo di aggregati a due livelli (n.1)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

- Trovare la media massima (conseguita da uno studente)

Calcolo di aggregati a due livelli (n.1)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

- Trovare la media massima (conseguita da uno studente)
- Risoluzione in 2 passi
 - trovare la media per ogni studente
 - trovare il valore massimo della media

Calcolo di aggregati a due livelli (n.1)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

➤ Trovare la media massima (conseguita da uno studente)

- passo 1: media per ogni studente

```
SELECT Matricola, AVG(Voto) AS MediaStudenti  
FROM ESAME-SUPERATO  
GROUP BY Matricola
```

Calcolo di aggregati a due livelli (n.1)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

➤ Trovare la media massima (conseguita da uno studente)

- passo 1: media per ogni studente

```
(SELECT Matricola, AVG(Voto) AS MediaStudenti  
FROM ESAME-SUPERATO  
GROUP BY Matricola) AS MEDIE
```

Calcolo di aggregati a due livelli (n.1)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

➤ Trovare la media massima (conseguita da uno studente)

- passo 2: valore massimo della media

```
SELECT ...
```

```
FROM (SELECT Matricola, AVG(Voto) AS MediaStudenti
```

```
FROM ESAME-SUPERATO
```

```
GROUP BY Matricola) AS MEDIE
```

Calcolo di aggregati a due livelli (n.1)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

➤ Trovare la media massima (conseguita da uno studente)

- passo 2: valore massimo della media

```
SELECT MAX(MediaStudenti)
```

```
FROM (SELECT Matricola, AVG(Voto) AS MediaStudenti
```

```
FROM ESAME-SUPERATO
```

```
GROUP BY Matricola) AS MEDIE;
```

Table functions (n.1)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

➤ Trovare la media massima (conseguita da uno studente)

```
SELECT MAX(MediaStudenti)
```

```
FROM (SELECT Matricola, AVG(Voto) AS MediaStudenti  
FROM ESAME-SUPERATO  
GROUP BY Matricola) AS MEDIE;
```

Table function

Table function

- Definisce una tabella temporanea che può essere utilizzata per ulteriori operazioni di calcolo
- La table function
 - ha la struttura di una **SELECT**
 - è definita all'interno di una clausola **FROM**
 - può essere referenziata come una normale tabella
- La table function permette di
 - calcolare più livelli di aggregazione
 - formulare in modo equivalente le interrogazioni che richiedono la correlazione

Table functions (n.2)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

- Per ogni anno di iscrizione, trovare la media massima (conseguita da uno studente)

Table functions (n.2)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

- Per ogni anno di iscrizione, trovare la media massima (conseguita da uno studente)
- Risoluzione in 2 passi
 - trovare la media per ogni studente
 - raggruppare gli studenti per anno di iscrizione e calcolare la media massima

Table functions (n.2)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

➤ Per ogni anno di iscrizione, trovare la media massima (conseguita da uno studente)

- passo 1

```
(SELECT Matricola, AVG(Voto) AS MediaStudente  
FROM ESAME-SUPERATO  
GROUP BY Matricola) AS MEDIE
```

Table functions (n.2)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

➤ Per ogni anno di iscrizione, trovare la media massima (conseguita da uno studente)

- passo 2

```
SELECT ...
```

```
FROM STUDENTE,
```

```
(SELECT Matricola, AVG(Voto) AS MediaStudente
```

```
FROM ESAME-SUPERATO
```

```
GROUP BY Matricola) AS MEDIE
```

```
WHERE STUDENTE.Matricola=MEDIE.Matricola
```

```
...
```

Table function

Table functions (n.2)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

➤ Per ogni anno di iscrizione, trovare la media massima (conseguita da uno studente)

- passo 2

```
SELECT ...
```

```
FROM STUDENTE,
```

```
(SELECT Matricola, AVG(Voto) AS MediaStudente  
FROM ESAME-SUPERATO
```

```
GROUP BY Matricola) AS MEDIE
```

```
WHERE STUDENTE.Matricola=MEDIE.Matricola
```

```
...
```

Table functions (n.2)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

➤ Per ogni anno di iscrizione, trovare la media massima (conseguita da uno studente)

- passo 2

```
SELECT ...
```

```
FROM STUDENTE,
```

```
    (SELECT Matricola, AVG(Voto) AS MediaStudente
```

```
    FROM ESAME-SUPERATO
```

```
    GROUP BY Matricola) AS MEDIE
```

```
WHERE STUDENTE.Matricola=MEDIE.Matricola
```

```
GROUP BY AnnoIscrizione
```

Table functions (n.2)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

➤ Per ogni anno di iscrizione, trovare la media massima (conseguita da uno studente)

- passo 2

```
SELECT AnnoIscrizione, MAX(MediaStudente)
```

```
FROM STUDENTE,
```

```
    (SELECT Matricola, AVG(Voto) AS MediaStudente
```

```
    FROM ESAME-SUPERATO
```

```
    GROUP BY Matricola) AS MEDIE
```

```
WHERE STUDENTE.Matricola=MEDIE.Matricola
```

```
GROUP BY AnnoIscrizione;
```

Correlazione tra interrogazioni (n.1)

➤ Per ogni prodotto, trovare il codice del fornitore che ne fornisce la quantità massima

```
SELECT CodP, CodF
FROM FP AS FPX
WHERE Qta = (SELECT MAX(Qta)
             FROM FP AS FPY
             WHERE FPY.CodP=FPX.CodP);
```

Condizione di correlazione

Correlazione tra interrogazioni (n.1)

- Per ogni prodotto, trovare il codice del fornitore che ne fornisce la quantità massima

Correlazione tra interrogazioni (n.1)

- Per ogni prodotto, trovare il codice del fornitore che ne fornisce la quantità massima

```
(SELECT CodP, MAX(Qta) AS QtaMax  
FROM FP  
GROUP BY CodP) AS MAXQ
```

Correlazione tra interrogazioni (n.1)

- Per ogni prodotto, trovare il codice del fornitore che ne fornisce la quantità massima

```
SELECT CodP, CodF
SELECT FP.CodP, CodF
FROM FP, (SELECT CodP, MAX(Qta) AS QtaMax
          FROM FP
          GROUP BY CodP) AS MAXQ
WHERE FP.CodP = MAXQ.CodP
AND FP.Qta = MAXQ.QtaMax;
```

Correlazione tra interrogazioni (n.2)

VIAGGIO (CodV, LuogoPartenza, LuogoArrivo,
OraPartenza, OraArrivo)

- Trovare il codice dei viaggi che hanno una durata inferiore alla durata media dei viaggi sullo stesso percorso (caratterizzato dallo stesso luogo di partenza e di arrivo)

```
SELECT CodV
FROM VIAGGIO AS VA
WHERE OraArrivo-OraPartenza <
      (SELECT AVG(OraArrivo-OraPartenza)
      FROM VIAGGIO AS VB      Condizioni di correlazione
      WHERE VB.LuogoPartenza=VA.LuogoPartenza
            AND VB.LuogoArrivo=VA.LuogoArrivo);
```

Correlazione tra interrogazioni (n.2)

VIAGGIO (CodV, LuogoPartenza, LuogoArrivo,
OraPartenza, OraArrivo)

- Trovare il codice dei viaggi che hanno una durata inferiore alla durata media dei viaggi sullo stesso percorso (caratterizzato dallo stesso luogo di partenza e di arrivo)

Correlazione tra interrogazioni (n.2)

VIAGGIO (CodV, LuogoPartenza, LuogoArrivo,
OraPartenza, OraArrivo)

- Trovare il codice dei viaggi che hanno una durata inferiore alla durata media dei viaggi sullo stesso percorso (caratterizzato dallo stesso luogo di partenza e di arrivo)

```
(SELECT LuogoPartenza, LuogoArrivo,  
        AVG(OraArrivo-OraPartenza) AS DurataMedia  
FROM VIAGGIO  
GROUP BY LuogoPartenza, LuogoArrivo) AS MEDIE
```

Correlazione tra interrogazioni (n.2)

VIAGGIO (CodV, LuogoPartenza, LuogoArrivo,
OraPartenza, OraArrivo)

- Trovare il codice dei viaggi che hanno una durata inferiore alla durata media dei viaggi sullo stesso percorso (caratterizzato dallo stesso luogo di partenza e di arrivo)

```
SELECT CodV
FROM VIAGGIO AS V,
     (SELECT LuogoPartenza, LuogoArrivo,
            AVG(OraArrivo-OraPartenza) AS DurataMedia
     FROM VIAGGIO
     GROUP BY LuogoPartenza, LuogoArrivo) AS MEDIE
WHERE V.LuogoPartenza = MEDIE.LuogoPartenza
AND V.LuogoArrivo = MEDIE.LuogoArrivo
AND V.OraArrivo-V.OraPartenza < MEDIE.DurataMedia;
```