

# Data Bases II

## Frammentazione

Michele Beretta

[michele.beretta@unibg.it](mailto:michele.beretta@unibg.it)



## Tema d'esame 30/10/2014

Si ha una base di dati distribuita su 3 sedi "Dalmine", "Caniana" e "Salvecchio". Lo schema globale è il seguente:

```
Corso(CodCorso, Sede, Titolo, Docente, Cfu)
PianoStudi(Matr, CodCorso)
Studente(Matr, Sede, Nome, CorsoLaurea)
```

Le tabelle Corso e Studente sono frammentate in base all'attributo Sede. Gli studenti possono inserire nel loro piano degli studi insegnamenti di altre sedi.

La tabella PianoStudi è frammentata in base alla frammentazione di Studente.

Scrivere ai livelli di trasparenza di frammentazione e di linguaggio i comandi SQL che:

1. spostano lo studente Mario Rossi dalla sede di "Caniana" alla sede di "Dalmine";
2. cambiano la sede del corso di "Basi di dati 2" da Dalmine a Salvecchio

## Soluzione punto 1

### Trasparenza di frammentazione

```
UPDATE Studente
SET Sede = "Dalmine"
WHERE Nome = "Mario Rossi" AND Sede = "Salvecchio"
```

### Trasparenza di linguaggio

```
INSERT INTO StudenteD @ Dalmine
SELECT Matr, "Dalmine", Nome, CorsoLaurea
FROM StudenteC @ Caniana
WHERE Nome = "Mario Rossi";
```

```
INSERT INTO PianoStudiD @ Dalmine
SELECT * FROM PianoStudiC @ Caniana
WHERE Matr = (SELECT Matr
               FROM StudenteC @ Caniana
               WHERE Nome = "Mario Rossi");
```

```
DELETE FROM PianoStudiC @ Caniana
WHERE Matr = (SELECT Matr
              FROM StudenteC @ Caniana
              WHERE Nome = "Mario Rossi");
```

```
DELETE FROM StudenteC @ Caniana
WHERE Nome = "Mario Rossi";
```

## Soluzione punto 2

### Trasparenza di frammentazione

```
UPDATE Corso
SET Sede = "Salvecchio"
WHERE Titolo = "Basi di dati 2"
```

### Trasparenza di linguaggio

```
INSERT INTO Corsos @ Salvecchio
SELECT CodCorso, "Salvecchio", Titolo, Docente, Cfu
FROM Corsod @ Dalmine
WHERE Titolo = "Basi di dati 2";

DELETE FROM Corsod @ Dalmine
WHERE Titolo = "Basi di dati 2";
```

## Tema d'esame 08/11/2018

Si ha una base di dati distribuita su  $n$  filiali. Lo schema globale è il seguente:

```
Conto(CodConto, Cliente, Filiale, Saldo)
```

```
Operazione(CodOperazione, CodConto, Ammontare, Causale, Data)
```

```
Movimento(CodMov, CodContoOrig, CodContoDest, Ammontare, Data, Causale)
```

La tabella Conto è frammentata in base all'attributo Filiale, mentre le tabelle Operazione e Movimento sono frammentate in modo derivato. La maggior parte dei movimenti operano tra conti della stessa filiale.

Descrivere la strategia di frammentazione da adottare in questo contesto. Mostrare poi ai livelli di trasparenza di frammentazione e di linguaggio la gestione di un comando SQL che cambia da 1 a 2 la filiale del conto 12345.

## Strategia di frammentazione

In algebra logica

$$\text{Conto}_i = \sigma_{\text{Filiale}=i}(\text{Conto})$$

$$\text{Operazione}_i = \text{Operazione} \bowtie \text{Conto}_i$$

$$\text{Movimento}_i = \text{Movimento} \bowtie_{\text{CodContoOrig} = \text{CodConto}} \text{Conto}_i$$

## Query

### Trasparenza di frammentazione

```
UPDATE Conto SET Filiale = 2 WHERE CodConto = 12345
```

### Trasparenza di linguaggio

```
INSERT INTO Conto2 @ 2  
SELECT CodConto, Cliente, 2, Saldo  
FROM Conto1 @ 1  
WHERE CodConto = 12345;
```

```
INSERT INTO Operazione2 @ 2  
SELECT *  
FROM Operazione1 @ 1  
WHERE CodConto = 12345;
```

```
INSERT INTO Movimento2 @ 2  
SELECT *  
FROM Movimento1 @ 1  
WHERE CodContoOrig = 12345;
```

```
DELETE FROM Movimento1 @ 1  
WHERE CodContoOrig = 12345;
```

```
DELETE FROM Operazione1 @ 1  
WHERE CodConto = 12345;
```

```
DELETE FROM Conto1 @ 1  
WHERE CodConto = 12345;
```