

Si consideri la base di dati:

Produzione(NumeroSerie,TipoParte,Modello,Qta,Macchina)

Prelievo(NumeroSerie,Lotto,Cliente,Venditore,Ammontare)

Cliente(Nome,Citta,Indirizzo)

Venditore(Nome,Citta,Indirizzo)

Progettare la frammentazione orizzontale delle tabelle Produzione e Prelievo in base al tipo di parte (che assume quattro valori: 'Keyboard', 'Screen', 'CPU-box' e 'Cable'), prevedendo quattro stabilimenti di produzione (uno per ogni componente) disposti a Milano, Torino, Roma e Napoli, e di Cliente e Venditore su tre città: Milano, Torino, Roma.

Il DB distribuito conterrà le seguenti tabelle

ProduzioneMiKbr(NumeroSerie,TipoParte,Modello,Qta,Macchina)

ProduzioneToScr(NumeroSerie,TipoParte,Modello,Qta,Macchina)

ProduzioneRmCpu(NumeroSerie,TipoParte,Modello,Qta,Macchina)

ProduzioneNaCbl(NumeroSerie,TipoParte,Modello,Qta,Macchina)

Produzione = ProduzioneMiKbr \cup ProduzioneToScr \cup ProduzioneRmCpu \cup

ProduzioneNaCpl

PrelievoMiKbr(NumeroSerie,Lotto,Cliente,Venditore,Ammontare)

PrelievoToScr(NumeroSerie,Lotto,Cliente,Venditore,Ammontare)

PrelievoRmCpu(NumeroSerie,Lotto,Cliente,Venditore,Ammontare)

PrelievoNaCbl(NumeroSerie,Lotto,Cliente,Venditore,Ammontare)

Prelievo = PrelievoMiKbr \cup PrelievoToScr \cup PrelievoRmCpu \cup PrelievoNaCpl

ClienteMi(Nome,Citta,Indirizzo)

ClienteTo(Nome,Citta,Indirizzo)

ClienteRm(Nome,Citta,Indirizzo)

Cliente = ClienteMi \cup ClienteTo \cup

ClienteRm

VenditoreMi(Nome,Citta,Indirizzo)

VenditoreTo(Nome,Citta,Indirizzo)

VenditoreRm(Nome,Citta,Indirizzo)

Venditore = VenditoreMi \cup

VenditoreTo \cup VenditoreRm

Esprimere a livello di trasparenza di frammentazione, di allocazione e di linguaggio le interrogazioni:

Determinare la quantità totale prodotta del componente con numero di serie '77Y6878'

Trasparenza di frammentazione

```
SELECT Qta  
FROM Produzione  
WHERE NumeroSerie='77Y6878'
```

Trasparenza di allocazione

```
SELECT Qta  
FROM ProduzioneMiKbr  
WHERE NumeroSerie='77Y6878'  
IF :empty then  
(SELECT Qta  
FROM ProduzioneToScr  
WHERE NumeroSerie='77Y6878'  
UNION  
SELECT Qta  
FROM ProduzioneRmCpu  
WHERE NumeroSerie='77Y6878'  
UNION  
SELECT Qta  
FROM ProduzioneNaCbl  
WHERE NumeroSerie='77Y6878')
```

Trasparenza di linguaggio

```
SELECT Qta  
FROM ProduzioneMiKbr@Milano  
WHERE NumeroSerie='77Y6878'  
IF :empty then  
(SELECT Qta  
FROM ProduzioneToScr@Torino  
WHERE NumeroSerie='77Y6878'  
UNION  
SELECT Qta  
FROM ProduzioneRmCpu@Roma  
WHERE NumeroSerie='77Y6878'  
UNION  
SELECT Qta  
FROM ProduzioneNaCbl@Napoli  
WHERE NumeroSerie='77Y6878')
```

Determinare i clienti che hanno comprato qualche lotto dal rivenditore 'Bianchi', che ha ufficio a Roma

Trasparenza di frammentazione

```
SELECT Cliente  
FROM Prelievo  
WHERE Venditore='Bianchi'
```

Trasparenza di allocazione

```
SELECT Cliente FROM PrelievoMiKbr  
WHERE Venditore='Bianchi'
```

union

```
SELECT Cliente FROM PrelievoToScr  
WHERE Venditore='Bianchi'
```

union

```
SELECT Cliente FROM PrelievoRmCpu  
WHERE Venditore='Bianchi'
```

union

```
SELECT Cliente FROM PrelievoNaCbl  
WHERE Venditore='Bianchi'
```

Trasparenza di linguaggio

```
SELECT Cliente FROM PrelievoMiKbr@Milano  
WHERE Venditore='Bianchi'
```

union

```
SELECT Cliente FROM PrelievoToScr@Torino  
WHERE Venditore='Bianchi'
```

union

```
SELECT Cliente FROM PrelievoRmCpu@Roma  
WHERE Venditore='Bianchi'
```

union

```
SELECT Cliente FROM PrelievoNaCbl@Napoli  
WHERE Venditore='Bianchi'
```

Determinare le macchine utilizzate per la produzione dei pezzi di tipo 'Keyboard' venduti al cliente 'Rossi'

Trasparenza di frammentazione

```
SELECT Macchina
FROM Produzione P1 JOIN Prelievo P2
ON (P1.NumeroSerie=P2.NumeroSerie)
WHERE Venditore = 'Rossi' AND
TipoParte = 'Keyboard'
```

Trasparenza di allocazione

Questa volta la query risulta più semplice perché un *solo* frammento è *totalmente* dedicato a componenti di tipo 'Keyboard'

```
SELECT Macchina
FROM ProduzioneMiKbr P1
JOIN PrelievoMiKbr P2
ON (P1.NumeroSerie=P2.NumeroSerie)
WHERE Venditore = 'Rossi'
```

Trasparenza di linguaggio

Anche questa query risulta più semplice perché un *solo* frammento è *totalmente* dedicato a componenti di tipo 'Keyboard'

```
SELECT Macchina
FROM ProduzioneMiKbr@Milano P1
JOIN PrelievoMiKbr@Milano P2
ON (P1.NumeroSerie=P2.NumeroSerie)
WHERE Venditore = 'Rossi'
```

Modificare l'indirizzo del rivenditore 'Rossi', che si trasferisce da 'via Po 45' di 'Milano' a 'Viale Trastevere 150' di Roma' (si ignorino i problemi legati alle vecchie vendite)

Trasparenza di frammentazione

```
UPDATE Venditore
SET Città = 'Roma',
Indirizzo='VialeTrastevere 150'
WHERE Nome = 'Rossi'
```

Trasparenza di allocazione

Devo eseguire una delete seguito da una insert

```
DELETE FROM VenditoreMi
WHERE Nome = 'Rossi'
INSERT INTO VenditoreRm
VALUES('Rossi', 'Roma',
'Viale Trastevere 150')
```

Trasparenza di linguaggio

Devo eseguire una delete seguito da una insert

```
DELETE FROM VenditoreMi@Milano
WHERE Nome = 'Rossi'
INSERT INTO VenditoreRm@Roma
VALUES('Rossi', 'Roma',
'Viale Trastevere 150')
```

Calcolare l'ammontare totale di tutti gli ordini ricevuti a Milano, Torino, Roma e Napoli (si noti che le funzioni aggregate sono distribuibili)

Trasparenza di frammentazione

```
SELECT sum(Ammontare) as Ammontare_totale  
FROM Prelievo
```

Trasparenza di allocazione

Bisogna identificare il contributo dei vari frammenti, distribuendo i frammenti sui quattro centri di produzione.

Per effettuare la somma di somma è necessario introdurre una vista ausiliaria

Trasparenza di allocazione

```
CREATE VIEW TotaliAmm ( Somma ) AS  
SELECT sum(Ammontare)  
FROM PrelievoMiKbr  
UNION ALL  
SELECT sum(Ammontare)  
FROM PrelievoToScr  
UNION ALL  
SELECT sum(Ammontare)  
FROM PrelievoRmCpu  
UNION ALL  
SELECT sum(Ammontare)  
FROM PrelievoNaCbl  
SELECT sum(Somma) as Ammontare_Totale  
FROM TotaliAmm
```

Trasparenza di linguaggio

```
CREATE VIEW TotaliAmm ( Somma ) AS  
SELECT SUM ( Ammontare )  
FROM PrelievoMiKbr@Milano  
UNION ALL  
SELECT SUM ( Ammontare )  
FROM PrelievoToScr@Torino  
UNION ALL  
SELECT SUM ( Ammontare )  
FROM PrelievoRmCpu@Roma  
UNION ALL  
SELECT SUM ( Ammontare )  
FROM PrelievoNaCbl@Napoli  
SELECT sum(Somma) as Ammontare_Totale  
FROM TotaliAmm
```