



UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

Dipartimento di Scienze Fisiche,
Informatiche e Matematiche

Basi di Dati

Riccardo Martoglia

Corso di Laurea in Informatica

E' vietata la copia e la riproduzione dei contenuti e immagini in qualsiasi forma. E' inoltre vietata la redistribuzione e la pubblicazione dei contenuti e immagini non autorizzata espressamente dall'autore o dall'Università di Modena e Reggio Emilia.

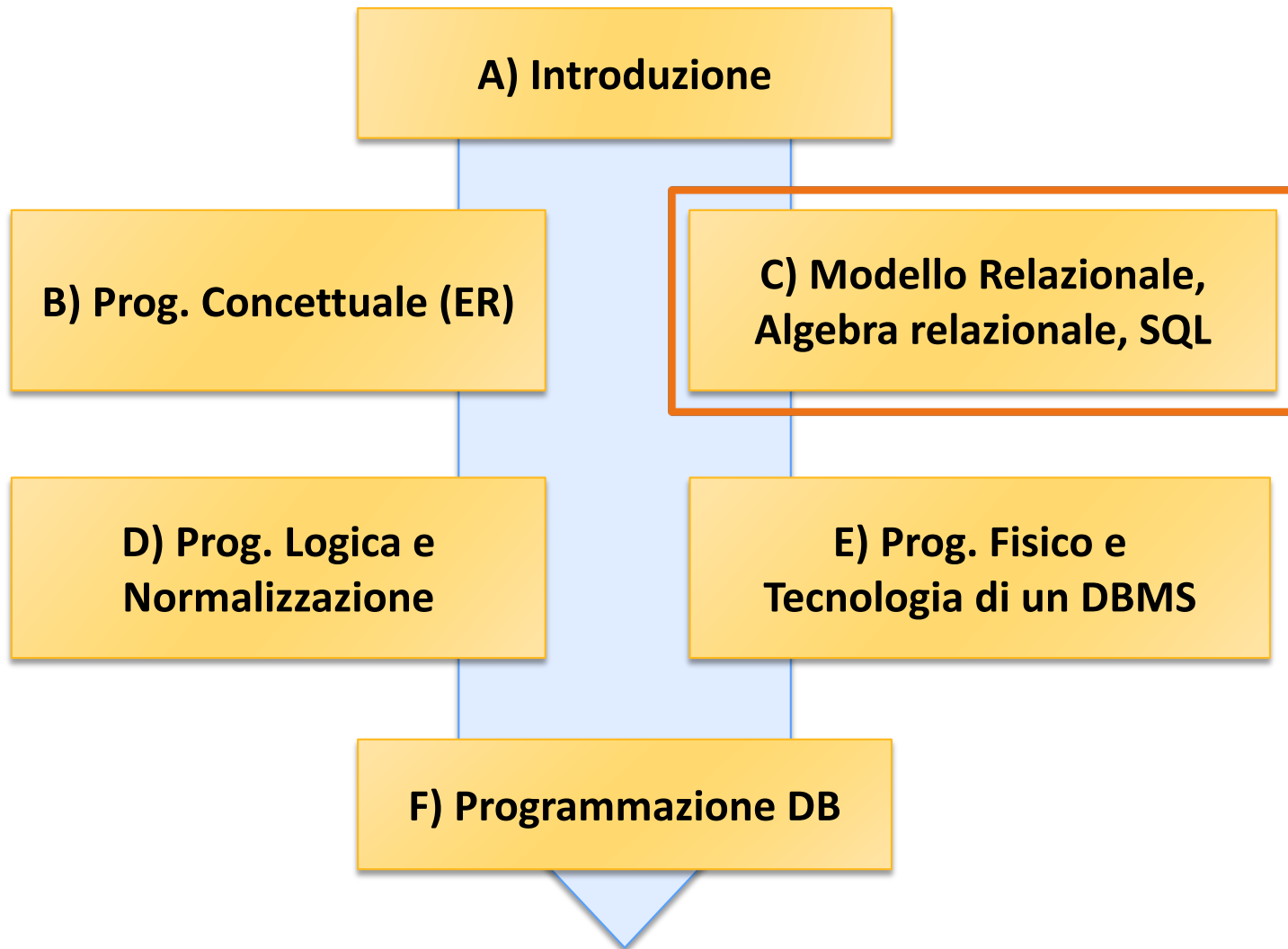


Basi di Dati



Aspetti avanzati SQL

Basi di Dati – Dove ci troviamo?



Aspetti avanzati SQL

- ▶ Aspetti avanzati DML (interrogazione)
 - ▶ Divisione
- ▶ Aspetti avanzati DDL (definizione schemi)
 - ▶ Creazione di indici
 - ▶ Modifica degli schemi
 - ▶ Gestione di viste
 - ▶ Autorizzazioni d'accesso

Divisione in SQL

- ▶ L'operazione di divisione, vista in algebra relazionale, non è definita in SQL. Pertanto, le interrogazioni che richiedono tale operatore , come ad esempio:
 - ▶ Selezionare i dati degli ordini che contengono tutti i prodotti di prezzo > 100

devono essere riformulate con una doppia negazione nel seguente modo:

- ▶ Selezionare i dati degli ordini per i quali non esiste alcun prodotto di prezzo > 100 che non sia contenuto in essi

Divisione in SQL

- Selezionare i dati degli ordini per i quali non esiste alcun prodotto di prezzo > 100 che non sia contenuto in essi

```
SELECT *  
FROM ORDINE O  
WHERE NOT EXISTS  
    ( SELECT *  
      FROM PRODOTTO P  
      WHERE P.PREZZO > 100  
        AND NOT EXISTS  
            ( SELECT *  
              FROM DETTAGLIO D  
              WHERE D.COD-PROD = P.COD-PROD  
                AND D.COD-ORD = O.COD-ORD))
```

Divisione in SQL

- Come sappiamo, invece, in algebra relazionale possiamo più semplicemente sfruttare l'operatore di divisione ...

$$\begin{array}{c} \text{ORDINE} \bowtie \\ \left(\pi_{\text{COD-ORD}, \text{COD-PROD}} \text{DETTAGLIO} \right. \\ \quad \div \\ \left. \pi_{\text{COD-PROD}} \left(\sigma_{\text{PREZZO} > 100} \text{PRODOTTO} \right) \right) \end{array}$$

Creazione di indici

- ▶ INDICI:

- ▶ meccanismi di accesso efficiente ai dati

- ▶ CREATE INDEX

```
CREATE INDEX DATA-IX  
ON ORDINI (DATA)
```

- ▶ CREATE UNIQUE INDEX

```
CREATE UNIQUE INDEX ORD-KEY  
ON ORDINI (ORD-COD)
```


Comandi di modifica degli schemi

- ▶ Necessari per garantire l'evoluzione della base di dati a fronte di nuove esigenze
- ▶ E' possibile:
 - ▶ CREARE NUOVI OGGETTI (CREATE)
 - ▶ MODIFICARE OGGETTI PREESISTENTI (ALTER)
 - ▶ CANCELLARE OGGETTI (DROP)

Cancellazione degli oggetti

- ▶ **DROP** (domini, tabelle, indici, view, asserzioni, procedure, trigger)
 - ▶ Es: DROP TABLE ORDINI
 - ▶ Es: DROP INDEX DATA-IX
- ▶ OPZIONI **RESTRICT** e **CASCADE**
 - ▶ **RESTRICT**: impedisce drop se gli oggetti comprendono istanze
 - ▶ **CASCADE**: applica drop agli oggetti collegati

Modifica degli oggetti

- ▶ **ALTER** (domini, tabelle)
 - ▶ Es: ALTER TABLE ORDINI ADD COLUMN NUM-FATT CHAR(6)
 - ▶ Es: ALTER TABLE ORDINI ALTER COLUMN IMPORTO ADD DEFAULT 0
 - ▶ Es: ALTER TABLE ORDINI DROP COLUMN DATA

Viste relazionali

- ▶ Offrono la "visione" di tabelle virtuali (schemi esterni)
- ▶ Classificate in
 - ▶ SEMPLICI (Selezione e proiezione su una sola tabella)
 - ▶ COMPLESSE

Sintassi delle viste relazionali

- ▶ **CREATE VIEW** <Nome> **AS** <QUERY>
- ▶ **CREATE VIEW** <Nome> (<Nome-attributi>) **AS** <QUERY>
- ▶ Le query possono includere al loro interno viste definite in precedenza

Esempio di vista semplice

- Ordini di importo superiore a 10.000 €

```
CREATE VIEW ORDINI-PRINCIPALI AS  
SELECT * FROM ORDINI  
WHERE IMPORTO > 10.000
```

ordini

1	3	2013-06-01	50.000
4	1	2014-07-01	12.000
6	3	2014-09-03	27.000

VISTA :
ordini principali



Uso della vista per query

► Vista:

```
CREATE VIEW ORDINI-PRINCIPALI AS  
SELECT *  
FROM ORDINI  
WHERE IMPORTO > 10.000
```

► Query:

```
SELECT NUM-CLI  
FROM ORDINI-PRINCIPALI
```

► Composizione della vista con la query:

```
SELECT NUM-CLI  
FROM ORDINI  
WHERE IMPORTO > 10.000
```

Modifiche tramite le viste

► Vista:

```
CREATE VIEW ORDINI-PRINCIPALI AS  
SELECT * FROM ORDINI  
WHERE IMPORTO > 10.000
```

► Modifica:

```
UPDATE ORDINI-PRINCIPALI  
SET IMPORTO = IMPORTO * 1.05  
WHERE NUM-CLI = 3
```

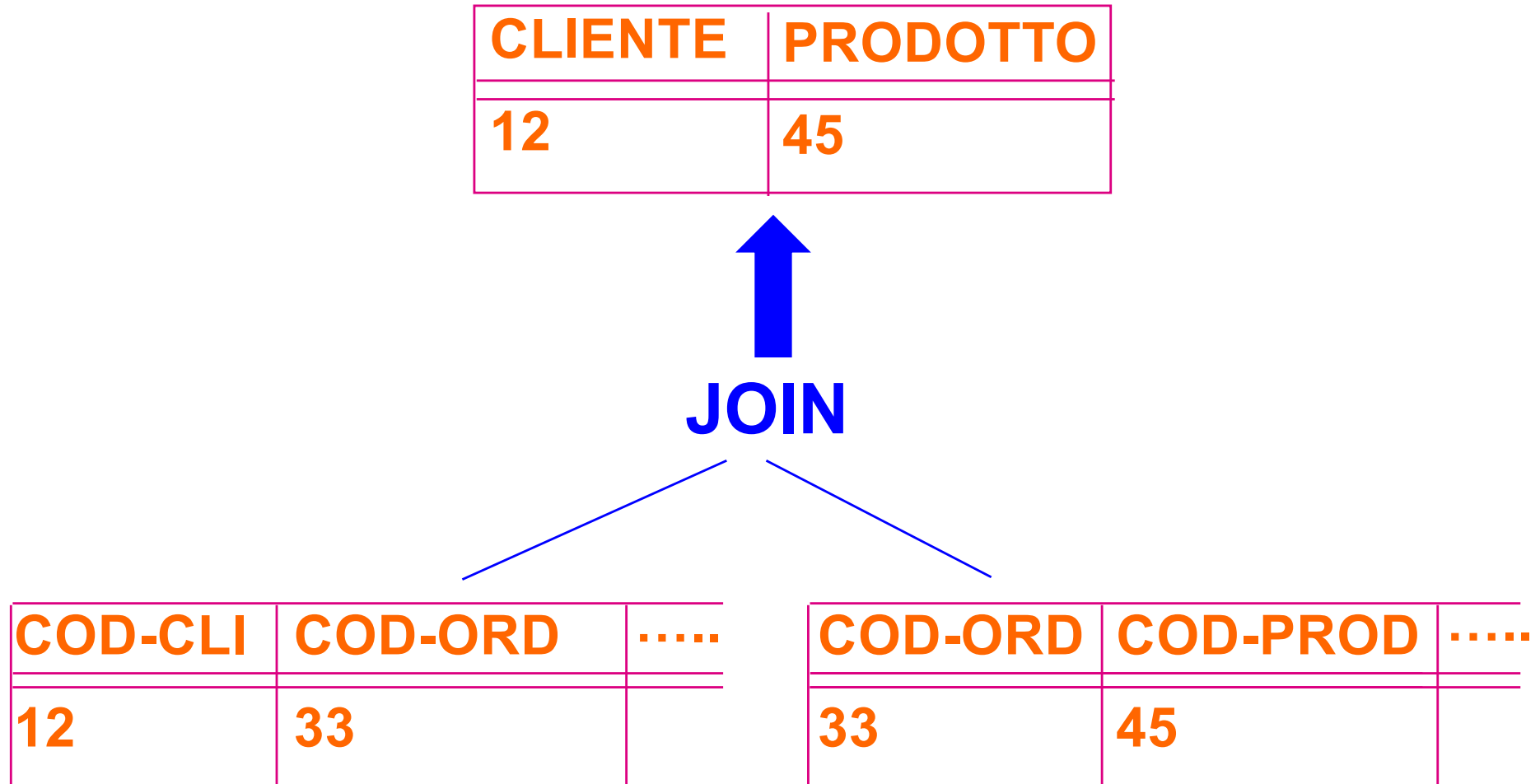
► Composizione della vista con la modifica:

```
UPDATE ORDINI  
SET IMPORTO = IMPORTO * 1.05  
WHERE NUM-CLI = 3  
AND IMPORTO > 10.000
```


Esempio di vista complessa

```
CREATE VIEW CLI-PRO (CLIENTE,PRODOTTO)
AS
SELECT COD-CLI, COD-PROD
FROM ORDINE JOIN DETTAGLIO
ON ORDINE.COD-ORD = DETTAGLIO.COD-ORD
```

Vista complessa (JOIN)



Interrogazione sulla vista complessa

- ▶ Query:

```
SELECT CLIENTE  
FROM CLI-PROD  
WHERE PRODOTTO = 45
```

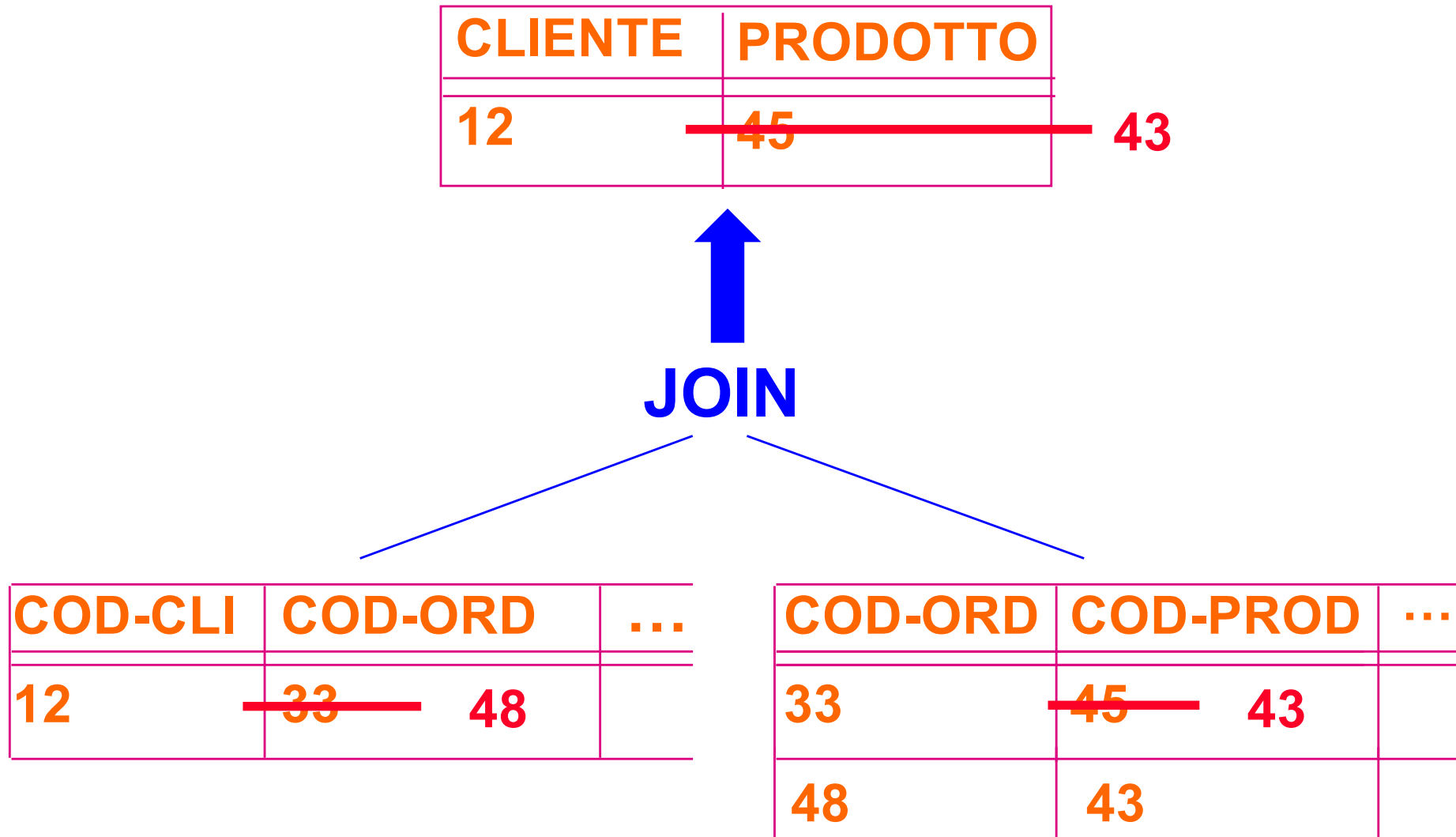
- ▶ Composizione della vista con la query:

```
SELECT COD-CLI  
FROM ORDINE JOIN DETTAGLIO  
ON ORDINE.COD-ORD = DETTAGLIO.COD-ORD  
WHERE COD-PROD = 45
```

Modifiche sulla vista complessa

- ▶ Non è possibile modificare le tabelle di base tramite la vista perché l'interpretazione è ambigua
 - ▶ Es: `UPDATE CLI-PROD SET PRODOTTO=43 WHERE CLIENTE=12`
- ▶ il cliente ha cambiato l'ordine
- ▶ il codice del prodotto è cambiato

Vista complessa (JOIN)



Autorizzazioni d'accesso

- ▶ **PRIVATEZZA**: protezione selettiva della base di dati in modo da garantire l'accesso solo agli utenti autorizzati.
- ▶ Meccanismi per identificare l'utente (tramite PAROLA CHIAVE o PASSWORD):
 - ▶ Quando si collega al sistema informatico
 - ▶ Quando accede al DBMS
- ▶ UTENTI individuali e GRUPPI di utenti

Autorizzazioni d'accesso

- ▶ Si attribuiscono agli UTENTI dei PRIVILEGI DI ACCESSO alle RISORSE
 - ▶ Utenti: identificati tramite password
 - ▶ Risorse: tabelle, view (e altre)
- ▶ **GRANT** <privilegi> **ON** <risorse> **TO** <utenti>

Principali privilegi di accesso

- ▶ SELECT [ATTRIBUTI]
- ▶ UPDATE [ATTRIBUTI]
- ▶ INSERT
- ▶ DELETE
- ▶ ALL PRIVILEGES

- ▶ ESEMPI:
 - ▶ GRANT ALL PRIVILEGES ON ORDINE TO User1
 - ▶ GRANT UPDATE(IMPORTO) ON ORDINE TO User2
 - ▶ GRANT SELECT ON ORDINE TO User2, User3

Concessione dei privilegi

- ▶ Il **creatore** di una risorsa ha tutti i privilegi
- ▶ Spesso l'utente **DATABASE ADMINISTRATOR** crea l'intera base di dati
- ▶ Chi detiene un privilegio può concederlo con **GRANT OPTION**:
 - ▶ **GRANT ALL PRIVILEGES ON ORDINE TO User1 WITH GRANT OPTION**

Esempio di uso, grant option

- ▶ Database administrator:
 - ▶ GRANT ALL PRIVILEGES ON ORDINE TO User1 WITH GRANT OPTION
- ▶ User1
 - ▶ GRANT SELECT ON ORDINE TO User2 WITH GRANT OPTION
- ▶ User2
 - ▶ GRANT SELECT ON ORDINE TO User3

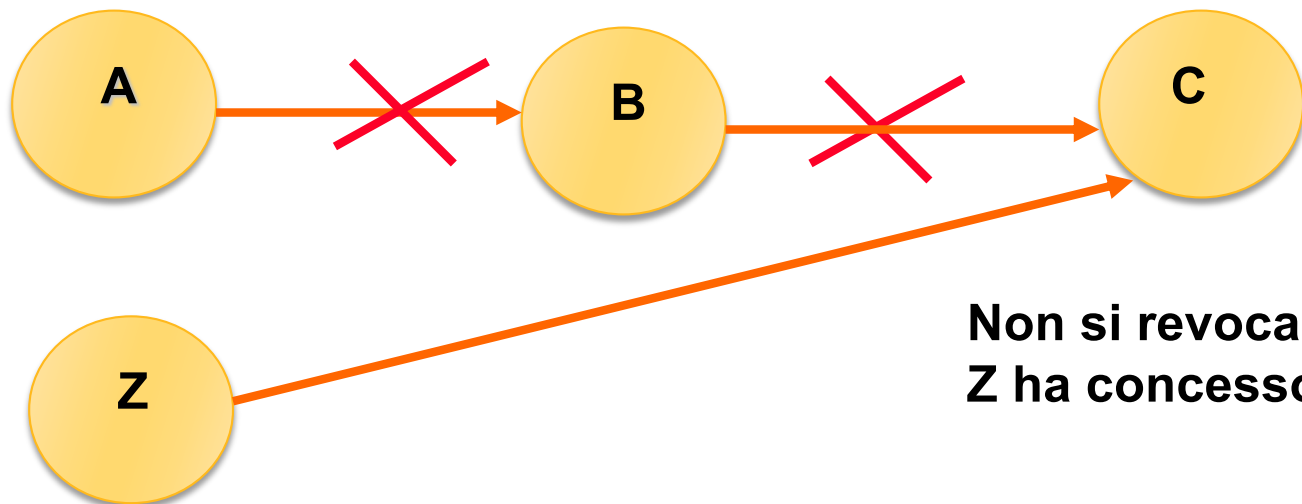
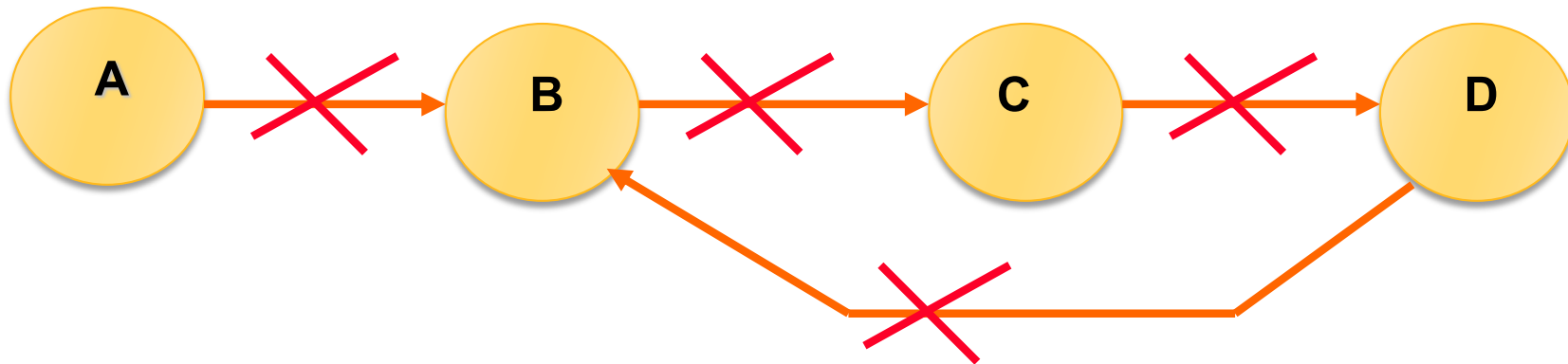
Revoca dei privilegi

- ▶ **REVOKE** <privilegi> **ON** <risorsa> **FROM** <utente>
 - ▶ REVOKE UPDATE ON ORDINE FROM User1
 - ▶ REVOKE SELECT ON ORDINE FROM User3
- ▶ Revoca dei privilegi con cascata
 - ▶ REVOKE <privilegi> ON <risorsa> FROM <utente> **CASCADE**

Revoca di un privilegio con cascata

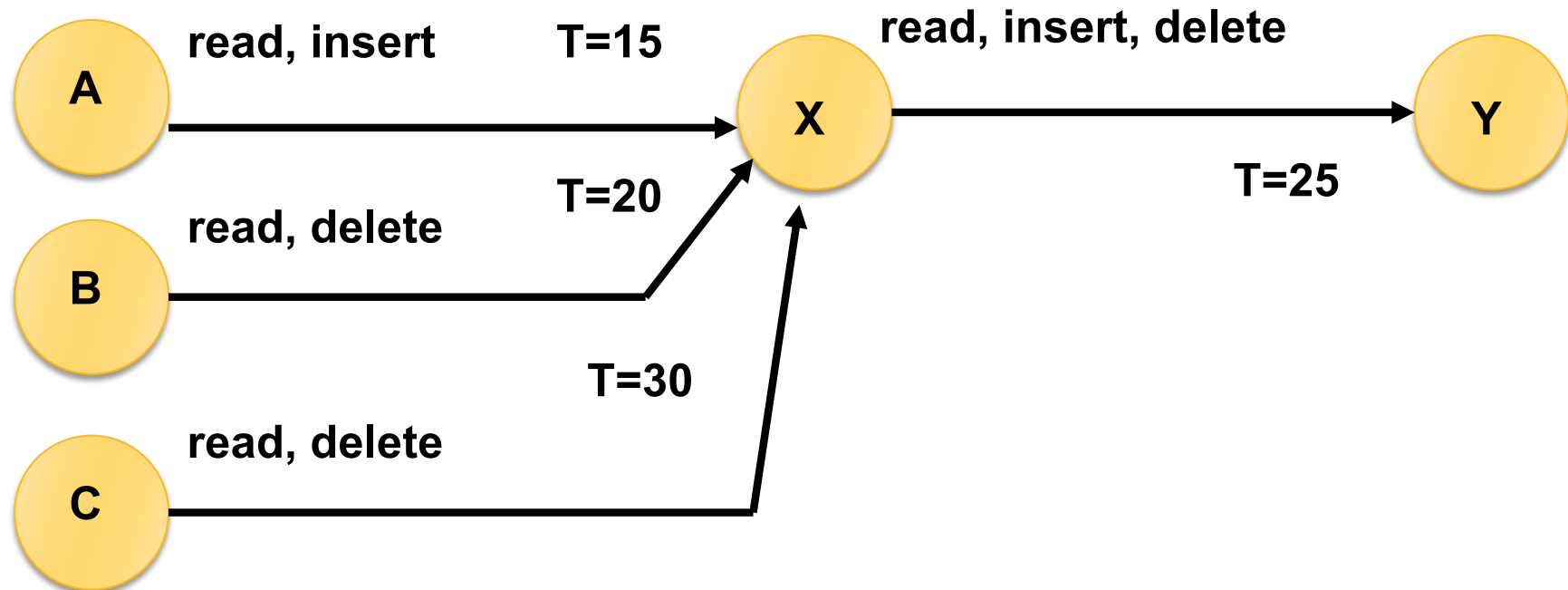
- ▶ Database administrator
 - ▶ GRANT SELECT ON ORDINE TO User1 WITH GRANT OPTION
- ▶ User1
 - ▶ GRANT SELECT ON ORDINE TO User2
- ▶ Database administrator
 - ▶ REVOKE SELECT ON ORDINE FROM User1 CASCADE

Revoca di un privilegio con cascata



**Non si revoca ciò che
Z ha concesso a C**

Algoritmo dei Time-Stamp



Al tempo $T=35$ B revoca (con cascata) tutti i diritti a X

Algoritmo dei Time-Stamp

CATALOGO DEI GRANT

User	tabella	grantor	read	insert	delete
X	EMP	A	15	15	-
X	EMP	B	20	-	20
Y	EMP	X	25	25	25
X	EMP	C	30	-	30

Ad ogni **GRANT** si inserisce una tupla con il tempo della concessione in corrispondenza del privilegio

Algoritmo dei Time-Stamp

User	tabella	grantor	read	insert	delete
X	EMP	A	15	15	-
X	EMP	B	20	-	20
Y	EMP	X	25	25	25
X	EMP	C	30	-	30

Se B revoca tutti i diritti a X si elimina la tupla (X,B)

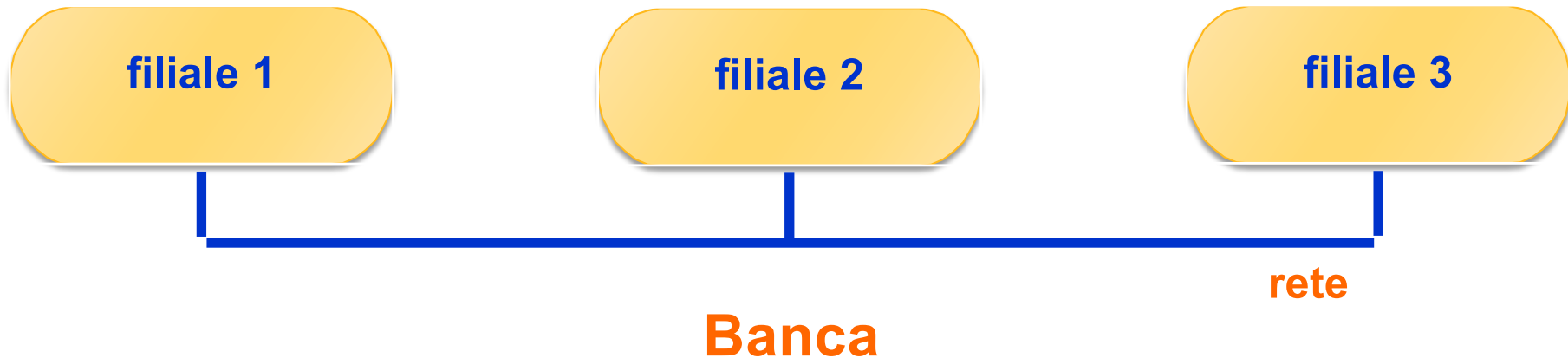
X può continuare a read, insert e delete perché li ha ricevuti anche da A e C

Ad Y si revoca il delete perché al tempo 25 X poteva averlo ricevuto solo da B

Viste e autorizzazioni di accesso

- ▶ Viste = unità di autorizzazione
 - ▶ Consentono la gestione ottimale della privacy

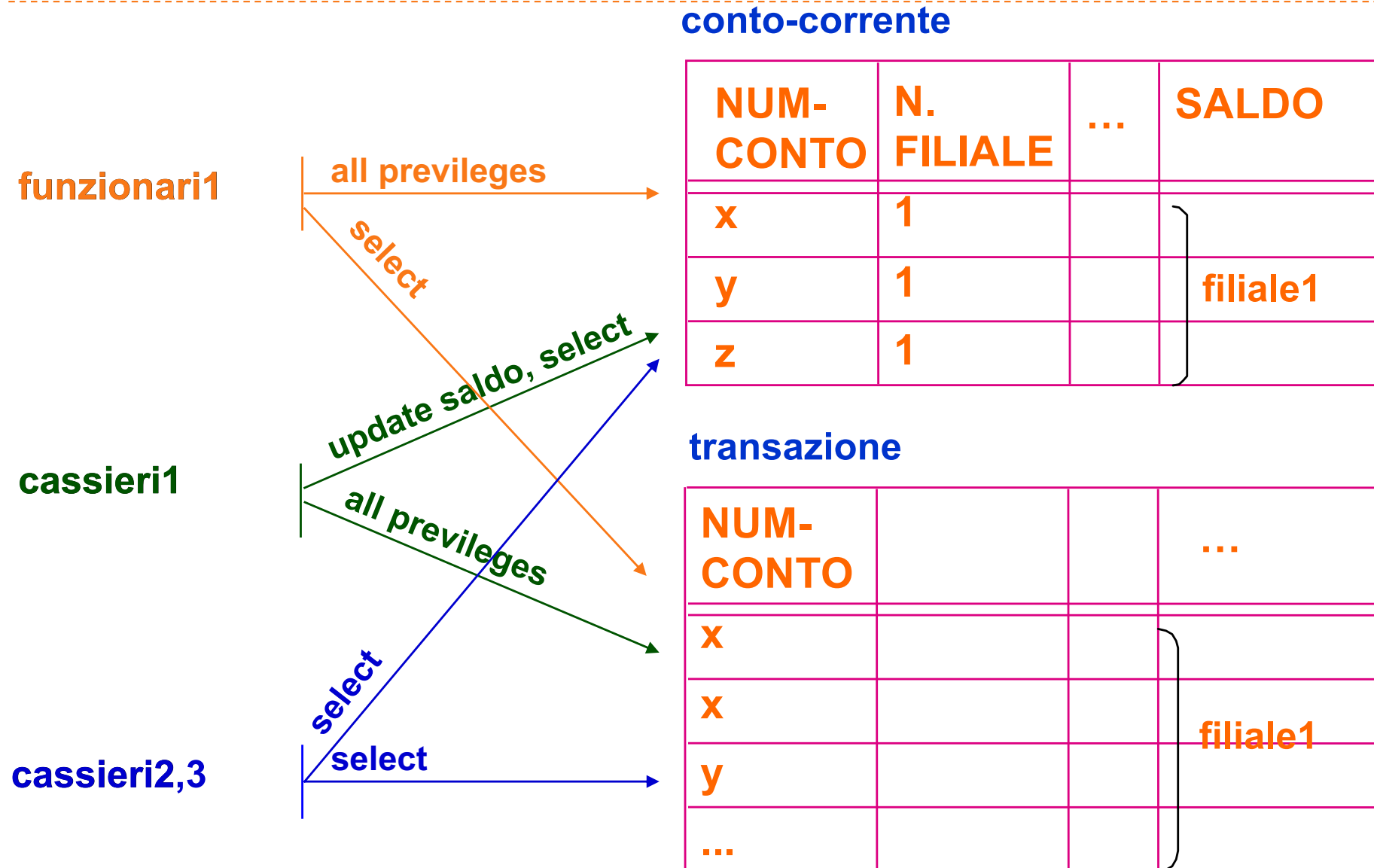
Esempio: gestione dei conti correnti



CONTO-CORRENTE (NUM-CONTO, FILIALE,
CLIENTE, COD-FISC, DATA-APERTURA, SALDO)

TRANSAZIONE (NUM-CONTO, DATA, PROGR,
CAUSALE, AMMONTARE)

Requisiti di accesso



Viste relative alla prima filiale

```
CREATE VIEW CONTO1 AS  
  SELECT *  
  FROM CONTO-CORRENTE  
  WHERE FILIALE = 1
```

```
CREATE VIEW TRANSAZIONE1 AS  
  SELECT *  
  FROM TRANSAZIONE  
  WHERE NUM-CONTO IN  
    ( SELECT NUM-CONTO  
      FROM CONTO1 )
```

Autorizzazioni relative ai dati della prima filiale

GRANT ALL PRIVILEGES ON CONTO1 TO FUNZIONARI1

GRANT UPDATE(SALDO) ON CONTO1 TO CASSIERI1

GRANT SELECT ON CONTO1 TO CASSIERI1, CASSIERI2,
CASSIERI3

GRANT SELECT ON TRANSAZIONE1 TO FUNZIONARI1

GRANT ALL PRIVILEGES ON TRANSAZIONE1 TO CASSIERI1

GRANT SELECT ON TRANSAZIONE1 TO CASSIERI2, CASSIERI3

Esercizi

- ▶ Riprendere le basi di dati per la gestione del personale ed esprimere:
 - ▶ i comandi per modificare la tabella IMPIEGATO aggiungendo il campo COD-FISC
 - ▶ una vista complessa che contiene i nomi degli impiegati che lavorano ai vari progetti
 - ▶ la vista che seleziona gli IMPIEGATI che lavorano al progetto “Wide”
 - ▶ le istruzioni che consentono all’utente MGR-WIDE il pieno controllo sulla vista