

Aspetti avanzati nella definizione degli schemi

Aspetti avanzati del DDL

- **Creazione di indici**
- **Modifica degli schemi**
- **Gestione di viste**
- **Autorizzazioni d'accesso**
- **Vincoli di integrità**
- **Procedure e regole attive**

Creazione di indici

INDICI:

meccanismi di accesso efficiente ai dati

CREATE INDEX

es.: **CREATE INDEX DATA-IX
ON ORDINI (DATA)**

CREATE UNIQUE INDEX

es.: **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**
- **È 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** (Join, raggruppamento...)

Sintassi delle viste relazionali

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

Esempio di vista semplice

- Ordini di importo superiore a 10.000.000

```
CREATE VIEW ORDINI-PRINCIPALI AS  
SELECT * FROM ORDINI  
WHERE IMPORTO > 10.000.000
```

ordini

1	3	1996-06-01	50.000.000
4	1	1997-07-01	12.000.000
6	3	1997-09-03	27.000.000

**VISTA :
ordini principali**



Uso della vista per query

Vista:

```
CREATE VIEW ORDINI-PRINCIPALI AS  
SELECT *  
FROM ORDINI  
WHERE IMPORTO > 10.000.000
```

Query:

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

Composizione della vista con la query:

```
SELECT NUM-CLI  
FROM ORDINI  
WHERE IMPORTO > 10.000.000
```

Modifiche tramite le viste

Vista: **CREATE VIEW ORDINI-PRINCIPALI AS**
SELECT * FROM ORDINI
WHERE IMPORTO > 10.000.000

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 > 10.000.000

Esempio di vista complessa (vista con raggruppamento)

```
CREATE VIEW TOT-ORDINI  
      (CLIENTE,TOTALE)
```

AS

```
SELECT NUM-CLI, SUM(IMPORATO)  
FROM ORDINE  
GROUP BY NUM-CLI
```

vista NON modificabile!!!

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)

CLIENTE	PRODOTTO
12	45

JOIN



COD-CLI	COD-ORD
12	33	

COD-ORD	COD-PROD
33	45	

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 la interpretazione è ambigua

- es.: **UPDATE CLI-PROD
SET PRODOTTO = 43
WHERE CLIENTE = 12**

- a** il cliente ha cambiato l'ordine
- b** il codice del prodotto è cambiato

Vista complessa (JOIN)

CLIENTE	PRODOTTO
12	45 42



JOIN

COD-CLI	COD-ORD
12	33 45	

COD-ORD	COD-PROD
33	45 42	
45	42	

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

1 Database administrator :

**GRANT ALL PRIVILEGES ON ORDINE
TO User1 WITH GRANT OPTION**

2 User1

**GRANT SELECT ON ORDINE TO User2
WITH GRANT OPTION**

3 User2

GRANT SELECT ON ORDINE TO User3

Revoca dei privilegi

REVOKE <privilegi> ON <risorsa> FROM <utente>

es:

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

1 Database administrator

**GRANT SELECT ON ORDINE TO User1
WITH GRANT OPTION**

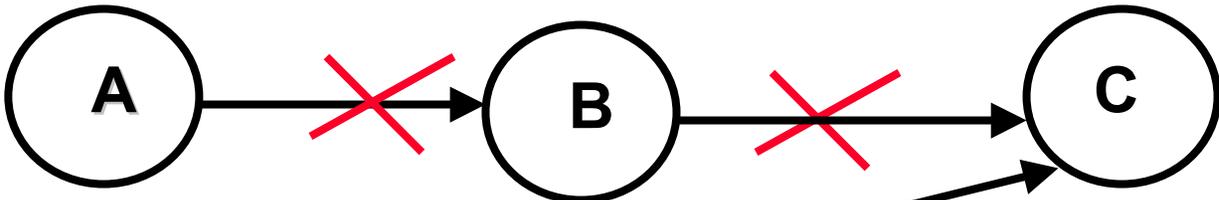
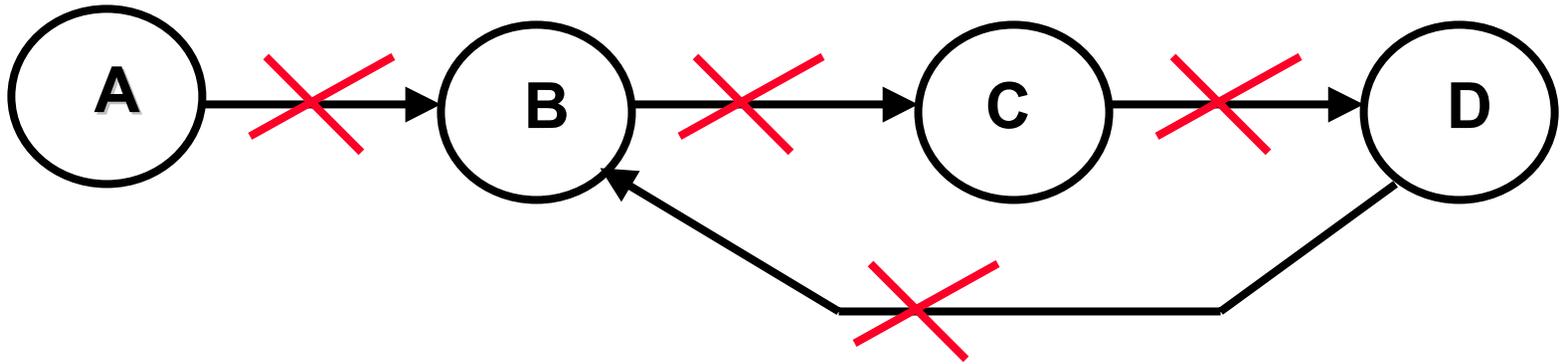
2 User1

GRANT SELECT ON ORDINE TO User2

3 Database administrator

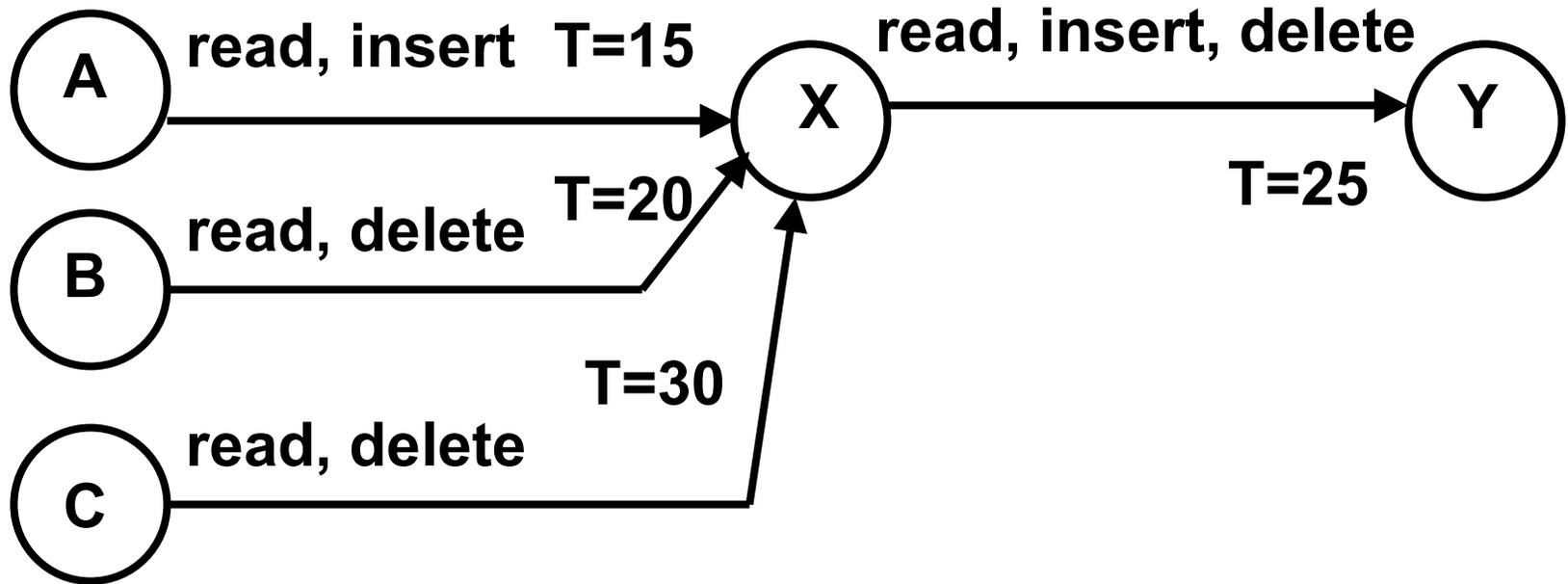
**REVOKE SELECT ON ORDINE FROM
User1 CASCADE**

Revoca di un privilegio con cascata



Non si revoca ciò che Z ha concesso a C

Algoritmo dei Time-Stamped



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

A X rimangono read (A e C), insert (A), delete (C)

A Y rimangono read e insert (X da A)

Algoritmo dei Time-Stamped

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-Stamped

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

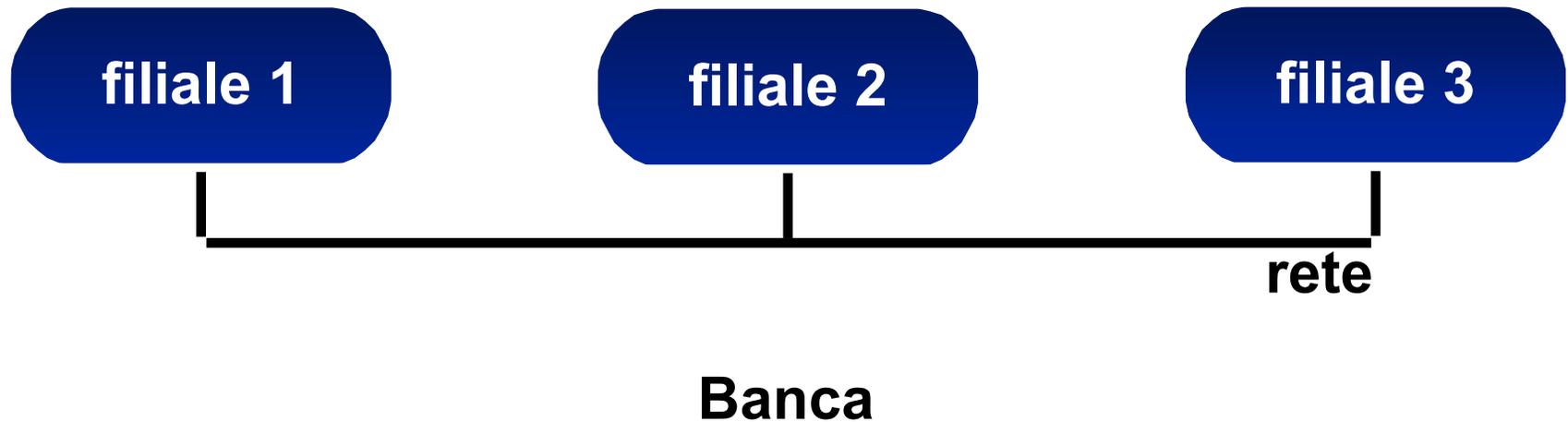
Ad Y si revoca il delete perché, al tempo 25, X poteva averlo ricevuto solo da B

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)**

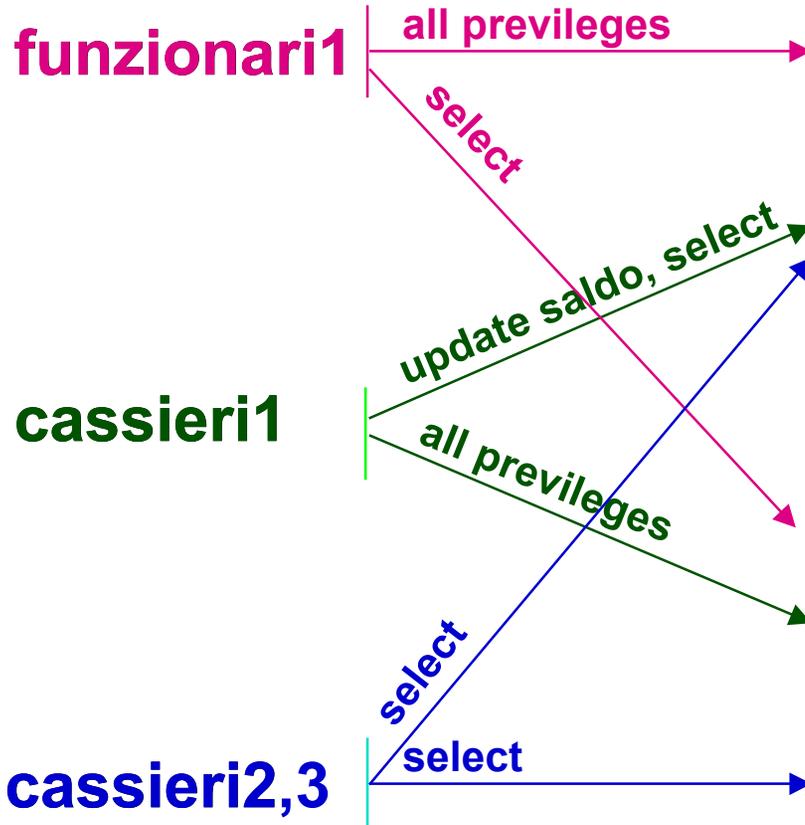
Requisiti di accesso

conto-corrente

NUM-CONTO	N. FILIALE	...	SALDO
x	1		} filiale1
y	1		
z	1		

transazione

NUM-CONTO			...
x			} filiale1
x			
y			
...			



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 il 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**