

Universita' di Milano Bicocca  
Corso di Basi di dati 1 in eLearning  
C. Batini  
6. SQL DDL  
6.2 Data Description Language - 2

# Vincoli di integrita'

# Cosa e' un vincolo di integrita'

- E' una proprieta' sempre valida per le istanze di una o piu' relazioni

# Tipi di vincoli

- Intrarelazionali - proprietà valida all'interno di una relazione
- Interrelazionali - proprietà relativa a più relazioni

# Vincoli intrarelazionali

# Vincoli intrarelazionali

1. NOT NULL
2. UNIQUE definisce chiavi
3. PRIMARY KEY: chiave primaria (una sola)
4. CHECK, permette di definire sia vincoli intra che inter relazionali, lo vedremo più avanti

# Vincolo NOT NULL

Indica che il valore nullo non è ammesso per il dominio a cui è associato il vincolo.



# Vincolo UNIQUE

- Si applica a uno (A) o un insieme di attributi (I)
- Impone che i valori dell'attributo A o dell'insieme I siano una chiave, cioè che tuple diverse abbiano valori diversi su A o I.
- Fa eccezione il valore nullo, che si assume sempre diverso nelle diverse occorrenze
- Due forme:
  - nella definizione di un attributo, se forma da solo la chiave
  - Es. `Cognome CHAR(20) UNIQUE`
  - come elemento separato, se definito su più attributi
  - Es. `Nome CHAR(20) NOT NULL,  
Cognome CHAR(20) NOT NULL,  
UNIQUE (Cognome, Nome)`



# Chiavi su più attributi, attenzione

```
Nome CHAR(20) NOT NULL,  
Cognome CHAR(20) NOT NULL,  
UNIQUE (Cognome, Nome),
```

```
Nome CHAR(20) NOT NULL UNIQUE,  
Cognome CHAR(20) NOT NULL UNIQUE,
```

non hanno lo stesso effetto! Perché la seconda dichiarazione corrisponde a due comandi distinti

# CREATE TABLE, esempio sviluppato fino a questo punto

```
CREATE TABLE Impiegato(  
  Matricola CHAR(6) PRIMARY KEY,  
  Nome CHAR(20) NOT NULL,  
  Cognome CHAR(20) NOT NULL,  
  Dipart CHAR(15),  
  Stipendio NUMERIC(9) DEFAULT 0,  
  FOREIGN KEY(Dipart) REFERENCES  
    Dipartimento(NomeDip),  
  UNIQUE (Cognome, Nome)  
)
```



## Clausola PRIMARY KEY

Serve per specificare la chiave primaria

Puo' essere usata una sola volta

Due forme (come per unique) per un attributo o piu' attributi

Gli attributi della chiave primaria non possono assumere valori nulli

Matricola CHAR(6) PRIMARY KEY

Matricola CHAR(6),

PRIMARY KEY (Matricola)

# CREATE TABLE, esempio sviluppato fino a questo punto

```
CREATE TABLE Impiegato(  
  Matricola CHAR(6) PRIMARY KEY,  
  Nome CHAR(20) NOT NULL,  
  Cognome CHAR(20) NOT NULL,  
  Dipart CHAR(15),  
  Stipendio NUMERIC(9) DEFAULT 0,  
  FOREIGN KEY(Dipart) REFERENCES  
    Dipartimento(NomeDip),  
  UNIQUE (Cognome, Nome)
```

)

# Vincoli interrelazionali

# Vincoli interrelazionali

- Vincoli di integrita' referenziale (IR)
- REFERENCES e FOREIGN KEY
- CHECK, vedremo più avanti perche' e' prima necessario trattare le interrogazioni SQL



## Vincoli di IR

- Creano un legame tra i valori di n attributi della tabella corrente (detta **interna**) e i valori di n attributi di una seconda tabella (detta **Esterna o Master o Principale**) di cui tipicamente costituiscono una chiave
- di nuovo due sintassi (come in precedenza)
  - per singoli attributi
  - su più attributi
- E' possibile definire politiche di reazione alla violazione

# Infrazioni (tabella interna)

<u>Codice</u>	Data	Vigile	Prov	Numero
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

Vigili  
(esterna  
o master)

<u>Matricola</u>	Cognome	Nome
3987	Rossi	Luca
3295	Neri	Piero
9345	Neri	Mario
7543	Mori	Gino



## Infrazioni (interna)

<u>Codice</u>	Data	Vigile	Prov	Numero
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

Auto  
(esterna  
o master)

<u>Prov</u>	<u>Numero</u>	Cognome	Nome
MI	39548K	Rossi	Mario
TO	E39548	Rossi	Mario
PR	839548	Neri	Luca



# CREATE TABLE, esempio con le due sintassi

```
CREATE TABLE Infrazioni(  
  Codice CHAR(6) NOT NULL PRIMARY KEY,  
  Data DATE NOT NULL,  
  Vigile
```

```
    INTEGER NOT NULL  
    REFERENCES Vigili(Matricola),  
  Provincia CHAR(2),  
  Numero CHAR(6),  
  FOREIGN KEY(Provincia, Numero)  
    REFERENCES Auto(Provincia, Numero)  
)
```

# Politiche di reazione alle violazioni

- Se viene eseguita una operazione di aggiornamento che porta alla violazione di un vincolo di integrita' referenziale, lo stato della base di dati diventa non valido.
- Sono percio' definite un insieme di politiche per evitare che questo accada.
- Tali politiche vanno dichiarate nella dichiarazione DDL dello schema.

# Politiche di reazione alle violazioni o referential triggered action

- Cause di violazione:
- 1. Operazione sulla tabella interna (TI)
  - inserimento di tupla nella TI
  - modifica di valore di attributo nella TI
  - In entrambi i casi il comando non viene eseguito
- 2. Operazione sulla tabella esterna/master (TE)
  - Quattro politiche possibili (vedi →)
- Motivo della asimmetria: la TE e' la tabella master, quindi e' la tabella rilevante per la integrita' complessiva della base di dati

# Politiche per comando su TE: modifica/cancellazione

## 1. Cascade:

- Operazione di Modifica → valore modificato anche nella TI
- Operazione di Cancellazione → la cancellazione viene propagata alla TI

## 2. Set null: NULL nella TI in entrambi i casi

## 3. Set default: default nella TI in entrambi i casi

## 4. No action: rifiutata in entrambi i casi

# Sintassi delle politiche nel linguaggio SQL-DDL

- Vedi il libro di riferimento.

## Politiche per comando su TE: modifica/cancellazione

Le due politiche possono essere differenti  
per modifica/cancellazione

Esempio

- Dipart CHARACTER (15)  
REFERENCES Dipartimento  
(NomeDip)  
ON DELETE SET NULL  
ON UPDATE CASCADE

# Istruzioni SQL di modifica di schema



# Modifiche degli schemi: ALTER

- ALTER DOMAIN, modifica domini (default, vincoli)
- ALTER TABLE, modifica tabelle.
- Attenzione:
  - Quando si definisce un nuovo vincolo, deve essere soddisfatto dalla istanza presente, altrimenti viene rifiutato.

# Modifiche degli schemi: DROP

- **DROP DOMAIN**, cancellazione del dominio (attributo)
- **DROP TABLE**, cancellazione della tabella
- Due opzioni:
  - **RESTRICT**, un oggetto (dominio, tabella) non e' rimosso se non e' vuoto
  - **CASCADE**, viene rimosso l'oggetto e tutto cio' che e' coinvolto nella definizione dell'oggetto

# In pratica

- In molti sistemi si utilizzano strumenti diversi dal codice SQL per definire lo schema della base di dati
- Es. Form in Access

# Esercizio 6.2.1

- Esegui l'esercizio 4.3 del libro di riferimento

## Esercizio 6.2.2

- Esegui l'esercizio 4.4 del libro di riferimento

## Esercizio 6.2.3

- Esegui l'esercizio 4.5 del libro di riferimento

## Esercizio 6.2.4

- Esegui l'esercizio 4.7 del libro di riferimento

# Concetti introdotti

- Politica di reazione alle violazioni