

Laboratorio di Basi di Dati e Multimedia

Laurea in Informatica Multimediale - A.A. 2008/09

Docente: Carlo Combi

Email: carlo.combi@univr.it

Lezione 2

Vincoli intrarelazionali

Proprietà che devono essere soddisfatte da ogni istanza della base di dati

Il soddisfacimento è definito rispetto a singole relazioni della base di dati

In SQL:

- **NOT NULL**
- **UNIQUE** definisce chiavi
- **PRIMARY KEY**: chiave primaria (una sola, implica **NOT NULL**)
- **CHECK**: vincolo generico

Vincoli interrelazionali

- Vincoli che coinvolgono più relazioni
- I più significativi sono i *vincoli di integrità referenziale* o vincoli di riferimento
 - In SQL i vincoli di integrità referenziale sono definiti mediante
FOREIGN KEY
(chiave esterna o chiave esportata)

FOREIGN KEY

Crea un legame tra i valori dell'attributo della tabella corrente (*interna*) e i valori dell'attributo di un'altra tabella (*esterna*)

- Impone che per ogni riga della tabella *interna* il valore dell'attributo, se diverso dal valore nullo, sia presente tra i valori di un attributo della tabella *esterna*.
- ATTENZIONE: L'attributo della tabella *esterna* a cui si fa riferimento deve essere soggetto a vincolo UNIQUE (o PRIMARY KEY).

FOREIGN KEY

- Nel vincolo possono essere coinvolti più attributi. Ciò accade, ad esempio, quando la chiave della tabella **esterna** è costituita da un insieme di attributi.
 - Si confrontano insiemi di valori invece che singoli valori.
- Un vincolo di integrità referenziale può essere definito in due modi:
 - Uso costruito **REFERENCES** su un attributo
 - Uso costruito **FOREIGN KEY** come vincolo di tabella

Costrutto REFERENCES

- Si usa il costrutto REFERENCES quando il vincolo è definito su un unico attributo.
- Con REFERENCES (nella tabella interna) si specificano la tabella esterna e l'attributo della tabella esterna con il quale l'attributo della tabella interna deve essere legato.

CREATE TABLE: uso del costrutto REFERENCES

Tabella Interna

```
CREATE TABLE Impiegato(  
    Matricola    CHAR(6)          PRIMARY KEY,  
    Nome        VARCHAR(20)      NOT NULL,  
    Cognome     VARCHAR(20)      NOT NULL,  
    NomeDipartimento VARCHAR(15)  
    REFERENCES Dipartimento(NomeDip));
```

Tabella Esterna

Attributo Chiave

CREATE TABLE:uso del costrutto REFERENCES

Tabella Esterna

Vincolo di
UNIQUE o
PRIMARY KEY

```
CREATE TABLE Dipartimento(  
  NomeDip VARCHAR(15) PRIMARY KEY,  
  Sede VARCHAR(20) NOT NULL,  
  Telefono VARCHAR(15));
```


ESEMPIO

Tabella Interna:
IMPIEGATO

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

Tabella Esterna:
DIPARTIMENTO

Vincolo
UNIQUE o
PRIMARY
KEY

<u>NomeDip</u>	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

Costrutto FOREIGN KEY

- Si usa il costrutto FOREIGN KEY quando il vincolo è definito su un insieme di attributi
- Con FOREIGN KEY (nella tabella **interna**) si elencano gli attributi della tabella **interna** coinvolti nel legame e con REFERENCES si specificano la tabella **esterna** e gli attributi della tabella **esterna** con il quale gli attributi della tabella **interna** devono essere legati

CREATE TABLE: FOREIGN KEY E REFERENCES

```
CREATE TABLE Impiegato(  
    Matricola CHAR(6) PRIMARY KEY,  
    Nome VARCHAR(20) NOT NULL,  
    Cognome VARCHAR(20) NOT NULL,  
    NomeDipartimento VARCHAR(15)  
    REFERENCES Dipartimento(NomeDip),  
    FOREIGN KEY(Nome,Cognome)  
    REFERENCES Anagrafica(Nome,Cognome));
```

Tabella Interna

Tabella Esterna

Attributi Chiave
(ordinati)

CREATE TABLE: FOREIGN KEY E REFERENCES

Tabella Esterna

```
CREATE TABLE Anagrafica(  
  CodFisc      CHAR(11)      PRIMARY KEY,  
  Nome        VARCHAR(20)    ,  
  Cognome     VARCHAR(20)    NOT NULL,  
  Indirizzo   VARCHAR(30),  
  UNIQUE(Nome,Cognome)  
);
```

Vincolo di
UNIQUE

ESEMPIO

Tabella Interna:
IMPIEGATO

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

Tabella Esterna:
ANAGRAFICA

Vincolo di
UNIQUE

<u>CodFisc</u>	Nome	Cognome	Indirizzo
RSSMRA...	Mario	Rossi	Via X
VRDPAO...	Paolo	Verdi	Via Y

Violazione dei vincoli e politiche di reazione

- E' possibile associare ad un vincolo di integrità referenziale una *politica di reazione alle violazioni*
 - SQL permette di decidere quale reazione adottare
- Per gli altri vincoli, in presenza di violazione, l'aggiornamento viene rifiutato

Violare i vincoli operando sulla tabella interna

- Si possono introdurre violazioni modificando il contenuto della tabella interna solo in due modi:
 - Modificando il valore dell'attributo referente
 - Inserendo una nuova riga
- Per queste operazioni SQL non offre nessun supporto:
 - Le operazioni vengono semplicemente impedito

ESEMPIO

Tabella Interna:
IMPIEGATO

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite
A00003	Marco	Bianchi	Marketing

Tentativo di inserimento che
causa **VIOLAZIONE!!!**

Tabella Esterna:
DIPARTIMENTO

<u>NomeDip</u>	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

ESEMPIO

Tabella Interna:
IMPIEGATO

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite
A00003	Mario	Bianchi	Marketing

L'inserimento viene impedito

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

Violare i vincoli operando sulla tabella esterna

- Diverse alternative per rispondere a violazioni generate da modifiche sulla tabella esterna (o tabella Master)
- La tabella interna (o tabella Slave) deve adeguarsi alle modifiche che avvengono sulla tabella Master
- Le violazioni possono avvenire per:
 - Modifiche dell'attributo riferito (MA)
 - Cancellazione righe dalla tabella Master (CR)

Politiche di reazione per modifica attributo riferito

- **Cascade**: il nuovo valore dell'attributo della tabella **esterna** viene riportato su tutte le corrispondenti righe della tabella **interna**
- Esempio: modifica di un valore dell'attributo **NomeDip** nella tabella **DIPARTIMENTO**
 - **DIPARTIMENTO**: Da **Vendite** a **VenditeMI**
 - **IMPIEGATO**: Da **Vendite** a **VenditeMI**

CASCADE (MA)

Tabella Esterna:
DIPARTIMENTO

Tabella Interna:
IMPIEGATO

<u>NomeDip</u>	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

<u>NomeDip</u>	Sede	Telefono
Acquisti	Verona	045/8008080
VenditeMI	Milano	02/8007070

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	VenditeMI

CASCADE (MA): risultato

<u>NomeDip</u>	Sede	Telefono
Acquisti	Verona	045/8008080
VenditeMI	Milano	02/8007070

Tabella Esterna:
DIPARTIMENTO

Tabella Interna:
IMPIEGATO

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	VenditeMI

Politiche di reazione per modifica attributo riferito

- **Set null**: all'attributo referente (tabella **interna**) viene assegnato valore nullo al posto del valore modificato nella tabella **esterna**
- Esempio: modifica di un valore dell'attributo **NomeDip** nella tabella **DIPARTIMENTO**
 - **DIPARTIMENTO**: Da **Vendite** a **VenditeMI**
 - **IMPIEGATO**: Da **Vendite** a **NULL**

SET NULL (MA)

Tabella Esterna:
DIPARTIMENTO

Tabella Interna:
IMPIEGATO

<u>NomeDip</u>	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

<u>NomeDip</u>	Sede	Telefono
Acquisti	Verona	045/8008080
VenditeMI	Milano	02/8007070

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	NULL

Politiche di reazione per modifica attributo riferito

- **Set default**: all'attributo referente viene assegnato un valore di default al posto del valore modificato nella tabella **esterna**
- Esempio: modifica di un valore dell'attributo **NomeDip** nella tabella **DIPARTIMENTO** supponendo che il valore di default sia **DipVendite**
 - **DIPARTIMENTO**: Da **Vendite** a **VenditeMI**
 - **IMPIEGATO**: Da **Vendite** a **DipVendite**

SET DEFAULT (MA)

Tabella Esterna:
DIPARTIMENTO

Tabella Interna:
IMPIEGATO

<u>NomeDip</u>	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

<u>NomeDip</u>	Sede	Telefono
Acquisti	Verona	045/8008080
VenditeMI	Milano	02/8007070

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	DipVendite

Politiche di reazione per modifica attributo riferito

- **No action**: non viene eseguita alcuna reazione. Il sistema può generare messaggio di errore ma la tabella interna non viene modificata
- Esempio: modifica di un valore dell'attributo **NomeDip** nella tabella **DIPARTIMENTO**
 - **DIPARTIMENTO**: Da **Vendite** a **VenditeMI**
 - **IMPIEGATO**: Da **Vendite** a **Vendite**

NO ACTION (MA)

Tabella Esterna:
DIPARTIMENTO

Tabella Interna:
IMPIEGATO

<u>NomeDip</u>	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

<u>NomeDip</u>	Sede	Telefono
Acquisti	Verona	045/8008080
VenditeMI	Milano	02/8007070

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

Politiche di reazione per cancellazione riga tabella esterna

- SQL mette a disposizione le stesse politiche di reazione:
 - **Cascade**: tutte le righe della tabella **interna** corrispondenti alla riga cancellata vengono cancellate
 - **Set null**: all'attributo referente viene assegnato il valore nullo al posto del valore presente nella riga cancellata dalla tabella **esterna**
 - **Set default**: all'attributo referente viene assegnato un valore di default
 - **No action**: non viene eseguita alcuna reazione

Politiche di reazione per cancellazione tupla tabella Master

- **Cascade**: tutte le righe della tabella **interna** corrispondenti alla riga cancellata vengono cancellate
- Esempio: cancellazione dalla tabella **DIPARTIMENTO** della riga che ha **NomeDip** uguale a **Vendite**

CASCADE (CR)

Tabella Esterna:
DIPARTIMENTO

<u>NomeDip</u>	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

Tabella Interna:
IMPIEGATO

<u>NomeDip</u>	Sede	Telefono
Acquisti	Verona	045/8008080

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti

Vincoli di integrità: sommario

- Vincoli su attributi

- Vincolo Attributo:=

- [NOT NULL [UNIQUE]] | [CHECK (Condizione)]
[REFERENCES Tabella [(Attributo {, Attributo})]]
[ON {DELETE|UPDATE} {NO ACTION | CASCADE |
SET NULL | SET DEFAULT}]

- Vincoli su tabella

- Vincolo Tabella:= UNIQUE(Attributo {, Attributo})

- | CHECK(Condizione) |

- | PRIMARY KEY [Nome] Attributo {, Attributo})

- | FOREIGN KEY [Nome] Attributo {, Attributo})

- REFERENCES Tabella [(Attributo {, Attributo})]

- [ON {DELETE|UPDATE} {NO ACTION | CASCADE |
SET NULL | SET DEFAULT}]

CREATE TABLE: esempio

```
CREATE TABLE Impiegato(  
    Matricola      CHAR(6)      PRIMARY KEY,  
    Nome          VARCHAR(20) NOT NULL,  
    Cognome       VARCHAR(20) NOT NULL,  
    NomeDipartimento VARCHAR(15)  
        REFERENCES Dipartimento(NomeDip),  
    FOREIGN KEY(Nome,Cognome)  
        REFERENCES Anagrafica(Nome,Cognome)  
    ON DELETE SET NULL  
    ON UPDATE CASCADE);
```


Modifica degli schemi:

ALTER TABLE

- Per aggiungere un nuovo attributo:

```
ALTER TABLE      NomeTabella  
ADD COLUMN        NuovoAttr Tipo
```

- Per rimuovere un attributo:

```
ALTER TABLE      NomeTabella  
DROP COLUMN       NomeAttributo
```

- Per modificare il valore di default di un attributo:

```
ALTER TABLE      Nometabella  
ALTER COLUMN      NomeAttributo  
{SET DEFAULT NuovoDefault | DROP DEFAULT}
```

Modifica degli schemi:

ALTER TABLE

- Per aggiungere un nuovo vincolo:

ALTER TABLE Nometabella

ALTER COLUMN NomeAttributo

ADD CONSTRAINT DefVincolo

- Per rimuovere un vincolo:

ALTER TABLE Nometabella

ALTER COLUMN NomeAttributo

DROP CONSTRAINT NomeVincoli

Modifica degli schemi:

DROP TABLE

- Per cancellare una tabella:

`DROP TABLE NomeTabella`

Modifiche degli schemi: esempi

- Aggiungere l'attributo "Sesso" alla tabella Impiegato:
`ALTER TABLE Impiegato
ADD COLUMN Sesso char;`
- Per l'attributo "Sesso" inserire come valore di default "F"
`ALTER TABLE Impiegato
ALTER COLUMN Sesso
SET DEFAULT 'F';`
- Rimuovere l'attributo "Sesso" dalla tabella Impiegato:
`ALTER TABLE Impiegato
DROP COLUMN Sesso;`
- Rimuovere la tabella Impiegato:
`DROP TABLE Impiegato;`

SQL: operazioni sui dati (DML)

- Modifica:
 - **INSERT**: inserimento tuple
 - **DELETE**: cancellazione tuple
DELETE FROM TABELLA WHERE Condizione
 - **UPDATE**: aggiornamento valori tuple
UPDATE Tabella
SET ATTRIBUTO=Expr, ..., Attributo=Expr
WHERE Condizione
- Interrogazione:
 - **SELECT** <target_list> FROM <table_list>
 - **WHERE** <condition>

INSERT


- Per inserire una tupla in una tabella

```
INSERT INTO NomeTabella  
[(<ElencoAttributi>)]  
VALUES (<Elenco di Valori>)
```

Istruzione DELETE

- Per eliminare righe dalle tabelle

DELETE FROM NomeTabella
[WHERE Condizione]



Vengono rimosse le righe
che soddisfano la
condizione

DELETE: esempio

- Eliminare le righe della tabella Dipartimento con nome del dipartimento uguale a “Vendite”

```
DELETE FROM Dipartimento  
WHERE NomeDip='Vendite';
```

- Attenzione ai vincoli di integrità referenziale con politica cascade

<u>NomeDip</u>	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

Istruzione UPDATE


- Per aggiornare uno o più attributi delle righe di una tabella:

UPDATE NomeTabella

SET ATTRIBUTO1 = Expr,

ATTRIBUTO2 = Expr

WHERE Condizione



Se la condizione non
compare vengono
aggiornate tutte le righe

UPDATE: esempio

- Aumentare lo stipendio di tutti gli impiegati del dipartimento Vendite di 100 euro.

UPDATE Impiegato

SET Stipendio = Stipendio + 100

WHERE NomeDip='Vendite';

- Risultato:

Matricola	Nome	Cognome	NomeDipartimento	Stipendio
A00001	Mario	Rossi	Acquisti	1000
A00002	Paolo	Verdi	Vendite	1000 -> 1100

Istruzione SELECT

- Il comando base del linguaggio SQL:

```
SELECT Attributo {,Attributo}  
FROM Tabella {,Tabella}  
[WHERE Condizione]
```

SELECT: esempio 1

- Trovare tutti i dati inseriti nella tabella Impiegato

SELECT *

FROM Impiegato;

- Risultato:

<u>Matricola</u>	Nome	Cognome	NomeDipartimento
A00001	Mario	Rossi	Acquisti
A00002	Paolo	Verdi	Vendite

SELECT: esempio 2

- Trovare Matricola, Nome, Cognome dei dati inseriti nella tabella Impiegato

```
SELECT Matricola, Nome, Cognome  
FROM Impiegato;
```

- Risultato:

<u>Matricola</u>	Nome	Cognome
A00001	Mario	Rossi
A00002	Paolo	Verdi