



# Laboratorio di Basi di Dati e Web (2006-07)

Docente: Alberto Belussi

*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.

Vincoli intrarelazionali di SQL (visti nella prima esercitazione):

- NOT NULL: richiede che l'attributo sia valorizzato
- UNIQUE: definisce chiavi
- PRIMARY KEY: chiave primaria (c'è solo una chiave primaria in una tabella, implica NOT NULL)
- CHECK: vincolo generico

# Vincoli interrelazionali

Vincoli che coinvolgono più relazioni

I più significativi sono i **vincoli di integrità referenziale** (consentono di mantenere consistenti i legami logici tra tuple).

In SQL la definizione dei vincoli di integrità referenziale si converte nella definizione di un vincolo **FOREIGN KEY** (chiave esportata).

# FOREIGN KEY

Crea un legame tra i valori di un attributo A (o di più attributi) della tabella corrente (*interna o Slave*) e i valori presenti nell'attributo B (o in più attributi) di un'altra tabella (*esterna o Master*).

E' una versione più debole dei vincoli di integrità referenziale visto in teoria. Infatti:

- ◆ impone che in ogni tupla della tabella *interna* il valore di A, se diverso dal valore nullo, sia presente tra i valori di B nella tabella *esterna*.
- ◆ l'attributo B della tabella *esterna* deve essere soggetto a un vincolo UNIQUE (o PRIMARY KEY). E' ammesso quindi che B non sia la chiave primaria purché sia però "identificante" per le tuple della tabella esterna.

# FOREIGN KEY

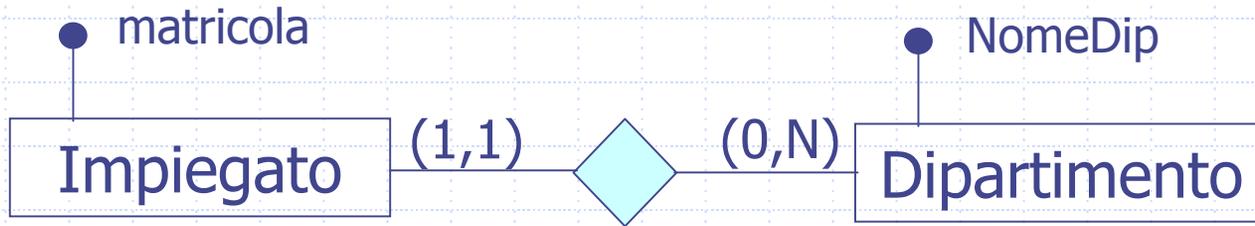
- ◆ Nel vincolo possono essere coinvolti più attributi, ad esempio quando la chiave primaria della tabella **esterna** è costituita da un insieme di attributi.
  - Si confrontano combinazioni di valori invece che singoli valori.
- ◆ Un vincolo di integrità referenziale può essere definito sintatticamente 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 solo attributo.
- ◆ Con REFERENCES (nella tabella **interna**) si specificano
  - la tabella **esterna** e
  - l'attributo della tabella **esterna** al quale l'attributo della tabella **interna** deve essere legato.

# Esempio schema base di dati

## Schema concettuale:



## Schema relazionale:

Impiegato(Matricola, Nome, Cognome, NomeDipartimento)

Dipartimento(NomeDip, Sede, Telefono)

# CREATE TABLE: uso del costrutto 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));
```

Tabella Slave

Chiave Esportata

Tabella Master

Chiave

# CREATE TABLE: uso del costrutto REFERENCES

Tabella Master

Vincolo di  
UNIQUE o  
PRIMARY KEY

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

# Esempio

Tabella Slave:  
IMPIEGATO

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

Tabella Master:  
DIPARTIMENTO

Vincolo  
UNIQUE o  
PRIMARY  
KEY

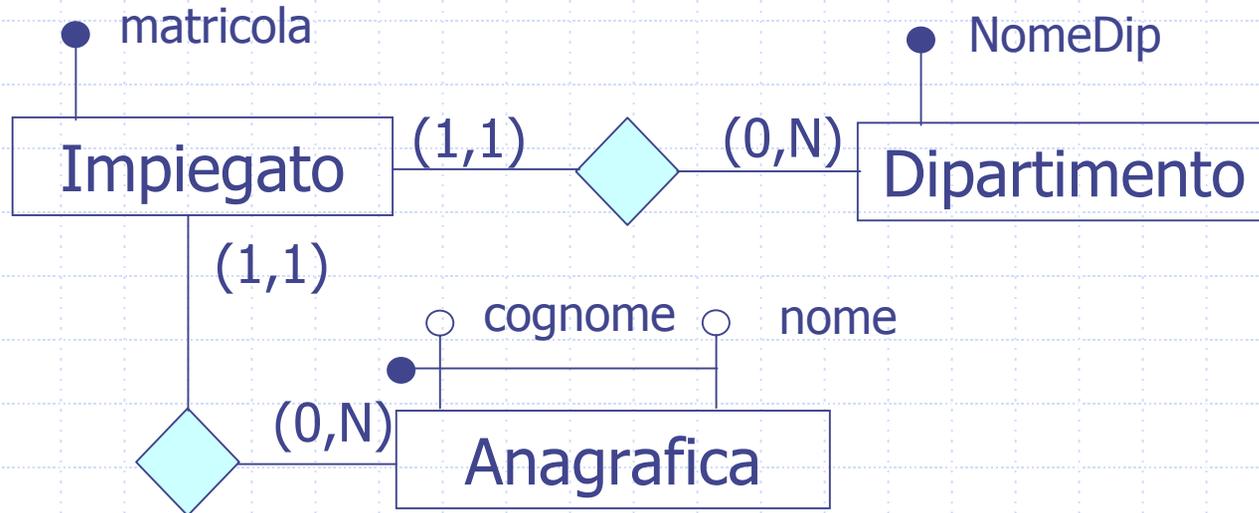
<u>NomeDip</u>	Sede	Telefono
Acquisti	Verona	045/8008080
Vendite	Milano	02/8007070

# Costrutto FOREIGN KEY

- ◆ Si usa il costrutto FOREIGN KEY quando il vincolo di integrità referenziale è definito su un insieme di attributi.
- ◆ Con FOREIGN KEY (nella tabella **interna**) si elencano:
  - gli attributi della tabella **interna** coinvolti nel legame e
  - usando la parola chiave REFERENCES si specificano la tabella **esterna** e gli attributi della tabella **esterna** ai quali gli attributi della tabella **interna** devono essere legati.

# Esempio schema base di dati

## Schema concettuale:



## Schema relazionale:

Impiegato(Matricola, Nome, Cognome, NomeDipartimento)

Dipartimento(NomeDip, Sede, Telefono)

Anagrafica(CodiceFiscale, Cognome, Nome, Indirizzo)

# CREATE TABLE: uso del costrutto FOREIGN KEY

Tabella Slave

```
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));
```

Chiave  
Esportata

Tabella Master

Chiave (attributi  
ordinati)

# CREATE TABLE: uso del costrutto FOREIGN KEY

Tabella Master

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

Vincolo di  
UNIQUE

# Esempio

Tabella Slave:  
IMPIEGATO

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

Tabella Master:  
ANAGRAFICA

Vincolo di  
UNIQUE

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

# Violazione 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.

# Violazione dei vincoli operando sulla tabella **Slave**

- ◆ 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 Slave:  
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 Master:  
DIPARTIMENTO

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

# ESEMPIO

Tabella Slave:  
IMPIEGATO

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

A00003	Marco	Bianchi	Marketing
--------	-------	---------	-----------

L'inserimento viene impedito

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

# Violazione dei vincoli operando sulla tabella **Master**

- ◆ Diverse alternative per rispondere a violazioni generate da modifiche sulla tabella **Master**.
- ◆ La tabella **Slave** deve adeguarsi alle modifiche che avvengono sulla tabella **Master**.
- ◆ Le violazioni possono avvenire per:
  - Modifiche dell'attributo riferito (MA)
  - Cancellazione di tuple dalla tabella **Master** (CT)

# Politiche di reazione per modifica attributo riferito 1/4

- ◆ **Cascade:** il nuovo valore assegnato all'attributo della tabella **Master** 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:** propagazione per CASCADE Da **Vendite** a **VenditeMI**

# CASCADE (MA)

Tabella Master:  
DIPARTIMENTO

Tabella Slave:  
IMPIEGATO

UPDATE

<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

Propagazione con  
cascade

# Politiche di reazione per modifica attributo riferito 2/4

- ◆ **Set null:** all'attributo referente della tabella **Slave** viene assegnato un valore nullo al posto del valore modificato nella tabella **Master**.
- ◆ Esempio: modifica di un valore dell'attributo **NomeDip** nella tabella **DIPARTIMENTO**
  - **DIPARTIMENTO:** Da **Vendite** a **VenditeMI**
  - **IMPIEGATO:** propagazione per SET NULL Da **Vendite** a **NULL**

# SET NULL (MA)

Tabella Master:  
DIPARTIMENTO

Tabella Slave:  
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 3/4

- ◆ **Set default:** all'attributo referente viene assegnato un valore di default al posto del valore modificato nella tabella **Master**.
- ◆ 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:** propagazione per SET DEFAULT Da **Vendite** a **DipVendite**

# SET DEFAULT (MA)

Tabella Master:  
DIPARTIMENTO

Tabella Slave:  
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 4/4

- ◆ **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:** rimane com'era

# NO ACTION (MA)

Tabella Master:  
DIPARTIMENTO

Tabella Slave:  
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 tupla tabella Master

- ◆ SQL mette a disposizione le stesse politiche di reazione:
  - **Cascade:** tutte le righe della tabella **Slave** 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 **Master** .
  - **Set default:** all'attributo referente viene assegnato un valore di default al posto del valore presente nella riga cancellata dalla tabella **Master**.
  - **No action:** non viene eseguita alcuna reazione.

# Politiche di reazione per cancellazione tupla tabella Master

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

# CASCADE (CR)

Tabella Master:  
DIPARTIMENTO

Tabella Slave:  
IMPIEGATO

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

<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 completo

```
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 NomeVincolo
```

# Modifica degli schemi: DROP TABLE

- ◆ Per cancellare una tabella:  
`DROP TABLE NomeTabella`

# Modifiche degli schemi: ESEMPI

- ◆ Aggiungere l'attributo "Stipendio" alla tabella Impiegato:  
`ALTER TABLE Impiegato  
ADD COLUMN Stipendio numeric(8,2);`
- ◆ Per l'attributo "Stipendio" inserire come valore di default "1000.00"  
`ALTER TABLE Impiegato  
ALTER COLUMN Stipendio  
SET DEFAULT 1000.00;`
- ◆ Rimuovere l'attributo "Stipendio" dalla tabella Impiegato:  
`ALTER TABLE Impiegato  
DROP COLUMN Stipendio;`
- ◆ 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 tuple di una tabella:

UPDATE NomeTabella

SET ATTRIBUTO1 = Expr,

ATTRIBUTO2 = Expr

WHERE Condizione

Se la condizione non viene specificata vengono aggiornate tutte le tuple

# UPDATE: esempio

- ◆ Aumentare lo stipendio di tutti gli impiegati Dipartimento 'Vendite' di 100 euro.

UPDATE Impiegato

SET Stipendio = Stipendio + 100

WHERE NomeDip='Vendite';

- ◆ Risultato:

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

# Fine

Vedere compito assegnato per la seconda esercitazione di laboratorio alla pagina

**<http://www.scienze.univr.it/fof/main?ent=oi&id=28310>**