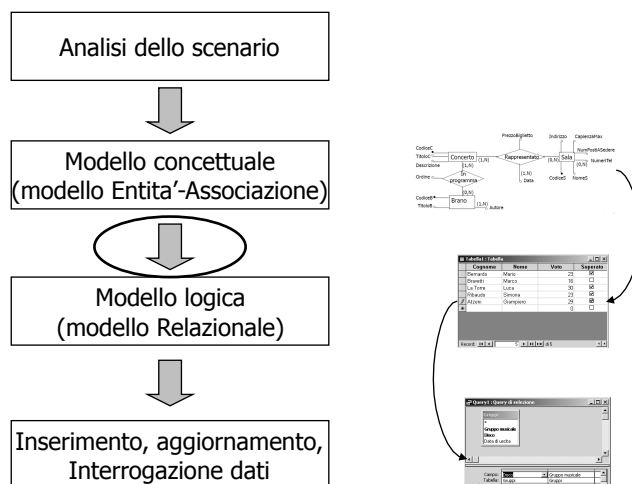


Dal modello concettuale al modello logico

Traduzione dal modello Entita'- Associazione al modello Relazionale



Ciclo di sviluppo di una base di dati (da parte dell'utente)



Informatica applicata alla comunicazione
multimediale

2

La progettazione logica

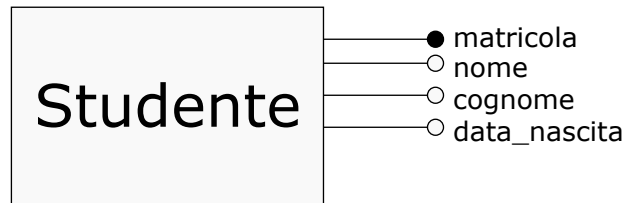
□ Obiettivo:

- Costruire uno schema logico in grado di descrivere le informazioni contenute nello schema Entità-Associazione prodotto nella fase di progettazione concettuale
- A partire da entità, associazioni e attributi:
 - quali tabelle e' necessario creare?
 - Per ogni tabella
 - quali attributi compongono la tabella?
 - quale e' la chiave primaria della tabella?
 - vincoli di tupla?
 - Quali sono le relazioni tra le tabelle?
 - Quali sono i vincoli tra le tabelle (vincoli di integrità referenziale)?

Ristrutturazione di uno schema Entità'-Associazione

- Prima di passare allo schema logico, lo schema Entità-Associazione va ristrutturato
- Eliminazione dallo schema EA di tutti i costrutti non direttamente rappresentabili nel modello relazionale:
 - Attributi composti
 - Attributi multi-valore
 - Gerarchie di generalizzazione
 - (non trattato in questo corso)
- Vedremo la ristrutturazione di uno schema EA nelle slide finali di questa presentazione
 - Per il momento si ipotizza di non avere attributi composti o multivalore

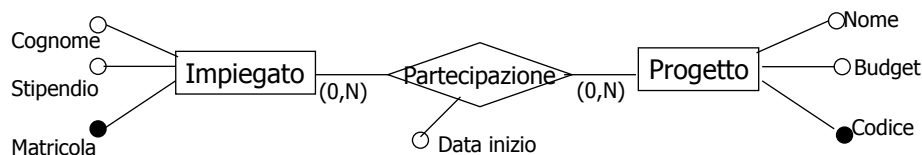
Traduzione verso il modello relazionale: le entità'



Le entità vengono tradotte con una relazione che comprende gli attributi e la cui chiave primaria corrisponde all'identificatore dell'entità

Studente(matricola, nome, cognome, data_nascita)

Verso il modello relazionale: associazione molti a molti



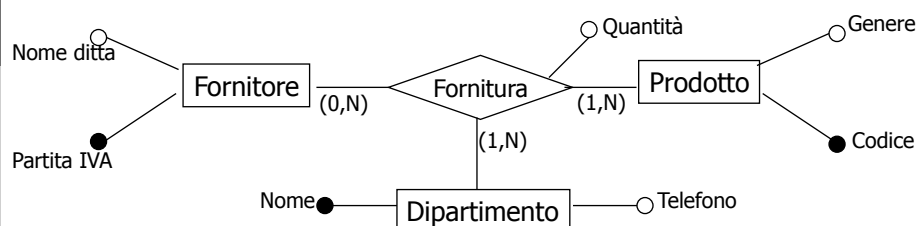
- ❑ Traduciamo l'associazione in una tabella che comprende:
 - i) le chiavi primarie delle entità coinvolte
 - ii) gli attributi propri dell'associazione
- ❑ Schema relazionale corrispondente:
 - Impiegato(Matricola, Cognome, Stipendio)
 - Progetto(Codice, Nome, Budget)
 - Partecipazione(Matricola, Codice, DataInizio)
- ❑ Due vincoli di integrità: tra Partecipazione e Impiegato (Matricola) e tra Partecipazione e Progetto (Codice)

Verso il modello relazionale: associazione molti a molti

- ❑ Per rendere più comprensibile lo schema è conveniente effettuare ridenominazioni
- ❑ Per esempio: in Partecipazione, Matricola \Rightarrow Impiegato, Codice \Rightarrow Progetto

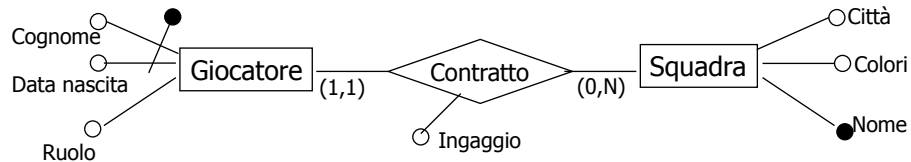
Impiegato(Matricola, Cognome, Stipendio)
 Progetto(Codice, Nome, Budget)
 Partecipazione(Impiegato, Progetto, DataInizio)

Verso il modello relazionale: associazione molti a molti



Fornitore(PartitaIVA, NomeDitta)
 Prodotto(Codice, Genere)
 Dipartimento(Nome, Telefono)
 Fornitura(Fornitore, Prodotto, Dipartimento, Quantità)

Verso il modello relazionale: associazione uno a molti



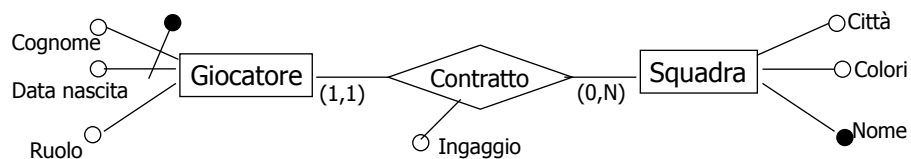
Potremmo procedere come nel caso precedente:

Giocatore(Cognome, DataNascita, Ruolo)
 Squadra(Nome, Città, Colori)
 Contratto(Giocatore, DataNascitaGiocatore, NomeSquadra, Ingaggio)

Informatica applicata alla comunicazione
multimediale

9

Verso il modello relazionale: associazione uno a molti



TRADUZIONE MIGLIORE

- Dato che l'associazione è uno a molti, possiamo dare una traduzione più compatta traducendo l'associazione mediante attributi nella relazione Giocatore, che ha cardinalità (1,1):

Giocatore(Cognome, DataNascita, Ruolo, NomeSquadra, Ingaggio)

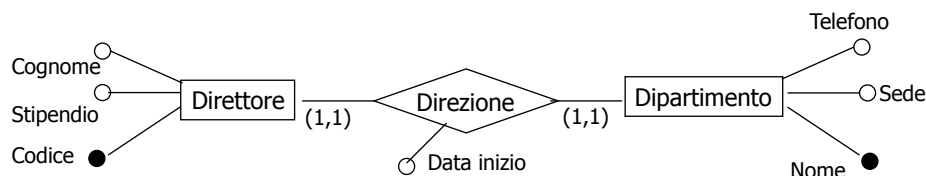
Squadra(Nome, Città, Colori)

- Un vincolo di integrità referenziale: tra NomeSquadra della relazione Giocatore e Nome della relazione Squadra

Informatica applicata alla comunicazione
multimediale

10

Verso il modello relazionale: associazione uno a uno



Due possibilità simmetriche e ugualmente valide:

Direttore(Codice, Cognome, Stipendio, DipartimentoDiretto, InizioDirezio)

Dipartimento(Nome, Telefono, Sede)

NB: devo inoltre imporre che i valori del campo DipartimentoDiretto della tabella Direttore siano tutti diversi tra loro! Altrimenti creo una relazione 1-N

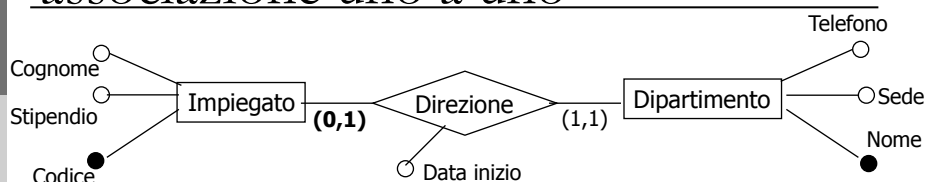
Direttore(Codice, Cognome, Stipendio)

Dipartimento(Nome, Telefono, Sede, Direttore, InizioDirezio)

NB: devo inoltre imporre che i valori del campo Direttore della tabella Dipartimento siano tutti diversi tra loro! Altrimenti creo una relazione 1-N

11

Verso il modello relazionale: associazione uno a uno



□ Partecipazione opzionale di una delle due entita'

- Traduciamo l'associazione mediante attributi della relazione Dipartimento, che partecipa all'associazione con cardinalità (1,1):

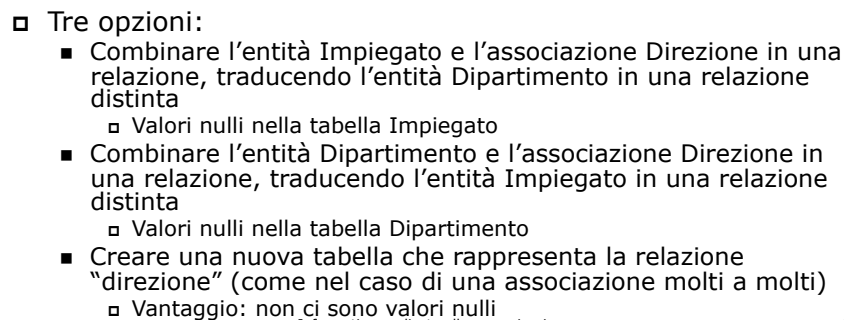
Impiegato(Codice, Cognome, Stipendio)

Dipartimento(Nome, Telefono, Sede, Direttore, InizioDirezio)

- Vincolo di integrità referenziale: tra Direttore della relazione Dipartimento e Codice della relazione Impiegato

NB: devo inoltre imporre che i valori del campo Direttore della tabella Dipartimento siano tutti diversi tra loro! Altrimenti creo una relazione 1-N

12



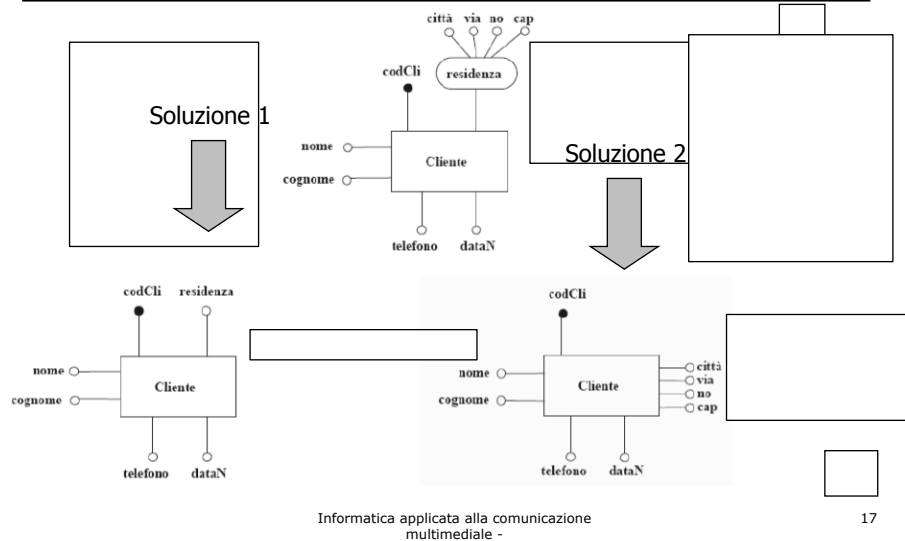
Ristrutturazione di uno schema Entita'-Associazione

- ❑ Prima di passare allo schema logico, lo schema Entità-Associazione va ristrutturato
- ❑ Eliminazione dallo schema EA di tutti i costrutti non direttamente rappresentabili nel modello relazionale:
 - Attributi composti
 - Attributi multi-valore

Eliminare gli attributi **composti**

- ❑ Eliminazione di un **attributo composto** A da un'entità E
- ❑ Soluzione 1: eliminazione dei sotto-attributi di A
 - attributo composto diventa attributo semplice
 - e' compito dell'applicazione garantire che il nuovo attributo contenga valori coerenti con la semantica dell'attributo composto ristrutturato
- ❑ Soluzione 2: considerare tutti i sotto-attributi di A come attributi di E
 - ridefinizione del dominio dell'attributo
- ❑ Se le componenti dell'attributo composto sono a loro volta attributi composti, si ri-applica la procedura

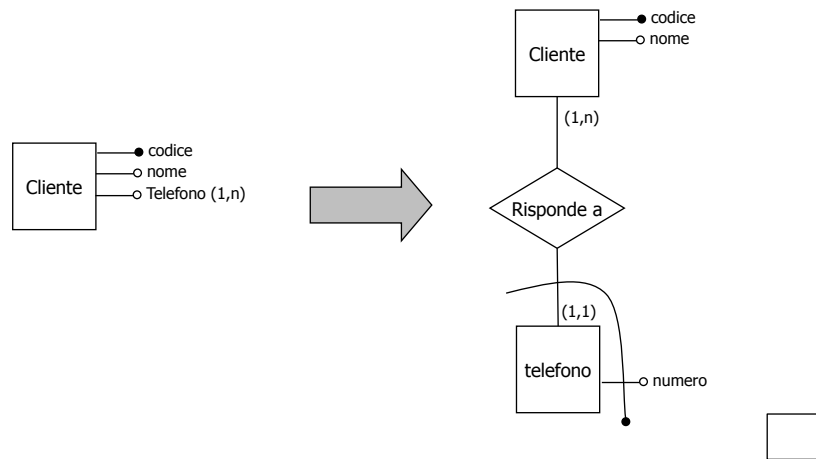
Eliminare gli attributi composti



Eliminare gli attributi multivalore

- ❑ Definizione di una nuova entità collegata all'entità di partenza
- ❑ L'attributo multi-valore è rappresentato con un attributo mono-valore
- ❑ Vincoli di cardinalità:
 - per l'entità che conteneva prima della ristrutturazione l'attributo multi-valore, coincide con il vincolo di cardinalità dell'attributo multi-valore (1:n)
 - per la nuova entità = 1:1

Eliminare gli attributi multivalore



Informatica applicata alla comunicazione
multimediale -

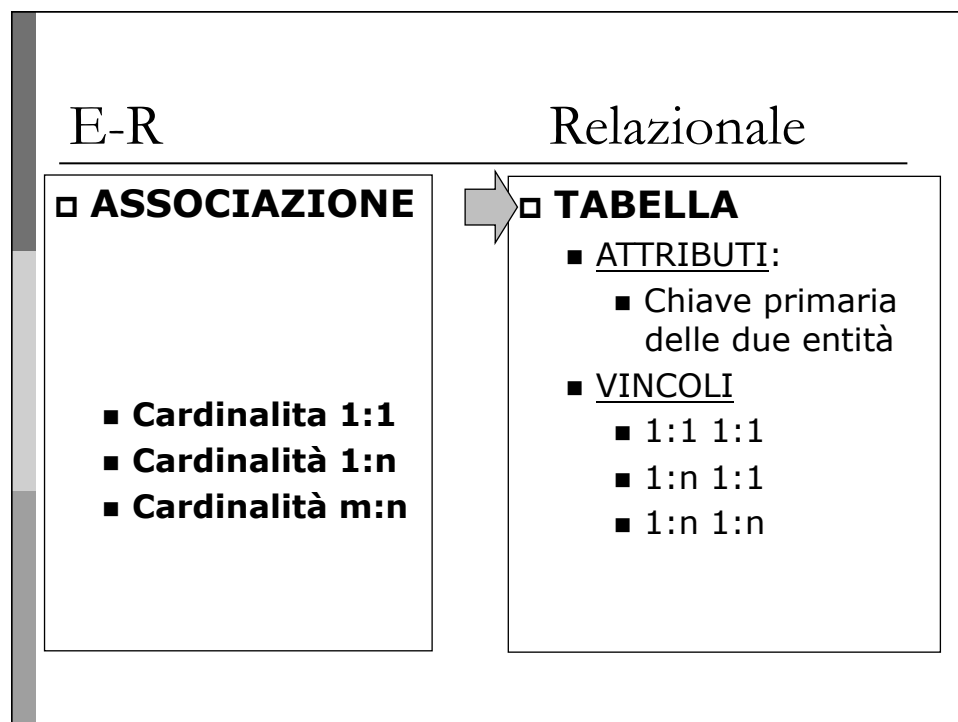
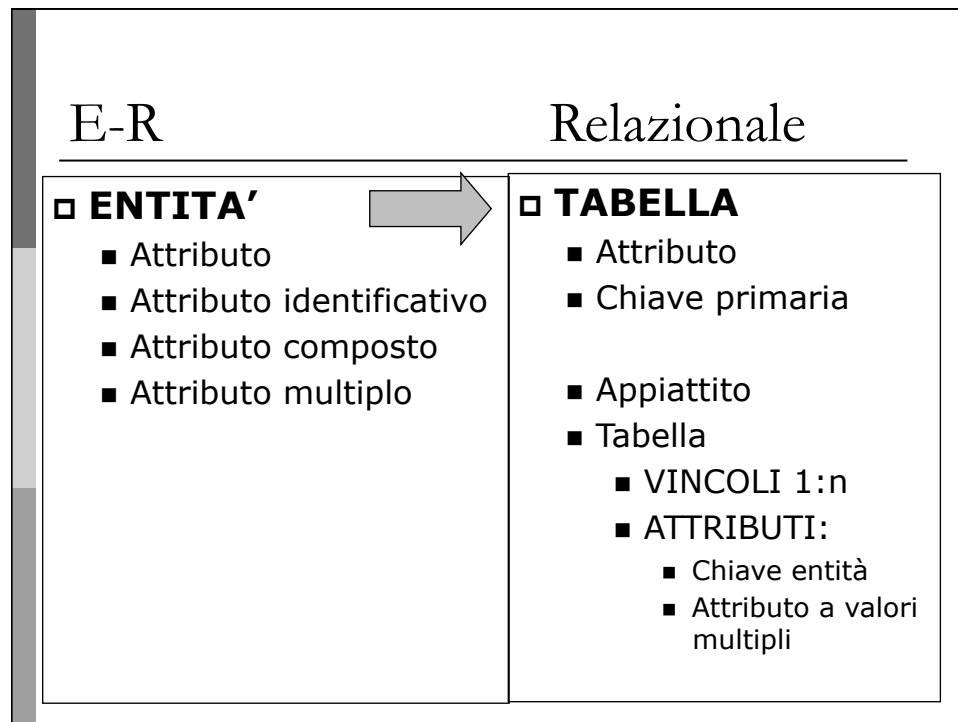
19

Riepilogo

Da Schema E-R  a Modello relazionale

Informatica applicata alla comunicazione
multimediale -

20



Algebra relazionale

Algebra relazionale

- ▣ Il modello logico Relazionale si fonda sull'algebra relazionale
- ▣ Permette di definire formalmente le proprietà delle basi di dati relazionali
- ▣ Permette di calcolare il risultato delle interrogazioni

Algebra relazionale

▣ Due operazioni fondamentali:

- Selezione: estrarre un sottoinsieme delle ennuple (righe) di una relazione (tabella)
[simbolo: σ] (taglio in orizzontale)
- Proiezione: estrarre dalla tabella un insieme di colonne (taglio in verticale)
[simbolo: π]

Operazioni insiemistiche

▣ Vedendo ogni tabella come un insieme è facile definire le seguenti operazioni

- Unione
- Intersezione
- Differenza
- Prodotto cartesiano

Compatibilità all'unione

- Stesso tipo di ennuple:
 - Stesso grado
 - Ogni coppia di attributi corrispondenti ha lo stesso dominio
- Necessaria per unione, intersezione e differenza

Compatibilità: esempio

LIBRI BIBLIOTECA

TITOLO	ISBN	AUTORE	COLLOCAZ.
stringa	stringa	stringa	intero

LIBRI PRESTITO

TITOLO_LIBRO	#ISBN	AUTORE	#COLL.
stringa	stringa	stringa	intero

Unione

□ $R \cup S$

- Tutte le ennuple della prima relazione e tutte le ennuple della seconda
 - Le duplicazioni sono eliminate

aaa	bbbb	cccc
ddd	bbbb	eee
xxx	yyy	zzz

ddd	bbbb	eee
www	yyy	zzz

aaa	bbbb	cccc
ddd	bbbb	eee
xxx	yyy	zzz
xxx	yyy	zzz

Intersezione

□ $R \cap S$

- Solo le ennuple che sono sia nella prima relazione che nella seconda

aaa	bbbb	cccc
ddd	bbbb	eee
xxx	yyy	zzz

ddd	bbbb	eee
www	yyy	zzz

ddd	bbbb	eee
-----	------	-----

Differenza

- $R - S$
- Solo le ennuple della prima relazione che *non* sono anche nella seconda relazione

aaa	bbbb	cccc
ddd	bbbb	eee
www	yyy	zzz

xxx	bbbb	fff
www	yyy	zzz

aaa	bbbb	cccc
ddd	bbbb	eee

Prodotto cartesiano

- $R \times S$
- Prodotto cartesiano delle ennuple di due relazioni
 - Non serve compatibilità all'unione!

aaa	bbbb	cccc
ddd	bbbb	eee

ddd	bbbb
www	yyy

aaa	bbbb	cccc	ddd	bbbb
ddd	bbbb	eee	www	yyy
aaa	bbbb	cccc	www	yyy
ddd	bbbb	eee	ddd	bbbb

Join

□ $R \bowtie S$

tutte le coppie di ennuple di R e S per cui vale una certa relazione tra un certo attributo di R e un certo attributo di S

□ L'operazione di join si fonda sull'esistenza delle relazioni tra le tabelle

Esempio: Join

$R \bowtie S_{B=E}$

A	B	C
aaa	bbbb	cccc
eee	xxxx	qqq
ddd	www	eee

D	E
ddd	bbbb
rrrr	xxxx

A	B	C	D	E
aaa	bbbb	cccc	ddd	bbbb
eee	xxxx	eee	rrrr	xxxx

Tipi di Join

- ▣ Equijoin: la relazione è una relazione di uguaglianza (come esempio)
- ▣ Join naturale: si elimina la ridondanza eliminando uno dei due attributi (quello di S)

A	B	C	D	E
aaa	bbbb	cccc	ddd	bbbb
eee	xxxx	eee	rrrr	xxxx

Join e selezione

- ▣ L'operazione di Join può essere vista come una sequenza di due operazioni semplici:
 1. Prodotto cartesiano
 2. Selezione delle ennuple per cui vale la condizione di join

A	B	C
aaa	bbbb	cccc
eee	xxxx	qqq
ddd	gggg	eee

D	E
ddd	bbbb
rrrr	xxxx

A	B	C	D	E
aaa	bbbb	cccc	ddd	bbbb
eee	xxxx	qqq	ddd	bbbb
ddd	gggg	eee	ddd	bbbb
aaa	bbbb	cccc	rrrr	xxxx
eee	xxxx	qqq	rrrr	xxxx
ddd	gggg	eee	rrrr	xxxx

A	B	C	D	E
aaa	bbbb	cccc	ddd	bbbb
eee	xxxx	eee	rrrr	xxxx

SQL

Interrogare una base di dati con
SQL

Tipi di operazioni: definire schema e inserire dati

□ Tabelle	<i>Definire tabelle</i>
■ Create	<i>creazione</i>
■ Drop	<i>cancellazione</i>
■ Alter	<i>modifica</i>
□ Ennuple	<i>Manipolare i dati</i>
■ Insert	<i>Inserire</i>
■ Delete	<i>Cancellare</i>
■ Update	<i>Modificare</i>

Esempio: definire una tabella

```
CREATE TABLE libri (
  isbn varchar(15) NOT NULL,
  titolo varchar(50),
  autore varchar(50) NOT NULL,
  pubblicato_da varchar(50) NOT NULL,
  anno year(4) NOT NULL default '0000',
  PRIMARY KEY (isbn)
)
```

Vincoli

- ▣ SQL permette di porre vincoli sullo schema di una base di dati:

```
CREATE TABLE nome_tabella  
(  
    PRIMARY KEY (nome_att_chiave),  
    FOREIGN KEY (nome_att)  
        REFERENCES nome_tabella_target (nome_att)  
    ON DELETE ...  
)
```

- ▣ Si può specificare comportamento che preserva integrità

Inserimento dei dati

- ▣ Per inserire righe [ennuple] in una tabella [relazione] si usa il comando INSERT:

```
INSERT INTO nome_tabella (att1, ..., attn)  
VALUES ('xxx', ... , 'yyy')
```

- ▣ Attenzione nell'inserimento: non è detto che il DSMS faccia rispettare i vincoli di integrità

Esempio

```
INSERT INTO `libri` ( `isbn` , `titolo` ,  
  `autore` , `pubblicato_da` , `anno` )  
VALUES ('ISBN 88-7192-22', 'Sistemi di  
basi di dati', 'R.E. Elmasri, S. B. Navathe',  
'Pearson', '1999');
```

Modificare dati

- ▣ Per modificare una riga (ennupla) di una tabella (relazione) si usa il comando UPDATE:

```
UPDATE nome_tabella  
  SET attr1 = xxx, ..., attn = yyy  
  WHERE condizione
```

```
UPDATE `libri` SET `autore` = 'R.E. Elmasri e  
S. B. Navathe' WHERE `isbn` = 'ISBN  
88-7192-22'
```

Operazioni di selezione

- ▣ Qui entra in gioco l'algebra relazionale
- ▣ Struttura della selezione in 3 clausole:

SELECT	attributi da includere
FROM	tabella da cui prenderli
WHERE	condizione

SELECT e algebra relazionale

- ▣ Il comando SELECT effettua una selezione delle righe
 - di una certa tabella (FROM)
 - in base a una condizione (WHERE)
 - proietta il risultato in base agli attributi indicati in SELECT
- ▣ Il risultato è una tabella, ma può avere righe duplicate!

In pratica:

FROM ...

In pratica: selezione

WHERE ...

Righe che
soddisfano la
condizione

In pratica: proiezione

SELECT ...

Attributi da
includere

Righe che
soddisfano la
condizione

Il comando SELECT

- ▣ Per porre la condizione in base alla quale si selezionano gli operatori si usano operatori di confronto

```
SELECT titolo, autore
FROM libro
WHERE titolo = "I promessi sposi"
```

Operatori di confronto

- ▣ Qui il confronto avviene con un numero, non con una stringa

```
SELECT titolo, autore  
FROM libro  
WHERE numero_pagine > 300
```

Seleziona i libri con più di 300 pagine

Sfruttare le relazioni tra tabelle

- ▣ E' possibile recuperare dati da più tabelle contemporaneamente basandosi sulle relazioni tra di esse (operazione JOIN)

```
SELECT Autore, Titolo, Nome_Editore  
FROM Libri, Case_Editrici  
WHERE Editore = Nome_Editore
```

- ▣ Confronta le due tabelle sulla base della relazione tra la colonna Editore di Libri e Nome_Editore di Case_Editrici

Condizione di Join

- ▣ Attraverso la clausola WHERE si può porre una condizione di join

```
SELECT Autore, Titolo, Nome_Editore  
FROM Libri, Case_Editrici  
WHERE Editore = Nome_Editore  
AND Luogo = "Roma"
```

- ▣ Estrae solo i libri pubblicati a Roma

Tabella.attributo

- ▣ Cosa succede se in tabelle diverse ci sono campi con lo stesso nome?

```
SELECT Autore, Titolo, Editore  
FROM Libri, Case_Editrici  
WHERE Libri.Editore = Case_Editrici.Editore  
AND Luogo = "Roma"
```

Attento con quel prodotto cartesiano

- ❑ L'operazione di prodotto cartesiano è rischiosa:
 - Determina come risultato tabelle molto grandi (esplosione combinatoria)
 - Si ha involontariamente in presenza di un join la cui condizione vale sempre

Order by

- ❑ La clausola ORDER BY, aggiunta a una SELECT, permette di ordinare il risultato in base a un attributo

```
SELECT Autore, Titolo, Nome_Editore, Data  
FROM Libri, Case_Editrici  
WHERE Editore = Nome_Editore  
ORDER BY Data
```

- ❑ Ordina il risultato in base alla data