

# Basi di Dati - Appunti lezioni

Sara Angeretti

@Sara1798

2022/2023

# Indice

<b>1</b>	<b>Introduzione al corso</b>	<b>7</b>
1.1	I professori . . . . .	7
1.2	Organizzazione del corso . . . . .	7
1.3	Organizzazione degli orari . . . . .	7
1.4	Organizzazione degli esami . . . . .	8
1.4.1	Esoneri . . . . .	8
1.4.2	Totale . . . . .	8
1.4.3	Laboratorio . . . . .	8
1.4.4	Valutazione . . . . .	8
1.5	Il corso . . . . .	9
1.5.1	Il programma . . . . .	9
1.5.2	Cosa vedremo . . . . .	9
<b>2</b>	<b>Basi di Dati</b>	<b>10</b>
2.1	Introduzione e definizioni . . . . .	10
2.1.1	Risorse . . . . .	10
2.1.2	Basi di Dati . . . . .	10
2.2	Sistemi informativi e sistemi informatici - una premessa . . . . .	10
2.2.1	Sistema Informatico . . . . .	12
2.3	Gestione delle informazioni . . . . .	12
2.4	Informazioni e dati . . . . .	13
2.5	Basi di Dati . . . . .	14
2.5.1	Altra definizione . . . . .	14
2.5.2	caratteristiche . . . . .	14
2.5.3	Perché studiare le basi di dati? . . . . .	15
2.5.4	Basi di dati multimediali . . . . .	15
2.5.5	Data Base Management System - DBMS . . . . .	15
2.5.6	Creazione di un database . . . . .	15
2.5.7	Interrogazione di un database . . . . .	15
2.6	Schemi e istanze . . . . .	16

<i>INDICE</i>	3
<b>3 Argomento 2</b>	<b>17</b>
<b>4 Previsione I parziale</b>	<b>18</b>
4.1 Modello dei dati relazionale . . . . .	18
<b>5 DBMS</b>	<b>19</b>
5.1 Linguaggi per basi di dati . . . . .	19
5.1.1 Pagine web statiche/dinamiche . . . . .	19
5.2 Personaggi ed interpreti . . . . .	20
5.3 Un esercizio per ripassare quanto detto . . . . .	20
5.3.1 Es1 . . . . .	20
5.3.2 Es2 . . . . .	21
5.3.3 Es3 . . . . .	21
<b>6 Progettazione di una base di dati</b>	<b>22</b>
6.1 Metodologie e modelli . . . . .	22
6.2 Progettazione . . . . .	24
6.2.1 Fase di progettazione concettuale . . . . .	25
6.2.2 Fase di progettazione logica . . . . .	25
6.2.3 Fase di progettazione fisica . . . . .	25
<b>7 Introduzione al modello Entità-Relazione</b>	<b>26</b>
7.1 Modello Entità-Relazione . . . . .	26
7.2 Entità . . . . .	26
7.2.1 Formalismi . . . . .	27
7.2.2 Occorrenza (o istanza) di entità . . . . .	28
7.3 Attributi . . . . .	28
7.4 Relazione - Associazione . . . . .	30
7.4.1 Formalismi . . . . .	30
7.4.2 Istanze di associazione . . . . .	31
7.5 Entity-Relationship . . . . .	33
7.5.1 Relazioni ricorsive . . . . .	34
7.6 Es. . . . .	35
7.7 Cardinalità di una relazione . . . . .	37
7.7.1 Classificazione delle relazioni . . . . .	37
7.7.2 Cardinalità degli attributi . . . . .	37
7.8 Identificatori . . . . .	37
7.9 Riassunto: Identificatori . . . . .	38
7.10 Ereditarietà . . . . .	38
7.10.1 Relazione IS-A (o IS-A) tra entità . . . . .	38
7.11 Generalizzazione tra entità . . . . .	39

7.12 esempio sessista e antico . . . . .	40
7.13 Altre proprietà . . . . .	41
7.14 Riassunto finale: tutto quello che abbiamo visto di ER . . . . .	41
<b>8 Progettazione concettuale</b>	<b>42</b>
8.1 Analisi dei dati . . . . .	42
8.1.1 Acquisizione e analisi dei requisiti . . . . .	43
8.2 Scrittura dei requisiti . . . . .	43
8.3 Strutturazione dei requisiti . . . . .	43
8.4 Specifiche sulle operazioni . . . . .	43
8.5 Dalle specifiche al modello ER . . . . .	43
8.6 Design Pattern . . . . .	43
8.6.1 Reificazione . . . . .	43
8.7 Strategie di progetto . . . . .	45
8.8 Esempi di progettazione concettuale dalle slides . . . . .	45
8.8.1 Es1 - Corso . . . . .	45
8.8.2 Es2 - Anagrafe . . . . .	46
8.8.3 Es3 - Campionato . . . . .	46
8.8.4 Es4 - Società di formazione . . . . .	46
<b>9 Modelli - intro</b>	<b>47</b>
9.1 Modello dei dati . . . . .	47
9.1.1 Architettura (semplificata) di un DBMS . . . . .	47
9.1.2 Indipendenza dei dati . . . . .	48
9.1.3 I modelli logici dei dati . . . . .	48
9.1.4 Il modello relazionale . . . . .	49
9.1.5 Termine "relazione" in tre accezioni . . . . .	50
<b>10 Modello Relazionale - MR</b>	<b>51</b>
10.1 . . . . .	51
10.1.1 Prodotto Cartesiano . . . . .	51
10.1.2 Relazione matematica . . . . .	51
10.1.3 Modello basato su valori . . . . .	52
10.1.4 Modello basato su puntatori . . . . .	53
10.1.5 Schema e istanza . . . . .	53
10.1.6 Un po' di notazione . . . . .	53
10.1.7 Tabelle e relazioni . . . . .	53
10.2 Ricapitolazione di quanto visto finora . . . . .	55
10.3 Tipi di vincoli . . . . .	55
10.3.1 Importanza delle chiavi . . . . .	56
10.4 Vincolo di integrità referenziale . . . . .	56

<i>INDICE</i>	5
10.5 Esercizi di esempio . . . . .	57
10.5.1 Es.: prestito di libri . . . . .	57
10.5.2 Altro esempio . . . . .	58
10.5.3 Altro esempio: stazione . . . . .	60
10.5.4 Altro esempio: azienda . . . . .	60
10.5.5 Altro esempio: radio . . . . .	61
10.6 Esempio di esercizio MR che troveremo all'esame . . . . .	62
<b>11 Esercizi di MR</b>	<b>63</b>
11.1 Es. A . . . . .	63
11.2 Es. B . . . . .	65
11.3 Es. C . . . . .	67
11.4 Es. D . . . . .	68
11.5 Es. E . . . . .	69
11.6 Es. F . . . . .	69
11.7 Es. G . . . . .	69
<b>12 Altri es. MR</b>	<b>70</b>
12.1 Es.1 - tavolo cliente . . . . .	70
12.2 Es.2 - gestione magazzino . . . . .	71
12.3 Es.3 - libreria . . . . .	72
<b>13 Algebra Relazionale</b>	<b>74</b>
13.1 Dai modelli ai linguaggi . . . . .	74
13.1.1 Linguaggi per Basi di Dati . . . . .	74
13.1.2 Tipi di linguaggi . . . . .	74
13.1.3 Linguaggi per Basi di Dati Relazionali . . . . .	75
13.2 AR vs SQL . . . . .	75
13.3 Algebra Relazionale . . . . .	76
13.4 Operatori insiemistici . . . . .	76
13.4.1 Le relazioni sono insiemi . . . . .	76
13.4.2 L'operatore Intersezione . . . . .	77
13.4.3 L'operatore Differenza . . . . .	77
13.5 Limiti degli operatori insiemistici . . . . .	77
13.5.1 L'operatore di ridenominazione . . . . .	77
13.6 Select . . . . .	77
13.7 Join . . . . .	77
<b>14 Progettazione Logica</b>	<b>78</b>
14.1 Progettazione Logica . . . . .	79
14.1.1 Obiettivo . . . . .	79

14.1.2 Ristrutturazione schema E-R . . . . .	81
14.1.3 Carico applicativo . . . . .	83
14.1.4 Schema di operazione . . . . .	83
14.1.5 Esempio 1 . . . . .	84
14.1.6 Tavola degli accessi . . . . .	84
14.1.7 Esempio 1 . . . . .	85
14.1.8 Esempio 2 . . . . .	85
14.1.9 Esempio di valutazione di costo . . . . .	85
14.1.10 Attività della ristrutturazione . . . . .	88
14.1.11 Analisi delle ridondanze . . . . .	88
14.1.12 Ridondanze . . . . .	88
14.1.13 Forme di ridondanza in uno schema E-R . . . . .	88

# Capitolo 1

## Introduzione al corso

Turno T1 - AL

Da cancellare, per Elia, turno T1 AL ma ammettono anche MZ alle lezioni.  
L'importante è che gli AL sostengono l'esame con il prof di AL e gli MZ con il prof di MZ.

### 1.1 I professori

- prof. Napoletano, di teoria (aula 1014 in U14).  
Per eventuali comunicazioni, mail: paolo.napoletano@unimib.it, ma **tassativamente** bisogna aggiungere la sigla [DB].
- prof.ssa Damiani, di esercitazione.
- prof. Raganato, di laboratorio.

### 1.2 Organizzazione del corso

- lezioni teoriche ed analisi di casi studio (32 ore, pari a quattro crediti) PN
- esercitazioni ed analisi casi studio (20 ore, pari a due crediti). CD
- Laboratorio / esercitazioni (20 ore, pari a crediti). AR

### 1.3 Organizzazione degli orari

- mercoledì: comincia a 10:30 spaccate

- giovedì: da definire perché c'è una lezione prima (statistica)
- venerdì lez: non ho capito io, ma per ora non li fa (marzo)
- venerdì lab: boh lo diranno.

## 1.4 Organizzazione degli esami

### 1.4.1 Esoneri

- Ci sono parziali, circa seconda metà di aprile (**due** es di progettazione concettuale e modello relazionale) e seconda metà di giugno (**tre** es di SQL, algebra relazionale, progettazione logica).
- Per ora, i parziali sono aperti a tutti, ma il prof Napoletano deve sentire il prof Schettini del turno MZ.

### 1.4.2 Totale

- Tutto il programma (quindi **cinque** es) è previsto per l'esame complessivo.

### 1.4.3 Laboratorio

- A frequenza facoltativa ma essenziale, prevede una prova unica (sempre facoltativa) alla fine del laboratorio che permettere di avere un punteggio  $-1 \leq p \leq 3$  da aggiungere alla media dei parziali o al voto del totale.
- Si lavorerà su MySQL, e prevede la progettazione concettuale e logica di una base di dati assegnata utilizzando lo strumento di Data Modeling fornito da MySQL.
- Il voto del laboratorio rimane valido tutto l'anno accademico, ovvero fino a Febbraio 2024 compreso, e andrà sempre sommato al voto preso.

### 1.4.4 Valutazione

- Per superare l'esame, il voto minimo *per parziale* è di **15/30**.
- Per superare l'esame, la *media dei parziali* deve essere di **18/30**.
- Verrà eventualmente sommato il voto del laboratorio (se sostenuto).

## 1.5 Il corso

Le lezioni non verranno registrate, ma sono disponibili quelle dell'anno precedente.

### 1.5.1 Il programma

1. Introduzione → prof. Napoletano
2. Metodologie e modelli per il progetto delle basi di dati → prof. Napoletano
3. La progettazione concettuale → prof. Napoletano
4. Il modello relazionale (*pausa compitini*) → prof. Napoletano
5. SQL → prof.ssa Damiani
6. Algebra relazionale → prof.ssa Damiani
7. La progettazione logica → prof. Napoletano

L'ordine degli argomenti è diverso da quello suggerito dal libro.

### 1.5.2 Cosa vedremo

Il corso e' dedicato a capire come e' organizzata una base di dati, a cosa serve, come si progetta, come si interroga e si crea.

# Capitolo 2

## Basi di Dati

Una cosa che impareremo sarà **organizzare** il lavoro che ci viene presentato, per esempio da un eventuale cliente. Per prima cosa ci servirà un *foglio dei requisiti*, ciò che dobbiamo progettare e sviluppare. Poi dovremo tradurre il passaggio da input ad output in una schematizzazione o mappa.

### 2.1 Introduzione e definizioni

#### 2.1.1 Risorse

Le risorse di una azienda (o ente, amministrazione):

- persone
- denaro
- materiali
- informazioni

#### 2.1.2 Basi di Dati

Insieme organizzato di dati utilizzati per il supporto allo svolgimento di attività (di un ente, azienda, ufficio, persona).

### 2.2 Sistemi informativi e sistemi informatici - una premessa

Che cos'è l'informatica? Una definizione:

## 2.2. SISTEMI INFORMATIVI E SISTEMI INFORMATICI - UNA PREMESSA 11

- *Scienza del trattamento razionale, specialmente per mezzo di macchine automatiche, dell'informazione, considerata come supporto alla conoscenza umana e alla comunicazione (Academie Francaise).*
- L'informatica ha due anime:
  - metodologica: i metodi per la soluzione di problemi e la gestione delle informazioni;
  - tecnologica: i calcolatori elettronici e i sistemi che li utilizzano;

N.B.: **sistema informativo ≠ sistema informatico**

**Sistema informativo** Componente (sottosistema) di una organizzazione che gestisce le informazioni di interesse (cioè utilizzate per il perseguimento degli scopi dell'organizzazione), le cui funzioni sono:

- acquisizione/memorizzazione
- aggiornamento
- interrogazione
- elaborazione

N.B.: Il concetto di “sistema informativo” è indipendente da qualsiasi automatizzazione!! Anche prima di essere automatizzati, molti sistemi informativi si sono evoluti verso una razionalizzazione e standardizzazione delle procedure e dell'organizzazione delle informazioni.

**Sistema informatico** Porzione automatizzata del sistema informativo: in pratica è la parte del sistema informativo che gestisce le informazioni con tecnologia informatica.

Ma perché le basi di dati sono così importanti? Proviamo a definirle con degli aggettivi o caratteristiche che spieghino come mai sono così interessanti:

- accessibili: le informazioni sono archiviate in modo ordinato;
- capienti: sono storate grandi quantità di dati;
- (facili da modificare;)
- ottimizzate: rapida ricerca delle informazioni; (comune a più sistemi)
- possibilità di raggruppare/filtrare le informazioni e schematizzarle/modellizzarle;
- sicurezza dei dati:;
- facilità di relazione dei dati;

- interfaccia per visualizzare in diversi modi;
- personalizzabili;
- scalabilità (Elia questa me la spieghi);
- interoperabilità: lavorabile con più linguaggi e strumenti;
- accesso concorrente alle informazioni: più persone possono lavorare allo stesso database o alla stessa sottosezione senza andare incontro ad inconsistenza dei dati;
- facilità di gestione delle ridondanze, che aiuta a ridurre al minimo l'inconsistenza dei dati;
- limitazione dell'inconsistenza dei dati: devono sempre essere consistenti, ovvero accessibili solo a chi ha diritto di farlo, il ruolo giusto;

Ma perché non usare una cosa più semplice come un FileSystem invece di un Database? Il primo mi aiuta con un'organizzazione logica, ma i Database sono dotati di strumenti (tipo la progettazione modulare) che sono più efficienti.

### 2.2.1 Sistema Informatico

Gestisce un sistema informativo in modo automatizzato.  
 Garantisce che i dati siano conservati in modo permanente sui dispositivi di memorizzazione.  
 Permette un rapido aggiornamento dei dati per riflettere rapidamente le loro variazioni.  
 Rende i dati accessibili alle interrogazioni degli utenti.  
 Può essere distribuito sul territorio.

## 2.3 Gestione delle informazioni

Parole chiave:

- Raccolta, acquisizione
- Archiviazione, conservazione
- Elaborazione, trasformazione, produzione
- Distribuzione, comunicazione, scambio

Nelle attività umane, le informazioni vengono gestite (registerate e scambiate) in forme diverse:

- idee informali
  - linguaggio naturale (scritto o parlato, formale o colloquiale, in una lingua o in un'altra)
  - disegni, grafici, schemi
  - numeri e codici
- e su vari supporti
- memoria umana, carta, dispositivi elettronici

Nelle attività standardizzate dei sistemi informativi complessi, sono state introdotte col tempo forme di organizzazione e codifica delle informazioni via via più precise (e in un certo senso artificiali).

## 2.4 Informazioni e dati

### Informazioni $\neq$ Dati

Nei sistemi informatici (e non solo), le informazioni vengono rappresentate in modo essenziale, spartano: **attraverso i dati**.

- **informazione**: notizia, dato o elemento che consente di avere conoscenza più o meno esatta di fatti, situazioni, modi di essere.
- **dato**: ciò che è immediatamente presente alla conoscenza, prima di ogni elaborazione; (in informatica) elementi di informazione costituiti da simboli che debbono essere elaborati.  
I dati hanno bisogno di essere interpretati.

*Esempio:*

'Mario' e '275' su un foglio di carta sono due dati.

Se il foglio di carta viene fornito in risposta alla domanda "A chi mi devo rivolgere per il problema X; qual è il suo numero di telefono?", allora i dati possono essere interpretati per fornire informazione e arricchire la conoscenza.

### Ma perché i dati?

La rappresentazione precisa di forme più ricche di informazione e conoscenza è difficile.

I dati costituiscono spesso una risorsa strategica, perché più stabili nel tempo di altre componenti (processi, tecnologie, ruoli umani)

I dati rimangono gli stessi nella *migrazione* da un sistema al successivo.

## 2.5 Basi di Dati

**DB:** Data Base Collezione di dati utilizzati per rappresentare le informazioni di interesse di un sistema informativo.

**DBMS:** Data Base Management System. Sistema software capace di gestire collezioni di dati che siano grandi, condivise e persistenti, assicurando la loro affidabilità e privatezza.

Accezione generica, **metodologica**: insieme organizzato di dati utilizzati per il supporto allo svolgimento delle attività di un ente (azienda, ufficio, persona).

Accezione specifica, **metodologica** e **tecnologica**: insieme di dati gestito da un DBMS.

### 2.5.1 Altra definizione

Possiamo definire una BdD anche come: insieme di archivi in cui ogni dato e' rappresentato logicamente una sola volta e puo' essere utilizzato da un insieme di applicazioni da diversi utenti secondo opportuni criteri di riservatezza.

### 2.5.2 caratteristiche

- i dati sono molti
- i dati hanno un formato definito
- i dati sono permanenti
- i dati sono raggruppati per insiemi omogenei di dati
- esistono relazioni specifiche tra gli insiemi di dati
- la ridondanza è minima e controllata: è assicurata la consistenza delle informazioni
- i dati sono disponibili per utenze diverse e concorrenti (anche contemporanee)
- i dati sono controllati: protetti da malfunzionamenti hardware e software
- indipendenza dei dati dal programma

### 2.5.3 Perché studiare le basi di dati?

Copia.

### 2.5.4 Basi di dati multimediali

Mi sono persa tutto.

### 2.5.5 Data Base Management System - DBMS

Un DBMS è un insieme di programmi che permettono di creare, usare e gestire una base di dati.

Quindi un DBMS è un sistema software general purpose che facilita il processo di definizione, costruzione e manipolazione del database per varie applicazioni.

### 2.5.6 Creazione di un database

Tre fasi:

- definizione
- creazione/popolazione
- manipolazione

### 2.5.7 Interrogazione di un database

ziofrass

```
SELECT [Nome] , [Cognome] , [Indirizzo] ,  
       [Città]  
FROM Studenti  
WHERE [Cognome] = "Rossi";
```

L'efficacia della query dipende da:

- conoscenza del contenuto del db
  - esperienza del linguaggio di interrogazione
- oppure
- semplicità ed efficacia dell'interfaccia di interrogazione

DataBase Management System (DBMS) • Sistema che gestisce collezioni di dati: - grandi - persistenti - condivise garantendo - privatezza - affidabilità - efficienza - efficacia

Hanno grandi dimensioni: dimensioni (molto) maggiori della memoria centrale dei sistemi di calcolo utilizzati; il limite deve essere solo quello fisico dei dispositivi. Sono persistenti: hanno un tempo di vita indipendente dalle singole esecuzioni dei programmi che le utilizzano. Sono condivise: ogni organizzazione (specie se grande) è divisa in settori o comunque svolge diverse attività; ciascun settore/attività ha un (sotto)sistema informativo (non necessariamente disgiunto).

### **MI SONO PERSA UNA QUINDICINA DI SLIDES**

## **2.6 Schemi e istanze**

In ogni base di dati esistono: - lo schema, sostanzialmente invariante nel tempo, che ne descrive la struttura, il significato (aspetto intensionale). • nell'esempio, le intestazioni delle tabelle - l'istanza, i valori attuali, che possono cambiare anche molto rapidamente (aspetto estensionale) • nell'esempio, il "corpo" di ciascuna tabella

lo schema costituisce l'aspetto intensionale, ovvero la descrizione "astratta" delle proprietà, ed è invariante nel tempo. • L'istanza (i valori degli attributi) costituiscono invece l'aspetto estensionale "concreto", che varia nel tempo al variare della situazione di ciò che stiamo descrivendo

Perso altre cinque SLIDES ho rinunciato ma cerca il sito DB engines rankings

# **Capitolo 3**

## **Argomento 2**

# Capitolo 4

## Previsione I parziale

### Quando

Settimana 17-23 aprile.

### Argomenti

**E-R:** abbiamo un testo in linguaggio naturale, dobbiamo creare uno schema concettuale con il linguaggio di modellazione E-R.

**M-R:** abbiamo un piccolo testo, dobbiamo trovarne le chiavi primarie, i vincoli di integrità, i vincoli di dominio.

Di solito è più facile il secondo esercizio, ma è importante fare bene il primo per poi fare il secondo.

### 4.1 Modello dei dati relazionale

Relazione -> teoria degli insiemi -> tramite sql

Due modelli:

- modello a documenti (no-sql)  
si usa come strumento mongodb
- modello a serie temporali (time series)

# Capitolo 5

## DBMS

### 5.1 Linguaggi per basi di dati

Diversi tipi:

**L. per definizione dei dati:** Data Definition Languages - DDL

Si occupano della definizione degli schemi logici, fisici e delle autorizzazioni di accesso.

**L. di manipolazione dei dati:** Data Manipulation Languages - DML

Si occupano dell'interrogazione (**consultazione**) e aggiornamento (**manipolazione**) delle basi di dati.

Alcuni linguaggi come SQL (Structured Query Language) hanno funzioni di entrambe le categorie.

#### 5.1.1 Pagine web statiche/dinamiche

Codice sorgente: linguaggio di *scripting* (che usa dei *tag*) HTML che è un linguaggio statico, che non permette dinamicità dei contenuti. Deve essere interpretato.

Sono nati altri linguaggi (tipo **php**, asp, jsp, etc...) che sono più *dinamici* e riescono a replicare meglio le richieste dell'utente generando contenuti on-demand.

Quando una pagina dinamica deve mostrare i dati, accede ad una base di dati volta per volta per recuperare informazioni da mostrare.

Viene fatta una **query** per **accedere** ad una sezione ben definita di dati (es. informazioni di un utente sul sito Esse3 delle segreterie).

Posso fare una query anche per **manipolare** i dati (es. iscrizione ad un esame).

## 5.2 Personaggi ed interpreti

- **progettisti** e realizzatori di DBMS;
- **progettisti della base di dati** e amministratori della base di dati (DBA);  
Questi dovrebbero avere pieni poteri senza sforare nella "privatezza" dei dati.  
Def. dalle slides: Persona o gruppo di persone responsabile del controllo centralizzato e della gestione del sistema, delle prestazioni, dell'affidabilità, delle autorizzazioni.  
Le funzioni del DBA includono quelle di progettazione, anche se in progetti complessi ci possono essere distinzioni.
- **progettisti** e programmatore **di applicazioni**;
- **utenti**:
  - utenti finali (terminalisti): eseguono applicazioni predefinite (transazioni)
  - utenti casuali: eseguono operazioni non previste a priori, usando linguaggi interattivi

## 5.3 Un esercizio per ripassare quanto detto

### 5.3.1 Es1

Quali delle seguenti affermazioni sono vere?

- l'indipendenza dei dati permette di scrivere programmi senza conoscere le strutture fisiche dei dati  
**VERO**
- l'indipendenza dei dati permette di modificare le strutture fisiche dei dati senza dover modificare i programmi che accedono alla base di dati  
**VERO**
- l'indipendenza dei dati permette di formulare interrogazioni senza conoscere le strutture fisiche  
**VERO**

### 5.3.2 Es2

Quali delle seguenti affermazioni sono vere?

- il fatto che le basi di dati siano condivise permette di ridurre ridondanze e inconsistenze
- il fatto che le basi di dati siano persistenti ne garantisce l'affidabilità  
**VERO**
- il fatto che le basi di dati siano condivise rende necessaria la gestione della privatezza e delle autorizzazioni  
**VERO**

### 5.3.3 Es3

Quali delle seguenti affermazioni sono vere?

- le istruzioni DML permettono di interrogare la base di dati ma non di modificarla  
**VERO**
- le istruzioni DDL permettono di specificare la struttura della base di dati ma non di modificarla
- – non esistono linguaggi che includono sia istruzioni DDL sia istruzioni DML – SQL include istruzioni DML e DDL – le istruzioni DML permettono di interrogare la base di dati e di modificarla  
**VERO**

da sistemare

Gli esercizi che seguono sono utili per fissare i concetti, ma non ai fini dell'esercizio dell'esame.

Cambia il pacco di slides.

# Capitolo 6

## Progettazione di una base di dati

Atzeni, Ceri, Paraboschi, Torlone, capitoli 6 e 7.

In questa parte del corso studieremo come progettare una base di dati...

Verra' illustrato ed esemplificato il processo di progettazione concettuale e logica delle basi di dati relazionali, che permette, partendo dai requisiti di utente, di arrivare a produrre strutture di basi di dati di buona qualità.

La progettazione di basi di dati è una delle attività del processo di sviluppo dei sistemi informativi va quindi inquadrata in un contesto più generale: il ciclo di vita dei sistemi informativi

### 6.1 Metodologie e modelli

Un concetto importante di un'applicazione è **il ciclo di vita**.

**Def.:** insieme e sequenzializzazione delle attività svolte da analisti, progettisti, utenti, nello sviluppo e nell'uso dei sistemi informativi.

Attività iterativa, quindi "un ciclo".



**Studio di fattibilità:** il progettista ha a che fare col committente (cliente)

**Raccolta e analisi dei requisiti:** viene prodotto un documento con i requisiti

**Progettazione:** ER -i PL (?)

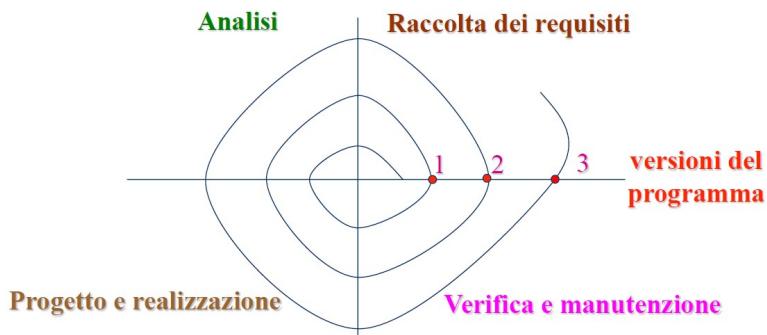
**Implementazione:** si comincia

**Validazione e collaudo:** di volta in volta, ci si confronta con l'utente o si controlla almeno di star seguendo i requisiti

**Funzionamento:** non ho capito ma credo sia la fase finale, quindi la consegna del progetto finito. GUARDA LA SLIDE 8

Noi ci concentreremo sulla parte di progettazione, in particolare sulla modellizzazione dei dati.

## Ciclo di vita (modello a spirale)



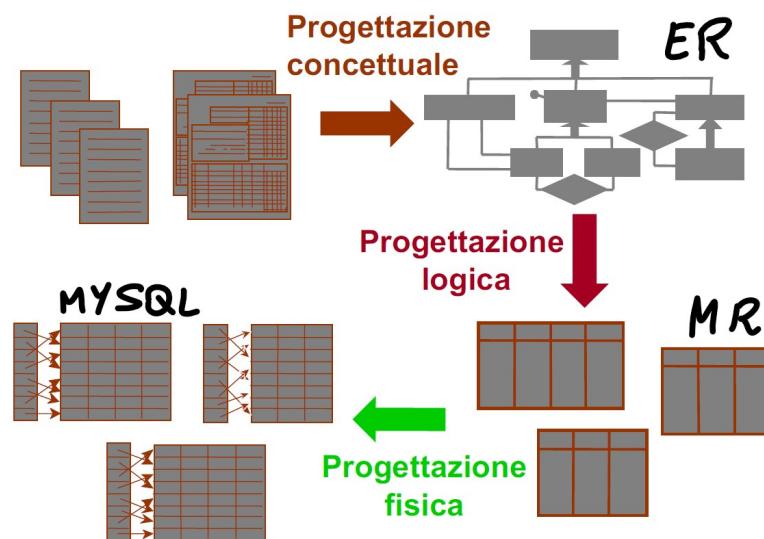
Ha balzato le slide di progettazione ma guardale.

## 6.2 Progettazione

Tre fasi:



A lezione vedremo p. concettuale e p. logica. La p. fisica sarà affrontata a laboratorio.



### 6.2.1 Fase di progettazione concettuale

Progettazione concettuale: traduce i requisiti del sistema informatico in una descrizione formalizzata, integrata delle esigenze aziendali, espressa in modo indipendente dalle scelte implementative (DBMS, SW e HW). formale: la descrizione deve essere espressa con un linguaggio non ambiguo e capace di descrivere in modo soddisfacente il sistema analizzato; integrata: la descrizione deve essere in grado di descrivere nella globalità l'ambiente analizzato; indipendente dall'ambiente tecnologico: la descrizione deve concentrarsi sui dati e sulle loro relazioni, e non sulle scelte implementative.

### 6.2.2 Fase di progettazione logica

La progettazione logica consiste nella traduzione dello schema concettuale nel modello dei dati del DBMS Il risultato è uno schema logico, espresso nel DDL del DBMS In questa fase si considerano anche aspetti legati ai vincoli ed all'efficienza La progettazione logica si articola in due sotto-fasi:  
•ristrutturazione dello schema concettuale •traduzione verso il modello logico

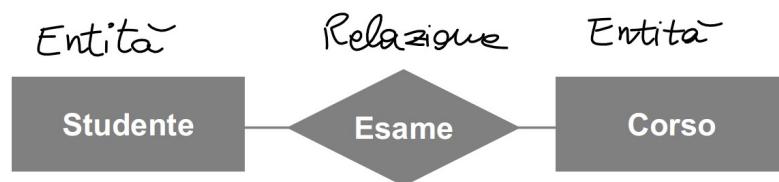
### 6.2.3 Fase di progettazione fisica

Progettazione concettuale: traduce i requisiti del sistema informatico in una descrizione formale, integrata e indipendente dalle scelte implementative (DBMS, SW e HW). Progettazione logica: traduce lo schema concettuale nel modello di rappresentazione dei dati adattato dal DBMS scelto Progettazione fisica: completa lo schema logico ottenuto con le specifiche proprie dell'hw/sw scelto. Il risultato e' lo schema fisico che descrive le strutture di memorizzazione ed accesso ai dati

# Capitolo 7

## Introduzione al modello Entità-Relazione

### 7.1 Modello Entità-Relazione



**Uno schema E-R, graficamente**

Il modello ENTITÀ-RELAZIONE (E-R) è un linguaggio grafico semi-formale per la rappresentazione di schemi concettuali • Il modello E-R si è ormai affermato come uno standard nelle metodologie di progetto e nei sistemi SW di ausilio alla progettazione • Ne esistono molte versioni, (più o meno) diverse l'una dall'altra. • Entity-Relationship, P.P. Chen 1976

È una sottosezione di UML

### 7.2 Entità

**Def.:** Classe di oggetti (fatti, persone, cose) della applicazione di interesse con proprietà comuni e con esistenza “autonoma” e della quale si vogliono registrare fatti specifici.

*Esempi:*

- impiegato

- dipartimento
- città
- conto corrente
- università
- studente

L'essere autonoma di un'entità è un concetto fondamentale: es. gli studenti sono entità, gli esami sostenuti no perché non hanno un'esistenza autonoma. È importante imparare, mentre stiamo leggendo un testo in linguaggio naturale, a capire chi può essere un'entità.

### 7.2.1 Formalismi

Si rappresentano con un rettangolo:



Ogni entità ha un nome che la identifica univocamente nello schema:

- nomi espressivi
- opportune convenzioni (singolare)

A livello estensionale un'entità è costituita da un insieme di oggetti, che sono chiamati le sue istanze.

Ciò significa che, se in uno schema  $S$  è definita una entità  $E$ , in ogni istanza  $I$  dello schema  $S$ , alla entità  $E$  è associato un insieme di oggetti (che viene denotato  $istanze(I, E))e_1, e_2, e_3, \dots, e_n$  (si chiama parte estensionale) che viene detto anche l'estensione di  $E$  nella istanza  $I$  dello schema  $S$ .

Una istanza di entità non è un valore che identifica un oggetto, ma è l'oggetto stesso.

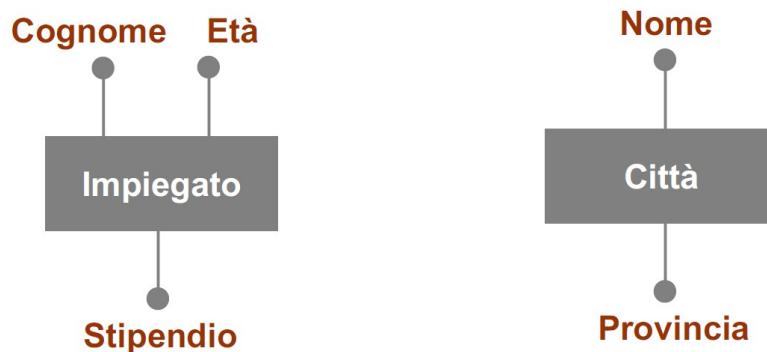
### 7.2.2 Occorrenza (o istanza) di entità

- oggetto della classe che l'entità rappresenta
- Nello schema concettuale rappresentiamo le entità, non le singole istanze (“astrazione”) Quindi: CONOSCENZA ASTRATTA -> entità CONOSCENZA CONCRETA -> istanza di entità

## 7.3 Attributi

Un attributo di entità è una proprietà locale di un'entità, di interesse ai fini dell'applicazione

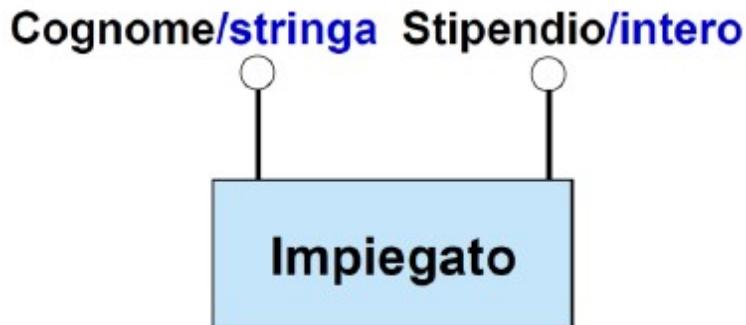
- Un attributo associa ad ogni istanza di entità un valore appartenente ad un insieme detto dominio dell'attributo (tipicamente, interi, caratteri, stringhe, ecc.)
- Si definisce un attributo per l'entità E quando si vuole rappresentare una proprietà locale delle istanze dell'entità E. Una proprietà di un oggetto si dice locale quando in ogni istanza dello schema il valore di tale proprietà dipende solamente dall'oggetto stesso, e non ha alcun rapporto con altri elementi dell'istanza dello schema



Ogni attributo di entità ha un nome che lo identifica in modo univoco nell'ambito della entità, ed è rappresentato da un cerchio collegato alla entità a cui appartiene.

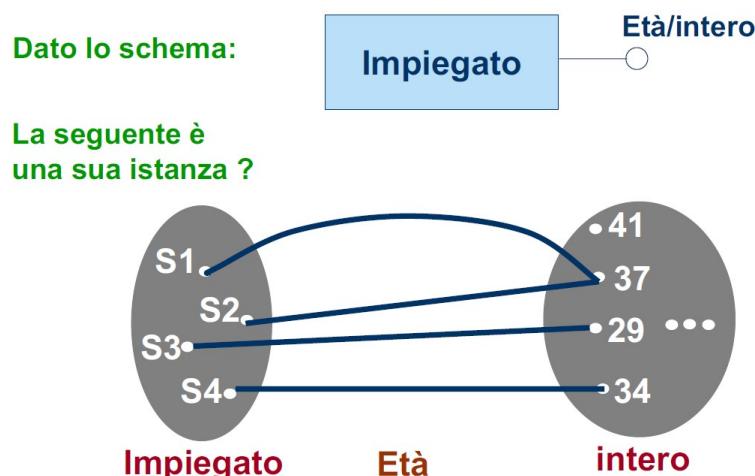


Ogni attributo e' definito su un dominio di valori. Un attributo associa ad ogni istanza di entità o associazione un valore nel corrispondente dominio. I domini solitamente non vengono specificati nell'E-R ma nella documentazione associata. Se li si vuole indicare, la notazione e' la seguente



Una entità può avere o no attributi.

#### Attributi di entità



Un'entità non può avere più di un valore per attributo. E non può averne neanche meno (oddio, veramente cambia in base al contesto, se ci fossero attributi facoltativi avrei dei **nulli**).

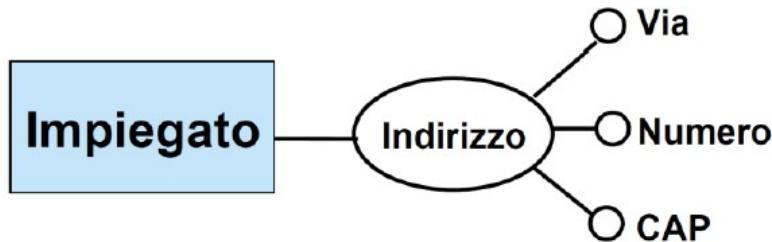
Gli attributi hanno un dominio, che ci specifica di che tipo di dati stiamo parlando.

#### Attributi nulli

I *nulli* sono un concetto fondamentale nelle basi di dati.

### Attributi composti

Si ottengono raggruppando attributi di una medesima entità o relazione che presentano affinità nel loro significato o uso.



## 7.4 Relazione - Associazione

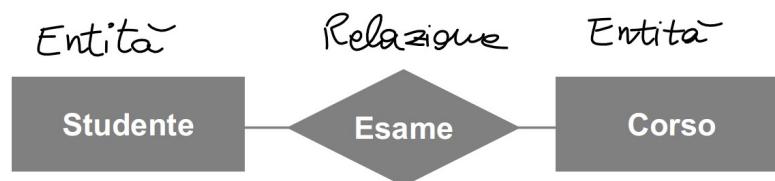
Fatto che descrive un'azione o una situazione e che stabilisce legami logici tra istanze di entità (associa, mette in relazione) nella realtà che stiamo considerando.

I legami possono essere fra piu' di due entità. Il numero di entità coinvolte in una relazione determina il suo grado (si vedano prossime slide).

NB: spesso useremo il termine ASSOCIAZIONE o RELATIONSHIP (per relazione) evitando confusione con la terminologia relazionale.

### 7.4.1 Formalismi

Si rappresentano con un rombo:



### Uno schema E-R, graficamente

Ogni relazione ha un nome che la identifica univocamente nello schema:

- nomi espressivi
- opportune convenzioni (singolare, sostantivi invece che verbi)

A livello estensionale una relazione R tra le entità E ed F è costituita da un insieme di coppie (x,y), tali che x è una istanza di E, ed y è una istanza di F. Ogni coppia è detta istanza della relazione R • Ciò significa che, se in uno schema S è definita una relazione R sulle entità E ed F, in ogni istanza I dello schema S, alla relazione R è associato un insieme di coppie (denotato da istanze(I,R)) (x<sub>1</sub>, y<sub>1</sub>), (x<sub>2</sub>, y<sub>2</sub>), (x<sub>3</sub>, y<sub>3</sub>), ... che viene detto anche l'estensione di R nella istanza I dello schema S • In altre parole, una relazione nel modello ER è, dal punto di vista della semantica, una relazione matematica. In ogni istanza I dello schema S si ha:

$$\text{istanze}(I, R) \subseteq \text{istanze}(I, E) \times \text{istanze}(I, F)$$

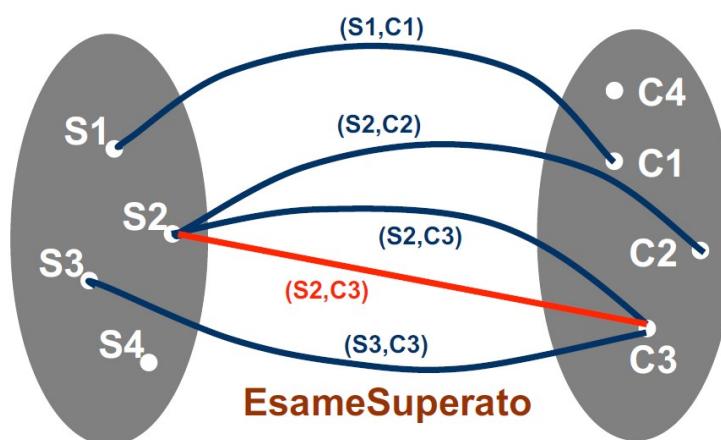
#### 7.4.2 Istanze di associazione

**Def.:** combinazione (aggregazione) di istanze di entità che prendono parte alla associazione.

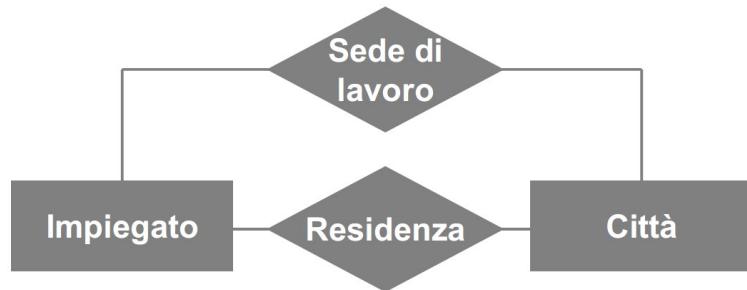
Es.: •Rossi insegna Basi di dati •Batini appartiene all'Università di Milano Bicocca •La ditta Rossi ordina PC •Bianchi lavora al magazzino 4 •Il tornio K22 è installato nell'officina 37 •il TIR 542 viaggia sulla tratta NA-MI

Es.:

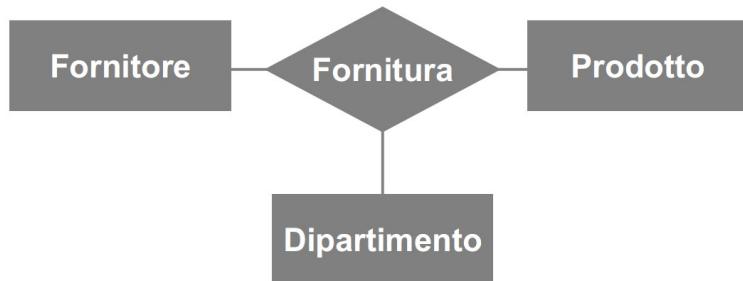
Dalla semantica delle relazioni segue immediatamente che non possono esistere due istanze della stessa relazione che coinvolgono le stesse istanze di entità.



N.B.: due entità possono essere coinvolte in più relationship.

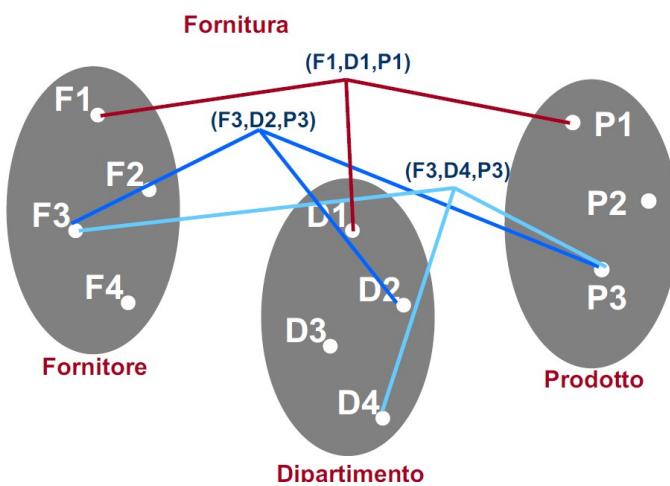


Le relazioni possono coinvolgere più di due entità.



Relazioni n-arie

## Relazione n-aria a livello estensionale



A livello estensionale (ovvero in ogni istanza I dello schema S) una relazione R tra le entità E<sub>1</sub>, E<sub>2</sub>, ..., E<sub>n</sub> è costituita da un insieme di n-ples (o tuple) (x<sub>1</sub>, x<sub>2</sub>, ..., x<sub>n</sub>), tali che x<sub>1</sub> è una istanza di E<sub>1</sub> in I, x<sub>2</sub> è una istanza di E<sub>2</sub> in

$I, \dots, x_n$  è una istanza di  $E_n$  in  $I$ . Ogni  $n$ -pla è detta istanza della relazione  $R$  nella istanza  $I$  dello schema  $S$  • Quindi, in ogni istanza  $I$  dello schema si ha:

$$\text{istanze}(I, R) \subseteq \text{istanze}(I, E_1) \times \dots \times \text{istanze}(I, E_n)$$

Una relazione può avere o no attributi.

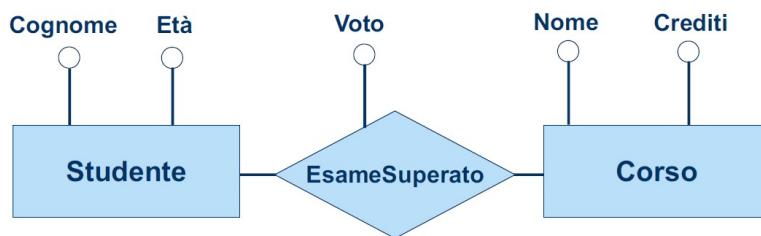
### Attributi di relazione

Un attributo di relazione è una proprietà locale di una relazione, di interesse ai fini dell'applicazione • Un attributo della relazione  $R$  tra le entità  $E_1, E_2, \dots, E_n$  modella una proprietà non di  $E_1$ , non di  $E_2, \dots$ , non di  $E_n$ , ma del legame tra  $E_1, E_2, \dots, E_n$  rappresentato da  $R$  • Un attributo associa ad ogni istanza di relazione un valore appartenente ad un insieme detto dominio dell'attributo

### Formalismi

Ogni attributo di relazione ha un nome che lo identifica in modo univoco nell'ambito della relazione, ed è rappresentato da un cerchio collegato alla relazione a cui appartiene.

#### Esempio



Ho perso un po' di slides

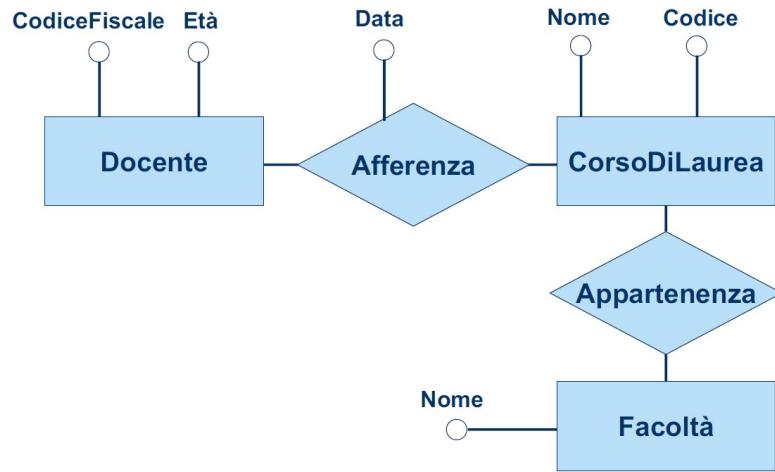
## 7.5 Entity-Relationship

BLOB: Binary Long Object. Sono i pallini pieni della rappresentazione degli attributi di entità.

cose

Es. Descrivere lo schema concettuale della seguente realtà: I docenti hanno un codice fiscale ed una età. I docenti operano nei corsi di laurea

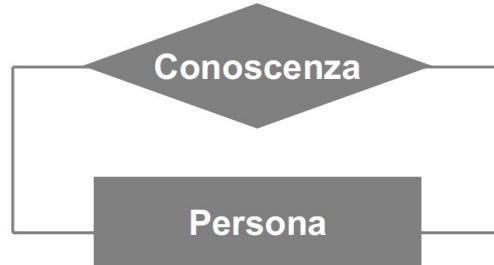
(si dice che afferiscono ai corsi di laurea). Interessa la data di afferenza dei docenti ai corsi di laurea. I corsi di laurea hanno un codice ed un nome, ed appartengono alle facoltà. Ogni facoltà ha un nome.



### 7.5.1 Relazioni ricorsive

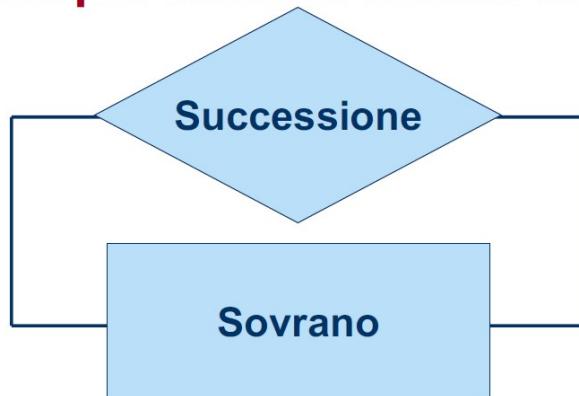
Una associazione può coinvolgere “due o più volte” la stessa entità (associazione ricorsiva o ad anello). Questo ovviamente la fa classificare come relazione **binaria**, anche se c’è rappresentata una sola entità noi stiamo facendo una relazione tra due istanze di quell’entità, quindi conta come due.

**Una associazione può coinvolgere “due o  
piu’ volte” la stessa entità  
(associazione ricorsiva o ad anello)**



Ma siamo in grado da questo grafico di capire ”chi viene prima e chi viene dopo”? No, se non operiamo sulla relazione: questo lo si fa aggiungendo dei ”ruoli” alla relazione.

## Una relazione può coinvolgere due o più volte la stessa entità



Problema: in una istanza di questo schema, data una coppia che è istanza di “Successione”, non si può individuare chi è il sovrano predecessore e chi il sovrano successore.

### I ruoli

Nelle relazioni dove una stessa entità è coinvolta più volte è necessario aggiungere la specifica dei “ruoli”.

### Proprietà delle rel. ricorsive

Un’associazione ad anello può essere o meno:

- Simmetrica:  $(a, b) \in A \Rightarrow (b, a) \in A$
- Riflessiva:  $(a, a) \in A$
- Transitiva:  $(a, b) \in A, (b, c) \in A \Rightarrow (a, c) \in A$
- L’associazione conoscenza è simmetrica, irriflessiva e intransitiva.

vabbeh mancano slides

ultima cosa vista: esempio dei docenti uno migliore dell’altro

## 7.6 Es.

Oggi faremo degli esercizi, che faccio prima a mano e poi vedo se farli a pc o fare solo riferimento.

## 36 CAPITOLO 7. INTRODUZIONE AL MODELLO ENTITÀ-RELAZIONE

Convenzioni: entità - sostantivo singolare, relazione - verbo singolare.

Slide 82: corsivo entità, grassetto relazioni.

Descrivere lo schema concettuale della seguente realtà:

Degli *impiegati* interessa il codice fiscale, il nome, il cognome, i dipartimenti ai quali **afferiscono** (con la data di afferenza), ed i *progetti* ai quali **partecipano**. Dei *progetti* interessa il nome, il budget, e la città in cui **hanno luogo** le corrispondenti attività. Alcuni progetti sono parti di altri progetti, e sono detti loro **sottoprogetti**. Dei *dipartimenti* interessa il nome, il numero di telefono, gli impiegati che li **dirigono**, e la *città* dove è **localizzata** la sede. Delle città interessa il nome e la regione.

Entità:

- impiegato
- progetto
- dipartimento

Relazioni:

- afferire
- partecipare
- avere luogo
- fare parte/sottoprogetto (ruolo)
- dirigere
- essere localizzato

**Ma domanda: quando un attributo viene classificato come entità?**

Come faccio, leggendo un testo, a capire quando un elemento che mi sembra un attributo possa invece essere effettivamente scritto come entità?

1. è descritto da più di un attributo
2. partecipa a più di una relazione
3. è autonoma

C'è una slide.

Come invece scegliere se entità o relazione?

C'è una slide.

Torniamo alla teoria.

## 7.7 Cardinalità di una relazione

Si indica come  $(x, y)$  dove  $x$  è la cardinalità minima dell'associazione e  $y$  è la cardinalità massima dell'associazione.

Si legge "Entità1 è in relazione con  $x$  Entità2 o al più con  $y$  Entità2".

Simboli standard:

- 0: cardinalità minima, indica partecipazione facoltativa;
- 1: cardinalità minima, indica partecipazione obbligatoria;
- $n$ : cardinalità massima.

### 7.7.1 Classificazione delle relazioni

Praticamente lo decide la cardinalità massima il nome.

- uno a uno:  $(1, 1) < - > (0, 1)$
- va beh
- .

### 7.7.2 Cardinalità degli attributi

È possibile associare delle cardinalità anche agli attributi, per:  
 - indicare opzionalità  
 - indicare attributi multivalore

## 7.8 Identificatori

### I. interni

identificatori interni: pallini pieni identificatori semplici: pallini vuoti

### I. esterni

Non così immediati. Uso un attributo composto, che mi comprende 2+ identificatori: l'identificatore esterno è valido e univoco se e solo se considero l'identificatore in relazione con una sola entità con cui l'entità di cui sto definendo l'attributo è in relazione.

Non posso ovviamente aggiungere attributi a caso per risolvere la situazione perché se il testo non lo prevede, non sono autorizzato.

Es.: in un'università uno studente è identificato univocamente dalla matricola, ma se vado a considerare per esempio tutte le università d'Italia posso avere la stessa matricola per più studenti con nomi e cognomi diversi. Come faccio? Considerando 1 e 1 sola entità di università, matricola diventa di nuovo univoca. Non posso assolutamente ad esempio aggiungere un attributo "codice id" univoco in tutte le università d'Italia, perché la consegna non lo prevede.

## 7.9 Riassunto: Identificatori

L'abbiamo vista la volta scorsa. Fa l'esempio dello studente e dell'università, in cui l'attributo Matricola di Studente è debole perché è sufficiente nel contesto di una sola università, dove è un identificativo univoco. Ma se consideriamo le università di tutt'Italia, non è più univoco.

Serve allora un identificatore esterno.

### Requisiti

Avere una relazione 1 a molti, ovvero  $(1, 1) \dashrightarrow (0, n)$ .

## 7.10 Ereditarietà

### 7.10.1 Relazione IS-A (o IS-A) tra entità

Se avessi una entità studente, con attributi matricola (pallino pieno) che nel contesto di 1 università è univoca quindi sufficiente, nome e cognome.

Fra questi ho particolari istanze chiamate studenti-lavoratori che hanno come attributi inizio-lavoro e sede-lavoro. Questo sottoinsieme (chiamato anche "relazione is-a") si comporta come in programmazione 2. Ovvero:

- studenti-lavoratori eredita tutti gli attributi di studenti
- non è vero il contrario

Ma avremmo potuto anche fare un solo attributo "sede di lavoro" per dire che se c'è, allora è uno studente-lavoratore per evitare tutto il sottoinsieme. Perché non l'abbiamo fatto?

Principalmente per facilità di rappresentazione e lettura del modello. Poi anche per specificare che gli attributi del sottoinsieme sono esclusivi, specifici delle istanze di quel sottoinsieme e non di tutte le istanze di studente.

Tornando alle slides:

- Fino ad ora non abbiamo detto nulla sul fatto se due entità possano o no avere istanze in comune
- È facile verificare che, in molti contesti, può accadere che tra due classi rappresentate da due entità nello schema concettuale sussista la relazione IS-A (o relazione di sottoinsieme), e cioè che ogni istanza di una sia anche istanza dell'altra. (Es. Studente, Studente della laurea breve)
- La relazione IS-A nel modello ER si può definire tra due entità, che si dicono "entità padre" ed "entità figlia" (o sottoentità, cioè quella che rappresenta un sottoinsieme della entità padre) (Es Studente è entità padre di Studente della laurea breve)

Poi ci sono 4-5 slides da sistemare sull'ereditarietà. Ovviamente NB che le relazioni che un eventuale sottoinsieme può avere con altre entità sarà specifica di quel sottoinsieme, così come per gli attributi vale anche per le relazioni.

## 7.11 Generalizzazione tra entità

Dalle slides:

- Finora, abbiamo considerato la relazione ISA che stabilisce che l'entità padre è più generale della sottoentità. Talvolta, però, l'entità padre può generalizzare diverse sottoentità rispetto ad un unico criterio. In questo caso si parla di generalizzazione.
- Nella generalizzazione, le sottoentità hanno insiemi di istanze disgiunti a coppie (anche se in alcune varianti del modello ER, si può specificare se due sottoentità della stessa entità padre sono disgiunte o no).
- Una generalizzazione può essere di due tipi:
  - Completa (o totale): l'unione delle istanze delle sottoentità è uguale all'insieme delle istanze dell'entità padre. Si rappresenta con una freccia piena.
  - Non completa (o parziale). Si rappresenta con una freccia non piena.

Parliamo di generalizzazione **esclusiva** quando un'entità è composta da due sottoinsiemi che raggruppano le istanze e nessuna istanza sta fuori da questi due sottoinsiemi (si dice anche completa) e l'intersezione dà l'insieme vuoto.

$$\cup_i E_i = E$$

$$\cap_i E_i = \emptyset$$

---

Se invece avessimo

$$\cup_i E_i = E$$

$$\cap_i E_i = \emptyset$$

Abbiamo 4 casi:

- completa, esclusiva
- completa, non esclusiva
- non completa, esclusiva
- non completa, non esclusiva

Come passo da non esclusiva a esclusiva? Andando a recuperare l'intersezione e rendendola entità a sé.

## 7.12 esempio sessista e antico

- Le persone hanno CF, cognome ed età; gli uomini anche la posizione militare;
- gli impiegati hanno lo stipendio e possono essere segretari, direttori o progettisti (un progettista può essere anche responsabile di progetto);
- gli studenti (che non possono essere impiegati) un numero di matricola;
- esistono persone che non sono né impiegati né studenti (ma i dettagli non ci interessano)

Soluzione: slide successiva

### 7.13 Altre proprietà

possono esistere gerarchie a più livelli e multiple generalizzazioni allo stesso livello • un'entità può essere inclusa in più gerarchie, come genitore e/o come figlia • se una generalizzazione ha solo un'entità figlia si parla di sottoinsieme

### 7.14 Riassunto finale: tutto quello che abbiamo visto di ER

Slide 157 in poi

Esercitazioni 175(immobili) tutto falso 176(città) tutto falso tranne le ultime due

# Capitolo 8

## Progettazione concettuale

Diverse fasi:

### 8.1 Analisi dei dati

Anche nota come "Analisi dei requisiti e progettazione concettuale". Comprende attività (interconnesse) di

- acquisizione dei requisiti
- analisi dei requisiti
- costruzione dello schema concettuale
- costruzione del glossario

#### Requisiti

Possibili fonti:

- utenti, attraverso:
  - interviste
  - documentazione apposita
- documentazione esistente:
  - normative (leggi, regolamenti di settore)
  - regolamenti interni, procedure aziendali
  - realizzazioni preesistenti
- modulistica

### 8.1.1 Acquisizione e analisi dei requisiti

Il reperimento dei requisiti è un'attività difficile e non standardizzabile. L'attività di analisi inizia con i primi requisiti raccolti e spesso indirizza verso altre acquisizioni.

## 8.2 Scrittura dei requisiti

### 8.3 Strutturazione dei requisiti

### 8.4 Specifiche sulle operazioni

### 8.5 Dalle specifiche al modello ER

## 8.6 Design Pattern

Soluzioni progettuali a problemi comuni. Sono largamente usati nell'ingegneria del software.

Vediamo alcuni pattern comuni nella progettazione concettuale di basi di dati.

Parliamo di **reificazione**: procedimento di creazione di un modello di dati basato su un concetto astratto predefinito.

### 8.6.1 Reificazione

#### Part-of

L'unicità è un tema importante: un identificativo unico mi identifica una sola istanza, non ci saranno altre istanze di quell'entità con quell'identificativo.

A volte un entità può essere legata a un'altra entità creando una relazione di tipo 1,N. Gli esempi mostrano come il concetto di part of possa essere di dipendenza (Sala non esiste senza Cinema) o meno (Tecnico è autonomo da Team).

NB: avevamo detto che l'identificatore esterno non è univoco sempre, ma ci serve una situazione univoca (es. della matricola dell'università, univoca in una università ma non tipo in tutta la Lombardia): perciò funziona solo in una relazione 1 a molti.

NBB: se la relazione è 1 a molti, l'entità che ha l'identificatore esterno è quella che ha la cardinalità 1. Grazie, copilot.

NBBB: la cardinalità della relazione si guarda col numerino a destra, ovvero la cardinalità massima. Diversi casi:

- $(*, 1) \dashv (*, n)$ : 1 a molti
- $(*, 1) \dashv (*, 1)$ : 1 a 1
- $(*, n) \dashv (*, 1)$ : molti a 1
- $(*, n) \dashv (*, n)$ : molti a molti

### **Istance-of**

A volte si viene a creare la necessità di creare un'entità astratta che prende concretezza in un'entità istanza. Volo possiede informazioni astratte sul Volo Reale che invece rappresenta il volo che avviene giornalmente. (vedi esempio grafico)

### **Reificazione di relazione binaria**

Nell'esempio degli esami, la relazione diventa un'entità dove l'attributo debole "data" diventa un attributo esterno quando reso univoco tramite un legame con un attributo "matricola" dell'entità "studente", che è univoco (pallino pieno). Questo è rappresentato dal secondo modello. Nell'esempio, non potendo avere ripetizioni, non posso sostenere due volte nello stesso giorno lo stesso esame. Posso due esami diversi. Il secondo modello grafico mi andrebbe bene. Però posso farlo guardando il terzo modello grafico.

### **Nota sull'identificazione esterna**

Non esistono due esami diversi che riguardano la stessa coppia di studente e corso. L'esempio mostra che un identificatore esterno può anche non comprendere attributi, e può coinvolgere una sola relazione attraverso un unico ruolo.

## 8.7 Strategie di progetto

### 8.8 Esempi di progettazione concettuale dalle slides

#### 8.8.1 Es1 - Corso

Cominciamo guardando un esercizio per vedere come approcciarsi allo sviluppo a partire da una specifica del tipo:

Vogliamo memorizzare dati relativi ai **partecipanti** ad un corso.  
Per ciascuno, vogliamo ricordare nome, cognome, data di nascita,  
se è sposato, e, nel caso lo sia, il numero di figli.

Vogliamo, poi ricordare le **città** in cui risiedono e le città in cui sono  
nati, insieme al numero di abitanti. Per le città capoluogo di  
regione, vogliamo ricordare la regione.

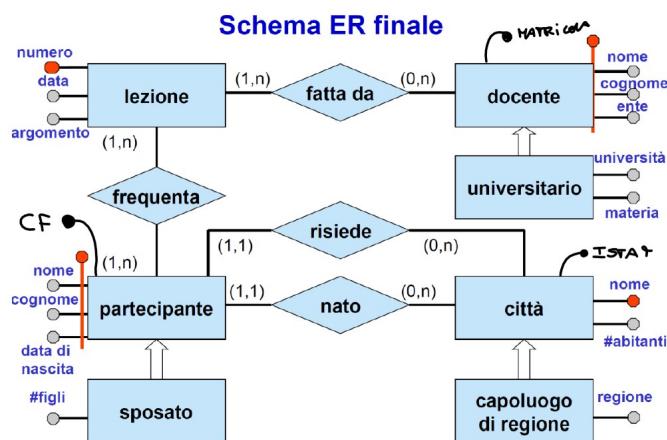
Vogliamo poi sapere le **lezioni** che i partecipanti hanno  
frequentato, con i (o il), **docenti** che le hanno svolte (nome,  
cognome, e tipo di enti di provenienza), il corrispondente  
argomento ed il giorno in cui si sono svolte. Ad ogni lezione va  
associato un numero progressivo.

Relativamente ai docenti provenienti dall'Università, si vuole  
ricordare la Università da cui provengono e la materia che hanno li  
in affidamento.

Le specifiche sono qua colorate in modi diversi per suggerirci le parti che  
possono essere raggruppate in sottoinsiemi.

La strategia usata è quella **top-down**.

NB: la parte "se è sposato, e nel caso lo sia il numero dei figli", ci suggerisce che sia meglio rappresentarlo come generalizzazione piuttosto che  
come attributo. Stesso discorso per quanto riguarda l'entità "città" e la  
generalizzazione che la collega alla sotto entità "capoluogo di regione".



Qua in penna ho aggiunto varianti alla soluzione proposta che dovrebbero essere ugualmente valide.

### 8.8.2 Es2 - Anagrafe

Definire uno schema E-R che descriva i dati di una applicazione relativa all'anagrafe del comune di Chissadove, con cittadini e famiglie. Vanno memorizzate:

- Informazioni sui cittadini nati nel comune e su quelli residenti in esso; ogni cittadino è identificato dal codice fiscale e ha cognome, nome, sesso e data di nascita; inoltre:
  - Per i nati nel comune, sono registrati anche gli estremi di registrazione (numero del registro e pagina)
  - Per i nati in altri comuni, è registrato il comune di nascita
- Informazioni sulle famiglie residenti, ognuna delle quali ha uno e un solo capofamiglia e zero o più membri, per ognuno dei quali è indicato (con la sigla) il grado di parentela (coniuge, figlio, genitore o altro); ogni cittadino residente appartiene ad una e una sola famiglia; tutti i membri di una famiglia hanno lo stesso domicilio (via, numero civico, interno)

Cercare di procedere secondo la strategia inside-out. Al termine, verificare le qualità dello schema ottenuto.

La strategia usata questa volta è quella **inside-out**.

### 8.8.3 Es3 - Campionato

La strategia usata questa volta è quella **top-down**.

### 8.8.4 Es4 - Società di formazione

Riprendiamo quanto visto a inizio lezione. La strategia usata questa volta è quella **top-down**.

# Capitolo 9

## Modelli - intro

### 9.1 Modello dei dati

Un modello dei dati è un insieme di concetti per organizzare i dati e descrivere la struttura.

Questo modello deve essere comprensibile a un elaboratore.

Componente fondamentale di ogni modello sono i meccanismi di strutturazione (analoghi dei costruttori di tipo).

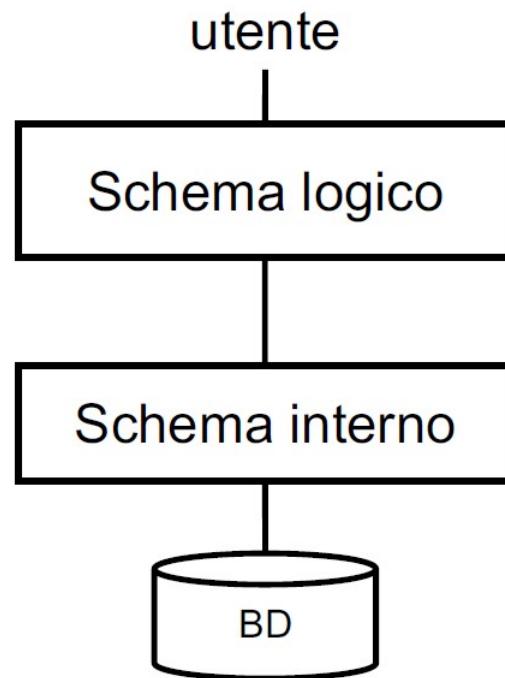
Ogni modello dei dati prevede alcuni costruttori che permettono di definire nuovi tipi sulla base di tipi predefiniti (elementari).

Il modello relazionale è il modello di dati più diffuso.

#### 9.1.1 Architettura (semplificata) di un DBMS

**Schema logico:** descrizione della base di dati nel modello logico adottato (ad esempio, la struttura della tabella).

**Schema interno (o fisico):** rappresentazione dello schema logico per mezzo di strutture di memorizzazione (realizzazione fisica attraverso file; ad esempio, record con puntatori, ordinati in un certo modo)



### 9.1.2 Indipendenza dei dati

Il livello logico è **indipendente** da quello fisico:

- una tabella è utilizzata nello stesso modo qualunque sia la sua realizzazione fisica (che può anche cambiare nel tempo)
- Perciò in questo corso vedremo solo il livello logico e non quello fisico

### 9.1.3 I modelli logici dei dati

Tre modelli logici tradizionali:

**gerarchico:** organizzazione ad albero

**reticolare:** organizzazione a grafo

**relazionale:** organizzazione a tabella

Più recenti (e meno diffusi):

**oggettuale:** organizzazione ad oggetti

**XML**

**NoSQL:** basata su documenti

Si chiamano modelli logici perché pur basandosi su strutture astratte queste riflettono una particolare organizzazione.

<https://db-engines.com/en/ranking>

#### 9.1.4 Il modello relazionale

Permette di definire tipi per mezzo del costruttore **relazione** che permette di organizzare i dati in insiemi di record a *struttura fissa*.

Una relazione è spesso rappresentata da una **tabella** (solo in questo specifico caso usabile come sinonimo di **relazione**):

- le righe: rappresentano specifici record
- le colonne: corrispondono ai campi dei record

L'ordine di righe e colonne è sostanzialmente *irrilevante*.

nome	cognome	Data di nascita	professione	tel
mario	rossi	21/10/80	impiegato	02 345678
sara	bianchi	17/03/77	avvocato	031 45678
marco	verdi	11/11/67	medico	06 789052

In una base di dati relazionale ci sono in generale più relazioni.

#### Un po' di storia

Proposto da E. F. Codd nel 1970 per favorire *l'indipendenza dei dati*.

Disponibile in DBMS reali nel 1981 (non è facile implementare l'indipendenza con efficienza e affidabilità!)

Basato:

- sul concetto matematico di relazione - concetto formale (con una variante)
- tabelle - concetto intuitivo

Definisce come sono organizzati i dati e non come sono poi memorizzati e gestiti dal sistema informatico.

### 9.1.5 Termine "relazione" in tre accezioni

**relazione matematica:** come nella teoria degli insiemi

**relazione:** secondo il modello relazionale dei dati

**associazione o correlazione:** relazione (dall'inglese *relationship*) che rappresenta una classe di fatti, nel modello Entity-Relationship.

# Capitolo 10

## Modello Relazionale - MR

### 10.1

#### 10.1.1 Prodotto Cartesiano

Dati due insiemi  $D_1, D_2$  (Estendibile a  $n$  insiemi distinti o non), definiamo nei seguenti modi:

- Prodotto cartesiano  $D_1 \times D_2$ :  
L'insieme di tutte le coppie **ordinate**  $(v_1, v_2)$  tali che  $v_1 \in D_1$  e  $v_2 \in D_2$
- Esempio

$$A = 1, 2, 4 \text{ e } B = a, b$$

Il prodotto cartesiano  $A \times B$  è composto da:

$$(1, a), (1, b), (2, a), (2, b), (3, a), (3, b),$$

ovvero sei coppie ordinate (il primo da A, il secondo da B) senza ripetizioni.

#### 10.1.2 Relazione matematica

- $D_1, \dots, D_n$  ( $n$  insiemi anche non distinti)
- prodotto cartesiano  $D_1 \times \dots \times D_n$ :  
l'insieme di tutte le  $n$ -uple ordinate  $(d_1, \dots, d_n)$  tali che  $d_1 \in D_1, \dots, d_n \in D_n$
- relazione matematica su  $D_1, \dots, D_n$ :  
un sottoinsieme di  $D_1 \times \dots \times D_n$ , ovvero del prodotto cartesiano.

- $D_1, \dots, D_n$  sono i domini della relazione

Il numero delle componenti del prodotto ( $n$ ) è detto **grado** della relazione; il numero di  $n$ -uple della relazione è la **cardinalità** della relazione.

**Es.:**

**Es. numeri di telefono:**

**Rel. matematica: alcune proprietà**

Una relazione matematica è un insieme di  $n$ -uple ordinate:  $(d_1, \dots, d_n)$  tali che  $d_1 \in D_1, \dots, d_n \in D_n$

Alcune proprietà:

- non c'è ordinamento fra le  $n$ -uple (in verticale);
- le  $n$ -uple sono distinte;
- ciascuna  $n$ -upla è ordinata (in orizzontale): l' $i$ -esimo valore proviene dall' $i$ -esimo dominio.

L'ultima proprietà ci dice che ciascuno dei domini ha due ruoli diversi, distinguibili attraverso la posizione: ovvero, la struttura è **posizionale**. Scambiare la posizione fa variare il risultato, portando quindi a possibili errori.

Questo si può sistemare aggiungendo alla tabella una riga di **intestazione**:

- a ciascun dominio si associa un nome (attributo), che ne descrive il "ruolo";
- gli attributi possono essere usati come intestazione;
- la struttura in questo modo non è più posizionale.

### 10.1.3 Modello basato su valori

Immagine

**Vantaggi:**

Indipendenza dalle strutture fisiche (si potrebbe avere anche con puntatori di alto livello) che possono cambiare dinamicamente • si rappresenta solo ciò che è rilevante dal punto di vista dell'applicazione • i dati sono portabili più facilmente da un sistema ad un altro • per accedere ai dati non serve sapere come sono memorizzati fisicamente

### 10.1.4 Modello basato su puntatori

Immagine

### 10.1.5 Schema e istanza

In ogni base di dati esistono:

- lo schema, sostanzialmente invariante nel tempo, che ne descrive la struttura (aspetto intensionale). Scritto con la lettera maiuscola.
- es.: le intestazioni delle tabelle
- l'istanza, i valori attuali, che possono cambiare anche molto rapidamente (aspetto estensionale). Scritto con la lettera minuscola.
- es.: il “corpo” di ciascuna tabella

#### Definizioni

Schema di relazione: un nome  $R1$  (MAIUSCOLO) con un insieme di attributi  $X=A1, \dots, An$ :  $R1(A1, \dots, An) = R1(X)$

Schema di base di dati: insieme di schemi di relazione con nomi diversi:  $R = R1(X1), \dots, Rk(Xk)$

Una tupla su un insieme di attributi  $X$  è una funzione  $t$  che associa a ciascun attributo  $A$  in  $X$  un valore del dominio di  $A$

- $t[A]$  (o  $t.A$ ) denota il valore della tupla  $t$  sull'attributo  $A$
- una relazione  $R$  con attributi  $A1, A2$  e  $A3$ , e corrispondenti domini  $D1, D2$  e  $D3$  si può denotare anche come  $R(A1 : D1, A2 : D2, A3 : D3)$

(Istanza di) relazione su uno schema  $R(X)$ : insieme  $r1$  (minuscolo) di tuple su  $X$

- (Istanza di) base di dati su uno schema  $R = R1(X1), \dots, Rn(Xn)$ : insieme di relazioni  $r = r1, \dots, rn$  (con  $ri$  relazione su  $Ri$ )

### 10.1.6 Un po' di notazione

### 10.1.7 Tabelle e relazioni

Una tabella rappresenta una relazione se

- i valori di ogni colonna sono fra loro omogenei
- le righe sono diverse fra loro
- le intestazioni delle colonne sono diverse tra loro

In una tabella che rappresenta una relazione

- l'ordinamento tra le righe è irrilevante
- l'ordinamento tra le colonne è irrilevante

### Chiave

insieme di attributi che identificano univocamente le tuple di una relazione  
 Formalmente:

- un insieme  $K$  di attributi è superchiave per  $r$  se  $r$  non contiene due tuple distinte  $t_1$  e  $t_2$  con  $t_1[K] = t_2[K]$
- $K$  è chiave per  $r$  se è una superchiave minimale per  $r$  (cioé non contiene un'altra superchiave)

Ma data una tabella qualsiasi, esiste sempre una superchiave (**NON** minimale)?  
 Dato che la tabella proviene da un prodotto cartesiano e nel prodotto cartesiano non ci sono ripetizioni, allora possiamo dire che la risposta è sì. E la superchiave corrisponderà all'insieme di **tutti** gli attributi della tabella.  
 All'esame ci verranno dati schemi di relazione, dobbiamo individuarne la superchiave e i vincoli.

### Informazione incompleta

Il modello relazionale impone ai dati una struttura rigida:

- le informazioni sono rappresentate per mezzo di tuple
- solo alcuni formati di tuple sono ammessi: quelli che corrispondono agli schemi di relazione

I dati disponibili possono non corrispondere al formato previsto:

- Hanno attributi in più che non sono nello schema
- Mancano di alcuni attributi = Informazione incompleta

### Informazione incompleta: motivazioni

#### Informazione incompleta nel modello relazionale

Si adotta una tecnica rudimentale ma efficace:

- valore nullo: denota l'assenza di un valore del dominio (e non è un valore del dominio)
- $t[A]$ , per ogni attributo  $A$ , è un valore del dominio  $\text{dom}(A)$  oppure il valore nullo NULL e altre cose

tutti questi vincoli ci aiutano a gestire la bd in modo che i dati non siano incosistenti.

tipi di vincoli:

per la slide 61, all'esame va bene così com'è nella slide oppure in linguaggio naturale

## 10.2 Ricapitolazione di quanto visto finora

### Modello relazionale

**Concetto relazione come variazione** (attributi, campi) di relazione matematica (tabella)

**Schema e istanza:**  $R(A_1, A_2, \dots, A_n)$  per la relazione,  $r(a_1, a_2, \dots, a_n)$  per le istanze

**Basi di dati come insieme di relazioni**  $R = (A_1, A_2, \dots, A_n)$

**Superchiave:** insieme di attributi per cui la relazione R di cui è superchiave in cui non c'è nessuna tupla che si ripete. L'insieme di tutti gli attributi di una relazione è un esempio di superchiave.

**Superchiave minimale (o CHIAVE):** quella che non contiene nessun'altra sottochiave, o nessun altro sottoinsieme di attributi.

**Valore nullo:** "null", esiste per alcuni attributi, mentre altri non possono avere valore nullo; è un po' come facevamo in E-R, quando mettavamo cardinalità (0, 1) per un attributo in modo da indicarne l'opzionalità.

**Vincoli di inter e intra relazionali:** vedremo oggi

**Chiave Primaria:** vedremo oggi

## 10.3 Tipi di vincoli

**vincoli intrarelazionali:**

vincoli su valori (o di dominio)

vincoli di tupla

vincoli di chiave (valuta le tuple nel complesso. es. non possono esistere due tuple con uno stesso valore per un particolare attributo A chiave).

**vincoli interrelazionali** (coinvolge più relazioni):

vincoli di integrità referenziale

Tre concetti:

- superchiave

- superchiave minimale (o Chiave)
- Chiave primaria

La **chiave primaria** non contiene attributi nulli. Sottoinsieme di attributi per cui è impossibile definire una superchiave minimale e per cui è impossibile avere attributi nulli.

Es.: tabella pag 65 universitari: la tabella avrà sempre una superchiave minimale, la matricola, che non può avere valore nullo (altrimenti come li identifichiamo gli studenti?) e questa è la **chiave primaria**.

### 10.3.1 Importanza delle chiavi

L'esistenza delle chiavi garantisce l'accessibilità a ciascun dato della base di dati.

Le chiavi permettono di correlare i dati in relazioni diverse: il modello relazionale è basato su valori.

Matricola	Cognome	Nome	Corso	Nascita
NULL	NULL	Mario	Ing Inf	5/12/78
78763	Rossi	Mario	Ing Civile	3/11/76
65432	Neri	Piero	Ing Mecc	10/7/79
87654	Neri	Mario	Ing Inf	NULL
NULL	Neri	Mario	NULL	5/12/78

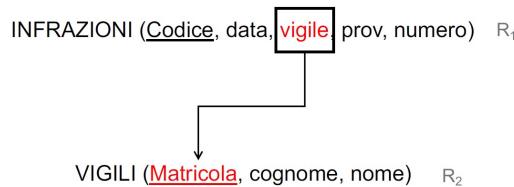
- questo schema di relazione ha come chiavi {Matricola} e {Cognome,Nome,DataNascita}
- La presenza di valori nulli nelle chiavi deve essere limitata  
(anche perché non ci si può riferire in altre relazioni della base di dati a certe tuple di questa istanza)

La slide 71 mi riporta un esempio in cui i dati sono rappresentati in 3 tavole correlate e relativamente indipendenti fra loro. Nella slide successiva viene mostrato l'output che avremmo se provassimo a rappresentare i dati in una sola tabella. Che non funzionerebbe tanto perché ci sarebbero troppe ripetizioni e poi avremmo anche molti valori nulli. Questa rappresentazione era da citare però perché mostra il comportamento di NoSQL.

## 10.4 Vincolo di integrità referenziale

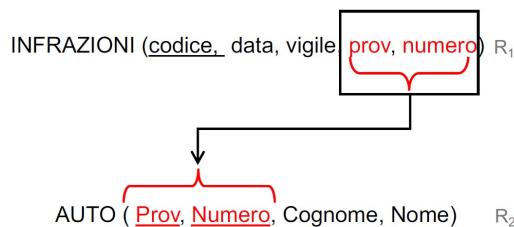
Un **vincolo di integrità referenziale** ("foreign key") fra gli attributi  $X$  di una relazione  $R_1$  e un'altra relazione  $R_2$  impone ai valori su  $X$  in  $R_1$  di comparire come valori della chiave primaria di  $R_2$ .

All'esame dovremo scegliere noi la chiave primaria (sottolineandola) e dire quali attributi vanno a formare il vincolo, ovvero a comporre la chiave primaria.



Integrità referenziale:

Graficamente



Leggi tu le slide, ora un esercizio di esempio.

## 10.5 Esercizi di esempio

### 10.5.1 Es.: prestito di libri

#### Cap. 2

Considerare le informazioni per la gestione dei prestiti di una biblioteca personale.

Il proprietario presta libri ai suoi amici, che indica semplicemente attraverso i rispettivi nomi o soprannomi (così da evitare omonimie) e fa riferimento ai libri attraverso i titoli (non possiede due libri con lo stesso titolo). Quando presta un libro, prende nota della data prevista di restituzione.

**Definire uno schema di relazione per rappresentare queste informazioni,** individuando opportuni domini per i vari attributi e mostrarne un'istanza in forma tabellare. Indicare la chiave (o le chiavi) della relazione.

Quello che dobbiamo fare noi è scegliere la chiave primaria e dire quali attributi vanno a formare il vincolo, ovvero a comporre la chiave primaria.

Nome	Libro	Data

PRESTI(Libro, Nome, Data\_presto)

AAA, Mary, 23/03/77

#### Cap. 2

PRESTITO (Libro, Nome, Data\_restituzione)

Libro	Nome	Data restituzione
Romeo and Juliet	John	15/04/95
Hamlet	Mary	20/03/95
Macbeth	Tom	1/04/95
King Lear	Peter	20/03/95

{Libro, Data\_restituzione} è chiave

{Libro, Nome} è possibile chiave

Nome è chiave per caso

### 10.5.2 Altro esempio

Facciamo una sorta di reverse engineering: prendiamo una tabella e cerchiamo di capire come è stata fatta.

Ovvero: descrivere in linguaggio naturale le informazioni organizzate nella base di dati. Individuare:

1. le chiavi (primaria, e almeno 1 superchiave)
2. i vincoli di integrità referenziale
3. gli attributi su cui NON possono essere ammessi valori nulli.

Per il punto 1.:

Pazienti		
codice	cognome	nome
A102	Harris	Lucy
B372	Rossini	Peter
B543	Johnson	Nadia
B444	Johnson	Lungi
S555	Rose	Jean

Ricoveri		
codice	datainizio	datafine
A102	2/05/94	9/05/94
A102	2/12/94	2/01/95
S372	5/10/94	3/12/94
B444	1/12/94	1/01/95
S555	5/10/94	1/11/94

Reparti		
codice	nome	dottore
A	Surgical	203
B	Paediatric	574
C	Medical	530

Dottori			
numero	cognome	nome	reparto
203	Black	Peter	A
574	Bisi	Mavis	B
461	Boyne	Steve	B
530	Clark	Nicola	C
405	Muzzi	Nicola	A
501	Mount	Mavis	A

Descrivere in linguaggio naturale le informazioni organizzate nella base di dati.

(base di dati per un ospedale o una clinica)

Per il punto 2.: c'è un vincolo di integrità referenziale non rispettato (ovvero, non esiste nella tabella dei pazienti il paziente identificato dal codice S372 che è possibile vedere nella tabella dei ricoveri).

Per il punto 3.: tutte le chiavi primarie ovviamente.

## Soluzioni

1. I valori nulli sono permessi solo negli attributi che non sono chiave primaria.
2. Alcuni non hanno molto senso (nome cognome in Dottori)
3. Codice per la relazione PAZIENTI
4. Paziente, datainizio, Paziente, datafine per la relazione RICOVERI (si assume che un paziente sia ricoverato -dimesso- una sola volta in un giorno)
5. Numero per la relazione DOTTORI
6. Codice per la relazione REPARTI
7. I vincoli referenziali sono fra:
  - Codice in RICOVERI e Codice in PAZIENTI
  - Reparto in RICOVERI e Codice in REPARTI
  - Dottore in REPARTO e Numero in DOTTORI
  - Reparto in DOTTORI e Codice in REPARTI
8. I valori nulli sono permessi solo negli attributi che non sono chiave primaria. Alcuni non hanno molto senso (nome cognome in Dottori)

### 10.5.3 Altro esempio: stazione

**Cap. 2**

Rappresentare per mezzo di una o più relazioni le informazioni contenute nell'orario delle partenze di una stazione ferroviaria: numero, orario, destinazione finale, categoria, fermate intermedie, di ciascuno dei treni in partenza.

Indicare: chiavi, vincoli di integrità referenziale dello schema.

Sol. aula:

```
TRENO (Tipo, \underline{Numero}, DestinazioneFinale, OrarioPartenza)
FERMATE (\underline{Nome}, Numero,) Orario)
```

Sol. slides:

```
PARTENZE (\underline{Numero}, OrarioPartenza, Destinazione Finale, Catego
FERMATE (\underline{Treno}, Stazione), Orario)
```

La prima relazione rappresenta tutti i treni in partenza dalla stazione ferroviaria, distinti per il Numero (chiave primaria della relazione).

La seconda rappresenta le fermate intermedie per ciascun treno in ciascuna stazione (chiave primaria composta da Treno e Stazione) con vincolo referenziale fra Treno in STOP e Numero in PARTENZE.

Manca qualcosa: relazione treno.

```
TRENO(\underline{Numero}, Categoria)
```

La terza rappresenta le fermate intermedie per ciascun treno in ciascuna stazione (chiave primaria composta da Treno, Stazione e orario).

### 10.5.4 Altro esempio: azienda

**Cap. 2**

Definire uno schema di base di dati per organizzare le informazioni di una azienda che ha impiegati (ognuno con CF, cognome, nome, data di nascita), filiali (con codice, sede, direttore - che è un impiegato).

Ogni impiegato lavora presso una filiale. Indicare: chiavi, vincoli di integrità referenziale dello schema.

Quante relazioni ci servono? 2. Ogni filiale ha più impiegati. Quindi una tabella per filiali, una per impiegati, una per azienda.

```
IMPIEGATO (\underline{CF}, Nome, Cognome, Nascita, Filiale)
FILIALE (\underline{Codice}, Sede, Direttore)
```

Ellisse con freccia da IMPIEGATO(..., Filiale) a FILIALE(Codice, ...) per indicare il vincolo di integrità referenziale.

Ellisse con freccia da FILIALE(..., Direttore) a IMPIEGATO per indicare il vincolo di integrità referenziale.

### 10.5.5 Altro esempio: radio

#### Cap. 2

Definire uno schema di base di dati che organizzi i dati necessari a generare la pagina dei programmi radiofonici quotidiani, con stazioni, ore e titoli dei programmi; per ogni stazione sono memorizzati, oltre al nome, anche la frequenza di trasmissione e la sede.

Quante relazioni ci servono? 2.

PROGRAMMI (\underline{Titolo}, Stazione, Orario)  
 STAZIONE (\underline{Nome}, Frequenza, Sede)

Magari all'interno di PROGRAMMI(...) basta da solo il Titolo per identificare univocamente un programma all'interno di una stazione.

#### Cap. 2

STAZIONI (**Nome**, Frequenza, Sede)  
**PROGRAMMI(Titolo,Stazione,Ora)**  
 con titolo unico in tutte le stazioni e in tutti gli orari  
 vincolo di integrità tra **Stazione** in PROGRAMMI e **Nome** in STAZIONI

**Opzione2:**  
**PROGRAMMI(Titolo,Stazione,Ora)**  
 con programma che va in onda con lo stesso titolo in stazioni diverse

**Opzione3:**  
**PROGRAMMI(Titolo,Stazione,Ora)**  
 con programma che va in onda più volte nella stessa stazione con lo stesso titolo in stazioni diverse

## 10.6 Esempio di esercizio MR che troveremo all'esame

### ESERCIZIO DI MR

Si consideri il seguente schema di base di dati che descrive la programmazione dei film nelle sale cinematografiche. Tenere conto del fatto che film diversi possono avere lo stesso titolo ma, in questo caso, non sono diretti dallo stesso regista. Per ogni film si vuole tenere traccia degli attori. Per ogni attore e regista si vuole tenere traccia del primo anno di attività.

```

PERSONA (id_persona, nome, cognome, telefono, nazionalità, sesso, anno_nascita, primo_anno_attività)
FILM (titolo, id_regista, genere, anno, durata)
PARTECIPAZIONE (id_attore, titolo_film, id_regista, ruolo)
CINEMA (id_cinema, nome, indirizzo)
SALACINEMATOGRAFICA (nome_sala, id_cinema, capienza, giorno_chiusura)
PROGRAMMAZIONE (nome_sala, id_cinema, film, id_regista, giorno_antepremiera, giorno_fine_programmazione)

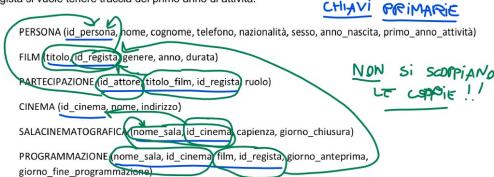
```

Definire tutte le chiavi primarie e i tutti i vincoli di integrità referenziale.

- Indicare inoltre:
- un vincolo di dominio
  - un vincolo di ennupla
  - indicare una superchiave non minimale
  - individuare una chiave che non sia stata scelta come chiave primaria.
- Le chiavi primarie ed i vincoli di integrità referenziale possono essere espressi in modo grafico direttamente sul testo.

### ESERCIZIO DI MR

Si consideri il seguente schema di base di dati che descrive la programmazione dei film nelle sale cinematografiche. Tenere conto del fatto che film diversi possono avere lo stesso titolo ma, in questo caso, non sono diretti dallo stesso regista. Per ogni film si vuole tenere traccia degli attori. Per ogni attore e regista si vuole tenere traccia del primo anno di attività.

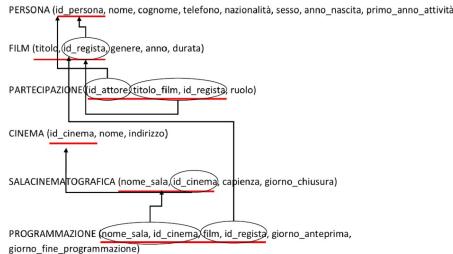


Definire tutte le chiavi primarie e i tutti i vincoli di integrità referenziale.

- Indicare inoltre:
- un vincolo di dominio
  - un vincolo di ennupla
  - indicare una superchiave non minimale
  - individuare una chiave che non sia stata scelta come chiave primaria.
- Le chiavi primarie ed i vincoli di integrità referenziale possono essere espressi in modo grafico direttamente sul testo.

### COME SI SVOLGE UN ESERCIZIO DI MR (VINCOLI DI INTEGRITÀ REFERENZ.)

Si consideri il seguente schema di base di dati che descrive la programmazione dei film nelle sale cinematografiche. Tenere conto del fatto che film diversi possono avere lo stesso titolo ma, in questo caso, non sono diretti dallo stesso regista. Per ogni film si vuole tenere traccia degli attori. Per ogni attore e regista si vuole tenere traccia del primo anno di attività.



# Capitolo 11

## Esercizi di MR

Svolgere gli esercizi nel seguente modo:

- Identificare le relazioni che modellano il dominio descritto nel testo
- Identificare per ogni relazione le chiavi primarie e i vincoli di integrità referenziale
- Identificare almeno un vincolo di dominio
- Identificare quali attributi ammettono valori nulli
- Identificare un vincolo di ennupla
- Identificare una superchiave
- Identificare una chiave alternativa

### 11.1 Es. A

Si consideri il seguente schema di basi di dati che descrive un campionato provinciale di pallavolo. Le squadre si incontrano due volte, nel girone di andata e in quello di ritorno. Si tenga presente che il risultato di una partita di pallavolo è al meglio dei cinque set, non è detto che una squadra giochi le partite nella palestra dove si allena, più squadre possono allenarsi o giocare in giorni/orari diversi ma nella stessa palestra.

#### Identifico le relazioni: attributi

- SQUADRA (nome, capitano, città, palestraAll, cittàAll, palestraPart, cittàPart, sponsor)

- PALESTRA (nome, città, indirizzo, numPosti)
- ATLETA (tesserino, nome, cognome, dataNascita, luogoNascita, squadra)
- ARBITRO (tesserino, nome, cognome, dataNascita, luogoNascita, numPartiteArbitrate)
- SPONSOR (pIva, nome, telefono)
- RISULTATO (squadraCasa, squadraOspite, setSquadraCasa, setSquadraOspite, arbitro, palestra, città, data, oraInizio, oraFine)

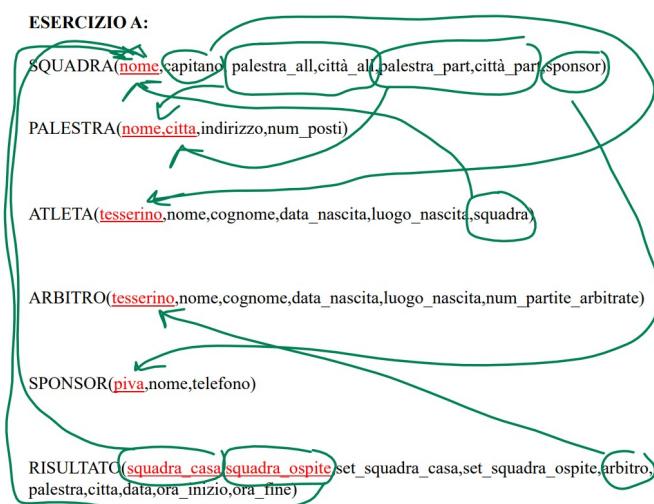
#### Identifico le chiavi primarie

- SQUADRA (nome, capitano, città, **palestraAll**, cittàAll, palestraPart, cittàPart, sponsor)
- PALESTRA (nome, città, indirizzo, numPosti)
- ATLETA (tesserino, nome, cognome, dataNascita, luogoNascita, squadra)
- ARBITRO (tesserino, nome, cognome, dataNascita, luogoNascita, numPartiteArbitrate)
- SPONSOR (pIva, nome, telefono)
- RISULTATO (squadraCasa, squadraOspite, setSquadraCasa, setSquadraOspite, arbitro, palestra, città, data, oraInizio, oraFine)

#### Identifico gli attributi che possono essere nulli

- SQUADRA (nome, capitano, città, palestraAll, cittàAll, palestraPart, cittàPart, *sponsor*)
- PALESTRA (nome, città, indirizzo, numPosti)
- ATLETA (tesserino, nome, cognome, dataNascita, luogoNascita, squadra)
- ARBITRO (tesserino, nome, cognome, dataNascita, luogoNascita, *numPartiteArbitrate*)
- SPONSOR (pIva, nome, telefono)

- RISULTATO (squadraCasa, squadraOspite, *setSquadraCasa*, *setSquadraOspite*, arbitro, palestra, città, data, oraInizio, oraFine)



### Identifico i vincoli di tupla

- *setSquadraCasa* e *setSquadraOspite* devono essere maggiori uguali di zero e minori uguali di 3.
- *setSquadraCasa* + *setSquadraOspite* deve essere maggiore uguale di 3 e minore uguale di 5.
- oraInizio ⋜ oraFine
- Una superchiave nella tabella sponsor è p.via, nome, telefono
- Una chiave alternativa è numTelefono in sponsor

## 11.2 Es. B

Si consideri il seguente schema di basi di dati che descrive le informazioni relative ad una olimpiade. Per ogni nazione si vuole tenere traccia dell'atleta porta bandiera, del numero di atleti, della mascotte e colore della divisa. Si vuole tenere traccia degli atleti e delle partecipazioni degli atleti alle gare con posizione raggiunta e fase della gara a cui sono arrivati. Per esempio nelle gare di corsa, le fasi possono essere: qualificazioni, ottavi, quarti, ecc. Ogni fase di gara ha due giudici e si svolgono ad una specifica data, ora e luogo. Per ogni atleta si vuole tenere traccia del codice tesserino atleta, nazione, cognome, nome e data di nascita.

### Identifico le relazioni

- NAZIONE (nome, portabandiera, numeroAtleti, mascotte, coloreDivisa)
- ATLETA (tesserino, nome, cognome, dataNascita, nazione)
- GARA (nomeGara, disciplina, recordMondiale, recordOlimpionico)
- SVOLGIMENTOGARA (gara, faseGara, luogo, data, ora, giudice1, giudice2)
- PARTECIPAZIONEGARA (atleta, gara, faseGara, posizioneClassifica, datiGara)
- GIUDICE (tesserino, nome, cognome, dataNascita, nazione)

### Identifico le chiavi primarie

- NAZIONE (nome, portabandiera, numeroAtleti, mascotte, coloreDivisa)
- ATLETA (tesserino, nome, cognome, dataNascita, nazione)
- GARA (nomeGara, disciplina, recordMondiale, recordOlimpionico)
- SVOLGIMENTOGARA (gara, faseGara, luogo, data, ora, giudice1, giudice2)
- PARTECIPAZIONEGARA (atleta, gara, faseGara, posizioneClassifica, datiGara)
- GIUDICE (tesserino, nome, cognome, dataNascita, nazione)

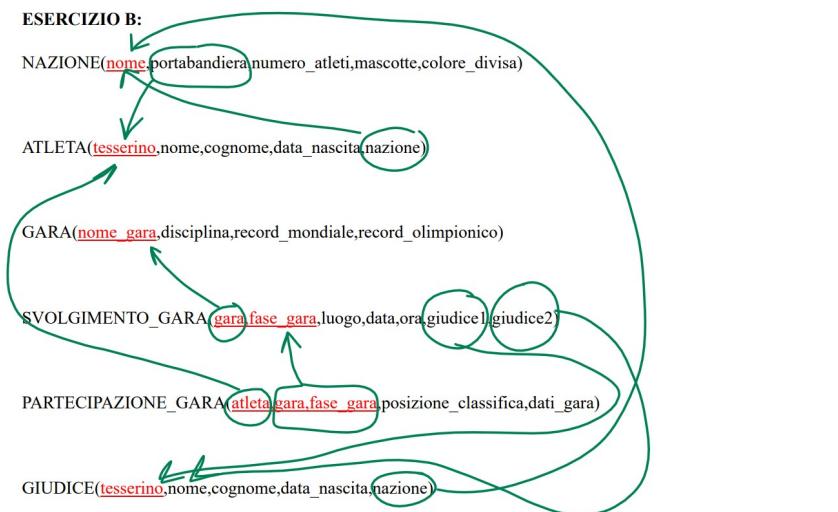
### Identifico gli attributi che possono essere nulli

Non è che abbia capito molto

- NAZIONE (nome, portabandiera, numeroAtleti, mascotte, coloreDivisa)
- ATLETA (tesserino, nome, cognome, dataNascita, nazione)
- GARA (nomeGara, disciplina, recordMondiale, recordOlimpionico)
- SVOLGIMENTOGARA (gara, faseGara, luogo, data, ora, giudice1, giudice2)

- PARTECIPAZIONEGARA (atleta, gara, faseGara, posizioneClassifica, datiGara)
- GIUDICE (tesserino, nome, cognome, dataNascita, nazione)

Boh da finire



## 11.3 Es. C

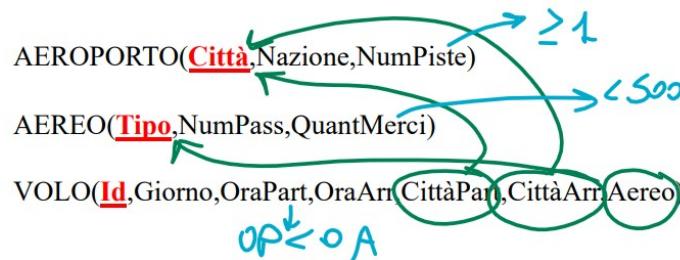
Si vuole rappresentare una base di dati relativa ai voli di una compagnia aerea. Si vuole tenere traccia dell'id del volo, del giorno della settimana, della città di partenza e di arrivo e ovviamente delle ore di partenza e arrivo. Un volo per esempio Milano-Parigi effettuato dall'aereo di tipo Airbus 300 che ha id=004 può essere svolto solo una volta durante una giornata. Per ogni tipologia di aereo si vuole tenere traccia del numero di passeggeri trasportabili, e quantità delle merci trasportabili. Ogni aeroporto è caratterizzato da una città, nazione e numero di piste. Si assuma la semplificazione che una città può avere un solo aeroporto.

### Identifico le relazioni

- AEROPORTO (Città, Nazione, NumPiste)
- AEREO (Tipo, NumPass, QuantMerci)
- VOLO (ID, Giorno, OraPart, OraArr, CittàPart, CittàArr, Aereo)

### Identifico le chiavi primarie

- AEROPORTO (Città, Nazione, NumPiste)
- AEREO (Tipo, NumPass, QuantMerci)
- VOLO (ID, Giorno, OraPart, OraArr, CittàPart, CittàArr, Aereo)



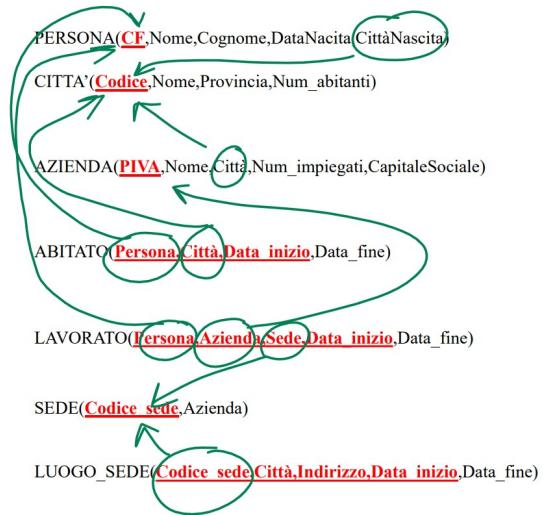
## 11.4 Es. D

Si vuole rappresentare una base di dati relativa ai lavoratori delle aziende italiane. Per ogni lavoratore si vuole tenere traccia delle informazioni anagrafiche, delle città in cui ha abitato e lavorato e delle aziende in cui ha lavorato. Di ogni città si vuole conservare: nome, provincia e numero di abitanti. Per ogni azienda in cui le persone hanno lavorato si vuole tenere traccia del nome, P.IVA, città, numero impiegati e capitale sociale. Una azienda potrebbe aver nel tempo cambiato sede. Quindi ad esempio la Microsoft potrebbe avere avuto sede dal 1998 al 2009 a Rozzano e dal 2010 a Milano. Ogni lavoratore potrebbe aver lavorato in più aziende nel corso della sua vita.

### Identifico le relazioni: attributi

- PERSONA (CF, Nome, Cognome, DataNacita, CittàNascita)
- CITTA' (Codice, Nome, Provincia, NumAbitanti)
- AZIENDA (PIVA, Nome, Città, NumImpiegati, CapitaleSociale)
- ABITATO (Persona, Città, DataInizio, DataFine)
- LAVORATO (Persona, Azienda, Sede, DataInizio, DataFine)

- SEDE (CodiceSede, Azienda)
- LUOGOSEDE (CodiceSede, Città, Indirizzo, DataInizio, DataFine)



## 11.5 Es. E

## 11.6 Es. F

## 11.7 Es. G

# Capitolo 12

## Altri es. MR

### 12.1 Es.1 - tavolo cliente

#### Specifiche

Il seguente schema relazionale riguarda un sistema di prenotazione e gestione dei tavoli di un ristorante. Il sistema in fase di prenotazione prevede una fase di registrazione in cui il cliente deve immettere i propri dati tra cui il codice fiscale, nome, cognome, e altro. Ogni tavolo ha un codice identificativo e numero di coperti massimo. Una prenotazione può riferirsi al turno pranzo o cena. Un tavolo può essere utilizzato solo una volta durante un turno. Un addetto del ristorante può servire più tavoli ma ogni tavolo è servito da un solo addetto. Ogni tavolo è dotato di un tablet attraverso cui è possibile sfogliare il menu e ordinare i pasti. Il tablet viene fornito dall'addetto al servizio al cliente ad inizio pranzo o cena. I tavoli sono disposti in diverse sale separate che possono essere 'normali' o sale in cui si proiettano eventi sportivi e concerti.

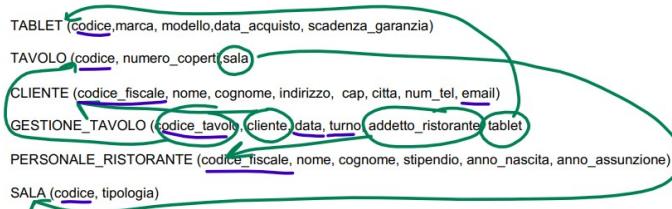
#### Relazioni

TABLET (codice, marca, modello, dataAcquisto, scadenzaGaranzia)  
TAVOLO (codice, numeroCoperti, sala)  
CLIENTE (codiceFiscale, nome, cognome, indirizzo, cap, citta, numTel, email)  
GESTIONETAVOLO (codiceTavolo, cliente, data, turno, addettoRistorante, tablet)  
PERSONALERISTORANTE (codiceFiscale, nome, cognome, stipendio, annoNascita, annoAssunzione)  
SALA (codice, tipologia)

### Chiavi primarie

TABLET (codice, marca, modello, dataAcquisto, scadenzaGaranzia)  
 TAVOLO (codice, numeroCoperti,sala)  
 CLIENTE (codiceFiscale, nome, cognome, indirizzo, cap, citta, numTel, email)  
 GESTIONETAVOLO (codiceTavolo, cliente, data, turno, addettoRistorante, tablet)  
 PERSONALERISTORANTE (codiceFiscale, nome, cognome, stipendio, annoNascita, annoAssunzione)  
 SALA (codice, tipologia)

### Chiavi esterne



### Vincoli di tupla

data.acquisto  $\neq$  scadenza + 2  
 anno-nascita  $\leq$  anno-assunzione + 18  
 CAP = seq numeri interi L = 5  
 numero-coperti  $\geq$  0

## 12.2 Es.2 - gestione magazzino

### Specifiche

Il seguente schema relazionale riguarda un sistema per la gestione di una palestra. In palestra possono entrare solo gli iscritti con abbonamento in corso di validità. All'entrata gli iscritti utilizzano un badge che permette di aprire il tornello di entrata solo se l'abbonamento è in corso di validità. Gli abbonamenti possono essere di diversa tipologia: mezza giornata, giornata piena ecc.. La palestra è composta da sale ognuna destinata ad una tipologia di attività: attrezzi, corpo libero ecc. In una data sala viene svolta una specifica attività e opera un solo addetto. Nelle sale possono esserci degli attrezzi. Dalla palestra vengono offerte gratuitamente (incluso nell'abbonamento) delle lezioni di gruppo come «lezione di aerobica», «zumba» ecc. che

sono svolte in un dato giorno e ora della settimana in una data sala sempre dallo stesso addetto.

### **Relazioni**

ATTREZZATURA (codice, marca, modello, dataAcquisto, scadenzaGaranzia, sala)  
 LEZIONE (codiceLezione, nome, giorno, ora, sala, addetto)  
 ISCRITTO (codiceFiscale, nome, cognome, indirizzo, cap, citta, numTel, email)  
 ABBONAMENTO(iscritto, dataInizio, dataFine, tipologia)  
 ADDETTO (codiceFiscale, nome, cognome, stipendio, annoNascita, annoAssunzione)  
 SALA (codice, tipologiaAttività, addetto)

### **Chiavi primarie**

ATTREZZATURA (codice, marca, modello, dataAcquisto, scadenzaGaranzia, sala)  
 LEZIONE (codiceLezione, nome, giorno, ora, sala, addetto)  
 ISCRITTO (codiceFiscale, nome, cognome, indirizzo, cap, citta, numTel, email)  
 ABBONAMENTO(iscritto, dataInizio, dataFine, tipologia)  
 ADDETTO (codiceFiscale, nome, cognome, stipendio, annoNascita, annoAssunzione)  
 SALA (codice, tipologiaAttività, addetto)

### **Vincoli di tupla**

data acquisto  $\downarrow$  scadenza garanzia  
 data inizio  $\downarrow$  data fine  
 anno nascita  $\leq$  anno assunzione

## **12.3 Es.3 - libreria**

### **Specifiche**

Lo schema seguente rappresenta una base di dati impiegata in una catena di librerie distribuite su tutto il territorio nazionale. In particolare, oltre ai dati sulle singole librerie, sono archiviati i dati del personale e dei libri in vendita. Si osservi che possono esistere diverse versioni dello stesso libro (autore e

titolo), edite da case editrici diverse. Infine, attraverso la relazione Catalogo è possibile conoscere in quali librerie una determinata versione di un libro sia disponibile per essere venduta, il numero di copie a disposizione, ed il costo.

### **Relazioni**

LIBRERIA(idLibreria, indirizzo, città, oraApertura, oraChiusura, turnoChiusura)  
 PERSONALE(idPersona, cognome, nome, idLibreria, reparto, turno)  
 CATALOGO(idLibro, idLibreria, reparto, numerocopie, costo)  
 LIBRO(idLibro, titolo, autori, casaEditrice, annoPubblicazione)

### **Chiavi primarie**

LIBRERIA(idLibreria, indirizzo, città, oraApertura, oraChiusura, turnoChiusura)  
 PERSONALE(idPersona, cognome, nome, idLibreria, reparto, turno)  
 CATALOGO(idLibro, idLibreria, reparto, numerocopie, costo)  
 LIBRO(idLibro, titolo, autori, casaEditrice, annoPubblicazione)

Chiavi alternative:

”indirizzo” e ”città” per Libreria.

Possibile superchiave:

tutta la tupla di una qualsiasi relazione, es la riga di Libro.

### **Vincoli di connupla**

orario apertura  $\neq$  orario chiusura

### **Vincoli di dominio**

costo  $\geq 0$

numerocopie  $\geq 0$

turnoChiusura  $\in$  lunedì, martedì, mercoledì, giovedì, venerdì, sabato, domenica

# Capitolo 13

## Algebra Relazionale

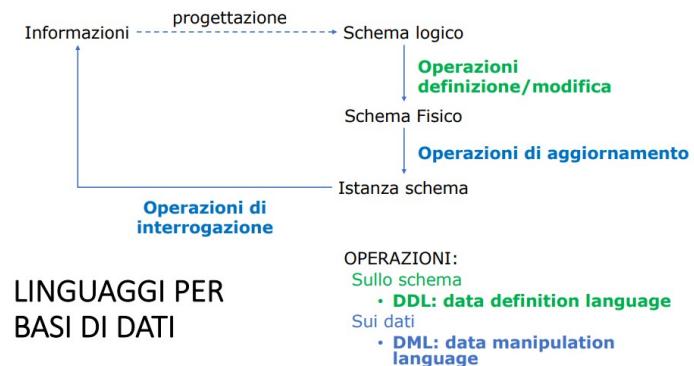
### 13.1 Dai modelli ai linguaggi

I modelli concettuali e logici permettono di descrivere informazioni, ma non sono direttamente interpretabili da un elaboratore. Dai modelli dobbiamo passare ai linguaggi, dotati di:

**sintassi**, che definisce le frasi corrette del linguaggio

**semantica**, che definisce le operazioni effettuate quando vengono eseguite gli operatori (o istruzioni o comandi) del linguaggio

#### 13.1.1 Linguaggi per Basi di Dati



#### 13.1.2 Tipi di linguaggi

##### Procedurali

Specificano le modalità di generazione del risultato ("come").

### Dichiarativi

Specificano le proprietà del risultato ("che cosa").

#### 13.1.3 Linguaggi per Basi di Dati Relazionali

**Algebra relazionale** : procedurale

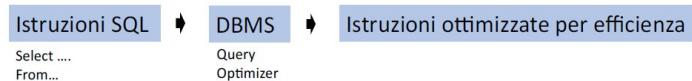
**Calcolo relazionale** : dichiarativo

**SQL** : (parzialmente) dichiarativo

**QBE** (Query by Example): dichiarativo

## 13.2 AR vs SQL

Capire l'algebra è la chiave per la comprensione dell'SQL. L'algebra relazionale è un linguaggio procedurale, mentre SQL è un linguaggio dichiarativo.



### AR

Linguaggio prettamente formale che forma la base per linguaggi 'reali'.

Linguaggio procedurale: si specifica l'algoritmo con cui ottenere il risultato.

Istruzioni equivalenti possono differire in termini di efficienza.

Relazioni intese in senso matematico =; Insiemi di tuple, definite su attributi.

Negli insiemi non ci possono essere elementi uguali.

### SQL

Linguaggio più usato per basi di dati relazionali.

Linguaggio (parzialmente) dichiarativo: si specifica il risultato da ottenere senza preoccuparsi di specificare l'algoritmo.

Istruzioni equivalenti differiscono solo per leggibilità.

Relazioni intese come tabelle.

Possono esserci righe uguali.

### 13.3 Algebra Relazionale

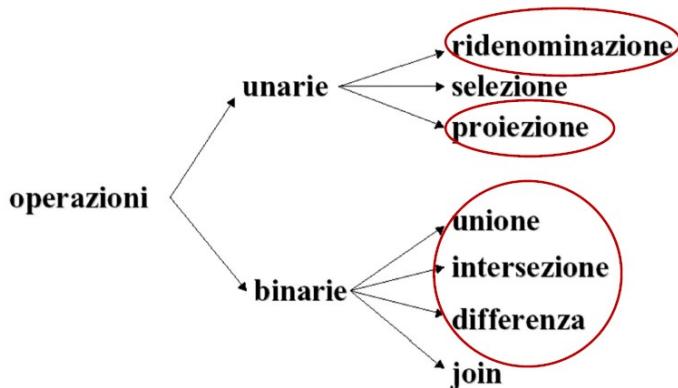
Insieme di operatori:

- su relazioni
- che producono relazioni
- e possono essere composti tra loro a formare nuove interrogazioni

#### Operatori AR



Operatori che vedremo in questa lezione



### 13.4 Operatori insiemistici

#### 13.4.1 Le relazioni sono insiemi

In matematica una relazione è un sottoinsieme del prodotto cartesiano di due o più insiemi.

Il prodotto cartesiano degli insiemi  $D_1, D_2, \dots, D_N$  indicato come  $D_1 \times D_2 \times \dots \times D_N$

$\dots \times D_n$  è l'insieme di tutte le n-uple ordinate  $(d_1, d_2, \dots, d_n)$  tali che  $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$ . Una relazione è un insieme quindi:

- non c'è ordinamento tra le diverse tuple
- le tuple sono distinte (non ce ne possono essere due uguali)
- ciascuna tupla è al suo interno ordinata: l'i-esimo valore proviene dall'i-esimo dominio

NB: una tabella è una relazione SOLO se non ci sono righe uguali.

### 13.4.2 L'operatore Intersezione

### 13.4.3 L'operatore Differenza

## 13.5 Limiti degli operatori insiemistici

### 13.5.1 L'operatore di ridenominazione

## 13.6 Select

## 13.7 Join

# Capitolo 14

## Progettazione Logica

Ci vorrà ancora un po' per i risultati, ma non manca tanto.  
La seconda parte, il secondo compitino, prevede

- la progettazione logica
- il secondo esercizio
- il terzo esercizio

Secondo il prof. Napoletano, la progettazione logica è abbastanza equiparabile al Modello Relazionale.

Viene presentato uno schema ER generico con proprietà ben determinate, senza una specifica, di cui bisogna identificare determinate regole di ri-strutturazione prima di tradurre il modello destrutturato in un elenco di tabelle.

## 14.1 Progettazione Logica



Come già precedentemente detto, non vedremo a lezione la progettazione fisica (è prevista a laboratorio però).

Richiede di scegliere il modello dei dati modello relazionale.

### Obiettivo

Definizione di uno schema logico relazionale corrispondente allo schema ER di partenza.

### Aspetti importanti

Semplificazione dello schema per renderlo rappresentabile mediante il modello relazionale ottimizzazione per aumentare l'efficienza delle interrogazioni.

#### 14.1.1 Obiettivo

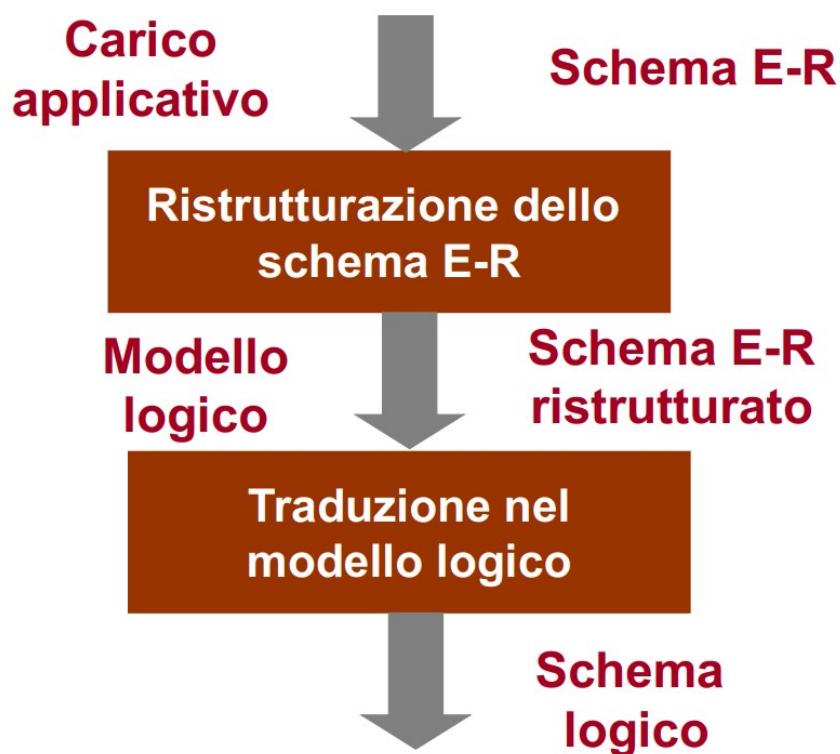
”Tradurre” lo schema concettuale in uno schema logico che rappresenti gli stessi dati in maniera corretta ed efficiente.

**Ingresso:**

- schema concettuale
- informazioni sul carico applicativo
- modello logico

**Uscita:**

- schema logico
- documentazione associata



Non si tratta di una pura e semplice traduzione

- alcuni aspetti non sono direttamente rappresentabili
- è inoltre necessario considerare le prestazioni

Modello ER     $\Leftrightarrow$     Modello relazionale  
 Corrispondenza dei concetti principali

<b>Modello ER</b>	<b>Modello relazionale</b>
Entità	$\Leftrightarrow$ Relazione ( <i>tabella</i> )
Relazione	$\Leftrightarrow$ Riferimento ( <i>chiave esterna</i> )
Generalizzazione	$\times$ Non presente
Attributo semplice	$\Leftrightarrow$ Attributo ( <icampo< i="">)</icampo<>
Attributo multivale	$\times$ Non presente

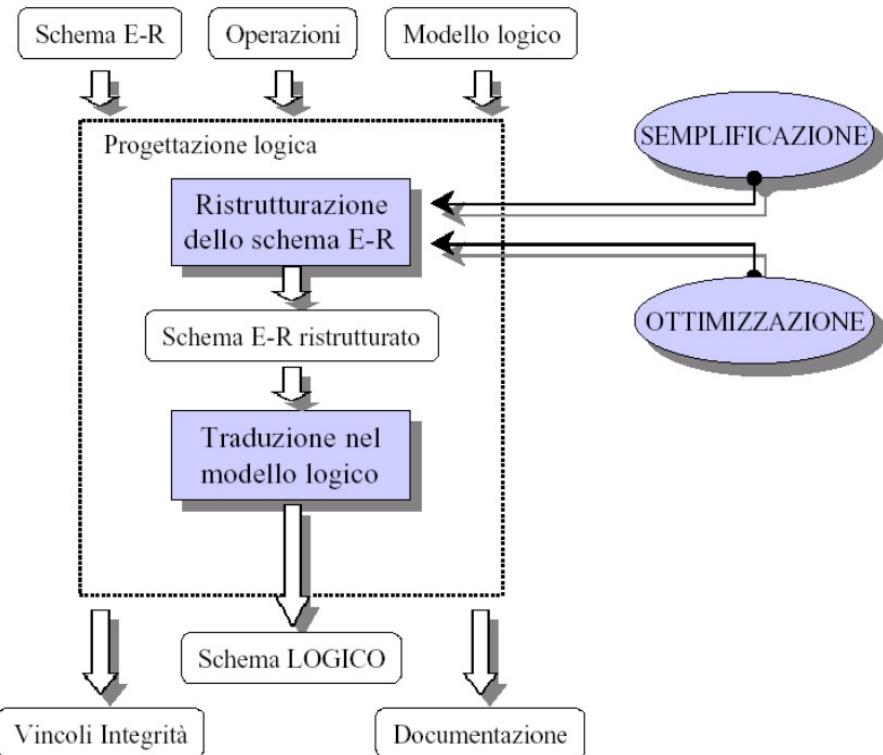
#### 14.1.2 Ristrutturazione schema E-R

Eliminazione dallo schema E/R di tutti i costrutti che non possono essere direttamente rappresentati nel modello logico target (relazionale nel nostro caso):

- Eliminazione degli attributi multivale
- Eliminazione delle generalizzazioni

Inoltre:

- Partizionamento/accorpamento di entità e associazioni
- Scelta degli identificatori primari
- Analisi ridondanze (non dovrebbero esserci, ...)



### Motivazioni:

- semplificare la traduzione
- ”ottimizzare” le prestazioni

### Osservazione:

- uno schema E-R ristrutturato non è (più) uno schema concettuale nel senso stretto del termine

**Per ottimizzare il risultato abbiamo bisogno di analizzare le prestazioni a questo livello.**

Ma:

- le prestazioni non sono valutabili con precisione su uno schema concettuale:
- Dipendono dalle caratteristiche del DBMS
- bisogna conoscere il volume dei dati e le caratteristiche delle operazioni

### 14.1.3 Carico applicativo

Consideriamo degli "indicatori" dei parametri che regolano le prestazioni:

- *tempo di esecuzione* delle operazioni di principale interesse: numero di istanze (di entità e relazioni) mediamente accedute durante l'esecuzione dell'operazione (*accessi*)
- *spazio di memoria* necessario per memorizzare i dati di interesse

Per valutare questi parametri bisogna conoscere (oltre allo schema):

#### volume dei dati:

- numero di istanze previste di entità e relazioni
- dimensione di ciascun attributo

#### caratteristiche delle operazioni:

- tipo: interattiva o batch
- frequenza: numero medio di esecuzioni in un certo periodo
- dati coinvolti

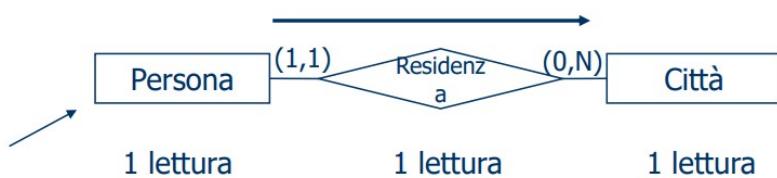
Si noti che la valutazione sarà necessariamente approssimata, in quanto le prestazioni effettive della base di dati dipendono anche da parametri fisici, difficilmente prevedibili in questa fase (DBMS utilizzato, indici, ...).

### 14.1.4 Schema di operazione

Lo schema di operazione descrive i dati coinvolti in un'operazione.

Corrisponde al frammento dello schema ER interessato all'operazione sul quale viene disegnato il cammino logico per accedere alle informazioni di interesse.

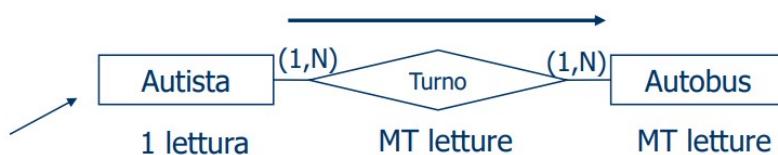
**Operazione:** data una Persona trovare la città di Residenza



Ma perché questo schema è importante? Per le operazioni che devo affrontare che possono essere molto ripetitive.

### 14.1.5 Esempio 1

**Operazione:** dato un Autista trovare gli autobus dei suoi turni



Per calcolare il numero di accessi a Turno e Autobus occorre conoscere il numero medio di Turni per Autista (MT).

### 14.1.6 Tavola degli accessi

Con lo schema di operazione si può fare una stima del costo di un'operazione contando il numero di accessi alle istanze di entità e relazioni. Il risultato può essere riassunto in una **tavola degli accessi**.

Concetto	Costrutto	Accessi	Tipo

Il tipo distingue gli accessi in scrittura (S) e in lettura (L). Le operazioni di scrittura sono in genere più onerose (esecuzione in modo esclusivo, aggiornamento degli indici).

### 14.1.7 Esempio 1

**Operazione:** Stampare il curriculum di uno Studente



Si suppone che la media degli esami sostenuti dagli studenti sia 10

Concetto	Costrutto	Accessi	Tipo
Studente	E	1	L
Esame	R	10	L
Corso	E	10	L

### 14.1.8 Esempio 2

**Operazione:** Registrare un Esame di un Corso ad uno Studente

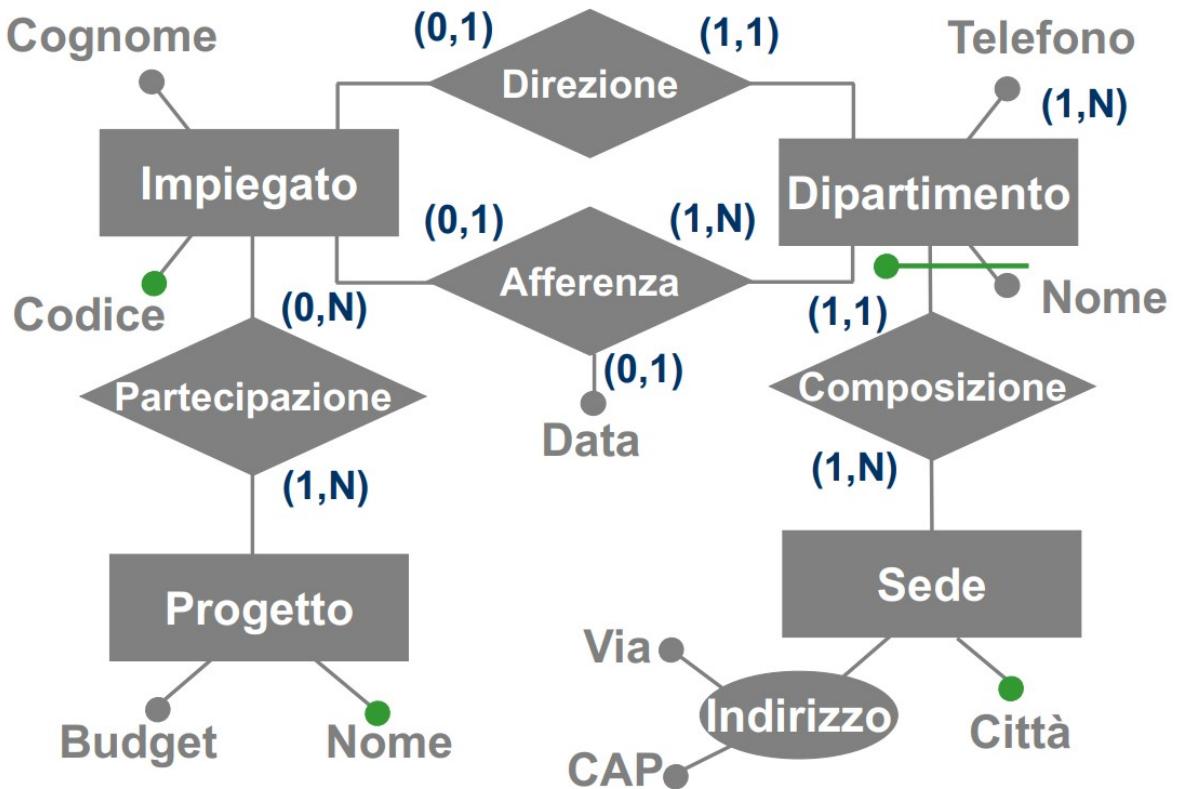


Si suppone che si inseriscano solo Esami nuovi

Concetto	Costrutto	Accessi	Tipo
Studente	E	1	L
Esame	R	1	S
Corso	E	1	L

### 14.1.9 Esempio di valutazione di costo

Schema E-R:



**Operazione:** trova tutti i dati di un impiegato, del dipartimento nel quale lavora e dei progetti ai quali partecipa.

Si costruisce una **tavola degli accessi** basata su uno **schema di navigazione**.

**Schema di navigazione:** parte dello schema E/R interessata dall'operazione, estesa con delle frecce che indicano in che modo l'operazione "naviga" i dati.

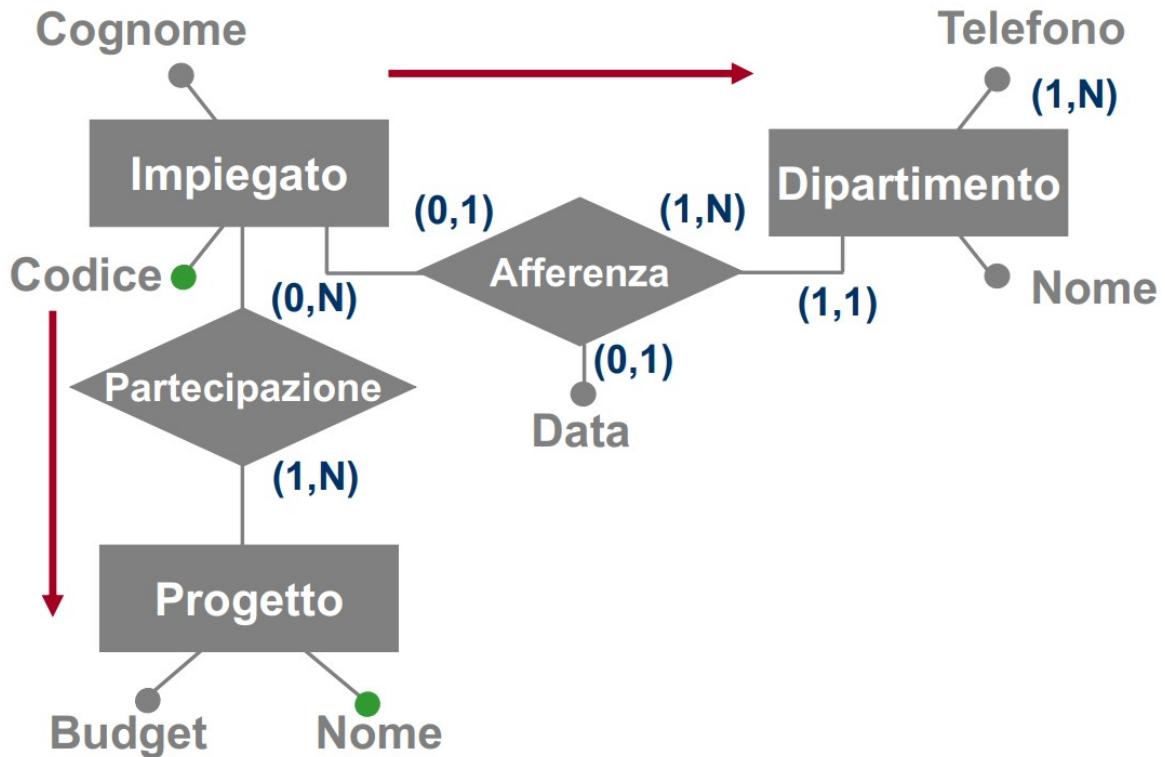


Tavola dei volumi:

Concetto	Tipo	Volume
Sede	E	10
Dipartimento	E	80
Impiegato	E	2000
Progetto	E	500
Composizione	R	80
Afferenza	R	1900
Direzione	R	80
Partecipazione	R	6000

Specifica il numero stimato di istanze per ogni entità (E) e associazione (R) dello schema. I valori sono necessariamente approssimati, ma indicativi. Il numero (medio) di partecipazioni di una istanza di entità alle istanze di relazione (dipende dalla cardinalità delle relazioni).

#### 14.1.10 Attività della ristrutturazione

Analisi delle ridondanze.  
Eliminazione delle generalizzazioni/gerarchie.  
Partizionamento/accorpamento di entità e relationship.  
Eliminazione attributi multivaleure.  
Scelta degli identificatori primari

#### 14.1.11 Analisi delle ridondanze

Una ridondanza in uno schema E-R è una informazione significativa ma derivabile da altre.

In questa fase si decide se eliminare le ridondanze eventualmente presenti o mantenerle.

#### 14.1.12 Ridondanze

Vantaggi

- semplificazione delle interrogazioni

Svantaggi

- appesantimento degli aggiornamenti
- maggiore occupazione di spazio

#### 14.1.13 Forme di ridondanza in uno schema E-R

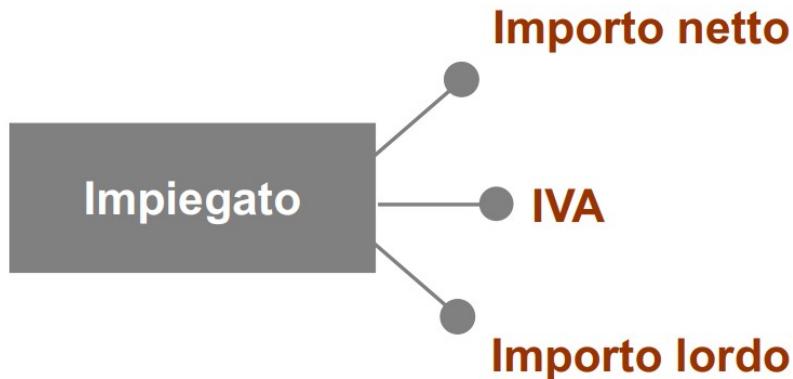
Attributi derivabili:

- da altri attributi della stessa entità (o relazione)
- da attributi di altre entità (o relazioni)

Associazioni derivabili dalla composizione di altre relazioni in presenza di cicli.

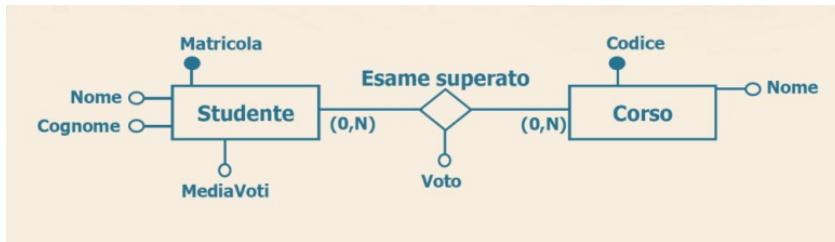
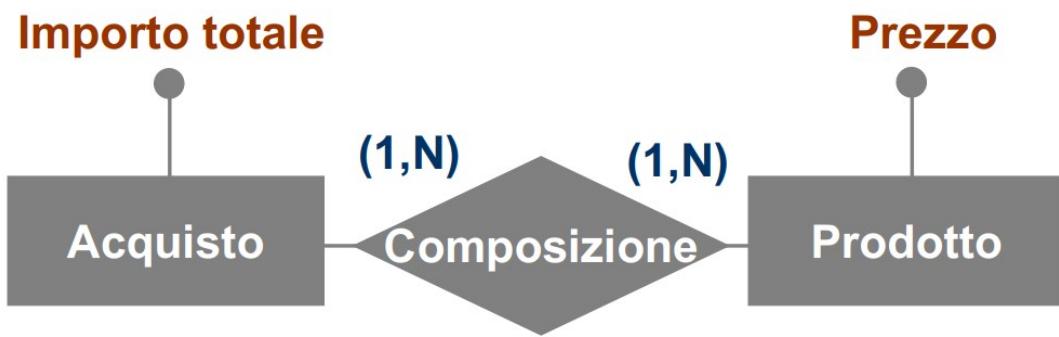
#### Attributi derivabili

Il valore di un attributo si calcola sulla base delle altre proprietà.



**Attributi derivabili da un'altra entità**

Attributo calcolato sulla base degli attributi di un'altra entità.



L'attributo MediaVoti è ridondante.

- utile per velocizzare le interrogazioni relative al calcolo della media dei voti degli studenti.
- se conservato, occorre integrare lo schema relazionale con l'indicazione di ridondanza dell'attributo.

Vabbeh facciamo che ascolto e basta, mi sono persa.  
È arrivato alla slide 49/142.