

Basi di Dati - Appunti lezioni

Sara Angeretti

@Sara1798

2022/2023

Indice

1	Introduzione al corso	5
1.1	I professori	5
1.2	Organizzazione del corso	5
1.3	Organizzazione degli orari	5
1.4	Organizzazione degli esami	6
1.4.1	Esoneri	6
1.4.2	Totale	6
1.4.3	Laboratorio	6
1.4.4	Valutazione	6
1.5	Il corso	7
1.5.1	Il programma	7
1.5.2	Cosa vedremo	7
2	Basi di Dati	8
2.1	Introduzione e definizioni	8
2.1.1	Risorse	8
2.1.2	Basi di Dati	8
2.2	Sistemi informativi e sistemi informatici - una premessa	8
2.2.1	Sistema Informatico	10
2.3	Gestione delle informazioni	10
2.4	Informazioni e dati	11
2.5	Basi di Dati	12
2.5.1	Altra definizione	12
2.5.2	caratteristiche	12
2.5.3	Perché studiare le basi di dati?	13
2.5.4	Basi di dati multimediali	13
2.5.5	Data Base Management System - DBMS	13
2.5.6	Creazione di un database	13
2.5.7	Interrogazione di un database	13
2.6	Schemi e istanze	14

<i>INDICE</i>	3
3 Argomento 2	15
4 Previsione I parziale	16
4.1 Modello dei dati relazionale	16
5 DBMS	17
5.1 Linguaggi per basi di dati	17
5.1.1 Pagine web statiche/dinamiche	17
5.2 Personaggi ed interpreti	18
5.3 Un esercizio per ripassare quanto detto	18
5.3.1 Es1	18
5.3.2 Es2	19
5.3.3 Es3	19
6 Progettazione di una base di dati	20
6.1 Metodologie e modelli	20
6.2 Progettazione	22
6.2.1 Fase di progettazione concettuale	23
6.2.2 Fase di progettazione logica	23
6.2.3 Fase di progettazione fisica	23
7 Introduzione al modello Entità-Relazione	24
7.1 Modello Entità-Relazione	24
7.2 Entità	24
7.2.1 Formalismi	25
7.2.2 Occorrenza (o istanza) di entità	26
7.3 Attributi	26
7.4 Relazione - Associazione	28
7.4.1 Formalismi	28
7.4.2 Istanze di associazione	29
7.5 Entity-Relationship	31
7.5.1 Relazioni ricorsive	32
7.6 Es.	33
7.7 Cardinalità di una relazione	35
7.7.1 Classificazione delle relazioni	35
7.7.2 Cardinalità degli attributi	35
7.8 Identificatori	35
7.9 Riassunto: Identificatori	36
7.10 Ereditarietà	36
7.10.1 Relazione IS-A (o IS-A) tra entità	36
7.11 Generalizzazione tra entità	37

7.12	esempio sessista e antico	38
7.13	Altre proprietà	39
7.14	Riassunto finale: tutto quello che abbiamo visto di ER	39

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 competitini*) → 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 PREMESSA⁹

- *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** \neq **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 (registrate 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 privacy.

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],  
      [Citta]  
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 - privacy - 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 sfiorare 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 programmatori **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 privacy 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...

Verrà 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 -> PL (?)

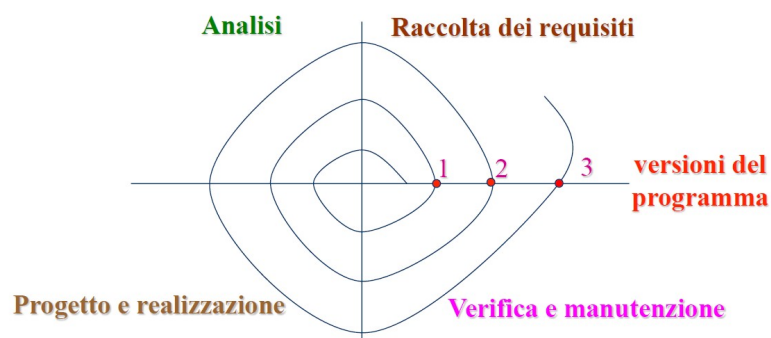
Implementazione: si comincia

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.

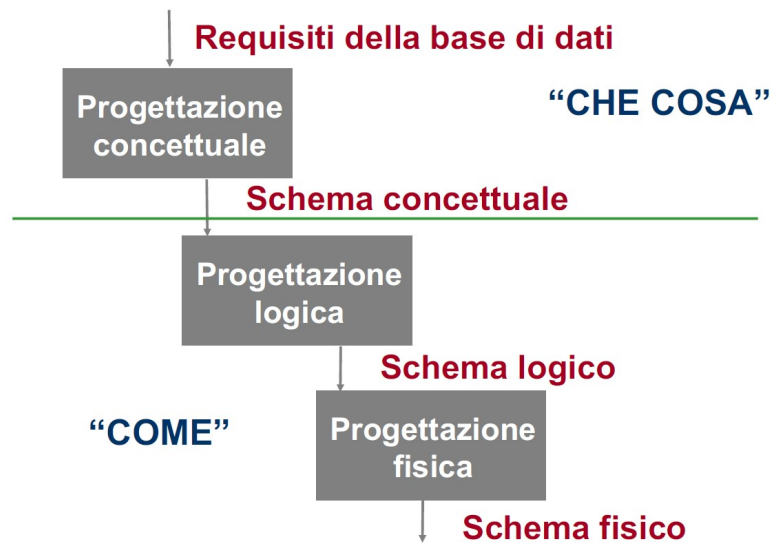
Ciclo di vita (modello a spirale)



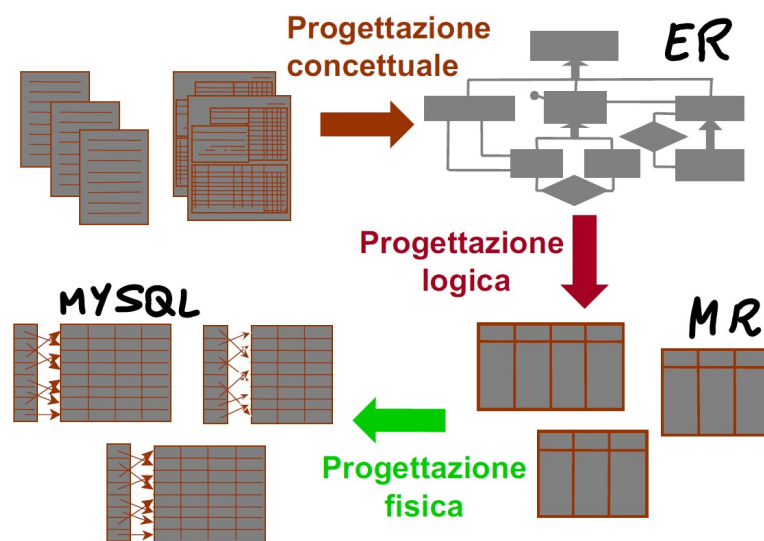
Ha balzato le slide di progettazione ma guardale.

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

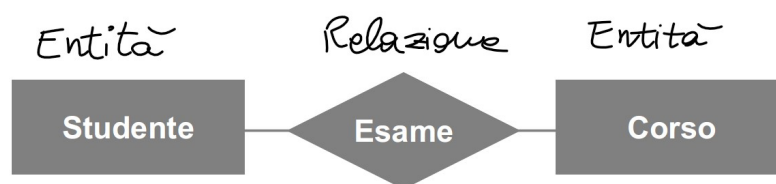
6.2.3 Fase di progettazione fisica

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 è 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.

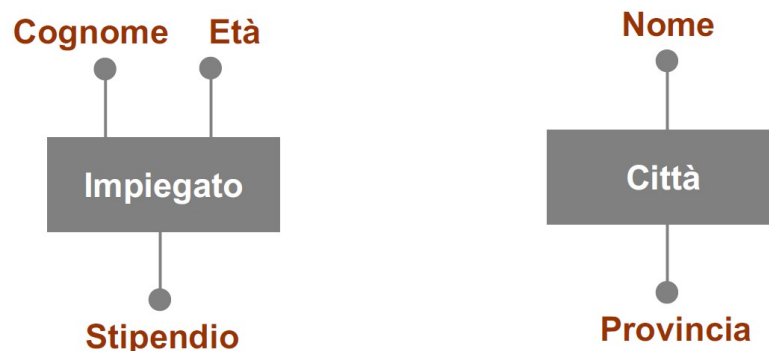
7.2.2 Occorrenza (o istanza) di entità

- 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

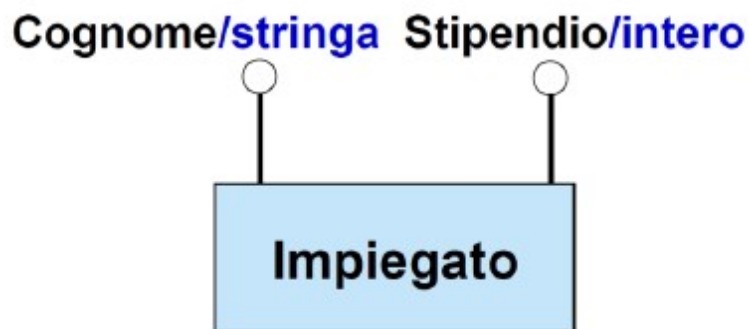
- Un attributo associa ad ogni istanza di entità un valore appartenente ad un insieme detto dominio dell'attributo (tipicamente, interi, caratteri, stringhe, ecc.)
- Si definisce un attributo per l'entità E quando si vuole rappresentare una proprietà locale delle istanze dell'entità E. Una proprietà 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.

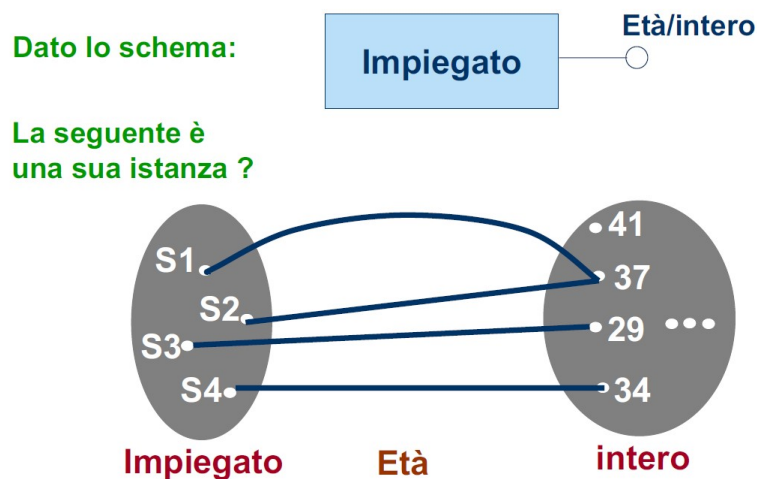


Ogni attributo è definito su un dominio di valori. Un attributo associa ad ogni istanza di entità o associazione un valore nel corrispondente dominio. I domini solitamente non vengono specificati nell'E-R ma nella documentazione associata. Se li si vuole indicare, la notazione è la seguente



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**).

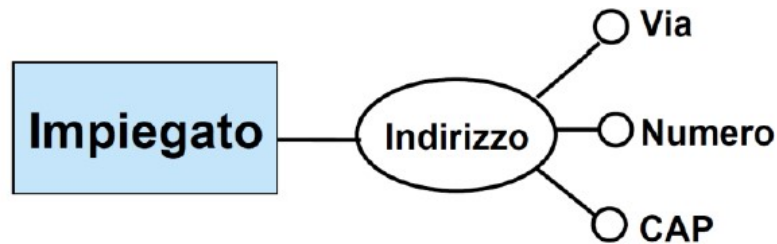
Gli attributi hanno un dominio, che ci specifica di che tipo di dati stiamo parlando.

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

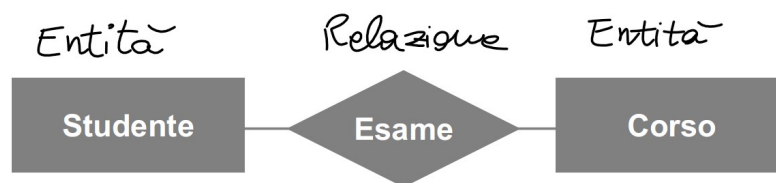
Fatto che descrive un'azione o una situazione e che stabilisce legami logici tra istanze di entità (associa, mette in relazione) nella realtà che stiamo considerando.

I legami possono essere fra più' 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 $istanze(I,R)$) $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots$ 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:

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

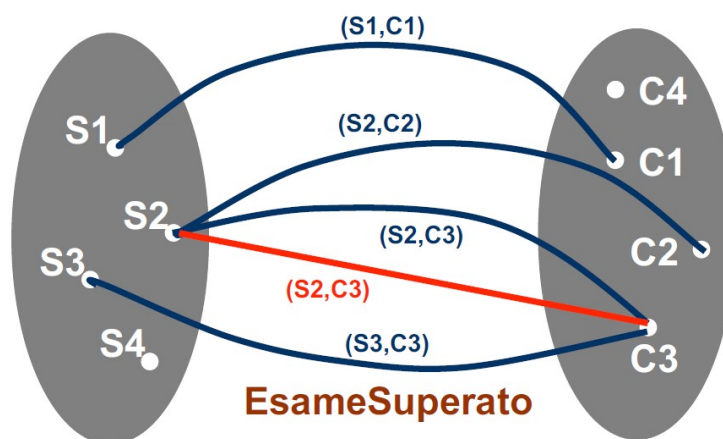
7.4.2 Istanze di associazione

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

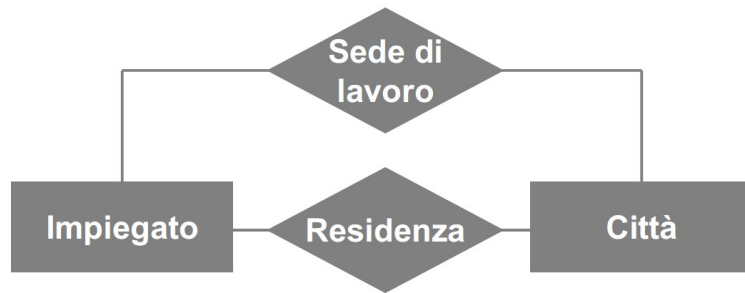
Es.: •Rossi insegna Basi di dati •Batini appartiene all'Università di Milano Bicocca •La ditta Rossi ordina PC •Bianchi lavora al magazzino 4 •Il tornio K22 è installato nell'officina 37 •il TIR 542 viaggia sulla tratta NA-MI

Es.:

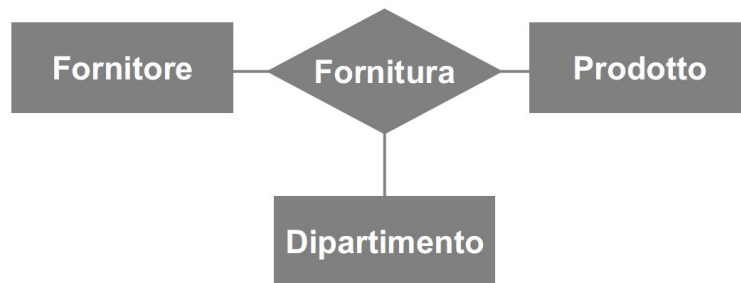
Dalla semantica delle relazioni segue immediatamente che non possono esistere due istanze della stessa relazione che coinvolgono le stesse istanze di entità.



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

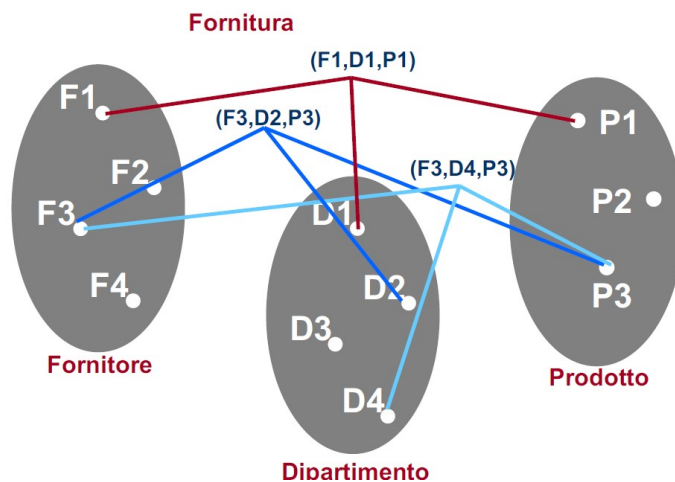


Le relationship possono coinvolgere più di due entità.



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_1, E_2, \dots, E_n è costituita da un insieme di n -ple (o tuple) (x_1, x_2, \dots, x_n) , tali che x_1 è una istanza di E_1 in I , x_2 è una istanza di E_2 in

I_1, \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.

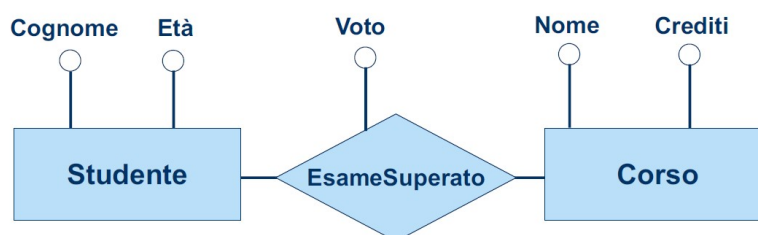
Attributi di relazione

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

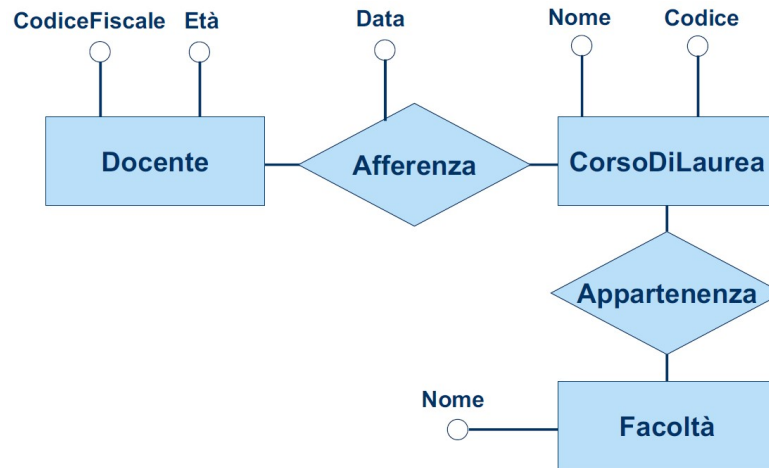
7.5 Entity-Relationship

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

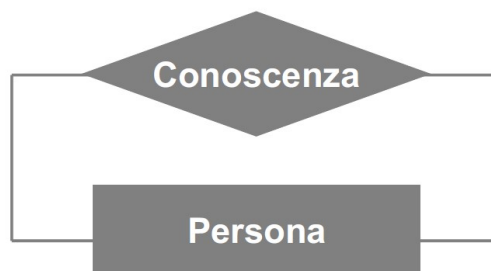
(si dice che afferiscono ai corsi di laurea). Interessa la data di afferenza dei docenti ai corsi di laurea. I corsi di laurea hanno un codice ed un nome, ed appartengono alle facoltà. Ogni facoltà ha un nome.



7.5.1 Relazioni ricorsive

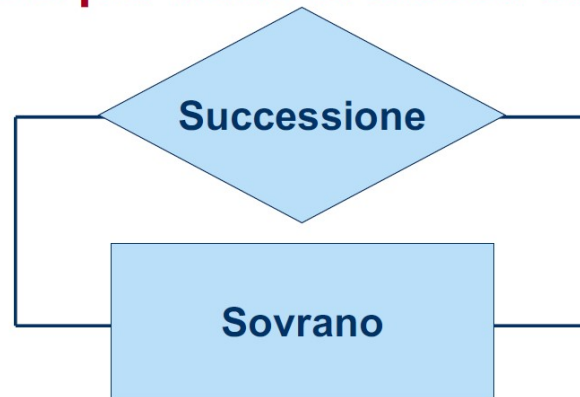
Una associazione può coinvolgere “due o più volte” la stessa entità (associazione ricorsiva o ad anello). Questo ovviamente la fa classificare come relazione **binaria**, anche se c'è rappresentata una sola entità noi stiamo facendo una relazione tra due istanze di quell'entità, quindi conta come due.

Una associazione può coinvolgere “due o più 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, irreflessiva 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.

34CAPITOLO 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 $(1, 1) \rightarrow (0, n)$.

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