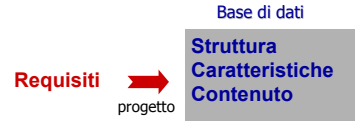


Progettazione di basi di dati

Progettazione di basi di dati



- Metodologia in 3 fasi
 - Progettazione concettuale
 - Progettazione logica
 - Progettazione fisica

Ciclo di vita dei sistemi informativi

- Comprende tutte le attività svolte nello sviluppo e utilizzazione dei sistemi informativi
 - varie figure: analisti, progettisti, utenti
- Le basi di dati sono solo una delle componenti di un sistema informativo
- E' un'attività iterativa... un ciclo di operazioni
 - spesso durante l'esecuzione di una attività occorre rivedere decisioni prese nelle attività precedenti

Fasi del ciclo di vita [1]

1. Studio di fattibilità
 - costi delle alternative, priorità di realizzazione, ...
2. Raccolta ed analisi dei requisiti
 - studio delle proprietà e delle funzionalità del sistema
 - interazione con vari tipi di utenti
 - analisi delle realizzazioni esistenti
 - descrizione informale dei dati e delle operazioni
 - descrizione dei requisiti hardware e software
3. Progettazione
 - progettazione dei dati e delle applicazioni
 - descrizione formali per mezzo di modelli per dati e applicazioni

Fasi del ciclo di vita [2]

4. Implementazione
 - Realizzazione del sistema informativo secondo la struttura prevista nella progettazione
 - Costruzione e popolamento della base di dati
 - Sviluppo del codice delle applicazioni
5. Validazione e collaudo
 - Verifica del corretto funzionamento e della qualità
6. Funzionamento
 - Utilizzo operativo del sistema informativo
 - Installazione, gestione e manutenzione
 - Possono essere richieste
 - Correzioni per malfunzionamenti
 - Revisioni delle funzionalità del sistema

Progettazione di basi di dati [1]

- Progettazione concettuale
 - Rappresentare le specifiche della realtà di interesse (specifiche sui dati) con una descrizione formale
 - Non ci si interessa a come le informazioni verranno codificate e all'efficienza dei programmi che utilizzeranno queste informazioni
 - Si utilizza un modello concettuale dei dati
 - Il prodotto è uno schema concettuale
 - rappresenta il contenuto informativo della base di dati
 - è utile anche a scopo documentativo

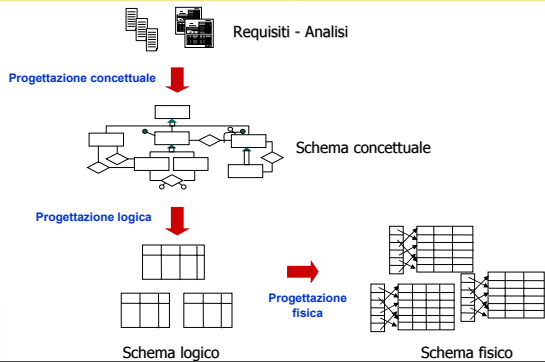
Progettazione di basi di dati [2]

- **Progettazione logica**
 - Lo schema concettuale viene tradotto nel modello di rappresentazione dei dati utilizzato dal DBMS
 - Si utilizza un **modello logico** dei dati
 - Il prodotto è uno **schema logico** dei dati
 - Si effettuano scelte progettuali per ottimizzare le operazioni che saranno effettuate sui dati
 - per il modello relazionale sono definite tecniche di normalizzazione per validare lo schema logico prodotto

Progettazione di basi di dati [3]

- **Progettazione fisica**
 - **Scelta dei parametri fisici di memorizzazione dei dati**
 - Es: organizzazione dei file, definizione degli indici, tipo e dimensioni dei campi, ...
 - Si tiene conto del modello logico e delle specifiche sulle operazioni per ottimizzare le prestazioni
 - Si fa riferimento ad un **modello fisico** dei dati
 - Il prodotto è uno **schema fisico** che dipende dal particolare DBMS utilizzato

Fasi della progettazione



Modelli concettuali

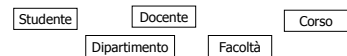
- Permettono di rappresentare i dati in modo indipendente da ogni sistema di memorizzazione
 - cercano di descrivere i concetti del mondo reale
 - sono utilizzati nelle fasi preliminari di progettazione
- Hanno **costrutti** per la rappresentazione di:
 - classi dei dati di interesse
 - correlazioni fra le classi di dati
- Solitamente ad ogni costrutto corrisponde una rappresentazione grafica

Il modello E-R

- Il modello **Entità-Relazione** è uno dei più diffusi modelli concettuali
- I costrutti del modello E-R sono:
 - Entità
 - Relazioni
 - Attributi
 - Identificatori
 - Generalizzazioni e sottoinsiemi
- Uno schema E-R è rappresentato con un diagramma

Entità e relazioni

- **Rappresentazione delle entità**

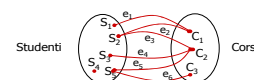


- **Rappresentazione delle relazioni**

- ogni relazione ha un nome ed è rappresentata con un **rombo** e da linee che connettono la relazione con le sue componenti

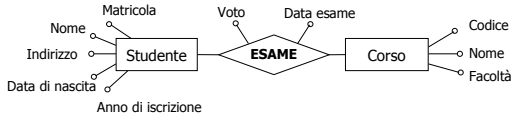


- esempio di istanze della relazione ESAME



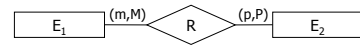
Attributi

- Descrivono proprietà elementari di entità o relazioni
- Ogni attributo associa a ciascuna istanza un valore appartenente al **dominio** dell'attributo
 - i domini non sono riportati nello schema ma sono descritti nella documentazione associata



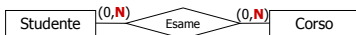
Cardinalità delle relazioni

- Per ogni partecipazione di una entità ad una relazione si specifica il **numero minimo** e **massimo** di volte che un'istanza dell'entità può essere coinvolta
 - Rappresenta un vincolo



ogni istanza di E_1 partecipa ad almeno m e al più a M istanze di R
 ogni istanza di E_2 partecipa ad almeno p e al più a P istanze di R

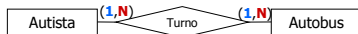
Relazioni “molti a molti”



Uno studente può non aver fatto esami o averne fatti un certo numero
 Nessuno studente o più studenti possono aver sostenuto l'esame di un corso



Una montagna può scalata o da molti alpinisti o da nessuno
 Un alpinista ha scalato almeno una montagna....



Un autista ha almeno un turno
 Ogni autobus è usato almeno in un turno

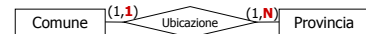
Relazioni “uno a molti”



Ogni Persona ha una sola Città di residenza
 Una Città può non avere Cittadini residenti



Una Persona è disoccupata o ha un impiego
 Una Società ha almeno un impiegato



Ogni Comune è in una sola Provincia
 Ogni Provincia ha almeno un Comune

Relazioni “uno a uno”



Un uomo è o meno sposato con una donna
 ... e viceversa



Ogni Patente ha un unico titolare
 Ogni persona può o meno avere la Patente



Ogni Spettatore acquista un solo Biglietto

Identificatori delle entità

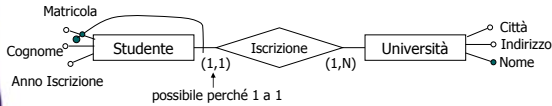
- Permettono di identificare in modo univoco le istanze dell'entità
 - ogni entità deve avere almeno un identificatore
- Tipologie di identificatori
 - **interno (chiave)**
attributi (uno o più) dell'entità
 - **esterno**
entità esterne attraverso relazioni con cardinalità (1,1) ed eventualmente attributi
- Le relazioni sono identificate dagli attributi delle entità coinvolte

Esempi di identificatori

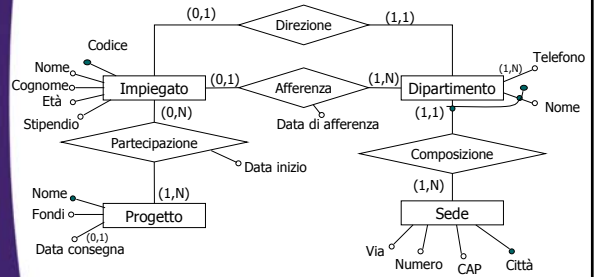
Identificatori interni



Identificatori esterni



Uno schema ER completo



Generalizzazione

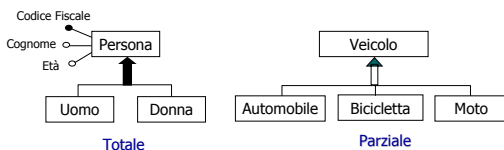
- Rappresenta un legame fra
 - Entità padre E
 - Entità figlie $E_1, E_2, \dots, E_n$ (specializzazioni o sottotipi)
- E è un'entità più generale delle entità figlie, nel senso che le comprende come casi particolari
 - Ogni istanza di E_i è anche istanza di E
 - Ogni istanza di E è istanza al più di una E_i
- Ereditarietà
 - Ogni proprietà di E (attributi, identificatori, relazioni e altre generalizzazioni) è ereditata anche dalle E_i

Esempi di generalizzazione



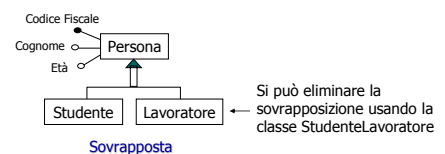
Generalizzazione totale

- La generalizzazione è **totale** se ogni istanza della classe padre è una istanza di almeno una delle figlie
- Altrimenti si dice **parziale**



Generalizzazione esclusiva

- La generalizzazione è **esclusiva** se ogni istanza della classe padre è una istanza di al più una delle classi figlie
- Altrimenti si dice **sovrapposta**
 - Si possono eliminare le classi sovrapposte introducendo le classi intersezione



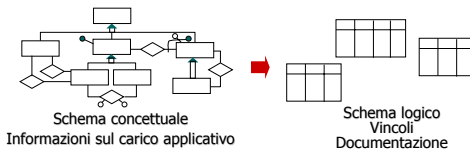
Generalizzazione "is-a"

- Nel caso di generalizzazione con un solo figlio si parla di **sottoinsieme (gerarchia is-a)**
- In tal caso la generalizzazione non può essere totale
 - le due entità sarebbero coincidenti



Progettazione logica

Progettazione logica



Costruzione di uno schema logico che descrive in modo corretto e efficiente tutte le informazioni presenti nello schema ER

Input: schema concettuale + informazioni sul carico applicativo + modello logico prescelto

Output: schema logico con vincoli e documentazione

Verso il modello logico

Riorganizzazione dello schema ER

- **semplificazione della traduzione**
 - non tutti i costrutti del modello ER hanno una traduzione naturale nel modello logico desiderato
 - Entità → relazione del modello relazionale con gli stessi attributi
 - Generalizzazione → ?
- **ottimizzazione del progetto**
 - si deve scegliere la soluzione che garantisce le migliori prestazioni

Traduzione verso il modello logico

- Ottimizzazioni mediante tecniche proprie del modello logico (normalizzazione per il modello relazionale)

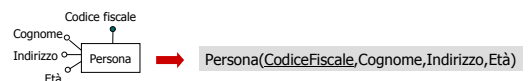
Ristrutturazione degli schemi ER

Schema ER iniziale + specifiche sul carico applicativo

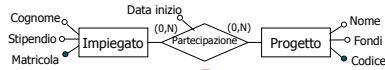
- **Analisi delle ridondanze**
 - Mantenere/rimuovere le ridondanze dello schema iniziale
- **Eliminazione delle generalizzazioni**
 - Le generalizzazioni sono analizzate e sostituite da altri costrutti
- **Partizionamento/accorpamento di entità e associazioni**
 - Suddivisione di un concetto (entità o relazione) in più concetti
 - Accorpamento di più concetti separati in un solo concetto
- **Scelta degli identificatori primari**
 - Per le entità che hanno più di un identificatore

Traduzione verso il modello relazionale

- Si traduce lo schema ER ristrutturato in uno schema relazionale equivalente
- **Trasformazioni di base**
 - Una entità diviene una relazione definita sugli stessi attributi e con chiave uguale all'identificatore
 - Una relazione ER diventa una relazione sulle chiavi delle relazioni che rappresentano le entità coinvolte (più gli attributi propri)



Relazioni multi-a-molti [1]



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

- La coppia di attributi Impiegato, Progetto identifica una istanza della relazione ER Partecipazione (è la chiave)
- Esiste un vincolo di integrità referenziale fra
 - Partecipazione.Impiegato e Impiegato.Matricola
 - Partecipazione.Progetto e Progetto.Codice

Relazioni multi-a-molti [2]

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

Impiegato

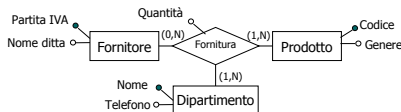
Cognome	Stipendio	Matricola
Rossi	500	0801
Verdi	800	0802
Bianchi	400	0823

Codice	Fondi	Nome
P0032	3500	Alpha
P0054	7800	Beta
P0012	2400	Gamma

Impiegato	DataInizio	Progetto
0801	22/3/2001	P0032
0801	12/11/2001	P0012
0802	10/7/2001	P0054

Progetto

Relazioni multi-a-molti [3]



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

- Esiste un vincolo di integrità referenziale fra
- Fornitura.Fornitore e Fornitore.PartitaIVA
 - Fornitura.Prodotto e Prodotto.Codice
 - Fornitura.Dipartimento e Dipartimento.Nome

Relazioni multi-a-molti [4]

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

NomeDitta	PartitaIVA
Rossi	08009382
Verdi	07092913
Bianchi	04563281

Codice	Genere
A0034	Computer
B3456	Stampante
V0567	Video

Fornitore

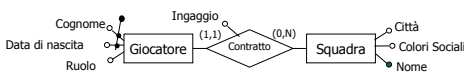
Fornitore	Prodotto	Dipartimento	Quantità
07092913	A0034	Fisica	5

Prodotto

Nome	Telefono
Ingegneria	0577233601
Fisica	0577232471
Biologia	0577234632

Dipartimento

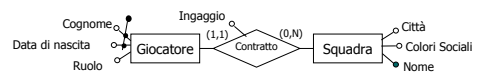
Relazioni uno-a-molti [1]



Giocatore(Cognome, DataNascita, Ruolo)
Contratto(Giocatore, Squadra, Ingaggio)
Squadra(Nome, Città, ColoriSociali)

- La chiave per la relazione Contratto è solo l'identificatore dell'entità Giocatore perché la cardinalità verso la relazione è (1,1)
- Le relazioni Giocatore e Contratto hanno un'unica chiave e quindi possono essere fuse insieme

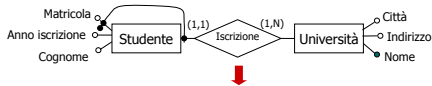
Relazioni uno-a-molti [2]



Giocatore(Cognome, DataNascita, Ruolo, NomeSquadra, Ingaggio)
Squadra(Nome, Città, ColoriSociali)

- Esiste un vincolo di integrità referenziale fra
 - Giocatore.NomeSquadra e Squadra.Nome
- La soluzione è valida anche se la cardinalità della relazione Contratto verso Giocatore è (0,1)
 - Si hanno giocatori con l'attributo NomeSquadra uguale a NULL

Relazioni con identificatore esterno [1]



Studente(Matericola, NomeUniversità, Cognome, AnnoIscrizione)
Università(Nome, Città, Indirizzo)

- Esiste un vincolo di integrità referenziale fra
 – Studente.NomeUniversità e Università.Nome
- Rappresentando l'identificatore esterno viene rappresentata anche la relazione fra le due entità

Relazioni con identificatore esterno [2]

Studente(Matericola, NomeUniversità, Cognome, AnnoIscrizione)
Università(Nome, Città, Indirizzo)

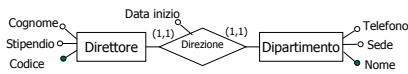
Studente

Matericola	NomeUniversità	Cognome	AnnoIscrizione
A80023633	UNISI	Rossi	1993
A80023633	UNIFI	Verdi	1996
A80023635	UNISI	Bianchi	2000

Nome	Città	Indirizzo
UNISI	Siena	Via Banchi di sotto
UNIFI	Firenze	P.zza S. Marco

Università

Relazioni uno-a-uno [1]



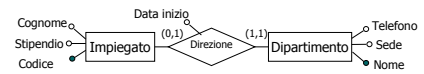
Direttore(Codice, Cognome, Stipendio, DipartimentoDiretto, InizioDirezione)
Dipartimento(Nome, Telefono, Sede)

oppure

Direttore(Codice, Cognome, Stipendio)
Dipartimento(Nome, Telefono, Sede, Direttore, InizioDirezione)

- La relazione è **biunivoca** e quindi può essere rappresentata in una qualsiasi delle relazioni che rappresentano le entità
- Si potrebbe anche definire un'unica relazione che fonde entrambe ma si perde la distinzione fra le due entità

Relazioni uno-a-uno [2]



Impiegato(Codice, Cognome, Stipendio)
Dipartimento(Nome, Telefono, Sede, Direttore, InizioDirezione)

- In questo caso la partecipazione della prima entità è **opzionale**
- Rappresentare la relazione Direzione in Dipartimento piuttosto che Impiegato è preferibile perché nell'altro caso si avrebbero molti valori nulli

Relazioni uno-a-uno [3]



Impiegato(Codice, Cognome, Stipendio)
Direzione(Direttore, Dipartimento, DataInizioDirezione)
Dipartimento(Nome, Telefono, Sede)

- In questo caso la partecipazione di entrambe le entità è opzionale
- Vantaggio: non si hanno valori nulli sugli attributi
- Occorre però introdurre una relazione in più
 – *Conviene se poche istanze partecipano alla relazione*