Introduzione ai Database

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

Le basi di dati

- Le basi di dati costituiscono uno dei maggiori campi di applicazione dell'informatica
- Qualunque organizzazione (azienda, usl, scuola, ...) ha un proprio sistema informativo, non necessariamente informatizzato, costituito da:
  - schedari e archivi cartacei: contengono le informazioni, organizzate secondo una certa struttura (schede, campi, codici, ...)
  - connessioni: gli archivi sono logicamente collegati tra loro, in funzione delle esigenze informative e operative (rimandi, collegamenti ad altre schede)
- Un data base è un insieme di archivi informatizzati connessi tra loro opportunamente, che rende possibile la consultazione e l'aggiornamento in tempo reale delle informazioni

Le basi di dati

- Caratteristiche e funzionalità fondamentali di un data base:
  - consistenza: coerenza interna alla base di dati, tra le diverse informazioni in essa rappresentate
  - indipendenza: delle modalità di accesso ai dati dalla struttura fisica di memorizzazione
  - concorrenza: accesso contemporaneo ai dati da parte di diversi operatori e programmi
  - integrità (robustezza): protezione e ripristino dei dati in caso di guasti hardware, crash di sistema ...
  - sicurezza (privatezza): accesso controllato e selettivo alle informazioni da parte di operatori abilitati, con diversi ruoli
  - efficienza, scalabilità, amministrazione, ...

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

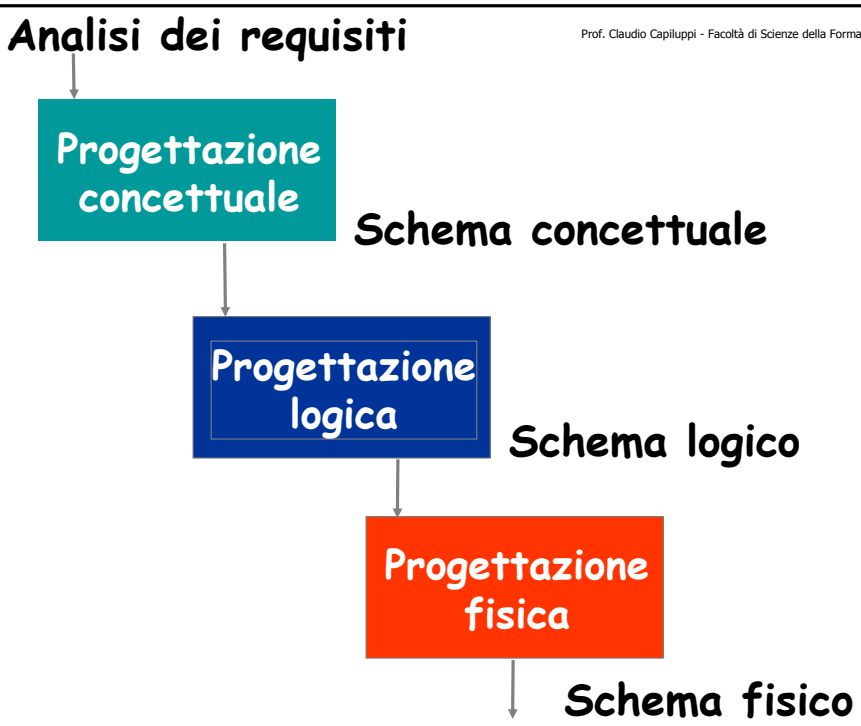
Le basi di dati

- DBMS (*data base management system*): software specializzato progettato per semplificare la realizzazione e la gestione di un data base, tale da consentire una adeguata informatizzazione di un sistema informativo tradizionale
- Un DBMS deve consentire la realizzazione di interrogazioni (query) e di applicazioni (procedure) basate sui dati contenuti nel database
- Un DBMS deve garantire le funzionalità appena elencate, prima tra tutte l'*indipendenza* dell'accesso alle informazioni dalla struttura fisica dei dati:
  - indipendenza tra la "struttura logica" (definizione dei dati ad alto livello) e la struttura fisica (dettagli di implementazione relativi a files e dispositivi dove vengono registrati i dati)
  - indipendenza tra dati e interrogazioni (viste o query): SQL
  - indipendenza tra dati e programmi: una modifica alla struttura fisica di un dato non deve comportare la necessità di modificare le procedure che lo utilizzano
- Soluzioni:
  - Dati: introduzione del riferimento alle informazioni per nome invece del riferimento per posizione fisica all'interno del file/record
  - Connessioni: diverse architetture

Le basi di dati

- Architetture, ovvero tipologie di DBMS:
  - Reticolare / Gerarchico / **Relazionale** / Ad Oggetti, ...
- Reticolare e Gerarchico:
  - utilizzano riferimenti fisici diretti fra record: introduzione nel db di dati fittizi (puntatori fisici), che non rappresentano informazioni reali, aggiunti allo scopo di creare connessioni tra record di diversi archivi
  - le interrogazioni possibili dipendono dalla struttura dei puntatori previsti
- Relazionale
  - basato esclusivamente sui valori dei dati rappresentati nel database
  - anche i collegamenti fra entità diverse sono rappresentati per mezzo dei valori reali assunti dai dati chiave corrispondenti (puntatori "logici")
  - qualsiasi query è ottenibile mediante operatori standard ("algebra relazionale")
- Ad Oggetti:
  - più recente e ancora poco diffusa, ispirata alla programmazione ad oggetti:
  - una classe (di oggetti) consiste non solo di dati (*proprietà*), ma anche delle procedure (*metodi*) per l'accesso e la manipolazione controllata dei dati stessi
- Principali DBMS relazionali:
  - Server (proprietari): Oracle, DB2, Ingres, SQL Server, Caché
  - Server (open-source): MySQL, PostgreSQL
  - Desktop: Jet (engine) – Access (dbms), MSDE

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06



Progettazione di una base di dati

- Analisi dei requisiti
  - Cosa vogliamo realizzare ? Quale realtà (mondo) vogliamo rappresentare nella base di dati
  - Quali dati memorizzare
  - Quali applicazioni vogliamo realizzare su di essi
  - Quali operazioni sono più frequenti, o devono essere particolarmente efficienti
- Progettazione Concettuale:
  - Modello E-R (Entity-Relationship): modello di rappresentazione della realtà
  - Modello UML (Unified Modelling Language): stessi concetti, simbolismo diverso
- Progettazione Logica:
  - Modello Relazionale: nuovo modello di ispirazione matematica ("relazione")
  - Normalizzazione: teoria sistematizzata per raffinare lo schema logico, attraverso l'analisi delle dipendenze funzionali
- Progettazione Fisica:
  - Tiene conto delle specifiche del DBMS che si utilizza per implementare la base di dati reale
  - Definizione di tipo e dimensioni di memoria dei singoli dati da memorizzare
  - Si considerano il carico di lavoro e le prestazioni del sistema, e si valutano ulteriori modifiche allo schema per rendere più efficiente il database:
    es. creazione di indici per velocizzare l'accesso a particolari informazioni

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

Analisi dei requisiti

- L' Analisi dei requisiti consiste nella individuazione :
  - dei problemi che il sistema dovrà permettere di risolvere
  - delle caratteristiche e funzionalità che l'applicazione dovrà garantire
  - aspetti statici (strutture dati) e dinamici (operazioni sui dati)
- Fonti per l'individuazione dei requisiti del sistema:
  - utenti dell'applicazione: utenti diversi possono fornire indicazioni diverse, in genere gli utenti a livello più alto possiedono una visione più ampia, ma meno dettagliata
  - documenti esistenti: moduli, regolamenti, procedure, normative interne all'organizzazione, già elementi costitutivi del sistema informativo
  - applicazioni preesistenti: da rimpiazzare o con cui il nuovo sistema dovrà interagire
- Si parte cercando di individuare gli aspetti essenziali, per procedere poi al raffinamento per approssimazioni successive: in questa fase sarà necessaria una notevole interazione tra analista-progettista e utenti
- Si arriva alla definizione delle specifiche, relative ai dati e alle operazioni sui dati che devono essere garantite, espresse generalmente in linguaggio naturale

La progettazione concettuale

Il modello Entità-Relazione

Progettazione concettuale

- La progettazione concettuale è un modo di formalizzare una rappresentazione della realtà vicino al modo di pensare umano
- Processi logici che intervengono nella progettazione concettuale:
  - Astrazione: separazione di alcuni aspetti della realtà da altri, cui sono in effetti connessi; di fronte alla complessità della realtà si opera una scelta distinguendo le caratteristiche che ci interessano dalle altre, che vengono ignorate
  - Unificazione: riduzione di più oggetti (parti) ad una sola unità
  - Classificazione: suddivisione di un insieme di unità/oggetti in classi
  - Generalizzazione: creazione di nuove classi più ampie, a partire da classi già formate
- Schema concettuale: rappresentazione del contenuto informativo della base di dati, in termini puramente concettuali:
  - senza preoccuparsi delle modalità di codifica e registrazione
  - né dell'efficienza delle procedure che faranno uso di queste informazioni, ovvero delle modalità di accesso ai dati
  - senza riferimento a un sistema informatico reale

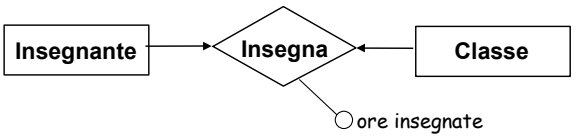
Progettazione concettuale

- **Entità**: insieme di oggetti del mondo reale distinguibili da altri oggetti, che possiamo raggruppare in una classe
- Una entità è caratterizzata da un insieme di **attributi**
  - La scelta degli attributi riflette la nostra analisi della realtà da cui abbiamo astratto gli aspetti di interesse: descrivono il livello di dettaglio con il quale vogliamo rappresentare le informazioni sulle entità
  - Un sottoinsieme minimale di attributi che identificano univocamente gli elementi dell'entità prende il nome di chiave (candidata)
  - Se sono presenti più chiavi candidate, una di esse viene designata come **chiave primaria**

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

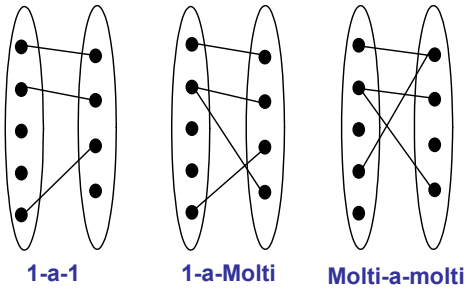
Progettazione concettuale

- **Relazione** o **Associazione**: connessione logica tra entità
  - di solito una associazione coinvolge due entità (associazione binaria), ma può coinvolgere anche più entità (es. ternaria), o connettere una entità con se stessa
  - una associazione è caratterizzata in primo luogo dalla **cardinalità** delle entità coinvolte: numero (minimo e massimo) di occorrenze di una associazione a cui un elemento dell'entità può partecipare
  - anche le associazioni possono avere *propri* attributi descrittivi: ad es. l'associazione Insegna tra le entità Insegnante e Classe ha l'attributo *ore insegnate*



Progettazione concettuale

- **Cardinalità** di una associazione tra entità: numero (minimo e massimo) di occorrenze della relazione a cui un elemento dell'entità può partecipare
  - l'associazione più comune è quella uno-a-molti (1:n): a un elemento di una entità corrispondono più elementi dell'altra entità
  - le associazioni uno-a-uno (1:1): sono abbastanza rare e, quando non vi siano altre considerazioni a sconsigliarlo, le due entità possono essere *unificate*
  - le associazioni molti-a-molti (n:m): a un elemento di una entità corrispondono più elementi dell'altra entità e viceversa; non sono così frequenti, ma sono le più problematiche
    es. l'associazione Insegna tra le entità Insegnante e Classe è di tipo (n:m)



Modello E-R

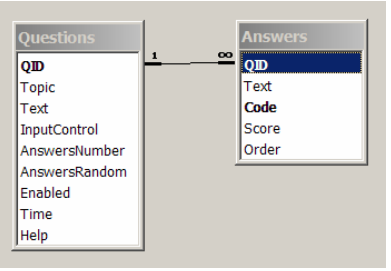
- Grafo E-R
  - le entità si rappresentano con i rettangoli
  - gli attributi si rappresentano con dei pallini (pieni se costituiscono la chiave primaria)
  - le associazioni si rappresentano con le losanghe (rombi)

  - la freccia indica che gli elementi dell'entità possono entrare nella relazione una o più volte
  - le cardinalità possono essere indicate con la notazione (non molto intuitiva):

es. una persona può risiedere in una sola città, nella quale possono vivere (da 0 a) n persone

Modello UML

- Rappresentazione UML
  - le entità vengono rappresentate da rettangoli, contenenti :
    - in alto il nome dell'entità
    - all'interno: gli attributi (evidenziando gli attributi che costituiscono la chiave primaria)
  - le associazioni sono indicate con una linea di collegamento tra le entità:
    - le cardinalità vengono riportate (più intuitivamente) accanto alla entità, ed in pratica risultano invertite rispetto al diagramma ER
    - notazione molto vicina alla logica del modello relazionale
      es. una domanda può avere n risposte:
  - La formalizzazione UML è molto vicina alla progettazione logica, utilizzata nella progettazione dei database relazionali, e adottata da molti DBMS (e CASE), incluso Access



Progettazione concettuale

- Top-down: si parte da concetti molto generali ed astratti per andare verso concetti più concreti e dettagliati
- Bottom-Up: si parte dai requisiti dettagliati, raggruppandoli in concetti più astratti
- Un buon punto di partenza è costruire un glossario dei termini a partire dalle specifiche definite nella fase di analisi dei requisiti, descritte generalmente in linguaggio naturale:
  - utile non solo per sanare ambiguità, individuare sinonimi, unificare i termini, del linguaggio naturale utilizzati nella definizione delle specifiche
  - ma soprattutto per individuare i concetti principali, che costituiranno le entità del mondo che stiamo andando a rappresentare

Progettazione concettuale

- Il progetto E-R è notevolmente soggettivo: ci sono spesso molti modi per modellare un dato scenario
- Scelte di progetto:
  - un concetto dovrebbe essere modellato come una entità o come un attributo ?
  - un concetto dovrebbe essere modellato come una entità o come una relazione?
  - identificare le relazioni: binarie o ternarie (n-arie) ?
- Indicazioni generali per la modellazione dei concetti:
  - se un concetto ha proprietà significative e descrive oggetti con esistenza autonoma -> entità
  - se è un concetto semplice e non ha proprietà rilevanti -> attributo (di un altro concetto a cui è riferibile)
  - se correla due o più concetti, già individuati come entità -> relazione
  - se una relazione è di cardinalità (n:m) e/o ha proprietà proprie -> entità
  - se un concetto è un caso particolare di un altro -> generalizzazione entità
- Lo schema concettuale risultante verrà in seguito rianalizzato e ristrutturato nella fase di **progettazione logica**, con una metodologia sistematica, oggettiva e ben formalizzata, di derivazione matematica

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

Progettazione concettuale

- *Esempio: Indirizzo* dovrebbe essere modellato come un attributo di Utente o invece come una entità, connessa a Utente da una relazione ?
- Dipende dalla realtà che dobbiamo rappresentare e dall'uso che vogliamo fare delle informazioni sull'indirizzo:
  - per prima cosa bisogna valutare se considerare l'indirizzo come una semplice stringa alfanumerica, o strutturarlo invece come un attributo *complesso*, costituito di un certo numero di attributi *atomici* : via, n.civico, località, comune, provincia, nazione, CAP, ...
  - se la struttura dell'attributo (via, n., località, ...) è importante per effettuare ricerche su una sua parte (es. trovare gli Utenti che risiedono in una determinata via), sarà conveniente modellare *indirizzo* come una entità, strutturata nei suoi attributi atomici
  - se si possono avere diversi indirizzi per utente, il concetto deve essere modellato come entità, perché gli attributi non possono (non devono) assumere valori multipli
  - a maggior ragione se l'attributo è complesso, per evitare una proliferazione di attributi ripetuti in Utente
- In generale, quando abbiamo un attributo che può concettualmente assumere più valori, cioè occorre poter registrare valori multipli dell'attributo per un elemento di una entità, si possono avanzare due soluzioni:
  - soluzione da evitare: moltiplicare il numero di attributi (indirizzo1, indirizzo2, ...)
  - soluzione corretta: modellare l'attributo come una nuova entità, associata alla prima da una relazione 1:n

Progettazione concettuale

- Esempio: Società di formazione
- Si vuole realizzare una base di dati per una società che eroga corsi, di cui vogliamo rappresentare i dati dei partecipanti ai corsi e dei docenti. Per gli studenti (circa 5000), identificati da un codice, si vuole memorizzare il codice fiscale, il cognome, l'età, il sesso, il luogo di nascita, il nome degli attuali datori di lavoro, i posti dove hanno lavorato in precedenza insieme al periodo, l'indirizzo e il numero di telefono, i corsi che hanno frequentato (in tutto circa 200) e il giudizio finale.
- Rappresentiamo anche i seminari che stanno attualmente frequentando e, per ogni giorno, i luoghi e le ore dove sono tenute le lezioni. I corsi hanno un codice, un titolo e possono avere varie edizioni con date di inizio e fine e numero di partecipanti. Per gli studenti liberi professionisti, vogliamo conoscere l'area di interesse e il titolo. Per quelli che lavorano alle dipendenze, vogliamo conoscere invece il livello e la posizione ricoperta.
- Per gli insegnanti (circa 300), rappresentiamo il cognome, l'età, il luogo di nascita, il nome del corso che insegnano, quelli che hanno insegnato nel passato e quelli che possono insegnare, e tutti i recapiti telefonici. I docenti possono essere dipendenti interni della società o collaboratori esterni.

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Partecipante	Persona che partecipa ai corsi	Studente	Corso Società
Docente	Docente dei corsi Può essere interno o esterno	Insegnante	Corso
Corso	Corso organizzato dalla società Può avere più edizioni	Seminario	Docente
Società	Ente presso cui i partecipanti lavorano o hanno lavorato	Posti	Partecipante

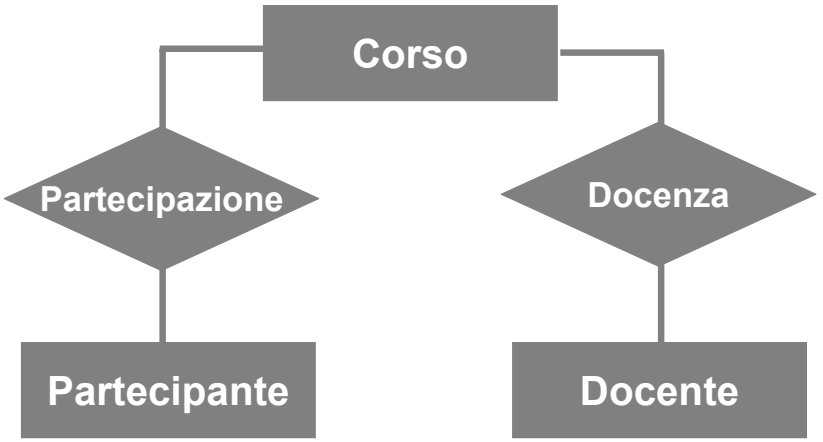
...

Frasi di carattere generale

Si vuole realizzare una base di dati per una società che eroga corsi, di cui vogliamo rappresentare i dati dei partecipanti ai corsi e dei docenti.

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

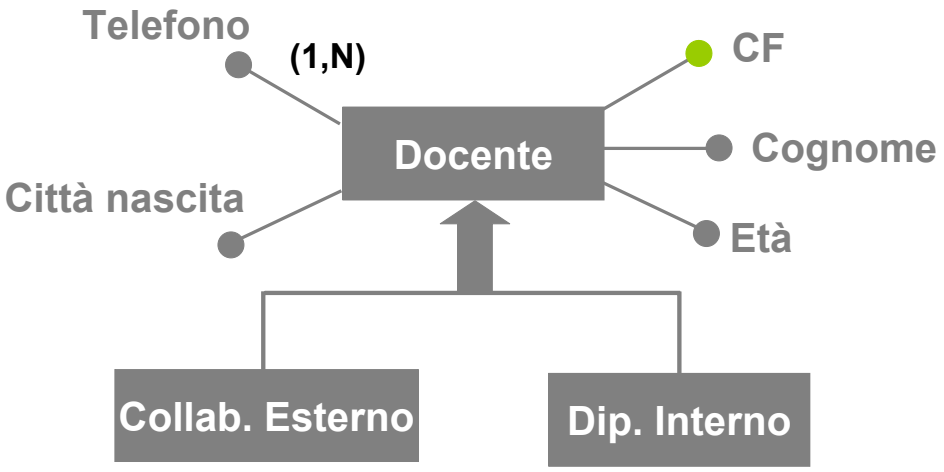
Schema scheletro



Frasi relative ai docenti

Per i docenti (circa 300), rappresentiamo il cognome, l'età, la città di nascita, tutti i numeri di telefono, il titolo del corso che insegnano, di quelli che hanno insegnato nel passato e di quelli che possono insegnare. I docenti possono essere dipendenti interni della società di formazione o collaboratori esterni.

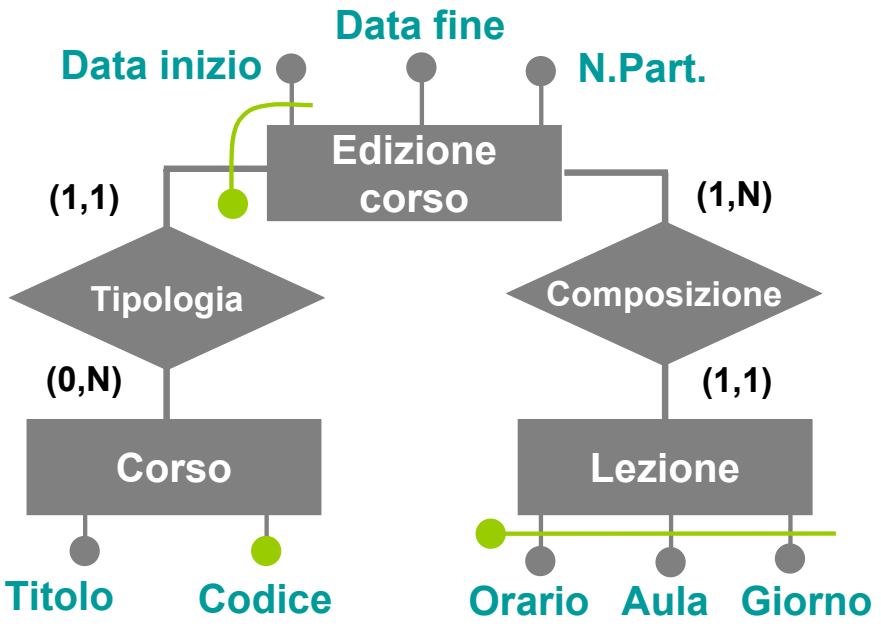
Raffinamento schema



Frasi relative ai corsi

Per i corsi (circa 200), rappresentiamo il titolo e il codice, le varie edizioni con date di inizio e fine e, per ogni edizione, rappresentiamo il numero di partecipanti e il giorno della settimana, le aule e le ore dove sono tenute le lezioni.

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06



Frasi relative ai partecipanti

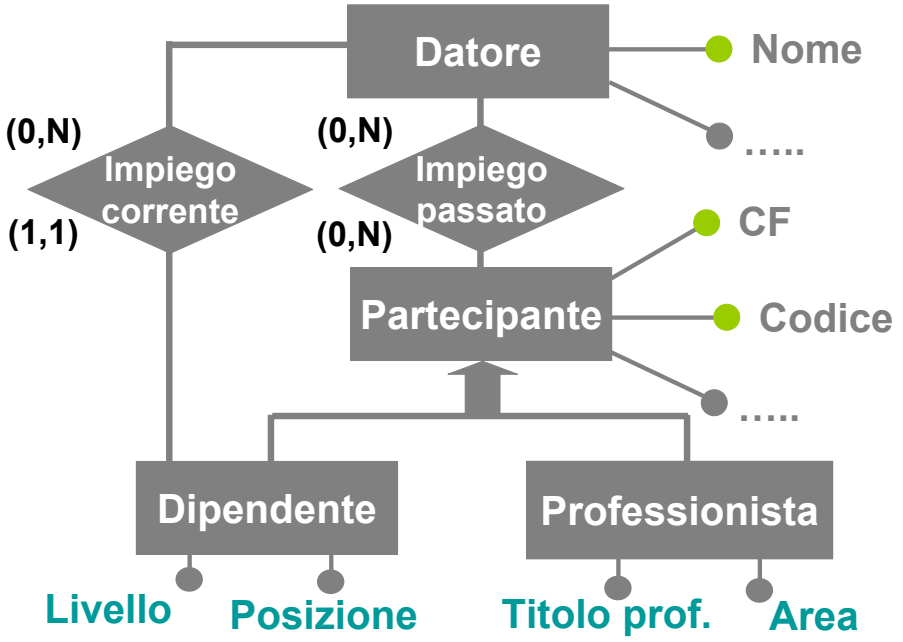
Per i partecipanti (circa 5000), identificati da un codice, rappresentiamo il codice fiscale, il cognome, l'età, il sesso, la città di nascita, i nomi dei loro attuali datori di lavoro e di quelli precedenti (insieme alle date di inizio e fine rapporto), le edizioni dei corsi che stanno attualmente frequentando e quelli che hanno frequentato nel passato, con la relativa votazione finale in decimi.

Frasi relative a tipi specifici di partecipanti

Per i partecipanti che sono liberi professionisti, rappresentiamo l'area di interesse e, se lo possiedono, il titolo professionale. Per i partecipanti che sono dipendenti, rappresentiamo invece il loro livello e la posizione ricoperta.

Frasi relative ai datori di lavoro dei partecipanti

Relativamente ai datori di lavoro presenti e passati dei partecipanti, rappresentiamo il nome, l'indirizzo e il numero di telefono.



Progetto (semplice, per esercizio)

- Base di dati bibliografica
- Si vogliono organizzare i dati di interesse per automatizzare la gestione dei riferimenti bibliografici, con tutte le informazioni da riportarsi in una bibliografia. Le pubblicazioni sono di due tipi, monografie (per le quali interessano editore, data e luogo di pubblicazione) e articoli su rivista (con nome della rivista, volume, numero, pagine e anno di pubblicazione); per entrambi i tipi si debbono ovviamente riportare i nomi degli autori. Per ogni pubblicazione deve esistere un codice identificativo.

Ridondanza e Consistenza

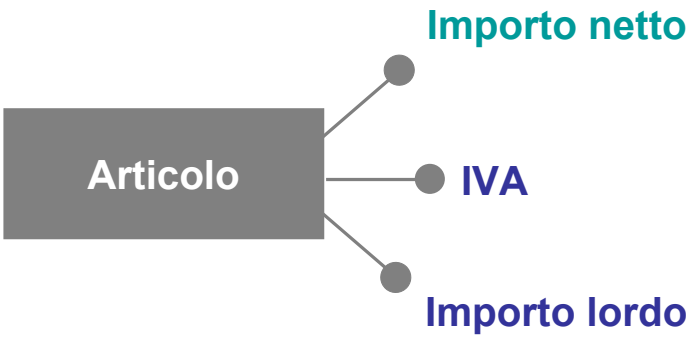
- Uno schema è minimale quando tutte le specificazioni dei concetti sono rappresentate una sola volta nel modello; non è minimale quando esistono delle ridondanze
- Una ridondanza in uno schema E-R è una informazione significativa ma derivabile da altre:
  - duplicazione di concetti (entità o attributi) e quindi, in definitiva, duplicazione di informazioni nel database
  - concetti (attributi) che possono essere derivati da altri, di solito attraverso funzioni di aggregazione (somma, conteggio, ...)
  - associazione derivabile dalla composizione di altre associazioni presenti
- In presenza di forme di ridondanza nel modello, si pone il problema del mantenimento della coerenza interna (*consistency*) della base di dati, tra informazioni duplicate, e di quelle derivabili da altre
- Problematica della ridondanza: ogni forma di ridondanza nella base di dati è una possibile fonte di incoerenza
- Si tratta di un problema concettuale, che ha un impatto di estrema importanza sulla gestione operativa del database

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

Ridondanza e Consistenza

- Problematiche operative collegate alla presenza di ridondanze nello schema del db:
  - maggiore occupazione (spreco) di memoria
  - le operazioni di aggiornamento, in particolare modifica e cancellazione, possono introdurre uno stato di incoerenza tra informazioni presenti nel database
  - alto costo di mantenimento della coerenza interna: le operazioni di aggiornamento risultano molto complesse e onerose (lente)
  - quando si modifica un dato duplicato o ridondante, occorre modificare tutte le sue occorrenze in modo coerente, cioè è necessario accedere a tutti i punti dove una stessa informazione si ripete, per mantenerle sincronizzate
  - occorre inoltre aggiornare eventuali ulteriori altri dati che dipendono dal dato modificato (es. totale fattura)
- Si può considerare l'introduzione di qualche elemento di ridondanza, valutando costi e rischi che tale scelta comporta, nei seguenti casi:
  - per ottenere una maggiore velocità di esecuzione di interrogazioni frequenti, e che altrimenti richiederebbero l'esecuzione di join e calcoli complessi
  - per avere una forma di controllo tra dati correlati (es. totale fattura)

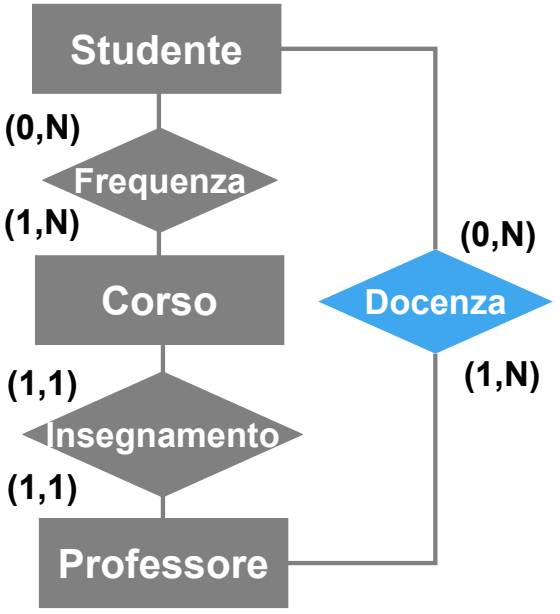
Attributo derivabile (interno)



Attributo derivabile da altra entità

Attributo derivabile da altra entità

Associazione derivabile



Progetti guida per i lavori di gruppo

- Per concretizzare questi concetti teorici, caliamoci nella progettazione di un database per una specifica realtà:
  - ciascun gruppo sceglie un problema che costituirà il suo punto di riferimento per l'applicazione dei concetti teorici in un caso concreto
  - scelto il tema di lavoro, saremo noi stessi a definire i requisiti e le specifiche del sistema che si vuole realizzare, prima in linguaggio naturale, poi in termini di costrutti E-R/UML, fino a disegnarne lo schema concettuale
- Temi :
  - Segreteria studenti università
  - Orario corsi università
  - Orario treni
  - Gestione reparto ospedaliero (ricoveri, dimissioni, ...)
  - Campionato di calcio (o quel che preferite ...)
  - Campionato di F1
  - Questionario per sondaggi demoscopici
  - Questionario per test didattici di verifica apprendimento
  - Gestione biblioteca
  - ...

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

La progettazione logica

Il modello relazionale

La progettazione logica

- La progettazione concettuale porta ad uno schema concettuale che è qualcosa di molto astratto, alquanto soggettivo, e che non tiene in alcun conto degli strumenti disponibili (DBMS) per implementare il sistema e l'applicazione
- Progettazione logica:
  - ristrutturazione dello schema concettuale, operata facendo riferimento all'architettura del DBMS che si utilizza: oggi la tipologia dominante è quella **Relazionale**
  - traduzione dei costrutti concettuali E-R nei concetti "logici" introdotti con il modello relazionale: si ragiona ancora a livello logico-concettuale e non fisico
- La progettazione logica costituisce la base per l'effettiva realizzazione operativa del database (relazionale) e delle applicazioni basate su di esso
- Il risultato della progettazione logica è dunque il ridisegno dello schema concettuale, che viene ora chiamato schema logico, attuato utilizzando concetti e formalismo introdotti dal modello relazionale
- Schema logico: rappresentazione ancora indipendente da dettagli fisici e aspetti implementativi, ma più concreta nel senso che fa riferimento alle strutture rese disponibili da un DBMS con architettura relazionale

Il modello relazionale

- Modello relazionale: formalizzazione di ispirazione matematica introdotta per rappresentare lo schema (logico) di un database relazionale
  - proposto da E. F. Codd nel 1970 per favorire l'indipendenza del disegno del db dall'implementazione fisica
  - fondato sul concetto di *relazione* matematica tra insiemi
  - disponibile in DBMS reali solo dal 1981: per la difficoltà di implementare l'indipendenza con efficienza !
- Il punto di forza di questa architettura è che ha reso le *operazioni* di accesso ai dati definitivamente indipendenti dalla *struttura* dei dati, cioè dall'implementazione fisica delle entità (-> tabelle) e soprattutto delle associazioni (-> collegamenti tra tabelle)
- Solo con l'introduzione del modello relazionale l'indipendenza tra il livello logico e il livello fisico risulta pienamente conseguita:
  - utenti e applicazioni possono fare riferimento solo al livello logico
  - le strutture descritte a livello logico sono poi implementate dal DBMS relazionale per mezzo di opportune strutture fisiche per la memorizzazione dei dati reali
  - ma per accedere ai dati non è necessario conoscere le strutture fisiche

Il modello relazionale

- Vantaggi teorici:
  - impostazione rigorosa: utilizza concetti matematici
  - ha permesso di definire in modo esatto concetti precedemente vaghi: dipendenza tra informazioni, ridondanza, vincolo
  - tutti i concetti vengono ricondotti ad una logica precisa, in modo convincente
  - anche le associazioni, concetti di natura apparentemente diversa dalle entità, risultano (vengono ricondotte a) parte integrante dei dati
  - le ridondanze vengono eliminate con un processo logico, la normalizzazione
- Vantaggi operativi:
  - fornisce una soluzione generale, semplice e lineare alle problematiche di disegno
  - le possibilità di collegamento tra diverse informazioni erano in precedenza basate su puntatori fisici che collegavano tra loro i record: nel modello relazionale tutti i collegamenti sono collegamenti "logici", istituiti tra dati reali
  - la progettazione (logica) non risulta più condizionata dalle operazioni di interrogazione che si devono garantire: qualunque query è comunque garantita e realizzabile mediante operazioni standard sui dati reali (algebra relazionale)
- Difetti:
  - minore efficienza: fino agli anni 90 l'architettura relazionale presentava ancora problemi prestazionali, e ha potuto affermarsi solo con lo sviluppo della potenza di calcolo a basso costo avvenuta degli ultimi 15 anni

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

Il modello relazionale

- Relazione: nel modello relazionale, il termine relazione è utilizzato nella sua accezione matematica (con una variante)
- In matematica, una relazione è definita come un sottoinsieme di un prodotto cartesiano di n insiemi (chiamati *domini* della relazione):

  $D_1, \ldots, D_n$ (n insiemi, anche non distinti)

  prodotto cartesiano $D_1 \times \ldots \times D_n$:
  insieme di tutte le n-uple $(d_1, \ldots, d_n)$ tali che $d_1 \in D_1, \ldots, d_n \in D_n$

  relazione matematica R su $D_1, \ldots, D_n$:
  un sottoinsieme dell'insieme $D_1 \times \ldots \times D_n$.

  $D_1, \ldots, D_n$ sono detti domini della relazione R

- In pratica, una relazione è un insieme di n-uple : $(d_1, \ldots, d_n)$

Relazione matematica

D1 = {a,b}

D2 = {x,y,z}

prodotto cartesiano D1 × D2 :

a	x
a	y
a	z
b	x
b	y
b	z

una relazione r ⊆ D1 × D2 :

a	x
a	z
b	y

Relazione matematica, proprietà

- Una relazione matematica è un insieme di n-uple :
  - $(d_1, \ldots, d_n)$ tali che $d_1 \in D_1, \ldots, d_n \in D_n$
  - la posizione degli elementi $(d_1, \ldots, d_n)$ all'interno dell' n-upla è rilevante: l' i-esimo valore proviene dall' i-esimo dominio
  - fra le n-uple non c'è invece ordinamento: in un insieme, l'ordine è irrilevante
  - le n-uple sono tutte distinte: in un insieme gli elementi sono distinti
- Nella definizione matematica, l'n-upla ha cioè una struttura posizionale: ciascuno dei domini ha ruoli diversi, distinguibili attraverso la posizione all'interno dell'n-upla
- Nella definizione di relazione nell'ambito del modello relazionale, invece, a ciascun dominio si associa un nome (attributo), che ne descrive il ruolo
- Relazione: tre accezioni
  - relazione (dall'inglese relationship) nel modello Entity-Relationship, tradotta anche con associazione (o correlazione o connessione o collegamento)
  - relazione matematica: come nella teoria degli insiemi
  - relazione secondo il modello Relazionale: insieme di n-uple costituite di attributi con riferimento per nome (non posizionale)

Relazione matematica, esempio

- Nella relazione matematica, ciascuno dei domini ha ruoli diversi, distinguibili attraverso la posizione all'interno dell'n-upla: se cambiamo l'ordine degli elementi cambia completamente il significato
- ↘ La struttura è posizionale

$$Partite \subseteq string \times string \times int \times int$$

Juve	Lazio	3	1
Lazio	Milan	2	0
Juve	Roma	0	2
Roma	Milan	0	1

Il modello relazionale

- Nell'ambito del modello relazionale, a ciascun dominio si associa un nome (attributo), che ne descrive il ruolo: la posizione dell'attributo all'interno dell'n-upla diventa quindi irrilevante
- ↘ Secondo quest'ultima definizione, una relazione ha una struttura non posizionale

Casa	Fuori	RetiCasa	RetiFuori
Juve	Lazio	3	1
Lazio	Milan	2	0
Juve	Roma	0	2
Roma	Milan	0	1

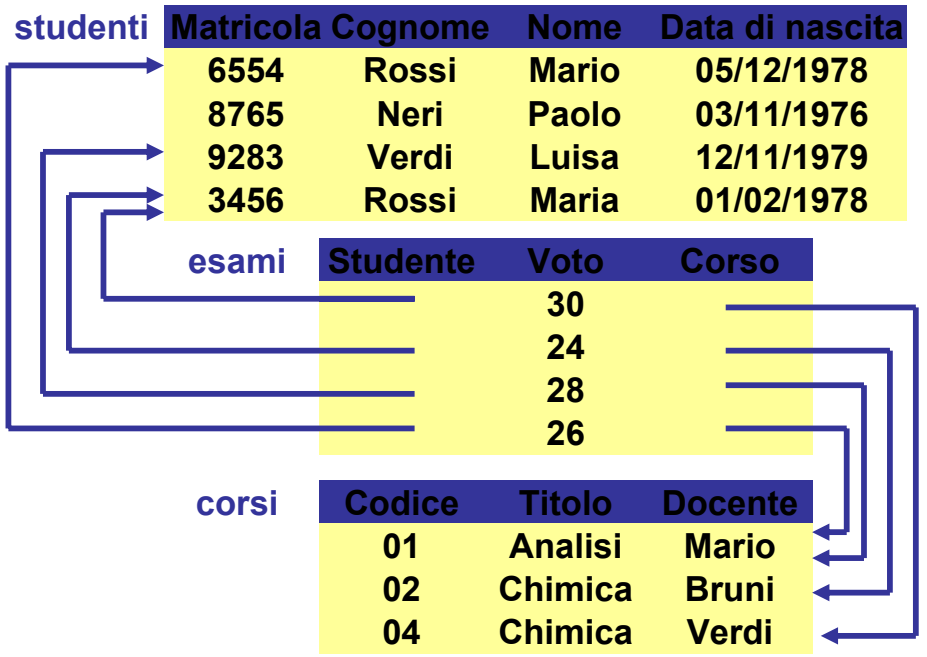
Il modello relazionale

- Il concetto di "relazione" permette di formalizzare in modo estremamente interessante il concetto informatico di "archivio"
- Una relazione ha una naturale rappresentazione per mezzo di una tabella:
  - le righe rappresentano in modo del tutto naturale le n-uple, cioè gli elementi dell'insieme (i record dell'archivio)
  - le colonne rappresentano gli attributi (i campi del record)
- Una tabella rappresenta una relazione se:
  - i valori di ogni colonna sono fra loro omogenei
  - le intestazioni delle colonne sono diverse tra loro
  - le righe sono diverse fra loro (nessun elemento dovrebbe comparire più volte)
- In una tabella che rappresenta una relazione :
  - l'ordinamento tra le righe è irrilevante
  - l'ordinamento (la posizione) tra le colonne è irrilevante
- Si ha una equivalenza sostanziale tra :
  - relazione, entità, tabella, archivio
  - n-upla di una relazione, elemento di una entità, riga di una tabella, record di un archivio
  - attributi, colonne della tabella, campi del record

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

Il modello relazionale

- Il modello relazionale è basato esclusivamente su valori:
  - nelle strutture dati non vengono introdotti puntatori fisici diretti tra record, come nei precedenti modelli (reticolare e gerarchico) basati su "record e puntatori"
  - i collegamenti tra relazioni diverse sono realizzati esclusivamente per mezzo dei valori degli attributi corrispondenti tra relazioni (entità) correlate: i collegamenti sono *logici* (per questo si dice anche che usa „puntatori logici")
- Vantaggi della struttura basata su valori
  - indipendenza del modello da strutture e dettagli di implementazione fisici
  - i puntatori sono direzionali, mentre i collegamenti logici istituiti tra informazioni corrispondenti sono bidirezionali
  - nel modello è rappresentata solo l'informazione reale utile per la descrizione del „mondo": i puntatori fisici erano dati aggiuntivi fittizi, legati ad aspetti realizzativi, e non significativi per il problema
  - i dati sono più facilmente esportabili da un sistema (DBMS) ad un altro: proprio per l'assenza di elementi estranei legati all'implementazione fisica, che sono diversi da un sistema all'altro



Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

studenti

Matricola	Cognome	Nome	Data di nascita
6554	Rossi	Mario	05/12/1978
8765	Neri	Paolo	03/11/1976
9283	Verdi	Luisa	12/11/1979
3456	Rossi	Maria	01/02/1978

esami

Studente	Voto	Corso
3456	30	04
3456	24	02
9283	28	01
6554	26	01

corsi

Codice	Titolo	Docente
01	Analisi	Mario
02	Chimica	Bruni
04	Chimica	Verdi

Progettazione logica

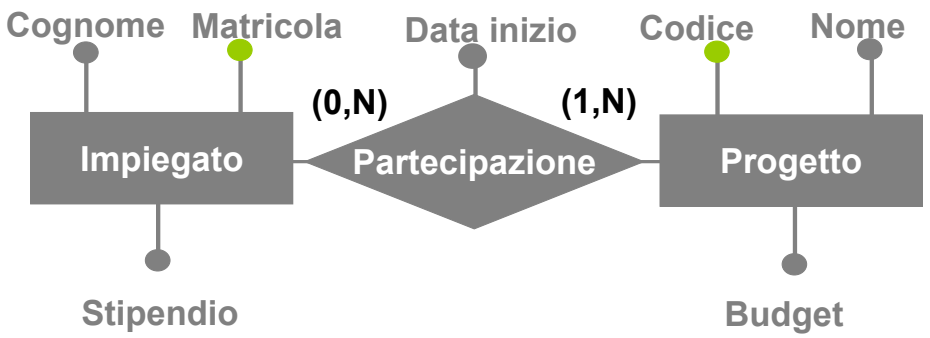
- Traduzione dello schema concettuale (E-R) nello schema "logico" relazionale:
  - Traduzione base: costrutti E-R (entità e associazioni) vengono tradotti in costrutti del modello relazionale (relazioni)
  - Raffinamento dello schema: analisi dipendenze funzionali e *Normalizzazione*
- Normalizzazione
  - procedimento formalizzato, assistito da una elegante teoria di derivazione matematica, che guida l'ottimizzazione del modello
  - permette di eliminare in modo rigoroso ed oggettivo tutte le fonti di ridondanza presenti nel modello
  - 1 FN -> 2 FN -> 3 FN (-> BC FN)

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

Traduzione da E-R a Relazionale

- Ciascuna entità diventa una relazione
- Ciascuna associazione (n:m) diventa una relazione
- Le relazioni (1:n) vengono tradotte più semplicemente, tramite inclusione della primaria chiave della relazione (1) come attributo nella relazione correlata (n), creando un collegamento logico tra le due tabelle
- Dunque nel modello relazionale, il concetto di relazione permette di unificare i concetti di entità e di associazione (relationship) del modello E-R, in un più ampio e ben definito concetto

Traduzione di una associazione molti a molti



Impiegato(<u>Matricola</u>, Cognome, Stipendio)

Progetto(<u>Codice</u>, Nome, Budget)

Partecipazione(<u>Matricola</u>, <u>Codice</u>, DataInizio)

La Normalizzazione

- **Le dipendenze funzionali**
- Intuitivamente tutti abbiamo un'idea di cosa significa che una cosa dipende da un'altra:
  - a volte questa dipendenza è determinata da una regola precisa (calcolo):
    es. Prezzo bistecca = Peso * Prezzo al kg
  - altre volte non è così ben determinata, ma c'è una relazione tra le due cose:
    es. NomeImpiegato → Stipendio
    il valore dell'attributo NomeImpiegato *determina* il valore del suo Stipendio
- In matematica si chiama funzione (o dipendenza funzionale) una relazione tra due classi X e Y indicata con:

$$X \rightarrow Y$$

Y si dice *funzione* di X, o *funzionalmente dipendente* da X

se per ogni coppia di elementi uguali di X : $x1 = x2$

gli elementi corrispondenti di Y sono anch'essi uguali: $y1 = y2$

- Ora pensiamo X e Y come due parti, cioè due sottoinsiemi di attributi, di una relazione Z : un gruppo di (uno o più) attributi può dipendere funzionalmente da un altro gruppo di (uno o più) attributi

La Normalizzazione

- Tutte le relazioni possono essere suddivise in due parti:
  - alcuni attributi (eventualmente tutti) formano la *parte unica* della relazione:
    - non vi sono ripetizioni fra le combinazioni dei loro valori
    - nessuno di questi attributi dipende dai rimanenti
  - gli altri attributi dipendono funzionalmente da quelli della parte unica
- Un insieme di attributi che forma la parte unica della relazione, e che perciò individua univocamente ogni n-upla, si dice una chiave (primaria) candidata
  - possono esistere più insiemi di attibuti che identificano univocamente l'n-upla: es. (Nome, Cognome, LuogoNascita, DataNascita, CodiceFiscale, Altezza, Peso, SegnoZodiacale)
  - ne scegliamo una, che verrà detta chiave primaria, e si indicherà nello schema logico mediante sottolineatura

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

La Normalizzazione

- La teoria della normalizzazione è centrale nel modello relazionale:
  - formalizza il concetto di ridondanza dei dati, fornendo un preciso criterio di minimalità nella rappresentazione delle informazioni
  - guida il processo per arrivare alla struttura logica migliore possibile (ottimale)
  - tutto il processo si basa sui concetti di dipendenza funzionale e di chiave primaria
- Nella teoria della normalizzazione si definiscono (almeno) 4 stati in cui una relazione può trovarsi, dette forme normali:
  1FN, 2FN, 3FN, e BCFN
- Quando una relazione non è in una forma normale, deve essere normalizzata:
  - in pratica deve essere decomposta sulla base delle dipendenze funzionali, al fine di separare i concetti, in relazioni più piccole, tra loro correlate
  - decomposizione senza perdita di informazione: dalle relazioni decomposte si deve poter ricostruire la relazione di partenza (mediante operazioni di JOIN)
- Se una relazione è in una data forma normale (es. BCNF oppure 3NF) è garantito che certi tipi di problemi sono evitati o minimizzati: sapere in che forma normale è una relazione ci aiuta a decidere se decomporla ulteriormente sia necessario o utile

La Normalizzazione

- La prima forma normale: 1FN
- Una relazione è in 1FN se ogni attributo contiene solo valori atomici, e non insiemi o liste di valori
  Es: SquadraCalcio (Nome, Sede, Presidente, Allenatore, ElencoGiocatori)
- La 1FN non ammette gli attributi complessi, che concettualmente rappresentano concetti indipendenti
  -> la relazione deve essere decomposta:
  - se l'attributo in questione non fa parte della chiave primaria, può essere eliminato dalla relazione di partenza e costituire una nuova relazione
  - nella nuova relazione viene riportata anche la chiave primaria della relazione di partenza:

    SquadraCalcio (Nome, Sede, Presidente, Allenatore)

    Giocatore (Tessera, Nome, Cognome, DataNascita, Ruolo, NomeSquadra)
  - l'inseririmento della chiave primaria NomeSquadra nella relazione Giocatore crea un collegamento logico tra le due entità, cioè istituisce l'associazione (1:n) che la dipendenza funzionale esprimeva
- Tutte le relazioni devono essere almeno normalizzate in 1FN:
  con il passaggio a 1NF si introducono associazioni (1:n) tra le relazioni

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

La Normalizzazione

- La seconda forma normale: 2FN
- Una relazione è in 2FN se nessun attributo dipende funzionalmente da solo alcuni degli attributi che costituiscono la chiave primaria
- Ciascun attributo deve cioè dipendere dall'intera chiave primaria, da tutti gli attributi della chiave in blocco
- La 2FN non ammette le dipendenze parziali dalla chiave primaria
  -> la relazione deve essere decomposta:

  Persona(Nome, LuogoNascita, AnnoN, MeseN, GiornoN, Altezza, Peso, SegnoZodiacale)
  - SegnoZodiacale non dipende funzionalmente da tutti gli attributi della chiave, ma solo da MeseN e GiornoN
  - in effetti questo determina una ridondanza: lo stesso segno zodiacale sarà ripetuto ogni volta che due persone sono nate lo stesso giorno e mese
  - l'attributo SegnoZodiacale può essere eliminato dalla relazione, e costituire una nuova relazione

    Persona (Nome, LuogoNascita, AnnoN, MeseN, GiornoN, Altezza, Peso)

    SegnoZodiacale (Mese, Giorno, Segno)
  - anche in questo caso risulta istituita una associazione (1:n) tra le due relazioni

La Normalizzazione

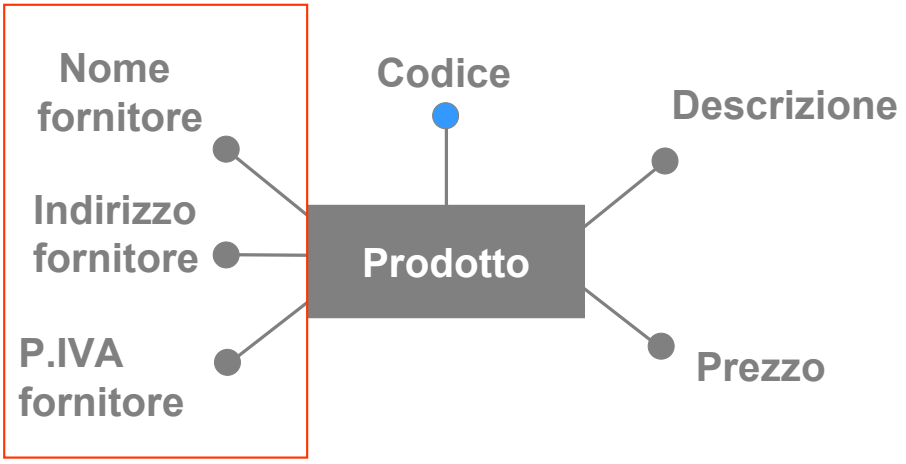
- La terza forma normale: 3FN
- Una relazione è in 3FN se nessun attributo dipende funzionalmente da altri attributi della relazione (che non siano la chiave primaria):
  - una relazione che non si trova in 3FN contiene una ridondanza analoga alla precedente, l'unica differenza è che la dipendenza parziale interna alla relazione non coinvolge la chiave, ma altri attributi
  - considerato che tali attributi a loro volta dipendono evidentemente dalla chiave, questo tipo di dipendenza prende il nome di *dipendenza transitiva*
- La 3FN non ammette le dipendenze transitive
  -> la relazione deve essere decomposta

  Persona(CF, Nome, LuogoN, AnnoN, MeseN, GiornoN, Altezza, Peso, SegnoZodiacale)
  - SegnoZodiacale non dipende funzionalmente dalla chiave CF, ma solo da MeseN e GiornoN: ma la ridondanza presente nella relazione è intuitivamente la stessa del caso precedente
  - in modo del tutto analogo al precedente, l'attributo SegnoZodiacale può essere eliminato dalla relazione, e costituire una nuova relazione:

    Persona (CF ,Nome, LuogoNascita, AnnoN, MeseN, GiornoN, Altezza, Peso)

    SegnoZodiacale (Mese, Giorno, Segno)
  - anche in questo caso risulta istituita una associazione (1:n) tra le due relazioni

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06



PartitaIVA → NomeFornitore, Indirizzo

Prodotto(Codice, Descrizione, Prezzo, PIVAFornitore)
Fornitore(PIVA, Nome, Indirizzo)

La Normalizzazione

- La forma normale di Boyce-Codd: BCFN
- Una relazione R è in forma normale di Boyce-Codd se, per ogni dipendenza funzionale (non banale) $X \rightarrow Y$ definita su di essa, X contiene una chiave K di R
  - in pratica: ciascun attributo deve dipendere dalla chiave, tutta la chiave, e nient'altro che la chiave
  - la forma normale BCFN richiede che i concetti in una relazione siano omogenei, nel senso che devono esserci solo attributi direttamente associati alla chiave
  - se una relazione rappresenta più concetti indipendenti, va decomposta in relazioni più piccole, una per ogni concetto
- Questa forma normale è la più restrittiva, e comprende tutte le precedenti, ma non è sempre ottenibile senza perdita di informazione
- Per i nostri scopi possiamo fermarci qui ...

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

La Normalizzazione

- Le associazioni (n:m) nel modello relazionale
- Esempio. Supponiamo di aver definito come punto di partenza della nostra analisi ed astrazione della realtà di nostro interesse la relazione Insegna:

  Insegna = (IDProf, NomeProf, IndirizzoProf, Classe, NomeScuola, IndirizzoScuola, Preside, NumeroOre)
- E' in 1FN ? Tutti gli attributi sono atomici quindi 1FN ok.
- E' in 2FN ? Analizzando le dipendenze funzionali scopriamo che:

  IDProf -> NomeProf, IndirizzoProf

  NomeScuola -> IndirizzoScuola, Preside

  E' necessario un (doppio) passaggio a 2FN:

  Insegnante = (IDProf, NomeProf, IndirizzoProf)

  Scuola = (NomeScuola, IndirizzoScuola, Preside)

  Insegna = (IDProf, Classe, NomeScuola, NumeroOre)
- La relazione risultante Insegna istituisce un collegamento tra le relazioni Insegnante e Scuola: l'associazione (n:m) tra Insegnante e Scuola risulta identificata e risolta in modo naturale, come conseguenza logica della normalizzazione

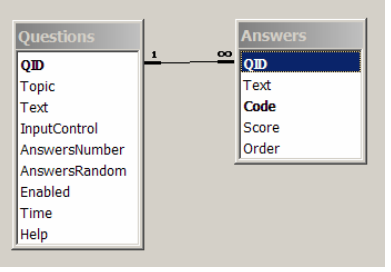
La Normalizzazione

- Il processo di normalizzazione permette di raggiungere con un procedimento logico e non soggettivo alcuni importanti risultati:
  - eliminazione delle ridondanze: è ora chiaro che le ridondanze sorgono quando esistono dipendenze funzionali non banali interne alle relazioni
  - creazione di associazioni (1:n): risultano implicitamente dai passaggi in 1-2-3FN, e sono realizzate riportando le chiavi primarie nelle relazioni correlate, come puntatori logici
  - le associazioni (n:m) vengono risolte in relazioni, connesse logicamente da relazioni (1:n) ad altre relazioni, in modo naturale e automatico, come conseguenza del passaggio in 2-3FN
- Per eliminare ulteriori ridondanze:
  - qualsiasi relazione esprimibile con una formula può essere sostituita con una procedura di calcolo (formula, vincolo), invece che con una relazione (es. ritorniamo al nostro esempio del segno zodiacale)

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

Integrità referenziale

- L'integrità referenziale è un vincolo che si impone tra due relazioni, collegate tra loro da un insieme di attributi X, per impedire l'inserimento di informazioni non consistenti
- L'integrità referenziale si applica ad una associazione (1:n) e prevede che i valori degli attributi X di ciascuna n-upla di R2 devono essere rappresentati nella relazione R1, dove costituiscono la chiave primaria
- In pratica, supponiamo di avere una relazione (1:n) tra le tabelle seguenti:
- la relazione R1 sul lato 1 della associazione ha come chiave primaria l'attributo QID
- la tabella R2 sul lato n non può contenere record (n-uple) non correlati, cioè con un QID non presente nella tabella R1
- E' possibile rinforzare ulteriormente l'integrità referenziale, impostando delle regole di sincronizzazione, che vengono attivate dalla modifica di un dato in R1:
  - aggiornamento a cascata dei record correlati in R2 su modifica di un QID in R1
  - cancellazione a cascata dei record correlati in R2 su cancellazione di un record in R1



JOIN e UNION

- Il JOIN è una operazione fondamentale in un database relazionale per ricostruire l'informazione complessiva a partire dalle relazioni normalizzate
- Le informazioni sono fisicamente organizzate in diverse tabelle, logicamente correlate: per rispondere a determinate domande è necessario mettere insieme le informazioni di più tabelle
- Ci sono molti possibili modi diversi di unire i dati di due tabelle…
- UNION:
  - nuova relazione costituita dall'insieme delle n-uple delle due relazioni che vengono unite: è una unione tra insiemi (A U B)
  - le tabelle vengono unite nel senso delle colonne: l'operazione ha senso per tabelle con la stessa struttura, cioè che contengono gli stessi campi.
- JOIN:
  - fusione tra le n-uple di due relazioni logicamente correlate (es. 1:1 o 1:n)
  - le tabelle vengono unite nel senso delle righe: la riga risultante è costituita dai campi della prima tabella + quelli della seconda tabella
  - Come si mettono in corrispondenza (*matching*) le righe delle due tabelle ?

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

JOIN

- Join interno (inner join):
  - il matching avviene sulla base dell'uguaglianza tra i valori di uno o più campi delle due tabelle: vengono appaiate e unite le righe che soddisfano tale condizione
  - solo i record presenti in entrambe le tabelle vengono inclusi nel risultato
  - Join "naturale": correla i dati di due relazioni sulla base dei valori assunti da tutti gli attributi con lo stesso nome
- Join esterno (outer join):
  - Left join: tutte le righe della tabella a sx vengono comunque incluse nel risultato, anche se non hanno righe correlate nella tabella a dx
  - Right join: tutte le righe della tabella a dx vengono comunque incluse
  - Outer join: esterno su entrambi i lati
- Prodotto cartesiano:
  - prodotto tra insiemi (A x B), contiene le n-uple ottenute combinando in tutti i modi possibili le n-uple dei due insiemi: operazione molto onerosa computazionalmente, genera un insieme di (n x m) elementi
  - Theta join: prodotto cartesiano seguito da una selezione, che conserva solo le n-uple di interesse descritte da una condizione booleana (qualsiasi)
  - Equi-join: caso particolare del precedente, in cui la selezione avviene su una condizione di uguaglianza tra un attributo di una tabella e uno dell'altra

La progettazione fisica

Implementazione in Access

Progettazione Fisica

- Implementazione fisica della base di dati utilizzando un DBMS reale:
  - specificazione dei parametri fisici di memorizzazione dei dati: tipo e dimensione di ciascun campo (attributo)
  - definizione di strutture accessorie per rendere più efficiente l'accesso a particolari informazioni: gli indici, detti anche chiavi secondarie (la chiave primaria è usualmente indicizzata automaticamente dal sistema)
- I tipi di dati disponibili dipendono dal DBMS, vediamo quelli supportati dal motore Jet (e quindi da Access):
  - Byte (1 byte), Integer (2 bytes), Long (4 bytes)
  - Single precision (4 bytes), Double precision (8 bytes), Decimal (12 bytes=28 cifre decimali)
  - Currency (strano tipo numerico decimale per le valute [$,£,€]: 15.4)
  - Text (1 – 255 caratteri), Memo (illimitato, non indicizzabile)
  - Datetime (data e ora)
  - Si/No (Boolean ovvero Vero/Falso)

Gli Indici

- Gli indici sono strutture dati ausiliarie che permettono di rendere più efficiente l'acceso ai record (n-uple) di una tabella (relazione) in base ad un attributo (o una combinazione di attributi)
- Un indice è una struttura dati che consente l'accesso diretto ai record di una tabella in base ai valori di uno o più attributi:
  - l'indice velocizza le operazioni di selezione, ricerca, join e ordinamento
  - per ciascuna tabella si possono definire più indici, semplici o composti
  - per ciascun indice si specificano il campo (o i campi) che ne fanno parte
- Un indice è costituito da:
  - un dizionario per l'attributo (o gli attributi) dell'indice
  - una lista invertita costituita, per ogni voce del dizionario, da puntatori diretti (fisici) a tutti i record individuati nella tabella
  - si tratta in sostanza di un indice analitico, analogo a quello di un libro che ci fornisce l'elenco delle pagine dove compare ciascun termine
  - il dizionario viene mantenuto ordinato, per mezzo di algoritmi molto sofisticati, basati su hash o su alberi binari, che permettono di eseguire ricerche molto più veloci di quelle sequenziali

Prof. Claudio Capiluppi - Facoltà di Scienze della Formazione - A.A. 2005/06

Gli Indici

- Gli indici velocizzano drasticamente le operazioni di accesso, ma rallentano (un po') quelle di inserimento e modifica
- E' consigliabile impostare un indice per gli attributi che :
  - sono chiave primaria di una relazione
  - sono oggetto di frequenti ricerche e selezioni (SELECT)
  - costituiscono collegamenti logici tra tabelle, e quindi costituiscono le informazioni chiave nelle operazioni di JOIN di tabelle correlate
  - sono oggetto di ordinamento: per avere un ordinamento delle n-uple, non essendo definito un ordinamento fisico tra record nelle tabelle
  - individuano categorie utilizzate in operazioni di aggregazione (GROUP BY)
- In assenza di opportuni indici, tutte le operazioni sulle relazioni possono avvenire solo mediante scansione sequenziale di tutti i record, con tempi proporzionali alla dimensione dell'archivio
- Il motore Jet supporta i seguenti tipi di indice :
  - univoco: non ammette valori duplicati (vincolo che previene l'inserimento di valori duplicati negli attributi indicizzati)
  - ignora null: non ammette valori null nei campi dell'indice, cioè i record contenenti valori null nei campi dell'indice non vengono inclusi nell'indice
  - primario: univoco e non ammette valori null