



Dispensa per la 3 RIM

Istituto Tecnico Statale Leonardo da Vinci – Borgomanero

A/S 2018/19

DOCENTE: PROF. MAURIZIO MASETTA

Argomenti

Basi di dati	
DBMS (DataBase Management System)	
Modello dei dati	
Modello relazionale	
Vincoli di integrità	
Progetto della base di dati	
Il modello Entità-Relazione	
Le gerarchie nel modello Entità-Relazione	
Sviluppo schema E-R	

Basi di dati (database)

Ogni organizzazione ha un **sistema organizzativo**, e più precisamente un insieme di risorse e regole per lo svolgimento coordinato delle attività al fine del perseguimento degli scopi. Dove per "risorse" si intendono personale, denaro, materiale, informazioni....ecc. La parte del sistema organizzativo che gestisce (acquisisce, elabora, conserva, produce) le informazioni di interesse, ovvero quelle informazioni utilizzate per il perseguimento degli scopi dell'organizzazione stessa, è detta **sistema informativo**. Le funzioni di un sistema informativo sono:

- raccolta, acquisizione delle informazioni
- archiviazione, conservazione delle informazioni
- elaborazione delle informazioni
- distribuzione, scambio di informazioni

Possiamo definire **sistema informatico** la porzione automatizzata del sistema informativo, e quindi la parte del sistema informativo che gestisce informazioni per mezzo della tecnologia informatica. Nelle attività umane, le informazioni vengono gestite in forme diverse, a seconda delle necessità, e trasmesse attraverso vari supporti, dalla memoria umana alla carta.... Nelle attività standardizzate dei sistemi informativi complessi, sono state introdotte col tempo forme di organizzazione e codifica delle informazioni. Ad esempio, nei servizi anagrafici si è iniziato con registrazioni discorsive e sono state poi introdotte informazioni via via più precise (e in un certo senso artificiali):

- nome e cognome;
- estremi anagrafici;
- codice fiscale;

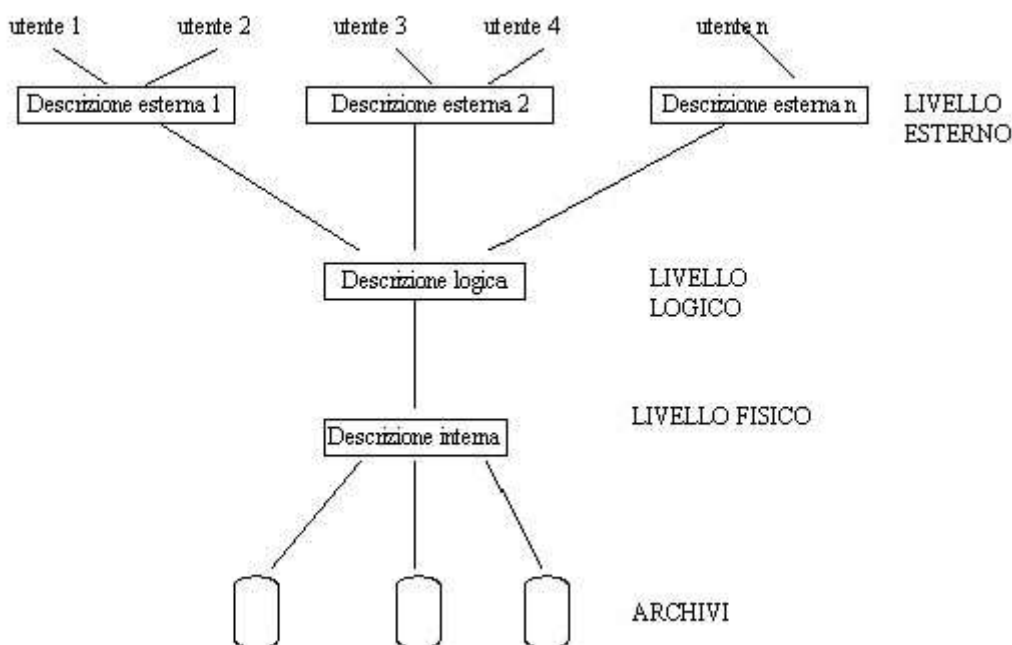
Nei sistemi informatici (e non solo in essi), le informazioni vengono quindi rappresentate attraverso dati. I dati sono fatti elementari, informazioni codificate, che hanno bisogno di essere interpretate per fornire conoscenza. Per esempio 'Mario' e '275' su un foglio di carta sono due dati. Se il foglio 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 ed arricchire la conoscenza. Una collezione di dati, utilizzati per rappresentare le informazioni di interesse per una o più applicazioni di una organizzazione è detta **base di dati** (database). I sistemi per la gestione di basi di dati sono detti **DBMS** (DataBase Management System). In ogni base di dati esistono:

- lo **schema**, sostanzialmente invariante nel tempo, che ne descrive la struttura e quindi per esempio, le intestazioni di una tabella
- l'**istanza**, costituita dai valori attuali, che possono cambiare molto e molto rapidamente, per esempio, il "corpo" di una tabella.

In un database, importante è il concetto di modello dei dati. Esso è un insieme di costrutti utilizzati per organizzare i dati di interesse e descriverne la dinamica. Il modello attualmente più utilizzato è il MODELLO RELAZIONALE, basato sul concetto di insieme e sulla strutturazione dei dati

tramite tabelle. La nozione di modello e di schema descritta sopra può essere ulteriormente sviluppata tenendo presente altre “dimensioni” nella descrizione dei dati. In particolare esiste una proposta di architettura standardizzata per DBMS articolata su tre livelli:

- **LIVELLO FISICO O INTERNO:** descrive la base di dati come un insieme di elementi contenuti nella memoria di massa, e quindi per mezzo di strutture fisiche di memorizzazione (implementazione della base di dati);
- **LIVELLO LOGICO:** evidenzia l’organizzazione dei dati dal punto di vista del loro contenuto informativo, descrivendo la struttura di ciascun dato e i collegamenti tra dati diversi, in base al modello logico utilizzato;
- **LIVELLO ESTERNO:** costituisce la descrizione di una porzione della base di dati. Uno schema esterno può prevedere organizzazioni dei dati diverse rispetto a quelle utilizzate nel livello logico. Uno schema esterno riflette il punto di vista di un particolare utente o insiemi di utenti. Pertanto è possibile associare ad uno schema logico vari schemi esterni. Per esempio se nel database dell’ università statale di Milano, c’è l’ elenco di tutti i corsi, uno studente di Ingegneria Elettronica potrebbe essere interessato solo ai corsi offerti dal manifesto del suo corso di laurea. Si può allora ideare uno schema esterno che consenta di accedere solo ai corsi di interesse.



Una volta archiviati i dati, si deve accedere ad essi. L’accesso ai dati può avvenire:

- con linguaggi testuali interattivi
- con comandi (come quelli del linguaggio interattivo) immersi in un linguaggio ospite (Pascal, C, Cobol, etc.)

- con comandi (come quelli del linguaggio interattivo) immersi in un linguaggio ad hoc, con anche altre funzionalità (p.es. per grafici o stampe strutturate)
- con interfacce amichevoli (senza linguaggio testuale).

Le operazioni fondamentali sulle basi di dati, si possono raggruppare su tre categorie:

1. **Amministrazione** (DB Administration). Progettazione, creazione e distruzione delle strutture dati o dell'intera base dati.
2. **Transazione** (Transaction). Inserimento, aggiornamento, cancellazione dei singoli dati
3. **Interrogazione** (Query). Consultazione (lettura) dei dati con appositi linguaggi di interrogazione (SQL)

È evidente che questi tre livelli presentano importanti differenze anche a livello di sicurezza della base-dati; il primo livello (spesso abbreviato in DBA) è il più delicato e andrà riservato a pochissime persone competenti. Il secondo è anche molto delicato, perchè va a modificare la base-dati; solo il personale autorizzato potrà accedere a questo livello. L'ultimo livello, quello delle interrogazioni, può essere esteso all'intera utenza non comportando pericoli per l'integrità della base-dati, è però verosimile che in molti casi sia necessario distinguere ulteriori livelli di sicurezza; alcune informazioni possono essere riservate o segrete (si pensi all'estratto conto di una banca).

DBMS

I DBMS (DataBase Management System) sono sistemi per la gestione di basi di dati. Essi sono prodotti software in grado di gestire collezioni di Dati (database). Un DBMS deve tenere conto di alcune caratteristiche delle basi di dati. Le basi di dati possono infatti essere **grandi** nel senso che possono avere anche dimensioni enormi e comunque in generale dimensioni molto maggiori della memoria centrale disponibile. Ovviamente possono esistere anche basi di dati “piccole” ma i sistemi debbono poter gestire i dati senza porre limiti alle dimensioni, a parte quelle fisiche dei dispositivi. Le basi di dati sono **condivise**, nel senso che applicazioni e utenti diversi debbono poter accedere, secondo opportune modalità, a dati comuni. E' importante notare che in questo modo si riduce la ridondanza dei dati poiché si evitano ripetizioni. Le basi di dati sono **persistenti** cioè hanno un tempo di vita che non è limitato a quello delle singole esecuzioni dei programmi che le utilizzano. I DBMS devono poi garantire **affidabilità** cioè la capacità del sistema di conservare sostanzialmente intatto il contenuto della base di dati in caso di malfunzionamento hardware e software. I DBMS devono garantire anche la **privatezza** dei dati. Ciascun utente viene abilitato a svolgere solo determinate azioni sui dati attraverso meccanismi di autorizzazione. Inoltre un DBMS deve essere efficiente ed efficace. Per **efficienza** si intende la capacità di svolgere le operazioni utilizzando un insieme di risorse (tempo e spazio) che sia accettabile per gli utenti. Per **efficacia** si intende la capacità della base di dati di rendere produttive in ogni senso, le attività dei suoi utenti. Alcuni tra i DBMS più utilizzati sono:

- Oracle - SQL server Microsoft - DB 2 (IBM) - Paradox (Borland) - ACCESS (Microsoft)

Modello dei dati

Un modello dei dati è un insieme di concetti utilizzati per organizzare una base di dati e descriverne la struttura in modo che essa risulti comprensibile ad un elaboratore. Ogni modello dei dati fornisce meccanismi di strutturazione, analoghi ai costruttori di tipo dei linguaggi di programmazione, che permettono di definire nuovi tipi sulla base di tipi elementari. Ogni modello dei dati prevede alcuni costruttori, ad esempio, il modello relazionale prevede il costruttore relazione, che permette di definire insiemi di record omogenei. Nell' esempio vediamo una rappresentazione di relazioni mediante una tabella.

CORSI	Corso	Docente	Aula
	Basi di dati	Rossi	DS3
	Sistemi	Neri	N3
	Reti	Bruni	N3
	Controlli	Bruni	G

Modellare i dati significa costruire una rappresentazione semplificata della realtà di un problema. Si può distinguere tra due tipologie principali di modello, quello logico e quello concettuale. Il **modello logico** è utilizzato nei DBMS per l'organizzazione dei dati in modo indipendente dalle strutture fisiche. Le strutture utilizzate da questi tipi di modello, pur essendo astratte, riflettono una particolare organizzazione (ad alberi, a grafi, a tabelle o a oggetti). Il modello logico si preoccupa di descrivere la composizione ed il formato dei dati. I principali modelli logici sono:

- **MODELLO GERARCHICO:**basato sull'uso di strutture ad albero (e quindi gerarchie,da cui il nome), definito durante la prima fase di sviluppo dei DBMS (anni Sessanta),ma tuttora ampiamente utilizzato; *(Non viene trattato in questa dispensa)*
- **MODELLO RETICOLARE:** e' sorto come estensione del linguaggio di programmazione Cobol per la gestione di strutture dati complesse. Il modello reticolare e' basato sui grafi, ovvero sulle strutture dati a reticolo. *(Non viene trattato in questa dispensa)*
- **MODELLO RELAZIONALE:** e' basato sul concetto di insieme e sulla strutturazione dei dati tramite tabelle.
- **MODELLO A OGGETTI:** sviluppato a partire dal 1985, caratterizza i DBMS di nuova concezione. Esso estende alla base di dati alcune caratteristiche dei linguaggi di programmazione orientati ad oggetti. *(Non viene trattato in questa dispensa)*

I modelli più utilizzati al momento risultano essere i database relazionali, perché caratterizzati da una definizione estremamente compatta ed elegante. Il **modello concettuale** permette invece di rappresentare i dati in modo indipendente da ogni sistema, cercando di descrivere i concetti del mondo reale. Quest' ultimo modello è infatti utilizzato nelle fasi preliminari di progettazione di basi di dati, il più noto è il modello Entity-Relationship.

Modello relazionale

Un modello dei dati è un insieme di concetti utilizzati per organizzare una base di dati e descriverne la struttura in modo che essa risulti comprensibile ad un elaboratore. Il modello relazionale è il modello logico dei dati, oggi più diffuso, soprattutto in ambiente di personal computer e di reti. Nato all'inizio degli anni '70 si basa sul concetto matematico di relazione tra insiemi. La teoria dei DB relazionali può esprimersi in modo rigoroso usando il linguaggio della logica formale o in alternativa quello dell' insiemistica. Il modello relazionale usa come struttura dati fondamentale la relazione (o tabella), e per operare sul DB definisce poche operazioni fondamentali, che si dimostrano sufficienti per qualsiasi operazione di interrogazione o transazione sul DB. Nel modello relazionale a ciascun dominio associamo un nome (**attributo**), unico nella relazione, che “descrive” il ruolo del dominio. Nella rappresentazione tabellare, gli attributi possono essere usati come intestazioni delle colonne.

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

Le informazioni sono rappresentate per mezzo di **ennuple** (La partita Juve-Lazio è finita 3-1). Ogni ennupla indica quindi che c'è una relazione tra i vari elementi della ennupla stessa, e ogni elemento viene estrapolato da un solo dominio. Si noti che l'ordinamento fra gli attributi è irrilevante, infatti la struttura è non posizionale. Una tabella rappresenta una relazione se:

- i valori di ciascuna colonna sono fra loro omogenei (dallo stesso dominio);
- le righe sono diverse fra loro;
- le intestazioni delle colonne sono diverse tra loro.
- l'ordinamento tra le righe è irrilevante;
- l'ordinamento tra le colonne è irrilevante.

Il modello relazionale impone ai dati una struttura rigida, infatti solo alcuni formati di ennuple sono ammessi. Per esempio una tabella di questo tipo:

Città	Prefettura
Roma	Via IV novembre
Firenze	
Tivoli	
Prato	

ci darebbe delle informazioni incomplete. Nel modello relazionale si adotta una tecnica rudimentale ma efficace, viene riconosciuto il concetto di **valore nullo** che denota l'assenza di un valore del dominio (e non è un valore del dominio). Si possono (e debbono) però imporre restrizioni sulla presenza di valori nulli.

Studenti	Matricola	Cognome	Nome	Nascita
	276545	Rossi	Maria	NULL
	NULL	Neri	Anna	23/04/1972
	NULL	Verdi	Fabio	12/02/1972

Per esempio nella tabella sopra, non ha senso mettere un valore nullo alla voce Matricola, in quanto è il campo che serve per identificare univocamente ogni studente. In genere quando si definisce una relazione si può specificare che i nulli sono ammessi solo su alcuni particolari attributi e non su altri. In molti casi però, nonostante siano stati collocati correttamente i valori nulli, non vero che qualsiasi tupla rappresenti informazioni corrette. Esistono infatti istanze di basi di dati che, pur sintatticamente corrette, non rappresentano informazioni possibili per l'applicazione di interesse. Per esempio nella seguente tabella abbiamo un voto pari a 32, che nel sistema italiano di valutazione non è ammissibile.

Esami	Studente	Voto	Lode	Corso
	276545	28	e lode	01
	276545	32		02
	788854	23		03
	200768	30	e lode	03

Sempre nella relazione ESAMI viene indicato che è stata attribuita la lode in un esame in cui il voto è 28, il che è impossibile. La terza tupla della relazione ESAMI, presenta per l'attributo Studente, un valore che non compare tra i numeri di matricola della relazione STUDENTI descritta di seguito:

Studenti	Matricola	Cognome	Nome	Nascita
	276545	Rossi	Maria	23/04/1968
	276545	Neri	Anna	23/04/1972
	788854	Verdi	Fabio	12/02/1972

In una base di dati è opportuno evitare situazioni come quelle appena descritte. Allo scopo è stato introdotto il concetto di vincolo di integrità. In generale ad uno schema di base di dati associamo un insieme di vincoli, e consideriamo corrette solo quelle istanze che soddisfano tutti i vincoli. Progettare un data-base relazionale richiede le seguenti fasi:

- Tracciare il diagramma entità-associazioni (DEA) del DB.
- Tradurre il DEA in un elenco di schemi relazionali.
- Individuare gli eventuali vincoli di integrità.
- Scegliere l'ambiente di sviluppo nel quale sarà realizzato il DB, e tradurre nel suo linguaggio gli schemi relazionali e le varie procedure.

Vincoli di integrità

Esistono istanze di basi di dati nel modello relazionale che, pur sintatticamente corrette, non rappresentano informazioni possibili per l'applicazione di interesse. Allo scopo di evitare che questo accada è stato introdotto il concetto di **vincolo di integrità**. Ogni vincolo può essere visto come un predicato che associa ad ogni istanza il valore vero o falso. Se il predicato assume valore vero diciamo che l'istanza soddisfa il vincolo. In generale ad uno schema di base di dati associamo un insieme di vincoli, e consideriamo corrette solo quelle istanze che soddisfano tutti i vincoli. Possiamo distinguere due categorie di vincoli. **Vincolo intrarelazionale**: se il suo soddisfacimento è definito rispetto a singole relazioni della base di dati. Può essere quindi un vincolo su una tupla, ovvero un vincolo che può essere valutato su ciascuna tupla indipendentemente dalle altre, o un vincolo su valori, in quanto impone una restrizione sul dominio dell'attributo.

Esami	Studente	Voto	Lode	Corso
	276545	28	e lode	01
	276545	32		02
	788854	23		03
	200768	30	e lode	03

Studenti	Matricola	Cognome	Nome	Nascita
	276545	Rossi	Maria	23/04/1968
	276545	Neri	Anna	23/04/1972
	788854	Verdi	Fabio	12/02/1972

Per esempio un **vincolo su una tupla** è $(Voto = 30) \text{ OR NOT } (Lode = \text{"e lode"})$ che sta ad indicare che nella relazione ESAMI, non posso avere una tupla in cui mi compare "e lode" associato a un voto diverso da 30. Mentre un **vincolo su valori** può essere $(Voto \geq 18) \text{ AND } (Voto \leq 30)$ in quanto impone una restrizione sul dominio dell'attributo voto.

Vincolo interrelazionale se coinvolge più relazioni. La terza tupla della relazione ESAMI, presenta per l'attributo Studente, un valore che non compare tra i numeri di matricola della relazione STUDENTI. Questa situazione indesiderata può essere vietata richiedendo che un numero di matricola compaia nella relazione ESAMI solo se compare nella relazione STUDENTI. I vincoli interrelazionali più importanti sono:

- Vincoli di chiave
- Vincoli di integrità referenziale

Vincoli di chiave, nel modello relazionale

Cominciamo con un esempio:

Matricola	Cognome	Nome	Corso	Nascita
6554	Rossi	Mario	Informatica	5/12/1978
8765	Rossi	Mario	Informatica	3/11/1976
4723	Verdi	Laura	Meccanica	10/7/1979
9283	Verdi	Mario	Informatica	3/11/1976
3456	Rossi	Laura	Meccanica	5/12/1978

Nella relazione sopra, possiamo notare che i valori delle varie tuple sull' attributo Matricola sono tutti diversi l'uno dall' altro. Infatti il valore della matricola identifica univocamente gli studenti. Analogamente possiamo notare che non ci sono coppie di tuple con gli stessi valori su ciascuno dei tre attributi Cognome, Nome, Nascita, infatti anche i dati anagrafici identificano univocamente le persone (in questo caso è vero). Intuitivamente una chiave è un insieme di attributi che identifica univocamente le tuple di una relazione. Per formalizzare la definizione, procediamo in due passi:

- un insieme K di attributi è **superchiave** per una relazione r se r non contiene due ennuple distinte t_1 e t_2 con $t_1[K] = t_2[K]$
- K è **chiave** per r se è una superchiave minimale, cioè non contiene un'altra superchiave per r

Nell' esempio sopra Matricola è una chiave in quanto:

- Matricola è superchiave in quanto non c'è nessuna riga per la quale abbiamo lo stesso valore.
- contiene un solo attributo e quindi è minimale

Cognome, Nome, Nascita è un'altra chiave:

- l'insieme Cognome, Nome, Nascita è superchiave in quanto non c'è nessuna riga per la quale abbiamo gli stessi valori per questi attributi.
- nessuno dei suoi sottoinsiemi che otteniamo eliminando un attributo è superchiave, in quanto ci sono righe con gli stessi valori

Si noti che l'insieme di tutti gli attributi è sempre una superchiave per ogni relazione. Il fatto che su ciascun schema di relazione possa essere definita almeno una chiave, garantisce la accessibilità a tutti i valori di una base di dati, e la loro univoca identificazione. Le chiavi sono lo strumento principale attraverso il quale vengono correlati i dati in relazioni (tabelle) diverse. In presenza di

valori nulli però, i valori degli attributi che formano la chiave non permettono di identificare le ennuple come desiderato e né permettono di risalire facilmente ad altre relazioni.

Matricola	Cognome	Nome	Nascita	Corso
NULL	Rossi	Luca	NULL	Informatica
8765	Rossi	Mario	01/05/61	Civile
4856	Neri	Mario	NULL	NULL
NULL	Neri	Mario	05/03/63	Civile

La presenza di valori nulli nelle chiavi deve quindi essere limitata. La soluzione pratica è che per ogni relazione scegliamo una chiave (la **chiave primaria**) su cui non ammettiamo valori nulli.

<u>Matricola</u>	Cognome	Nome	Nascita	Corso
6554	Rossi	Luca	NULL	Informatica
8765	Rossi	Mario	01/05/61	Civile
4856	Neri	Mario	NULL	NULL
6590	Neri	Mario	05/03/63	Civile

Per convenzione gli attributi che compongono la chiave primaria sono sottolineati.

Vincoli di integrità referenziale

Consideriamo la seguente base di dati nel modello relazionale:

infrazioni	<u>Codice</u>	Data	Vigile	Prov	Numero
	65524	3/9/1997	343	MI	3K9886
	87635	4/12/1997	476	MI	6D5563
	82236	4/12/1997	343	RM	7C5567
	35632	6/1/1998	476	RM	7C5567
	76543	5/3/1998	548	MI	6D5563

vigili	<u>Matricola</u>	Cognome	Nome
	343	Rossi	Luca
	476	Neri	Pino
	548	Nicolosi	Gino

automobili	<u>Prov</u>	<u>Numero</u>	Proprietario	...
	MI	3K9886	Nestore	...
	MI	6D5563	Nestore	...
	RM	7C5567	Menconi	...
	RM	1A6673	Mussone	...
	MI	5E7653	Marchi	...

Le informazioni della relazione INFRAZIONI sono rese significative e complete attraverso il riferimento alle altre due relazioni. I riferimenti sono significativi in quanto i valori della relazione

INFRAZIONI sono uguali ai valori effettivamente presenti nelle alte due. Informazioni in relazioni diverse sono quindi correlate attraverso valori comuni, in particolare, valori delle chiavi primarie solitamente. Un vincolo di **integrità referenziale** fra un insieme di attributi X (per esempio Prov e Numero) di una relazione R1 (INFRAZIONI) e un'altra relazione R2 (AUTOMOBILI), impone ai valori su X di ciascuna ennupla dell'istanza di R1 di comparire come valori della chiave (primaria) dell'istanza di R2. Nell'esempio, esistono vincoli di integrità referenziale fra l'attributo Vigile della relazione INFRAZIONI e la relazione VIGILI e tra gli attributi Prov e Numero di INFRAZIONI e la relazione AUTO. Garantendo il vincolo di integrità referenziale, dalla relazione INFRAZIONI possiamo risalire univocamente a qualsiasi valore delle relazione AUTOMOBILI e VIGILI

Progetto della base di dati

In un progetto di una base di dati possiamo distinguere tre fasi:

- la progettazione concettuale;
- la progettazione logica;
- la progettazione fisica. ((Non viene trattato in questa dispensa)

La progettazione concettuale

Scopo della progettazione concettuale è tradurre il risultato dell'analisi dei requisiti in una descrizione formale che dovrà essere indipendente dal DBMS. La descrizione formale fa riferimento al modello concettuale. Il modello concettuale più utilizzato è l' Entity-Relationship, uno strumento che ci permetterà di produrre uno SCHEMA CONCETTUALE, cioè una rappresentazione semplificata che dovrà comunque contenere tutti gli aspetti interessanti per la base di dati da realizzare. In questa fase ci si deve preoccupare di rappresentare il contenuto informativo della base di dati eliminando le ambiguità tipiche delle frasi in linguaggio naturale.

La progettazione logica

L' obiettivo della progettazione logica è quello di costruire uno schema logico in grado di descrivere in maniera corretta ed efficiente , tutte le informazioni contenute nello schema E-R, prodotto nella fase di progettazione concettuale. Non si tratta però di una semplice traduzione da un modello ad un altro, perché prima di passare allo schema logico, lo schema E-R va ristrutturato per soddisfare due esigenze: quella di semplificare la traduzione e quella di ottimizzare il progetto. La semplificazione dello schema si rende necessaria in quanto non tutti i costrutti del modello E-R hanno una traduzione naturale nel modello logico. Per esempio mentre una entità può essere facilmente rappresentata da una relazione del modello relazionale(una tabella avente gli stessi

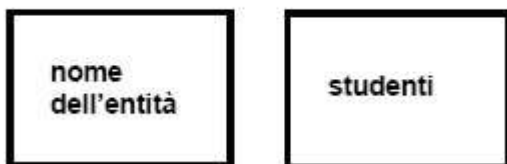
attributi di un' entità) , altre cose, come le gerarchie ISA sono più complicate da rappresentare. Pertanto è necessario prevedere sia una attività di riorganizzazione, sia una attività di traduzione. Possiamo allora dividere la progettazione logica in due parti:

- Ristrutturazione dello schema E-R: fase indipendente dal modello logico scelto che si basa su criteri di ottimizzazione dello schema concettuale.
- Traduzione verso il modello logico: fa riferimento a uno specifico modello logico (nel nostro caso al modello relazionale), e può includere una ulteriore ottimizzazione che si basa sulle caratteristiche del modello logico stesso.

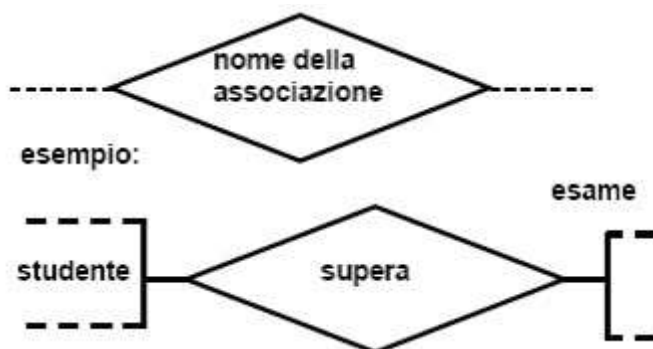
Il modello Entità-Relazione

Il modello Entità-Relazione (E-R) è un modello concettuale di dati, e come tale fornisce una serie di strutture (costrutti), atte a descrivere la realtà in una maniera facile da comprendere e che prescinde dai criteri di organizzazione dei dati nei calcolatori. Il modello E-R usa simboli grafici per favorire l'immediatezza della comprensione. Gli schemi E-R sono infatti schemi essenzialmente grafici con aggiunte di frasi di specifica e di vincolo. L' **entità** rappresenta una classe di oggetti del mondo reale (oggetti sia materiali che immateriali). Ogni entità è caratterizzata da un nome e viene rappresentata con un rettangolo.

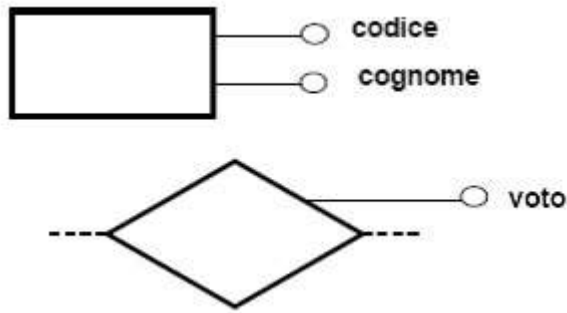
esempio:



Associazione o **relazione** rappresenta un legame logico tra entità. Ogni associazione è caratterizzata da un nome ed è rappresentata in questo modo:



Le **proprietà o attributi** rappresentano caratteristiche delle entità e delle associazioni. Ogni istanza di entità e di associazione possiede un valore per ciascuna proprietà e ogni proprietà è caratterizzata da un nome.

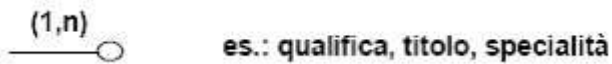


Facciamo però una prima classificazione sulle proprietà:

Proprietà scalare :una proprietà semplice, ad un solo valore:

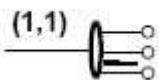


Proprietà multipla : sono ammessi n valori:



Il simbolo (n,m) esprime la **cardinalità** della proprietà dove n e m rappresentano il numero minimo e massimo di valori dell' attributo associati ad ogni occorrenza di entità o relazione;

Proprietà composta: una proprietà composta da più proprietà, per esempio la data (gg,mm,aaaa), o l' indirizzo (spec. top.,denominazione, civico,cap);

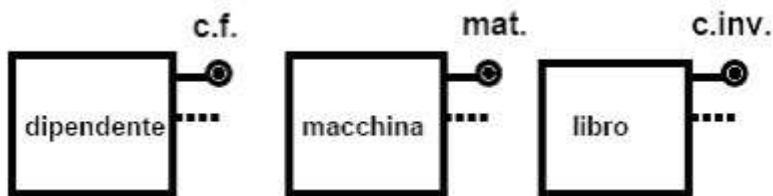


Proprietà multipla composta: una proprietà multipla composta da più proprietà, per esempio il telefono (stato,città, numero), si può infatti avere più di un numero di telefono.

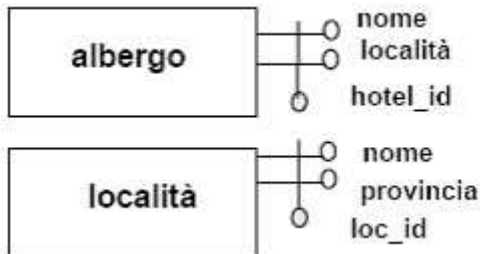
Di seguito un esempio:



Un **identificatore** identifica in modo univoco la singola istanza di entità:



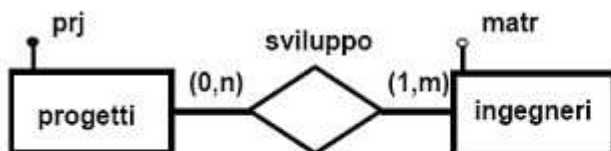
Un identificatore di un'entità può essere composto, e viene indicato in tale caso nel seguente modo:



La **cardinalità delle associazioni** è differente da quella delle proprietà. Nelle relazioni per cardinalità si intende il numero di volte che una data istanza di entità deve o può partecipare alla associazione.

- (1,1) : obbligatoria, una sola volta
- (1,n) : obbligatoria, almeno una volta
- (0,1) : opzionale, una sola volta
- (0,n) : opzionale, n volte

associazioni N:M



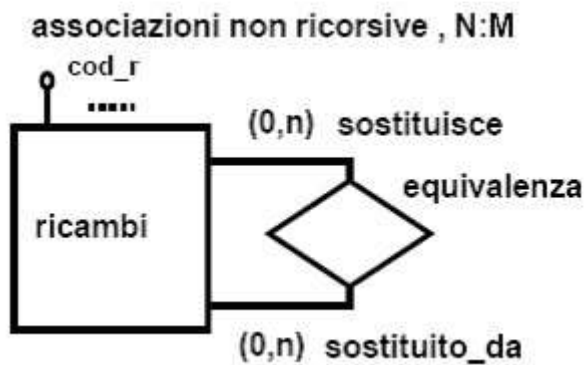
- con vincolo: un ingegnere deve partecipare ad almeno un progetto (1,m)
- senza vincolo: ad un progetto possono partecipare ingegneri, ma può esistere anche un progetto senza ingegneri (0,n)
(detta anche associazione molti a molti)

associazioni 1:1

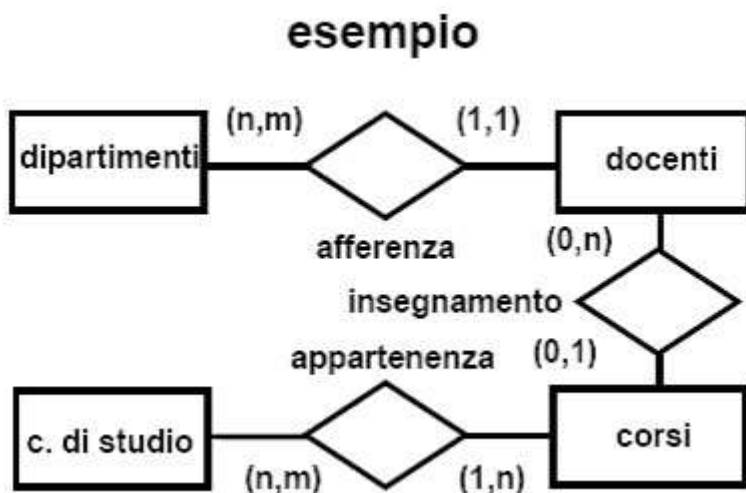


- con doppio vincolo: un reparto deve avere un direttore ed il direttore è uno solo (1,1)
un direttore deve dirigere uno ed un solo reparto (1,1)

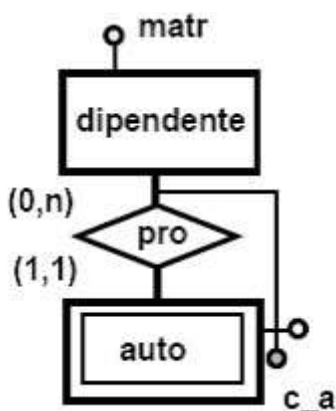
Vi sono poi le **auto-associazioni**: associazioni aventi come partecipanti istanze provenienti dalla stessa entità.



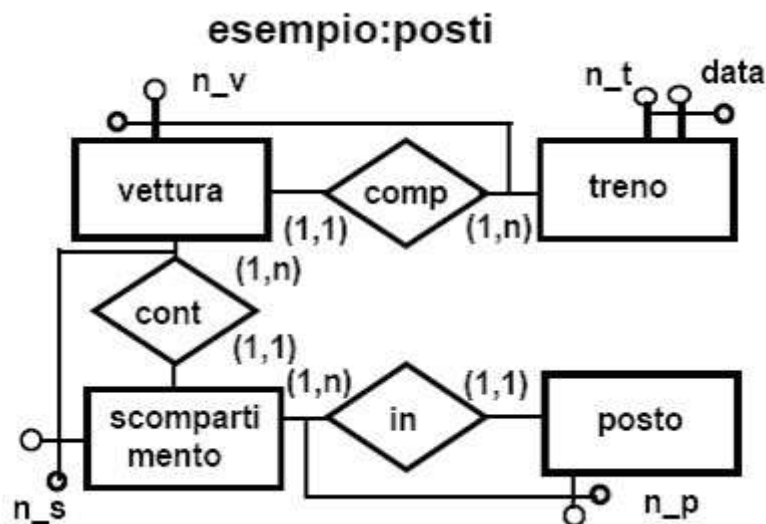
In questo esempio un ricambio può essere sostituito da zero o più ricambi e un ricambio sostituisce zero o più ricambi. Con tutti questi costrutti si possono creare schemi concettuali più o meno complessi. Per esempio schemi di questo tipo:



Ci sono però delle **entità deboli**, ovvero quelle entità che contengono istanze la cui presenza nel sistema è accettata solo se sono presenti determinate istanze di altre entità da cui queste dipendono. In caso di eliminazione dell'istanza di riferimento le istanze deboli collegate devono essere eliminate. L'identificatore dell'entità debole deve contenere l'identificatore dell'entità da cui dipende.



In questo esempio indichiamo che un dipendente può possedere zero o più auto, e una macchina può essere posseduta da un solo dipendente. Auto è però una entità debole, perché dipende dall'esistenza di un dipendente. La regola che l'entità debole deve contenere l'identificatore dell'entità da cui dipende viene rappresentato con una **chiave esterna** come indicato nell'esempio sopra. Uno schema con entità deboli può essere il seguente:



I posti esistono se c'è una scompartimento, che a sua volta esiste se c'è una vettura, che a sua volta esiste se c'è un treno.

Per saperne di più consulta i seguenti approfondimenti:

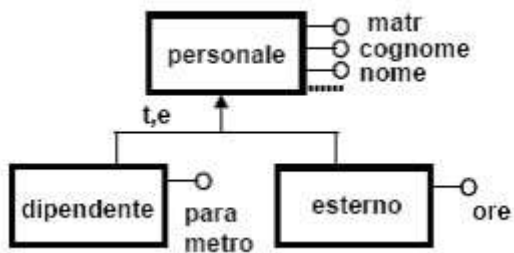
- Le gerarchie nel modello Entità-Relazione
- Sviluppo schema E-R

Le gerarchie nel modello Entità-Relazione

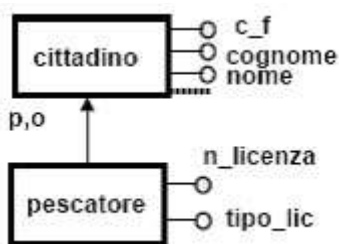
Spesso nella analisi di un settore aziendale può risultare che più entità risultino simili o casi particolari l'una dell'altra. Emerge quindi la necessità di evidenziare sottoclassi di alcune classi. A tale proposito è stato introdotto il concetto di **gerarchia** che definisce il legame logico che esiste tra classi e sottoclassi. La gerarchia concettuale è il legame logico tra un'entità padre E ed alcune entità figlie E1 E2 .. En dove:

- E è la generalizzazione di E1 E2 .. En
- E1 E2 .. En sono specializzazioni di E
- una istanza di Ek è anche istanza di E
- una istanza di E può essere una istanza di Ek

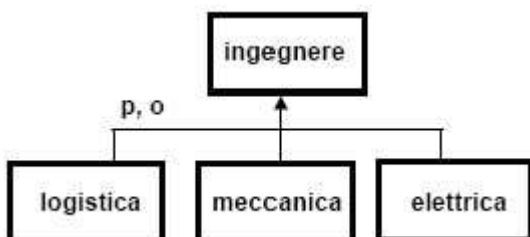
Per esempio un'azienda si avvale dell'opera di professionisti esterni, quindi il suo personale si suddivide in esterni e dipendenti:



La lettera **t** indica **totale**, ovvero ogni istanza dell'entità padre deve far parte di almeno una entità figlia. La lettera **e** sta per **esclusiva**: ogni istanza dell'entità padre deve far parte di una sola delle entità figlie. Quindi l'intero personale può essere per forza o dipendente o esterno. Altro esempio può essere un comune che gestisce l'anagrafe ed i servizi per i suoi cittadini e alcuni di questi richiedono la licenza di pesca (quindi solo alcuni cittadini sono pescatori):



La lettera **p** sta per **parziale**, ovvero ogni istanza dell'entità padre può far parte oppure no di una entità figlia. La lettera **o** sta per **overlapping** (con sovrapposizioni) ed indica che ogni istanza dell'entità padre può far parte di una o più entità figlie. Questo concetto può risultare più semplice con il seguente esempio:



In questo caso la lettera **o** indica che possono esistere ingegneri meccanici, elettrici, e della logistica, ma anche sia meccanici che elettrici o meccanici e logistici. In sostanza le tre qualifiche non si escludono, ogni ingegnere può assumere più qualifiche. Nelle gerarchie si ha una **ereditarietà delle proprietà**. Più precisamente le proprietà dell'entità padre non devono essere replicate sull'entità figlia in quanto questa le eredita. Quindi le proprietà dell'entità padre fanno parte del tipo dell'entità figlia. Nell'esempio dei dipendenti il tipo di dipendente è: (matricola, cognome, nome, indirizzo, data_nascita, parametro) il tipo di esterno è: (matricola, cognome, nome, indirizzo, data_nascita, ore). Le gerarchie concettuali sono anche denominate **gerarchie ISA**.

Sviluppo schema E-R

Lo sviluppo dello schema concettuale E-R , come tutti i progetti di ingegneria, si può eseguire seguendo quattro strategie fondamentali:

- Top-Down
- Bottom-Up
- Inside-Out
- Mista

Top-Down

Si parte dalle specifiche e si costruisce uno schema iniziale. Dallo schema iniziale si arriva per raffinamenti successivi a schemi intermedi e poi allo schema finale. I raffinamenti prevedono l'uso di trasformazioni elementari che operano sul singolo concetto per descriverlo con maggior dettaglio.

Bottom-up

Le specifiche nascono suddivise per sottoprogetti descrittivi frammenti limitati della realtà da schematizzare. Prima si sviluppano i sottoschemi separati e dopo si fondono i sottoschemi per ottenere lo schema finale.

Inside-out

E' una variante della bottom-up, si sviluppano schemi parziali in aggiunta a sottoschemi già definiti precedentemente e separatamente.

Mista

Strategia mista: si parte da uno schema scheletro generale preciso, poi lo si suddivide in sottoschemi da dettagliare e fondere alla fine.