

# Database

4 ore

# Sistemi Informativi

- Base di dati:
  - Insieme organizzato di dati utilizzati per il supporto allo svolgimento di attività (di un ente, azienda, ufficio, persona)
- Sistema informativo
  - Componente di un'organizzazione che gestisce (acquisisce, elabora, conserva, produce) le informazioni di interesse (cioè utilizzate per il perseguimento degli scopi dell'organizzazione)
  - Ogni organizzazione ha un sistema informativo: Università ha informazioni sugli studenti iscritti, sul personale dipendente docente e non docente, sui corsi, sui progetti, ecc.

# Sistemi Informativi

- Il concetto di Sistema Informativo è indipendente dal concetto Sistema Informatico
  - Banche, anagrafe, elenco di battesimi, matrimoni, ecc. esistono da secoli
  - A noi interessa la gestione del sistema informativo con strumenti informatici
- Gestione delle informazioni
  1. Raccolta, acquisizione
  2. Archiviazione, conservazione
  3. Elaborazione, trasformazione
  4. Distribuzione, comunicazione, scambio.

# Sistemi Informativi

- Nelle attività umane, le informazioni vengono gestite in forme diverse
  - Idee informali
  - Linguaggio naturale
  - Disegni, grafici, schemi
  - Numeri e codici
- ...e su vari supporti
  - Mente umana, carta, dispositivi elettronici

Adi 29 magio 1577  
Ill.mo e R.mo mons. Carlo Cardinale  
Benvenuto de' tituli di S. prassede  
Arcivescovo di milo. A me ha fatto  
gratia della parochiale chiesa di m.  
s.ta maria di Caluayrate gherada  
nelli corpi s. di porta orientale di  
milo eretta prima da sua Ill.mo  
de' chierici in parte et sacramento  
de' popoli le anime quante salutano fieri di

- *Adi 30 giugno 1577 E stata battezzata da me prete Lazaro di Conti Rettore primo della parochiale chiesa di m.a s.ta Maria di Caluayrate s(u i) corpi santi di porta orientale di milano nouamente eretta dall' Ill.mo e R.mo mons.re Car.le S.ta prassede Arciuesc.o di milano et a me fattone gratia una figliola nasciuta adi 29 detto di Ambrosio di Lupi e paolina di Galli sua moglie habitanti nella cassina app.ta La chabella E gli è stato posto nome Anastasia.*

# Sistemi Informativi

- Nei 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 poi
  - Nome e cognome
  - Estremi anagrafici
  - Codice fiscale
- Ovvero le informazioni vengono rappresentate in modo essenziale, attraverso i “dati”
  - Cercare l’informazione di interesse in un testo come quello precedente è problema di comprensione del linguaggio naturale molto complesso!

# Sistemi Informativi

- I “dati” non sono ancora “informazione”: è solo nel momento in cui li mettiamo in relazione tra loro e assegniamo loro un significato diventano “informazione”
  - Mario, 2755678 sono due dati elementari. Di per sé non mi forniscono alcuna informazione
  - Se interpretiamo il primo dato come un nome e il secondo come un numero telefonico di Genova, abbiamo informazione. La coppia <nome> <numero> mi fornisce risposta alla domanda “qual è il numero telefonico del mio amico Mario?”
  - Il numero potrebbe essere interpretato come una matricola universitaria. L’informazione contenuta in questo caso sarebbe diversa.
  - Se la coppia <Nome> <numero> è su un foglio appeso sul frigo con il dato aggiuntivo Idraulico, l’informazione è ancora differente...

# Sistemi Informativi

- Perché usare i “dati” per rappresentare conoscenza?
  - La rappresentazione tramite forme più ricche è difficile da trattare
  - In un sistema basato su “dati” è facile accedere alle informazioni volute: si pensi alla ricerca di un numero di telefono...
  - DataBase Management System: sistema informatico che gestisce collezioni di dati 1) grandi, 2) persistenti 3) condivise garantendo 1) privatezza 2) affidabilità 3) efficienza.
  - Esempi di DBMS: Access, DB2, Oracle, Informix, Sybase, SQLServer

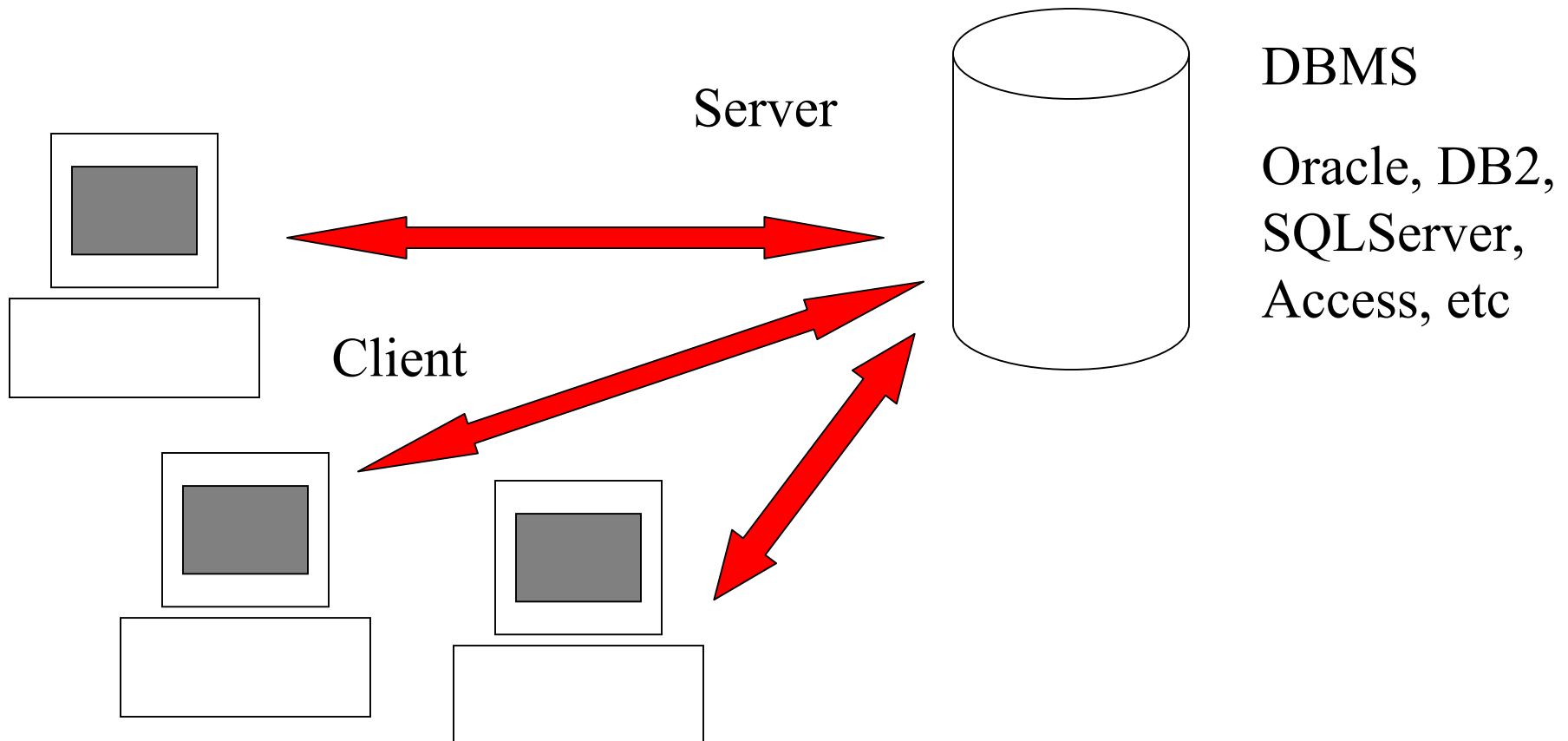


# Sistemi Informativi

- Grandi
  - Dimensioni molto maggiori della memoria centrale. E' necessario avere molto spazio su disco per contenere i dati
- Persistenti
  - I dati hanno un tempo di vita indipendente dalle singole esecuzioni dei programmi che li utilizzano. E' necessario avere ridondanza e meccanismi di backup dei dati
- Condivise
  - Ogni organizzazione è divisa in settori che svolgono diverse attività. Spesso ogni settore ha un sistema informativo proprio, che dialoga con quelli delle altre unità.

# Sistemi Informativi

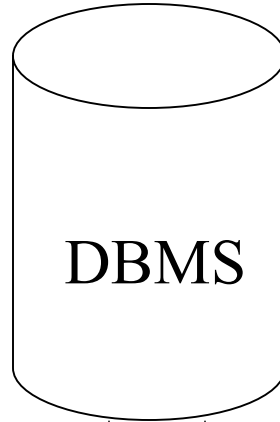
- La struttura è Client-Server



# Basi di Dati

- Le basi di dati sono condivise: brevi accenni a problemi di tipo informatico...
  - E' necessario avere meccanismi di controllo degli accessi, per impedire agli utenti non autorizzati di modificare la base di dati.
  - Bisogna impedire modifiche contemporanee della base di dati da parte di più utenti, per non andare in uno stato consistente.
  - Cosa succede se, da due diversi sportelli bancari, vengono effettuati depositi sullo stesso conto?

# Basi di Dati

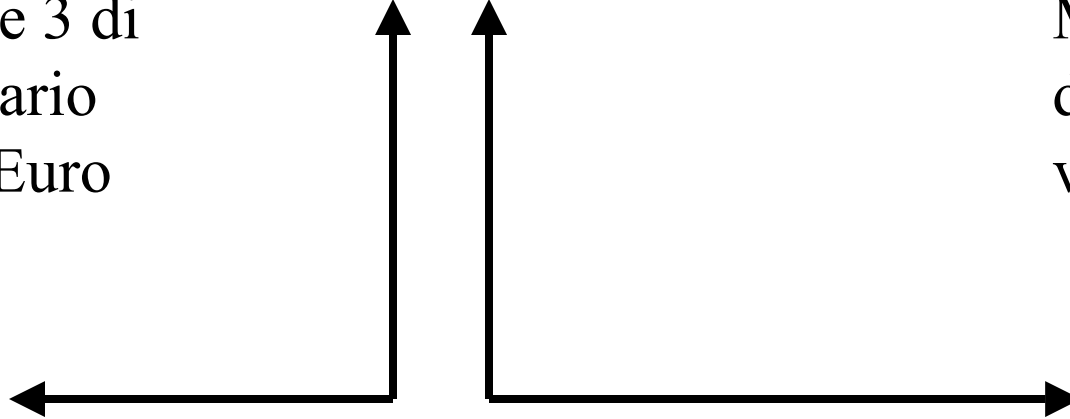


C/C 016344 intestato a  
Pippo Meneghelli  
5000 Euro

Luigi, Filiale 3 di  
Genova S.Ilario  
versa 1000 Euro



Marietto, Filiale 8  
di Imperia Parasio  
versa 2000 Euro



# Basi di Dati

- A Genova, il bancario Luigi si prepara a fare un deposito di 1000 euro sul conto del sig. Pippo Meneghelli. Il programma client si connette al Database e legge che, attualmente, ci sono 5000 euro sul conto.
- A Imperia, il bancario Marietto si prepara a fare un bonifico di 2000 euro sullo stesso conto. Il suo client legge che, attualmente, ci sono 5000 euro sul conto.
- Il programma client di Luigi somma 1000 a 5000. Alla fine dell'operazione ci sono 6000 sul conto.
- Il programma client di Marietto somma 2000 a 5000. Alle fine dell'operazione ci sono 7000 sul conto.
- Dovrebbero esserci 8000 euro, come conseguenza dei due bonifici!

# Basi di Dati

- Esistono dei sistemi per ovviare a questi inconvenienti. In generale si dice che le Transazioni (passaggi da uno stato a un altro del database) devono essere ACIDe...
  - Atomiche (la lettura e la somma non possono essere interrotte da un'altra operazione)
  - Consistenti (devono portare in uno stato consistente)
  - Isolabili (ogni transazione deve essere indipendente dall'esecuzione di altre attività)
  - Durevoli (il risultato deve essere registrato in maniera definitiva, anche in presenza di guasti)

# Basi di Dati

- Che differenza c'è tra un DBMS e un File System?
  - Anche i FileSystem permettono di memorizzare dati in maniera persistente.
  - Il DBMS fornisce strumenti e servizi per accedere ai dati ed estrapolare informazione in maniera efficiente.
  - Un DBMS si appoggia su un FileSystem per memorizzare dati.
  - Spesso si distingue tra livello logico (come sono organizzati i dati nel DBMS) dal livello fisico (come sono memorizzati nel FileSystem). Trattiamo solo il livello logico.

# Basi di Dati

- Come vengono organizzati i dati a livello logico?
  - Il modello più diffuso è quello relazionale; in cui i dati vengono messi in relazione tra loro come fossero “attributi” che contribuiscono a creare una certa “informazione”
  - Ad esempio l’informazione “lezioni del corso di laurea in Ingegneria Ligustica” potrebbe essere una collezione di quadruple: Nome docente, Nome corso, Crediti Formativi, N. Telefonico Docente.
- In ogni base di dati quindi esiste
  - lo schema (invariante nel tempo) che ne descrive la struttura.
  - Un certo numero di istanze (varianti nel tempo), ovvero n-uple di valori che definiscono l’informazione contenuta nel database



Lo schema mi dice che ogni riga (record) deve contenere i seguenti attributi

	Nome Insegnante	Nome Materia	Crediti Formativi	Numero Telefonico
	Filippo Tubinga	Analisi elettroquantica degli ipoconduttori	10	010 256796
	Mario Meloni	Lingue Salmistrade	5	010 222760
	Piero Paciughi	Storia del fumetto argentino	5	010 255656
..	Piero Paciughi	Storia del fumetto brasiliano	5	010 255656

Questa è un'istanza, in cui ogni attributo assume un valore ben specifico.

# Basi di Dati

- Per estrarre informazione dal DataBase, il linguaggio più noto è SQL (Structured Query Language)
  - Nell'esempio precedente, per sapere il Numero Telefonico del docente che insegna Lingue Salmistrade:  
**SELECT** Nome Insegnante, Numero Telefonico  
**FROM** Lezioni Di Ingegneria Ligustica  
**WHERE** Nome Materia = Lingue Salmistrade

Telef : Query di selezione		
	Nome Insegnante	Numero Telefonico
	Mario Meloni	010 222760

# Basi di Dati

**SELECT** Nome Insegnante, Numero Telefonico  
*(seleziona gli attributi Nome Insegnante, Numero Telefonico)*

**FROM** Lezioni Di Ingegneria Ligustica  
*(dalla tabella Lezioni Di Ingegneria Ligustica)*

**WHERE** Nome Materia = Lingue Salmistrade  
*(per quelle istanze dove Nome Materia = Lingue Salmistrade)*

- Si possono fare query molto più complesse che coinvolgono più tabelle.

# Basi di Dati

- Alcuni DBMS permettono un'interazione non testuale che si basa su un principio simile
  - ACCESS si basa sulla creazione di QUERY
  - In entrambi i casi si possono fare ricerche molto più complesse di quelle viste fino ad ora...

Campo:	Nome Insegnante	Numero Telefonico	Nome Materia
Tabella:	Lezioni di Ingegneria	Lezioni di Ingegneria	Lezioni di Ingegneria
Ordinamento:			
Mostra:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteri:			"Lingue Salmistrade"
Oppure:			

# Basi di Dati

- Personaggi e interpreti
  - Progettisti e realizzatori di DBMS
  - Progettisti della base di dati e amministratori (DBA)
  - Progettisti e programmatori di applicazioni che accedono alla base di dati
  - Utenti: terminalisti che eseguono applicazioni predefinite e utenti casuali
- Spesso l'utente "casalingo" che crea il proprio DataBase con Access (ad esempio per classificare i libri nella propria biblioteca) svolge tutti questi ruoli contemporaneamente

# Il modello relazionale

- Più specificamente, definiamo “relazione” un insieme una o più n-uple ordinate
  - Nel caso precedente, “lezioni del corso di laurea in Ingegneria Ligustica” era una “relazione” costituita da quadruple.
  - Anche la relazione “Risultati Domenica 13 febbraio” è costituita da 4-uple (collezioni di 4 valori) del tipo “squadra di calcio” “squadra di calcio” “n. goal” “n. goal”. “squadra di calcio” e “n. goal” vengono detti “domini”.
  - Il dominio mi impedisce di inserire valori non validi: non posso scrivere “pistola” o “carota” dove devo inserire una “squadra di calcio”. Allo stesso modo, non posso scrivere “Lazio” al posto di un punteggio.

# Il modello relazionale

- Purtroppo, siccome alcuni domini sono ripetuti, è importante l'ordine in cui i valori sono scritti! La tupla Genoa Lazio 1 8 non è uguale a Lazio Genoa 1 8. Per questo si introducono gli attributi, un ruolo che viene associato ad ogni dominio permettendo di variare l'ordine dei valori.
- In questa tabella, gli attributi sono “Squadra in casa” “squadra fuori” “goal casa” e “goal fuori”

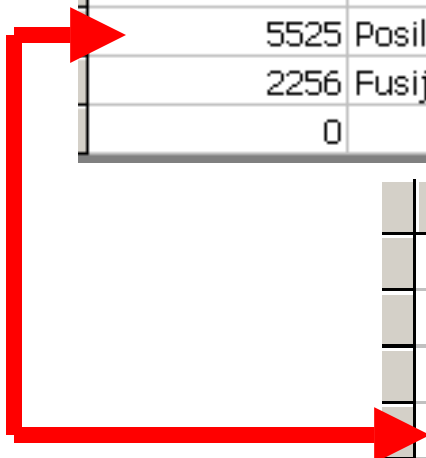
Squadra in casa	Squadra fuori	Goal casa	Goal fuori
Genoa	Lazio	1	8
Milan	Parma	0	11
Lanerossi Vicenza	Ternana	12	2
Roma Termini	Lucca	3	6
Vittorio Al mare	Fiesole	2	2

# Il modello relazionale

- Il modello è basato sui valori:
  - Se abbiamo più relazioni, i riferimenti tra dati in relazioni diverse possono essere ricavati per mezzo dei valori che compaiono nelle n-uple

Matricola	Cognome	Nome	Data di nascita
1566	Pernach	Guido	12/20/1985
1677	Manolo	Ezio	3/22/1983
5525	Posillipani	Marta	2/12/1975
2256	Fusiju	Takaori	6/8/1994
0			

	Studente	Voto	Corso
	1566	16	Lingue Salmistrade
	1677	22	Storia del fumetto argentino
	2256	31	Storia del fumetto argentino
	5525	19	Storia del fumetto brasiliano





# Il modello relazionale

- Supponiamo di cercare tutti gli studenti che hanno preso 31 in almeno un corso
- E' possibile in SQL scrivere

**SELECT** Cognome, Nome, Voto

**FROM** Studenti, Esami

**WHERE** Studenti.Matricola = Esami.Studente AND  
Esami.Voto = 31

	Cognome	Nome	Voto
•	Fusiju	Takaori	31

# Il modello relazionale

- ...o creare la seguente Query in ACCESS

Campo:	Cognome	Nome	Voto	Studente
Tabella:	Studenti	Studenti	Esami	Esami
Ordinamento:				
Mostra:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteri:			31	[Studenti].[Matricola]
Oppure:				

# Il modello relazionale

- I valori che vengono usati per mettere in relazione due tabelle (la matricola nel caso precedente) spesso devono identificare univocamente la n-upla (non essere ripetuti!) in almeno una delle due tabelle...
  - Si pensi al caso in cui due studenti abbiano la stessa matricola nella tabella “Studenti”. Quale dei due viene fornito come risultato? Uno solo dei due ha preso 31!
  - E’ però possibile (e anzi auspicabile) che la stessa matricola sia ripetuta più volte nella tabella “Esami”: lo stesso studente ha sostenuto più di una prova.
  - Per ogni tabella è conveniente definire un attributo o un insieme di attributi che identifica univocamente ogni record: questo insieme viene detto “chiave”
  - In una tabella è possibile avere più chiavi!

chiave



Studente	Voto	Corso
1566	18	Lingue Salmistrade
1677	22	Storia del fumetto argentino
2256	31	Storia del fumetto argentino
5525	19	Storia del fumetto brasiliano
1677	22	Lingue Salmistrade
2256	30	Analisi elettroquantica degli ipoconduttori

Relazione  
“Esami”

Relazione  
“Studenti”

chiave



Matricola	Cognome	Nome	Data di nascita
1566	Pernach	Guido	12/20/1985
1677	Manolo	Ezio	3/22/1983
5525	Posillipani	Marta	2/12/1975
2256	Fusiju	Takaori	6/8/1994
0			

# Il modello relazionale

- Nella tabella studenti la chiave è l'attributo “matricola”
  - E' minimale, perché contiene un solo attributo.
  - In un'altra tabella, è possibile che la chiave sia costituita dalla coppia di attributi “nome” e “cognome”, oppure da “nome”, “cognome” e “data di nascita”
  - Se la chiave è minimale, cioè costituita da un solo attributo, è più semplice collegare le relazioni Studenti ed Esami. Si provi a immaginare l'interrogazione precedente in SQL con tabelle in cui la chiave è costituita da “nome” e “cognome”

**SELECT** Cognome, Nome, Voto

**FROM** Studenti, Esami

**WHERE** Studenti.Cognome = Esami.Cognome AND

Studenti.Nome=Esami.Nome AND

Esami.Voto = 31

# Il modello relazionale

- Esiste sempre una chiave?
  - Una relazione non può contenere n-uple uguali.
  - Ogni relazione ha almeno una chiave, costituita da tutti i suoi attributi.
  - A volte, se la chiave è costituita da molti attributi, si può aggiungere un identificatore numerico ai record per avere una chiave minimale...
  - Ad esempio, nella relazione “Esami” potremmo aggiungere un codice univoco per ogni istanza, che identifica la prova sostenuta da un dato studente con il voto risultante.

# Informazione incompleta

- Il modello relazionale impone ai dati una struttura rigida
  - Le informazioni sono rappresentate per mezzo di n-uple
  - Solo alcuni formati di n-uple sono ammessi, quelli che corrispondono agli schemi della relazione
  - E' però possibile che i dati disponibili non corrispondano al formato previsto; ad esempio è possibile che un dato non sia presente (nell'esempio precedente, la data di nascita dello studente)
  - Non conviene accettare n-uple con valori incompleti: talvolta è però necessario e si adotta un valore speciale (NULL) che corrisponde all'assenza del dato
  - Se abbiamo valore nullo nella “chiave” non è facile identificare le n-uple e realizzare facilmente riferimenti tra relazioni

# Vincoli di integrità

- Esistono istanze di basi di dati che, pur sintatticamente corrette, non rappresentano informazioni possibili per l'applicazione
  - Nell'esempio precedente, il voto 31 non è valido in un sistema di punteggio in trentesimi.
  - Vincolo di integrità è una proprietà che deve essere soddisfatta dalle istanze che rappresentano informazioni corrette per l'applicazione
  - Un vincolo associa ad ogni istanza il valore “vero” o “falso” a seconda che sia rispettato o meno.
- Un vincolo particolare è quello di “chiave primaria”; definiamo una chiave su cui non sono ammessi valori NULLI



# Vincoli di integrità

- Perché mettere vincoli?
  - Permettono una descrizione più accurata della realtà e sono utili nella progettazione
- Vincoli intrarelazionali (all'interno di una stessa relazione)
  - Vincoli su valori (o di dominio): 31 non è lecito
  - Vincoli di enunpla: il voto 27 non può avere l'attributo "lode"
- Vincoli interrelazionali

# Vincoli di integrità

- Esempio di Vincoli di dominio con Access

Generale	Ricerca
Dimensione campo	Intero lungo
Formato	
Posizioni decimali	Automatiche
Maschera di input	
Etichetta	Voto
Valore predefinito	0
Valido se	[Voto]>17 And [Voto]<31
Messaggio errore	Il Voto deve essere compreso tra 18 e 30
Richiesto	No
Indicizzato	Sì (Duplicati non ammessi)

# Vincoli di integrità

- Integrità referenziale
  - Informazioni in relazioni diverse sono correlate attraverso valori comuni, in particolare i valori delle chiavi (primarie)
  - Le correlazioni devono essere coerenti.
  - Un vincolo di integrità referenziale impone ai valori in una relazione di apparire come valori della chiave primaria di un'altra relazione.
  - Ad esempio, potremmo mettere un vincolo simile sull'attributo “studenti” nella tabella “esami” e la chiave primaria “matricola” della tabella “studenti”; altrimenti, quando troviamo uno studente che ha preso 30, non sappiamo a quale nome e cognome collegarlo!

Matricola	Cognome	Nome	Data di nascita
1566	Pernach	Guido	12/20/1985
1677	Manolo	Ezio	3/22/1983
5525	Posillipani	Marta	2/12/1975
2256	Fusiju	Takaori	6/8/1994
0			

- La matricola 4555 ha preso 30... ma a che studente corrisponde?

Studente	Voto	Corso
1566	16	Lingue Salmistrade
1677	22	Storia del fumetto argentino
2256	31	Storia del fumetto argentino
5525	19	Storia del fumetto brasiliano
1677	22	Lingue Salmistrade
2256	30	Analisi elettroquantica degli ipoconduttori
4555	30	Lingue Salmistrade

# Creare una Base di Dati

- Il linguaggio SQL permette di creare tabelle, oltre che interrogarle (NUMERIC e CHAR sono domini):

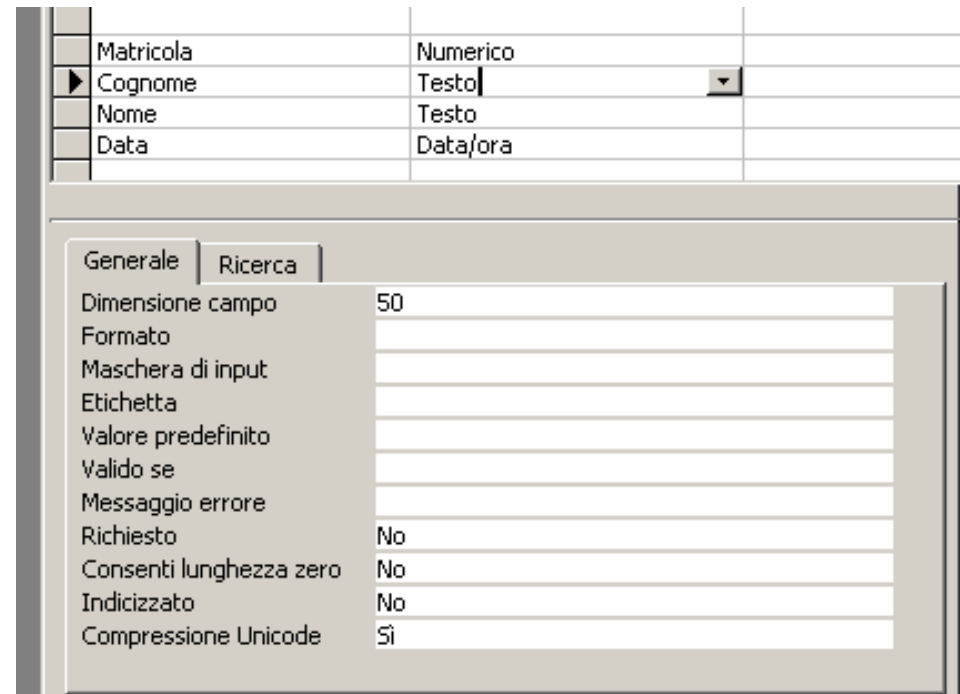
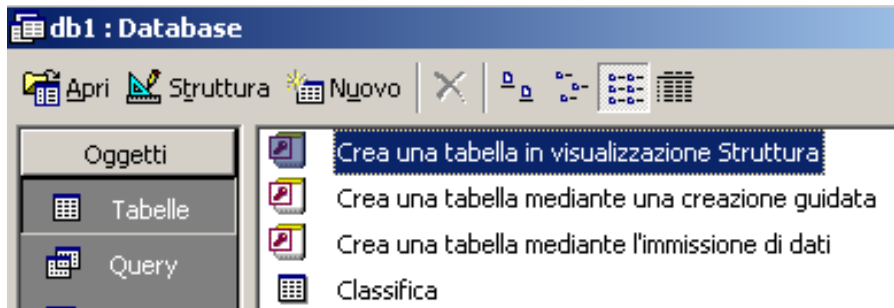
```
CREATE TABLE Studenti (  
  Matricola NUMERIC PRIMARY KEY,  
  Cognome CHAR(20) NOT NULL,  
  Nome CHAR(20) NOT NULL,  
  Data di Nascita CHAR(10) NOT NULL  
)
```

- E' possibile creare domini più restrittivi

```
CREATE DOMAIN Voto  
AS SMALLINT DEFAULT NULL  
CHECK (value >=18 AND value <=30)
```

# Creare una Base di Dati

- ACCESS, oltre a SQL, permette di creare Tabelle graficamente selezionando gli attributi con strumenti appropriati
  - Notiamo di nuovo come sia possibile aggiungere vincoli agli attributi



# Progettare una Base di Dati

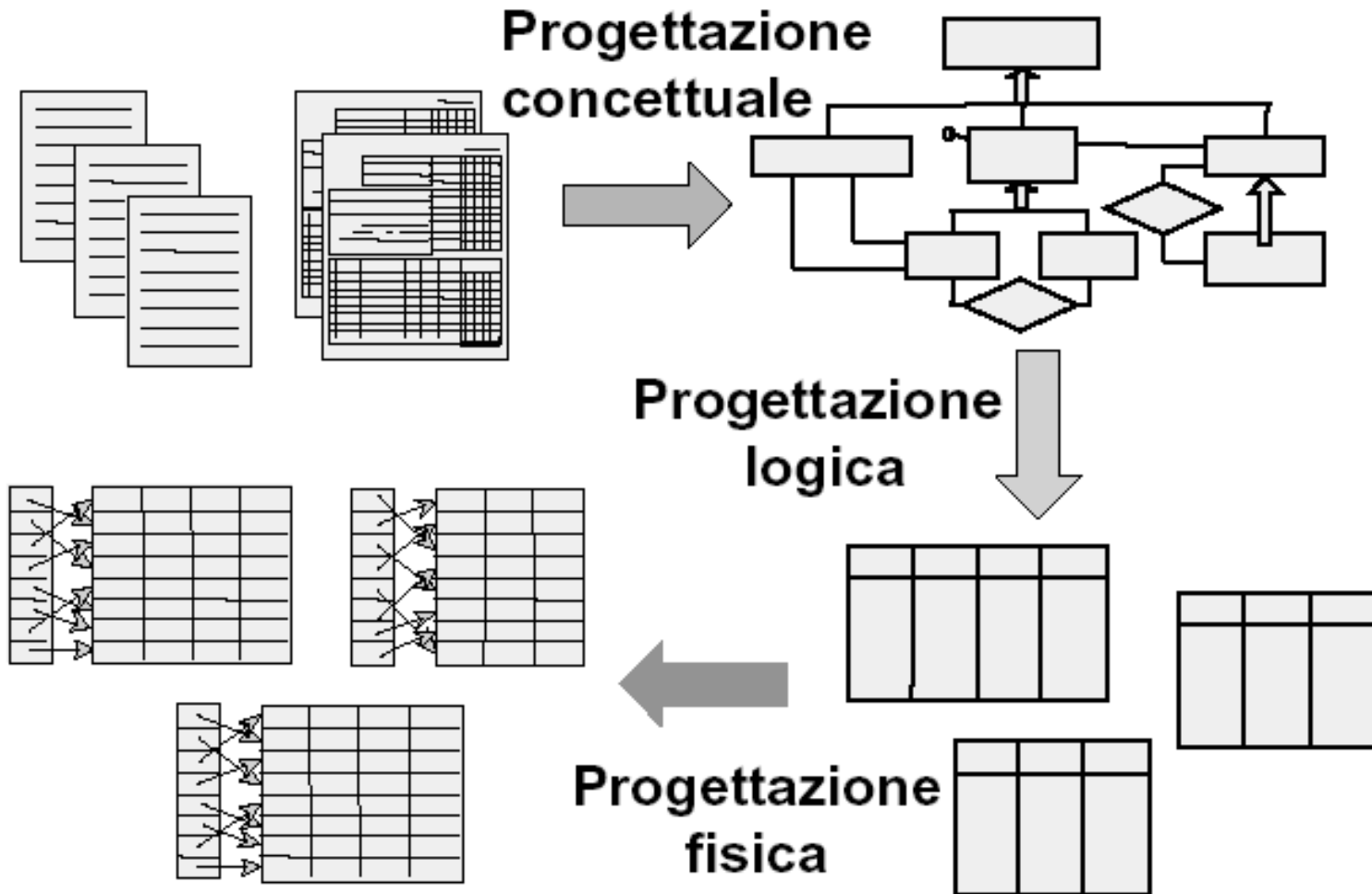
- E' la parte più interessante nello sviluppo di un sistema informativo.
  - Schema fisico (come sono organizzati i dati su Hard Disk): tanti file? Un solo file? Distribuiti su più HardDisk? Con quali nomi?
  - Schema logico (utilizzato nei DBMS per l'organizzazione dei dati, è indipendente dallo schema fisico): modello relazionale, a oggetti, ecc.
  - Schema concettuale (permette di rappresentare i dati in modo indipendente dallo schema logico), cerca di descrivere concetti del mondo reale.
  - Gli schemi concettuali sono usati nelle fasi preliminari di progettazione. Il più noto è il modello Entità-Relazione.

# Progettare una Base di Dati

- Con una Base di Dati complessa, se proviamo a definire direttamente lo schema logico
  - Non sappiamo bene da dove cominciare
  - Rischiamo di perderci subito nei dettagli
  - Dobbiamo pensare subito a come correlare le varie tabelle (chiavi etc.)
  - I modelli logici sono rigidi, con regole ben prefissate.
- I modelli concettuali
  - Servono per ragionare sulla realtà, indipendentemente dagli aspetti realizzativi
  - Permettono di rappresentare le classi di dati di interesse e le loro correlazioni
  - Prevedono efficaci rappresentazioni grafiche (utili anche per documentazione e comunicazione)



# Progettare una Base di Dati



# Modello E-R

- Il più diffuso modello concettuale è il modello Entità-Relazione.
- I costrutti disponibili sono
  - Entità
  - Relazione
  - Attributo
  - Identificatore
  - Generalizzazione
  - ...

# Entità

- Entità: classe di oggetti con proprietà comuni e esistenza autonoma
  - es. studente, città, conto corrente...
- Occorrenza (o istanza) di entità: elemento della classe
  - es. lo studente Paolo Pizzagalli, la città Parigi, il conto corrente 1777282...
- Nello schema concettuale vengono rappresentate le entità, non le singole istanze.

# Entità

- Rappresentazione grafica di entità
  - Ogni entità ha un nome univoco, sufficientemente espressivo e scelto seguendo opportune convenzioni (ad es. uso del singolare)



Studente

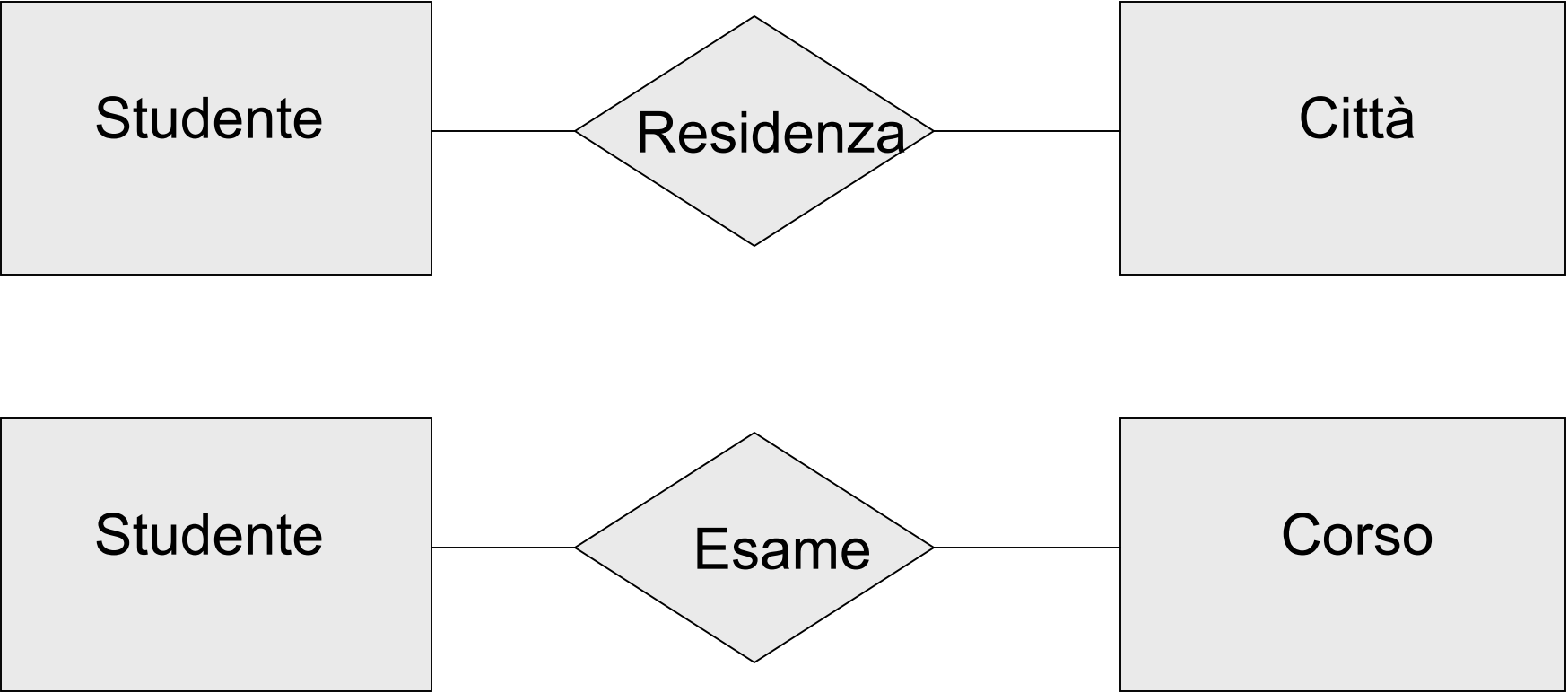
Città

Conto Corrente

# Relazione

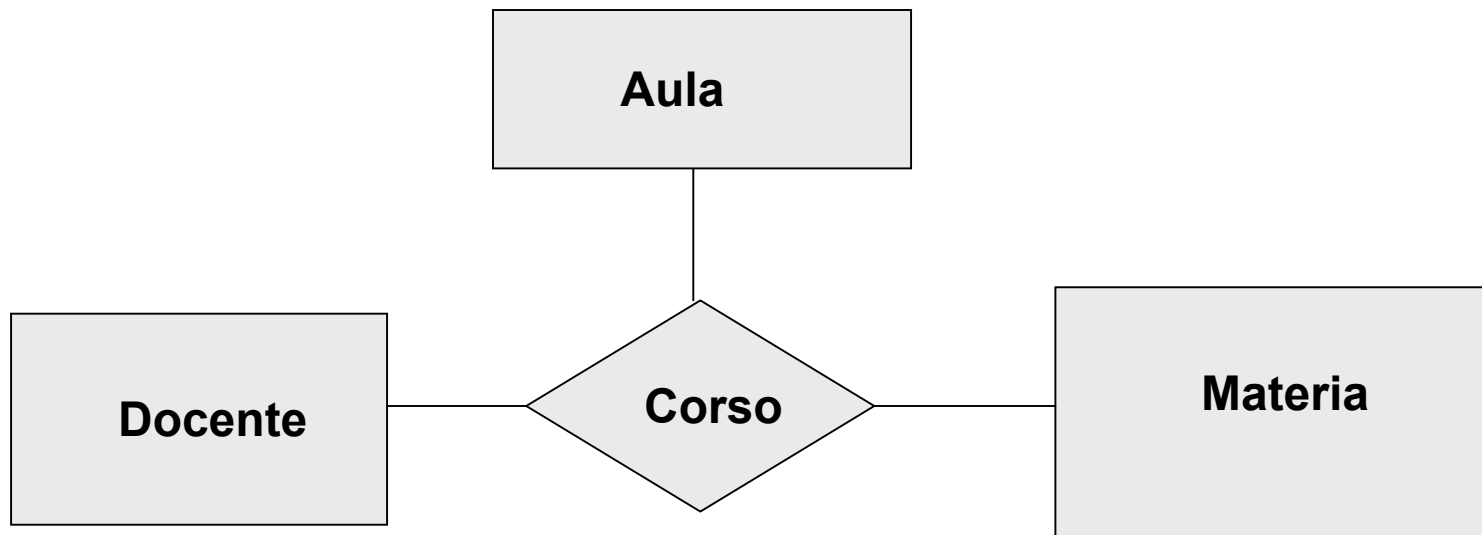
- Relazione: Legame logico tra due o più entità, rilevante nell'applicazione di interesse
  - Residenza tra Studente e Città
  - Esame tra Studente e Corso
  - Ogni entità ha un nome univoco, sufficientemente espressivo e scelto seguendo opportune convenzioni (ad es. uso del singolare, uso di sostantivi invece che verbi, ecc.)

# Relazione



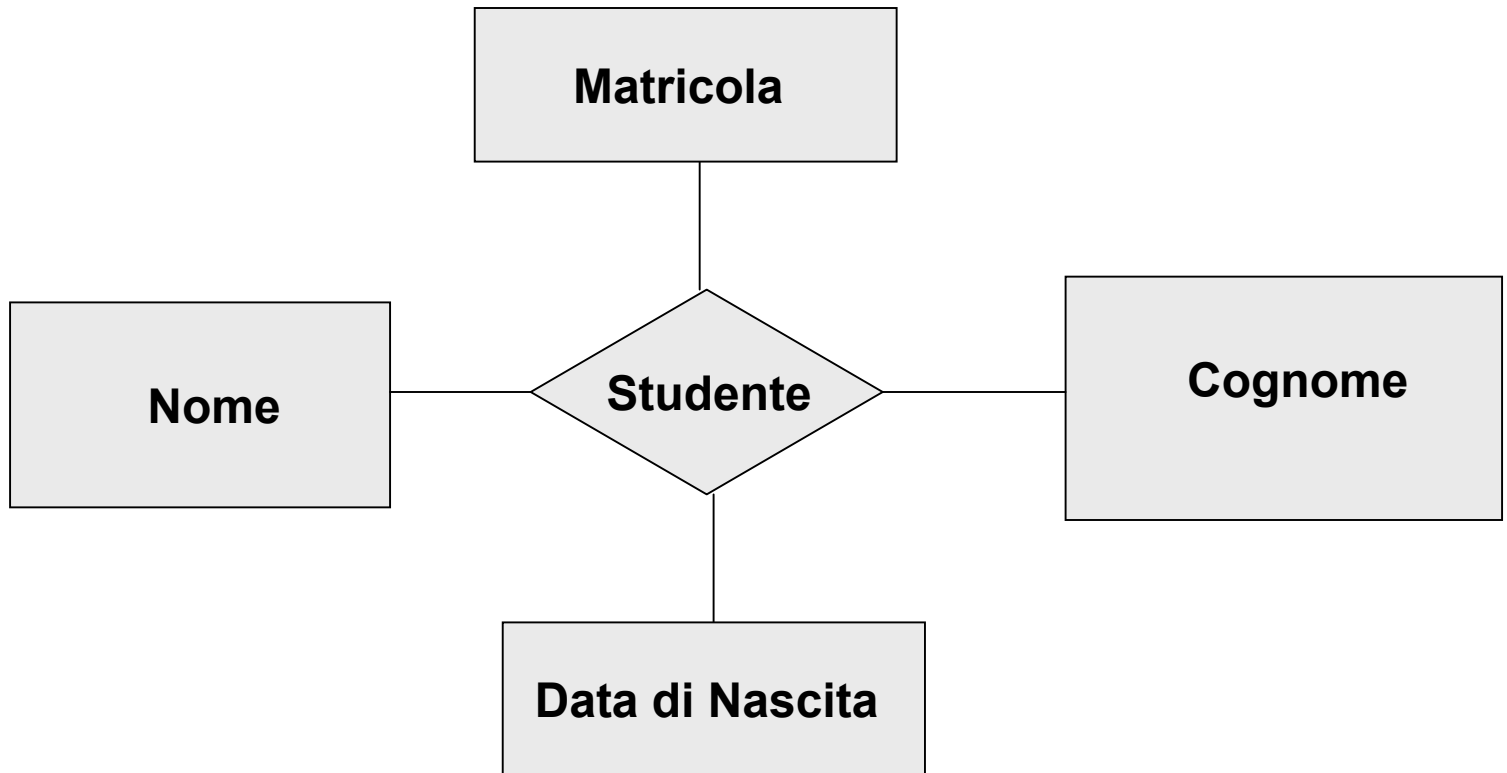
# Relazione

- La relazione può coinvolgere anche più entità



# Relazione

- O ancora...





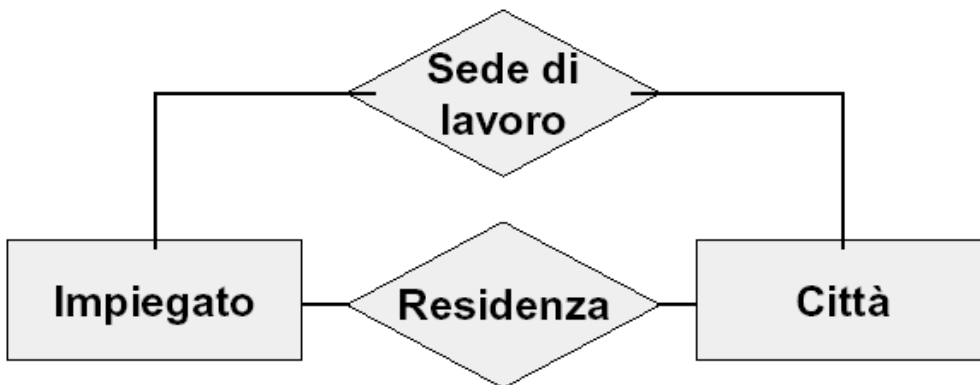
# Relazione

- Una occorrenza di una relazione binaria è una coppia di occorrenze di entità, una per ciascuna entità coinvolta.
  - Ad es. un'occorrenza di Residenza potrebbe essere la coppia Paolo Pizzagalli, Parigi
- Una occorrenza di una relazione n-aria è una n-upla di occorrenze di entità, una per ciascuna entità coinvolta
  - Ad es. una occorrenza di Corso potrebbe essere la terna Lingue Salmistrade, Piero Farozzi, Aula 04.
  - Nell'ambito di una relazione non ci possono essere occorrenze (coppie, n-uple) ripetute.

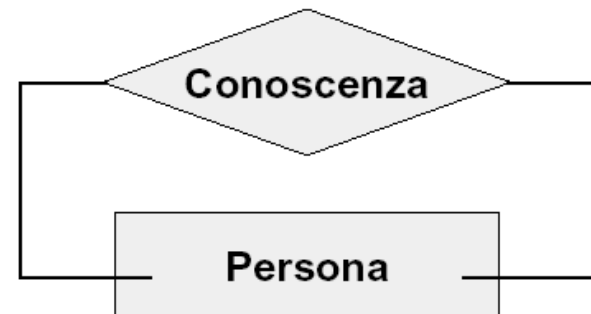
# Relazione

- Si possono immaginare relazioni diverse sulle stesse entità...

**Due relationship sulle stesse entità**

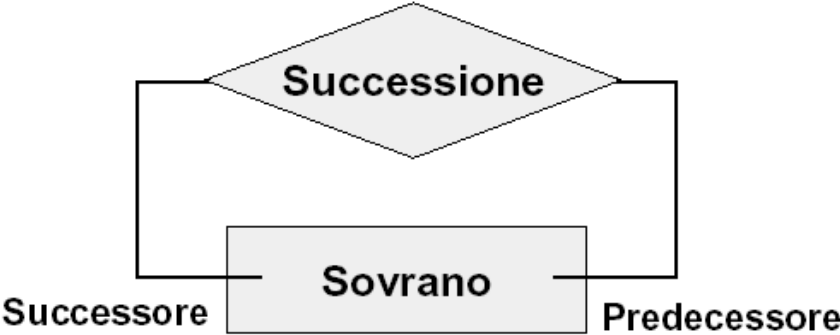


**Relationship ricorsiva:  
coinvolge “due volte” la stessa entità**

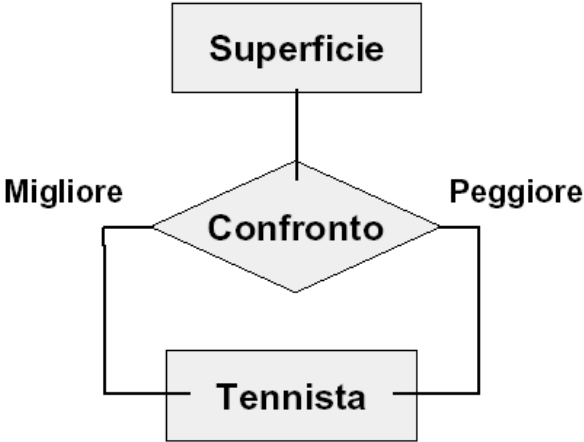


# Relazione

Relationship ricorsiva con "ruoli"

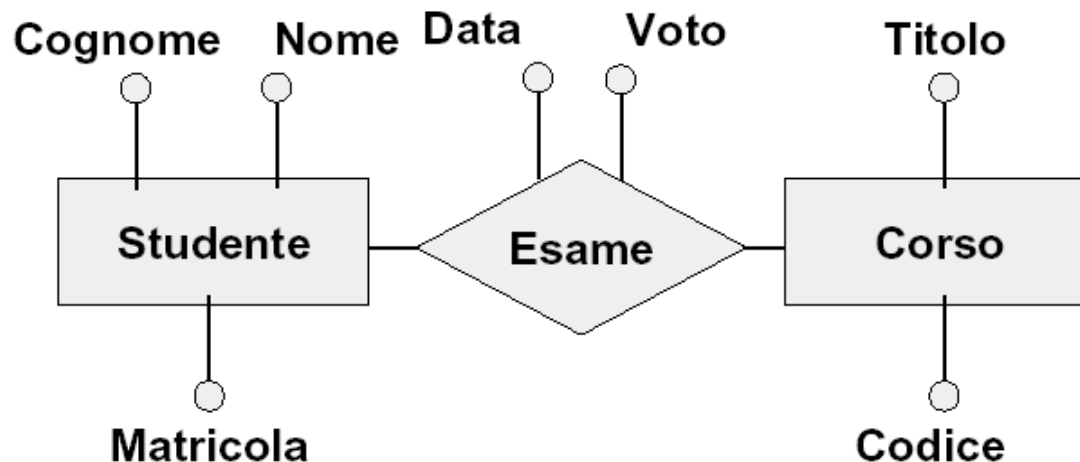


Relationship ternaria ricorsiva



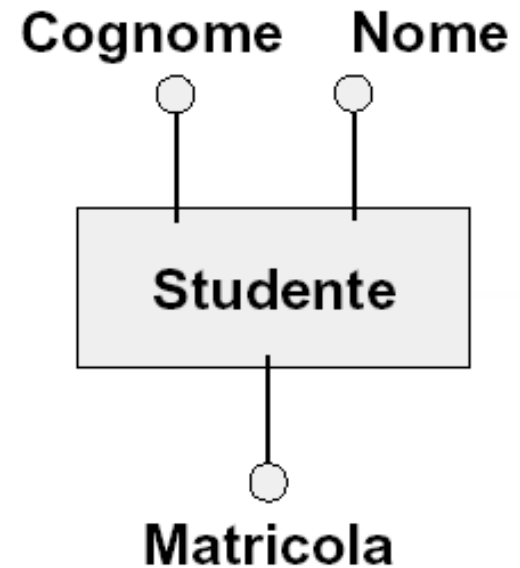
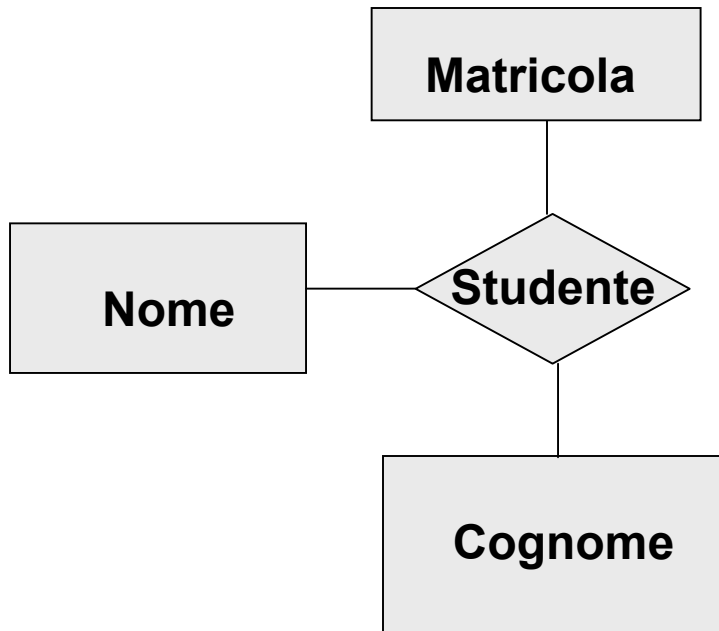
# Attributi

- **Attributo**
  - Proprietà elementare di un'entità o di una relazione, di interesse ai fini dell'applicazione.
  - Associa ad ogni occorrenza di entità o di relazione un valore appartenente a un insieme detto dominio dell'attributo



# Attributi

- Quale di queste due rappresentazioni è corretta?
  - Non esiste una regola fissa... Un suggerimento: verificare se cognome è un'entità coinvolta anche in altre relazioni.



# Cardinalità

- Cardinalità
  - Di relazione coppia di valori associati ad ogni entità che partecipa a una relazione
  - Specificano il numero minimo e massimo di occorrenze della relazione cui ciascuna occorrenza di entità può partecipare.
- Per semplicità usiamo solo tre valori
  - Cardinalità minima 0 (partecipazione opzionale) o 1 (partecipazione obbligatoria)
  - Cardinalità massima 1 o N (nessun limite sul numero di relazioni cui può partecipare)

# Cardinalità



Ogni studente ha una  
residenza (non più di una)

Ogni città può avere un  
numero qualsiasi di studenti  
residenti.

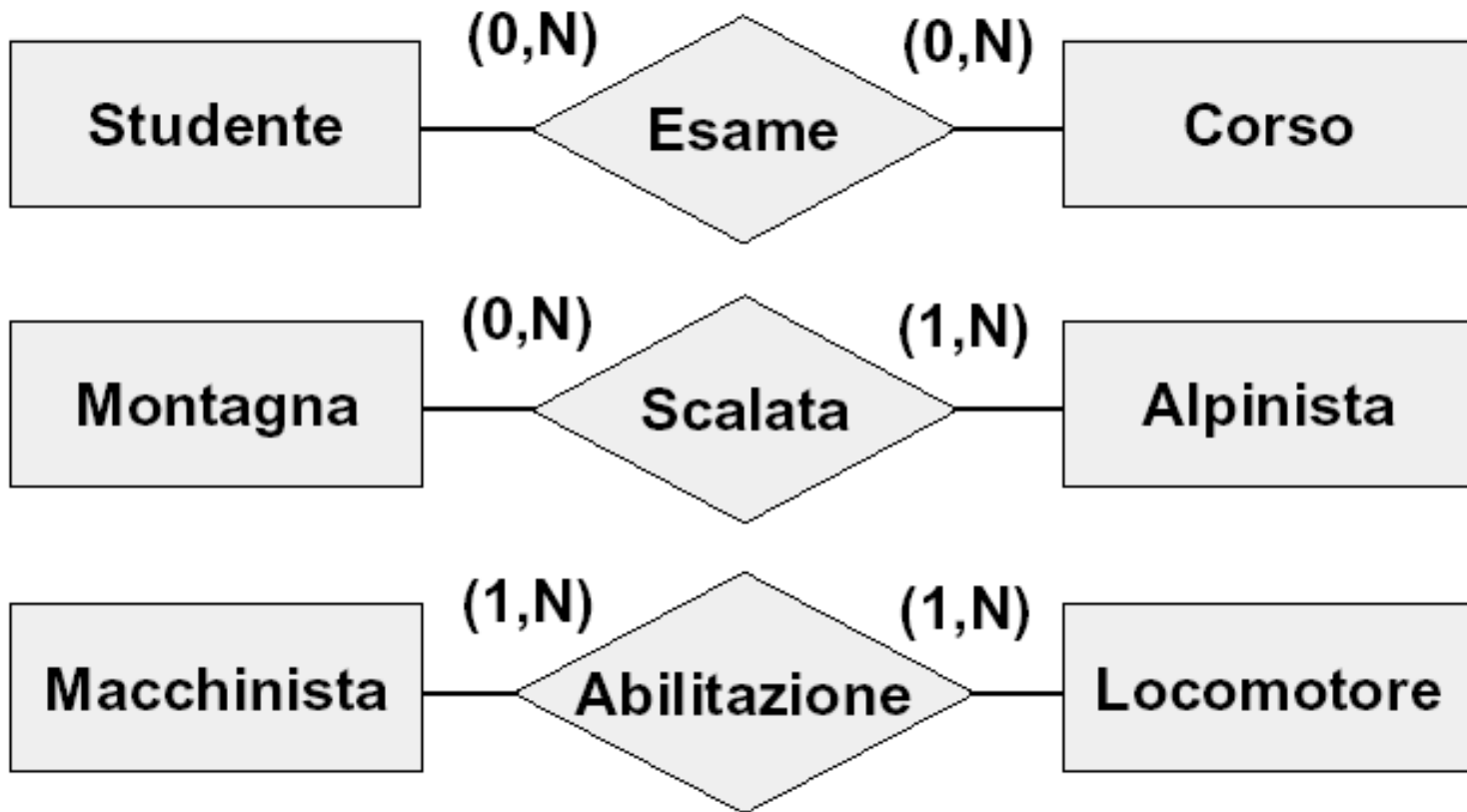
# Cardinalità

- Con riferimento alla cardinalità massima abbiamo relazioni
  - Uno a uno
  - Uno a molti
  - Molti a molti
- Alcune considerazioni
  - Attenzione al verso nelle relazioni uno a molti.
  - Le relazioni uno a uno di tipo obbligatorio sono molto rare...



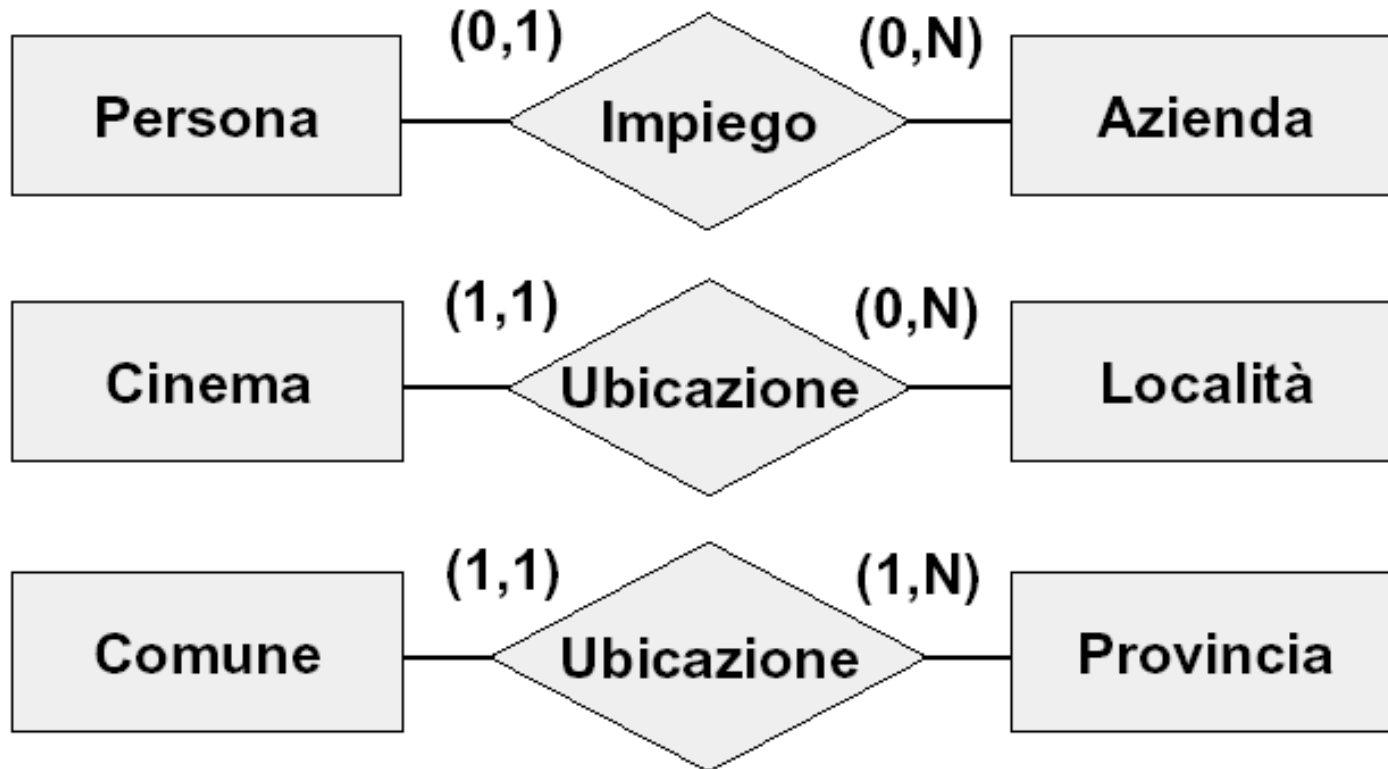
# Cardinalità

- Relazioni “molti a molti”



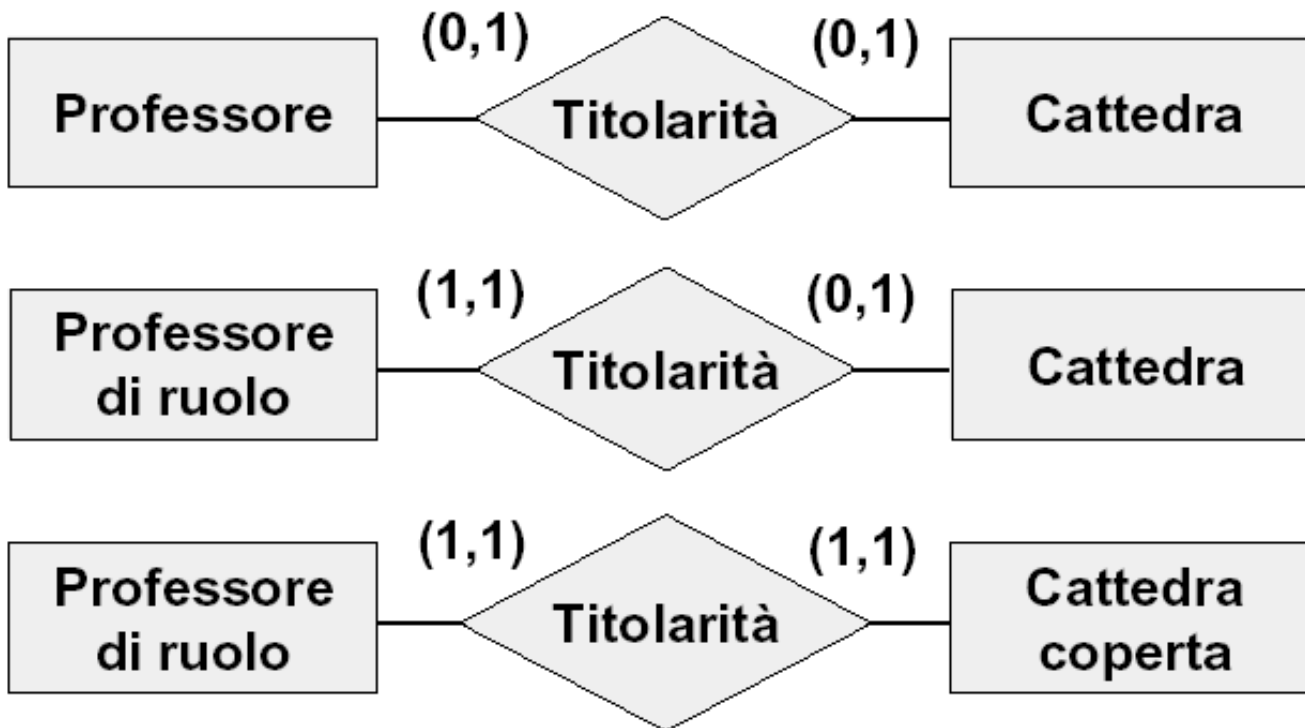
# Cardinalità

- Relazioni “uno a molti”



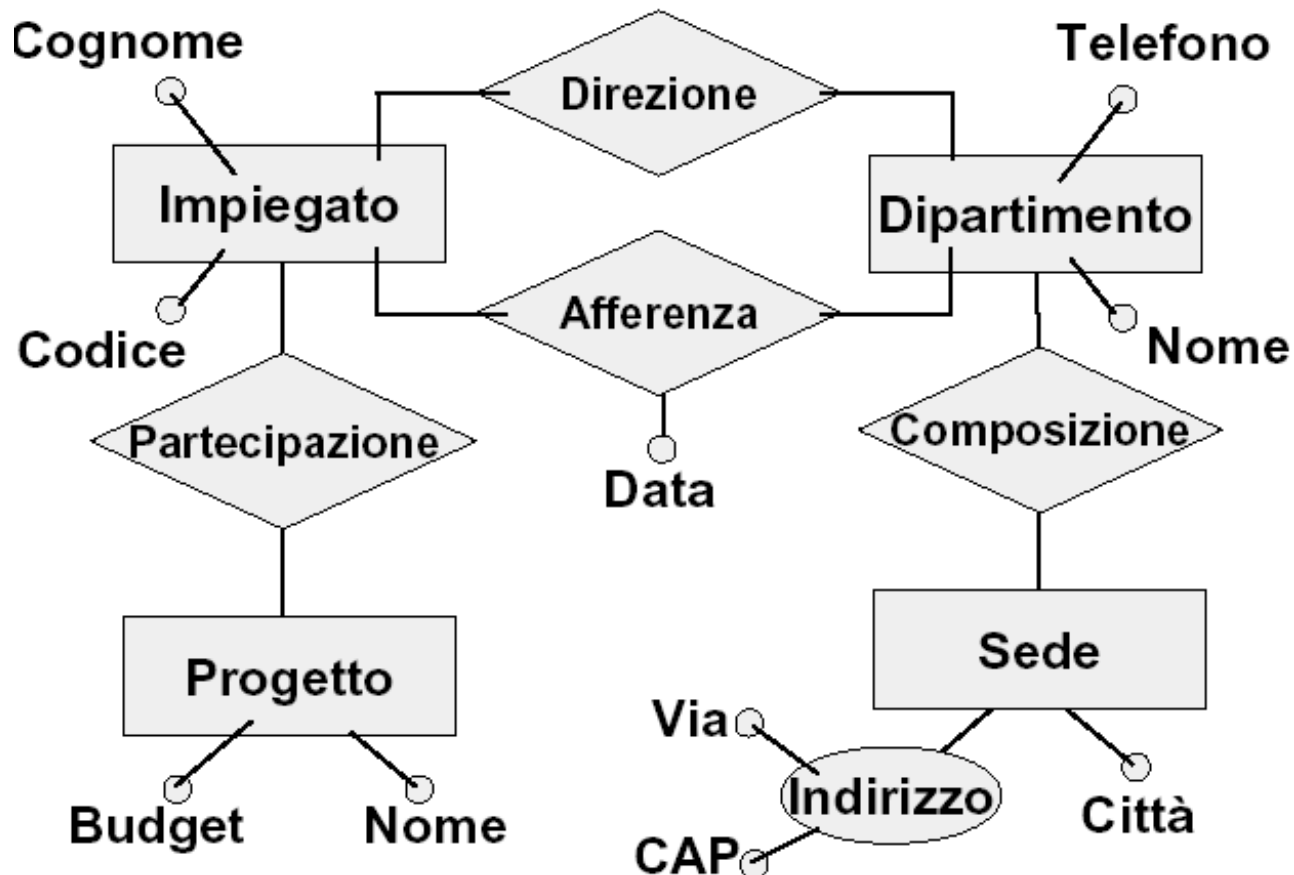
# Cardinalità

- Relazioni “uno a uno”



# Modello E-R

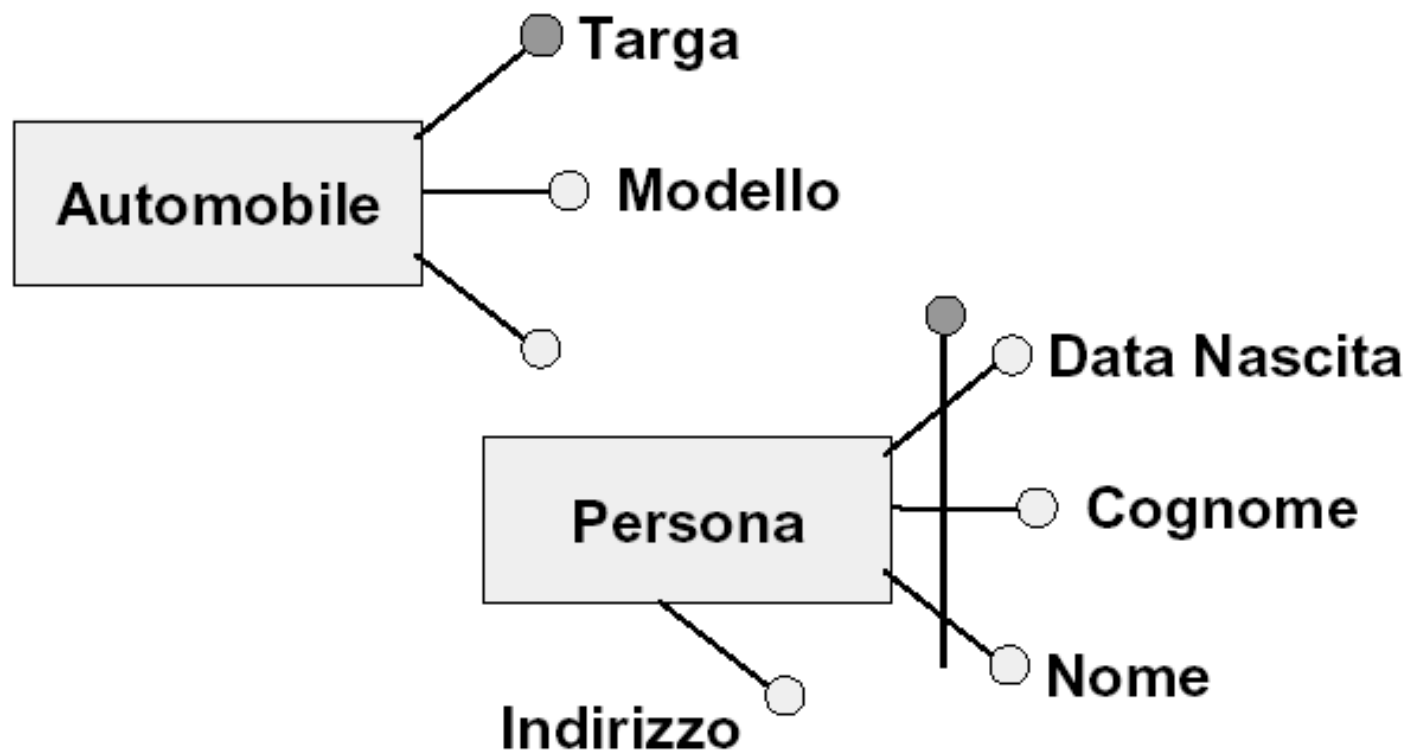
- Si possono costruire schemi piuttosto complessi...



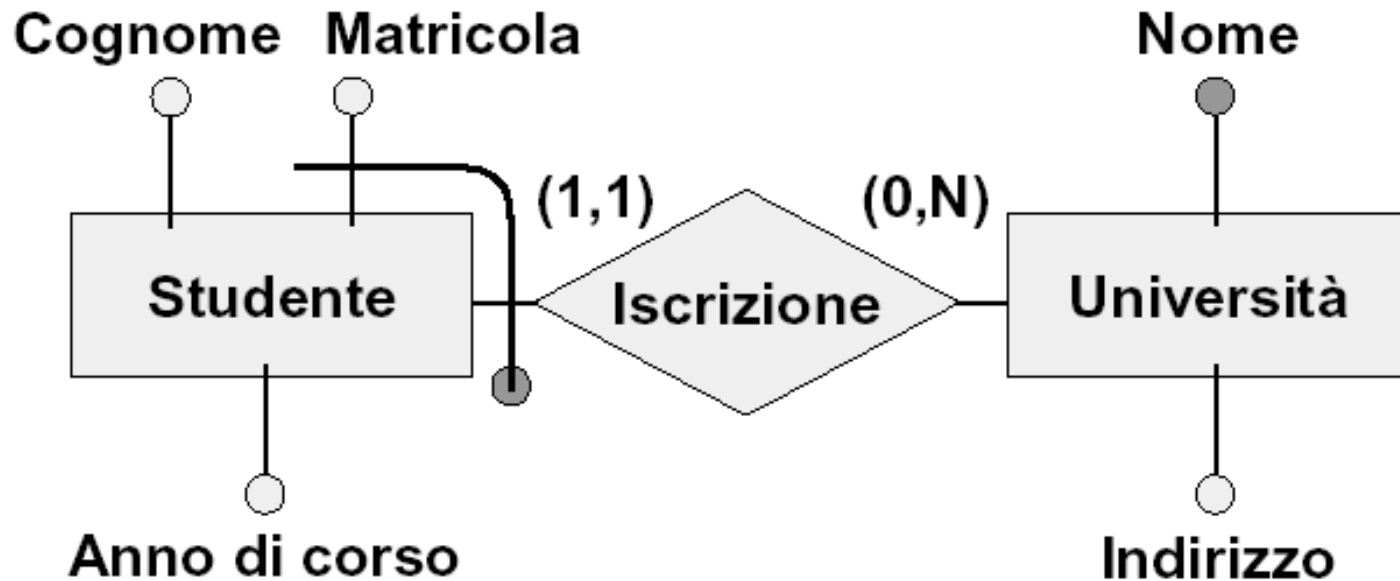
# Identificatori

- Identificatore: strumento per l'identificazione univoca delle occorrenze di un'entità
  - Identificatore interno: costituito da attributi dell'entità
  - Identificatore esterno: attributi + entità esterne attraverso relazione
  - Ogni entità deve possedere almeno un identificatore ma può averne in generale più di uno
  - Una identificazione esterna è possibile solo attraverso una relazione a cui l'entità da identificare partecipa con cardinalità (1,1)

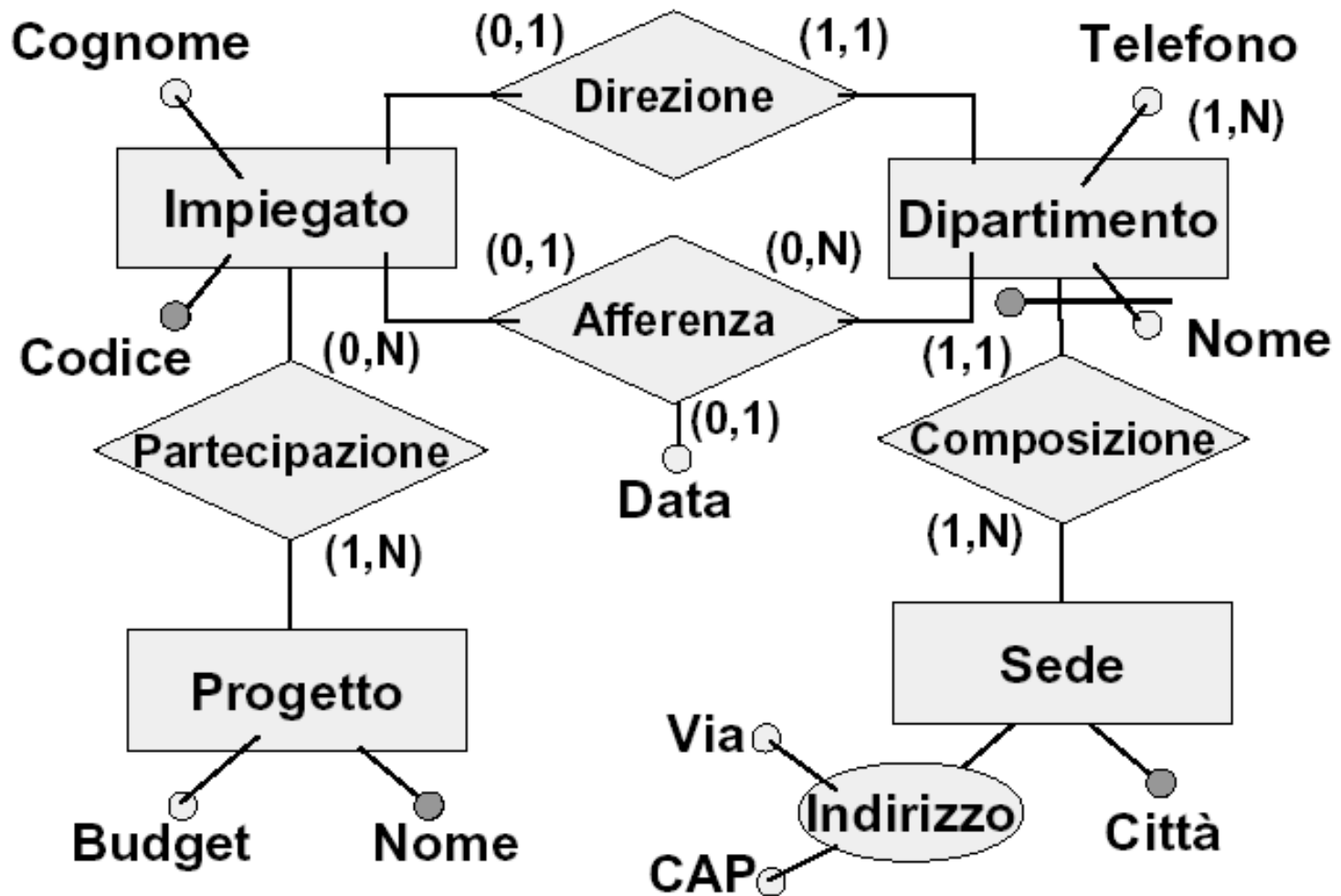
## Identificatori interni



# Identificatori



# Identificatori

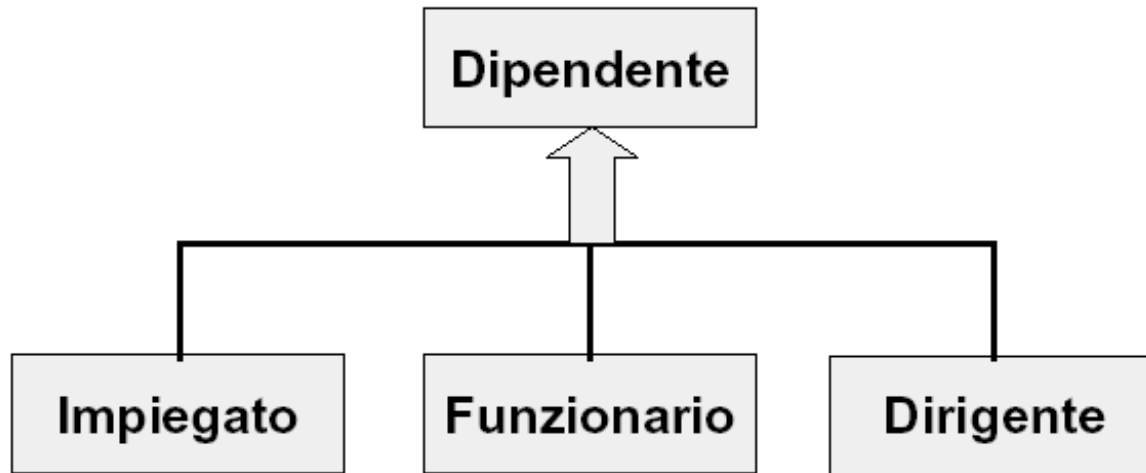




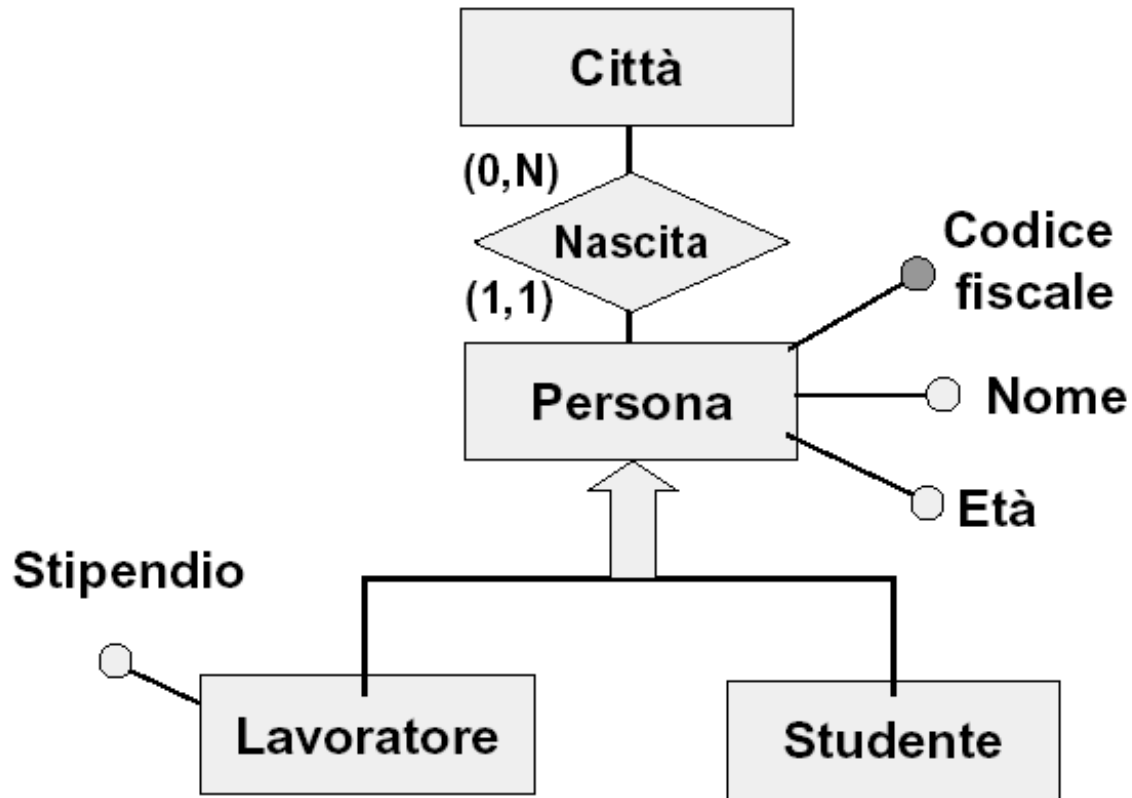
# Generalizzazione

- Mette in relazione una o più entità  $E_1, E_2, \dots, E_n$  con una entità  $E$  che le comprende come casi particolari
  - $E$  è generalizzazione di  $E_1, E_2, \dots, E_n$
  - $E_1, E_2, \dots, E_n$  sono specializzazioni di  $E$
- Se  $E$  (genitore) è generalizzazione di  $E_1, E_2, \dots, E_n$  (figlie)
  - ogni occorrenza di  $E_1, E_2, \dots, E_n$  è occorrenza anche di  $E$
  - Tutte le proprietà (attributi, relazioni, altre generalizzazioni) dell'entità genitore vengono ereditate dalle figlie e non rappresentate esplicitamente.

# Generalizzazione



# Generalizzazione



# Modello E-R

- Traduzione in modello relazionale
  - Obiettivo: tradurre lo schema concettuale in uno schema logico che rappresenti gli stessi dati in maniera corretta ed efficiente
  - Il modello E-R viene semplificato il più possibile con strumenti opportuni
  - Le entità diventano relazioni sugli stessi attributi
  - Le relazioni E-R diventano relazioni sugli identificatori delle entità coinvolte e sui loro stessi attributi

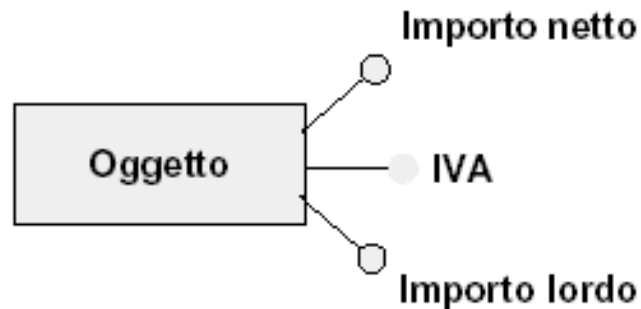
# Modello E-R

- Strumenti di ristrutturazione dello schema concettuale
  - Analisi delle ridondanze (informazione significativa ma derivabile da altre)
  - Eliminazione delle generalizzazioni
  - Partizionamento/accorpamento di entità e relazioni
  - Scelta degli identificatori primari

# Analisi delle ridondanze

- Esempio di ridondanza
  - Bisogna decidere se mantenerle o eliminarle

Attributo derivabile



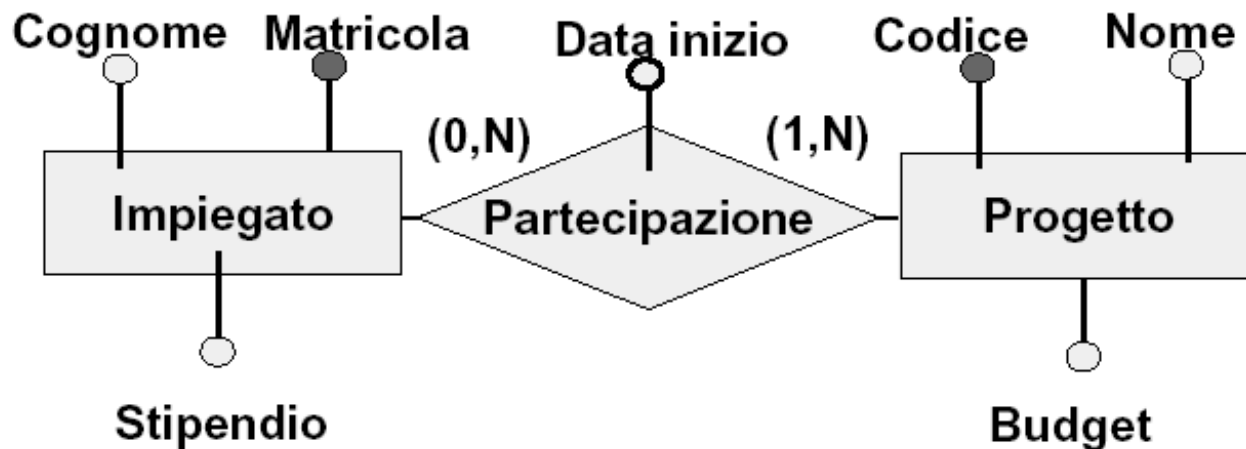
Attributo derivabile da  
altra entità



# Scelta degli attributi primari

- Operazione indispensabile per la traduzione nel modello relazionale (chiave primaria)
  - Criteri: non devono essere opzionali, semplici, utilizzati nelle operazioni più frequenti o importanti
  - Se nessuno degli identificatori soddisfa questi requisiti? Si introducono nuovi attributo (codici) contenenti valori speciali generati appositamente per questo scopo.
- Nella traduzione in modello relazionale
  - Le entità diventano relazioni sugli stessi attributi
  - Le relazioni E-R diventano relazioni sugli identificatori delle entità coinvolte e sui loro stessi attributi
  - **Gli attributi primari diventano la chiave primaria**

# Scelta degli attributi primari



Impiegato(Matricola, Cognome, Stipendio)

Progetto(Codice, Nome, Budget)

Partecipazione(Matricola, Codice, DataInizio)



# Modello E-R

