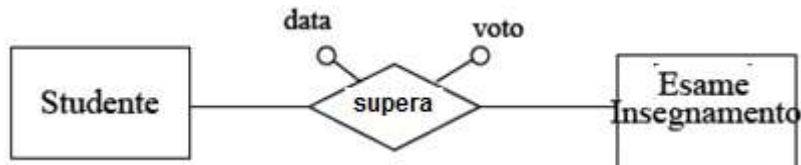


DEA, Dizionario dei dati ed uso di RDBMS tipo desktop

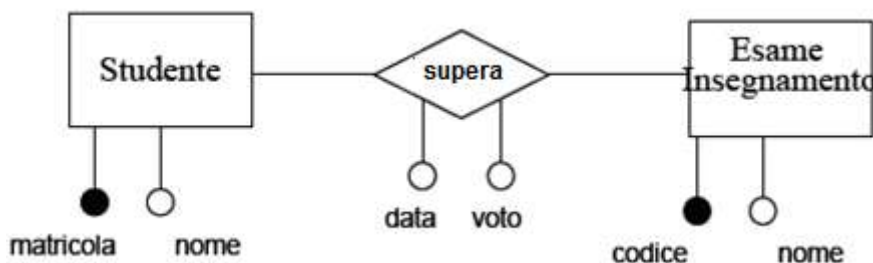
Mini-mondo: superamento di esami universitari (relativi ad insegnamenti diversi di unica facoltà nell'ateneo di Genova) da parte di studenti

Analisi: per dato contesto --> scelta di entità, associazioni, attributi

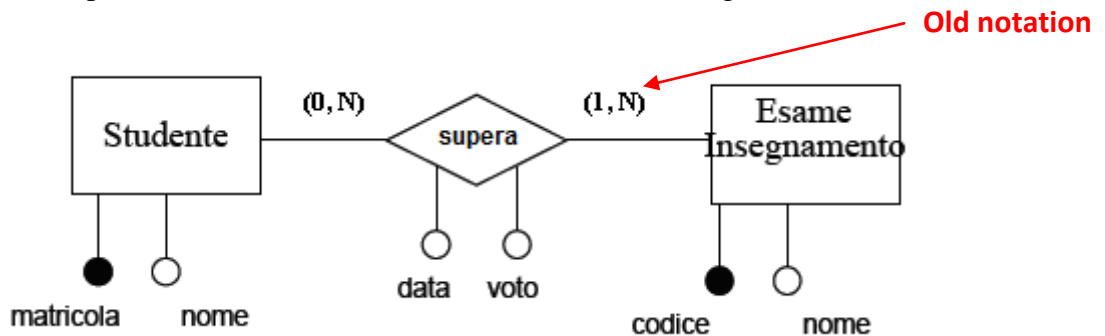


- data e voto non sono proprietà né di uno Studente né dell'Esame dell'insegnamento in un certo corso, ma del legame che si crea in occasione di tale esame

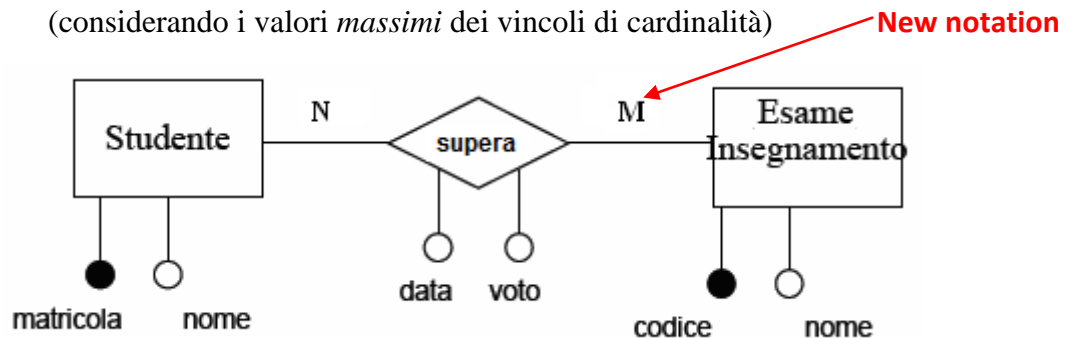
Chiave primaria, vincoli di cardinalità e tipi di associazione



- Ogni studente **può non superare** esami o superarne diversi --> cardinalità *diretta* dell'entità Studente (0, N)
- In una ipotesi realistica, un esame di dato insegnamento viene superato da almeno uno studente o più --> cardinalità *inversa* dell'entità Esame_Insegnamento (1, N)



Il tipo dell'associazione è *molti a molti* cioè N:M
(considerando i valori *massimi* dei vincoli di cardinalità)



Nb: piuttosto che ricorrere all'uso di linee tratteggiate, l'**opzionalità** si esplicherà in documentazione a corredo (dizionario dei dati - [schema delle associazioni](#))

Una **corretta documentazione** illustra nello **schema concettuale** o **DEA** il **tipo dell'associazione** e sintetizza *vincoli di cardinalità*; altri *vincoli non esprimibili* in tale diagramma sono descritti in schemi a corredo detti “**Dizionario dei dati**”.

Dizionario dei dati

Schema delle **entità** :

Nome_Entità	Descrizione	Attributi	Identificatore
Studente	studente universitario di una facoltà dell'ateneo di Genova
Esame_Insegnamento

Schema delle **associazioni**:

Nome_Relazione	Descrizione	Entità_Componenti	Attributi
Supera	Studente (0, N) Esame_Insegnamento (1,N)

Elenco dei vincoli *non esprimibili*

Vincoli di integrità sui dati ed **altri** (le regole dette *Business Rules*): infatti oltre al tipo di conoscenza descritta (*Conoscenza Concreta*), abbiamo nel mondo reale un altro tipo di conoscenza, costituita da condizioni, da regole che pongono dei limiti alla conoscenza concreta.

Tra i vincoli di integrità si deve prevedere, ad esempio, che uno studente possa superare solo **una volta** un esame di un certo insegnamento; tra gli altri vincoli si può, ad esempio, ricordare che uno studente non può superare un esame con voto inferiore a 18, che uno studente che non ha pagato le tasse previste non può superare un esame, che in alcuni casi il superamento di un esame (Analisi I) è prerequisito per poterne superare un altro (Analisi II).

Traduzione di entità e di associazioni in **tabelle** dette **relazioni**

Schema logico

`studente (matricola, nome)`

matricola	nome
0000123456	Mario Rossi
0000654321	Giovanna Bianchi
....

`esame_insegnamento (codice, nome)`

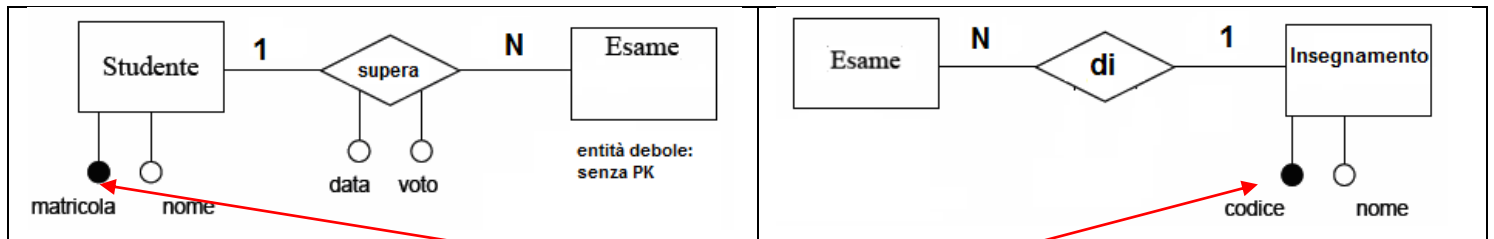
codice	nome
01234	Sistemi Informativi
05432	Informatica
....

`supera (matricola, codice, data, voto)`

matricola	codice	data	voto
0000123456	01234	15-04-2014	30
0000654321	01234	16-04-2014	28
0000654321	05432	14-10-2013	30
....

si è spezzata l'associazione *molti a molti* in due associazioni 1 a N cioè *uno a molti*, traducendo in tre tabelle

Allo stesso schema si poteva giungere anche raffinando l'analisi, nella considerazione di rendere più facile l'inserimento corretto dei vari insegnamenti, ed identificando tre entità:



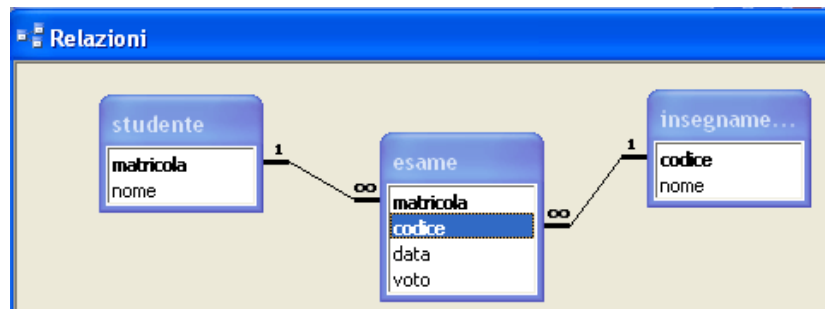
L'entità lato N si traduce in tabella: **esame** (*matricola, codice, data, voto*)

con *matricola* e *codice* FK

Quindi l'entità lato N si traduce in tabella che, tra i vari campi, come *chiave esterna*, contiene la chiave primaria dell'entità lato I

Uso **DBMS Access** o **Base** per implementare (*livello fisico*) salvando in database **università**:

- creando associazioni di tipo *uno a molti* con **integrità referenziale**



Si completi la conoscenza, nell'uso, di DBMS *relazionale* tipo *desktop* verificando gli effetti dell'**integrità referenziale** :

- Nella tabella "esame" impostare la proprietà **duplicati non ammessi** sul campo "matricola" e nella tabella stessa verificare l'impossibilità che uno stesso studente superi più volte un esame di un dato insegnamento
- Nella tabella "esame" verificare l'impossibilità di superamento da parte di uno studente inesistente
- Nella tabella "studente" verificare l'impossibilità di cancellare uno studente
- Nella tabella "insegnamento" verificare l'impossibilità di cancellare un corso
- Catturare come immagine la **struttura** delle tabelle *commentata* da inserire in una relazione per illustrare il DataBase utilizzato
- Impostare le **query** significative catturandone il risultato come immagine da inserire in una relazione con l'indice sotto esemplificato. In particolare si creino le query per selezionare:
 - tutti gli studenti che hanno superato l'esame d'Informatica
 - tutti gli studenti che hanno superato l'esame d'Informatica con voto > 25
 - tutti gli studenti che hanno superato un qualunque esame a marzo
- Realizzare un **Report** che per ogni insegnamento evidenzi gli studenti che lo hanno superato in ordine alfabetico con il relativo voto e la data di superamento e con titolo uguale al nome della facoltà ad esempio come in figura:

Ingegneria

<i>insegnamento_nome</i>	<i>studente_nome</i>	<i>data</i>	<i>voto</i>
<i>Informatica</i>	Mario Rossi	14/10/2013	30
<i>Sistemi Informativi</i>	Giovanna Bianchi	16/04/2014	28
	Mario Rossi	15/04/2014	30

- **Esportare** le tre tabelle salvandole come fogli di calcolo potendo meglio illustrare le potenzialità del DBMS utilizzato
- **Consegnare** il database realizzato, comprimendolo in unica cartella con la relazione (con indice tipo sottoelencato) e i fogli di calcolo

Indice

1. Analisi

Problema generale:

Obiettivi realmente raggiunti:

2. Schema E/R

3. Dizionario dei dati

4. Schema logico

5. Scelta implementativa

6. DataBase utilizzato

7. Query significative