Progettazione di basi di dati

Preliminari

- Progettare una base di dati: definirne il contenuto e la struttura che esso deve avere
- metodologie di progettazione: le basi di dati sono sempre più complesse e sofisticate $\Rightarrow$ è necessario un approccio sistematico
- obbiettivo della progettazione è produrre i seguenti risultati:
  - uno schema dei dati
  - un insieme di sottoschemi di applicazione
  - un insieme di (specifiche di) programmi applicativi

Preliminari

- Fasi della progettazione:
  - raccolta ed analisi dei requisiti
  - progettazione concettuale
  - progettazione logica
  - progettazione fisica
- ogni fase si basa su un **modello**, che permette di generare una rappresentazione formale della base di dati ad un dato livello di astrazione
- utilizzo di modello appropriato permette di
  - verificare le caratteristiche della base di dati
  - comunicare con i futuri utenti della base di dati

Fasi della progettazione

Raccolta ed analisi dei requisiti

- è la fase in cui si raccolgono le specifiche informali ed eterogenee che i vari utenti danno delle procedure da automatizzare mediante un DBMS
- requisiti informativi: caratteristiche dei dati
- requisiti sui processi: operazioni sui dati
- requisiti sui vincoli di integrità: proprietà dei dati e delle operazioni
- disambiguazione delle specifiche dell'utente

Fasi della progettazione

Progettazione concettuale

- a partire dai requisiti informativi viene creato uno *schema concettuale*, cioè una descrizione formalizzata e integrata delle esigenze aziendali, espressa in modo indipendente dal DBMS adottato
- *modello concettuale*: descrizione ad alto livello indipendente dall'implementazione
- prima rappresentazione formale e del tutto indipendente dall'implementazione della base di dati (indipendente anche dal *tipo* di DBMS che sarà utilizzato - relazionale, object-relational, gerarchico, …)

Fasi della progettazione

Progettazione logica

- traduzione dello schema concettuale nel modello dei dati del DBMS
- risultato: *schema logico* nel DDL del DBMS
- aspetti considerati durante la progettazione logica:
  - integrità e consistenza
  - sicurezza
  - efficienza
- sottofasi:
  - ristrutturazione dello schema concettuale
  - traduzione canonica
  - valutazione ed eventuale modifica dello schema

Fasi della progettazione

Progettazione fisica

- in questa fase vengono scelte le caratteristiche fisiche di realizzazione della base di date
- risultato: *schema fisico* che descrive le strutture di memorizzazione e accesso ai dati
  (es. clustering, indici, …)

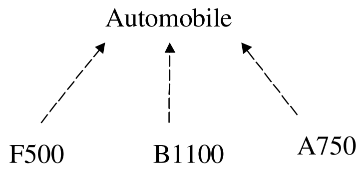
Modelli concettuali

- Concetto di astrazione: procedimento mentale che si adotta quando si concentra l'attenzione su alcune caratteristiche di un insieme di entità, trascurando le altre giudicate non rilevanti
- quattro tipi:
  - astrazione di classificazione
  - astrazione di aggregazione
  - astrazione di generalizzazione
  - astrazione di associazione

Modelli concettuali

Astrazione di classificazione

- definizione di una classe a partire da un insieme di oggetti aventi proprietà comuni



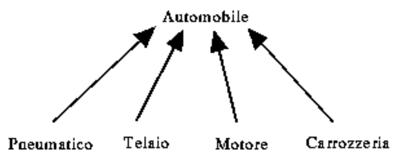
- F500 è un' istanza della classe Automobile
- relazione di *instance_of*

9

Modelli concettuali

Astrazione di aggregazione

- definizione di una classe a partire da un insieme di classi che costituiscono le sue componenti



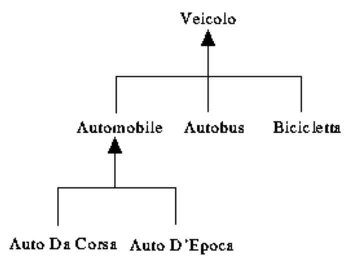
- il telaio è una parte dell'automobile
- relazione di *part_of*

10

Modelli concettuali

Astrazione di generalizzazione

- definizione di una classe (superclasse) a partire da un insieme di classi aventi proprietà comuni (sottoclassi)



11

Modelli concettuali

Astrazione di generalizzazione

- le istanze di Automobile sono un sottoinsieme delle istanze di Veicolo
- equivalentemente, ogni automobile è un veicolo
- relazione di *subset_of* o *is_a*

- si noti che le sottoclassi possono possedere caratteristiche proprie

12

Modelli concettuali

Astrazione di associazione

- definizione di un collegamento fra due o più classi

- le associazioni possono avere proprietà
- esempio: data (in cui si è presa la residenza) può essere una proprietà dell'associazione Risiede

13

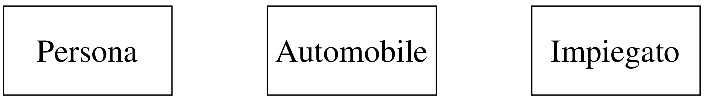
Modello Entity-Relationship

- Uno dei modelli più utilizzati nell'ambito della progettazione concettuale
- ha rappresentazione grafica: diagramma ER
- concetti fondamentali:
  - entità (istanze di entità)
  - associazioni (istanze di associazione)
  - attributi

14

Modello Entity-Relationship

- **Entità**: insieme di oggetti della realtà che possiedono caratteristiche comuni (es. persone, automobili, …)
- **Istanze di entità**: oggetti appartenenti a una certa entità (es. io, la mia auto, …)
- graficamente:



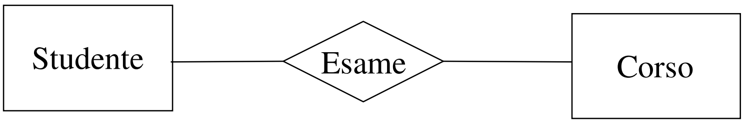
Modello Entity-Relationship

- **Associazione**: legame logico tra entità
- **Istanze di associazione**: combinazione delle istanze delle entità che prendono parte ad una associazione
- graficamente:

- p istanza di Persona, c istanza di Città
- (p, c) istanza di Risiede

Modello Entity-Relationship

- si noti che l'insieme delle istanze di un'associazione è un sottoinsieme del prodotto Cartesiano degli insiemi delle istanze di entità che partecipano all'associazione
- quindi le stesse istanze di entità non possono partecipare più volte alla stessa associazione
- esempio



uno studente s può sostenere un'unica volta l'esame del corso c, perché (s,c) può comparire un'unica volta nell'insieme delle istanze di Esame

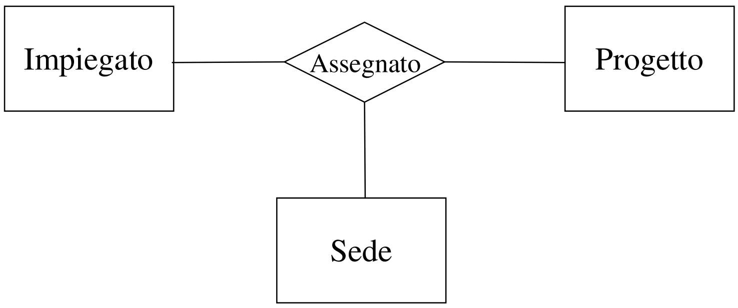
Modello Entity-Relationship

- **Grado**: numero di classi che partecipano ad un'associazione
- associazione unaria: grado 1

- associazione binaria: grado 2

Modello Entity-Relationship

- associazione n-aria: grado n > 2

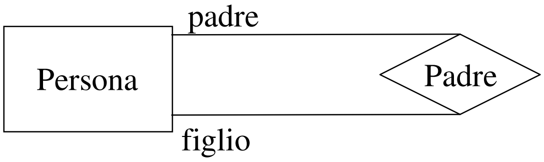


19

Progettazione di basi di dati

Modello Entity-Relationship

- **Ruolo**: funzione che un'istanza di entità esercita nell'ambito di un'associazione

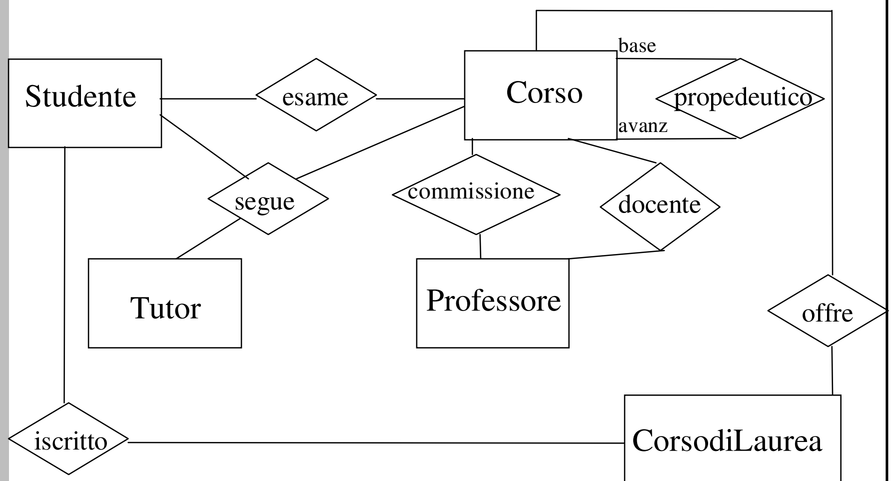


- nel caso di associazione unaria il ruolo è obbligatorio

20

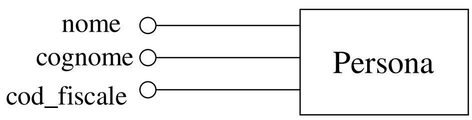
Progettazione di basi di dati

Modello Entity-Relationship



Modello Entity-Relationship

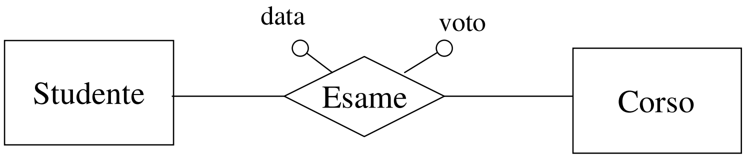
- **Attributo**: proprietà elementare posseduta da un'entità o da un'associazione
- graficamente:



- nome, cognome, cod_fiscale sono attributi di Persona

Modello Entity-Relationship

- anche le associazioni possono avere attributi
- esempio:



data e voto non sono proprietà né di uno Studente né di un Corso, ma del legame Studente-Corso che si crea in occasione di un esame

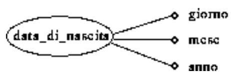
- gli attributi possono essere visti come funzioni che associano un valore ad un'istanza di entità o associazione

Modello Entity-Relationship

- **Dominio di un attributo:** insieme dei valori legali per l'attributo
- domini possibili:
  - interi, reali, booleani, caratteri
  - intervalli di interi e di caratteri
  - stringhe di caratteri
  - domini definiti dall'utente
- notazione:
  - v_i, v_j intervallo compreso fra v_i e v_j
  - (v_i, ... v_j) insieme di valori possibili v_i, ..., v_j

Modello Entity-Relationship

- **Attributo composito:** possiede dei sottoattributi
- es. data_nascita con sottoattributi giorno, mese, anno



- i domini si possono distinguere in:
  - semplici: domini degli attributi non compositi
  - compositi: domini degli attributi compositi ovvero prodotto Cartesiano degli insiemi di valori associati ai domini componenti

    se $D = D_1 \times D_2 \times \ldots \times D_n$ allora $< d_1, ..., d_n >$ t.c. $d_i \in D_i$ è valore possibile

25

Modello Entity-Relationship

- Esempio: si consideri l'entità Persona, i cui attributi e relativi domini sono:

  nome: stringa(20)

  cognome: stringa(20)

  cod_fiscale: stringa(16)

  data_di_nascita: giorno $\times$ mese $\times$ anno

  titolo_di_studio: stringa(50)

  dove i domini giorno, mese, ed anno sono:

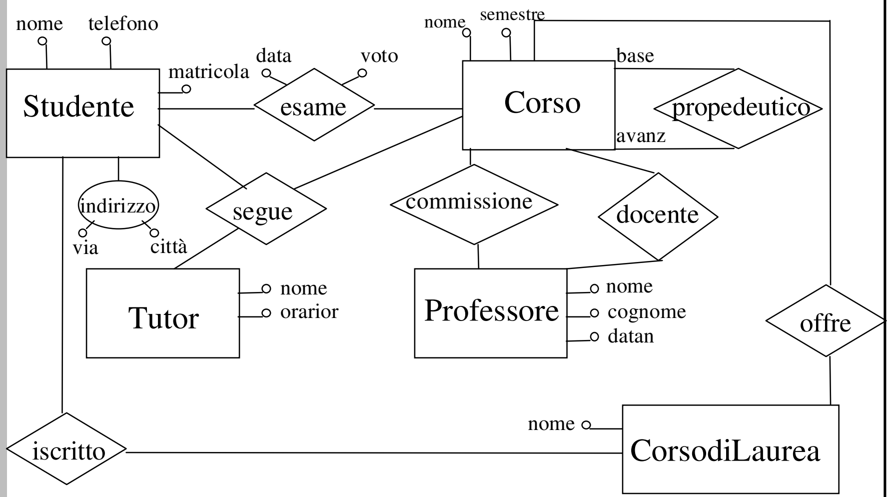
  giorno = 1, ..., 31

  mese = {Gen,Feb,Mar,Apr,Mag,Giu,Lug,Ago,Set,Ott,Nov,Dic}

  anno = 1900, ..., 2100

26

Modello Entity-Relationship



Modello Entity-Relationship

Vincoli di integrità

- **impliciti**: automaticamente verificati dal sistema
  ogni occorrenza di una base di dati relativa ad uno schema ER li deve verificare
- **espliciti**: definiti esplicitamente da chi progetta lo schema ER
  - vincoli di cardinalità (per associazioni e attributi)
  - vincoli di identificazione
- anche i domini degli attributi sono dei vincoli di integrità

Modello Entity-Relationship

Vincoli impliciti:

- ogni istanza di associazione deve riferirsi ad istanze di entità presenti nell'occorrenza della base di dati
- istanze diverse della stessa associazione devono riferirsi a differenti combinazioni di istanze delle entità partecipanti all'associazione

29

Modello Entity-Relationship

Vincoli espliciti di cardinalità - associazioni

- numero minimo e massimo di istanze dell'associazione a cui un'istanza dell'entità può partecipare
- valori più comuni:
  - cardinalità minima (c_min): 0, 1
  - cardinalità massima (c_max): n, ovvero qualunque intero > 1
- data una classe C e un'associazione A:
  - c_min=0 $\Rightarrow$ esistono istanze di C che non partecipano ad alcuna istanza di A
  - c_min=1 $\Rightarrow$ ogni istanza di C partecipa almeno ad una istanza di A

30

Modello Entity-Relationship

- data un'entità E e un'associazione A:
  - c_max=1 ⇒ ogni istanza di E può partecipare al più ad una istanza di A
  - c_max=n ⇒ non esiste limite al numero massimo di istanze di A a cui ogni istanza di E può partecipare
  - c_max=c_min=1 ⇒ ogni istanza di E partecipa ad una ed una sola istanza di A
  - c_min=0, c_max=n ⇒ ogni istanza di E può partecipare ad un numero qualsiasi di istanze di A
- nei diagrammi si può indicare la coppia (c_min,c_max) sulla linea che congiunge E ad A
- se non si indica niente il valore di default è (0,n)

Modello Entity-Relationship

- esempio

  - c_min di Automobile rispetto a Proprietario è 0: esistono automobili non possedute da alcuna persona
  - c_min di Persona rispetto a Proprietario è 0: esistono persone che non posseggono alcuna automobile
  - c_max di Persona rispetto a Proprietario è n: ogni persona può essere proprietaria di un numero arbitrario di automobili
  - c_max di Automobile rispetto a Proprietario è 1: ogni automobile può avere al più un proprietario

Modello Entity-Relationship

- Terminologia: nel caso di un'associazione binaria A tra due entità E_1 ed E_2 (o unaria con $E_1 = E_2$)
  - A si dice associazione **uno a uno** se c_max di E_1 ed E_2 rispetto ad A è 1
  - A si dice associazione **uno a molti** se c_max di E_1 rispetto ad A è n e c_max di E_2 rispetto ad A è 1
  - A si dice associazione **molti a uno** se c_max di E_1 rispetto ad A è 1 e c_max di E_2 rispetto ad A è n
  - A si dice associazione **molti a molti** se c_max di E_1 ed E_2 rispetto ad A è n
- inoltre un'associazione è detta **opzionale** se c_min è 0, è detta **obbligatoria** se c_min è 1

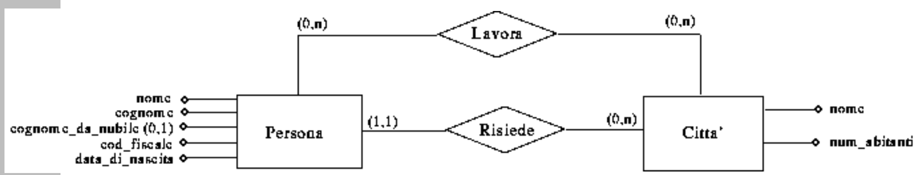
Modello Entity-Relationship

Vincoli espliciti di cardinalità - attributi

- numero minimo e massimo di valori dell'attributo che possono essere associati ad un'istanza della corrispondente associazione od entità
- nei diagrammi si può indicare la coppia (c_min,c_max) sulla linea che congiunge l'attributo all'associazione/ entità
- se non si indica niente il valore di default è (1,1)

Modello Entity-Relationship

- Si parla di attributi:
  - **opzionali**: se la cardinalità minima è 0 (es. cognome_da_nubile)
  - **monovalore**: se la cardinalità massima è 1 (es. cod_fiscale)
  - **multivalore**: se la cardinalità massima è n (es. telefono)
- esempio di diagramma con vincoli di cardinalità



35

Progettazione di basi di dati

Modello Entity-Relationship

Vincoli espliciti di identificazione

- identificatori o chiavi: insieme di attributi che identificano univocamente le istanze dell'entità
- devono essere minimali: qualsiasi sottoinsieme proprio non è un identificatore
- si noti che gli identificatori hanno senso solo per le entità e non per le associazioni
- nell'insieme di istanze di un'associazione si hanno tutte tuple distinte ⇒ non c'è bisogno di identificatori

36

Progettazione di basi di dati

Modello Entity-Relationship

- a volte non è possibile identificare un'istanza di entità solo sulla base dei suoi attributi, cioè due istanze diverse possono coincidere su tutti gli attributi
- si utilizza allora il fatto che tale istanza partecipi ad una particolare istanza di associazione con una data istanza di un'altra entità
- l'entità viene detta **debole**

37

Progettazione di basi di dati

Modello Entity-Relationship

- identificatori o chiavi:
  - interni: uno o più attributi dell'entità
  - esterni: uno o più associazioni collegate all'entità a cui si riferiscono

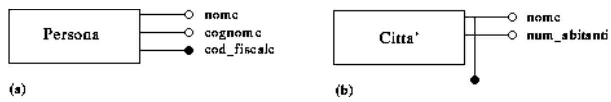
    (identificazione esterna **da** tale entità **attraverso** tale associazione)
  - misti: attributi o associazioni

  - semplici: un elemento
  - compositi: più di un elemento

38

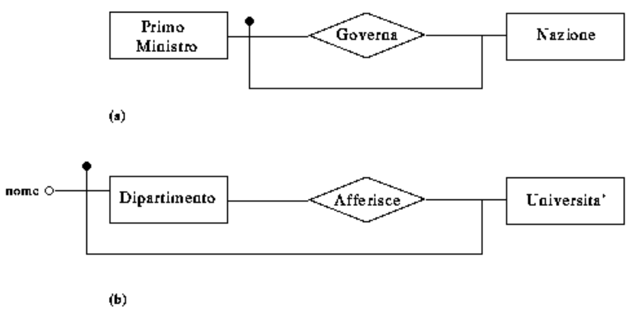
Progettazione di basi di dati

Modello Entity-Relationship



- (a) identificatore interno semplice
- (b) identificatore interno composito

Modello Entity-Relationship

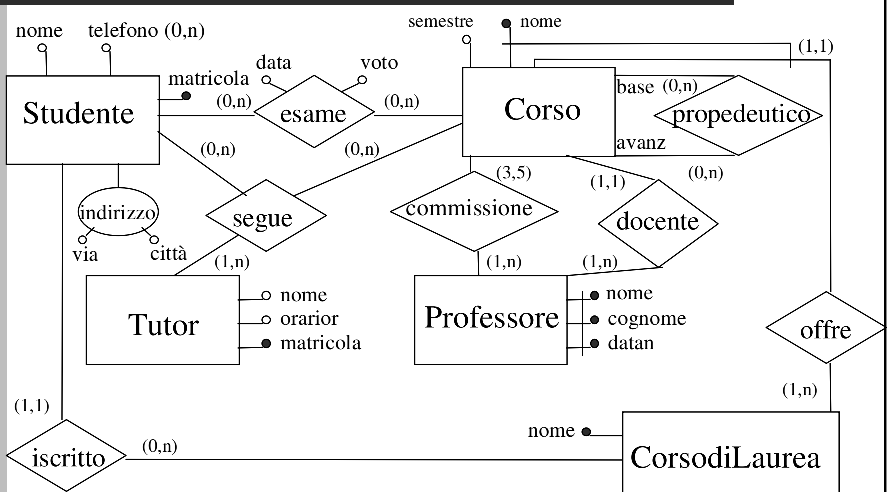


- (a) identificatore esterno
- (b) identificatore misto

Modello Entity-Relationship

- le entità deboli hanno sempre cardinalità (1,1) rispetto all'associazione attraverso cui avviene l'identificazione
  - nel caso di identificazione esterna l'associazione sarà uno a uno
  - nel caso di identificazione mista l'associazione sarà uno a molti
- a volte viene evidenziato graficamente che un'entità è debole evidenziando il rispettivo rettangolo con una doppia linea
- è possibile che un'entità abbia più chiavi, a livello concettuale va bene indicarle tutte, nel passaggio allo schema logico bisognerà sceglierne una

Modello Entity-Relationship

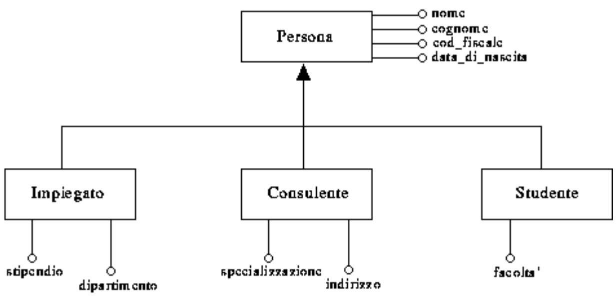


Modello Entity-Relationship

- Costrutti di base:
  entità, associazione, attributo
- costrutto ulteriore
  (non presente nella proposta originaria [Chen 1976])
  gerarchie di generalizzazione
- una entità E è una generalizzazione delle entità E_1, …, E_n se ogni istanza delle entità E_1, …, E_n è anche un'istanza di E
  - E entità padre
  - E_1, …, E_n entità figlie

Modello Entity-Relationship

- Esempio



Modello Entity-Relationship

- Vincoli impliciti
- se una entità E_1 è definita come generalizzazione di una entità E_2:
  - l'insieme delle istanze di E_2 deve essere contenuto in quello delle istanze di E_1
  - ogni attributo di E_1 è anche un attributo di E_2
  - ad ogni associazione cui partecipa E_1 partecipa anche E_2

Modello Entity-Relationship

- ogni generalizzazione può essere:
  - **totale**: ogni istanza di E è istanza di almeno un'entità E_i
    es.: Persona - Uomo, Donna
  - **parziale**: esiste almeno un'istanza di E che non è istanza di alcuna entità E_i
    es.: Persona - Studente, Impiegato

Modello Entity-Relationship

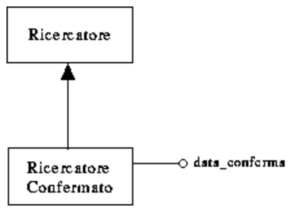
- ogni generalizzazione può essere inoltre
  - **esclusiva**: ogni istanza di E è istanza di al più un'entità E_i

    es.: Persona - Uomo, Donna
  - **condivisa**: esiste almeno un'istanza di E che è istanza di più di un'entità E_i

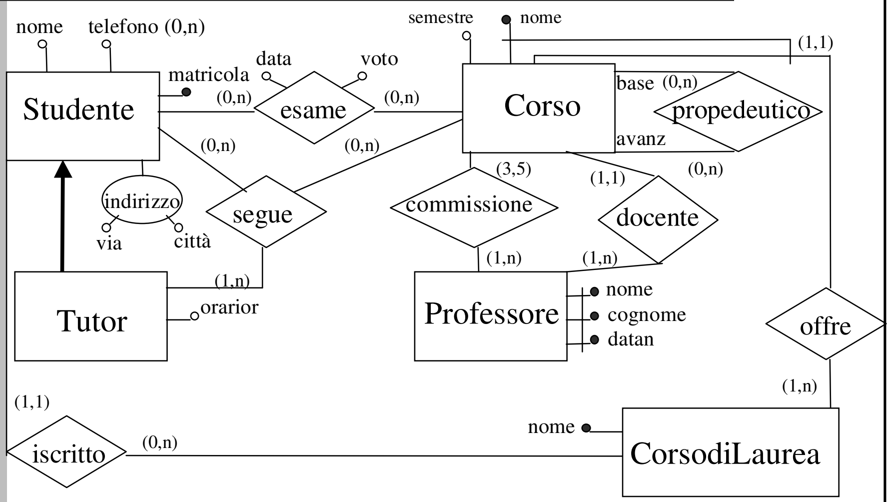
    es.: Persona - Studente, Impiegato
- tali caratteristiche possono essere indicate come vincoli espliciti della gerarchia di generalizzazione

Modello Entity-Relationship

- Caso particolare di generalizzazione (parziale ed esclusiva): relazione di sottoinsieme
- definire una relazione di sottoinsieme tra una entità E_1 ed una entità E_2 significa specificare che ogni istanza di E_1 è anche istanza di E_2
- esempio



Modello Entity-Relationship

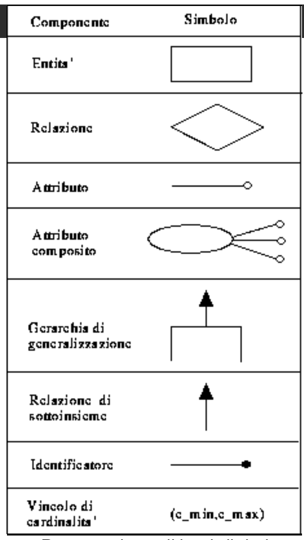


Modello Entity-Relationship

Riassumendo:

- Astrazione di classificazione: entità, associazioni, attributi
- Astrazione di aggregazione:
  - entità come aggregazione di attributi
  - associazione come aggregazione di attributi ed entità
  - attributo composito come aggregazione di sottoattributi
- Astrazione di generalizzazione: gerarchie di generalizzazione tra entità, associazioni e attributi
- Astrazione di associazione: associazione

Modello Entity-Relationship



51

Modello Entity-Relationship

- uno schema ER non è sufficiente, da solo, a rappresentare tutti gli aspetti di un'applicazione
- cosa manca?
  - i nomi dei vari concetti possono non essere sufficienti per comprenderne il significato
  - non tutti i vincoli di integrità sono esprimibili in un diagramma ER - esempi:
    - ogni studente ha al più un tutor per ogni corso
    - uno studente non può essere tutor di se stesso
    - per sostenere un esame è necessario avere sostenuto tutti gli esami propedeutici
- necessità di documentazione di supporto

52

Progettazione logica

- Due fasi principali:
  - fase di ristrutturazione
  - fase di traduzione
- **ristrutturazione**: eliminazione dallo schema ER di tutti i costrutti che non possono essere direttamente rappresentati nel modello relazionale
  - eliminazione degli identificatori esterni
  - eliminazione degli attributi compositi e multivalore
  - eliminazione delle gerarchie di generalizzazione
- risultato: schema ER ristrutturato

53

Progettazione logica

- **traduzione**: traduzione con regole di trasformazione di entità, attributi e associazioni dello schema ER in relazioni del modello relazionale:

Schema ER	Schema relazionale
entità attributi associazioni	relazioni

- risultato: schema relazionale

54

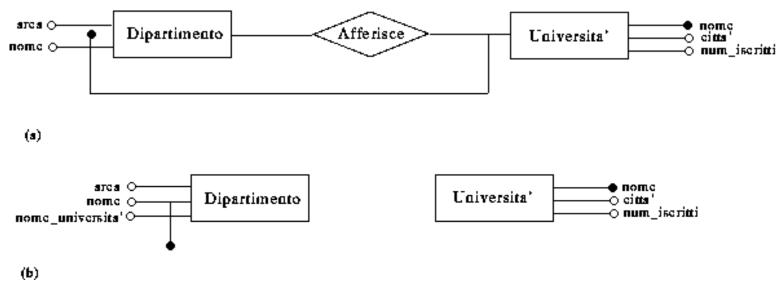
Fase di ristrutturazione

Eliminazione degli identificatori esterni

- E_1 ha identificatore (chiave) misto od esterno da E_2 attraverso A
- distinguiamo due casi:
  (a) E_2 ha un identificatore interno
  (b) E_2 ha da un identificatore misto od esterno
- caso (a):
  - l'identificatore di E_1è trasformato in un identificatore interno aggiungendo agli attributi di E_1 l'identificatore interno di E_2
  - l'associazione A può essere eliminata

Fase di ristrutturazione

- Esempio



Fase di ristrutturazione

- caso (b): E_2 è a sua volta caratterizzata da un identificatore esterno o misto da E_3
- due casi ulteriori:
  - E_3 ha un identificatore interno ⇒ l'eliminazione dell'identificatore esterno di E_1 avviene come segue:
    - trasformazione dell'identificatore di E_2 in un equivalente identificatore interno
    - trasformazione dell'identificatore di E_1 in un equivalente identificatore interno

    i passi precedenti sono eseguiti con la procedura del caso (a)
  - E_3 ha un identificatore esterno o misto ⇒ si applica ricorsivamente il passo precedente per E_3

57

Progettazione di basi di dati

Fase di ristrutturazione

Eliminazione degli attributi compositi e multivalore

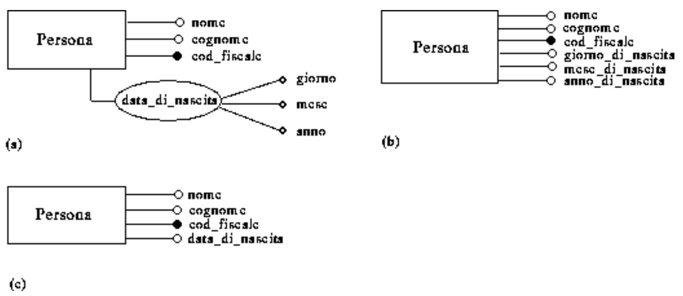
- Il modello relazionale consente solo la specifica di attributi semplici e monovalore
- **attributi compositi** si può procedere in due modi (ricorsivamente):
  - si considerano tutti i sottoattributi come attributi dell'entità, oppure
  - si eliminano i sottoattributi e si considera l'attributo composito come un attributo semplice (ridefinizione del dominio)

58

Progettazione di basi di dati

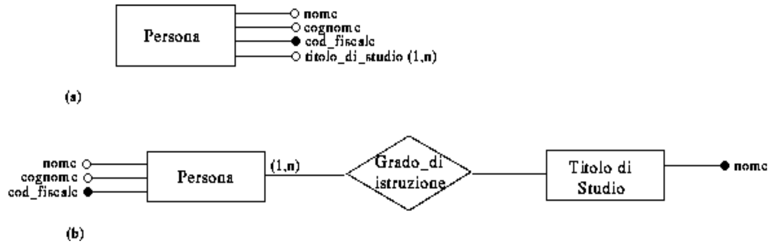
Fase di ristrutturazione

- Esempio



Fase di ristrutturazione

- **attributi multivalore** si definisce una nuova entità, collegata all'entità di partenza con un'associazione, che modella l'attributo multivalore mediante un attributo a valore singolo
- l'associazione introdotta sarà ovviamente uno a molti



Fase di ristrutturazione

Eliminazione delle gerarchie di generalizzazione

- il modello relazionale non prevede gerarchie di generalizzazione
- consideriamo E generalizzazione di E_1, ..., E_n
- tre alternative:

  (a) eliminazione delle entità figlie

  (b) eliminazione dell'entità padre

  (c) sostituzione della generalizzazione con associazioni

61

Progettazione di basi di dati

Fase di ristrutturazione

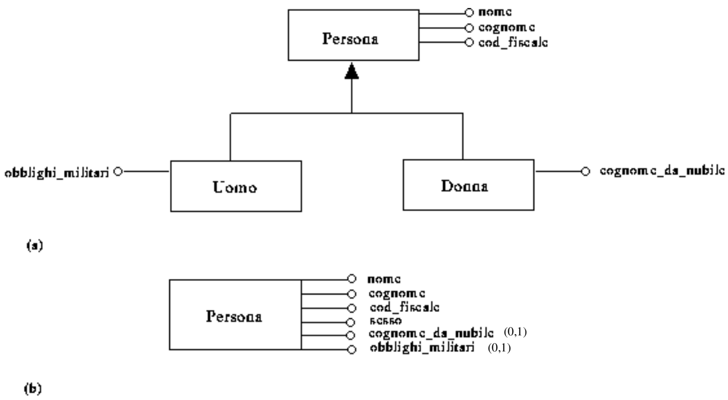
- Caso (a): eliminazione delle entità figlie
  - E_1, ..., E_n vengono eliminate e i loro attributi vengono inseriti in E
  - ad E viene aggiunto un attributo *a* per tenere traccia delle entità figlie
- generalizzazioni totali ⇒ *a* mai nullo
- generalizzazioni parziali ⇒ *a* nullo per istanza dell'entità padre che non appartiene a nessuna entità figlia

62

Progettazione di basi di dati

Fase di ristrutturazione

- Esempio

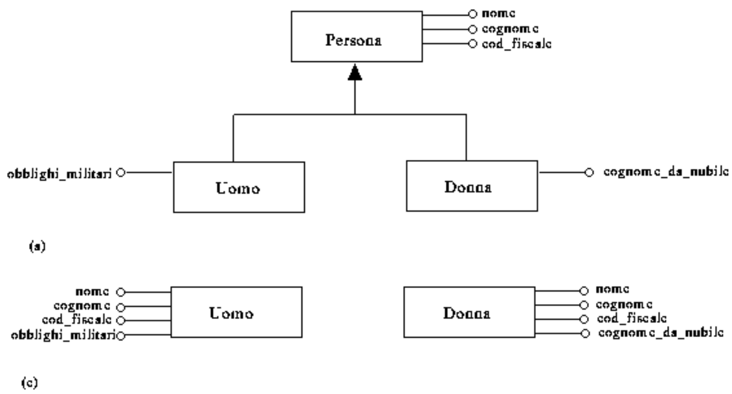


Fase di ristrutturazione

- Caso (b): eliminazione dell'entità padre
  - E viene eliminata e i suoi attributi vengono inseriti in E_1, ..., E_n
  - si può applicare solo nel caso di generalizzazioni totali

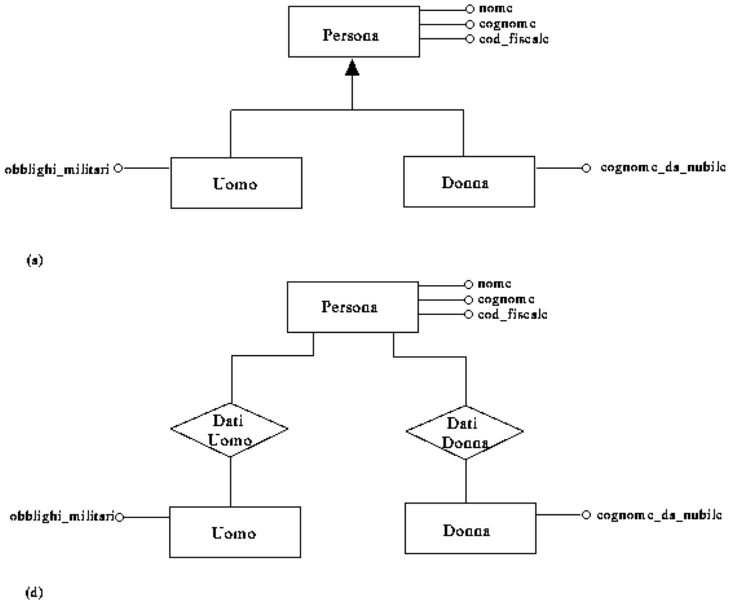
Fase di ristrutturazione

- esempio



Fase di ristrutturazione

- Caso (c): sostituzione della generalizzazione con associazioni
  - E, E_1, ..., E_n rimangono invariate
  - la gerarchia di generalizzazione è sostituita da un insieme di associazioni uno a uno, ognuna delle quali lega l'entità padre con una diversa entità figlia



67

Fase di ristrutturazione

Quale scegliere?

- (a) eliminazione delle entità figlie
  - spreco di memoria per la presenza di valori nulli
  conviene se le operazioni non fanno distinzione fra le istanze delle varie entità
- (b) eliminazione dell'entità padre
  + risparmio di memoria
  - applicabile solo per generalizzazioni totali
  conviene se esistono operazioni che si riferiscono ad istanze di una particolare entità figlia

68

Fase di ristrutturazione

Quale scegliere?

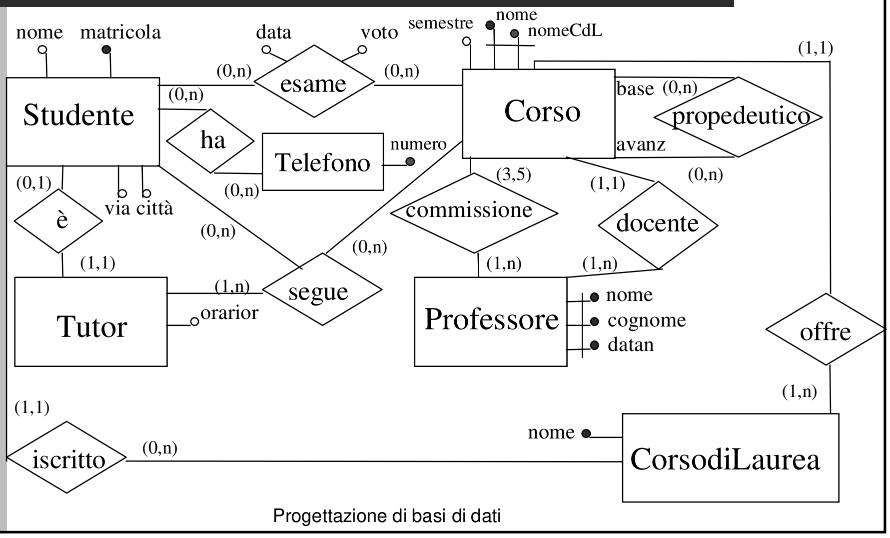
- (c) sostituzione della generalizzazione con associazioni

  + risparmio di memoria

  - incremento del numero degli accessi (anche se tuple di dimensione minore)

  conviene se esistono operazioni che si riferiscono alternativamente a entità padre o figlie
- sono possibili anche combinazioni delle tre trasformazioni presentate

Fase di ristrutturazione



Fase di traduzione

Entità

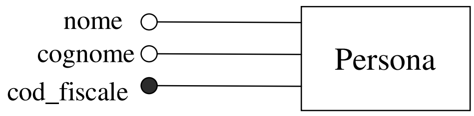
- Per ogni entità si genera una relazione che ha un attributo per ogni attributo dell'entità

entità ⇒ relazione

attributo di entità ⇒ attributo di relazione

identificatore di entità ⇒ chiave di relazione

- esempio:



Persona(nome, cognome, <u>cod_fiscale</u>)

Fase di traduzione

Associazioni

- La traduzione delle associazioni dipende da:
  - grado (numero di entità partecipanti)
  - vincoli di cardinalità
- due alternative:
  - l'associazione viene rappresentata inserendo opportuni attributi (chiavi esterne) in una delle relazioni rappresentanti le entità partecipanti
  - l'associazione stessa viene modellata con una relazione
- vediamo per ogni tipo di associazione

Fase di traduzione

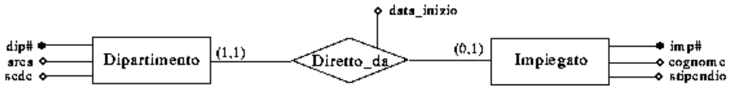
Associazione binaria uno a uno

- L'associazione viene modellata mediante attributi inseriti nelle relazioni che modellano le entità partecipanti
- due casi:
  - (a) partecipazione obbligatoria di una sola entità
  - (b) partecipazione opzionale od obbligatoria di entrambe le entità

73

Fase di traduzione

- caso (a): la relazione che rappresenta l'entità per cui l'associazione è obbligatoria contiene come chiave esterna la chiave della relazione che rappresenta l'altra entità e come attributi gli attributi dell'associazione
- esempio



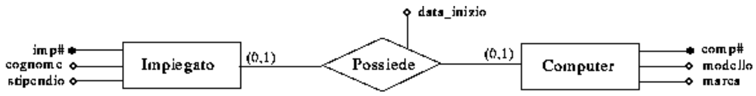
Dipartimento(<u>Dip#</u> ,Area,Sede,Imp#,Data_inizio)

Impiegato(<u>Imp#</u>,Cognome,Stipendio)

74

Fase di traduzione

- caso (b): come il caso (a), ma la relazione può essere scelta indistintamente
- esempio



Impiegato(Imp#,Cognome,Stipendio,Comp#,Data_inizio)
Computer(Comp#,Modello,Marca)

oppure

Impiegato(Imp#,Cognome,Stipendio)
Computer(Comp#,Modello,Marca,Imp#,Data_inizio)

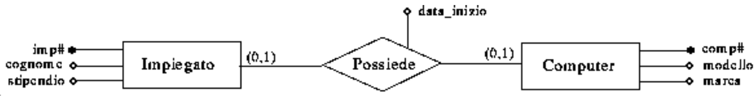
Fase di traduzione

- nel caso particolare di partecipazione opzionale di entrambe le entità si può decidere di introdurre una relazione nuova per modellare l'associazione
- tale relazione contiene:
  - le chiavi delle entità partecipanti
  - gli attributi dell'associazione

- vantaggio: mai valori nulli
- svantaggio: una relazione in più

Fase di traduzione

- esempio



Impiegato(<u>Imp#</u>,Cognome,Stipendio)

Computer(<u>Comp#</u>,Modello,Marca)

Possiede(Imp#,Comp#,Data_inizio)

con chiave indifferentemente Imp# o Comp#

77

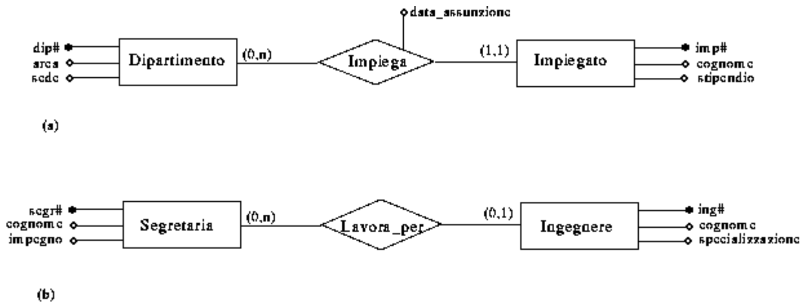
Fase di traduzione

Associazione binaria uno a molti

- Si inseriscono nella relazione dell'entità dal lato uno
  - la chiave della relazione corrispondente all'entità dal lato n come chiave esterna
  - gli attributi dell'associazione come attributi

78

Fase di traduzione



- (a) Dipartimento (Dip#,Area,Sede)
  Impiegato (Imp#,Cognome,Stipendio,Dip#,Data_assunzione)
- (b) Segretaria(Segr#,Cognome,Impegno)
  Ingegnere(Ing#,Cognome,Specializzazione,Segr#)

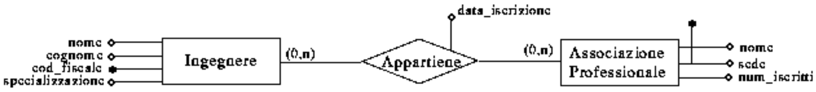
Fase di traduzione

- nel caso particolare di partecipazione opzionale dell'entità dal lato uno si può decidere di introdurre una relazione nuova per modellare l'associazione
- tale relazione contiene:
  - le chiavi delle entità partecipanti
  - gli attributi dell'associazione
- esempio (b)
  Segretaria(Segr#,Cognome,Impegno)
  Ingegnere(Ing#,Cognome,Specializzazione)
  Lavora_Per(Ing#,Segr#)

Fase di traduzione

Associazione binaria molti a molti

- Nuova relazione con attributi le chiavi di entrambe le entità che partecipano all'associazione (chiavi esterne) e gli attributi dell'associazione
- esempio



Ingegnere(Nome,Cognome,<u>Cod_fiscale</u>,Specializzazione)

Associazione_Professionale(<u>Nome</u>, <u>Sede</u>, Num_iscritti)

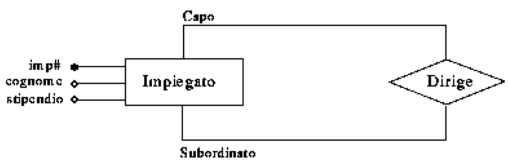
Appartiene(<u>NomeA</u>,<u>SedeA</u>,<u>Cod_fiscaleI</u>,Data_iscrizione)

81

Fase di traduzione

Associazione unaria

- La traduzione avviene come per le associazioni binarie con attributi distinti per ruoli distinti
- esempio



82

Fase di traduzione

- se uno a molti o uno a uno:
  Impiegato(<u>Imp#</u>,Cognome,Stipendio,Capo#)
- se molti a molti:
  Impiegato(<u>Imp#</u>,Cognome,Stipendio)
  Dirige(<u>Capo#</u>,<u>Subordinato#</u>)

83 Progettazione di basi di dati

Fase di traduzione

- se si introduce una relazione nuova nel caso uno a uno e uno a molti (associazione opzionale per entrambi i ruoli), ci sono due alternative per scegliere la chiave della relazione che rappresenta l'associazione:
  - se la relazione è uno a uno, la chiave è uno qualsiasi dei due attributi corrispondenti ai ruoli giocati dall'entità nell'associazione
  - se la relazione è uno a molti, la chiave è costituita dall'attributo che corrisponde al ruolo dal lato uno della associazione

84 Progettazione di basi di dati

Fase di traduzione

- esempio

  Impiegato(<u>Imp#</u>,Cognome,Stipendio)
  Dirige(Capo#,Subordinato#)

- per determinare la chiave di Dirige:
  - se un impiegato può avere più capi ed un capo può avere più subordinati, la chiave è Capo# e Subordinato#
  - se ogni impiegato ha un solo capo ed ogni capo ha un solo subordinato, la chiave può essere indifferentemente Capo# o Subordinato#
  - se un impiegato può avere un solo capo la chiave è Subordinato#
  - se un impiegato può avere un solo subordinato la chiave è Capo#

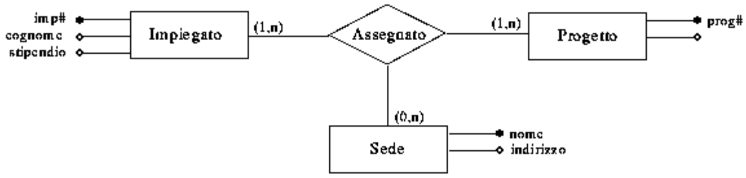
Fase di traduzione

Associazione n-aria

- Nuova relazione contenente:
  - chiavi delle entità partecipanti (che diventano chiave della relazione)
  - attributi dell'associazione

Fase di traduzione

- esempio



Impiegato(Imp#,Cognome,Stipendio)
Progetto(Prog#,Budget)
Sede(Nome,Indirizzo)
Assegnato(Imp# ,Prog# ,NomeSede)

Fase di traduzione

Metodologia di traduzione

- due passi fondamentali:

  1) generazione delle relazioni corrispondenti alle entità dello schema ER e degli attributi delle relazioni generate

  2) generazione delle relazioni corrispondenti alle associazioni presenti nello schema ER che non sono state mappate nelle relazioni generate al passo 1

Fase di traduzione

Passo 1

(a) entità ⇒ relazione
attributo di entità ⇒ attributo di relazione
identificatore di entità ⇒ chiave di relazione

(b) associazione A binaria uno a uno tra E_1 ed E_2
⇒ nella relazione che rappresenta E_1 si aggiunge:
- chiave di E_2
- attributi di A

dove E_1 partecipa obbligatoriamente ad A
(se A è obbligatoria sia per E_1 che per E_2 la scelta è indifferente)

89

Progettazione di basi di dati

Fase di traduzione

(c) associazione A binaria uno a molti tra E_1 ed E_2 con E_1 dal lato uno
⇒ nella relazione che rappresenta E_1 si aggiunge:
- chiave di E_2
- attributi di A

(d) associazione A unaria uno a uno su entità E
⇒ nella relazione che rappresenta E si aggiunge:
- chiave di E per uno dei ruoli
- attributi di A

90

Progettazione di basi di dati

Fase di traduzione

(e) associazione A unaria uno a molti su entità E

⇒ nella relazione che rappresenta E si aggiunge:

- la chiave di E per il ruolo dal lato molti
- attributi di A

91

Fase di traduzione

Passo 2

(a) associazione A binaria molti a molti o n-aria

⇒ nuova relazione contenente:

- chiavi delle relazioni delle entità partecipanti
- attributi di A

(b) associazione A unaria molti a molti su entità E

⇒ nuova relazione contenente:

- attributi corrispondenti alla chiave di E per ogni ruolo
- attributi di A

92

Fase di traduzione

Studente(<u>Matricola</u>,Nome,Via,Città,NomeCdL)
Tutor(<u>Matricola</u>,OrarioR)
CorsodiLaurea(<u>Nome</u>)
Professore(<u>Nome,Cognome,DataN</u>)
Corso(<u>Nome,NomeCdL</u>,Semestre,NomeP,CognomeP,DataNP)
Esame(<u>MatricolaS,NomeC,NomeCdLC</u>,Data,Voto)
PropedeuticoA(<u>NomeCB,NomeCdLCB,NomeCA,NomeCdLCA</u>)
Commissione(<u>NomeC,NomeCdL,NomeP,CognomeP,DataNP</u>)
Segue(<u>MatricolaS,MatricolaT,NomeC,NomeCdL</u>)