Progettazione logica

- Obiettivo della progettazione logica è costruire uno schema logico che descriva in modo corretto ed efficiente tutte le informazioni dello schema concettuale
- non si tratta però di una semplice traduzione da un modello ad un altro
- prima di passare allo schema logico lo schema concettuale va ristrutturato per soddisfare due esigenze:
  - semplificare la traduzione
  - ottimizzare il progetto
- la semplificazione è necessaria perché non tutti i costrutti del modello concettuale (ER o ad oggetti) hanno una traduzione naturale nei modelli logici (es. generalizzazioni)

1

Progettazione logica

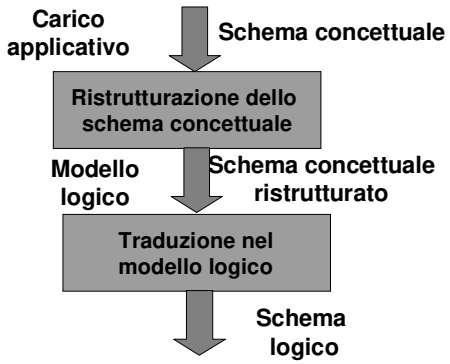
- La progettazione logica è quindi articolata in due fasi:
  - **ristrutturazione dello schema concettuale** fase di riorganizzazione indipendente dal modello logico scelto che si basa su criteri di ottimizzazione dello schema e di semplificazione della fase successiva
  - **traduzione verso il modello logico** fa riferimento ad uno specifico modello logico (nel nostro caso il modello relazionale) e può includere una ulteriore ottimizzazione che si basa sulle caratteristiche del modello logico

2

Progettazione logica

- prima fase
  - dati di ingresso: lo schema concettuale e il carico applicativo previsto (dimensioni dei dati e caratteristiche delle operazioni)
  - risultato: un nuovo schema (ER o ad oggetti) che non è più uno schema concettuale in senso stretto
- seconda fase
  - dati di ingresso: lo schema risultato della prima fase e il modello logico
  - risultato: lo schema logico
- nella seconda fase è possibile effettuare verifiche di qualità dello schema e eventuali ulteriori ottimizzazioni
- i prodotti finali sono lo schema logico finale, i vincoli di integrità su di esso e la relativa documentazione

3



4

Progettazione logica

Analisi delle prestazioni su schemi concettuali

- uno schema concettuale può essere modificato per modificare alcuni indici di prestazione del progetto
- ci si basa su schematizzazioni per effettuare studi di massima su
  - **costo delle operazioni** valutato in termini di occorrenze di classi e associazioni mediamente visitate per effettuare l'operazione
  - **occupazione di memoria** valutato in termini dello spazio di memoria necessario per memorizzare i dati dello schema

5

Progettazione logica

Analisi delle prestazioni su schemi concettuali

- per tali studi occorrono informazioni su
  - volume dei dati
    - numero di occorrenze di ogni classe e associazione dello schema
    - dimensioni di ciascun attributo
  - caratteristiche delle operazioni
    - tipo dell'operazione (interattiva o batch)
    - frequenza (numero medio di esecuzioni in un certo intervallo di tempo)
    - dati coinvolti (classi e/o associazioni)
  - si noti che il carico può essere valutato concentrandosi solo sulle operazioni principali (regola ottanta-venti)

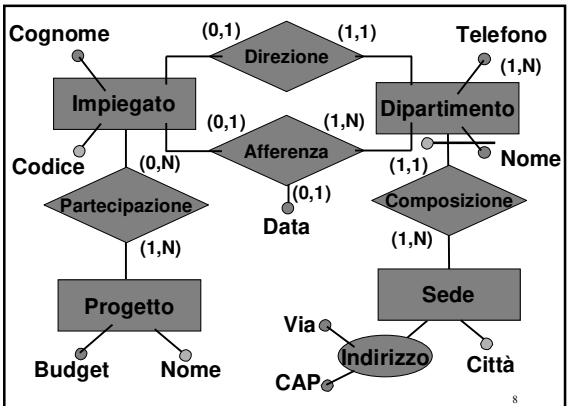
6

Progettazione logica

Analisi delle prestazioni su schemi concettuali

- Tavola dei volumi
  concetto tipo volume
- Tavola delle operazioni
  operazione tipo frequenza
- nella tavola dei volumi il volume delle associazioni dipende dal numero di occorrenze, che a sua volta dipende dal numero di occorrenze delle classi coinvolte e dal numero (medio) di partecipazioni di oggetti di tali classi all'associazione (legato alla cardinalità)
- ad esempio uno studente che effettua un trasferimento ha sostenuto in media 5 esami

7



8

Progettazione logica

Analisi delle prestazioni su schemi concettuali

•Tavola dei volumi

Concetto	Tipo	Volume
Sede	E	10
Dipartimento	E	80
Impiegato	E	2000
Progetto	E	500
Composizione	R	80
Afferenza	R	1900
Direzione	R	80
Partecipazione	R	6000

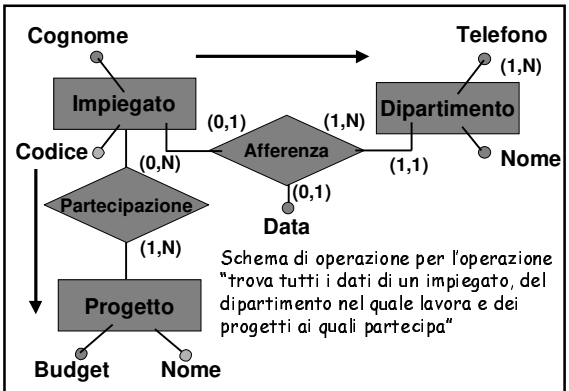
9

Progettazione logica

Analisi delle prestazioni su schemi concettuali

- Per ogni operazione **schema di operazione**: frammento dello schema concettuale interessato dall'operazione su cui è indicato il cammino logico per accedere alle informazioni
- utilizzando lo schema di operazione è possibile stimare il costo di un'operazione contando il numero di accessi alle occorrenze di classi e associazioni necessari per eseguire l'operazione
- in questo modo viene costruita una tavola degli accessi per ogni operazione
- Tavola degli accessi
  concetto tipo num_accessi tipo_accesso (S/L)

10



Schema di operazione per l'operazione "trova tutti i dati di un impiegato, del dipartimento nel quale lavora e dei progetti ai quali partecipa"

11

Progettazione logica

Analisi delle prestazioni su schemi concettuali

•Tavola degli accessi dell'operazione

Concetto	Costrutto	Accessi	Tipo
Impiegato	Entità	1	L
Afferenza	Relazione	1	L
Dipartimento	Entità	1	L
Partecipazione	Relazione	3	L
Progetto	Entità	3	L

12

Progettazione logica

Ristrutturazione di schemi concettuali

- La fase di ristrutturazione di uno schema concettuale si può suddividere in una serie di passi:
  - **analisi delle ridondanze** si decide se eliminare o mantenere eventuali ridondanze dello schema
  - **eliminazione delle generalizzazioni**
  - **partizionamento/accorpamento di classi e associazioni**
  - **scelta degli identificatori primari** in caso ve ne sia più di uno

13

Progettazione logica

Ristrutturazione di schemi concettuali

- **analisi delle ridondanze**
  - una ridondanza corrisponde alla presenza di un dato che può essere derivato da altri dati
  - i casi più frequenti di ridondanza sono
  - attributi derivabili occorrenza per occorrenza da altri attributi della stessa classe/associazione
    es. età e data di nascita
  - attributi derivabili da altri attributi di altre classi/associazioni, di solito attraverso aggregazione
    es. media di uno studente dai voti dei suoi esami
  - attributi derivabili da operazioni di conteggio delle occorrenze
    es. numero esami sostenuti da uno studente
  - associazioni derivabili dalla composizione di altre associazioni in presenza di cicli
    es. Prof tiene Corso, Studente segue Corso, Prof prof_di Studente

14

Progettazione logica

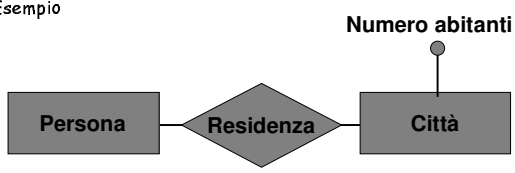
Ristrutturazione di schemi concettuali

- **analisi delle ridondanze**
- la presenza di un dato derivato
  - ha come vantaggio una riduzione del numero degli accessi necessari per calcolare il dato derivato
  - ha come svantaggi
    - una maggiore occupazione di memoria (spesso trascurabile)
    - necessità di effettuare operazioni aggiuntive per mantenere il dato derivato aggiornato
  - la decisione di mantenere o eliminare una ridondanza va presa confrontando il costo di esecuzione delle operazioni che coinvolgono il dato ridondante e la relativa occupazione di memoria, nei casi di presenza e assenza della ridondanza

15

Progettazione logica

Ristrutturazione di schemi concettuali
Analisi delle ridondanze
Esempio



16

Progettazione logica

Ristrutturazione di schemi concettuali
Analisi delle ridondanze
Esempio - Tavola dei volumi e operazioni

Concetto	Tipo	Volume
Città	E	200
Persona	E	1000000
Residenza	R	1000000

- Operazione 1: memorizza una nuova persona con la relativa città di residenza (500 volte al giorno)
- Operazione 2: stampa tutti i dati di una città (incluso il numero di abitanti) (2 volte al giorno)

17

Progettazione logica

Ristrutturazione di schemi concettuali
Analisi delle ridondanze
Esempio - Tavola degli accessi in presenza di ridondanza

Operazione 1

Concetto	Costrutto	Accessi	Tipo
Persona	Entità	1	S
Residenza	Relazione	1	S
Città	Entità	1	L
Città	Entità	1	S

Operazione 2

Concetto	Costrutto	Accessi	Tipo
Città	Entità	1	L

18

Progettazione logica

Ristrutturazione di schemi concettuali
Analisi delle ridondanze
Esempio - Tavola degli accessi in assenza di ridondanza

Operazione 1

Concetto	Costrutto	Accessi	Tipo
Persona	Entità	1	S
Residenza	Relazione	1	S

Operazione 2

Concetto	Costrutto	Accessi	Tipo
Città	Entità	1	L
Residenza	Relazione	5000	L

Progettazione logica

Ristrutturazione di schemi concettuali
Analisi delle ridondanze
Esempio

- Presenza di ridondanza - Costi:
  - Operazione 1: 1500 accessi in scrittura e 500 accessi in lettura al giorno
  - Operazione 2: trascurabile
  - Totale di 3500 accessi al giorno (contiamo doppi gli accessi in scrittura)
- Assenza di ridondanza - Costi:
  - Operazione 1: 1000 accessi in scrittura
  - Operazione 2: 10000 accessi in lettura al giorno
  - Totale di 12000 accessi al giorno

Progettazione logica

Ristrutturazione di schemi concettuali
- **eliminazione delle generalizzazioni**
- il modello relazionale non prevede gerarchie di generalizzazione
- consideriamo C generalizzazione di $C_1, \ldots, C_n$
- tre alternative:

  (a) eliminazione delle classi figlie

  (b) eliminazione della classe padre

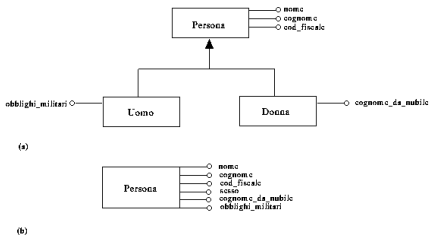
  (c) sostituzione della generalizzazione con associazioni

Progettazione logica

Ristrutturazione di schemi concettuali
- **eliminazione delle generalizzazioni**
- Caso (a): eliminazione delle classi figlie
  - $C_1, \ldots, C_n$ vengono eliminate e i loro attributi vengono inseriti in C
  - a C viene aggiunto un attributo a per tenere traccia delle classi figlie
- generalizzazioni totali $\Rightarrow$ a mai nullo
- generalizzazioni parziali $\Rightarrow$ a nullo per istanza della classe padre che non appartiene a nessuna classe figlia

Progettazione logica

Ristrutturazione di schemi concettuali
- **eliminazione delle generalizzazioni**
- esempio caso (a)



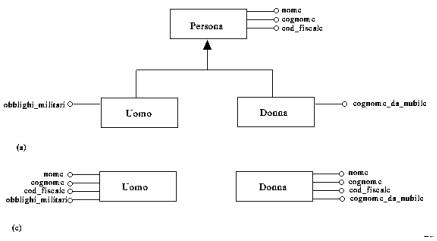
Progettazione logica

Ristrutturazione di schemi concettuali
- **eliminazione delle generalizzazioni**
- Caso (b): eliminazione della classe padre
  - C viene eliminata e i suoi attributi vengono inseriti in $C_1, \ldots, C_n$
  - si può applicare solo nel caso di generalizzazioni totali

Progettazione logica

Ristrutturazione di schemi concettuali

- **eliminazione delle generalizzazioni**
- esempio caso (b)



Progettazione logica

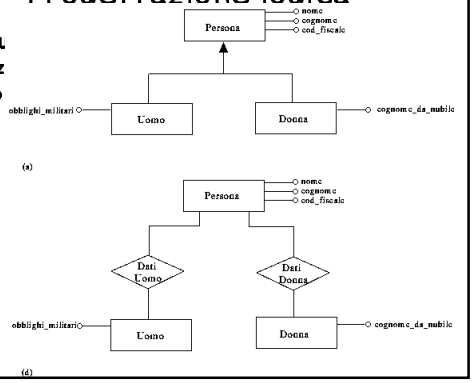
Ristrutturazione di schemi concettuali

- **eliminazione delle generalizzazioni**
- Caso (c): sostituzione della generalizzazione con associazioni
  - $C, C_1, \ldots, C_n$ rimangono invariate
  - la gerarchia di generalizzazione è sostituita da un insieme di associazioni uno a uno, ognuna delle quali lega la classe padre con una diversa classe figlia

Progettazione logica

Ristrutt

- **eliminaz**
- esempio



Progettazione logica

Ristrutturazione di schemi concettuali

- **eliminazione delle generalizzazioni**
- la scelta può essere effettuata considerando vantaggi e svantaggi di ognuna delle scelte possibili relativamente ad occupazione di memoria e costo delle operazioni coinvolte
- è possibile però stabilire delle regole generali
- (a) eliminazione delle classi figlie
  - spreco di memoria per la presenza di valori nulli
  - conviene se le operazioni non fanno distinzione fra le istanze delle varie classi
- (b) eliminazione della classe padre
  - \+ risparmio di memoria
  - applicabile solo per generalizzazioni totali
  - conviene se esistono operazioni che si riferiscono ad istanze di una particolare classe figlia

Progettazione logica

Ristrutturazione di schemi concettuali

- **eliminazione delle generalizzazioni**
- (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 classi padre o figlie
- sono possibili anche combinazioni delle tre trasformazioni presentate
- per generalizzazioni su più livelli si procede analogamente analizzando una generalizzazione alla volta partendo dal fondo

Progettazione logica

Ristrutturazione di schemi concettuali

- **partizionamento/accorpamento di concetti**
- classi e associazioni in uno schema concettuale possono essere partizionati o accorpati per garantire una maggiore efficienza delle operazioni
- gli accessi si riducono separando attributi di uno stesso concetto che vengono acceduti da operazioni diverse e raggruppando attributi di concetti diversi che vengono acceduti dalle stesse operazioni
- le tecniche viste possono essere usate per prendere le decisioni

Progettazione logica

Ristrutturazione di schemi concettuali

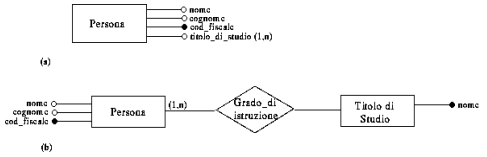
- **partizionamento/accorpamento di concetti**
- partizionamento di classi
- esempio: Ipmiegato sostituita da DatiAnagrafici (Cognome, Nome, Indirizzo, DataNascita) e DatiLavorativi (Livello, Stipendio, Anzianità)
- utile se le operazioni frequenti su Impiegato coinvolgono solo informazioni anagrafiche o lavorative, non insieme
- esempio di decomposizione verticale
- è possibile anche decomposione orizzontale
- esempio: Impiegato decomposto in Analista e Venditore
- utile se le operazioni operano o sugli analisti o sui venditori (corrisponde a introdurre generalizzazione)

31

Progettazione logica

Ristrutturazione di schemi concettuali

- **partizionamento/accorpamento di concetti**
- eliminazione di attributi multivalore
- si definisce una nuova classe, collegata alla classe di partenza con un'associazione, che modella l'attributo multivalore mediante un attributo a valore singolo
- l'associazione introdotta sarà ovviamente uno a molti



Progettazione logica

Ristrutturazione di schemi concettuali

- **partizionamento/accorpamento di concetti**
- accorpamento di classi
- l'accorpamento è l'operazione inversa del partizionamento
- esempio: Persona e Appartamento, legate da associazione uno-a-uno IntestatoA vengono accorpate in un'unica classe
- l'accorpamento ha senso se i dati relativi alle persone sono sempre richiesti insieme a quelli sull'appartamento che occupano
- un problema è la presenza di valori nulli se IntestatoA è opzionale per Persona
- gli accorpamenti si effettuano in genere su associazioni di tipo uno-a-uno

33

Progettazione logica

Ristrutturazione di schemi concettuali

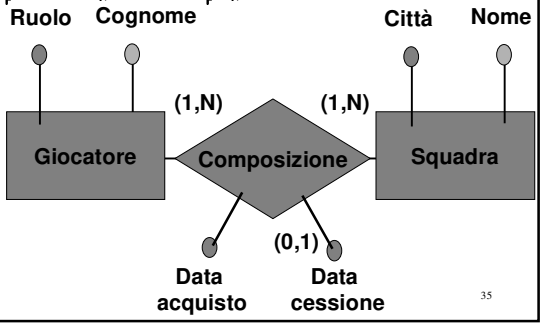
- **partizionamento/accorpamento di concetti**
- altri tipi di partizionamento/accorpamento
- il discorso visto per le classi si può estendere alle associazioni
- in alcuni casi può convenire decomporre un'associazione in due (o più) associazioni tra le stesse classi per separare occorrenze dell'associazione originale accedute sempre separatamente
- viceversa può convenire accorpare due (o pi ù) associazioni tra le stesse classi (che si riferiscono ad uno stesso concetto) in un'unica associazione, quando vengono sempre accedute contemporaneamente

34

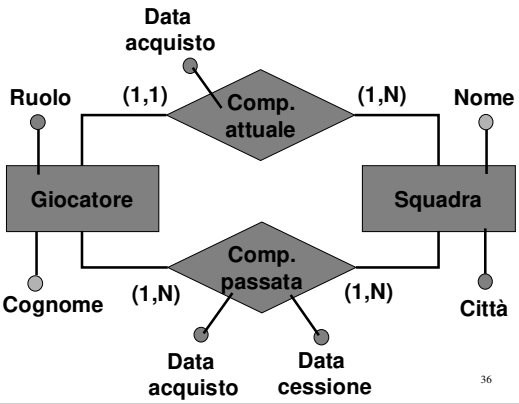
Progettazione logica

Ristrutturazione di schemi concettuali

- **partizionamento/accorpamento di concetti**



35



36

Progettazione logica

Ristrutturazione di schemi concettuali

- **scelta degli identificatori principali**
- operazione indispensabile per la traduzione nel modello relazionale
- nel modello relazionale infatti le chiavi vengono usate per stabilire legami tra dati in relazioni diverse
- criteri
  - **assenza di opzionalità** gli identificatori non possono assumere valori nulli
  - **semplicità** un identificatore composto da uno o pochi attributi è da preferire a identificatori con molti attributi e gli identificatori interni sono da preferire a quelli esterni
  - **utilizzo nelle operazioni più frequenti o importanti**

Progettazione logica

Ristrutturazione di schemi concettuali

- **scelta degli identificatori principali**
- se nessuno degli identificatori soddisfa tali requisiti si introducono nuovi attributi (codici) contenenti valori speciali generati appositamente per questo scopo
- è importante tenere traccia anche degli identificatori non selezionati come primari ma che vengono utilizzati da qualche operazione per accedere ai dati
- per questi identificatori è possibile definire in fase di progettazione fisica degli indici secondari

Progettazione logica

Traduzione verso il modello relazionale

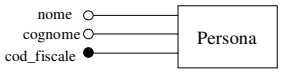
Classi

- Per ogni classe si genera una relazione che ha un attributo per ogni attributo della classe

  classe ⇒ relazione

  attributo di classe ⇒ attributo di relazione

  identificatore ⇒ chiave di relazione
- esempio:



Persona(nome, cognome, cod_fiscale)

Progettazione logica

Traduzione verso il modello relazionale

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

Progettazione logica

Traduzione verso il modello relazionale

Associazione binaria uno a uno

- L'associazione viene modellata mediante attributi inseriti nelle relazioni che modellano le classi partecipanti
- due casi:

  (a) partecipazione obbligatoria di una sola classe

  (b) partecipazione opzionale od obbligatoria di entrambe le classi

Progettazione logica

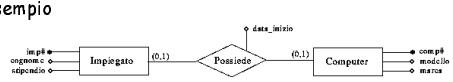
- **Traduzione verso il modello relazionale**
- caso (a): la relazione che rappresenta la classe per cui l'associazione è obbligatoria contiene come chiave esterna la chiave della relazione che rappresenta l'altra classe e come attributi gli attributi dell'associazione
- esempio

Dipartimento(Dip#, Area, Sede, Imp#, Data_inizio)

Impiegato(Imp#, Cognome, Stipendio)

Progettazione logica

- **Traduzione verso il modello relazionale**
- 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)

Progettazione logica

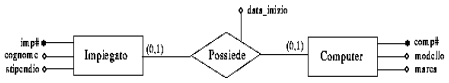
Traduzione verso il modello relazionale

- nel caso particolare di partecipazione opzionale di entrambe le classi si può decidere di introdurre una relazione nuova per modellare l'associazione
- tale relazione contiene:
  - le chiavi delle classi partecipanti
  - gli attributi dell'associazione
- vantaggio: mai valori nulli
- svantaggio: una relazione in più

Progettazione logica

Traduzione verso il modello relazionale

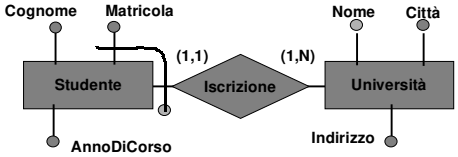
- esempio



Impiegato(Imp#,Cognome,Stipendio)
Computer(Comp#,Modello,Marca)
Possiede(Imp#,Comp#,Data_inizio)
con chiave indifferentemente Imp# o Comp#

Progettazione logica

Traduzione verso il modello relazionale
Identificatori esterni



Studente(Matricola, Università, Cognome, AnnoDiCorso)
Università(Nome, Città, Indirizzo)

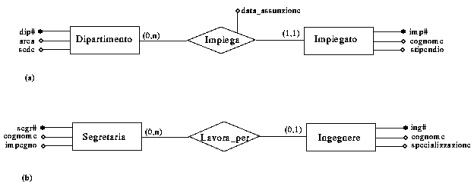
Progettazione logica

Traduzione verso il modello relazionale

Associazione binaria uno a molti

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

Progettazione logica



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

Progettazione logica

Traduzione verso il modello relazionale

- nel caso particolare di partecipazione opzionale della classe dal lato uno si può decidere di introdurre una relazione nuova per modellare l'associazione
- tale relazione contiene:
  - le chiavi delle classi partecipanti
  - gli attributi dell'associazione
- esempio (b)

  Segretaria(Segr#,Cognome,Impegno)

  Ingegnere(Ing#,Cognome,Specializzazione)

  Lavora_Per(Ing#,Segr#)

Progettazione logica

Traduzione verso il modello relazionale

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,Cod_fiscale,Specializzazione)

Associazione_Professionale(Nome, Sede, Num_iscritti)

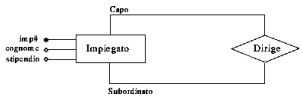
Appartiene(NomeA,SedeA,Cod_fiscaleI,Data_iscrizione)

Progettazione logica

Traduzione verso il modello relazionale

Associazione unaria

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



Progettazione logica

Traduzione verso il modello relazionale

Associazione unaria

- se uno a molti o uno a uno:

  Impiegato(Imp#,Cognome,Stipendio,Capo#)
- se molti a molti:

  Impiegato(Imp#,Cognome,Stipendio)

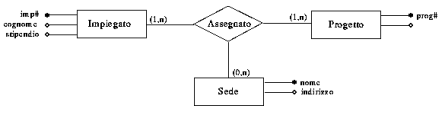
  Dirige(Capo#,Subordinato#)

Progettazione logica

Traduzione verso il modello relazionale

Associazione n-aria

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



Progettazione logica

Traduzione verso il modello relazionale

Associazione n-aria

Impiegato(Imp#,Cognome,Stipendio)

Progetto(Prog#,Budget)

Sede(Nome,Indirizzo)

Assegnato(Imp# ,Prog# ,NomeSede)

Progettazione logica

Documentazione di schemi logici

- il risultato della progettazione logica non è costituito solo dallo schema di base di dati ma anche da una documentazione ad esso associata
- le business rules definite per lo schema concettuale sono ereditate dallo schema logico (se i nomi non sono stati modificati in fase di traduzione)
- è invece importante documentare i vincoli di integrità referenziale introdotti dalla traduzione (ad esempio mediante formalismo grafico)

55