

## Capitoli 5 e 6

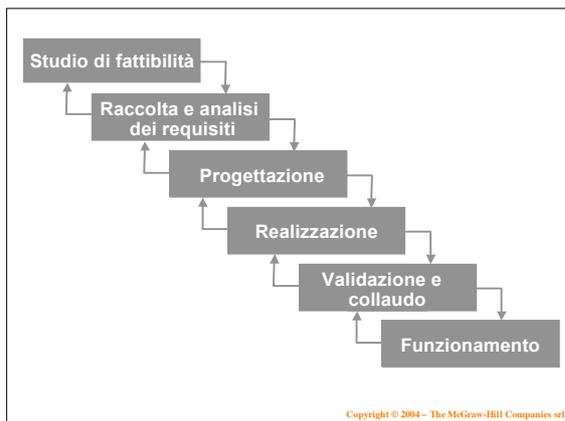
- 5) Il modello entità relazione
- 6) La progettazione logica: dallo schema ER al relazionale

Copyright © 2004 – The McGraw-Hill Companies srl

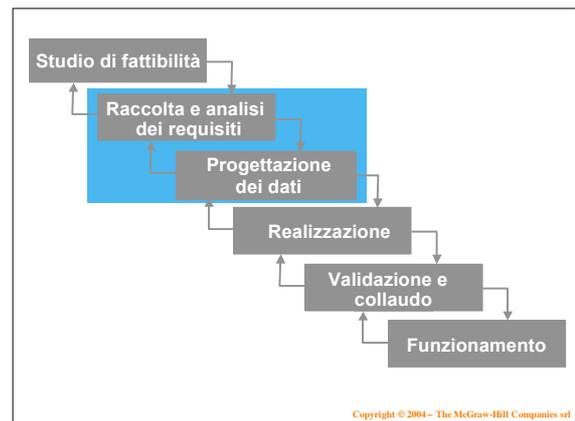
## Progettazione di basi di dati

- È una delle attività del processo di sviluppo dei sistemi informativi
- va quindi inquadrata in un contesto più generale:
- il ciclo di vita dei sistemi informativi:
  - Insieme e sequenzializzazione delle attività svolte da analisti, progettisti, utenti, nello sviluppo e nell'uso dei sistemi informativi
  - attività iterativa, quindi ciclo

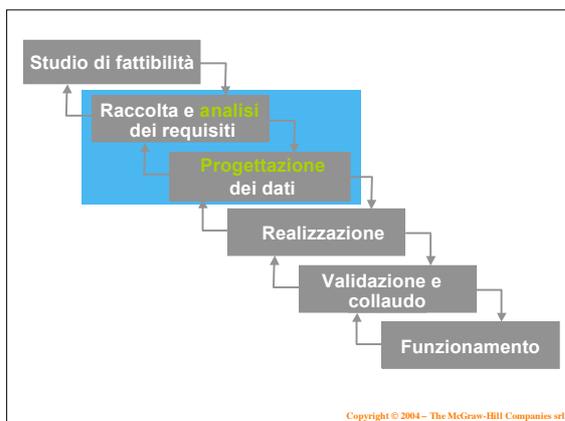
Copyright © 2004 – The McGraw-Hill Companies srl



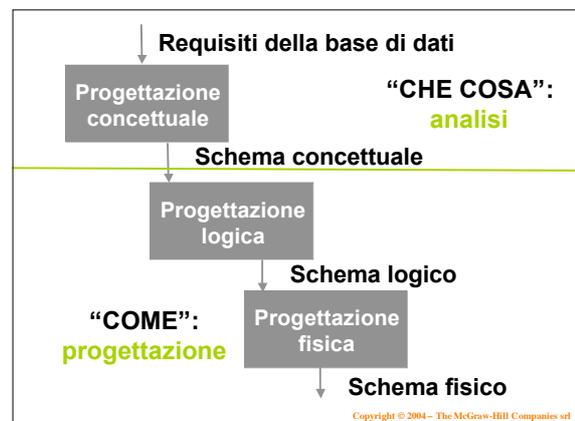
Copyright © 2004 – The McGraw-Hill Companies srl



Copyright © 2004 – The McGraw-Hill Companies srl



Copyright © 2004 – The McGraw-Hill Companies srl

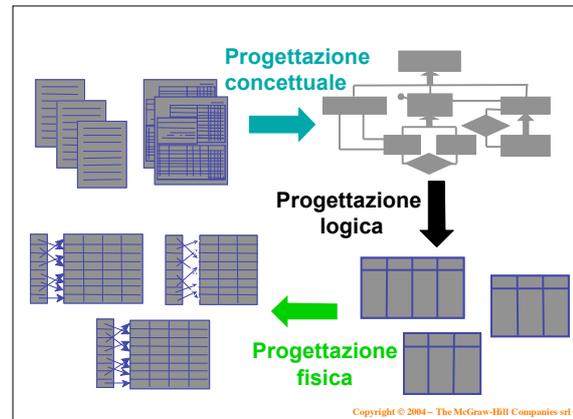


Copyright © 2004 – The McGraw-Hill Companies srl

I prodotti della varie fasi sono  
schemi di alcuni modelli di dati:

- Schema concettuale
- Schema logico
- Schema fisico

Copyright © 2004 – The McGraw-Hill Companies srl



### Modello dei dati

- insieme di costrutti utilizzati per organizzare i dati di interesse e descriverne la dinamica
- componente fondamentale: **meccanismi di strutturazione** (o **costruttori di tipo**)
- come nei linguaggi di programmazione esistono meccanismi che permettono di definire nuovi tipi, così ogni modello dei dati prevede alcuni costruttori
- ad esempio, il **modello relazionale** prevede il costruttore **relazione**, che permette di definire insiemi di record omogenei

Copyright © 2004 – The McGraw-Hill Companies srl

### Due tipi (principali) di modelli

- **modelli logici**: utilizzati nei DBMS esistenti per l'organizzazione dei dati
  - utilizzati dai programmi
  - indipendenti dalle strutture fisicheesempi: **relazionale**, reticolare, gerarchico, a oggetti
- **modelli concettuali**: permettono di rappresentare i dati in modo indipendente da ogni sistema
  - cercano di descrivere i concetti del mondo reale
  - sono utilizzati nelle fasi preliminari di progettazioneil più noto è il modello **Entity-Relationship**

Copyright © 2004 – The McGraw-Hill Companies srl

### Modelli concettuali, perché?

- Proviamo a modellare una applicazione definendo direttamente lo schema logico della base di dati:
  - da dove cominciamo?
  - rischiamo di perderci subito nei dettagli
  - dobbiamo pensare subito a come correlare le varie tabelle (chiavi etc.)
  - i modelli logici sono rigidi

Copyright © 2004 – The McGraw-Hill Companies srl

### Modelli concettuali, perché?

- servono per ragionare sulla realtà di interesse, 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)

Copyright © 2004 – The McGraw-Hill Companies srl

## I costrutti del modello E-R

- Entità
- Relationship
- Attributo
- Identificatore
- Generalizzazione
- ....

Copyright © 2004 – The McGraw-Hill Companies srl

## Entità

- Classe di oggetti (fatti, persone, cose) della applicazione di interesse con proprietà comuni e con esistenza "autonoma"
- Esempi:
  - impiegato, città, conto corrente, ordine, fattura

Copyright © 2004 – The McGraw-Hill Companies srl

## Relationship

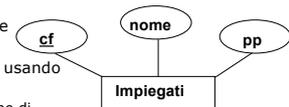
- Legame logico fra due o più entità, rilevante nell'applicazione di interesse
- Esempi:
  - Residenza (fra persona e città)
  - Esame (fra studente e corso)



Copyright © 2004 – The McGraw-Hill Companies srl

## Basi del modello ER

- Entità: oggetto del mondo reale distinguibile da altri oggetti
- Una entità è descritta (nel DB) usando un insieme di attributi
  - Insieme di entità: una collezione di entità simili. Ad esempio, tutti gli impiegati
  - Tutte le entità in un insieme di entità hanno lo stesso insieme di attributi
  - Ciascun insieme di entità ha una chiave
  - Ciascun attributo ha un dominio
  - Si può tradurre facilmente un insieme di entità in una tabella relazionale

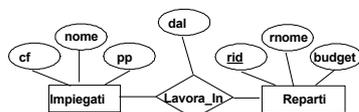


```

CREATE TABLE Impiegati
(cf CHAR(15),
nome CHAR(20),
pp INTEGER,
PRIMARY KEY (cf))
    
```

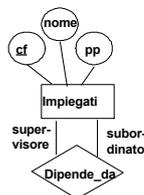
Copyright © 2004 – The McGraw-Hill Companies srl

## Basi del modello ER (segue)



- Relazione: associazione tra 2 o più entità. Ad esempio, Attishoo lavora nel reparto Farmacia
- Insieme di relazioni: collezione di relazioni simili
  - Un insieme n-ario di relazioni R correla n insiemi di entità E1.. En; ciascuna relazione in R coinvolge entità e1 ∈ E1, ... en ∈ En
    - Lo stesso insieme di entità può partecipare a diversi insiemi di relazioni, o in "ruoli" differenti all'interno dello stesso insieme

Copyright © 2004 – The McGraw-Hill Companies srl



## Basi del modello ER (segue)

- Gli insiemi di relazioni possono anche avere attributi descrittivi (ad esempio l'attributo *dal* di *Lavora\_In*)
- Nel tradurre un insieme di relazioni ER in una tabella relazionale, gli attributi della tabella devono includere:
  - chiavi per ciascun insieme di entità partecipante (come chiavi esterne)
    - questo insieme di attributi forma la superchiave per la relazione
  - tutti gli attributi descrittivi

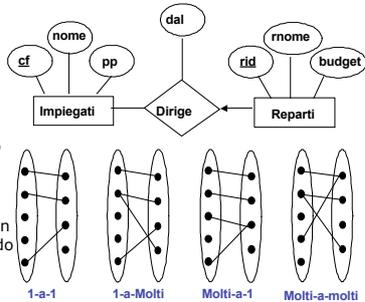
```

CREATE TABLE Lavora_In(
cf CHAR(15),
rid INTEGER,
dal DATE,
PRIMARY KEY (cf, rid),
FOREIGN KEY (cf)
REFERENCES Impiegati,
FOREIGN KEY (rid)
REFERENCES Reparti)
    
```

Copyright © 2004 – The McGraw-Hill Companies srl

### Vincoli di chiave

- Consideriamo Lavora\_In: un impiegato può lavorare in molti reparti; un reparto può avere molti impiegati
- Di contro, ciascun reparto ha al più un direttore, in accordo con il vincolo di chiave su Dirige



Traduzione nel modello relazionale?

Copyright © 2004 – The McGraw-Hill Companies srl

### Traduzione dei diagrammi ER con vincoli di chiave

- Tradurre una relazione in una tabella:

- notare che adesso la chiave è *rid*!
- separare le tabelle per Impiegati e Reparti

```
CREATE TABLE Dirige(
cf CHAR(15),
rid INTEGER,
dal DATE,
PRIMARY KEY (rid),
FOREIGN KEY (cf) REFERENCES Impiegati,
FOREIGN KEY (rid) REFERENCES Reparti)
```

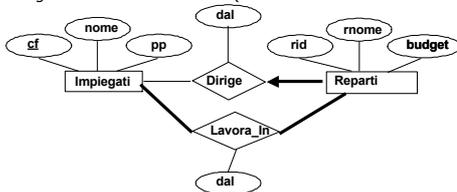
- Poiché ciascun reparto ha un unico direttore, possiamo invece combinare Dirige e Reparti

```
CREATE TABLE Rep_Dir(
Rid INTEGER,
Rnome CHAR(20),
Budget REAL,
Cf CHAR(15),
Dal DATE,
PRIMARY KEY (rid)
FOREIGN KEY (cf) REFERENCES Impiegati)
```

Copyright © 2004 – The McGraw-Hill Companies srl

### Vincoli di partecipazione

- C'è un direttore per ogni reparto?
  - Se sì, questo è un vincolo di partecipazione: la partecipazione di Reparti in Dirige viene detta *totale* (piuttosto che *parziale*)
    - Ogni valore *rid* nella tabella Reparti deve apparire in una riga della tabella Gestisce (con un valore di *cf* non nullo!)

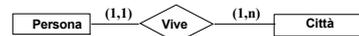


Copyright © 2004 – The McGraw-Hill Companies srl

### Vincoli di cardinalità

- Vincoli di chiave e di partecipazione considerati insieme sono anche detti vincoli di cardinalità (essi definiscono una cardinalità massima e minima, rispettivamente)
- I vincoli di cardinalità definiscono il massimo (minimo) numero di istanze della relazioni cui partecipa una istanza dell'entità

Es.:



Copyright © 2004 – The McGraw-Hill Companies srl

### Vincoli di partecipazione in SQL

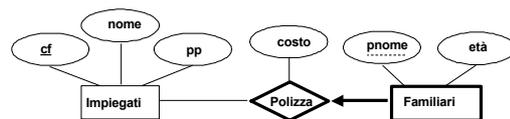
- Possiamo catturare i vincoli di partecipazione che coinvolgono un insieme di entità in una relazione binaria, ma poche altre cose (senza ricorrere a vincoli CHECK)

```
CREATE TABLE Rep_Dir(
Rid INTEGER,
Rnome CHAR(20),
Budget REAL,
Cf CHAR(15),
Dal DATE,
PRIMARY KEY (rid),
FOREIGN KEY (cf) REFERENCES Impiegati,
ON DELETE NO ACTION)
```

Copyright © 2004 – The McGraw-Hill Companies srl

### Entità deboli

- Una entità debole può essere indentificata univocamente solo considerando la chiave primaria di un'altra entità (proprietario)
- L'insieme di entità proprietarie e l'insieme di entità deboli devono partecipare in un insieme di relazioni uno-a-molti (1 proprietario, molte entità deboli)
- L'insieme di entità deboli deve avere partecipazione totale in questo insieme di relazioni identificanti



Copyright © 2004 – The McGraw-Hill Companies srl

### Traduzione di insiemi di entità deboli

- Un insieme di entità deboli e l'insieme di relazioni identificanti sono tradotte in una tabella singola
  - Quando l'entità proprietaria viene cancellata, anche tutte le sue entità deboli devono essere cancellate

```
CREATE TABLE Politica_Rep(
  Rnome CHAR(20),
  età INTEGER,
  costo REAL,
  cf CHAR(15) NOT NULL,
  PRIMARY KEY (rnome, cf)
  FOREIGN KEY (cf) REFERENCES Impiegati,
  ON DELETE CASCADE)
```

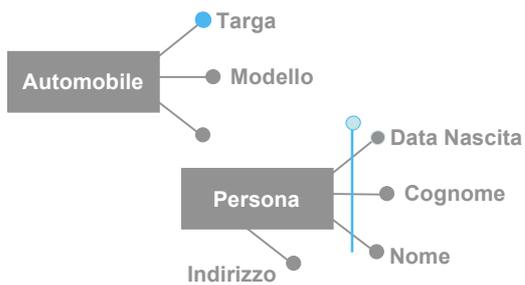
Copyright © 2004 – The McGraw-Hill Companies srl

### Identificatore di una entità

- “strumento” per l'identificazione univoca delle occorrenze di un'entità
- costituito da:
  - attributi dell'entità
    - identificatore interno
  - (attributi +) entità esterne attraverso relationship
    - identificatore esterno

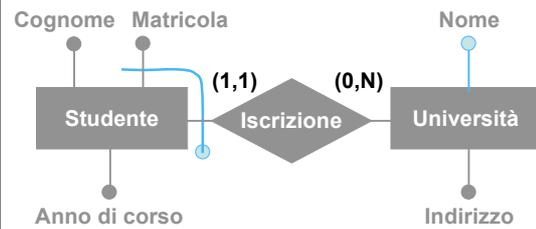
Copyright © 2004 – The McGraw-Hill Companies srl

### Identificatori interni



Copyright © 2004 – The McGraw-Hill Companies srl

### Identificatore esterno

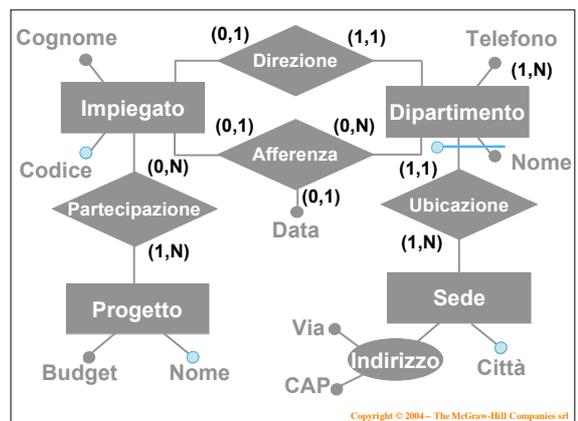


Copyright © 2004 – The McGraw-Hill Companies srl

### Alcune osservazioni

- ogni entità deve possedere almeno un identificatore, ma può averne in generale più di uno
- una identificazione esterna è possibile solo attraverso una relationship a cui l'entità da identificare partecipa con cardinalità (1,1)
- perché non parliamo degli identificatori delle relationship?

Copyright © 2004 – The McGraw-Hill Companies srl



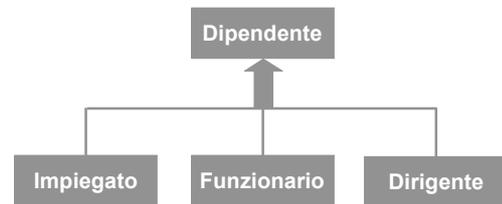
Copyright © 2004 – The McGraw-Hill Companies srl

## Generalizzazione

- mette in relazione una o più entità E1, E2, ..., En con una entità E, che le comprende come casi particolari
  - E è **generalizzazione** di E1, E2, ..., En
  - E1, E2, ..., En sono **specializzazioni** (o sottotipi) di E

Copyright © 2004 - The McGraw-Hill Companies srl

## Rappresentazione grafica

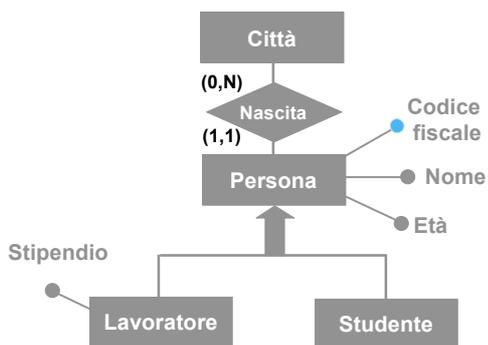


Copyright © 2004 - The McGraw-Hill Companies srl

## Proprietà delle generalizzazioni

- Se E (genitore) è generalizzazione di E1, E2, ..., En (figlie):
- ogni proprietà di E è significativa per E1, E2, ..., En
  - ogni occorrenza di E1, E2, ..., En è occorrenza anche di E

Copyright © 2004 - The McGraw-Hill Companies srl



Copyright © 2004 - The McGraw-Hill Companies srl

## Ereditarietà

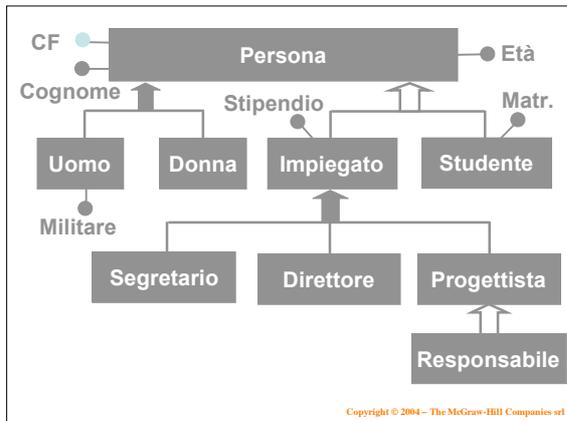
- tutte le proprietà (attributi, relationship, altre generalizzazioni) dell'entità genitore vengono **ereditate** dalle entità figlie e non rappresentate esplicitamente

Copyright © 2004 - The McGraw-Hill Companies srl

## Esercizio

- Le persone hanno CF, cognome ed età; gli uomini anche la posizione militare; gli impiegati hanno lo stipendio e possono essere segretari, direttori o progettisti (un progettista può essere anche responsabile di progetto); gli studenti (che non possono essere impiegati) un numero di matricola; esistono persone che non sono né impiegati né studenti (ma i dettagli non ci interessano)

Copyright © 2004 - The McGraw-Hill Companies srl



### Progettazione concettuale usando il modello ER

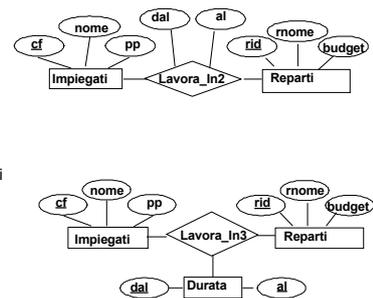
- **Scelte di progetto:**
    - Un concetto dovrebbe essere modellato come una entità o come un attributo?
    - Un concetto dovrebbe essere modellato come una entità o come una relazione?
    - Identificare le relazioni: binarie o ternarie? Aggregazione?
  - **Vincoli nel modello ER:**
    - molta della semantica dei dati può (e dovrebbe) essere catturata
    - però alcuni vincoli non possono essere catturati dai diagrammi ER
  - **Necessità di ulteriori raffinamenti dello schema:**
    - lo schema relazionale ottenuto dai diagrammi ER è un buon punto di partenza. Ma il progetto ER è soggettivo e non può esprimere certi vincoli; quindi lo schema relazionale può aver bisogno di raffinamenti
- Copyright © 2004 – The McGraw-Hill Companies srl

### Entità verso Attributi

- *Indirizzo* dovrebbe essere un attributo di Impiegati o una entità (connessa a Impiegati da una relazione)?
  - Dipende dall'uso che vogliamo fare delle informazioni sull'indirizzo, e dalla semantica dei dati:
    - se abbiamo diversi indirizzi per impiegati, *indirizzo* deve essere una entità (poiché gli attributi non possono assumere più valori)
    - se la struttura (città, via, etc) è importante, ad esempio vogliamo trovare gli impiegati in una data città, *indirizzo* deve essere modellato come una entità (poiché i valori degli attributi sono atomici)
- Copyright © 2004 – The McGraw-Hill Companies srl

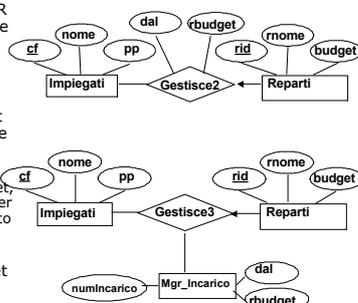
### Entità verso Attributi (segue)

- *Lavora\_In2* non permette a un impiegato di lavorare in un reparto per due o più periodi
- Simile al problema della registrazione di indirizzi multipli per un impiegato: vogliamo registrare diversi valori degli attributi descrittivi per ciascuna istanza della relazione



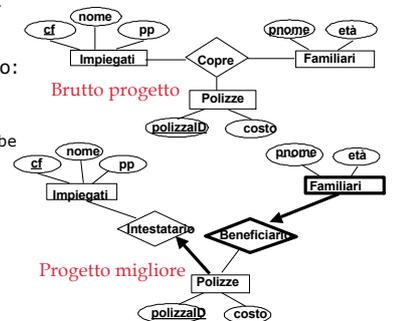
### Entità verso Relazioni

- Il primo diagramma ER va bene se un direttore ha un budget discrezionale separato per ciascun reparto
- Che succede se un direttore ha un budget discrezionale che copre *tutti* i reparti da lui diretti?
  - Ridondanza di *rbudget*, che è memorizzato per ciascun reparto gestito dal direttore
- Ingannevole: suggerisce che *rbudget* sia legato al reparto



### Relazioni binarie verso relazioni ternarie

- Se ogni polizza è posseduta da un solo impiegato:
  - un vincolo di chiave su Polizze significherebbe che una polizza può coprire solo un familiare!
- Quali sono i vincoli aggiuntivi nel secondo diagramma?



### Relazioni binarie verso relazioni ternarie (segue)

- Il vincolo di chiave ci permette di combinare Acquirente con Polizze e Beneficiario con Familiari
  - I vincoli di partecipazione portano a vincoli NOT NULL
  - Che succederebbe se Polizze fosse un insieme di entità deboli?
- ```
CREATE TABLE Polizze(
  polizzaID INTEGER,
  costo REAL,
  cf CHAR(15) NOT NULL,
  PRIMARY KEY (polizzaID),
  FOREIGN KEY (CF) REFERENCES Impiegati
  ON DELETE CASCADE)

CREATE TABLE Familiari (
  Pnome CHAR(20),
  Età INTEGER,
  polizzaID INTEGER,
  PRIMARY KEY (pnome, polizzaID),
  FOREIGN KEY (polizzaID) REFERENCES Polizze
  ON DELETE CASCADE)
```

Copyright © 2004 – The McGraw-Hill Companies srl

### Relazioni binarie verso relazioni ternarie (segue)

- L'esempio precedente illustra un caso in cui due relazioni binarie sono migliori di una relazione ternaria
- Un esempio nell'altra direzione: una relazione ternaria Contratti mette in relazione gli insiemi Pezzi, Reparti e Fornitori, e ha l'attributo descrittivo *qta*. Nessuna combinazione di relazioni binarie è un sostituto adeguato:
- F "può fornire" P, D "ha bisogno" di P, e D "tratta con" F non implicano che D sia d'accordo sul comprare P da F
- Come registriamo *qta*?

Copyright © 2004 – The McGraw-Hill Companies srl

### Vincoli oltre il modello ER

- Dipendenze funzionali:
  - ad esempio, un reparto può ordinare due pezzi distinti dallo stesso fornitore
    - Non si può esprimere in termini della relazione ternaria Contratti
  - La normalizzazione raffina il progetto ER tenendo in considerazione le DF
- Dipendenze di inclusione:
  - caso speciale: chiavi esterne (si possono esprimere nel modello ER)
  - ad esempio, almeno una persona deve dipendere da ciascun direttore (l'insieme dei valori *cf* in Dirige deve essere un sottoinsieme dei valori *supervisore\_cf* in Dipende\_da) Chiave esterna? Esprimibile nel modello ER?
- Vincoli generali
  - Ad esempio, budget discrezionale del direttore minore del 10% dei budget combinati di tutti i reparti che gestisce

Copyright © 2004 – The McGraw-Hill Companies srl

### Riassunto della progettazione concettuale

- La progettazione concettuale segue l'analisi dei requisiti
  - Porta a una descrizione ad alto livello dei dati che devono essere memorizzati
- Il modello ER è molto usato per la progettazione concettuale
  - I costrutti sono espressivi, vicini al modo in cui la gente pensa alle applicazioni
- Costrutti di base: entità, relazioni e attributi (di entità e relazioni)
- Alcuni costrutti addizionali: entità deboli, gerarchie ISA e aggregazione
- Nota: ci sono molte varianti del modello ER

Copyright © 2004 – The McGraw-Hill Companies srl

### Riassunto della progettazione concettuale (segue)

- Il progetto ER è soggettivo. Ci sono spesso molti modi per modellare un dato scenario! Analizzare le alternative può essere complicato, specialmente per una grande azienda. Scelte comuni includono:
- Entità verso attributo, entità verso relazione, relazione binaria o n-aria, uso delle gerarchie ISA, uso dell'aggregazione
- Garantire un buon progetto della base di dati: lo schema relazionale risultante dovrebbe essere analizzato e ulteriormente raffinato. Le informazioni delle DF e le tecniche di normalizzazione sono particolarmente utili.

Copyright © 2004 – The McGraw-Hill Companies srl

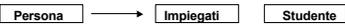
### Metodologie di progetto

- Le metodologie per la progettazione concettuale sono basate su:
  - raffinamento incrementale dello schema
  - astrazione come meccanismo di raffinamento
  - controllo della qualità dello schema concettuale
  - decomposizione in sottoschemi
  - analisi integrata di dati e funzioni
- Le principali metodologie sono
  - top-down
  - bottom-up
  - mista

Copyright © 2004 – The McGraw-Hill Companies srl

### Top-down

- Implementa una serie di trasformazioni dello schema iniziando con concetti molto astratti per andare verso concetti più concreti e dettagliati
- Il primo schema contiene (in una rappresentazione molto astratta) tutto il contenuto di informazione della realtà da rappresentare
- Meccanismo di trasformazione: raffinamento

Es. 

Copyright © 2004 – The McGraw-Hill Companies srl

### Bottom-up

- Parte dai requisiti dettagliati, raggruppandoli in concetti più astratti
- Meccanismo di trasformazione: astrazione



Problemi: conflitti tra le definizioni dei concetti, ridondanze, necessità di ristrutturazione (ma PIÙ NATURALE)

Copyright © 2004 – The McGraw-Hill Companies srl

### Mista

- Decomposizione “controllata” dei requisiti
- Struttura globale dello schema
- Generazione bottom-up dei sottoschemi
- Integrazione dei sottoschemi guidata dalla struttura

Copyright © 2004 – The McGraw-Hill Companies srl

### Quale costrutto E-R va utilizzato per rappresentare un concetto presente nelle specifiche?

Bisogna basarsi sulle definizioni dei costrutti del modello E-R:

- se ha proprietà significative e descrive oggetti con esistenza autonoma
  - entità
- se è semplice e non ha proprietà
  - attributo
- se correla due o più concetti
  - relazione
- se è caso particolare di un altro
  - generalizzazione

Copyright © 2004 – The McGraw-Hill Companies srl

### Una metodologia

- Analisi dei requisiti
  - Analizzare i requisiti ed eliminare le ambiguità
  - Costruire un glossario dei termini
  - Raggruppare i requisiti in insiemi omogenei
- Passo base
  - Definire uno schema scheletro con i concetti più rilevanti
- Passo iterativo (da ripetere finché non si è soddisfatti)
  - Raffinare i concetti presenti sulla base delle loro specifiche
  - Aggiungere concetti per descrivere specifiche non descritte
- Analisi di qualità (ripetuta e distribuita)
  - Verificare le qualità dello schema e modificarlo

Copyright © 2004 – The McGraw-Hill Companies srl

### Qualità dello schema concettuale

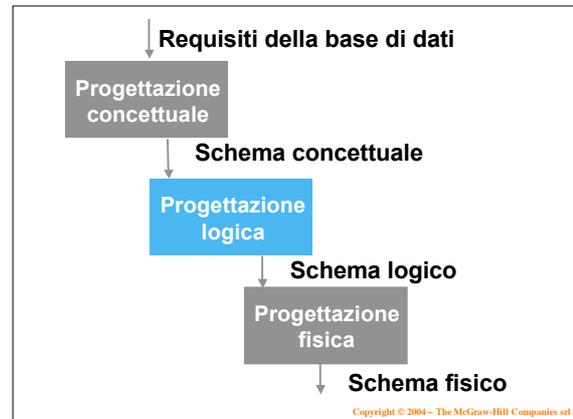
- Correttezza: lo schema rappresenta correttamente (sintatticamente e semanticamente) i requisiti iniziali
- Completezza: lo schema rappresenta tutti i requisiti
- Minimalità: lo schema rappresenta solo i requisiti e ogni loro aspetto appare solo una volta
- Leggibilità: lo schema è facile da interpretare ed esprime i requisiti in modo naturale
- Modificabilità: lo schema può essere facilmente modificato se i requisiti cambiano
- Auto-documentabilità: lo schema non ha bisogno di materiale aggiuntivo esterno

Copyright © 2004 – The McGraw-Hill Companies srl

### Da concettuale a logico

- Traduzione di uno schema concettuale (E-R) in uno schema (relazionale) logico
- Fare attenzione ai vincoli di integrità!
- La prima ottimizzazione si basa sulla frequenza degli accessi alle entità e alle relazioni
- Due fasi:
  - ristrutturazione dello schema concettuale e prima ottimizzazione
  - traduzione e ulteriore ottimizzazione (basata sul modello logico)

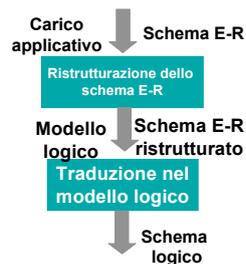
Copyright © 2004 – The McGraw-Hill Companies srl



Copyright © 2004 – The McGraw-Hill Companies srl

### Dati di ingresso e uscita

- **Ingresso:**
  - schema concettuale
  - informazioni sul carico applicativo
  - modello logico
- **Uscita:**
  - schema logico
  - documentazione associata



Copyright © 2004 – The McGraw-Hill Companies srl

### Ristrutturazione

- Analisi delle ridondanze
- Traduzione delle gerarchie IS-A
- Partizionamento delle entità
- Fusione delle entità
- Scelta delle chiavi primarie

Copyright © 2004 – The McGraw-Hill Companies srl

### Ristrutturazione schema E-R

- Non si tratta di una pura e semplice traduzione
  - alcuni aspetti non sono direttamente rappresentabili
  - è necessario considerare le prestazioni
- Motivazioni:
  - semplificare la traduzione
  - "ottimizzare" le prestazioni
- Osservazione:
  - uno schema E-R ristrutturato non è (più) uno schema concettuale nel senso stretto del termine

Copyright © 2004 – The McGraw-Hill Companies srl

### Per ottimizzare il risultato abbiamo bisogno di analizzare le prestazioni a questo livello

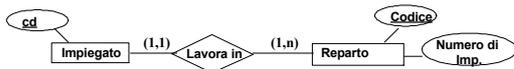
- Ma le prestazioni non sono valutabili con precisione su uno schema concettuale!
- "indicatori" significativi: parametri che regolano le prestazioni
  - spazio: numero di occorrenze previste
  - tempo: numero di occorrenze (di entità e relationship) visitate durante un'operazione

Copyright © 2004 – The McGraw-Hill Companies srl

### Ridondanze

- Pro
  - Maggiore velocità di esecuzione di alcune interrogazioni
- Contro
  - Aumento in memoria
  - Alto costo del mantenimento dell'integrità

Es.:



Copyright © 2004 - The McGraw-Hill Companies srl

### Valutazione del costo delle Transazioni

1. Per ciascun reparto, visualizzare il corrispondente numero di impiegati (ogni giorno)
2. Assegnare ogni nuovo impiegato a un reparto (ogni giorno)
3. Spostare un impiegato da un reparto a un altro (ogni 3 giorni)

Dimensione di entità e relazioni:

Impiegati - 100  
Lavora in - 100  
Reparto - 10

Copyright © 2004 - The McGraw-Hill Companies srl

### Modifiche di tempo e memoria con ridondanza

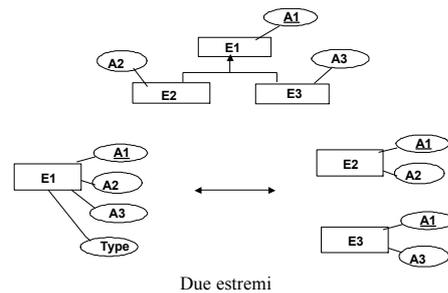
- Operazione 1: 10 accessi a Reparto; -100 accessi a Impiegato
- Operazione 2: 1 accesso a Impiegato; +1 accesso a Reparto
- Operazione 3: 1 accesso a Impiegato; +2 accessi a Reparto

Incremento di memoria: 10

Accessi (ogni 3 giorni):  $-100 \cdot 3 + 1 \cdot 3 + 2$

Copyright © 2004 - The McGraw-Hill Companies srl

### Traduzione in gerarchia IS-A



Due estremi

Copyright © 2004 - The McGraw-Hill Companies srl

### Fusione contro divisione

- Fusione
  - Grande occupazione di memoria quando le sotto-entità hanno molti attributi
  - Inutile se l'entità principale ha pochi attributi e/o relazioni (utile nel caso opposto)
  - Valori *null*
- Divisione
  - Nessuno spreco di memoria se non c'è sovrapposizione tra le sotto-entità
  - Inutile se ci sono molti accessi di tipo *join* alle sotto-entità
  - Necessità di procedure per il controllo dell'integrità

Copyright © 2004 - The McGraw-Hill Companies srl

### Vincoli di integrità

- Fusione
  - $E1 = E \text{ e } TIPO = E1$
  - $E2 = E \text{ e } TIPO = E2$
- Divisione
  - $E = E1 \cup E2$
  - $R = R1 \cup R2$  (per ogni R cui partecipa E)
- $E1 \cap E2 = \phi$  (se non sono permesse sovrapposizioni)

Copyright © 2004 - The McGraw-Hill Companies srl

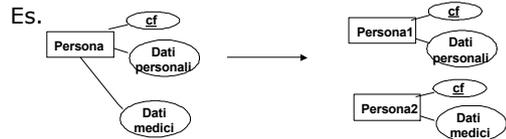
### Altre soluzioni

- Traduzione mista (parte delle entità della gerarchia sono fuse, altre vengono separate)
- Ridondanza (vengono lasciate tutte le entità)
  - Operazioni di aggiornamento molto complesse (controllo dell'integrità)
  - Spreco di memoria

Copyright © 2004 - The McGraw-Hill Companies srl

### Partizionamento e fusione delle entità

- Partizionamento: le entità vengono separate in base agli accessi delle transazioni



Vincolo di integrità:  $Persona1[CF] = Persona2[CF]$

Nota: quando si divide, ricordare che le proprietà delle entità non sono solo attributi, ma anche relazioni e gerarchie

Copyright © 2004 - The McGraw-Hill Companies srl

### Partizionamento e fusione delle entità (cont)

- Le entità cui si accede spesso possono essere fuse



Vincoli di integrità:  $E12[A1] = E1[A1]$   
 $E12[A2] = E2[A2]$

Copyright © 2004 - The McGraw-Hill Companies srl

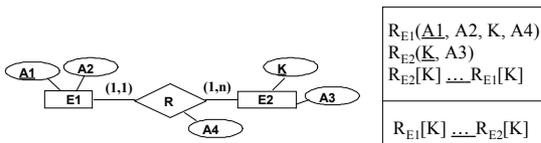
### Chiavi primarie

- Per ciascuna entità si deve scegliere una chiave primaria
- La chiave primaria di una relazione è data dall'unione delle chiavi primarie delle entità che vi partecipano

Copyright © 2004 - The McGraw-Hill Companies srl

### Traduzione nel modello relazionale

- Ciascuna entità diventa una relazione, con la chiave primaria corrispondente
- Ciascuna relazione (m:n) diventa una relazione, con la chiave primaria corrispondente
- Ciascuna relazione (1:n) e (1:1) viene tradotta tramite inclusione della chiave esterna



- Le relazioni con cardinalità minima uguale a 0 possono originare valori null

Copyright © 2004 - The McGraw-Hill Companies srl

### Note

- Tutti i vincoli di integrità che si originano durante la progettazione logica e/o la traduzione devono essere importati nello schema relazionale (principalmente come vincoli di inclusione di chiave)
- Tutte le relazioni devono essere in 1NF (vedi normalizzazione), cioè tutti gli attributi sono atomici e monovalore
- Le relazioni ottenute traducendo lo schema ER possono essere ulteriormente normalizzate

Copyright © 2004 - The McGraw-Hill Companies srl