

# BASI DI DATI

## PROGETTO DELLA BASE DI DATI

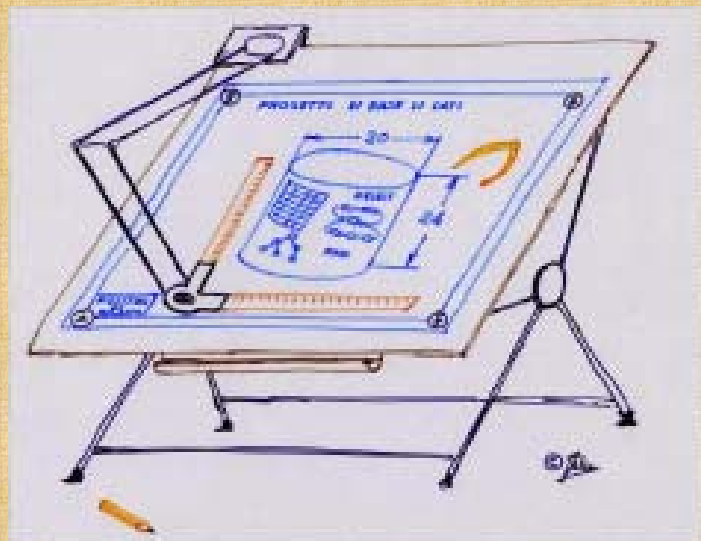
**Prof. Fabio A. Schreiber**

Dipartimento di Elettronica e Informazione  
Politecnico di Milano

tratto da: Atzeni, Ceri, Paraboschi, Torlone - Basi di Dati -  
McGraw-Hill



## PROGETTAZIONE DELLA BASE DI DATI



© Fabio A. Schreiber

Progettazione 1

## CICLO DI VITA DEL SISTEMA INFORMATIVO

- **RACCOLTA E ANALISI DEI REQUISITI**
- **PROGETTAZIONE**
- **IMPLEMENTAZIONE**
- **VALIDAZIONE E COLLAUDO**
- **ESERCIZIO E MANUTENZIONE**

© Fabio A. Schreiber

Progettazione 2

## ANALISI DEI REQUISITI

- **INIVIDUAZIONE DI**
    - UNITA' ORGANIZZATIVE OMOGENEE UTENTI DEL SISTEMA
    - ATTIVITA' CHE IL SISTEMA DOVRA' SUPPORTARE
    - PROPRIETA' E FUNZIONALITA' DEL SISTEMA
    - REQUISITI DI MASSIMA DI HW E SW PER SUPPORTARE IL SISTEMA
- RICHIEDE DI INTERAGIRE CON GLI UTENTI A TUTTI I LIVELLI, IN PARTICOLARE CON LA DIREZIONE AZIENDALE**

© Fabio A. Schreiber

Progettazione 3

## ANALISI DEI REQUISITI

- **PRODUZIONE DI**
  - DESCRIZIONE INFORMALE DEI DATI COINVOLTI E DELLE OPERAZIONI SU DI ESSI
  - PIANO DI SVILUPPO DEL SISTEMA
    - **PRIORITA'**
    - **TEMPI DI REALIZZAZIONE**
  - STUDIO DI FATTIBILITA'
    - **RISORSE**
      - IMPIANTI
      - PERSONALE
    - **COSTI**
    - **PROBLEMI DEL PASSAGGIO E ADATTAMENTO AL NUOVO SISTEMA**

© Fabio A. Schreiber

Progettazione 4

## ANALISI DEI REQUISITI

### LINEE GUIDA

- SCEGLIERE TERMINI SPECIFICI
  - EVITARE SINONIMI E OMONIMI
- STANDARDIZZARE (FORMALIZZARE) LE FRASI
  - SE <COND> ALLORA <CASO 1> ALTRIMENTI <CASO 2>
- EVITARE CIRCONLOCUZIONI
  - OPERATORE ECOLOGICO = SPAZZINO (!)
- ESPLICITARE TUTTI I RIFERIMENTI
  - “I PARTECIPANTI DI UNA DITTA INDICANO IL NUMERO DI TELEFONO”: **DI CHI????**

© Fabio A. Schreiber

Progettazione 5

## USO DEL LINGUAGGIO NATURALE il progetto di un'altalena



come è stata proposta dal finanziatore del progetto



come è stata descritta



come è stata progettata dal progettista



come è stata prodotta dal programmatore

© Fabio A. Schreiber



come è stata installata presso l'utente



cosa voleva veramente l'utente

Progettazione 6

## ANALISI DEI REQUISITI: ESEMPIO

### FUNZIONAMENTO DI UN MAGAZZINO

- IL MAGAZZINO E' COMPOSTO DA SCAFFALI
- I FORNITORI FORNISCONO I PRODOTTI
- I CLIENTI ORDINANO I PRODOTTI
- GLI SCAFFALI CONTENGONO I PRODOTTI
- GLI OPERAI SONO ADDETTI AGLI SCAFFALI

© Fabio A. Schreiber

Progettazione 7

## ANALISI DEI REQUISITI: DIZIONARIO

TERMINE	DESCRIZIONE	SINONIMI	LEGAMI
fornitore	p.IVA, denominaz., indirizzo, num. Tel.		prodotto
cliente	p.IVA, denominaz., indirizzo, num. Tel.	acquirente	prodotto
prodotto	codice, nome, genere	articolo, voce	fornitore scaffale cliente
scaffale	supporto numerato	ripiano	operaio prodotto
operaio	dati anagrafici, matricola, qualifica	addetto, magazziniere	scaffale

© Fabio A. Schreiber

Progettazione 8

## PROGETTAZIONE

- **PROGETTAZIONE DEI DATI**  
INDIVIDUA L'ORGANIZZAZIONE E LA STRUTTURA DELLA BASE DI DATI
  - CONCETTUALE
  - LOGICA
  - FISICA
- **PROGETTAZIONE DELLE APPLICAZIONI**  
– SCHEMATIZZA LE OPERAZIONI SUI DATI E PROGETTA IL SOFTWARE APPLICATIVO

© Fabio A. Schreiber

Progettazione 9



## IMPLEMENTAZIONE

- VENGONO **DEFINITE** LE COMPONENTI HW E SW DI BASE E DI SVILUPPO **DA ACQUISIRE** SUL MERCATO
- VIENE **COSTRUITA E POPOLATA** LA BASE DI DATI
- VENGONO **SVILUPPATI** I PROGRAMMI APPLICATIVI

© Fabio A. Schreiber

Progettazione 10

## VALIDAZIONE E COLLAUDO

- VERIFICA DELLA **CORRETTEZZA DEI DATI**
- VERIFICA DEL **CORRETTO FUNZIONAMENTO DELLE APPLICAZIONI**
- VERIFICA DELLE **PRESTAZIONI** NELLE CONDIZIONI OPERATIVE
  - **THROUGHPUT**
  - **TEMPO DI RISPOSTA**
- VERIFICA DELLA **SICUREZZA** DEL SISTEMA E DEI DATI
- VERIFICA DELLA **RESISTENZA AI GUASTI**
- EVENTUALE PERIODO DI **PARALLELO** CON IL SISTEMA PRECEDENTE

© Fabio A. Schreiber

Progettazione 11

## ESERCIZIO E MANUTENZIONE

- **AMMINISTRAZIONE DEL SISTEMA**
  - CONTROLLO DEL CORRETTO FUNZIONAMENTO DEL SISTEMA
  - AUDITING DI SICUREZZA
  - GESTIONE DEGLI UTENTI
- **MANUTENZIONE**
  - **CORRETTIVA**
    - RIMUOVE EVENTUALI GUASTI O ERRORI
  - **ADATTATIVA**
    - RIVEDE SPECIFICHE E IMPLEMENTAZIONE IN FUNZIONE DI MUTATE ESIGENZE APPLICATIVE
  - **EVOLUTIVA**
    - FA CRESCERE LE FUNZIONALITA' E LE PRESTAZIONI DEL SISTEMA NEL TEMPO

© Fabio A. Schreiber

Progettazione 12

## PROGETTAZIONE CONCETTUALE

TRADUCE IL RISULTATO DELL'ANALISI DEI REQUISITI IN UNA **DESCRIZIONE FORMALE**

- DI **FACILE COMPrensione** PER GLI ESPERTI DI ORGANIZZAZIONE AZIENDALE
- CHE RAPPRESENTI IL **CONTENUTO INFORMATIVO** DELLA BASE DI DATI
- INDIPENDENTE DAL DBMS

UTILIZZA UN **MODELLO CONCETTUALE**

© Fabio A. Schreiber

Progettazione 13

## PROGETTAZIONE CONCETTUALE

### SCHEMA CONCETTUALE

RAPPRESENTAZIONE SEMPLIFICATA  
CONTENENTE **TUTTI E SOLI** GLI ASPETTI  
INTERESSANTI PER LA GESTIONE  
DELL'AZIENDA

- ELIMINA LE **AMBIGUITA'** DEL LINGUAGGIO NATURALE
- ELIMINA I POSSIBILI **CONFLITTI DI TERMINOLOGIA** PRESENTI NEL GLOSSARIO

E' IL DOCUMENTO DI PROGETTO  
FONDAMENTALE CHE DEVE ESSERE  
**VALIDATO DALLA DIREZIONE AZIENDALE**

© Fabio A. Schreiber

Progettazione 14

## MODELLI CONCETTUALI

- **LINGUAGGI FORMALI**
  - ESEGUIBILI
  - NON ESEGUIBILI
- **RAPPRESENTAZIONI GRAFICHE CORREDATE DA SCHEDE DESCRITTIVE**
  - **MODELLO ENTITA'- RELAZIONE (E-R)** (P. P. Chen, 1976)
    - NELLE SUE DIVERSE VARIANTI E' IL PIU' DIFFUSO
    - SONO DISPONIBILI STRUMENTI CASE DI AUSILIO ALLA PROGETTAZIONE
  - **UNIFIED MODELING LANGUAGE (UML)**
    - STRUMENTO PIU' RECENTE CHE PUO' ESSERE UTILIZZATO A VARI LIVELLI ANCHE PER IL PROGETTO CONGIUNTO DATI-FUNZIONI

© Fabio A. Schreiber

Progettazione 15



## IL MODELLO ENTITA' RELAZIONE (E-R)

- IL MODELLO E-R USA **SIMBOLI GRAFICI** PER FAVORIRE LA COMPrensIONE DEI CONCETTI ANCHE E SOPRATTUTTO AI NON SPECIALISTI
- IL MODELLO E-R- E' QUINDI UN **GRAFO** CHE USA **NODI ETEROGENEI** PER RAPPRESENTARE **ENTITA' E RELAZIONI**
- I GRAFI E-R VENGONO AFFIANCATI DA SIMBOLI DI VINCOLO E DA SCHEDE DI DETTAGLIO

© Fabio A. Schreiber

Progettazione 16

## ENTITA'

- RAPPRESENTA UNA **CLASSE DI OGGETTI** DEL MONDO RALE DI INTERESSE PER L'APPLICAZIONE
- GLI OGGETTI POSSONO ESSERE
  - MATERIALI (automobili, impiegati, ...)
  - IMMATERIALI (conti correnti, corsi, ...)
- OGNI ENTITA' E' **CARATTERIZZATA DA UN NOME**

© Fabio A. Schreiber

Progettazione 17

## ENTITA'

IL SIMBOLO GRAFICO CHE RAPPRESENTA  
UN'ENTITA' E' IL **RETTANGOLO**

NOME DELLA  
ENTITA'

ESEMPI

STUDENTE

ESAME

CONTO  
CORRENTE

© Fabio A. Schreiber

Progettazione 18

## ASSOCIAZIONE (O RELAZIONE)

- RAPPRESENTA UN **LEGAME LOGICO** TRA ENTITA' DI INTERESSE PER L'APPLICAZIONE
- UN'ISTANZA DI UN'ASSOCIAZIONE E' UNA **ENNUPLA** TRA SINGOLE ISTANZE DELLE ENTITA' COINVOLTE
- OGNI ASSOCIAZIONE E' **DISTINTA DA UN NOME**
- CI POSSONO ESSERE **PIU' ASSOCIAZIONI DIVERSE** TRA LE STESSE ENTITA'

© Fabio A. Schreiber

Progettazione 19

## ASSOCIAZIONE

IL SIMBOLO GRAFICO CHE RAPPRESENTA UN'ASSOCIAZIONE E' IL **ROMBO**



ESEMPIO

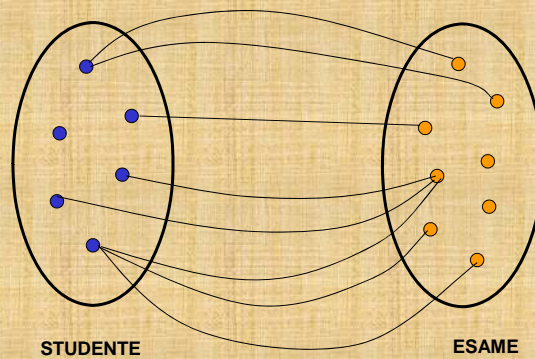


© Fabio A. Schreiber

Progettazione 20

## ASSOCIAZIONE TRA INSIEMI

L'ASSOCIAZIONE RAPPRESENTA UNA **RELAZIONE TRA INSIEMI**



© Fabio A. Schreiber

Progettazione 21

## CARATTERISTICHE DELLE ASSOCIAZIONI

### CARDINALITA' (min, Max)

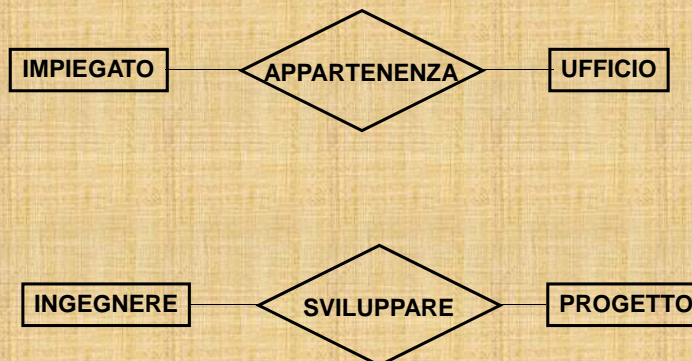
NUMERO DI VOLTE CHE UNA DATA  
ISTANZA DI ENTITA' DEVE O PUO'  
PARTECIPARE ALL'ASSOCIAZIONE

- (1,1) OBBLIGATORIA, UNA SOLA VOLTA
- (1,n) OBBLIGATORIA, FINO AD n VOLTE
- (0,1) OPZIONALE , UNA SOLA VOLTA
- (0,n) OPZIONALE, FINO AD n VOLTE

© Fabio A. Schreiber

Progettazione 22

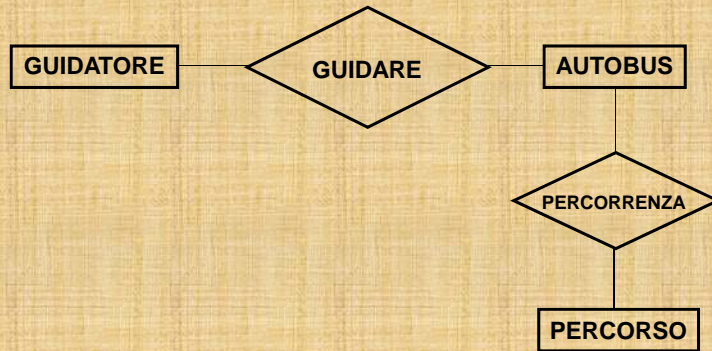
## ESEMPI DI SCHEMI



© Fabio A. Schreiber

Progettazione 23

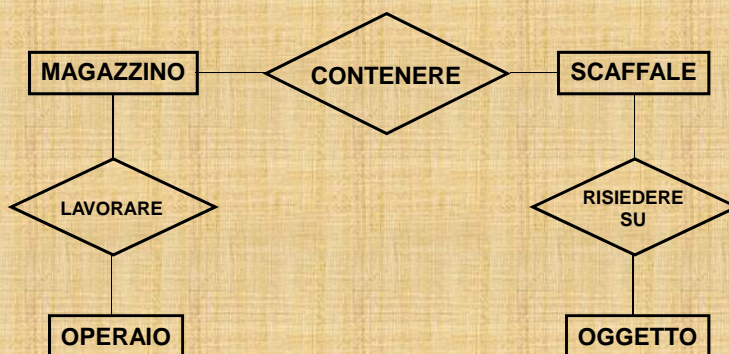
## ESEMPI DI SCHEMI



© Fabio A. Schreiber

Progettazione 24

## ESEMPI DI SCHEMI

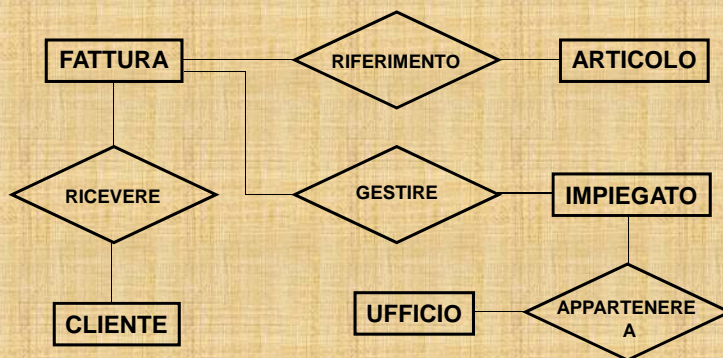


© Fabio A. Schreiber

Progettazione 25



## ESEMPI DI SCHEMI



© Fabio A. Schreiber

Progettazione 26

## PROPRIETA' (O ATTRIBUTI)

- RAPPRESENTANO **CARATTERISTICHE** DELLE ENTITA' O DELLE ASSOCIAZIONI DI INTERESSE PER L'APPLICAZIONE
- ENTITA' E ASSOCIAZIONI VENGONO DESCRITTE MEDIANTE **L'AGGREGAZIONE DI PROPRIETA'**
- OGNI ISTANZA DI ENTITA' O DI ASSOCIAZIONE POSSIEDE UN **VALORE** PER CIASCUN ATTRIBUTO
- OGNI ATTRIBUTO E' **CARATTERIZZATO DA UN NOME**

© Fabio A. Schreiber

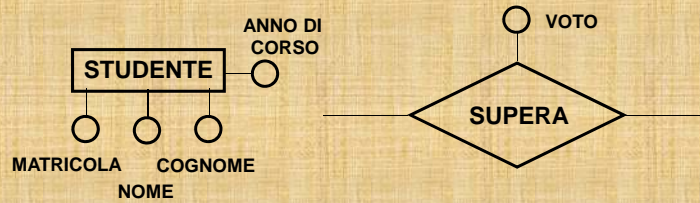
Progettazione 27

## ATTRIBUTI

IL SIMBOLO GRAFICO CHE RAPPRESENTA UN ATTRIBUTO E' IL **CIRCOLETTO**



### ESEMPI



© Fabio A. Schreiber

Progettazione 28

## ATTRIBUTI

- **SCALARE**  
– L'ATTRIBUTO HA UN SOLO VALORE



- **MULTIPLO**  
– L'ATTRIBUTO AMMETTE PIU' VALORI



(min, Max) RAPPRESENTANO LA CARDINALITA'  
MINIMA E MASSIMA DELL'ATTRIBUTO

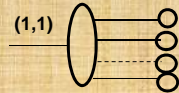
© Fabio A. Schreiber

Progettazione 29

## ATTRIBUTI

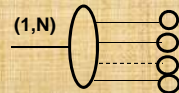
- **COMPOSTO**

– LA PROPRIETA' E' A SUA VOLTA FORMATA DA ELEMENTI PIU' SEMPLICI



DATA (gg, mm, aaaa)  
INDIRIZZO (spec. topon., denomin., civico, cap, citta')

- **MULTIPLO COMPOSTO**



TELEFONO (stato, citta', numero, interno)

© Fabio A. Schreiber

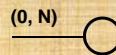
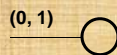
Progettazione 30

## ATTRIBUTI

- **OPZIONALE**

– SI AMMETTE CHE IL VALORE POSSA **NON ESISTERE**

- DIVERSO DA sconosciuto
- DIVERSO DA non applicabile
- DIVERSO DA valore = 0 !!!



© Fabio A. Schreiber

Progettazione 31

## MONDO REALE E MODELLO E-R

ENTITA', RELAZIONI E ATTRIBUTI NON SONO FATTI ASSOLUTI, MA **DIPENDONO DAL CONTESTO APPLICATIVO**

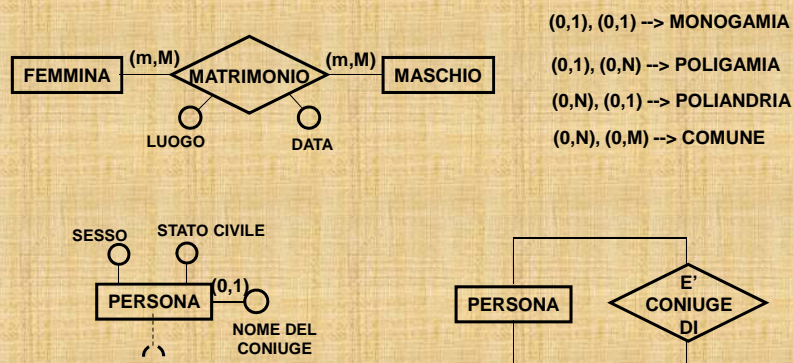
- SE IL CONCETTO E' **SIGNIFICATIVO** PER IL CONTESTO APPLICATIVO --> **ENTITA'**
- SE IL CONCETTO E' **MARGINALE** E DESCRIVIBILE IN MODO **SEMPLICE** --> **ATTRIBUTO**
- SE IL CONCETTO DEFINISCE UN **LEGAME** TRA ENTITA' --> **RELAZIONE**

© Fabio A. Schreiber

Progettazione 32

## ESEMPI DI SCHEMI

E' POSSIBILE CHE LA **STESSA REALTA'** VENGA MODELLATA CON **SCHEMI DIVERSI**



© Fabio A. Schreiber

Progettazione 33

## ASTRAZIONI

- UN **MODELLO E' UN'ASTRAZIONE** DELLA REALTA'
- NEL CASO DEL MODELLO E-R E' UTILE CONSIDERARE TRE TIPI DI ASTRAZIONI
  - **CLASSIFICAZIONE** (is-member-of)
  - **AGGREGAZIONE** (is-part-of)
  - **GENERALIZZAZIONE/SPECIALIZZAZIONE** (is-a)

© Fabio A. Schreiber

Progettazione 34

## CLASSIFICAZIONE (is-member-of)

- DEFINISCE UNA **CLASSE DI OGGETTI** CON PROPRIETA' COMUNI
- **IDENTIFICA UN'ENTITA'** ATTRAVERSO I SUOI ATTRIBUTI

### ESEMPI

ATTRIBUTO: colore = CLASSE {rosso, giallo, verde, ...}

ATTRIBUTO: tipo = CLASSE {corsa, turismo, montagna, ...}

ATTRIBUTO: proprietario = CLASSE {Pietro, Giuseppe, ...}

ENTITA': bicicletta = CLASSE {colore, tipo, proprietario}

© Fabio A. Schreiber

Progettazione 35



## AGGREGAZIONE (is-part-of)

- DEFINISCE UNA **NUOVA CLASSE** A PARTIRE DA ALTRE CLASSI COMPONENTI
- **IDENTIFICA UN'ENTITA'** ATTRAVERSO I SUOI COMPONENTI

### ESEMPI

ENTITA': ruota

ENTITA': telaio

ENTITA': sellino

ENTITA': bicicletta = AGGREGAZIONE {ruota, telaio, sellino}

© Fabio A. Schreiber

Progettazione 36

## SOTTOINSIEMI DI UN INSIEME

### COPERTURA

#### – RISPETTO ALLA **PARTECIPAZIONE**

- **TOTALE (t)**

- TUTTI GLI ELEMENTI DELL'INSIEME FANNO PARTE DI UN SOTTOINSIEME

- **PARZIALE (p)**

- ALCUNI ELEMENTI POSSONO NON FAR PARTE DI ALCUN SOTTOINSIEME

#### – RISPETTO ALLA **SOVRAPPOSIZIONE**

- **ESCLUSIVA (e)**

- OGNI ELEMENTO PUO' FAR PARTE DI UN SOLO SOTTOINSIEME

- **SOVRAPPOSTA (s)**

- UN ELEMENTO PUO' FAR PARTE DI PIU' SOTTOINSIEMI

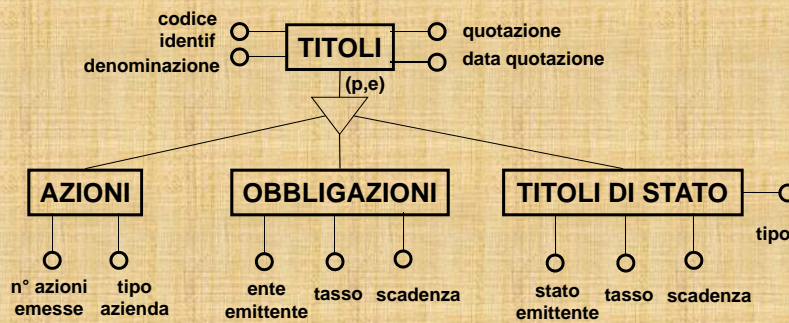
UNA SUDDIVISIONE (t,e) SI DICE **PARTIZIONE** DELL'INSIEME

© Fabio A. Schreiber

Progettazione 37

## GENERALIZZAZIONE (is-a)

- DEFINISCE UNA CLASSE COME **GERARCHIA DI SOTTOINSIEMI**



GLI ATTRIBUTI DEL PADRE VENGONO **EREDITATI** DAI FIGLI

© Fabio A. Schreiber

Progettazione 38

## IDENTIFICATORE (CHIAVE)

UN IDENTIFICATORE, COSTITUITO DA UNO O PIU' ATTRIBUTI, CARATTERIZZA **IN MODO UNIVOCO** LA SINGOLA ISTANZA DI ENTITA' O DI ASSOCIAZIONE

- TOTALE, OBBLIGATORIO, UNICO, ESPLICITO
- PUO' ESSERE COMPOSTO, MA NON DEVE AVERE null
- IN GENERE NON E' MODIFICABILE
- L'IDENTIFICATORE DI UN'ASSOCIAZIONE E' COSTITUITO **DALL'UNIONE DEGLI IDENTIFICATORI DELLE ENTITA'** CHE PARTECIPANO

id ●—

ESEMPI

c.f. ●— DIPENDENTE

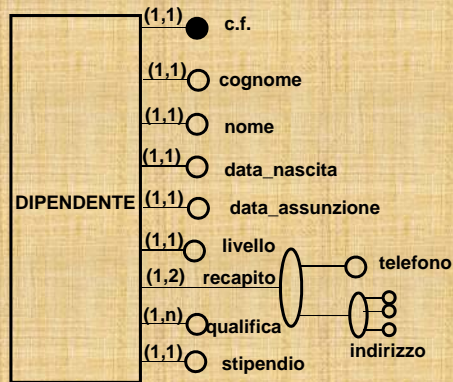
n° telaio ●— ATOMOBILE

ISBN ●— LIBRO

© Fabio A. Schreiber

Progettazione 39

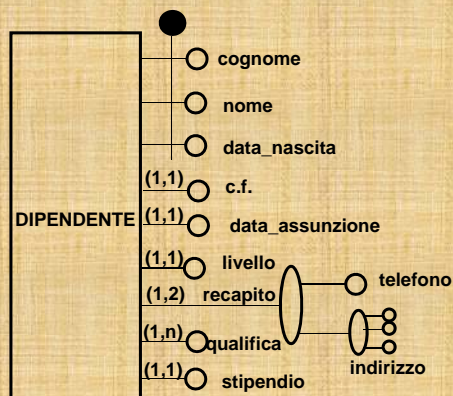
## IDENTIFICATORE (CHIAVE)



© Fabio A. Schreiber

Progettazione 40

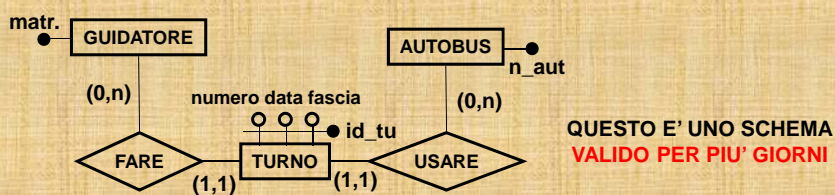
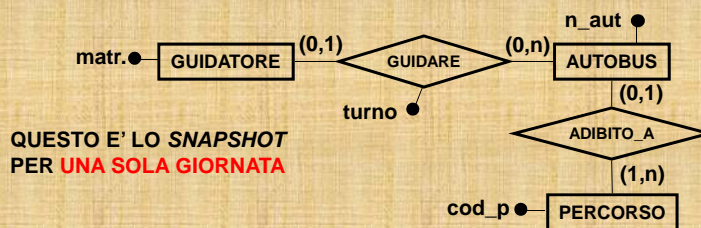
## IDENTIFICATORE COMPOSTO



© Fabio A. Schreiber

Progettazione 41

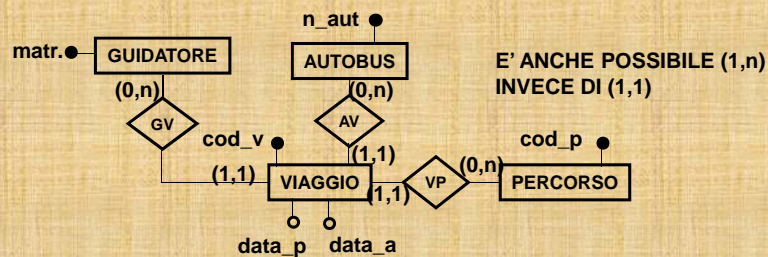
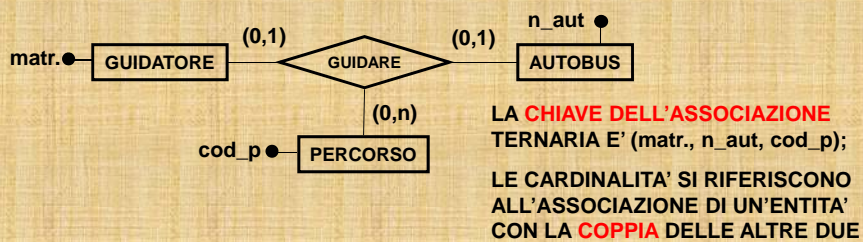
## ESEMPI DI SCHEMI



© Fabio A. Schreiber

Progettazione 42

## ESEMPI DI SCHEMI

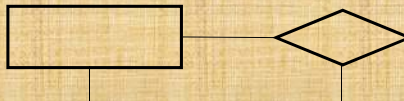


© Fabio A. Schreiber

Progettazione 43

## AUTO-ASSOCIAZIONI (RELAZIONI UNARIE)

- SONO ASSOCIAZIONI AVENTI COME PARTECIPANTI **ISTANZE PROVENIENTI DALLA STESSA ENTITA'**

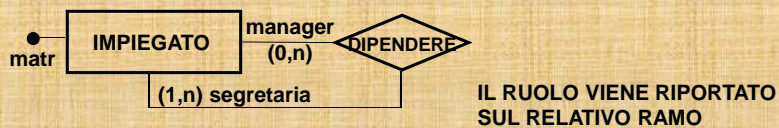


- VENGONO ANCHE CHIAMATE RELAZIONI DI **RUOLO** O AD ANELLO

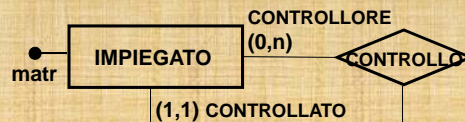
© Fabio A. Schreiber

Progettazione 44

## AUTO-ASSOCIAZIONI (RELAZIONI UNARIE)



ASSOCIAZIONE **RICORSIVA**,  
GERARCHIA (1:n)



© Fabio A. Schreiber

Progettazione 45



## RIDONDANZE NEGLI SCHEMI

- LA SPECIFICA DI UNA BASE DI DATI ESPRESSA IN **LINGUAGGIO NATURALE** PUO' PORTARE ALL'**INTRODUZIONE DI RIDONDANZA**
- LE RIDONDANZE NEGLI SCHEMI DEVONO ESSERE **IDENTIFICATE ED ELIMINATE** MODIFICANDOLI OPPORTUNAMENTE

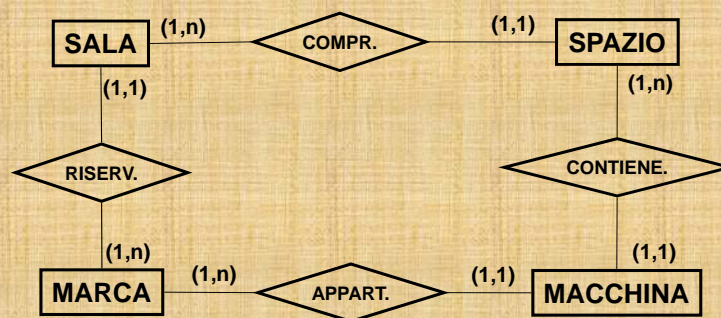
### ESEMPIO

- UNA SOCIETA' GESTISCE SALE PER ESPOSIZIONI
- LE SALE SONO RISERVATE A MARCHE DI MACCHINE
- LE SALE COMPRENDONO SPAZI DI ESPOSIZIONE
- GLI SPAZI CONTENGONO MACCHINE
- LE MACCHINE APPARTENGONO AD UNA CERTA MARCA

© Fabio A. Schreiber

Progettazione 46

## RIDONDANZE NEGLI SCHEMI



POTREBBE ESSERCI RIDONDANZA

© Fabio A. Schreiber

Progettazione 47

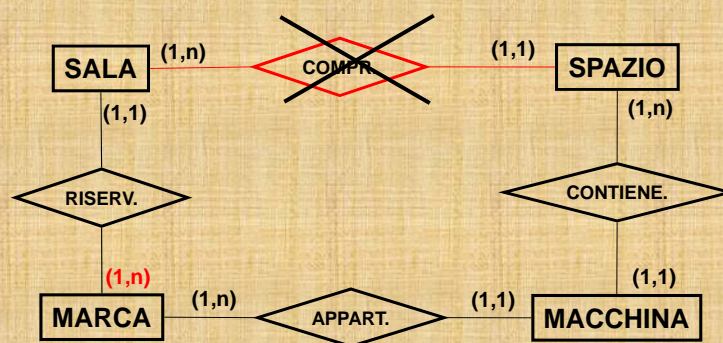
## RIDONDANZE NEGLI SCHEMI

- **VINCOLO**
  - UNA SALA CONTIENE UNA SOLA MARCA
- SE IL VINCOLO PER LA SISTEMAZIONE DELLE MACCHINE NEGLI SPAZI VIENE RISPETTATO **IL CICLO E' RIDONDANTE**
- NON E' POSSIBILE ELIMINARE UN LATO **A CASO** SENZA PERDERE INFORMAZIONE

© Fabio A. Schreiber

Progettazione 48

## RIDONDANZE NEGLI SCHEMI

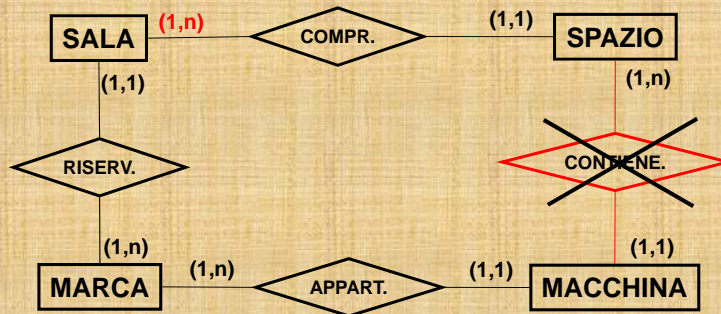


**DATO UNO SPAZIO NON E' PIU' POSSIBILE RISALIRE ALLA SALA DI APPARTENENZA (CI POSSONO ESSERE PIU' SALE DEDICATE ALLA STESSA MARCA)**

© Fabio A. Schreiber

Progettazione 49

## RIDONDANZE NEGLI SCHEMI

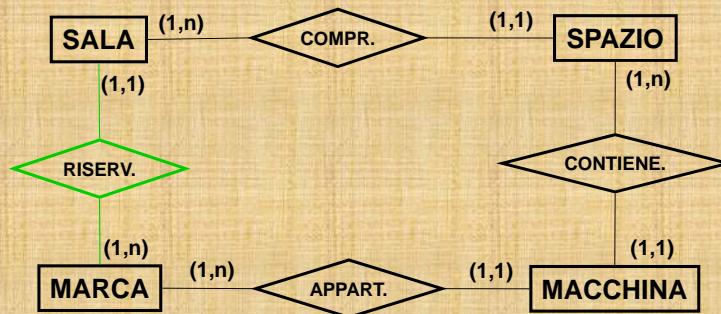


**DATA UNA MACCHINA NON E' PIU' POSSIBILE RISALIRE ALLO SPAZIO CHE LA CONTIENE (CI SONO PIU' SPAZI IN UNA SALA)**

© Fabio A. Schreiber

Progettazione 50

## RIDONDANZE NEGLI SCHEMI

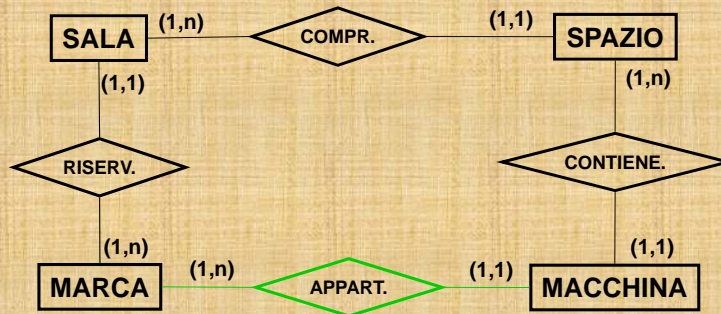


**L'INFORMAZIONE VIENE PRESERVATA**

© Fabio A. Schreiber

Progettazione 51

## RIDONDANZE NEGLI SCHEMI



L'INFORMAZIONE VIENE PRESERVATA

© Fabio A. Schreiber

Progettazione 52

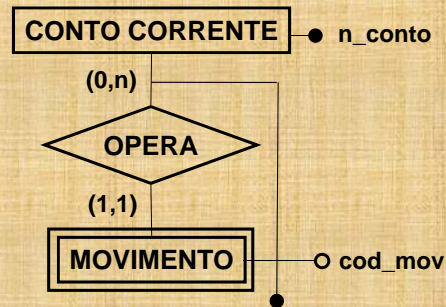
## ENTITA' DEBOLI

- CONTENGONO ISTANZE ACCETTATE DAL SISTEMA SOLO **IN PRESENZA DI DETERMINATE ISTANZE DI ALTRE ENTITA'** DALLE QUALI ESSE DIPENDONO
- IN CASO DI ELIMINAZIONE DELL'ISTANZA DI RIFERIMENTO DEVONO **ESSERE ELIMINATE ANCHE LE ISTANZE DEBOLI** AD ESSA COLLEGATE (**DIPENDENZA ESISTENZIALE**)
- L'IDENTIFICATORE DELL'ENTITA' DEBOLE DEVE CONTENERE L'IDENTIFICATORE DELL'ENTITA' DALLA QUALE DIPENDE

© Fabio A. Schreiber

Progettazione 53

## ENTITA' DEBOLI



IDENTIFICATORE = {n\_conto, cod\_mov}

© Fabio A. Schreiber

Progettazione 54

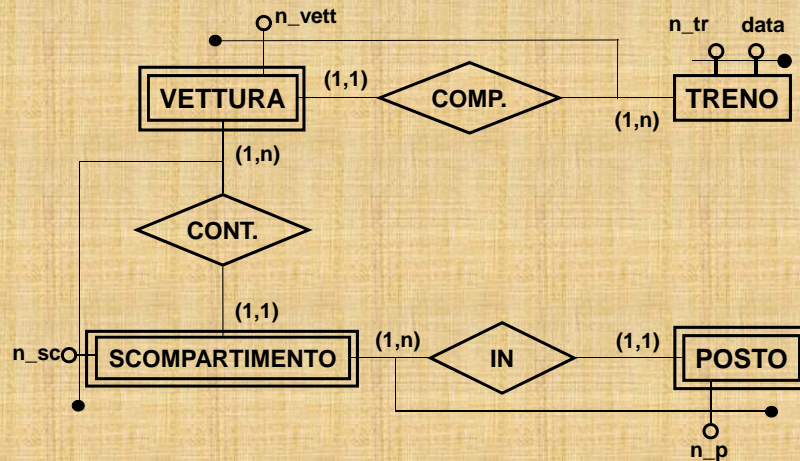
## ESEMPIO DI ENTITA' DEBOLI

- I TRENI SONO IDENTIFICATI DA UN NUMERO E DA UNA DATA E SONO COMPOSTI DA VETTURE CHE CONTENGONO I POSTI DA PRENOTARE
- LE VETTURE SONO NUMERATE E I POSTI SONO NUMERATI ALLO STESSO MODO IN OGNI VETTURA
- SE LE VETTURE HANNO SCOMPARTIMENTI, I POSTI STANNO ALL'INTERNO DI UNO SCOMPARTIMENTO

© Fabio A. Schreiber

Progettazione 55

## ESEMPIO DI ENTITA' DEBOLI



SE VIENE CANCELLATO IL TRENO, VENGONO CANCELLATI ANCHE TUTTI I POSTI

© Fabio A. Schreiber

Progettazione 56

## LE GERARCHIE

- ALCUNE ENTITA' POSSONO ESSERE **MOLTO SIMILI O CASI PARTICOLARI** L'UNA DELL'ALTRA
- RISULTA UTILE **"PORRE A FATTOR COMUNE" LE CARATTERISTICHE COMUNI E EVIDENZIARE I MOTIVI DI DISTINZIONE**
- IL LEGAME LOGICO CHE UNISCE LA CLASSE DEGLI ATTRIBUTI COMUNI (**PADRE**) CON LE SOTTOCLASSI SPECIALIZZATE (**FIGLI**) E' LA **GERARCHIA DI SPECIALIZZAZIONE**

© Fabio A. Schreiber

Progettazione 57



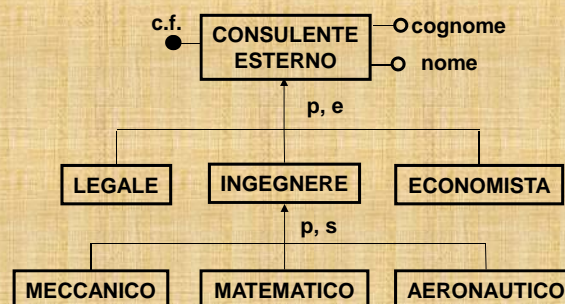
## LE GERARCHIE

- OGNI ISTANZA DI UN'ENTITA' FIGLIO E' **NECESSARIAMENTE** ANCHE ISTANZA DELL'ENTITA' PADRE
- UN'ISTANZA DELL'ENTITA' PADRE **PUO'** ESSERE ISTANZA DI **UNA O PIU'** ENTITA' FIGLIO
- NEL CASO DI **PARTIZIONE** ANCHE OGNI ISTANZA DELL'ENTITA' PADRE E' **NECESSARIAMENTE** ISTANZA DI UNA E UNA SOLA ENTITA' FIGLIO

© Fabio A. Schreiber

Progettazione 58

## ESEMPIO DI GERARCHIA



© Fabio A. Schreiber

Progettazione 59

## EREDITARIETA' DELLE PROPRIETA'

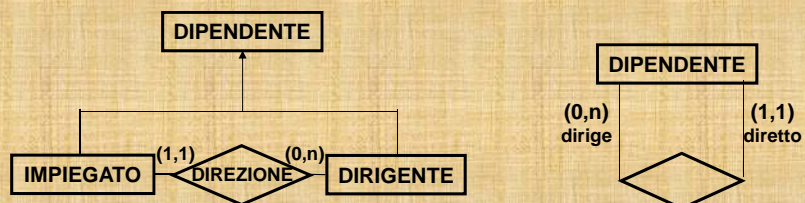
- GLI ATTRIBUTI DELL'ENTITA' PADRE NON DEVONO ESSERE REPLICATI SULLE ENTITA' FIGLIE
  - GLI ATTRIBUTI DELL'ENTITA' PADRE VENGONO **EREDITATI DALLE ENTITA' FIGLIE** E FANNO PARTE DEL SUO TIPO
  - **NON E' VERO IL VICEVERSA**

© Fabio A. Schreiber

Progettazione 60

## GERARCHIE ISA

LA GERARCHIA ISA CONTIENE PIU' INFORMAZIONE DELL'ASSOCIAZIONE UNARIA RICORSIVA



© Fabio A. Schreiber

Progettazione 61

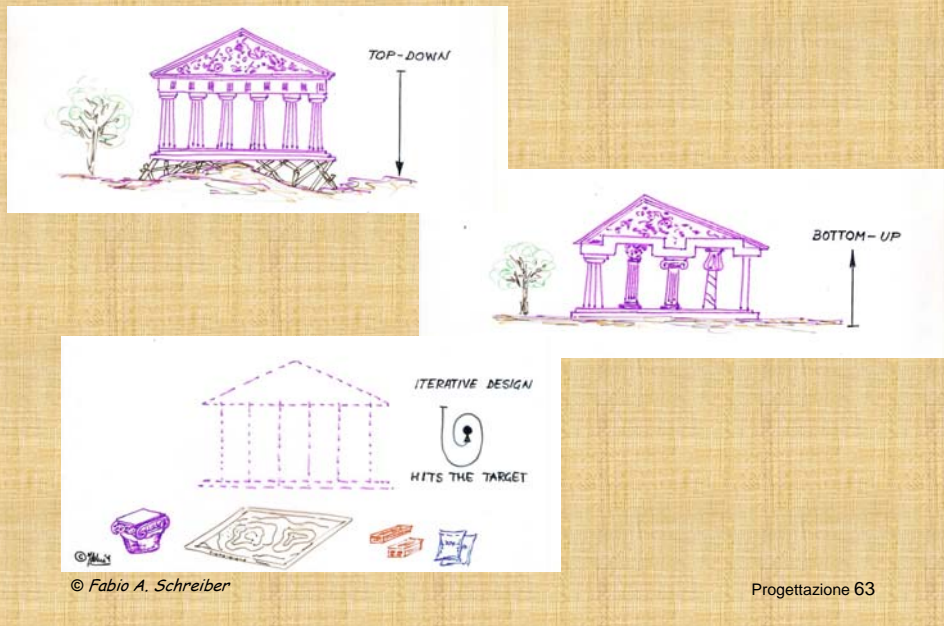
## STRATEGIE DI PROGETTO

- TOP - DOWN
- BOTTOM - UP
- INSIDE - OUT
- MISTA

© Fabio A. Schreiber

Progettazione 62

## STRATEGIE DI PROGETTO



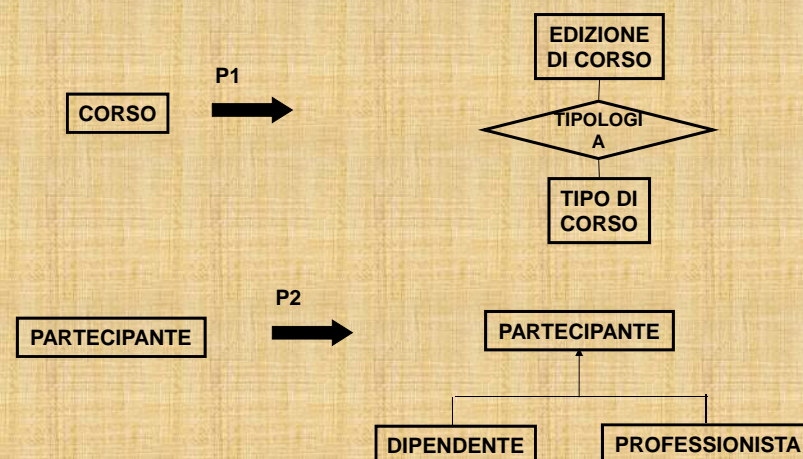
## STRATEGIA TOP - DOWN

- SI COSTRUISCE UNO **SCHEMA INIZIALE** A PARTIRE DALLE SPECIFICHE
- PER **RAFFINAMENTI SUCCESSIVI** SI ARRIVA A SCHEMI INTERMEDI E QUINDI ALLO SCHEMA FINALE
- LA TRASFORMAZIONE PREVEDE L'USO DI **PRIMITIVE ELEMENTARI** CHE OPERANO SUL SINGOLO CONCETTO PER DESCRIVERLO CON MAGGIORE DETTAGLIO

© Fabio A. Schreiber

Progettazione 64

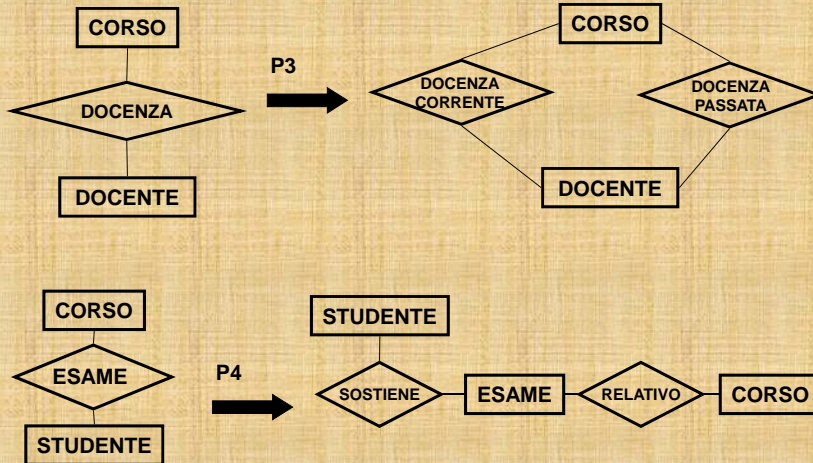
## PRIMITIVE DI TRASFORMAZIONE



© Fabio A. Schreiber

Progettazione 65

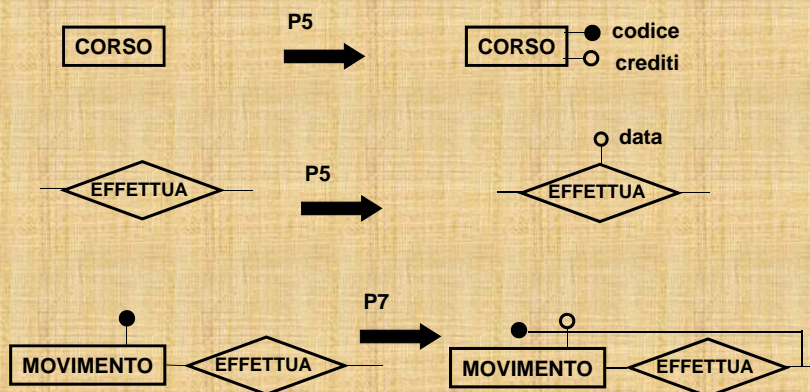
## PRIMITIVE DI TRASFORMAZIONE



© Fabio A. Schreiber

Progettazione 66

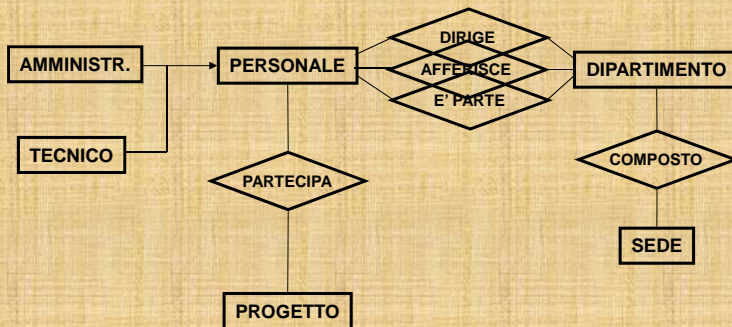
## PRIMITIVE DI TRASFORMAZIONE



© Fabio A. Schreiber

Progettazione 67

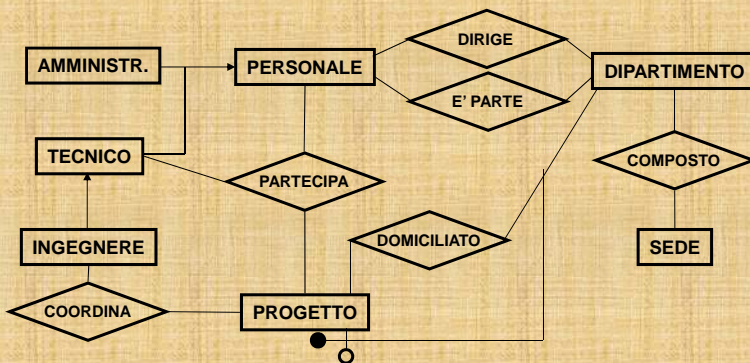
## ESEMPIO DI SVILUPPO TOP-DOWN



© Fabio A. Schreiber

Progettazione 68

## ESEMPIO DI SVILUPPO TOP-DOWN



© Fabio A. Schreiber

Progettazione 69



## STRATEGIA TOP-DOWN

- **VANTAGGI**
  - IL PROGETTISTA DESCRIVE INIZIALMENTE LO SCHEMA TRASCURANDO I DETTAGLI
  - LE PRECISAZIONI AVVENGONO GRADUALMENTE APPROFONDENDO L'ANALISI
- **PROBLEMA**
  - NON SI ADATTA AD APPLICAZIONI MOLTO COMPLESSE POICHE' E' DIFFICILE AVERE SIN DALL'INIZIO UNA VISIONE GLOBALE SUFFICIENTEMENTE PRECISA DI TUTTE LE PARTI DEL SISTEMA

© Fabio A. Schreiber

Progettazione 70

## STRATEGIA BOTTOM-UP

- LE SPECIFICHE NASCONO **SUDDIVISE PER SOTTOPROGETTI** DESCRIVENTI FRAMMENTI LIMITATI DELLA REALTA' DA MODELLARE
- SI SVILUPPANO **SOTTOSCHEMI SEPARATI**
- SI **FONDONO I SOTTOSCHEMI** PER OTTENERE LO SCHEMA FINALE

© Fabio A. Schreiber

Progettazione 71

## STRATEGIA BOTTOM-UP

- **VANTAGGI**
  - DIVERSI PROGETTISTI ELABORANO GLI SCHEMI PARZIALI, OGNUNO AVENDO UNA VISIONE PIU' PRECISA DEL PROPRIO SETTORE
  - SI ADATTA ANCHE AD APPLICAZIONI MOLTO COMPLESSE
- **PROBLEMA**
  - SI POSSONO INCONTRARE CONFLITTI E DIFFICOLTA' DI INTEGRAZIONE NON FACILI DA RISOLVERE

© Fabio A. Schreiber

Progettazione 72

## ALTRE STRATEGIE

- **INSIDE-OUT**

VARIANTE DELLA BOTTOM-UP NELLA QUALE SI SVILUPPANO SCHEMI PARZIALI IN AGGIUNTA A SOTTOSCHEMI GIA' DEFINITI PRECEDENTEMENTE E SEPARATAMENTE
- **MISTA**
  - SI PARTE DA UNO SCHEMA SCHELETRO GENERALE
  - LO SI SUDDIVIDE IN SOTTOSCHEMI DA DETTAGLIARE
  - SI FONDONO I SOTTOSCHEMI

IN LINEA DI MASSIMA E' CONVENIENTE ADDOTTARE A POSTERIORI UNA SOLUZIONE **TOP-DOWN COME DOCUMENTAZIONE**

© Fabio A. Schreiber

Progettazione 73

## PRORPIETA' GENERALI DI UNO SCHEMA E-R

- **COMPLETEZZA**  
RISPETTO ALL'APPLICAZIONE
- **CORRETTEZZA**
  - SINTATTICA
  - SEMANTICA
- **LEGGIBILITA'**
  - GRAFICA
    - MINIMIZZARE CURVE E INCROCI
    - FARE IN MODO CHE IL GRAFO SIA PLANARE
    - DISPORRE GLI ELEMENTI SU UNA GRIGLIA
  - CONCETTUALE

© Fabio A. Schreiber

Progettazione 74

## PRORPIETA' GENERALI DI UNO SCHEMA E-R

- **MINIMALITA'**  
ASSENZA DI CICLI RIDONDANTI
- **AUTO-ESPLICATIVITA'**  
MINIMIZZARE LA DOCUMENTAZIONE DI SUPPORTO

© Fabio A. Schreiber

Progettazione 75

## PROGETTO LOGICO

© Fabio A. Schreiber

Progettazione 76

## PROGETTO LOGICO

- **LO SCHEMA E-R DESCRIVE IL DOMINIO APPLICATIVO AD UN CERTO LIVELLO DI ASTRAZIONE**
  - FORNISCE UNA **DESCRIZIONE SINTETICA VISIVA**
  - RAPPRESENTA BUONA PARTE DELLA **SEMANTICA** DELL'APPLICAZIONE
  - E' UTILE PER **SCAMBIARE INFORMAZIONI** TRA I MEMBRI DELL'UNITA' DI PROGETTO E PER MANTENERE LA **DOCUMENTAZIONE**

© Fabio A. Schreiber

Progettazione 77

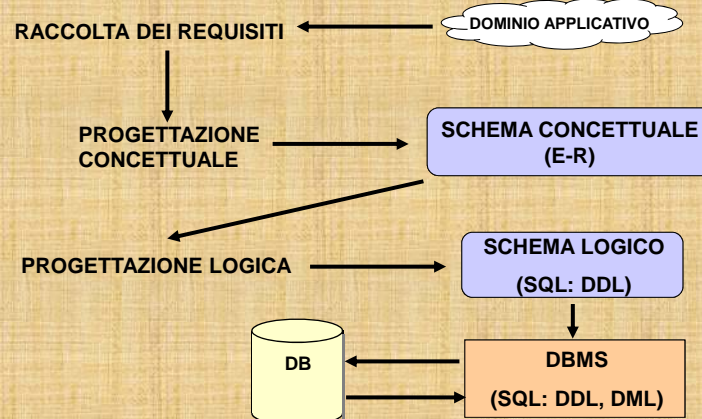
## PROGETTO LOGICO

- NON ESISTONO DBMS IN GRADO DI OPERARE **DIRETTAMENTE** A PARTIRE DAGLI SCHEMI **E-R**
  - OCCORRE TRADURLI IN **ALTRI FORMALISMI** (SCHEMI LOGICI)
  - LA TRADUZIONE PUO' ESSERE FATTA IN MODO **SEMIAUTOMATICO**
  - LE SCELTE ALTERNATIVE DEVONO TENER CONTO DELL'**EFFICIENZA DELLO SCHEMA LOGICO** RISULTANTE E DELLE **OPERAZIONI DA EFFETTUARE**

© Fabio A. Schreiber

Progettazione 78

## PROCESSO DI PROGETTAZIONE



© Fabio A. Schreiber

Progettazione 79

## PROGETTO LOGICO

SI POSSONO INDIVIDUARE ALCUNE **LINEE GUIDA** PER LE SCELTE DI PROGETTO

- CONSIDERARE LE **PROPRIETA' LOGICHE COMUNQUE PREVALENTI** RISPETTO A CONSIDERAZIONI DI EFFICIENZA
- TENERE **SULLA STESSA ENTITA' INFORMAZIONI CHE VERRANO FREQUENTEMENTE CONSULTATE ASSIEME**
- TENERE SU **ENTITA' SEPARATE** INFORMAZIONI DESTINATE AD ESSERE **CONSULTATE SEPARATAMENTE**
- **LIMITARE L'INCIDENZA DI VALORI NULLI PER ATTRIBUTI OPZIONALI**

© Fabio A. Schreiber

Progettazione 80

## FASI DEL PROGETTO LOGICO

LE TRASFORMAZIONI E LE TRADUZIONI TRA LO SCHEMA E-R E LO SCHEMA LOGICO AVVENGONO PER FASI

- 1 - ELIMINAZIONE DELLE GERARCHIE ISA
- 2 - SELEZIONE DELLE CHIAVI PRIMARIE
  - ELIMINAZIONE DELLE IDENTIFICAZIONI ESTERNE
- 3 - NORMALIZZAZIONE DEGLI ATTRIBUTI COMPOSTI O MULTIPLI
- 4 - TRADUZIONE DI ENTITA' E ASSOCIAZIONI IN SCHEMI DI RELAZIONI
- 5 - VERIFICA DI NORMALIZZAZIONE

© Fabio A. Schreiber

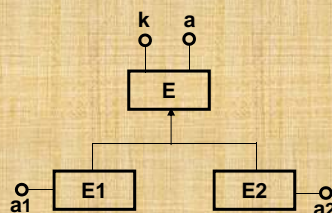
Progettazione 81



## ELIMINAZIONE DELLE GERARCHIE

IL MODELLO RELAZIONALE E' "PIATTO" E **NON PUO' RAPPRESENTARE LE GERARCHIE CHE SONO SOSTITuite DA ENTITA' E ASSOCIAZIONI**

- MANTENIMENTO DELLE ENTITA' CON ASSOCIAZIONI
- COLLASSO VERSO L'ALTO
- COLLASSO VERSO IL BASSO



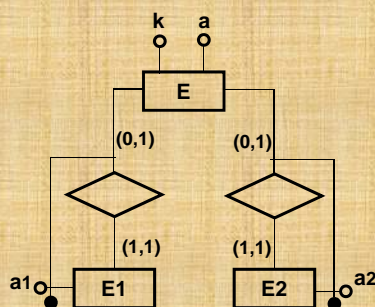
L'APPLICABILITA' E LA CONVENIENZA DI CIASCUNA SOLUZIONE DIPENDONO DALLE **PROPRIETA' DI COPERTURA E DALLE OPERAZIONI PREVISTE**

© Fabio A. Schreiber

Progettazione 82

## MANTENIMENTO DELLE ENTITA'

- TUTTE LE ENTITA' VENGONO MANTENUTE
- LE ENTITA' FIGLIE SONO ASSOCIATE ALL'ENTITA' PADRE
- LE ENTITA' FIGLIE SONO IDENTIFICATE ESTERNAMENTE TRAMITE L'ASSOCIAZIONE

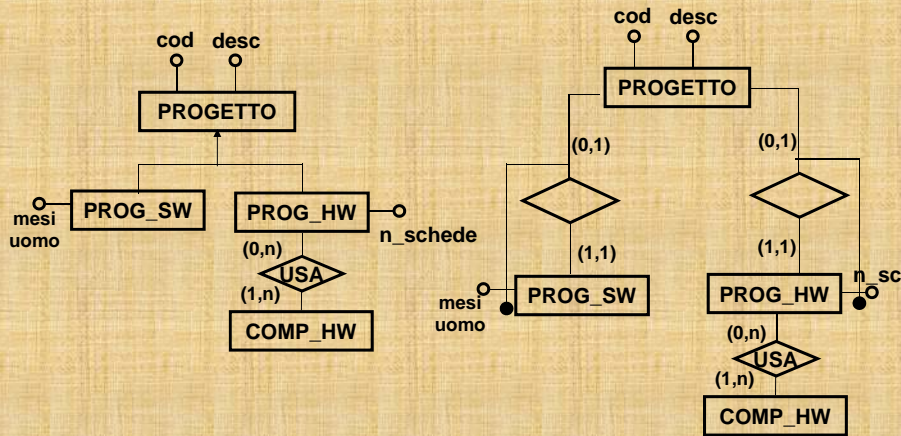


SOLUZIONE **SEMPRE POSSIBILE** INDIPENDENTEMENTE DALLA COPERTURA

© Fabio A. Schreiber

Progettazione 83

## MANTENIMENTO DELLE ENTITA'

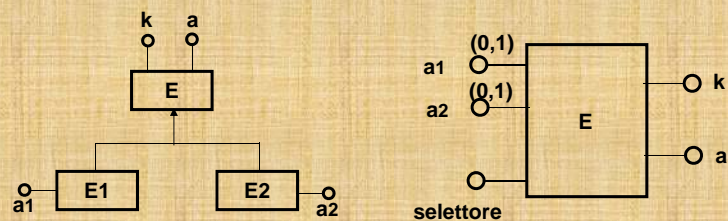


© Fabio A. Schreiber

Progettazione 84

## ELIMINAZIONE DELLE GERARCHIE

IL COLLASSO VERSO L'ALTO RIUNISCE TUTTE LE ENTITA' FIGLIE NELL'ENTITA' PADRE



selettore E' UN ATTRIBUTO CHE SPECIFICA SE UNA Istanza DI E APPARTIENE A UNA DELLE SOTTOENTITA'

© Fabio A. Schreiber

Progettazione 85

## COLLASSO VERSO L'ALTO

- FAVORISCE LE OPERAZIONI CHE **CONSULTANO INSIEME** GLI ATTRIBUTI DELL'ENTITA' PADRE E QUELLI DI UN'ENTITA' FIGLIA
  - SI ACCEDE AD UNA SOLA ENTITA' INVECE CHE A DUE ATTRAVERSO UN'ASSOCIAZIONE
- GLI ATTRIBUTI **OBBLIGATORI PER LE ENTITA' FIGLIE** DIVENTANO **OPZIONALI PER L'ENTITA' PADRE**

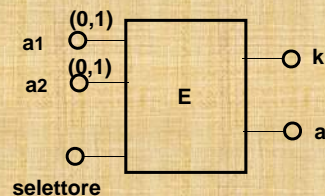
© Fabio A. Schreiber

Progettazione 86

## COLLASSO VERSO L'ALTO

### COPERTURA DELL'ISA

- **TOTALE, ESCLUSIVA**  
selettore HA n VALORI QUANTE SONO LE SOTTOENTITA'
- **PARZIALE ESCLUSIVA**  
selettore HA n+1 VALORI PER POTER SEGNALARE LE ISTANZE CHE NON APPARTENGONO AD ALCUNA SOTTOENTITA'



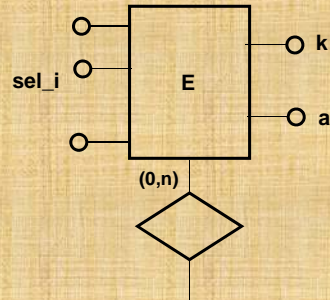
© Fabio A. Schreiber

Progettazione 87

## COLLASSO VERSO L'ALTO

### COPERTURA DELL'ISA

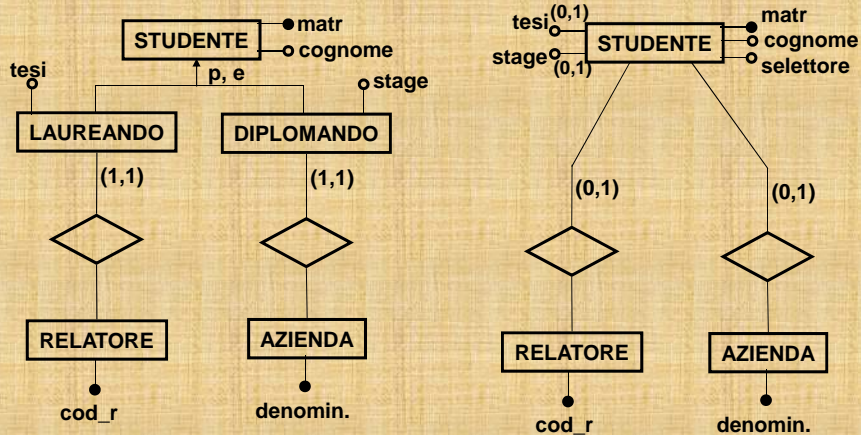
- **SOVRAPPOSTA, TOTALE**
  - OCCORRONO TANTI selettori BOOLEANI QUANTE SONO LE SOTTOENTITA'
  - $sel_i$  E' vero SE L'ISTANZA APPARTIENE ALLA  $i$ -esima SOTTOENTITA'
- **SOVRAPPOSTA, PARZIALE**
  - I selettori POSSONO ESSERE ANCHE TUTTI falsi
- LE ASSOCIAZIONI CONNESSE ALLE SOTTOENTITA' SI TRASPORTANO SU  $E$
- LE EVENTUALI CARDINALITA' MINIME DIVENTANO 0



© Fabio A. Schreiber

Progettazione 88

## COLLASSO VERSO L'ALTO



IL DOMINIO DI selettore E' (l, d, null)

© Fabio A. Schreiber

Progettazione 89

## COLLASSO VERSO IL BASSO

SI ELIMINA L'ENTITA' PADRE TRASFERENDONE GLI ATTRIBUTI **SU TUTTE** LE ENTITA' FIGLIE

- UN'ASSOCIAZIONE DEL PADRE E' **REPLICATA** TANTE VOLTE QUANTE SONO LE ENTITA' FIGLIE
- E' CONVENIENTE QUANDO CI SIANO **MOLTI ATTRIBUTI DI SPECIALIZZAZIONE**
  - VERSO L'ALTO SI AVREBBE UN ECCESSO DI VALORI NULLI
- FAVORISCE LE OPERAZIONI NELLE QUALI SI ACCEDA **SEPARATAMENTE ALLE ENTITA' FIGLIE**

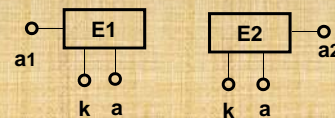
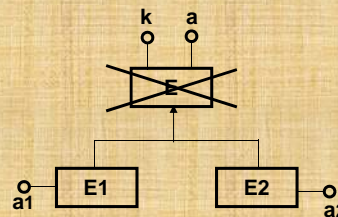
© Fabio A. Schreiber

Progettazione 90

## COLLASSO VERSO IL BASSO

LIMITI DI APPLICABILITA'

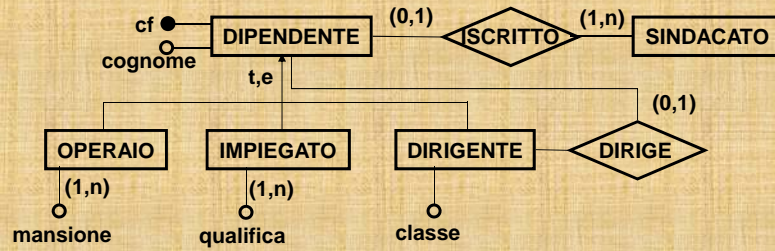
- **COPERTURA PARZIALE**
  - NON SI PUO' FARE  
NON SI SAPREBBE DOVE METTERE LE ENTITA' CHE NON APPARTENGONO A E1 O A E2
- **COPERTURA SOVRAPPOSTA**
  - INTRODUCE RIDONDANZA  
GLI ATTRIBUTI DI E VENGONO RIPETUTI PER TUTTE LE ISTANZE CHE APPARTENGONO A PIU' SOTTOENTITA'



© Fabio A. Schreiber

Progettazione 91

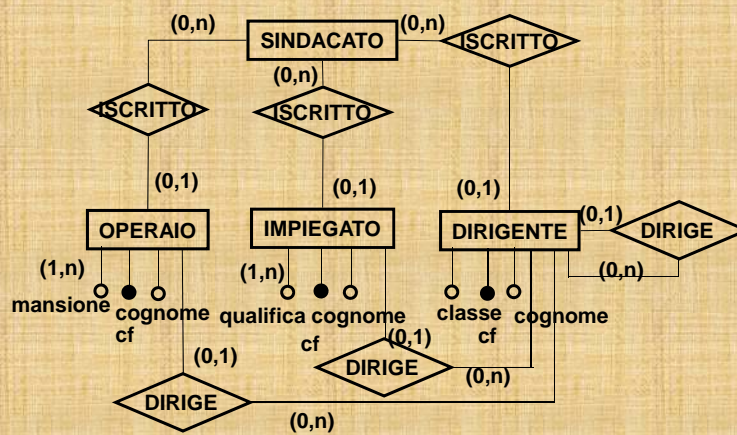
## COLLASSO VERSO IL BASSO



© Fabio A. Schreiber

Progettazione 92

## COLLASSO VERSO IL BASSO



© Fabio A. Schreiber

Progettazione 93



## SCELTA DELLA CHIAVE PRIMARIA

TRA I POSSIBILI IDENTIFICATORI DI UN'ENTITA' E' NECESSARIO SCEGLIERE UNA **CHIAVE PRIMARIA**

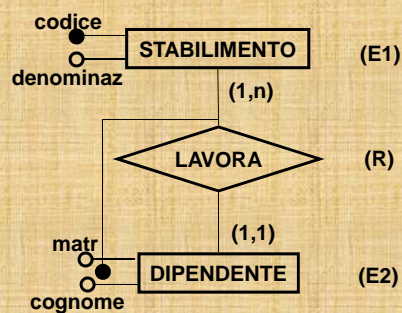
- IL DBMS DEVE POTER **GARANTIRNE L'UNICITA' DEI VALORI**
- CRITERI EURISTICI DI SCELTA
  - SCEGLIERE L'IDENTIFICATORE **USATO PIU' FREQUENTEMENTE** PER ACCEDERE ALL'ENTITA'
  - PREFERIRE **CHIAVI SEMPLICI** A CHIAVI COMPOSTE
  - PREFERIRE **CHIAVI INTERNE** A CHIAVI ESTERNE

© Fabio A. Schreiber

Progettazione 94

## IDENTIFICATORI ESTERNI

UNA COMPONENTE DI IDENTIFICAZIONE ESTERNA DI UNA ENTITA'  $E_2$  DA UNA ENTITA'  $E_1$  ATTRAVERSO UNA RELAZIONE  $R$  COMPORTA IL TRASPORTO DELLA CHIAVE PRIMARIA DI  $E_1$  SU  $E_2$



© Fabio A. Schreiber

Progettazione 95

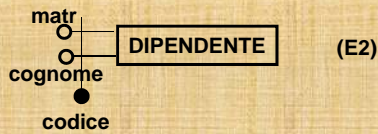
## IDENTIFICATORI ESTERNI

- L'ASSOCIAZIONE VIENE RAPPRESENTATA ATTRAVERSO LA CHIAVE E PUO' ESSERE ELIMINATA



- LA CHIAVE TRASPORTATA E' CHIAVE ESTERNA

- CON PIU' IDENTIFICAZIONI IN CASCATA E' NECESSARIO INIZIARE LA PROPAGAZIONE DALL'ENTITA' CHE NON HA IDENTIFICAZIONI ESTERNE

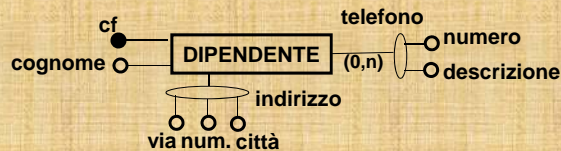


© Fabio A. Schreiber

Progettazione 96

## ATTRIBUTI COMPOSTI O RIPETUTI

- IL MODELLO RELAZIONALE AMMETTE **SOLO ATTRIBUTI ATOMICI**
  - ELIMINAZIONE DEGLI ATTRIBUTI COMPOSTI O RIPETUTI



### DUE POSSIBILI SOLUZIONI

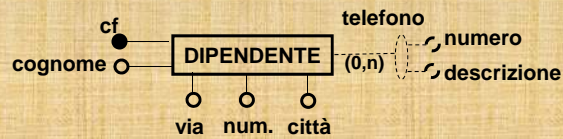
© Fabio A. Schreiber

Progettazione 97

## ATTRIBUTI COMPOSTI

- PRIMA SOLUZIONE

- ELIMINARE L'ATTRIBUTO COMPOSTO E CONSIDERARE I SUOI COMPONENTI COME ATTRIBUTI SEMPLICI
- SI PERDE LA VISIONE UNITARIA
- SI MANTIENE L'ARTICOLAZIONE DEI COMPONENTI



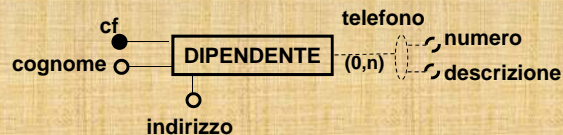
© Fabio A. Schreiber

Progettazione 98

## ATTRIBUTI COMPOSTI

- SECONDA SOLUZIONE

- ELIMINARE I COMPONENTI E CONSIDERARE L'ATTRIBUTO COME SEMPLICE
- LO SCHEMA E' SEMPLIFICATO
- SI PERDE PARTE DEI DETTAGLI
  - L'INDIRIZZO DIVENTA UNA STRINGA MONOLITICA



© Fabio A. Schreiber

Progettazione 99

## ATTRIBUTI RIPETUTI

- SI CREA UNA **NUOVA ENTITA'** CHE CONTIENE L'ATTRIBUTO
- LA NUOVA ENTITA' E' COLLEGATA A QUELLA ORIGINARIA
  - UN VALORE PUO' **COMPARIRE UNA SOLA VOLTA** NELLA RIPETIZIONE
  - UN VALORE PUO' **COMPARIRE PIU' DI UNA VOLTA** NELLA RIPETIZIONE

© Fabio A. Schreiber

Progettazione 100

## ATTRIBUTI RIPETUTI

UN VALORE PUO' **COMPARIRE UNA SOLA VOLTA** NELLA RIPETIZIONE

LA NUOVA ENTITA' HA UN IDENTIFICATORE COMPOSTO DALLA CHIAVE DELL'ENTITA' ORIGINARIA PIU' L'ATTRIBUTO RIPETUTO



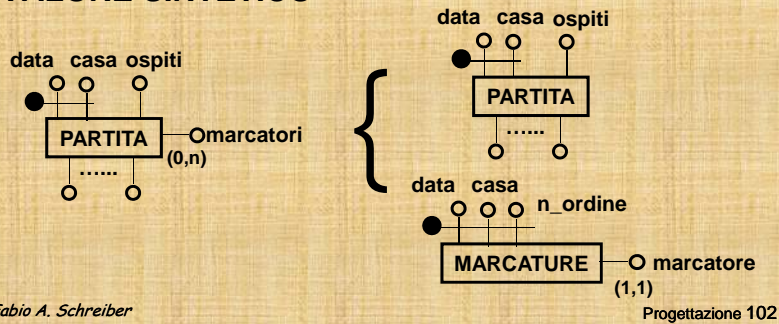
© Fabio A. Schreiber

Progettazione 101

## ATTRIBUTI RIPETUTI

UN VALORE PUO' COMPARIRE **PIU' DI UNA VOLTA** NELLA RIPETIZIONE

LA NUOVA ENTITA' HA LO STESSO IDENTIFICATORE DI QUELLA ORIGINARIA PIU' UN VALORE SINTETICO



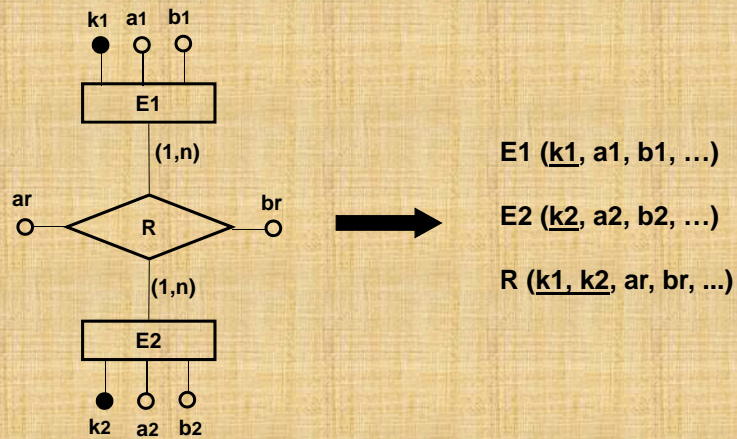
## TRADUZIONE STANDARD

- **OGNI ENTITA' E' TRADOTTA CON UNA RELAZIONE (TABELLA)**
  - CON GLI **STESSI ATTRIBUTI**
  - CON LA **STESSA CHIAVE**
- **OGNI ASSOCIAZIONE E' TRADOTTA CON UNA RELAZIONE (TABELLA)**
  - CON GLI **STESSI ATTRIBUTI**
  - SI **AGGIUNGONO GLI IDENTIFICATORI** DI TUTTE LE ENTITA' COLLEGATE
  - LA **CHIAVE E' COMPOSTA DALLE CHIAVI DELLE ENTITA' COLLEGATE** (POTREBBE ESSERE SUPERCHIAVE)

© Fabio A. Schreiber

Progettazione 103

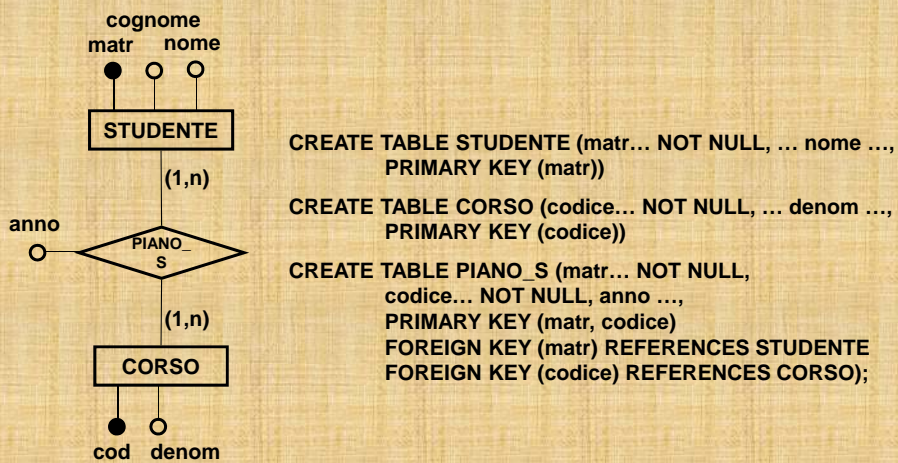
## TRADUZIONE STANDARD



© Fabio A. Schreiber

Progettazione 104

## TRADUZIONE STANDARD



© Fabio A. Schreiber

Progettazione 105



## ALTRE TRADUZIONI

- LA TRADUZIONE STANDARD
  - E' **SEMPRE** POSSIBILE
  - E' **L'UNICA** PER LE ASSOCIAZIONI (n:m)
- ALTRE FORME DI TRADUZIONE SONO POSSIBILI PER LE ASSOCIAZIONI (1.1) E (1:n)
- LE ALTRE FORME DI TRADUZIONE FONDONO IN UNA STESSA RELAZIONE ENTITA' E ASSOCIAZIONI

© Fabio A. Schreiber

Progettazione 106

## ALTRE TRADUZIONI

- DANNO ORIGINE AD UN MINOR NUMERO DI RELAZIONI E QUINDI AD UNO **SCHEMA PIU' SEMPLICE**
- RICHIEDONO UN **MINOR NUMERO DI JOIN** PER LA NAVIGAZIONE ATTRAVERSO UNA ASSOCIAZIONE
- PENALIZZANO LE OPERAZIONI CHE CONSULTANO GLI **ATTRIBUTI DI UNA SOLA ENTITA'** CHE E' STATA FUSA

© Fabio A. Schreiber

Progettazione 107

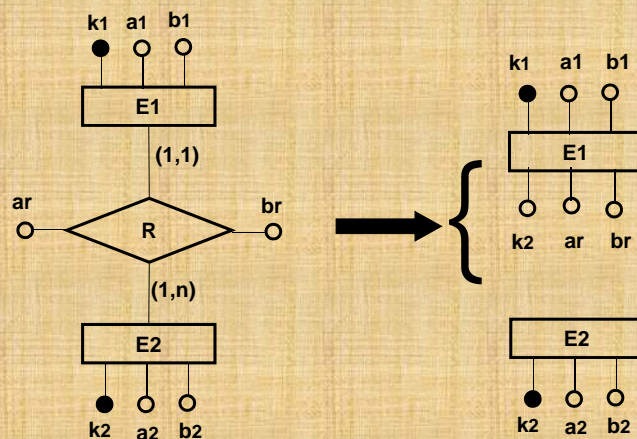
## ASSOCIAZIONE BINARIA 1:n

- SE  $E1$  PARTECIPA CON CARDINALITA' (1,1) **PUO' ESSERE FUSA** CON L'ASSOCIAZIONE  
 $E1$  ( $k1$ ,  $a1$ ,  $b1$ ,  $k2$ ,  $ar$ ,  $br$ )  
 $E2$  ( $k2$ ,  $a2$ ,  $b2$ )
- SE  $E1$  PARTECIPA CON CARDINALITA' (0,1) QUESTA SOLUZIONE **HA VALORI NULLI IN**  
 $k2$ ,  $ar$ ,  $br$  PER LE ISTANZE DI  $E1$  CHE NON  
 PARTECIPANO ALL'ASSOCIAZIONE

© Fabio A. Schreiber

Progettazione 108

## ASSOCIAZIONE BINARIA 1:n



© Fabio A. Schreiber

Progettazione 109

## ASSOCIAZIONE BINARIA 1:n

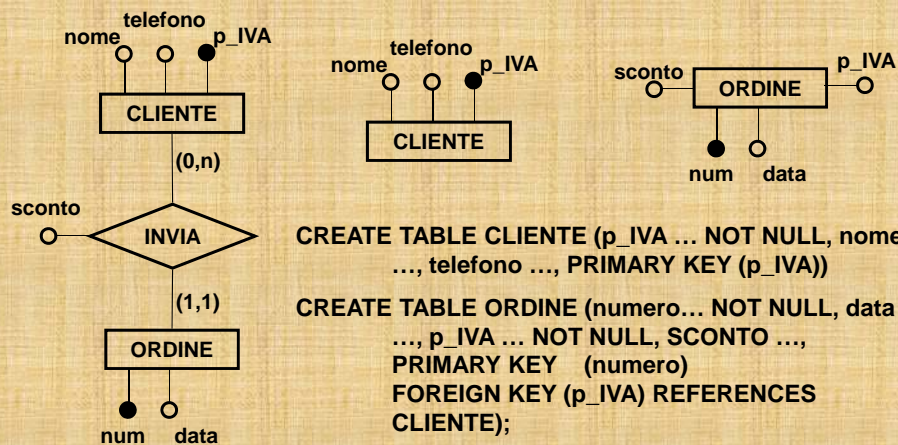
- IN QUESTO CASO LA PARTECIPAZIONE DI E1 E' (0,1) O (0,n)
  - AD UN DATO VALORE DI k1 CORRISPONDE **UNO E UNO SOLO** VALORE DI k2
  - NON E' VERO IL CONTRARIO
- SI PUO' QUINDI DIRE CHE **k1 IMPLICA k2** (DIPENDENZA FUNZIONALE DA k1 A k2)
- SE SI UTILIZZA IL MODELLO STANDARD A TRE RELAZIONI LA CHIAVE DELLA RELAZIONE CHE TRADUCE L'ASSOCIAZIONE SI RIDUCE A k1

E1 (k1, a1, b1)    E2 (k2, a2, b2)    R (k1, k2, ar, br)

© Fabio A. Schreiber

Progettazione 110

## ASSOCIAZIONE BINARIA 1:n



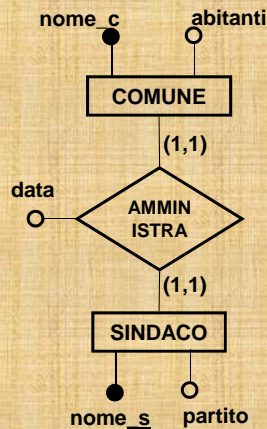
CREATE TABLE CLIENTE (p\_IVA ... NOT NULL, nome ..., telefono ..., PRIMARY KEY (p\_IVA))

CREATE TABLE ORDINE (numero... NOT NULL, data ..., p\_IVA ... NOT NULL, SCONTO ..., PRIMARY KEY (numero) FOREIGN KEY (p\_IVA) REFERENCES CLIENTE);

© Fabio A. Schreiber

Progettazione 111

## ASSOCIAZIONE BINARIA 1:1



SI TRADUCE CON **UNA SOLA RELAZIONE** CHE  
COMPRENDE TUTTI GLI ATTRIBUTI

CREATE TABLE AMMINISTRAZIONE (nome\_c  
... NOT NULL, abitanti ..., nome\_s ...  
NOT NULL UNIQUE, partito ..., data ...  
PRIMARY KEY (nome\_c));

© Fabio A. Schreiber

Progettazione 112

## ASSOCIAZIONE BINARIA 1:1

- **CARDINALITA' MINIME ENTRAMBE 1**
  - VERRA' SCELTA COME CHIAVE PRIMARIA LA PIU' SIGNIFICATIVA TRA k1 E k2
- **CARDINALITA' DI E2 E' (0,1) E QUELLA DI E1 E' (1,1)**
  - LA CHIAVE PRIMARIA SARA' k2
  - CI SARANNO VALORI NULLI DI k1 IN CORRISPONDENZA DELLE ISTANZE DI E2 CHE NON SI ASSOCIANO
- **CARDINALITA' (0,1) PER ENTRAMBE**
  - BISOGNA MANTENERE **DUE RELAZIONI DISTINTE** POICHE' ALTRIMENTI CI SAREBBERO VALORI NULLI SULLA CHIAVE

© Fabio A. Schreiber

Progettazione 113

## ASSOCIAZIONE BINARIA 1:1

- **TRADUZIONE CON DUE RELAZIONI**
  - L'ASSOCIAZIONE PUO' ESSERE COMPATTATA CON L'ENTITA' CHE PARTECIPA OBBLIGATORIAMENTE
  - SE ENTRAMBE SONO OBBLIGATORIE SI SCEGLIE QUELLA PIU' SIGNIFICATIVA
- **TRADUZIONE CON TRE RELAZIONI**
  - LA CHIAVE DELLA RELAZIONE CHE TRADUCE L'ASSOCIAZIONE PUO' ESSERE INDIFFERENTEMENTE  $k_1$  O  $k_2$  (NON CI SONO PROBLEMI DI VALORI NULLI)

© Fabio A. Schreiber

Progettazione 114

## AUTO-ASSOCIAZIONE (n:m)

### VIENE TRADOTTA CON

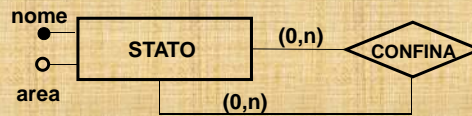
- UNA RELAZIONE PER L'ENTITA'
- UNA RELAZIONE PER L'ASSOCIAZIONE
  - QUESTA CONTIENE **DUE VOLTE** LA CHIAVE DELL'ENTITA'
  - E' NECESSARIO **DIVERSIFICARE I NOMI** DEI DUE ATTRIBUTI PER EVITARE L'OMONIMIA

© Fabio A. Schreiber

Progettazione 115



## AUTO-ASSOCIAZIONE (n:m)



```
CREATE TABLE STATO (nome ... NOT NULL, area ...
PRIMARY KEY (nome))
```

```
CREATE TABLE CONFINA (stato_a ... NOT NULL, stato_b ... NOT NULL,
PRIMARY KEY (stato_a, stato_b)
FOREIGN KEY (stato_a) REFERENCES STATO
FOREIGN KEY (stato_b) REFERENCES STATO);
```

© Fabio A. Schreiber

Progettazione 116

## AUTO-ASSOCIAZIONE (1:n)

### VIENE TRADOTTA CON

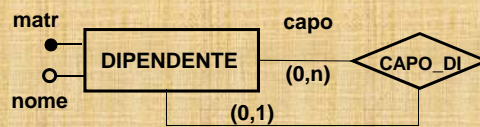
- UNA SOLA RELAZIONE
- L'ATTRIBUTO CHIAVE VIENE RIPETUTO DUE VOLTE
  - UNA VOLTA COME CHIAVE
  - LA SECONDA VOLTA COME RIFERIMENTO ALL'ISTANZA CONNESSA
- I DUE NOMI VENGONO DIVERSIFICATI

© Fabio A. Schreiber

Progettazione 117



## AUTO-ASSOCIAZIONE (1:n)



CREATE TABLE DIPENDENTE (matr ... NOT NULL, nome ..., capo ...,  
PRIMARY KEY (matr)  
FOREIGN KEY (capo) REFERENCES DIPENDENTE);

© Fabio A. Schreiber

Progettazione 118

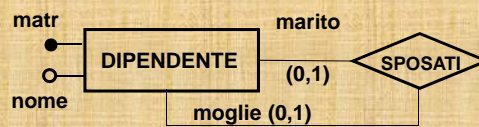
## AUTO-ASSOCIAZIONE (1:1)

- NEL CASO DI AUTO-ASSOCIAZIONE (1:1)  
IL RUOLO ASSUME MAGGIORE  
IMPORTANZA
- SU ENTRAMBI I RAMI E' BENE  
**SPECIFICARE IL RUOLO**
- CONVIENE LA SOLUZIONE A **DUE**  
**RELAZIONI** PER EVITARE
  - RIDONDANZE
  - VINCOLI
  - ECCESSO DI VALORI NULLI

© Fabio A. Schreiber

Progettazione 119

## AUTO-ASSOCIAZIONE (1:1)



CREATE TABLE DIPENDENTE (matr ... NOT NULL, nome ..., capo ...,  
PRIMARY KEY (matr))

CREATE TABLE SPOSATI (moglie ... NOT NULL, marito ... NOT NULL  
PRIMARY KEY (moglie)  
FOREIGN KEY (moglie) REFERENCES DIPENDENTE  
FOREIGN KEY (marito) REFERENCES DIPENDENTE);

© Fabio A. Schreiber

Progettazione 120

## ASSOCIAZIONE n-ARIA

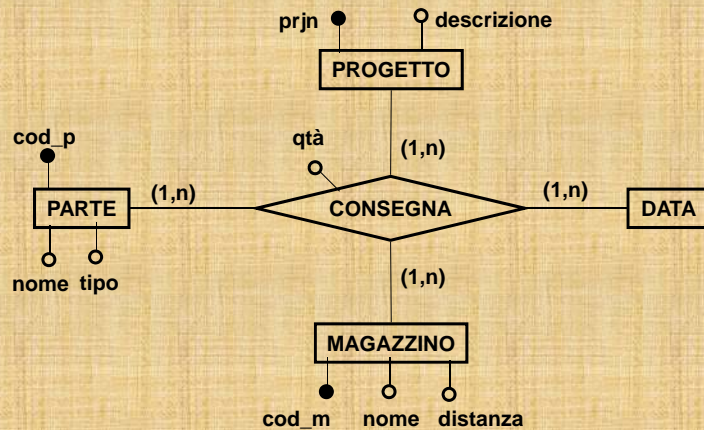
### SEGUE LA TRADUZIONE STANDARD

- NELLA RELAZIONE CHE TRADUCE  
L'ASSOCIAZIONE TALVOLTA **LA CHIAVE**  
OTTENUTA COMPONENTO LE CHIAVI DI  
TUTTE LE ENTITA' PARTECIPANTI **E' UNA**  
**SUPERCHIAVE** (LA CHIAVE VERA E' UN SUO  
SOTTOINSIEME)

© Fabio A. Schreiber

Progettazione 121

## ASSOCIAZIONE n-ARIA



© Fabio A. Schreiber

Progettazione 122

## ASSOCIAZIONE n-ARIA

```
CREATE TABLE PROGETTO (prj ... NOT NULL, descrizione ...,
PRIMARY KEY (prj))
CREATE TABLE PARTE (cod_p ... NOT NULL, nome ..., tipo ...,
PRIMARY KEY (cod_p))
CREATE TABLE MAGAZZINO (cod_m ... NOT NULL, nome ...,
distanza ..., PRIMARY KEY (cod_m))
CREATE TABLE CONSEGNA (prj ... NOT NULL, cod_p ... NOT
NULL, cod_m ... NOT NULL, data ... NOT NULL, qtà ...
PRIMARY KEY (prj, cod_p, cod_m, data)
FOREIGN KEY (prj) REFERENCES PROGETTO
FOREIGN KEY (cod_p) REFERENCES PARTE
FOREIGN KEY (cod_m) REFERENCES MAGAZZINO);
```

**NON C'E' UNA RELAZIONE PER DATA** IN QUANTO E' UN'ENTITA' FITTIZIA USATA PER GARANTIRE L'UNICITA' DELLE CONSEGNE (COMPARE NELLA CHIAVE)

© Fabio A. Schreiber

Progettazione 123

## ASSOCIAZIONE n-ARIA

- PUO' SUCCEDERE CHE NEL CORSO DEL PROGETTO CI SI ACCORGA CHE **UNA PARTE PUO' ESISTERE IN UN SOLO MAGAZZINO**
- IN TAL CASO **cod\_p IMPLICA cod\_m** E QUEST'ULTIMO DIVENTA **RIDONDANTE** NELLA CHIAVE DELLA RELAZIONE

```
.....  
CREATE TABLE CONSEGNA (prj ... NOT NULL, cod_p ... NOT  
NULL, cod_m ... NOT NULL, data ... NOT NULL, qtà ...  
PRIMARY KEY (prj, cod_p, data)  
FOREIGN KEY (prj) REFERENCES PROGETTO  
FOREIGN KEY (cod_p) REFERENCES PARTE);
```

© Fabio A. Schreiber

Progettazione 124

## ASSOCIAZIONE n-ARIA

- NELL'ESEMPIO PRECEDENTE LA DIPENDENZA TRA MAGAZZINO E PARTE NON ERA STATA ESPRESSA SULLA RELAZIONE n-ARIA
- SE IL PROGETTO **CONCETTUALE E' BEN FATTO** CASI DEL GENERE NON SONO FREQUENTI
- DIVERSO E' IL CASO IN CUI SI VOGLIANO ESPRIMERE DEI VINCOLI CHE RICHIEDEREBBERO UN USO COMPLICATO DI ENTITA' DI COLLEGAMENTO CON IDENTIFICAZIONE ESTERNA
- LA **VERIFICA DELLE CHIAVI** DELLE RELAZIONI E' QUINDI **IMPORTANTE** E LA MATERIA E' OGGETTO DELLA **TEORIA DELLA NORMALIZZAZIONE**

© Fabio A. Schreiber

Progettazione 125