

Progetto logico

Progetto logico

- Lo schema E/R descrive un dominio applicativo ad un dato livello di astrazione
- Lo schema E/R serve per:
 - fornire una descrizione sintetica e visiva
 - rappresentare buona parte della semantica dell'applicazione
 - scambiare informazioni sull'attività progettuale tra i membri del team di progetto e mantenere una documentazione

2

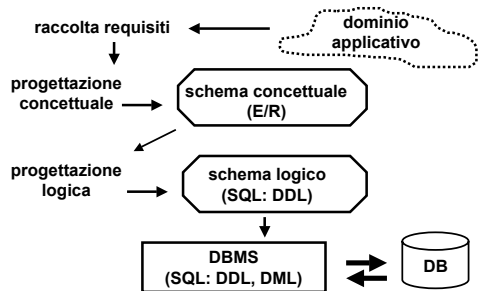
Progetto logico

- Non esistono DBMS in grado di operare direttamente sui concetti di schemi E/R

- è quindi necessario tradurli in altri schemi di dati (logico relazionale in queste lezioni)
- questa traduzione può essere eseguita in modo semi-automatico
- le scelte alternative devono tenere conto dell'efficienza dello schema logico risultante e delle operazioni da effettuare (derivanti da flussi e processi)

3

processo di design



4

scelte alternative

si possono individuare alcune linee guida:

- considerare le proprietà logiche comunque primarie rispetto ai motivi di efficienza
- tenere sulla stessa entità informazioni che verranno di frequente consultate insieme
- tenere su entità separate informazioni che verranno consultate separatamente
- limitare l'incidenza di valori nulli per attributi opzionali

5

fasi del progetto

il progetto produce trasformazioni e traduzioni dello schema E/R con le seguenti fasi:

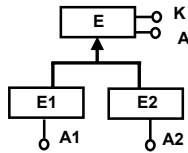
- 1 eliminazione delle gerarchie isa
- 2 selezione delle chiavi primarie, eliminazione delle identificazioni esterne
- 3 normalizzazione degli attributi composti o multipli
- 4 traduzione di entità e associazioni in schemi di relazioni
- 5 verifica di normalizzazione

6

eliminazione delle gerarchie

il modello relazionale non rappresenta le gerarchie, le gerarchie sono sostituite da entità e associazioni:

- 1) mantenimento delle entità con associazioni
- 2) collasso verso l'alto
- 3) collasso verso il basso

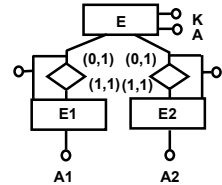


l'applicabilità e la convenienza delle soluzioni dipendono dalle proprietà di copertura e dalle operazioni previste

7

mantenimento delle entità

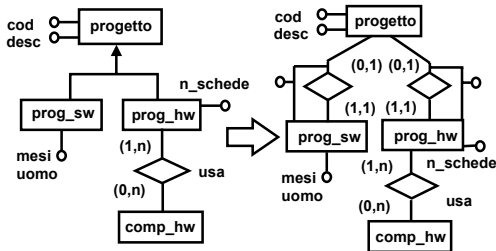
- tutte le entità vengono mantenute
- le entità figlie sono in associazione con l'entità padre
- le entità figlie sono identificate esternamente tramite l'associazione



questa soluzione è sempre possibile, indipendentemente dalla copertura

8

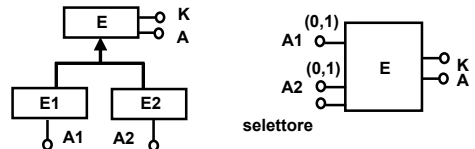
mantenimento entità - es.:



9

eliminazione delle gerarchie

- Il collasso verso l'alto riunisce tutte le entità figlie nell'entità padre



selettore è un attributo che specifica se una istanza di E appartiene a una delle sottoentità

10

isa: collasso verso l'alto

- il collasso verso l'alto favorisce operazioni che consultano insieme gli attributi dell'entità padre e quelli di una entità figlia:
 - in questo caso si accede a una sola entità, anziché a due attraverso una associazione
- gli attributi obbligatori per le entità figlie divengono opzionali per il padre
 - si avrà una certa percentuale di valori nulli

11

isa: collasso verso l'alto

Copertura dell'ISA

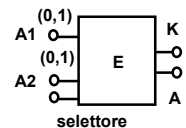
totale esclusiva:

selettore ha N valori,

quante sono le sottoentità

parziale esclusiva:

selettore ha N+1 valori; il valore in più serve per le istanze che non appartengono ad alcuna sottoentità



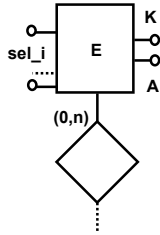
12

isa: collasso verso l'alto

Copertura dell'ISA:

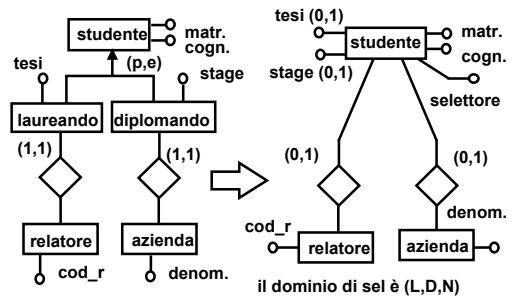
overlapping: occorrono tanti selettori booleani quante sono le sottoentità, sel_i è "vero" per ogni istanza di E che appartiene a E_i se la copertura è parziale i selettori possono essere tutti "falsi"

le associazioni connesse alle sottoentità si trasportano su E, le eventuali cardinalità minime diventano 0



13

isa: collasso verso l'alto



14

isa: collasso verso il basso

Collasso verso il basso:

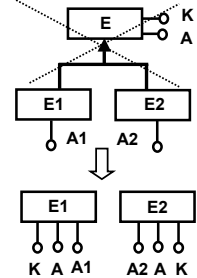
- si elimina l'entità padre trasferendone gli attributi su tutte le entità figlie
 - una associazione del padre è replicata, tante volte quante sono le entità figlie
 - la soluzione è interessante in presenza di molti attributi di specializzazione (con il collasso verso l'alto si avrebbe un eccesso di valori nulli)
 - favorisce le operazioni in cui si accede separatamente alle entità figlie

15

isa: collasso verso il basso

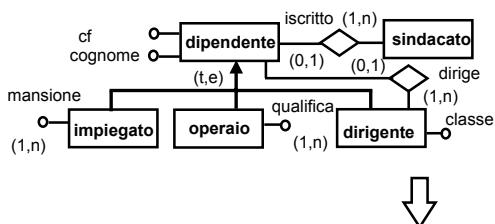
limiti di applicabilità:

- se la copertura è parziale non si può fare: dove mettere gli E che non sono né E1, né E2?
- se la copertura è overlapping introduce ridondanza: per una istanza presente sia in E1 che in E2 si rappresentano due volte gli attributi di E



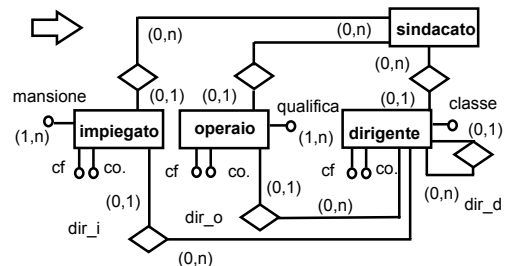
16

collasso verso il basso: es.



17

collasso verso il basso: es.



18

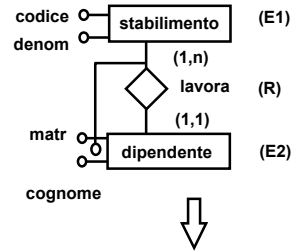
Scelta della chiave primaria

- È necessario che tra i diversi identificatori di una entità venga designata una chiave primaria: per la chiave primaria occorrerà, infatti, che il DBMS sia provvisto di strumenti per garantire l'unicità dei valori
- criteri euristici di scelta:
 - primo: scegliere la chiave che è usata più frequentemente per accedere all'entità
 - secondo: si preferiscono chiavi semplici a chiavi composte, interne anziché esterne

19

identificatori esterni

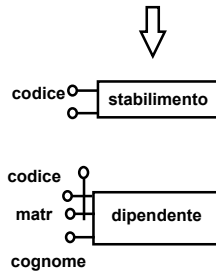
- una componente di identificazione esterna di una entità E2 da una entità E1 attraverso una associazione R comporta il trasporto della chiave primaria di E1 su E2



20

identificatori esterni

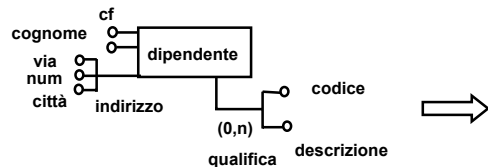
- in questo modo l'associazione è rappresentata attraverso la chiave, e può essere eliminata
- la chiave trasportata è chiave esterna
- in presenza di più identificazioni in cascata, è necessario iniziare la propagazione dall'entità che non ha identificazioni esterne



21

attributi composti/ripetuti

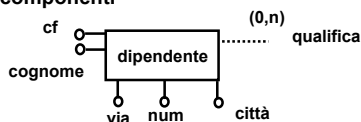
le relazioni non possono contenere attributi composti o attributi ripetuti, ma solamente attributi "atomici"



22

attributi composti

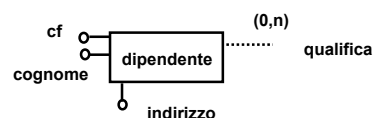
- Due possibili soluzioni
 - eliminare l'attributo composto e considerare i suoi componenti come attributi semplici
 - in questo modo si perde la visione unitaria ma si mantiene l'articolazione dei componenti



23

attributi composti

- eliminare i componenti e considerare l'attributo come semplice
 - in questo modo lo schema risulta semplificato, perdendo parte della struttura



24

attributi ripetuti

la definizione di relazione impone che, se una entità E ha un attributo A ripetuto, si crei una nuova entità che contenga l'attributo e sia collegata a E:

Caso a) - un valore può comparire una volta sola nella ripetizione:

la nuova entità EA ha l'identificatore composto dall'identificatore di E più l'attributo A

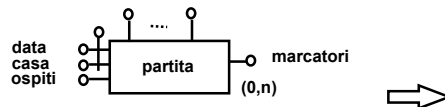


25

attributi ripetuti

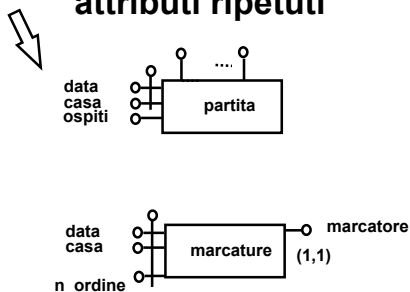
Caso b) - un valore può comparire più volte nella ripetizione:

la nuova entità EA ha l'identificatore composto dall'identificatore di E più un valore identificante sintetico (ad esempio, un numero d'ordine)



26

attributi ripetuti



27

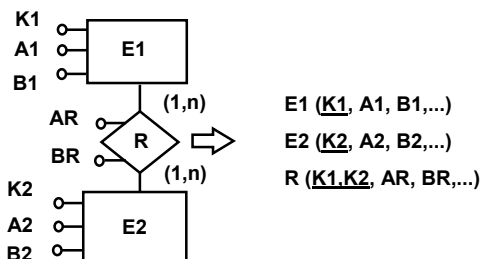
Traduzione

Traduzione standard

- ogni entità è tradotta con una relazione con gli stessi attributi
 - la chiave è l'identificatore dell'entità stessa (già visto)
- ogni associazione è tradotta con una relazione con gli stessi attributi, cui si aggiungono gli identificatori di tutte le entità che essa collega (già visto)
 - la chiave è composta dalle chiavi delle entità collegate

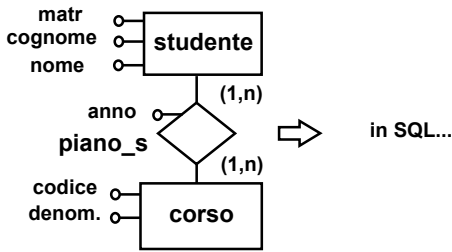
29

traduzione standard



30

traduzione standard: es.



31

traduzione standard: es.

```
CREATE TABLE STUDENTE (MATR... NOT NULL,
..., NOME..., PRIMARY KEY (MATR));

CREATE TABLE CORSO (CODICE... NOT NULL,
DENOM ..., PRIMARY KEY (CODICE));

CREATE TABLE PIANO_ST (MATR... NOT NULL,
CODICE... NOT NULL, ANNO...
PRIMARY KEY (MATR, CODICE),
FOREIGN KEY (MATR) REFERENCES STUDENTE
FOREIGN KEY (CODICE) REFERENCES CORSO);
```

32

altre traduzioni

- La traduzione standard è sempre possibile ed è l'unica possibilità per le associazioni N a M
- Altre forme di traduzione delle associazioni sono possibili per altri casi di cardinalità (1 a 1, 1 a N)
- Le altre forme di traduzione fondono in una stessa relazione entità e associazioni

33

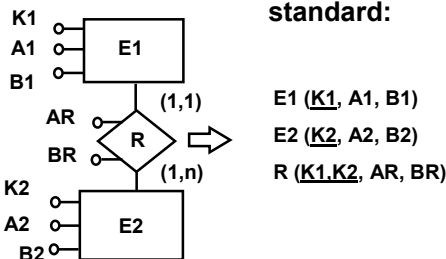
altre traduzioni

- Le altre forme di traduzione:
 - danno origine a un minor numero di relazioni e generano quindi uno schema più semplice
 - richiedono un minor numero di join per la navigazione attraverso un'associazione, ovvero per accedere alle istanze di entità connesse tramite l'associazione
 - penalizzano le operazioni che consultano soltanto gli attributi di una entità che è stata fusa

34

Associazione binaria 1 a N

- traduzione standard:



E1 (K1, A1, B1)
E2 (K2, A2, B2)
R (K1, K2, AR, BR)

35

associazione binaria 1 a N

- Se E1 partecipa con cardinalità (1,1) può essere fusa con l'associazione, ottenendo una soluzione a due relazioni:

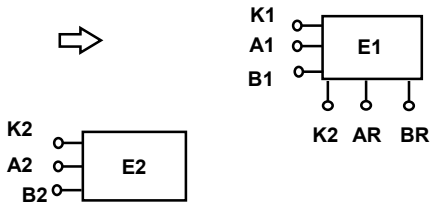
E1(K1, A1, B1, K2, AR, BR)
E2(K2, A2, B2)

- Se E1 partecipa con cardinalità (0,1) la soluzione a due relazioni ha valori nulli in K2, AR, BR per le istanze di E1 che non partecipano all'associazione

36

Associazione binaria 1 a N

- equivale a:



37

associazione binaria 1 a N

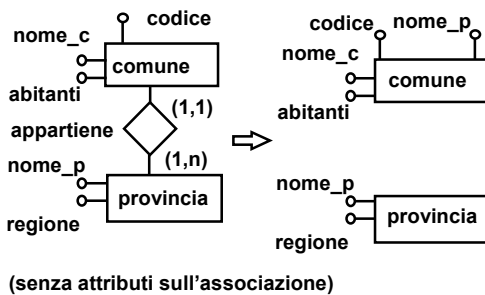
- **Attenzione** : in questo caso, poiché la partecipazione di E1 è 0,1 o 1,1, si nota facilmente che ad un dato valore di K1 corrisponde uno e un sol valore di K2 (non è vero il contrario), quindi si può dire che K1 implica K2 o, anche, che esiste una dipendenza funzionale da K1 a K2
- nella soluzione a 3 relazioni la chiave della relazione che traduce l'associazione è riducibile a K1:

$E1(\underline{K1}, A1, B1)$, $E2(\underline{K2}, A2, B2)$

$R(\underline{K1}, K2, AR, BR)$

38

ass. binaria 1 a N es.



39

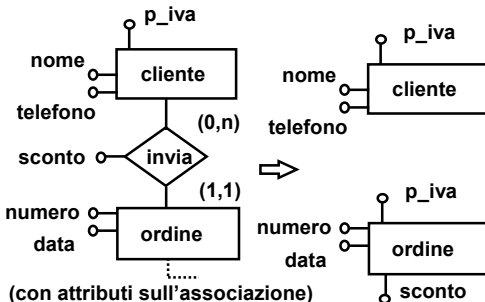
ass. binaria 1 a N es.

```
CREATE TABLE PROVINCIA
(NOME_P ... NOT NULL,
REGIONE ... PRIMARY KEY (NOME_P));
```

```
CREATE TABLE COMUNE
(CODICE ... NOT NULL, NOME_C ...
ABITANTI ..., NOME_P ... NOT NULL
PRIMARY KEY (CODICE)
FOREIGN KEY NOME_P
REFERENCES PROVINCIA);
```

40

ass. binaria 1 a N es.



41

ass. binaria 1 a N es.

traduzione con due relazioni:

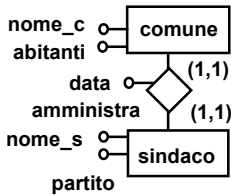
```
CREATE TABLE CLIENTE (P_IVA..... NOT NULL,
NOME ...,TELEFONO ..., PRIMARY KEY (P_IVA));
```

```
CREATE TABLE ORDINE (NUMERO ... NOT NULL,
DATA ... P_IVA ... NOT NULL, SCONTO ...,
PRIMARY KEY (NUMERO)
FOREIGN KEY P_IVA REFERENCES CLIENTE);
```

42

Associazione binaria 1 a 1

- traduzione con una relazione:



E12 (K1, A1, B1,
K2, A2, B2,
AR, BR)



43

associazione binaria 1 a 1

```
CREATE TABLE AMMINISTRAZIONE
(NOME_C ... NOT NULL, ABITANTI ...,
 NOME_S ... NOT NULL UNIQUE,
 INDIRIZZO ..., DATA
 PRIMARY KEY (NOME_C));
```

se le cardinalità minime sono entrambe 1 la chiave può essere indifferentemente K1 o K2 si sceglierà quella più significativa

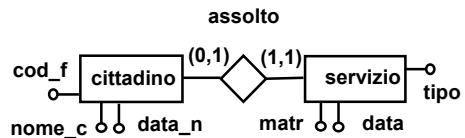
44

associazione binaria 1 a 1

- se la cardinalità di E2 è 0,1 e quella di E1 è 1,1 allora la chiave sarà K2; E2 è l'entità con maggior numero di istanze alcune delle quali non si associano, ci saranno quindi valori nulli in corrispondenza di K1, K1 in questo caso non potrebbe essere scelta
- se la cardinalità è 0,1 da entrambe le parti allora le relazioni saranno due per l'impossibilità di assegnare la chiave all'unica relazione a causa della presenza di valori nulli sia su K1 che su K2

45

associazione binaria 1 a 1



```
CREATE TABLE CITTADINO
(COD_F ... NOT NULL, NOME_C ... NOT NULL,
 INDIRIZZO ..., DATA_N ..., MATR ..., DATA...,
 TIPO ....., PRIMARY KEY (COD_F));
```

46

associazione binaria 1 a 1

- Traduzione con due relazioni
 - l'associazione può essere compattata con l'entità che partecipa obbligatoriamente (una delle due se la partecipazione è obbligatoria per entrambe) la discussione sulla chiave è analoga al caso di traduzione con una relazione

E1 (K1, A1, B1,...)

E2 (K2, A2, B2,... K1, AR, BR)

47

associazione binaria 1 a 1

- Traduzione con tre relazioni
 - la chiave della relazione che traduce l'associazione può essere indifferentemente K1 o K2, non ci sono problemi di valori nulli

E1 (K1, A1, B1,...)

E2 (K2, A2, B2,...)

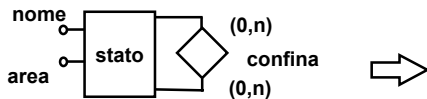
R (K1, K2, AR, BR,...)

48

Auto associazione N a M

viene tradotta con:

- una relazione per l'entità ed
- una per l'associazione,
 - quest'ultima contiene due volte la chiave dell'entità, è necessario però modificare i nomi degli attributi, per non avere omonimia



49

auto associazione N a M

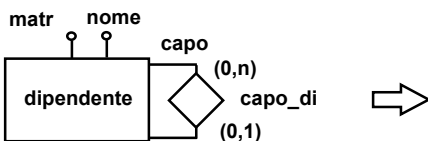
```
CREATE TABLE STATO
(NOME ... NOT NULL, AREA ...
PRIMARY KEY (NOME));
```

```
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);
```

50

auto associazione 1 a N

- è traducibile con una sola relazione che contiene due volte l'attributo chiave: una volta come chiave ed una come riferimento all'istanza connessa, con nome diverso per specificare il ruolo



51

auto associazione 1 a N

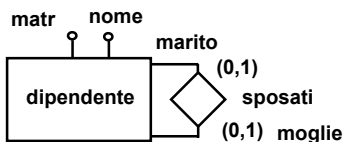
```
CREATE TABLE DIPENDENTE
(MATR ... NOT NULL, NOME ..., CAPO ...
PRIMARY KEY (MATR)
FOREIGN KEY (CAPO)
REFERENCES DIPENDENTE);
```

- nel caso di associazione 1 ad 1 il concetto di ruolo assume maggiore importanza:



52

auto associazione 1 a 1



- su entrambi i rami è bene specificare il ruolo: conviene la soluzione con due relazioni per evitare ridondanze, vincoli ed eccesso di valori nulli.

53

auto associazione 1 a 1

```
CREATE TABLE DIPENDENTE (MATR ... NOT
NULL, NOME ..., 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);
```

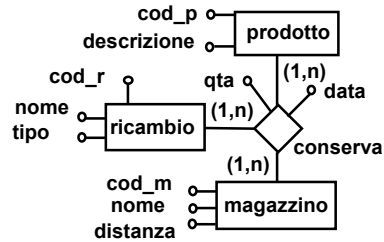
54

Associazione n-aria

- segue la traduzione standard
- talvolta, nella relazione che traduce l'associazione, la chiave ottenuta componendo le chiavi di tutte le entità partecipanti è una superchiave, cioè una chiave composta il cui set di componenti non è minimale (la chiave vera è un sottoinsieme)
- Esempio: prodotti-ricambi-magazzini

55

associazione n-aria



56

associazione n-aria

```

CREATE TABLE PRODOTTO (COD_P... NOT NULL,
DESCRIZIONE... , PRIMARY KEY (PRD));
CREATE TABLE RICAMBIO (COD_R ... NOT NULL,
NOME..., TIPO..., PRIMARY KEY (COD_R));
CREATE TABLE MAGAZZINO (COD_M.... NOT
NULL, NOME ..., DISTANZA..., PRIMARY KEY
(COD_M));
    
```

57

associazione n-aria

l'associazione diventa:

```

CREATE TABLE CONSERVA (COD_P ... NOT NULL,
COD_R... NOT NULL, COD_M... NOT NULL,
DATA... NOT NULL, QTA ...
PRIMARY KEY (COD_P, COD_R, COD_M)
FOREIGN KEY (COD_P)
REFERENCES PRODOTTO
FOREIGN KEY (COD_M)
REFERENCES MAGAZZINO
FOREIGN KEY (COD_R)
REFERENCES RICAMBIO);
    
```

ipotizziamo che COD_M sia ridondante ➡

58

associazione n-aria

- un ricambio esiste in un solo magazzino, quindi COD_R è associato ad un solo COD_M, cioè determina COD_M, allora la presenza di COD_M nella chiave è ridondante:

```

CREATE TABLE CONSERVA (COD_P ... NOT NULL,
COD_R... NOT NULL, COD_M... NOT NULL,
DATA..., QTA ...
PRIMARY KEY (COD_P, COD_R)
FOREIGN KEY (COD_P) REFERENCES PRODOTTO
FOREIGN KEY (COD_R) REFERENCES RICAMBIO);
    
```

- COD_M non è più parte della chiave

59

commento

- nel caso precedente la dipendenza tra magazzino e ricambio non era stata espressa sulla associazione n-aria; abbiamo ipotizzato di scoprirla nella fase di progetto logico
- se il progetto concettuale è ben fatto casi del genere non sono frequenti
- il ricontrollo delle chiavi delle relazioni è quindi importante e se ne occupa la teoria della normalizzazione

60