



3 – Modello Relazionale e Progettazione logica

Prof. A. Longheu – Basi di dati

Modello Relazionale

- > Il modello relazionale fu proposto nel 1970 da E.F. Codd
- > Primi sistemi relazionali appaiono sul mercato nel 1981
- > Il modello logico si basa su due concetti:
 - relazione**: teoria matematica derivata dalla teoria degli insiemi
 - tabella**: intuitivo ma informalel'uso delle tabelle rende semplice da usare il modello, mentre la formalizzazione matematica ne consolida la potenza (ad esempio, verifica della correttezza o completezza di un db)
- > Le relazioni possono essere efficacemente e semplicemente rappresentate tramite tabelle.

Caratteristiche delle relazioni

- > Ogni riga della tabella viene chiamata ***n-upla*** o ***t-upla***
- > Non è definito alcun ordinamento fra le *n*-uple, quindi due tabelle con le stesse righe, anche in ordine diverso, rappresentano la stessa relazione
- > Le *n*-uple di una relazione devono essere distinte l'una dall'altra, quindi una tabella rappresenta una relazione solo se le sue righe sono tutte fra loro diverse
- > Per definire il significato e il ruolo di ogni colonna essa viene etichettata mediante un ***attributo***. In tal modo, l'ordine delle colonne risulta irrilevante.

<i>SqCasa</i>	<i>SqOspite</i>	<i>Casa</i>	<i>Ospite</i>
Juventus	Roma	0	1
Lazio	Milan	2	0
Torino	Sampdoria	1	1

Attributi

T-upla

3

Database Relazionali - Generalità

Una base di dati relazionale è costituita da più relazioni (una di solito non è sufficiente). Alcuni attributi contengono valori comuni (ripetuti cioè in relazioni differenti) necessari per definire collegamenti fra le relazioni

Studenti

<i>Matricola</i>	<i>Cognome</i>	<i>Nome</i>	<i>Nascita</i>
276545	Rossi	Maria	21/10/1971
565211	Bruni	Aldo	12/03/1973
446212	Lorenzi	Silvio	06/06/1970
656422	Vulturi	Paolo	22/08/1974

Corsi

<i>Codice</i>	<i>Docente</i>
A04	Sergi
A10	Palazzo
A23	La Corte
A45	Ingrassia

Esami

<i>Studente</i>	<i>Voto</i>	<i>Corso</i>
276545	28	A04
565211	26	A10
446212	25	A23
656422	30	A45

4

Valori Nulli

In alcuni casi è necessario assegnare ad un attributo un valore non significativo perché:

- non se ne conosce il valore
- non è significativo per gli altri attributi
- deve essere inserito in un secondo momento

Non sempre esiste un valore al quale sia possibile assegnare il significato di “nullo”, per cui viene introdotto un valore “speciale” NULL.

Non tutte le colonne possono essere nulle (ad esempio, un numero di matricola nella tabella STUDENTI).

Occorre un controllo per evitare che la presenza multipla di più NULL in righe diverse generi ambiguità o faccia perdere di significato la tupla considerata.

Esami

Studente	Voto	Corso
NULL	28	NULL
NULL	26	A10
NULL	26	A10
656422	30	A45

Callout boxes: "inutilizzabile" points to the first row (NULL, 28, NULL); "ambiguo" points to the second and third rows (NULL, 26, A10).

5

Vincoli di integrità

I vincoli di integrità sono proprietà che devono essere soddisfatte dalle istanze per potere rappresentare informazioni corrette. I vincoli sono di due categorie:

1. vincoli **intrarelazionali** (all'interno di una relazione)
 - vincolo su valori (ad esempio, VOTO compreso fra 18 e 30)
 - vincolo di t-tupla (ad esempio, LODE non può esserci se il VOTO è inferiore a 30)
2. vincoli **interrelazionali** (su più relazioni), ad esempio una tupla in ESAMI per essere valida deve contenere un numero di matricola che esista anche in STUDENTI

Studenti

Matricola	Cognome	Nome	Nascita
276545	Rossi	Maria	21/10/1971
565211	Bruni	Aldo	12/03/1973
446212	Lorenzi	Silvio	06/06/1970
446212	Vulturi	Paolo	22/08/1974

Esami

Studente	Voto	LoDe	Corso
276545	36	-	A04
565211	28	X	A10
446212	25	-	A23
999100	30	X	A45

6

Chiave - 1

insieme di attributi utilizzato per identificare univocamente le t-uple di una relazione

> **superchiave**: un insieme K di attributi è superchiave di R se non contiene due t-uple distinte, ad esempio come chiave si può prendere l'insieme di tutti gli attributi della relazione .

> **chiave**: un insieme K è chiave se è una superchiave minimale (cioè non contiene al suo interno altre superchiavi)

Studenti

Cognome	Nome	Nascita	Corso
Rossi	Maria	21/10/1971	Ing. Meccan.
Bruni	Aldo	12/03/1973	Ing. Civile
Lorenzi	Silvio	06/06/1970	Ing. Inform.
Valturi	Paolo	22/08/1974	Ing. Inform.

{ *Cognome*, *Nome*, *Nascita*, *Corso* } è superchiave, e contiene la superchiave minimale (ossia, la chiave) { *Cognome*, *Nome*, *Nascita* } nell'ipotesi che siano sufficienti a distinguere le singole persone. 7

Chiave - 2

- Per trovare una chiave, si cerca una superchiave, si scartano tutti gli attributi possibili, restando con quelli indispensabili, che saranno una chiave.
- In una relazione possono esistere più chiavi, tutte egualmente valide.
- Nei casi reali, è quasi impossibile trovare attributi che siano sempre identificanti e disponibili (noti). Per questo si introduce un attributo avente il compito specifico di fare da chiave, motivo della nascita di matricole, numeri di targa, codici fiscali eccetera.
- La presenza indiscriminata di valori nulli negli attributi può rendere impossibile o inutile l'uso di una chiave.

Matricola	Cognome	Nome	Nascita	Corso
NULL	Rossi	Maria	NULL	Ing. Meccan.
565211	Bruni	Aldo	12/03/1973	Ing. Civile
NULL	Lorenzi	Silvio	06/06/1970	Ing. Inform.
446212	Lorenzi	Silvio	NULL	Ing. Inform.

impossibile

inutile

Chiavi : { *Cognome*, *Nome*, *Nascita* } e { *Matricola* }

Se esiste almeno un valore nullo su *tutte* le chiavi, la tupla non è più identificabile

Se esiste almeno un valore nullo su *almeno* una chiave, potrebbe sorgere ambiguità

Per evitare problemi, si definisce una delle chiavi come **primaria**, e si impone per essa l'assenza di valori nulli.

Vincoli di Integrità Referenziale

Un **Vincolo di Integrità Referenziale** (*foreign key*) fra un insieme di attributi X di una relazione R1 ed un'altra relazione R2 è soddisfatto se i valori su X per ogni tupla di R2 figurano come chiave primaria su R2.

Codice	Articolo	Prov.	Numero	Agente
1123	567	RM	2F8534	55-89
1256	125	RM	6D99909	12-88
1433	322	FI	D45668	82-33
2300	219	NA	F81642	23-43

Prov.	Numero	Proprietario
RM	6H5567	Rossi Mario
RM	6D99909	Broni Aldo
SA	2F8534	Sineri Luigi
BA	998342	Russo Paolo

Agente	Nome
40-10	Mico Sergio
12-88	Lumi Piero
82-33	Ronchi Aldo
23-43	Busi Nemo

9

Basi di Dati Relazionali

Rispetto agli altri modelli (reticolare, gerarchico), il modello relazionale è basato su valori, quindi:

- richiede di rappresentare solo ciò che è rilevante da parte dell'utente (altri modelli richiedono di tenere conto di altri campi, necessari alla gestione interna). Questo implica che:

- è semplice spostare l'informazione da un contesto ad un altro (ad esempio, da un calcolatore ad un altro).
- la rappresentazione logica non fa riferimento a quella fisica.



Progettazione Logica

- > Obiettivo della **Progettazione Logica** e' quello di costruire uno schema logico ,in un determinato *modello* (ad es. *relazionale*), che descriva in maniera *corretta* ed *efficiente* tutte le informazioni contenute nello schema E-R prodotto dalla progettazione concettuale.
- > Poichè abbiamo scelto il modello relazionale, la progettazione logica consiste nel tradurre il disegno (modello E-R prodotto durante la progettazione concettuale) in un insieme di tabelle
- > Non si tratta però di una semplice *traduzione*, perché occorre in genere anche fare delle scelte in base alle operazioni richieste nel DB
- > Il modello E-R e il modello relazionale sono distinti, usati in due fasi distinte della progettazione, anche se la parola "relazione" li accumuna. Il significato però e' diverso: la relazione nel modello E-R lega le entità fra loro (rombo), nel modello relazionale la relazione è una tabella. Quando si traducono entità e relazioni (rettangoli e rombi) entrambi diventano "relazioni" (tabelle) nel senso del modello relazionale.
- > Si potrebbe usare il modello E-R anche senza usare il modello relazionale (ad esempio usando quello gerarchico), o viceversa.

11



Progettazione Logica

La progettazione logica si articola in due fasi:

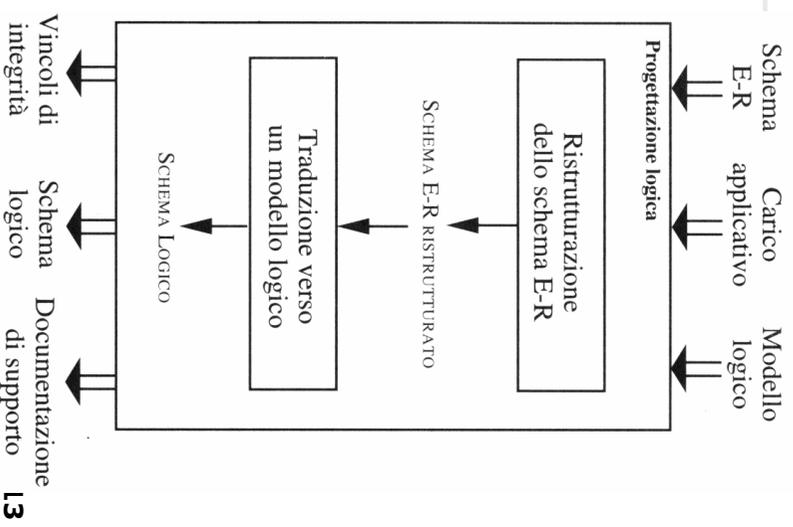
- > **Ristrutturazione dello schema E-R:** e' una fase indipendente dal modello logico e si basa su criteri di *ottimizzazione* dello schema e di successiva *semplificazione*.
- > **Traduzione verso il Modello Logico:** fa riferimento ad un modello logico (ad es. relazionale) e puo' includere ulteriore *ottimizzazione* che si basa sul modello logico stesso (es. normalizzazione).

12

Ristrutturazione

L'input ed output della fase di ristrutturazione dello schema E-R sono:

- > **Input:** Schema Concettuale E-R iniziale, Carico Applicativo previsto (in termini di dimensione dei dati e caratteristica delle operazioni)
- > **Output:** Schema E-R ristrutturato che rappresenta i dati e tiene conto degli aspetti realizzativi



Ristrutturazione Determinazione Carico Applicativo

- > L'input della ristrutturazione è lo schema E-R, ricavato dalla fase di progettazione concettuale, ed il carico applicativo.
 - > il **carico applicativo** è composto dagli **indici di prestazione** per la valutazione di schemi E-R. Tali indici sono due:
 - Costo di un'operazione:** in termini di numero di occorrenze di entità' ed associazioni che mediamente vanno visitate per rispondere a quella operazione sulla base di dati
 - Occupazione di memoria:** viene valutata in termini dello spazio di memoria (misurato in byte) necessario per memorizzare i dati del sistema.
- il carico applicativo serve per ristrutturare lo schema, ottimizzandolo in funzione delle operazioni più pesanti, ed occupando meno memoria possibile

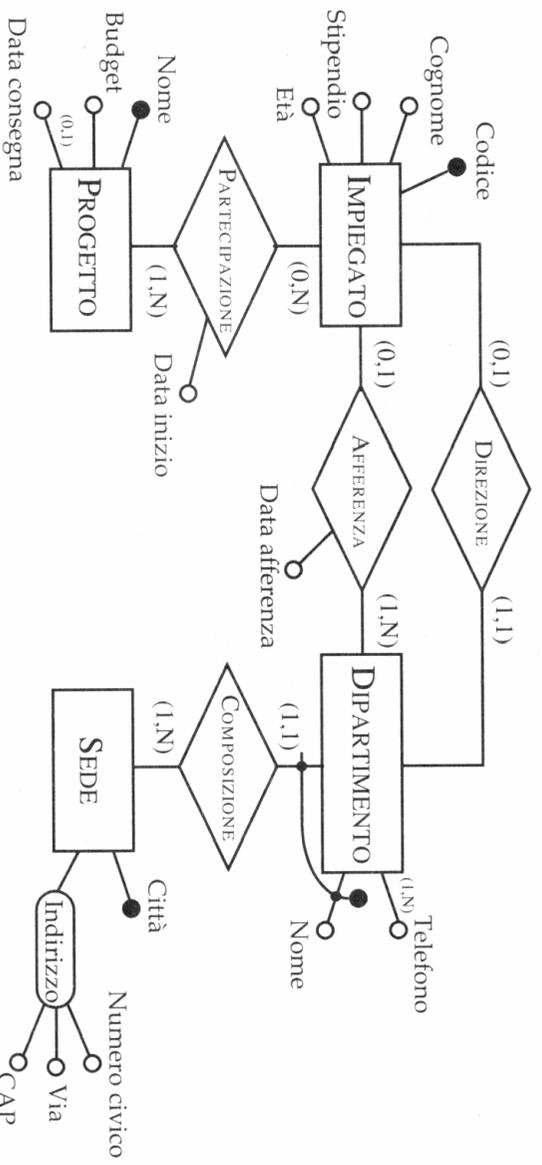
Ristrutturazione Determinazione Carico Applicativo

- > Per studiare questi indici abbiamo anzitutto bisogno di conoscere altri due set di valori:
- > **Volume dei dati:** a) numero (medio) di occorrenze di ogni entita' ed associazione b) dimensioni di ciascun attributo
- > **Caratteristiche delle operazioni:** a) tipo di operazione (interattiva o batch) b) frequenza (esecuzione/tempo) c) dati coinvolti (entita' e o associazioni)

15

Ristrutturazione Determinazione Carico Applicativo

Esempio:ditta con sedi in citta' diverse



16

Ristrutturazione

Determinazione Carico Applicativo

Operazioni dell'esempio

- > **Operazione 1:** assegna un impiegato ad un progetto
- > **Operazione 2:** trova i dati di un impiegato, del dipartimento nel quale lavora e dei progetti in cui e' coinvolto
- > **Operazione 3:** trova i dati di tutti gli impiegati di un certo dipartimento
- > **Operazione 4:** per ogni sede, trova i dipartimenti con il cognome del direttore e l'elenco degli impiegati.

17

Ristrutturazione

Determinazione Carico Applicativo

TABELLA DEI VOLUMI

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

TABELLA DELLE OPERAZIONI

Concetto	Tipo	Frequenza
Op. 1	I	50 al giorno
Op. 2	I	100 al giorno
Op. 3	I	10 al giorno
Op. 4	B	2 a settimana

18



Ristrutturazione Determinazione Carico Applicativo

- > il volume dei dati (tabella dei volumi) e le caratteristiche delle operazioni (tabella delle operazioni), vengono poi utilizzati come “pesi” per calcolare gli indici di prestazione.
- > Partendo dalla **stima dei costi** di ogni operazione, tale indice si determina contando il numero di accessi alle occorrenze di entita’ e relazioni necessario per eseguire l’operazione, usando poi come “peso” la frequenza dell’operazione stessa.
- > Prendiamo per esempio l’operazione 2 e facciamo riferimento allo schema di operazione

19



Ristrutturazione Determinazione Carico Applicativo

Dobbiamo accedere ad un’occorrenza di *Impiegato* e di *Afferenza* e quindi di *Dipartimento*. Successivamente , per avere i dati dei progetti a cui lavora, dobbiamo accedere (in media) a tre occorrenze di *Partecipazione* e quindi a tre entita’ *Progetto*. Tutto viene riassunto nella tavola degli accessi, che costituisce una stima del costo...

20

Ristrutturazione Determinazione Carico Applicativo

TAVOLA DEGLI ACCESSI

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Impiegato	Entita'	1	L
Afferenza	Relazione	1	L
Dipartimento	Entita'	1	L
Partecipazione	Relazione	3	L
Progetto	Entita'	3	L

Il Tipo indica se si tratta di un accesso in lettura o in scrittura; questo serve perche' la scrittura è solitamente più onerosa (quantitativamente si potrebbe ad esempio dire che $1 \text{ WR} = 2 * \text{RD}$)

21

Ristrutturazione Determinazione Carico Applicativo

TAVOLA DEGLI ACCESSI

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Impiegato	Entita'	1	L
Afferenza	Relazione	1	L
Dipartimento	Entita'	1	L
Partecipazione	Relazione	3	L
Progetto	Entita'	3	L

L'assegnazione del peso "assoluto" all'operazione 2 può essere data sommando gli accessi ($1+1+1+3+3$), ricordando che se fossero in scrittura andrebbero pesati (ad esempio $2x$), e moltiplicando il risultato per la frequenza dell'operazione quindi $9*100 =$

900 azioni elementari al giorno (peso assoluto dell'op.2)

Ripetendo per tutte le operazioni, si arrivano ad avere pesi quantitativi per stabilire quali operazioni sono effettivamente le più "pesanti".

Questi pesi rappresentano l'indice di prestazione "costo delle operazioni"

22

Ristrutturazione

Determinazione Carico Applicativo

Per valutare il secondo indice, l'occupazione di memoria, è sufficiente utilizzare la tabella dei volumi da cui prelevare il numero di occorrenze (record, righe, tuple), aggiungere la dimensione della singola occorrenza ed ottenere quindi l'occupazione di memoria di tutte le entità e relazioni, ad esempio Progetto ha

- > Nome (char[30])=30 byte (se un char=1 byte)
- > budget (long)=4 byte
- > data consegna (data)=8 byte (se GGMMAAAA)
- > Il totale è di 42 byte * 500 = 2100 byte

23

Ristrutturazione

- > Una volta determinato il carico applicativo, ed avendo dalla progettazione concettuale lo schema E-R, tutti gli input della ristrutturazione sono noti.
- > Si passa quindi alla ristrutturazione vera e propria, per determinare lo schema ristrutturato (output)
- > La ristrutturazione consta di **quattro fasi**:
 - analisi delle ridondanze
 - eliminazione delle generalizzazioni
 - partizionamento/accorpamento di entità ed associazioni
 - scelta degli identificatori principali (chiavi)

24

Ristrutturazione

- > **Analisi delle Ridondanze:** si decide se eliminare o no eventuali ridondanze.
- > **Eliminazione delle Generalizzazioni:** tutte le generalizzazioni vengono analizzate e sostituite da altro.
- > **Partizionamento/Accorpamento di entita' ed associazioni:** si decide se partizionare concetti in piu' parti o viceversa accorpare.
- > **Scelta degli identificatori primari:** si sceglie un identificatore per quelle entita' che ne hanno piu' di uno

25

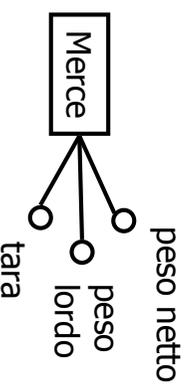
Ristrutturazione

Analisi delle ridondanze

Esempi di Ridondanza

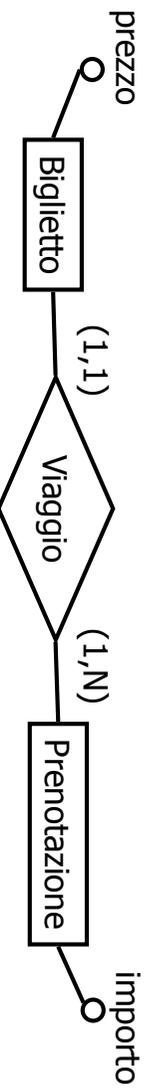
Attributi derivabili da altri della stessa entita':

- peso lordo, peso netto e tara



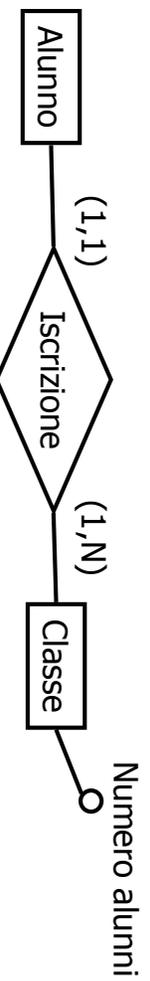
Attributi derivabili da attributi di altre entità (o associazioni):

- prezzo del biglietto e importo della prenotazione



Attributi derivabili da operazioni di conteggio:

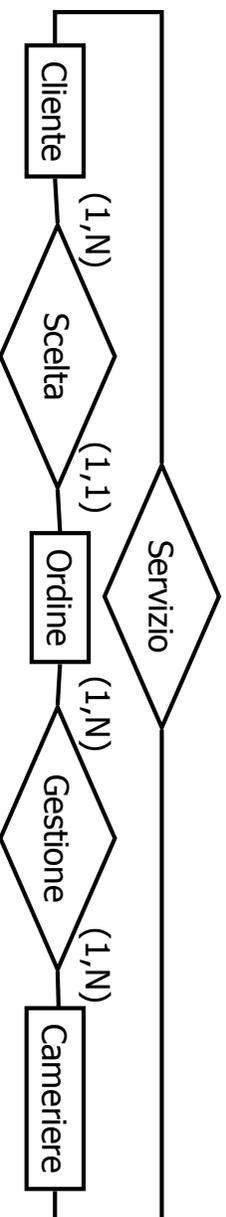
- Numero alunni ricavabile contando il numero di istanze di Alunno



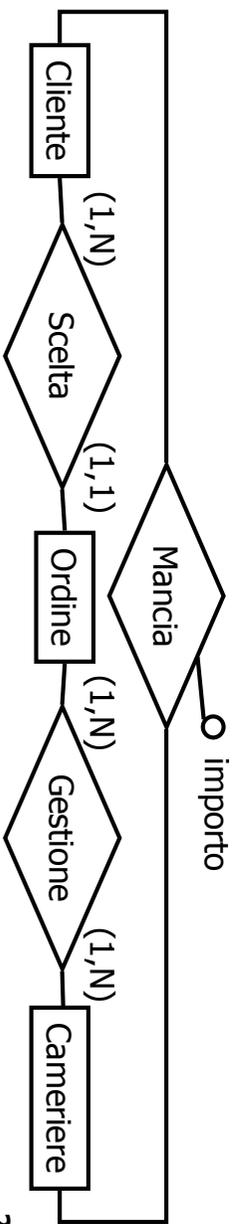
26

Ristrutturazione Analisi delle ridondanze

- > Associazioni derivabili dalla composizione di altre associazioni in presenza di cicli: *Servizio* da *Gestione* e *Scelta*



- > Notare tuttavia che i cicli non sempre generano ridondanze:



27

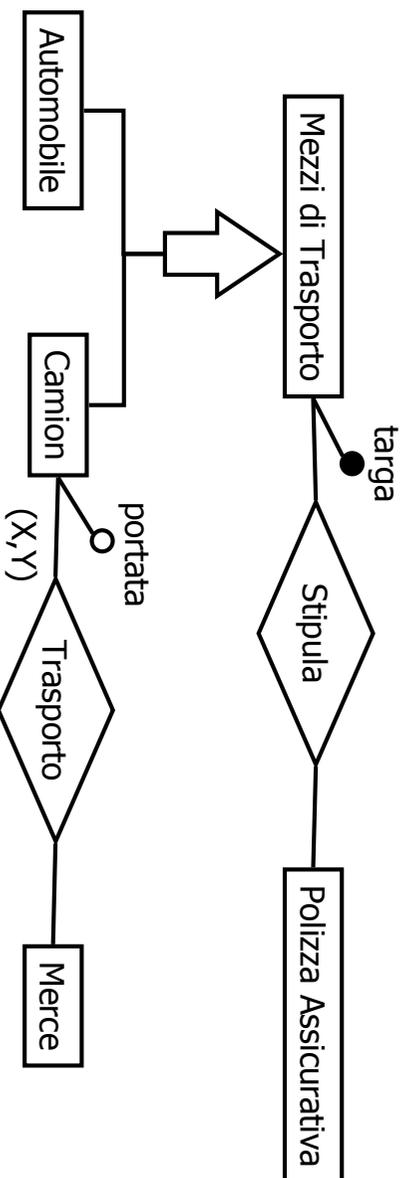
Ristrutturazione Analisi delle ridondanze

- > *Vantaggi* nell'avere e mantenere un dato derivabile: riduce gli accessi per calcolare il dato derivato.
- > *Svantaggi* : occupazione di memoria e necessita' di effettuare operazioni aggiuntive per mantenere il dato aggiornato
- > Decisione: mantenere o eliminare? Basta confrontare i costi di esecuzione delle operazioni sull'oggetto
- > Quindi non è che le ridondanze si devono levare sempre e comunque, piuttosto bisogna sapere se ci sono (rilevazione) e poi scegliere di lasciarle o meno; qui interviene il carico applicativo perché si valuta per ogni operazione se essa trae vantaggio lasciando la ridondanza oppure eliminandola; questo lo si ripete per tutte le operazioni (eventualmente considerando solo quelle più pesanti), verificando anche la diversa occupazione di memoria. Alla fine si sceglie cosa fare, ristrutturando quindi eventualmente lo schema E-R (cambiare il disegno)

28

Ristrutturazione Eliminazione delle generalizzazioni

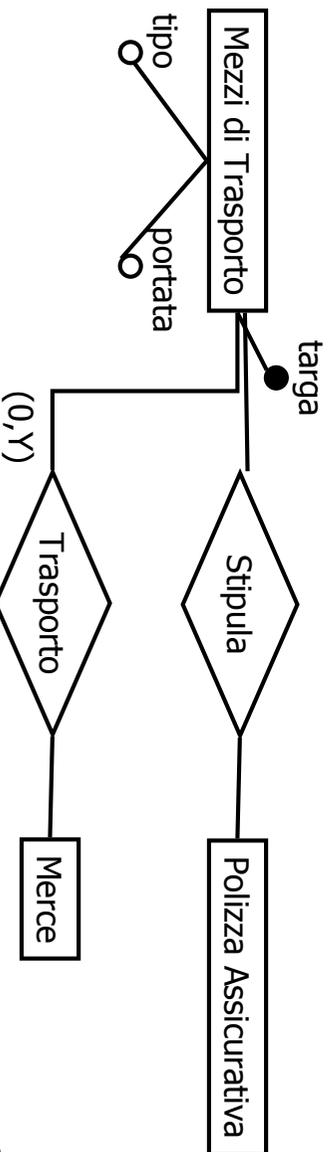
- > Il modello relazionale **non permette di rappresentare gerarchie** quindi occorre trasformarle in altri concetti.



29

Ristrutturazione Eliminazione delle generalizzazioni

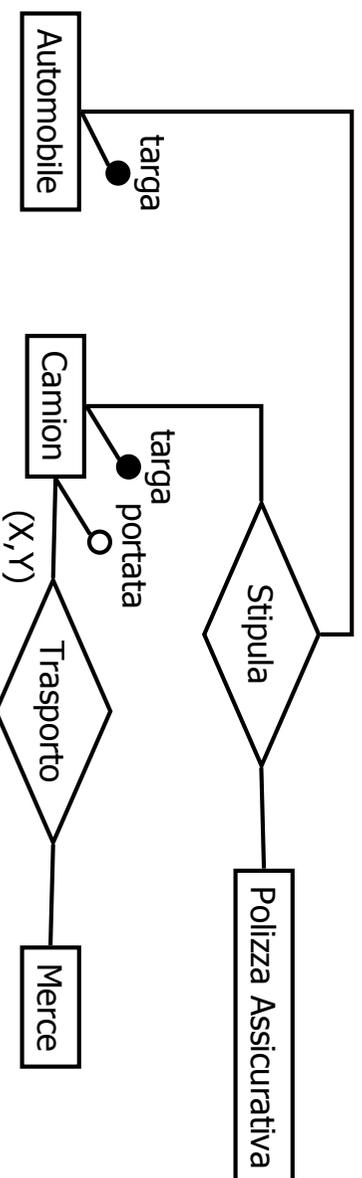
La **prima soluzione** prevede l'accorpamento dei figli nel padre: le entità 'figlie' vengono eliminate e le loro proprietà (attributi, associazioni, generalizzazioni) vengono aggiunte al padre, che deve avere in più un attributo (TIPO) per distinguere le entità figlie.



30

Ristrutturazione Eliminazione delle generalizzazioni

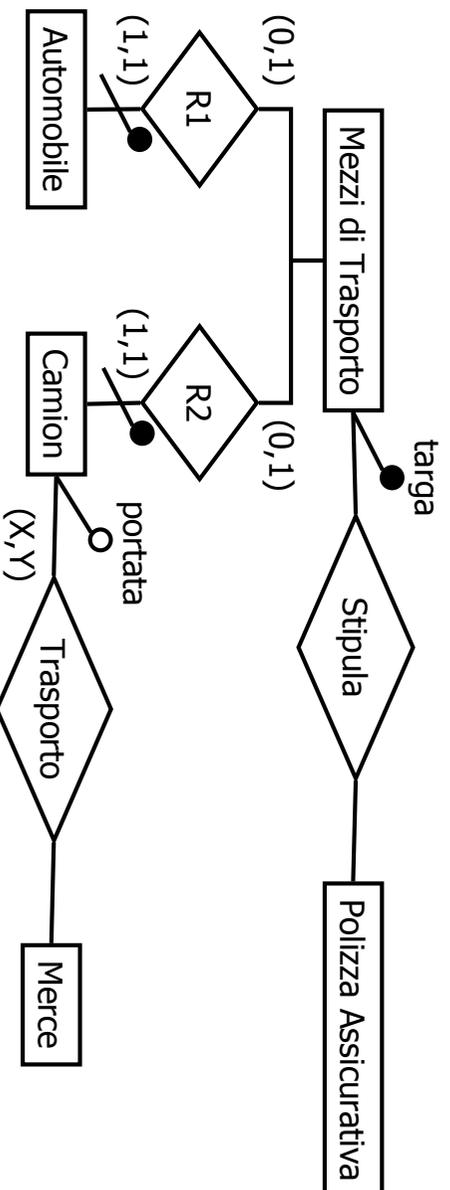
La **seconda soluzione** prevede l'accorpamento del padre nei figli: l'entità 'padre' viene eliminata e per ereditarietà i suoi attributi, identificatori e relazioni vanno ai figli.



31

Ristrutturazione Eliminazione delle generalizzazioni

La **terza soluzione** prevede la sostituzione: la generalizzazione si trasforma in associazioni uno ad uno che legano padre e figli



32



Ristrutturazione Eliminazione delle generalizzazioni

Quale delle tre scelte si deve fare?

Ancora una volta, occorre considerare le operazioni e vedere a quali entità accedono; se le operazioni più pesanti ad esempio accedono quasi sempre solo al padre, occorre lasciarlo e si potrebbero eliminare i figli, e così via

33



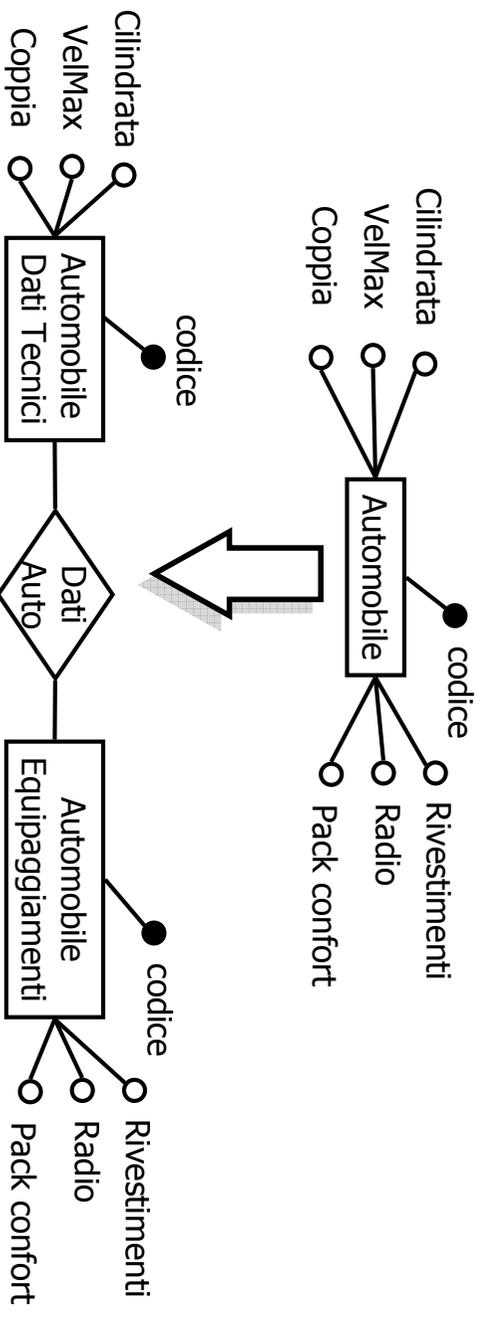
Ristrutturazione Partizionamento / accorpamento

- > I principi generali sono:
- > gli accessi si riducono **separando attributi di uno stesso concetto che vengono acceduti da operazioni diverse**
- > gli accessi si riducono **raggruppando attributi di concetti diversi che vengono acceduti dalle medesime operazioni**

34

Ristrutturazione Partizionamento / accorpamento

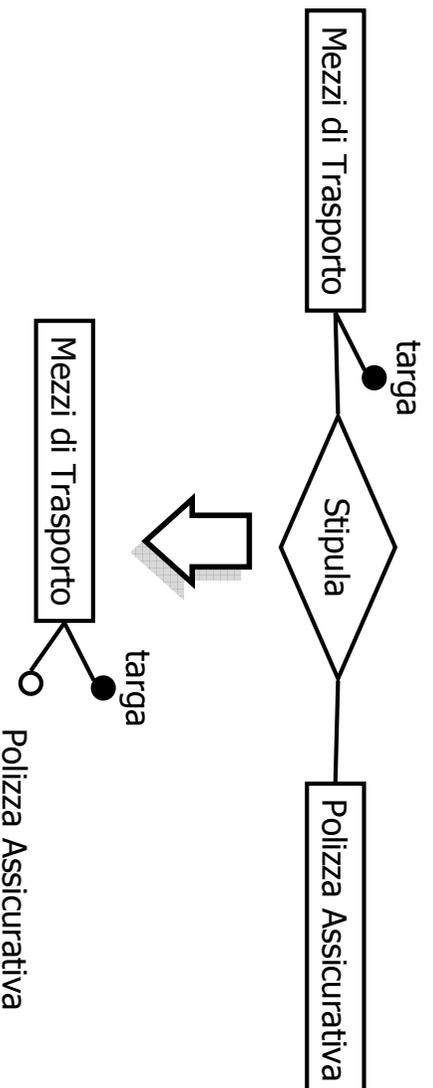
Esempio di **partizionamento**:



35

Ristrutturazione Partizionamento / accorpamento

Esempio di **accorpamento**:



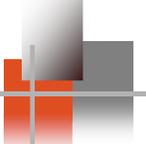
36



Ristrutturazione Partizionamento / accorpamento

- > L'accorpamento e' giustificato se le operazioni piu' frequenti su *Mezzi di trasporto* richiedono sempre i dati relativi alla *polizza assicurativa* e quindi vogliamo risparmiare gli accessi alla relazione che li lega.
- > Normalmente gli accorpamenti si fanno su relazioni uno ad uno, raramente su uno a molti e mai su molti a molti.

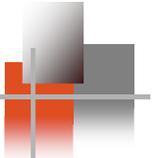
37



Ristrutturazione Scelta degli identificatori

- > Gli identificatori con *valori nulli* non possono essere principali (non garantiscono l'accesso a tutti i record).
- > Un identificatore *semplice* (pochi attributi) e' da preferire (indici piccoli fanno risparmiare memoria e facilitano la programmazione).
- > Un identificatore *interno* e' preferibile ad uno esterno (stesso motivo del precedente)
- > Un identificatore usato da *molte operazioni* per accedere alle occorrenze di un'entita' e' da preferire (in quanto ottimizza il maggior numero di operazioni)

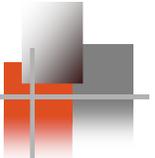
38



Traduzione

- > Completata la fase di ristrutturazione, lo schema E-R ottenuto durante la progettazione concettuale (in parole povere, il disegno) potrebbe avere subito dei cambiamenti, che sono ottimizzazioni fatte per favorire le operazioni e ridurre il consumo di memoria
- > Lo schema E-R così cambiato deve essere tradotto in un insieme di tabelle del modello relazionale (il cosiddetto schema relazionale); per fare questo, si seguono alcune linee guida...

39



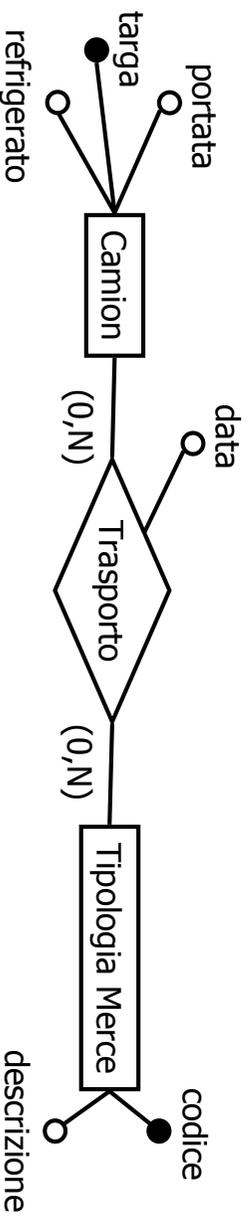
Traduzione Associazione molti a molti (m-n)

- > Nel caso di **associazione molti a molti**, la traduzione segue le regole:
- > Ogni **entita'** coinvolta si traduce in una relazione (tabella) con lo stesso nome dell'entità, avente i suoi stessi attributi e per chiave il suo identificatore.
- > La **relazione** si traduce in una relazione (tabella) con lo stesso nome avente gli stessi attributi e come *chiave* l'insieme di tutti gli *identificatori* delle entita' coinvolte.

40

Traduzione

Associazione molti a molti (m-n)



Ad esempio, dallo schema E-R qui sopra, si ricava lo schema relazionale seguente:

- > *Camion* (*targa*, *portata*, *refrigerato*)
- > *Tipologia Merce* (*codice*, *descrizione*)
- > *Trasporto* (*idcamion*, *idmerce*, *data*)

> *idcamion* e *idmerce* sono rispettivamente *targa* e *codice* nelle entità camion e tipologia merce (è possibile rinominare); tra essi sussiste un vincolo referenziale

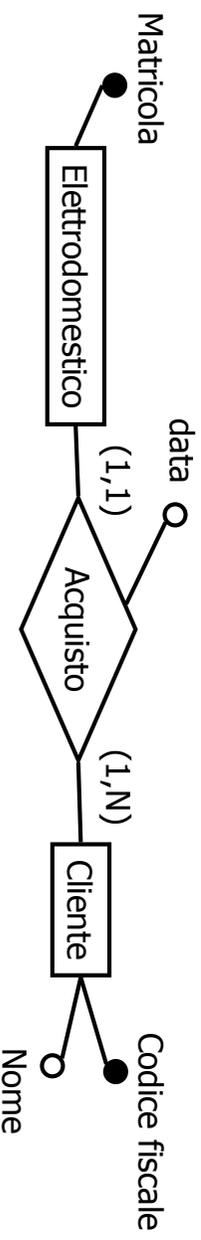
Traduzione

Associazione uno a molti (1-n)

- > Nel caso di **associazione uno a molti**, la traduzione segue le regole:
 - > L'**entità** partecipante con cardinalità (1,1) si traduce in una relazione (tabella) con lo stesso nome dell'entità, avente i suoi stessi attributi, per *chiave* il suo *identificatore* e che ingloba anche la chiave della entità che partecipa con cardinalità (1,n) (tale attributo diventa quindi una chiave esterna), più tutti gli eventuali attributi della associazione (che non avrà quindi una tabella distinta come nel caso m-n).
 - > L'**entità** partecipante con cardinalità (1,n) si traduce in una relazione (tabella) con lo stesso nome dell'entità, avente i suoi stessi attributi, per *chiave* il suo *identificatore*

Traduzione

Associazione uno a molti (1-n)



Lo schema relazionale che si ottiene è il seguente:

Elettrodomestico(Matricola, *Cfcliente*, *dataacquisto*)
Cliente(Codicefiscale, *Nome*)

- > *Cfcliente* (chiave esterna di *Elettrodomestico*) corrisponde a *Codicefiscale*; fra i due quindi sussiste un vincolo di integrità' referenziale
- > La *dataacquisto* corrisponde alla *data* in *Acquisto*

43

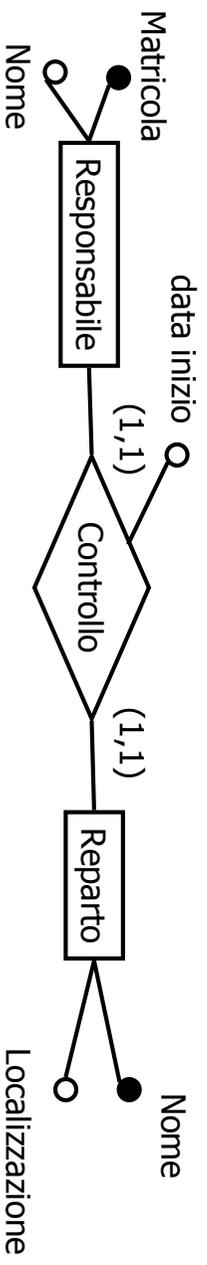
Traduzione

Associazione uno a uno (1-1)

- > Una delle **entità**' si traduce in una relazione (tabella) con lo stesso nome dell'entità, avente i suoi stessi attributi, per *chiave* il suo *identificatore* e che ingloba anche la chiave della altra entità (tale attributo diventa quindi una chiave esterna), più tutti gli eventuali attributi della associazione (come nel caso 1-n, l'associazione non avrà una tabella distinta).
- > L'**entità**' restante si traduce in una relazione (tabella) con lo stesso nome dell'entità, avente i suoi stessi attributi, per *chiave* il suo *identificatore*

44

Traduzione Associazione uno a uno (1-1)



In base alle regole, sono possibili due soluzioni:

Responsabile (Matricola, Nome, NomeReparto, datainizio)

Reparto (Nome, Localizzazione)

NomeReparto (chiave esterna di Responsabile)=Nome di Reparto

oppure

Responsabile (Matricola, Nome)

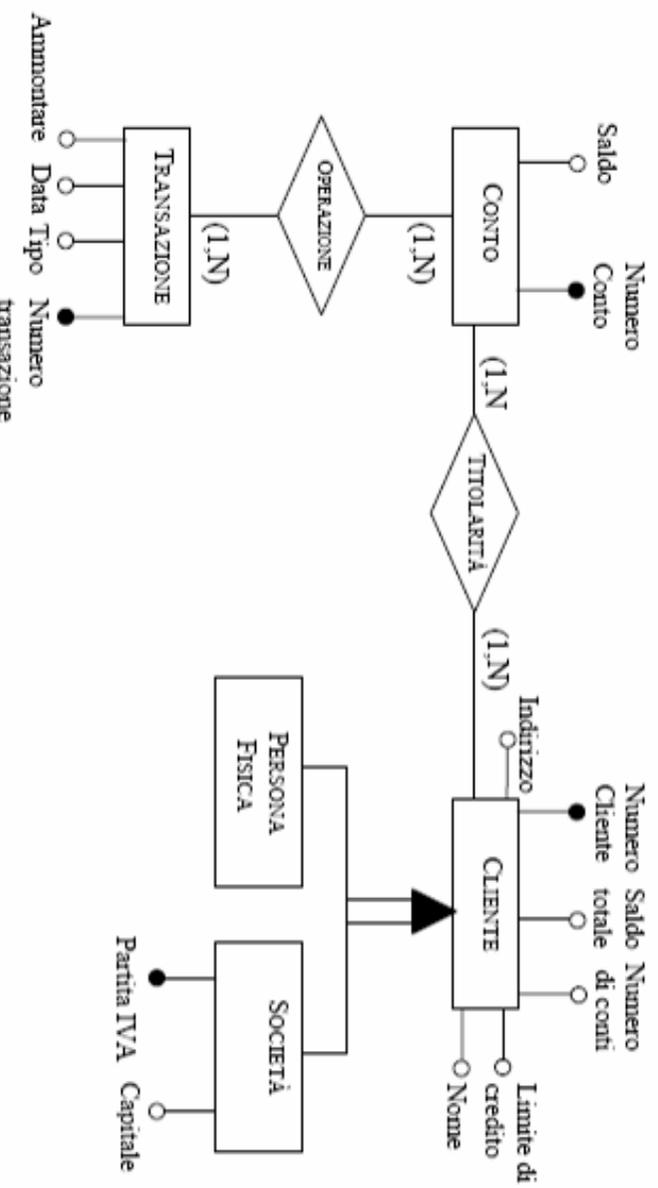
Reparto (Nome, Localizzazione, Responsabile, datainizio)

Responsabile (chiave esterna di Reparto)=Matricola di Responsabile

Quindi sostanzialmente una delle due relazioni si fa carico della relazione con l'altra, inserendo la chiave (esterna) ed il campo *data* della relazione. Esiste in teoria la terza opzione, fondere tutto in un'unica tabella; tale opzione viene generalmente scartata (a meno che non vi siano motivazioni fondate) perché in sede di progettazione concettuale era stato deciso di avere due entità distinte.

Progettazione logica Esempio 1

Esempio di Schema E-R da tradurre:



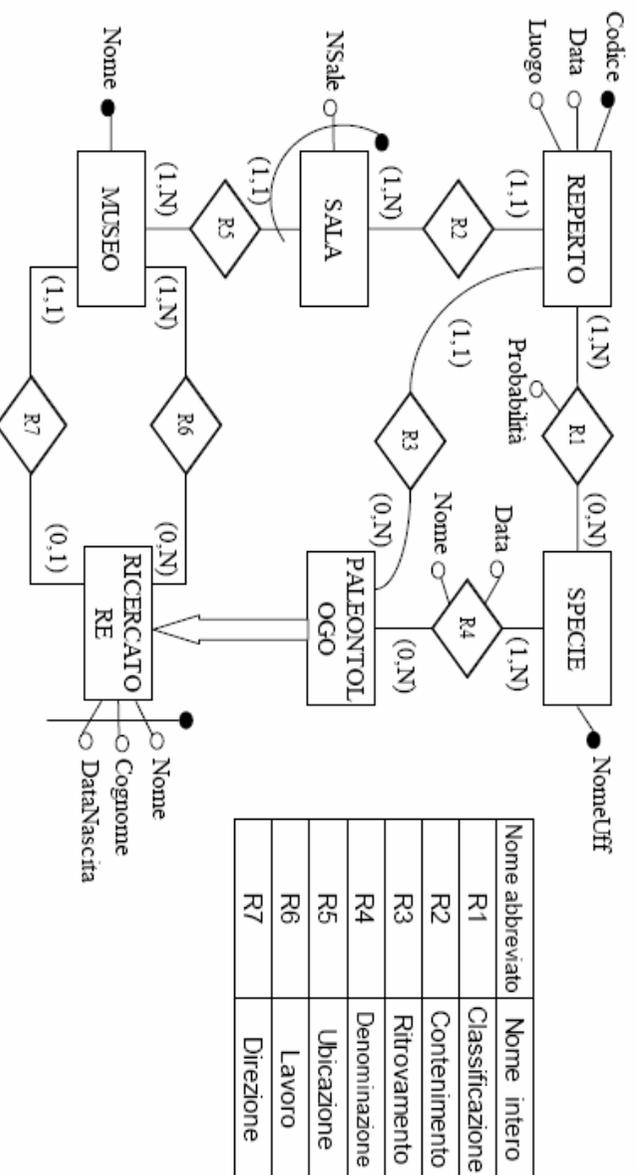
Progettazione logica Esempio 1

Dopo la Ristrutturazione, lo schema logico è il seguente:

TRANSAZIONE(Numero transazione, Tipo, Data, Ammontare)
 CONTO(Numero Conto, Saldo)
 CLIENTE(Numero cliente, Saldo Totale, Limite di credito, Nome, Indirizzo, Partita IVA*, Capitale*)
 OPERAZIONE(Numero conto, Numero transazione)
 TITOLARITÀ(Numero conto, Numero cliente)

Progettazione logica Esempio 2

Lo schema E-R già visto...



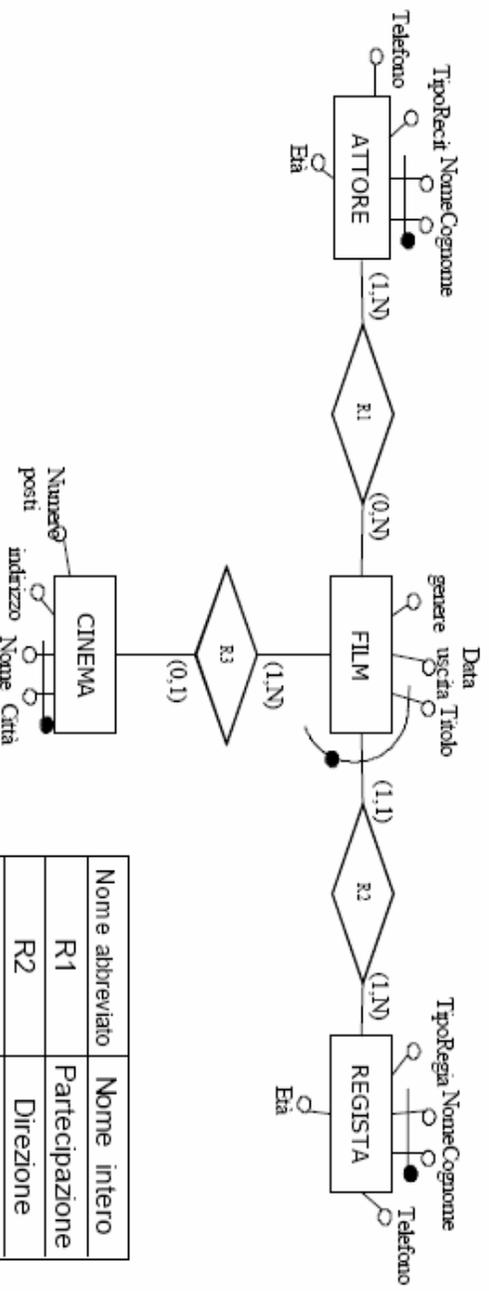
Progettazione logica Esempio 2

...e la sua traduzione in schema logico:

- Reperto (Codice, Data, Luogo, NomeMuseo*, Nsala*, NomeR*, CognR*, DataNR*)
- Sala (NomeMuseo, Nsala)
- Classificazione (Codice, NomeUff, Probabilità)
- Ricercatore (Nome, Cognome, DataNascita, Tipo)
- Museo (Nome, NomeDir*, CognomeDir*, DataNascitaDir*)
- Specie (NomeUff)
- Denominazione (NomeUff, NomeR, CognomeR, DataNascitaR, Nome, Data)
- Lavoro (NomeMuseo, NomeR, CognomeR, DataNascitaR)

I campi sottolineati sono i campi chiave, mentre quelli contrassegnati con * sono le chiavi esterne

Progettazione logica Esempio 3



Nome abbreviato	Nome intero
R1	Partecipazione
R2	Direzione
R3	Proiezione

Progettazione logica Esempio 3

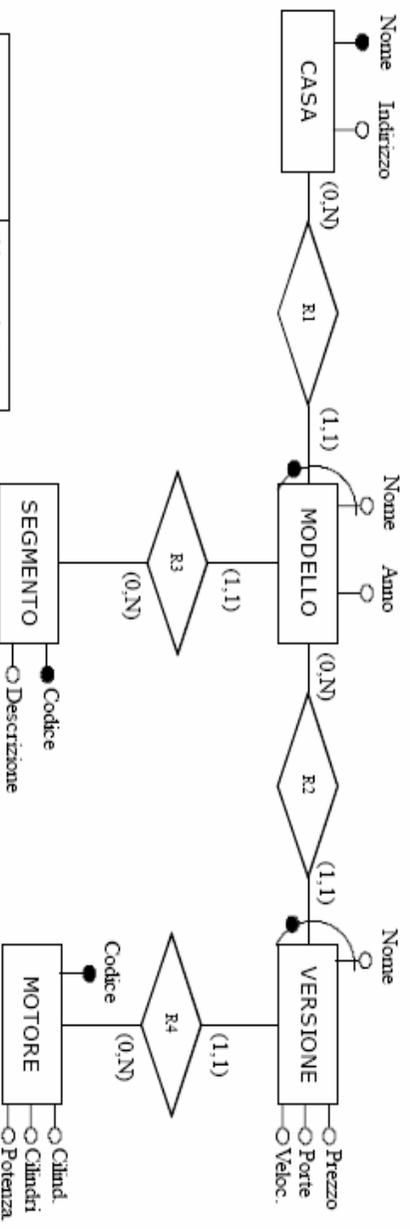
...e la sua traduzione in schema logico:

Attore (Nome, Cognome, tipoR, tel, eta)
 Film(titolo, nomeR, cognR, datauscita, genere)
 Partecipazione(nA, cA, titolofilm, nR, cR)
 Regista(nome, cognome, tipoR, tel, eta)
 Cinema(nome, citta, indirizzo, posti, titolo*, nomeR*, cognR*)

51

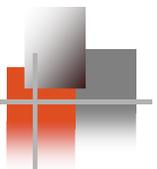
Progettazione logica Esempio 4

Altro schema E-R già visto...



Nome e abbreviato	Nome intero
R1	Produce
R2	Composto
R3	Localizzato
R4	Composta

52



Progettazione logica

Esempio 4

...e la sua traduzione in schema logico:

Casa(Nome, indirizzo)

Modello(Nome, Nomecasa, anno, segmento)

Segmento(codice, descrizione)

Versione(nomeV, nomeM, nomeC, prezzo, porte, vel, motore*)

Motore(codice, cilindrata, numcil, potenzaCV)