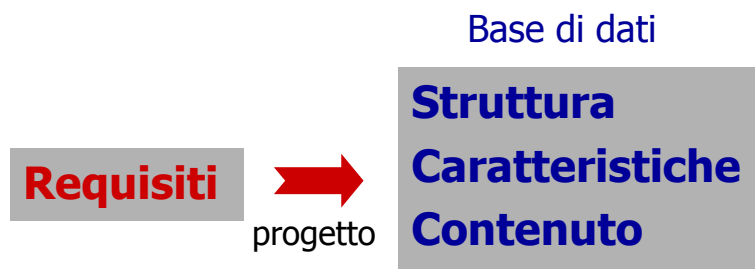


Progettazione di basi di dati



Metodologia in 3 fasi

- Progettazione concettuale
- Progettazione logica
- Progettazione fisica

1

Ciclo di vita dei sistemi informativi

- Comprende tutte le attività svolte da **analisti**, **progettisti**, **utenti** nello sviluppo e utilizzazione dei sistemi informativi
- E' un'attività iterativa... un ciclo di operazioni
 - spesso durante l'esecuzione di una attività occorre **rivedere** decisioni prese nelle attività precedenti
- Le basi di dati sono solo una delle componenti di un sistema informativo

2



Fasi del ciclo di vita [1]

- **Studio di fattibilità**
 - definizione dei costi delle alternative
 - priorità di realizzazione delle componenti del sistema
- **Raccolta ed analisi dei requisiti**
 - Studio delle proprietà e delle funzionalità del sistema
 - Interazioni con gli utenti (a livelli diversi)
 - Analisi delle realizzazioni esistenti
 - Studio della normativa (es. riservatezza dei dati)
 - Descrizione informale dei dati e delle operazioni
 - Requisiti hardware e software

3



Fasi del ciclo di vita [2]

- **Progettazione**
 - Progettazione dei dati
 - Progettazione delle applicazioni

Descrizioni formali per mezzo di modelli per dati e applicazioni
- **Implementazione**
 - Realizzazione del sistema informativo secondo la struttura prevista nella progettazione
 - Costruzione e popolazione della base di dati
 - Sviluppo del codice delle applicazioni

4



Fasi del ciclo di vita [3]

- **Validazione e collaudo**
 - Verifica del corretto funzionamento e della qualità
 - Si devono considerare tutte le possibili condizioni operative
- **Funzionamento**
 - Utilizzo operativo del sistema informativo. Possono essere richieste
 - Correzioni per malfunzionamenti
 - Revisioni delle funzionalità del sistema
 - Installazione, gestione e manutenzione

5



Analisi e progettazione

- L'**analisi** descrive **cosa** rappresentare in una base di dati
 - costruzione di un modello astratto rispetto alle realizzazioni
 - analisi delle procedure aziendali
 - riguarda i **dati** (**specifiche sui dati**) e le **funzioni** (**specifiche sulle operazioni**)
- La **progettazione** definisce **come** automatizzare le procedure
 - organizzazione dei dati in ingresso e uscita
 - architettura hardware e software
 - organizzazione dei moduli software

6



Progettazione di basi di dati [1]

Progettazione concettuale

- Rappresentare le specifiche della realtà di interesse (**specifiche sui dati**) con una descrizione formale
- Si utilizza un **modello concettuale** dei dati
- Il prodotto è uno **schema concettuale** che rappresenta il contenuto informativo della base di dati ed è utile anche a scopo documentativo
- La rappresentazione è indipendente dai modelli utilizzati per rappresentare i dati nei DBMS
- Non ci si interessa a come le informazioni verranno codificate e all'efficienza dei programmi che utilizzeranno queste informazioni

7



Progettazione di basi di dati [2]

Progettazione logica

- Lo schema concettuale viene tradotto nel modello di rappresentazione dei dati utilizzato dal DBMS (es. modello relazionale)
- Si utilizza un **modello logico** dei dati
- Il prodotto è uno **schema logico** dei dati che fornisce una rappresentazione concreta del contenuto della base di dati
- Si effettuano le scelte progettuali per ottimizzare le operazioni che saranno effettuate sui dati
- Per il modello relazionale sono definite tecniche di **normalizzazione** per validare lo schema logico prodotto

8

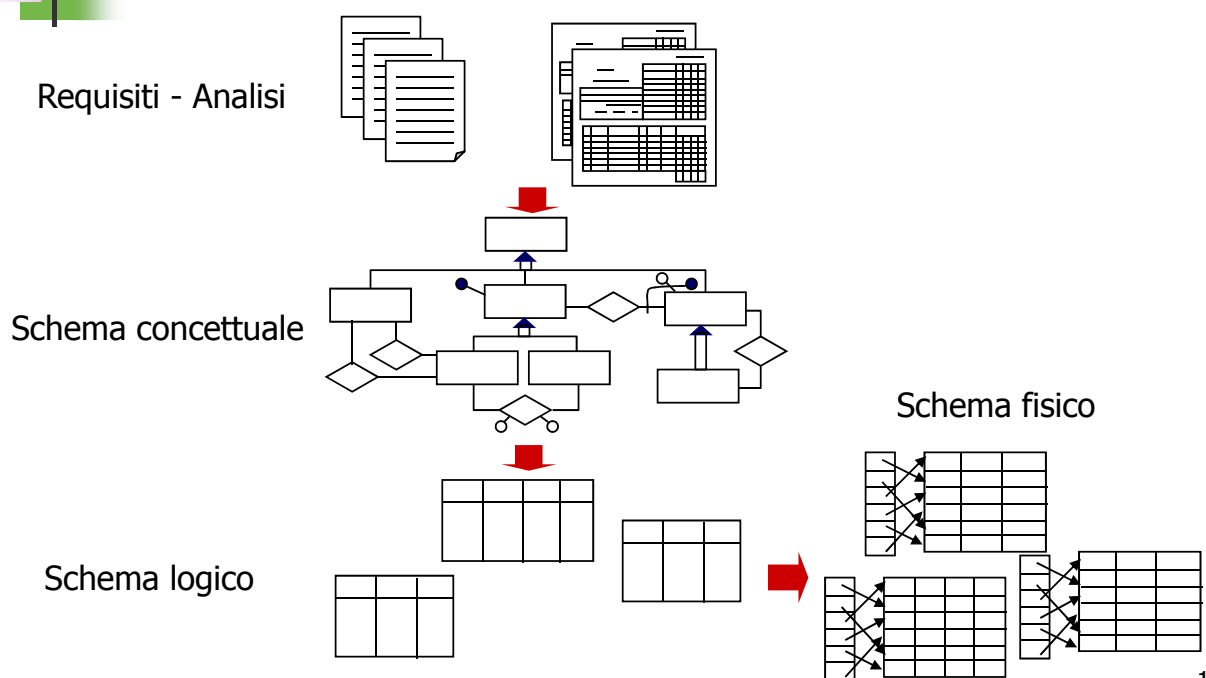
Progettazione di basi di dati [3]

Progettazione fisica

- Si scelgono i parametri fisici di memorizzazione dei dati (organizzazione dei file, definizione degli indici, tipo e dimensioni dei campi)
- Si tiene conto del modello logico e delle specifiche sulle operazioni per ottimizzare le prestazioni del sistema
- Si fa riferimento ad un **modello fisico** dei dati
- Il prodotto è uno **schema fisico** che dipende dal particolare DBMS utilizzato

9

Fasi della progettazione e materiale prodotto



10



Modelli concettuali

- Rappresentazione dei dati in modo indipendente da ogni sistema e rappresentazione dei dati
- Descrizione dei **concetti** del mondo reale
- Rappresentazione mediante **costrutti** di
 - classi dei dati di interesse
 - correlazioni fra le classi di dati
- I costrutti definiscono schemi che descrivono l'organizzazione e la struttura delle istanze dei dati
- Per ogni costrutto esiste una sua rappresentazione grafica

11



Il modello E-R

- Il modello **Entità-Relazione** è uno dei più diffusi modelli concettuali
- I costrutti del modello E-R sono
 - Entità
 - Relazioni
 - Attributi
 - Identificatori
 - Generalizzazioni e sottoinsiemi
- Uno schema E-R è rappresentato con un diagramma grafico

12



Entità

- Classe di oggetti che hanno **proprietà comuni** ed esistenza *autonoma*
 - Studente
 - Docente
 - Corso
 - Dipartimento
 - Facoltà
- Una **istanza** di una entità è un oggetto della classe rappresentata dall'entità
 - Facoltà di Ingegneria

13



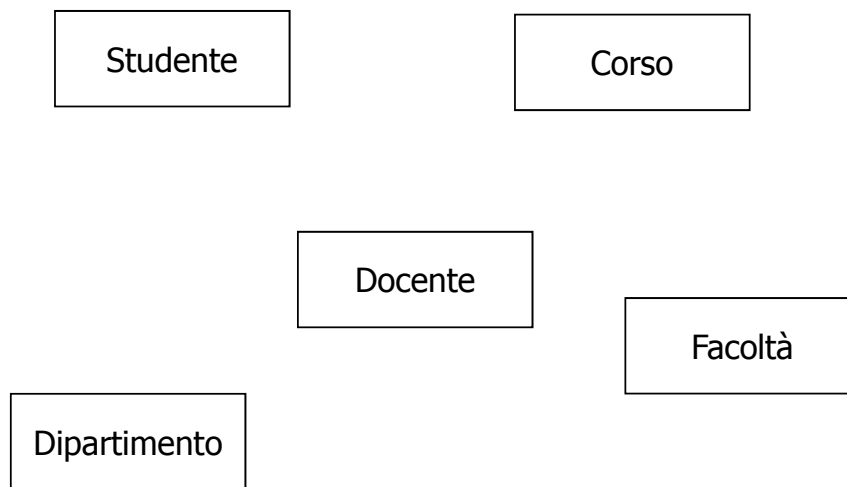
Entità e occorrenze

- Una occorrenza di entità non è un valore (o valori) che identifica un oggetto ma l'oggetto in se stesso
- Una occorrenza di entità esiste indipendentemente dalle proprietà ad essa associate
- Ogni entità ha un nome che la identifica univocamente nello schema concettuale
 - Uso di nomi significativi ed espressivi
 - Rispetto di alcune convenzioni (es. usare il singolare)
- Graficamente le entità sono rappresentate da **rettangoli**

14



Rappresentazione delle Entità



15



Relazione

- Legame logico fra due o più entità
 - Esame lega l'entità Studente con l'entità Corso
- Da non confondere col termine relazione del modello relazionale dei dati
- Una **istanza** di relazione è una n-upla di istanze di entità, una per ciascuna entità coinvolta
 - La coppia (*Mario Rossi*, *Basi di dati*) è un'istanza della relazione binaria Esame fra le entità Studente e Corso se
 - *Mario Rossi* è un'istanza dell'entità Studente
 - *Basi di dati* è un'istanza dell'entità Corso

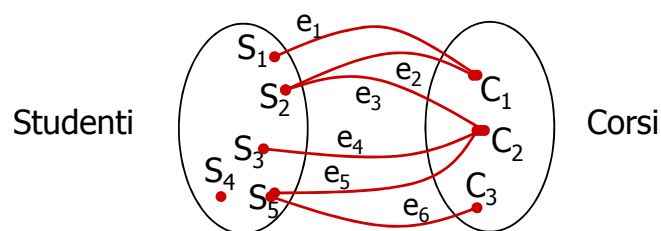
16

Rappresentazione delle relazioni

- In uno schema E-R ogni relazione ha un nome ed è rappresentata graficamente con un **rombo** e da linee che connettono la relazione con le sue componenti



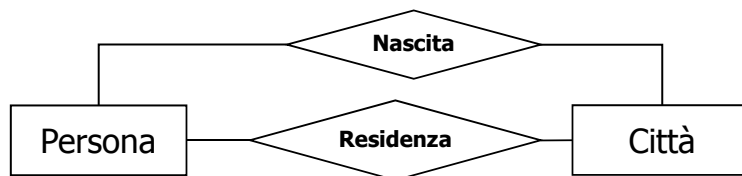
- Esempio di istanze della relazione ESAME



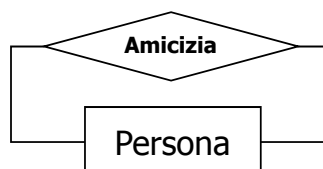
17

Esempi di relazioni

- Due entità possono essere coinvolte in più relazioni



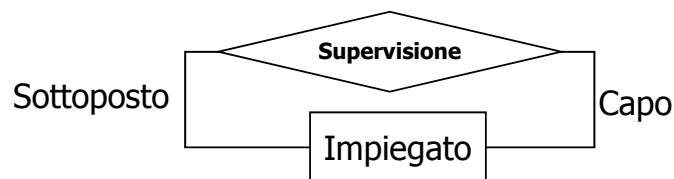
- Una relazione può essere **ricorsiva** ovvero una relazione fra un'entità e se stessa



18

Relazioni con ruoli

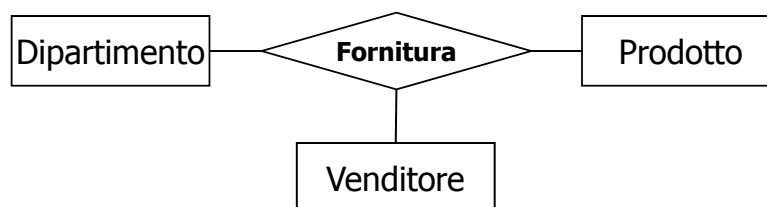
- Una relazione ricorsiva può essere **non simmetrica**, ovvero nella coppia le due istanze dell'entità giocano ruoli diversi
- Si associano degli identificatori alle linee uscenti dalla relazione



19

Relazioni con più di due entità

- Una relazione può coinvolgere più di due entità



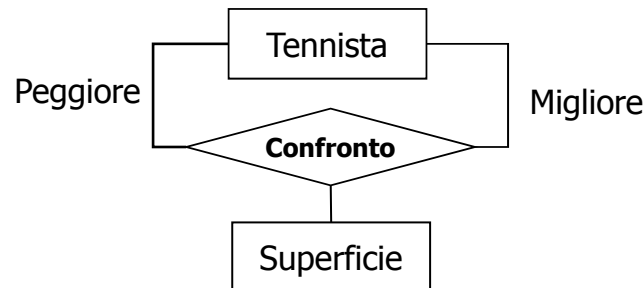
- Le istanze sono triple (Venditore, Prodotto, Dipartimento)

A	fornisce	stampanti	al Dip.	Ingegneria
A		calcolatori		Fisica
B		calcolatori	...	Ingegneria

20

Relazioni con più di due entità e ricorsive

- Possono esistere relazioni ternarie ricorsive



- Le istanze sono triple (Tennista,Tennista,Superficie)

Migliore		Peggiorore		Superficie
Agassi	Migliore di	Becker	su	erba
Chang		Noa	..	terra rossa
Becker		Agassi	..	cemento

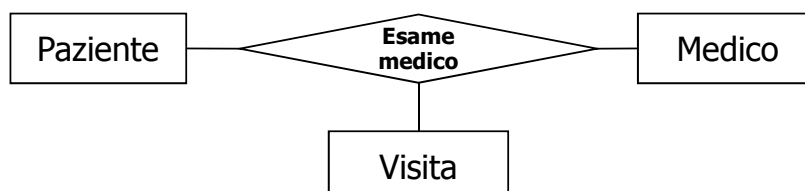
21

Relazioni e valori multipli

- Una relazione è un insieme e come tale può contenere solo istanze distinte (non ci possono essere n -uple ripetute)
 - Sottoinsieme del prodotto cartesiano degli insiemi di istanze coinvolte



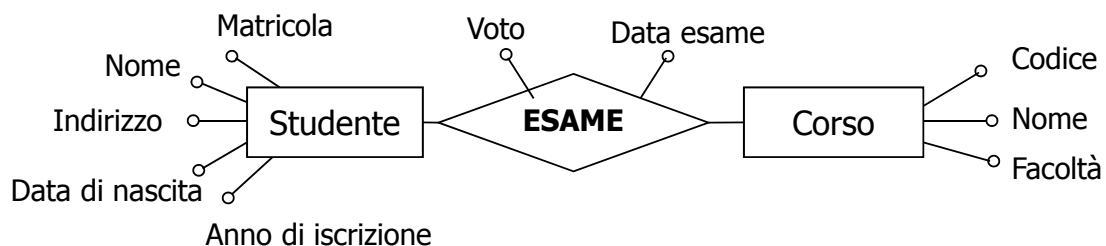
- Un paziente può avere più visite con lo stesso medico
- Occorre trasformare visita in una entità e definire una relazione multipla



22

Attributi

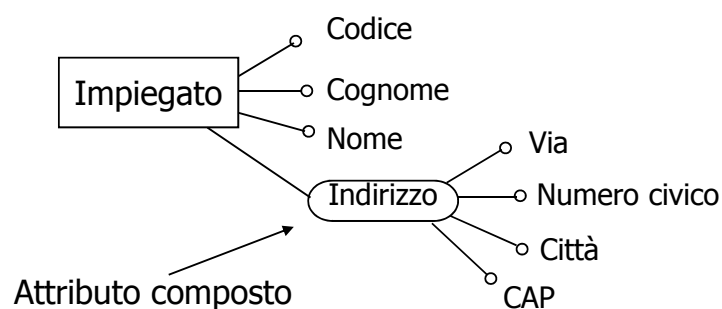
- Descrivono proprietà elementari di entità o relazioni
- Ogni attributo associa a ciascuna istanza un valore appartenente al **dominio** dell'attributo
 - I domini non sono riportati nello schema ma sono descritti nella documentazione associata



23

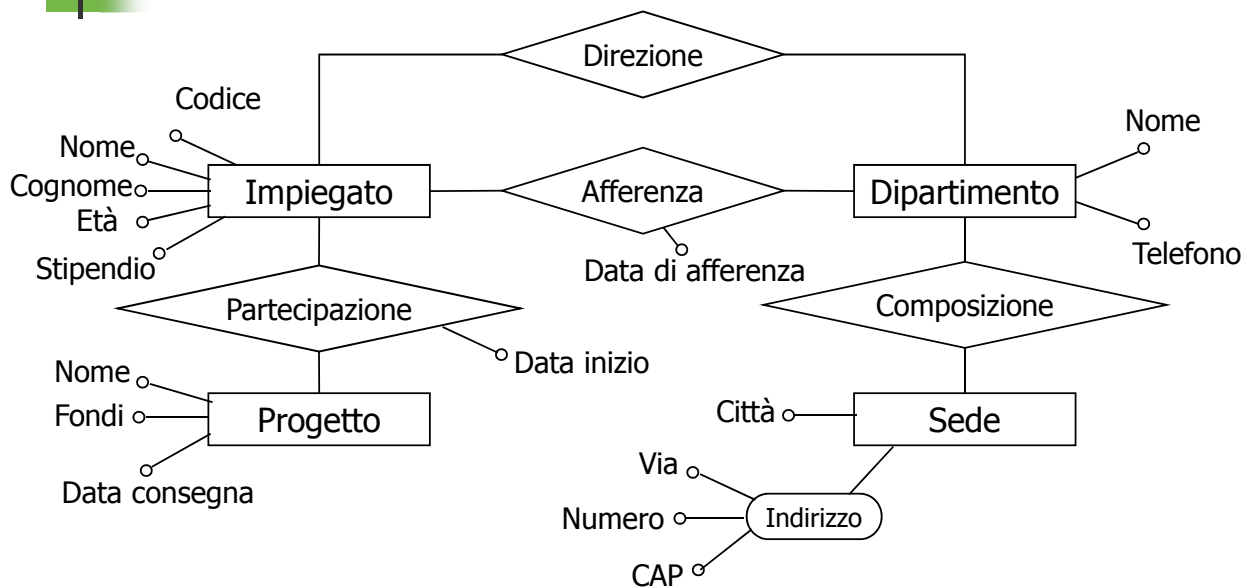
Attributi composti

- Si possono raggruppare gli attributi in base ad una affinità nel loro significato o uso
- L'attributo composto ha un nome che lo individua e un insieme di attributi atomici che lo compongono



24

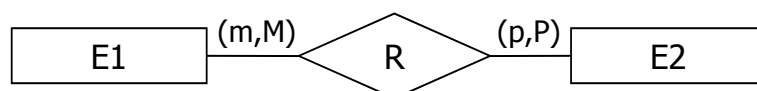
Uno schema ER



25

Cardinalità delle relazioni

- Per ogni partecipazione di una entità ad una relazione si specificano il **numero minimo e massimo** cui un'istanza dell'entità può partecipare
- Rappresenta un **vincolo**



- ogni istanza di E1 partecipa ad almeno m e al più a M istanze di R
- ogni istanza di E2 partecipa ad almeno p e al più a P istanze di R

26

Esempio di cardinalità



- Ad ogni Impiegato è assegnato almeno un incarico
- Ogni Impiegato ha al più 5 incarichi
- Un Incarico può anche non essere ricoperto
- Ad un Incarico possono essere assegnati al massimo 50 Impiegati

27

Tipi di cardinalità

- In genere si specificano solo tre valori per i vincoli di cardinalità: 0, 1, N (intero maggiore di 1)

Cardinalità minima

- 0 - la partecipazione alla relazione è **opzionale**
- 1 - la partecipazione alla relazione è **obbligatoria**

Cardinalità massima

- 1 - rappresenta una funzione (parziale se la cardinalità minima è 0) che associa una sola istanza dell'altra entità
- N - rappresenta una associazione con un numero arbitrario di istanze dell'altra entità

28

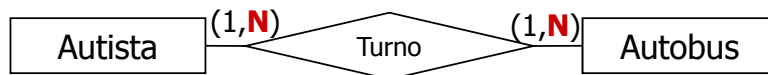
Relazioni "molti a molti"



- Uno studente può non aver fatto esami o un certo numero di esami
- Nessun studente o più studenti possono aver sostenuto l'esame di un corso



- Una montagna può non essere stata scalata o scalata da molti alpinisti
- Un alpinista ha scalato almeno una montagna....



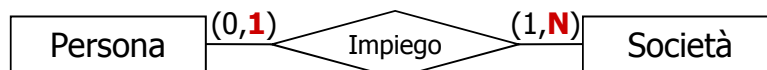
- Un autista ha almeno un turno
- Ogni autobus è usato almeno in un turno

29

Relazioni "uno a molti"



- Ogni Persona ha una sola Città di residenza
- Una Città può non avere Cittadini residenti



- Una Persona è disoccupata o ha un impiego
- Una Società ha almeno un impiegato



- Ogni Comune è in una sola Provincia
- Ogni Provincia ha almeno un Comune

30

Relazioni "uno a uno"



- Un uomo è o meno sposato con una donna
- ... e viceversa



- Ogni Patente ha un unico titolare
- Ogni persona può o meno avere la Patente

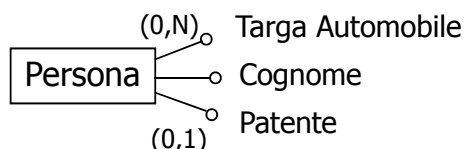


- Ogni Spettatore acquista un solo Biglietto

31

Cardinalità di attributi

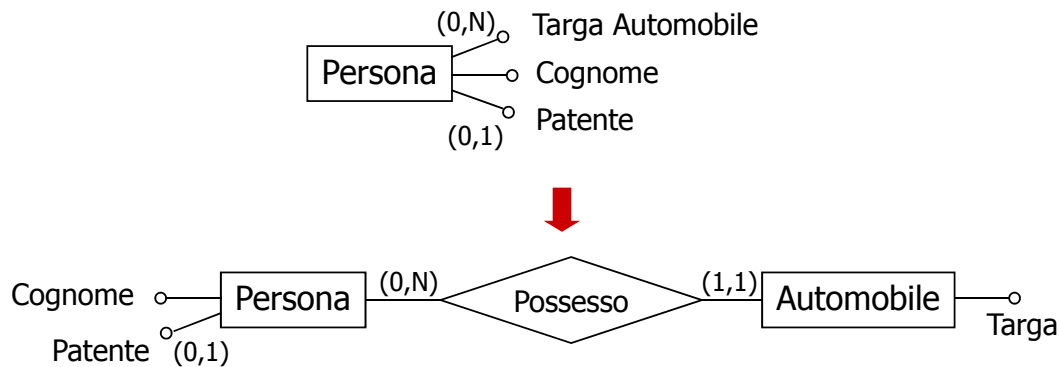
- Per gli attributi si possono specificare il **numero massimo e minimo** di valori dell'attributo associati all'istanza
 - Nella maggior parte dei casi è (1,1) e viene omessa
 - Se la cardinalità è (0,1) l'attributo è **opzionale** (può assumere il valore NULL)
 - Se la cardinalità massima è N l'attributo può avere più valori per una istanza (**attributo multivalore**)



32

Cardinalità di attributi

- Gli **attributi multivalore** possono spesso essere sostituiti con entità legate da relazioni con l'entità a cui si riferiscono



33

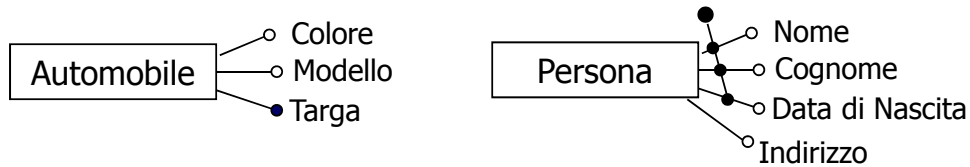
Identificatori delle entità

- Permettono di identificare in modo univoco le istanze dell'entità
- Può essere costituito da
 - **identificatore interno (chiave)**
attributi (uno o più) dell'entità con cardinalità $(1,1)$
 - **identificatore esterno**
entità esterne attraverso relazioni con cardinalità $(1,1)$ ed eventualmente attributi
- Ogni entità deve avere almeno un identificatore
- Le relazioni sono identificate dagli attributi delle entità coinvolte

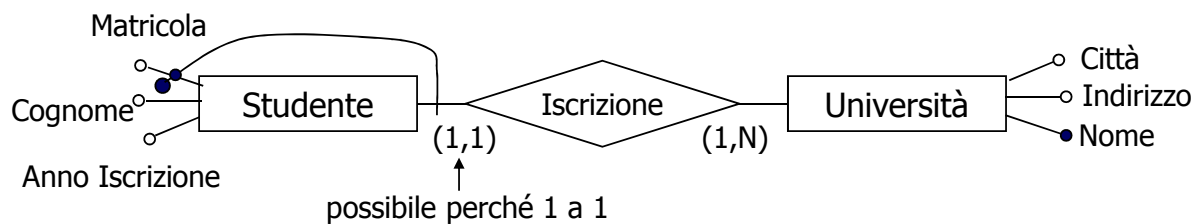
34

Esempi di identificatori

Identificatori interni

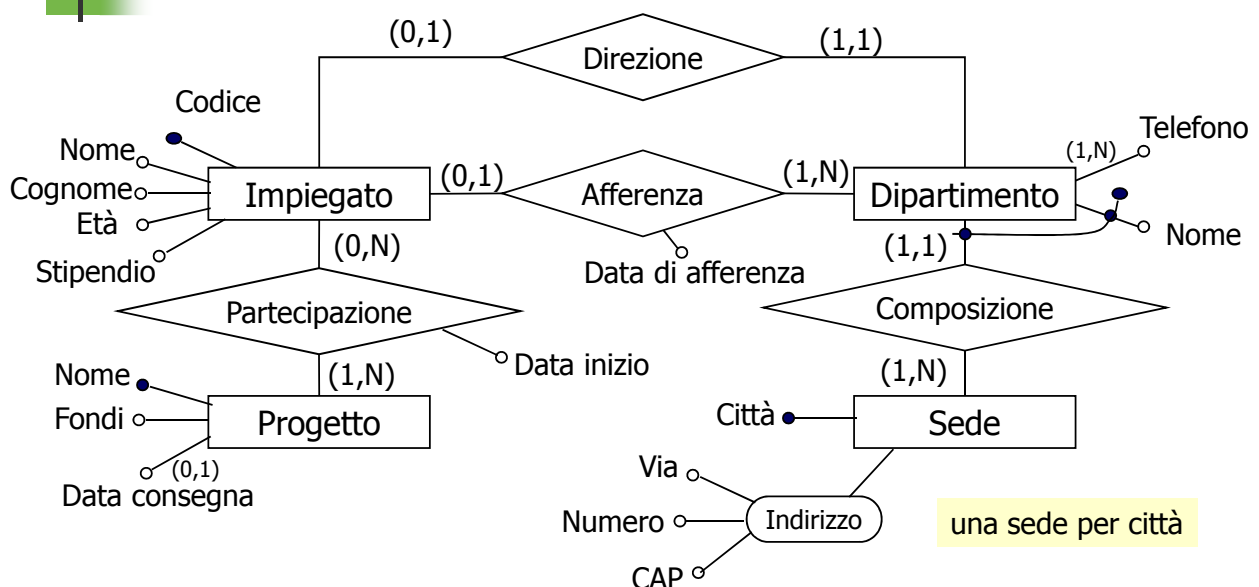


Identificatori esterni



35

Uno schema ER più completo



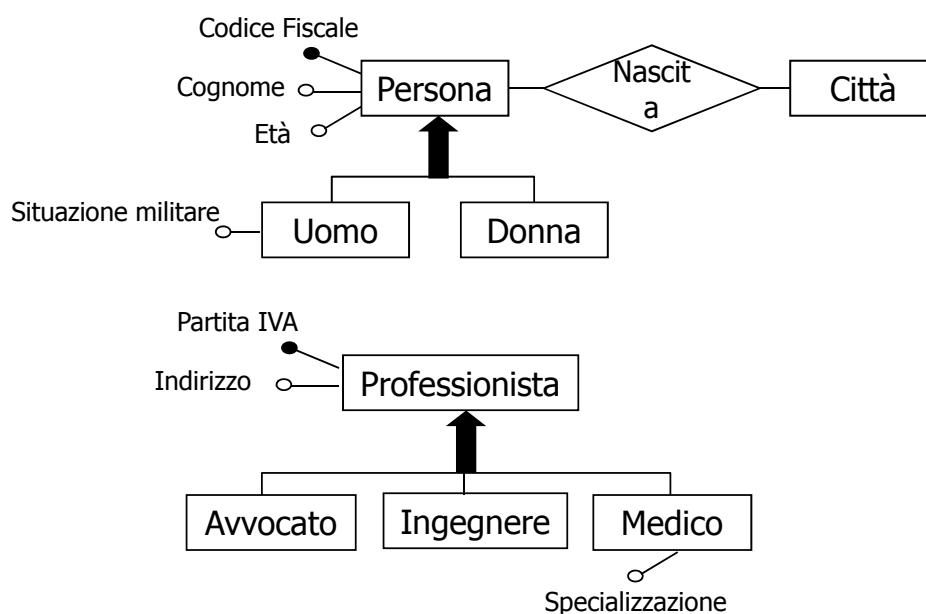
36

Generalizzazione

- Rappresenta un legame fra
 - Entità **padre** E
 - Entità **figlie** $E_1, E_2, \dots, E_n$ (specializzazioni o sottotipi)E è un'entità più generale delle entità figlie, nel senso che le comprende come casi particolari
- Ogni istanza di E_i è anche istanza di E
- Ogni istanza di E è istanza al più di una E_i
- **Ereditarietà**
Ogni proprietà di E (attributi, identificatori, relazioni e altre generalizzazioni) è ereditata anche dalle E_i

37

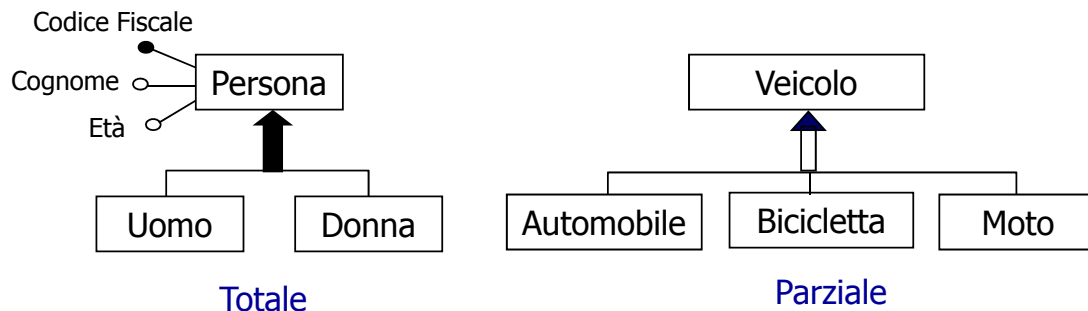
Esempi di generalizzazione



38

Generalizzazione totale

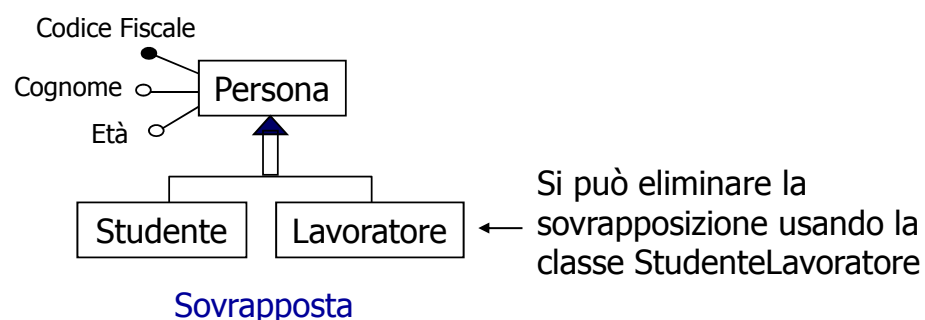
- La generalizzazione è **totale** se ogni istanza della classe padre è una istanza di almeno una delle classi figlie
- Altrimenti si dice **parziale**



39

Generalizzazione esclusiva

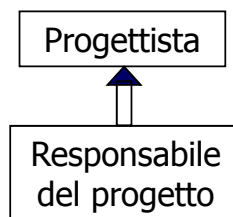
- La generalizzazione è **esclusiva** se ogni istanza della classe padre è una istanza di al più una delle classi figlie
- Altrimenti si dice **sovrapposta**
 - Si possono eliminare le classi sovrapposte introducendo le classi intersezione



40

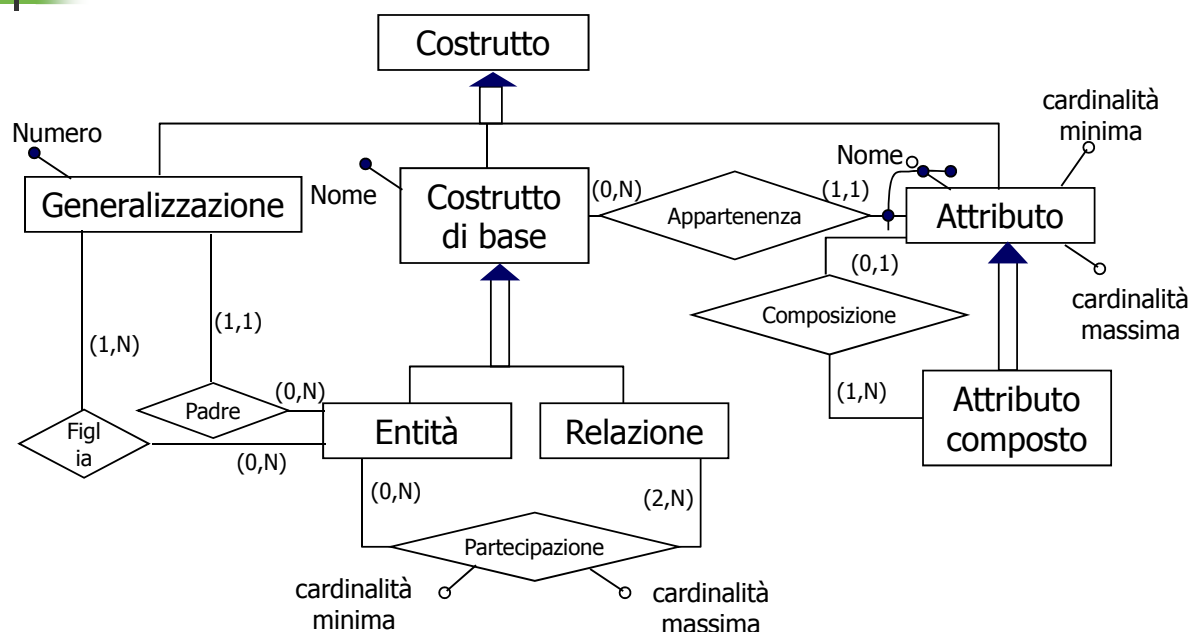
Generalizzazione "is-a"

- Nel caso di generalizzazione con un solo figlio si parla di **sottoinsieme** (gerarchia is-a)
- In questo caso la generalizzazione non può essere totale (le due entità sarebbero coincidenti)

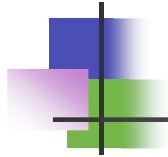


41

Il modello ER in... ER



42



Documentazione

Dizionario dei dati

Spiegazione del significato dei concetti... oltre il nome

- Entità
- Relazioni
- Attributi
- Gerarchie

Vincoli non esprimibili

Impossibilità di rappresentare alcune proprietà dei dati

- Vincoli di integrità dei dati

43



Regole aziendali [1]

- Descrizione delle proprietà che non si riescono a rappresentare con il modello concettuale
- Regole del particolare dominio applicativo

Descrizioni dei concetti

- Si utilizza il linguaggio naturale
- Si costruisce un **glossario** (per entità e relazioni)

44



Regole aziendali [2]

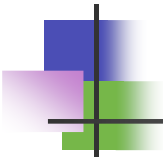
Vincoli di integrità

- Vincoli espressi con un costrutto del diagramma ER (es. cardinalità delle relazioni)
- Vincoli non esprimibili con i costrutti del modello
- Definizione formale o con **linguaggio naturale**

Derivazioni

- Possibilità di ricavare un concetto con una inferenza o un calcolo aritmetico da altri concetti dello schema

45



Asserzioni

- Regole per descrivere i vincoli di integrità
- Sono affermazioni che **devono essere verificate** nella base di dati
- Le affermazioni sono **atomiche**
- Sono enunciate in **modo dichiarativo**

*<concetto> **deve/non deve** <espressione sui concetti>*

I concetti a cui si fa riferimento possono essere presenti direttamente nello schema ER o derivabili

46



Derivazioni

- Le regole specificano le operazioni che permettono di ottenere il concetto derivato

*<concetto> **si ottiene** <operazione sui concetti>*

Esempio

- (RD1) il numero di studenti di una facoltà **si ottiene** contando gli studenti che vi sono iscritti
- (RD2) lo stipendio netto **si ottiene** sottraendo allo stipendio lordo il suo 33%

47



Regole aziendali e DBMS

- Le regole che esprimono vincoli e derivazioni sono implementate nella base di dati
 - Uso di clausole di SQL
(**assertion**, **foreign key** e **references** , **check**, espressioni)
 - Uso di **trigger** o regole attive (basi di dati attive)
Regole *Evento-Condizione-Azione*
(when *Evento* if *Condizione* then *Azione*)
 - Uso di procedure scritte in un linguaggio di programmazione

48

Dizionario dei dati

- E' composto da due tabelle
 - Tabella di descrizione delle entità

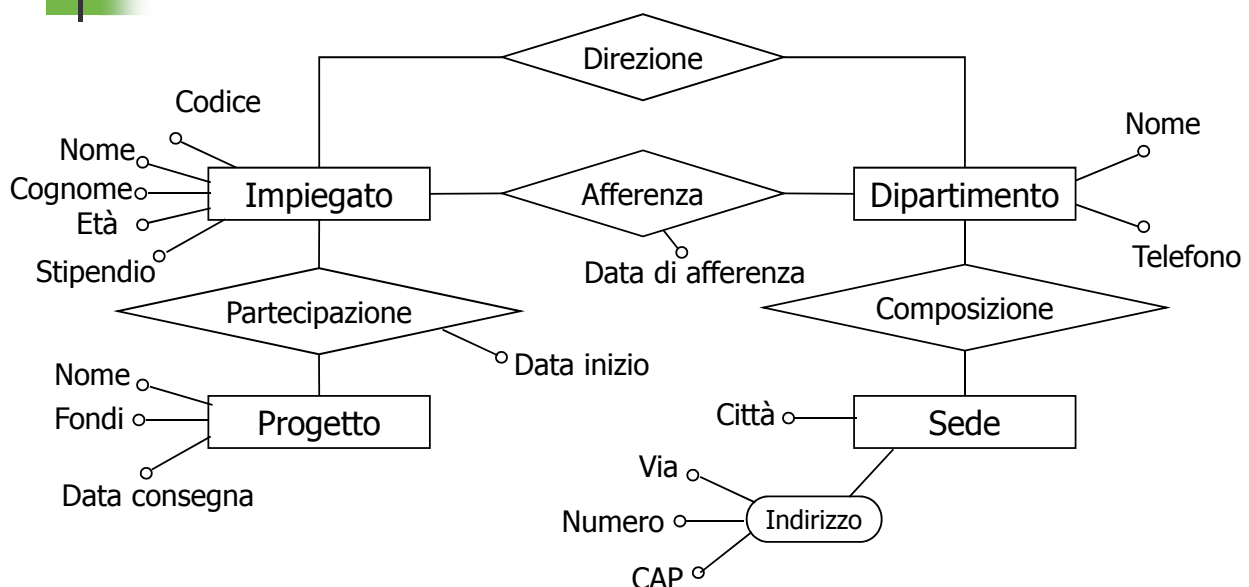
Entità	Descrizione	Attributi	Identificatore
Nome dell'entità	Definizione informale	Elenco di tutti gli attributi con eventuali descrizioni	Possibili identificatori

- Tabella di descrizione delle relazioni

Relazione	Descrizione	Entità coinvolte	Attributi
Nome della relazione	Definizione informale	Elenco delle entità coinvolte con cardinalità	Elenco di tutti gli attributi con eventuali descrizioni

49

Uno schema ER



50

Tabella delle entità

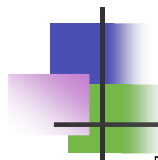
Entità	Descrizione	Attributi	Identificatore
Impiegato	Impiegato che lavora nell'azienda	Codice, Cognome, Nome, Stipendio, Età	Codice
Progetto	Progetti aziendali sui quali lavorano gli impiegati	Nome, Fondi, Data consegna	Nome
Dipartimento	Dipartimenti delle sedi dell'azienda	Telefono, Nome	Nome, Sede
Sede	Sede dell'azienda in una certa città	Città, Indirizzo (Numero, Via e CAP)	Città

51

Tabella delle relazioni

Relazioni	Descrizione	Entità	Attributi
Direzione	Associa un dipartimento al suo direttore	Impiegato (0,1) Dipartimento (1,1)	
Afferenza	Associa un impiegato al suo dipartimento	Impiegato (0,1) Dipartimento (1,N)	Data afferenza
Partecipazione	Associa gli impiegati ai progetti sui quali lavorano	Impiegato (0,N) Progetto (1,N)	Data inizio
Composizione	Associa una sede ai dipartimenti di cui è composta	Dipartimento (1,1) Sede (1,N)	

52



Regole di vincolo e derivazione

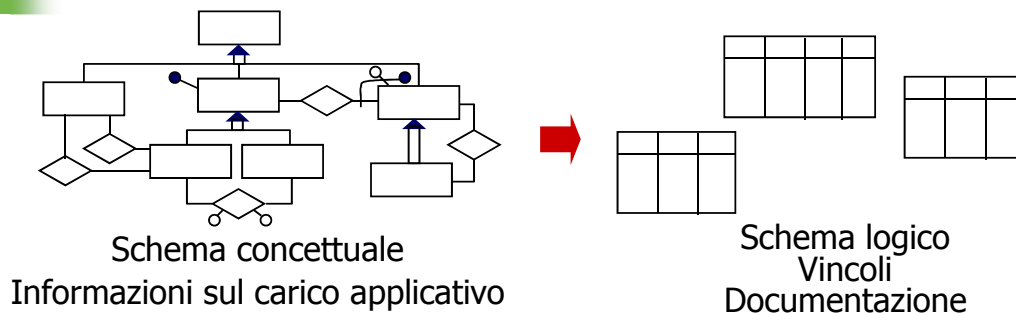
Regole di vincolo

- (RV1) Il Direttore di un dipartimento deve afferire a tale dipartimento
- (RV2) Un impiegato non deve avere uno stipendio maggiore del direttore del dipartimento al quale afferisce
- (RV3) Un dipartimento con sede a Roma deve essere diretto da un impiegato con più di dieci anni di anzianità
- (RV4) Un impiegato che non afferisce a nessun dipartimento non deve partecipare a nessun progetto

Regole di derivazione

- (RD1) I fondi di un progetto si ottengono moltiplicando per 3 la somma degli stipendi degli impiegati che vi partecipano

Progettazione logica



- Costruzione di uno schema logico che descrive in modo corretto e efficiente tutte le informazioni presenti nello schema ER
 - **Input:** schema concettuale + informazioni sul carico applicativo + modello logico prescelto
 - **Output:** schema logico con vincoli e documentazione

1

Verso il modello logico

Riorganizzazione dello schema ER

- **semplificazione della traduzione**
 - non tutti i costrutti del modello ER hanno una traduzione naturale nel modello logico desiderato
 - Entità → relazione del modello relazionale con gli stessi attributi
 - Generalizzazione → ?
- **ottimizzazione del progetto**
 - si deve scegliere la soluzione che garantisce le migliori prestazioni

Traduzione verso il modello logico

- Ottimizzazioni mediante tecniche proprie del modello logico (normalizzazione per il modello relazionale)

2



Analisi delle prestazioni su schemi ER

- Lo schema ER può essere modificato per ottimizzare alcuni **indici di prestazione**
 - **Spazio**: dimensione dei dati
 - **Tempo**: costo di un'operazione valutato come numero di entità e relazioni che mediamente devono essere visitate
- Le prestazioni effettive non sono valutabili perché dipendono da parametri fisici (caratteristiche del DBMS)
- Per studiare le prestazioni occorre conoscere la **dimensione dei dati** e le **caratteristiche delle operazioni**

3



Volume dei dati

Volume dei dati

- numero di istanze per ogni entità e relazione dello schema
- dimensione di ogni attributo
- Nella **tavola dei volumi** si riportano per tutti i concetti (entità e associazioni) i volumi previsti a regime

Concetto	Tipo	Volume
<Nome>	E/R	<dim>

- Per le relazioni il volume dipende da
 - numero di istanze coinvolte nella relazione
 - il numero (medio) di partecipazioni di una istanza di entità alle istanze di relazione (dipende dalla cardinalità delle relazioni)

4

Caratteristiche delle operazioni

Caratteristiche delle operazioni

- tipo dell'operazione (interattiva/batch)
- frequenza (numero medio/unità di tempo)
- dati coinvolti (entità/relazioni)
- Nella tavola delle operazioni si riporta per ogni operazione la frequenza prevista e se l'operazione è **interattiva** (I) o **batch** (B)

Operazione	Tipo	Frequenza
<Nome>	I/B	<freq>

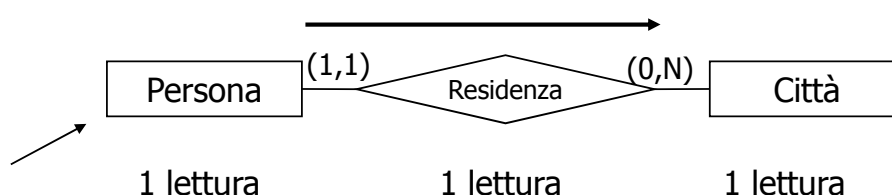
- Si considerano le operazioni principali previste secondo la regola dell'80-20: si suppone che l'80% del tempo dipenda dal 20% delle operazioni più frequenti

5

Schema di operazione

- Lo **schema di operazione** descrive i dati coinvolti in un'operazione
- Corrisponde al frammento dello schema ER interessato all'operazione sul quale viene disegnato il cammino logico per accedere alle informazioni di interesse

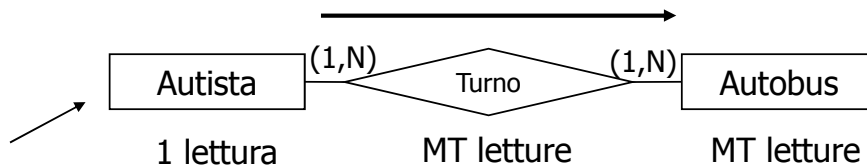
Operazione: data una Persona trovare la città di Residenza



6

Esempio 1

Operazione: dato un Autista trovare gli autobus dei suoi turni

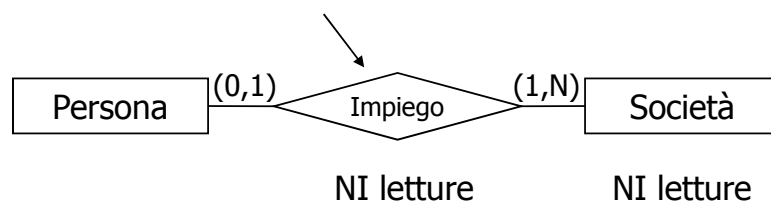


- Per calcolare il numero di accessi a Turno e Autobus occorre conoscere il numero medio di Turni per Autista (MT)

7

Esempio 2

Operazione: Trovare le Società delle Persone che hanno un Impiego

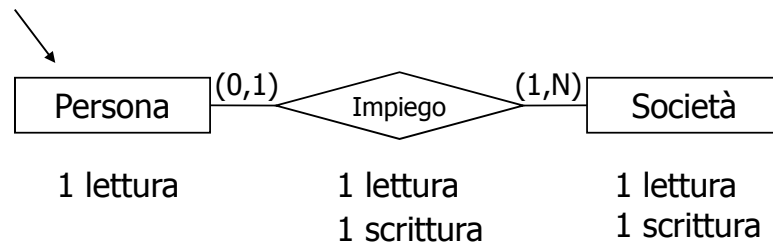


- Per calcolare il numero di accessi a Persona e Società occorre conoscere il numero di Persone occupate (NI)

8

Esempio 3

Operazione: Assegnare ad una Persona nota un Impiego in una Società



- Si deve accedere alla Persona
- Verificare se la Società non è presente e nel caso scriverla
- Verificare se la Persona aveva già un Impiego e (sovra)scrivere la nuova associazione

9

Tavola degli accessi

- Con lo schema di operazione si può fare una stima del costo di un'operazione contando il numero di accessi alle istanze di entità e relazioni
- Il risultato può essere riassunto in una **tavola degli accessi**

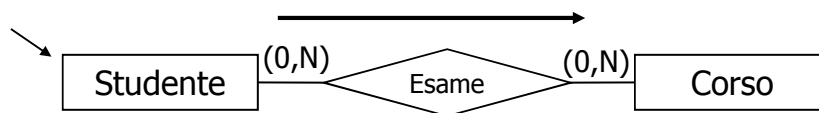
Concetto	Costrutto	Accessi	Tipo

- Il tipo distingue gli accessi in **scrittura** (S) e in **lettura** (L)
- Le operazioni di scrittura sono in genere più onerose (esecuzione in modo esclusivo, aggiornamento degli indici)

10

Esempio 1

Operazione: Stampare il curriculum di uno Studente



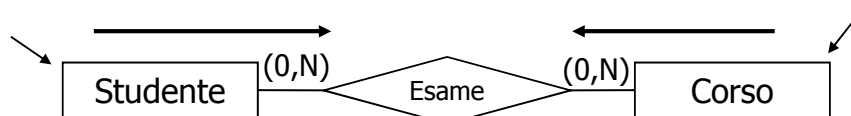
- Si suppone che la media degli esami sostenuti dagli studenti sia 10

Concetto	Costrutto	Accessi	Tipo
Studente	E	1	L
Esame	R	10	L
Corso	E	10	L

11

Esempio 2

Operazione: Registrare un Esame di un Corso ad uno Studente

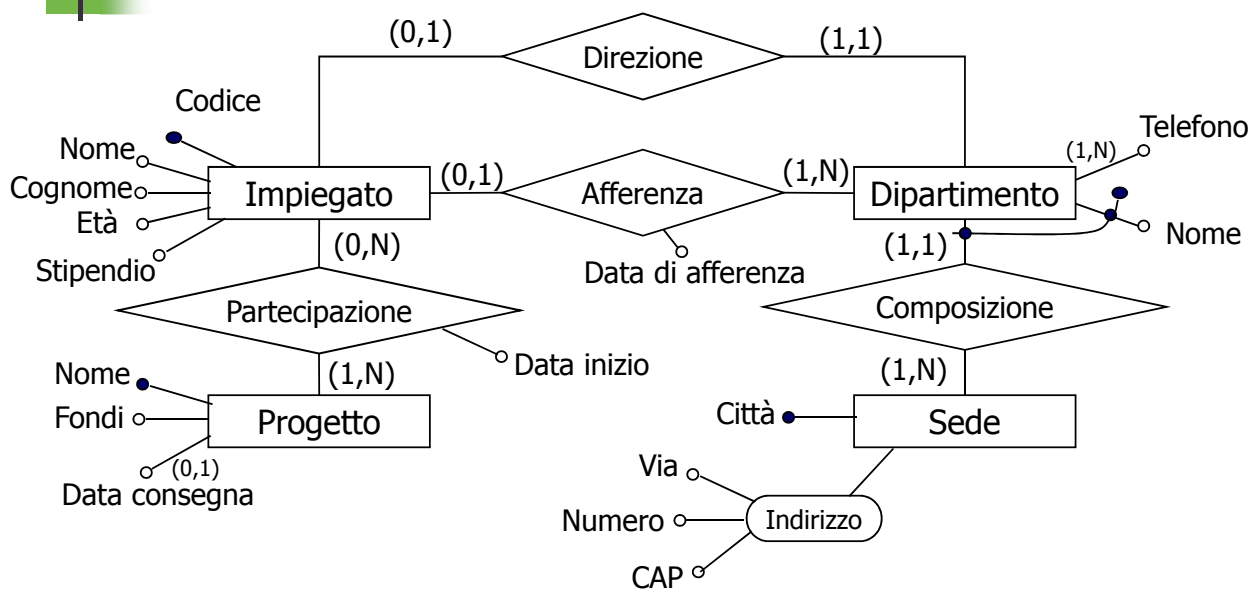


- Si suppone che si inseriscano solo Esami nuovi

Concetto	Costrutto	Accessi	Tipo
Studente	E	1	L
Esame	R	1	S
Corso	E	1	L

12

Schema ER



13

Operazioni

- **Operazione 1:** assegna un impiegato a un progetto
- **Operazione 2:** trova i dati di un impiegato, del dipartimento nel quale lavora e dei progetti ai quali partecipa
- **Operazione 3:** trova i dati di tutti gli impiegati di un certo dipartimento
- **Operazione 4:** per ogni sede, trova i suoi dipartimenti con il cognome del direttore e l'elenco degli impiegati

Operazione	Tipo	Frequenza
Op. 1	I	50/giorno
Op. 2	I	100/giorno
Op. 3	I	10/giorno
Op. 4	B	2/settimana

14

Volumi

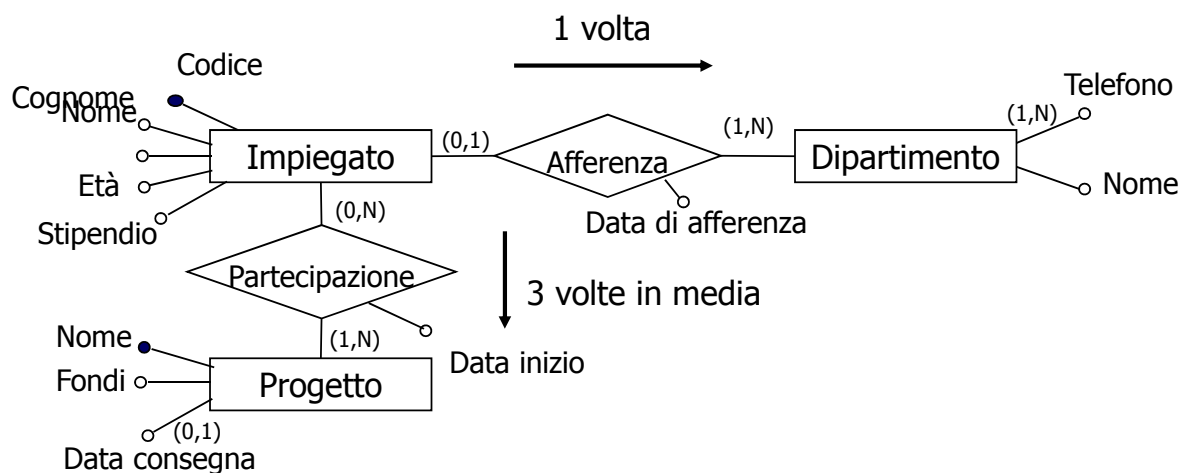
Concetto	Tipo	Volume
Sede	E	10
Dipartimento	E	80
Impiegato	E	2000
Progetto	E	500
Composizione	R	80
Afferenza	R	1900
Direzione	R	80
Partecipazione	R	6000

- **Composizione:** è pari al numero di Dipartimenti (1 Dipartimento - 1 Sede)
- **Afferenza:** è paragonabile (leggermente inferiore) al numero di Impiegati
- **Partecipazione:** si assume che in media un Impiegato partecipi a 3 Progetti
- **Direzione:** ogni Dipartimento ha un Direttore

15

Schema di operazione

- **Operazione 2:** trova i dati di un impiegato, del dipartimento nel quale lavora e dei progetti ai quali partecipa



16



Tavola degli accessi

- Le informazioni richieste sono prelevate da
 - **Impiegato**: 1 accesso
 - **Afferenza**: 1 accesso (ogni Impiegato afferisce al più a un Dipartimento)
 - **Dipartimento**: 1 accesso
 - **Partecipazione**: 3 accessi (in media 1 Impiegato partecipa a 3 Progetti)
 - **Progetto**: 1 accesso
- Tutti gli accessi sono in lettura

Concetto	Costrutto	Accessi	Tipo
Impiegato	E	1	L
Afferenza	R	1	L
Dipartimento	E	1	L
Partecipazione	R	3	L
Progetto	E	3	L

17



Ristrutturazione degli schemi ER

- Schema ER iniziale + specifiche sul carico applicativo
 - **Analisi delle ridondanze**
 - Mantenere/rimuovere le ridondanze dello schema iniziale
 - **Eliminazione delle generalizzazioni**
 - Le generalizzazioni sono analizzate e sostituite da altri costrutti
 - **Partizionamento/accorpamento di entità e associazioni**
 - Suddivisione di un concetto (entità o relazione) in più concetti
 - Accorpamento di più concetti separati in un solo concetto
 - **Scelta degli identificatori primari**
 - Per le entità che hanno più di un identificatore

18

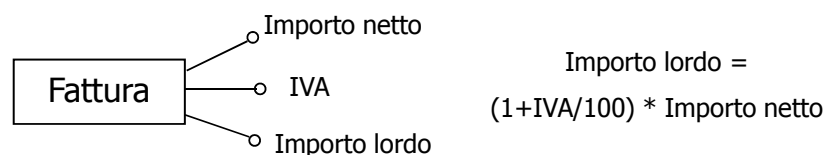
Analisi delle ridondanze

- **Ridondanza** = presenza di un dato che può essere derivato (ottenuto attraverso una serie di operazioni) da altri dati
 - **Vantaggi**
 - riduzione degli accessi necessari per calcolare il dato
 - **Svantaggi**
 - maggiore occupazione di memoria (spesso trascurabile)
 - meno semplice da implementare: necessità di operazioni aggiuntive per mantenere il dato aggiornato
- La decisione viene presa confrontando i costi delle operazioni nel caso di presenza e assenza della ridondanza

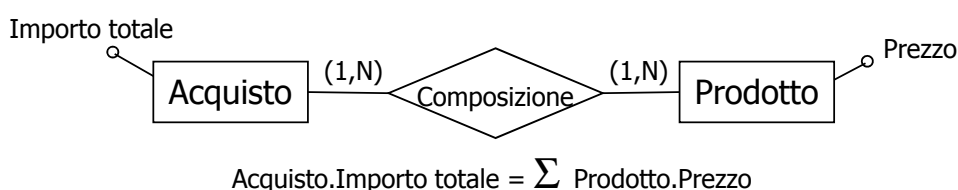
19

Esempi di ridondanze 1

- Attributi derivati da altri attributi della stessa istanza di entità (relazione)



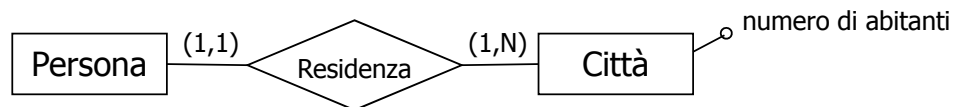
- Attributi derivati da attributi di altre entità (relazione) di solito attraverso funzioni aggregate



20

Esempi di ridondanze 2

- Attributi derivati da operazioni di conteggio di occorrenze



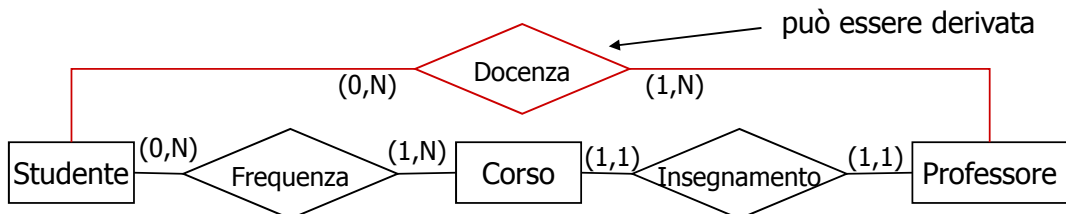
$$\text{Città.numero di abitanti} = \sum \text{Residenza}$$

Data una città si devono contare le istanze della relazione Residenza in cui essa è coinvolta

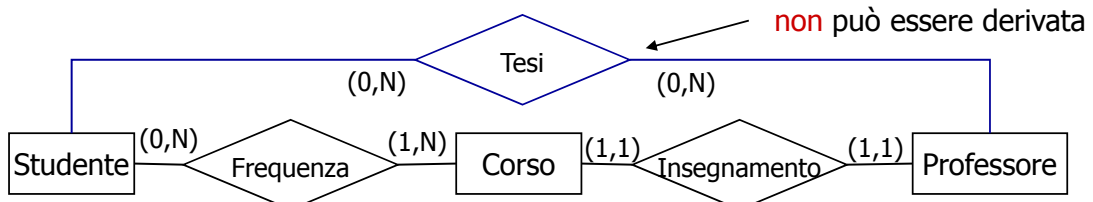
21

Esempi di ridondanze 3

- Associazioni derivate dalla composizione di altre associazioni in presenza di cicli

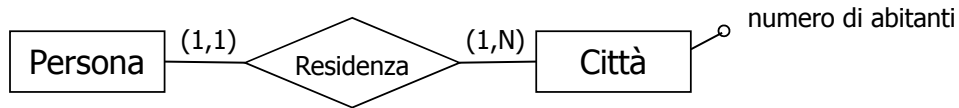


- Occorre però considerare la semantica della relazione coinvolta nel ciclo



22

Esempio di analisi 1



- **Operazione 1:** memorizza una persona con la relativa città di residenza
- **Operazione 2:** stampa tutti i dati di una città

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

Tavola dei volumi

Operazione	Tipo	Frquenza
Operazione 1	I	500/giorno
Operazione 2	I	10/giorno

Tavola delle operazioni

23

Esempio di analisi 2

- **Occupazione di memoria**
 - 1 integer per città (4 byte)
 - $4 \times 200 = 800 \text{ byte} < 1\text{Kb}$
- **Costo delle operazioni**

Operazione 1: memorizza una persona con la relativa città di residenza

con ridondanza

Concetto	Costrutto	Accessi	Tipo
Persona	E	1	S
Residenza	R	1	S
Città	E	1	L
Città	E	1	S

senza ridondanza

Concetto	Costrutto	Accessi	Tipo
Persona	E	1	S
Residenza	R	1	S

legge il numero di abitanti precedente
e scrive il numero aggiornato

24

Esempio di analisi 3

Costo delle operazioni

Operazione 2: stampa tutti i dati di una città

con ridondanza

Concetto	Costrutto	Accessi	Tipo
Città	E	1	L

senza ridondanza

Concetto	Costrutto	Accessi	Tipo
Città	E	1	L
Residenza	R	5000	L

conta in media 1000000/200 abitanti per città

Totali

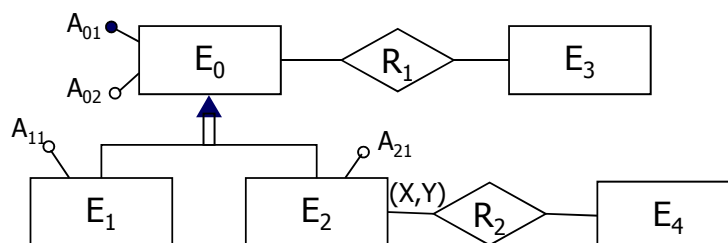
Operazione	L	S
Operazione 1	1 x 500/giorno	3 x 500/giorno
Operazione 2	1 x 10/giorno	0

Operazione	L	S
Operazione 1	1 x 500/giorno	1 x 500/giorno
Operazione 2	5000 x 10/giorno	0

25

Eliminazione delle gerarchie

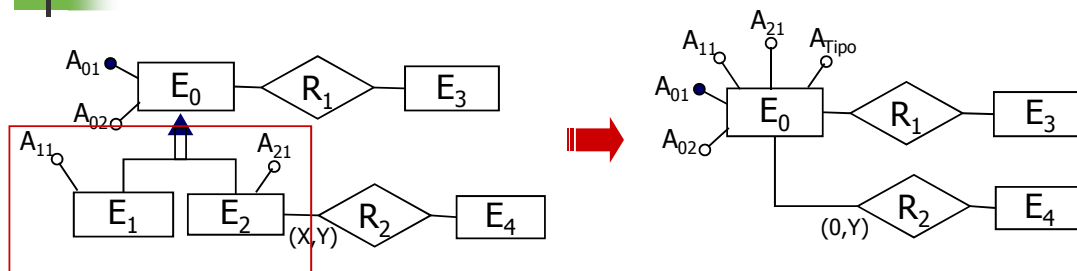
- Il modello relazionale non permette di rappresentare direttamente la generalizzazione



- Occorre trasformare le generalizzazioni in entità e relazioni
 - accorpamento delle figlie della generalizzazione nel genitore
 - accorpamento del genitore della generalizzazione nelle figlie
 - sostituzione della generalizzazione con relazioni

26

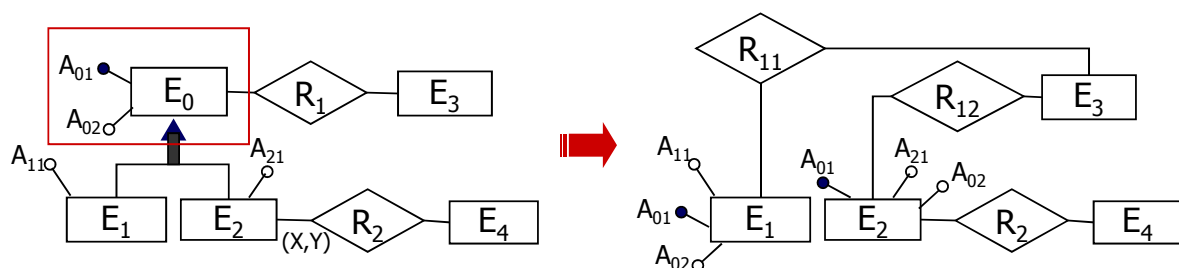
Accorpamento delle figlie



- Le entità figlie sono eliminate
- Gli attributi delle figlie sono aggiunte al padre
- Si aggiunge un attributo per distinguere il **tipo** (quale figlia è - o nessuna)
- Gli attributi che provengono da una figlia possono essere nulli
- Le relazioni (es. R_1) che provengono da una sola figlia hanno cardinalità minima pari a 0

27

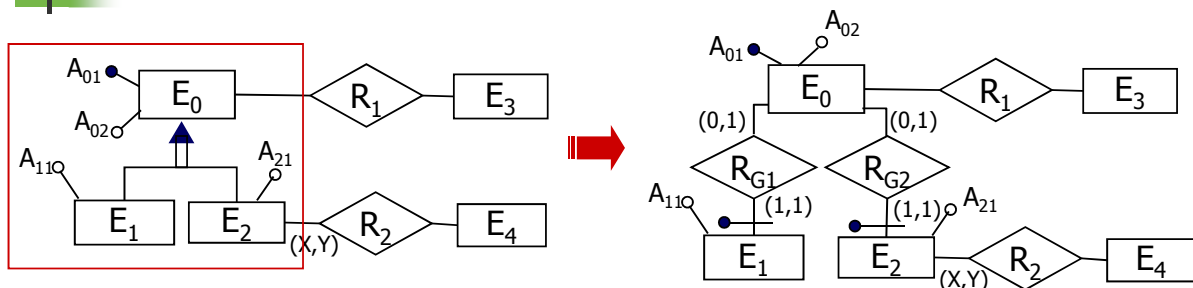
Accorpamento del padre



- L'entità padre è eliminata
- Gli attributi, le relazioni e l'identificatore del padre sono aggiunti alle figlie
- Ogni relazione definita sul padre genera una relazione distinta per ogni figlia

28

Sostituzione con relazioni



- Si introduce una relazione uno-a-uno fra l'entità padre e ciascuna entità figlia
- Occorre inserire il vincolo che ogni istanza dell'entità padre può partecipare solo ad una relazione di legame con le entità figlie
- Se la generalizzazione è totale ogni istanza dell'entità padre partecipa necessariamente ad una (sola) delle relazioni di legame con le figlie

29

Scelta delle alternative 1

Accorpamento delle figlie nel padre

- Contro
 - Si ha spreco di memoria per i valori nulli
- Pro
 - Accesso contestuale agli attributi del padre e della figlia
 - E' la soluzione più semplice

Accorpamento del padre nelle figlie

- Contro
 - Possibile solo se la generalizzazione è totale (non si possono rappresentare istanze che non sono in nessuna delle classi figlie)
- Pro
 - Migliore se si effettuano prevalentemente operazioni solo su istanze di una classe figlia
 - Non ci sono di principio valori nulli

30

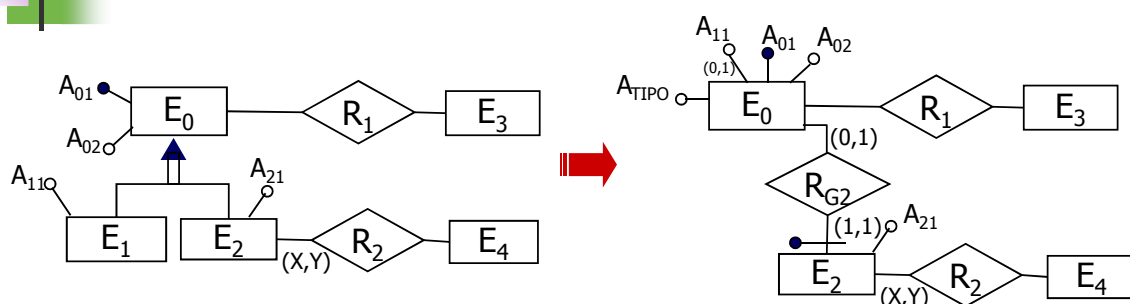
Sceita delle alternative 2

Sostituzione con relazioni

- Contro
 - Si incrementa il numero degli accessi per mantenere la consistenza delle istanze rispetto ai vincoli introdotti
- Pro
 - Conviene quando la generalizzazione non è totale e ci sono operazioni che fanno distinzione fra le entità padre e le entità figlie
 - Non è necessario introdurre valori nulli
 - Genera entità con pochi attributi (le tabelle corrispondenti sono più piccole e più tuple possono essere gestite in memoria principale)

31

Soluzioni miste



- Si è accorpata E_1 col padre E_0
- L'attributo A_{TIPO} serve per distinguere le istanze di E_1 da quelle di E_0
- Per E_2 invece si è introdotta una relazione R_{G2} che rappresenta la generalizzazione

32

Partizionamento/accorpamento

■ Motivazioni

- gli accessi si riducono separando attributi di uno stesso concetto che vengono considerati in operazioni diverse
- gli accessi si riducono raggruppando attributi di concetti diversi a cui si accede contemporaneamente

■ Tipologie

■ Partizionamento di entità

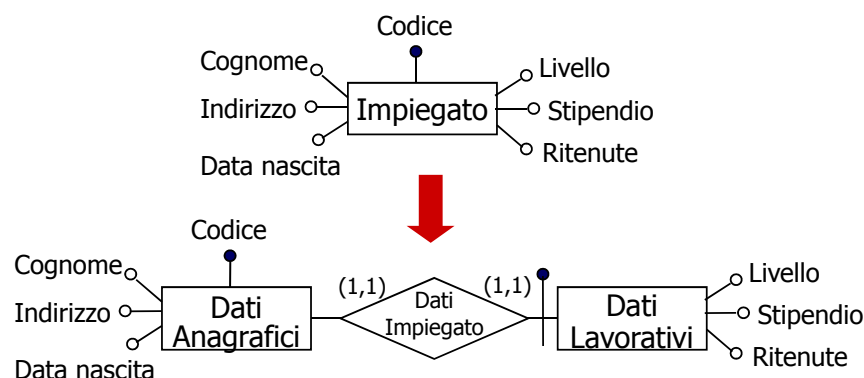
- **Verticale** - opera su attributi
- **Orizzontale** - opera sulle tuple. Corrisponde a introdurre una generalizzazione

■ Partizionamento (orizzontale) di relazioni

■ Accorpamento di entità

33

Partizionamento di entità

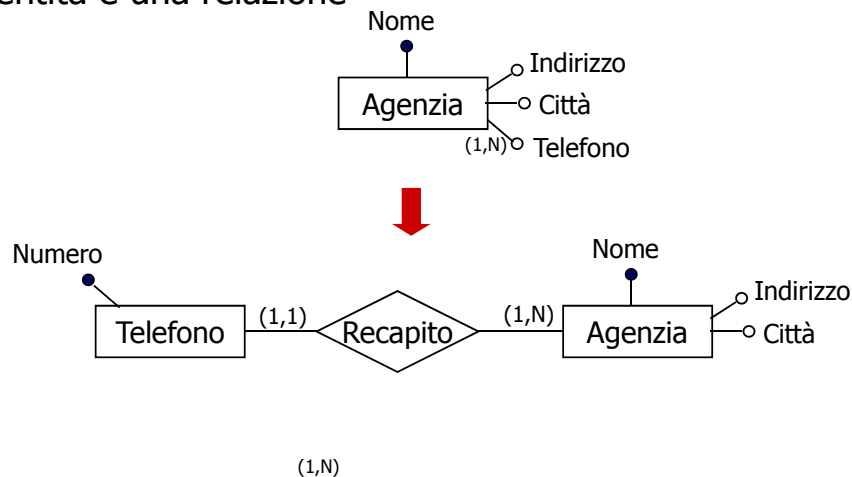


- Conviene se le operazioni richiedono solo informazioni di carattere anagrafico o lavorativo

34

Eliminazione di attributi multivalore

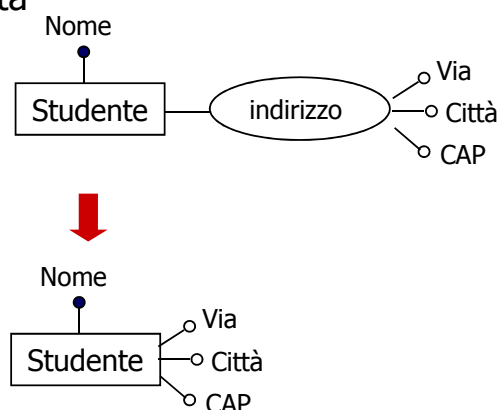
- Il modello relazionale non permette di rappresentare direttamente attributi multivalore
- Si può eliminare un attributo multivalore introducendo una entità e una relazione



35

Eliminazione di attributi composti

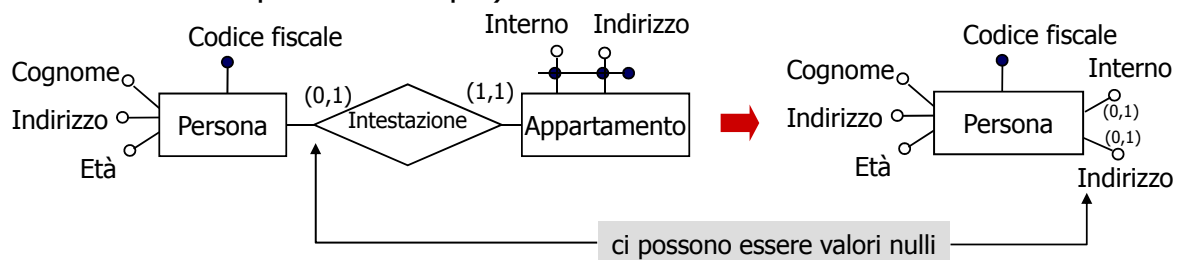
- Il modello relazionale non permette di rappresentare direttamente attributi composto
- Si può eliminare un attributo composto associando gli attributi direttamente all'entità



36

Accorpamento di entità

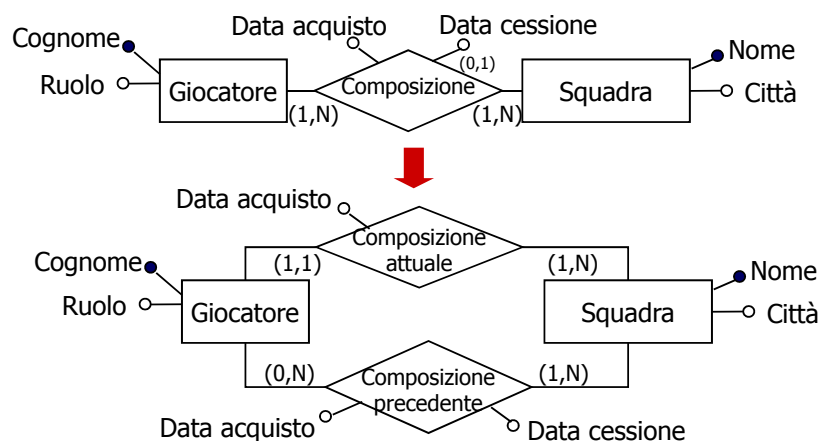
- Due entità **legate da una relazione** possono essere fuse in un'unica entità contenente gli attributi di entrambe quando le operazioni fanno sempre riferimento a tutti gli attributi delle due entità
- In questo modo si risparmiano gli accessi per recuperare i dati attraverso la relazione che lega le due entità
- Si effettuano per **relazioni uno-a-uno** (raramente per uno-a-molti dato che si generano ridondanze: gli attributi delle istanze della prima entità sono ripetuti in N tuple)



37

Partizionamento/accorpamento di relazioni

- Si può **decomporre una relazione** fra due entità in due o più relazioni fra le medesime entità per separare istanze della relazione originale a cui si accede sempre separatamente
- Viceversa si possono **accorpare due relazioni** tra le medesime entità



38



Scelta degli identificatori principali

- Le **chiavi** sono essenziali nel modello relazionale
 - permettono il collegamento fra relazioni
 - la chiave primaria permette la definizione di indici
- Quando un'entità ha più identificatori occorre scegliere la chiave principale (o introdurre una ad hoc → Codice)
 - assenza di valori nulli
 - identificatori con uno o pochi attributi
 - dimensioni indici ridotte, join più semplici
 - preferenza per gli identificatori interni
 - rispetto a identificatori esterni che coinvolgono molte entità
 - identificatori usati nelle operazioni principali

39



Ottimizzazione degli schemi ER: riassumendo

L'ottimizzazione dello schema ER può avere obiettivi diversi a seconda dell'applicazione

1. minimizzare l'occupazione della memoria
2. migliorare l'efficienza nell'esecuzione delle operazioni, minimizzando il numero di accesso alla memoria secondaria
3. semplificare lo schema del database e rendere più semplice il lavoro dei programmatore

Si osservi che con il migliorare dell'efficienza dell'hardware per molte applicazioni i punti 1 e 2 hanno un'importanza limitata. In tali casi la semplificazione dello schema del database diventa l'obiettivo primario!!

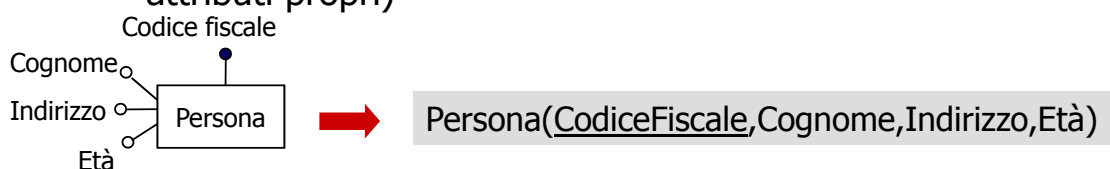
40

Traduzione verso il modello relazionale

41

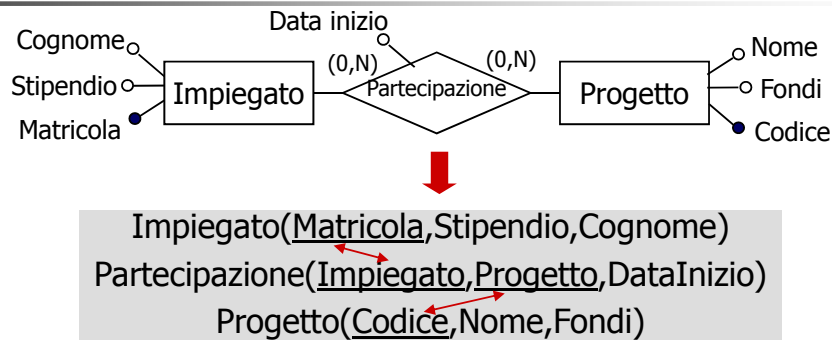
Traduzione verso il modello relazionale

- Si traduce lo schema ER ristrutturato in uno schema relazionale equivalente
- Trasformazioni di base
 - Una entità diviene una relazione definita sugli stessi attributi e con chiave uguale all'identificatore
 - Una relazione ER diventa una relazione sulle chiavi delle relazioni che rappresentano le entità coinvolte (più gli attributi propri)



42

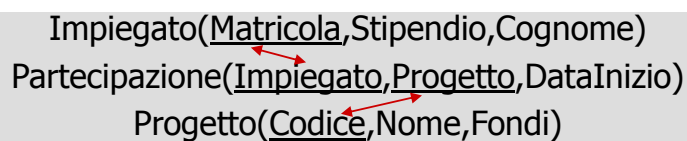
Relazioni multi-a-molti 1



- La coppia di attributi Impiegato, Progetto identifica una istanza della relazione ER Partecipazione (è la chiave)
- Esiste un vincolo di integrità referenziale fra
 - Partecipazione.Impiegato e Impiegato.Matricola
 - Partecipazione.Progetto e Progetto.Codice

43

Relazioni multi-a-molti 2



Impiegato

Cognome	Stipendio	<u>Matricola</u>
Rossi	500	0801
Verdi	800	0802
Bianchi	400	0823

<u>Codice</u>	Fondi	Nome
P0032	3500	Alpha
P0054	7800	Beta
P0012	2400	Gamma

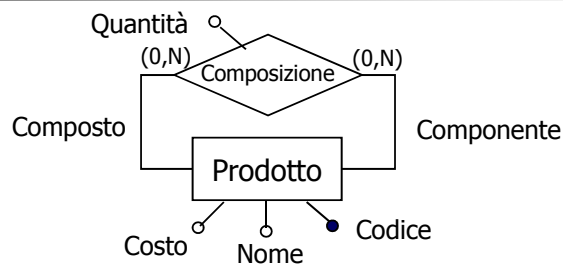
Progetto

<u>Impiegato</u>	<u>DataInizio</u>	<u>Progetto</u>
0801	22/3/2001	P0032
0801	12/11/2001	P0012
0802	10/7/2001	P0054

Partecipazione

44

Relazioni multi-a-molti 3



Prodotto(Codice, Nome, Costo)
 Composizione(Composto, Componente, Quantità)

- Esiste un vincolo di integrità referenziale fra
 - Composizione.Composto e Prodotto.Codice
 - Composizione.Componente e Prodotto.Codice

45

Relazioni multi-a-molti 4

Prodotto(Codice, Nome, Costo)
 Composizione(Composto, Componente, Quantità)

Prodotto

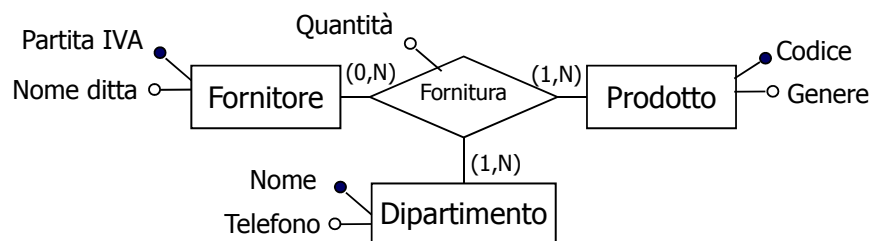
Nome	Costo	<u>Codice</u>
Cilindro	500	C003
Cuscinetto	800	C023
Sfera	300	V823

<u>Composto</u>	<u>Componente</u>	Quantità
C023	C003	2
C023	V823	20

Composizione

46

Relazioni multi-a-molti 5



Fornitore(PartitaIVA, NomeDitta)
 Prodotto(Codice, Genere), Dipartimento(Nome, Telefono)
 Fornitura(Fornitore, Prodotto, Dipartimento, Quantità)

- Esiste un vincolo di integrità referenziale fra
 - Fornitura.Fornitore e Fornitore.PartitaIVA
 - Fornitura.Prodotto e Prodotto.Codice
 - Fornitura.Dipartimento e Dipartimento.Nome

47

Relazioni multi-a-molti 6

Fornitore(PartitaIVA, NomeDitta)
 Prodotto(Codice, Genere), Dipartimento(Nome, Telefono)
 Fornitura(Fornitore, Prodotto, Dipartimento, Quantità)

NomeDitta	<u>PartitaIVA</u>					<u>Codice</u>	Genere
Rossi	08009382					A0034	Computer
Verdi	07092913					B3456	Stampante
Bianchi	04563281					V0567	Video

Fornitore

<u>Fornitore</u>	<u>Prodotto</u>	<u>Dipartimento</u>	Quantità
07092913	A0034	Fisica	5

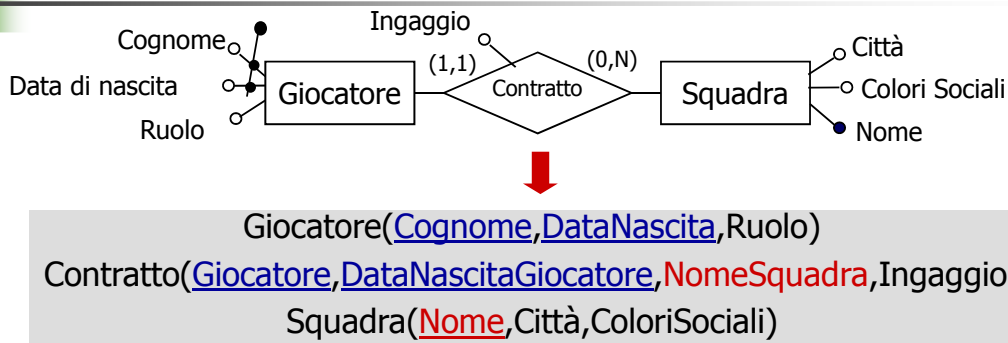
Prodotto

<u>Nome</u>	Telefono
Ingegneria	0577233601
Fisica	0577232471
Biologia	0577234632

Dipartimento

48

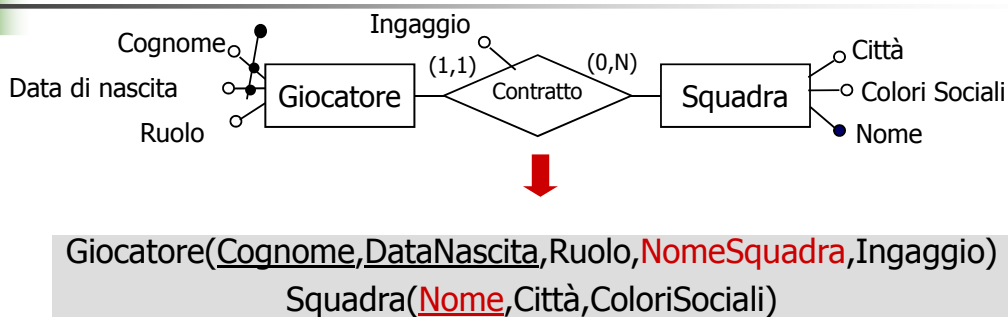
Relazioni uno-a-molti 1



- La chiave per la relazione Contratto è solo l'identificatore dell'entità Giocatore perché la cardinalità verso la relazione è (1,1)
- Le relazioni Giocatore e Contratto hanno un'unica chiave e quindi possono essere fuse insieme

49

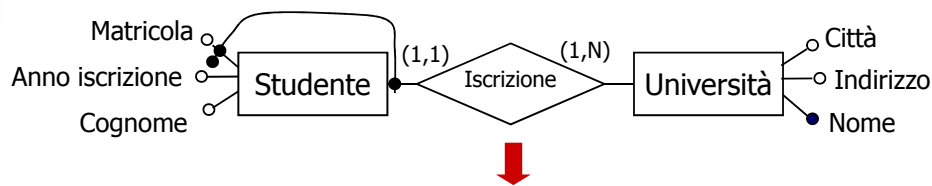
Relazioni uno-a-molti 2



- Esiste un vincolo di integrità referenziale fra
 - Giocatore.NomeSquadra e Squadra.Nome
- La soluzione è valida anche se la cardinalità della relazione Contratto verso Giocatore è (0,1)
 - Si hanno giocatori con l'attributo NomeSquadra uguale a NULL

50

Relazioni con identificatore esterno 1



Studente(Matricola, NomeUniversità, Cognome, AnnoIscrizione)
 Università(Nome, Città, Indirizzo)

- Esiste un vincolo di integrità referenziale fra
 - Studente.NomeUniversità e Università.Nome
- Rappresentando l'identificatore esterno viene rappresentata anche la relazione fra le due entità

51

Relazioni con identificatore esterno 2

Studente(Matricola, NomeUniversità, Cognome, AnnoIscrizione)
 Università(Nome, Città, Indirizzo)

Studente

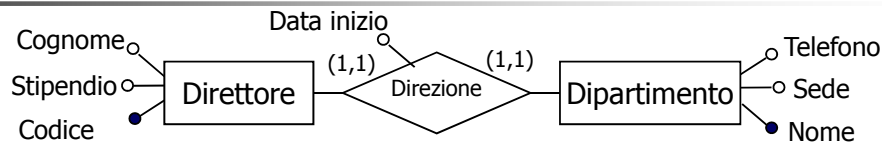
<u>Matricola</u>	<u>NomeUniversità</u>	Cognome	AnnoIscrizione
A80023633	UNISI	Rossi	1993
A80023633	UNIFI	Verdi	1996
A80023635	UNISI	Bianchi	2000

<u>Nome</u>	Città	Indirizzo
UNISI	Siena	Via Banchi di sotto
UNIFI	Firenze	P.zza S. Marco

Università

52

Relazioni uno-a-uno 1



Direttore(Codice, Cognome, Stipendio, DipartimentoDiretto, InizioDirezione)
 Dipartimento(Nome, Telefono, Sede)

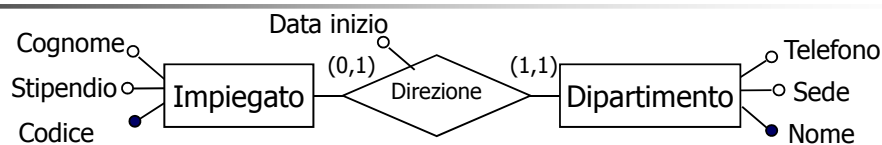
oppure

Direttore(Codice, Cognome, Stipendio)
 Dipartimento(Nome, Telefono, Sede, Direttore, InizioDirezione)

- La relazione è biunivoca e quindi può essere rappresentata in una qualsiasi delle relazioni che rappresentano le entità
- Si potrebbe anche definire un'unica relazione che fonde entrambe ma si perde la distinzione fra le due entità

53

Relazioni uno-a-uno 2

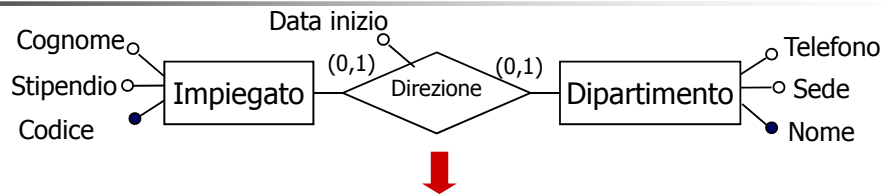


Impiegato(Codice, Cognome, Stipendio)
 Dipartimento(Nome, Telefono, Sede, Direttore, InizioDirezione)

- In questo caso la partecipazione della prima entità è opzionale
- Rappresentare la relazione Direzione in Dipartimento piuttosto che Impiegato è preferibile perché nell'altro caso si avrebbero molti valori nulli

54

Relazioni uno-a-uno 3



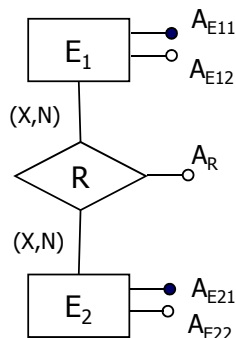
Impiegato(Codice,Cognome,Stipendio)
 Direzione(Direttore,Dipartimento,DataInizioDirezione)
 Dipartimento(Nome,Telefono,Sede)

- In questo caso la partecipazione di entrambe le entità è opzionale
- Questa soluzione ha il vantaggio di non presentare valori nulli sugli attributi
- Occorre però introdurre una relazione in più
- Conviene se il numero di istanze che partecipano alla relazione sono poche

55

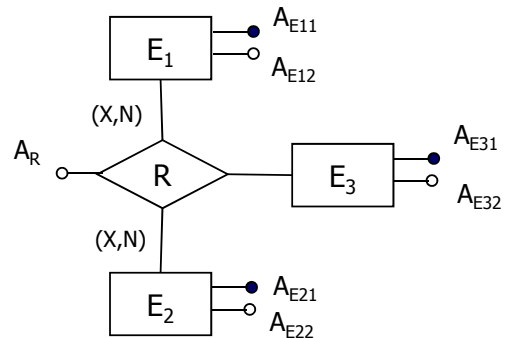
Traduzione delle relazioni 1

Relazione binaria
multi-a-molti



$E_1(A_{E11}, A_{E12})$
 $R(A_{E11}, A_{E21}, A_R)$
 $E_2(A_{E21}, A_{E22})$

Relazione ternaria
multi-a-molti

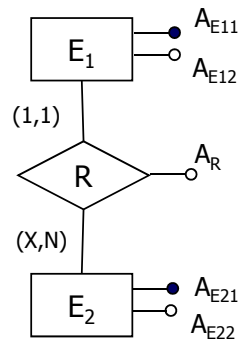


$E_1(A_{E11}, A_{E12})$
 $E_2(A_{E21}, A_{E22})$
 $E_3(A_{E31}, A_{E32})$
 $R(A_{E11}, A_{E21}, A_{E31}, A_R)$

56

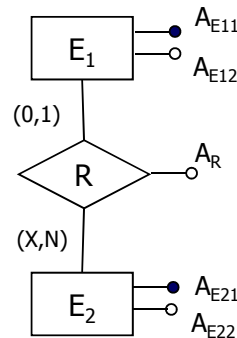
Traduzione delle relazioni 2

Relazione uno-a-molti
con partecipazione obbligatoria



$E_1(\underline{A_{E11}}, A_{E12}, A_{E21}, A_R)$
 $E_2(\underline{A_{E21}}, A_{E22})$

Relazione uno-a-molti
con partecipazione opzionale



$E_1(\underline{A_{E11}}, A_{E12})$
 $R(\underline{A_{E11}}, \underline{A_{E21}}, A_R)$
 $E_2(\underline{A_{E21}}, A_{E22})$

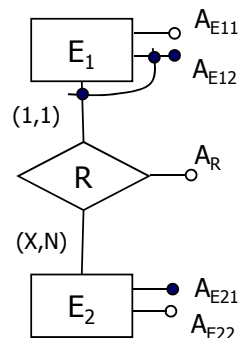
o $E_1(\underline{A_{E11}}, A_{E12}, \underline{A_{E21}}, A_R)$
 $E_2(\underline{A_{E21}}, A_{E22})$

possono essere nulli

57

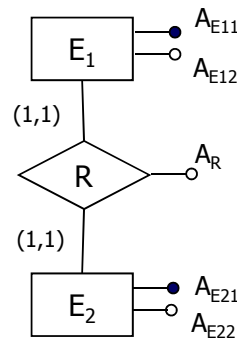
Traduzione delle relazioni 3

Relazione con
identificatore esterno



$E_1(\underline{A_{E12}}, \underline{A_{E21}}, A_{E11}, A_R)$
 $E_2(\underline{A_{E21}}, A_{E22})$

Relazione uno-a-uno
con partecipazione obbligatoria
di entrambe le entità



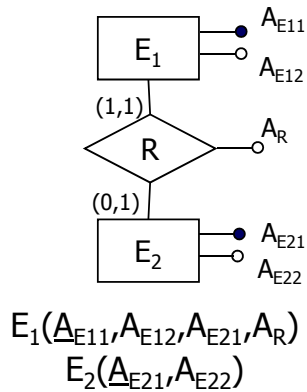
$E_1(\underline{A_{E11}}, A_{E12}, A_{E21}, A_R)$
 $E_2(\underline{A_{E21}}, A_{E22})$

o $E_2(\underline{A_{E21}}, A_{E22}, A_{E11}, A_R)$
 $E_1(\underline{A_{E11}}, A_{E12})$

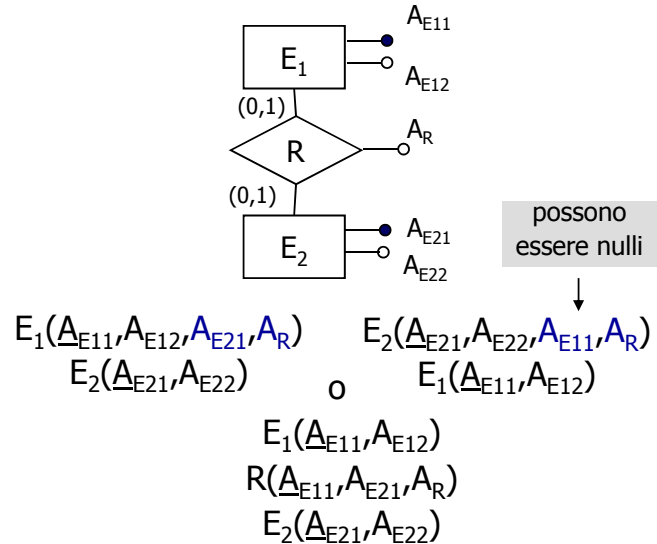
58

Traduzione delle relazioni 4

Relazione uno-a-uno
con partecipazione opzionale
di una entità



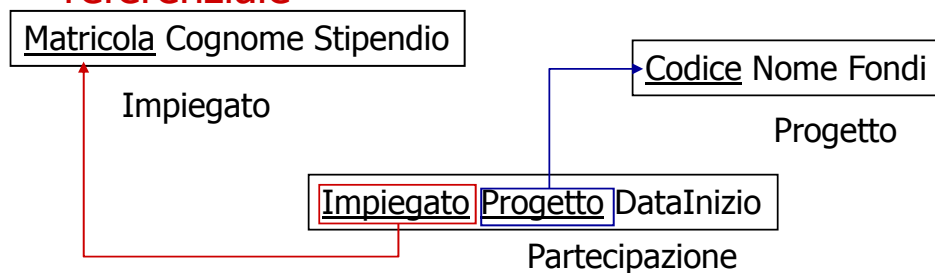
Relazione uno-a-uno
con partecipazione opzionale
di entrambe le entità



59

Documentazione degli schemi logici

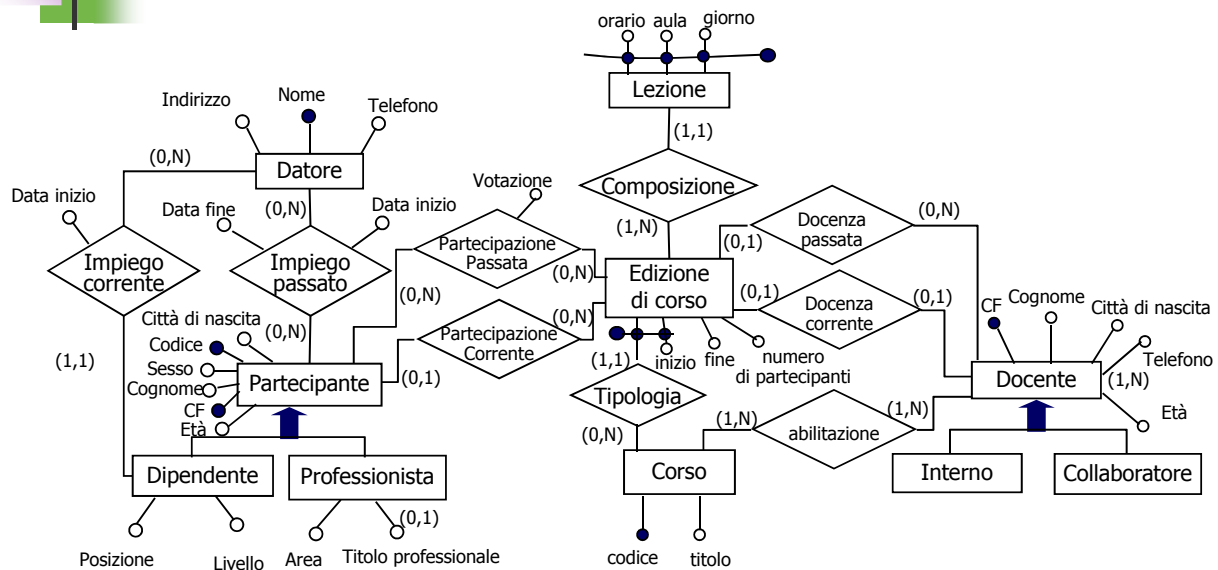
- Si devono documentare i **vincoli di integrità referenziale**



- Il diagramma è utile anche per visualizzare i **cammini di join**

60

Esempio: società di formazione



61

Esempio: operazioni

- **Operazione 1**: inserisci un nuovo partecipante indicando tutti i suoi dati
- **Operazione 2**: assegna un partecipante ad una edizione di un corso
- **Operazione 3**: inserisci un nuovo docente indicando tutti o suoi dati e i corsi che può insegnare
- **Operazione 4**: assegna un docente abilitato a una edizione di corso
- **Operazione 5**: stampa tutte le informazioni sulle edizioni passate di un corso con titolo, orari delle lezioni e numero dei partecipanti
- **Operazione 6**: stampa tutti i corsi offerti, con informazioni sui docenti che possono insegnarli
- **Operazione 7**: per ogni docente, trova tutti i partecipanti a tutti i corsi da lui insegnati
- **Operazione 8**: effettua una statistica su tutti i partecipanti a un corso, con tutte le informazioni su di essi, sulla edizione alla quale hanno partecipato e la rispettiva votazione

62

Esempio: il carico

Concetto	Tipo	Volume
Lezione	E	8000
Edizione corso	E	1000
Corso	E	200
Docente	E	300
Collaboratore	E	250
Interno	E	50
Partecipante	E	5000
Dipendente	E	4000
Professionista	E	1000
Datore	E	8000
Part. Passata	R	10000
Part. Corrente	R	500
Composizione	R	8000
Tipologia	R	1000
Doc. passata	R	900
Doc. corrente	R	100
Abilitazione	R	500
Impiego corrente	R	4000
Impiego passato	R	1000

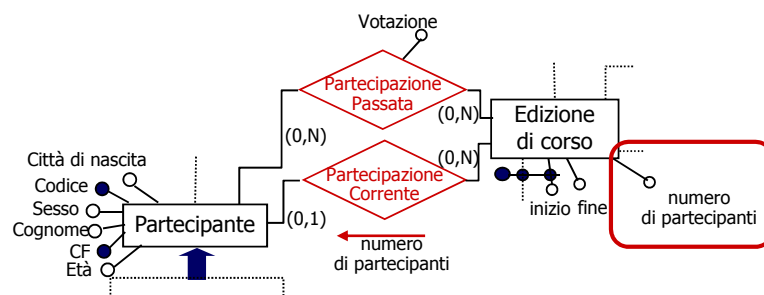
Tavola dei volumi

Operazione	Tipo	Frequenza
Op. 1	I	40/giorno
Op. 2	I	50/giorno
Op. 3	I	2/giorno
Op. 4	I	15/giorno
Op. 5	I	10/giorno
Op. 6	I	20/giorno
Op. 7	I	5/sett.
Op. 8	B	10/mese

Tavola delle operazioni

63

Esempio: ridondanze

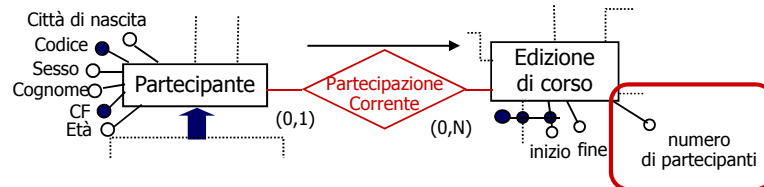


- **Occupazione di memoria:** Volume Edizione Corso x 4 byte = 4000 byte
- **Operazioni coinvolte:**
 - Op. 2: assegna un partecipante ad una edizione di un corso
 - Op. 5: stampa tutte le informazioni sulle edizioni passate di un corso
 - Op. 8: può essere trascurata perché poco frequente e batch

64

Esempio: operazione 2

Op. 2: assegna un partecipante ad una edizione di un corso



con ridondanza

Concetto	Costrutto	Accessi	Tipo
Partecipante	E	1	L
Par. corrente	R	1	S
Edizione corso	E	1	L
Edizione corso	E	1	S

$2 \times 50 \text{ S} + 2 \times 50 \text{ L} / \text{giorno}$

senza ridondanza

Concetto	Costrutto	Accessi	Tipo
Partecipante	E	1	L
Par. corrente	R	1	S

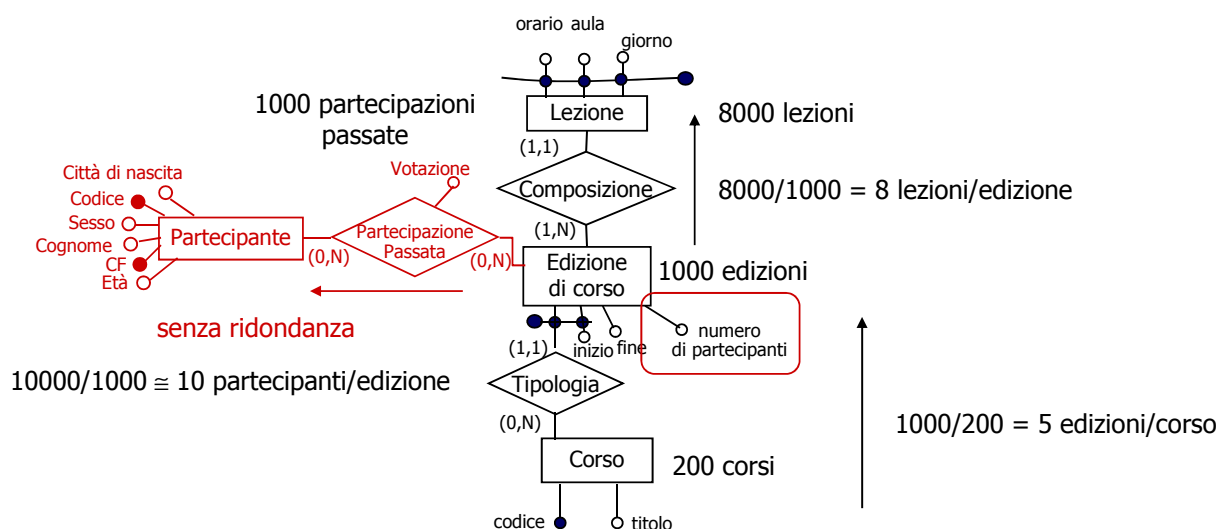
aggiornamento di "numero di partecipanti"

$1 \times 50 \text{ S} + 1 \times 50 \text{ L} / \text{giorno}$

65

Esempio: operazione 5/a

Op. 5: stampa tutte le informazioni sulle edizioni passate di un corso con titolo, orari delle lezioni e numero dei partecipanti



66

Esempio: operazione 5/b

con ridondanza

Concetto	Costrutto	Accessi	Tipo
Corso	E	1	L
Tipologia	R	1	L
Edizione corso	E	5	L
Composizione	R	5x8=40	L
Lezione	E	40	L

87 x 10 L /giorno

senza ridondanza

Concetto	Costrutto	Accessi	Tipo
Corso	E	1	L
Tipologia	R	1	L
Edizione corso	E	5	L
Part. passata	R	5x10=50	L
Composizione	R	5x8=40	L
Lezione	E	40	L

137 x 10 L /giorno

conteggio dei partecipanti alle edizioni passate

67

Esempio: scelta ridondanza

Totali

Operazione	L	S
Operazione 2	100/giorno	100/giorno
Operazione 5	870/giorno	0

con ridondanza

Operazione	L	S
Operazione 2	50/giorno	50/giorno
Operazione 5	1370/giorno	0

senza ridondanza

- Considerando doppio il costo per le operazioni di scrittura
 - totale con ridondanza: 1170
 - totale senza ridondanza: 1520
- Si decide quindi di mantenere la ridondanza

68

Esempio: gerarchie /a

Gerarchia dei docenti

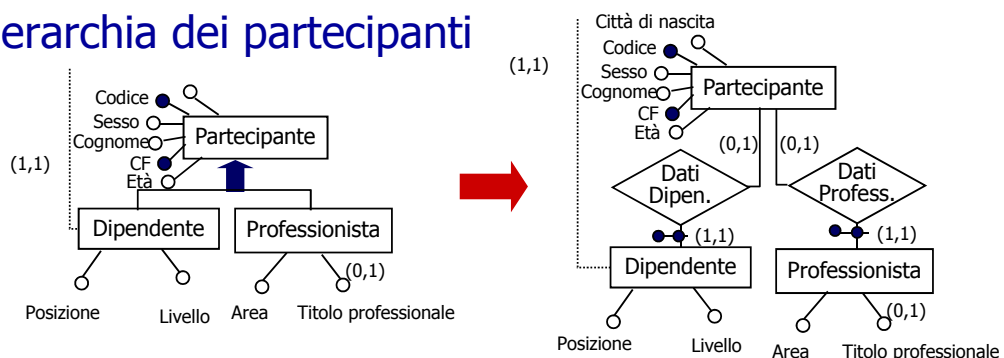


- Le operazioni che fanno riferimento ai docenti (3,4,6,7) non distinguono i sottotipi
- I due sottotipi non hanno attributi specifici
- Si accorpano le figlie nel padre aggiungendo l'attributo **Tipo** che ha per dominio l'insieme dei valori **C** (collaboratore) e **I** (interno).

69

Esempio: gerarchie /b

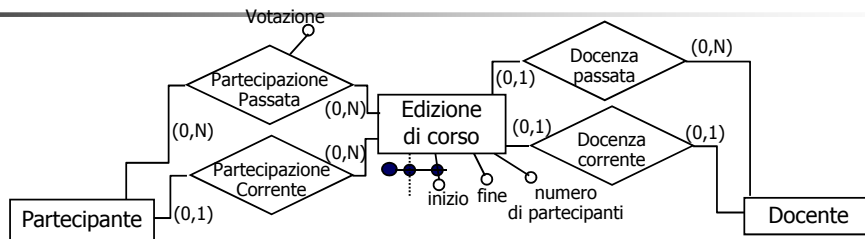
Gerarchia dei partecipanti



- Le operazioni che fanno riferimento ai partecipanti (1,2,8) non distinguono i sottotipi
- I due sottotipi hanno attributi specifici
- E' preferibile lasciare i due sottotipi come entità e usare due relazioni per evitare di avere entità con troppi attributi e/o con attributi con valori nulli

70

Esempio: partizionamento /a

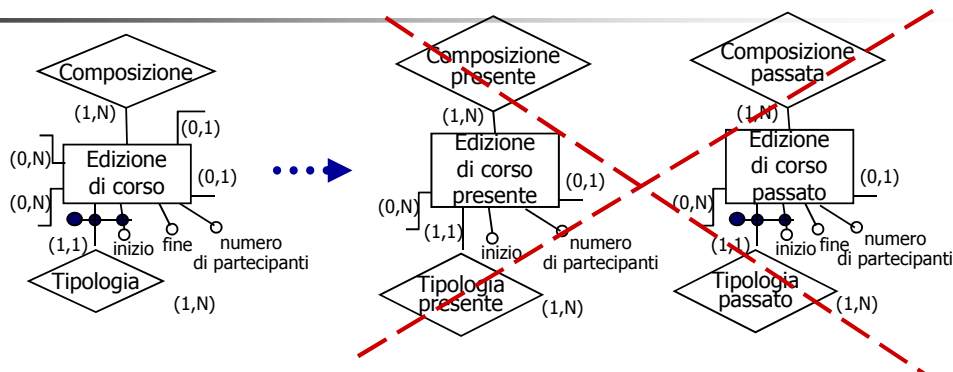


- L'operazione 5 distingue fra edizioni correnti e passate
- Le relazioni *Partecipazione corrente* e *Partecipazione passata* fanno riferimento a edizioni correnti e passate
- Le relazioni *Docenza corrente* e *Docenza passata* fanno riferimento a edizioni correnti e passate

conviene partizionare l'entità *Edizione di corso* ?

71

Esempio: partizionamento /b

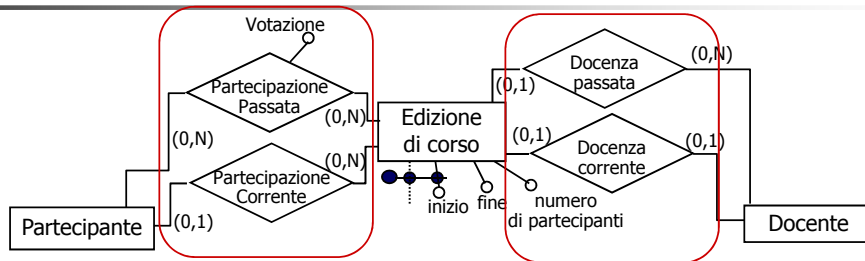


- Andrebbero duplicate le relazioni *Composizione* e *Tipologia*
- Le operazioni 7 e 8 che non fanno differenza tra edizioni correnti e passate richiederebbero al visita di due entità distinte

Non partizioniamo tale entità

72

Esempio: accorpamento /a

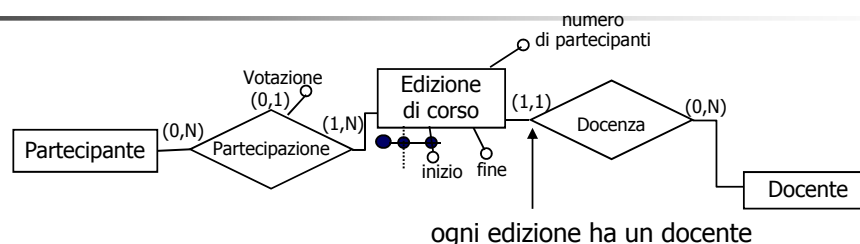


- Le operazioni in pratica non richiedono la differenza fra i concetti relativi alle relazioni *Docenza* e *Partecipazione* corrente e passata
- Occorre trasferire le occorrenze da una relazione all'altra quando una edizione di corso termina
- Il numero medio di istanze di *Partecipazione corrente* è basso (500)

si accorpano i due concetti

73

Esempio: accorpamento /b

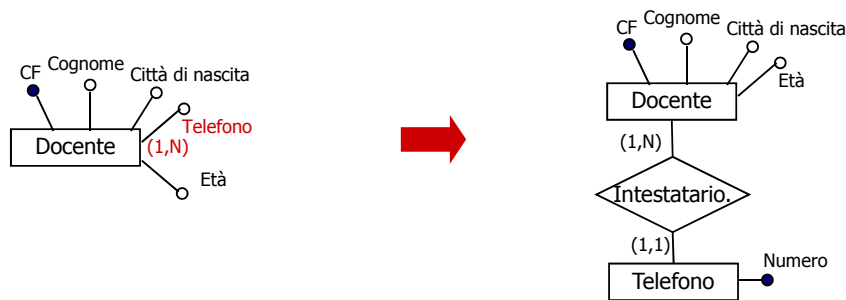


- L'attributo *Votazione* della relazione *Partecipazione* non si applica alle partecipazioni presenti ed è quindi nullo
 - Lo speco di memoria è solo di $500 \times 4 \text{ byte} = 2000 \text{ byte}$
- Le cardinalità minime dalla parte di *Edizione di corso* sono 1 perché ogni edizione ha un *Docente* e almeno un *Partecipante*

74

Esempio: attributo multivalore

- Si elimina l'attributo multivalore introducendo l'entità Telefono legata con una relazione uno-a-molti con l'entità Docente



75

Esempio: identificatori

Partecipante

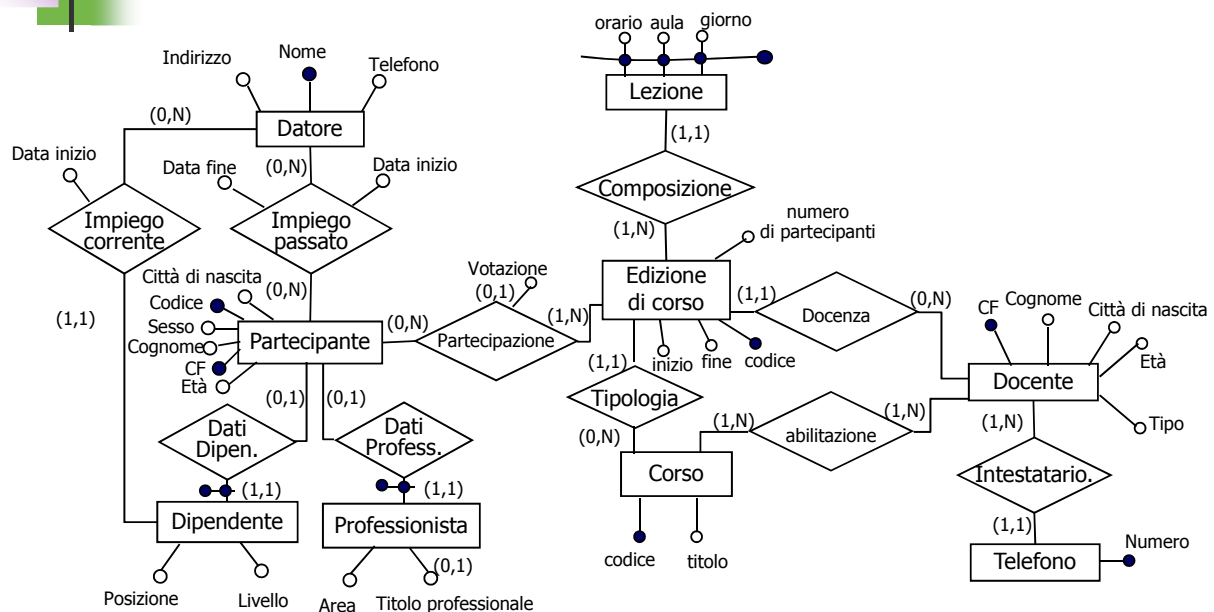
- Codice: 2 byte ←
- CF: 16 byte

Edizione Corso

- Data Inizio e Tipologia Corso (esterno)
- Deve essere usato in Partecipazione e Docenza con molte occorrenze
- E' preferibile inserire un codice ad hoc

76

Esempio: schema ristrutturato



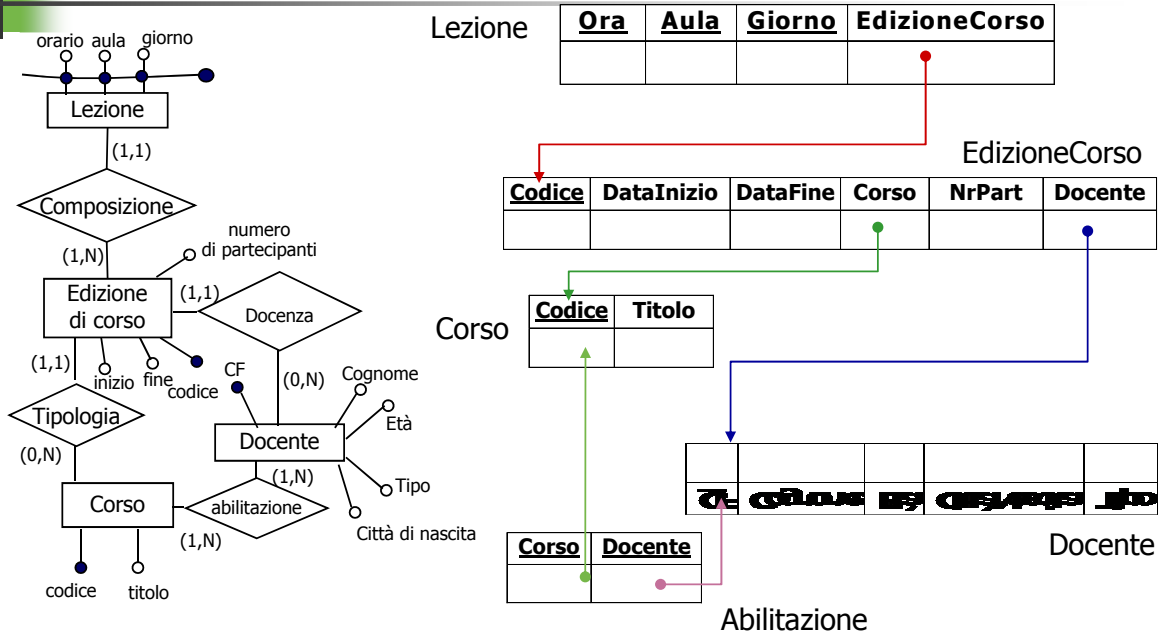
77

Esempio: schema relazionale

EdizioneCorso(Codice, DataInizio, DataFine, Corso, Docente, NrPart)
 Lezione(Ora, Aula, Giorno, EdizioneCorso)
 Docente(CF, Cognome, Età, CittàNascita, Tipo)
 Telefono(Numero, Docente), Corso(Codice, Titolo)
 Abilitazione(Corso, Docente)
 Partecipante(Codice, CF, Cognome, Età, CittàNascita, Sesso)
 Partecipazione(Partecipante, EdizioneCorso, Votazione*)
 Datore(Nome, Telefono, Indirizzo)
 ImpiegoPassato(Partecipante, Datore, DataInizio, DataFine)
 Professionista(Partecipante, Area, Titolo*)
 Dipendente(Partecipante, Livello, Posizione, Datore, DataInizio)

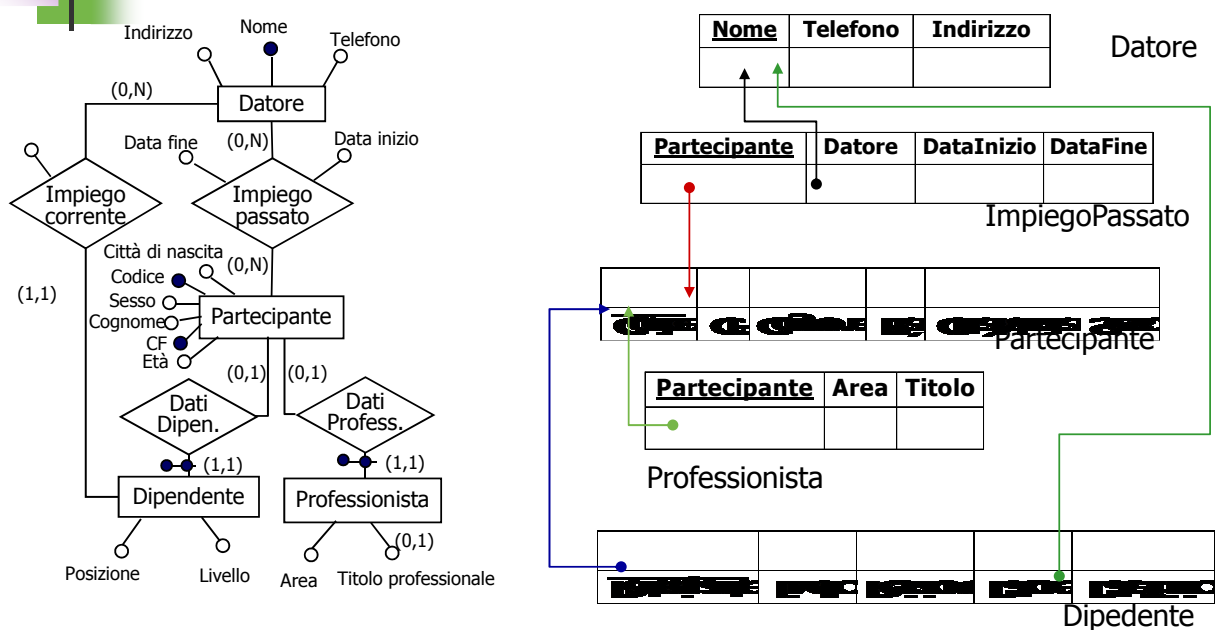
78

Esempio: schema relazionale /a



79

Esempio: schema relazionale /b



80