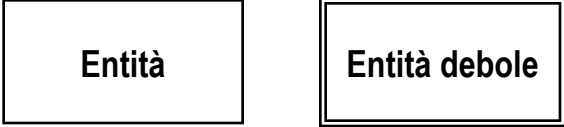
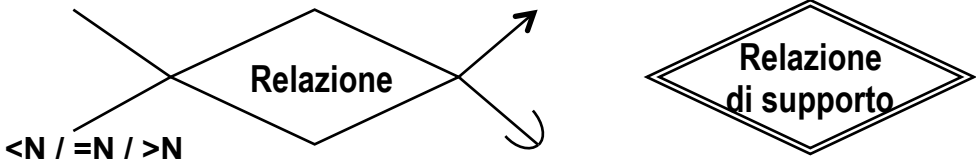
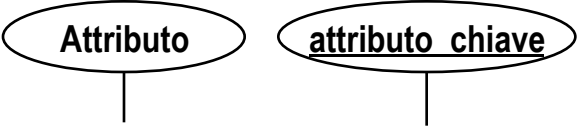



Elementi del modello E/R

Entità	
Relazioni	
Attributi	
Generalizzazione ISA	

Materiale alternativo (non sostitutivo)

Esercizio 1 – Campionato di calcio

Si vuole realizzare un database per l'ultima stagione del campionato.

Il campionato si articola in più 'giornate', ciascuna identificata da un numero progressivo. Per ogni giornata, si vuole memorizzare la classifica, ovvero quanti punti ha ciascuna squadra.

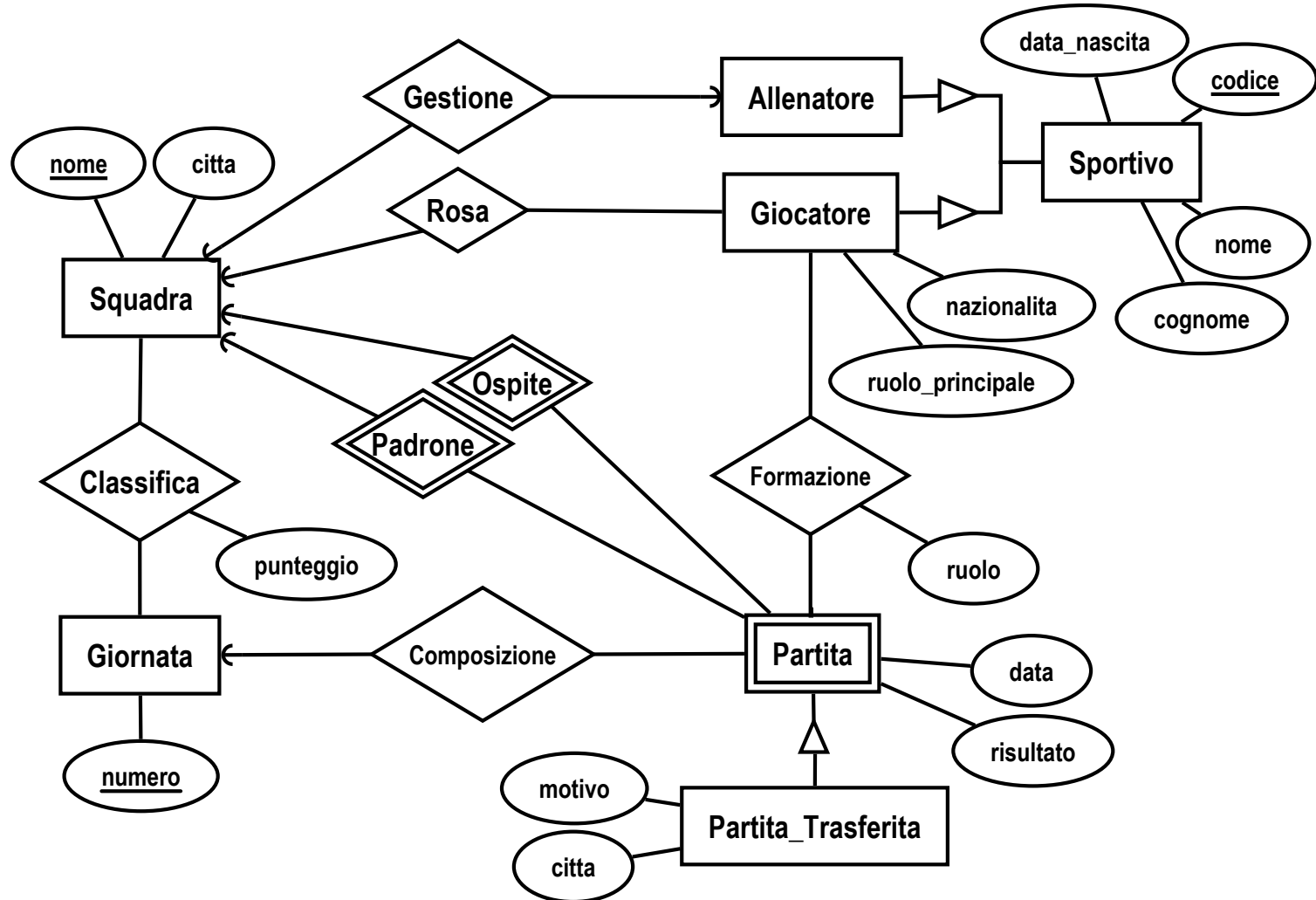
Per ogni giornata si disputano più partite, ciascuna delle quali si svolge in una certa città in una certa data, coinvolge una squadra ospite ed una squadra padrona di casa con le rispettive formazioni e termina con un risultato. Le partite sono normalmente giocate nella città della squadra ospite, ma alcune partite possono essere trasferite, nel qual caso si vuole memorizzare motivo del trasferimento e città di destinazione.

Ogni squadra è associata ad una città ed ha una rosa di giocatori ed un allenatore, che per semplicità si assume non cambiano squadra durante la stagione di campionato. Per giocatori e allenatore si conoscono nome, cognome e data di nascita. Per i giocatori sono noti inoltre nazionalità e ruolo principale, ma si vuole anche memorizzare i ruoli (possibilmente diversi) con cui sono stati schierati in ciascuna partita.

Quesiti:

1. Si produca il modello E/R del database
2. Si converta il modello realizzato in uno schema relazionale

Esercizio 1 – Soluzione (1/5)



Esercizio 1 – Soluzione (2/5)

Osservazioni sul modello E/R:

La documentazione delle assunzioni è fondamentale

Di seguito solo le più peculiari:

- Si ipotizza che le squadre siano identificate dal nome, questo lascia la libertà di avere squadre diverse nella stessa città
- Poiché il DB memorizza una sola stagione del campionato, ciascuna partita può essere identificata mediante i nomi delle due squadre ospite e padrona di casa che la disputano
- Alternativamente si può ipotizzare che anche la giornata sia utilizzata nella chiave, permettendo così partite delle stesse squadre in giornate diverse
- Poiché il testo tratta in modo analogo giocatori e allenatori, specificando attributi simili, si è aggiunta una generalizzazione Sportivo (poteva essere chiamata anche Persona) per mettere in rilievo il legame tra le due entità
- Nel definire le cardinalità delle relazioni, assumiamo che non possano esserci nel DB giocatori e allenatori che non lavorino per qualche squadra,
- Così come ogni squadra deve avere necessariamente un allenatore

Esercizio 1 – Soluzione (3/5)

Elementi E/R di partenza Relazioni derivate

Squadra ⇒ Squadra (nome, città)

Classifica ⇒ Classifica (nome_squadra, numero_giornata, punteggio)

~~Giornata ⇒ Giornata (numero)~~

Partita, Composizione ⇒ Partita (nome_ospite, nome_padrone, data, risultato, numero_giornata,
Partita_Trasferita, motivo_trasferimento, città_trasferimento)

Allenatore, Gestione ⇒ Allenatore (codice, nome, cognome, data_nascita, nome_squadra)

Giocatore, Rosa ⇒ Giocatore (codice, nome, cognome, data_nascita, ruolo_principale,
nazionalità, nome_squadra)

Formazione ⇒ Formazione (codice_giocatore, nome_ospite, nome_padrone, ruolo)

Esercizio 1 – Soluzione (4/5)

Osservazioni sulla traduzione:

- la relazione Giornata può essere rimossa in quanto contenuta in Classifica o in Partita, sotto l'ipotesi che il DB contenga una giornata soltanto se c'è almeno una squadra con un punteggio per quella giornata o una partita per quella giornata
- In questa soluzione si è preferito ristrutturare la generalizzazione per Partita accorpendo l'entità figlia nell'entità padre
 - separare le due entità replicando gli attributi di Partita in Partita_Trasferita comporterebbe il dover duplicare le relazioni molti a molti Formazione e Composizione, creando così due tabelle aggiuntive
 - non occorre un attributo aggiuntivo 'tipo_partita' in quanto è sufficiente controllare se gli attributi 'motivo_trasferimento' e/o 'citta_trasferimento' sono o non sono NULL (risp. Partita oppure Partita_Trasferita)
 - Questa soluzione è riportata per completezza e compattezza, seppur corretta deve essere chiaro per lo studente che non si tratta di una soluzione molto elegante.

Esercizio 1 – Soluzione (5/5)

Osservazioni sulla traduzione (continua):

- si è preferito ristrutturare la generalizzazione per Sportivo accorpando l'entità padre nelle entità figlie

- a supporto di questa soluzione vi è il fatto che Sportivo non partecipa a relazioni che andrebbero altrimenti replicate; così facendo, inoltre, non si hanno attributi potenzialmente NULL
- non occorre un attributo aggiuntivo 'tipo_partita' in quanto è sufficiente controllare se gli attributi 'motivo_trasferimento' e/o 'citta_trasferimento' sono o non sono NULL (risp. Partita oppure Partita_Trasferita)

[confronta con la soluzione adottata per le partite trasferite!]

- la relazione gestione può essere accorpata tanto in Squadra quanto in Allenatore
 - a meno di conoscere in anticipo le query che insisteranno sul DB, la scelta tra le due è indifferente
 - Tutto dipende da quale entità vogliamo privilegiare
 - qualora, invece, la cardinalità in uno dei due rami della relazione fosse stata < 1 invece che $= 1$, allora sarebbe convenuto accorpare la relazione sul ramo con cardinalità $= 1$ in modo da evitare valori NULL sull'attributo aggiunto

Esercizio 2 – Noleggio film

Un'azienda di noleggio film vuole realizzare un database per gestire l'archivio dei film in catalogo e l'attività di noleggio.

Ogni film è caratterizzato da titolo, anno di produzione, uno o più attori ciascuno dei quali interpreta un personaggio, e da un unico regista, che può essere anche attore. Per ogni artista, sia esso attore o regista, si conoscono nome e cognome, nome d'arte (a volte), anno di nascita e eventuale lista di premi vinti durante la carriera.

Per ogni film l'azienda possiede una o più copie noleggiabili, ciascuna delle quali identificata da un codice di magazzino e caratterizzata da anno di acquisto e tipo di supporto (videocassetta, DVD, Blu-ray).

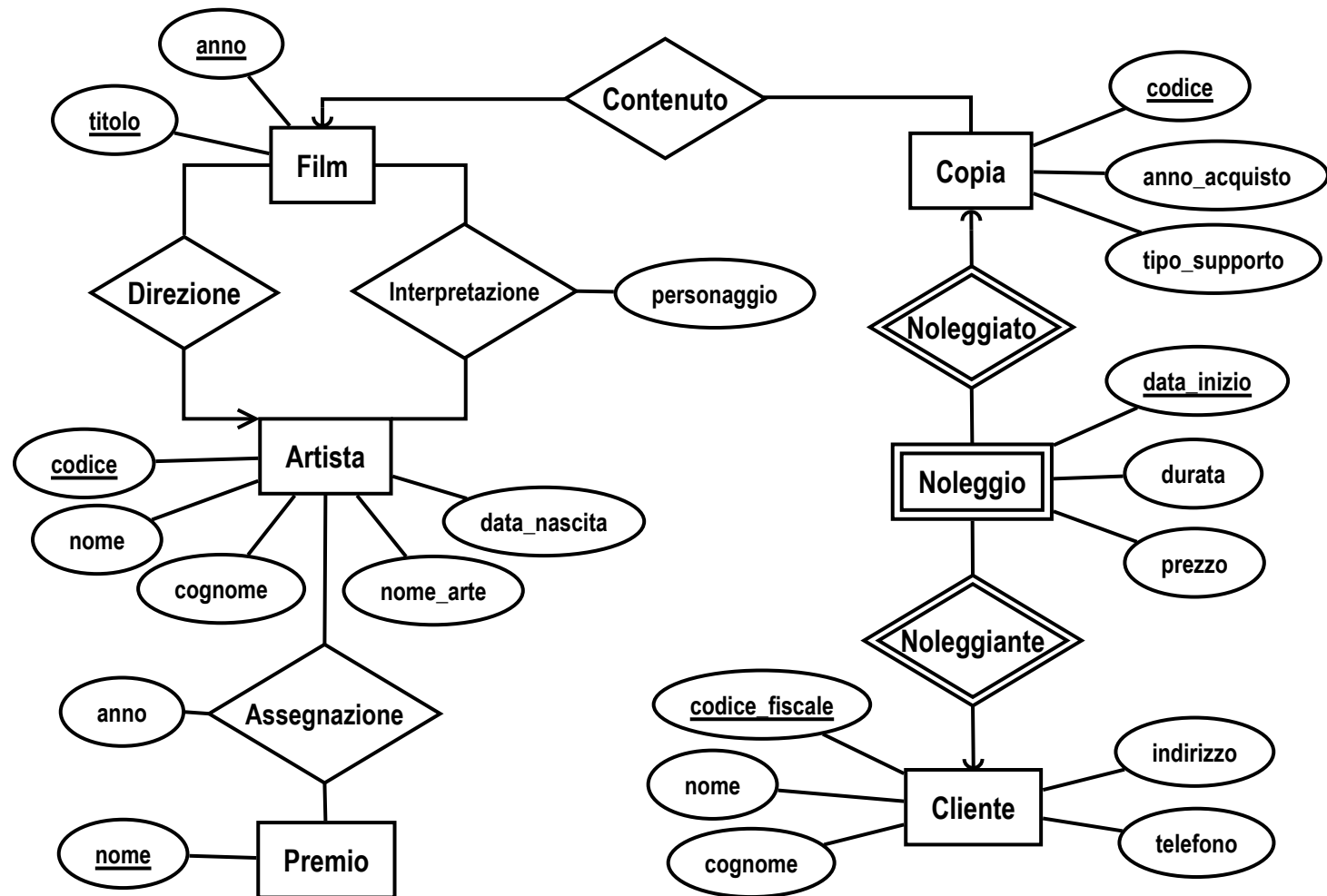
Il database deve memorizzare i dati relativi ai noleggi in corso e ai noleggi effettuati in passato. Per ogni noleggio si conosce il film e il codice della copia noleggiata (al più una), il cliente, il prezzo pagato, la data e la durata prevista del noleggio.

Per ogni cliente si vuole mantenere codice fiscale, nome, cognome, indirizzo e recapito telefonico.

Quesiti:

1. Si produca il modello E/R del database
2. Si converta il modello realizzato in uno schema relazionale

Esercizio 2 – Soluzione (1/4)



Esercizio 2 – Soluzione (2/4)

Osservazioni sul modello E/R:

- si è preferito non aggiungere una generalizzazione tra Artista, Regista e Attore in quanto hanno gli stessi attributi e la distinzione tra Regista e Attore è già veicolata dalle relazioni Interpretazione e Direzione
 - non sarebbe stato comunque un errore aggiungere la generalizzazione
 - in tal caso, fare attenzione che non si può modellare Regista ISA Attore, in quanto un Regista può essere un attore, ma non lo deve essere necessariamente come richiesto dalla relazione ISA
- si ipotizza che un noleggio sia effettuato esattamente da un Cliente e riguardi esattamente una Copia
- si noti che i noleggi non possono essere modellati mediante relazione binaria tra Copia e Cliente, in quanto così facendo si escluderebbe la possibilità che un cliente noleggi la stessa copia più volte nel tempo (con data, durata e possibilmente prezzo diversi)

Esercizio 2 – Soluzione (3/4)

Elementi E/R di partenza Relazioni derivate

Assegnazione, Premio \Rightarrow Assegnazione (codice_artista, nome_premio, anno)

Artista \Rightarrow Artista (codice, nome, cognome, nome_arte, data_nascita)

Interpretazione \Rightarrow Interpretazione (codice_artista, titolo_film, anno_film, personaggio)

Direzione \Rightarrow Direzione (codice_artista, titolo_film, anno_film)

Film \Rightarrow Film (titolo, anno)

Copia
Contenuto \Rightarrow Copia (codice, anno_acquisto, tipo_supporto, titolo_film, anno_film)

Noleggio, Noleggiante, \Rightarrow Noleggio (codice_fiscale_cliente, codice_copia, data_inizio, prezzo)
Noleggiato

Cliente \Rightarrow Cliente (codice_fiscale, nome, cognome, indirizzo, telefono)

Esercizio 2 – Soluzione (4/4)

Osservazioni sulla traduzione:

- nell'ipotesi che un Premio sia aggiunto al database soltanto se c'è qualche Artista che lo ha vinto, allora la relazione Premio con cui si traduce l'entità omonima può essere rimossa, in quanto contenuta nella relazione per Assegnazione
- Direzione può essere sia accorpata in Film (in quanto relazione molti a uno) che tradotta in una relazione ad hoc, in quanto la non tutti i film hanno un regista e la prima soluzione porterebbe ad avere alcuni attributi 'codice_regista' NULL

Esercizio 3 – Compagnia aerea

Una compagnia aerea vuole realizzare un database per gestire i suoi voli.

Ogni volo di linea è caratterizzato da un codice identificativo, ora e aeroporto di partenza, ora e aeroporto di e arrivo. Gli aeroporti sono caratterizzati anch'essi da un codice identificativo, nonché da città e stato dove sono situati.

Per ogni volo si vogliono memorizzare i viaggi effettuati, soppressi o in programma. Un viaggio è identificato dalla data ed è caratterizzato dal velivolo impiegato e dalla lista del personale viaggiante, che consiste in un comandante, in un vicepilota e negli assistenti di viaggio (almeno uno).

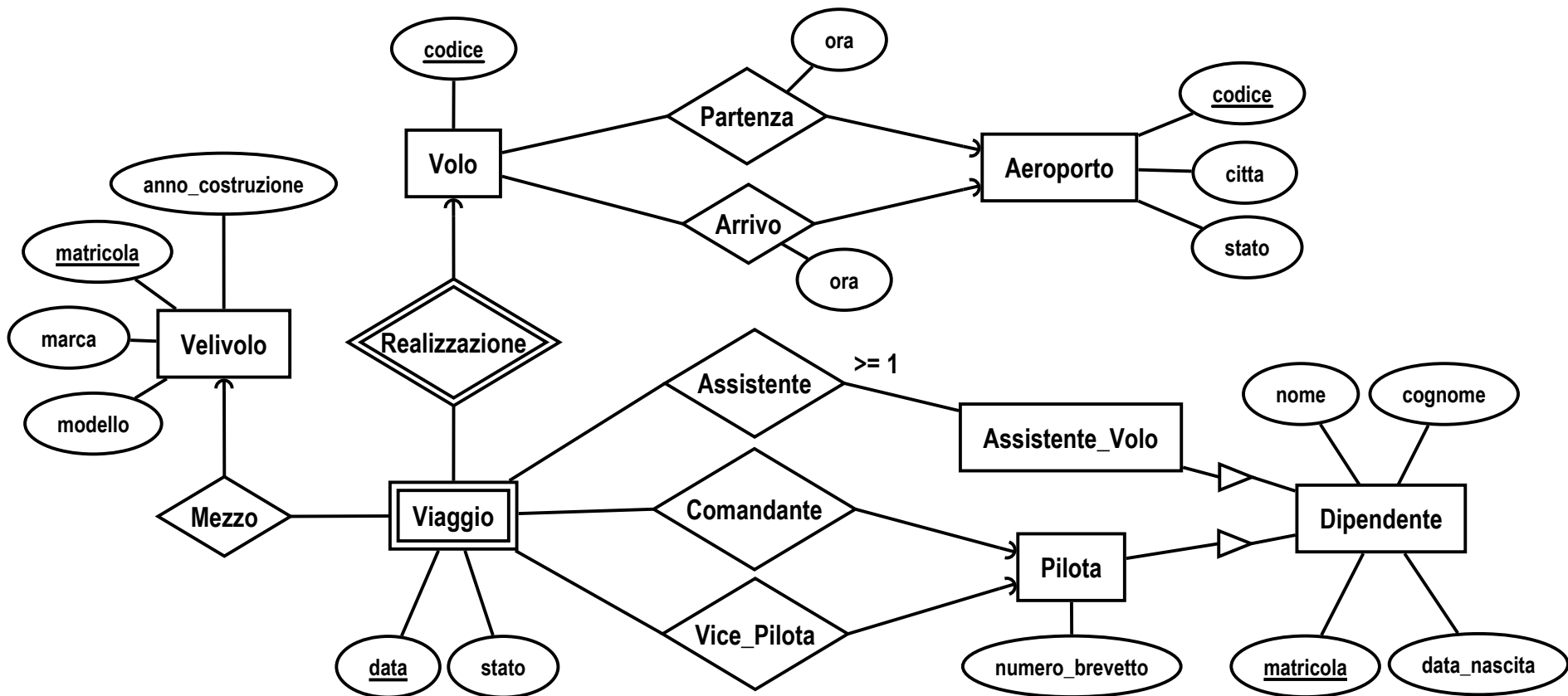
I velivoli sono identificati da un numero di matricola, e per essi sono noti marca, modello e anno di costruzione.

Ciascun membro del personale viaggiante è identificato da una matricola e possiede nome, cognome e data di nascita. Il personale viaggiante è costituito da assistenti di volo e piloti; per i secondi si vuole memorizzare anche il numero di brevetto.

Quesiti:

1. Si produca il modello E/R del database
2. Si converta il modello realizzato in uno schema relazionale

Esercizio 3 – Soluzione (1/4)



Esercizio 3 – Soluzione (2/4)

Osservazioni sul modello E/R

- i due attributi 'ora' posti sulle relazioni partenza e arrivo si possono anche collocare sull'entità Volo (usando nomi diversi, e.g. ora_partenza, ora_arrivo)
 - la scelta di metterli sulle relazioni è concettualmente più pertinente, in quanto si tratta di attributi che caratterizzano partenza e arrivo di un volo
 - ciò non incide comunque sullo schema relazionale risultante
- l'entità Dipendente modella un generico membro del personale viaggiante e si specializza in Pilota e Assistente_Volo

Esercizio 3 – Soluzione (3/4)

Elementi E/R di partenza Relazioni derivate

Aeroporto ⇒ Aeroporto (codice, città, stato)

Volo, Partenza, Arrivo ⇒ Volo (codice, ora_partenza, codice_aeroporto_partenza, ora_arrivo, codice_aeroporto_arrivo)

Velivolo ⇒ Velivolo (matricola, anno_costruzione, marca, modello)

Viaggio, Mezzo,
Realizzazione ⇒ Viaggio (data, codice_volo, stato, matricola_velivolo, matricola_comandante, matricola_vice_pilota)

Pilota ⇒ Pilota (matricola, nome, cognome, data_nascita, numero_brevetto)

Assistente_Volo ⇒ Assistente_Volo (matricola, nome, cognome, data_nascita)

Assistente ⇒ Assistente (data_viaggio, codice_volo, matricola assistente volo)

Esercizio 3 – Soluzione (4/4)

Osservazioni sulla traduzione

- la generalizzazione con entità padre in Dipendente può essere ragionevolmente tradotta sia accorpendo l'entità padre nelle figlie, come proposto, sia accorpendo le entità figlie nel padre
 - il primo caso ha come vantaggio la mancata introduzione di attributi possibilmente null
 - il secondo caso facilita invece query su Dipendente che devono estrarre indifferentemente piloti o assistenti di volo, in quanto l'informazione è tutta in una relazione

Esercizio 4 – Archivio fotografico

Un gruppo editoriale vuole realizzare un database per il suo archivio fotografico digitale.

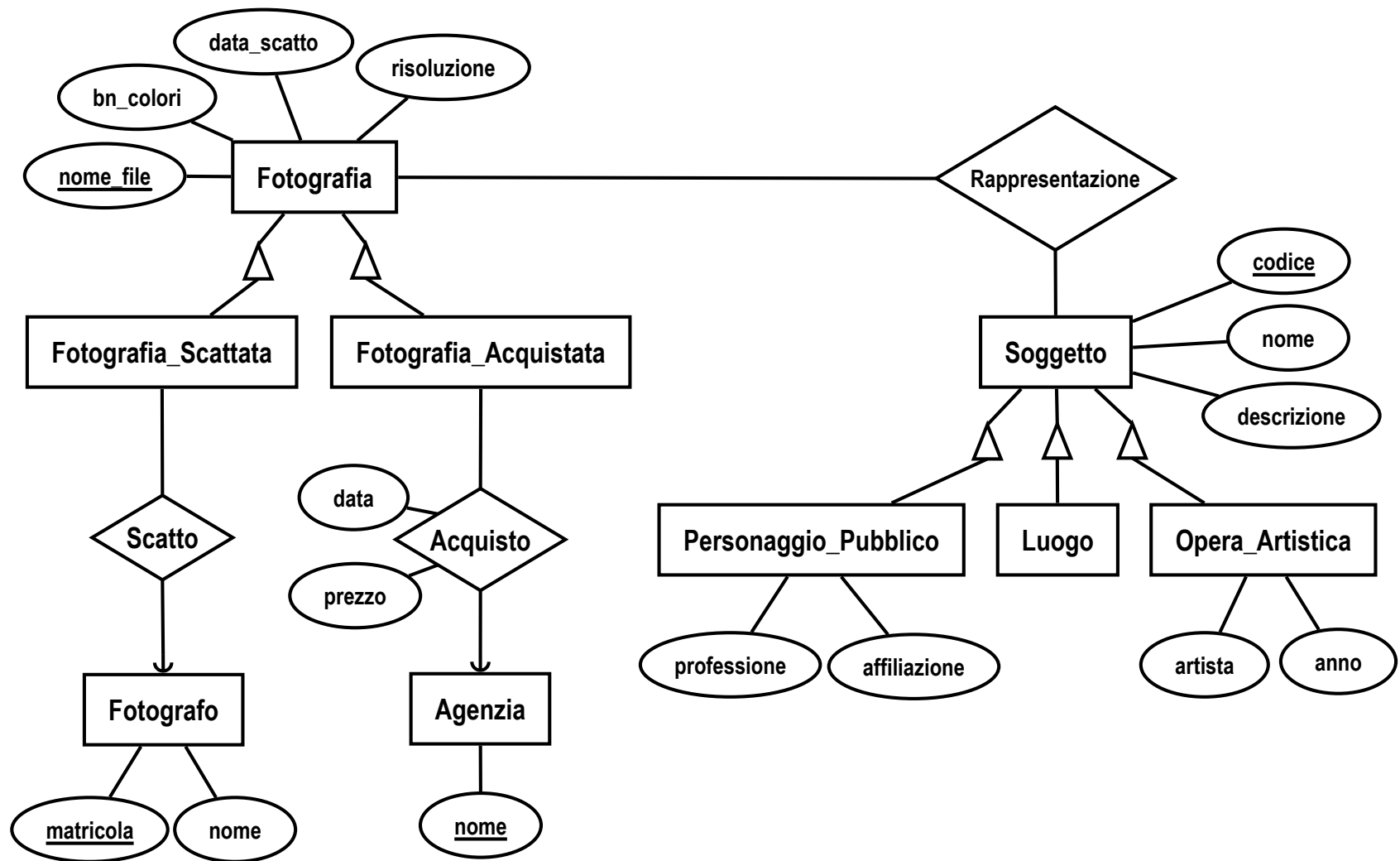
Le fotografie sono identificate dal nome del file con cui sono archiviate nel sistema, possono essere in bianco e nero o a colori e sono caratterizzate da data di scatto e risoluzione digitale. Le fotografie possono essere state scattate da un fotografo del giornale, nel qual caso è noto il nome del fotografo e la sua matricola aziendale, oppure acquistate da un'agenzia esterna, nel qual caso si conosce il nome dell'agenzia, la data di acquisto ed il prezzo pagato.

Le fotografie sono catalogate in base ad un catalogo di soggetti possibili. Questi possono essere personaggi pubblici, luoghi o opere artistiche e sono caratterizzati da nome, descrizione e codice identificativo. Per i personaggi pubblici sono spesso disponibili anche la professione e l'affiliazione (e.g. il partito politico di appartenenza). Per le opere artistiche può essere noto il nome dell'artista e l'anno di realizzazione.

Quesiti:

1. Si produca il modello E/R del database
2. Si converta il modello realizzato in uno schema relazionale

Esercizio 4 – Soluzioni (1/5)



Esercizio 4 – Soluzione (2/5)

Osservazioni sul modello E/R

- si assume che una fotografia possa avere più soggetti, così come lo stesso soggetto possa essere rappresentato da più fotografie
- per semplicità, si modella la distinzione tra fotografie in B/N e a colori tramite un attributo booleano `bn_colore`, e non mediante generalizzazione (non necessaria qui)
- si è deciso di modellare esplicitamente le due tipologie di fotografie scattate da fotografi del giornale ed acquistate da agenzie, ma non era indispensabile

Esercizio 4 – Soluzione (3/5)

Elementi E/R di partenza Relazioni derivate

Fotografia ⇒ Fotografia (nome_file, bn_colori, data_scatto, risoluzione)

Fotografia_Scattata, ⇒ Fotografia_Scattata (nome_file, matricola_fotografo)
Scatto

Fotografia_Acquistata, ⇒ Fotografia_Acquistata (nome_file, nome_agenzia, data, prezzo)
Acquisto

Fotografo ⇒ Fotografo (matricola, nome)

Rappresentazione ⇒ Rappresentazione (nome_file, codice_soggetto)

Soggetto ⇒ Soggetto (codice, nome, descrizione)

Personaggio_Pubblico ⇒ Personaggio_Pubblico (codice, professione, affiliazione)

Luogo ⇒ Luogo (codice)

Opera_Artistica ⇒ Opera_Artistica (codice, artista, anno)

Esercizio 4 – Soluzione (4/5)

Osservazioni sulla traduzione

- la gerarchia con radice in Fotografia viene ristrutturata sostituendo ISA con relazioni
 - si preferisce non accorpare l'entità padre nelle entità figlie per non duplicare la relazione molti a molti Rappresentazione
 - si preferisce non accorpare le entità figlie nell'entità padre a causa delle relazioni molti a uno Scatto e Acquisto, le quali comporterebbero l'aggiunta di 4 attributi (possibilmente NULL) alla relazione per Fotografia, in aggiunta ai 4 già presenti
- la gerarchia con radice in Soggetto viene ristrutturata anch'essa sostituendo ISA con relazioni
 - ancora, si preferisce non accorpare il padre nelle entità figlie per non triplicare Rappresentazione
 - analogamente a prima, si preferisce non accorpare le entità figlie nel padre per evitare il proliferare di attributi possibilmente NULL in Soggetto

Esercizio 4 – Soluzione (5/5)

Osservazioni sulla traduzione

- la relazione Agenzia viene mantenuta anche se il suo schema è contenuto in quello di Fotografia_Acquistata:
 - assumiamo infatti che il DB possa contenere anche agenzie da cui non si sono ancora acquistate delle foto, per cui può quindi esistere una tupla in Agenzia ma non in Fotografia_Acquistata
- a causa dell'assenza di attributi aggiuntivi in Fotografia_Scattate e Fotografia_Acquistate, le loro traduzioni in pratica corrisponde alla traduzione delle relazioni Scatto e Acquisto
 - se si preferisce, si possono ridenominare le relazioni ottenute da Fotografia_Scattata a Scatto e da Fotografia_Acquistata ad Acquisto

Esercizi dal libro

- Modellazione E/R:
 - 4.1.1 e 4.1.2 (con specifica chiavi e entità deboli)
 - 4.2.1
- Traduzione da E/R a relazionale:
 - 4.5.1
 - 4.6.1 e 4.6.2

Esercizi dal libro – 4.1.1 e 4.1.2

Exercise 4.1.1: Design a database for a bank, including information about customers and their accounts. Information about a customer includes their name, address, phone, and customer ID. Accounts have numbers, types (e.g., savings, checking) and balances. Also record the customer(s) who own an account. Draw the E/R diagram for this database. Be sure to include arrows where appropriate, to indicate the multiplicity of a relationship.

Exercise 4.1.2: Modify your solution to Exercise 4.1.1 as follows:

- a) Change your diagram so a customer can have only one account.
- b) Further change your diagram so an account can have only one customer.
- ! c) Change your original diagram of Exercise 4.1.1 so that a customer can have a set of addresses (which are street-city-state triples) and a set of phones. Remember that we do not allow attributes to have nonprimitive types, such as sets, in the E/R model.
- ! d) Further modify your diagram so that customers can have a set of addresses, and at each address there is a set of phones.

Esercizi dal libro – 4.2.1

Exercise 4.2.1: In Fig. 4.14 is an E/R diagram for a bank database involving customers and accounts. Since customers may have several accounts, and accounts may be held jointly by several customers, we associate with each customer an “account set,” and accounts are members of one or more account sets. Assuming the meaning of the various relationships and attributes are as expected given their names, criticize the design. What design rules are violated? Why? What modifications would you suggest?

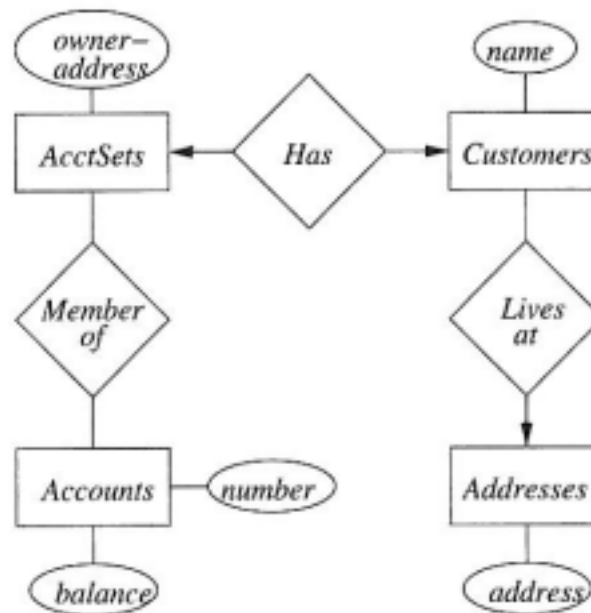


Figure 4.14: A poor design for a bank database

Esercizi dal libro – 4.5.1

Exercise 4.5.1 : Convert the E/R diagram of Fig. 4.29 to a relational database schema.

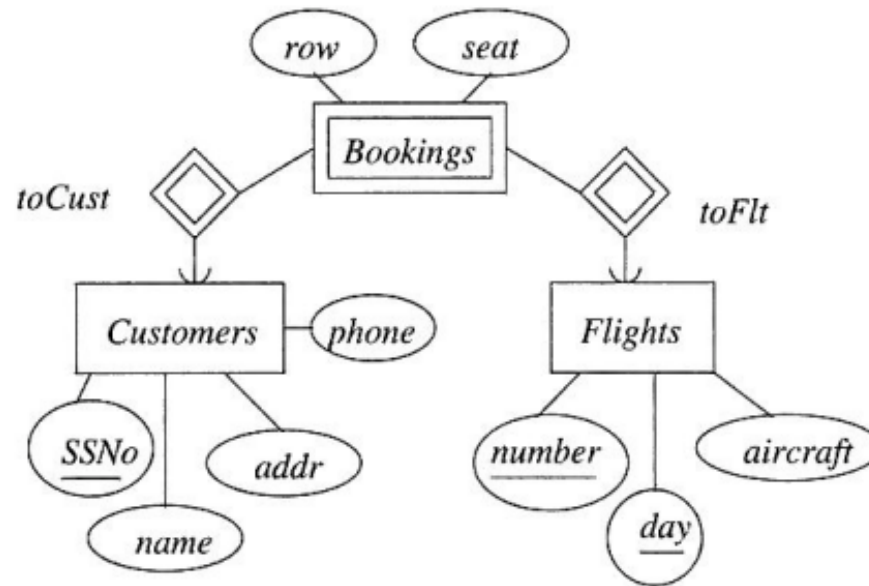


Figure 4.29: An E/R diagram about airlines

Esercizi dal libro – 4.6.1

Exercise 4.6.1: Convert the E/R diagram of Fig. 4.32 to a relational database schema, using each of the following approaches:

- a) The nulls method.
- b) The straight-E/R method.

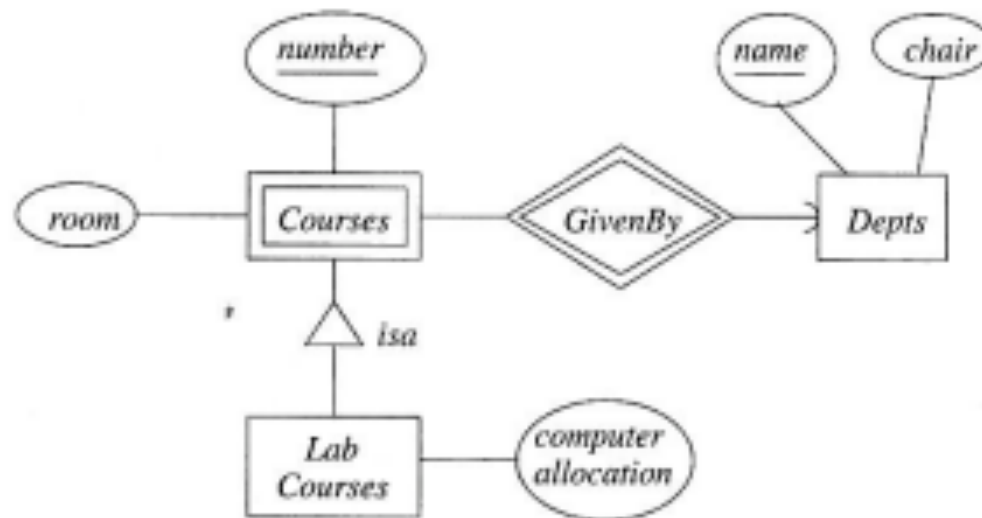


Figure 4.32: E/R diagram for Exercise 4.6.1

Esercizi dal libro – 4.6.2

! Exercise 4.6.2: Convert the E/R diagram of Fig. 4.33 to a relational database schema, using:

- a) The nulls method.
- b) The straight-E/R method.

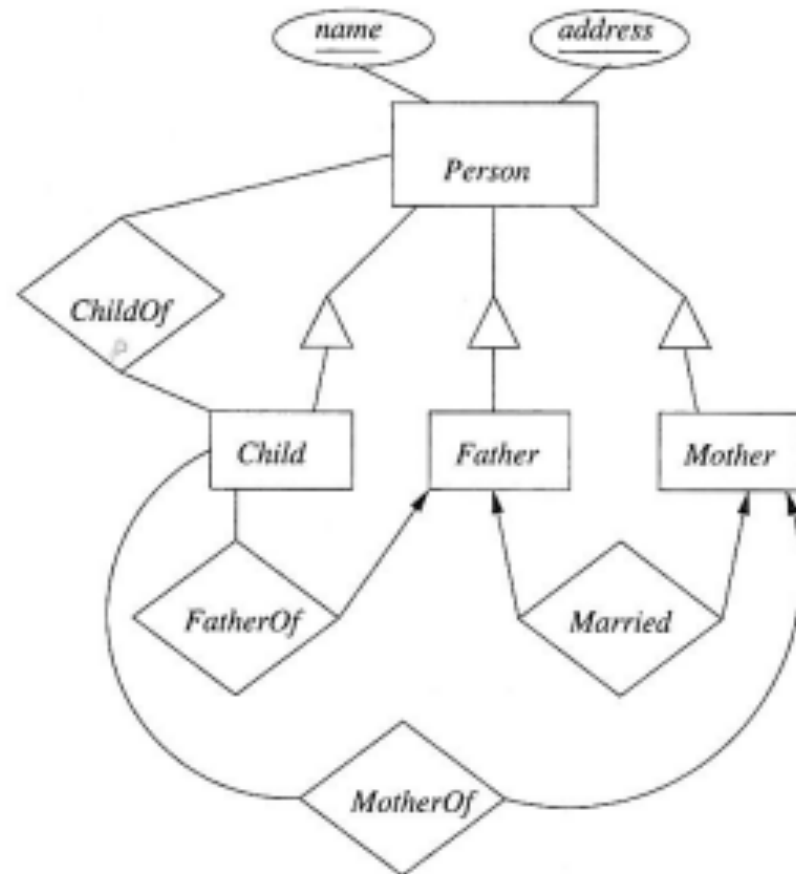


Figure 4.33: E/R diagram for Exercise 4.6.2