

## **Esercizi progettazione DB – 2011/2012 – 5B**

<b>ESERCIZIO 1: CLUB.....</b>	<b>2</b>
<b>ESERCIZIO 2: MEDICI .....</b>	<b>4</b>
<b>ESERCIZIO 3: SQUADRE .....</b>	<b>6</b>
<b>ESERCIZIO 4: CORSI DI BALLO .....</b>	<b>8</b>
<b>ESERCIZIO 5: AUTO .....</b>	<b>10</b>
<b>ESERCIZIO 6: FILIALI .....</b>	<b>12</b>
<b>ESERCIZIO 7: APPARTAMENTI .....</b>	<b>14</b>
<b>ESERCIZIO 8: CIRCOLO DI TENNIS (COMPITO).....</b>	<b>16</b>
<b>ESERCIZIO 9: FILM (COMPITO) .....</b>	<b>18</b>
<b>ESERCIZIO 10: MOSTRA CANINA .....</b>	<b>20</b>
<b>ESERCIZIO 11: AUTO USATE .....</b>	<b>24</b>
<b>ESERCIZIO 12: OSPEDALE (COMPITO) .....</b>	<b>25</b>
<b>ESERCIZIO 13: NOLEGGIO DVD (COMPITO) .....</b>	<b>27</b>
<b>ESERCIZIO 14: VENDITORI (NORMALIZZAZIONE).....</b>	<b>29</b>
<b>ESERCIZIO 15: SCRUTINI .....</b>	<b>31</b>
<b>ESERCIZIO 16: SUPERMERCATO .....</b>	<b>37</b>
<b>ESERCIZIO 17: VETERINARIO .....</b>	<b>39</b>

## Esercizio 1: Club

Progettare l'archivio dei soci di un club

### Svolgimento

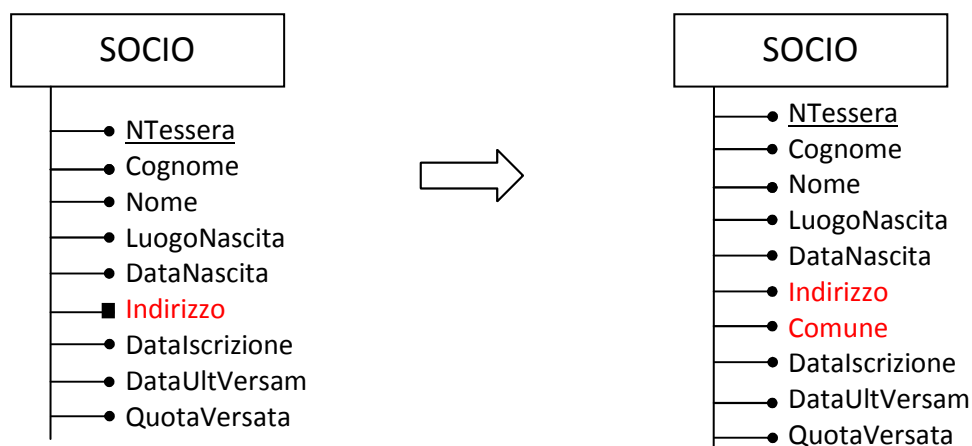
#### ANALISI DELLA REALTÀ DI INTERESSE

Il progetto è rivolto alla gestione informatizzata dei soci di un club. Si ritengono rilevanti i seguenti dati: Numero Tessera, Cognome, Nome, Luogo di Nascita, Data di Nascita, Indirizzo di residenza, Data di prima iscrizione, Data Ultimo Versamento, Quota versata per l'anno corrente.

IPOTESI AGGIUNTIVE.

Si suppone che l'archivio sia di tipo **storico** perché in caso di rientro del socio i suoi dati sono già disponibili e non è necessario attribuirgli un nuovo numero di tessera.

#### SCHEMA CONCETTUALE



VINCOLI ESPLICITI

V1 (SOCIO): DataUltVersam >= DataIscrizione

V2 (SOCIO): DataNascita > #01/01/1900#

#### SCHEMA LOGICO

tblSoci(NTessera, Cognome, Nome, LuogoNascita, DataNascita, Indirizzo, Comune, DataIscrizione, DataRinnovo, QuotaVersata)

#### SCHEMA FISICO

Percorso fisico: **Utenti\5Bs\DB**

Nome del Database: **db01\_Soci.accdb**

## Tabelle

### tblSoci

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	NTessera	Contatore	Intero lungo	SI	
	Cognome	Testo	30	SI	
	Nome	Testo	30	SI	
	LuogoNascita	Testo	40	SI	
	DataNascita	Data/Ora		SI	> #01/01/1900#
	Indirizzo	Testo	40	SI	
	Comune	Testo	30	SI	
	DataIscrizione	Data/Ora		SI	
	DataUltVersam	Data/Ora		SI	>=DataIscrizione
	QuotaVersata	Valuta		SI	

## Esercizio 2: Medici

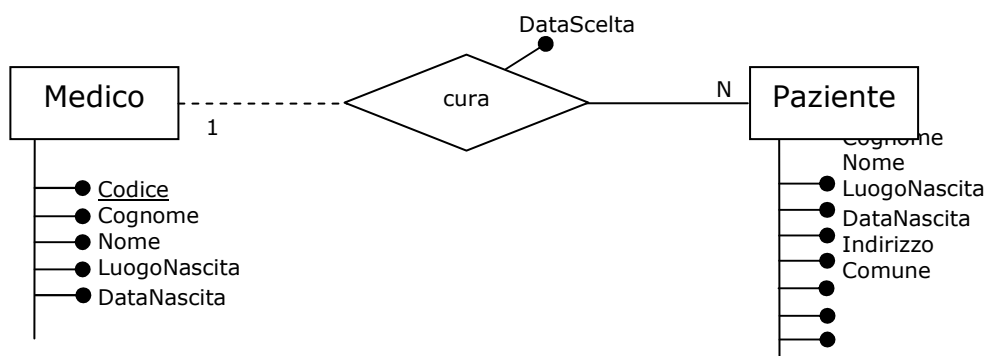
Progettare una base di dati per memorizzare i dati dei medici generici che fanno capo ad una ASL e dei relativi pazienti. Di ogni medico devono essere registrati un codice, cognome, nome, data e luogo di nascita; di ogni paziente devono essere registrati codice fiscale, cognome, nome, data e luogo di nascita, indirizzo.

### Svolgimento

#### ANALISI DELLA REALTÀ DI INTERESSE

L'ambito del progetto è la gestione informatizzata di una Azienda Sanitaria Locale, limitatamente ai medici di base con l'elenco dei rispettivi pazienti.

#### SCHEMA CONCETTUALE



#### SCHEMA LOGICO

tblMedici(Codice, Cognome, Nome, LuogoNascita, DataNascita)

tblPazienti(CF, Cognome, Nome, LuogoNascita, DataNascita, Indirizzo, Comune, CodiceMedico, DataScelta)

#### SCHEMA FISICO

Percorso fisico: **Utenti\5Bs\DB**

Nome del Database: **db02\_ASL.accdb**

#### Tabelle

##### tblMedici

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Codice	Testo	6	SI	
	Cognome	Testo	30	SI	
	Nome	Testo	30	SI	
	LuogoNascita	Testo	40	SI	
	DataNascita	Data/Ora		SI	> #01/01/1930#

**tblPazienti**

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	CF	Testo	16	SI	
	Cognome	Testo	30	SI	
	Nome	Testo	30	SI	
	LuogoNascita	Testo	40	SI	
	DataNascita	Data/Ora		SI	> #01/01/1900#
	Indirizzo	Testo	40	SI	
	Comune	Testo	30	SI	
FK	CodiceMedico	Testo	6	SI	
	DataScelta	Data/Ora		SI	

Chiave esterna: CodiceMedico

referenzia: tblMedici (Codice)

## Esercizio 3: Squadre

Progettare una base di dati per memorizzare i dati delle squadre di calcio dei campionati italiani in corso, compresi: giocatori, città e serie di appartenenza.

### Svolgimento

#### ANALISI DELLA REALTÀ DI INTERESSE

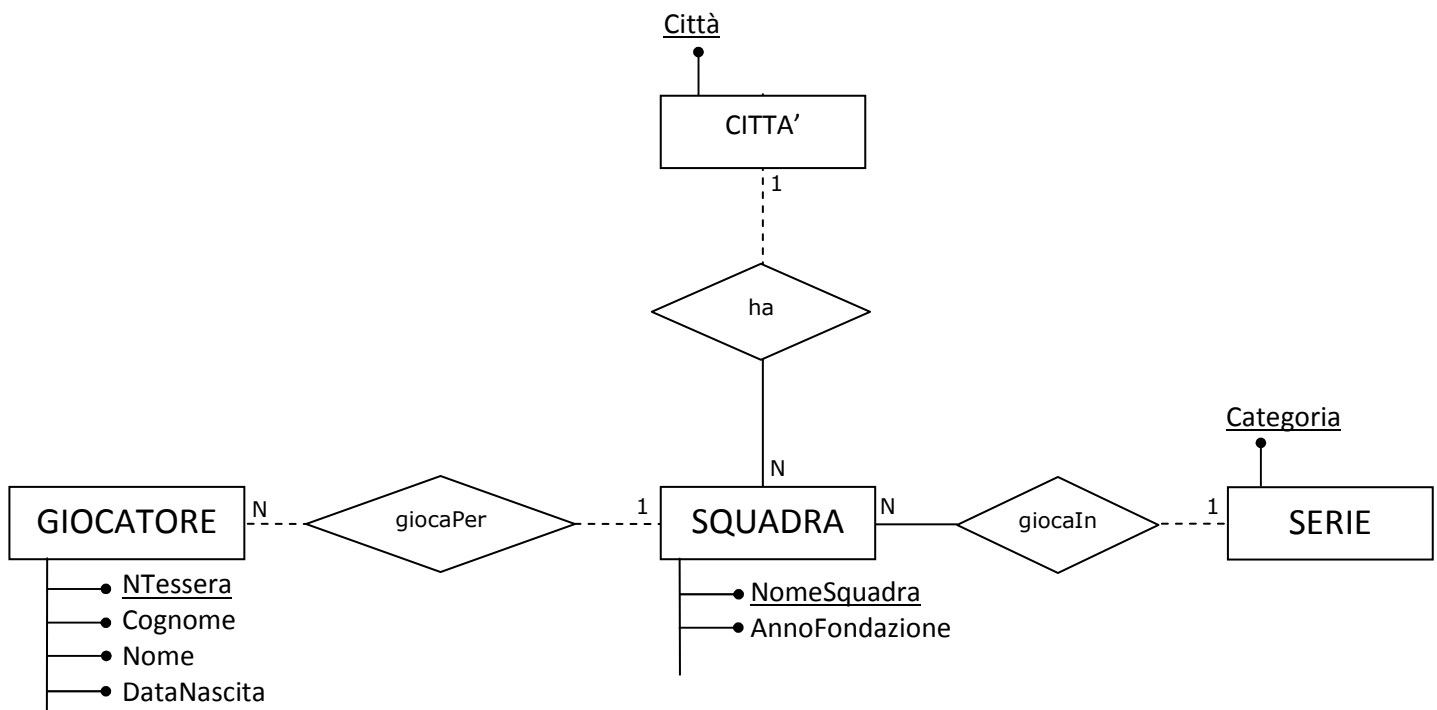
Occorre progettare un database per la gestione informatizzata dei campionati italiani di calcio relativi alla sola stagione in corso. Le entità individuate sono: Squadra, Giocatore, Città, Serie. Tra queste, la Squadra è l'entità a cui risultano associate tutte le altre.

#### IPOTESI AGGIUNTIVE

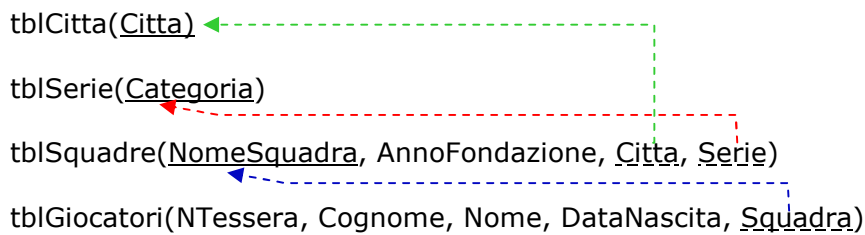
Ogni giocatore ha una tessera della FIGC, con un numero che lo identifica univocamente, qualunque sia la squadra in cui gioca. Se un giocatore rimane senza squadra (svincolato), il suo numero di tessera rimane a lui riservato.

Nel DB ci saranno solo le squadre che si sono iscritte ad una qualunque serie per partecipare al relativo campionato.

#### SCHEMA CONCETTUALE



## SCHEMA LOGICO



## SCHEMA FISICO

Percorso fisico: **Utenti\5Bs\DB**

Nome del Database: **db03\_ASLSquadre.accdb**

### Tabelle

#### tblCitta

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Citta	Testo	20	Sì	

#### tblSerie

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Categoria	Testo	20	Sì	

#### tblSquadre

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	NomeSquadra	Testo	20	Sì	
	AnnoFondazione	Intero		Sì	
FK	Citta	Testo	20	Sì	
	Serie	Testo	20	Sì	

Chiave esterna: Citta

referenzia: tblCitta(Citta)

#### tblGiocatori

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	NTessera	Testo	5	Sì	
	Cognome	Testo	30	Sì	
FK	Nome	Testo	30	Sì	
	Squadra	Testo	20	Sì	

Chiave esterna: Squadra

referenzia: tblSquadre(NomeSquadra)

## Esercizio 4: Corsi di ballo

Progettare un DB per gestire i dati dei corsi di ballo erogati da una scuola di danza. Per ogni corso devono essere registrati il nome, l'orario, il nome dell'istruttore e gli iscritti.

### Svolgimento

#### ANALISI DELLA REALTÀ DI INTERESSE

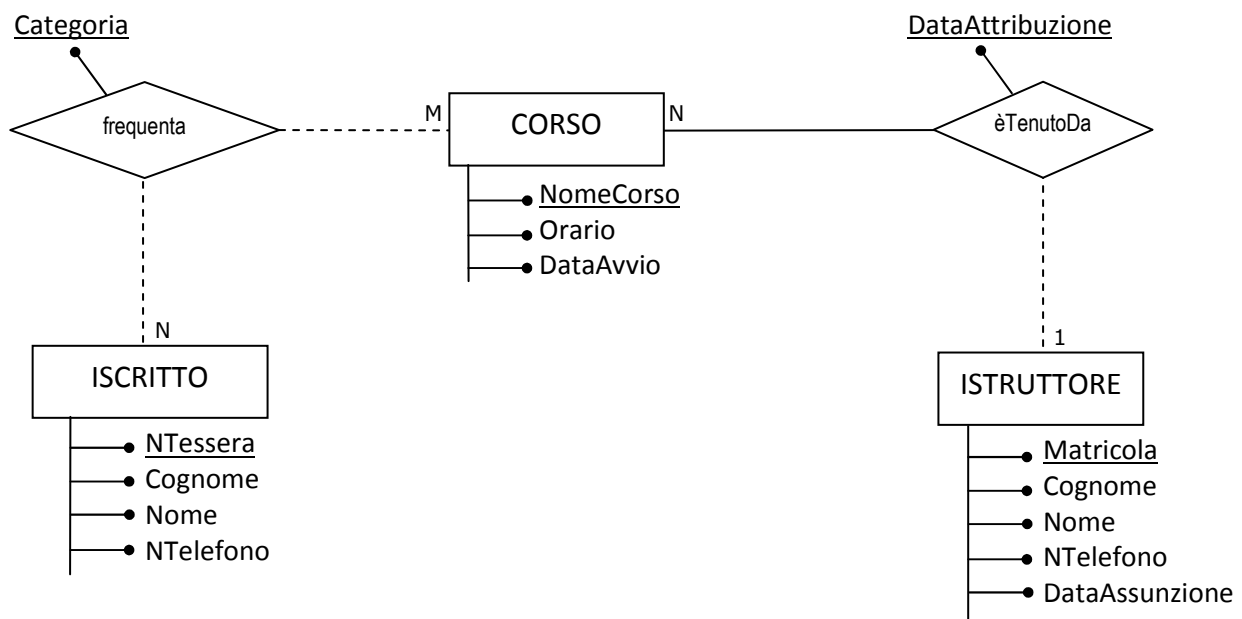
La realtà di interesse è costituita dalla scuola di danza, di cui si devono gestire i corsi. Le entità individuate sono Corso, Istruttore, Iscritto. Ogni corso è tenuto da un solo istruttore.

#### IPOTESI AGGIUNTIVE

Si suppone che l'archivio sia di tipo **attuale** perché è molto probabile che la traccia intenda far riferimento alla gestione solo dei corsi attualmente erogati.

Si suppone che un istruttore possa anche tenere più corsi, e che un cliente possa iscriversi a più corsi.

#### SCHEMA CONCETTUALE



#### SCHEMA LOGICO

```
tblIstruttori(Matricola, Cognome, Nome, Telefono, DataAssunzione)
tblCorsi(NomeCorso, Orario, DataAvvio, Istruttore, DataAttribuzione)
tblIscritti(NTessera, Cognome, Nome, Telefono)
tblIscrizioni(NTessera, NomeCorso, DataIscrizione)
```



## SCHEMA FISICO

Percorso fisico: **Utenti\5Bs\DB**

Nome del Database: **db04\_ASLScuolaBallo.accdb**

### Tabelle

#### tblIstruttori

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Matricola	Testo	5	Sì	
	Cognome	Testo	20	Sì	
	Nome	Testo	20	Sì	
	Telefono	Testo	15	Sì	
	DataAssunzione	Data/ora		Sì	

#### tblCorsi

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	NomeCorso	Testo	20	Sì	
	Orario	Data/ora			
	DataAvvio	Data/ora			
FK	Istruttore	Testo	5		
	DataAttribuzione	Data/ora			

Chiave esterna: Istruttore

referenzia: tblIstruttori(Matricola)

#### tblIscritti

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	NTessera	Testo	5	Sì	
	Cognome	Testo	20	Sì	
	Nome	Testo	20	Sì	
	Telefono	Testo	15	Sì	

#### tblIscrizioni

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
FK, K	NTessera	Testo	5	Sì	
FK, K	NomeCorso	Testo	20	Sì	
	DataIscrizione	Data/ora		Sì	

Chiave esterna: NTessera

referenzia: tblIscritti(NTessera)

Chiave esterna: NomeCorso

referenzia: tblCorsi(NomeCorso)

## Esercizio 5: Auto

Un automobilista vuole gestire le informazioni che riguardano gli interventi effettuati sulla sua auto da meccanici, carrozzieri, elettrauti, ma anche rifornimenti di benzina, cambio dell'olio, ecc. Per ogni intervento occorre memorizzare la data, la durata, la descrizione dell'intervento, il costo, la persona che ha fornito il servizio, indicandone la qualifica, l'indirizzo e il recapito telefonico.

### Svolgimento

#### ANALISI DELLA REALTÀ DI INTERESSE

La realtà di interesse è costituita dagli interventi effettuati sulle auto di una persona. Ogni intervento riguarda una sola auto. Un'auto può ricevere uno o più interventi.

#### IPOTESI AGGIUNTIVE

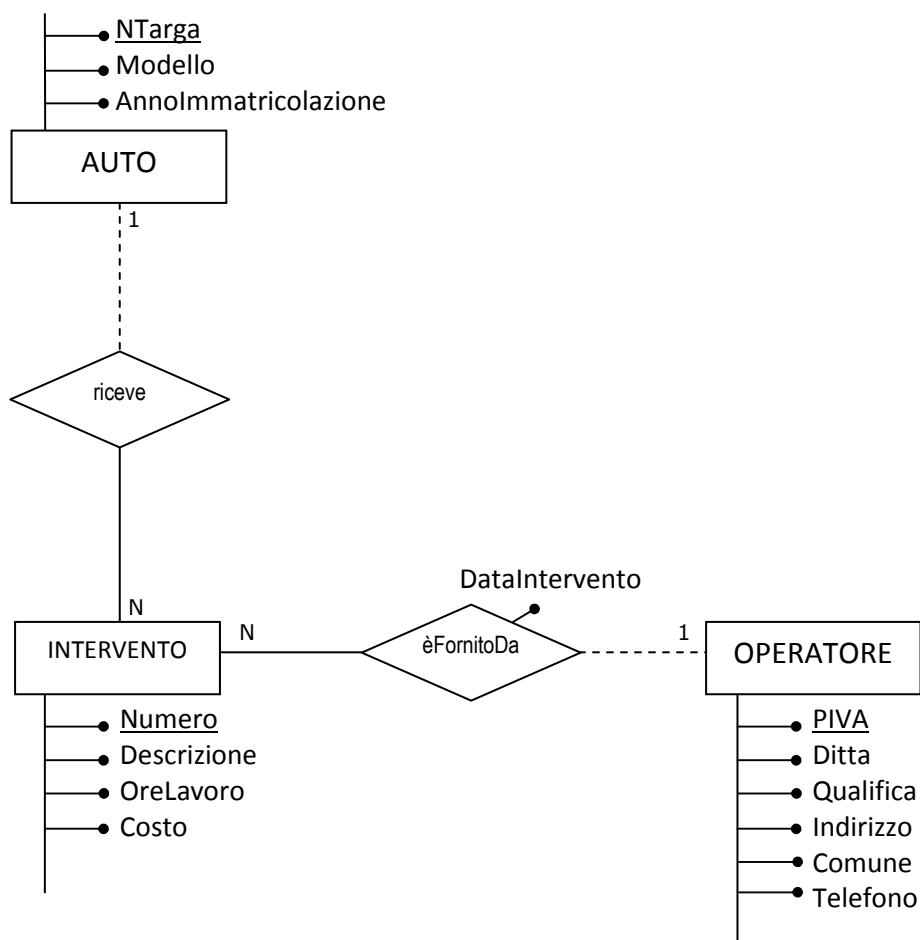
La persona può avere più auto.

Ogni intervento ha un numero d'ordine.

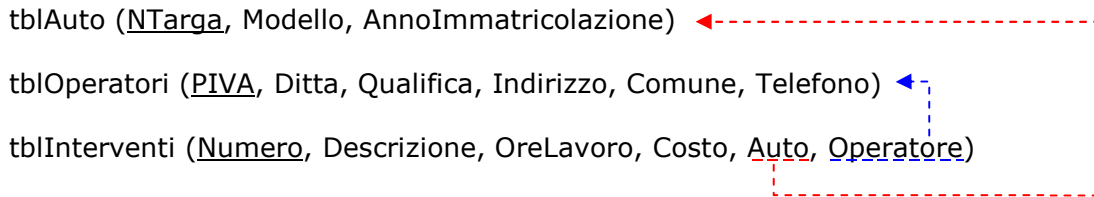
La durata di un intervento è formulata come numero di ore di lavoro.

Ogni operatore può avere una sola qualifica.

#### SCHEMA CONCETTUALE



## SCHEMA LOGICO



## SCHEMA FISICO

Percorso fisico: **Utenti\5AS\DB**  
 Nome del Database: **db05\_Auto.accdb**

### Tabelle

#### tblRazze

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	NTarga	Testo	8	Sì	
	Modello	Testo	20	Sì	
	AnnoImmatricolazione	Numerico	Integer	Sì	>= 1900

#### tblOperatori

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	PIVA	Testo	18	SI	
	Ditta	Testo	30	SI	
	Qualifica	Testo	20	SI	
	Indirizzo	Testo	30	SI	
	Comune	Testo	20	SI	
	Telefono	Testo	15	SI	

#### tblInterventi

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Numero	Numerico	Integer	SI	
	Descrizione	Testo	50	SI	
	OreLavoro	Numerico	Byte	SI	
	Costo	Valuta			
FK	Auto	Testo	8		
FK	Operatore	Testo	18		

Chiave esterna (Auto)                      referencia: tblAuto(Targa)

Chiave esterna (Operatore)              referencia: tblOperatori(PIVA)

## Esercizio 6: filiali

Progettare una base di dati per memorizzare i dati delle filiali una grande banca e dei direttori che le gestiscono. Ogni filiale ha un solo direttore, che si occupa solo di quella filiale.

### Svolgimento

#### ANALISI DELLA REALTÀ DI INTERESSE

La realtà d'interesse è il sistema di filiali di una grande banca.

Il DB da progettare non è storico, perché si registrano solo i dati attuali.

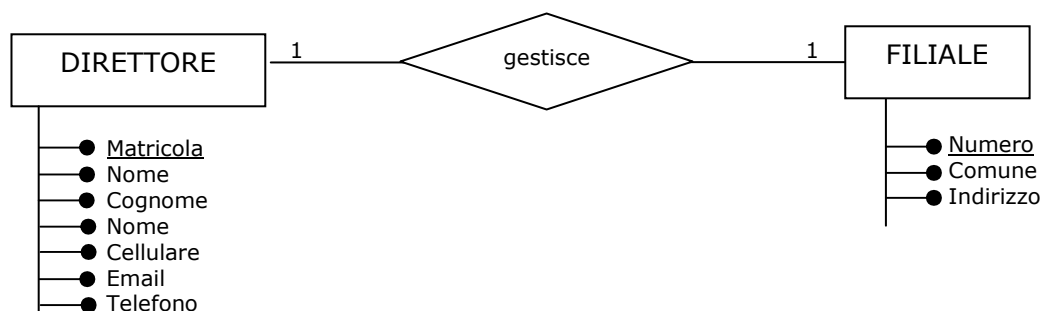
Si individuano come entità FILIALE e DIRETTORE. Tra esse si individua un'associazione 1:1.

#### IPOTESI AGGIUNTIVE

Una filiale deve avere obbligatoriamente un direttore e un direttore deve occuparsi obbligatoriamente di una filiale.

Si ritiene opportuno registrare, per ogni filiale, un numero (che la identifica univocamente), il comune e l'indirizzo, mentre, per ogni direttore, una matricola (che lo identifica univocamente), il cognome, il nome, un numero di cellulare e un indirizzo email.

#### SCHEMA CONCETTUALE



#### SCHEMA LOGICO

tblFiliali (Numero, Comune, Indirizzo, MatrDir, CognomeDir, NomeDir, CellDir, EmailDir)

#### SCHEMA FISICO

Percorso fisico: **Utenti\5AS\DB**

Nome del Database: **db06\_ Filiali.accdb**

## Tabelle

### tblFiliali

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Numero	Numerico	Intero lungo	SI	
	Comune	Testo	25	SI	
	Indirizzo	Testo	40	SI	
SK	MatrDir	Testo	4	NO	
	CognomeDir	Testo	25	SI	
	NomeDir	Testo	25	SI	
	CellDir	Testo	13	NO	
SK	Email	Testo	40	NO	

## Esercizio 7: appartamenti

Progettare una base di dati per memorizzare i dati degli appartamenti di uno stabile e dei rispettivi proprietari. Si ipotizzi che ogni appartamento abbia un solo proprietario e che ogni proprietario detenga un solo appartamento.

### Svolgimento

#### ANALISI DELLA REALTÀ DI INTERESSE

La realtà d'interesse è l'insieme degli appartamenti di uno stabile.

Il DB da progettare non è storico, perché si registrano solo i dati attuali.

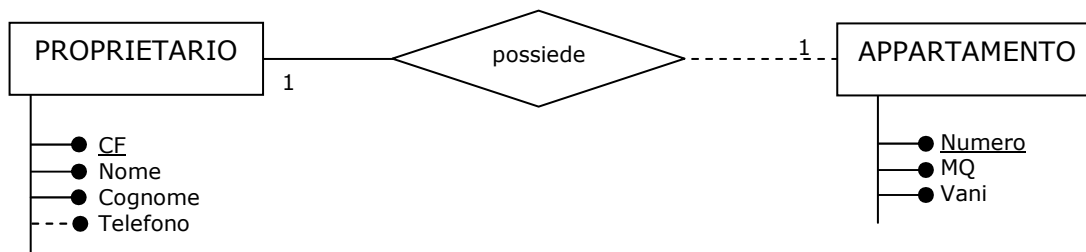
Si individuano come entità PROPRIETARIO e APPARTAMENTO. Tra esse si individua un'associazione 1:1.

#### IPOTESI AGGIUNTIVE

Una filiale deve avere obbligatoriamente un proprietario e un direttore deve occuparsi obbligatoriamente di una filiale.

Si ritiene opportuno registrare, per ogni filiale, un numero (che la identifica univocamente), il comune e l'indirizzo, mentre, per ogni direttore, una matricola (che lo identifica univocamente), il cognome, il nome, un numero di cellulare e un indirizzo email.

#### SCHEMA CONCETTUALE



#### SCHEMA LOGICO

tblAppartamenti (Numero, MQ, Vani)

tblProprietari (CF, Cognome, Nome, {Telefono}, NumeroApp)

#### VINCOLI ESPLICITI

V1 (tblAppartamenti): Numero > 0

V2 (tblAppartamenti): MQ > 40

#### SCHEMA FISICO

Percorso fisico: **Utenti\5AS\DB**

Nome del Database: **db07\_ Appartamenti.accdb**

## Tabelle

### tblAppartamenti

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Numero	Numerico	Byte	SI	>0
	MQ	Numerico	Byte	SI	>40
	Vani	Numerico	Byte	SI	>1

### tblProprietari

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	CF	Testo	16	SI	
	Cognome	Testo	25	SI	
	Nome	Testo	25	SI	
	Telefono	Testo	13	NO	
SK, FK	NumeroApp	Numerico	Byte	SI	

## Esercizio 8: Circolo di tennis (compito)

Un circolo di tennis vuole memorizzare le prenotazioni dei propri campi da parte dei propri soci. Ogni prenotazione viene effettuata da un solo socio, riguarda un solo campo, ha validità di un'ora intera di un dato giorno, e può andare dalle 9:00 alle 21:00.

Di ogni socio interessano i dati anagrafici e il recapito. Di ogni campo interessa sapere il tipo (ad es. coperto e in terra battuta, coperto e in cemento, scoperto, etc.). I tipi non sono predefiniti.

Progettare la base di dati fino allo schema fisico.

### Svolgimento

#### ANALISI DELLA REALTÀ DI INTERESSE

Oggetto dell'automazione è la gestione della prenotazioni dei campi da tennis di un circolo.

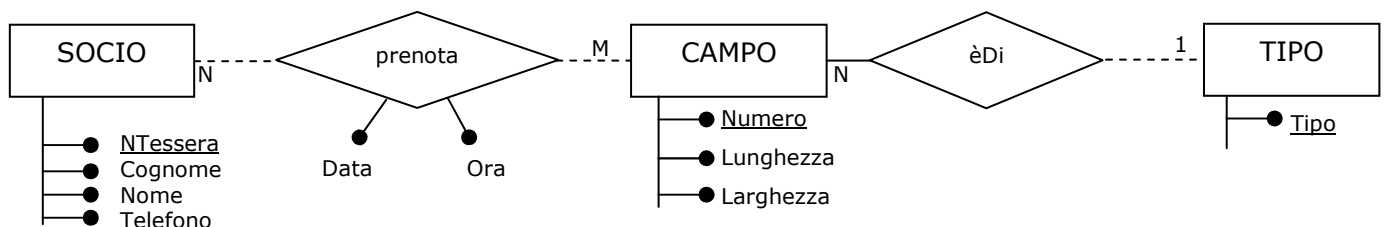
Il DB da progettare è storico, perché conserva le prenotazioni nel tempo (dalla traccia non si rileva la necessità di cancellarle).

Si individuano come entità CAMPO, SOCIO e TIPO CAMPO. La prenotazione può essere vista come associazione tra CAMPO e SOCIO.

#### IPOTESI AGGIUNTIVE

Si ritiene che i dati di interesse di ogni socio siano un numero di tessera (che lo identifica univocamente), cognome, nome e recapito telefonico), mentre per il campo si ritiene di dover registrare lunghezza, larghezza e un identificativo, che potrebbe essere un numero.

#### SCHEMA CONCETTUALE



#### SCHEMA LOGICO

tblSoci (NTessera, Cognome, Nome, Telefono)  
tblTipi (Tipo)  
tblCampi (Numero, Lunghezza, Larghezza, Tipo)  
tblPrenota (Socio, Campo, Data, Ora)



N.B.: La particolare chiave di tblPrenota è dovuta alla necessità di evitare che uno stesso campo possa essere prenotato da più persone nello stesso momento (giorno e ora).



## SCHEMA FISICO

Percorso fisico: **Utenti\5AS\DB**

Nome del Database: **db07\_ Appartamenti.accdb**

### Tabelle

#### tblSoci

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	NTessera	Numerico	Integer	SI	>0
	Cognome	Testo	20	SI	
	Nome	Testo	20	SI	
	Telefono	Testo	15		

#### tblTipi

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Tipo	Testo		SI	

#### tblCampi

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Numero	Numerico	Byte	SI	
	Lunghezza	Numerico	Integer	SI	
	Larghezza	Numerico	Integer	SI	
	Tipo	Testo	20	SI	

## Esercizio 9: Film (compito)

Si vogliono trattare informazioni relative a produttori e attori di film. Degli attori interessano le generalità anagrafiche. Dei produttori, la Ragione Sociale, l'anno di fondazione e il recapito. Di un film interessano il titolo, l'anno di produzione, gli attori e il produttore. Nello stesso anno non possono essere prodotti film con lo stesso titolo.

Progettare la base di dati fino allo schema fisico

### Svolgimento

#### ANALISI DELLA REALTÀ DI INTERESSE

La realtà d'interesse è costituita da attori e produttori di film. Il Database registra dati anche di film passati, per cui è storico.

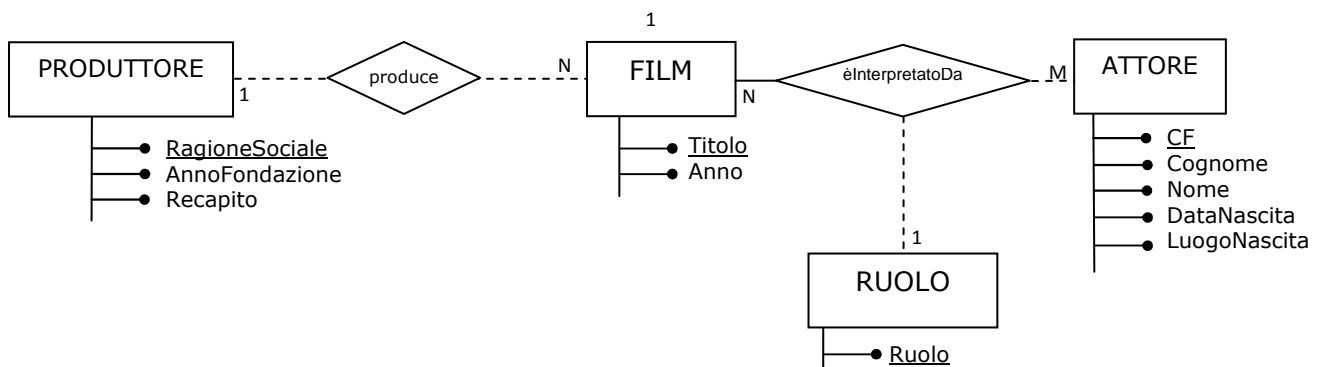
Si individuano come entità PRODUTTORE, ATTORE, FILM.

#### IPOTESI AGGIUNTIVE

Degli attori si ritiene di dover registrare codice fiscale, che li identifica univocamente, cognome, nome, data di nascita e luogo di nascita.

Si ritiene di dover registrare anche il ruolo che gli attori interpretano nel film (protagonista, co-protagonista, comparsa, ecc.), supponendo che un attore non possa avere più di un ruolo nello stesso film).

#### SCHEMA CONCETTUALE



#### SCHEMA LOGICO

tblProduttori (RagioneSociale, AnnoFondazione, Recapito)

tblFilm (Titolo, Anno, Produttore)

tblRuoli (Ruolo)

tblAttori (CF, Cognome, Nome, DataNascita, LuogoNascita)

tblInterpretazioni (Attore, Film, Ruolo)

## SCHEMA FISICO

Percorso fisico: **Utenti\5AS\DB**

Nome del Database: **db09\_Film.accdb**

### Tabelle

#### tblProduttori

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	RagioneSociale	Testo	20	Sì	
	AnnoFondazione	Numerico	Integer	Sì	
	Recapito	Testo	40	Sì	

#### tblFilm

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Titolo	Testo	20	SI	
	Anno	Numerico	Integer	SI	
FK	Produttore	Testo	20	SI	

Chiave esterna (Produttore)      referencia: tblProduttori(RagioneSociale)

#### tblRuoli

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Ruolo	Testo	20	SI	

#### tblAttori

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	CF	Testo	16	SI	
	Cognome	Testo	20	SI	
	Nome	Testo	20	SI	
	DataNascita	Data/Ora			
	LuogoNascita	Testo	20		

#### tblInterpretazioni

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
FK,K	Attore			SI	
FK,K	Film			SI	
	Ruolo			SI	

Chiave esterna (Attore)      referencia: tblAttori(CF)

Chiave esterna (Film)      referencia: tblFilm(Titolo)

## Esercizio 10: MOSTRA CANINA

Progettare una base di dati per la gestione di una mostra canina. Di ogni cane, identificato da un codice, interessano il nome, la data di nascita, l'altezza, il peso, la razza di appartenenza, e i dati del proprietario. Le razze si distinguono dal nome, e possiedono un'altezza e un peso standard. Ogni giudice, identificato da un codice, esprime un voto su ciascun cane.

Realizzare l'analisi della realtà d'interesse, con eventuali ipotesi aggiuntive, lo schema concettuale, lo schema logico, lo schema fisico e un'applicazione per la gestione dei dati. Sviluppare in Access sia la base di dati, sia l'applicazione.

### Svolgimento

#### ANALISI DELLA REALTÀ DI INTERESSE

La realtà d'interesse è una mostra canina. Il Database da realizzare ha breve durata nel tempo, perché è destinato alla gestione di una mostra canina, quindi è senz'altro di tipo attuale. Sia i giudici che i cani in concorso sono in numero imprecisato. Ogni cane sarà valutato da tutti i giudici. Ogni giudice esprime un solo voto su ciascun cane. Si individuano come entità CANE, PROPRIETARIO, RAZZA, GIUDICE.

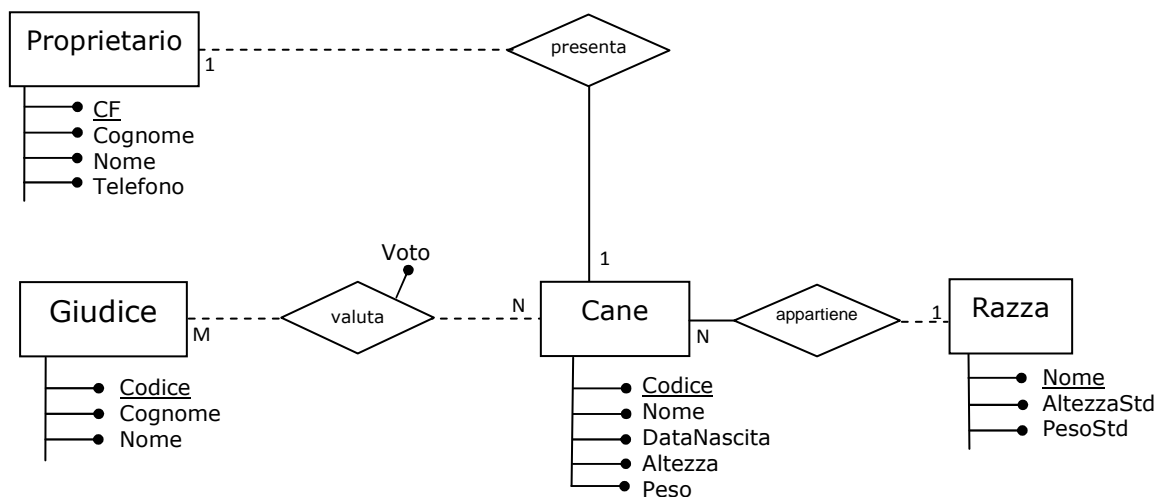
#### IPOTESI AGGIUNTIVE

Si suppone che ogni proprietario possa presentare un solo cane alla mostra.

Di ogni proprietario si ritiene di dover registrare il codice fiscale, che lo identifica univocamente, cognome, nome, indirizzo e città di residenza.

Di ogni giudice, oltre al codice, si ritiene di dover registrare almeno cognome e nome.

#### SCHEMA CONCETTUALE



#### SCHEMA LOGICO

tblRazze (NomeR, AltezzaStd, PesoStd) ←

tblCani (Codice, NomeCane, Razza, Altezza, Peso, DataNascita, CFPr, CognomePr, NomePr, TelefonoPr) ←

tblGiudici (Codice, Cognome, Nome) ←

tblValutazioni (CodGiudice, CodCane, Voto) ←

**Vincoli espliciti:**

V1 (tblRazze): AltezzaSt &gt;= 20

V2 (tblRazze): PesoSt &gt;= 2

V3 (tblCani): Altezza &gt;0

V4 (tblCani): Peso &gt;0

V5 (tblValutazioni): Voto Between 1 AND 10

**SCHEMA FISICO**Percorso fisico: **Utenti\5AS\DB**Nome del Database: **db09\_MostraCanina.accdb****Tabelle****tblRazze**

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	NomeR	Testo	20	Sì	
	AltezzaSt	Numerico	Byte	Sì	>= 20
	PesoSt	Numerico	Byte	Sì	>= 2

**tblCani**

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Codice	Numerico	Byte	SI	
	NomeCane	Testo	10	SI	
FK	Razza	Testo	20	SI	
	Altezza	Numerico	Byte	SI	>0
	Peso	Numerico	Byte	SI	>0
	DataNascita	Data		SI	
SK	CFPr	Testo	16	SI	
	CognomePr	Testo	20	SI	
	NomePr	Testo	20	SI	
	TelefonoPr	Testo	13	SI	

Chiave esterna (Razza)

referenzia: tblRazze(NomeR)

**tblGiudici**

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Codice	Testo	3	SI	
	Cognome	Testo	20	SI	
	Nome	Testo	20	SI	

**tblValutazioni**

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
FK,K	Giudice	Testo	3	SI	
FK,K	Cane	Numerico	Byte	SI	
	Voto	Numerico	Byte	SI	Between 1 and 10

Chiave esterna (Cane)

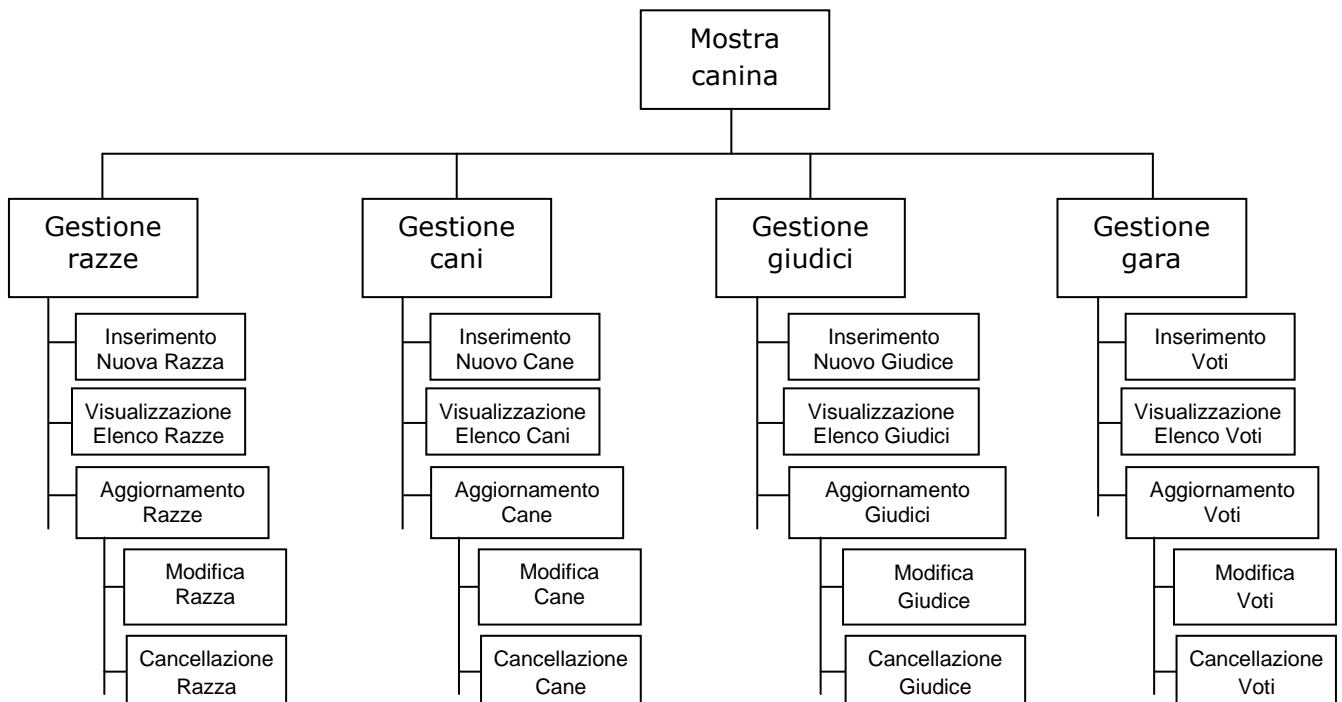
referenzia: tblCani(NumeroC)

Chiave esterna (Giudice)

referenzia: tblGiudici(CodGiud)

# APPLICAZIONE

## Diagramma delle funzioni



## Maschere

### Maschera di I livello

È il menu principale, che richiama le maschere per la gestione dei dati contenuti nelle singole tabelle, una per ogni tabella.



### Maschere di II livello



Mostra canina

### Gestione Giudici

- ☒ Inserimento giudici
- ☐ Aggiornamento giudici
- ☐ Visualizzazione elenco giudici
- ☐ Indietro



Mostra canina

### Gestione Gara

- ☒ Inserimento valutazioni
- ☐ Aggiornamento valutazioni
- ☐ Visualizzazione valutazioni
- ☐ Indietro



### Maschere di III livello

Per ogni tabella sono previste le operazioni di inserimento, aggiornamento, visualizzazione (3 maschere).

Per la tabella delle razze, esse sono:

Inserimento Razze

Nome

AltezzaStd

PesoStd

Salva Annulla Esci

Record: 1 di 1 Nessun filtro Cerca

Aggiornamento Razze

Nome

AltezzaStd

PesoStd

Salva Elimina Annulla Esci

Record: 1 di 11 Nessun filtro Cerca

Visualizzazione Razze

Nome	AltezzaStd	PesoStd
Alano Blu	60	48
Alano fulvo	65	50
Australian terrier	67	45
Bassotto standard	45	25
Bassotto tedesco	40	28
Cane lupo	70	60
Cane lupo cecoslov.	65	55
Golden retriever	50	45
Labrador retriever	55	15

Esci

Per le altre tabelle si procede analogamente.

## Esercizio 11: Auto usate

La Mini Car & C. vende auto usate, con la collaborazione di vari venditori. Prima di metterle in vendita registriamo su un database le loro caratteristiche (targa, numero di telaio, anno di immatricolazione, marca, modello, cilindrata, colore). Per ogni auto venduta ci interessa sapere la data e il prezzo di vendita, nonché il venditore.

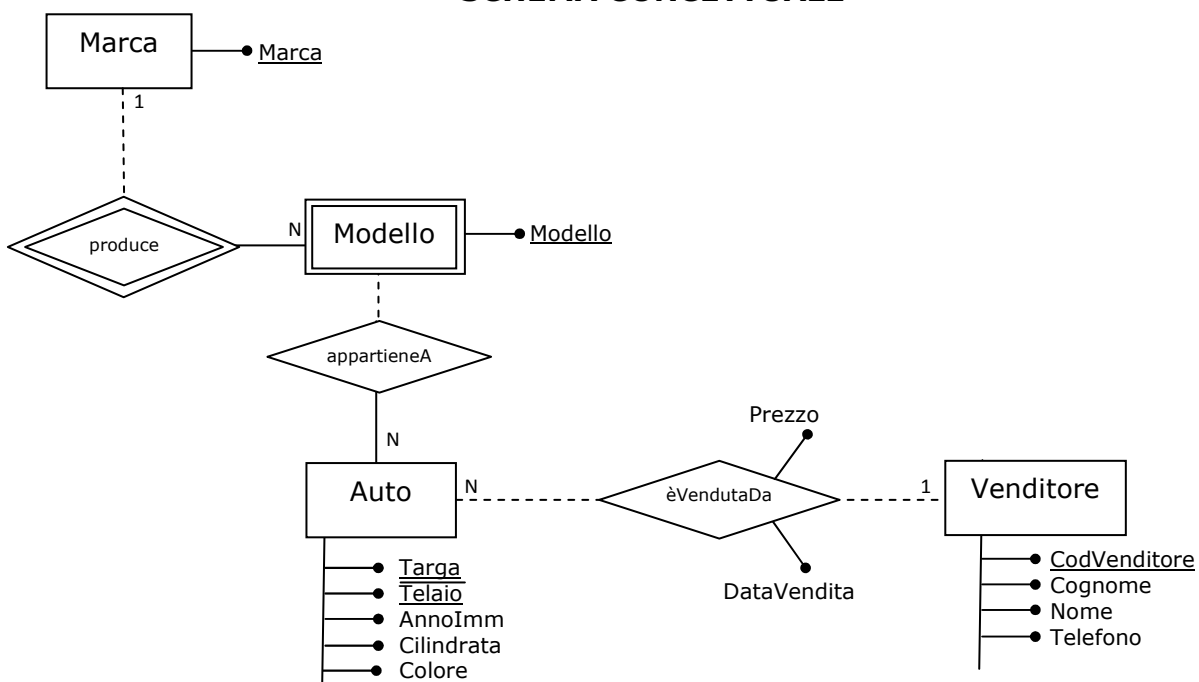
Progettare la base di dati fino allo schema fisico, e produrre l'analisi delle funzioni completa di diagramma delle funzioni e disegno dell'interfaccia utente (menu, maschere di immissione / visualizzazione / modifica / cancellazione e opportuni pulsanti).

### Svolgimento

#### ANALISI DELLA REALTÀ DI INTERESSE

La realtà d'interesse è un rivenditore di auto usate. Il Database da realizzare è di tipo storico, perché si deve tenere traccia delle vendite effettuate. Si individuano come entità AUTO, MARCA, MODELLO, VENDITORE.

#### SCHEMA CONCETTUALE



#### SCHEMA LOGICO

tblMarche(Marca)

tblModelli(Marca, Modello)

tblVenditori (CodVenditore, Cognome, Nome, Telefono)

tblAuto(Telaio, Targa, AnnoImm, Cilindrata, Colore, Marca, Modello, {CodVenditore, DataVendita, Prezzo})



## Esercizio 12: Ospedale (compito)

Il database dell'ospedale S. Riccardo detiene le informazioni anagrafiche dei pazienti attualmente in cura, compreso il gruppo sanguigno e il motivo del ricovero. L'ospedale è organizzato in reparti. Di ogni reparto sono rilevanti il nome e il piano. Per i dottori presenti nella struttura si registrano informazioni anagrafiche essenziali e la data di assunzione. Ogni medico è assegnato a un solo reparto e viene identificato da un codice progressivo all'interno del reparto. Il regolamento dell'ospedale vieta che i propri dottori, in caso di malattia, siano ricoverati internamente.

Progettare la base di dati fino allo schema fisico, e produrre l'analisi delle funzioni completa di diagramma delle funzioni e disegno dell'interfaccia utente: menu, maschere di immissione / visualizzazione / modifica / cancellazione e opportuni pulsanti (almeno una maschera di ciascun tipo, relativa a una tabella referenziante).

### Svolgimento

#### ANALISI DELLA REALTÀ DI INTERESSE

La realtà di interesse è l'ospedale San Riccardo. Il database da realizzare è di tipo attuale.

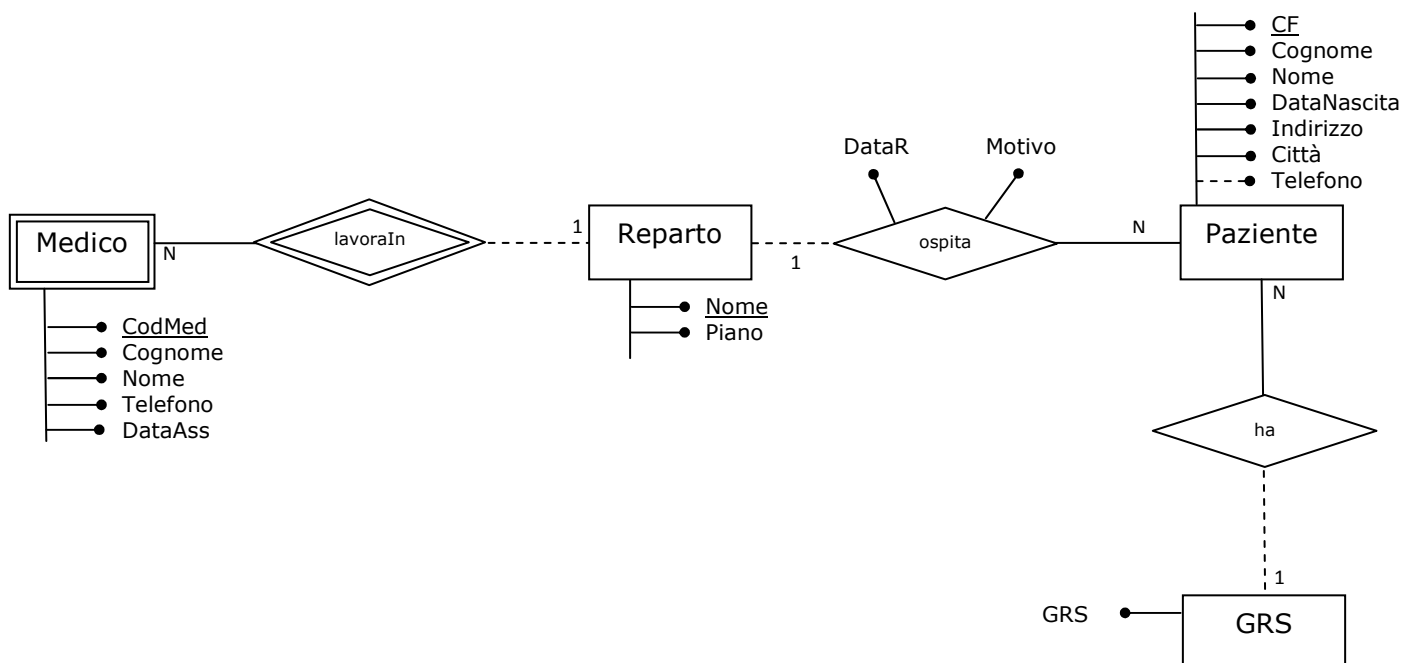
##### SCELTE IMPLEMENTATIVE:

Si ritiene più opportuno che il motivo del ricovero sia un testo digitabile liberamente dall'utente.

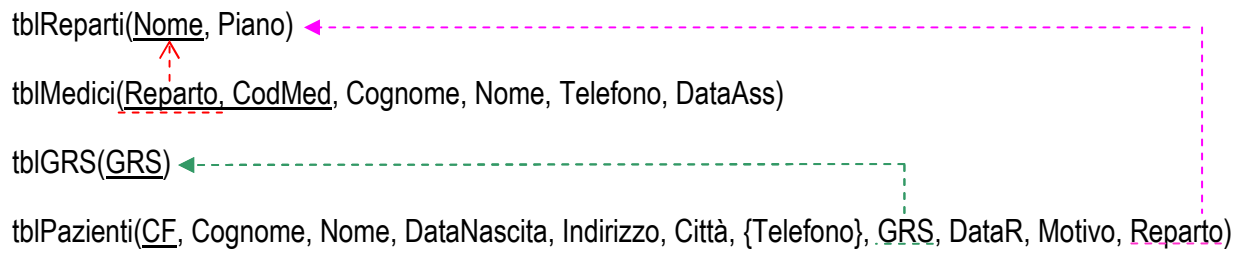
Dati anagrafici essenziali per medici e pazienti si ritiene debbano essere Cognome, Nome, Data e Luogo di Nascita.

ENTITÀ INDIVIDUATE: Paziente, Medico, Reparto, GruppoSanguigno.

#### SCHEMA CONCETTUALE



## SCHEMA LOGICO



## Esercizio 13: Noleggio DVD (compito)

VideoMovies noleggia film in DVD, elencati nel proprio catalogo. Per ogni film vanno registrate le copie disponibili (identificate da un numero progressivo), e il loro stato (disponibile: si / no). Occorre anche archiviare i clienti e tutti i noleggi che effettuano. Di ogni <noleggio>, oltre all'identificativo del cliente e al numero della copia presa in prestito, va registrata la data, la durata in giorni e l'importo.

Progettare la base di dati fino allo schema fisico, e produrre l'analisi delle funzioni completa di diagramma delle funzioni e disegno dell'interfaccia utente: menu, maschere di immissione / visualizzazione / modifica / cancellazione e opportuni pulsanti (almeno una maschera di ciascun tipo, relativa a una tabella referenziante).

### Svolgimento

#### ANALISI DELLA REALTÀ DI INTERESSE

La realtà di interesse è la videoteca VideoMovies.

Il database da realizzare è di tipo storico.

Entità individuate sono: Film, Copia, Cliente.

#### IPOTESI AGGIUNTIVE

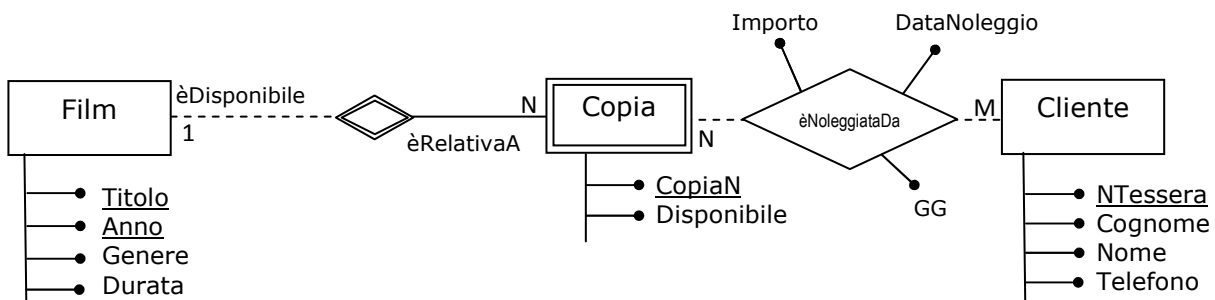
Non ci sono due film dello stesso anno e con lo stesso titolo.

La durata minima di un noleggio è un giorno (si rileva dalla traccia).

#### SCELTE IMPLEMENTATIVE

Si ritiene necessario archiviare, dei clienti, solo nome, cognome e numero di telefono.

#### SCHEMA CONCETTUALE



#### SCHEMA LOGICO

tblFilm(Titolo, Anno, Genere, Durata)

tblCopia(Titolo, Anno, CopiaN, Disponibile)

tblClienti(NTessera, Cognome, Nome, Telefono)

tblNoleggi(NTessera, Titolo, Anno, CopiaN, DataNoleggio, Importo, GG)

### SCHEMA FISICO

Tabella	Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se	Referenzia
<b>tblFilm</b>	PK	Titolo	Testo	30	Sì		
	PK	Anno	Numerico	Intero	Sì		
		Genere	Testo	30	Sì		
		Durata	Numerico	Intero	Sì		
<b>tblCopie</b>	PK, FK	Titolo	Testo	30	Sì		tblFilm
	PK, FK	Anno	Numerico	Intero	Sì		
	PK	NumeroC	Numerico	Byte	Sì		
		Disponibile	Booleano		Sì		
<b>tblClienti</b>	PK	Ntessera	Numerico	Int. lungo	Sì		
		Nome	Testo	20	Sì		
		Cognome	Testo	20	Sì		
		Tel	Testo	15	Sì		
<b>tblNoleggi</b>	FK	NTessera	Numerico	Int. lungo	Sì		tblClienti
	PK, FK	Titolo	Testo	30	Sì		tblCopie
	PK, FK	Anno	Numerico	Intero	Sì		
	PK, FK	NumeroC	Numerico	Byte	Sì		
	PK	DataNoleggio	Data/Ora		Sì		
		Importo	Valuta		Sì		
		GG	Numerico	Byte	Sì		

## Esercizio 14: Venditori (Normalizzazione)

Lo schema logico di un DB è costituito dalla seguente tabella (relazione):

tblVenditori(Cognome, Nome, Recapito, Telefono, Onomastico, AutoVendute)

Verificare se la relazione è normalizzata e, nel caso in cui non lo sia, eseguire la normalizzazione.

1FN)

La tabella non è in 1FN perché

- Recapito è un attributo composto da Indirizzo, CAP, Città;
- Onomastico è composto da Giorno e Mese;
- AutoVendute è composto da Targa e Prezzo di ogni auto venduta, cioè l'attributo è multiplo; inoltre in rapporto 1:N con la chiave primaria.

Per condurla in 1FN, la tabella tblVenditori deve essere divisa in due tabelle:

- tblVenditori(Cognome, Nome, Indirizzo, CAP, Città, Telefono, Mese, Giorno)
- tblAutoVendute (Targa, Prezzo, CognomeV, NomeV)

Se il rapporto è N:M, sono due le tabelle che si aggiungono. Es.:  
tblStudenti (Matr, Cognome, Nome, Sport)  
diventa

tblStudenti (Matr, Cognome, Nome) ←  
tblSport(NomeSport) ←  
tblSportPraticati(Matr, Sport)

2FN)

La tabella tblVenditori non è in 2FN perché Mese e Giorno dell'onomastico dipendono funzionalmente dal campo Nome, che è parte della chiave primaria. Pertanto lo schema logico viene così modificato:

- tblOnomastici(Nome, Mese, Giorno)
- tblVenditori(Cognome, Nome, Indirizzo, CAP, Città, Telefono)
- tblAutoVendute (Targa, Prezzo, CognomeV, NomeV)

3FN)

La tabella tblVenditori non è in 3FN perché il campo non chiave Città dipende dal campo non chiave CAP della stessa tabella.

Si divide allora tblVenditori in due tabelle e il modello logico diventa:

- tblOnomastici(Nome, Mese, Giorno)
  - tblComuni(CAP, Città)
  - tblVenditori(Cognome, Nome, Indirizzo, CAP, Telefono)
  - tblAutoVendute (Targa, Prezzo, CognomeV, NomeV)
- 
- ```
graph TD; ONOMASTICI[tblOnomastici] -- blue dashed arrow --> VENDITORI[tblVenditori]; COMUNI[tblComuni] -- green dashed arrow --> VENDITORI; AUTOVENDUTE[tblAutoVendute] -- red dashed arrow --> VENDITORI;
```

## Esercizio 15: Scrutini

Realizzare un database per gestire i voti finali, nelle varie materie, degli studenti di una classe. Va memorizzato anche il numero di ore di assenza in ciascuna materia. L'elenco delle materie non è noto a priori, ma viene introdotto dall'utente.

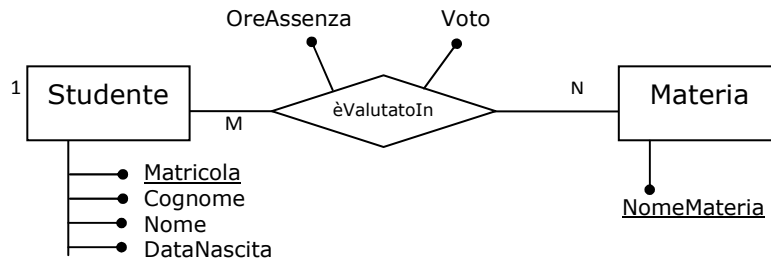
Eseguire le seguenti operazioni:

- Creare il DB con i comandi DDL.
- Popolare il DB con i seguenti dati, usando il comando INSERT parametrizzato:

| Studente                    | Materia, assenze, voto                                                    |
|-----------------------------|---------------------------------------------------------------------------|
| Astori Guido<br>(12/2/92)   | Italiano, 45, 7 – Matematica, 15, 6 - Inglese, 22, 7 – Informatica, 11, 6 |
| Seccia Vittoria<br>(4/6/90) | Italiano, 25, 6 – Matematica, 21, 5 - Inglese, 15, 5 – Informatica, 21, 4 |
| Dani Alberto<br>(23/3/91)   | Italiano, 30, 7 – Matematica, 33, 7 - Inglese, 13, 6 – Informatica, 31, 7 |
| Soleno Tina<br>(15/12/92)   | Italiano, 12, 8 – Matematica, 12, 9 - Inglese, 15, 8 – Informatica, 10, 7 |
| Ficco Andrea<br>(5/5/1992)  | Italiano, 14, 6 – Matematica, 14, 6 - Inglese, 20, 5 – Informatica, 10, 6 |
| Tiro Giancarlo<br>(21/5/93) | Italiano, 65, 4 – Matematica, 35, 4 - Inglese, 42, 5 – Informatica, 15, 5 |
| Tesore Angela<br>(21/12/91) | Italiano, 12, 7 – Matematica, 12, 6 - Inglese, 12, 5 – Informatica, 25, 5 |
| Piano Guido<br>(5/5/91)     | Italiano, 10, 6 – Matematica, 11, 6 - Inglese, 20, 6 – Informatica, 30, 5 |
| Maturi Nando<br>(23/3/93)   | Italiano, 31, 7 – Matematica, 23, 6 - Inglese, 10, 6 – Informatica, 21, 8 |
| Citti Alassia<br>(4/6/94)   | Italiano, 15, 8 – Matematica, 21, 7 - Inglese, 15, 9 – Informatica, 11, 9 |

- Produrre le seguenti query:
  1. Cognome e nome degli studenti in ordine alfabetico.
  2. Elenco degli studenti il cui cognome inizia con la lettera T.
  3. Elenco degli studenti il cui nome contiene la lettera N.
  4. Elenco degli studenti nati nel 1993.
  5. Elenco delle materie in ordine alfabetico.
  6. Elenco delle valutazioni insufficienti (matricola dello studente, materia, voto) in ordine alfabetico di materia e di studente.
  - 6 bis. Elenco delle valutazioni insufficienti (cognome e nome dello studente, materia, voto) in ordine alfabetico di materia e di studente.
- Eseguire poi le seguenti operazioni:
  7. Modificare il nome della materia "Inglese" in "Lingua Inglese", senza perdere le valutazioni nella materia stessa.
  8. Modificare il nome di Maturi in "Fernando" senza perdere le sue valutazioni.
- Chiudere il DB e crearne una copia chiamandola db15Test. Aprire la copia così creata ed eseguire le seguenti operazioni:
  9. Cancellare la materia Italiano e tutte le valutazioni correlate.
  10. Cancellare uno studente scelto dall'utente e tutte le sue valutazioni.

## SCHEMA CONCETTUALE



## SCHEMA LOGICO

tblStudenti(Matricola, Cognome, Nome, DataNascita) ←  
tblMaterie(NomeMateria) <←  
tblValutazioni(Matricola, Materia, OreAssenza, Voto)

## CREAZIONE DELLE TABELLE

```
CREATE TABLE tblStudenti
(
    Matricola TEXT(7) NOT NULL PRIMARY KEY,
    Cognome TEXT(30) NOT NULL,
    nome TEXT(30) NOT NULL,
    DataNascita DATE NOT NULL,

    UNIQUE (Cognome, Nome, DataNascita),
    CONSTRAINT 'La data di nascita deve essere maggiore del 1950'
    CHECK (DataNascita >=#01/01/1950#)
);
```

```
CREATE TABLE tblMaterie
(
    NomeMateria TEXT(20) NOT NULL PRIMARY KEY
);
```

```
CREATE TABLE tblValutazioni
(
    Matricola TEXT(7) NOT NULL REFERENCES tblStudenti
    ON UPDATE CASCADE ON DELETE CASCADE,
    Materia TEXT(20) NOT NULL REFERENCES tblMaterie
    ON UPDATE CASCADE ON DELETE CASCADE,
```



```
OreAssenze BYTE,
Voto        BYTE,
PRIMARY KEY (Matricola, Materia),
CONSTRAINT 'Voto tra 1 e 10' CHECK (Voto BETWEEN 1 AND 10)
);
```

## INSERIMENTO DEI DATI NELLE TABELLE

```
INSERT INTO tblMaterie
VALUES ([Nome della materia]);
```

```
INSERT INTO tblStudenti
VALUES ([Matricola], [Cognome], [Nome], [DataNascita]);
```

```
INSERT INTO tblValutazioni
VALUES ([Matricola], [Materia], [Ore di assenza], [Voto]);
```

| tblMaterie |             |
|------------|-------------|
|            | NomeMateria |
| +          | Informatica |
| +          | Inglese     |
| +          | Italiano    |
| +          | Matematica  |

| tblStudenti |           |         |           |             |
|-------------|-----------|---------|-----------|-------------|
|             | Matricola | Cognome | nome      | DataNascita |
| +           | 00001     | Astori  | Guido     | 12/02/1992  |
| +           | 00002     | Seccia  | Vittoria  | 04/06/1990  |
| +           | 00003     | Dani    | Alberto   | 23/03/1991  |
| +           | 00004     | Soleno  | Assuntina | 12/12/1992  |
| +           | 00005     | Ficco   | Andrea    | 05/05/1992  |
| +           | 00006     | Tiro    | Giancarlo | 21/05/1993  |
| +           | 00007     | Tesore  | Angela    | 21/12/1991  |
| +           | 00008     | Piano   | Guido     | 05/05/1991  |
| +           | 00009     | Maturi  | Nando     | 23/03/1993  |
| +           | 00100     | Citti   | Alessia   | 04/06/1994  |

| tblValutazioni |             |            |      |  |
|----------------|-------------|------------|------|--|
| Matricola      | Materia     | OreAssenze | Voto |  |
| 00006          | Italiano    | 65         | 4    |  |
| 00006          | Matematica  | 35         | 4    |  |
| 00007          | Informatica | 25         | 5    |  |
| 00008          | Informatica | 30         | 5    |  |
| 00006          | Informatica | 15         | 5    |  |
| 00005          | Inglese     | 20         | 5    |  |
| 00002          | Inglese     | 15         | 5    |  |
| 00007          | Inglese     | 12         | 5    |  |
| 00006          | Inglese     | 42         | 5    |  |
| 00002          | Matematica  | 21         | 5    |  |
| 00001          | Informatica | 11         | 6    |  |
| 00005          | Informatica | 10         | 6    |  |
| 00002          | Informatica | 21         | 6    |  |
| 00009          | Inglese     | 10         | 6    |  |
| 00003          | Inglese     | 13         | 6    |  |
| 00008          | Inglese     | 20         | 6    |  |
| 00005          | Italiano    | 14         | 6    |  |
| 00002          | Italiano    | 25         | 6    |  |
| 00008          | Italiano    | 10         | 6    |  |
| 00007          | Matematica  | 12         | 6    |  |
| 00008          | Matematica  | 11         | 6    |  |
| 00001          | Matematica  | 15         | 6    |  |
| 00009          | Matematica  | 23         | 6    |  |
| 00005          | Matematica  | 14         | 6    |  |
| 00003          | Informatica | 31         | 7    |  |
| 00004          | Informatica | 10         | 7    |  |
| 00001          | Inglese     | 22         | 7    |  |
| 00003          | Italiano    | 30         | 7    |  |
| 00001          | Italiano    | 45         | 7    |  |
| 00009          | Italiano    | 31         | 7    |  |
| 00007          | Italiano    | 12         | 7    |  |
| 00100          | Matematica  | 21         | 7    |  |
| 00003          | Matematica  | 33         | 7    |  |
| 00009          | Informatica | 21         | 8    |  |
| 00004          | Inglese     | 15         | 8    |  |
| 00004          | Italiano    | 12         | 8    |  |
| 00100          | Italiano    | 15         | 8    |  |
| 00100          | Informatica | 11         | 9    |  |
| 00100          | Inglese     | 15         | 9    |  |
| 00004          | Matematica  | 12         | 9    |  |

## QUERY SQL

### 1. Cognome e nome degli studenti in ordine alfabetico

```
SELECT Cognome, Nome  
FROM tblStudenti  
ORDER BY 1,2;
```

### 2. SELECT Cognome, Nome

```
FROM tblStudenti  
WHERE Cognome LIKE 'T%'  
ORDER BY 1, 2;
```

### 3. Elenco degli studenti il cui nome contiene la lettera N

```
SELECT Cognome, Nome  
FROM tblStudenti  
WHERE Cognome LIKE '%n%'  
ORDER BY 1, 2;
```

### 4. Elenco degli studenti nati nel 1993

```
SELECT *  
FROM tblStudenti  
WHERE DataNascita Between #1/1/1993# And #12/31/1993#;
```

### 5. Elenco delle materie in ordine alfabetico

```
SELECT NomeMateria AS Materia  
FROM tblMaterie  
ORDER BY NomeMateria;
```

### 6. Elenco delle valutazioni insufficienti (matricola dello studente, materia, voto) in ordine alfabetico di materia e di studente

```
SELECT *  
FROM tblValutazioni  
WHERE Voto<6  
ORDER BY Materia, Matricola;
```

### 6 bis. Elenco delle valutazioni insufficienti (cognome e nome dello studente, materia, voto) in ordine alfabetico di materia e di studente

#### Equi Join

```
SELECT Cognome, Nome, Materia, Voto  
FROM tblStudenti, tblValutazioni  
WHERE tblStudenti.Matricola=tblValutazioni.Matricola And Voto<6  
ORDER BY 3, 1, 2;
```

#### Inner Join

```
SELECT Cognome, Nome, Materia, Voto  
FROM tblStudenti INNER JOIN tblValutazioni  
ON tblStudenti.Matricola=tblValutazioni.Matricola  
WHERE Voto<6  
ORDER BY 3, 1, 2;
```

## UPDATE

**7. Modificare il nome della materia "Inglese" in "Lingua Inglese", senza perdere le valutazioni nella materia stessa.**

```
UPDATE tblMaterie SET Nomemateria = 'Lingua Inglese'  
WHERE Nomemateria='Inglese';
```

**8. Modificare il nome di Maturi in "Fernando" senza perdere le sue valutazioni.**

```
UPDATE tblStudenti SET Nome = 'Fernando'  
WHERE Cognome='Maturi' And Nome='Nando';
```

## DELETE

**9. Cancellare la materia Italiano e tutte le valutazioni correlate.**

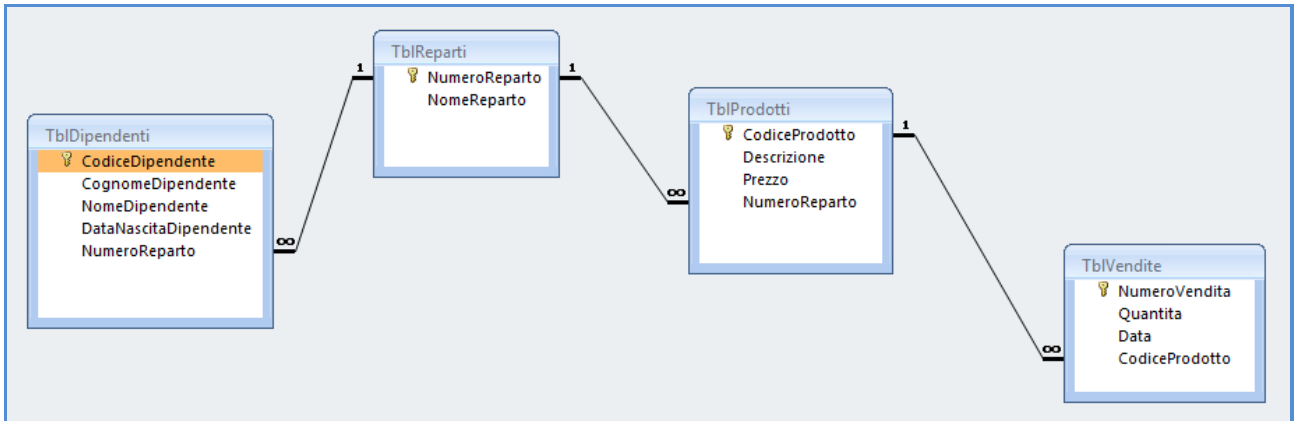
```
DELETE *  
FROM tblMaterie  
WHERE NomeMateria='Italiano';
```

**10. Cancellare uno studente scelto dall'utente e tutte le sue valutazioni.**

```
DELETE *  
FROM tblStudenti  
WHERE Matricola=[Digita la matricola da cancellare:];
```

## Esercizio 16: Supermercato

Si consideri il database:



che contiene dati relativi ad un supermercato.

Realizzare le seguenti interrogazioni:

1. Prezzo medio, minimo e massimo dei prodotti.
2. Numero dei prodotti.
3. Elenco dei prodotti, con descrizione, prezzo e nome del reparto, ordinato per reparto e descrizione.
4. Elenco dei dipendenti, con cognome e nome del dipendente e nome del reparto, ordinato per reparto e cognome e nome.
5. Prezzo medio, minimo e massimo dei prodotti per ogni reparto, in ordine alfabetico dei reparti.
6. Elenco dei prodotti di un reparto di cui si dà il nome, dal più al meno costoso.
7. Vendite di un giorno assegnato, con descrizione, quantità e nome del reparto.
8. Incasso di un giorno.
9. Elenco dei nomi dei reparti con numero di prodotti per reparto.
10. Elenco dei nomi dei reparti con prezzo minimo e massimo per reparto.
11. Elenco dei nomi dei reparti con incasso totale per reparto.
12. Elenco dei prodotti con il numero di pezzi venduti per ciascuno.
13. Elenco dei prodotti per i quali sono stati venduti più di 10 pezzi.
14. Elenco dei prodotti con prezzo superiore ad una cifra specificata.
15. Elenco dei prodotti con prezzo più alto.
16. Elenco dei prodotti con prezzo superiore alla media.

### 1. Prezzo medio, minimo e massimo dei prodotti.

```
SELECT AVG(Prezzo) AS [Prezzo medio],  
       MIN(Prezzo) AS [Prezzo minimo],  
       MAX(Prezzo) AS [Prezzo massimo]  
FROM tblProdotti;
```

### 2. Numero dei prodotti.

```
SELECT Count(*) AS [Numero prodotti]  
FROM tblProdotti;
```

**3. Elenco dei prodotti, con descrizione, prezzo e nome del reparto, ordinato per reparto e descrizione.**

```
SELECT Descrizione, Prezzo, NomeReparto AS Reparto
FROM TblReparti INNER JOIN TblProdotti
    ON TblReparti.NumeroReparto = TblProdotti.NumeroReparto
ORDER BY 3, 1;
```

**4. Elenco dei dipendenti, con cognome e nome del dipendente e nome del reparto, ordinato per reparto e cognome e nome.**

```
SELECT CognomeDipendente AS Cognome,
       NomeDipendente     AS Nome,
       NomeReparto        AS Reparto
FROM TblReparti INNER JOIN TblDipendenti
    ON TblReparti.NumeroReparto = TblDipendenti.NumeroReparto
ORDER BY 3, 1, 2;
```

**5. Prezzo medio, minimo e massimo dei prodotti per ogni reparto, in ordine alfabetico dei reparti.**

```
SELECT NomeReparto AS Reparto,
       AVG(Prezzo) AS [Prezzo medio],
       MIN(Prezzo) AS [Prezzo minimo],
       MAX(Prezzo) AS [Prezzo massimo]
FROM TblReparti INNER JOIN TblProdotti
    ON TblReparti.NumeroReparto = TblProdotti.NumeroReparto
GROUP BY NomeReparto
ORDER BY 1;
```

## Esercizio 17: Veterinario

Uno studio veterinario vuole realizzare un database per registrare i dati di tutti i "pazienti". Di ciascuno dei essi devono essere registrati: il nome, la data di nascita, il tipo (cane, gatto, criceto,...), la razza, il nome del padrone (che può anche possedere più animali). Di ogni visita devono essere registrati gli interventi effettuati (visita generica, vaccino, intervento chirurgico, pulizia, ...), che possono essere più di uno per ogni visita, la data e qualche annotazione (eventuale diagnosi e farmaci prescritti, data di successivo intervento, ...).

Il candidato esamini la situazione sopra descritta, precisi eventuali ipotesi aggiuntive e realizzi:

1. La progettazione concettuale della realtà indicata attraverso la produzione di un modello (ad esempio E/R), indicando le entità, gli attributi di ogni entità e le relazioni.
2. Una traduzione del modello concettuale in modello logico.
3. La definizione delle relazioni della base di dati in linguaggio SQL;
4. Le seguenti interrogazioni espresse in SQL:
  - a) elenco di tutti gli animali pazienti, con nome, tipo, razza e proprietario;
  - b) elenco di tutti i cani;
  - c) elenco di tutti i proprietari di animali;
  - d) elenco di tutti i proprietari di più di un animale;
  - e) elenco di tutte le visite effettuate su un animale.

Il veterinario vuole inoltre realizzare un sito Internet per il proprio studio.

Il candidato progetti e realizzi almeno uno dei seguenti moduli:

- la home page del sito;
- la pagina per accedere all'area riservata del sito, con un modulo in cui l'utente deve inserire il suo nome utente e la sua password e cliccare su un pulsante per accedere.

### Svolgimento

#### ANALISI DELLA REALTÀ DI INTERESSE

La realtà di interesse è uno studio veterinario.

L'archivio è chiaramente storico, dal momento che è necessario mantenere i dati di tutte le visite effettuate.

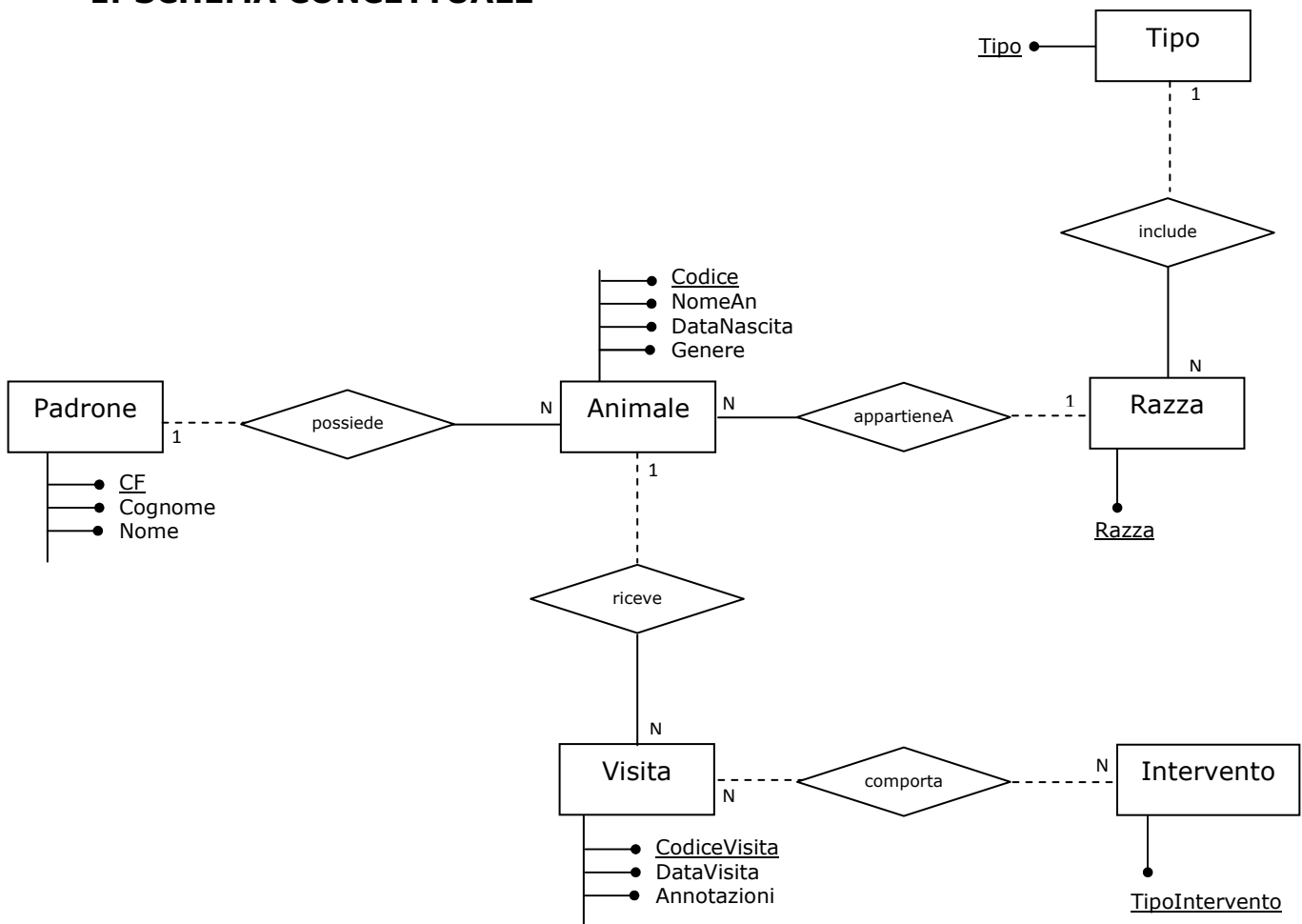
Si individuano come entità: Padrone, Animale, Razza, Tipo, Visita, Intervento.

Del padrone dell'animale si ritiene opportuno registrare solo cognome, nome e codice fiscale 8che lo identifica univocamente).

Ad ogni animale si ritiene di dover attribuire un codice (che lo identifichi univocamente). Avrà poi un nome, una data di nascita e un genere (maschio o femmina). Ogni animale appartiene ad un certo tipo (cane, gatto, ecc.) ed è di una certa razza. È evidente la presenza di relazioni tra le entità Padrone e Animale (1:N), Razza e Animale (1:N), Tipo e Razza (1:N).

Ad ogni visita si attribuisce un codice, ad esempio un numero d'ordine. Si registrano poi data e annotazioni. Intervento è una delle possibili operazioni che vengono fatte sull'animale in occasione della visita. C'è un'associazione tre Animale e Visita (1:N) e tra Visita e Intervento (1:N).

## 1. SCHEMA CONCETTUALE



## 2. SCHEMA LOGICO

tblPadroni(CF, Cognome, Nome) ←  
 tblTipi(Tipo) ←  
 tblRazze(Razza, Tipo) ←  
 tblAnimali(Codice, Nome, DataNascita, Genere, Razza, Padrone) ←  
 tblVisite(CodiceVisita, DataVisita, Annotazioni, Animale) ←  
 tblInterventi(TipoIntervento) ←  
 tblVisiteInterventi(CodiceVisita, Intervento) ←

VINCOLI ESPLICITI

V1(tblAnimali): Genere in ('M','F')



### SCHEMA FISICO

| Tabella                    | Chiave | Nome campo     | Tipo     | Dimensione | Richiesto | Valido se     | Referenzia |
|----------------------------|--------|----------------|----------|------------|-----------|---------------|------------|
| <b>tblPadroni</b>          | PK     | CF             | Testo    | 16         | Sì        |               |            |
|                            |        | Cognome        | Testo    | 20         | Sì        |               |            |
|                            |        | Nome           | Testo    | 20         | Sì        |               |            |
| <b>tblTipi</b>             | PK     | Tipo           | Testo    | 15         | Sì        |               |            |
| <b>tblRazze</b>            | PK     | Razza          | Testo    | 20         | Sì        |               |            |
|                            | FK     | Tipo           | Testo    | 15         | Sì        |               | tblTipi    |
| <b>tblAnimali</b>          | PK     | Codice         | Testo    | 4          | Sì        |               |            |
|                            |        | Nome           | Testo    | 10         | Sì        |               |            |
|                            |        | DataNascita    | Data     |            | Sì        |               |            |
|                            |        | Genere         | Testo    | 1          | Sì        | In ('F', 'M') |            |
|                            | FK     | Razza          | Testo    | 20         | Sì        |               | tblRazze   |
|                            | FK     | Padrone        | Testo    | 16         | Sì        |               | tblPadroni |
| <b>tblVisite</b>           | PK     | CodiceVisita   | Numerico | Intero     | Sì        |               |            |
|                            |        | DataVisita     | Data     |            | Sì        |               |            |
|                            |        | Annotazioni    | Memo     |            |           |               |            |
|                            | FK     | Animale        | Testo    | 4          | Sì        |               | tblAnimali |
| <b>tblInterventi</b>       | PK     | TipoIntervento | Testo    | 20         | Sì        |               |            |
| <b>tblVisiteInterventi</b> | PK, FK | CodiceVisita   | Numerico | Intero     | Sì        |               |            |
|                            | PK, FK | Intervento     | Testo    | 20         | Sì        |               |            |

### 3. Definizione delle relazioni in linguaggio SQL

```
CREATE TABLE tblPadroni
```

```
(
  CF          CHAR(16) NOT NULL PRIMARY KEY,
  Cognome     TEXT(20)  NOT NULL,
  Nome        TEXT(20)  NOT NULL
);
```

```
CREATE TABLE tblTipi
```

```
(
  Tipo        Text(15)  NOT NULL   PRIMARY KEY
);
```

```

CREATE TABLE tblRazze
(
  Razza      Text(20)  NOT NULL  PRIMARY KEY,
  Tipo       Text(15)  NOT NULL  REFERENCES tblTipi
                                     ON UPDATE CASCADE
                                     ON DELETE CASCADE
);

```

```

CREATE TABLE tblAnimali
(
  Codice      CHAR(4)   NOT NULL  PRIMARY KEY,
  Nome        TEXT(10)  NOT NULL,
  DataNascita DATE       NOT NULL,
  Genere      TEXT(1)   NOT NULL,
  Razza       TEXT(20)  NOT NULL  REFERENCES tblRazze
                                     ON UPDATE CASCADE
                                     ON DELETE CASCADE,
  Padrone     CHAR(16)  NOT NULL  REFERENCES tblPadroni
                                     ON UPDATE CASCADE
                                     ON DELETE CASCADE,
  CONSTRAINT 'maschio M, femmina F' CHECK (Genere = 'M' OR Genere = 'F')
);

```

```

CREATE TABLE tblVisite
(
  CodiceVisita Integer  NOT NULL  PRIMARY KEY,
  DataVisita   DATE     NOT NULL,
  Annotazioni  Memo,
  Animale      TEXT(4)   NOT NULL  REFERENCES tblAnimali
                                     ON UPDATE CASCADE
                                     ON DELETE CASCADE
);

```

```

CREATE TABLE tblInterventi
(
  TipoIntervento Text(20) NOT NULL  PRIMARY KEY
);

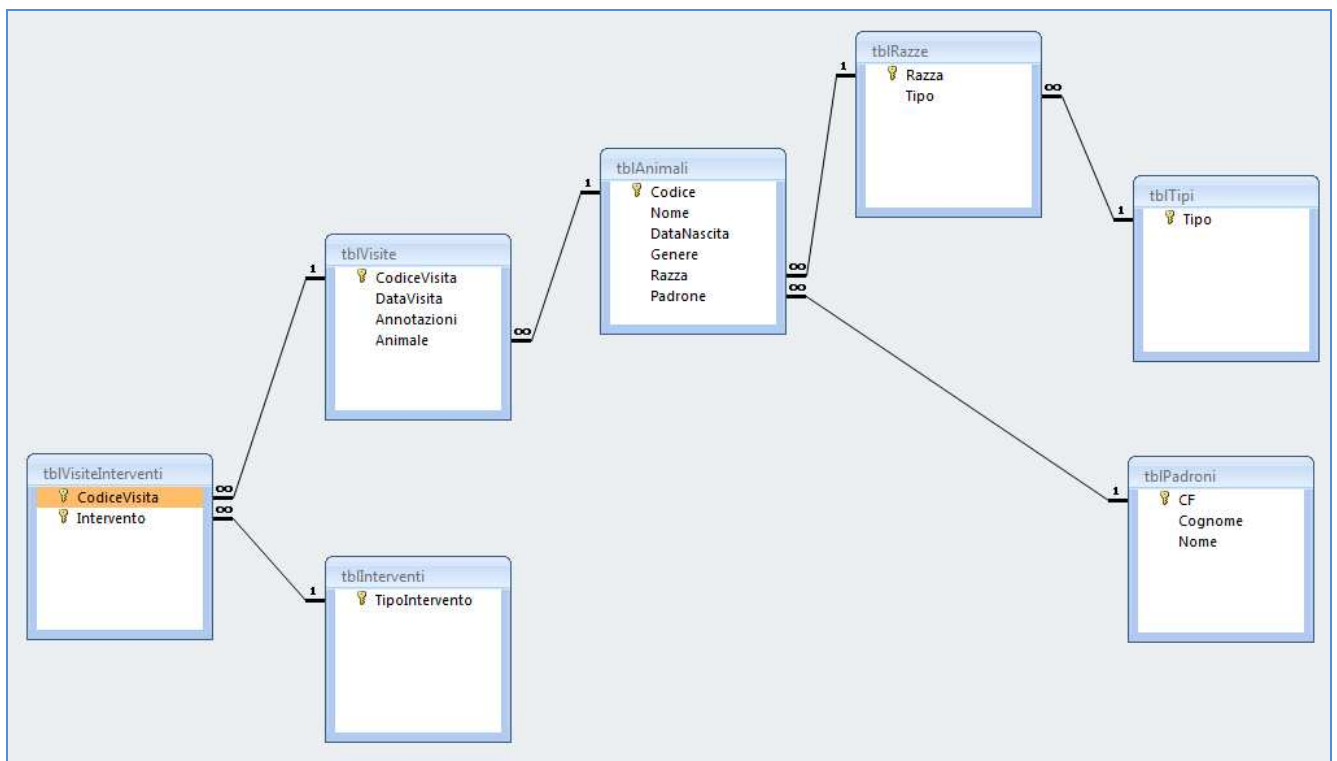
```

```

CREATE TABLE tblVisiteInterventi
(
  CodiceVisita Integer  NOT NULL  REFERENCES tblVisite
                                     ON UPDATE CASCADE
                                     ON DELETE CASCADE,
  Intervento   Text(20)  NOT NULL  REFERENCES tblInterventi
                                     ON UPDATE CASCADE
                                     ON DELETE CASCADE,
  PRIMARY KEY (CodiceVisita, Intervento)
);

```

## SCHEMA RELAZIONI



## 4. Query SQL

a) elenco di tutti gli animali pazienti, con nome, tipo, razza e proprietario;

```
SELECT Codice, Nome AS [Nome animale], A.Razza, Tipo, Cognome, Nome, Padrone AS  
[Codice Fiscale Proprietario]  
FROM (tblAnimali A INNER JOIN tblRazze R ON A.Razza = R.Razza)  
INNER JOIN tblPadroni P ON A.Padrone = P.CF  
ORDER BY Codice;
```

b) elenco di tutti i cani;

```
SELECT Codice, Nome AS [Nome], Data Nascita AS [Data di nascita], Genere, A.Razza  
FROM tblAnimali A INNER JOIN tblRazze R ON A.Razza = R.Razza  
ORDER BY Codice;
```

c) elenco di tutti i proprietari di animali;

```
SELECT Cognome, Nome, CF AS [Codice Fiscale]  
FROM tblAnimali A INNER JOIN tblPadroni P ON A.Padrone = P.CF  
ORDER BY Codice;
```

d) elenco di tutti i proprietari di più di un animale;

```
SELECT Cognome, Nome, CF AS [Codice Fiscale], COUNT (*) AS [Numero di animali]
FROM tblAnimali A INNER JOIN tblPadroni P ON A.Padrone = P.CF
GROUP BY Cognome, Nome, CF
HAVING COUNT (*) > 1
ORDER BY Codice;
```

e) elenco di tutte le visite effettuate su un animale;

```
SELECT CodiceVisita, DataVisita, Intervento, Annotazioni
FROM tblVisite V INNER JOIN tblVisiteInterventi VI ON V.CodiceVisita = VI.CodiceVisita
WHERE Animale = [Digitare il codice del paziente:]
ORDER BY 2, 1, 3;
```

## **Esercizi di progettazione logica relazionale di una base di dati**

È riportato di seguito un insieme di esercizi di progettazione logica relazione di basi di dati.

Ogni esercizio è caratterizzato dallo schema E-R di una base di dati. Per ogni esercizio si richiede di definire uno schema logico relazionale normalizzato per il diagramma E-R proposto.

**Esercizio 1.****Vendita all'ingrosso di piante**

La Figura 1 mostra il diagramma E-R relativo allo schema concettuale di una base dati per la gestione della vendita all'ingrosso di piante.

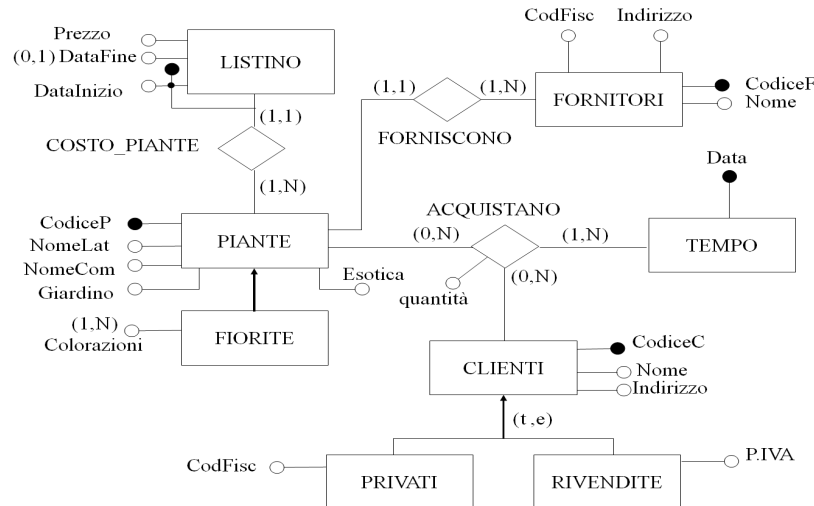


Figure 1: Diagramma ER per la gestione della vendita all'ingrosso di piante

1. Definire uno schema logico relazionale normalizzato per il diagramma E-R proposto.

**Soluzione**

Uno schema logico relazionale normalizzato per lo schema E-R proposto è il seguente:

**FORNITORI**(CodiceF, CodFisc, Nome, Indirizzo)

La generalizzazione sulle piante può essere tradotta eliminando la entità figlia **FIORITE**. Si introduce l'attributo "TipoPianta" per discriminare tra le piante fiorite e non fiorite.

**PIANTE**(CodiceP, TipoPianta, NomeLat, NomeCom, Giardino, Esotica, CodiceF)  
**COLORAZIONI**(CodicePFiorita, Colore)  
**LISTINO**(CodiceP, DataInizio, DataFine\*, Prezzo)

La generalizzazione sui clienti può essere tradotta eliminando le entità figlie **PRIVATI** e **RIVENDITE**. Si introduce l'attributo "TipoCliente" per discriminare tra i clienti privati e le rivendite.

**CLIENTI**(CodiceC, TipoCliente, Nome, Indirizzo, CodFisc\*, PIVA\*)  
~~**TEMPO**(Data)~~  
**ACQUISTANO**(CodiceP, CodiceC, Data, Quantità)

la tabella **TEMPO**(Data) può essere eliminata dato che tutti i valori dell'attributo Data sono anche presenti nella tabella **ACQUISTANO**().

## Esercizio 2. Gestione delle piscine comunali di Torino

La Figura 2 mostra il diagramma E-R relativo allo schema concettuale di una base dati per la gestione delle piscine del comune di Torino.

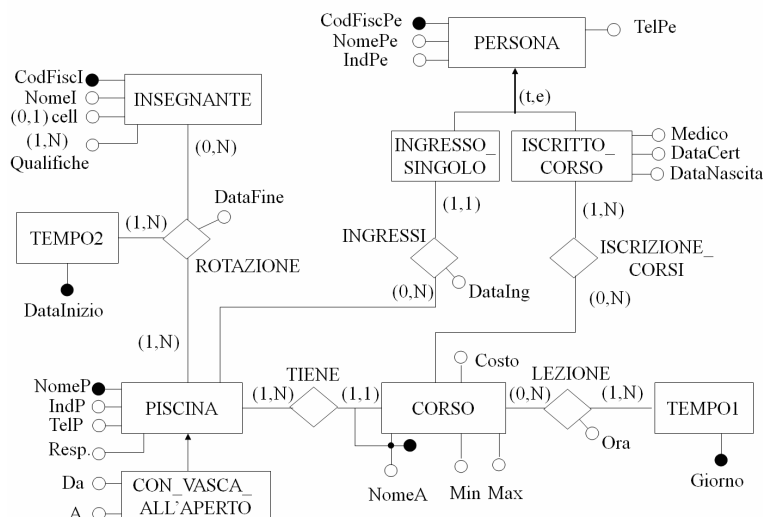


Figure 2: Diagramma ER per la gestione delle piscine gestite dal comune di Torino

1. Definire uno schema logico relazionale normalizzato per il diagramma E-R proposto.

### Soluzione

Uno schema logico relazionale normalizzato per lo schema E-R proposto è il seguente.

La generalizzazione sulle piscine è stata tradotta togliendo l'entità figlia CON\_VASCA\_ALL'APERTO e mantenendo l'entità padre PISCINA. Si è introdotto l'attributo "Tipo" per discriminare tra le piscine con e senza vasca all'aperto.

```
PISCINA(NomeP, IndP, TelP, Resp., Tipo, Da*, A*)
CORSO(NomeP, NomeA, Min, Max, Costo)
TEMPO1(Giorno)
LEZIONE(NomeP, NomeA, Giorno, Ora)
```

La generalizzazione sulle persone è stata tradotta mantenendo sia l'entità padre PERSONA che le due entità figlie ISCRITTO\_CORSO e INGRESSO\_SINGOLO:

```
PERSONA(CodFiscPe, NomePe, IndPe, TelPe)
ISCRITTO_CORSO(CodFiscPeIscrittoCorso, Medico, DataCert, DataNascita)
INGRESSO_SINGOLO(CodFiscPeIngressoSingolo, NomeP, DataIng)
ISCRIZIONE_CORSI (CodFiscPeIscrittoCorso, NomeP, NomeA)

INSEGNANTE(CodFiscI, NomeI, Cell*)
QUALIFICHE(CodFiscI, QualificaI)

TEMPO2(DataInizio)
ROTAZIONE(CodFiscI, DataInizio, NomeP, DataFine)
```

### Esercizio 3. Noleggio di videocassette

La Figura 3 mostra il diagramma E-R relativo allo schema concettuale di una base dati per la gestione di una catena di centri di servizio per il noleggio delle videocassette.

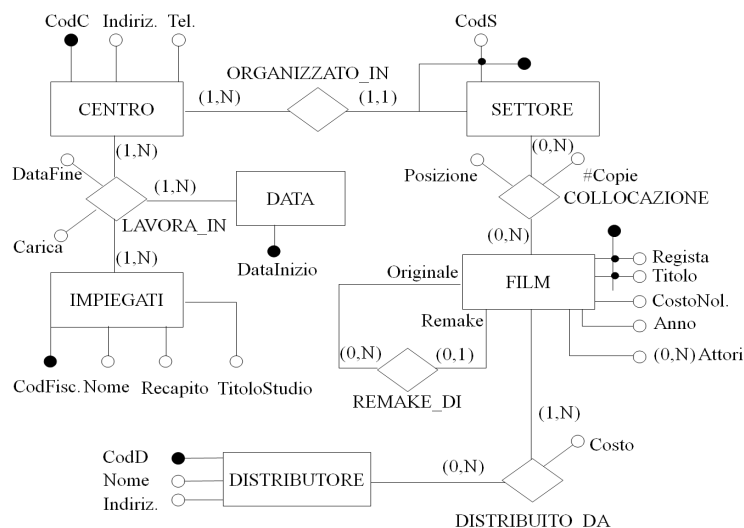


Figure 3: Diagramma ER per la gestione di una catena di centri di servizio per il noleggio delle videocassette

1. Definire uno schema logico relazionale normalizzato per il diagramma E-R proposto.

#### Soluzione

Uno schema logico relazionale normalizzato per lo schema E-R proposto è il seguente:

```

IMPIEGATI(CodFisc, Nome, Recapito, TitoloStudio)
CENTRO(CodC, Indiriz, Tel)
DATA(DataInizio)
LAVORA_IN(CodC, CodFisc, DataInizio, DataFine, Carica)
SETTORE(CodC, CodS)
FILM(Regista, Titolo, CostoNol, Anno)
ATTORI(Regista, Titolo, Attore)
COLLOCAZIONE(CodC, CodS, Regista, Titolo, #Copie, Posizione )
REMAKE_DI(RegistaRemake, TitoloRemake, RegistaFilmOriginale,
TitoloFilmOriginale)
DISTRIBUTORE(CodD, Nome, Indiriz)
DISTRIBUITO_DA(Regista, Titolo, CodD, Costo)

```

La relazione REMAKE\_DI poteva anche essere tradotta come riportato di seguito:

```

FILM(Regista, Titolo, RegistaFilmOriginale*, TitoloFilmOriginale*)

```



### Esercizio 4. Corsi di lingua inglese

La Figura 4 mostra il diagramma E-R relativo allo schema concettuale di una base dati per la gestione dei corsi di lingua inglese, tenuti presso un istituto di insegnamento.

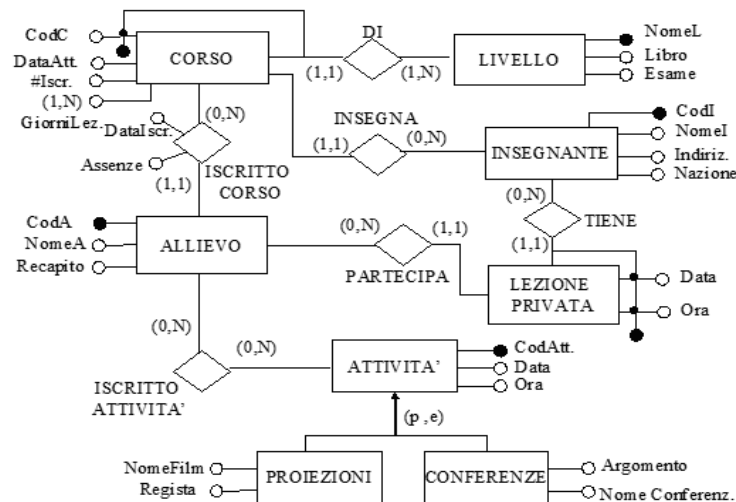


Figure 4: Diagramma ER per la gestione dei corsi di lingua inglese, tenuti presso un istituto di insegnamento

1. Definire uno schema logico relazionale normalizzato per il diagramma E-R proposto.

### Soluzione

Uno schema logico relazionale normalizzato per lo schema E-R proposto è il seguente:

```

LIVELLO(NomeL, Libro, Esame)
INSEGNANTE(CodI, NomeI, Indirizzo, Nazione)
CORSO(NomeL, CodC, DataAtt., #Iscr., CodI)
GIORNI LEZIONE(NomeL, CodC, Giorno)
ALLIEVO(CodA, NomeA, Recapito, NomeL, CodC, DataIscr, Assenze)

LEZIONE PRIVATA(CodI, Data, Ora, CodA)
ATTIVITÀ(CodAtt, Data, Ora)
ISCRITTO ATTIVITÀ(CodA, CodAtt)
PROIEZIONI(CodAttProiez, NomeFilm, Regista)
CONFERENZE(CodAttConf, Argomento, Nome Conferenz)

```

### Esercizio 5. Campionato Europeo Edizione 2000

La Figura 5 mostra il diagramma E-R relativo allo schema concettuale di una base dati per la gestione delle partite di calcio svolte nell'ambito del Campionato Europeo edizione 2000.

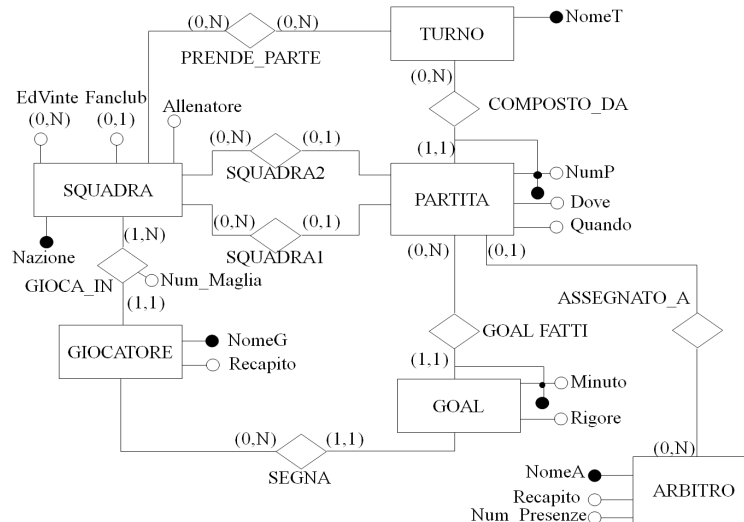


Figure 5: Diagramma ER per la gestione delle partite di calcio svolte durante il Campionato Europeo Edizione 2000

1. Definire uno schema logico relazionale normalizzato per il diagramma E-R proposto.

#### Soluzione

Uno schema logico relazionale normalizzato per lo schema E-R proposto è il seguente:

```

SQUADRA(Nazione, Allenatore, Fanclub*)
EDVINTE(AnnoEdizione, Nazione)
TURNO(NomeT)
PRENDE PARTE(NomeT, Nazione)

GIOCATORE (NomeG, Recapito, Nazione, Num_Maglia)

ARBITRO (NomeA, Recapito, Num_Presenze)
PARTITA (NomeT, NumP, Dove, Quando, NomeA*,
Nazione_Squadra1*, Nazione_Squadra2*)

GOAL(NomeT, NumP, Minuto, NomeG, Rigore)
  
```

## Esercizio 6. Gestione di una farmacia

La Figura 6 mostra il diagramma E-R relativo allo schema concettuale di una base dati per la gestione dei prodotti disponibili in una farmacia.

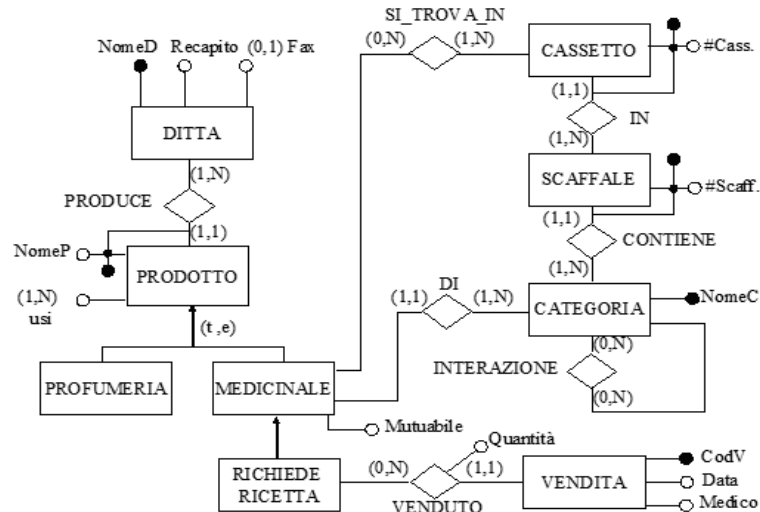


Figure 6: Diagramma ER per la gestione dei prodotti disponibili in una farmacia

1. Definire uno schema logico relazionale normalizzato per il diagramma E-R proposto.

### Soluzione

Uno schema logico relazionale normalizzato per lo schema E-R proposto è il seguente:

**DITTA**(NomeD, Recapito, Fax\*)

Le generalizzazione sul prodotto stata tradotta eliminando l'entità figlia PROFUMERIA e mantenendo l'entità figlia MEDICINALE. L'attributo "TipoProdotto" è un attributo discriminante tra i prodotti di profumeria e gli altri prodotti.

Le generalizzazione sui medicinali è stata tradotta eliminando l'entità figlia RICHIESTE RICETTA. L'attributo "TipoMedicinale" è un attributo discriminante tra i medicinali che richiedono la ricetta e gli altri medicinali.

**PRODOTTO**(NomeD, NomeP, TipoProdotto)

**USI\_DEL\_PRODOTTO**(NomeD, NomeP, Uso)

**MEDICINALE**(NomeD, NomeP, Mutuabile, NomeC, TipoMedicinale)

**VENDITA**(CodV, Data, Medico, NomeD, NomeP, Quantità)

**CATEGORIA**(NomeC)

**INTERAZIONE**(NomeC, NomeCInteragente)

**SCAFFALE**(NomeC, #Scaff)

**CASSETTO**(NomeC, #Scaff, #Cass)

**SI\_TROVA\_IN**(NomeD, NomeP, NomeC, #Scaff, #Cass)

### Esercizio 7. Cartellone dello Stabile

La Figura 7 mostra il diagramma E-R relativo allo schema concettuale di una base dati per la vendita dei biglietti per gli spettacoli del cartellone del teatro Stabile.

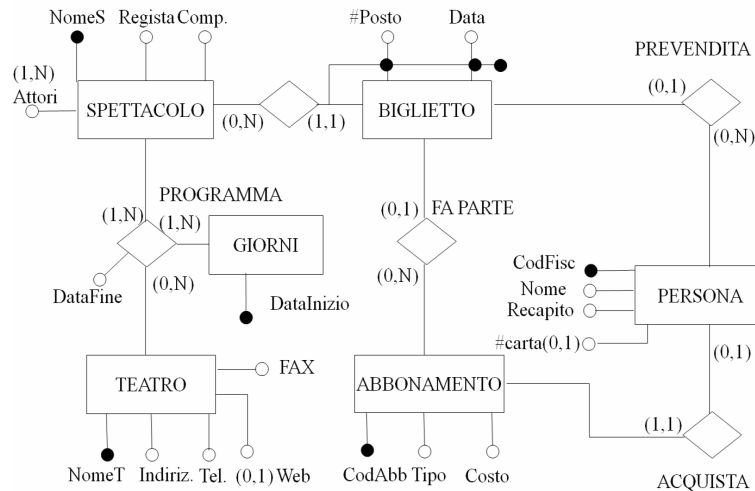


Figure 7: Diagramma ER per la vendita dei biglietti per gli spettacoli del cartellone del teatro Stabile

1. Definire uno schema logico relazionale normalizzato per il diagramma E-R proposto.

#### Soluzione

Uno schema logico relazionale normalizzato per lo schema E-R proposto è il seguente:

```

SPETTACOLO(NomeS, Regista, Comp)
ATTORI(NomeS, NomeAttore)
TEATRO(NomeT, Indirizzo, Tel, Web*, Fax)
GIORNI(DataInizio)
PROGRAMMA(NomeS, NomeT, DataInizio, DataFine)
PERSONA(CodFisc, Nome, Recapito, #carta)
ABBONAMENTO(CodAbb, Tipo, Costo, CodFisc)
BIGLIETTO(NomeS, #Posto, Data, CodAbb*, CodFisc*)

```

Le relazioni FA PARTE e PREVENDITA potevano anche essere tradotte:

```

BIGLIETTO(NomeS, #Posto, Data, CodAbb*)
PREVENDITA(NomeS, #Posto, Data, CodFisc)

```

Questa soluzione è preferibile qualora la maggior parte dei biglietti faccia parte di un abbonamento, mentre un sottoinsieme più ristretto di biglietti venga venduto in prevendita.

Oppure

```

BIGLIETTO(NomeS, #Posto, Data, CodFisc*)
FA PARTE(NomeS, #Posto, Data, CodAbb)

```

Questa soluzione è preferibile qualora la maggior parte dei biglietti venga venduto in prevendita, mentre un sottoinsieme più ristretto di biglietti faccia parte di un abbonamento.

Oppure

```
BIGLIETTO(NomeS, #Posto, Data)  
FA PARTE(NomeS, #Posto, Data, CodAbb)  
PREVENDITA(NomeS, #Posto, Data, CodFisc)
```

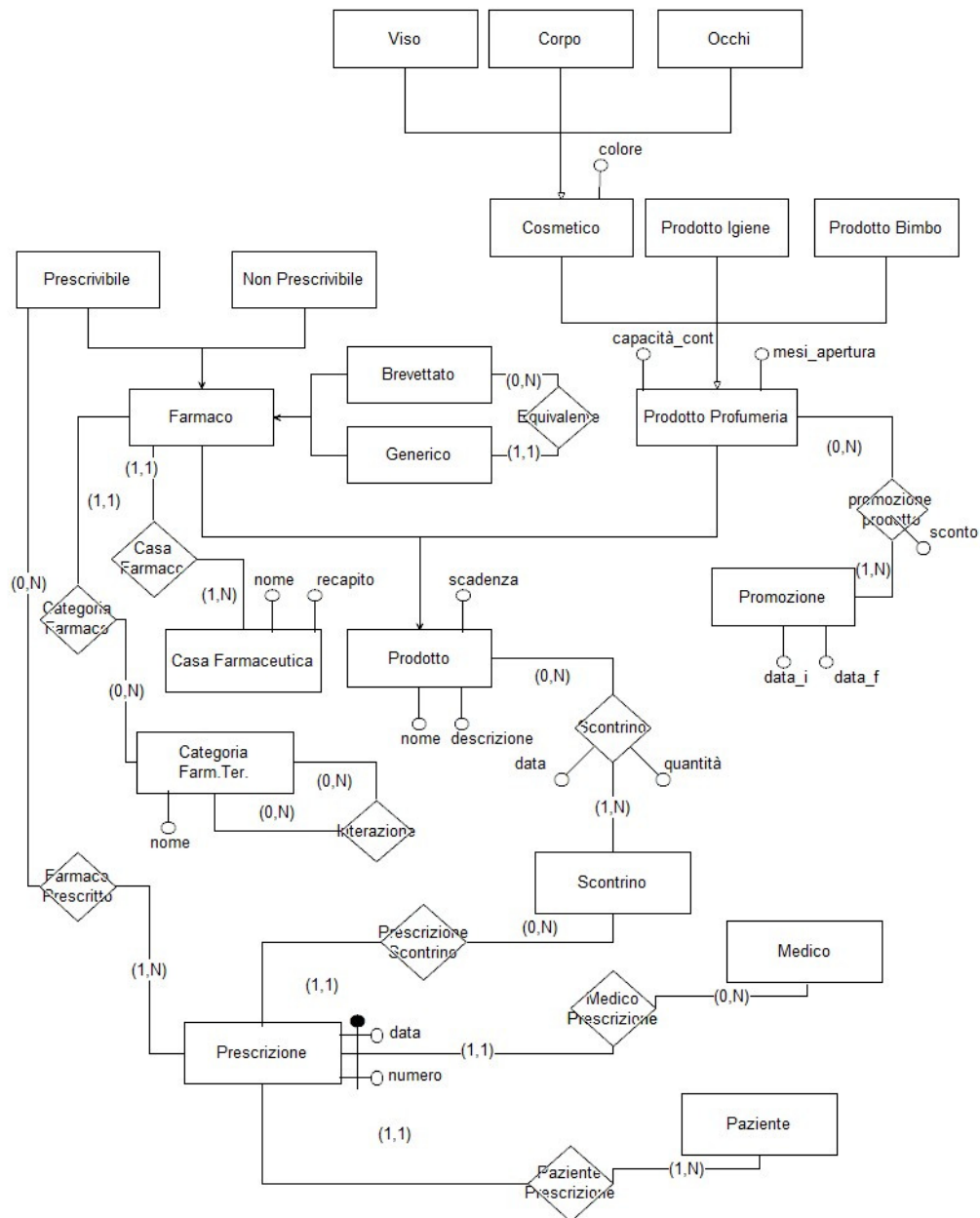
Questa soluzione è preferibile qualora la maggior parte dei biglietti non sia venduta nè in prevendita nè faccia parte di un abbonamento.

## Basi di Dati Relazionali

=====

### Esercizio N° 1

Ristrutturare il seguente schema E-R:



### **Esercizio N° 2**

1. Si utilizzi il modello E-R per rappresentare la seguente classificazione di Persona:

- Le persone sono identificate univocamente dal proprio codice fiscale e sono caratterizzati dal nome, il cognome ed l'età.
- Per gli uomini, si vuole memorizzare anche la posizione militare.
- Si vuole memorizzare anche l'occupazione delle persone, in particolare si vuole sapere se una persona è studente o lavoratore. I lavoratori (o impiegati) hanno lo stipendio e possono essere segretari, direttori o progettisti. In particolare, un progettista può essere anche responsabile di progetto.

Si assume che gli studenti, identificati dal proprio numero di matricola, non possono essere anche impiegati, ma che esistono persone che non sono né impiegati né studenti.

2. Ristrutturare lo schema E-R ottenuto.

### **Esercizio N° 3**

Di seguito, vengono formulate le richieste riguardanti le informazioni relative agli **utenti** di un social network:

- Il social network (SN) prevede due tipi di iscrizioni: il primo tipo riguarda l'iscrizione di una persona al sito e richiede l'inserimento dei dettagli personali quali il nome, il cognome, il sesso, la data e la città di nascita; il secondo è rivolto agli enti (tra i quali attività commerciali, associazioni culturali, gruppi musicali ecc) e richiede la specifica del nome e della tipologia di attività. Opzionalmente gli enti possono specificare la città sede dell'attività e la data di creazione.
- Ogni iscritto, caratterizzato univocamente da un indirizzo e-mail e dotato di una password per l'accesso al sito, crea un profilo sul SN in una certa data. Questo può essere pubblico, cioè visibile a tutti gli utenti del sistema, o privato, cioè visualizzabile solo dai propri amici.
- Per ogni utente, si vuole conoscere la lista di tutti gli utenti amici e la lista di quelli bloccati.
- Un'altra informazione visualizzata sul profilo è il numero totale di amici.

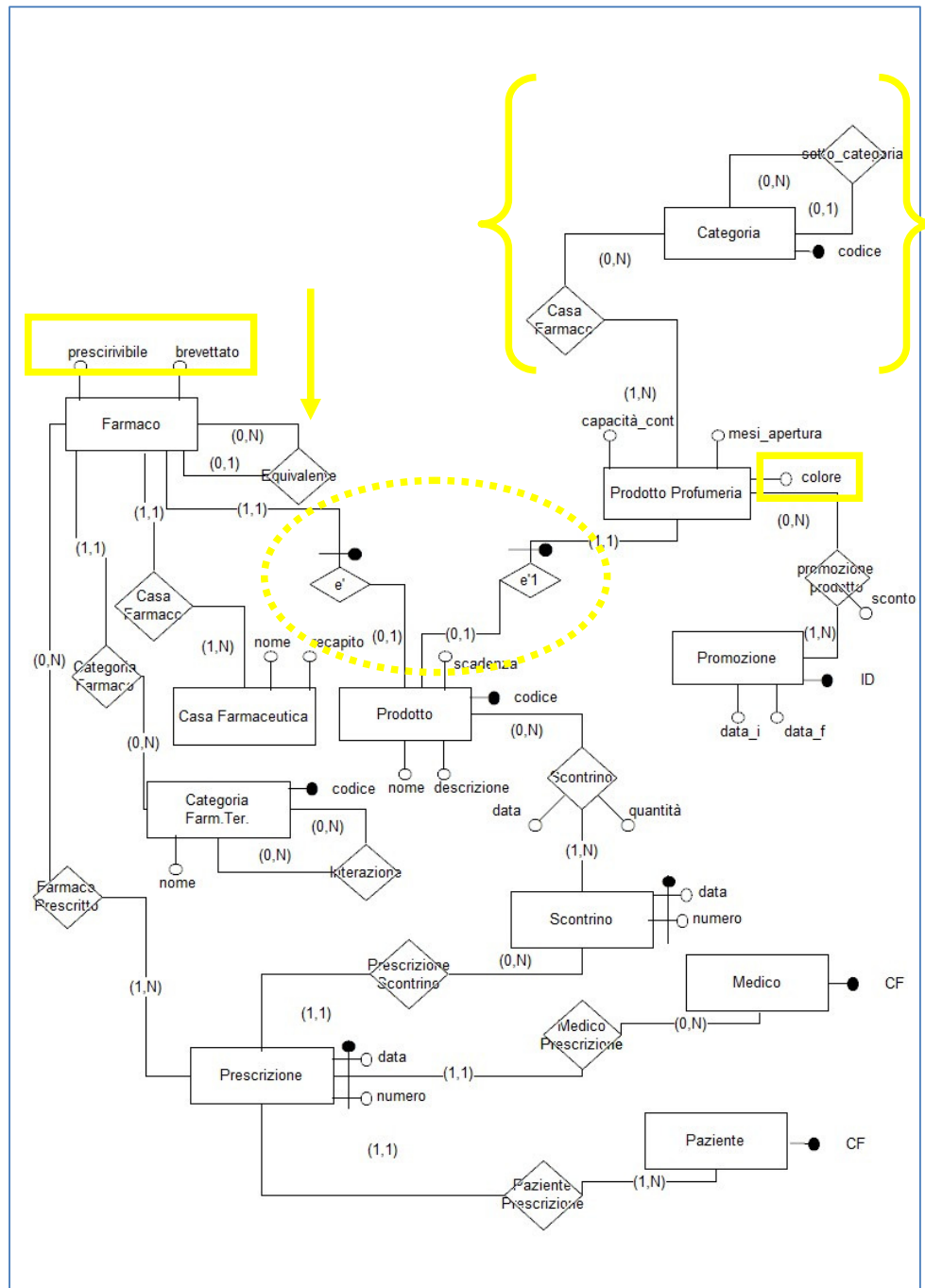
Utilizzare il modello E-R per rappresentare la realtà di interesse e tradurre lo schema ottenuto in uno schema relazionale.

# Basi di Dati Relazionali

Esercitazione – 27 Aprile 2010

## Esercizio N° 1

Di seguito viene mostrata una possibile ristrutturazione dello schema concettuale:





***Esercizio N° 2***

***Esercizio N° 3***

## CAPITOLO 9

### ESERCIZI SULLA PROGETTAZIONE LOGICA

(le soluzioni sono riportate da pag. 6 in poi)

(Nota: gli esercizi non sono sempre in ordine di difficoltà)

#### Esercizio 1

Si prendano tutti gli schemi Entità-Relazione considerati negli esercizi sulla modellazione concettuale pubblicati sulla pagina web del corso. Effettuate le necessarie ristrutturazioni, tradurre ciascuno schema in uno schema relazionale.

#### Esercizio 2

Si consideri lo schema Entità-Relazione in Figura 1. Effettuare le necessarie ristrutturazioni dello schema ed effettuare poi la traduzione verso il modello relazionale.

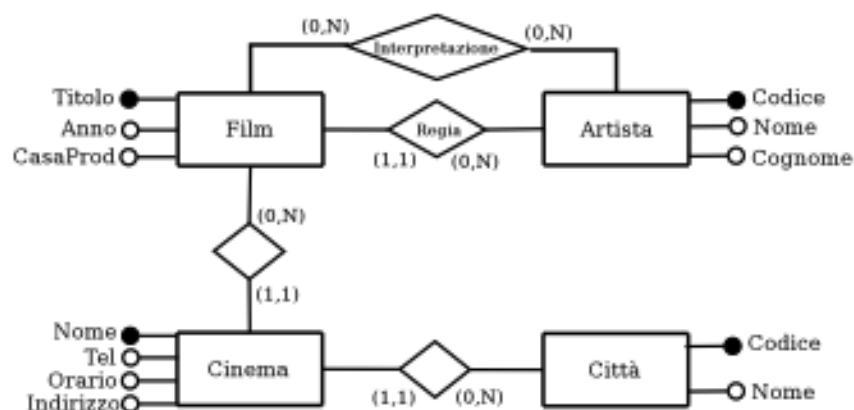


Figura 1: Schema E-R per l'esercizio 2

#### Esercizio 3

Si consideri lo schema Entità-Relazione in Figura 2. Tradurre lo schema in uno schema del modello relazionale (dopo l'eventuale ristrutturazione dello schema ER).

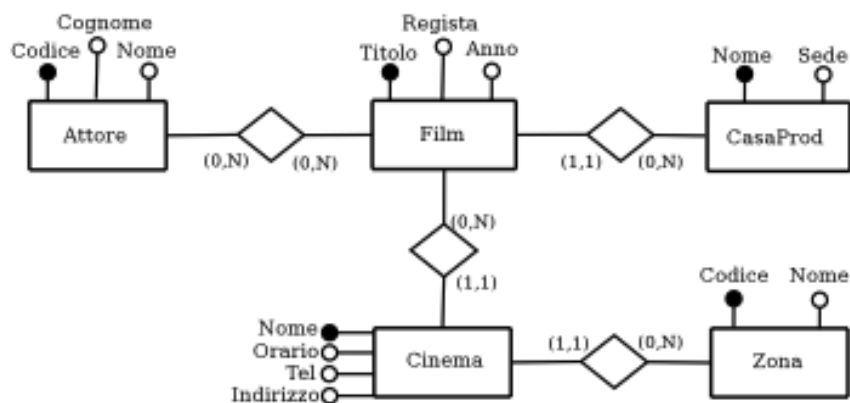


Figura 2: Schema E-R per l'esercizio 3

## Esercizio 4

Si consideri lo schema Entità-Relazione in Figura 3. Tradurre lo schema in uno schema del modello relazionale.

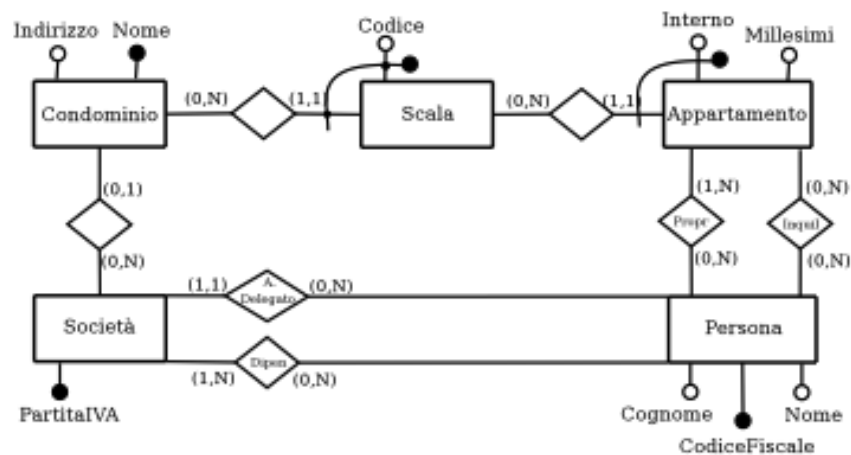


Figura 3: Schema E-R per l'esercizio 4

## Esercizio 5

Si consideri lo schema Entità-Relazione in Figura 4. Tradurre lo schema in uno schema del modello relazionale.

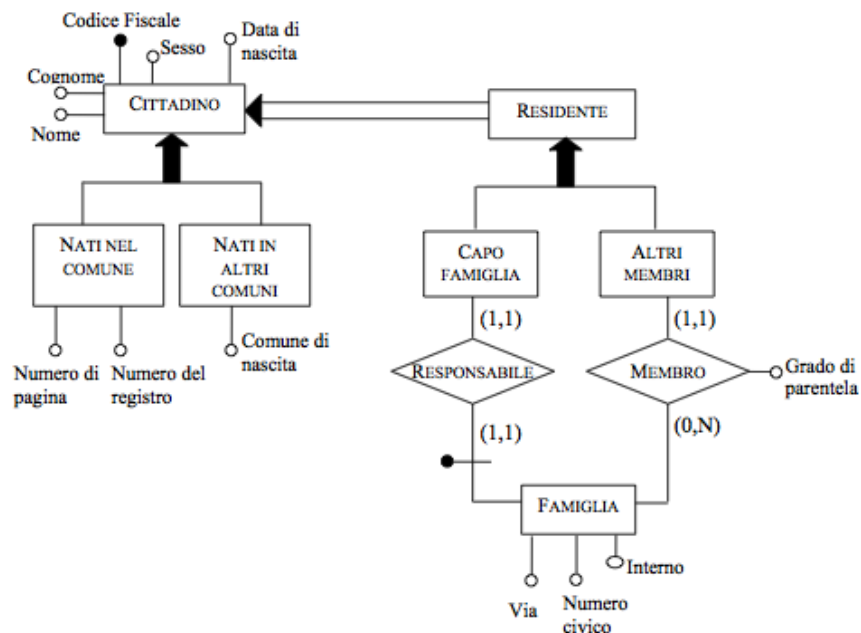


Figura 4: Schema E-R per l'esercizio 5

## Esercizio 6

Si consideri lo schema Entità-Relazione in Figura 5. Tradurre lo schema in uno schema del modello relazionale.

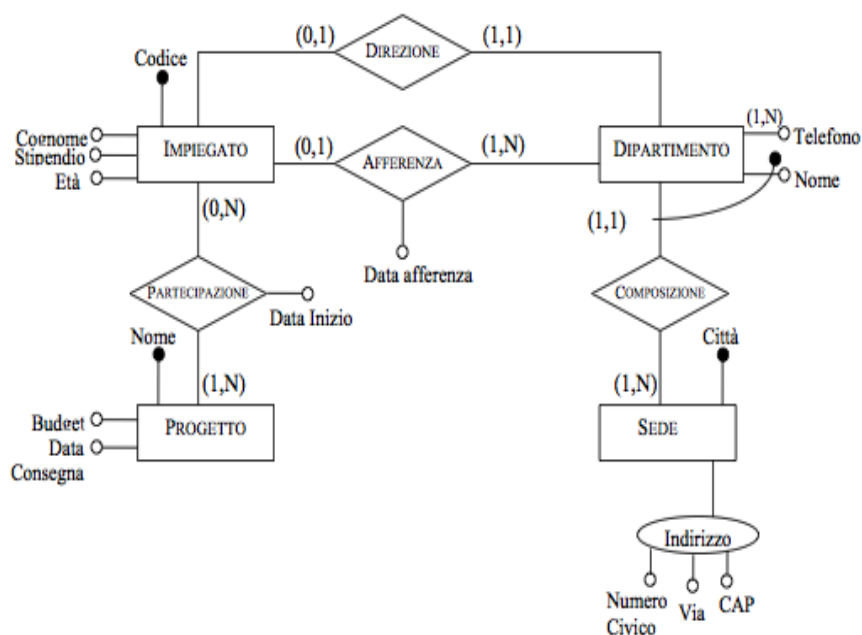


Figura 5: Schema E-R per l'esercizio 6

## Esercizio 7

Si consideri lo schema Entit -Relazione in Figura 6. Tradurre lo schema in uno schema del modello relazionale.

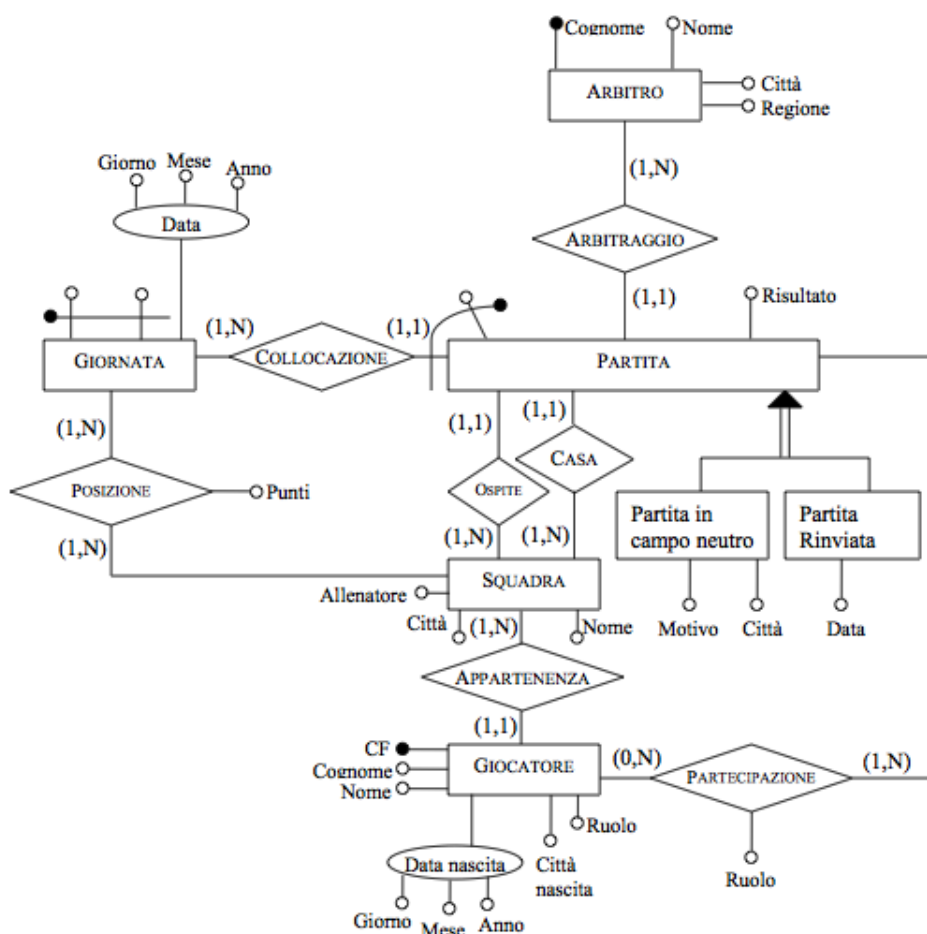


Figura 6: Schema E-R per l'esercizio 7

## Esercizio 8

Si consideri lo schema Entità-Relazione in Figura 7. Tradurre lo schema in uno schema del modello relazionale.

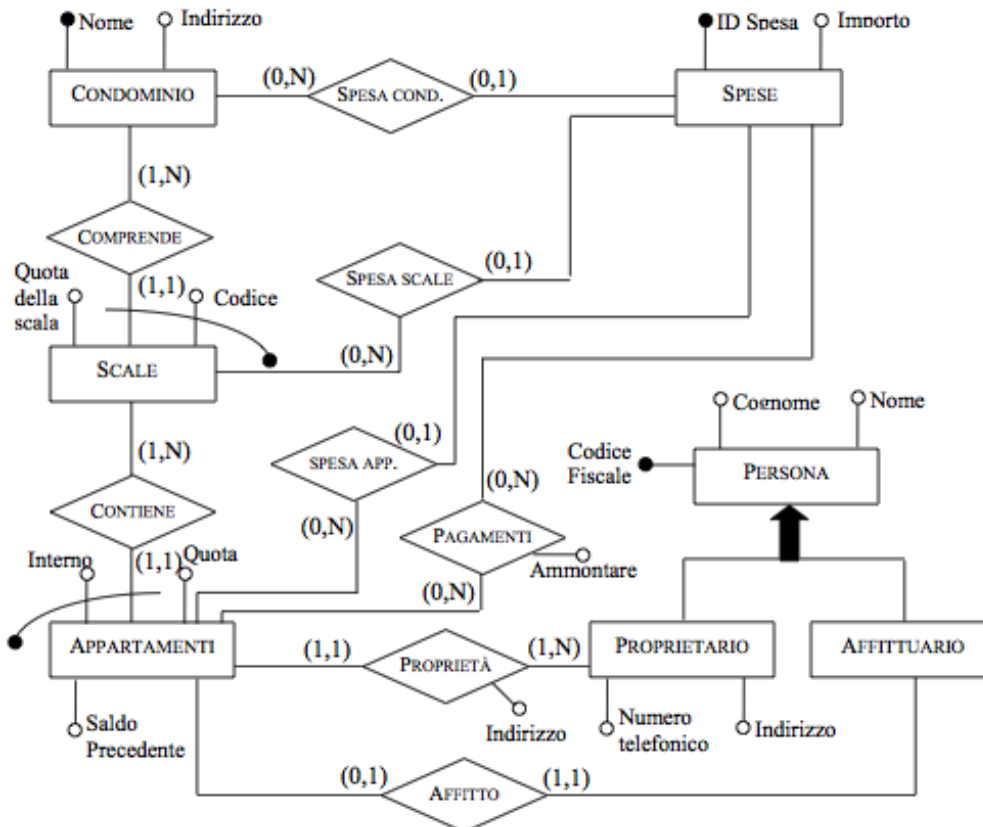


Figura 7: Schema E-R per l'esercizio 8

## Esercizio 9

Tradurre lo schema Entità-Relazione in Figura 8 in uno schema relazionale.

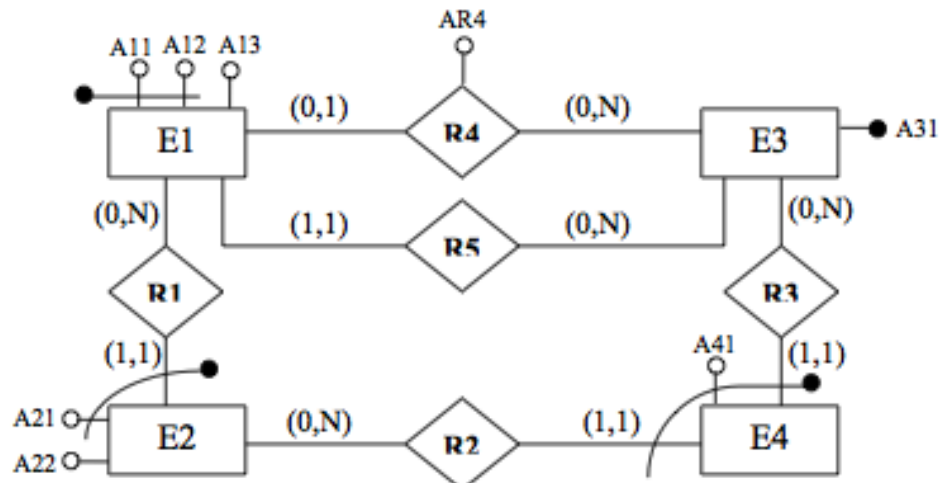


Figura 8: Schema ER per l'esercizio 9

## Esercizio 10

Tradurre lo schema Entità-Relazione in Figura 9 in uno schema relazionale.

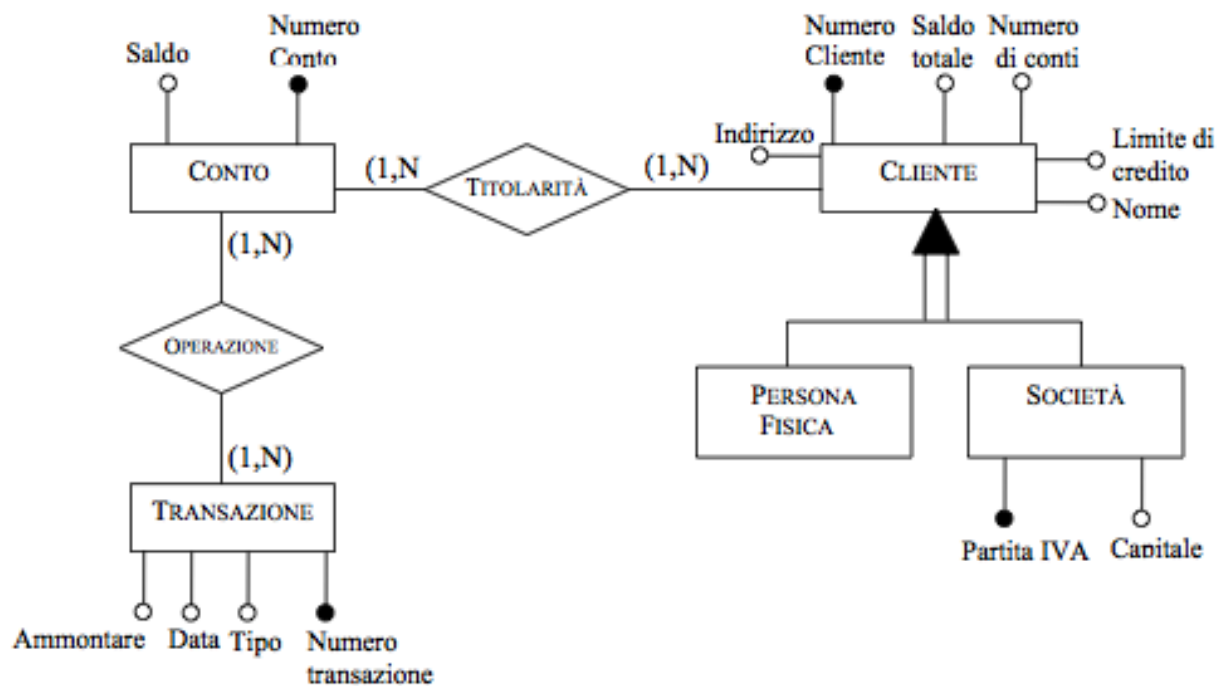


Figura 9: Schema ER per l'esercizio 10

## Esercizio 11

Tradurre lo schema Entità-Relazione in Figura 10 in uno schema relazionale.

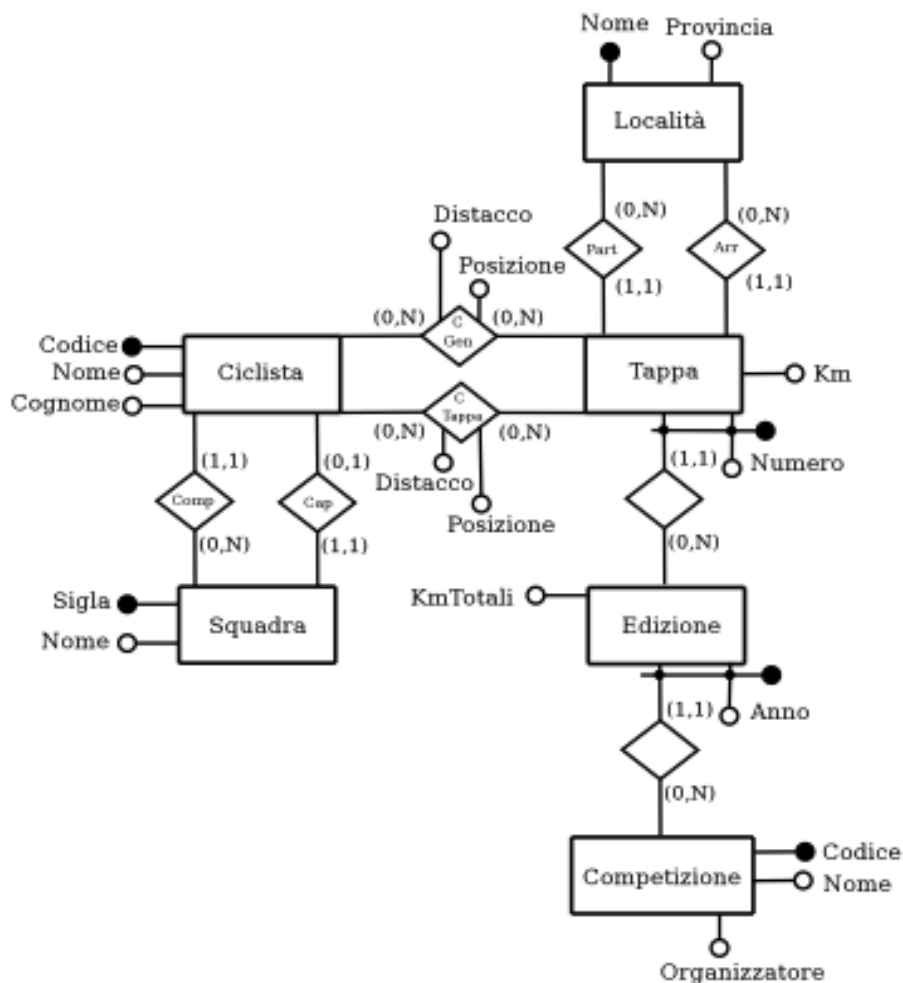


Figura 10: Schema ER per l'esercizio 11

# SOLUZIONI

## Esercizio 2

Schema relazionale:

- ARTISTI (Codice, Nome, Cognome)
- FILM (Titolo, Anno, CasaProduttrice, Artista)  
con vincolo di integrità referenziale fra l'attributo **Artista** e la relazione ARTISTI
- INTERPRETAZIONI (Film, Artista)  
con vincoli di integrità referenziale fra l'attributo **Artista** e la relazione ARTISTI e fra l'attributo **Film** e la relazione FILM
- CITTA (Codice, Nome)
- CINEMA (Nome, Tel, Orario, Indirizzo, Citta, Film)  
con vincoli di integrità referenziale fra l'attributo **Film** e la relazione FILM e fra l'attributo **Citta** e la relazione CITTA

## Esercizio 3

Schema relazionale:

- ATTORI (Codice, Cognome, Nome)
- FILM (Titolo, Regista, Anno)
- ATTORIFILM (Attore, Film)  
con vincoli di integrità referenziale fra l'attributo **Attore** e la relazione ATTORI e fra l'attributo **Film** e la relazione FILM
- CASEPRODUTTRICI (Nome, Sede)
- ZONE (Codice, Nome)
- CINEMA (Nome, Orario, Tel, Indirizzo, Zona, Film)  
con vincoli di integrità referenziale fra l'attributo **Film** e la relazione FILM e fra l'attributo **Zona** e la relazione ZONE.

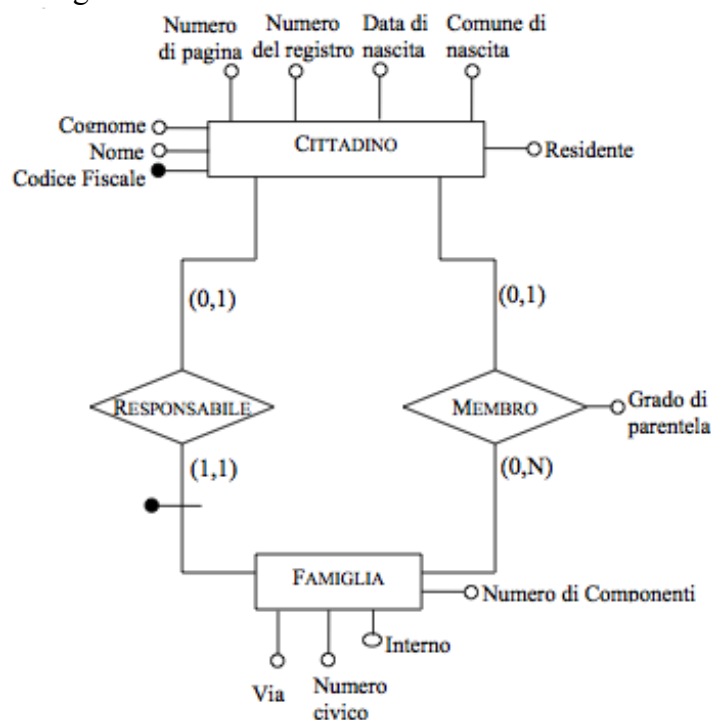
## Esercizio 4

Schema relazionale:

- CONDOMINI (Nome, Indirizzo, PartitalvaSocieta)  
con vincolo di integrità referenziale fra l'attributo **PartitalvaSocieta** e la relazione SOCIETA
- SCALA (Codice, Condominio)  
con vincolo di integrità referenziale fra l'attributo **Condominio** e la relazione CONDOMINIO
- APPARTAMENTO (Interno, CodiceScala, Condominio, Millesimi)  
con vincolo di integrità referenziale fra gli attributi **CodiceScala**, **Condominio** e la relazione CONDOMINI
- PROPRIETA (InternoAppartamento, CodiceScala, Condominio, CodiceFiscalePersona)  
con vincoli di integrità referenziale fra gli attributi **InternoAppartamento**, **CodiceScala**, **Condominio** e la relazione APPARTAMENTO e fra l'attributo **CodiceFiscalePersona** e la relazione PERSONA
- INQUILINO (InternoAppartamento, codiceScala, CodiceFiscalePersona)  
con vincoli di integrità referenziale fra gli attributi **InternoAppartamento**, **CodiceScala**, **Condominio** e la relazione APPARTAMENTO e fra l'attributo **CodiceFiscalePersona** e la relazione PERSONA
- PERSONA (CodiceFiscale, Cognome, Nome)
- SOCIETA (Partitalva, CodiceFiscaleDelegato)  
con vincolo di integrità referenziale fra l'attributo **CodiceFiscaleDelegato** e la relazione PERSONA
- DIPENDENTI (PartitalvaSocieta, CodiceFiscalePersona)  
con vincoli di integrità referenziale fra l'attributo **PartitalvaSocieta** e la relazione SOCIETA` e fra l'attributo **CodiceFiscalePersona** e la relazione PERSONA.

## Esercizio 5

Lo schema ER ristrutturato è il seguente:



Il corrispondente schema relazionale è il seguente:

- CITTADINO(CodiceFiscale, Cognome, Nome, NumeroDiPagina, NumeroDelRegistro, DataDiNascita, ComuneDiNascita, Residente, Famiglia\*, GradoDiParentela)
  - FAMIGLIA(CapoFamiglia, Via, NumeroCivico, Interno, NumeroDiComponenti)
- con vincolo di integrità referenziale tra CapoFamiglia e la relazione CITTADINO

## Esercizio 6

Schema relazionale:

- IMPIEGATO(Codice, Cognome, Stipendio, Eta, Dipartimento, Sede, DataAfferenza)
  - DIPARTIMENTO(Nome, Sede)
  - TELEFONO(Dipartimento, Numero)
  - SEDE(Citta, CAP, Via, NumeroCivico)
  - PROGETTO(Nome, Budget, Data Consegna)
  - PARTECIPAZIONE(Impiegato, Progetto, DataInizio)
- con vincolo di integrità referenziale tra Dipartimento e la relazione DIPARTIMENTO, e tra Sede e la relazione SEDE
- con vincolo di integrità referenziale tra Sede e la relazione SEDE
- con vincolo di integrità referenziale tra Dipartimento e la relazione DIPARTIMENTO
- con vincolo di integrità referenziale tra Impiegato e la relazione IMPIEGATO e tra Progetto e la relazione PROGETTO.

## Esercizio 7

Schema relazionale:

- ARBITRO(Cognome, Nome, Citta, Regione)
  - GIORNATA(Numero, Serie, Giorno, Mese, Anno)
  - SQUADRA(Nome, Citta, Allenatore)
  - GIOCATORE(CodiceFiscale, Cognome, Nome, Ruolo, CittaDiNascita, Squadra)
  - PARTITA(Numero, DNumero, DSerie, Risultato, Arbitro, Casa, Ospite)
  - PARTITAINCAMPONEUTRO(Partita, Numero, Serie, Motivo, Citta)
- con vincolo di integrità referenziale tra Squadra e la relazione SQUADRA.
- con vincoli di integrità referenziale tra DNumero e DSerie e la relazione GIORNATA, tra Arbitro e ARBITRO e tra Casa e Ospite con la relazione SQUADRA.
- con vincoli di integrità referenziale tra Partita, Numero e Serie con la relazione PARTITA.



- PARTITA RINVIATA(Partita, Numero, Serie, Data)  
con vincoli di integrità referenziale tra Partita, Numero e Serie con la relazione PARTITA.
- POSIZIONE(Squadra, Numero, Serie, Punteggio)  
con vincoli di integrità referenziale tra Squadra e la relazione SQUADRA e tra Numero e Serie e la relazione GIORNATA.
- PARTECIPAZIONE(Giocatore, Partita, Numero, Serie, Ruolo)  
con vincoli di integrità referenziale tra Giocatore e la relazione GIOCATORE e tra Partita, Numero, Serie e la relazione PARTITA.

## Esercizio 8

Schema relazionale:

- CONDOMINIO(Nome, Indirizzo)
- SPESE(IDSpesa, Importo)
- PROPRIETARIO(CodiceFiscale, Cognome, Nome, Indirizzo, NumeroTelefonico)
- AFFITTUARIO(CodiceFiscale, Cognome, Nome)
- SCALE(Codice, Nome, QuoteDellaScala)
- APPARTAMENTI(Interno, Codice, Nome, Quota, CFProprietario, CFAffittuario)
- SPESA COND(IDSpesa, Nome)
- SPESA SCALE(IDSpesa, Nome, Codice)
- SPESA APP(IDSpesa, Nome, Codice, Interno)
- PAGAMENTI(IDSpesa, Nome, Codice, Interno, Ammontare)

## Esercizio 9

Schema relazionale:

- E1(A11, A13, A13, A31R4\*, AR4\*, A31R5)
- E3(A31)
- E2(A21, A11, A12, A22)
- E4(A41, A31, A21, A11, A12)

## Esercizio 10

Schema relazionale:

- TRANSAZIONE(NumeroTransazione, Tipo, Data, Ammontare)
- CONTO(NumeroConto, Saldo)
- CLIENTE(NumeroCliente, SaldoTotale, LimiteDiCredito, Nome, Indirizzo, PartitaIVA\*, Capitale\*)
- OPERAZIONE(NumeroConto, NumeroTransazione)
- TITOLARITA(NumeroConto, NumeroCliente)

## Esercizio 11

Schema relazionale:

- SQUADRE (Sigla, Nome, Capitano)  
con vincolo di integrità referenziale fra l'attributo Capitano e la relazione CICLISTI
- CICLISTI (Codice, Cognome, Nome, Squadra)  
con vincolo di integrità referenziale fra l'attributo Squadra e la relazione SQUADRE
- LOCALITA (Nome, Provincia)
- COMPETIZIONI (Codice, Nome, Organizzatore)
- EDIZIONI (AnnoEdizione, Competizione, KmTotali)  
con vincolo di integrità referenziale fra l'attributo Competizione e la relazione COMPETIZIONI
- TAPPE (NumeroTappa, AnnoEdizione, Competizione, LocPartenza, LocArrivo)  
con vincoli di integrità referenziale fra gli attributi Competizione, AnnoEdizione e la relazione EDIZIONI, fra l'attributo LocPartenza e la relazione LOCALITA e fra l'attributo LocArrivo e la relazione LOCALITA
- CLASSIFICATA TAPPA (NumeroTappa, AnnoEdizione, Competizione, Ciclista, Posizione, Distacco)  
con vincoli di integrità referenziale fra gli attributi NumeroTappa, Competizione, AnnoEdizione e la relazione TAPPE e fra Ciclista e la relazione CICLISTA
- CLASSIFICA GENERALE (NumeroTappa, AnnoEdizione, Competizione, Ciclista, Posizione, Distacco)  
con vincoli di integrità referenziale fra gli attributi NumeroTappa, Competizione, AnnoEdizione e la relazione TAPPE e fra Ciclista e la relazione CICLISTA.