

Esercizi progettazione DB – 2011/2012 – 5As

ESERCIZIO 1: ALUNNI	2
ESERCIZIO 2: MEDICI.....	4
ESERCIZIO 3: VARIANTE ALL’ESERCIZIO N°1.....	6
ESERCIZIO 4: ISTITUTO DI VENDITE GIUDIZIARIE	8
ESERCIZIO 5: LISTINO	11
ESERCIZIO 6: FILIALI.....	13
ESERCIZIO 7: FORMULA 1 (COMPITO)	14
ESERCIZIO 8: IMPRESA EDILE (COMPITO)	16
ESERCIZIO 9: MOSTRA CANINA.....	18
ESERCIZIO 10: MULTISALA CINEMATOGRAFICA	22
ESERCIZIO 11: AUTO USATE	26
ESERCIZIO 12: UNIVERSITÀ (COMPITO)	27
ESERCIZIO 13: EURONICS (COMPITO).....	31
ESERCIZIO 14: VENDITORI (NORMALIZZAZIONE).....	33
ESERCIZIO 15: SCRUTINI.....	35
ESERCIZIO 16: VETERINARIO	41

Esercizio 1: Alunni

Progettare una base di dati per memorizzare gli alunni che <frequentano> le varie classi di un Istituto scolastico.

Svolgimento

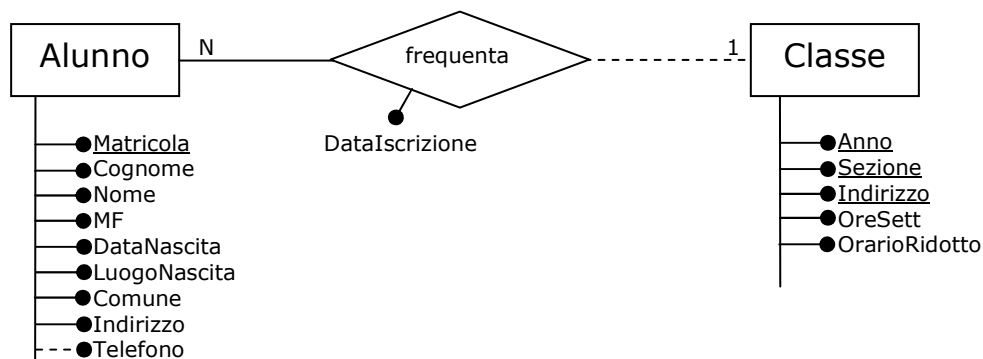
ANALISI DELLA REALTÀ DI INTERESSE

Il progetto è rivolto alla gestione informatizzata degli alunni di una scuola. Si ritengono rilevanti i seguenti dati: Matricola, che identifica univocamente ogni alunno e viene attribuita dall'Istituto, Cognome, Nome, Luogo di Nascita, Data di Nascita, Indirizzo di residenza, classe frequentata. Per le classi si ritiene rilevante il numero di ore settimanali e l'attuazione o meno della riduzione di orario.

IPOTESI AGGIUNTIVE

Si suppone che l'archivio sia di tipo **attuale (non storico)**, perché i dati degli studenti e delle classi degli aa.ss. precedenti non vengono conservati.

SCHEMA CONCETTUALE



Vincoli espliciti

V1 (Alunno): MF IN ('M', 'F')

V2 (Classe): Anno BETWEEN 1 AND 5

V3 (Classe): Sezione BETWEEN 'A' AND 'Z'

SCHEMA LOGICO

tblClassi (Anno, Sezione, Indirizzo, OreSett, OrarioRidotto)

tblAlunni (Matricola, Cognome, Nome, MF, DataNascita, LuogoNascita, Comune, **Recapito**, {Telefono}, DataIscrizione, Anno, Sezione, Indirizzo)

Nota

Nello schema logico l'attributo *Indirizzo* dell'entità *Alunno* viene rinominato in *Recapito* per evitare il conflitto di nomi con la chiave esterna *Indirizzo* verso l'entità *Classe*.

SCHEMA FISICO

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

Nome del Database: **db01_Alunni.accdb**

Tabelle

tblClassi

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Anno	Numerico	Byte	SI	>=1 AND <=5
K	Sezione	Testo	1	SI	>='A' AND <='Z'
K	Indirizzo	Testo	1	SI	>='A' AND <='Z'
	OreSett	Numerico	Byte	SI	>0 AND <=36
	OrarioRidotto	Booleano		SI	

tblAlunni

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Matricola	Testo	5	SI	
	Cognome	Testo	30	SI	
	Nome	Testo	30	SI	
	MF	Testo	1	SI	= 'M' OR = 'F'
	DataNascita	Data/Ora		SI	
	LuogoNascita	Testo	30	SI	
	Comune	Testo	30	SI	
	Recapito	Testo	40	SI	
	Telefono	Testo	13	NO	
	DataIscrizione	Data/Ora		SI	
FK	Anno	Numerico	Byte	SI	
FK	Sezione	Testo	1	SI	
FK	Indirizzo	Testo	1	SI	

Chiave esterna (Anno, Sezione, Indirizzo)

referenzia: tblClassi (Anno, Sezione, Indirizzo)

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, la specializzazione principale; di ogni paziente devono essere registrati codice fiscale, cognome, nome, data e luogo di nascita, indirizzo.

Svolgimento

ANALISI DELLA REALTÀ DI INTERESSE

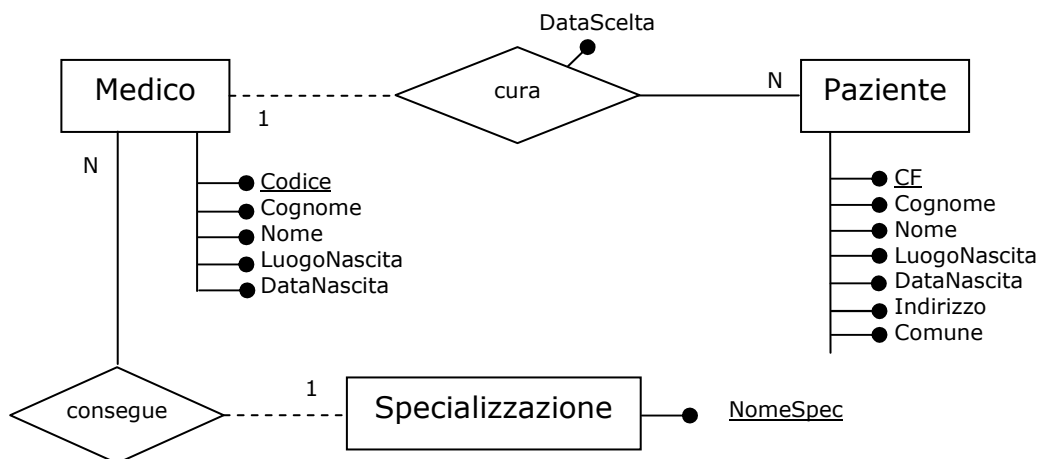
Per una organizzazione all'interno di una ASL utile a relazionare i medici ai propri pazienti, si vuole progettare la base di dati in cui l'ambito della ASL rappresenta la realtà di interesse, mentre Medico e Paziente costituiscono le due entità.

E' noto che ogni paziente debba necessariamente essere assistito da un medico di base.

IPOTESI AGGIUNTIVE

Si suppone che per le necessità gestionali della ASL sia sufficiente un archivio solo attuale, e pertanto l'associazione tra Medico e Paziente è di tipo 1:N, non potendo un paziente essere assistito contemporaneamente da più medici.

SCHEMA CONCETTUALE



SCHEMA LOGICO

tblSpecializzazioni (NomeSpec)

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

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

SCHEMA FISICO

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

Nome del Database: **db02_ASL**

Tabelle

tblSpecializzazioni

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

tblMedici

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Codice	Testo	5	SI	
	Cognome	Testo	30	SI	
	Nome	Testo	30	SI	
	LuogoNascita	Testo	30	SI	
	DataNascita	Data/Ora		SI	
FK	Specializzazione	Testo	20		

Chiave esterna: Specializzazione

referenzia: tblSpecializzazioni (NomeSpec)

tblPazienti

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	CF	Testo	16	SI	
	Cognome	Testo	30	SI	
	Nome	Testo	30	SI	
	LuogoNascita	Testo	30	SI	
	DataNascita	Data/Ora		SI	
	Indirizzo	Testo	30	SI	
	Comune	Testo	30	SI	
FK	CodiceMedico	Testo	5	SI	
	DataScelta	Data/Ora		SI	

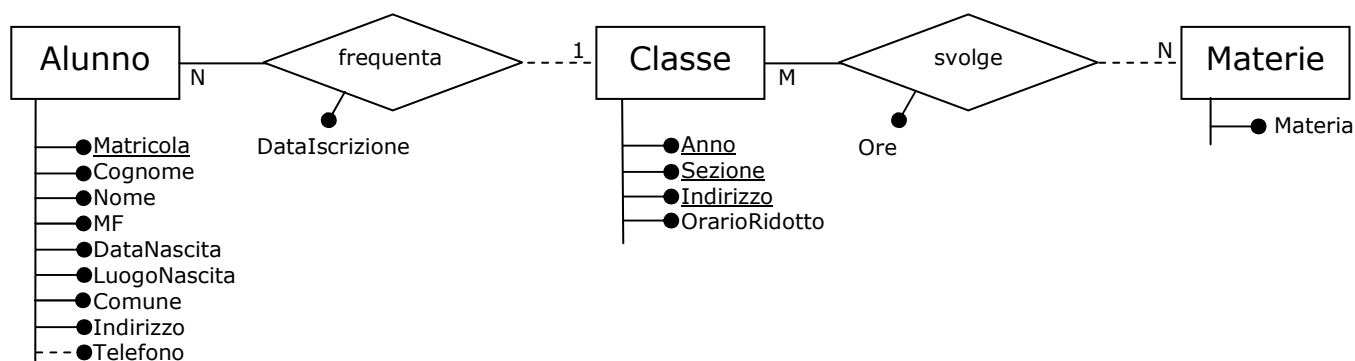
Chiave esterna: CodiceMedico

referenzia: tblMedici (Codice)

Esercizio 3: Variante all'esercizio N°1

Aggiungere al DB degli alunni il numero di ore di lezione svolte per ogni materia in ciascuna classe.

SCHEMA CONCETTUALE



SCHEMA LOGICO

tblClassi (Anno, Sezione, Indirizzo, OrarioRidotto)

tblAlunni (Matricola, Cognome, Nome, MF, DataNascita, LuogoNascita, Comune, Recapito, {Telefono}, DataIscrizione, Anno, Sezione, Indirizzo)

tblMaterie(Materia)

tblClassiMaterie(Anno, Sezione, Indirizzo, Materia, Ore)

SCHEMA FISICO

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

Nome del Database: **db03_Alunni.accdb**

Tabelle

tblClassi

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Anno	Numerico	Byte	SI	>=1 AND <=5
K	Sezione	Testo	1	SI	>='A' AND <='Z'
K	Indirizzo	Testo	1	SI	>='A' AND <='Z'
	OrarioRidotto	Booleano		SI	

tblAlunni

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Matricola	Testo	5	SI	
	Cognome	Testo	30	SI	
	Nome	Testo	30	SI	

	MF	Testo	1	SI	= 'M' OR = 'F'
	DataNascita	Data/Ora		SI	
	LuogoNascita	Testo	30	SI	
	Comune	Testo	30	SI	
	Recapito	Testo	40	SI	
	Telefono	Testo	13	NO	
	DataIscrizione	Data/Ora		SI	
FK	Anno	Numerico	Byte	SI	
FK	Sezione	Testo	1	SI	
FK	Indirizzo	Testo	1	SI	

Chiave esterna (Anno, Sezione, Indirizzo)

referenzia: tblClassi (Anno, Sezione, Indirizzo)

tblMaterie

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Materia	Testo	40	SI	>=1 AND <=5

tblMaterieClassi

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
FK,K	Anno	Numerico	Byte	SI	
FK,K	Sezione	Testo	1	SI	
FK,K	Indirizzo	Testo	1	SI	
FK,K	Materia	Testo	40	SI	
	Ore	Byte		SI	

Chiave esterna (Anno, Sezione, Indirizzo)

referenzia: tblClassi (Anno, Sezione, Indirizzo)

Chiave esterna (Materia)

referenzia: tblMaterie(Materia)

Esercizio 4: istituto di vendite giudiziarie

Un Istituto di Vendite Giudiziarie specializzato in opere d'arte, necessita di un database per archiviare le opere vendute con le rispettive caratteristiche essenziali e il relativo acquirente. L'istituto classifica le opere d'arte in: Pitture, Sculture, Bassorilievi. Tutte le opere hanno un autore e un anno di realizzazione (che possono anche essere sconosciuti). Per le pitture sono rilevanti le dimensioni della tela e la tecnica pittorica. Per le sculture è rilevante il materiale utilizzato, le dimensioni (3D) e il peso. Per i bassorilievi è rilevante anche l'epoca storica e il luogo di ritrovamento.

Sviluppare il progetto fino allo schema fisico, e popolare le tabelle con alcuni dati di prova.

Svolgimento

ANALISI DELLA REALTÀ DI INTERESSE

Il progetto è rivolto alla gestione informatizzata di un istituto di vendite giudiziarie.

Devono essere archiviati i dati delle opere vendute e degli acquirenti.

Le opere possono essere pitture, sculture e bassorilievi. Tutte le opere hanno, come caratteristiche comuni, l'autore e l'anno di realizzazione. Ogni tipo di opera ha poi caratteristiche proprie:

- dimensioni (2D) della tela e tecnica pittorica per le pitture;
- dimensioni (3D) e peso per le sculture;
- epoca storica e il luogo di ritrovamento per i bassorilievi.

IPOTESI AGGIUNTIVE

Le dimensioni (2D) della tela e dei bassorilievi (che sono comunque a due dimensioni) sono intese come Larghezza x Altezza.

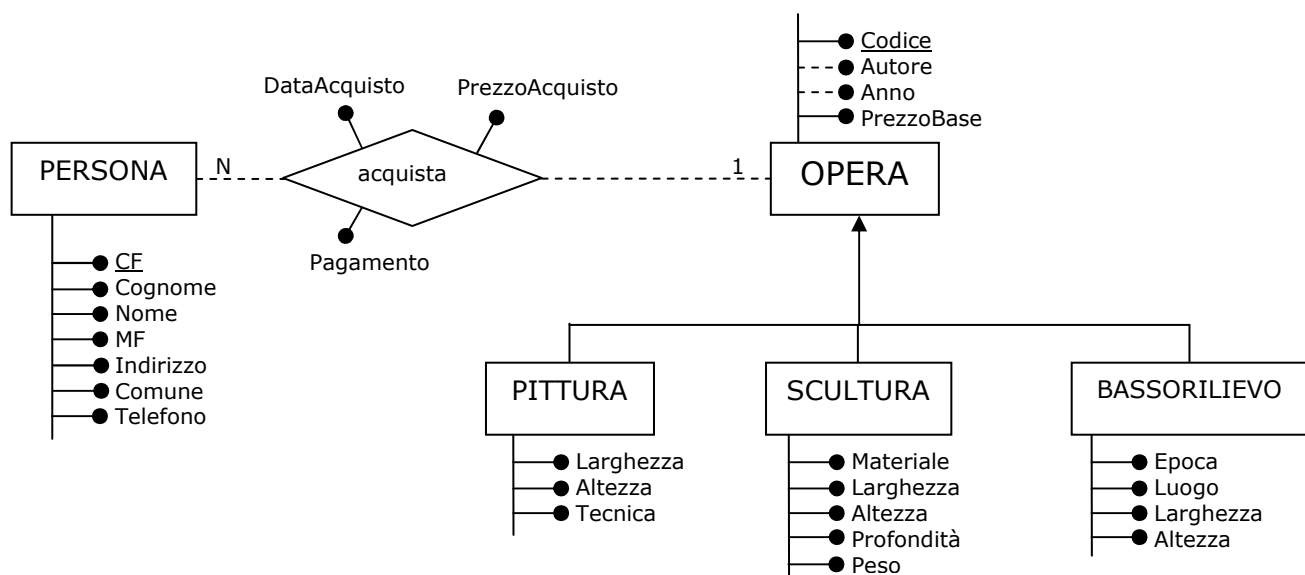
Le dimensioni (3D) delle sculture sono intese come Larghezza x Altezza x Profondità.

Per ogni acquirente si registreranno i principali dati anagrafici (codice fiscale, cognome, nome, sesso, indirizzo completo) e un recapito telefonico.

Per ogni opera si ritiene opportuno registrare anche il prezzo base.

Per ogni acquisto, si ritiene opportuno registrare la data, il prezzo e la forma di pagamento, che si ipotizza possa essere solo in contanti o a rate, per cui è sufficiente un carattere per la memorizzazione di quest'informazione ('C', 'R').

SCHEMA CONCETTUALE



SCHEMA LOGICO

tblPersone (CF, Cognome, Nome, MF, Indirizzo, Comune, Telefono) ←

tblPitture (Codice, Autore, Anno, PrezzoBase, Larghezza, Altezza, Tecnica, DataAcquisto, PrezzoAcquisto, Pagamento, Acquirente)

tblSculture (Codice, Autore, Anno, PrezzoBase, Materiale, Larghezza, Altezza, Profondita, Peso, DataAcquisto, PrezzoAcquisto, Pagamento, Acquirente)

tblBassorilievi (Codice, Autore, Anno, PrezzoBase, Epoca, Luogo, Larghezza, Altezza, DataAcquisto, PrezzoAcquisto, Pagamento, Acquirente)

SCHEMA FISICO

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

Nome del Database: **db04_ VenditeGiudiziarie.accdb**

Tabelle

tblPersone

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	CF	Testo	16	SI	
	Cognome	Testo	20	SI	
	Nome	Testo	20	SI	
	MF	Testo	1	SI	= 'M' OR = 'F'
	Indirizzo	Testo	30	SI	
	Comune	Testo	25	SI	
	Telefono	Testo	15	SI	

tblPitture

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Codice	Testo	5	SI	
	Autore	Testo	30	SI	
	Anno	Numerico	Intero	SI	
	PrezzoBase			SI	
	Larghezza	Numerico	Byte	SI	
	Altezza	Numerico	Byte	SI	
	Tecnica	Testo	30	SI	
	DataAcquisto	Data/Ora	13	NO	
	PrezzoAcquisto	Valuta		SI	
	Pagamento	Testo	1	SI	= 'R' OR = 'C'
FK	Acquirente	Testo	1	SI	

Chiave esterna (Acquirente) referencia: tblPersone(CF)

tblSculture

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Codice	Testo	5	SI	
	Autore	Testo	30	SI	
	Anno	Numerico	Intero	SI	
	PrezzoBase	Valuta		SI	
	Materiale	Testo	20	SI	

	Larghezza	Numerico	Byte	SI	
	Altezza	Numerico	Byte	SI	
	Profondita	Numerico	Byte		
	Peso	Testo	30	SI	
	DataAcquisto	Data/Ora	13	NO	
	PrezzoAcquisto	Valuta		SI	
	Pagamento	Testo	1	SI	= 'R' OR = 'C'
FK	Acquirente	Testo	1	SI	

Chiave esterna (Acquirente) referenzia: tblPersone(CF)

tblBassorilievi

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Codice	Testo	5	SI	
	Autore	Testo	30	SI	
	Anno	Numerico	Intero	SI	
	PrezzoBase	Valuta		SI	
	Epoca	Testo	20	SI	
	Luogo	Testo	20	SI	
	Larghezza	Numerico	Byte	SI	
	Altezza	Numerico	Byte	SI	
	DataAcquisto	Data/Ora	13	NO	
	PrezzoAcquisto	Valuta		SI	
	Pagamento	Testo	1	SI	= 'R' OR = 'C'
FK	Acquirente	Testo	1	SI	

Chiave esterna (Acquirente) referenzia: tblPersone(CF)

Esercizio 5: listino

Si consideri un sistema per la gestione di listini di vendita in valute diverse. Un listino è relativo a una zona di vendita e comprende i prezzi di un insieme di prodotti. Un prodotto può comparire in più listini; in alcuni listini uno stesso prodotto può comparire più volte, quotato in valute diverse (ad esempio: lire, dollari e marchi). A ogni listino è associato uno sconto.

Svolgimento

ANALISI DELLA REALTÀ DI INTERESSE

Il progetto è rivolto alla informatizzazione di un sistema di listini.

Un listino fa capo ad una sola zona.

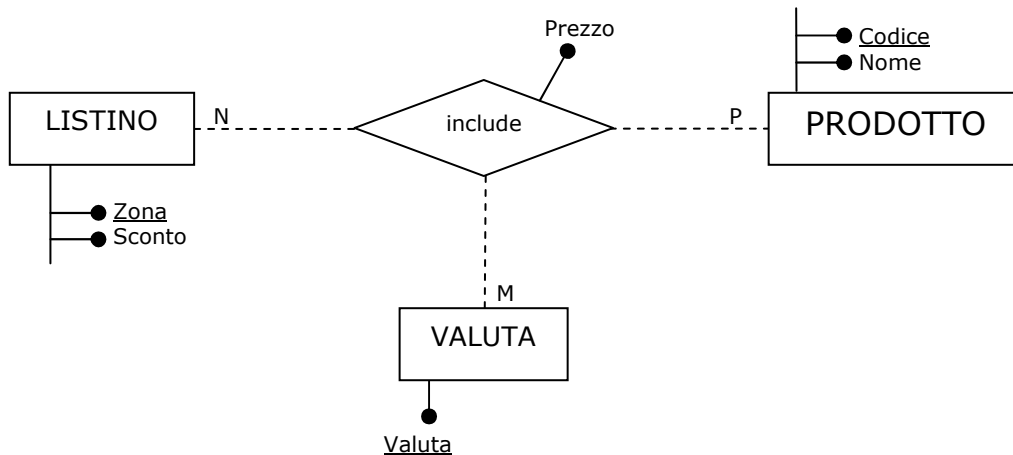
Ogni listino include più prodotti.

Un prodotto può comparire più volte in un listino, ma con valute diverse.

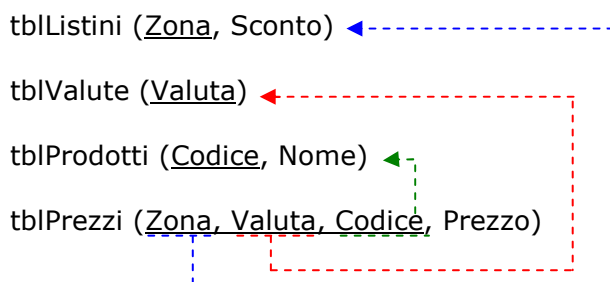
IPOTESI AGGIUNTIVE

Si ipotizza (la traccia non è esplicita in questo senso) che una zona abbia un solo listino, per cui si può identificare ogni listino tramite la zona di appartenenza.

SCHEMA CONCETTUALE



SCHEMA LOGICO



SCHEMA FISICO

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

Nome del Database: **db07_ Listini.accdb**

Tabelle

tblListini

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Zona	Testo	20	Sì	
	Sconto	Numerico	Byte	Sì	

tblValute

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Valuta	Testo	15	SI	

Chiave esterna (Nazione) referencia: tblNazioni(Nazione)

tblProdotti

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Codice	Testo	3	SI	
	Nome	Testo	30	SI	

tblPrezzi

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
FK,K	Zona	Testo	20	SI	
FK,K	Valuta	Valuta		SI	
FK,K	Codice	Testo	3	SI	
	Prezzo				

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.

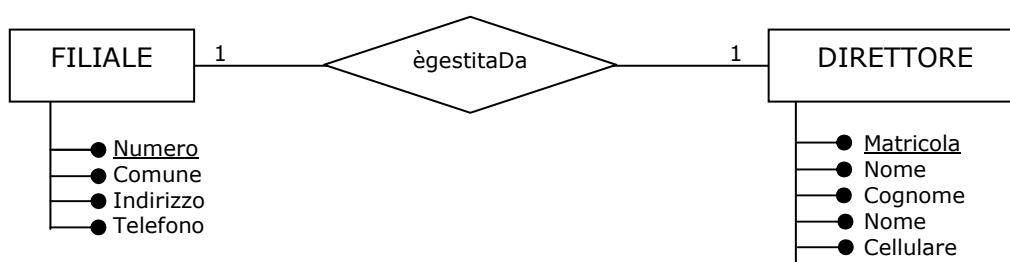
Si individuano come entità FILIALE e DIRETTORE. Tra esse si individua un'associazione 1:1.

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, l'indirizzo e il numero di telefono, mentre, per ogni direttore, una matricola (che lo identifica univocamente), il cognome, il nome e un numero di cellulare.

SCHEMA CONCETTUALE



SCHEMA LOGICO

tblListini (Numero, Comune, Inirizzo, Telefono, MatrDir, CognomeDir, NomeDir, CellDir)

SCHEMA FISICO

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

Nome del Database: **db06_ Banca.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	Telefono	Testo	13	SI	
	MatrDir	Testo	4	NO	
	CognomeDir	Testo	25	SI	
	NomeDir	Testo	25	SI	
SK	CellDir	Testo	13	NO	

Esercizio 7: Formula 1 (compito)

Si devono raccogliere e organizzare le informazioni relative al campionato mondiale di F1 dell'anno in corso. Il campionato è articolato in una serie di gare, ciascuna delle quali si svolge in una diversa data dell'anno e località del mondo. In un giorno si può tenere una sola gara; in una località si può tenere una sola gara.

Ad ogni gara partecipano i piloti iscritti al campionato. Ogni pilota, identificato da un numero, fa parte di una scuderia, di cui si devono registrare il nome e la nazionalità di appartenenza. Una scuderia può avere più piloti. Di ciascun pilota si deve tenere memoria del cognome, del nome e della data di nascita.

In ogni gara pilota può guadagnare punti.

Sono richiesti:

- l'analisi della realtà di interesse, con eventuali ipotesi aggiuntive;
- lo schema concettuale;
- lo schema logico;
- lo schema fisico.

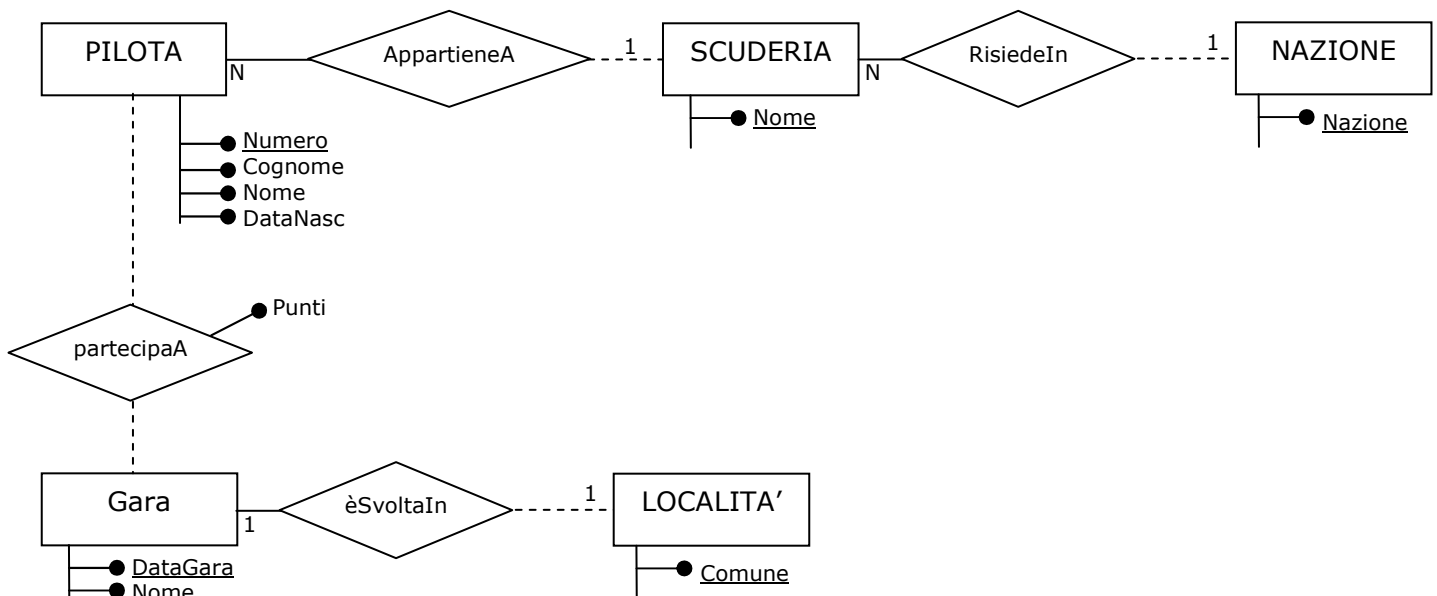
Svolgimento

ANALISI DELLA REALTÀ DI INTERESSE

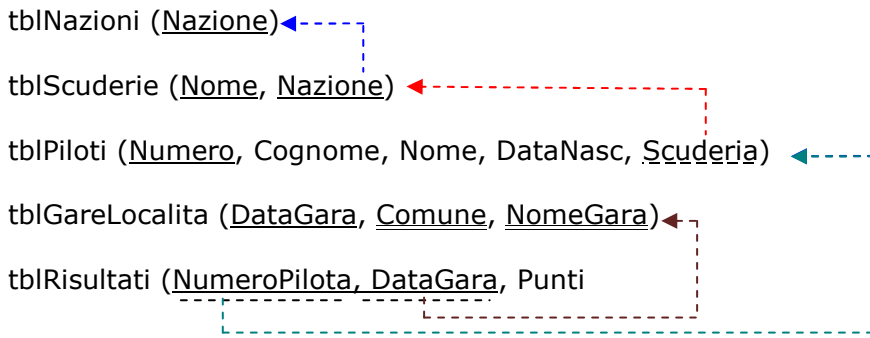
La realtà d'interesse è il Campionato mondiale di F1 corrente. Il DB da progettare quindi non è storico, perché si registrano solo i dati attuali.

Si individuano come entità LOCALITA', SCUDERIA, PILOTA, GARA e NAZIONE.

SCHEMA CONCETTUALE



SCHEMA LOGICO



SCHEMA FISICO

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

Nome del Database: **db07_Formula1.accdb**

Tabelle

tblNazioni

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

tblScuderie

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Nome	Testo	15	SI	
FK	Nazione	Testo	20	SI	

Chiave esterna (Nazione) referencia: tblNazioni(Nazione)

tblPiloti

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Numero	Numerico	Byte	SI	
	Cognome	Testo	20	SI	
	Nome	Testo	20	SI	
	DataNasc	Data		SI	
FK	Scuderia	Testo	15	SI	

tblGareLocalita

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	DataGara	Data		SI	
SK	Comune	Testo	30	SI	
SK	NomeGara	Testo	40	SI	

tblRisultati

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K,FK	NumeroPilota	Numerico	Byte	SI	
K,FK	DataGara	Data		SI	
	Punti	Numerico	Byte	SI	

Chiave esterna (NumeroPilota) referencia: tblScuderie(Numero)
 Chiave esterna (DataGara) referencia: tblGareLocalita(DataGara)

Esercizio 8: Impresa edile (compito)

Si progetti un sistema informativo per la gestione di un'impresa edile.

Il sistema deve gestire i cantieri in corso e i dipendenti.

Di ogni cantiere, oltre alla propria localizzazione sul territorio, alla descrizione, alla data di inizio lavori e alla data di fine prevista, si devono memorizzare i dati dei dipendenti assegnati. Un dipendente potrebbe lavorare in più cantieri. Per ogni dipendente, oltre ai dati anagrafici, si devono conoscere la qualifica (muratore, carpentiere, autista, geometra, ecc.) e, per ogni cantiere in cui sta lavorando, la data di inizio incarico. Ad ogni qualifica corrisponde una fascia stipendiale.

Sono richiesti:

- l'analisi della realtà di interesse, con eventuali ipotesi aggiuntive;
- lo schema concettuale;
- lo schema logico;
- lo schema fisico.

Svolgimento

ANALISI DELLA REALTÀ DI INTERESSE

La realtà d'interesse è un'impresa edile, di cui si devono registrare dati relativi ai cantieri in corso. Il DB da progettare quindi non è storico, perché si registrano solo i dati attuali.

Si individuano, come entità, CANTIERE, DIPENDENTE e QUALIFICA.

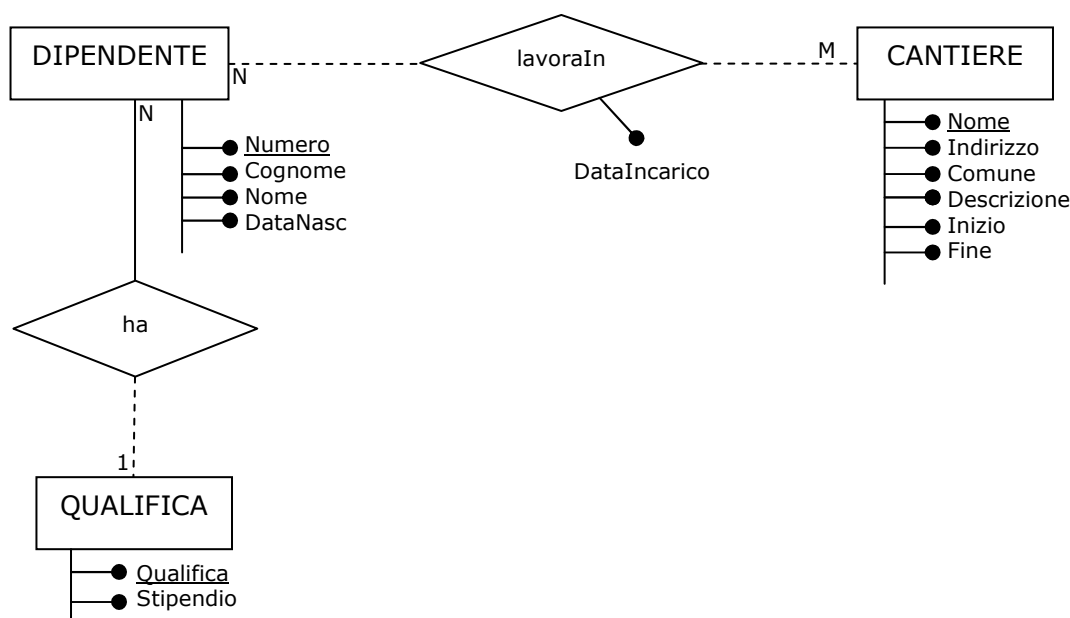
IPOTESI AGGIUNTIVE

L'impresa può gestire cantieri in più comuni.

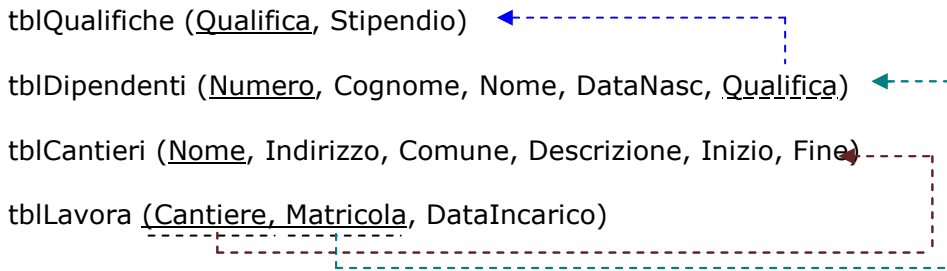
Ad ogni cantiere l'impresa attribuisce un nome che lo identifichi univocamente.

Per località si intende un'indicazione della posizione del cantiere, ad esempio la via.

SCHEMA CONCETTUALE



SCHEMA LOGICO



SCHEMA FISICO

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

Nome del Database: **db07_ Formula1.accdb**

Tabelle

tblNazioni

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

tblScuderie

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Nome	Testo	15	SI	
FK	Nazione	Testo	20	SI	

Chiave esterna (Nazione) referencia: tblNazioni(Nazione)

tblPiloti

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	Numero	Numerico	Byte	SI	
	Cognome	Testo	20	SI	
	Nome	Testo	20	SI	
	DataNasc	Data		SI	
FK	Scuderia	Testo	15	SI	

tblGareLocalita

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K	DataGara	Data		SI	
SK	Comune	Testo	30	SI	
SK	NomeGara	Testo	40	SI	

tblRisultati

Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
K,FK	NumeroPilota	Numerico	Byte	SI	
K,FK	DataGara	Data		SI	
	Punti	Numerico	Byte	SI	

Chiave esterna (NumeroPilota)

referencia: tblScuderie(Numero)

Chiave esterna (DataGara)

referencia: tblGareLocalita(DataGara)

Esercizio 9: 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.

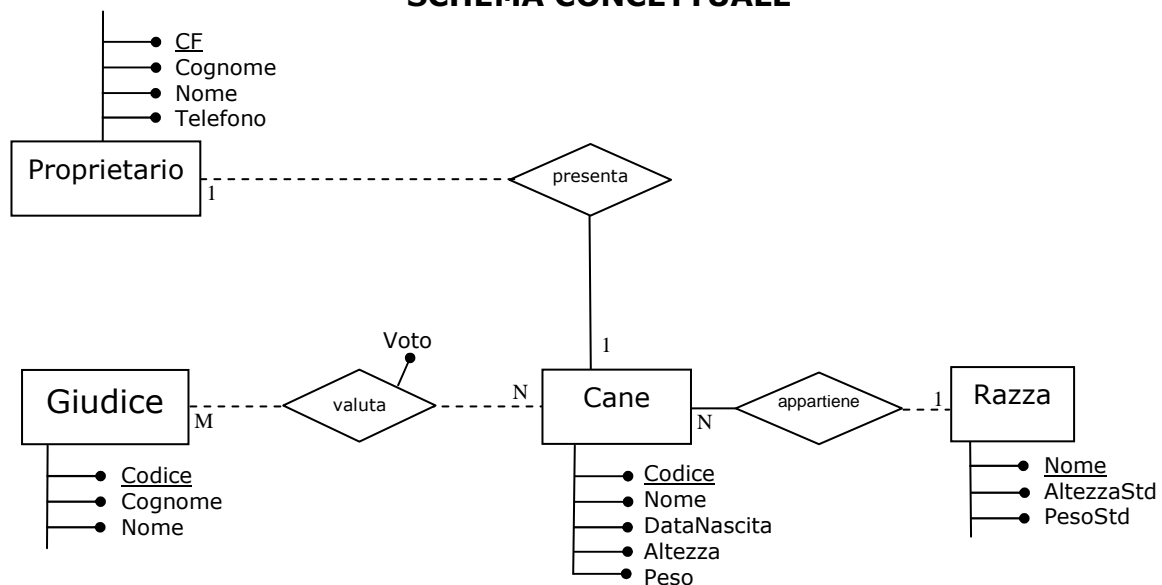
IPOTESI AGGIUNTIVE

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 >= 20

V2 (tblRazze): PesoSt >= 2

V3 (tblCani): Altezza >0

V4 (tblCani): Peso >0

V5 (tblValutazioni): Voto Between 1 AND 10

SCHEMA FISICOPercorso 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) referencia: 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

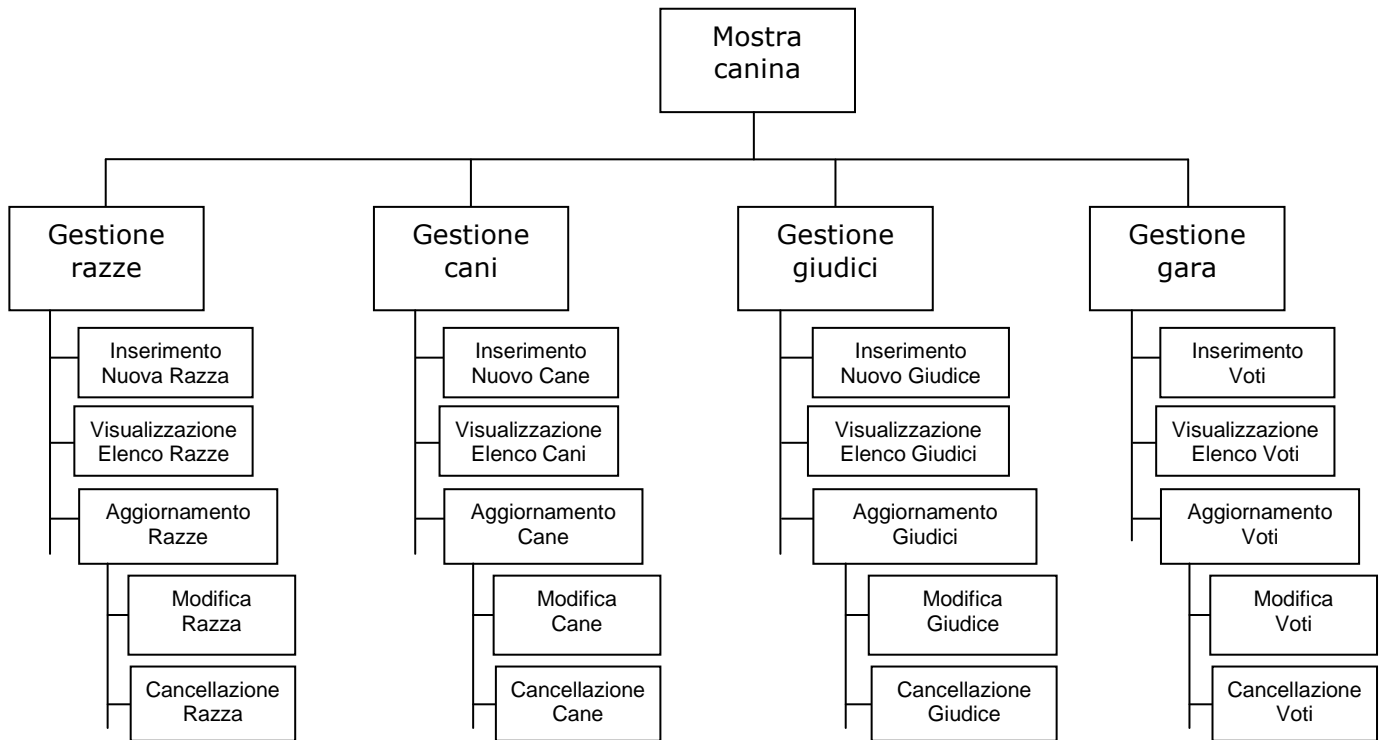
Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se
FK,K	Giudice	Testo	3	SI	
FK,K	Cane	Numerico	Byte	SI	
	Voto	Numerico	Byte	SI	Between 1 and 10

Chiave esterna (Cane) referencia: tblCani(NumeroC)

Chiave esterna (Giudice) referencia: tblGiudici(CodGiud)

APPLICAZIONE

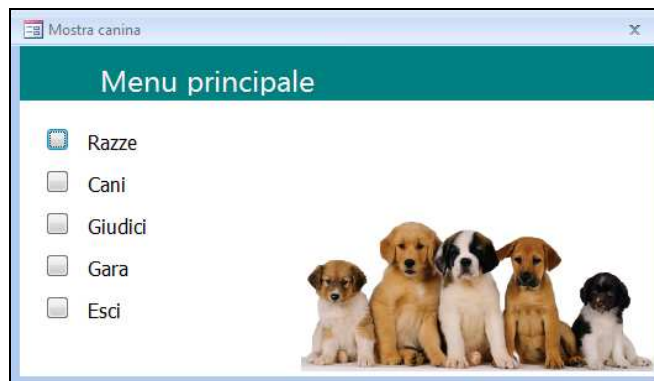
Diagramma delle funzioni



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 10: MULTISALA CINEMATOGRAFICA

Progettare il sistema informativo per una multisala cinematografica, al fine di gestire le prenotazioni per le varie rappresentazioni.

Svolgimento

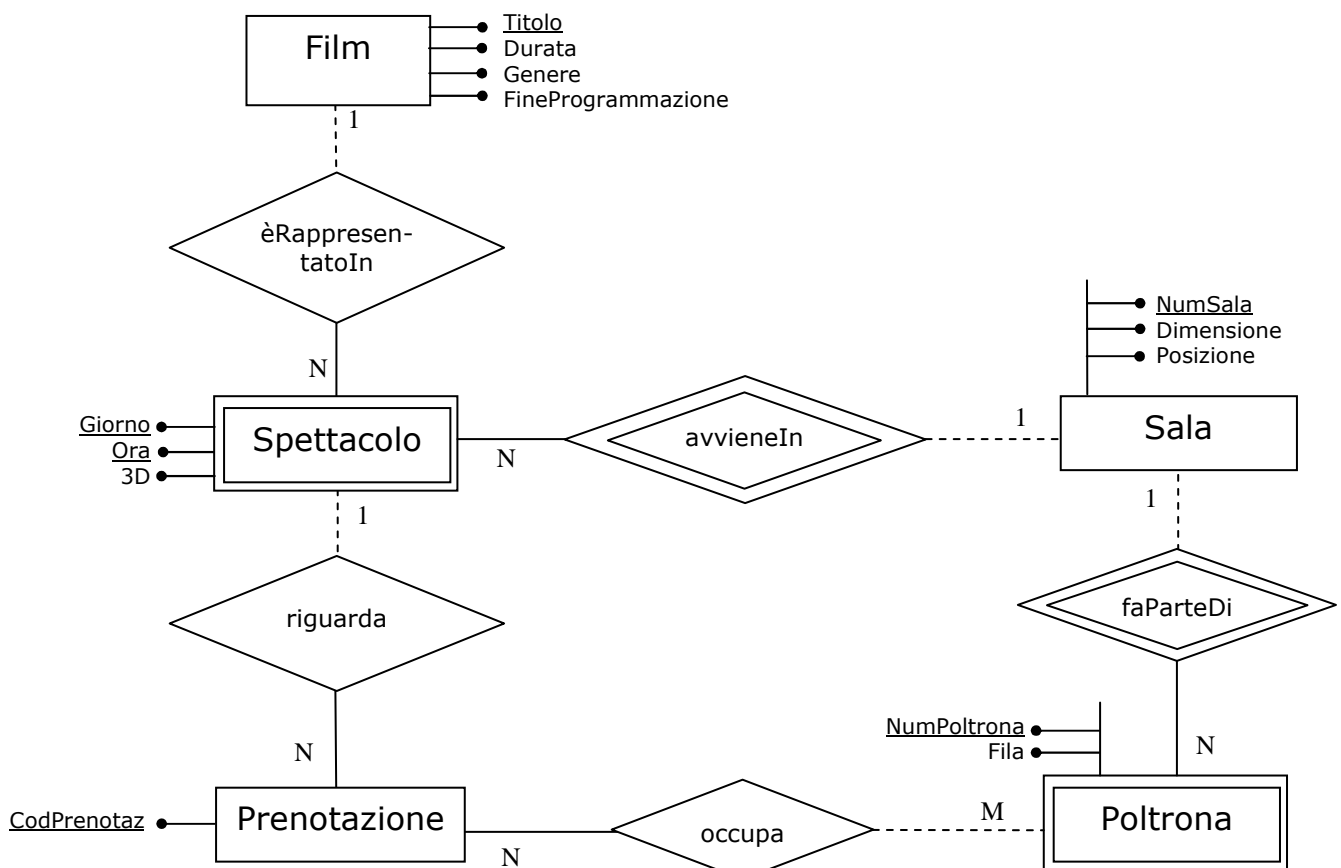
ANALISI DELLA REALTÀ DI INTERESSE

Il progetto riguarda una multisala in cui avvengono proiezioni cinematografiche. Il Database non dovrà mantenere traccia delle prenotazioni scadute, per cui è solo attuale. La traccia non fornisce una descrizione dettagliata della realtà di interesse, per cui è necessario formulare numerose ipotesi aggiuntive, sulla base dell'esperienza personale.

IPOTESI AGGIUNTIVE

1. La multisala si compone di un numero imprecisato di sale, ciascuna delle quali <contiene> poltrone numerate da 1 in avanti, e disposte in file individuate da lettere, il cui solo scopo però è agevolare lo spettatore nel raggiungere il proprio posto.
2. Ogni sala è identificata da un numero. Informazioni utili sono la dimensione in mq e la posizione (ad es.: piano basso a sinistra, piano basso a destra, piano alto a sinistra, piano alto a destra).
3. Di un film interessano il titolo, la durata, il genere. Questi dati sono forniti dal produttore (non rilevante in questa realtà) e pertanto non è necessaria un'entità di riferimento per il genere. Rilevante è altresì la data di Fine Programmazione, per evitare di accettare prenotazioni per date successive.
4. Un film <è rappresentato> in più spettacoli, per ciascuno dei quali si specificano data, ora, sala, e se la proiezione avviene in modalità stereoscopica (3D). Un film potrebbe essere rappresentato contemporaneamente in più sale.
5. Non interessa gestire un'anagrafe degli spettatori, e neppure identificare chi effettua la prenotazione.
6. È possibile prenotare solo per i film attualmente in programmazione.
7. Per far valere una prenotazione e ritirare i relativi biglietti alla cassa, è sufficiente il numero della prenotazione assegnato dall'operatore.
8. Con una prenotazione un cliente può occupare più poltrone ma per un solo spettacolo. Nulla impedisce che il cliente prenoti per più spettacoli, purché con numeri diversi perché ogni prenotazione scade subito prima dell'inizio del relativo spettacolo. Per far sì che una prenotazione possa riguardare più poltrone ma per un solo spettacolo, è necessario inquadrarla come entità.

SCHEMA CONCETTUALE



SCHEMA LOGICO

tblFilm (Titolo, Durata, Genere, FineProgrammazione) ←

tblSale (NumSala, Dimensione, Posizione) ←

tblPoltrone (NumSala, NumPoltrona, Fila) ←

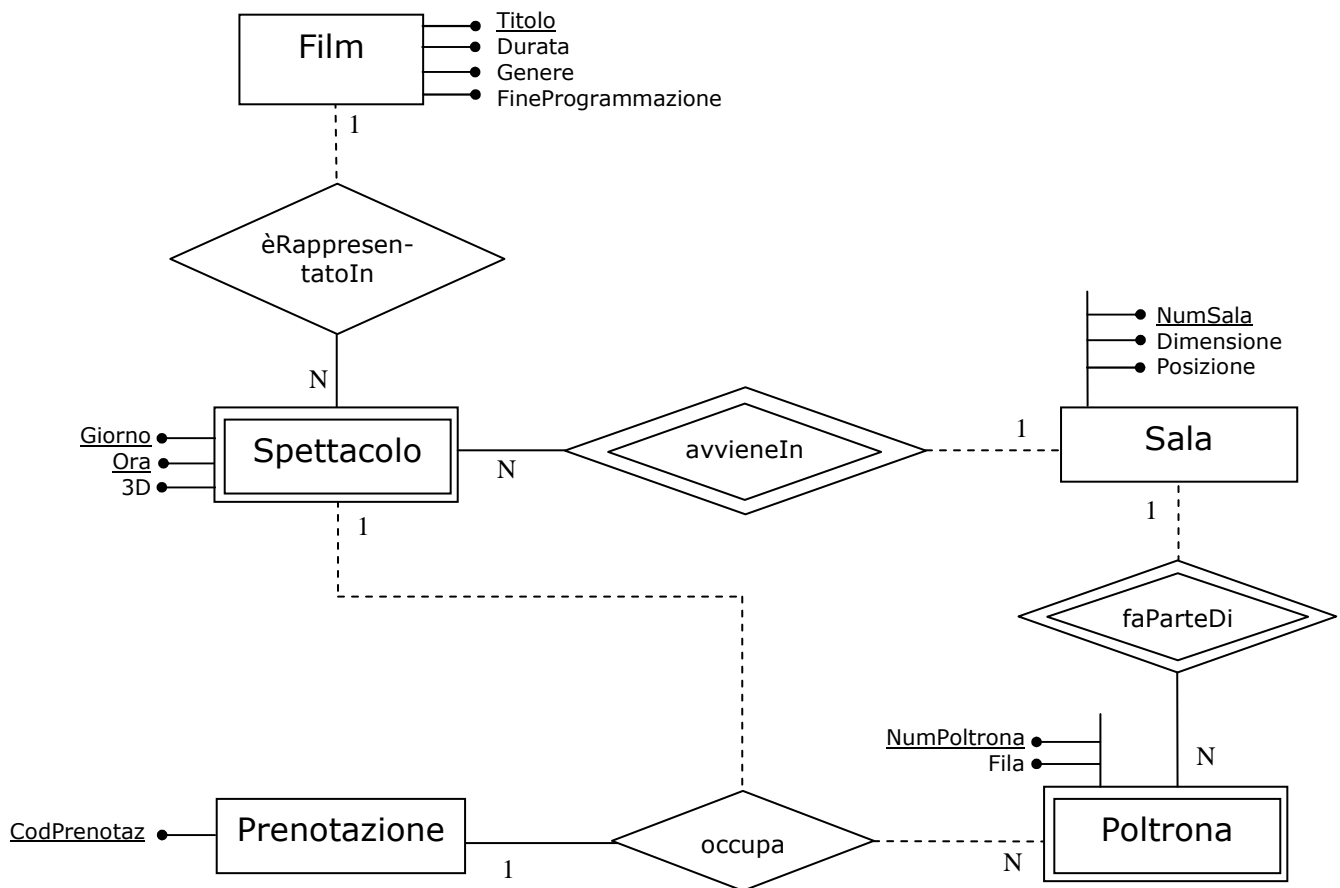
tblSpettacoli (NumSala, Giorno, Ora, Film, 3D) ←

tblPrenotazioni (CodPrenotaz, NumSala, Giorno, Ora) ←

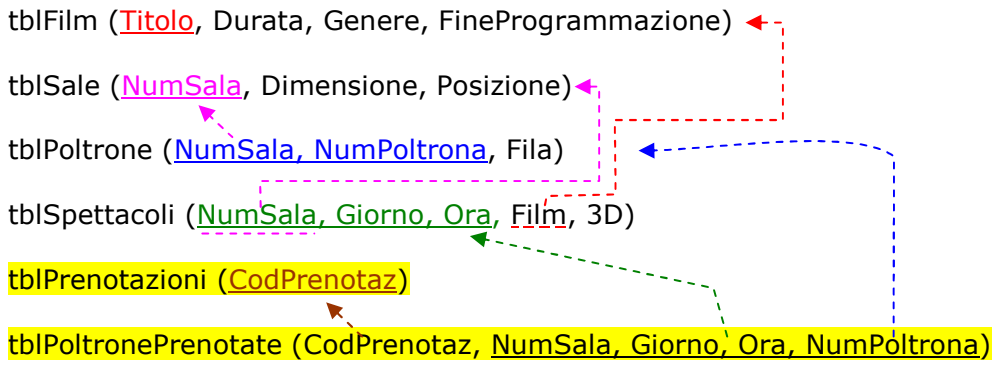
tblPoltronePrenotate (CodPrenotaz, NumPoltrona) ←

SECONDA VERSIONE

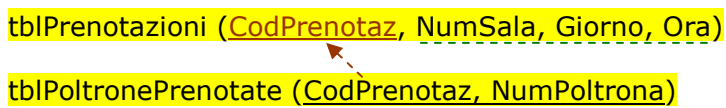
MODELLO CONCETTUALE



SCHEMA LOGICO



In seguito alla normalizzazione, poiché con la stessa prenotazione non si può prenotare per spettacoli diversi:



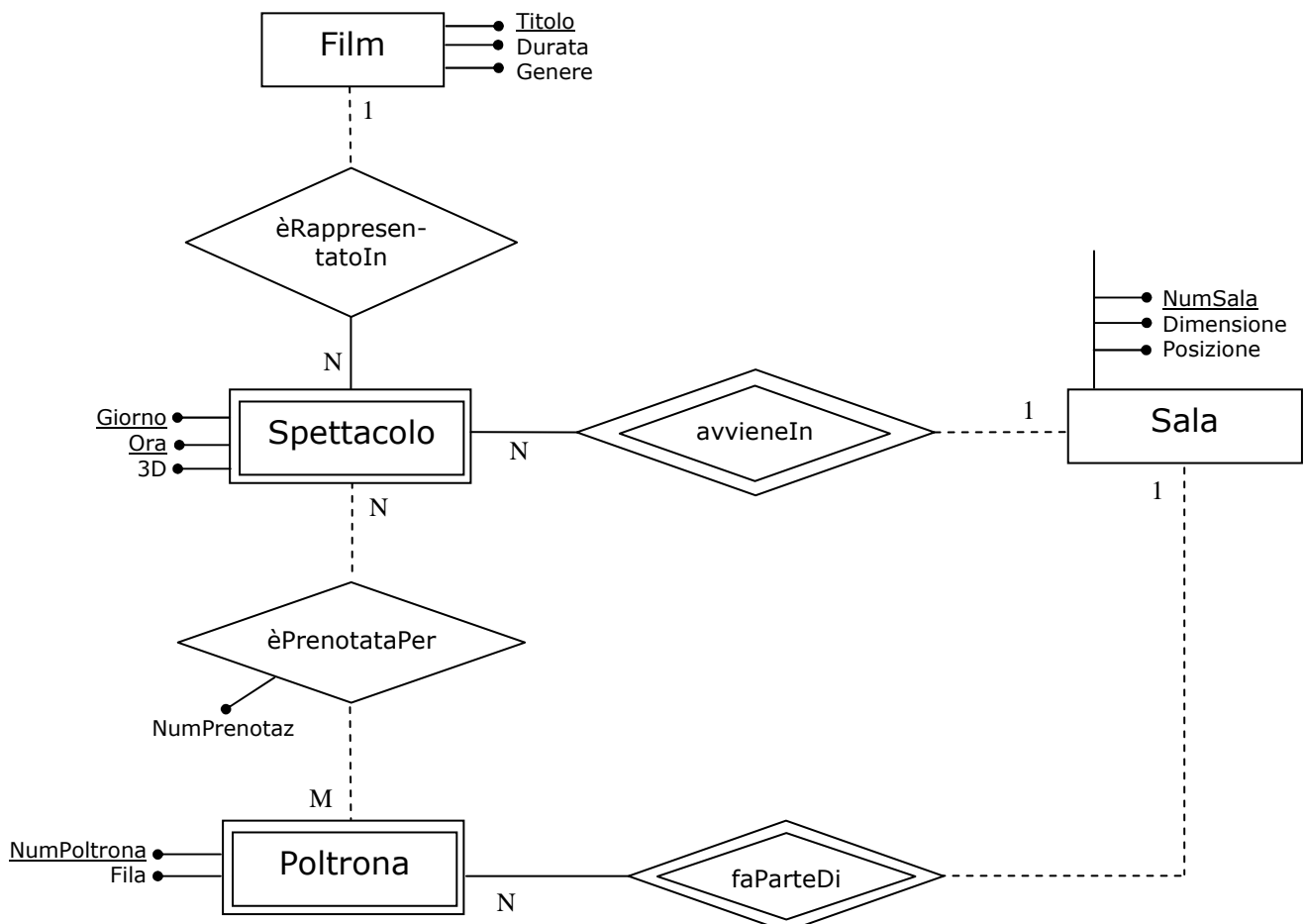
che dimostra che lo schema concettuale corretto era quello della versione precedente.

TERZA VERSIONE

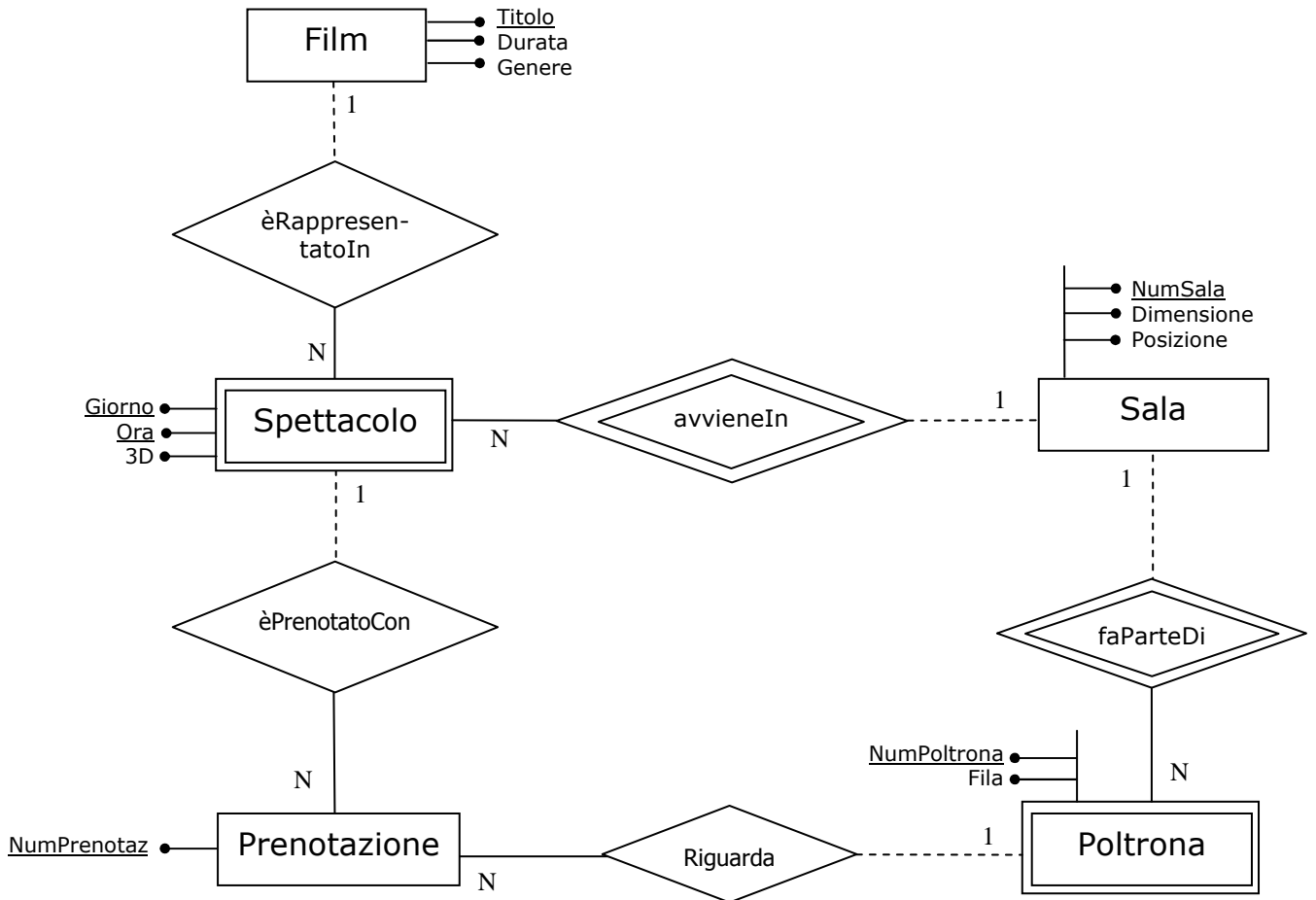
VARIANTE IPOTESI AGGIUNTIVE

8. Un numero di prenotazione occupa una sola poltrona per uno solo spettacolo.

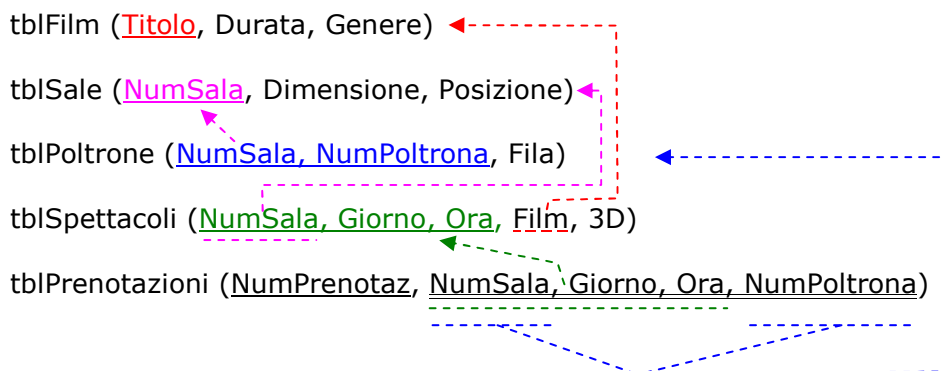
SCHEMA CONCETTUALE 1



SCHEMA CONCETTUALE 2



SCHEMA LOGICO



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.

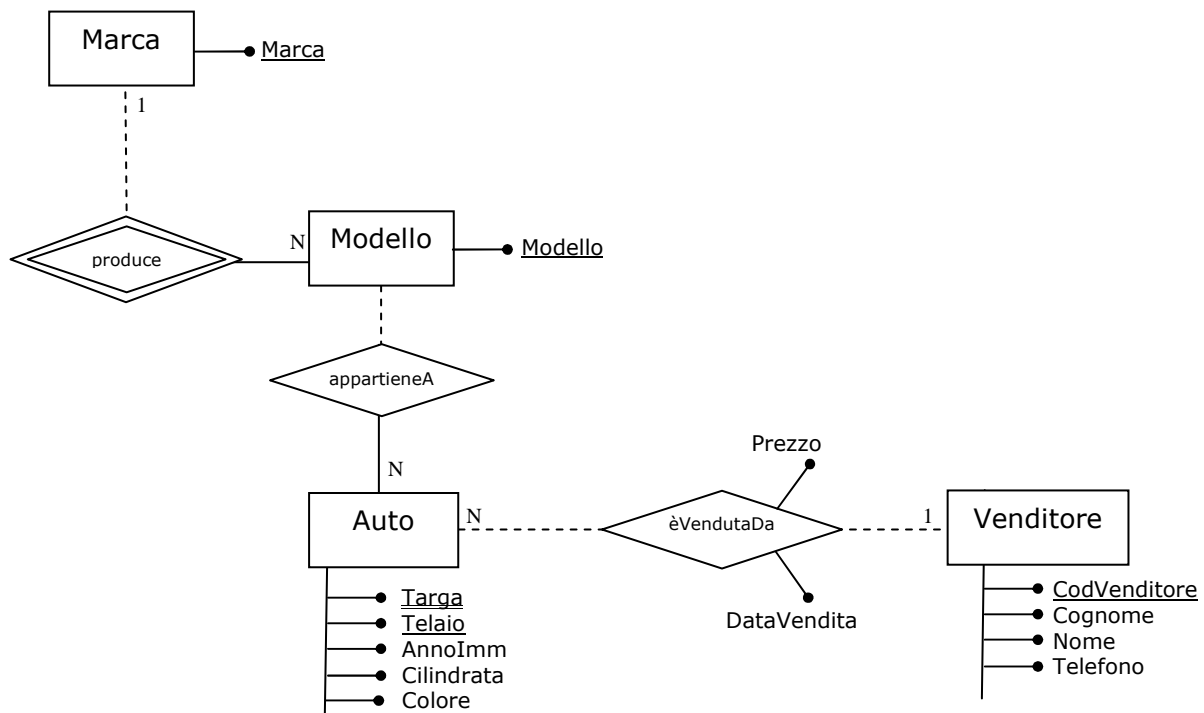
Progettare la base di dati fino allo schema fisico, e produrre l'analisi delle funzioni completa di diagramma delle funzioni e disegno dell'interfaccia utente (menu, maschere di immissione / visualizzazione / modifica / cancellazione e opportuni pulsanti).

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: Università (compito)

L'Università degli Studi di Foggia necessita di un sistema informatico per gestire la situazione dei docenti e dei corsi di laurea. L'università è organizzata in Facoltà, <suddivise> in Dipartimenti. Ogni facoltà ha un nome e una sua sede all'interno della città. Ogni dipartimento è caratterizzato da una sigla univoca di massimo 10 caratteri, e da un nome in chiaro. Ogni facoltà eroga uno o più corsi di Laurea, di Primo o Secondo Livello. Ogni corso di laurea è identificato da un codice di 5 lettere e da un nome univoco. Ogni docente viene assunto, in data da registrare, da una facoltà, e può assumere il ruolo di Associato, Straordinario o Ordinario. La facoltà gli attribuisce un numero progressivo, e ne memorizza il cognome, nome, numero di cellulare ed eventualmente l'e-mail.

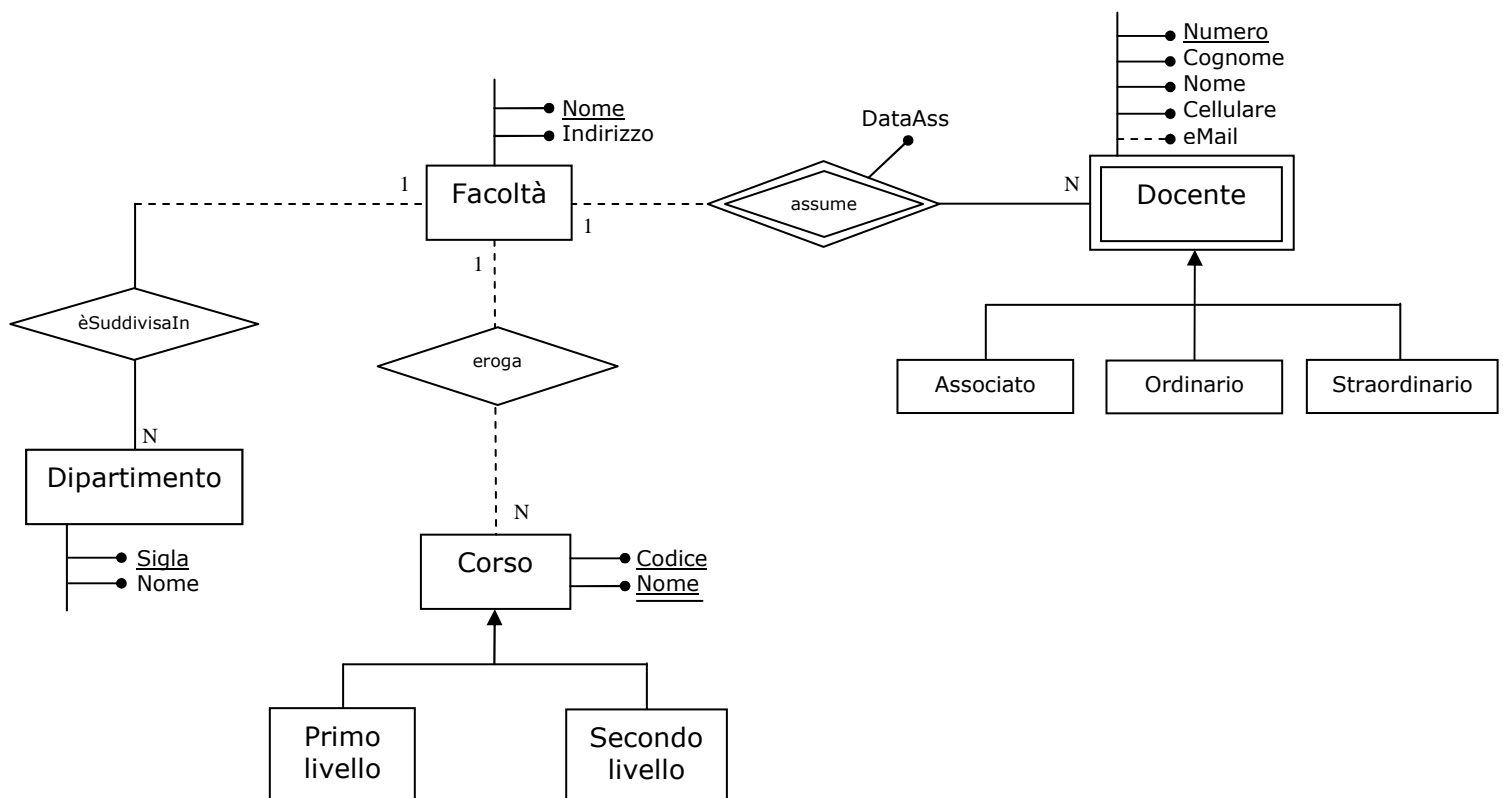
Progettare la base di dati fino allo Schema Fisico, quindi produrre l'Analisi delle Funzioni completa di Diagramma delle Funzioni e Disegno dell'interfaccia utente: Menu, Maschere per l'immissione / visualizzazione / modifica / cancellazione dei dati e opportuni pulsanti (almeno una maschera di ciascun tipo, relativa a una tabella referenziante).

Svolgimento

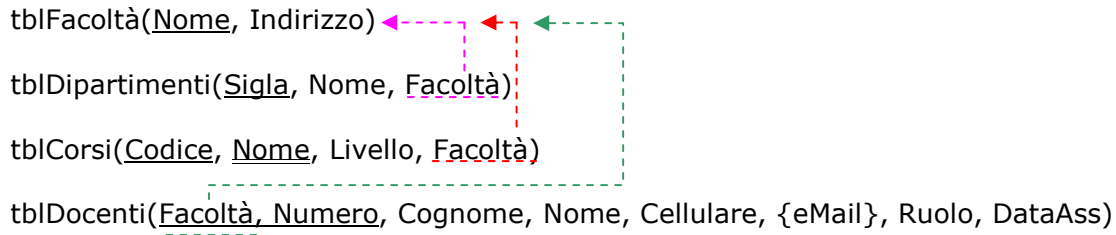
ANALISI DELLA REALTÀ DI INTERESSE

La realtà d'interesse è l'Università degli Studi di Foggia. Il DB è attuale perché si parla della "situazione dei docenti e dei corsi di laurea".

SCHEMA CONCETTUALE



SCHEMA LOGICO



Vincoli espliciti:

V1 (tblCorsi): Livello in (1, 2)

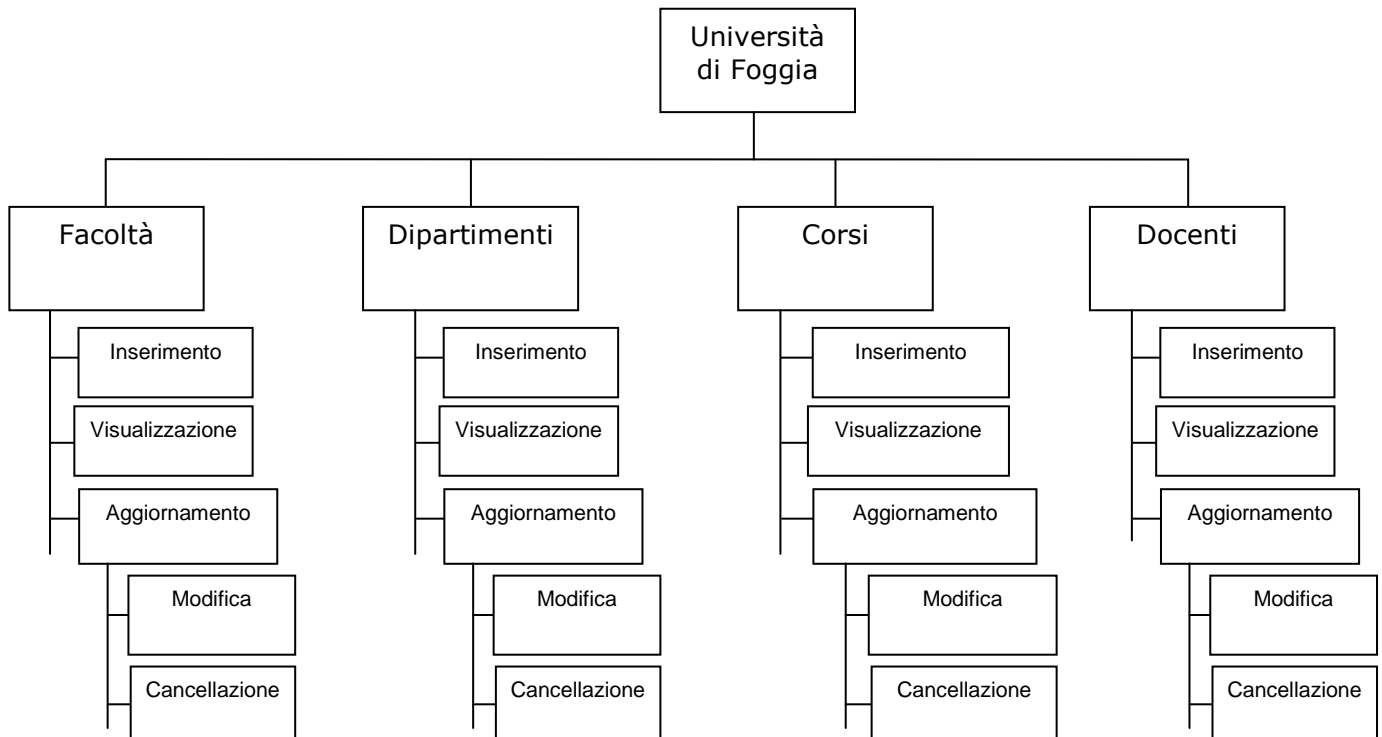
V2 (tblDocenti): Ruolo in 'A', 'O', 'S'

SCHEMA FISICO

Tabella	Chiave	Nome campo	Tipo	Dimensione	Richiesto	Valido se	Referenzia
tblFacoltà	PK	Nome	Testo	30	Sì		
		Indirizzo	Testo	30	Sì		
tblDipartimenti	PK	Sigla	Testo	5	Sì		
		Nome	Testo	30	Sì		
		Facoltà	Testo	30	Sì		tblFacoltà
tblCorsi	PK	Codice	Testo	5	Sì		
	SK	Nome	Testo	30	Sì		
		Livello	Byte	1	Sì	in (1, 2)	
		Facoltà	Testo	30	Sì		tblFacoltà
tblDocenti	PK	Numero	Numerico	Byte	Sì		
		Cognome	Testo	20	Sì		
		Nome	Testo	20	Sì		
		Cellulare	Testo	15	Sì		
		Email	Testo	20	No		
		Ruolo	Testo	1	Sì	in ('A', 'O', 'S')	
		Facoltà	Testo	30	Sì		tblFacoltà
		DataAss	Data/Ora		Sì		

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.

Università di Foggia	
<input type="checkbox"/>	Facoltà
<input type="checkbox"/>	Dipartimenti
<input type="checkbox"/>	Corsi
<input type="checkbox"/>	Docenti
<input type="checkbox"/>	Uscita

Maschere di II livello

Si sceglie di sviluppare la maschera relativa alla tabella dei corsi. Le altre sono analoghe.

Corsi	
<input type="checkbox"/>	Inserimento
<input type="checkbox"/>	Visualizzazione elenco
<input type="checkbox"/>	Aggiornamento
<input type="checkbox"/>	Indietro

Maschere di III livello

Maschere relative alla gestione della tabella dei corsi.

Inserimento Corsi

Codice

Nome

Livello

☒ Primo livello

☐ Secondo livello

Facoltà

▼

Salva

Annulla

Esci

◀◀

▶▶

*

Aggiornamento Corsi

Codice

Nome

Livello

☒ Primo livello

☐ Secondo livello

Facoltà

▼

Salva

Elimina

Annulla

Esci

◀◀

▶▶

Visualizzazione Elenco Corsi

Codice	Nome	Livello	Facoltà

Esci

Esercizio 13: Euronics (compito)

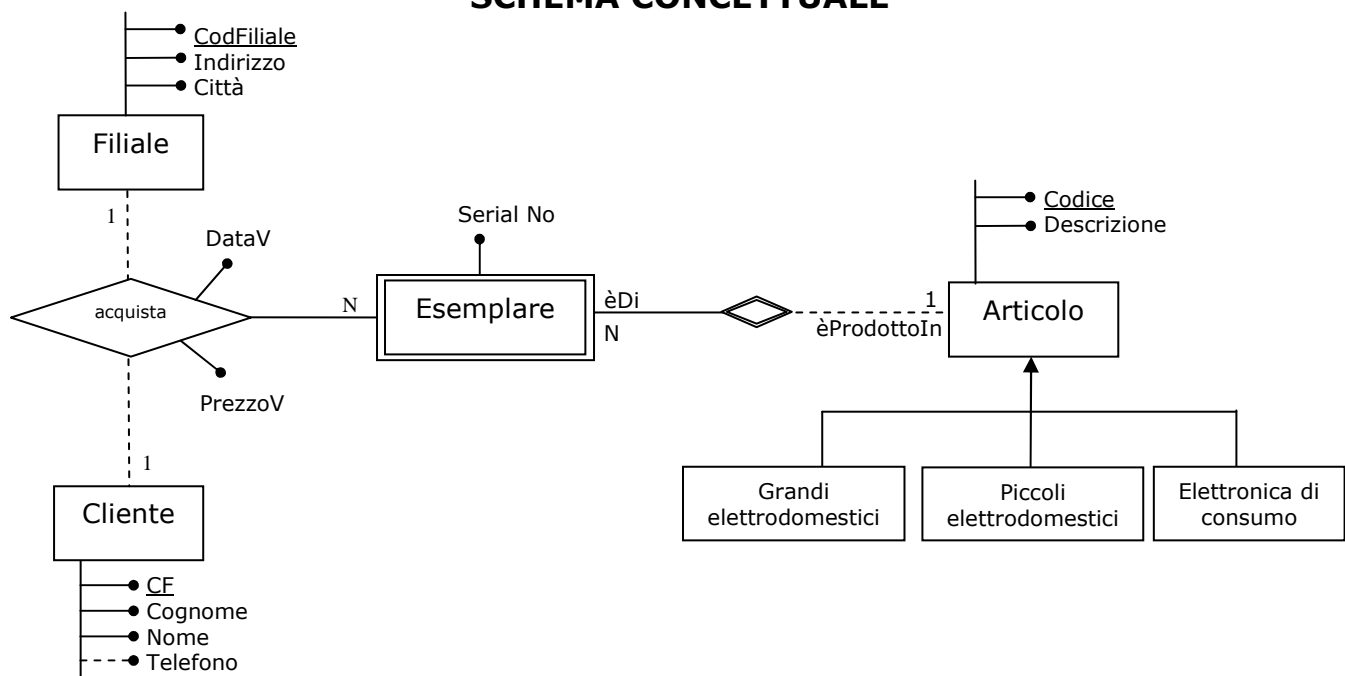
Il gruppo Euronics necessita di un database centrale, accessibile anche in remoto dalle varie filiali, per la gestione delle vendite al dettaglio da parte delle filiali stesse, con riferimento all'esercizio corrente (*anno solare*). A fine esercizio tutti i dati delle vendite, dopo essere stati valutati, saranno cancellati. Ogni **filiale** è caratterizzata da un codice e un indirizzo. Gli articoli sono inquadrati nelle seguenti categorie: Grandi Elettrodomestici, Piccoli Elettrodomestici, Elettronica di Consumo. Di ogni **Articolo** è inoltre rilevante il Codice e la Descrizione. Ogni **esemplare** venduto è identificato da un numero seriale. Dei Clienti vengono considerati rilevanti solo il Codice Fiscale, il Cognome e Nome, e facoltativamente il recapito telefonico. Ogni volta che una filiale vende un articolo a un **cliente**, vanno registrati: la filiale, l'acquirente, l'identificativo dell'articolo acquistato (ai fini della garanzia) nonché la data e il prezzo di vendita.

Progettare la base di dati fino allo Schema Fisico, quindi produrre l'Analisi delle Funzioni completa di Diagramma delle Funzioni e Disegno dell'interfaccia utente: Menu, Maschere per l'immissione / visualizzazione / modifica / cancellazione dei dati e opportuni pulsanti (*almeno una maschera di ciascun tipo, relativa a una tabella referenziante*).

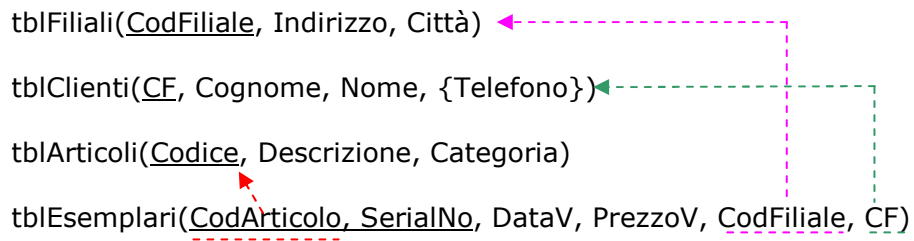
ANALISI DELLA REALTÀ DI INTERESSE

La realtà d'interesse è il gruppo Euronics. Il DB è attuale perché è detto che, a fine esercizio, tutti i dati delle vendite, dopo essere stati valutati, saranno cancellati.
Entità individuate: Filiale, Articolo, Esemplare, Cliente.

SCHEMA CONCETTUALE



SCHEMA LOGICO



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 da 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)

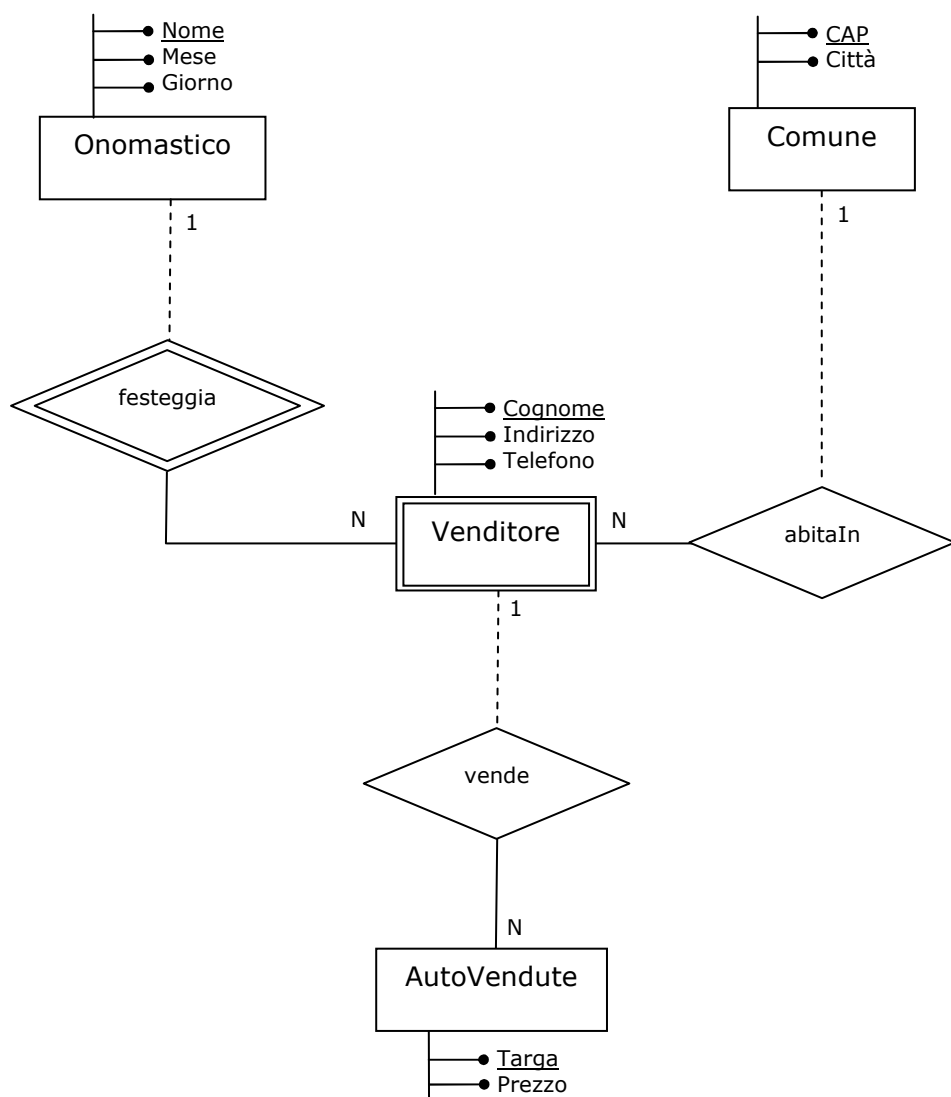
3FN)

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)

Lo schema concettuale correttamente realizzato, che avrebbe dato luogo ad uno schema logico già in forma normale, è il seguente:



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.

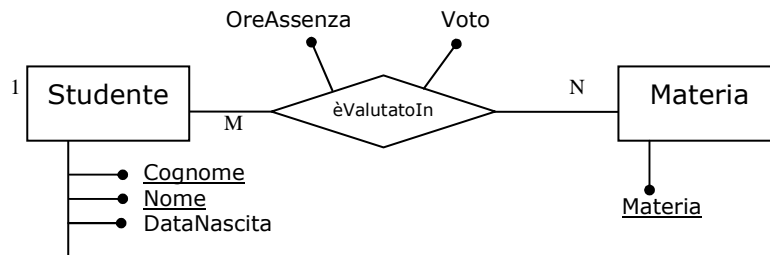
Eseguire le seguenti operazioni:

- 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 (12/2/92)	Italiano, 45, 7 – Matematica, 15, 6 - Inglese, 22, 7 – Informatica, 11, 6
Seccia Vittoria (4/6/90)	Italiano, 25, 6 – Matematica, 21, 5 - Inglese, 15, 5 – Informatica, 21, 4
Dani Alberto (23/3/91)	Italiano, 30, 7 – Matematica, 33, 7 - Inglese, 13, 6 – Informatica, 31, 7
Soleno Tina (15/12/92)	Italiano, 12, 8 – Matematica, 12, 9 - Inglese, 15, 8 – Informatica, 10, 7
Ficco Andrea (5/5/1992)	Italiano, 14, 6 – Matematica, 14, 6 - Inglese, 20, 5 – Informatica, 10, 6
Tiro Giancarlo (21/5/93)	Italiano, 65, 4 – Matematica, 35, 4 - Inglese, 42, 5 – Informatica, 15, 5
Tesore Angela (21/12/91)	Italiano, 12, 7 – Matematica, 12, 6 - Inglese, 12, 5 – Informatica, 25, 5
Piano Guido (5/5/91)	Italiano, 10, 6 – Matematica, 11, 6 - Inglese, 20, 6 – Informatica, 30, 5
Nauri Nando (23/3/93)	Italiano, 31, 7 – Matematica, 23, 6 - Inglese, 10, 6 – Informatica, 21, 8
Citti Alassia (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 (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 sua 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(Cognome, Nome, DataNascita)
tblMaterie(Materia)
tblValutazioni(Cognome, Nome, Materia, OreAssenza, Voto)
```

CREAZIONE DELLE TABELLE

```
CREATE TABLE tbIStudenti
(
    Cognome      Text(20)      NOT NULL,
    Nome         Text(20)      NOT NULL,
    DataNascita  Date NOT      NOT NULL,
    PRIMARY KEY (Cognome, Nome)
);

CREATE TABLE tbIMaterie
(
    Materia      Text(15)      NOT NULL      PRIMARY KEY
);

CREATE TABLE tbIVoti
(
    Cognome      Text(20)      NOT NULL,
    Nome         Text(20)      NOT NULL,
    Materia      Text(15)      NOT NULL REFERENCES tbIMaterie
                                     ON UPDATE CASCADE ON DELETE CASCADE,
    OreAssenza   Byte          NOT NULL,
    Voto         Byte          NOT NULL,
    PRIMARY KEY (Cognome, Nome, Materia),
    FOREIGN KEY (Cognome, Nome) REFERENCES tbIStudenti
                                     ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT 'Voto massimo: 10' CHECK (Voto <=10)
);
```

INSERIMENTO DEI DATI NELLE TABELLE

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

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

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

tblMaterie	
	Materia
+	Informatica
+	Inglese
+	Italiano
+	Matematica

tblStudenti			
	Cognome	Nome	DataNascita
+	Astori	Guido	12/02/1992
+	Citti	Alessia	04/06/1994
+	Dani	Alberto	23/03/1991
+	Ficco	Andrea	05/05/1992
+	Maturi	Nando	23/03/1993
+	Piano	Guido	05/05/1991
+	Seccia	Vittoria	04/06/1990
+	Soleno	Tina	12/12/1992
+	Tesore	Angela	21/12/1991
+	Tiro	Giancarlo	21/05/1993

tblVoti					
	Cognome ▾	Nome ▾	Materia ▾	OreAssenza ▾	Voto ▾
	Astori	Guido	Informatica	11	6
	Astori	Guido	Inglese	22	7
	Astori	Guido	Italiano	45	7
	Astori	Guido	Matematica	15	6
	Citti	Alessia	Informatica	11	9
	Citti	Alessia	Inglese	15	9
	Citti	Alessia	Italiano	15	8
	Citti	Alessia	Matematica	21	7
	Dani	Alberto	Informatica	31	7
	Dani	Alberto	Inglese	13	6
	Dani	Alberto	Italiano	30	7
	Dani	Alberto	Matematica	33	7
	Ficco	Andrea	Informatica	10	6
	Ficco	Andrea	Inglese	20	5
	Ficco	Andrea	Italiano	14	6
	Ficco	Andrea	Matematica	14	6
	Maturi	Nando	Informatica	21	8
	Maturi	Nando	Inglese	10	6
	Maturi	Nando	Italiano	31	7
	Maturi	Nando	Matematica	23	6
	Piano	Guido	Informatica	30	5
	Piano	Guido	Inglese	20	6
	Piano	Guido	Italiano	10	6
	Piano	Guido	Matematica	11	6
	Seccia	Vittoria	Informatica	21	6
	Seccia	Vittoria	Inglese	15	5
	Seccia	Vittoria	Italiano	25	6
	Seccia	Vittoria	Matematica	21	5
	Soleno	Tina	Informatica	10	7
	Soleno	Tina	Inglese	15	8
	Soleno	Tina	Italiano	12	8
	Soleno	Tina	Matematica	12	9
	Tesore	Angela	Informatica	25	5
	Tesore	Angela	Inglese	12	5
	Tesore	Angela	Italiano	12	7
	Tesore	Angela	Matematica	12	6
	Tiro	Giancarlo	Informatica	15	5
	Tiro	Giancarlo	Inglese	42	5
	Tiro	Giancarlo	Italiano	65	4
	Tiro	Giancarlo	Matematica	35	4

QUERY SQL

1. Cognome e nome degli studenti in ordine alfabetico

```
SELECT Cognome, Nome  
FROM tblStudenti  
ORDER BY Cognome, Nome DESC;
```

2. Elenco degli studenti il cui cognome inizia con la lettera T

```
SELECT *  
FROM tblStudenti  
WHERE Cognome LIKE 'T%'  
ORDER BY Cognome, Nome;
```

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

```
SELECT *  
FROM tblStudenti  
WHERE Nome LIKE '%N%';
```

4. Elenco degli studenti nati nel 1993

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

5. Elenco delle materie in ordine alfabetico

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

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

```
SELECT Cognome, Nome, Materia, Voto  
FROM tblVoti  
WHERE Voto<6  
ORDER BY Materia, Cognome, Nome;
```

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: 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 (che lo identifica univocamente).

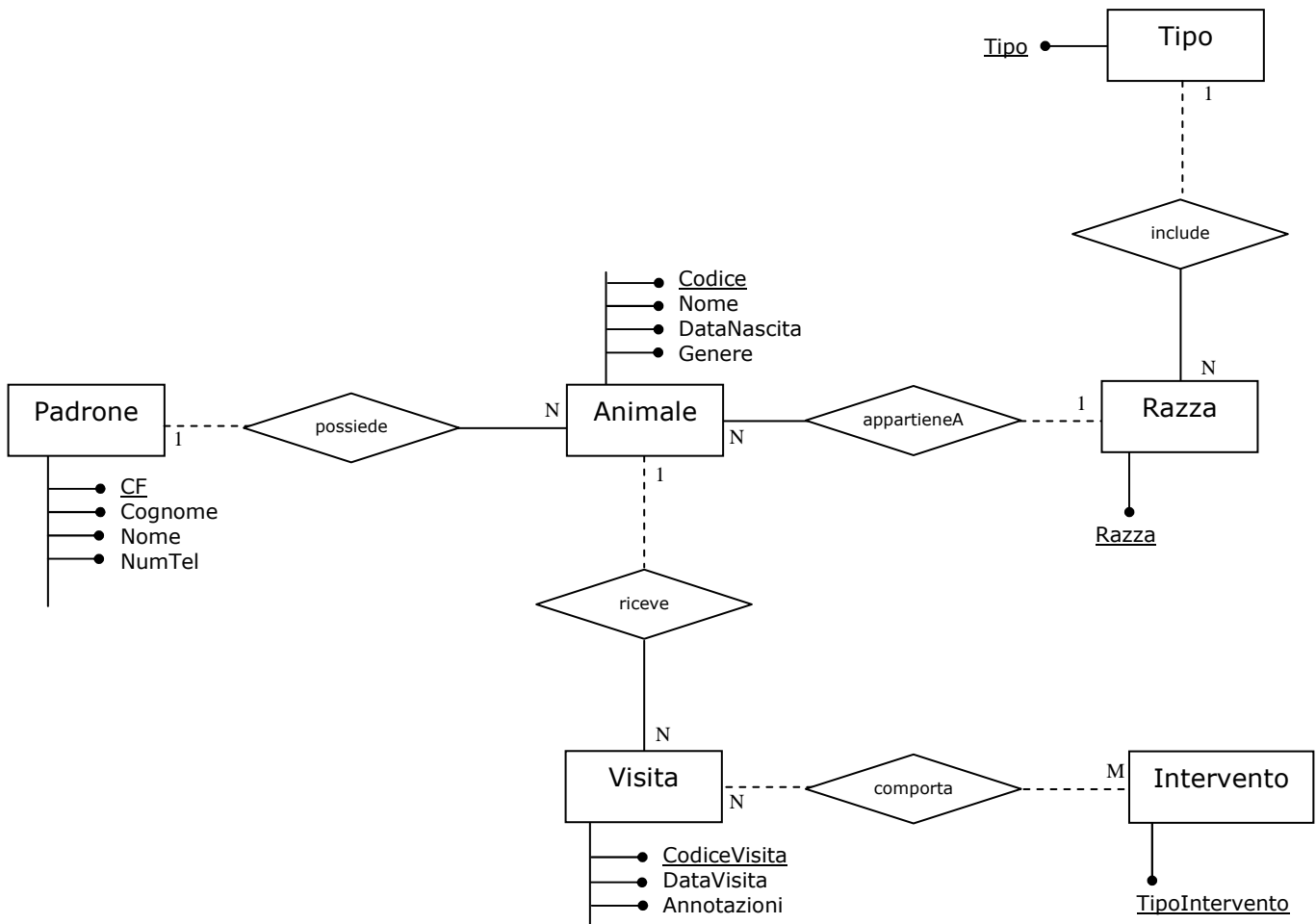
Ad ogni animale si ritiene di dover attribuire un codice (che lo identifichi univocamente). Avrà poi un nome, una data di nascita e un genere (maschio o femmina). Ogni animale appartiene ad un certo tipo (cane, gatto, ecc.) ed è di una certa razza.

È evidente la presenza di relazioni tra le entità Padrone e Animale (1:N), Razza e Animale (1:N), Tipo e Razza (1:N).

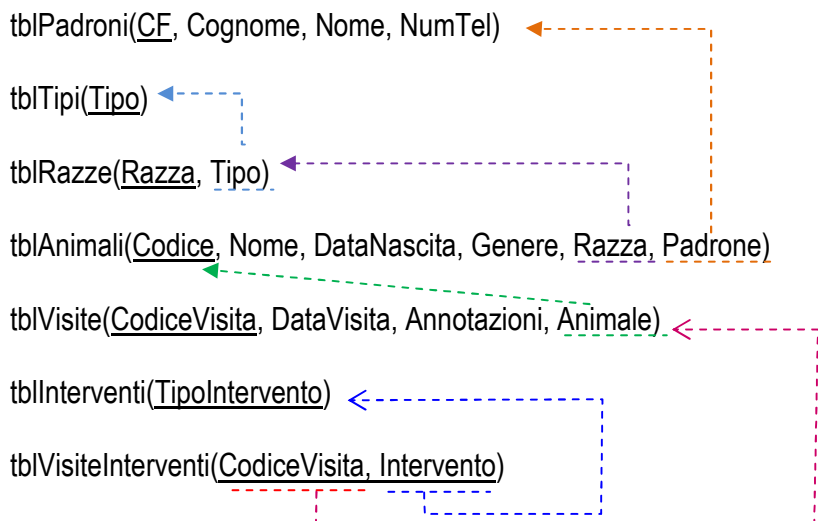
Ad ogni visita si attribuisce un codice, ad esempio un numero d'ordine. Si registrano poi data e annotazioni.

Intervento è una delle possibili operazioni che vengono fatte sull'animale in occasione della visita. C'è un'associazione tra Animale e Visita (1:N) e tra Visita e Intervento (1:N).

1. SCHEMA CONCETTUALE



2. SCHEMA LOGICO



VINCOLI ESPLICITI

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

SCHEMA FISICO

Tabella	Chiave	Nome campo	Tipo	Dimen sione	Richie sto	Valido se	Referenzia
tblPadroni	PK	CF	Testo	16	Sì		
		Cognome	Testo	20	Sì		
		Nome	Testo	20	Sì		
		NumTel	Testo	15	No		
tblTipi	PK	Tipo	Testo	15	Sì		
tblRazze	PK	Razza	Testo	20	Sì		
	FK	Tipo	Testo	15	Sì		tblTipi
tblAnimali	PK	Codice	Testo	4	Sì		
		NomeAni	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
tblVisite	PK	CodiceVisita	Numerico	Intero	Sì		
		DataVisita	Data		Sì		
		Annotazioni	Memo		No		
	FK	Animale	Testo	4	Sì		tblAnimali
tbl Interventi	PK	TipoIntervento	Testo	20	Sì		
tblVisite Interventi	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,
  NumTel      TEXT(15)  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,
  NomeAni      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
);

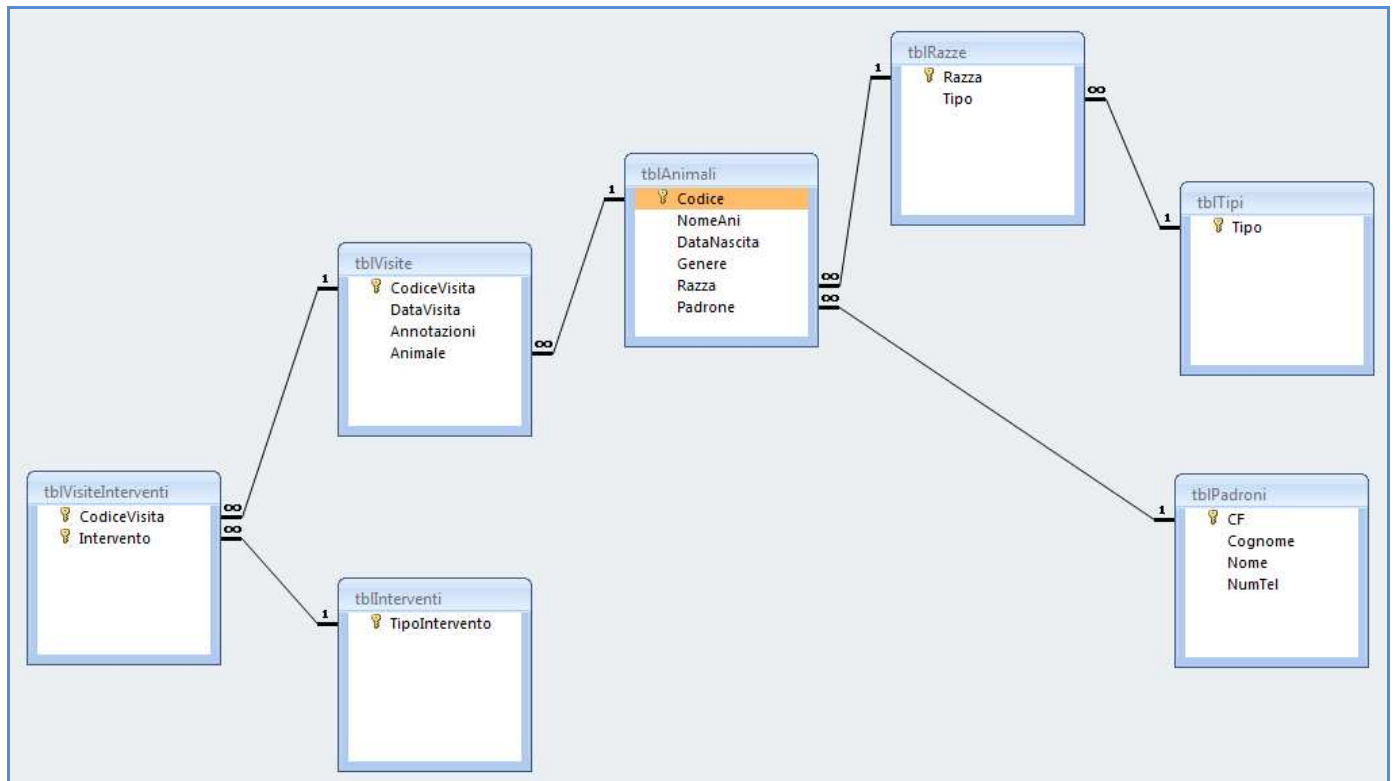
CREATE TABLE tblVisiteInterventi
(
  CodiceVisita Integer   NOT NULL   REFERENCES tblVisite
                                     ON UPDATE CASCADE
                                     ON DELETE CASCADE,
```

```

Intervento          Text(20)  NOT NULL
REFERENCES tblIntervi
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, NomeAni 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, NomeAni 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], NumTel AS [Telefono]
 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;
```