

ESERCIZIO 36 P. A 82 del libro (MULTE)

36

A SCUOLA DI LAVORO



Un piccolo comune deve realizzare una base di dati relativa alle multe per violazioni al Codice della strada. Si intendono gestire dati relativi alle seguenti categorie:

- agenti, descritti da: matricola e nominativo;
- infrazioni, descritte da: codice infrazione, data, agente che l'ha elevata, denominazione dell'infrazione («divieto di sosta», «eccesso di velocità», ecc.), importo e targa dell'auto multata;
- auto, descritte da: targa, marca («Fiat», «Citroen», ecc.), modello («Cinquecento», «Picasso», ecc.), automobilista proprietario;
- automobilisti, descritti da: codice fiscale, nominativo, indirizzo, città, CAP.

Svolgere la progettazione concettuale e logica realizzando:

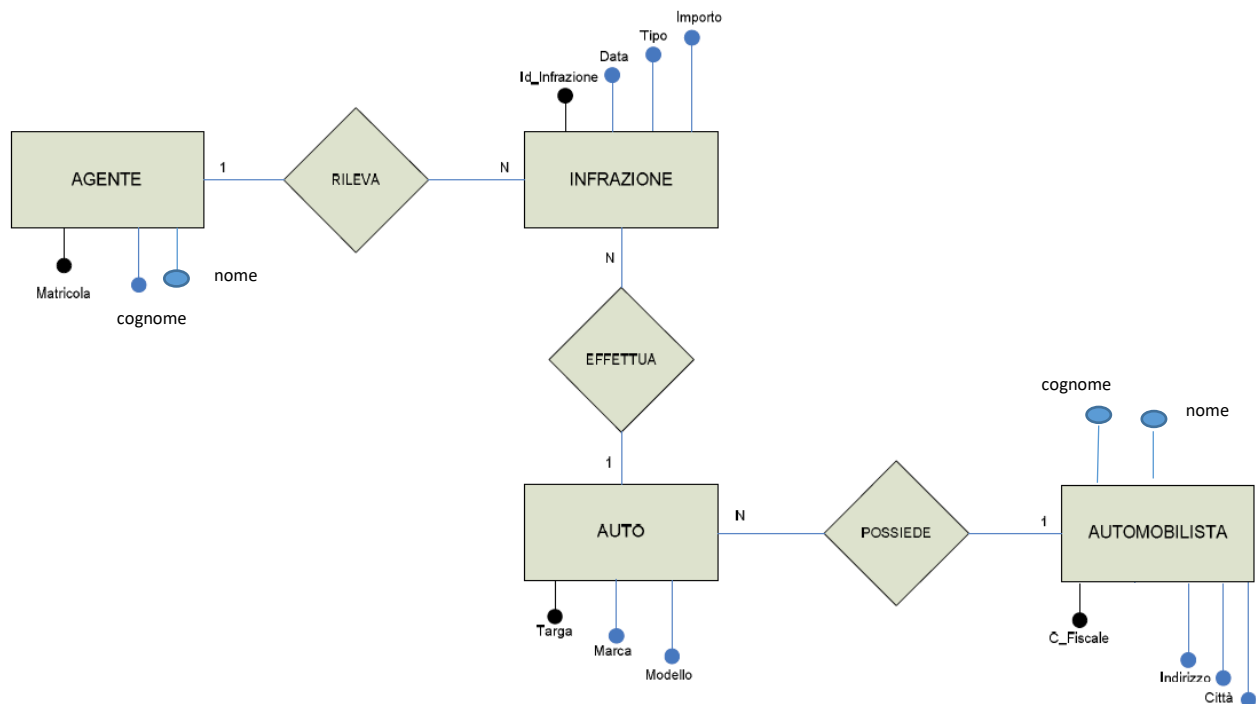
1. Il diagramma E/R
2. Il database schema
3. Le tabelle descrittive di ciascuna relazione
4. Il codice SQL per la realizzazione delle relazioni

Verificare che dal modello logico relazionale realizzato sia possibile determinare le seguenti informazioni:



- elenco delle infrazioni elevate a partire dal 1 gennaio 2000;
- elenco degli agenti che hanno elevato almeno un'infrazione;
- elenco degli automobilisti con i dati dell'auto con la quale hanno subito infrazioni;
- elenco che riporta data dell'infrazione, nome dell'agente che l'ha elevata, targa, modello e marca dell'auto multata.

1 Diagramma E/R



2 DB SCHEMA

AGENTI (matricola, cognome, nome)

AUTOMOBILISTI (codice fiscale, cognome, nome, indirizzo, città)

AUTOMOBILI (targa, marca, modello, *codice_fiscale_proprietario*)

FK *codice_fiscale_proprietario* → AUTOMOBILISTI. codice_fiscale

INFRAZIONI (id_infrazione, data, denominazione, importo, *matricola*, *targa*)

FK *matricola* → AGENTI.matricola

FK *targa* → AUTOMOBILI.matricola

3 TABELLE DESCRITTIVE DELLE RELAZIONI:

AGENTI

Tabella che contiene i dati anagrafici degli agenti

ATTRIBUTO	CHIAVE	TIPO	DIMENSIONE	NOT NULL	NOTE
matricola	PK	Integer auto_increment		X	
cognome		varchar	30	X	
nome		varchar	30	X	

PK: matricola

AUTOMOBILISTI

Tabella che contiene i dati anagrafici degli automobilisti

ATTRIBUTO	CHIAVE	TIPO	DIMENSIONE	NOT NULL	NOTE
codice_fiscale	PK	char	16	X	
cognome		varchar	30	X	
nome		varchar	30	X	
indirizzo		varchar	30	X	
citta		varchar	30	X	
provincia		char	2	X	

PK: codice_fiscale

AUTOMOBILI

Tabella che contiene i dati identificativi di ogni automobile

ATTRIBUTO	CHIAVE	TIPO	DIMENSIONE	NOT NULL	NOTE
targa	PK	char	7	X	
marca		varchar	15	X	
modello		varchar	15	X	
codice_fiscale_proprietario	FK	char	16	X	

PK: targa

FK: codice_fiscale_proprietario → Automobilisti.codice_fiscale

INFRAZIONI

Tabella che contiene i dati identificativi di ogni infrazione commessa

ATTRIBUTO	CHIAVE	TIPO	DIMENSIONE	NOT NULL	NOTE
id_infrazione	PK	integer auto_increment		X	
data		date		X	
denominazione		varchar	255	X	
importo		float		X	
matricola	FK	integer		X	
targa	FK	char	7	X	

PK: id_infrazione

FK: matricola → Agenti.matricola

FK: targa → Automobili.targa

4 CODICE SQL PER LA CREAZIONE DELLE RELAZIONI

Relazione AGENTI:

```
create table agenti(  
    matricola integer AUTO_INCREMENT primary key not null,  
    cognome varchar(30) not null,  
    nome varchar(30) not null  
);
```

Relazione AGENTI

```
create table automobilisti(  
    codice_fiscale char(16) PRIMARY KEY NOT NULL,  
    cognome varchar(30) NOT NULL,  
    nome varchar(30) NOT NULL,  
    indirizzo varchar(30) NOT NULL,  
    citta varchar(30) NOT NULL,  
    provincia char(2) NOT NULL  
);
```

Relazione AUTOMOBILI

```
create table automobili(  
    targa char(7) PRIMARY KEY NOT NULL,  
    marca varchar(15) NOT NULL,  
    modello varchar(15) NOT NULL,  
    indirizzo varchar(30) NOT NULL,  
    codice_fiscale_proprietario char(16) NOT NULL,  
    CONSTRAINT fk_automobili_proprietari FOREIGN KEY(codice_fiscale_proprietario)  
REFERENCES automobilisti(codice_fiscale)  
);
```

Relazione INFRAZIONI

```
create table infrazioni(  
    id_infrazione integer AUTO_INCREMENT PRIMARY KEY NOT NULL,  
    data date NOT NULL,  
    denominazione varchar(255) NOT NULL,  
    indirizzo varchar(30) NOT NULL,  
    importo float NOT NULL,  
    matricola integer NOT NULL,  
    targa char(7) not null,  
    CONSTRAINT fk_infrazioni_agenti FOREIGN KEY(matricola) REFERENCES  
agenti(matricola),  
    CONSTRAINT fk_infrazioni_auto FOREIGN KEY(targa) REFERENCES automobili(targa)  
);
```