

Parte 1: modello relazionale, SQL

Si consideri la seguente base di dati relazionale relativa ad una piccola biblioteca:

TABLE <u>Iscritti</u> (NTessera: integer PRIMARY KEY, Nome: varchar(100) NOT NULL, DataIscrizione: date, Et�: integer, NomeGenitore: varchar(100))	TABLE <u>Libri</u> (Codice: char(6) PRIMARY KEY, Titolo: varchar(200) NOT NULL, PrimoAutore: char(8) REFERENCES Autori(Codice), Donatore: integer REFERENCES Iscritti(NTessera), VietatoMinori: boolean)
TABLE <u>Prestiti</u> (Codice: integer PRIMARY KEY, Iscritto: integer REFERENCES Iscritti(NTessera), Libro: char(6) REFERENCES Libri(Codice), InizioPrestito: date, FinePrestito: date)	TABLE <u>Autori</u> (Codice: char(8) PRIMARY KEY, Nome: varchar(80) NOT NULL, DataNascita: date NOT NULL, DataMorte: date) TABLE <u>Recensione</u> (Codice: char(8) PRIMARY KEY, Iscritto: integer REFERENCES Iscritti(NTessera), Libro: char(6) REFERENCES Libri(Codice), TestoRecensione: varchar(2000), Voto: integer)

NOTE: il nome del genitore viene indicato solo per i minorenni. Il donatore di un libro e l'anno di donazione vengono indicati solo per i libri effettivamente donati da un iscritto. I voti assegnati ai libri vanno da 0 a 10. Laddove serva calcolare la differenza in giorni tra due date D1 e D2, utilizzare l'espressione DATEDIFF(d, D2, D1). Es.: DATEDIFF(d,dataA,dataB)<30.

Si formulino le seguenti interrogazioni tramite il linguaggio SQL oppure l'algebra relazionale.

1. Elencare i libri recensiti eppure mai presi in prestito. **(6 punti)**
2. Elencare gli iscritti che hanno donato almeno 2 libri vietati ai minori. **(7 punti)**
3. Mostrare tutte le coppie di recensioni che assegnano ad uno stesso libro dei voti discordanti di almeno 3 punti (ad esempio un 4 ed un 7). **(7 punti)**
4. Elencare i prestiti durati solo 15 giorni o meno, ma per i quali risulta ugualmente che l'iscritto coinvolto abbia fatto una recensione del libro preso in prestito. **(6 punti)**

Inoltre, si estenda la base di dati (aggiungendo nuove tabelle e/o modificando quelle esistenti):

5. per descrivere gli *incontri di lettura* organizzati, con data e luogo dell'evento, nonch  gli iscritti della biblioteca che vi hanno partecipato. **(6 punti)**

Parte 2: Progetto concettuale e logico, XML

Si considerino i seguenti fatti riguardanti i condomini gestiti da un amministratore:

- ogni *condominio* gestito è caratterizzato dal proprio codice, dall'indirizzo, dal numero di piani, dall'anno di costruzione, oltre che dagli appartamenti di cui è composto;
- ogni *appartamento* è caratterizzato da un proprio codice identificativo di 6 lettere, dal condominio a cui appartiene, dal proprio numero di “interno”, dal piano a cui si trova, dalla propria estensione in metri quadrati, dalla presenza o meno di un garage, dai proprietari dell'appartamento (possono essere anche più di uno);
- ogni *proprietario* di appartamento è caratterizzato dal proprio codice fiscale, dal nome, dagli appartamenti che possiede (limitatamente ai condomini gestiti dal nostro amministratore) e dall'appartamento presso il quale ha la propria residenza – solo nel caso che tale residenza sia effettivamente uno degli appartamenti presenti nella nostra base di dati.

1. Si rappresentino i fatti sopra descritti in uno schema concettuale UML **(9 punti)**
2. Si traduca lo schema concettuale in uno schema relazionale **(9 punti)**
3. Si costruisca un esempio di istanza della base di dati composta da due condomini, di cui uno con 2 appartamenti ognuno posseduto da due proprietari. **(4 punti)**
4. Si costruisca un documento XML relativo al condominio con 2 appartamenti indicato nel punto 3, che rappresenti cioè tutte le informazioni ad esso collegate. **(8 punti)**
5. Si dia un DTD (Document Type Definition = definizione del tipo di documento) per il documento XML del punto 4. **(2 punti)**