

Basi di dati — Homework 3 giugno 2008

L'homework va consegnato secondo modi e termini indicati sul sito del corso.

Domanda 1 Indicare quali delle seguenti affermazioni sono vere e quali false:

1. nei sistemi relazionali, le viste possono essere utili al fine di rendere più semplice la scrittura delle interrogazioni
2. in uno schema concettuale può essere utile introdurre ridondanze al fine di descrivere meglio la realtà di interesse
3. nei sistemi relazionali, le viste possono essere utili al fine di rendere più efficienti le interrogazioni
4. nei sistemi relazionali, le viste introducono ridondanze nei dati memorizzati
5. in uno schema concettuale può essere utile introdurre ridondanze al fine di rendere più efficienti le interrogazioni

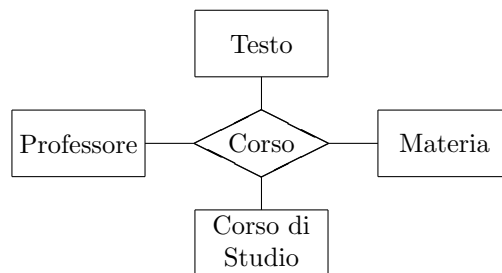
Domanda 2 Mostrare uno schema concettuale che rappresenti una realtà i cui dati siano organizzati per mezzo del seguente schema relazionale.

- CICLISTA(Codice, Cognome, Nome, Squadra)
- COMPETIZIONE(Codice, Nome, Organizzatore, KmTotali)
- TAPPA(Numero, Competizione, Partenza, Arrivo, Km) con vincolo di integrità referenziale fra *Competizione* e la relazione *Competizione*
- CLASSIFICATAPPA(NumeroTappa, Competizione, Ciclista, Posizione, Distacco) con vincoli di integrità referenziale fra gli attributi *NumeroTappa*, *Competizione* e la relazione *Tappa* e fra *Ciclista* e la relazione *Ciclista*
- CLASSIFICAGENERALE(NumeroTappa, Competizione, Ciclista, Posizione, Distacco) con vincoli di integrità referenziale fra gli attributi *NumeroTappa*, *Competizione* e la relazione *Tappa* e fra *Ciclista* e la relazione *Ciclista*

Domanda 3 Modificare lo schema ottenuto in risposta alla domanda precedente, assumendo che

- ciascuna competizione si ripeta ogni anno, con lo stesso organizzatore ma diverso numero di Km totali.
- per ogni località di partenza e arrivo interessi memorizzare la provincia
- ogni squadra abbia una sigla, un nome e un capitano (che è un ciclista)

Domanda 4 A seguito di una prima, superficiale analisi di una realtà di interesse, è stato prodotto lo schema in figura:



Modificare lo schema (decomponendo la relationship ed aggiungendo ulteriori entità, se necessario; indicare le cardinalità delle relationship e eventuali necessità di identificatori esterni) tenendo conto delle seguenti specifiche:

- per ogni materia possono esistere più corsi, tenuti dallo stesso professore o da professori diversi
- ogni corso è relativo ad una e una sola materia
- ogni professore tiene zero o più corsi
- ogni corso ha uno e un solo professore ed è offerto ad uno e un solo corso di studio
- per ogni corso di studio esiste al più un corso di una data materia
- tutti i corsi di una data materia hanno lo stesso libro di testo (uno e uno solo)

Domanda 5 Svolgere quanto richiesto alla domanda precedente, ma con riferimento alle seguenti specifiche:

- per ogni materia possono esistere più corsi, tenuti dallo stesso professore o da professori diversi
- ogni corso è relativo ad una e una sola materia
- ogni professore tiene zero o più corsi
- ogni corso ha uno o più professori ed è offerto ad uno e un solo corso di studio
- per ogni corso di studio esiste al più un corso di una data materia
- ogni corso ha uno e un solo libro di testo; i corsi di una data materia non hanno necessariamente lo stesso libro di testo

Domanda 6 Considerare la base di dati relazionale definita per mezzo delle seguenti istruzioni:

```
create table impiegati (  
    cf numeric not null primary key,  
    titolo char(5),  
    cognome char(20) not null,  
    nome char(20) not null,  
    stipendio numeric not null,  
    dip numeric not null references dipartimenti(codice)  
);  
create table Dipartimenti (  
    codice numeric not null primary key,  
    nomeDip char(20) not null unique,  
    indirizzo char(30) not null );
```

e il metodo mostrato sotto, che avrebbe lo scopo di stampare, per ciascun dipartimento, le informazioni sintetiche (nome e indirizzo) e, subito dopo, la lista degli impiegati.

Riguardo a tale metodo:

1. il codice SQL contiene un errore, a causa del quale la stampa può non avvenire correttamente; individuare tale errore e correggerlo
2. modificare il metodo in modo che stampi anche le informazioni sui dipartimenti che non hanno impiegati (che sono invece ignorati nella versione attuale)

```
static void stampaDati(Connection connection) throws SQLException {  
    Statement statement = connection.createStatement();  
    String query = "select * from dipartimenti JOIN impiegati ON codice = dip " ;  
    ResultSet resultSet = statement.executeQuery(query);  
    boolean primo = true ;  
    int codice = 0;  
    int codicePrec = 0;  
    while (resultSet.next()){  
        primo = false ;  
        codice = resultSet.getInt("codice");  
        if (primo || !(codice==codicePrec)){  
            System.out.println("Dipartimento: " + codice +  
                " Nome: " + resultSet.getString("NomeDip") +  
                " Indirizzo: " + resultSet.getString("Indirizzo") +  
                "\nImpiegati");  
        }  
        String titolo = resultSet.getString("titolo");  
        // Nota bene (dalla documentazione di Java)  
        // boolean wasNull(): reports whether the last column read had a value of SQL NULL.  
        if ((resultSet.wasNull())){ titolo = " ";}  
        System.out.println(" " + resultSet.getInt("cf") + " " + titolo +  
            resultSet.getString("Nome") + " " + resultSet.getString("Cognome"));  
        codicePrec = codice ;  
    }  
}
```