

Basi di Dati – Corso di Laurea in Informatica Umanistica

Esercitazione 3: 3 dicembre 2015

Parte 2 – SQL (aggregati), progettazione concettuale e logica

Docente: Giovanna Rosone

Si considerino i seguenti fatti riguardanti una base di dati relativa ad un'agenzia pubblicitaria.

- L'agenzia è composta da uno staff creativo e da uno staff amministrativo.
- I membri sia dello staff amministrativo che di quello creativo vengono pagati secondo il loro livello di impiego.
- I membri degli staff possono ricevere più livelli durante il loro impiego: è necessario sapere quando un membro dello staff incomincia ad essere pagato per un certo livello e quando finisce di essere pagato per quel livello.
- Bisogna specificare la qualifica dello staff creativo.
- Solo gli amministratori possono ricevere un bonus.
- La retribuzione dei livelli oscilla nel tempo. Ogni livello può avere una retribuzione diversa nel tempo, ma una particolare retribuzione viene usata univocamente per un livello alla volta.
- I Clienti contattano persone dello staff amministrativo e creativo per specificare i requisiti della loro campagna pubblicitaria.
- I Clienti possono lavorare su più campagne pubblicitarie contemporaneamente.
- Lo staff creativo può partecipare a più campagne pubblicitarie.
- Le campagne sono tipicamente prodotte da squadre di impiegati provenienti dallo staff creativo.

- 1) Si definisca uno schema concettuale che rappresenti le informazioni richieste. (Punti 10)
- 2) Si traduca lo schema concettuale del punto 1) in uno schema logico di base di dati nel modello relazionale e creare le tabelle usando SQL. (Punti 8)
- 3) Si risolvano, in riferimento al seguente schema:

```
TABLE Utenti
{Codice integer PRIMARY KEY,
Cognome char(30) NOT NULL,
Nome char(20) NOT NULL,
Sesso char(1) NOT NULL,
Nazionalità char(20) NOT NULL,
AnnoNascita integer NOT NULL,
Città char(20) NOT NULL
}
```

```
TABLE Prodotti
{Codice integer PRIMARY KEY,
Descrizione char(256) NOT NULL,
Categoria varchar(20) NOT NULL,
Prezzo double NOT NULL
}
```

```
TABLE Acquisti
{Codice integer PRIMARY KEY
Prodotto integer REFERENCES Prodotti(Codice),
Utente integer REFERENCES Utenti(Codice) NOT NULL,
DataAcquisto Date NOT NULL
}
```

```
TABLE Recensioni
{Codice integer PRIMARY KEY,
Utente integer REFERENCES Utenti (Codice) NOT NULL,
Prodotto integer REFERENCES Prodotti (Codice) NOT NULL,
Data Date NOT NULL,
Voto integer NOT NULL,
Testo varchar(40) NOT NULL,
Check (Voto>0 AND Voto<6)
}
```

le seguenti interrogazioni utilizzando SQL:

- a. Elenco che, per ogni categoria (diversa da “dvd”), con almeno 4 prodotti, riporta il codice del prodotto e il numero di tali prodotti;
- b. Elenco che riporta, la descrizione del prodotto che ha ricevuto un voto superiore alla media dei voti di tutti i prodotti;
- c. Scrivere una vista che elenchi tutti gli utenti che hanno scritto almeno una recensione e che non hanno effettuato acquisti. Usare le select nidificate. Non usare Join.
- d. Scrivere una vista che per ogni utente (nome, cognome, codice) riporta il numero di acquisti da lui stesso effettuati;
- e. (difficile) Elencare il codice dell’utente che ha effettuato più acquisti.
- f. (difficile) Elencare i prodotti con un prezzo inferiore a tutti i prezzi dei DVD (ovvero dei prodotti con categoria dvd)