

22 GENNAIO 2016

NOME
COGNOME
MATRICOLA

ESERCIZIO 1 (12 punti)

Dato il seguente schema relazionale, che modella i dati di una videoteca:

UTENTE(NumTessera, Nome, Cognome, Città, Età)

NOLEGGIO(NumT, TitoloF, Data, Note)

FILM(Titolo, Genere, Anno, StatoCopia)

Con vincoli di integrità referenziale:

NOLEGGIO.NumT → UTENTE.NumTessera

NOLEGGIO.TitoloF → FILM.Titolo

- a) (3 pt) Scrivere in **SQL** la query che restituisce titolo e genere dei film che non sono stati mai noleggiati nel periodo: 1/1/2016 - 21/1/2016. [VINCOLO: Non è possibile utilizzare il join tra tabelle nella clausola FROM].
- b) (3 pt) Scrivere in **SQL** la query che determina l'età media degli utenti di Bologna che hanno noleggiato almeno un film nel periodo 1/1/2016 al 21/1/2016.
- c) (3 pt) Scrivere in **SQL** la query che determina, per ogni genere presente nella tabella FILM, nome e cognome dell'utente (o degli utenti) che ha (hanno) noleggiato il numero massimo di film appartenenti a quel genere. → GROUP BY
- d) (2 pt) Scrivere in **SQL** il codice della tabella NOLEGGIO, imponendo i seguenti vincoli: (i) Data deve essere sempre definito; (ii) Note è una stringa con lunghezza massima pari a 2000 caratteri, ed è inizializzato a "Nessun Commento". Imporre il seguente vincolo: un utente può noleggiare al massimo 3 film nella stessa data.
- e) (1 pt) Assumendo che FILM sia una Collezione in **MONGO-DB**, scrivere la query MONGO-DB che restituisce titolo e genere (*) dei film che sono stati rilasciati nel 2014 (l'Anno si assume di tipo stringa) ed il cui StatoCopia sia pari a "Eccellente".
(*) restituire solo i campi indicati

ESERCIZIO 2 (12 punti)

Si vuole progettare una base di dati per una piattaforma di gestione di annunci di lavoro nella regione Emilia Romagna. La piattaforma consente la creazione di profili: ogni profilo dispone di nickname (univoco), data creazione, costo attivazione. Inoltre, è prevista la possibilità di inviare messaggi tra profili: ogni messaggio dispone di titolo, corpo, mittente (profilo della piattaforma) e destinatario (altro profilo). Sono previste solo due tipologie di profili: UTENTI o AZIENDA. Un profilo UTENTE dispone di: nome, cognome, data nascita, indirizzo (via/città/CAP), uno o più recapiti telefonici. Si vuole tenere traccia delle -eventuali- esperienze lavorative e dei titoli di studio di ogni UTENTE. Nel dettaglio, per ogni esperienza lavorativa si vuole memorizzare: periodo di inizio, periodo di fine, mansione (campo testo), e descrizione. Per ogni titolo di studio si vuole memorizzare: data conseguimento, nome dell'organizzazione presso il quale è stato conseguito, descrizione, voto. Per i profili di tipo AZIENDA, si vogliono gestire le informazioni seguenti: nome, sede legale, tipologia (campo testo), indirizzo email, link al sito web. Le AZIENDE possono pubblicare annunci di lavoro sulla piattaforma. Ogni annuncio dispone di: numero progressivo, titolo, data inserimento, descrizione. Sono previste solo due tipologie di annunci: annunci con scadenza o annunci aperti. I primi (annunci con scadenza) dispongono anche di: data chiusura, mesi uomo, salario. Gli annunci aperti dispongono di un campo skill (testo). E' prevista la possibilità per gli UTENTI di sottomettere la propria candidatura agli annunci con scadenza (e solo su quelli). Ogni candidatura dispone di una data inserimento e di un campo note (testo), e fa riferimento ad un utente e ad un annuncio. Più utenti possono candidarsi per lo stesso annuncio; viceversa, lo stesso annuncio può ricevere più candidature. Infine, si vuole tenere traccia dei colloqui svolti. Ogni colloquio dispone di una data, un luogo, un esito (testo), e coinvolge un'AZIENDA e un UTENTE registrati sulla piattaforma. Non esistono limiti al numero di colloqui svolti da ciascun UTENTE/AZIENDA.

a) (6pt) Costruire il modello Entità-Relazione (E-R) della base di dati.

b) (4pt) Tradurre il modello E-R nel modello logico relazionale, preferendo la soluzione che minimizzi la presenza di valori NULL nelle tabelle generate. Indicare i vincoli di integrità referenziale tra gli attributi dello schema.

c) (2pt) Indicare quale operazione ha il costo più alto tra quelle elencate sotto:

- Inviare un nuovo messaggio tra due profili presenti nella piattaforma (Interattiva, 50 volte/mese).
 - Visualizzare tutti i titoli di studio e le esperienze lavorative associati ad un dato profilo UTENTE (Interattiva, 2 volte/mese).
 - Rimuovere tutti gli annunci associati ad un dato profilo AZIENDA (Batch, 10 volte/mese).
- Tabella dei volumi: 20 annunci per AZIENDA, 4 titoli studio per UTENTE, 2 esperienze lavorative per UTENTE. α (peso operazioni scrittura)=2, w_1 (peso operazioni interattive)=1, w_B (peso operazioni batch)=0.5

ESERCIZIO 3 (4 punti)

Dato il seguente schema: $R(ABCDE)$, con le seguenti dipendenze funzionali:

$$AB \rightarrow C, BD \rightarrow AE, B \rightarrow AD$$

- a) Indicare se ABD sia chiave o meno della relazione R. **Giustificare la risposta.**
- b) La relazione è in forma normale di Boyce e Codd (FNBC)? La relazione è in terza forma normale (3FN)? **Giustificare le risposte.**

$$ABD_F^+ = \{A, B, D, C, E\} \rightarrow \text{SUPERCHIAVE MA NON MINIMALE} \\ \text{QUINDI NON È CHIAVE.}$$

$$AB_F^+ = \{A, B, C, D, E\}$$

$$AD_F^+ = \{A, D, \}$$

$$BD_F^+ = \{B, D, A, E, C, \}$$

$$B_F^+ = \{B, A, D, C, E\} \rightarrow \text{CHIAVE}$$

• FNBC \rightarrow SI VISTO CHE TUTTE LE DF HANNO A SX UNA SUPERCHIAVE DELLA RELAZIONE

• 3FN \rightarrow SI VISTO CHE IN FNBC

ESERCIZIO 4 (2 punti)

a) (1 pt) Fornire una breve descrizione dell'algoritmo delle K-medie (k-means). A cosa serve? Quali operazioni esegue? (in sintesi)

b) (1 pt) Dato il seguente stato dei log:

CK(T₀, T₁) B(T₂) U(T₂, O₆, B₆, A₆) B(T₃) U(T₃, O₇, B₇, A₇) C(T₂) CK(T₀, T₁, T₃) B(T₄)
 U(T₄, O₂, B₄, A₄) I(T₄, O₈, A₈) C(T₀) B(T₅) I(T₅, O₅, A₅) C(T₅) **guasto**

Applicando l'algoritmo di ripresa a caldo, indicare la composizione degli insiemi UNDO e REDO (ossia, quali sono le transazioni di cui occorre fare l'undo e di quali il redo).

UNDO = { B₄, T₁, O₆, B₆, A₆, T₃, O₇, B₇, A₇, T₄, O₂, B₄, A₄, O₈, A₈, O₅, A₅ }

REDO = { T₅, T₀, T₂ }

CK(T₀, T₁, T₃)

UNDO(T₀, T₁, T₃)

REDO { }

B(T₄) UNDO(T₀, T₁, T₃, T₄)

C(T₀) UNDO(T₁, T₃, T₄)

REDO { T₀ }

B(T₅) UNDO(T₁, T₃, T₄, T₅)

REDO { T₀ }

C(T₅) UNDO(T₁, T₃, T₄)

REDO { T₀, T₅ }