Basi di Dati I, Esempio Seconda Prova Intercorso, 8/12/2022

Mara Sangiovanni

DIETI, Corso di Laurea in Informatica, Università di Napoli 'Federico II', Italy
E-mail: mara.sangiovanni@unina.it

Si consideri il seguente schema relazionale che descrive un frammento della base di dati per gestire le informazioni relative a viaggi in autostrada. Nella tabella tariffe sono indicati i costi dal casello di ingresso al casello di uscita per una determinata categoria di automobili. Un viaggio inizia da un casello in ingresso nel giorno dataI al tempo TempoI e finisce nel casello Uscita nel giorno DataF al tempo TempoF. Per i viaggi iniziati ma non ancora conclusi DataF, TempoF, Uscita e Tariffa sono NULL. La tabella CHECK contiene le rilevazioni dei tutor fatte al passaggio dell'automobile (identificata dalla targa) nel tragitto. Ogni tutor ha una velocità massima consentita descritta nella tabella $PCHECK$. Se non vi sono infrazioni il campo Infrazione ha valore NULL.

$AUTO(\underline{Targa}, CF\_P, Categoria)$
$TARIFFE(Ingresso, Uscita, KM, Categoria, Costo)$
$PCHECK(\underline{PuntoCheck}, VelocitaMax)$
$CHECK(PuntoCheck, Targa, Velocita, Data, Tempo, Infrazione)$
$VIAGGIO(\underline{CodV}, Targa, DataI, DataF, TempoI, TempoF, Ingresso, Uscita, Tariffa, KM)$

Esercizio 01 *(Punti 6), 15 minuti Si scriva il seguente trigger. Quando viene inserito un check per un viaggio si controlla se la velocità rilevata è superiore alla velocità massima. Se è superiore, si pone a TRUE il campo infrazione del CHECK.*

Esercizio 02 *(Punti 6, 15 minuti) Si scriva il seguente trigger. Quando viene aggiornato un viaggio esprimendo un valore per il casello di uscita si aggiornano anche gli attributi Km e Tariffa recuperando i valori dalla tabella TARIFFE (la tariffa dipende da ingresso, uscita e categoria dell'auto).*

Si consideri il seguente schema relazionale che descrive un frammento della base di dati per gestire gli accessi ad una biblioteca digitale di periodici. Le riviste sono identificate dal codice $ISNN$ e sono strutturate in fascicoli (identificati da $CodF$). Ogni fascicolo contiene articoli identificati dall'attibuto Doi. L'utente identificato dal codice fiscale CF ha un profilo associato $Codprofilo$ che regola le possibilità di accesso (numero di articoli consultati al giorno e al mese). In $ACCESSO$ ogni istanza memorizza l'accesso di un utente an un articolo. In $PAROLECHIAVE$ viene memorizzato un insieme di parole chiave significative per una rivista. Le parole chiave sono usate per descrivere astrattamente gli articoli in base al loro contenuto. L'associazione tra parole chiave e articoli viene memorizzata nella relazione $DESCRIZIONE$.

$RIVISTA(\underline{ISNN}, Titolo, Editore, Perodicita)$
$FASCICOLO(\underline{CodF}, Titolo, ISNN, Anno, Numero)$
$ARTICOLO(\underline{Doi}, CodF, Titolo, Autore, Sommario, PagI, PagF)$
$UTENTE(\underline{CF}, email, CodProfilo, Nome, Cognome, DataN)$
$ACCESSO(\underline{CF, Doi}, Data)$
$PROFILO(\underline{CodProfilo}, Tipo, MaxGiorno, MaxMese)$
$PAROLECHIAVE(\underline{Parola}, ISNN)$
$DESCRIZIONE(\underline{Parola}, Doi)$

Esercizio 03 *(Punti 7, 20 minuti) Si implementi un trigger azionato quando viene inserito un nuovo articolo. Il trigger cerca la presenza nel sommario dell'articolo delle parole chiave associate alla rivista dell'articolo. Se viene trovata la presenza di una parola chiave questa viene memorizzata nella tabella $DESCRIZIONE$.*

Esercizio 04 *(Punti 7, 25 minuti) Si scriva una funzione in SQL DINAMICO che riceve in ingresso una stringa di parole chiave separate dal carattere +. La funzione restituisce la stringa di doi degli articoli a cui sono associate TUTTE le parole chiave nella stringa.*

Si consideri il seguente schema relazionale che descrive un frammento della base di dati per gestire depositi di immagini condivise tra utenti. Le foto sono contenute in album. Un album può contenere foto ed altri album (la struttura degli album è analoga a quella di un filesystem con file e cartelle). Ciascuna foto e ciscun album ha un proprietario (owner) espresso da un identificativo di utente. Nella tabella $ALBUM$ l'attributo $InAlbum$ è l'identificativo dell'album contenete (si assuma che un album alla radice della gerarchia di contenimento è contenuto nell'album di sistema di identificativo SYSALBUM). Gli attributi *Aggiunta* e *Rimossa* riportano la data di inserimento della foto e la eventuale data di rimozione (attributo parziale). Foto ed album possono avere dei tag associati. L'elenco dei tag ammissibili si trova nella tabella $HASHTAG$. I tag associati alle foto e agli album si trovano nelle tabelle $TAGFOTO$ e $TAGALBUM$ rispettivamente. La tablella $VISIBLE$ indica invece l'accessibilità dei file ai vari utenti. Il proprietario $CodProp$ dell'album $codA$ concede il diritto di visualizzazione a $CodUt$ su tutte le foto contenute in $codA$. Infine, nella tabella LOG vengono registrate tutte le operazioni fatte sulle foto dagli utenti: $codF$ è la foto su cui si fa l'operazione, $CodU$ l'utente che fa l'operazione ed $Operation$ il tipo di operazione (ad es. inserimento, cancellazione, visualizzazione).

$HASHTAG(parola)$
$FOTO(\underline{CodF}, uri, nome, titolo, owner, CodAlbum, Aggiunta, Rimossa)$
$ALBUM(\underline{CodA}, nome, titolo, owner, InAlbum)$
$UTENTE(\underline{CodU}, nome, cognome, email)$
$TAGFOTO(CodF, parola)$
$TAGALBUM(CodA, parola)$
$VISIBLE(CodProp, CodUt, CodA)$
$LOG(CodF, CodU, Time, Operation)$.

Esercizio 05 *(Punti 10) Si scriva una procedura PLSQL che riceve in ingresso l'identificativo di un album e che restituisce una stringa contenete tutti i tag associati all'album e agli album in esso contenuti (ad ogni livello di profondità) senza ripetizioni. Si consiglia di avvalersi di una tabella $TMP(CodA)$ (che si suppone già definita) dove memorizzare preventivamente l'albero degli album radicato nell'album passato come parametro.*

Esercizio 06 *(Punti 7) Usando SQL DINAMICO si scriva una funzione che riceve in ingresso una lista di tag separati dal carattere @ e che restituisce una stringa degli uri delle foto (separati da @) a cui sono associati tutti i tag passati per parametro.*