

Basi di Dati e Sistemi Informativi I, Prove scritte AA 2007/08

Adriano Peron

Facoltà di Scienze M.F.N., Corso di Laurea in Informatica, Dipartimento di Scienze Fisiche,
Università di Napoli 'Federico II', Italy
E-mail: peron@na.infn.it

1 5 maggio 2008 - Prova intercorso

Si consideri il seguente schema relazionale che descrive la gestione di sale cinematografiche:

SALA(CodSala, Nome, Schermo, Audio, Capienza)

POSTO(CodSala, CodPosto, Fila, NumPosto)

PROIEZIONE(CodicePr, Film, CodSala, Data, OraInizio, OraFine)

BIGLIETTO(CodicePr, CodSala, CodPosto, Prezzo).

SALA, fornisce alcune caratteristiche tecniche della sala di proiezione;

POSTO descrive i singoli posti a sedere di una sala qualificandoli con il codice della sala (CodSala), il numero della fila (Fila), il numero del posto nella fila (NumPosto);

PROIEZIONE descrive le proiezioni dei film nelle sale qualificandoli con il film proiettato, il codice della sala, la data e l'ora iniziale e finale della proiezione;

BIGLIETTO contiene informazione sui posti venduti per ogni proiezione (CodicePr).

Esercizio 11 *Si scriva una espressione in algebra relazionale (senza usare operazioni di conteggio) che, se valutata, fornisce il codice delle proiezioni per le quali tutti i posti disponibili sono stati occupati.*

Esercizio 12 *Scrivere una vista in SQL che per ogni film fornisca il numero delle proiezioni già effettuate (la data è precedente alla data corrente), il numero complessivo degli spettatori, e il numero medio di spettatori per proiezione.*

Esercizio 13 *Si fornisca l'implementazione più ragionevole per il seguente insieme di vincoli:*

1. *Il numero di posti di una sala è uguale alla capienza dichiarata nella tabella SALA;*
2. *Non deve essere emesso un biglietto per una proiezione inesistente o per un posto inesistente;*
3. *Non si può emettere più di un biglietto per una proiezione su di uno stesso posto;*
4. *Le proiezioni di una giornata in una stessa sala non si devono sovrapporre per orario.*

Esercizio 14 *Si supponga di avere una tabella RICAVI(Film, Incasso) dove per ogni film viene indicato l'ammontare dei biglietti venduti per le proiezioni del film. Si scriva un comando SQL che inserisca nella tabella RICAVI una riga per ogni film presente nelle proiezioni che non sia già presente nella tabella stessa inizializzando l'incasso a 0; Si scriva poi un comando SQL che aggiorna la tabella aggiungendo all'ammontare indicato nei vari film la somma ricavata nelle proiezioni dei film della giornata corrente.*

2 16 giugno - compito e II prova intercorso

Per il compito intero si svolgono i primi quattro esercizi.

Per la seconda prova intercorso si svolgono gli ultimi tre esercizi.

Si consideri il seguente schema relazionale che descrive la strutturazione di un documento nelle sue parti (sezioni, sottosezioni, etc) e l'associazione tra parole chiave e le sezioni dove esse sono presenti.

SEZIONE(*Documento*, *CodSezione*, *Titolo*, *Testo*)
STRUTTURA(*Documento*, *SezContenente*, *SezContenuta*, *Posizione*)
PAROLE(*Parola*)
PRESENZE(*Parola*, *Documento*, *Sezione*)

SEZIONE descrive le sezioni dei documenti (ogni sezione ha un titolo ed un testo. Una sezione può essere composta da sezioni. Tale composizione è descritta in *STRUTTURA* dove *SezContenente* è il codice della sezione composta e *SezContenuta* è il codice della sottosezione e *Posizione* è una stringa di caratteri *i* che rappresenta un intero e che indica che *SezContenuta* è la *i*-esima sottosezione di *SezContenente*. *PAROLE* fornisce la lista delle parole interessanti (parole chiave) per il recupero delle sezioni. Se una parola chiave è presente in una sezione, è riportata una riga corrispondente in *PRESENZE* (*Sezione* in *PRESENZE* indica il codice di una sezione).

Esercizio 21 Si scriva una espressione dell'algebra relazionale che se valutata fornisca tutti i documenti in cui sono presenti tutte le parole chiave.

Esercizio 22 Si scrivano nel modo più opportuno i seguenti vincoli:

- se per una sezione è presente una sottosezione in posizione $i > 1$, allora deve essere presente per la stessa sezione anche una sottosezione in posizione $i - 1$ (per semplicità si può assumere per questo esercizio che il tipo di posizione sia intero).
- Nello stesso documento, il titolo delle sezioni è unico.
- Quando viene inserita una nuova parola chiave nella tabella *PAROLA* la tabella *PRESENZE* deve essere aggiornata con la indicazione delle sezioni che contengono la parola.
- La cancellazione di una parola chiave da *PAROLE* deve avere come conseguenza la cancellazione di tutte le indicazioni di presenza di quella parola nelle sezioni.

Esercizio 23 Si scriva un metodo *PLSQL* che riceve in ingresso il nome di un documento e il codice di una sezione e che restituisce una stringa di caratteri con il titolo della sezione ed il titolo di tutte le sue sottosezioni (a tutti i livelli). Per l'ordinamento dei titoli nella stringa di output si consideri il seguente criterio: ad ogni sezione va associata una stringa che rappresenta il cammino nell'albero delle sottosezioni: alla sezione passata per parametro viene associata la stringa vuota; se *s* è la stringa per una sezione *X* che ha una sottosezione *Y* in posizione *h* allora la stringa per *Y* è *s.h*. Per determinare i titoli delle sottosezioni si consiglia di usare una tabella *TEMP*(*documento*, *codiceesez*, *stringa*) che si suppone essere stata già definita.

Esercizio 24 Si consideri la relazione

$R(A, B, C, D)$ con le seguenti dipendenze funzionali:

- $A, B \rightarrow C$;

- $A, B, C \rightarrow D$;
- $C \rightarrow A$;

Trovare le chiavi.

Trovare una copertura canonica per le dipendenze.

Dire quali dipendenze di R soddisfano i requisiti della terza forma normale e quali soddisfano i requisiti della forma normale di Boyce-Code.

Se la tabella non è in terza forma normale scomporla applicando l'algoritmo di normalizzazione.

Esercizio 25 *Si scriva un trigger che quando viene inserita una nuova parola chiave nella tabella $PAROLA$ provveda ad aggiornare la tabella $PRESENZE$ con la indicazione delle sezioni che contengono la parola. Si scriva inoltre un trigger che quando viene inserita una nuova sezione provveda ad aggiornare la tabella $PRESENZE$ relativamente a tutte le parole chiave presenti nella nuova sezione.*

3 21 luglio 2008

Si consideri il seguente schema relazionale che descrive parte dei metadati di uno sche relazionale (la definizione delle tabelle e dei loro attributi, le chiavi primarie ed esterne eventualmente associate alle tabelle).

$TABELLE(\underline{TabId}, Nome, Schema)$
 $ATTRIBUTI(\underline{TabId}, Nome, Tipo, Totale)$
 $PK(\underline{TabId}, Attributo)$
 $FK(\underline{TabId}, NomeFK, RefTabId, OnDel, OnUp)$
 $FKK(\underline{TabId}, NomeFK, Att, RefAtt)$.

$TABELLE$ descrive i nomi delle tabelle definite in uno schema. $ATTRIBUTI$ descrive l'insieme di attributi di ciascuna tabella, indicando il tipo e la totalità dell'attributo (Totale è un booleano). PK indica, se esiste una chiave primaria per una tabella, l'insieme degli attributi che formano la chiave primaria (se non vi è chiave primaria per una tabella non vi sono righe corrispondenti in PK). FK indica le eventuali chiavi esterne per una tabella. Per la chiave esterna viene indicato il nome della tabella referenziata e le azioni OnDelete, OnUpdate da intraprendere in caso di violazione del vincolo di integrità referenziale. Le coppie di attributi corrispondenti (attributo della chiave esterna, attributo referenziato nella tabella referenziata) sono invece riportate nella tabella FKK .

Esercizio 31 *Si scriva una espressione dell'algebra relazionale che se valutata fornisca nome della tabella, numero di attributi presenti nella chiave primaria, numero di chiavi esterne per tabelle in cui tutti gli attributi della chiave primaria sono di tipo integer.*

Esercizio 32 *Si scrivano nel modo più opportuno i seguenti vincoli:*

- Se in FKK vengono messi in corrispondenza due attributi referenziante-referenziato, allora i due attributi devono essere dello stesso tipo e l'attributo referenziato deve far parte della chiave primaria della tabella referenziata.
- Quando viene cancellato un attributo da una tabella, devono essere cancellate tutte le chiavi primarie o esterne che coinvolgono in qualche forma l'attributo (anche referenziandolo).

Esercizio 33 *Si scriva una funzione PLSQL che riceve in ingresso il nome di uno schema e di una tabella e che restituisce una stringa corrispondente alla definizione nello standard SQL della tabella comprensiva delle eventuali chiavi primaria ed esterne.*

Esercizio 34 *Si consideri la relazione*

$R(A, B, C, D, E)$ *con le seguenti dipendenze funzionali:*

- $A \rightarrow B, E;$
- $B \rightarrow A;$
- $A, C \rightarrow D;$
- $B, C \rightarrow D;$

Trovare le chiavi.

Trovare una copertura canonica per le dipendenze.

Dire quali dipendenze di R soddisfano i requisiti della terza forma normale e quali soddisfano i requisiti della forma normale di Boyce-Code.

Se la tabella non è in terza forma normale scomporla applicando l'algoritmo di normalizzazione.

4 17 settembre 2008

Si consideri il seguente schema relazionale che descrive una base di dati per test. La base di dati contiene la descrizione del tipo di test (TIPOTEST) indicando per ogni tipo di domanda (ad esempio, algebra, geometria, logica) quante domande sono presenti (Quantita) di un determinato livello di difficoltà (Livello). Tale composizione si trova nella tabella COMPTEST. La tabella DOMANDA contiene le domande vere e proprie. Le risposte multiple alle domande sono invece memorizzate nella tabella RISPOSTA. (Per ogni risposta multipla prevista per una domanda vi è una riga corrispondente nella tabella RISPOSTA e la risposta corretta ha l'attributo Corretta con valore assegnato TRUE). La tabella ISTANZE contiene istanze del test. Ogni istanza di test ha un numero progressivo univoco (NTest), ha una tipologia di test associata (CodiceT) ed ha associato un insieme di domande presenti nella tabella DOMANDE.

TIPOTEST(CodiceT, Descrizione)
COMPTEST(CodiceT, Tipodomanda, Quantita, Livello)
DOMANDA(CodiceD, Tipodomanda, Livello, Testo)
RISPOSTA(CodiceD, Risposta, Corretta)
ISTANZE(NTest, CodiceT, CodiceD).

Esercizio 41 *Si scriva una espressione dell'algebra relazionale che se valutata fornisca i codici di istanze di test in cui tutte le domande prevedono lo stesso numero di risposte.*

Esercizio 42 *Si scrivano nel modo più opportuno i seguenti vincoli:*

- Per ogni domanda ci deve essere una ed una sola risposta indicata come corretta.
- Ogni codice test presente in ISTANZE deve corrispondere a un codice di test presente e ogni codice domanda deve corrispondere a un domanda presente.
- Ogni istanza di test deve rispettare le regole di composizione per il tipo di test di cui è istanza.

Esercizio 43 *Si scriva una procedura PLSQL che riceve in ingresso il codice di un tipo di test e che ha come effetto la costruzione di una istanza di test (inserimento della composizione della istanza nella tabella ISTANZE). Il numero della istanza di test generato deve essere ottenuto incrementando di 1 il più grande numero di istanza di test già memorizzata. Per semplicità, se la tipologia di test prevede un numero n di domande di un certo tipo e di un certo livello, si associano al test le prime n domande di quel tipo e di quel livello ottenibili interrogando opportunamente la tabella delle domande.*

Esercizio 44 *Alla luce del problema considerato, si consideri la tabella ISTANZE. Per la tabella si introduca un insieme di dipendenze funzionali ragionevoli rispetto alla descrizione del problema. Si indichino le chiavi, si dica se la tabella è in terza forma normale e in forma di Boyce-Codd. Se non è in una delle due forme normali, si applichi l'algoritmo di normalizzazione.*

5 7 gennaio 2009

Si consideri il seguente schema relazionale che descrive le tabelle definite in un database e le associazioni esistenti tra le tabelle.

TABELLE(CodTab, Nome)

ATTRIBUTI(CodTab, Nome, Tipo, Totale)

CHIAVI(CodTab, NomeAttrib)

ASSOCIAZIONI(CodAss, Nome, Grado, Creato)

COMPASSOC(CodAss, CodTab, Ruolo, Cardinalita)

TABELLE descrive le tabelle esistenti nel database. *ATTRIBUTI* descrive gli attributi associati ad ogni tabella. Per ogni attributo viene fornito il nome, il tipo e la indicazione di parzialità (Totale =0) o totalità (Totale=1). *CHIAVI* indica quali sono gli attributi che determinano la chiave primaria della tabella. *ASSOCIAZIONI* indica quali sono le associazioni presenti tra le tabelle (si pensi alle associazioni di un Class Diagram tra le classi o le associazioni tra le Entità in uno schema Entità-Relazioni). Grado indica il grado dell'associazione e Creato è un attributo che indica se l'associazione è stata già codificata nel database (Creato=1) oppure se deve essere ancora creata (Creato=0). Per ogni legame di una associazione verso una tabella è presente nella tabella *LEGASSOCIAZIONI* una riga che indica la tabella verso cui si stabilisce il legame, il nome del ruolo del legame, e la cardinalità del legame (il grado di partecipazione di una Tabella all'associazione). I valori per la cardinalità sono 1 o M (molti).

Esercizio 51 (8 punti) *Si scriva una interrogazione in algebra relazionale che, se valutata, fornisce il nome delle tabelle che NON sono coinvolte in nessuna associazione ricorsiva (è ricorsiva quando la tabella partecipa con almeno due ruoli distinti alla associazione).*

Esercizio 52 (8 punti) *Si scriva una interrogazione in SQL che restituisca le coppie delle tabelle che NON sono legate da alcuna associazione binaria.*

Esercizio 53 (8 punti) *Si scriva un Trigger che viene attivato quando l'attributo Creato di ASSOCIAZIONI viene posto da 0 a 1. L'effetto è quello di creare una tabella per l'associazione, se si tratta di una associazione molti a molti o di grado maggiore a 2. Gli attributi della tabella creata sono gli attributi chiave delle tabelle coinvolte dall'associazione. Per creazione di una tabella si intende l'inserimento della sua descrizione in TABELLE ed ATTRIBUTI. Se si tratta di una associazione 1 a molti si aggiungono gli attributi chiave della tabella che partecipa con 1 agli attributi della tabella che partecipa con molti*

Esercizio 54 (8 punti) *Si consideri la relazione ORGANICO(Qualifica, Specializ, Anzianita, Stipendio, Reparto) che descrive lo schema di retribuzione delle figure professionali di una azienda. La relazione gode delle seguenti dipendenze logiche:*

- Specializzazione \rightarrow Reparto;
- Qualifica, Specializzazione, Anzianita, Reparto \rightarrow Stipendio;

Trovare le chiavi.

Trovare una copertura canonica per le dipendenze.

Dire se ORGANICO è in terza forma normale e se è in forma normale di Boyce-Code.

Se la tabella non è in terza forma normale scomporla applicando l'algoritmo di normalizzazione.

6 29 gennaio 2009

Si consideri il seguente schema relazionale che descrive la usuale strutturazione logica di folder e file in un file system.

FOLDER(*CodFolder*, *Nome*, *Visibile*, *InFolder*)
ABILITAZIONE(*CodUt*, *CodFolder*)
FILE(*CodFile*, *Nome*, *estensione*, *dimensione*, *data*, *InFolder*)
Utente(*CodUt*, *Nome*, *Cognome*, *Log*, *Passwd*).

FOLDER descrive le cartelle. Una cartella può essere contenuta in un'altra cartella (att. *InFolder* che risulta nullo solo se il folder risulta essere la radice della gerarchia). L'attributo *Visibile* può avere valore 'ALL' (il suo contenuto è visibile a tutti) oppure 'PRIVATE' (il suo contenuto è visibile ai soli utenti abilitati). Gli utenti abilitati ad accedere ai vari folder sono elencati nella tabella *ABILITAZIONE*. Un utente abilitato ad accedere ad un folder è abilitato ad accedere a tutti i file del folder e a tutti i folder visibili contenuti nel folder.

Esercizio 61 *Si scriva una espressione dell'algebra relazionale che se valutata fornisca il nome dei folder PRIVATE che garantiscono comunque l'accesso a tutti gli utenti.*

Esercizio 62 *Si scriva in SQL una interrogazione che restituisca lo stesso risultato dell'esercizio precedente.*

Esercizio 63 *Si scriva un metodo PLSQL che riceve in ingresso il codice di un folder e che restituisce lo spazio di memoria occupato dai file contenuti a tutti i livelli nel folder stesso. Per determinare il valore richiesto si consiglia di usare una tabella temporanea TEMP per memorizzare i risultati intermedi del conteggio (la struttura di TEMP può essere liberamente scelta a seconda delle necessità).*

Esercizio 64 *Si scriva un trigger che viene azionato quando viene revocato ad un utente il permesso di accesso ad un folder (cancellazione della riga in ABILITAZIONE). Il trigger provvede a revocare allo stesso utente le eventuali abilitazioni che esso possiede nei folder contenuti (a tutti i livelli) nel folder in questione. Come nel precedente esercizio si consiglia di usare una tabella temporanea.*