

# Basi di Dati e Sistemi Informativi I, Prove scritte

## AA 2010/11

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](mailto:peron@na.infn.it)

### 1 Scritto del 2 febbraio 2010

Si consideri il seguente schema relazionale che descrive un database per la gestione di condivisione di file in uno schema peer to peer.

*UTENTE*(*UsrId*, *MaxUp*, *MaxDown*, *Stato*)

*FILE*(*UsrId*, *IdFile*)

*SCARICO*(*IdFile*, *UsrUp*, *UsrDown*)

*CODA*(*IdFile*, *UsrDown*, *Time*)

*UTENTE* fornisce il numero massimo di operazioni di Upload e Download simultaneamente possibili per un utente e il suo stato che può assumere lo stato *On* o *Off*. *FILE* indica i file di cui gli utenti sono in possesso. *SCARICO* indica quali sono le operazioni di scarico in corso con indicazione dell'utente che fa l'upload ed il download. *CODA* indica le richieste in attesa di scarico, con *Time* il tempo in cui la richiesta è stata inoltrata.

**Esercizio 11** *Si scriva una espressione dell'algebra relazionale che se valutata fornisca lo userid degli utenti che hanno il maggior numero di download in corso.*

**Esercizio 12** *(Facoltativo) Si scriva una vista SQL che per ogni utente fornisca il numero di upload e download in corso, il numero di file in attesa di download e il tempo massimo di attesa.*

**Esercizio 13** *Si scrivano nel modo più opportuno i seguenti vincoli:*

- *nella tabella CODA non ci possono essere più richieste dello stesso utente per lo stesso file.*
- *un utente non può avere un numero di upload in corso superiore al suo MaxUp.*
- *Quando un utente passa da stato On a stato Off tutte le operazioni di scarico che lo coinvolgono sono cancellate.*
- *I file sono associati ad utenti presenti in UTENTE.*

**Esercizio 14** *Quando un utente commuta il suo stato da Off a On i file che possiede divengono disponibili. Di conseguenza un trigger cerca di attivare il maggior numero di operazioni di scarico presenti nella tabella CODA usando le disponibilità dell'utente secondo i seguenti criteri: Hanno precedenza le richieste più vecchie; le richieste riguardano file posseduti dall'utente; il numero di operazioni di carico attivate non può superare il MaxUp dell'utente; una richiesta non può essere attivata se l'utente ha già in corso un numero di download pari al numero massimo a lui consentito.*

*Attivare una richiesta significa inserire una riga corrispondente nella tabella SCARICO e rimuovere la richiesta dalla tabella CODA. Scrivere il trigger in questione.*

**Esercizio 15** *Si consideri lo schema relazionale*

$R(A, B, C, D)$  *La relazione gode delle seguenti dipendenze logiche:*

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

*Dire quali dipendenze dello schema soddisfano la terza forma normale e quali la forma normale di Boyce-Codd. Ridurre lo schema in forma normale di Boyce-Codd.*

## 2 Scritto del 3 marzo 2010

Si consideri il seguente schema relazionale che descrive un database per la memorizzazione di partite in un gioco per scacchiera.

$PEZZO(\underline{CodiceP}, Tipo, Colore, iX, iY)$

$MOSSA(\underline{Partita}, \underline{Ordine}, daX, daY, aX, aY)$

$ELIMINA(\underline{Partita}, \underline{Ordine}, CodiceP)$

$PEZZO$  fornisce la descrizione dei pezzi del gioco ( $iX$  e  $iY$  indicano le coordinate della posizione iniziale del pezzo sulla scacchiera).

$MOSSA$  memorizza le mosse per ogni partita:  $Ordine$  è un intero che indica l'ordine della mossa nella partita;  $daX$  e  $daY$  sono le coordinate della casella di partenza e  $aX$ ,  $aY$  sono le coordinate della casella di arrivo.  $ELIMINA$  indica i pezzi eliminati durante la partita:  $ordine$  indica la mossa che ha portato alla eliminazione.

**Esercizio 21** *Si scriva una espressione dell'algebra relazionale che se valutata fornisca tutte le partite in cui sono stati eliminati tutti i pezzi neri.*

**Esercizio 22** *Si scriva una vista SQL che risponda al seguente schema logico:  $VISTA(Partita, Orizzontali, Verticali)$ .  $Orizzontali$  e  $Verticali$  indicano il numero delle mosse orizzontali e verticali della partita. Nella vista devono comparire solo le partite che hanno un numero di mosse superiore o uguale alla media delle mosse di tutte le partite.*

**Esercizio 23** *Si scriva una procedura PLSQL che riceve in ingresso il codice di una partita e l'ordine di una mossa. La procedura deve restituire una stringa di terne che descrivono lo stato della scacchiera nella partita al compimento della mossa di ordine fornito. Le terne hanno la forma  $(X, Y, occ)$  dove  $X$  e  $Y$  sono le coordinate della casella e  $occ$  vale  $NULL$  se la casella è vuota e contiene il codice di un pezzo se, invece, la casella è occupata da un pezzo. (Si osservi che, una casella  $X, Y$  contiene un pezzo alla fine della mossa  $i$  se il pezzo è posto in quella casella da una mossa  $j$  con  $j < i$  e nessuna altra mossa  $k$  con  $j < k < i$  ha spostato il pezzo o ha eliminato il pezzo mettendoci un altro pezzo al suo posto).*

**Esercizio 24** *Si consideri lo schema relazionale*

$R(A, B, C, D)$  *La relazione gode delle seguenti dipendenze logiche:*

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

*Dire quali sono le chiavi. Dire quali dipendenze dello schema soddisfano la terza forma normale e quali la forma normale di Boyce-Codd. Ridurre lo schema in forma normale di Boyce-Codd.*

### 3 Scritto del 9 aprile 2010

Si consideri il seguente schema relazionale che descrive un frammento dei metadati di Oracle 10g.

*UserTables*(*TableName*)

*UserTabColumns*(*TableName*, *ColumnName*, *DataType*, *Position*)

*UserConstraints*(*TabelName*, *ConstraintName*, *ConstraintType*, *SearchCondition*, *RConstraintName*)

*UserConsColumns*(*ConstraintName*, *TableName*, *ColumnName*, *Position*)

*UserTabColumns* descrive gli attributi associati alle colonne (nome colonna e tipo di dato e posizione). *UserConstraints* descrive i vincoli associati ad ogni tabella. In particolare, *ConstraintType* assume il valore C per vincolo Check (l'attributo *SearchCondition* contiene il vincolo associato al Check), P per vincolo PRIMARY KEY, R per vincolo FOREIGN KEY, U per vincolo UNIQUE. Un vincolo FOREIGN KEY in una tabella A fa riferimento ad un corrispondente vincolo PRIMARY KEY della tabella referenziata B mediante l'attributo *RConstraintName* (indica il nome del vincolo di chiave primaria nella tabella B a cui la chiave esterna della tabella A fa riferimento). Si assuma al fine dell'esercizio che il nome del vincolo sia chiave. *UserConsColumns* indica il nome degli attributi (e la loro posizione nel vincolo) che formano vincoli di tipo P,R e U.

**Esercizio 31** *Si scriva una interrogazione in algebra relazionale che, se valutata, fornisca il nome delle tabelle che hanno il massimo numero di attributi di tipi INTEGER nella chiave primaria.*

**Esercizio 32** *Si scriva una interrogazione in SQL che restituisca le coppie dei nomi delle tabelle che sarebbero compatibili per operazioni insiemistiche (tutti gli attributi in uguale posizione devono avere lo stesso tipo).*

**Esercizio 33** *Si scriva una funzione che prende in ingresso il nome di una tabella e restituisce una stringa contenente la dichiarazione secondo la sintassi di SQL dei vincoli PRIMARY KEY ed UNIQUE presenti nella tabella (si ricordi che in una tabella può esserci più di un vincolo UNIQUE).*

**Esercizio 34** *Si consideri la relazione  $R(A, B, C, D)$ . La relazione gode delle seguenti dipendenze logiche:*

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

*Trovare tutte le chiavi. Dire se lo schema soddisfa la forma normale di Boyce Codd e in caso negativo applicare l'algoritmo di scomposizione (Boyce Codd) fino ad ottenere la forma normale. Dire se la forma normale ottenuta preserva i dati e le dipendenze della forma di partenza (si dica preliminarmente che cosa significa preservare dati e dipendenze).*

## 4 Scritto del 22 giugno 2010

Si consideri il seguente schema relazionale che descrive un data base per articoli scientifici su rivista.

*RIVISTA*(ISNN, Nome, Editore)  
*ARTICOLO*(CodA, ISNN, Titolo, Volume, Numero, Anno, pagI, pagF)  
*AUTORE*(CodAT, Cognome, Nome, Istituto)  
*SCRIVE*(CodA, CodAT, posizione)  
*REFERENZA*(CodA, CodARiferito)

*RIVISTA* descrive gli attributi della rivista. *ARTICOLO* raccoglie istanze di articolo. Un articolo è associato ad una rivista tramite la chiave esterna ISNN. *AUTORE* raccoglie le istanze degli autori di articoli. L'associazione tra articoli e autori viene stabilita dallo schema *SCRIVE*, dove CodA è chiave esterna verso un articolo, CodAT verso autore, e *posizione* è un intero che indica l'ordine dell'autore nella lista degli autori di un articolo (primo autore, secondo autore...etc). Lo schema *REFERENZA* indica gli articoli citati da un articolo di codice *CodA* (CodA riferenzia CodARiferimento).

**Esercizio 41** Si scriva una interrogazione in algebra relazionale che, se valutata, fornisce il titolo degli articoli in cui gli autori compaiono in ordine alfabetico.

**Esercizio 42** Si implementino in modo opportuno i seguenti vincoli:

1. Due articoli pubblicati nello stesso numero dello stesso volume della stessa rivista non possono sovrapporsi nel numero delle pagine.
2. Due autori del medesimo articolo non possono stare nella stessa posizione nell'elenco degli autori.
3. le referenze fanno riferimento ad articoli esistenti nel database.
4. un articolo che cita un altro articolo non può essere pubblicato in un anno precedente a quello dell'articolo citato.

**Esercizio 43** Si supponga che sia stata già definita una tabella

*CITAZIONI*(CodA, NumCitPure, NumCitSelf)

dove CodA è un articolo, NumCitPure è il numero di volte che viene citato tramite una referenza da un articolo che non ha autori in comune con gli autori di CodA e NumCitSelf è il numero di volte che viene citato tramite una referenza da un articolo che ha autori in comune con gli autori di CodA. Si scrivano dei trigger che realizzino i seguenti vincoli. Quando viene inserito un articolo viene inserito in *CITAZIONI* una riga corrispondente con il numero di citazioni inizializzato a 0. Quando viene inserita una riga in *REFERENZE* viene aggiornato il conteggio delle citazioni in modo opportuno.

**Esercizio 44** Si scriva una funzione PLSQL che dato in ingresso il codice di un autore restituisca in uscita la descrizione (stringa di caratteri) dei suoi articoli (in ordine crescente di anno). Per ciascun articolo si indichino gli autori, il titolo, il nome della rivista, volume, numero, anno e pagine.

## 5 Scritto del 16 luglio 2010

Si consideri il seguente schema relazionale che descrive un database per questionari a risposta multipla (test).

*TEST*(CodT, Data, Utente, Risultato)

*DOMANDE*(CodD, domanda, tipo, livello)

*RISPOSTE*(CodD, ordine, corretta, risposta)

*COMPTEST*(CodT, CodD, Ordine, OrdineRisposta).

*TEST* memorizza collezioni di istanze di questionari somministrati ad utenti. Risultato è attributo parziale (viene associato un valore solo alla correzione del questionario). *DOMANDE* memorizza una collezione di domande con le quali i test sono composti. Per ogni domanda vi sono più risposte (*RISPOSTE*) e delle risposte una soltanto è corretta (attributo corretto con valore 1). Ordine (A,B,C etc ....) indica l'ordine in cui viene presentata la risposta. *COMPTEST* descrive la effettiva composizione di una istanza di questionario: CodT indica la istanza di questionario, CodD la domanda che vi fa parte, Ordine (1,2,3...) indica l'ordine di presentazione della domanda, OrdineRisposta indica la scelta dell'utente tra le risposte per la domanda.

**Esercizio 51** *Si scriva una espressione dell'algebra relazionale che se valutata fornisca i codici dei test in cui vi sia al più una domanda sbagliata tra le domande dello stesso tipo.*

**Esercizio 52** *Si immagini che i test vengano prima predisposti nel database lasciando le risposte a NULL e che le risposte vengano successivamente caricate aggiornando l'attributo Risposta. Si scriva un trigger il quale provvede ad aggiornare il campo risultato nella tabella TEST quando vengono caricate le risposte.*

*Per il calcolo del punteggio si segua la regola: la risposta non data vale 0, la risposta corretta vale 1, la risposta sbagliata vale - (1 / numerorisposte). Si ragioni su come contenere il numero di volte in cui si fa l'aggiornamento dell'attributo risultato suggerendo quale strategia andrebbe adottata.*

**Esercizio 53** *Si scriva una funzione che riceve in ingresso il codice di un questionario e fornisce in uscita il testo del questionario. Le domande devono essere nell'ordine previsto. Le risposte a ciascuna domanda devono essere nell'ordine previsto. La risposta scelta dall'utente deve avere come prefisso i caratteri '(X)'.*

**Esercizio 54** *Si consideri lo schema relazionale*

*R(A, B, C, D) La relazione gode delle seguenti dipendenze logiche:*

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

*Dire quali dipendenze dello schema soddisfano la terza forma normale e quali la forma normale di Boyce-Codd. Ridurre lo schema in forma normale di Boyce-Codd.*

## 6 Scritto del 15 settembre 2010

Si consideri la bozza di Class Diagram che descrive l'anagrafe di una città. Una persona può aver avuto nella sua vita diverse residenze, qualificate da una data di inizio (DATAI) e da una data di fine (DATAF). Per la residenza attualmente in corso la DATAF ha valore nullo. Per persone viventi la data di morte (DATAM) ha valore nullo. Nella tabella *STATOF* viene indicata la composizione del nucleo familiare: CapoF indica il CF del capofamiglia, membro indica il CF del membro del nucleo familiare e relazione indica la relazione che lega il Membro a CapoF: coniuge, figlio, padre, etc.

$VIA(\underline{CodV}, Nome)$   
 $RESIDENZA(\underline{CF}, DATAI, DATAF, CodV, Numero)$   
 $RESIDENTE(\underline{CF}, Nome, Cognome, DataN, DataM)$   
 $STATOFAM(CapoF, Membro, relazione)$

**Esercizio 61** (Punti 8) Scrivere una espressione in algebra relazionale che fornisca per ogni via, il numero delle persone viventi che hanno avuto sempre la residenza in quella via (attenzione, potrebbero eventualmente aver cambiato il numero civico di residenza).

**Esercizio 62** (Punti 8) Si implementino nel modo più opportuno i seguenti vincoli:

- Quando viene apposta la data di morte di un residente viene anche contestualmente chiusa con la stessa data la sua residenza.
- Tutti i membri dello stesso nucleo familiare devo avere la medesima residenza.
- Una persona può avere al più una residenza correntemente attiva.
- Una persona può essere a capo di un unico nucleo familiare.

**Esercizio 63** (Punti 8) Si scriva una procedura PLSQL che riceve in ingresso il codice di una via e un numero civico e che restituisce una stringa contenente la descrizione dei nuclei familiari residenti a quell'indirizzo. L'elenco deve essere ordinato lessicograficamente per cognome-nome del capo famiglia. Per ogni nucleo, va indicato l'elenco (cognome-nome-età) dei membri del nucleo familiare ordinato in modo decrescente per età.

**Esercizio 64** (Punti 8) 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.