

# **Basi di Dati e Sistemi Informativi I, Prove scritte AA 2006/07**

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 27 marzo 2008

Si consideri il seguente schema relazionale che descrive la strutturazione di un libro.

*LIBRO*(*ISBN*, *Titolo*, *Anno*)

*SEZIONI*(*ISBN*, *Codice*, *Titolo*, *Tipo*)

*INCLUSIONE*(*Contenete*, *Incluso*)

*LIBRO* descrive le classi. *SEZIONI* descrive le sezioni di un libro. La struttura del libro può essere arbitrariamente complessa (una sezione può contenere altre sezioni, le quali possono contenere a loro volta altre sezioni etc) e la struttura che descrive l'inclusione delle sezioni è dunque un albero di altezza arbitraria. Si assume che esista una sezione che rappresenta l'intero libro (radice nell'albero della struttura). *INCLUSIONE* indica la relazione di inclusione tra le sezioni.

**Esercizio 11** (8 punti) *Si scriva una interrogazione in algebra relazionale che, se valutata, fornisce il titolo dei libri che sono costituiti solo da sezioni prive di sottosezioni.*

**Esercizio 12** (8 punti) *Si scriva una interrogazione SQL che restituisce il titolo dei libri la cui struttura è un albero di grado  $k$  (ogni sezione o non ha sottosezioni oppure ha esattamente  $k$  sottosezioni).*

**Esercizio 13** (8 punti) *Si scriva un metodo PLSQL che riceve in ingresso il codice di una sezione e che restituisce una stringa di caratteri contenente nell'ordine, il titolo della sezione stessa, la lista di tutti i titoli delle sezioni che la contengono (nell'ordine di risalita dalla sezione alla radice), il titolo del libro.*

**Esercizio 14** (8 punti) *Si consideri la relazione *CORSI*(*SSD*, *TITOLO*, *GRUPPO*, *SEMESTRE*, *AA*, *DOCENTE*) che descrive la attivazione di corsi in un determinato Anno Accademico. Valgono le seguenti dipendenze funzionali*

- *Docente*  $\rightarrow$  *SSD*;
- *TITOLO*  $\rightarrow$  *SSD*;
- *TITOLO*, *AA*  $\rightarrow$  *SEMESTRE*;
- *TITOLO*, *AA*, *GRUPPO*, *SEMESTRE*  $\rightarrow$  *DOCENTE*;

*Trovare le chiavi.*

*Trovare una copertura canonica per le dipendenze.*

*Dire se *CORSI* 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.*

## 2 8 gennaio 2008

Si consideri il seguente schema relazionale che descrive la strutturazione di classi in un class diagram di UML.

*CLASS*(*Nome*, *Descrizione*)

*REFINE*(*SuperClass*, *SubClass*, *Total*, *Disjoint*)

*ASSOCIAZIONI*(*Nome*, *Descrizione*)

*COMPASSOC*(*Associazione*, *Classe*, *Ruolo*, *CardMin*, *CardMax*)

*CLASS* descrive le classi. *REFINE* descrive la relazione di specializzazione delle classi (*SubClass* è la specializzazione della classe *SuperClass*); *Total* e *Disjoint* sono due tipi booleani che indicano se la specializzazione è totale e disgiunta. *ASSOCIAZIONI* indica quali sono le associazioni presenti tra le classi. *COMPASSOC* indica quali sono le classi che partecipano alla associazione: *Associazione* è il nome della associazione a cui *Classe* partecipa; *Ruolo* indica il nome del ruolo della partecipazione, *CardMin* e *CardMax* indicano la cardinalità minima e massima della partecipazione della classe alla associazione.

**Esercizio 21** (7 punti) Si scriva una interrogazione in algebra relazionale che, se valutata, fornisce il nome delle classi che *NON* sono coinvolte in nessuna associazione ricorsiva (è ricorsiva quando la classe partecipa con due ruoli distinti alla associazione).

**Esercizio 22** (7 punti) Si scriva una vista che per ogni classe indicata dal nome restituisce il numero delle classi che la specializzano, e il numero delle associazioni binarie uno a uno, uno a molti, e molti a molti a cui la classe partecipa.

**Esercizio 23** (8 punti) Si scriva un metodo *PLSQL* che riceve in ingresso il nome di una classe e che restituisce una stringa di caratteri contenente, il nome della classe e la lista delle associazioni attributi (nome associazioni e ruolo separati da una virgola) visibili in quella classe (sono visibili nella classe tutte le associazioni della classe e quelle delle sue superclassi).

**Esercizio 24** (7 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.

### 3 6 settembre 2007

Si consideri il seguente schema relazionale che descrive la strutturazione di classi in un class diagram di UML.

*CLASS*(*Nome*, *Descrizione*)

*REFINE*(*SuperClass*, *SubClass*)

*METODI*(*Class*, *Nome*, *TipoOut*)

*PARAMETRI*(*Class*, *NomeM*, *NomePar*, *TipoPar*)

*CLASS* descrive le classi. *REFINE* descrive la relazione di specializzazione delle classi (*SubClass* è la specializzazione della classe *SuperClass*; una classe non è specializzata se non compare come *SuperClass* in nessuna riga in *REFINE*). *METODI* indica quali metodi vengono associati ad una classe: *Class* indica il nome della classe di appartenenza, *Nome* il nome del metodo, *TipoOut* il tipo di dato restituito dal metodo. *PARAMETRI* indica quali parametri vengono associati ad un metodo (*Class* indica il nome della classe di appartenenza del metodo, *NomeM* il nome del metodo, *NomePar* il nome del parametro e *TipoPar* il tipo del parametro).

**Esercizio 31** (7 punti) Si scriva una interrogazione in algebra relazionale (senza l'uso di operazioni di conteggio) che, se valutata, fornisce il nome delle classi non specializzate che hanno esclusivamente metodi con un (ed uno solo) parametro.

**Esercizio 32** (7 punti) Si scriva una interrogazione SQL che recupera il nome del metodo e della sua classe di appartenenza per metodi che costituiscono un overriding di metodi presenti nella classe padre. Si ricorda che un metodo *A* costituisce un overriding del metodo *B* se *A* ha lo stesso nome di *B* ma almeno un parametro diverso da *B*.

**Esercizio 33** (9 punti) Si scriva un metodo PLSQL che riceve in ingresso il nome di una classe e che restituisce una stringa di caratteri contenente, il nome della classe e la lista dei nomi dei metodi visibili in quella classe (sono visibili nella classe tutti i nomi associati alla classe e quelli associati alle sue superclassi).

**Esercizio 34** (7 punti) Si consideri la relazione  $R(A, B, C, D)$  e le seguenti dipendenze logiche:

- $AB \rightarrow C$ ;
- $A \rightarrow D$ ;
- $BD \rightarrow C$

Trovare le chiavi.

Trovare una copertura canonica per le dipendenze.

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

Se lo schema non è in terza forma normale scomporlo applicando l'algoritmo di normalizzazione.

## 4 24 luglio 2007

Si consideri il seguente schema relazionale che descrive la strutturazione di classi in un class diagram di UML.

*CLASS*(*Nome*, *Descrizione*)

*REFINE*(*SuperClass*, *SubClass*, *Total*, *Disjoint*)

*ATTRIBUTI*(*Class*, *Nome*, *Tipo*)

*BASICTYPE*(*Nome*)

*CLASS* descrive le classi. *REFINE* descrive la relazione di specializzazione delle classi (*SubClass* è la specializzazione della classe *SuperClass*); *Total* e *Disjoint* sono due tipi booleani che indicano se la specializzazione è totale e disgiunta. *ATTRIBUTI* indica quali attributi vengono associati ad una classe. Il tipo di un attributo può essere di tipo basico (contenuto nello schema *BASICTYPE*) oppure il nome di una classe.

**Esercizio 41** (7 punti) Si scriva una interrogazione in algebra relazionale che, se valutata, fornisce il nome delle classi non specializzate che prevedono solo attributi di tipo *BASICTYPE*.

**Esercizio 42** (7 punti) Si scriva una interrogazione SQL che recupera coppie di nomi di classi in modo che per ogni attributo della seconda classe vi sia un attributo della prima classe che abbia lo stesso nome e lo stesso tipo.

**Esercizio 43** (9 punti) Si scriva un metodo PLSQL che riceve in ingresso il nome di una classe e che restituisce una stringa di caratteri contenente, il nome della classe e la lista degli attributi (nome e tipo degli attributi separati da una virgola) visibili in quella classe (sono visibili nella classe tutti gli attributi associati alla classe ed quelli associati alle sue superclassi).

**Esercizio 44** (Facoltativo 5 punti) Si esprimano i seguenti vincoli:

- Tutte le indicazioni di specializzazione della stessa classe devono avere gli stessi valori per gli attributi *total* e *disjoint*.
- I valori dell'attributo *tipo* nello schema *ATTRIBUTI* devono essere o di tipo *BASICTYPE* o devono essere nomi di classi.

**Esercizio 45** (7 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.

## 5 29 giugno 2007

Si consideri il seguente schema relazionale che descrive la composizione di pezzi meccanici

*PEZZI*(CodP, Articolo, Modello, Reparto, Scaffale, Quantita, Costo)

*ASSEMBLAGGIO*(Composto, Componente, Quantita)

*PEZZI* descrive i pezzi meccanici presenti in magazzino (viene indicato il reparto dove viene custodito il pezzo e lo scaffale nel reparto; Quantita indica il numero di pezzi presenti in magazzino);

*ASSEMBLAGGIO*, descrive il modo in cui i pezzi posso essere assemblati, in particolare *Composto* indica il coddice di un pezzo meccanico che richiede l'assemblaggio interno di altri pezzi indicati dall'attributo *Componente* (è il codice di un pezzo);

**Esercizio 51** (7 punti) Si scriva una interrogazione in algebra relazionale che, se valutata, fornisce il codice dei pezzi che sono composti (al primo livello di composizione) da esattamente due (due e solo due) pezzi componenti distinti (Si intenda da due tipologie differenti di pezzi componenti, non da due esemplari dello stesso pezzo). Si risolva l'esercizio preferibilmente senza ricorrere ad operazioni di conteggio.

**Esercizio 52** (7 punti) Un pezzo è di base quando non prevede l'assemblaggio interno di nessun altro pezzo. Si scriva una interrogazione SQL che recupera il codice dei pezzi in cui vengono assemblati SOLO pezzi di base.

**Esercizio 53** (10 punti) Il costo di un pezzo si ottiene sommando al costo del pezzo stesso, il costo di tutte le sue componenti (si faccia attenzione al fatto che una componente può essere a sua volta composta). Per semplicità si assuma che un pezzo possa occorrere una sola volta nella composizione di un altro pezzo). Si scriva un metodo PLSQL che riceve in ingresso il codice di un pezzo e che restituisce il costo del pezzo. Per il calcolo del costo si suggerisce di usare una tabella temporanea *TEMP*(CodP, Costo) per memorizzare i risultati intermedi della scomposizione del pezzo nelle sue componenti).

**Esercizio 54** (7 punti) Si consideri la relazione *MAGAZZINO*(Articolo, Modello, Scaffale, Stanza) che descrive la collocazione dei modelli degli articoli (un articolo può avere vari modelli) negli scaffali delle stanze di un magazzino (una stanza ha vari scaffali). La relazione gode delle seguenti proprietà:

- Gli esemplari di un articolo che hanno lo stesso modello stanno nello stesso scaffale;
- Gli esemplari del medesimo articolo stanno nella stessa stanza;
- Gli esemplari di un articolo che stanno nella stessa stanza e nello stesso scaffale appartengono allo stesso modello.

Scrivere le dipendenze suggerite nella descrizione sopra riportata.

Trovare una copertura canonica per le dipendenze.

Dire se la tabella *MAGAZZINO* 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 I Prova intercorso

Si consideri il seguente schema relazionale che descrive la gestione di un magazzino

*MAGAZZINO*(CodA, Descrizione, Collocazione, Quantita)

*ORDINE*(CodO, DataOrdine, CodFornitore, DataArrivo)

*COMPORDINE*(CodA, CodO, Quantita)

*FORNITORE*(PI, RagSociale, Via, Citta)

*CATALOGO*(CodA, PI, Prezzo)

*MAGAZZINO* descrive gli articoli presenti in magazzino (Quantità indica il numero di pezzi presenti in magazzino);

*ORDINE*, descrive un ordine di articoli ad un fornitore. La data di arrivo è un attributo parziale il cui valore è inserito solo quando la merce viene fornita dal fornitore; *COMPORDINE* descrive la composizione dell'ordine (articoli ordinati e relativa quantità);

*FORNITORE* descrive l'elenco dei fornitori;

*CATALOGO* descrive il listino prezzi attuato dai fornitori per gli articoli.

**Esercizio 61** *Si scriva una interrogazione in algebra relazionale che, se valutata, fornisce il codice degli ordini e la ragione sociale dei fornitori per ordini non ancora arrivati ed urgenti (un ordine è urgente quando tutti gli articoli dell'ordine hanno scorta zero in magazzino).*

**Esercizio 62** *Si scriva una interrogazione in SQL che fornisca le PI dei fornitori che hanno nel catalogo tutti gli articoli che nel magazzino hanno scorta zero.*

**Esercizio 63** *Si esprima nel modo più adeguato il seguente insieme di vincoli:*

- Un articolo non compare più di una volta nello stesso ordine;
- Il fornitore di un ordine deve avere in catalogo tutti gli articoli dell'ordine;
- La data di arrivo di un ordine è successiva a quella di invio (DataOrdine);
- Un ordine può riferirsi solo ad articoli presenti nel magazzino
- Quando viene inserita la data di arrivo dell'ordine viene aggiornata automaticamente la quantità degli articoli in magazzino;

## 7 II Prova intercorso

Si consideri il seguente schema relazionale che descrive la composizione di pezzi meccanici

*PEZZI*(*CodP*, *Articolo*, *Modello*, *Reparto*, *Scaffale*, *Quantita*, *Costo*)

*ASSEMBLAGGIO*(*Composto*, *Componente*, *Quantita*)

*PEZZI* descrive i pezzi meccanici presenti in magazzino (viene indicato il reparto dove viene custodito il pezzo e lo scaffale nel reparto; *Quantita* indica il numero di pezzi presenti in magazzino);

*ASSEMBLAGGIO*, descrive il modo in cui i pezzi posso essere assemblati, in particolare *Composto* indica il coddice di un pezzo meccanico che richiede l'assemblaggio interno di altri pezzi indicati dall'attributo *Componente* (è il codice di un pezzo);

**Esercizio 71** *Il costo di un pezzo si ottiene sommando al costo del pezzo stesso, il costo di tutte le sue componenti (si faccia attenzione al fatto che una componente può essere a sua volta composta). Per semplicità si assuma che un pezzo possa occorrere una sola volta nella composizione di un altro pezzo). Si scriva un metodo PLSQL che riceve in ingresso il codice di un pezzo e che restituisce il costo del pezzo. Per il calcolo del costo si suggerisce di usare una tabella temporanea TEMP(CodP, Costo) per memorizzare i risultati intermedi della scomposizione del pezzo nelle sue componenti).*

**Esercizio 72** *Si consideri la relazione MAGAZZINO(Articolo, Modello, Scaffale, Stanza) che descrive la collocazione dei modelli degli articoli (un articolo può avere vari modelli) negli scaffali delle stanze di un magazzino (una stanza ha vari scaffali). La relazione gode delle seguenti proprietà:*

- *Gli esemplari di un articolo che hanno lo stesso modello stanno nello stesso scaffale;*
- *Gli esemplari del medesimo articolo stanno nella stessa stanza;*
- *Gli esemplari di un articolo che stanno nella stessa stanza e nello stesso scaffale appartengono allo stesso modello.*

*Scrivere le dipendenze suggerite nella descrizione sopra riportata.*

*Trovare una copertura canonica per le dipendenze.*

*Dire se la tabella MAGAZZINO 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.*