

Basi di Dati e Sistemi Informativi I, 13 febbraio 2015

DIETI, Corso di Laurea in Informatica

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

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

REFINE(*SuperClass*, *SubClass*)

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

BASICTYPE(*Nome*)

METODO(*CodM*, *Class*, *Nome*, *TipoOut*, *Posizione*)

PARAMETRI(*CodM*, *Class*, *Nome*, *Tipo*, *Posizione*)

CLASS descrive le classi. *REFINE* descrive la relazione di specializzazione delle classi (*SubClass* è la specializzazione della classe *SuperClass*; entrambi gli attributi sono totali). *ATTRIBUTI* descrive gli 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. Posizione indica l'ordine in cui compare l'attributo nella classe. *METODO* descrive i metodi associati ad una classe. Il *Tipout* di ritorno di un metodo può essere di tipo basico (contenuto nello schema *BASICTYPE*) oppure il nome di una classe oppure la costante *Void*. Posizione indica l'ordine in cui compare il metodo nella classe. *PARAMETRI* descrive i parametri associati ad un metodo. Il *Tipout* di un attributo può essere di tipo basico (contenuto nello schema *BASICTYPE*) oppure il nome di una classe. Posizione indica l'ordine in cui compare il parametro nel metodo.

Esercizio 01 (7 punti) Si scriva una interrogazione in algebra relazionale che, se valutata, fornisce per ogni classe, il codice della classe, il nome della classe, il numero di attributi di tipo *Basictype*, il numero di attributi non *Basictype*, il numero di metodi senza parametri.

Esercizio 02 (8 punti) Si scriva una interrogazione SQL che restituisca delle terne, in cui le prime due componenti sono nomi diversi di classi e la terza componente è il nome di un metodo. La condizione che la terna deve soddisfare è che in ciascuna delle due classi vi sia un metodo avente quel nome e che i due metodi abbiano la medesima segnatura (i codici sono comunque distinti). Due metodi hanno la stessa segnatura se hanno lo stesso nome, lo stesso tipo di ritorno e gli stessi parametri (in ogni posizione stesso nome e stesso tipo).

Esercizio 03 (6 punti) Si assuma che nella tabella *BASICTYPE* sia contenuto il valore *Undefined*. Si scriva un trigger che viene attivato quando viene cancellata una classe. Il suo effetto è quello di sostituire preventivamente, in tutti gli attributi, i parametri e il valore di ritorno dei metodi che avevano come tipo associato il nome della classe da cancellare, il nome della classe che si deve cancellare con il valore *basicty Undefined*.

Esercizio 04 (9 punti) Si scriva un metodo *PLSQL* che riceve in ingresso il codice di una classe e che restituisce una stringa contenente la segnatura di tutti i metodi visibili nella classe (la segnatura ha la forma *nomemetodo(par1 tipo1, ..., parn tipon):tiporitorno*). Si ricorda che i metodi visibili sono quelli presenti nella classe e in tutte le sue superclassi.

Esercizio 05 (6 punti) Si consideri la relazione $R(A, B, C, D, E)$ che soddisfa le seguenti dipendenze funzionali:

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

Trovare le chiavi. Dire se R 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.