

# Basi di Dati

Corso di Laurea in “Informatica”

04 febbraio 2010

## Note

1. Su tutti i fogli contenenti le soluzioni indicare, IN STAMPATELLO, la data dell'appello ed il proprio cognome, nome e numero di matricola.
2. Non è consentita la consultazione di alcunché.
3. L'orario di consegna scritto alla lavagna è tassativo.
4. Il testo del compito va consegnato insieme a tutti i fogli; marcare in modo evidente i fogli di brutta (che vanno consegnati insieme ai fogli contenenti le soluzioni).

## Esercizi

1. Mostrare lo schema concettuale Entità-Relazione per il database di un sistema di *bug tracking*, il cui scopo è la gestione della tracciabilità degli errori nei prodotti software. Si richiede di modellare le informazioni seguenti:
  - (a) Gli utenti del servizio di bug tracking sono identificati da una login e sono dotati di password, indirizzo email ed un nome opzionale.
  - (b) I prodotti software per i quali è fornito il servizio di bug tracking: identificati da un nome, per essi si memorizzano una breve descrizione, l'indirizzo del sito web, la data di registrazione presso il bug tracker, il manutentore ufficiale (un utente registrato) e, per i prodotti obsoleti, la data di cessazione di attività.
  - (c) Per ogni prodotto software, l'insieme delle componenti logiche dalle quali è composto. Tali componenti sono caratterizzate da un nome univoco all'interno del prodotto e possono essere associate ad un numero arbitrario di sviluppatori (utenti registrati). Il manutentore di un prodotto è considerato sviluppatore di tutte le sue componenti e non deve essere associato esplicitamente ad esse.
  - (d) Oltre a poter essere manutentori/sviluppatori di un prodotto/componente, tutti gli utenti registrati possono contribuire al servizio mediante la segnalazione di guasti e la proposta di correzioni.
  - (e) Una segnalazione di guasto viene detta *problem report* (PR). Ogni PR è associato ad un solo componente di un prodotto software. Di esso si conoscono la data di creazione, una breve descrizione, l'autore (un utente) e lo stato, che deve essere un valore tra: *nuovo* (default), *confermato*, *non confermato*, *duplicato* e *risolto*.
  - (f) Tutti i PR, tranne quelli con stato *nuovo*, devono avere associato uno sviluppatore al quale sono stati assegnati; per essi si conosce anche la data dell'ultimo cambiamento di stato effettuato.
  - (g) Tutti i PR con stato *duplicato* hanno associato un altro PR (quello del quale sono un duplicato, che deve far parte dello stesso prodotto software).
  - (h) Ad ogni PR è associata una sequenza ordinata di *commenti*, caratterizzati da un numero progressivo, testo, data e autore (un utente). Alcuni commenti hanno associata una correzione (detta *patch*), caratterizzata da un testo ed uno stato (*in attesa*, *accettata*, *rifiutata*).

2. Tradurre lo schema concettuale dell'esercizio precedente in uno schema logico relazionale, codificando opportunamente i vincoli dello schema.
3. Con riferimento allo schema relazionale sviluppato nell'esercizio precedente, esprimere le seguenti interrogazioni in linguaggio SQL.
  - (a) Fornire l'elenco degli sviluppatori registrati con l'indicazione del numero di componenti di prodotti ai quali collaborano, in ordine decrescente di componenti.
  - (b) Definire la vista relazionale "PR aperti" che elenca, per ogni prodotto software non obsoleto, le segnalazioni di guasto il cui stato è *nuovo* o *confermato*; analogamente, definire la vista "PR chiusi" per le segnalazioni di guasti di prodotti non obsoleti il cui stato è diverso da *nuovo* e *confermato*.
  - (c) Estrarre l'elenco dei PR aperti per i quali non si siano registrate variazioni (creazione, cambiamento di stato o aggiunta di commenti) nell'ultima settimana.
  - (d) Per ogni prodotto software non obsoleto calcolare il *saldo mensile*, ovvero la differenza tra il numero di PR che sono stati creati e quelli che sono stati chiusi nel corso degli ultimi 30 giorni.
  - (e) Esprimere come espressione dell'algebra relazionale il vincolo che impone che tutti i PR non nuovi siano stati assegnati ad un utente che sia effettivamente uno sviluppatore della corrispondente componente del prodotto software.
  - (f) Scrivere un trigger che, a fronte della cancellazione di un utente, aggiorni il database in modo che tutti i PR ed i commenti ad esso riferibili siano associati al manutentore del prodotto software corrispondente.
  - (g) Calcolare il punteggio di ogni utente, secondo la seguente formula:

$$\# \text{ commenti creati} + 5 \times \# \text{ PR creati} + 10 \times \# \text{ patch accettate}.$$