Database relazionali – SQL

Studente: Gabriele Nigro

ciro.santoro@sideagroup.com

Domande

1. Qual è la differenza fra dati e informazioni?  
     
   **Il dato è essenzialmente un simbolo, privo di contesto, come può esserlo una cifra, una parola, un’immagine  
   Per ottenere informazioni a partire dai dati, questi vanno contestualizzati  
   Quindi, una cifra(dato) diventa informazione se questo viene magari indicizzato come “età”**
2. Che cosa una base di dati, a cosa serve?  
     
   **Una base di dati è un’insieme di dati, strutturati usando un qualche modello e legati tra loro in qualche modo variabile  
   Si usa una base di dati, in modo contrapposto ad un classico singolo file su memoria, quando si hanno esigenze relative a scalabilità, quando la mole di dati da gestire è molto grande, quando si vuole strutturare in un certo modo i dati a disposizione o quando questi dati devono poter essere usati in contemporanea da più utenti**
3. Quali modelli logici esistono per rappresentare i dati?  
     
   **I modelli gerarchico e reticolare, dove il primo fa uso di una struttura ad albero, il secondo un simil-albero dove ogni nodo può avere più di un nodo padre.  
   Sono utili se si volesse rappresentare ad esempio le directory di un file system, dato che il sistema operativo usa una struttura del genere. La loro struttura risulta perciò anche rigida.  
     
   Oggi esistono modelli più flessibili e adatti a soluzioni reali e concrete, quali il relazionale, non relazionale, ad oggetti...**
4. Descrivi il modello relazionale dei dati.

**Nel modello relazionale dei dati, questi sono organizzati in relazioni, strutture tabellari divise quindi in righe e colonne. Ogni riga è un record, ogni colonna un campo del record e l’ordine di queste ultime è irrilevante, non essendo un sistema posizionale.**

**Tra le relazioni vengono poi stabilite delle associazioni, sulla base di valori uguali in colonne corrispondenti.**

1. A cosa serve il linguaggio SQL? Fai alcuni esempi di istruzioni SQL.  
     
   **Il linguaggio SQL è il linguaggio dichiarativo usato per definire e manipolare basi di dati.  
   Consiste di due parti: Data Definition Language usato per creare/eliminare schemi degli oggetti (database, tabelle, viste...) della base di dati  
   Data Manipulation Language per recuperare e modificare il contenuto delle basi di dati.**

**Un esempio di istruzione DDL è CREATE, usato appunto per creare nuovi oggetti quali DB o relazioni: create nome\_tabella(nome\_attributo tipo\_attributo eventualiVincoli e poi vincoli di tabella). L’opposto di questo è il comando DROP.  
  
Per il DML si deve citare il comando SELECT, usato per fornire una query con la quale interrogare il database ed estrarne i contenuti, usando una forma dove si specificano quali campi di quali tabelle si desiderano, le condizioni che le tuple del risultato devono soddisfare ed eventuali altre operazioni da applicare al result set.**

1. Quali comandi SQL appartengono alla categoria DML, Data Manipulation Language?  
     
   **I comandi di SELECT, UPDATE, INSERT, DELETE**
2. Cosa è un dbms? Elenca alcuni dbms.  
     
   **DBMS sta per DataBase Management System, è una classe di software usati appunto per creare e gestire basi di dati, offrendo diversi servizi sia agli utenti che agli amministratori.  
   Un DBMS non avrà problemi a gestire la quantità di dati immagazzinati e si occuperà di garantire protezione da accessi di utenti non autorizzati, accesso efficace ed efficiente ai dati e gestione di operazioni concorrenti**

**Esistono molti DBMS, distinti in base al modello usato per rappresentare i dati.  
Tra quelli relazionali ad esempio c’è MySQL o OracleDB. Ci sono anche quelli non relazionali come PostgreSQL o alcuni che servono ad effettuare particolari analisi ed operazioni sui dati, come alcuni prodotti Apache.**

1. Con quale istruzione SQL posso visualizzare l’elenco dei database esistenti?  
   **show databases;**
2. Con quale istruzione SQL posso selezionare un database da usare?  
   **use nome\_database;**
3. Cosa cambia tra le seguenti definizioni di attributi in una tabella?  
   Nome CHAR(30)  
   Cognome VARCHAR(30)  
     
   **Char indica una dimensione fissa: stringhe meno lunghe vengono riempite con spazi, stringhe più lunghe vengono troncate. Varchar imposta una lunghezza massima ma non inserisce spazi alla fine di una stringa più corta.**
4. Cosa cambia tra le seguenti definizioni di attributi in una tabella?  
   Tempo1 DATE  
   Tempo2 TIME  
   Tempo3 DATETIME  
     
   **Date si usa per sole date, in un formato quale ad esempio yyyy-mm-dd  
   Time si usa per soli orari, in un formato come ad esempio ore:minuti:secondi  
   Datetime è un’unione dei due**.
5. Scrivere il comando per creare una tabella per contenere un campo numerico, uno testuale e uno booleano. Il campo numerico è la chiave primaria della tabella.

**Create table tabella1 (**

**campoChiave numeric primary key,**

**campoTestuale varchar(n),**

**campoBooleano boolean (se supportato dal dbms, altrimenti valori numerici 1 e 0)**

**);**

**//n = lunghezza massima campo**

1. Scrivere il comando per creare una tabella contenente due campi, uno testuale ed uno numerico. Il campo numerico è una chiave esterna che riferisce la tabella del punto 12.  
     
   **create table tabella2 (**

**campoTestuale varchar(n) primary key, – presumo --**

**campoNumerico numeric references tabella1(campoChiave);**

1. Quale tipo di relazione esiste tra le tabelle dei punti 12 e 13? Quali altri tipi di relazioni possono esistere tra due tabelle?

**Tra di loro c’è una relazione uno a molti: ad una istanza di tabella1 può essere associata 1 o più istanze di tabella2, ma ogni istanza di tabella2 si riferisce ad una singola istanza di tabella1  
  
Esistono anche relazioni uno ad uno e molti a molti, definite in modo analogo**

1. Scrivere il comando per inserire una riga nella tabella del punto 12.  
     
   **insert into tabella values(numero, stringa, true/false);**
2. Il risultato del seguente predicato è TRUE o FALSE?  
   A = TRUE, B = FALSE, C = FALSE  
   (A OR B) AND (NOT(C))  
     
   **A or B restituisce True  
   not(c) = not(false) = true  
   Quindi true and true = TRUE**
3. Il risultato del seguente predicato è TRUE o FALSE?  
   A = TRUE, B = FALSE, C = FALSE  
   (A OR B) OR (NOT(C)) OR (C AND B AND A) OR (NOT(NOT(NOT(A)))) OR B

**Questa è una sequenza di clausole, quindi predicati legati da degli OR**

**Basta uno solo di questi per rendere vero tutto il predicato**

**A or B = True or False = True, quindi il predicato è TRUE**