

# Laboratorio di Basi di dati 1

## VIII Esercitazione 20 maggio 2003

Si consideri la base di dati `GestioneAziendaleN` (dove  $N$  è il numero del gruppo a cui appartenete) ed in particolare le tabelle create durante l'esercitazione V o VII mediante l'esecuzione dello script `creadb_es5.sql`. SOLO per chi non aveva eseguito tale script:

1. Selezionare la base di dati creata durante la prima esercitazione `GestioneAziendaleN`, (dove  $N$  è il numero del gruppo a cui appartenete) che rappresenta i dati relativi all'Università di Genova.
2. Cancellare le tabelle create in precedenza, oppure rinominarle.
3. Scaricare lo script `creadb_es5.sql` dalla pagina Web del laboratorio di basi di dati <http://www.disi.unige.it/person/MesitiM/teach/BD1-02-03/lab.html>
4. Dal Query Analyzer aprire lo script in locale ed eseguirlo. Siate sicuri di aver selezionato il database corretto, prima di eseguire lo script, altrimenti modificate di conseguenza la prima riga dello script e decommentatela.

### Per tutti:

- Copiare in locale (sul desktop o dove preferite) il file Java `templateJava.java` dalla pagina Web del laboratorio di basi di dati <http://www.disi.unige.it/person/MesitiM/teach/BD1-02-03/lab.html>.
- Associare un identificatore per accedere la vostra base di dati tramite ODBC. Tale identificatore verrà utilizzato come `subname` da utilizzare per identificare la vostra base di dati in JDBC (si ricorda che verrà sempre utilizzato il driver JDBC-ODBC bridge, che si appoggia su ODBC). Per farlo:
  - Da `Control Panel/Administrative Tools` lanciare l'applicazione `Data Sources (ODBC)`.
  - A questo punto, selezionare `User DSN` e selezionare `Add`. Questo ci permette di creare un identificatore associato alla nostra base di dati.
  - Scegliere `SQL Server` come tipo di driver.
  - Specificare `Murphy` come server e scegliere un identificatore. Andare avanti.
  - Selezionare l'opzione 'autenticazione basata su SQL Server', ed introdurre la propria login e password. Andare avanti.
  - Scegliere la base di dati. Andare avanti.
  - Selezionare `Fine`.

Il file `templateJava.java` dovrà essere modificato opportunamente per eseguire le seguenti operazioni (generare un nuovo programma per ognuna delle operazioni indicate nel seguito). Ogni nuovo programma dovrà utilizzare l'identificatore ODBC generato come `subname` nella stringa che specifica il server.

**Se i comandi `javac` e `java` non vengono riconosciuti:** da linea di comando modificare i path come segue: `set PATH = %PATH%;c:\programs\jdk1.3.1\bin`

1. *Connessione alla base di dati.* Creare un programma Java `JdbcConn.java` che apre una connessione con il server e la chiude, riportando eventuali errori.

Compilare ed eseguire tale programma.

2. *Esecuzione query non preparate, senza uso del cursore.* Creare un programma `JdbcQueryNonPrep1.java` in java che apre una connessione con il server, crea uno statement SQL per determinare il numero totale di studenti, esegue tale statement e stampa questo numero a video, con un opportuno messaggio. Infine, chiude la connessione. Eventuali errori devono essere riportati all'utente.

Compilare ed eseguire tale programma.

3. *Esecuzione query non preparate, con cursore.* Creare un programma Java `JdbcQueryNonPrep2.java` che apre una connessione con il server, crea uno statement SQL per determinare il cognome e il nome degli studenti che sono già in tesi (quindi il campo `Relatore` non è null). Quindi stampa queste informazioni a video (utilizzando un cursore) secondo il seguente formato:

Cognome: <cognome>, Nome: <nome>

Infine, chiude la connessione. Eventuali errori devono essere riportati all'utente.

Compilare ed eseguire tale programma.

4. *Esecuzione query preparate.* Creare un programma Java `JdbcQueryPrep.java` che prende in input due parametri da linea di comando, che rappresentano i codici identificativi di due studenti. Quindi, apre una connessione con il server, crea uno statement SQL preparato per determinare il codice identificativo dei corsi per i quali uno studente, non specificato nello statement, ha già sostenuto l'esame (quindi il codice dello studente è un parametro per la query). Successivamente, assegnare al parametro della query la prima stringa inserita da linea di comando ed esegue la query, stampando opportunamente il risultato a video (utilizzando un cursore). Assegna quindi al parametro della query la seconda stringa inserita da linea di comando ed esegue la query, stampando opportunamente il risultato a video (utilizzando un cursore). Infine, chiude la connessione. Eventuali errori devono essere riportati all'utente.

Compilare ed eseguire varie volte tale programma, passando diversi parametri.

5. *Esecuzione update non preparato.* Creare un programma Java `JdbcUpdate.java` che apre una connessione con il server, crea uno statement SQL non preparato per assegnare ad ogni professore uno stipendio pari a 2500 euro. Quindi esegue l'update. Eventuali errori devono essere riportati all'utente.

Compilare ed eseguire tale programma. Verificare l'avvenuta esecuzione della procedura sulla base di dati.

6. *Esecuzione stored procedure.* **Se non l'avete già fatto la settimana scorsa:** Da Query Analyzer, creare una tabella `ErroriCorsi(DenominazioneCorso)`, dove `DenominazioneCorso` rappresenta il nome di un corso. Creare quindi una procedura `CambiaDenCorso` che prende in input la denominazione di un corso `Den-old` e una nuova denominazione `Den-new`. Se non esiste alcun corso chiamato `Den-old`, la procedura deve inserire `Den-old` nella tabella `ErroriCorsi`. Altrimenti, deve sostituire `Den-old` con `Den-new`, in ogni tupla in cui `Den-old` appare.

**Per tutti:** Creare un programma Java `JdbcProc.java` che prende in input da linea di comando due parametri, che rappresentano rispettivamente il nome di un corso esistente e un nuovo nome. Quindi apre una connessione con il server, crea uno statement SQL non preparato per eseguire la chiamata alla procedura `CambiaDenCorso`, passandogli come parametri le stringhe fornite da linea di comando. Quindi eseguire lo statement come se fosse un update (utilizzando il metodo `executeUpdate`). Eventuali errori devono essere riportati all'utente.

Compilare ed eseguire varie volte tale programma, passando diversi parametri. Verificare l'avvenuta esecuzione della procedura sulla base di dati.