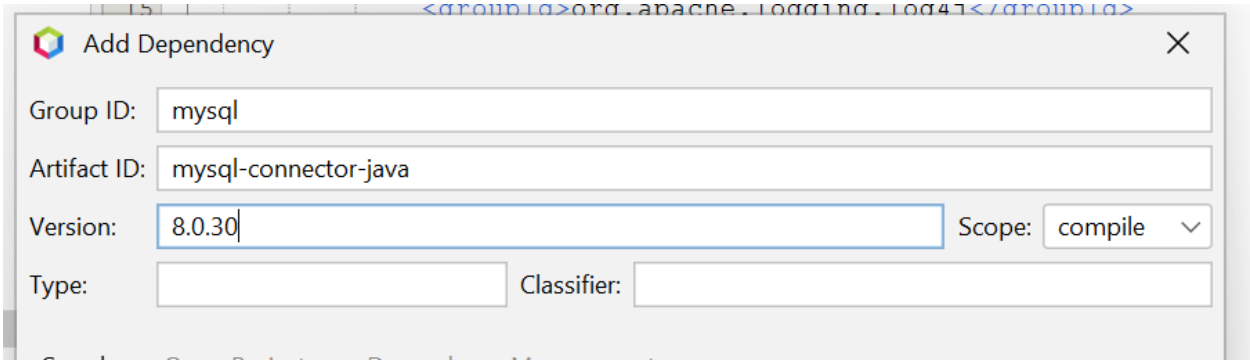
Programmazione avanzata

Lezione 6

Interazione con il database MySQL

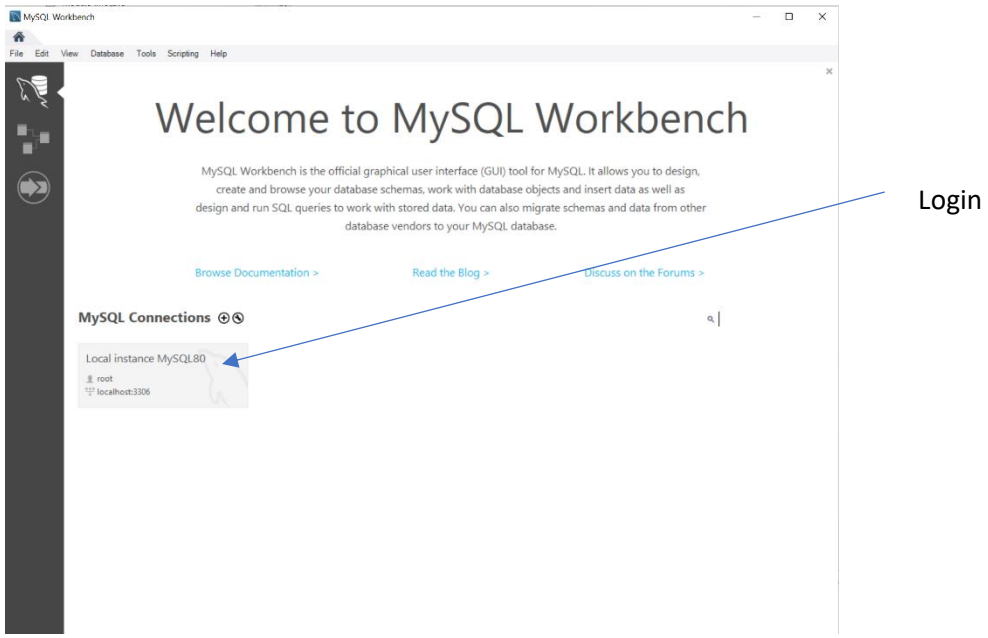
Aggiungere le dipendenze

La connessione ad un'istanza di un database MySQL avviene attraverso una serie di librerie che devono essere installate localmente. Per questo motivo la libreria mysql-connector-java deve essere inclusa tra le dipendenze:

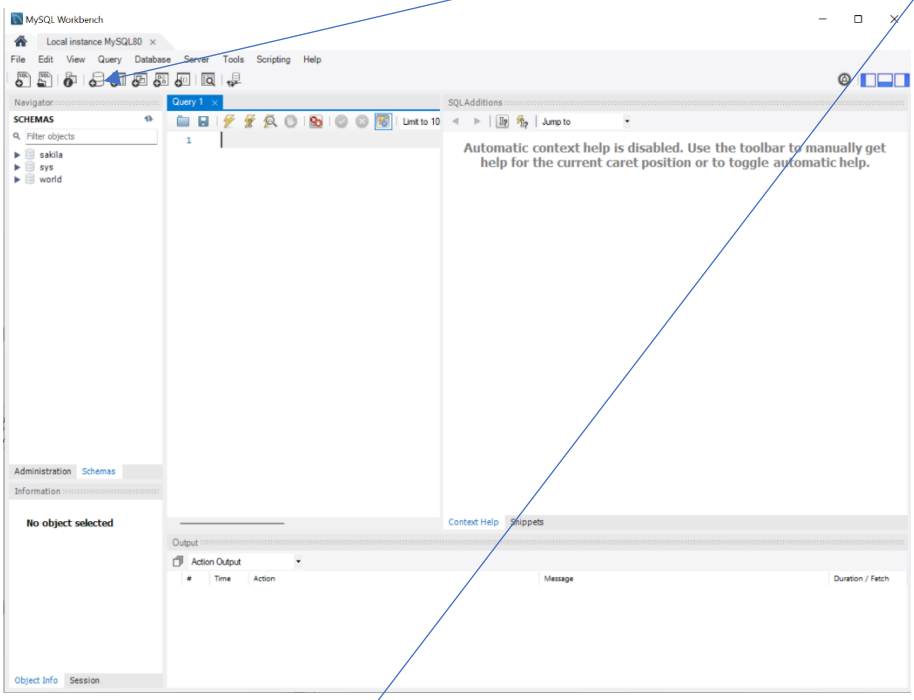


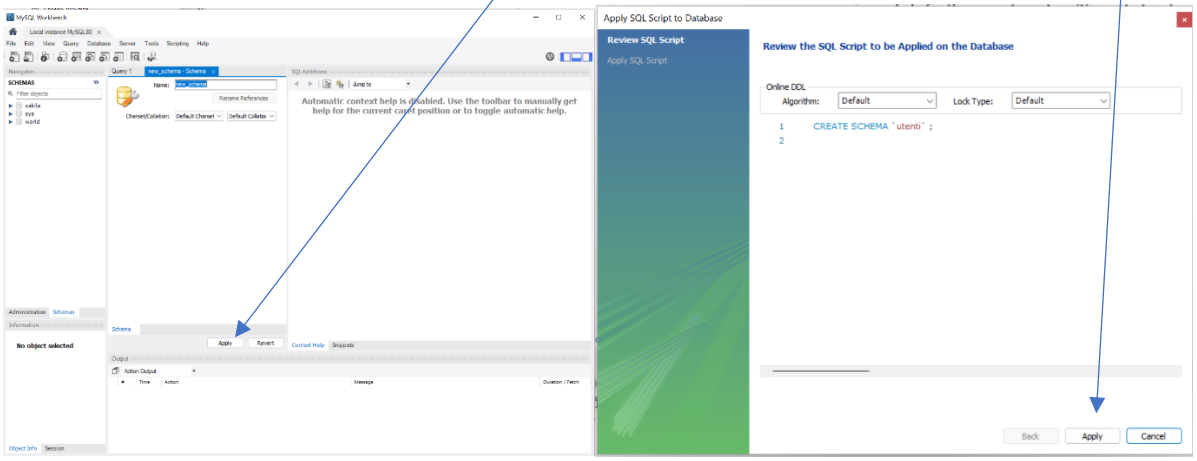
Creazione del database e inserimento primi dati

Il primo passo è quello di creare il database all'interno del server MySQL attraverso MySQL workbench.

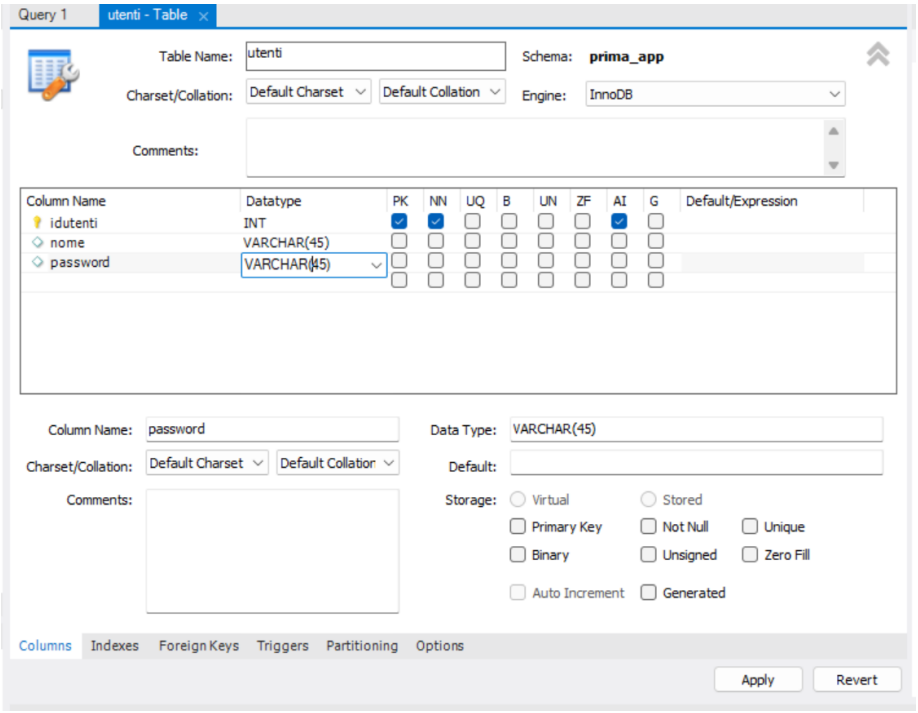


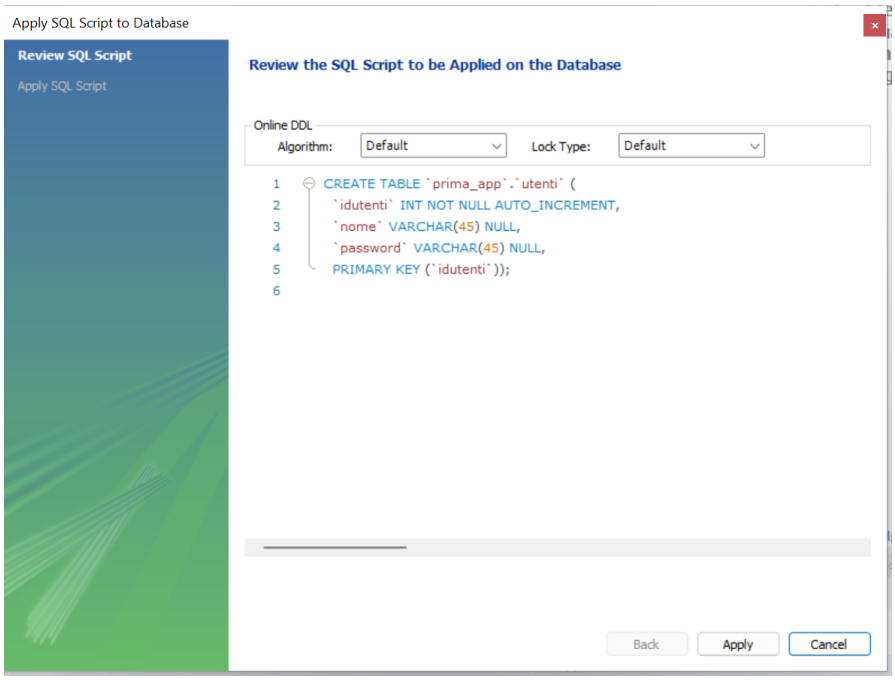
Crea un nuovo database



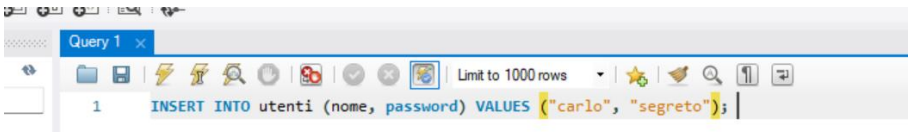


Creare una nuova tabella all'interno del database. Ad esempio, creiamo una tabella per memorizzare le informazioni degli utenti del sistema:





Inseriamo all'interno della tabella le informazioni di almeno un utente:



Interagire con il database

Il database può interagire con il programma attraverso una connessione al database:

```
1.  Connection co = DriverManager.getConnection("jdbc:mysql://localhost:3306/prima_app",
    "root","root");
```

La connessione al database necessita:

- Della stringa di connessione "jdbc:mysql://localhost:3306/prima_app" nella quale è incluso l'indirizzo IP del server MySQL e il nome del database;
- Il nome utente per connettersi al database
- La password per la connessione

Una volta effettuata la connessione si può procedere con interrogare il database, ad esempio recuperando tutte le righe della tabella:

```
1.  Statement st = co.createStatement();
2.  ResultSet rs = st.executeQuery("SELECT * FROM utenti;");
3.  while(rs.next())
4.      logger.info(rs.getString("nome"));
5.
```

L'interrogazione viene effettuata attraverso la classe Statement[1] che rappresenta una query SQL non parametrica. Una volta creata la struttura Statement si procede con l'interrogazione (riga 2) la quale ritorna un ResultSet. Il ResultSet è un oggetto iterabile che permette di scorrere tutti gli elementi del database recuperando i vari campi in base al nome della colonna. Nel caso di aggiornamenti sui dati si utilizza la executeUpdate che restituisce un intero rappresentante il numero di righe coinvolte da una INSERT, DELETE, o UPDATE.

Nel caso di query parametriche, ad esempio con delle condizioni nel where, si può usare la classe PreparedStatement[2] che rappresenta degli statement precompilati, con parametri di ingresso posizionati ma dal valore non definito. Mentre uno statement viene compilato dal DBMS ad ogni esecuzione, il prepared statement è compilato solo una volta ed è conveniente nel caso di esecuzioni multiple della stessa query con parametri diversi.

```
1.  try ( Connection co = DriverManager.getConnection("jdbc:mysql://localhost:3306/prima_app",
    "root", "root");
2.        PreparedStatement ps = co.prepareStatement("INSERT INTO utenti (nome, password) VALUES
    (?, ?)");) {
3.
4.        ps.setString(1, "utente1");
5.        ps.setString(2, "password");
6.
7.        ps.executeUpdate();
8.  } catch (SQLException e) {
9.        logger.error(e.getMessage());
10. }
```

[1] http://docs.oracle.com/javase/8/docs/api/java/sql/Statement.html
[2] http://docs.oracle.com/javase/tutorial/jdbc/basics/prepared.html

Nell’esempio si usa la classe PreparedStatement per inserire un nuovo utente nel database. Come si vede i valori parametrici in questo caso vengono inseriti nella query successivamente (righe 4-5) e in seguito la query viene eseguita (riga 7). Come si può vedere la connessione e la creazione dello PreparedStatement viene effettuato all’interno del costrutto try, se la connessione o la compilazione della query falliscono allora viene lanciata un’eccezione che viene subito catturata. Nell’esempio precedente, invece, le operazioni non erano eseguite all’interno di un costrutto try/catch, in quel caso la funzione doveva specificare il potenziale lancio di eccezioni del tipo SQLException.

Esercizio

Modificare il server in maniera tale da verificare le credenziali inviate dal client andando a verificare la presenza del nome utente e la correttezza della password dal database. I dati degli utenti nel database possono essere inseriti a mano tramite MySQL Workbench.

OPZIONALE: modificare il server per aggiungere gli utenti nel database durante la fase di inizializzazione (se questi non sono presenti nel database).