

## Esercitazione 6 – JDBC

- Modificare il progetto per includere il connettore di mysql (contiene il Driver per la connessione via JDBC a un dbms MySQL)

(In eclipse selezionare dal menu a tendina associato al progetto la voce Properties e modificare Libraries aggiungendo un jar esterno)

- Usare gli strumenti di Mysql per

1) creare un database con nome MapDB

```
CREATE DATABASE MapDB;
```

2) creare l'utente "MapUser" con **password** "map" che sia dotato dei diritti di accesso in lettura e scrittura per mapDB

```
CREATE USER 'MapUser'@'localhost' IDENTIFIED BY 'map';
```

```
GRANT SELECT ON MapDB.* TO MapUser@localhost IDENTIFIED BY 'map';
```

2) creare e popolare la tabella denominata provaC (alla stessa maniera possono essere create le tabelle servo, provinsert into MapDB.provac values('A', ....

```
CREATE TABLE MapDB.provaC(
```

```
    X varchar(10),
```

```
    Y float(5,2),
```

```
    C float(5,2)
```

```
);
```

```
insert into MapDB.provaC values('A',2,1);
```

```
insert into MapDB.provaC values('A',2,1);
```

```
insert into MapDB.provaC values('A',1,1);
```

```
insert into MapDB.provaC values('A',2,1);
```

```
insert into MapDB.provaC values('A',5,1.5);
```

```
insert into MapDB.provaC values('A',5,1.5);
```

```
insert into MapDB.provaC values('A',6,1.5);
```

```

insert into MapDB.provaC values(' B', 6, 10);

insert into MapDB.provaC values(' A', 6, 1.5);

insert into MapDB.provaC values(' A', 6, 1.5);

insert into MapDB.provaC values(' B', 10, 10);

insert into MapDB.provaC values(' B', 5, 10);

insert into MapDB.provaC values(' B', 12, 10);

insert into MapDB.provaC values(' B', 14, 10);

insert into MapDB.provaC values(' A', 1, 1);

commit;

```

- Nel progetto map6 creare il package `database` da popolare con le classi: `DbAccess`, `TableData`, `Column`, `TableSchema`, `DatabaseConnectionException`, `Example`, `EmptyTypeException`
- Definire la classe `DatabaseConnectionException` che estende `Exception` per modellare il fallimento nella connessione al database.
- Definire la classe `DbAccess` che realizza l'accesso alla base di dati.

## Attributi

`private final String DRIVER_CLASS_NAME = "com.mysql.cj.jdbc.Driver";` (Per utilizzare questo Driver scaricare e aggiungere al classpath il connettore mysql connector)

`private final String DBMS = "jdbc:mysql";`

`private String SERVER="localhost";` contiene l'identificativo del server su cui risiede la base di dati (per esempio localhost)

`private String DATABASE = "MapDB";` contiene il nome della base di dati

`private final String PORT=3306;` La porta su cui il DBMS MySQL accetta le connessioni

`private String USER_ID = "MapUser";` contiene il nome dell'utente per l'accesso alla base di dati

`private String PASSWORD = "map";` contiene la password di autenticazione per l'utente identificato da `USER_ID`

`private Connection conn;` gestisce una connessione

## Metodi

*public void initConnection() throws DatabaseConnectionException:* impartisce al class loader l'ordine di caricare il driver mysql, inizializza la connessione riferita da *conn*. Il metodo solleva e propaga una eccezione di tipo *DatabaseConnectionException* in caso di fallimento nella connessione al database.

*public Connection getConnection():* restituisce *conn*;

*public void closeConnection():* chiude la connessione *conn*;

- Definire le classi *Column* e *Table\_Schema* (fornite da docente) che modellano lo schema di una tabella nel database relazionale
- Definire la classe *EmptySetException* che estende *Exception* per modellare la restituzione di un resultset vuoto.
- Definire la classe *Example* (fornita dal docente) che modella una transazione letta dalla base di dati.
- Definire la classe *Table\_Data* (parzialmente fornita dal docente) che modella l'insieme di transazioni collezionate in una tabella. La singola transazione è modellata dalla classe *Example*.

*public List<Example> getTransazioni(String table) throws SQLException, EmptySetException*

**Input:** nome della tabella nel database.

**Output:** Lista di transazioni memorizzate nella tabella.

**Comportamento:** : Ricava lo schema della tabella con nome *table*. Esegue una interrogazione per estrarre le tuple *distinte* da tale tabella. Per ogni tupla del resultset, si crea un oggetto, istanza della classe *Example*, il cui riferimento va incluso nella lista da restituire. In particolare, per la tupla corrente nel resultset, si estraggono i valori dei singoli campi (usando *getFloat()* o *getString()*), e li si aggiungono all'oggetto istanza della classe *Example* che si sta costruendo.

Il metodo può propagare un eccezione di tipo *SQLException* (in presenza di errori nella esecuzione della query) o *EmptySetException* (se il resultset è vuoto)

(fornita dal docente)

*public Set<Object> getDistinctColumnValues (String table, Column column) throws SQLException*

**Input:** Nome della tabella insert into MapDB.provac values('A', nome della colonna nella tabella

**Output:** Insieme di valori distinti ordinati in modalità ascendente che l'attributo identificato da nome *column* assume nella tabella identificata dal nome *table*.

**Comportamento:** Formula ed esegue una interrogazione SQL per estrarre i valori distinti ordinati di *column* e popolare un insieme da restituire (scegliere opportunamente in Set da

*utilizzare).*

```
public enum QUERY_TYPE {  
  
    MIN, MAX  
  
}
```

- Modificare nella classe **Data** la dichiarazione del membro **data** come segue

```
private List<Example> data=new ArrayList<Example>();
```

- Rimpiazzare il costruttore della classe **Data** con un costruttore che si occupa di caricare i dati (schema e esempi) di addestramento da una tabella della base di dati. Il nome della tabella è un parametro del costruttore. Il costruttore solleva una eccezione di tipo `TrainingDataException` se la connessione al database fallisce, la tabella non esiste, la tabella ha meno di due colonne, la tabella ha zero tuple, l'attributo corrispondente all'ultima colonna della tabella non è numerico.
- Modificare **MainTest** in modo da acquisire il nome della tabella contenente i dati per l'apprendimento da tastiera.