

## Esercitazione 6 – JDBC

- 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

`String DRIVER_CLASS_NAME = "com.mysql.cj.jdbc.Driver";` (Verificare l'effettivo nome della classe Driver nel jar che si decide di utilizzare; Per utilizzare questo Driver scaricare e aggiungere al classpath il connettore mysql connector)

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

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

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

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

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

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

`Connection conn;` gestisce una connessione

### Metodi

`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

`Connection getConnection();` restituisce `conn`;

`void closeConnection();` chiude la connessione `conn`;

- Definire la classe `Table_Schema` (fornita da docente) che modella lo schema di una tabella nel database relazionale
- Definire la classe `NoValueException` che estende `Exception` per modellare l'assenza di un

valore all'interno di un resultset

- 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> getDistinctTransazioni(String table) throws SQLException, EmptySetException
```

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

**Output:** Lista di transazioni distinte 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)

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

**Input:** Nome della tabella, 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 Object getAggregateColumnValue(String table, Column column, QUERY_TYPE aggregate) throws SQLException, NoValueException
```

**Input:** Nome di tabella, nome di colonna , operatore SQL di aggregazione (min,max)

**Output:** Aggregato cercato.

**Comportamento:** Formula ed esegue una interrogazione SQL per estrarre il valore aggregato (valore minimo o valore massimo) cercato nella colonna di nome `column` della tabella di nome `table`. Il metodo solleva e propaga una `NoValueException` se il resultset è vuoto o il valore

calcolato è pari a null

N.B. aggregate è di tipo QUERY\_TYPE dove QUERY\_TYPE è la classe enumerativa fornita dal docente

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

- Rimuovera la inner class [Example](#) e usare al suo posto la classe [database.Example](#). Rimpiazzare il costruttore della classe [Data](#) con un costruttore che si occupa di caricare i dati di addestramento da una tabella della base di dati. Il nome della tabella è un parametro del costruttore.