



# Progetto e persistenza dei dati

Andrea Polini

Ingegneria del Software  
Corso di Laurea in Informatica

# Interazione con DB relazionale

Il contesto specifico è quello di un programma Java che voglia interagire con un DBMS relazionale:

- Scaricare il JDBC connector (file Jar generalmente scaricabile dal sito del DBMS utilizzato)
- Aggiungere il file al classpath/Maven/Gradle
- Studiare le API del package `java.sql`

# Connessione al DB usando java.sql.DriverManager

```
public class DBManager { private static DBManager instance;
    private String url, user, pwd; private Connection conn = null;

    private DBManager() { }

    ...

    private void connect() { try {
        Class.forName("org.postgresql.Driver"); } catch
        (ClassNotFoundException e) { System.out.println("Where is your
        PostgreSQL JDBC Driver? " + "Include in your library
        path!"); e.printStackTrace(); } try { conn =
        DriverManager.getConnection(url, user, pwd);
        System.out.println("Database connected, ready to go!"); }
        catch (SQLException e) { System.out.println("Problems in
        opening a connection to the DB"); e.printStackTrace(); }

    ...
}
```

# Test della connessione al DB

```
public boolean DBtest() { Boolean result = true; try { if (conn ==
    null || !conn.isClosed()) { connect(); result = false; }
    DatabaseMetaData data = conn.getMetaData();
    System.out.println("Details on DBMS - " +
        data.getDatabaseProductName() + "\n" + " version: " +
        data.getDriverMajorVersion() + "\n" + " catalogs: " +
        data.getCatalogs().getCursorName() + "\n" + " schemas: " +
        data.getSchemas().getRow() + "\n"); close(); } catch
    (SQLException e) { e.printStackTrace(); } return result; }
```

# Esecuzione di una Query e gestione dei risultati

```
public ArrayList<DatiTavolo> getTavoli() { String SQL = "SELECT *  
FROM pizzeria4.tavolo;"; int count = 0; ArrayList<DatiTavolo>  
tavoli = new ArrayList<DatiTavolo>(); DatiTavolo tavolo; int i =  
1; try (Connection conn = DriverManager.getConnection(url, user,  
pwd); statement stmt = conn.createStatement(); ResultSet rs =  
stmt.executeQuery(SQL);) { while (rs.next()) { i = 1; tavolo =  
new DatiTavolo(rs.getString(i++), rs.getInt(i++),  
rs.getString(i)); tavoli.add(tavolo); } } catch  
(SQLException ex) { ex.printStackTrace(); } return tavoli; }
```

# Impedance Mismatch

Il problema si riferisce al fatto che i modelli OO e relazionali hanno una concezione differente del modello dei dati:

- Uso di puntatori di memoria
- Relazioni e meccanismi di Join

Il recupero dei dati da DB relazionali richiede di **effettuare innumerevoli azioni ripetitive** per trasformare i dati recuperati dal DB in oggetti, o per preparare i dati incapsulati in oggetti al fine del riversamento nelle tabelle di un DB.

Definizione di framework per la gestione della persistenza “Object Relational Mapping” (ORM)

# Impedance Mismatch

Il problema si riferisce al fatto che i modelli OO e relazionali hanno una concezione differente del modello dei dati:

- Uso di puntatori di memoria
- Relazioni e meccanismi di Join

Il recupero dei dati da DB relazionali richiede di **effettuare innumerevoli azioni ripetitive** per trasformare i dati recuperati dal DB in oggetti, o per preparare i dati incapsulati in oggetti al fine del riversamento nelle tabelle di un DB.

Definizione di framework per la gestione della persistenza “**Object Relational Mapping**” (ORM)