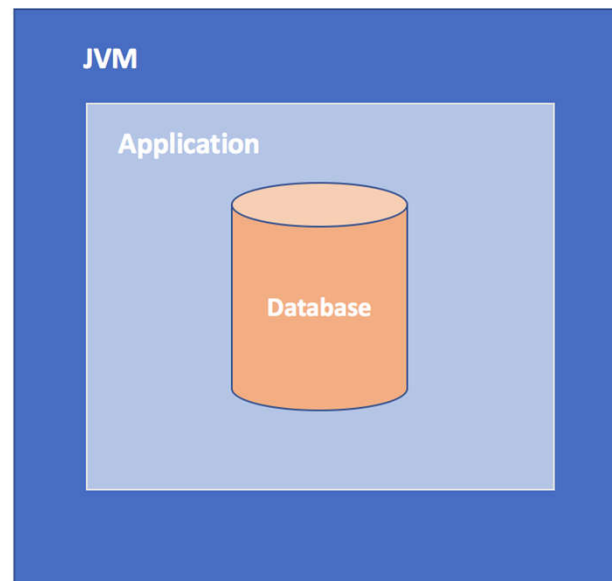# APACHE DERBY
EMBEDDED DATABASE

Embedded Database

- A Java application, accesses the database using a JDBC driver.

- An *embedded database* means that the database is integrated as an inseparable part of an application software.

- An embedded [database](#) is a [database](#) technology in which [database](#) management solutions are embedded into an application, rather than provided as standalone [database](#) systems.

Embedded Database

- An embedded [database](#) is a [database](#) technology in which [database](#) management solutions are embedded into an application, rather than provided as standalone [database](#) systems.
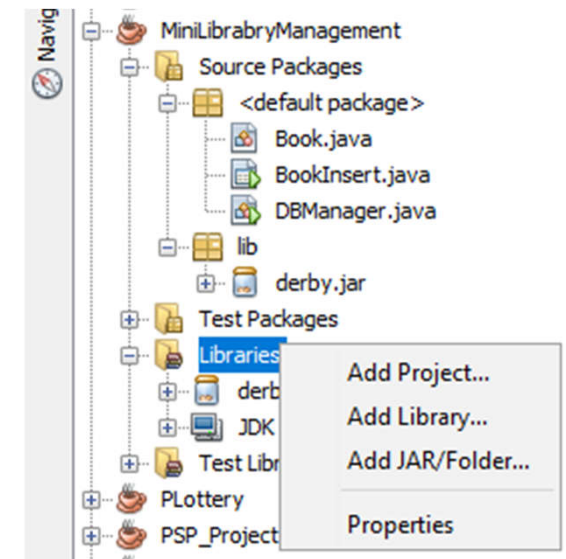
Embedded Database

- There are many embedded database systems..

  - **HSQLDB** (HyperSQL **Database**)
  - **H2**
  - **Apache Derby**

Embedded Database

- Copy derbey.jar from this path
  - C:\Program Files (x86)\Java\jdk1.8.0_131\db\lib

- Paste it on your project
- Right click on Libraries folder of your project from netbean's Project panel.
- Select 'Add JAR/Folder'
  - -> select the copied derbey.jar file from your project folder

Embedded Database

- Code:
  - Declare these variables

```java
private static final String DRIVER = "org.apache.derby.jdbc.EmbeddedDriver";
private static final String JDBC_URL = "jdbc:derby:ProjectName;create=true";
```

  - create=true // means if there is no database with this name, create a new one.

  - Create a connection:

```java
try {
    Connection con = DriverManager.getConnection(JDBC_URL);
} catch (SQLException ex) {
    System.out.println("*****database connection error - "+ex.toString());
}
```

Embedded Database

- Code:
  - Create your tables if not existed (first time only)

```java
try {
    DatabaseMetaData dbm = con.getMetaData();

    if(!dbm.getTables(null, null, "BOOK", null).next()){
        String sql1 = "CREATE TABLE BOOK ( "
                + "bID            VARCHAR(10), "
                + "bName          VARCHAR(40), "
                + "author         VARCHAR(40), "
                + "publication    VARCHAR(40), "
                + "isbn VARCHAR(20), "
                + "subject VARCHAR(30), "
                + "status INTEGER)";
        try {
            Statement stmt = con.createStatement();
            stmt.execute(sql1);
            System.out.println("book table crated");
        } catch (SQLException ex) {
            System.out.println("sql error "+ex.getMessage());
        }
    }
} catch (SQLException ex) {
```

Embedded Database

- Code:
  - Inserting into tables

```java
String sql = "INSERT INTO BOOK (bID, bName, author, "
        + "publication, isbn, subject, status) "
        + "VALUES (\'" + this.getNewBookID() +"\',"
        + "\'" + b.getbName() +"\',"
        + "\'" + b.getAuthor()+"\',"
        + "\'" + b.getPublication()+"\',"
        + "\'" + b.getIsbnNo()+"\',"
        + "\'" + b.getSubject()+"\',"
        + " 1 )";
System.out.println(sql);
Statement stmt;
try {
    stmt = con.createStatement();
    stmt.execute(sql);
} catch (SQLException ex) {
    javax.swing.JOptionPane.showConfirmDialog(null, "insert error "+ex.getMessage());
}
```

Embedded Database

- Code:
  - Fetch data

```java
String sql = "SELECT * FROM BOOK";
Statement stmt;
try {
    stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery(sql);
    while(rs.next()){
        System.out.println("name "+rs.getString("bName"));
        System.out.println("name "+rs.getString("bID"));
        System.out.println("author"+rs.getString("author"));
        System.out.println("isbn"+rs.getString("isbn"));
        System.out.println("-------------\n\n");
    }
} catch (SQLException ex) {
    Logger.getLogger(DBManager.class.getName()).log(Level.SEVERE, null, ex);
}
```

- https://www.youtube.com/watch?v=9GEW6kv4dKU

- http://db.apache.org/derby/quick_start.html
- http://db.apache.org/derby/papers/DerbyTut/embedded_intro.html

Links for UI design

- https://www.youtube.com/watch?v=uZ4ufLRZNHo
- https://www.youtube.com/watch?v=XAowXcmQ-kA&t=1404s
- https://www.youtube.com/watch?v=hYPa3P_BImA
- https://www.youtube.com/watch?v=zA82Mp5BXC4