

# 3. Бази данни

# Типове бази данни

- Не-Релационна - данните са представени чрез връзки между самите данни
- Релационна - съхраняват данните по предварително структуриран начин - в таблици, подредени в редове (записи) и колони

# Популярни SQL заявки

- CREATE DATABASE <db-name>

- CREATE TABLE <table-name> (

Id

SERIAL PRIMARY KEY

NOT

NULL,

kol2

TEXT

NOT NULL,

Kol3

REAL

NOT NULL)

QUERY X

FROM <table-name>

WHERE X > 3

ASC

# Как се създава връзката база - java код

- Драйвери - това са библиотеки, които знаят как да достъпваш конкретен тип база данни
- Типове
  - JDBC - low-level тип драйвери - използват се SQL заявки, за да се говори с базата
    - Ние ще започнем с MySQL
  - JPA - използват се джава методи за да се говори с базата. Пример за JPA е Hibernate
- При JDBC типовете библиотеки, драйвъра трябва да се 'зареди' в кода, за да се използва, при JPA той се указва в конфигурационни файлове

# Създаване на базата

- Може да стане от java кода, може и през командния ред
  - Отворете MySQL Shell
  - Създайте базата: `create database bazatami;`

# Създаване на таблица

```
String uri = "jdbc:mysql://localhost:3306/<db-name>";  
Properties props = setLoginForDB(); //set the db user  
Class.forName("com.mysql.jdbc.Driver"); //load the MySQL driver  
Connection conn = DriverManager.getConnection(uri, props);
```

```
Statement stmt = conn.createStatement();  
createTable(stmt);
```

```
conn.setAutoCommit(false);  
insertRows(stmt);  
conn.commit();  
stmt.close();
```

```
conn.close();
```

# Database login

- Необходимо за да знае базата кой я използва и какви права има
- Пример:

```
Properties props = new Properties();
```

```
props.setProperty("user", "pesho");
```

```
props.setProperty("password", "parola");
```

# Create Table

```
String sql = "CREATE TABLE skisEtc " +  
            "(id      SERIAL PRIMARY KEY NOT NULL," + // auto-generated integer  
key  
            " product VARCHAR(50)      NOT NULL," + // text, 50 chars at most  
            " category VARCHAR(50)     NOT NULL," + // ditto  
            " price   REAL);           // floating-point  
stmt.executeUpdate(sql);
```

- Stmt е от тип Statement и се генерира от Connection обекта.  
Представява 1 заявка към базата.



# Четене на ред от базата

```
ResultSet rs = stmt.executeQuery("SELECT * FROM books;");

    while (rs.next()) {

        int id = rs.getInt("id");

        String product = rs.getString("product");

        String category = rs.getString("category");

        float price = rs.getFloat("price");

        System.out.format("%d %s %s %.2f\n", id, product,
category, price);
```

# Компилиране и стартиране

<https://drive.google.com/open?id=0B4vR4GxbKqXDZzMzRTFSSjNjNkE>

# Домашно:

- Направете прост уебсайт за заемане и връщане на книги
- Основната страница показва всички книги, там има и два бутона - вземи и върни книга
- Вземи книга отваря страница с поле за търсене, при търсене се отварят всички отговарящи книги и могат да се изберат
- При избиране, книгата се маркира като взета/върнатата