

Webové a nativní aplikace

Toto cvičení může sloužit jako návod pro tvorbu jednoduché webové nebo nativní klientské aplikace Vaší semestrální práce. Pro větší objektivitu jsou ukázány programy jak pro komerční SŘBD Oracle, tak pro open-source SŘBD PostgreSQL.

Vytvoření ukázkových databází

Pro jednoduchost vytvoříme v databázi o jedné tabulce, která bude uchovávat základní informace o osobách v tabulce OSOBY. Pro SŘBD Oracle použijeme skript [osoby_oracle.sql](#), pro SŘBD PostgreSQL skript [osoby_postgres.sql](#). Důvodem použití dvou skriptů jsou různé datové typy obou databázových systémů.

Protože oba skripty budeme spouštět přímo ze stroje, kde jsou instalovány oba SŘBD, je nutné je přenést na daný stroj např. programem WinSCP, poté se na tento stroj přihlásit prostřednictvím SSH2 klienta, např. PuTTY a oba skripty vykonat z adresáře, kam byly uloženy. Pro SŘBD Oracle to bude vypadat např. takto:

```
$ sqlplus
```

```
SQL*Plus: Release 11.2.0.2.0 Production on Mon Mar 23 15:16:19 2015
```

```
Copyright (c) 1982, 2010, Oracle. All rights reserved.
```

```
Enter user-name: login
```

```
Enter password: *****
```

```
Connected to:
```

```
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

```
SQL> start osoby_oracle.sql
```

```
Table created.
```

```
1 row created.
```

```
1 row created.
```

```
1 row created.
```

```
Commit complete.
```

```
SQL> exit
```

```
Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

```
$
```

Pro SŘBD PostgreSQL to vypadá následovně:

```
$ psql -d db2_vyuka -U db2_vyuka
```

```
Password for user db2_vyuka:
```

```
psql (9.1.15)
```

```
Type "help" for help.
```

```
db2_vyuka=> \i osoby_postgres.sql
```

```
CREATE TABLE
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
db2_vyuka=> \q
```

```
$
```

Nativní aplikace: Java + JDBC

Tvorba nativní aplikace v Javě využívá pro komunikaci s databází rozhraní JDBC (*Java DataBase Connectivity*). Toto rozhraní je navrženo tak, aby bylo možné pro ukládání dat použít téměř jakékoli SŘBD a zdrojový kód programu přitom zůstane stejný. Tato výhoda je vykoupena nutností každého výrobce SŘBD vytvořit odpovídající distribuci JDBC, někdy i několik verzí.

Verze JDBC se dají kategorizovat podle:

- verze instalovaného SŘBD serveru
- verze instalované Javy resp. JDK

Mezi základní funkce JDBC můžeme řadit tyto:

`getConnection()`
poskytuje připojení do databáze konkrétnímu uživateli.

`prepareStatement()`
umožňuje připravit příkaz jazyka SQL, který bude v databázi spuštěn. Tento příkaz může při spuštění požadovat parametry.

`executeQuery()`
`executeUpdate()`
umožňuje vykonat (spustit) připravený příkaz jazyka SQL v databázi.

`close()`
ukončení spojení s databází

Ukázkové programy v Javě, které komunikují s databází Oracle nebo PostgreSQL jsou ke stažení níže, včetně potřebných distribucí JDBC:

- Program [OracleJDBC.java](#) komunikující se SŘBD Oracle a příslušná distribuce JDBC: [ojdbc6.jar](#).
- Program [PostgresJDBC.java](#) komunikující se SŘBD PostgreSQL a příslušná distribuce JDBC: [postgresql-9.1-903.jdbc4.jar](#).

Programy jsou bohatě komentovány, ukazují základní operace s daty v databázi a jsou téměř identické. Oba programy se liší pouze v řetězci použitého JDBC driveru a v řetězci, sloužící k připojení k databázi:

	název třídy ovladače	připojovací řetězec do databáze
Oracle	<code>oracle.jdbc.driver.OracleDriver</code>	<code>jdbc:oracle:thin:@students.kiv.zcu.cz:1521:students</code>
PostgreSQL	<code>org.postgresql.Driver</code>	<code>jdbc:postgresql://students.kiv.zcu.cz:5432/db2_vyuka</code>

Pokud použijeme vývojové prostředí Eclipse pro editaci a spuštění programu, budeme postupovat takto:

1. založíme projekt
2. v projektu vytvoříme adresář `lib` a do něj importujeme příslušnou JDBC knihovnu
3. ve vlastnostech projektu nastavíme cestu k importované JDBC knihovně
4. do projektu naimportujeme daný zdrojový soubor
5. zdrojový soubor zeditujeme - doplníme správné informace pro spojení s databází
6. program spustíme a otestujeme

Webová aplikace: [PHP 5 + Oracle](#)

Pro komunikaci se SŘBD Oracle se v PHP již používá funkcí *Oracle Call Interface*, zkráceně OCI. Tato množina funkcí je dostupná v modulu/rozšíření nesoucí název `php_oci8`. Toto rozšíření poskytuje celou řadu funkcí, jak pracovat s databází. Mezi nejdůležitější funkce můžeme zařadit tyto:

`oci_connect()`

poskytuje připojení do databáze konkrétnímu uživateli.

`oci_parse()`

umožňuje připravit příkaz jazyka SQL, který bude v databázi spuštěn. Tento příkaz může při spuštění požadovat parametry.

`oci_execute()`

umožňuje vykonat (spustit) připravený příkaz jazyka SQL v databázi.

`oci_close()`

ukončení spojení s databází

Ukázkový program v PHP, který komunikuje s databází Oracle je [zde](#). Program je bohatě komentován a ukazuje základní operace s daty v databázi. Pro jeho bezchybný běh je nutné odstranit koncovku TXT a umístit jej do příslušného publikačního adresáře. Pro studijní účely se jedná o adresář `~/public-kiv/public_html` na stroji `students.kiv.zcu.cz`.

Výsledná aplikace je poté dostupná přes následující

URL:`http://students.kiv.zcu.cz/~login/oracle_oci.php`

Webová aplikace: [PHP 5 + PostgreSQL](#)

Pro komunikaci se SŘBD PostgreSQL se v PHP používá funkcí, které jsou dostupné v modulu/rozšíření `php_pgsq1`. Toto rozšíření poskytuje celou řadu funkcí, jak pracovat s databází. Mezi nejdůležitější funkce můžeme zařadit tyto:

`pg_connect()`

poskytuje připojení do databáze konkrétnímu uživateli.

`pg_prepare()`

umožňuje připravit příkaz jazyka SQL, který bude v databázi spuštěn. Tento příkaz může při spuštění požadovat parametry.

`pg_execute()`

umožňuje vykonat (spustit) připravený příkaz jazyka SQL v databázi.

`pg_close()`

ukončení spojení s databází