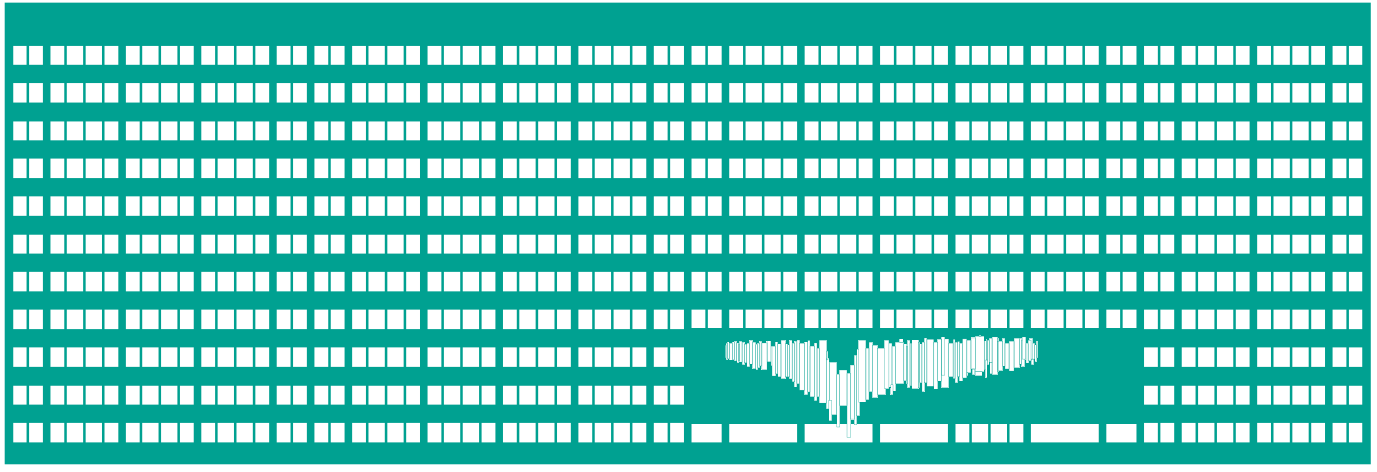
VŠB TECHNICKÁ UNIVERZITA OSTRAVA

VSB TECHNICAL UNIVERSITY OF OSTRAVA



www.vsb.cz

Database and Information Systems

db.cs@vsb.cz

Department of Computer Science
Faculty of Electrical Engineering and Computer Science
VSB - Technical University of Ostrava

2019/2020

Task 1: Stored procedures and functions

1. Create stored procedure[1] `AddStudent` with 4 parameters `p_login, p_fname, p_lname, p_tallness`, which will insert new record into table. Run procedure using command `EXECUTE`.
2. Create function[2] `FAddStudent`, which will work the same way like the procedure `AddStudent` and will return 'ok', if record is successfully inserted, otherwise 'error' (Use `Exception`). Use `dbms_output.put_line` to print the result of the function[3].

---

[1] `https://docs.oracle.com/en/database/oracle/oracle-database/18/lnpls/CREATE-PROCEDURE-statement.html`

[2] `https://docs.oracle.com/en/database/oracle/oracle-database/18/lnpls/CREATE-FUNCTION-statement.html`

[3] Do not forget `SET SERVEROUTPUT ON`

Task 2: Table Teacher

```
CREATE TABLE Teacher (
login CHAR(6) NOT NULL PRIMARY KEY,
fname VARCHAR2(30) NOT NULL,
lname VARCHAR2(50) NOT NULL,
department INT NOT NULL,
specialization VARCHAR2(30) NULL);
```

Task 2: Variables

1. Create stored procedure `StudentBecomeTeacher` with 2 parameters `p_login` and `p_department`, which will move student with login `p_login` from table `Student` into table `Teacher` (command SELECT INTO[4]).
2. Modify stored procedure `StudentBecomeTeacher` to be one transaction.
3. Create stored procedure `AddStudent2` with 3 parameters `p_fname`, `p_lname` a `p_tallness`, which will create login from last name (parameter `p_lname`) with adding '00' and insert record into table (use SUBSTR[5]).

---

[4] https://docs.oracle.com/en/database/oracle/oracle-database/18/lnpls/SELECT-INTO-statement.html

[5] https://docs.oracle.com/en/database/oracle/oracle-database/18/lnpls/plsql-language-fundamentals.html

Table `Student`

Note: unless noted otherwise all tasks are for table `Student`:

```
CREATE TABLE Student (
  login     CHAR(6) PRIMARY KEY,
  fname     VARCHAR(30) NOT NULL,
  lname     VARCHAR(50) NOT NULL,
  email     VARCHAR(50) NOT NULL,
  tallness INT NOT NULL);
```

Task 3: Control Structures[6]

1. Add to table `Student` attribute `isTall`, which can be 0 or 1.
2. Create stored procedure `IsStudentTall` with one input parameter `p_login`, which will find record with current login and set attribute `isTall` to 0, if attribute `tallness` is less than the average tallness, otherwise 1 (use command IF).
3. Create function `LoginExist` with one input parameter `p_login`, which will return true if record with current login `p_login` exists. Use function `LoginExist` to extend procedure `AddStudent2`, which will be generating new login until it will find unused login (use command LOOP).

[6] `https://docs.oracle.com/en/database/oracle/oracle-database/18/lnpls/plsql-control-statements.html`

Task 4: Cursor

1. Change stored procedure `IsStudentTall` to go through all records and set corresponding atribute `isTall`. In this case, the procedure will be without parameter. Use type `student%ROWTYPE` and commands OPEN, FETCH, CLOSE.
2. Change stored procedure `IsStudentTall` to use cycle FOR.