MTAT.03.105
Introduction to Databases

Lecture #11
Function and Procedures

Ljubov Jaanuska
(ljubov.jaanuska@ut.ee)
Lecture 10. Summary

• Views
• Revision on queries
Lecture 11. What will you learn

• Functions
• Procedures
• Revision on queries
3GL inside 4GL

- Functions
  - Return only **one values**
- Procedures
  - Statements which do not return anything (insert, delete, update etc.)
  - Return a **table**
Functions

SELECT POWER(2, 6);

CREATE FUNCTION f_name (input_params)
RETURNS <data_type>
BEGIN
    ....
END;
Example 1

Function to calculate power

```sql
CREATE FUNCTION f_power ( int1 integer, int2 integer)
RETURNS INTEGER
BEGIN
RETURN power(int1,int2);
END;

SELECT f_power(3,2);
SELECT f_power(4,3);
```
Functions

SQL is 4GL

3GL features:

• create variable:
  DECLARE <v_name> <data_type>

• assign value:
  SET <v_name> = <value>

• assign query single result:
  SELECT ... INTO <v_name> FROM ...
Example 2
Function to calculate power

```
ALTER FUNCTION f_power ( int1 integer, int2 integer)
RETURNS  INTEGER
BEGIN
  DECLARE  p_res INTEGER;
  SET  p_res =power(int1,int2);
  RETURN  p_res;
END;

SELECT f_power(3,2);
SELECT f_power(4,3);
```
Example 3

Function which gets player’s id as an input and returns player’s first name

```
CREATE FUNCTION f_name( i_id integer)
RETURNS varchar(50)
BEGIN
    DECLARE first_name varchar(50);
    SELECT p_name INTO first_name FROM Player
    WHERE p_id = i_id;
    RETURN first_name;
END;

SELECT f_name(88);
SELECT p_id, f_name(p_id) FROM Player;
```
CREATE PROCEDURE p_name ({{<params>}})
[RESULT ({{column_names}})]
BEGIN
...
END

NB! params: {IN | OUT | INOUT} <name> <data_type>
Example 4

Procedure to add a new club into table Club

```sql
CREATE PROCEDURE p_new_club
(IN n_name VARCHAR(100), IN n_address VARCHAR(100) )
BEGIN
    INSERT INTO Club (c_name, address) VALUES (n_name, n_address);
END;

CALL p_new_club ('Valga Valge', 'Valga');
```
Example 5

Procedure to add a new club into table Club and show a message

```
ALTER PROCEDURE p_new_club
(IN n_name VARCHAR(100),
 IN n_address VARCHAR(100),
OUT n_id INTEGER)
BEGIN
    DECLARE i_id INTEGER;
    INSERT INTO Club (c_name, address)
    VALUES (n_name, n_address);
    SELECT @@identity INTO i_id;
    MESSAGE 'New club id: ' || i_id TO CLIENT;
    SET n_id = i_id;
END;
```

```
CREATE VARIABLE p_int INTEGER;
CALL p_new_club ('MyPHPAdmin club', 'Narva', p_int);
SELECT p_int ;
```
Example 6

Procedure which show club members as a table

```
CREATE PROCEDURE p_club_members(IN i_club_id INTEGER)
RESULT (
    first_name VARCHAR(50),
    second_name VARCHAR(50),
    c_date DATE)
BEGIN
    SELECT p_name, surname, CURRENT DATE FROM Player
    WHERE club_id = i_club_id ORDER BY p_name;
END;

CALL p_club_members(51);
SELECT * FROM p_club_members(51);
```
Functions vs Procedures

Functions
• Return a single value of the pre-defined data type; the data type is defined in `RETURNS`
• Used in expressions (e.g. inside functions and procedures)
• Only input parameters
• `SELECT`

Procedures
• Parameter types: `IN`, `INOUT` ja `OUT`
• May return several parameters and a table
• Used in queries after `FROM`
• `CALL`
Delete

DROP FUNCTION <f_name>

DROP PROCEDURE <f_name>
Revision on queries (self-study/group work)

1. Which player has won the most games?
2. Which club is the most successful?
3. What is an average number of points scored by each club?
4. Which club scored the most points?
5. How many clubs competed in each tournament?
6. Which district provides the most players?
7. How many people are there with the same first names?
8. How many players are in each club?
9. Which name, surname beginning letter is the most common?
10. Show the ID's of the players when the result of the game was 2:1.
Home Tasks

1. Create function `f_club_size()` which returns the number of the club members. An input parameter is the club id.
2. Create function `f_name_surname()` which returns the player’s name in the form of “surname, name”. An input parameter is the player’s id.
3. Create function `f_player_load()` which returns the number of the games the player has played so far. An input parameter is the player’s id.
4. Create function `f_player_wins()` which returns the number of wins of the player during the specific tournament. Input parameters are player’s id and tournament id.
5. Create function `f_player_tournament_points()` which returns the number of points (= half of the result) that a player earned during the tournament. Input parameters are player’s id and tournament id.
6. Create procedure `p_new_player()` which adds a new player into table Player. Input parameters are player’s first name, last name and club id.
Home Tasks

7. Create procedure `p_info()` which shows a table with two columns and contains the following information:
   - club names and number of club members in each club (use function `f_club_size`)
   - tournament names and the number of games in each tournament
   - player’s name and the number of games (use functions `f_name_surname` and `f_player_load`)

8. Create procedure `p_top_10()` which shows en best player in the specific tournament. Input parameters are the tournament id. Use view `v_rating`.

9. Create procedure `p_win_draw_loss()`, which shows numbers of wins, draws and losses for each player in the specific tournament. The result must contain the following columns: player’s id, first name, last name, nr of wins, nr of draws, nr of losses. Use function `f_player_wins` and similar to it.
Anonymous feedback

PLEASE SUBMIT via Moodle by **25.04 23:55**
- .db file
- .log file

Please fill in an anonymous feedback if you haven’t done it yet:
[https://goo.gl/forms/1TXenNUTKgKoNZwh2](https://goo.gl/forms/1TXenNUTKgKoNZwh2)

Each answer is important!