

Indian Institute of Information Technology Sri City

Database Management Systems LAB-09

Date: 02/11/2021

TOPIC: Stored Functions, Error Handling in Stored Procedures

Instructors: Dr. Odelu Vanga, Dr. Rakesh and Dr. Annushree

STORED FUNCTIONS:

The stored function is almost similar to the procedure in MySQL, but it has some differences that are as follows:

- The function parameter may contain only the **IN parameter** but can't allow specifying this parameter, while the procedure can allow **IN, OUT, INOUT parameters**.
- The stored function can return only a single value defined in the function header.
- The stored function may also be called within SQL statements.
- It may not produce a result set.

Thus, we will consider the stored function when our program's purpose is to compute and return a single value only or create a user-defined function.

SYNTAX:

```
DELIMITER $$  
CREATE FUNCTION function_name (param1 datatype, param2 datatype, ...)  
RETURNS datatype  
[NOT] {Characteristics}  
BEGIN  
-- fun body  
END $$  
DELIMITER ;
```

Parameter Name	Descriptions
FUNCTION_NAME	It is the name of the stored function that we want to create in a database. It should not be the same as the built-in function name of MySQL.

PARAM	It contains the list of parameters used by the function body. It does not allow to specify IN, OUT, INOUT parameters.
Datatype	It should any valid MySQL data type.
characteristics	The CREATE FUNCTION statement only accepted when the characteristics (DETERMINISTIC, NO SQL, or READS SQL DATA) are defined in the declaration.
fun_body	This parameter has a set of SQL statements to perform the operations. It requires at least one RETURN statement. When the return statement is executed, the function will be terminated automatically.

NOTE: To skip specifying ‘characteristics ‘ in function syntax run the below command

“SET GLOBAL log_bin_trust_function_creators = 1;”

EXAMPLE:

```
delimiter //
create function my_fun(value numeric(12,2))
returns int
deterministic
begin
declare r int;
select count(*) into r from account where balance>value;
return (r);
end//
```

Calling a function:

```
select my_fun(500);
select balance,my_fun(balance) from account order by balance;
```

Dropping a function:

```
DROP FUNCTION [IF EXISTS] function_name;
```

To know the procedures already existing in database:

```
SHOW FUNCTION STATUS WHERE db = <database_name>;

SELECT routine_name
FROM
information_schema.routines
WHERE
routine_type = 'FUNCTION' AND routine_schema = '<your_db_name>';
```

Example problem:

1. Display all account numbers and 'Level' of the customer based on the balance amount.
(Level='PLATINUM' if balance >=900 ; Level='Gold' if balance>=700 and <900;
Level='Silver' if balance<700)

```
delimiter //
create function customerlevel(id varchar(10))
returns varchar(20)
deterministic
begin
    declare bal numeric(12,2) default 0;
    declare level varchar(20);

    select balance into bal from account natural join depositor where
account number = id;
    if bal > 700 then
        set level='PLATINUM';
    elseif bal<=700 and bal>300 then
        set level='GOLD';
    else
        set level='SILVER';
    end if;
    return (level);
end //

mysql> select account_number,customerlevel(account_number) from depositor;
```

```

+-----+-----+
| account_number | customerlevel(account_number) |
+-----+-----+
| A-101          | GOLD                           |
| A-102          | GOLD                           |
| A-201          | PLATINUM                       |
| A-215          | GOLD                           |
| A-217          | PLATINUM                       |
| A-222          | GOLD                           |
| A-305          | GOLD                           |
+-----+-----+

```

MYSQL ERROR HANDLING IN STORED PROCEDURES:

- When an error occurs inside a stored procedure, it is important to handle it appropriately, such as continuing or exiting the current code block's execution, and issuing a meaningful error message.
- MySQL provides an easy way to define handlers that handle from general conditions such as warnings or exceptions to specific conditions e.g., specific error codes.

DECLARING A HANDLER:

- To declare a handler, you use the DECLARE HANDLER statement as follows:

```
DECLARE action HANDLER FOR condition_value statement;
```
- If a condition whose value matches the `condition_value`, MySQL will execute the statement and continue or exit the current code block based on the action.

The action accepts one of the following values:

- CONTINUE : the execution of the enclosing code block (BEGIN ... END) continues.
- EXIT : the execution of the enclosing code block, where the handler is declared, terminates.

The `condition_value` specifies a particular condition or a class of conditions that activate the handler. The `condition_value` accepts one of the following values:

- A MySQL error code.
- A standard SQLSTATE value.

The statement could be a simple statement or a compound statement enclosing by the BEGIN and END keywords.

MYSQL ERROR HANDLING EXAMPLES

1. DECLARE CONTINUE HANDLER FOR SQLEXCEPTION
SET hasError = 1;
2. DECLARE CONTINUE HANDLER FOR NOT FOUND
SET RowNotFound = 1;
3. DECLARE CONTINUE HANDLER FOR 1062
SELECT 'Error, duplicate key occurred';

EXAMPLE:

```
DELIMITER $$
CREATE PROCEDURE emp_insert(
    IN name VARCHAR(20),
    IN id INT
)
BEGIN
    -- exit if the duplicate key occurs
    DECLARE EXIT HANDLER FOR 1062
    BEGIN
        SELECT CONCAT('Duplicate key (' ,name,',',',id,') occurred') AS message;
    END;

    -- insert a new row into the EMP table
    INSERT INTO emp(emp id,emp name)
    VALUES(id,name);

    -- return the row inserted recently
    Select * from emp where emp_id=id;
END $$
```

Now, Call the procedure emp_insert() to inert records into the table.

```
Call emp_insert('John',10);
Call emp_insert('mart',20);
Call emp_insert ('Jo',30);
```

Fourth, attempt to insert a row whose values already exist in the emp table:

```
Call emp_insert('mart',20);
```

REFERENCES:

Course material is inspired from the following websites and some examples are taken from geeksforgeeks, w3schols only for student understanding perspective.

1. <https://www.mysqltutorial.org/mysql-stored-function/>
2. <https://www.mysqltutorial.org/mysql-error-handling-in-stored-procedures/>