

Experiment No 9

Aim: Implementation of PL/SQL Functions & Procedures

Class: SE Comp

Year: 2020-21

Performed by: Danyl Fernandes, 72

Performance Date: 18-04-2021

Theory:

Stored Functions:

- A stored function is a special kind stored program that returns a single value. Typically, you use stored functions to encapsulate common formulas or business rules that are reusable among SQL statements or stored programs.
- Different from a stored procedure, you can use a stored function in SQL statements wherever an expression is used. This helps improve the readability and maintainability of the procedural code.
- To create a stored function, you use the CREATE FUNCTION statement.

Syntax:

```
DELIMITER $$
CREATE FUNCTION function_name(
    param1,
    param2, ...
)
RETURNS datatype
[NOT] DETERMINISTIC
BEGIN
    -- statements
END $$
DELIMITER;
```

Example:

customers
* customerNumber
customerName
contactLastName
contactFirstName
phone
addressLine1
addressLine2
city
state
postalCode
country
salesRepEmployeeNumber
creditLimit

```
DELIMITER $$
CREATE FUNCTION CustomerLevel(
    credit DECIMAL(10,2)
)
RETURNS VARCHAR(20)
DETERMINISTIC
BEGIN
    DECLARE customerLevel VARCHAR(20);
    IF credit > 50000 THEN
        SET customerLevel = 'PLATINUM';
    ELSEIF (credit >= 50000 AND
        credit <= 10000) THEN
        SET customerLevel = 'GOLD';
    ELSEIF credit < 10000 THEN
        SET customerLevel = 'SILVER';
    END IF;
    -- return the customer level
    RETURN (customerLevel);
END$$
```

Stored Procedure:

- A procedure (often called a stored procedure) is a collection of pre-compiled SQL statements stored inside the database.
- It is a subroutine or a subprogram in the regular computing language.
- A procedure always contains a name, parameter lists, and SQL statements.
- We can invoke the procedures by using triggers, other procedures and applications such as Java, Python, PHP, etc.
- It was first introduced in MySQL version 5. Presently, it can be supported by almost all relational database systems.

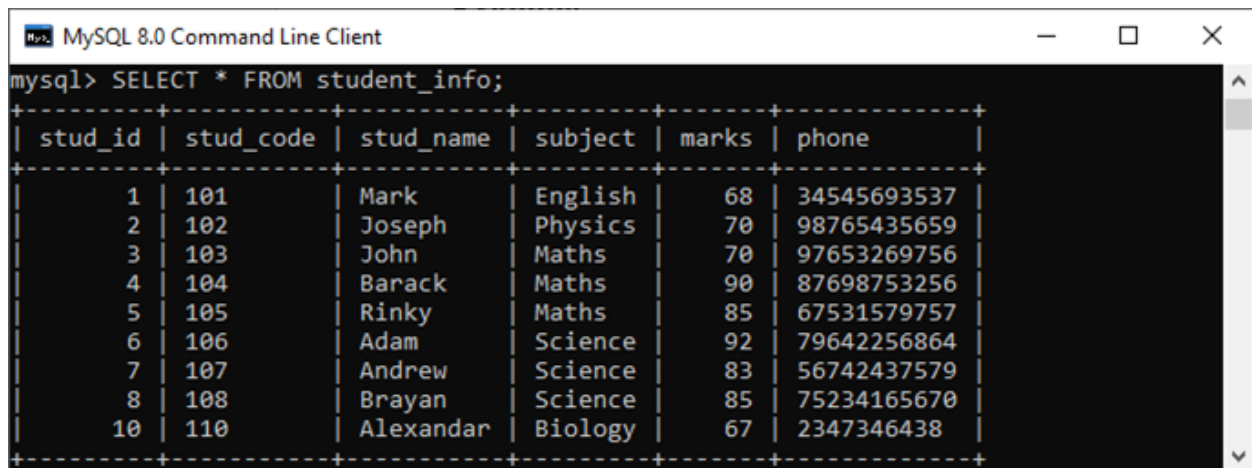
Features:

- Stored Procedure increases the performance of the applications. Once stored procedures are created, they are compiled and stored in the database.
- Stored procedure reduces the traffic between application and database server. Because the application has to send only the stored procedure's name and parameters instead of sending multiple SQL statements.
- Stored procedures are reusable and transparent to any applications.
- A procedure is always secure. The database administrator can grant permissions to applications that access stored procedures in the database without giving any permissions on the database tables.

Syntax:

```
DELIMITER &&
CREATE PROCEDURE procedure_name [[IN | OUT | INOUT]
parameter_name datatype [, parameter datatype]) ]
BEGIN
    Declaration_section
    Executable_section
END &&
DELIMITER ;
```

Example:



The screenshot shows a MySQL 8.0 Command Line Client window. The command prompt shows the command: `mysql> SELECT * FROM student_info;`. The result is a table with 6 columns: `stud_id`, `stud_code`, `stud_name`, `subject`, `marks`, and `phone`. The table contains 10 rows of data.

stud_id	stud_code	stud_name	subject	marks	phone
1	101	Mark	English	68	34545693537
2	102	Joseph	Physics	70	98765435659
3	103	John	Maths	70	97653269756
4	104	Barack	Maths	90	87698753256
5	105	Rinky	Maths	85	67531579757
6	106	Adam	Science	92	79642256864
7	107	Andrew	Science	83	56742437579
8	108	Brayan	Science	85	75234165670
10	110	Alexandar	Biology	67	2347346438

DELIMITER &&

CREATE PROCEDURE get_merit_student ()

BEGIN

 SELECT * FROM student_info WHERE marks > 70;

 SELECT COUNT(stud_code) AS Total_Student FROM student_info;

END &&

DELIMITER;

Procedures w/ Parameters:

IN Parameter example:

CREATE PROCEDURE my_proc_IN (IN var1 INT)

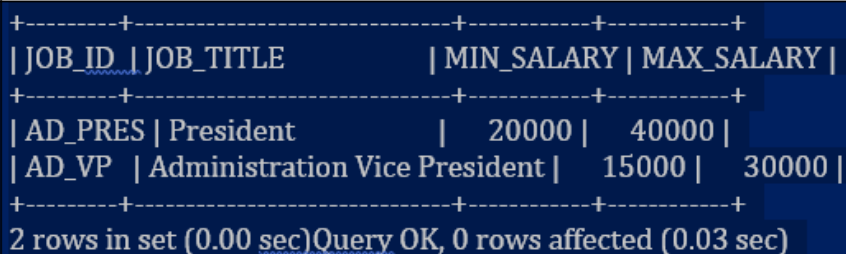
BEGIN

SELECT * FROM jobs LIMIT var1;

END\$\$

To execute the first 2 rows from the 'jobs' table execute the following command:

mysql> CALL my_proc_in(2)\$\$



The screenshot shows the output of the `CALL my_proc_in(2)$$` command. It displays the first two rows of the 'jobs' table. The output is a table with 4 columns: `JOB_ID`, `JOB_TITLE`, `MIN_SALARY`, and `MAX_SALARY`. The first row is for the President and the second row is for the Administration Vice President. Below the table, it says "2 rows in set (0.00 sec)Query OK, 0 rows affected (0.03 sec)".

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
AD_PRES	President	20000	40000
AD_VP	Administration Vice President	15000	30000

2 rows in set (0.00 sec)Query OK, 0 rows affected (0.03 sec)

```
mysql> CALL my_proc_in(5)$$
```

```
+-----+-----+-----+-----+
| JOB_ID | JOB_TITLE           | MIN_SALARY | MAX_SALARY |
+-----+-----+-----+-----+
| AD_PRES | President           | 20000      | 40000      |
| AD_VP   | Administration Vice | 15000      | 30000      |
| AD_ASST | Administration Assistant | 3000      | 6000      |
| FI_MGR  | Finance Manager     | 8200       | 16000      |
| FI_ACCOUNT | Accountant         | 4200       | 9000       |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)Query OK, 0 rows affected (0.05 sec)
```

OUT Parameter example:

The following example shows a simple stored procedure that uses an OUT parameter. Within the procedure MySQL MAX() function retrieves maximum salary from MAX_SALARY of jobs table:

```
CREATE PROCEDURE my_proc_OUT (OUT highest_salary INT)
BEGIN
SELECT MAX(MAX_SALARY) INTO highest_salary FROM JOBS;
END$$
```

In the body of the procedure, the parameter will get the highest salary from MAX_SALARY column. After calling the procedure the word OUT tells the DBMS that the value goes out from the procedure. Here highest_salary is the name of the output parameter and we have passed its value to a session variable named @M, in the CALL statement:

```
mysql> CALL my_proc_OUT(@M)$$
```

```
mysql> SELECT @M$$+-----+
| @M |
+-----+
| 40000 |
+-----+
1 row in set (0.00 sec)
```

Functions:

Query:

```
DELIMITER $
CREATE OR REPLACE function helloGreeting() returns varchar(20)
BEGIN
    DECLARE msg varchar(20);
    SET msg = "Yo, this is dan!";
    RETURN msg;
END;
$
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0077 seconds.)

```
CREATE OR REPLACE function helloGreeting() returns varchar(20) BEGIN DECLARE msg varchar(20); SET msg = "Yo, this is dan!"; RETURN msg;
END;
```

[\[Edit inline\]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Query:

```
show function status where db = 'exp3'
```

Your SQL query has been executed successfully.

```
SHOW FUNCTION STATUS WHERE db = 'exp3'
```

☐ Profiling [\[Edit inline\]](#) [\[Edit \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

+ Options

Db	Name	Type	Definer	Modified	Created	Security_type	Comment	character_set_client	collation_connection	Database Collation
exp3	helloGreeting	FUNCTION	root@localhost	2021-04-18 06:09:53	2021-04-18 06:09:53	DEFINER		utf8mb4	utf8mb4_unicode_ci	utf8mb4_gen

Query results operations

Query:

```
select helloGreeting()
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0018 seconds.)

```
select helloGreeting()
```

☐ Show all

Number of rows:

25



Filter rows:

Search this table

+ Options

helloGreeting()

Yo, this is dan!

Query:

```
DELIMITER $
CREATE OR REPLACE function getRankFromRating(
    rating int
) RETURNS varchar(20)
BEGIN
    DECLARE rank varchar(20);
    IF rating > 5 THEN
        SET rank= 'Excellent';
    ELSEIF rating > 4 && rating <= 5 THEN
        SET rank = 'Good';
    ELSE
        SET rank = 'fair';
    END IF;
    RETURN rank;
END;
$
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0114 seconds.)

```
CREATE OR REPLACE function getRankFromRating( rating int ) RETURNS varchar(20) BEGIN DECLARE rank varchar(20); IF rating > 5 THEN SET
rank= 'Excellent'; ELSEIF rating > 4 && rating <= 5 THEN SET rank = 'Good'; ELSE SET rank = 'fair'; END IF; RETURN rank; END;
```

[\[Edit inline\]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Query:
































```
select sid, sname, rating, getRankFromRating(rating)
as rank from sailor
```

✓ Showing rows 0 - 20 (21 total, Query took 0.0049 seconds.)

```
select sid, sname, rating, getRankFromRating(rating) as rank from
```

☐ Show all | Number of rows: 25 Filter rows:

+ Options

				sid	sname	rating	rank
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	Rahul	5	Good
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	Danyl	5	Good
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	Shrinath	5	Good
<input type="checkbox"/>	 Edit	 Copy	 Delete	4	Jack	4	fair
<input type="checkbox"/>	 Edit	 Copy	 Delete	5	Sean	3	fair
<input type="checkbox"/>	 Edit	 Copy	 Delete	6	Matt	3	fair
<input type="checkbox"/>	 Edit	 Copy	 Delete	7	Hans	4	fair
<input type="checkbox"/>	 Edit	 Copy	 Delete	8	Drishyam	2	fair
<input type="checkbox"/>	 Edit	 Copy	 Delete	9	Anakin	1	fair
<input type="checkbox"/>	 Edit	 Copy	 Delete	10	Bob	3	fair

Procedures:

Query:

```
DELIMITER $  
CREATE or REPLACE PROCEDURE greet()  
BEGIN  
    DECLARE msg varchar(20);  
    SET msg = 'Hello';  
SELECT msg;  
END;  
$
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0052 seconds.)

```
CREATE or REPLACE PROCEDURE greet() BEGIN DECLARE msg varchar(20); SET msg = 'Hello'; SELECT msg; END;
```

Query:

```
CALL greet();
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0020 seconds.)

```
CALL greet()  
.....
```

☐ Show all

Number of rows:

25



Filter rows:

Search this table

+ Options

msg

Hello

Query:

```
DELIMITER $
CREATE or REPLACE PROCEDURE getRankFromRating(
    IN rating int, OUT rank varchar(20)
)
BEGIN
DECLARE rank varchar(20);
IF rating > 4 THEN
    SET rank = 'excellent';
ELSE
SET rank= 'good';
END IF;
END;
$
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0068 seconds.)

```
CREATE or REPLACE PROCEDURE getRankFromRating( IN rating int, OUT rank varchar(20) ) BEGIN IF rating > 4 THEN SET rank = 'excellent';
ELSE SET rank= 'good'; END IF; END;
```

[\[Edit inline\]](#) [\[Edit\]](#) [\[Create PHP code\]](#)

Query:

```
call getRankFromRating(6, @rank);
select @rank
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

```
call getRankFromRating(6, @rank)
```

[\[Edit inline\]](#) [\[Edit\]](#) [\[Create PHP code\]](#)

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available. ⓘ

✓ Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
select @rank
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: | Filter rows:

+ Options

@rank
excellent

Query:

```
DELIMITER $
CREATE or REPLACE PROCEDURE getRecordByName(
    IN sname varchar(20)
)
BEGIN
    SELECT * FROM sailor WHERE name = sname;
END;
$
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0060 seconds.)

```
CREATE or REPLACE PROCEDURE getRecordByName( IN sname varchar(20) ) BEGIN SELECT * FROM sailor WHERE name = sname; END;
```

[\[Edit inline\]](#) [\[Edit\]](#) [\[Create PHP code\]](#)

Query:

```
CALL getRecordByName("Rahul")
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0009 seconds.)

```
CALL getRecordByName ("Rahul")
```

☐ Show all

Number of rows:

25



Filter rows:

Search this table

+ Options

sid	sname	address	rating	age
1	Rahul	Mumbai	5	34

Query:

```
DELIMITER $
CREATE or REPLACE PROCEDURE getRankFromRating(
    IN name varchar(20), OUT srnk int
)
BEGIN
    SELECT rating INTO srnk FROM sailor WHERE sname = name;
END;
$
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0074 seconds.)

```
CREATE or REPLACE PROCEDURE getRankFromRating( IN name varchar(20), OUT srnk int ) BEGIN DECLARE rank varchar(20); SELECT rating INTO
rank FROM sailor WHERE sname = name; END;
```

[\[Edit inline\]](#) [\[Edit\]](#) [\[Create PHP code\]](#)

Query:

```
CALL getRankFromRating("Shrinath", @rank);
SELECT @rank
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0069 seconds.)

```
CALL getRankFromRating("Shrinath", @rank)
```

[\[Edit inline\]](#) [\[Edit\]](#) [\[Create PHP code\]](#)

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available. ⓘ

✓ Showing rows 0 - 0 (1 total, Query took 0.0055 seconds.)

```
SELECT @rank
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

☐ Show all | Number of rows: | Filter rows:

+ Options

@rank

5