PL/SQL stands for “Procedural language extensions to SQL.”

|  |  |
| --- | --- |
| **SQL is declarative language.** | **PLSQL is procedural language.** |
| SQL can be embedded in PLSQL. | PLSQL can’t be embedded in SQL. |

|  |  |
| --- | --- |
| **No Supported Control Structures.** | **Control Structures are available Like, For loop, While loop.** |
| **Data variable are not available** | **Data variable are available.** |

### 

### [**What are the *differences between* stored *procedure* and *functions* in mysql**](http://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=0CEYQFjAD&url=http%3A%2F%2Fwww.allinterview.com%2Fshowanswers%2F28431.html&ei=8rdtUP68FcrZrQe9uICgAQ&usg=AFQjCNH2x6lb31zqb-d7sNbMMg8Mg7ctVw&sig2=PxGAO6vhNURXH7BC1lILxA&cad=rja)

**a.** Functions are normally used for **computations** where as

procedures are normally used for **executing business logic**.

d. Stored procedure returns always integer value by default

zero. Whereas function returns type could be scalar or

table or table values

**e.** Stored procedure is precompiled execution plan where as

functions are not.

**f.** A function can call directly by SQL statement like

select func\_name from dual while procedure cannot.

**g.**Stored procedure has the security and reduces the network

traffic and also we can call stored procedure in any no. of

applications at a time.

**h.** A Function can be used **in the** **SQL Queries** while a

procedure cannot be used **in SQL queries .**that cause a major

difference b/w function and procedures.

## PL/SQL Functionalities

|  |  |
| --- | --- |
| **Stored Procedure** | **Function** |
| is used to perform business logic. | is used to perform calculation. |
| must not have the return type. | must have the return type. |
| may return 0 or more values. | may return only one values. |
| We can call functions from the procedure. | Procedure cannot be called from function. |
| Procedure supports input and output parameters. | Function supports only input parameter. |
| Exception handling using try/catch block can be used in stored procedures. | Exception handling using try/catch can't be used in user defined functions. |

By default we use semicolon (;) as a delimiter.

To change delimeter in MySQL

DELIMITER ;

To call Procedure: **CALL procedureName( parameters ) ;**

**To show all Procedures: show procedure status;**

**To get** the **definition (code)** of a **stored procedure** using this **command** in **MySQL**:

**drop procedure proc-name;**

The **delimiter** is the character or string of characters which is used to complete an SQL statement.

A **stored procedure** can have **any number** of **statements**

**DELIMITER //**

**CREATE PROCEDURE GetAllEmpDepts()**

**BEGIN**

**SELECT fname, lname FROM employee;**

**SELECT dname, mgrssn FROM department;**

**END //**

**DELIMITER ;**

**DELIMITER $$** ->>First, change the default delimiter to $$

**CREATE PROCEDURE sp\_name()**

**BEGIN**

*--* ***statements***

**END** **$$** ->>to end the stored procedure.

**DELIMITER;** ->> Third, change the default delimiter back to a semicolon (;)

In MySQL, a parameter has one of three modes: **IN(default),OUT, or INOUT.**

**1. Stored Procedure.**

If you want to save this query on the database server for execution later, one way to do it is to use a stored procedure.

DELIMITER $$

CREATE PROCEDURE GetCustomers()

BEGIN

    SELECT customerName, city, state, postalCode, country

    FROM customers ORDER BY customerName;

END$$

DELIMITER ;

DELIMITER $$

**CREATE** **PROCEDURE** getCustomer( **IN** id INT)

BEGIN

    SELECT customerName, city, state, postalCode, country

    FROM customers where Cust\_Id=id;

END$$

DELIMITER ;

Once you save the stored procedure, you can invoke it by using the CALL statement:

**CALL GetCustomers();**

Funtion

Different from a [stored procedure](https://www.mysqltutorial.org/mysql-stored-procedure-tutorial.aspx), 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.

*drop function if exists fn1;*

*delimiter $$*

*create function fn1()* ***returns varchar(20)***

*begin*

*declare x1 varchar(20);*

*call pro1(x1);*

***return*** *(concat(x1, " Saleel"));*

*end $$*

*delimiter ;*

*drop function if exists fn1;*

*delimiter $$*

*create function fn1() returns varchar(20)*

*begin*

*declare x varchar(20);*

*declare y varchar(20);*

*select "Hello", "World" into x, y;*

*return (concat(x," ", y));*

*end $$*

*delimiter ;*

PL/SQL is a block structured language. The programs of PL/SQL are logical blocks that can contain any number of nested sub-blocks.

PL/SQL includes procedural language elements like conditions and loops. It allows declaration of constants and variables, procedures and functions, types and variable of those types and triggers. It can support Array and handle exceptions (runtime errors).

Introduction to MySQL cursor

To handle a result set inside a [stored procedure](http://www.mysqltutorial.org/mysql-stored-procedure-tutorial.aspx), you use a cursor. A cursor allows you to [iterate](http://www.mysqltutorial.org/stored-procedures-loop.aspx) and process a set of rows individually returned by a query.

1. Link Overview It

<https://www.javatpoint.com/pl-sql-interview-questions>