

MySQL: Programming

Tushar B. Kute,
<http://tusharkute.com>



- SQL is a dual-mode language. It is both an interactive database language used for ad hoc queries and updates, and a programmatic database language used by application programs for database access.
- For the most part, the SQL language is identical in both modes.

- The dual-mode nature of SQL has several advantages:
 - It is relatively easy for programmers to learn how to write programs that access the database.
 - Capabilities available through the interactive query language are also automatically available to application programs.
 - The SQL statements to be used in a program can be tried first using interactive SQL and then can be coded into the program.
 - Programs can work with tables of data and query results instead of navigating their way through the database

Programming SQL

- Embedded SQL
 - In this approach, SQL statements are embedded directly into the program's source code, intermixed with the other programming language statements.
 - Special embedded SQL statements are used to retrieve data into the program.
 - A special SQL precompiler accepts the combined source code and, along with other programming tools, converts it into an executable program.

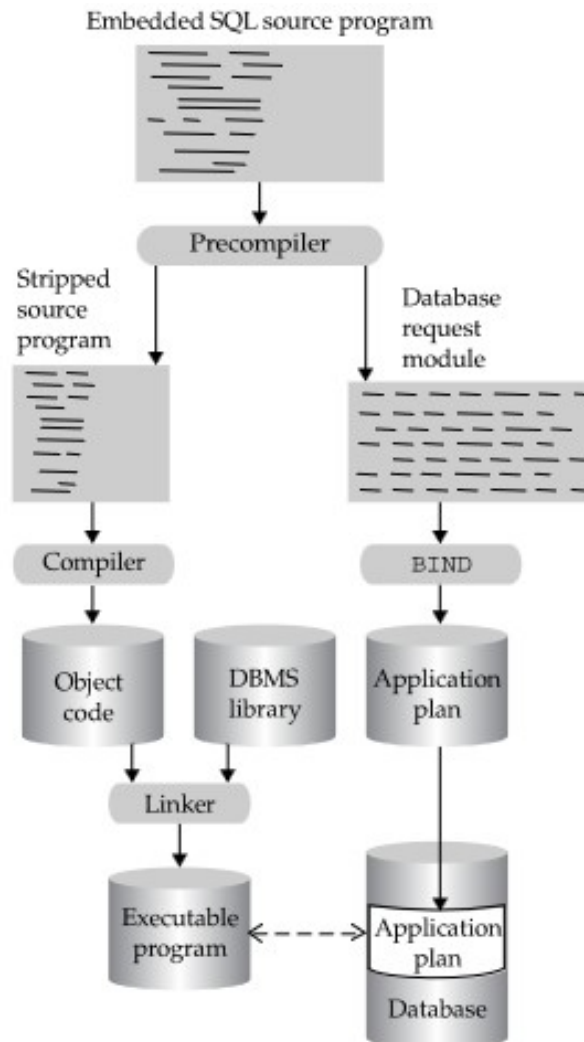
Programming SQL

- Application program interface
 - In this approach, the program communicates with the DBMS through a set of function calls called an application program interface (API).
 - The program passes SQL statements to the DBMS through the API calls and uses API calls to retrieve query results.
 - This approach does not require a special precompiler.

Programming SQL

DBMS	Callable API	Embedded SQL Language Support
DB2	ODBC, JDBC, JSQL	APL, Assembler, BASIC, COBOL, FORTRAN, Java, PL/I
Informix	ODBC, JDBC	C, COBOL, Java
Microsoft SQL Server	DB library (dblib), ODBC	C
MySQL	C-api (proprietary), ODBC, JDBC, Perl, PHP, Ruby, other scripting languages	None
Oracle	Oracle Call Interface (OCI), ODBC, JDBC, JSQL, PHP, Perl	C, COBOL, FORTRAN, Pascal, PL/I, Java
Sybase	DB library (dblib), ODBC, JDBC, SQLJ	C, COBOL, Java

Embedded SQL



MySQL Programs

- mysql
 - The command-line tool for interactively entering SQL statements or executing them from a file in batch mode.
- mysqladmin
 - A client that performs administrative operations, such as creating or dropping databases, reloading the grant tables, flushing tables to disk, and reopening log files. mysqladmin can also be used to retrieve version, process, and status information from the server.
- mysqlcheck
 - A table-maintenance client that checks, repairs, analyzes, and optimizes tables.

MySQL Programs

- `mysqldump`
 - A client that dumps a MySQL database into a file as SQL, text, or XML.
- `mysqlimport`
 - A client that imports text files into their respective tables using LOAD DATA.
- `mysqlpump`
 - A client that dumps a MySQL database into a file as SQL.

MySQL Programs

- `mysqlsh`
 - MySQL Shell is an advanced client and code editor for MySQL Server. See MySQL Shell 8.0. In addition to the provided SQL functionality, similar to `mysql`, MySQL Shell provides scripting capabilities for JavaScript and Python and includes APIs for working with MySQL. X DevAPI enables you to work with both relational and document data
- `mysqlshow`
 - A client that displays information about databases, tables, columns, and indexes.
- `mysqlslap`
 - A client that is designed to emulate client load for a MySQL server and report the timing of each stage. It works as if multiple clients are accessing the server.

Running a MySQL script

- `mysql -u root -p college < test.sql`

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.

Stored Procedure

- If we consider the enterprise application, we always need to perform specific tasks such as database cleanup, processing payroll, and many more on the database regularly. Such tasks involve multiple SQL statements for executing each task.
- This process might be easy if we group these tasks into a single task. We can fulfill this requirement in MySQL by creating a stored procedure in our database.
- A procedure is called a recursive stored procedure when it calls itself. Most database systems support recursive stored procedures.

Stored Procedure: 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 Procedure: Features

- 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:

- The following syntax is used for creating a stored procedure in MySQL.
- It can return one or more value through parameters or sometimes may not return at all.
- By default, a procedure is associated with our current database. But we can also create it into another database from the current database by specifying the name as `database_name.procedure_name`.

Syntax:

```
DELIMITER &&
```

```
CREATE PROCEDURE get_data()
```

```
BEGIN
```

```
    SELECT * FROM student WHERE marks > 70;
```

```
    SELECT COUNT(id) AS Total_Student FROM student;
```

```
END &&
```

```
DELIMITER ;
```

How to call procedure?

- We can use the CALL statement to call a stored procedure.
- This statement returns the values to its caller through its parameters (IN, OUT, or INOUT).
- The following syntax is used to call the stored procedure in MySQL:

```
CALL procedure_name ( parameter(s) )
```

IN Parameter

- A stored procedures and functions may have input, output, and input/output parameters.
- Input parameter is a parameter whose value is passed into a stored procedure/function module.
- The value of an IN parameter is a constant; it can't be changed or reassigned within the module.

IN Parameter

DELIMITER &&

CREATE PROCEDURE get_data(IN var1 INT)

BEGIN

SELECT * FROM student WHERE marks > 70 limit var1;

SELECT COUNT(id) AS Total_Student FROM student;

END &&

DELIMITER ;

OUT Parameter

- Output parameter is a parameter whose value is passed out of the stored procedure/function module, back to the calling PL/SQL block.
- An OUT parameter must be a variable, not a constant. It can be found only on the left-hand side of an assignment in the module.
- You cannot assign a default value to an OUT parameter outside of the module's body.
- In other words, an OUT parameter behaves like an uninitialized variable.

OUT Parameter

```
DELIMITER &&  
CREATE PROCEDURE get_data(OUT top INT)  
BEGIN  
    SELECT MAX(marks) INTO top FROM student;  
END &&  
DELIMITER ;
```

OUT Parameter

- `mysql> call get_data(@top);`
- `mysql> select @top;`

INOUT Parameter

- An input/output parameter is a parameter that functions as an IN or an OUT parameter or both.
- The value of the IN/OUT parameter is passed into the stored procedure/function and a new value can be assigned to the parameter and passed out of the module.
- An IN/OUT parameter must be a variable, not a constant. However, it can be found on both sides of an assignment.
- In other words, an IN/OUT parameter behaves like an initialized variable.

INOUT Parameter

DELIMITER &&

CREATE PROCEDURE get_data(INOUT var1 INT)

BEGIN

 SELECT marks INTO var1 FROM student where id = var1;

END &&

DELIMITER ;

INOUT Parameter

- `mysql> set @a = 3;`
- `mysql> call get_data (@a);`
- `mysql> select @a;`

Display the procedures

- When we have several procedures in the MySQL server, it is very important to list all procedures. It is because sometimes the procedure names are the same in many databases. We can list all procedure stored on the current MySQL server as follows:

```
SHOW PROCEDURE STATUS [LIKE 'pattern' |  
                        WHERE search_condition]
```

- This statement displays all stored procedure names, including their characteristics. If we want to display procedures in a particular database, we need to use the WHERE clause.

Display the procedures

- `SHOW PROCEDURE STATUS WHERE db = 'college';`

Drop procedure

- MySQL also allows a command to drop the procedure.
- When the procedure is dropped, it is removed from the database server also.
- The following statement is used to drop a stored procedure in MySQL:

```
DROP PROCEDURE [IF EXISTS ]  
procedure_name;
```

Benefits of procedures

- Reduce the Network Traffic:
 - Multiple SQL Statements are encapsulated in a stored procedure. When you execute it, instead of sending multiple queries, we are sending only the name and the parameters of the stored procedure
- Easy to maintain:
 - The stored procedure are reusable. We can implement the business logic within an SP, and it can be used by applications multiple times, or different modules of an application can use the same procedure.
 - This way, a stored procedure makes the database more consistent. If any change is required, you need to make a change in the stored procedure only.

Benefits of procedures

- Secure:
 - The stored procedures are more secure than the AdHoc queries.
 - The permission can be granted to the user to execute the stored procedure without giving permission to the tables used in the stored procedure.
 - The stored procedure helps to prevent the database from SQL Injection

Drawbacks of procedures

- If we use stored procedures, the memory uses of every connection that uses those stored procedures will increase substantially.
 - Also, if we overuse many logical applications inside stored procedures, the CPU usage will increase. It is because the database server is not well designed for logical operations.
- Stored procedure's constructs are not designed to develop complex and flexible business logic.

Drawbacks of procedures

- It is difficult to debug stored procedures.
 - Only a few database management systems allow us to debug stored procedures. Unfortunately, MySQL does not provide facilities for debugging stored procedures.
- It is not easy to develop and maintain stored procedures.
 - Developing and maintaining stored procedures are often required a specialized skill set that not all application developers possess.
 - It may lead to problems in both application development and maintenance phases.

Thank you

This presentation is created using LibreOffice Impress 7.4.1.2, can be used freely as per GNU General Public License



@mitu_skillologies



@mITuSkillologies



@mitu_group



@mitu-skillologies



@MITUSkillologies

kaggle

@mituskillologies

Web Resources

<https://mitu.co.in>

<http://tusharkute.com>



@mituskillologies

contact@mitu.co.in
tushar@tusharkute.com