MYSQL PROJECT

KEY AREAS

- Syntax and Creation
- Benefits and Drawbacks
- Parameter Handling
- Control Flow Structures within Procedures/Functions
- Best Practices

SYNTAX AND CREATION

SYNTAX

Refers to the rules governing the structure of code in a programming language. In terms of functions or procedures, syntax involves defining the function with its name, parameters, and body.

CREATION

Involves actually writing the code to implement the function or procedure according to the syntax rules of the programming language being used.

BENEFITS AND DRAWBACKS

BENEFITS

Benefits of functions procedures include code reuse, modularization, and abstraction, leading to cleaner ,more maintainable code.

DRAWBACKS

DRAWBACKS MAY INCLUDE POTENTIAL OVERHEAD (ESPECIALLY IN INTERPRETED LANGUAGES), DIFFICULTY IN DEBUGGING NESTED FUNCTION CALLS, AND POTENTIAL FOR CODE REDUNDANCY IF FUNCTIONS ARE POORLY DESIGNED

PARAMETER HANDLING

Parameter handling involves defining and managing the inputs (parameters) that a function or procedure accepts.

CONTROL FLOW STRUCTURES WITHIN PROCEDURES/FUNCTIONS

Control flow structures, such as conditionals (if-else statements) and loops (for, while), can be used within functions or procedures to control the execution flow based on certain conditions or iterate over data.

BEST PRACTICES

Keep functions small and focused: Functions should ideally perform a single, well-defined task to enhance reusability and maintainability.

Error handling: Implement robust error handling mechanisms to gracefully handle unexpected conditions and provide informative error messages.

1-EXAMPLES PROCEDURES/FUNCTIONS

PROCEDURES

```
mysql> DELIMITER // mysql>
CREATE PROCEDURE p1 () BEGIN
DECLARE fanta INT DEFAULT 55;
DROP TABLE t2; LOOP INSERT INTO
t3 VALUES (fanta); END LOOP; END//
Query OK, 0 rows affected (0.01 sec)
mysql> SHOW PROCEDURE CODE
p1// +-----+-------------------------+
| Pos | Instruction | +-----+----------
-----------------+ | 0 | set fanta@0 55
| | 1 | stmt 9 "DROP TABLE t2" | | 2
| stmt 5 "INSERT INTO t3 VALUES
(fanta)" | | 3 | jump 2 | +-----+--------
-----------------------+ 4 rows in set
(0.00 sec)
```

FUNCTIONS

```
mysql> CREATE FUNCTION
test.hello (s CHAR(20))
RETURNS CHAR(50)
DETERMINISTIC RETURN
CONCAT('Hello, ',s,'!'); Query OK,
0 rows affected (0.00 sec)
mysql> SHOW FUNCTION CODE
test.hello; +-----+--------------------
----------+ | Pos | Instruction | +--
---+-----------------------------+ | 0
| freturn 254 concat('Hello,
',s@0,'!') | +-----+--------------------
----------+ 1 row in set (0.00 sec)
```

1-SHOW PROCEDURE/FUNCTION STATUS STATEMENT

This statement is a MySQL extension that is available only for servers that have been built with debugging support. It displays a representation of the internal implementation of the named stored procedure. A similar statement, `SHOW FUNCTION CODE`, displays information about stored functions

To use either statement, you must be the user named as the routine `DEFINER`, have the `SHOW_ROUTINE` privilege, or have the `SELECT` privilege at the global level.

If the named routine is available, each statement produces a result set. Each row in the result set corresponds to one "instruction" in the routine. The first column is `Pos`, which is an ordinal number beginning with 0. The second column is `Instruction`, which contains an SQL statement (usually changed from the original source), or a directive which has meaning only to the stored-routine handler.

In this example, the nonexecutable `BEGIN` and `END` statements have disappeared, and for the `DECLARE` ***`variable_name`*** statement, only the executable part appears (the part where the default is assigned). For each statement that is taken from source, there is a code word `stmt` followed by a type (9 means `DROP`, 5 means `INSERT`, and so on). The final row contains an instruction `jump 2`, meaning `GOTO instruction #2`.

QUESTIONS

1 - What are the primary advantages of using functions or procedures in programming?

2 -Can you explain the concept of "test-driven development" and its relevance to writing functions or procedures?

ANSWERS

1 - They are tested procedures that get the job done without the need to (re-)create from scratch; They run faster than newly created procedures, the ones that still need to be compiled first;

2 - Test-driven development (TDD) benefits include better code quality, clear documentation, confident refactoring, faster debugging, modular development, improved design, stakeholder confidence, team collaboration, predictability, regression prevention, easier onboarding, and long-term cost savings.