Module 15: MySQL Functions, Triggers and Events

Stored Function:

Need for User Defined Functions:

- □ If we find repeatedly coding the same expression within a SQL statement, we may want to create a scalar-valued function for the expression.
- □ We can use that function in place of the expression, which can save coding time and make the code easier to maintain.

User Defined Functions:

- □ User defined function also called a stored function is an executable database object that contains a block of procedural SQL code.
- □ They are useful as they help us to extend the functionalities of MySQL server.
- → □ With MySQL, we can

- □ Function can accept zero or more input parameters and return a value such as string, integer or real using RETURN keyword.
- □ To call a stored function we can use it in any expression like built-in function.

Create and Drop Function:

- □ Create function Syntax:
  - □ **CREATE FUNCTION function_name([par1 data_type1][, par2 data_type2]…)**
  - □ **RETURNS datatype**
  - □ **[characteristic]**
  - □ **sql_block**

- □ To create a function we use CREATE FUNCTION followed by the name of the function.
- □ After the name of the function , we write set of parenthesis.
- □ Within the parenthesis we code the parameters for the function.
- □ After the parenthesis we write RETURNS keyword followed by the data type that's returned by the function.
- □ Sql code starts with BEGIN keyword and ends with END keyword.
- □ Before ending we write RETURN statement to return a value to the calling program.

- □ Drop function Syntax:
  - □ **DROP FUNCTION [IF EXISTS] function_name;**

Function Limitations:

- □ Statements that return a result set.
- □ Statements that perform explicit or implicit commit or rollback.
- □ Stored functions cannot be recursive.
- □ Stored function or trigger cannot modify a table that is already being used (for reading or writing) by the statement that invoked the function or trigger.
- □ Function cannot have dynamic SQL.

Stored Function – MCQs – 5

Q1) which of the following is used for constructing function?

Options:

A. PRODUCE FUNCTION function_name…
B. CONSTRUCT FUNCTION function_name …
C. CREATE FUNCTION function_name …
D. CREATE function_name

Solution:

Q2) how many values can be returned by function?

Options:

A. 1
B. 2
C. 0
D. 3

Solution:

Q3) How to invoke function named f1()?

Options:

A. call f1()
B. select f1()
C. f1()
D. call f1(10)

Solution:

Q4) what is the result of following code:

```
create function fa2() returns int deterministic
begin
declare c int;
select count(*) into c from stu_20 where rno> 2;
end$$
```

Options:

A. when invoked will return 2.
B. When invoked will return 3.
C. will not compile.
D. when invoked will return 1.

Solution:

Q5) what is the result of following code:

```
create function fa3() returns int deterministic
begin
declare c int;
select max(eyear) into c from stu_20;
return c;
end$$

call fa3() $$
```

Options:

A. Will not compile
B. Will display 4 as output
C. Will display 1 as output
D. Will display simmy as output.

Solution:

Stored Procedure vs. Stored Function

	Stored Procedure	Stored Function
Invoke	Invoked with CALL statement.	
Return	May return one or more values through parameters or may not return any at all.	
Parameters	Parameters can be input-only, output-only or for both input and output.	
Usage	Execute business logic	
Statements	Will allow DDL and DML statements.	
Calling	Stored procedures can call stored function.	
Allowed	• Return result set. • Perform transaction. • Perform exception handling.	

Triggers:

- → Trigger is a named block of code that

- Triggers are used to enforce business rules or verify data that can't be enforced by constraints.
- We can fire a trigger BEFORE or AFTER an INSERT, UPDATE or DELETE statement is executed on a table.
- We must specify a FOR EACH ROW clause to create a row level trigger that fires once for each row that's modified.
- We can use the OLD and NEW keywords to get and set the values for the columns that are stored in the old row and the new row.

- **Creating Trigger Syntax is:**
  - **CREATE TRIGGER trigger_name**
  - **{BEFORE | AFTER } {INSERT | UPDATE | DELETE } ON table_name**
  - **FOR EACH ROW**
  - **sql_block**

- To create a trigger we write, CREATE TRIGGER followed by trigger_name.
- After the trigger name we write BEFORE to indicate the trigger to fire before I/U/D operation.
- ON clause is to identify the name of the table.
- We write FOR EACH ROW to indicate the trigger is a row-level-trigger which fires for each row that's modified.
- Begin and End indicates the body of the trigger(not needed for single statement)
- Within body we use NEW keyword to work with the new values in a row that's being inserted or updated.
- Within body we use OLD keyword with the old values in a row that's being updated or deleted.

- **View Triggers:**
  - 1) List of all triggers in current database:
    - SHOW TRIGGERS
  - 2) List of all triggers in specified database:
    - SHOW TRIGGERS IN database_name

- **DROP Triggers:**
  - DROP TRIGGER [IF EXISTS] trigger_name

Trigger Limitations:

- □ **We cannot write select * from st50;**
  - □ ERROR 1415 (0A000): Not allowed to return a result set from a trigger
- □ **Triggers cannot begin or end transaction.**
  - □ ERROR 1422 (HY000): Explicit or implicit commit is not allowed in stored function or trigger.
- □ **Trigger cannot return a value.**
  - □ ERROR 1313 (42000): RETURN is only allowed in a FUNCTION
- □ **Trigger cannot be called explicitly.**
  - □ ERROR 1305 (42000): PROCEDURE test3aug.sai2 does not exist
- □ **Triggers are not permitted on views.**
  - □ ERROR 1347 (HY000): 'test3aug.stu2' is not BASE TABLE

Trigger MCQS:

Q1) Trigger is a _______ that executes in response to certain action.

Options:

A. Stored Function   B. Stored Package

C. Stored Procedure   D. Stored Event

Solution:

Q2) Trigger time can be _________.

Select all that apply

Options:

A. BEFORE   B. STOPPED

C. AFTER   D. DURING

Solution:

Q3) Trigger cannot be applied on following DML operations.

Select all that apply

Options:

A. INSERT   B. SELECT

C. UPDATE   D. DELETE

Solution:

Q4) Which keyword is used to specify the table_name on which trigger would be applied?

Options:

A. IN   B. FROM

C. UPON   D. ON

Solution:

Q5) In BEFORE Trigger new keyword can be used with ________

Options:

A. INSERT B. SELECT

C. UPDATE D. DELETE

Solution:

Q6) In AFTER Trigger old keyword can be used with ________

Options:

A. INSERT B. SELECT

C. UPDATE D. DELETE

Solution:

Events:

→ ❑ MySQL "event" or "scheduled event" is named block of code that

❑ MySQL event is also known as "temporal trigger" because it is triggered by time and not by table update/insert/delete like a trigger.

❑ An event can be onetime event that occurs once or a recurring event that occurs regularly at a specified interval.

❑ **Creating Event:**

- ❑ **CREATE EVENT event_name**
- ❑ **ON SCHEDULE {AT timestamp | EVERY interval } DO**
- ❑ **sql_block**

❑ We create event CREATE EVENT event_name, followed by unique event name within the database.

❑ In ON SCHEDULE we specify the schedule:

- ❑ 1) For one time event we write:
  - ❑ AT timestamp [+ INTERVAL]
- ❑ 2) For recurring event we write:
  - ❑ EVERY interval

❑ After DO keyword we place the SQL statements.

❑ **Show events:**

- ❑ SHOW EVENTS → events on the server
- ❑ SHOW EVENTS IN database_name → all events in db.

❑ **DROP Event:**

- ❑ DROP Event [IF EXISTS] event_name

□ **ALTER Event:**

- □ ALTER EVENT event_name {ENABLE | DISABLE}
- □ ALTER EVENT old_event_name RENAME TO new_event_name
- □ ALTER EVENT event_name ON SCHEDULE schedule
- □ ALTER EVENT event_name DO sql_body

Event MCQS:

Q1) Which of the following will execute a block of code at a particular time.

Options:

A. Stored Function B. Stored Package
C. Stored Procedure D. Stored Event

Solution:

Q2) Which keyword can be used to specify one time event.

Options:

A. AT B. ON
C. IS D. IN

Solution:

Q3) Which keyword can be used to specify recurring event.

Options:

A. AT B. REPEAT
C. EVERY D. IN

Solution:

Q4) What all can be altered with events. Choose all that apply

Options:

A. event name B. event timing
C. event body D. event size

Solution: