**Stored Routines:**

**Routine: A Usual, fixed action, or series of actions , repeated periodically.**

Diagram

Description automatically generated

Diagram

Description automatically generated

**Stored Routine:**

* An SQL Statement, or a set of SQL statements, that can be stored on the database server.
* Whenever a user needs to run the query in question, they can call, reference, or invoke the routine.

Text

Description automatically generated with medium confidence

Diagram

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text, application, email

Description automatically generated

A picture containing website

Description automatically generated

Diagram

Description automatically generated

Graphical user interface, diagram

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Graphical user interface, text

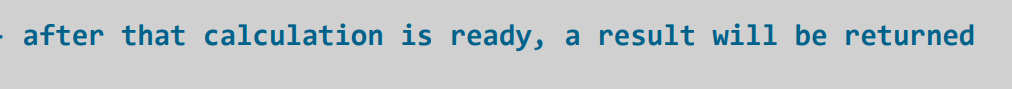
Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated



Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A picture containing text

Description automatically generated

**Stored procedures (SPs) are the most significant innovation in MySQL 5.0.**

These are custom SQL procedures or functions that are stored and executed directly by the MySQL server.

With SPs you have an SQL-based programming language at your service. SPs make it possible to store part of the logic of a client-server database application.

Stored procedures are a collection of SQL commands that are stored and executed in the MySQL

server.

Text, timeline

Description automatically generated

Timeline

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, website

Description automatically generated with medium confidence

**Error Code: 1418.**

Error Code: 1418. This function has none of DETERMINISTIC, NO SQL, or READS SQL DATA in its declaration and binary logging is enabled…

Let’s begin by saying that a log is a software component where you can save information about some events or errors that happened during the execution of a certain application. A log is preserved for traceability or debugging reasons and this is how it is used in MySQL as well.

Consequently, a Binary Log is a log that contains database changes. This type of logging affects the way in which we need to structure our code when creating MySQL functions.

When the Binary Log has been enabled, it will always check whether a function is changing the data in the database and what is the result to be produced. The situation can be described like this.

Unless we specify what the exact behavior of our function should be, our code will lead to an error. This error is with code 1418 and states that the function has none of the following characteristics in its declaration: DETERMINISTIC, NO SQL, or READS SQL DATA.

To solve this error, we must include one (or more) of these characteristics in our code in the way shown in the previous video. They must be placed right after we ‘ve specified the return type of the function. Here’s the syntax to use:

create function <function name> <function parameters> returns <type> <characteristics> …

Let’s check the meaning of these characteristics:

· DETERMINISTIC – it states that the function will always return identical result given the same input

· NO SQL – means that the code in our function does not contain SQL (rarely the case)

· READS SQL DATA – this is usually when a simple SELECT statement is present

When none of those is present in our code, MySQL assumes that our function is non deterministic and that it changes data. This might not be the case, but still, in the end, an error is raised just because MySQL cannot know a priori what our function will do. Adding one of those to our code will prevent this error of showing up.

That said, there is another way to stop the error - by disabling the binary log when creating functions. And we can achieve this by executing the following command:

SET @@global.log\_bin\_trust\_function\_creators := 1;

Technically speaking, this operation isn’t the safest one out there. Nevertheless, for the purposes of this course, it is the one that will solve the potential problems regardless of the version of MySQL.

In conclusion, remember that sometimes the Binary Log may be disabled anyway and you don’t have to take any of the above actions.

**UDF\_01**

delimiter $$

create function f\_emp\_avg\_sal3(p\_emp\_no integer) returns decimal(10,2)

reads sql data

begin

declare v\_avg\_salary decimal(10,2);

select avg(s.salary)into v\_avg\_salary from employees e join salaries s on e.emp\_no=s.emp\_no where e.emp\_no=p\_emp\_no;

return v\_avg\_salary;

end$$

delimiter ;

select f\_emp\_avg\_sal3(11300);

**UDF\_02**

delimiter $$

create function emp\_info(p\_f\_name varchar(255),p\_l\_name varchar(255)) returns decimal(10,2)

deterministic

begin

declare v\_max\_from\_date date;

declare v\_sal decimal(10,2);

select max(from\_date) into v\_max\_from\_date from employees e

join

salaries s on e.emp\_no = s.emp\_no where e.first\_name = p\_f\_name and e.last\_name = p\_l\_name;

select s.salary into v\_sal from employees e

join

salaries s on e.emp\_no=s.emp\_no where e.first\_name=p\_f\_name and e.last\_name =p\_l\_name and

s.from\_date = v\_max\_from\_date;

return v\_sal;

end$$

delimiter ;

select emp\_info('Aruna','Journel');

Calendar

Description automatically generated

A picture containing calendar

Description automatically generated

A picture containing timeline

Description automatically generated

Calendar

Description automatically generated

A picture containing text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**Locks:**

Exclusive lock mode prevents the associated resource from being shared. This lock mode is obtained to modify data. The first transaction to lock a resource exclusively is the only transaction that can alter the resource until the exclusive lock is released.

Share lock mode allows the associated resource to be shared, depending on the operations involved. Multiple users reading data can share the data, holding share locks to prevent concurrent access by a writer (who needs an exclusive lock). Several transactions can acquire share locks on the same resource.

An exclusive or write lock gives a process exclusive access for writing to the specified part of the file. While a write lock is in place, no other process can lock that part of the file.

A shared or read lock prohibits any other process from requesting a write lock on the specified part of the file. However, other processes can request read locks.