

Funciones Almacenadas. MySQL

Funciones Almacenadas.

Una función almacenada es un programa almacenado de tipo especial que devuelve un solo valor. Por lo general, utiliza funciones almacenadas para encapsular fórmulas comunes o reglas comerciales que son reutilizables entre sentencias SQL o programas almacenados.

A diferencia de un procedimiento almacenado, se puede usar una función almacenada en sentencias SQL donde se use una expresión. Esto ayuda a mejorar la legibilidad y la facilidad de mantenimiento del código de procedimiento.

Para crear una función almacenada, use la instrucción CREATE FUNCTION.

Sintaxis de MySQL CREATE FUNCTION.

A continuación se ilustra la sintaxis básica para crear una nueva función almacenada:

```
DELIMITER $$

CREATE FUNCTION function_name(
    param1,
    param2,...
)
RETURNS datatype
[NOT] DETERMINISTIC
BEGIN
    -- statements
END $$

DELIMITER ;
```

En esta sentencia:

- Primero, especifique el nombre de la función almacenada que desea crear después de las palabras clave CREATE FUNCTION.
- En segundo lugar, enumere todos los parámetros de la función almacenada dentro de los paréntesis seguidos del nombre de la función. Por defecto, todos los parámetros son los parámetros IN. No puede especificar modificadores IN, OUT o INOUT a los parámetros.
- En tercer lugar, especifique el tipo de datos del valor de retorno en la declaración RETURNS, que puede ser cualquier tipo de datos MySQL válido.

- Cuarto, especifique si una función es determinista o no, utilizando la palabra clave DETERMINISTIC. Una función determinista siempre devuelve el mismo resultado para los mismos parámetros de entrada, mientras que una función no determinista devuelve resultados diferentes para los mismos parámetros de entrada. Si no usa DETERMINISTIC o NOT DETERMINISTIC, MySQL usa la opción NOT DETERMINISTIC por defecto.
- Quinto, escriba el código en el cuerpo de la función almacenada en el bloque BEGIN END. Dentro de la sección del cuerpo, debe especificar al menos una instrucción RETURN. La instrucción RETURN devuelve un valor a los programas que llaman. Cada vez que se alcanza la instrucción RETURN, la ejecución de la función almacenada finaliza inmediatamente.

Ejemplo de MySQL CREATE FUNCTION

Tomemos el ejemplo de crear una función almacenada. Utilizaremos la tabla de customers en la base de datos de muestra para la demostración.

La siguiente instrucción CREATE FUNCTION crea una función que devuelve el nivel del cliente en función del crédito:

```
DELIMITER $$

CREATE FUNCTION CustomerLevel(
    credit DECIMAL (10,2)
)
RETURNS VARCHAR (20)
DETERMINISTIC
BEGIN
    DECLARE customerLevel VARCHAR (20);

    IF credit > 50000 THEN
        SET customerLevel = 'PLATINUM' ;
    ELSEIF (credit >= 50000 AND
        credit <= 10000) THEN
        SET customerLevel = 'GOLD' ;
    ELSEIF credit < 10000 THEN
        SET customerLevel = 'SILVER' ;
    END IF ;
    -- return the customer level
    RETURN (customerLevel);
END $$
DELIMITER ;
```

Una vez que se crea la función, puede verla en MySQL Workbench en la sección Funciones.

O puede ver todas las funciones almacenadas en la base de datos de los classicmodels actuales utilizando el SHOW FUNCTION STATUS siguiente manera:

```
SHOW FUNCTION STATUS
WHERE db = 'classicmodels' ;
```

Llamar a una función y asignar el resultado a una variable.

```
SET @resultado=CustomerLevel(200); -- 200 es un ejemplo, podría ser otra cantidad
```

Llamar a una función almacenada en una instrucción SQL.

La siguiente declaración utiliza la función almacenada CustomerLevel:

```
SELECT
    customerName,
    CustomerLevel(creditLimit)
FROM
    customers
ORDER BY
    customerName;
```

Llamar a una función almacenada en un procedimiento almacenado

La siguiente instrucción crea un nuevo procedimiento almacenado que llama a la función almacenada CustomerLevel() :

```
DELIMITER $$

CREATE PROCEDURE GetCustomerLevel(
    IN customerNo INT ,
    OUT customerLevel VARCHAR (20)
)
BEGIN

    DECLARE credit DEC (10,2) DEFAULT 0;

    -- get credit limit of a customer
    SELECT
        creditLimit
    INTO credit
    FROM customers
    WHERE
        customerNumber = customerNo;

    -- call the function
    SET customerLevel = CustomerLevel(credit);
END $$

DELIMITER ;
```

A continuación se ilustra cómo llamar al procedimiento almacenado GetCustomerLevel() :

```
CALL GetCustomerLevel(-131,@customerLevel);
SELECT @customerLevel;
```

Es importante tener en cuenta que si una función almacenada contiene sentencias SQL que consultan datos de tablas, entonces no debe usarla en otras sentencias SQL; de lo contrario, la función almacenada reducirá la velocidad de la consulta.

Introducción a la declaración MySQL DROP FUNCTION

La instrucción DROP FUNCTION descarta una función almacenada.

Aquí está la sintaxis de la instrucción DROP FUNCTION :

```
DROP FUNCTION [ IF EXISTS ] function_name;
```

La sintaxis, especifica el nombre de la función almacenada que desea eliminar después de las palabras clave DROP FUNCTION.

La opción IF EXISTS permite descartar condicionalmente una función almacenada si existe. Evita que surja un error si la función no existe.

Ejemplo de MySQL DROP FUNCTION

Utilizaremos la tabla de orders en la base de datos de muestra para la demostración.

- Primero, cree una nueva función llamada OrderLeadTime que calcule el número de días entre la fecha ordenada y la fecha requerida:

```
DELIMITER $$

CREATE FUNCTION OrderLeadTime (
    orderDate DATE ,
    requiredDate DATE
)
RETURNS INT
DETERMINISTIC
BEGIN
    RETURN requiredDate - orderDate;
END $$

DELIMITER ;
```

- En segundo lugar, use la instrucción DROP FUNCTION para descartar la función OrderLeadTime:

```
DROP FUNCTION OrderLeadTime;
```

- Tercero, use la función DROP FUNCTION para eliminar una función no existente:

```
DROP FUNCTION IF EXISTS NonExistingFunction;
```

MySQL emite una advertencia:

```
0 row(s) affected, 1 warning(s): 1305 FUNCTION
classicmodels.NonExistingFunction does not exist
```

Si desea ver la advertencia en detalle, use la instrucción SHOW WARNINGS :

```
SHOW WARNINGS ;
```

Listado de funciones almacenadas usando la instrucción SHOW FUNCTION STATUS

El SHOW FUNCTION STATUS es como el SHOW PROCEDURE STATUS pero para las funciones almacenadas.

Aquí está la sintaxis básica de la instrucción SHOW FUNCTION STATUS :

```
SHOW FUNCTION STATUS
[ LIKE 'pattern' | WHERE search_condition];
```

La instrucción SHOW FUNCTION STATUS devuelve todas las características de las funciones almacenadas. La siguiente declaración muestra todas las funciones almacenadas en el servidor MySQL actual:

```
SHOW FUNCTION STATUS ;
```

Tenga en cuenta que SHOW FUNCTION STATUS solo muestra la función a la que tiene un privilegio de acceso.

Si solo desea mostrar las funciones almacenadas en una base de datos particular, puede usar una cláusula WHERE en SHOW FUNCTION STATUS como se muestra en la siguiente declaración:

```
SHOW FUNCTION STATUS
WHERE search_condition;
```

Por ejemplo, esta declaración muestra todas las funciones almacenadas en los classicmodels base de datos de muestra :

```
SHOW FUNCTION STATUS
WHERE db = 'classicmodels' ;
```

Si desea encontrar las funciones almacenadas cuyos nombres contienen una palabra específica, puede usar la cláusula LIKE :

```
SHOW FUNCTION STATUS
LIKE '%pattern%' ;
```

La siguiente declaración muestra todas las funciones almacenadas cuyos nombres contienen la palabra Customer :

```
SHOW FUNCTION STATUS LIKE '%Customer%' ;
```

Listado de funciones almacenadas usando el diccionario de datos

El diccionario de datos MySQL tiene una tabla de routines que almacena información sobre las funciones almacenadas de todas las bases de datos en el servidor MySQL actual.

Esta consulta encuentra todas las funciones almacenadas en una base de datos particular:

```
SELECT
    routine_name
FROM
    information_schema.routines
WHERE
    routine_type = 'FUNCTION'
    AND routine_schema = '<database_name>' ;
```

Por ejemplo, la siguiente declaración devuelve todas las funciones almacenadas en la base de datos de classicmodels:

```
SELECT
    routine_name
FROM
    information_schema.routines
WHERE
    routine_type = 'FUNCTION'
    AND routine_schema = 'classicmodels' ;
```

Mostrar funciones almacenadas utilizando MySQL Workbench

Si usa MySQL Workbench, puede ver todas las funciones almacenadas desde una base de datos.

- Paso 1. Conéctese a la base de datos que desea mostrar las funciones almacenadas.
- Paso 2. Abra el menú Funciones , verá una lista de funciones que pertenecen a la base de datos.