Eventos

Índice

[1. Planificador de eventos 3](#_Toc72306674)

[2. Sentenza CREATE EVENT 3](#_Toc72306675)

[3. Exemplos de creación de eventos 4](#_Toc72306676)

[4. Sentenzas SHOW EVENTS e SHOW CREATE EVENT 5](#_Toc72306677)

[5. Sentenza ALTER EVENT 6](#_Toc72306678)

[6. Sentenza DROP EVENT 6](#_Toc72306679)

1. Planificador de eventos

MySQL, a partir da versión 5.1, inclúe unha característica que se chama 'Planificador de eventos', que permite executar sentenzas SQL ou procedementos almacenados tendo en conta un calendario establecido, indicando o momento, ou o intervalo de tempo no que se executarán. Proporcionan unha gran potencia ao servidor utilizado conxuntamente cos procedementos almacenados e os disparadores.

A configuración dun servidor MySQL inclúe a variable *event\_scheduler*. Dependendo do valor de dita variable, o planificador de eventos estará habilitado ou non. Para habilitalo, hai que modificar o valor que ten a variable global *event\_scheduler*, executando a sentenza:

**set** **global** event\_scheduler **=** **on;**

No caso de que a variable tome o valor *off,* pódense crear eventos e non se mostra ningún erro, pero non funciona o calendario, e polo tanto non se van a executar. Pódese consultar o valor que ten a variable executando a sentenza:

**show variables like** 'event\_scheduler'**;**



1. Sentenza CREATE EVENT

A sentenza CREATE EVENT permite crear e programar eventos. O evento creado queda asociado á base de datos que estea activa, ou ben á base de datos á que se fai referencia cando se utilizan nomes cualificados. O evento só se executará se o planificador de eventos está habilitado. Sintaxe:

CREATE [DEFINER = { usuario| CURRENT\_USER }]

EVENT [IF NOT EXISTS] nome\_evento

ON SCHEDULE calendario

[ON COMPLETION [NOT] PRESERVE]

[ENABLE | DISABLE | DISABLE ON SLAVE]

[COMMENT 'comentario']

DO corpo\_evento;

* A cláusula DEFINER permite indicar o nome do usuario que vai ser considerado o creador do evento. Se non se especifica nada tómase CURRENT\_USER que fai referencia ao usuario actual que está creando o evento.
* *nome\_evento*: é o nome que vai ter o evento. Non é sensible a maiúsculas e minúsculas polo que *meuEvento* e *meuevento* serían iguais. Ten que ser único dentro do esquema dunha base de datos.
* A cláusula ON SCHEDULE determina **cando** ou **con que frecuencia** e **durante canto tempo** se executan as sentenzas que forman o corpo do evento.
* *calendario*: O calendario da programación ten a seguinte sintaxe:

AT dato\_timestamp [+ INTERVAL intervalo] …

| EVERY intervalo

[STARTS dato\_timestamp [+ INTERVAL intervalo] …]

[ENDS dato\_timestamp [+ INTERVAL intervalo] …]

No calendario pódense utilizar as opcións:

* AT para indicar o momento en que se van a executar as sentenzas contidas no corpo do evento.
* EVERY para indicar que as sentenzas teñen que executarse cada certo período de tempo. No caso de utilizar a opción EVERY, de forma opcional, tamén se pode indicar o momento en que empezan a executarse (STARTS) e o momento en que se deixan de executar (ENDS).
* *intervalo*: A definición do intervalo ten a seguinte sintaxe:

cantidade {YEAR | QUARTER | MONTH | DAY | HOUR | MINUTE | WEEK | SECOND | YEAR\_MONTH | DAY\_HOUR | DAY\_MINUTE | DAY\_SECOND | HOUR\_MINUTE | HOUR\_SECOND | MINUTE\_SECOND}

* A opción [ON COMPLETION [NOT] PRESERVE] permite indicar ao servidor se hai que borrar o evento unha vez que xa se executou. Se non se especifica nada, o comportamento normal é que un evento se borre de forma automática cando se deixa de executar, ven sexa porque só se executa unha vez (AT), ou porque acabou o tempo en que se tiña que executar (ENDS).
* A opción [ENABLE | DISABLE | DISABLE ON SLAVE] permite habilitar ou deshabilitar o evento no momento da creación. Pódese modificar máis tarde o seu estado coa sentenza ALTER EVENT.
* A opción [COMMENT 'comentario'] permite engadir un comentario na descrición do evento.
* *corpo\_evento*: Pode ser unha sentenza, ou conxunto de sentenzas SQL en forma de bloque de programación empezando por BEGIN e rematando en END. No caso de estar formado por unha soa sentenza non sería necesario utilizar os delimitadores de bloque BEGIN e END. O corpo do evento pode incluír chamadas a rutinas almacenadas (procedementos e funcións definidas polo usuario).

1. Exemplos de creación de eventos

A creación dun evento require como mínimo:

* As palabras claves CREATE EVENT, e un nome para o evento (*'nome\_evento'*) que ten que ser único dentro do esquema dunha base de datos.
* A cláusula ON SCHEDULE para establecer o calendario de execución.
* A cláusula DO na que se escribe o conxunto de sentenzas que van a executar.

Exemplo 1: Na táboa *concerto* da base de datos *practicas1* está almacenado o prezo dos concertos programados. Decídese incrementar os prezos dos concertos un 15%. Este incremento entrará en vigor dentro de 24 horas. Para este exemplo, pódese crear un evento que se execute unha soa vez dentro de 24 horas, tomando como referencia a data do sistema e se elimine despois automaticamente:

-- activar base de datos practicas1

**use** practicas1**;**

-- creación do evento

**drop** **event** **if** **exists** actualizaPrezo**;**

**create** **event** actualizaPrezo

**on** **schedule** **at** now**()** **+** **interval** 24 **hour**

**do** **update** concerto **set** prezo **=** prezo **\*** 1.15**;**

Exemplo 2: Un caixeiro automático so permite retirar 1.000 € diarios por conta. O sistema irá sumando os reintegros feitos no día en cada conta, restándolle o seu importe ao límite dispoñible que está gardado na columna *limite\_dia* na táboa de *contas*. Ás 00:00 de cada día habería que actualizar o límite a retirar e establecer os 1.000 € para o día que empeza. Para este exemplo, é útil crear un evento que se execute tódolos días ás 00:00.

-- activar base de datos practicas1

**use** practicas1**;**

-- creación da táboa contas se non existe

**create** **table** **if** **not** **exists** contas**(**

idConta **char(**20**),**

dataApertura **date,**

saldo **decimal(**12**,**2**),**

limiteDia **decimal(**10**,**2**),**

podeRetirar **bit** **default** 0

**)** **engine** **=** myisam**;**

-- creación do evento

**drop** **event** **if** **exists** actualizaLimite**;**

**create** **event** actualizaLimite

**on** **schedule** **every** 1 **day** starts '2015-01-01 00:00:00'

**do** **update** contas **set** limiteDia **=** 1000**;**

Este evento é recorrente, e executarase ata que se borre, porque non se utilizou a cláusula ENDS no momento da creación do evento. No caso de utilizar a cláusula ENDS, o evento deixa de executarse na data e hora sinalados e bórrase automaticamente.

1. Sentenzas SHOW EVENTS e SHOW CREATE EVENT

A información dos eventos creados gárdase no dicionario de datos, igual que o resto de obxectos das bases de datos. No caso de MySQL, a información sobre os disparadores pódese consultar en *mysql.event* e en *information\_schema.events*. MySQL tamén dispón das sentenzas SHOW EVENTS e SHOW CREATE EVENTS.

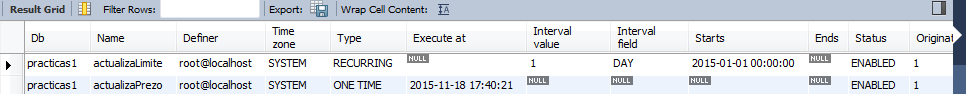
SHOW EVENTS permite ver os eventos creados. A sintaxe é:

SHOW EVENTS [{FROM | IN} *nome\_bd*]

[LIKE '*patrón*' | WHERE *expresión*]

No caso de non especificar o nome da base de datos (*nome\_bd*), móstranse os eventos asociados á base de datos activa. Pódense utilizar cláusulas LIKE ou WHERE para mostrar só os eventos que cumpran unha condición, e non todos. Exemplo:

**show** **events** **from** practicas1**;**

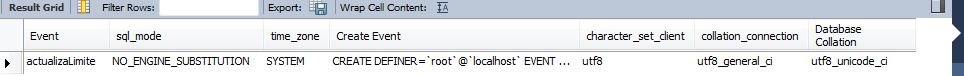


* A sentenza SHOW CREATE EVENT, permite ver información de como foi creado o evento, incluído o código SQL utilizado para a creación. Sintaxe:

SHOW CREATE EVENT nome\_evento

Exemplo:

**SHOW** **CREATE** **EVENT** practicas1**.**actualizaLimite**;**



1. Sentenza ALTER EVENT

Un evento só pode ser modificado polo usuario que o creou (DEFINER) ou por usuarios que teñan permisos sobre ese evento. O usuario que executa a orde ALTER pasará a ser considerado como o usuario que crea o evento (DEFINER).

A sentenza ALTER EVENT permite modificar un evento, sen necesidade de borralo e volvelo a crear. Sintaxe:

ALTER [DEFINER = { usuario| CURRENT\_USER }]

EVENT nome\_evento

[ON SCHEDULE calendario]

[ON COMPLETION [NOT] PRESERVE]

[RENAME TO novo\_nome\_evento]

[ENABLE | DISABLE | DISABLE ON SLAVE]

[COMMENT 'comentario']

[DO corpo\_evento]

A sintaxe das cláusulas DEFINER, ON SCHEDULE, ON COMPLETION, ENABLE/DISABLE, COMMENT e DO, é exactamente igual que na sentenza CREATE EVENT, pero só se pode executar para eventos que xa existan. A cláusula RENAME permite cambiar o nome a un evento.

Nunha sentenza ALTER EVENT pódense modificar unha ou máis cláusulas do evento, especificando só aquelas cláusulas nas que queremos facer cambios. As cláusulas que se omiten non se modificarán e permanecerán os valores que se lles deu no momento da creación. Exemplos:

* Para cambiar só o calendario do evento *actualizaLimite* do exemplo 1, hai que executar a seguinte sentenza:

**alter** **event** actualizaLimite

**on** **schedule every** 1 **day**

**starts** '2015-11-01 00:00:00'

**ends** '2016-12-31 00:00:00'**;**

Neste caso, faise un cambio que afecta ao calendario, logo só hai que incluír a cláusula ON SCHEDULE.

* É posible modificar o nome dun evento, ou movelo dunha base de datos a outra coa cláusula RENAME TO. Exemplo:

**alter** **event** practicas1.actualizaLimite

**rename to** utilidades.actualizaLimite**;**

Neste exemplo, o evento *actualizaLimite* pásese da base de datos *practicas1* á base de datos *utilidades*.

* Pódese habilitar e deshabilitar un evento coa cláusula ENABLE/DISABLE. Exemplo:

**alter** **event** utilidades.actualizaLimite

**disable**

**comment** 'Deshabilitado por Julia Mendez o 12/12/2015'**;**

Despois de executar a sentenza, o evento deixa de executarse pero permanece gardado no servidor e volverá a executarse cando se volva a habilitar. Ademais engade un comentario, empregando a cláusula COMMENT. Neste exemplo pódese ver que se poden modificar varias cláusulas na mesma sentenza ALTER EVENT.

1. Sentenza DROP EVENT

A sentenza DROP EVENT permite borrar un evento. Sintaxe:

DROP EVENT [IF EXISTS] nome\_evento

Cando se borra unha base de datos bórranse os eventos que estean asociados a ela.