



TEMA 6.

PROGRAMACIÓN EN BASES DE
DATOS.

SESIÓN 1.



INTRODUCCIÓN

- **RUTINAS** formadas por una serie de instrucciones que permiten realizar una tarea. Una vez almacenada la rutina, ésta podrá ser invocada o llamada a ejecución en cualquier momento.
- Las rutinas que se pueden desarrollar en MySQL y, en general, en cualquier SGBD relacional son:
 - Funciones.
 - Procedimientos.
 - Disparadores o Triggers.
 - Cursores.

INTRODUCCIÓN

■ VENTAJAS

- Se automatizan procesos que constan de varias instrucciones. No hay que reescribir esas instrucciones.
- Desde los clientes se tiene que enviar muchísima menos información al servidor. El servidor ya tiene las rutinas almacenadas.
- Si están perfectamente comprobadas las rutinas, hay mayor seguridad de que los procesos se realicen correctamente.

INTRODUCCIÓN

■ DESVENTAJAS

- Portabilidad. Hay bastantes diferencias en el lenguaje SQL para crear rutinas en los diferentes SGBD por lo que una base de datos con rutinas creadas en un SGBD puede no ser portable a otro SGBD por esas rutinas.
- Pueden producirse errores de ejecución de una rutina que sean difícilmente detectables.

VARIABLES DE USUARIO Y DE SISTEMA

- En MySQL podemos usar dos tipos de variables:
 - Variables de sistema:
 - Las crea el servidor cuando se inicia y/o cuando se inicia una sesión.
 - El valor que tengan estas variables configuran el comportamiento del servidor y de las sesiones.
 - Por ejemplo, la variable autocommit que hemos visto al estudiar las transacciones es una variable de sistema y de sesión.
 - Sólo los usuarios con los privilegios adecuados podrán modificar los valores de estas variables.
 - Variables de usuario:
 - Las declara o crea un usuario para usarlas y modificarlas dentro de la sesión.
 - Cuando se cierra una sesión, todas las variables de usuario que se hubieran creado en la sesión desaparecen.

VARIABLES DEL SISTEMA

- Las variables de sistema pueden ser:
 - Globales
 - Sesión
- El servidor crea y mantiene varias variables de sistema que indican cómo está configurado.
- Todas ellas tienen valores por defecto.
- Puede cambiarse el valor al arrancar el servidor usando opciones en la línea de comandos o en ficheros de configuración.
- En la mayoría de ellas (las dinámicas) podemos modificar su valor en tiempo de sesión usando el comando SET.

VARIABLES DEL SISTEMA

- Las **variables globales** establecen configuraciones globales del servidor:
 - Iniciado el servidor, se puede modificar el valor de las variables globales que sean dinámicas ejecutando el comando **SET GLOBAL variable=valor**.
- Las **variables de sesión** configuran las sesiones o para conexiones individuales de clientes:
 - El valor de las variables de sesión que son dinámicas se puede cambiar mediante un comando **SET SESSION variable=valor**.
- Muchas de ellas son tanto globales como de sesión (realmente tienen un valor global y tienen un valor para cada sesión).

VARIABLES DEL SISTEMA

- Se puede consultar las variables de sistema y sus valores usando el comando **SHOW VARIABLES**.
- Para modificar una variable global, se puede escribir SET GLOBAL.
- Para modificar una variable de sesión, se puede escribir SET, SET SESSION, SET LOCAL, SET @@SESSION, SET @@LOCAL.

```
SET SESSION autocommit=0;  
SET autocommit=0;  
SET GLOBAL max_connections=5;  
SET GLOBAL character_set_results=utf8;  
SET SESSION character_set_results=utf8;  
SET @@SESSION.character_set_results=utf8;  
SET character_set_results=utf8;  
SET LOCAL character_set_results=utf8;  
SET @@local.character_set_results=utf8;
```


VARIABLES DEL SISTEMA

- Si queremos ver los valores de las variables de sesión cuyo nombre comienza por auto_, ejecutaríamos:
 - SHOW SESSION VARIABLES LIKE 'auto_%';

Variable_name	Value
auto_increment_increment	1
auto_increment_offset	1

- El resultado nos da el valor de dos variables:
- Auto_increment_increment nos da cuanto se incrementa cada vez una variable de tipo autoincrementada.
- Auto_increment_offset nos dice en que valor comienza el valor de una variable autoincrementada.

VARIABLES DEL SISTEMA

- También se pueden obtener los valores de las variables de sesión usando SELECT.
- Para consultar su valor se debe escribir el nombre de la variable precedido de @@
 - `SELECT @@autocommit, @@max_connections, @@character_set_results;`

VARIABLES DEL SISTEMA

- autocommit: Vale 0 si está activado el estado transaccional.
- basedir: Ruta del directorio raíz de MySQL. No es dinámica.
- character_set_server: Conjunto de caracteres utilizado en el servidor.
- collation_server: Colación por defecto del servidor.
- datadir: Directorio donde se guardan las bases de datos. No es dinámica.
- init_file: Nombre del archivo de configuración del servidor, por defecto MY.INI. No dinámica.
- log: Vale true si se activa el registro de consultas. No dinámica

VARIABLES DEL SISTEMA

- log_updates: Vale true si se ha activado el registro de actualizaciones. No dinámica.
- max_connections: Máximo número de conexiones permitidas de forma simultánea.
- max_user_connections: Máximo número de sesiones que puede tener iniciadas un usuario.
- port: Número de puerto que usa MySQL para escuchar conexiones TCP/IP. No dinámica.
- skip_networking: Vale true si el servidor sólo admite conexiones locales. No dinámica.
- table_type: Tipo de tabla predeterminado al crear tablas sin especificar su tipo. No dinámica.

VARIABLES DE ESTADO

- Son variables del sistema que indican información sobre sus operaciones.
- Para ver estas variables se utiliza la sentencia **SHOW STATUS**, o con **SHOW STATUS LIKE '%';**
- Son variables que no se pueden modificar directamente.

VARIABLES DE ESTADO

- questions: El número de consultas que han sido enviadas al servidor.
- uptime: El número de segundos que el servidor ha estado funcionando.
- slow_queries: El número de consultas que han tardado más de long_query_time segundos
- max_used_connections: El número máximo de conexiones que han sido utilizadas simultáneamente desde que el servidor ha sido iniciado.
- innodb_rows_inserted: El número de registros insertados en tablas InnoDB.
- connections: El número de intentos de conexión (con éxito o no) al servidor MySQL.
- aborted_connects: El número de intentos de conexión al servidor MySQL que han fallado.
- com_select: Número de instrucciones select ejecutadas en la sesión.
- com_insert: Número de instrucciones insert ejecutadas en la sesión.

VARIABLES DE USUARIO

- Un usuario puede crear variables propias o de usuario.
- Las variables de usuario se declaran con @nombre_var, donde el nombre de variable nombre_var puede consistir de caracteres alfanuméricos y los caracteres '.', '_', y '\$'.
- Una forma de establecer una variable de usuario es empleando una instrucción **SET**:

```
SET @nombre_var = expresion;
```

VARIABLES DE USUARIO

- Otra forma de crear variables de usuario y/o asignarles valores es hacerlo asignándoles un valor devuelto por una consulta.
- Por ejemplo:

```
SELECT max(numcontrato) INTO @nummayor FROM contratos;  
SELECT fini INTO @fecha FROM contratos WHERE numcontrato=@nummayor;
```

COMPROBAMOS LO APRENDIDO

HOJA 1 DE EJERCICIOS.