



# PRÁCTICA 2

## TEMA 2

C.E.S ACADEMIA LOPE DE VEGA

CFGS: 2º Administración de Sistemas Informáticos en Red

Curso: 2017/2018

Asignatura: ADMINISTRACIÓN DE SISTEMAS GESTORES DE BASES DE DATOS

Prof. Miguel Castro Ortiz

Autor: Rafael Osuna Ventura

**1. ¿En el ámbito de las bases de datos, ¿qué entendemos por conector? Busca información de los distintos tipos de conectores que disponemos para mysql y relacionalos con su lenguaje de programación.**

Son estándares de acceso a las bases de datos desarrollados por los diversos manejadores de bases de datos, el objetivo de estos conectores es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar el sistema manejador de las bases de datos.

Tipos de conectores:

- ODBC (Open DataBase Connectivity).
- MDAC (Microsoft Data Access Components).
- OLE DB (Object Linking and Embedding for Databases).
- ADO (ActiveX Data Objects).
- DAO (Data Access Objects).
- RDS (Remote Data Services).

**2. ¿Cuál es la función del Pool de conexiones?**

Su función es mantener y gestionar un número de conexiones físicas, que se irán reutilizando automáticamente para aumentar la eficiencia.

**a. ¿Qué problema soluciona?**

Con ello se soluciona el problema respecto a las conexiones a bases de datos ya que estas son limitadas en número y además suponen un coste de procesador al abrirlas y cerrarlas.

**b. Explica su funcionamiento.**

Lo explicaremos paso a paso:

- Se usa la UCP para la fábrica de origen de datos JDBC (oracle.ucp.jdbc.PoolDataSourceFactory) para obtener una instancia del pool habilitada para el origen de datos utilizando el método de getPoolDataSource. La instancia de origen de datos debe ser del tipo de PoolDataSource.
- Se establecen las propiedades de conexión que se requieren para obtener una conexión física a una base de datos. Estas propiedades se establecen en la instancia de origen de datos, e incluyen: la dirección, el nombre de usuario y

contraseña para conectarse a la base de datos y la factoría de conexión utilizado para obtener la conexión física. Estas propiedades son específicas de un controlador JDBC y la base de datos.

- Establezca todas las propiedades del pool con el fin de anular el comportamiento predeterminado del grupo de conexión. Las propiedades del pool se establecen en la instancia de origen de datos.
- Obtener una conexión utilizando la instancia de origen de datos. La conexión devuelta es una consecuencia lógica de manejar a una conexión física en el pool de la conexión de la fuente de datos
- Utilice la conexión para realizar un trabajo sobre la base de datos
- Cerrar la conexión y regresar al pool

**c. El pool de conexiones proporciona un número prefijado y limitado de conexiones abiertas a la espera de ser utilizadas. ¿Qué problemas de eficiencia podría tener si establezco un número de conexiones muy superior a las que voy a utilizar? ¿Y si establezco un número de conexiones muy inferior?**

Si establezco un número de conexiones muy superior a las que voy a utilizar el problema estará en que agotaré los recursos de un sistema, afectando así al rendimiento de una aplicación y disponibilidad. El pool tiene una propiedad que asegura que esto no pase.

Si establezco un número de conexiones muy superior a las que voy a utilizar el problema estará en que se desperdician recursos del sistema en el mantenimiento de conexiones que son innecesarias. Si establezco un número de conexiones muy superior a las que voy a utilizar el problema reside

**d. Busca alguna estrategia para controlar el tamaño del pool de conexiones, que de alguna manera me permita ajustar el número de conexiones de una forma más óptima dependiendo de las conexiones que voy a utilizar a posteriori.**

Las principales características a configurar:

- Configuración de reutilización de la conexión Se configuran las características para poder reutilizar una conexión. Cuando supera un numero de reutilizaciones, se elimina del pool.
- Tiempo máximo de reutilización Permite controlar el tiempo necesario para que una conexión pueda volver a ser reutilizada. Mantiene como ocupadas, conexiones dentro del pool hasta que superan el tiempo estipulado

- Tiempo de espera para considerar una conexión abandonada El tiempo de espera que se permite a los abandonados para recuperar la conexión de nuevo en el grupo de conexión después de que una conexión no ha sido utilizada por un importe específico de tiempo.
- Ajuste del Tiempo de espera de conexión El tiempo de espera que una solicitud de conexión específica espera para obtener una conexión, si ya no hay ninguna conexión en el pool. La solicitud recibe una excepción si se alcanza el valor de tiempo de espera. Esta característica mejora la usabilidad de tiempo de espera global de la aplicación, minimizando la cantidad de tiempo que una aplicación está bloqueada y ofrece la capacidad de implementar una recuperación.
- Ajuste del tiempo de espera de conexión inactiva Especifica el tiempo que una conexión disponible puede permanecer inactiva antes de que se cierre y se retire del pool. Esta propiedad de tiempo de espera sólo es aplicable a las conexiones disponibles y no afecta a las conexiones que estén prestadas.

**3. Mediante el uso de un motor de almacenamiento podemos personalizar cómo se almacena la información de cada tabla. Busca información acerca de los diferentes motores de acceso personalizables, describiendo sus ventajas y para qué funciones son más óptimos.**

MySQL soporta varios motores de almacenamiento que tratan con distintos tipos de tabla. Los motores de almacenamiento de MySQL incluyen algunos que tratan con tablas transaccionales y otros que no lo hacen:

- MyISAM trata tablas no transaccionales. Proporciona almacenamiento y recuperación de datos rápida, así como posibilidad de búsquedas fulltext. MyISAM se soporta en todas las configuraciones MySQL, y es el motor de almacenamiento por defecto a no ser que tenga una configuración distinta a la que viene por defecto con MySQL.
- El motor de almacenamiento MEMORY proporciona tablas en memoria. El motor de almacenamiento MERGE permite una colección de tablas MyISAM idénticas ser tratadas como una simple tabla. Como MyISAM, los motores de almacenamiento MEMORY y MERGE tratan tablas no transaccionales y ambos se incluyen en MySQL por defecto. El motor de almacenamiento MEMORY anteriormente se conocía como HEAP.

- Los motores de almacenamiento InnoDB y BDB proporcionan tablas transaccionales. BDB se incluye en la distribución binaria MySQL-Max en aquellos sistemas operativos que la soportan. InnoDB también se incluye por defecto en todas las distribuciones binarias de MySQL 5.0 . En distribuciones fuente, puede activar o desactivar estos motores de almacenamiento configurando MySQL a su gusto.
- El motor de almacenamiento EXAMPLE es un motor de almacenamiento "tonto" que no hace nada. Puede crear tablas con este motor, pero no puede almacenar datos ni recuperarlos. El objetivo es que sirva como ejemplo en el código MySQL para ilustrar cómo escribir un motor de almacenamiento. Como tal, su interés primario es para desarrolladores.
- NDB Cluster es el motor de almacenamiento usado por MySQL Cluster para implementar tablas que se particionan en varias máquinas. Está disponible en distribuciones binarias MySQL-Max 5.0. Este motor de almacenamiento está disponible para Linux, Solaris, y Mac OS X . Añadiremos soporte para este motor de almacenamiento en otras plataformas, incluyendo Windows en próximas versiones.
- El motor de almacenamiento ARCHIVE se usa para guardar grandes cantidades de datos sin índices con una huella muy pequeña.

#### **4. Muestra las variables del sistema. ¿Qué información contienen estas variables?**

Mysql tiene muchas variables de sistema que indican cómo está configurado. Cada una tiene un valor por defecto y la mayoría puede ser modificada dinámicamente mientras el servidor está corriendo.

Existen de dos tipos:

- Globales: afectan la operación del servidor completo
- de Sesión: afectan la operación de las conexiones cliente individuales

Cuando el servidor inicia, establece todas las variables globales a su valor por defecto. Cuando se inicia una conexión desde un cliente, sus variables de sesión se inicializan usando los valores actuales de las correspondientes variables globales.

```
mysql> show variables;
```

Variable_name	Value
auto_increment_increment	1
auto_increment_offset	1
autocommit	ON
automatic_sp_privileges	ON
avoid_temporal_upgrade	OFF
back_log	80
basedir	/usr/
big_tables	OFF
bind_address	127.0.0.1
binlog_cache_size	32768
binlog_checksum	CRC32
binlog_direct_non_transactional_updates	OFF
binlog_error_action	ABORT_SERVER

**5. Muestra las variables de estado del sistema. ¿Qué información contienen estas variables? ¿En qué se diferencian con las variables del sistema?**

La información que contienen estas variables son las operaciones realizadas en el servidor. La diferencia es que si es de estado se refiere al estado del servidor en un momento dado y de sistema se refiere al comportamiento general.

```
mysql> show global status
-> ;
```

Variable_name	Value
Aborted_clients	0
Aborted_connects	1
Binlog_cache_disk_use	0
Binlog_cache_use	0
Binlog_stmt_cache_disk_use	0
Binlog_stmt_cache_use	0
Bytes_received	446
Bytes_sent	530
Com_admin_commands	0
Com_assign_to_keycache	0
Com_alter_db	0
Com_alter_db_upgrade	0
Com_alter_event	0
Com_alter_function	0
Com_alter_instance	0
Com_alter_procedure	0
Com_alter_server	0
Com_alter_table	0
Com_alter_tablespace	0
Com_alter_user	0

## 6. Muestra las siguientes variables:

### a. La ruta al directorio de instalación de MySQL.

```
mysql> show variables like "datadir";
+-----+-----+
| Variable_name | Value          |
+-----+-----+
| datadir       | /var/lib/mysql/ |
+-----+-----+
1 row in set (0,00 sec)
```

### b. La dirección IP asociada al servidor.

```
mysql> show variables where variable_name in ('hostname','port');
+-----+-----+
| Variable_name | Value          |
+-----+-----+
| hostname      | asgbd-VirtualBox |
| port          | 3306           |
+-----+-----+
2 rows in set (0,00 sec)
```

### c. La ruta al directorio de datos.

```
mysql> show variables like "datadir";
+-----+-----+
| Variable_name | Value          |
+-----+-----+
| datadir       | /var/lib/mysql/ |
+-----+-----+
1 row in set (0,00 sec)
```

### d. El motor de acceso a datos por defecto.

```
mysql> show engines;
+-----+-----+-----+-----+
| Engine          | Support | Comment |
+-----+-----+-----+-----+
| InnoDB          | DEFAULT | Supports transactions, row-level locking, and foreign keys |
| CSV             | YES     | CSV storage engine |
+-----+-----+-----+-----+
```

### e. La zona horaria del servidor.

```
mysql> describe mysql.time_zone_name;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Name           | char(64)      | NO   | PRI | NULL    |       |
| Time_zone_id   | int(10) unsigned | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0,00 sec)
```

7. Busca 5 variables tipo HAVE. Muestra su valor. A continuación, cambia su valor.

```
mysql> show variables like '%have%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_compress | YES   |
| have_crypt     | YES   |
| have_dynamic_loading | YES   |
| have_geometry  | YES   |
| have_openssl   | DISABLED |
| have_profiling | YES   |
| have_query_cache | YES   |
| have_rtree_keys | YES   |
| have_ssl       | DISABLED |
| have_statement_timeout | YES   |
| have_symlink   | YES   |
+-----+-----+
11 rows in set (0,01 sec)
```

8. Muestra las variables del sistema “auto\_increment\_increment” y “auto\_increment\_offset”.

```
mysql> show variables like 'auto_increment_increment';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| auto_increment_increment | 1 |
+-----+-----+
1 row in set (0,00 sec)

mysql> show variables like 'auto_increment_offset';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| auto_increment_offset | 1 |
+-----+-----+
1 row in set (0,01 sec)
```

a. Crea una base de datos con el nombre de LOPE DE VEGA.

b. Crea una tabla llamada 2ASIR con los campos (num\_matricula, nombre, apellido1, apellido2).

```
mysql> create table 2asir (num_matricula VARCHAR(20), nombre VARCHAR(20), apellido1 VARCHAR(20), apellido2 VARCHAR(20));
Query OK, 0 rows affected (0,02 sec)
```

c. Cambia las variables anteriormente mencionadas para que cada vez que inserte a un alumno en la tabla 2ASIR el núm\_matrícula salte de 20 en 20.



d. Inserta 5 estudiantes en la tabla creada anteriormente.

e. Muestra todos los datos que contiene la tabla y comprueba que está todo correcto.

```
mysql> select *from 2asir;
+-----+-----+-----+-----+
| num_matricula | nombre | apellidos1 | apellidos2 |
+-----+-----+-----+-----+
| 12345         | rafa   | osuna      | ventura     |
| 123235        | rafa   | cior       | ventura     |
| 12356855      | rafa   | cior       | vendsf      |
| 234           | antonio | cior       | asdsf       |
+-----+-----+-----+-----+
4 rows in set (0,00 sec)
```

### 9. ¿Qué son las variables tipo COM?

Son todas variables de estado que contabilizan el número de veces que cierto comando se ha ejecutado en la sesión actual. Su formato siempre es Com\_xxx siendo xxx el comando sql computado, por ejemplo, Com\_insert contabiliza el número de inserciones realizadas.

### 10. ¿Qué información proporcionan las variables tipo HADLER?

Variables relacionadas con operaciones de lectura y escritura sobre las tablas, su formato es Handler\_xxx siendo xxx la cadena que indica la operación realizada.

a. Muestra las variables tipo HANDLER.

```
mysql> show status like 'handler%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Handler_commit | 0     |
| Handler_delete | 0     |
| Handler_discover | 0     |
| Handler_external_lock | 0     |
| Handler_mrr_init | 0     |
| Handler_prepare | 0     |
| Handler_read_first | 0     |
| Handler_read_key | 0     |
| Handler_read_last | 0     |
| Handler_read_next | 0     |
| Handler_read_prev | 0     |
| Handler_read_rnd | 0     |
| Handler_read_rnd_next | 140   |
| Handler_rollback | 0     |
| Handler_savepoint | 0     |
| Handler_savepoint_rollback | 0     |
| Handler_update | 0     |
| Handler_write | 136   |
+-----+-----+
18 rows in set (0,05 sec)
```

**11. Muestra todas las tablas asociadas al diccionario de datos.**

```
mysql> select table_name from information_schema.tables;
+-----+
| table_name |
+-----+
| CHARACTER_SETS |
| COLLATIONS |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLUMNS |
| COLUMN_PRIVILEGES |
| ENGINES |
| EVENTS |
| FILES |
| GLOBAL_STATUS |
| GLOBAL_VARIABLES |
| KEY_COLUMN_USAGE |
| OPTIMIZER_TRACE |
| PARAMETERS |
| PARTITIONS |
| PLUGINS |
| PROCESSLIST |
| PROFILING |
| REFERENTIAL_CONSTRAINTS |
| ROUTINES |
| SCHEMATA |
| SCHEMA_PRIVILEGES |
| SESSION_STATUS |
| SESSION_VARIABLES |
+-----+
```

**12. Muestra los campos de la tabla “jugador” de la base de datos “liga”.**

```
mysql> show columns from liga.jugador;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_jugador | char(8) | NO | PRI | | |
| puntos_total | int(9) | YES | | NULL | |
| minutos | int(4) | YES | | NULL | |
| altura | int(5) | YES | | NULL | |
| equipo | char(8) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0,00 sec)
```

- a. A continuación, muestra a esa misma información, pero accediendo a la tabla correspondiente del diccionario de datos.

```
mysql> select column_name from information_schema.columns where table_name="jugador";
+-----+
| column_name |
+-----+
| id_jugador |
| puntos_total |
| minutos |
| altura |
| equipo |
+-----+
```

**13. Necesitamos conocer las restricciones que tiene la tabla clientes de la base de datos ebanca. Realiza la consulta a la tabla correspondiente del diccionario de datos.**

```
mysql> select * from table_constraints where table_schema="ebanca";
```

CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_SCHEMA	TABLE_NAME	CONSTRAINT_TYPE
def	ebanca	PRIMARY	ebanca	clientes	PRIMARY KEY
def	ebanca	idni	ebanca	clientes	UNIQUE
def	ebanca	PRIMARY	ebanca	cuentas	PRIMARY KEY
def	ebanca	PRIMARY	ebanca	movimientos	PRIMARY KEY
def	ebanca	fk1_dni	ebanca	movimientos	FOREIGN KEY
def	ebanca	fk2_ccuenta	ebanca	movimientos	FOREIGN KEY
def	ebanca	PRIMARY	ebanca	tipos	PRIMARY KEY
def	ebanca	fk1_tdni	ebanca	tipos	FOREIGN KEY
def	ebanca	fk2_tccuenta	ebanca	tipos	FOREIGN KEY

9 rows in set (0,00 sec)

**14. Comprueba las variables relacionadas con el registro general de consultas.**

**a. Activa el Log.**

```
mysql> set global general_log=1;
Query OK, 0 rows affected (0,03 sec)

mysql> show variables like 'general_log';
```

Variable_name	Value
general_log	ON

**b. Cambia la ruta para que el log lo almacene en el escritorio.**

**c. Realiza varias consultas en la base de datos.**

**d. Muestra el Log y relaciona la información con las consultas creadas anteriormente**

## 15. Comprueba las variables relacionadas con el registro binario.

a. Activa dentro de MySQL el Log. ¿Qué problema tienes? Actívalo de la forma correcta

```
mysql> show variables like '%bin';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| log_bin       | ON    |
| sql_log_bin   | ON    |
+-----+-----+
2 rows in set (0,00 sec)
```

b. Una vez que lo tengas activado, realiza operaciones que modifiquen datos en la base de datos “LOPE DE VEGA”.

c. Muestra el fichero Log relacionando su información con las operaciones realizadas en el apartado anterior.

```
asgbd@asgbd-VirtualBox:/var/log/mysql$ sudo mysqlbinlog mysql-bin.000005
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
/*!50003 SET @OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/;
DELIMITER /*!*/;
# at 4
#171031 10:38:23 server id 1  end_log_pos 123 CRC32 0x59cad534  Start: binlog
v 4, server v 5.7.19-0ubuntu0.16.04.1-log created 171031 10:38:23 at startup
# Warning: this binlog is either in use or was not closed properly.
ROLLBACK/*!*/;
BINLOG '
j0T4WQ8BAAAAdwAAAHsAAAAABAAQANS43LjE5LTB1YnVudHUwLjE2LjA0LjEtbG9nAAAAAAAAAAAA
AAAAAAAAAAAAAAAAACPRPhZEzgNAAGAEgAEBAQEgAAXwAEGggAAAAICAgCAAAACgoKKioAEjQA
ATTvYlk=
'/*!*/;
# at 123
#171031 10:38:23 server id 1  end_log_pos 154 CRC32 0xc8f84676  Previous-GTIDs
# [empty]
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog */ /*!*/;
DELIMITER ;
# End of log file
/*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;
```

d. Convierte el último fichero binario en fichero de texto.

```
asgbd@asgbd-VirtualBox:/var$ cd ..
asgbd@asgbd-VirtualBox:/$ nano mysql-bin.000005 > /home/asgbd/Escritorio/log.t
xt
```

## 16. Comprueba las variables relacionadas con el registro de consultas lentas.

### a. Activa el Log.

```
mysql> set global slow_query_log=1;  
Query OK, 0 rows affected (0,00 sec)
```

```
mysql> show variables like 'slow%';
```

Variable_name	Value
slow_launch_time	2
slow_query_log	ON
slow_query_log_file	/var/lib/mysql/asgbd-VirtualBox-slow.log

```
3 rows in set (0,00 sec)
```

### b. Modifica las opciones para que considere como lentas las consultas que tardan más de 20 segundos.

```
mysql> show variables like 'long_query_time';
```

Variable_name	Value
long_query_time	20.000000

```
1 row in set (0,00 sec)
```

### c. Modifica el Log para que lo muestre en forma de tabla.

### d. Realiza una consulta que tarde 25 segundos.

```
mysql> set global log_output="TABLE";  
Query OK, 0 rows affected (0,00 sec)
```

```
mysql> select sleep(25);
```

sleep(25)
0

```
1 row in set (25,00 sec)
```

**e. Muestra la tabla y señala el registro que muestre que hemos realizado una consulta lenta**

```
mysql> select * from slow_log;
+-----+-----+-----+-----+-----+-----+-----+-----+
| start_time | user_host | query_time | lock | | |
| time | rows_sent | rows_examined | db | last_insert_id | insert_id |
| server_id | sql_text | thread_id |
+-----+-----+-----+-----+-----+-----+-----+
| 2017-10-30 20:07:31.079517 | root[root] @ localhost [] | 00:00:25.000769 | 00:00:00.000000 | 1 | 0 | lopedevaga | 0 |
| 1 | select sleep(25) | 4 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0,00 sec)

mysql>
```