

## **Motores de almacenamiento en MySQL 5.0** **por Mario López y Juan A. Sánchez**

*15 de enero de 2008*

### **¿Qué es un motor de almacenamiento en un sistema gestor de bases de datos?**

Un motor de almacenamiento es una parte esencial de un SGDB puesto que se encarga de crear, recuperar, actualizar y borrar los datos de una base de datos<sup>1</sup>.

### **¿Por qué son importantes los motores de almacenamiento?**

Los datos en MySQL pueden ser almacenados de diversas formas, con distintas técnicas que nos aportan distintas funcionalidades. Aquí es donde encontramos la importancia de los motores de almacenamiento, ya que, dependiendo del motor que elijamos obtendremos distintas ventajas y dependerá de nosotros saber sacarles partido de forma acorde a nuestras necesidades.

MySQL nos provee por defecto de una serie de motores que podemos usar fácilmente (a parte de otros desarrollados por terceras partes que también podemos instalar), de hecho esta flexibilidad es una de las bazas que han jugado a favor de la popularidad de MySQL.

### **¿Cómo saber que motores están disponibles?**

Es muy sencillo, basta con escribir en una consola MySQL la siguiente consulta:

```
show engines;
```

Y obtendremos algo como esto:

---

<sup>1</sup> [http://en.wikipedia.org/wiki/Storage\\_engine](http://en.wikipedia.org/wiki/Storage_engine)

Engine	Support	Comment
MyISAM	DEFAULT	Default engine as of MySQL 3.23 with great performance
MEMORY	YES	Hash based, stored in memory, useful for temporary tables
InnoDB	YES	Supports transactions, row-level locking, and foreign keys
BerkeleyDB	NO	Supports transactions and page-level locking
BLACKHOLE	YES	/dev/null storage engine (anything you write to it disappears)
EXAMPLE	NO	Example storage engine
ARCHIVE	YES	Archive storage engine
CSV	YES	CSV storage engine
ndbcluster	DISABLED	Clustered, fault-tolerant, memory-based tables
FEDERATED	YES	Federated MySQL storage engine
MRG_MYISAM	YES	Collection of identical MyISAM tables
ISAM	NO	Obsolete storage engine

## Motores y diferencias

Las comparativas entre motores se suelen hacer basándose en cuatro funcionalidades clave:

**Tipos de datos:** aunque la mayoría son comunes hay algunos específicos que pueden ser decisivos bajo determinadas circunstancias.

**Bloqueo de datos:** la forma en la que el motor protege un dato que está siendo modificado para evitar problemas de acceso concurrente a los datos y mantener la integridad referencial.

**Indexado:** las diferentes técnicas de indexado pueden influir drásticamente en el rendimiento de una base de datos.

**Transacciones:** dota de fiabilidad a los datos mientras se realizan operaciones, te permite utilizar los datos pero sólo te permite guardarlos cuando se comprueba que las otras condiciones que pudiesen requerirse se han cumplido.

Y tenemos que hablar también de...

### ACID<sup>2</sup>

Se denomina ACID a la propiedad de una base de datos para realizar transacciones seguras. Así pues *ACID compliant* define a un sistema de gestión de bases de datos que puede realizar transacciones seguras.

En concreto **ACID** es un acrónimo de **A**tomicity, **C**onsistency, **I**solation and **D**urability: Atomicidad, Consistencia, Aislamiento y Durabilidad en español.

<sup>2</sup> <http://es.wikipedia.org/wiki/ACID>

- **Atomicidad:** es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias. Si una operación consiste en una serie de pasos, u ocurren todos o ninguno
- **Consistencia:** es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.
- **Aislamiento:** es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que la realización de dos transacciones sobre la misma información nunca generará ningún tipo de error.
- **Durabilidad:** es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

## Motores de almacenamiento

### MyISAM

Se basa en el antiguo ISAM, al que añade muchas mejoras, es el motor que usa MySQL por defecto. Es una buena combinación entre funcionalidad y rendimiento aunque carece de algunas características interesantes.

Características más importantes:

- Límite de  $2^{32}$  registros
- Máximo de 64 índices por tabla
- Máximo de 16 columnas por índice
- Los datos son independientes de la máquina y el sistema operativo
- Permite campos índice como NULL
- BLOB<sup>3</sup> y TEXT pueden ser índices
- Permite un gran tamaño en las tablas (hasta 256TB)
- No soporta transacciones
- Bloquea los datos a nivel de tabla
- No permite "claves ajenas"

Este motor pone especial empeño en la rapidez de las operaciones de lectura (predominio de SELECT), es una de las razones por las que MySQL es tan popular en la web, ya que la mayoría de las operaciones que se realizan son de este tipo. Que no tenga que hacer comprobaciones de integridad referencial también influye en su velocidad.

Ejemplo de creación de una tabla:

---

<sup>3</sup> binary large object (generalmente son imágenes, archivos de sonido y otros objetos multimedia)

```
CREATE TABLE pruebaMyISAM (  
    codigo varchar(5) default NOT NULL,  
    descripcion varchar(255) default NULL,  
    PRIMARY KEY (codigo)  
) ENGINE=MyISAM;
```

En general no hará falta indicar el uso de este motor pues es el que se usa por defecto.

## MERGE

Permite combinar varias tablas de igual estructura en una única tabla, pudiendo así realizar consultas sobre una tabla que nos devuelve datos de varias.

Características más importantes:

- Límite de  $2^{32}$  registros
- Las tablas "base" deben ser MyISAM
- Bloqueo a nivel de tabla
- No tiene índices, usa los de las tablas "base" (salvo FULLTEXT)
- La lectura es más lenta al tener que ir consultando la clave en cada una de las tablas subyacentes
- No permite REPLACE
- No soporta transacciones
- En su creación no comprueba que las tablas que usa existan y tengan una estructura idéntica

Una de sus funcionalidades puede ser partir una tabla muy grande en otras más pequeñas y, al unir las con MERGE, permitirnos trabajar con ellas como si fuesen una sola.

Ejemplo de creación de una tabla a partir de otras dos:

```
mysql> CREATE TABLE t1 (  
->     a INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
->     message CHAR(20));  
mysql> CREATE TABLE t2 (  
->     a INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
->     message CHAR(20));  
mysql> INSERT INTO t1 (message) VALUES ('Testing'),('table'),('t1');  
mysql> INSERT INTO t2 (message) VALUES ('Testing'),('table'),('t2');  
mysql> CREATE TABLE total (  
->     a INT NOT NULL AUTO_INCREMENT,  
->     message CHAR(20), INDEX(a)  
->     TYPE=MERGE UNION=(t1,t2) INSERT_METHOD=LAST;
```

## MEMORY (HEAP)

Guarda todos los datos en memoria, de forma que si se cae el servidor o reiniciamos MySQL se pierden los datos, aunque la estructura de las tablas se guarda.

Características más importantes:

- Bloquea los datos a nivel de tabla
- Puede usar índices HASH
- No soporta BLOB ni TEXT
- No soporta transacciones
- Resulta extremadamente fácil perder los datos

Resultan útiles como tablas temporales para determinadas consultas ya que al estar en memoria y poder tener Hash como índice resultan extremadamente rápidas, es una buena elección cuando necesitamos realizar operaciones muy rápidas sobre conjuntos pequeños de datos.

## FEDERATED

Este motor se incluyó en la versión 5.03. La novedad de este motor es que permite el acceso a una base de datos MySQL remota como si fuese local, en realidad tenemos una tabla local que representa a otra remota, ambas deben ser idénticas.

Características más importantes:

- Permite acceso a BBDD remotas
- MySQL no instala este motor por defecto
- No soporta transacciones
- No contempla el bloqueo de datos
- No permite ALTER

Ejemplo de tabla en la que se le indica la dirección de los datos:

```
CREATE TABLE federated_table (  
    id      int(20) NOT NULL auto_increment,  
    name    varchar(32) NOT NULL default '',  
    other   int(20) NOT NULL default '0',  
    PRIMARY KEY (id),  
    KEY name (name),  
    KEY other_key (other)  
)  
ENGINE=FEDERATED  
DEFAULT CHARSET=latin1  
COMMENT='mysql://root@remote_host:9306/federated/test_table';
```

## **ARCHIVE**

Se utiliza básicamente para almacenar grandes cantidades de datos sin índices en muy poco espacio, ya que los comprime con zlib alcanzando un nivel de ahorro de espacio considerable.

Características más importantes:

- Gran compresión de los datos
- Sólo permite INSERTS y SELECTS
- Bloquea los datos a nivel de registro
- Almacena los datos en un buffer hasta que los comprime e inserta
- No soporta transacciones

Este motor resulta especialmente útil para el almacenamiento de históricos o logs ya que suelen ocupar gran cantidad de espacio y no es necesario modificarlos con posterioridad.

## **CSV**

Almacena la información utilizando el formato de valores separados por comas (comma-separated values), de forma que cada tabla es un fichero que contiene los datos. No soporta indexado y su fin principal es permitir exportar los datos de forma que puedan ser importados fácilmente por algunas suites ofimáticas.

Características más importantes:

- Útil para exportar e importar datos
- No soporta indexación ni transacciones

## **BLACKHOLE**

El sorprendente uso de este motor es no almacenar los datos sino crear un log con la consulta SQL utilizada. Como no almacena ningún dato lógicamente no soporta índices, ni transacciones...

Su principal utilidad es mantener un servidor esclavo que mantenga un log del sistema principal.

## **NDB**

Es el motor de almacenamiento de los clúster de MySQL, las bases de datos se reparten por los diferentes nodos de un clúster.

### Características más importantes:

- Proporciona alta disponibilidad mediante redundancia.
- Proporciona alto rendimiento mediante fragmentación de datos sobre los grupos de nodos.
- Proporciona alta escalabilidad mediante la combinación de las dos características anteriores.
- Los datos se guardan en memoria, pero los logs van a disco.

Es una buena elección cuando disponiendo de varios servidores necesitamos a la vez velocidad, transacciones y redundancia de datos; replicación síncrona; y resistencia a caídas de servidores.

### **BerkeleyDB (o BDB)**

Este motor, independiente de MySQL, provee altas prestaciones. Posee un mecanismo de almacenamiento basado en hash de alta eficiencia, lo que facilita el rápido acceso a los datos de forma directa a costa de la lentitud en el acceso secuencial.

### Características más importantes:

- MySQL no lo instala por defecto
- Máximo de 31 índices por tabla
- Máximo de 16 columnas por índice
- Hasta 256TB
- Sí soporta transacciones
- Usa índices HASH
- MySQL necesita una clave primaria por cada tabla BDB, en caso de no existir creará una oculta
- El bloqueo interno de las tablas se hace a nivel de página (8192 bytes)
- Abrir muchas tablas es bastante lento
- Cada tabla se almacena en la ruta de creación definida y no se puede cambiar de directorio salvo usando mysqldump

### **InnoDB**

Está considerado como uno de los motores más avanzados para el almacenamiento de datos en MySQL. Provee un motor sólido con soporte completo de transacciones (es **ACID compliant**), permite el bloqueo de datos a nivel de registro permitiendo gran flexibilidad a la hora de utilizar las tablas, controla la integridad referencial, permite claves ajenas y tiene un sistema de recuperación de caídas.

No obstante la piedra de toque de InnoDB es su mecanismo de indexación y cache de los registros pues mantiene una caché de índices y datos en memoria y en disco proporcionando un muy alto rendimiento.

### Características más importantes:

- ACID compliant
- Permite claves ajenas y transacciones, soporte de integridad referencial
- Bloqueo de datos a nivel de registro y no bloquea la lectura durante los selects (mejora la concurrencia)
- Sistema de recuperación de caídas
- Cambiar la ubicación de la base de datos/tabla es complicado
- Una tabla no puede tener más de 1000 columnas
- El tamaño de sus logs debe ser inferior a 4GB
- El tamaño máximo para una tabla es de 64TB
- No permite índices de FULLTEXT
- No mantiene un contador interno de registros (select count(\*) from tabla lento al tener recorrer todo el índice)

### Ejemplo de uso de transacciones:

```
mysql> CREATE TABLE CUSTOMER (A INT, B CHAR (20), INDEX (A))
-> ENGINE=InnoDB;
Query OK, 0 rows affected (0.00 sec)
mysql> BEGIN;
Query OK, 0 rows affected (0.00 sec)
mysql> INSERT INTO CUSTOMER VALUES (10, 'Heikki');
Query OK, 1 row affected (0.00 sec)
mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)
mysql> SET AUTOCOMMIT=0;
Query OK, 0 rows affected (0.00 sec)
mysql> INSERT INTO CUSTOMER VALUES (15, 'John');
Query OK, 1 row affected (0.00 sec)
mysql> ROLLBACK;
Query OK, 0 rows affected (0.00 sec)
mysql> SELECT * FROM CUSTOMER;
+-----+-----+
| A     | B       |
+-----+-----+
|    10 | Heikki  |
+-----+-----+
1 row in set (0.00 sec)
```

## FALCON

Es el futuro motor de almacenamiento para MySQL 6.0, actualmente se encuentra en estado alpha, y está diseñada para entornos de servidores Web de alto volumen.

### Características más importantes:

- Cumple ACID
- Permite recuperación de estados / datos en caso de caídas
- Permite una alta concurrencia (bloqueo a nivel de registro)
- Caché de datos muy rápida y potente
- Incluye tablas para la monitorización de rendimiento y errores
- Permite una configuración simple

## Referencias

<http://www.linuxplanet.com/linuxplanet/tutorials/6034/1/>  
<http://bicosyes.com/motores-de-almacenamiento-de-mysql/>  
<http://jayant7k.blogspot.com/2006/06/mysql-storage-engines.html>  
<http://articles.techrepublic.com.com/5100-22-1058872.html>  
<http://en.wikipedia.org/wiki/InnoDB>  
<http://en.wikipedia.org/wiki/MyISAM>  
<http://dev.mysql.com/doc/refman/5.0/es/memory-storage-engine.html>  
<http://dev.mysql.com/doc/refman/5.0/es/myisam-storage-engine.html>  
[http://en.wikipedia.org/wiki/MySQL\\_Federated](http://en.wikipedia.org/wiki/MySQL_Federated)  
<http://jmonne.blogspot.com/2007/07/innodb.html>  
<http://lampologia.blogspot.com/2007/09/instalacin-de-mysql-60-con-falcon.html>  
<http://dev.mysql.com/doc/refman/5.0/es/storage-engines.html>  
<http://www.mysql-hispano.org/page.php?id=45&pag=6>  
<http://www.mysql.com/mysql60/top-reasons.html>