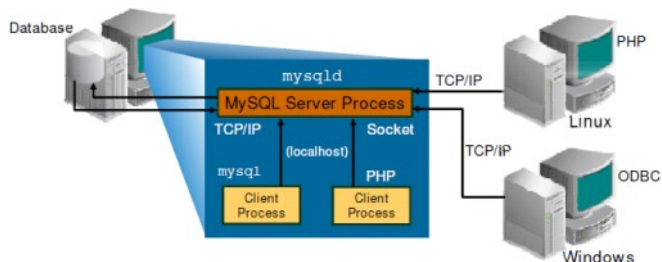


# MySQLAdmin - Conclusiones/Resumen

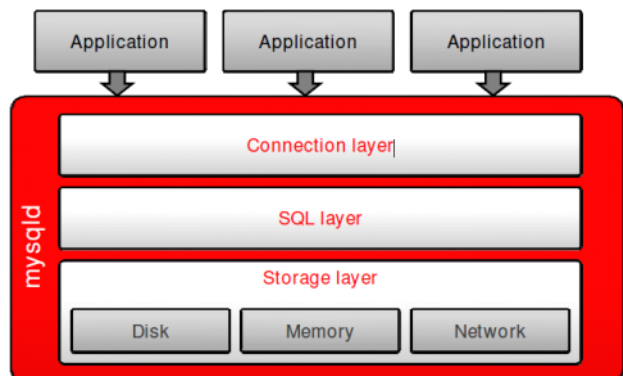
## MySQLAdmin - Arquitectura

### Arquitectura cliente - servidor

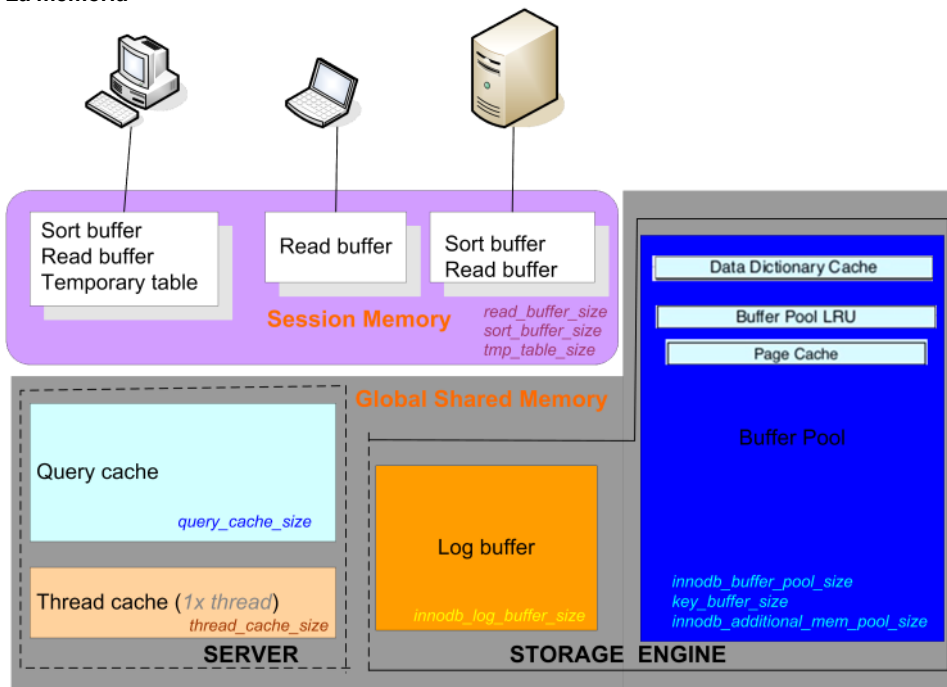
MySQL client/server model



3 capas



### La memoria

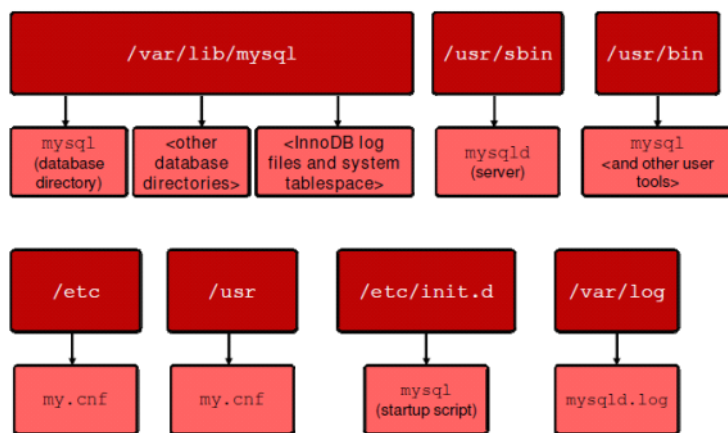


## MySQLAdmin - System Administration

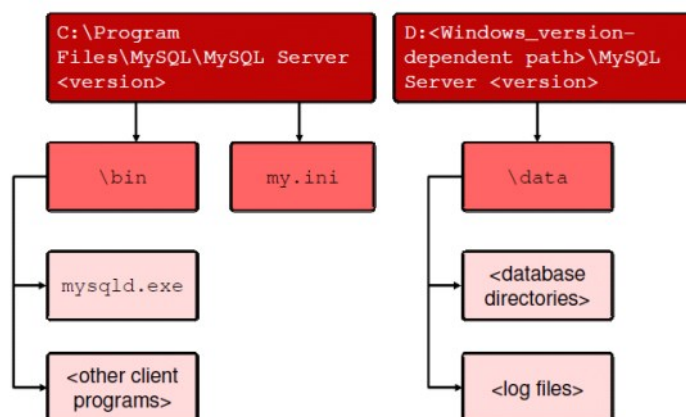
La instalacion:

2. Install all the downloaded RPM files.
  - a. Execute the `rpm -i <rpm_filename>` command for each RPM file.
  - b. The installation performs the following tasks automatically as it runs:
    - Extracts RPM files to their default locations
    - Registers a startup script named `mysql` in the `/etc/init.d` directory
    - Executes `mysql_install_db`, a script that creates the system databases and default `my.cnf` file, sets up a random password for the `root` accounts, and saves that password in the installing user's home directory in a file called `.mysql_secret`
    - Sets up a login account with user and group names for `mysql` (for administering and running the server)

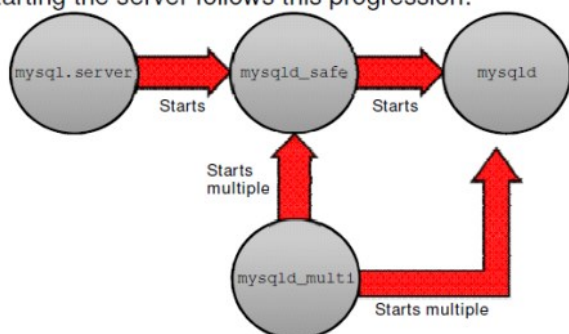
## Linux MySQL Server Installation Directories



## Windows MySQL Server Installation Directory



Starting the server follows this progression:



start mysql

```
# service mysql start
```

```
#!/etc/init.d/mysql start
```

stop mysql

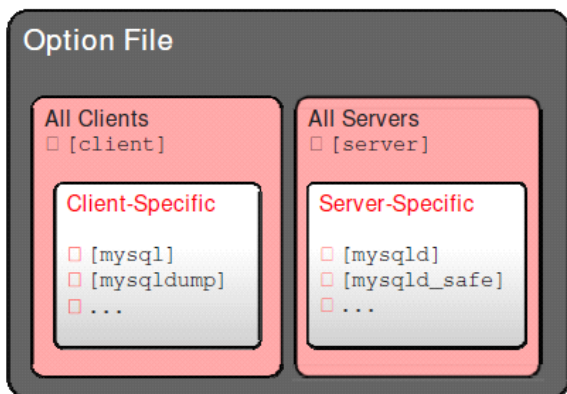
```
# service mysql stop
```

```
#!/etc/init.d/mysql stop
```

```
$ mysqladmin -uroot -p shutdown
```

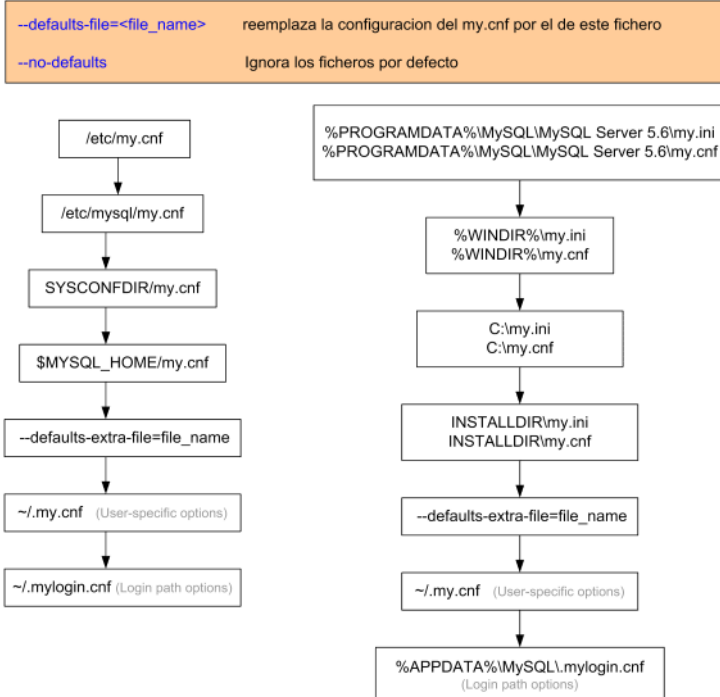
## MySQLAdmin - Server Configuration

Fichero : my.cnf



`my_print_defaults`

`mysql --print-defaults`



### Login paths

**mysql\_config\_editor:** Almacena las credenciales de acceso en un fichero de configuración especial.

Crea el fichero `<user_home>/mylogin.cnf`

Cada login\_path contiene una credencial de acceso unica

Uso: `mysq --login_path=<nombre>`

## Server System Variables

Variable categories:

GLOBAL	GLOBAL and SESSION	SESSION
Affects all sessions	Affects new session	Affects current session
Requires SUPER privilege	No privilege required	
Set variables at run time SET GLOBAL <variable> SET @@global.<variable>	Set variables at run time SET SESSION <variable> SET @@session.<variable>	

```
mysql> SHOW VARIABLES;
mysql> SHOW VARIABLES LIKE '%LOG%';

SELECT @@GLOBAL.sql_mode;
SELECT @@SESSION.sql_mode;
```

• Cambiar su contenido

```
SET GLOBAL sql_mode = 'modes';
SET SESSION sql_mode = 'modes';
```

## Ficheros de Log

Log File	Options	File Name	Programs
		Table Name	
Error	--log-error	host_name.err	N/A
General	--general_log	host_name.log	N/A
	general_log_file	general_log	
Slow Query	--slow_query_log	host_name-slow.log	mysqldumpslow
	--long_query_time	slow_log	
Binary	--log-bin --expire-logs-days	host_name-bin.000001	mysqlbinlog
Audit	--audit_log --audit_log_file ...	audit.log	N/A

## Binary Logs

	Statement-Based	Row-Based
Size of log files	Small	Large
Replication limitations	Not all statements can be replicated.	All statements can be replicated.
Master/Slave MySQL versions	Slave can be a newer version with a different row structure.	Slave must be an identical version and row structure.
Locking	INSERT and SELECT require a greater number of row locks.	INSERT, UPDATE, and DELETE require fewer locks on slaves.
Point-in-time recovery	Yes	Yes (more difficult due to binary format of log events)

Log shipping system:

```
show binary logs;
```

```
show binlog events \G;
```

```
SHOW MASTER STATUS
```

```
mysqlbinlog <fichero>
```

## Mantenimiento

```
expire_logs_days
```

```
PURGE { BINARY | MASTER } LOGS
      { TO 'log_name' | BEFORE datetime_expr }
```

```
PURGE BINARY LOGS TO 'mysql-bin.010';
```

```
PURGE BINARY LOGS BEFORE '2008-04-02 22:46:26';
```

#### **Deshabilitar**

```
sql_log_bin=0 con SET GLOBAL o SESSION (no en my.cnf)
```

```
#log-bin      No inicializando la variable
```

# MySQLAdmin - Conclusiones: Clients and Tools

Conectarnos a un cliente

```
shell> mysql --host=localhost --user=myname --password=mypass
shell> mysql --login-path=mypath
```

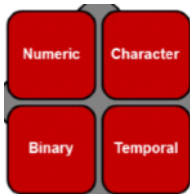
Otras opciones:

Para dummies	Opcion <b>--safe-updates</b> <b>\$ mysql --safe-updates</b>
<b>-t</b> formato tabla	# <b>mysql --login-path=admin -t</b>  14:54 (none)> select date(now()); +-----+   date(now())   +-----+   2015-09-06   +-----+ 1 row in set (0.00 sec)
<b>-B</b> formato batch	# <b>mysql --login-path=admin -B</b> select date(now()); date(now()) 2015-09-06
<b>-H</b> formato HTML	<b>mysql --login-path=admin -H</b>  14:54 (none)> select date(now()); <TABLE BORDER=1><TR><TH>date(now())</TH></TR><TR><TD>2015-09-06</TD></TR></TABLE>1 row in set (0.00 sec)
<b>-X</b> formato XML	# <b>mysql --login-path=admin -X</b>  14:54 (none)> select date(now()); <?xml version="1.0"?>  <resultset statement="select date(now());" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <row> <field name="date(now())">2015-09-06</field> </row> </resultset>

<b>HELP</b>	mysql> <b>HELP</b> Ayuda de comandos
<b>STATUS</b>	mysql> <b>STATUS</b> Identifica la conexion
<b>tee / notee</b>	mysql> <b>tee &lt;file&gt;</b> Escribe ademas de la salida standard a un fichero
<b>PROMPT</b>	mysql > <b>PROMPT(\u@\h) [\d]\&gt;</b> Cambia el prompt del cliente mysql
<b>SOURCE</b>	mysql > <b>SOURCE /usr/stage/world_innodb.sql</b> Ejecuta un fichero

# MySQLAdmin - Conclusiones: Data Types

MySQL soporta los siguientes tipos de Datos:



## Numeric Data Types

Class	Type	Description
Integer	TINYINT	Very small integer data type
Integer	SMALLINT	Small integer data type
Integer	MEDIUMINT	Medium-sized integer data type
Integer	INT	Normal- (average-) sized integer data type
Integer	BIGINT	Large integer data type
Floating-Point	FLOAT	Small, single-precision (four-byte) floating-point number
Floating-Point	DOUBLE	Normal, double-precision (eight-byte) floating-point number
Fixed-Point	DECIMAL	Exact-value numbers that have an integer part, a fractional part, or both
BIT	BIT	Bit-field values

## Binary Data Types

Class	Type	Description
Binary	BINARY	Similar to the CHAR (fixed-length) type, but stores binary byte strings instead of nonbinary character strings
Binary	VARBINARY	Similar to the VARCHAR (variable-length) type, but stores binary byte strings instead of nonbinary character strings
BLOB	TINYBLOB	BLOB column with a maximum length of 255 bytes
BLOB	BLOB	BLOB column with a maximum length of 65,535 bytes
BLOB	MEDIUMBLOB	BLOB column with a maximum length of 16,777,215 bytes
BLOB	LONGBLOB	BLOB column with a maximum length of 4,294,967,295 bytes

## Character String Data Types

Class	Type	Description
Text	CHAR	Fixed-length character string, up to a maximum of 255 characters
Text	VARCHAR	Variable-length character string, up to a maximum of 65,535 characters
Text	TINYTEXT	Variable-length character string, up to a maximum of 255 characters
Text	TEXT	Variable-length character string, up to a maximum of 65,535 characters
Text	MEDIUMTEXT	Variable-length character string, up to a maximum of 16,777,215 characters
Text	LONGTEXT	Variable-length character string, up to a maximum of 4,294,967,295 characters
Integer	ENUM	Enumeration consisting of a fixed set of legal values
Integer	SET	Set consisting of a fixed set of legal values

## Temporal Data Types

Type	Format	Example
DATE	YYYY-MM-DD	2006-08-04
TIME	hh:mm:ss[.uuuuuu]	12:59:02.123456
DATETIME	YYYY-MM-DD hh:mm:ss[.uuuuuu]	2006-08-04 12:59:02.123
TIMESTAMP	YYYY-MM-DD hh:mm:ss[.uuuuuu]	2006 08 04 12:59:02.12
YEAR	YYYY	2006

## Spatial Data Types

MySQL supports spatial data type extensions to enable the generation, storage, and analysis of geographic features.



# MySQLAdmin - Conclusiones: Obtaining Metadata

Para interrogar al Diccionario de datos por nuestro Sistema de Información y su salud.

## Metadata Access Methods

- View data that describes the structure of the database.
- Query the **INFORMATION\_SCHEMA** database tables.
  - They contain data about all objects managed by the MySQL database server.
- Use **SHOW** statements.
  - MySQL-proprietary statements for obtaining database and table information
- Use the **DESCRIBE** (or **DESC**) statement.
  - A shortcut to inspect table structure and column properties
- Use the **mysqlshow** client.
  - A command-line interface to the **SHOW** syntax

### SHOW Statements

SHOW DATABASES

SHOW INDEX

SHOW CREATE TABLE

SHOW CHARACTER SET

SHOW TABLES

SHOW COLLATION

SHOW OPEN TABLES

SHOW PROCESSLIST

SHOW COLUMNS

SHOW TABLE STATUS

# MySQLAdmin - Conclusiones: Transactions and Locking

**TRANSACCIÓN** Una colección de pasos de ejecución de manipulación de datos que son tratados como una sola unidad de trabajo

En una transacción el resultado es O TODOS O NINGUNO.

## Transaction SQL Control Statements

<b>START TRANSACTION (or BEGIN)</b>	Inicio de una transacion
<b>SAVEPOINT</b>	Asigna un punto en la transacción para futuras referencias
<b>COMMIT</b>	Hace permanente los cambios realizados en la transacción
<b>ROLLBACK</b>	Canela los cambios realizados en la transacción
<b>ROLLBACK TO SAVEPOINT</b>	Cancelo cambios ejecutados <b>después</b> del SAVEPOINT
<b>RELEASE SAVEPOINT</b>	Elimina el SAVEPOINT
<b>SET AUTOCOMMIT</b>	Habilita o deshabilita el autocommit en la conexión actual. EL habilitar AUTOCOMMIT <u>no desactiva las transacciones</u> . Se puede habilitar y deshabilitar por session.



OJO al **COMMIT** implicito !!!

## Isolation Level Problems

Isolation Level	Dirty Read	Non-Repeatable Read	Phantom Read
Read Uncommitted	Possible	Possible	Possible
Read Committed	Not possible	Possible	Possible
Repeatable Read	Not possible	Not possible	Possible*
Serializable	Not possible	Not possible	Not possible

\* Not possible for InnoDB, which uses snapshots for Repeatable Read

### – Syntax examples:

```
SET GLOBAL TRANSACTION ISOLATION LEVEL <isolation_level>;
SET SESSION TRANSACTION ISOLATION LEVEL <isolation_level>;
SET TRANSACTION ISOLATION LEVEL <isolation_level>;
```

Consulta : `mysql> select @@tx_isolation, @@global.tx_isolation, @@session.tx_isolation;`

```
+-----+-----+-----+
| @@tx_isolation | @@global.tx_isolation | @@session.tx_isolation |
+-----+-----+-----+
| REPEATABLE-READ | REPEATABLE-READ      | REPEATABLE-READ      |
+-----+-----+-----+
```

## Bloqueos

- **LOCK IN SHARE MODE** clause: A shared lock, which means that no other transactions can take exclusive locks but other transactions can also use shared locks. Because normal reads do not lock anything, they are not affected by the locks.



- **FOR UPDATE** (EXCLUSIVE LOCK) clause: Locks each selected row with an exclusive lock, preventing others from acquiring any lock on the rows but allowing reading of the rows.

	S	X
S	Compatible	Conflict
X	Conflict	Conflict

## Implicit Locks

The MySQL server locks the table (or row) based on the commands issued and the storage engines being used:

Operation	InnoDB	MyISAM
SELECT	No lock*	Table-level shared lock
UPDATE/DELETE	Row-level exclusive lock	Table-level exclusive lock
ALTER TABLE	Table-level shared lock	Table-level shared lock

\* No lock unless `SERIALIZABLE` level, `LOCK IN SHARE MODE`, or `FOR UPDATE` is used

# MySQLAdmin - Conclusiones: Storage Engines

A MySQL storage engine is a low-level engine inside the database server that takes care of storing and retrieving data, and can be accessed through an internal MySQL API or, in some situations, can be accessed directly by an application.

## Displaying the Storage Engine Setting

- Use **SELECT** to confirm the session storage engine:

```
SELECT @@default_storage_engine;
```

- Use **SHOW** to confirm the storage engine, per a table:

```
SHOW CREATE TABLE City\G
SHOW TABLE STATUS LIKE 'CountryLanguage'\G
```

- Use **INFORMATION\_SCHEMA** to confirm the storage engine, per a table:

```
SELECT TABLE_NAME, ENGINE FROM
INFORMATION_SCHEMA.TABLES
WHERE TABLE_NAME = 'City'
AND TABLE_SCHEMA = 'world_innodb'\G
```

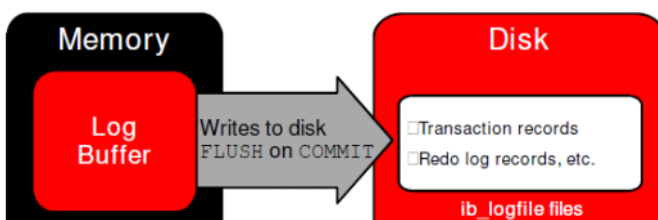
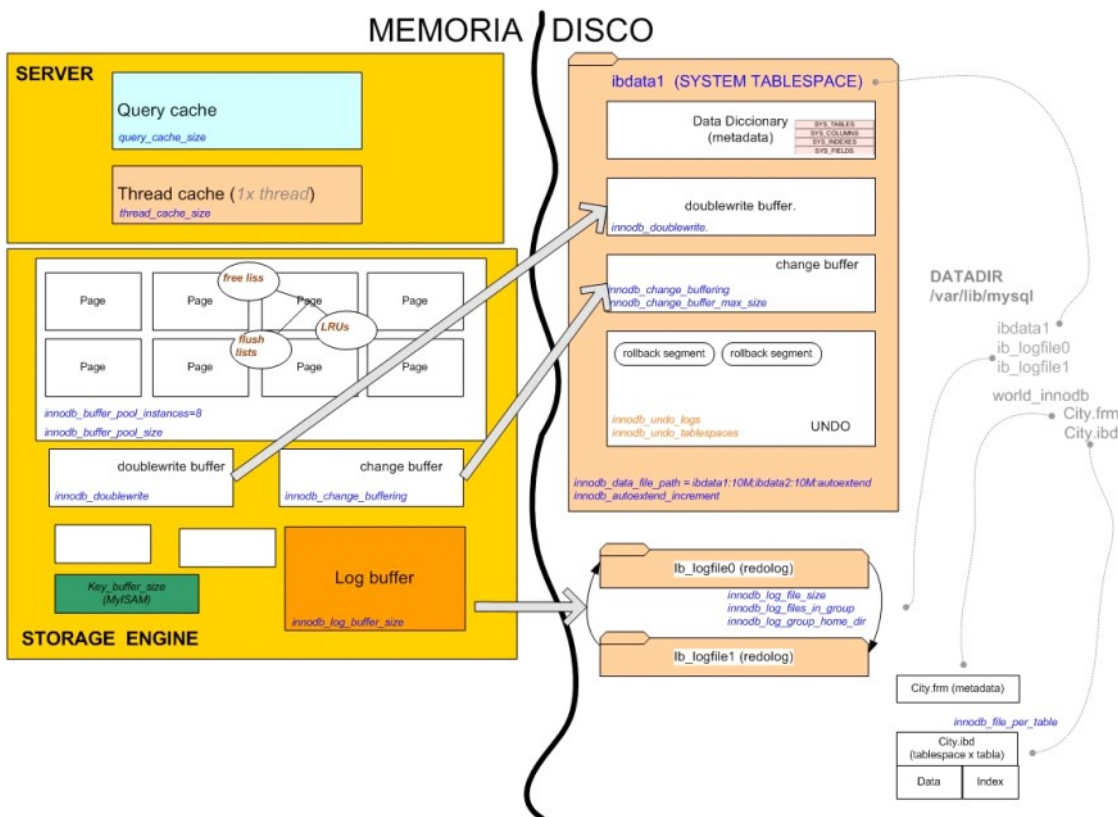
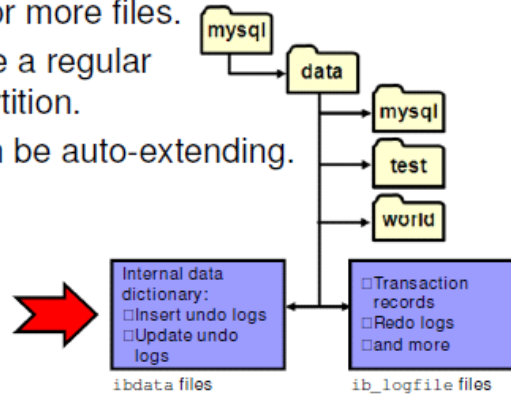
Que tiene InnoDB

**Table 14.1 InnoDB Storage Engine Features**

<i>Storage limits</i>	64TB	<i>Transactions</i>	Yes	<i>Locking granularity</i>	Row
<i>MVCC</i>	Yes	<i>Geospatial data type support</i>	Yes	<i>Geospatial indexing support</i>	Yes <sup>[a]</sup>
<i>B-tree indexes</i>	Yes	<i>T-tree indexes</i>	No	<i>Hash indexes</i>	No <sup>[b]</sup>
<i>Full-text search indexes</i>	Yes <sup>[c]</sup>	<i>Clustered indexes</i>	Yes	<i>Data caches</i>	Yes
<i>Index caches</i>	Yes	<i>Compressed data</i>	Yes <sup>[d]</sup>	<i>Encrypted data <sup>[e]</sup></i>	Yes
<i>Cluster database support</i>	No	<i>Replication support <sup>[f]</sup></i>	Yes	<i>Foreign key support</i>	Yes
<i>Backup / point-in-time recovery <sup>[g]</sup></i>	Yes	<i>Query cache support</i>	Yes	<i>Update statistics for data dictionary</i>	Yes
<sup>[a]</sup> InnoDB support for geospatial indexing is available in MySQL 5.7.5 and higher. <sup>[b]</sup> InnoDB utilizes hash indexes internally for its Adaptive Hash Index feature. <sup>[c]</sup> InnoDB support for FULLTEXT indexes is available in MySQL 5.6.4 and higher. <sup>[d]</sup> Compressed InnoDB tables require the InnoDB Barracuda file format. <sup>[e]</sup> Implemented in the server (via encryption functions), rather than in the storage engine. <sup>[f]</sup> Implemented in the server, rather than in the storage engine. <sup>[g]</sup> Implemented in the server, rather than in the storage engine.					

## InnoDB System Tablespace

- ❑ InnoDB metadata, undo log, and buffers are stored in a system “tablespace” by default.
- ❑ This is a single logical storage area, which can consist of one or more files.
- ❑ Each file can be a regular file or a raw partition.
- ❑ The last file can be auto-extending.



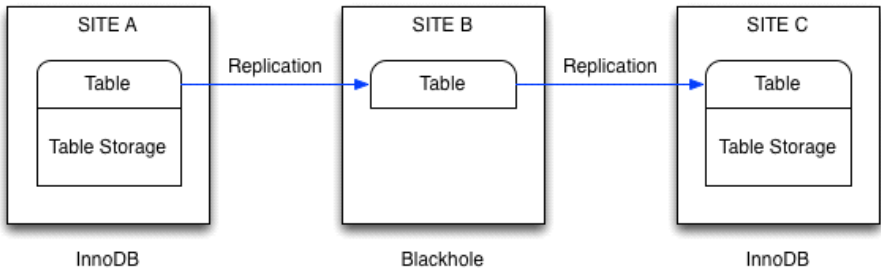
**innodb\_buffer\_pool\_dump\_at\_shutdown and innodb\_buffer\_pool\_load\_at\_startup.**

## Multiversioning

A rollback segment is an InnoDB storage area that contains the undo log.

- **DB\_TRX\_ID**: Six-byte field that indicates the transaction identifier for the last transaction that inserted or updated the row. Also, a deletion is treated internally as an update where a special bit in the row is set to mark it as deleted.
- **DB\_ROLL\_PTR**: Seven-byte field called the roll pointer. It points to an undo log record written to the rollback segment. If the row was updated, the undo log record contains the information necessary to rebuild the content of the row before it was updated.
- **DB\_ROW\_ID**: Six-byte field that contains a row ID that automatically increments as new rows are inserted. If InnoDB generates a clustered index automatically, the index contains row ID values. Otherwise, the DB\_ROW\_ID column does not appear in any index.

MyISAM	<p>The MyISAM storage engine stores each table on disk in three files (<b>.frm</b>, <b>.MYD</b>, and <b>.MYI</b>) and has the following features:</p> <ul style="list-style-type: none"><li>• Support for <b>FULLTEXT</b> searching and spatial data types</li><li>• Flexible <b>AUTO_INCREMENT</b></li><li>• Compressed, read-only tables, which save space</li><li>• Table-level locking to manage contention between queries</li><li>• Portable storage format</li><li>• Ability to specify the number of rows for a table</li><li>• Ability to control the updating of non-unique indexes for loading data into an empty table</li></ul>				
MEMORY	<p>The MEMORY storage engine creates tables with contents that are stored in memory, represented on disk by a <b>.frm</b> file.</p> <p>It has the following features:</p> <ul style="list-style-type: none"><li>• Table data and indexes that are stored in memory</li><li>• Very fast performance due to in-memory storage</li><li>• Fixed-length row storage format</li><li>• Table contents that do not survive restart</li><li>• Maximum size option <b>--max-heap-table-size</b></li><li>• Table-level locking</li></ul> <p>MEMORY tables:</p> <ul style="list-style-type: none"><li>• Cannot contain <b>TEXT</b> or <b>BLOB</b> columns</li><li>• Can use different character sets for different columns</li></ul>				
ARCHIVE	<p>The ARCHIVE storage engine is used for storing large volumes of data in a compressed format, allowing for a very small footprint. It has these primary characteristics:</p> <ul style="list-style-type: none"><li>• Represented by <b>.frm</b> file</li><li>• Data file: <b>.ARZ</b></li><li>• Does not support indexes</li><li>• Supports <b>INSERT</b> and <b>SELECT</b>, but not <b>DELETE</b>, <b>REPLACE</b>, or <b>UPDATE</b></li><li>• Supports <b>ORDER BY</b> operations and <b>BLOB</b> columns</li><li>• Accepts all but spatial data types</li></ul> <table><tr><td>Geospatial data type support</td><td>Yes</td><td>Geospatial indexing support</td><td>No</td></tr></table> <ul style="list-style-type: none"><li>• Uses row-level locking</li><li>• Supports <b>AUTO_INCREMENT</b> columns</li></ul>	Geospatial data type support	Yes	Geospatial indexing support	No
Geospatial data type support	Yes	Geospatial indexing support	No		

<b>BLACKHOLE</b>	<p>The BLACKHOLE storage engine acts as a “black hole” that accepts data but throws it away and does not store it. This storage engine has these primary characteristics:</p> <ul style="list-style-type: none"> <li>• Represented by <b>.frm</b> file</li> <li>• Used for replication</li> <li>• Supports all kinds of indexes</li> <li>• Retrievals always return an empty result.</li> <li>• Verification of dump file syntax</li> <li>• Measurement of the overhead from binary logging</li> <li>• “No-op” storage engine that can be used for finding performance bottlenecks not related to the storage engine</li> <li>• Transaction-aware</li> </ul>  <pre> graph LR     subgraph SITE_A [SITE A]         direction TB         T1[Table] --- TS1[Table Storage]     end     subgraph SITE_B [SITE B]         direction TB         T2[Table]     end     subgraph SITE_C [SITE C]         direction TB         T3[Table] --- TS3[Table Storage]     end     T1 -- Replication --&gt; T2     T2 -- Replication --&gt; T3     SITE_A --- InnoDB_A[InnoDB]     SITE_B --- Blackhole[Blackhole]     SITE_C --- InnoDB_C[InnoDB] </pre>
<b>CSV</b>	<p>Los ficheros que la componen tiene extensión <b>.frm y .CSV</b></p> <p><a href="https://dev.mysql.com/doc/refman/5.6/en/csv-storage-engine.html">https://dev.mysql.com/doc/refman/5.6/en/csv-storage-engine.html</a></p>

# MySQLAdmin - Conclusiones: Partitioning

¿Esta habilitado el partitioning? > **show plugins ;**

```
partition          ACTIVE
```

```
SELECT PLUGIN_NAME as Name,
       PLUGIN_VERSION as Version,
       PLUGIN_STATUS as Status
FROM   INFORMATION_SCHEMA.PLUGINS
WHERE  PLUGIN_TYPE='STORAGE ENGINE';
```

Deshabilitar partitioning

**mysqld --skip-partition**

Tipo		Variante	
RANGE	<pre>CREATE TABLE employees (   id INT NOT NULL,   fname VARCHAR(30),   lname VARCHAR(30),   hired DATE NOT NULL DEFAULT '1970-01-01',   separated DATE NOT NULL DEFAULT '9999-12-31',   job_code INT NOT NULL,   store_id INT NOT NULL ) PARTITION BY RANGE (store_id) (   PARTITION p0 VALUES LESS THAN (6),   PARTITION p1 VALUES LESS THAN (11),   PARTITION p2 VALUES LESS THAN (16),   PARTITION p3 VALUES LESS THAN (21),   PARTITION p4 VALUES LESS THAN MAXVALUE );</pre>	COLUMNS	<pre>CREATE TABLE rcx (   a INT,   b INT,   c CHAR(3),   d INT) PARTITION BY RANGE COLUMNS(a,d,c) (   PARTITION p0 VALUES LESS THAN (5,10,'ggg'),   PARTITION p1 VALUES LESS THAN (10,20,'mmm'),   PARTITION p2 VALUES LESS THAN (15,30,'sss'),   PARTITION p3 VALUES LESS THAN (MAXVALUE,MAXVALUE,MAXVALUE));</pre>
LIST	<pre>CREATE TABLE h2 (   c1 INT,   c2 INT) PARTITION BY LIST(c1) (   PARTITION p0 VALUES IN (1, 4, 7),   PARTITION p1 VALUES IN (2, 5, 8) );</pre>	COLUMNS	
HASH	<pre>CREATE TABLE employees (   id INT NOT NULL,   fname VARCHAR(30),   lname VARCHAR(30),   hired DATE NOT NULL DEFAULT '1970-01-01',   separated DATE NOT NULL DEFAULT '9999-12-31',   job_code INT,   store_id INT ) PARTITION BY HASH(store_id) PARTITIONS 4;</pre>	LINEAR	<pre>CREATE TABLE t1 (col1 INT, col2 CHAR(5), col3 DATE) PARTITION BY LINEAR HASH( YEAR(col3) ) PARTITIONS 6;</pre> <p><a href="https://dev.mysql.com/doc/refman/5.6/en/partitioning-linear-hash.html">https://dev.mysql.com/doc/refman/5.6/en/partitioning-linear-hash.html</a></p>
KEY	<pre>CREATE TABLE k1 (   id INT NOT NULL PRIMARY KEY,   name VARCHAR(20) ) PARTITION BY KEY() PARTITIONS 2;</pre> <p>If there is no primary key but there is a unique key, then the unique key is used for the partitioning key:</p> <pre>CREATE TABLE k1 (   id INT NOT NULL,   name VARCHAR(20),</pre>	LINEAR	<pre>CREATE TABLE tk (   col1 INT NOT NULL PRIMARY KEY,   col2 CHAR(5),   col3 DATE ) PARTITION BY LINEAR KEY ( ) PARTITIONS 3;</pre>



```

    UNIQUE KEY (id)
)
PARTITION BY KEY()
PARTITIONS 2;

```

```
SHOW CREATE TABLE
```

```
SHOW TABLE STATUS
```

```
select * from INFORMATION_SCHEMA.PARTITIONS where table_name = 'rc1'
```

```
EXPLAIN PARTITIONS SELECT
```

```

| ADD PARTITION (partition_definition)
| DROP PARTITION partition_names
| TRUNCATE PARTITION {partition_names | ALL}
| COALESCE PARTITION number Mismo comando que DROP PARTITION para HASH y KEY
| REORGANIZE PARTITION partition_names INTO (partition_definitions) Renombrar particiones
| EXCHANGE PARTITION partition_name WITH TABLE tbl_name
| ANALYZE PARTITION {partition_names | ALL}
| CHECK PARTITION {partition_names | ALL}
| OPTIMIZE PARTITION {partition_names | ALL} Reclama el espacio no-usado
| REBUILD PARTITION {partition_names | ALL} Tiene el mismo efecto que eliminar todas la filas y volverlas a insertar (desfragmentacion)
| REPAIR PARTITION {partition_names | ALL} Repara las particiones corruptas
| REMOVE PARTITIONING

```

Partitions are stored in their own files in the data directory by default. Use DATA DIRECTORY to specify an alternative partition location:

```

CREATE TABLE entries (id INT, entered DATE)
PARTITION BY RANGE(YEAR(entered)) (
PARTITION p0 VALUES LESS THAN (2000) DATA DIRECTORY = '/data/p0',
PARTITION p1 VALUES LESS THAN MAXVALUE DATA DIRECTORY = '/data/p1'
);

```

## Subpartition

RANGE and LIST partitioned tables can be subpartitioned. The subpartitions themselves can be HASH or KEY.

```

CREATE TABLE orders_range_hash ( ...
...) ENGINE = InnoDB
PARTITION BY RANGE(id)
SUBPARTITION BY HASH(store_id)
SUBPARTITIONS 2 (
PARTITION p0 VALUES LESS THAN(10000),
PARTITION p1 VALUES LESS THAN(20000),
PARTITION p2 VALUES LESS THAN(30000),
PARTITION p3 VALUES LESS THAN(40000),
PARTITION p4 VALUES LESS THAN(50000)
);

```

## Limitaciones y Restricciones del Partitioning

### General

- The maximum number of partitions per table is **8192**.
- Spatial types are not supported.
- Temporary tables cannot be partitioned.
- It is not possible to partition log tables.

### Foreign keys and indexes

- Foreign keys are not supported.
- FULLTEXT indexes are not supported.
- **No global indexes**: Each partition has its own indexes.

Subpartitioning is possible only:

- – When partitioning by RANGE and LIST
- – By LINEAR HASH or LINEAR KEY
- Expressions used for RANGE, LIST, and HASH partitions must evaluate as an integer.
  - RANGE COLUMNS and LIST COLUMNS allow a wider range of data types.
- You cannot use TEXT or BLOB in partitioning expressions.
- UDFs, stored functions, variables, some operators, and some built-in functions are not allowed.
  - Operators: |, &, ^, <<, >>, ~
- SQL modes should not be changed after table creation.
- Subqueries are not supported in partitioning expressions.
- All columns used in the partitioning expression must be part of all of the table's unique indexes.

# MySQLAdmin - Conclusiones: User Management

There are several ways to set a MySQL user password:

- **CREATE USER...IDENTIFIED BY**
- **GRANT...IDENTIFIED BY**
- **SET PASSWORD**
- **mysqladmin password**
- **UPDATE** grant tables (*not recommended*)

The **SET PASSWORD** statement is the most common method for setting or changing an account password, as

Crear cuenta de usuario	<code>CREATE USER ... IDENTIFIED BY ...</code>
Asignar / Cambiar de password	<code>GRANT ... IDENTIFIED BY ..</code>  <code>SET PASSWORD = PASSWORD('clave')</code> <code>SET PASSWORD FOR 'user'@'host' = PASSWORD('clave')</code>  <code>mysqladmin -u... -h.. -p... password 'clave'</code>
Cuenta anonima	<code>CREATE USER ''@localhost</code>
Expirar password	<code>ALTER USER 'user'@'host' PASSWORD EXPIRE;</code>
Renombrar cuenta de usuario	<code>RENAME USER old_user TO new_user</code>
	<code>DROP USER 'user'@'host';</code>
Authentication Plugins	<b>auth_socket</b> Allows only MySQL users who are logged in via a UNIX socket from a UNIX account with the same name  <pre>[mysqld] plugin-load=auth_socket.so</pre> <b>authentication_pam Enterprise Edition</b> Allows you to log in using an external authentication mechanism  <pre>[mysqld] plugin-load=authentication_pam.so</pre>
Pluggin: Password Validation  <b>validate_password</b>	Variable <b>validate_password_policy</b> Niveles  LOW: Comprueba la longitud minima > <code>validate_password_length</code>  MEDIUM : LOW + Clave contiene num de caract numericos > <code>validate_password_number_count</code> Clave contiene num de "mixed" caract > <code>validate_password_mixed_case_count</code> Clave contiene num de caract especiales > <code>validate_password_special_char_count</code>  STRONG: MEDIUM + cuatro o más caracteres no deben coincidir con palabras el archivo de diccionario especificado en el <code>validate_password_dictionary</code>

	<pre> +-----+-----+   Variable_name   Value   +-----+-----+   validate_password_dictionary_file            validate_password_length            8         validate_password_mixed_case_count   1         validate_password_number_count      1         validate_password_policy            MEDIUM     validate_password_special_char_count   1       +-----+-----+ </pre> <p>INSTALL PLUGIN validate_password SONAME 'validate_password.so';</p> <p>[mysqld]  plugin-load=validate_password.so  validate-password=FORCE_PLUS_PERMANENT</p>
Comprobar la fortaleza de una clave	<p>VALIDATE_PASSWORD_STRENGTH()  —  Returns an integer in the range 0 (weak) to 100 (strong)</p>
	<pre> mysql&gt; show grants for current_user \G;  mysql&gt; show grants \G; </pre>
Disabling Client Access Control	<p>The <b>--skip-grant-tables</b> option has the following effects:</p> <ul style="list-style-type: none"> <li>•When connected, the user has full privileges to do anything.</li> <li>•This option disables account management statements such as CREATE USER, GRANT, REVOKE, and SET PASSWORD.</li> </ul> <p><b>--skip-networking</b>  option to prevent network access and allow access only on local socket, named pipe, or shared memory.</p>
Account Resource Limits	<p>Limit the following server resources for individual accounts:</p> <ul style="list-style-type: none"> <li>– <b>MAX_QUERIES_PER_HOUR</b>: The number of queries that an account can issue per hour</li> <li>– <b>MAX_UPDATES_PER_HOUR</b>: The number of updates that an account can issue per hour</li> <li>– <b>MAX_CONNECTIONS_PER_HOUR</b>: The number of times an account can connect to the server per hour</li> <li>– <b>MAX_USER_CONNECTIONS</b>: The number of simultaneous connections allowed</li> </ul> <pre> mysql&gt; GRANT ALL ON customer.* TO 'francis'@'localhost' -&gt;      WITH MAX_QUERIES_PER_HOUR 20 -&gt;          MAX_UPDATES_PER_HOUR 10 -&gt;          MAX_CONNECTIONS_PER_HOUR 5 -&gt;          MAX_USER_CONNECTIONS 2; </pre>

# MySQLAdmin - Conclusiones: Security

To check whether a **mysqld** server supports SSL, examine the value of the [have\\_ssl](#) system variable:


```
mysql> SHOW VARIABLES LIKE 'have_ssl';
+-----+-----+
| Variable_name | Value      |
+-----+-----+
| have_ssl      | DISABLED   |
+-----+-----+
1 row in set (0.00 sec)
```

Si el valor está DISABLED , hay que habilitarlo con --ssl

```
mysql> desc mysql.user;
```

Field	Type	Null	Key	Default	Extra
Host	char(60)	NO	PRI		
User	char(16)	NO	PRI		
Password	char(41)	NO			
ssl_type	enum('', 'ANY', 'X509', 'SPECIFIED')	NO			
ssl_cipher	blob	NO		NULL	
x509_issuer	blob	NO		NULL	
x509_subject	blob	NO		NULL	

# MySQLAdmin - Conclusiones: Table Maintenance

<b>ANALYZE</b>	<p>Updates index statistics</p> <p><b>ANALYZE [NO_WRITE_TO_BINLOG   LOCAL] TABLE <i>tbl_name</i> [, <i>tbl_name</i>] ...</b></p> <p>Works with <b>InnoDB</b>, <b>NDB</b>, and <b>MyISAM</b> tables</p>				
<b>CHECK TABLE</b>	<p>Checks the integrity of the table</p> <p>Funciona con InnoDB, CSV, MyISAM, y tablas ARCHIVE</p> <p>Mensajes de salida de la ejecución deben ser:</p> <table> <tr> <td>Correcto</td><td>OK Table is already up to date</td></tr> <tr> <td>ERROR</td><td>corrupted not closed properly</td></tr> </table>	Correcto	OK Table is already up to date	ERROR	corrupted not closed properly
Correcto	OK Table is already up to date				
ERROR	corrupted not closed properly				
<b>CHECKSUM</b>	<p>Reports a checksum for the table</p> <p><b>CHECKSUM TABLE <i>tbl_name</i> [, <i>tbl_name</i>] ... [ QUICK   EXTENDED ]</b></p> <p><b>InnoDB y MyISAM.</b></p>				
<b>OPTIMIZE</b>	<p>Optimizes the table</p> <p><b>OPTIMIZE [NO_WRITE_TO_BINLOG   LOCAL] TABLE <i>tbl_name</i> [, <i>tbl_name</i>] ...</b></p> <p>Aplica: InnoDB, MyISAM, y ARCHIVE</p>				
<b>REPAIR</b>	<p>Repairs the table</p> <p><b>REPAIR [NO_WRITE_TO_BINLOG   LOCAL] TABLE <i>tbl_name</i> [, <i>tbl_name</i>] ... [QUICK] [EXTENDED] [USE_FRM]</b></p> <p>Aplica: MyISAM y ARCHIVE</p> <div>  <p>Realizar un backup antes de utilizar el REPAIR bajo algunas circunstancias, la operación podría ocasionar pérdida de datos</p> <p>Si se produce un "crash" durante un REPAIR TABLE se debería ejecutar otro nuevamente tras reiniciar el servidor antes de hacer cualquier otra operación con la tabla.</p> </div>				

Herramientas para hacer por línea de comando : **mysqlcheck** y **myisamchk**

Options to control the type of maintenance performed:

Both	mysqlcheck only	myisamchk only
<b>--analyze</b> or <b>-a</b> <b>--check</b> or <b>-c</b> <b>--check-only-changed</b> or <b>-C</b> <b>--fast</b> or <b>-F</b> <b>--medium-check</b> or <b>-m</b> <b>--quick</b> or <b>-q</b>	<b>--auto-repair</b> <b>--extended</b> or <b>-e</b> <b>--repair</b> or <b>-r</b>	<b>--extend-check</b> or <b>-e</b> <b>--recover</b> or <b>-r</b>

## Troubleshooting con InnoDB

InnoDB se recupera con cada reinicio del servidor. En caso de errores en tablas InnoDB usar **--innodb\_force\_recovery**

## Mantenimiento de tablas MyISAM

```
mysqld]
myisam-recover=FORCE, BACKUP
```

## Mantenimiento de tablas Memory

Las tabla MEMORY no liberan memoria cuando se eliminan con una sentencia DELETE.  
Liberar memoria **ALTER TABLE ENGINE=MEMORY**



## **Mantenimiento de tablas ARCHIVE**

Para mejorar la compresión utilizar: OPTIMIZE y/o REPAIR

# MySQLAdmin - Conclusiones: Exporting and Importing Data

## Exporting Data by Using SELECT with INTO OUTFILE

```
SELECT * FROM TABLA
WHERE ...
INTO OUTFILE '/var/tmp/report.txt'
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\n';"
```

```
FIELDS
TERMINATED BY 'string'
ENCLOSED BY 'char'
ESCAPED BY 'char'
LINES TERMINATED BY 'string'
```

## Importing Data by Using LOAD DATA INFILE

```
LOAD DATA INFILE '/tmp/City.txt'
INTO TABLE City
FIELDS TERMINATED BY ',';
```

Desde un cliente remoto, la fórmula es: **LOAD DATA LOCAL INFILE**

Deshabilitar carga de ficheros remotos **--local-infile=0**

Restringir la carga de ficheros desde un único directorio **secure\_file\_priv**

```
LOAD DATA INFILE '/tmp/City.txt' INTO TABLE City IGNORE 2 LINES
```

Uso de variables (@Name) , SET col=...

Control de Duplicados funciona igual que en los INSERT

- Con IGNORE se descarta la fila y
- Con REPLACE reemplaza la fila existente con la nueva del fichero

# MySQLAdmin - Conclusiones: Programming Inside MySQL

## Stored Procedures

<b>DELIMITER //</b>	<b>CALL</b> <database>.<nombre> ()
<b>CREATE PROCEDURE</b> nombre ([parámetro1,parámetro2,...]) [Atributos de la rutina]	<b>SHOW CREATE PROCEDURE</b>
<b>BEGIN instrucciones</b>	<b>SHOW CREATE FUNCTION</b>
<b>DECLARE ...</b>	<b>SHOW PROCEDURE STATUS</b>
	<b>SHOW FUNCTION STATUS</b>
<b>SET ...</b>	
<b>RETURN</b>	<b>INFORMATION_SCHEMA.ROUTINES</b>
<b>END //</b>	mysql.proc
	mysql.procs_priv

## TRIGGERS

Si un trigger falla, en Storage Engine transaccionales se ejecuta un rollback

Triggers NO se disparan:

- En causas de FOREIGN KEYS CASCADE
- Durante la replicación ROW-BASED

```
CREATE [DEFINER={usuario|CURRENT_USER}]
TRIGGER nombre_del_trigger {BEFORE|AFTER}
{UPDATE|INSERT|DELETE}
ON nombre_de_la_tabla
FOR EACH ROW
<bloque_de_instrucciones>
```

```
SHOW CREATE TRIGGER nombre_trigger;
SHOW TRIGGERS;
```

```
INFORMATION_SCHEMA.TRIGGERS
Privilege: TRIGGER
```

## EVENTS

```
SET GLOBAL event_scheduler=ON;
```

```
CREATE
  [DEFINER = { user | CURRENT_USER }]
  EVENT
  [IF NOT EXISTS]
  event_name
  ON SCHEDULE schedule
  [ON COMPLETION [NOT] PRESERVE]
  [ENABLE | DISABLE | DISABLE ON SLAVE]
  [COMMENT 'comment']
  DO event_body;
```

```
SHOW CREATE EVENT evento \G;
SHOW EVENTS
```

```
SUPER para cambiar event_scheduler
```

```
mysql> GRANT EVENT ON myschema.* TO user1@srv1;  
mysql> GRANT EVENT ON *.* TO user1@srv1;
```

Use **REVOKE** to cancel an event privilege:

```
REVOKE EVENT ON myschema.* FROM user1@srv1;
```

[INFORMATION\\_SCHEMA.EVENTS](#)

# MySQLAdmin - Conclusiones: Backup and Recovery

## Comparing Backup Methods

Method	Hot/Warm/ Cold	Storage Engines	Logical/ Physical	Consistent	Availability
MySQL Enterprise Backup	Hot (InnoDB)/ warm (other)	All	Physical	Yes	Commercially available
<b>mysqlhotcopy</b>	Warm	MyISAM	Physical	Yes	Freely available
<b>mysqldump</b>	Hot (InnoDB)/ warm (other)	All	Logical	Yes	Freely available
Snapshots	Hot	All	Physical	Yes	Need snapshot volume or file system
Replication	Hot	All	Logical or physical	Yes	Freely available
SQL Statements	Warm	All	Logical	No	Freely available
OS Copy Commands	Cold or warm	All	Physical	Yes	Freely available

### MySQL Enterprise Backup

#### Backup Físico

Que incluye?

InnoDB data

ibdata\* files: Shared tablespace files  
 .ibd files: Per-table data files  
 ib\_logfile\* files: Log files

All files in the data directory to include:

.opt files: Database configuration information  
 .TRG files: Trigger parameters  
 .MYD files: MyISAM data files  
 .MYI files: MyISAM index files  
 .FRM files: Table data dictionary file

Minimum privileges required:

- **RELOAD** on all databases and tables
- **CREATE TEMPORARY TABLES** on **mysql** database
- **CREATE, INSERT, and DROP** on tables:
  - **mysql.ibbackup\_binlog\_marker**
  - **mysql.backup\_progress**
  - **mysql.backup\_history**
- **SELECT** on **mysql.backup\_history** table
- **REPLICATION CLIENT** to retrieve the binlog position stored with the backup
- **SUPER** to optimize locking and minimize disruption

### mysqlhotcopy

#### Backup Físico

Aplica \_ MyISAM y ARCHIVE

Solo en UNIX. En Local

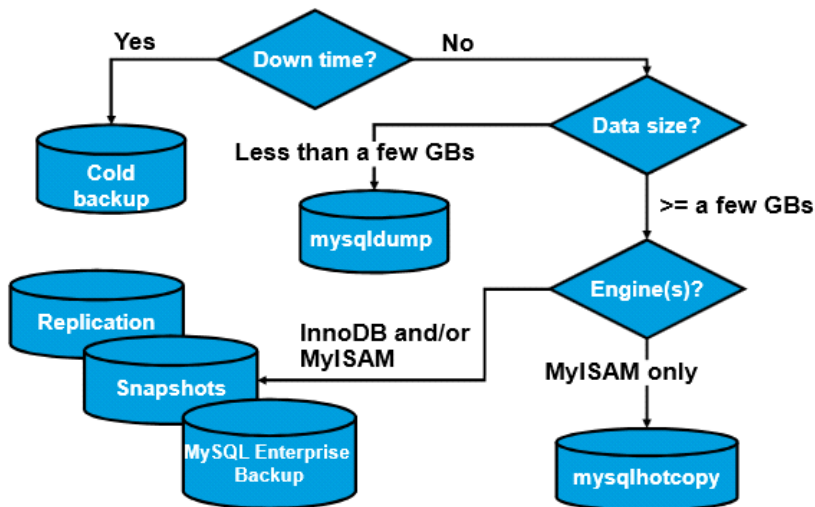
### Raw InnoDB Backups (= transportable)

Permite copiar un tablespace de un server a otro.

tablespace)	<p>1. <b>FLUSH TABLES</b> <i>table, table, table</i> <b>FOR EXPORT</b></p> <p>2. Copiar desde el sistema operativo</p> <p>3. <b>UNLOCK TABLES</b></p> <p>Pasos <b>RESTORE completo</b></p> <p>1.- Parar el mysql</p> <p>2.- Copiar los ficheros del backup</p> <p>3.- Arrancar mysql</p> <p>El <b>RESTORE</b> individual de tablas hacerlo con</p> <p><b>ALTER TABLE ... IMPORT TABLESPACE</b></p>
Raw MyISAM and ARCHIVE Backups	<p>Permite copiar tablas de MyISAM entre servidores</p> <p>1.- <b>FLUSH TABLES</b> <i>table, table, table</i> <b>WITH READ LOCK</b></p> <p>2.- Copiar desde el sistema operativo</p> <p>3.- <b>FLUSH LOGS</b></p> <p>4.- <b>UNLOCK TABLES</b></p> <p>Pasos <b>RESTORE</b></p> <p>1.- Parar el mysql</p> <p>2.- Copiar los ficheros del backup</p> <p>3.- Arrancar mysql</p>
LVM Snapshots	Backup Físico para toda la BBDD
mysqldump	<p>Backup <b>Lógico</b></p> <div> <p>Backup</p> <pre>mysqldump --user=&lt;user&gt; --password=&lt;password&gt; --opt db_name &gt; backup.file</pre> <p>Restore</p> <pre>mysql --login-path=&lt;login-path&gt; &lt;database&gt; &lt; backup_file.sql</pre> </div>
mysqlimport	<p>Igual que el LOAD DATA INTO</p> <pre>shell&gt; cd &lt;backup_dir&gt; shell&gt; mysqlimport -u&lt;user&gt; -p&lt;password&gt; &lt;database&gt; table.tsv</pre>
Otros ficheros que hacer Backup	<p><b>Binary</b> Logs</p> <p>Ficheros de configuración my.cnf o my.ini</p> <p>Replicación: master.info, relay-log.info</p> <p>Replicación slave data files: SQL_LOAD-*</p> <p>...</p>
Hacer el Backup desde el slave	<p>Pasos <b>BACKUP</b></p> <p>1.- Parar MySQL o parar el slave <b>STOP SLAVE SQL_THREAD (+ FLUSH TABLES)</b></p> <p>2.- Hacer el backup</p> <p>3.- Arrancar MySQL o arrancar el slave <b>START SLAVE SQL_THREAD</b></p>



## Backup Strategy

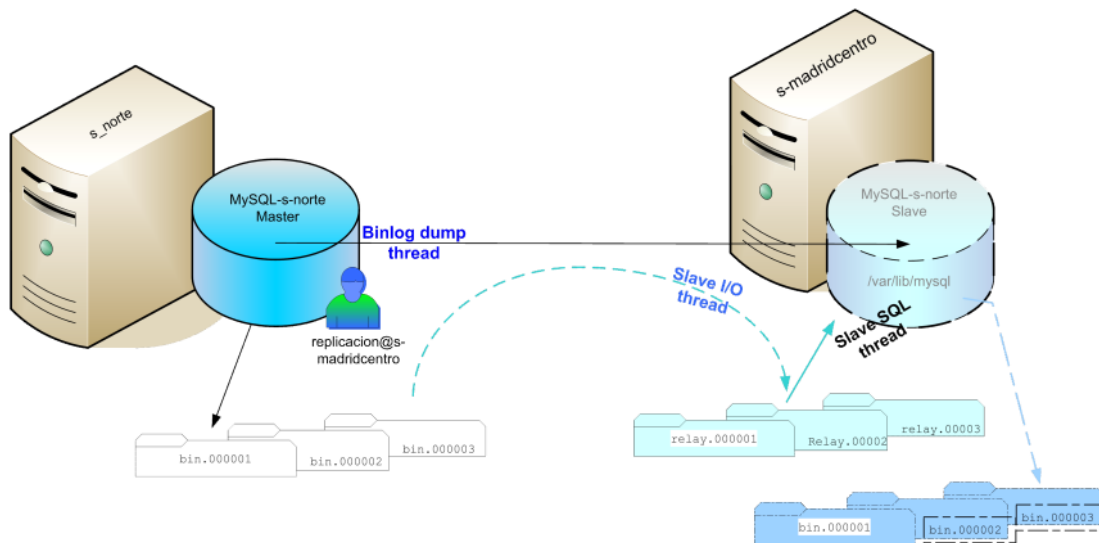


### Utilizar los binary logs para una recuperación

`mysqlbinlog [options] log_file ... |mysql`

```
server1# mysqlbinlog --database=sakila /var/log/mysql/mysql-bin.000215  
--stop-position=352 | mysql -uroot -p  
server1# mysqlbinlog --database=sakila /var/log/mysql/mysql-bin.000215  
--start-position=429 | mysql -uroot -p
```

# MySQLAdmin - Conclusiones: Replication



## Cómo funciona?

En el maestro se graba los cambios realizados en el binlog

El esclavo tiene 2 componentes:

1. Abre un thread E/S que conecta al maestro, y le pide los cambios en los binlogs y Escribe estos cambios en los relaylogs
2. Abre otro Thread SQL que lee los relaylogs y aplica los cambios localmente
  - a. Puede que se escriban estos cambios en los binlogs del esclavo si también es un maestro

## Implementar una Replicación

### 1.- Configurar el MASTER

```
[mysql]
#
# -- Configuración en master para la replicación -----
#
server-id=1
log_bin=XXX-bin
```

Crear un usuario con el privilegio REPLICATION SLAVE

```
mysql> GRANT REPLICATION SLAVE ON *.* TO 'replicacion'@'server' IDENTIFIED BY 'cLave';
```

Realizar un **backup** de la Base de Datos **MASTER**.

Identifica el punto desde el que se desea comenzar a replicar

```
mysql> SHOW MASTER STATUS;
```

### 2.- Configurar el SLAVE

En cada servidor SLAVE de un grupo de replicación, es necesario establecer un ID de servidor único en cada uno de ellos y distinto del MASTER

```
# -----
[mysql]
#
# -- Configuración en master para la replicación -----
#
#
server-id=2
```

Restaurar el backup

### 3.- Establecer la replicación

Se ejecuta el comando CHANGE MASTER TO para establecer la replicación desde el slave.

```
mysql> CHANGE MASTER TO
MASTER_HOST='server_master', MASTER_PORT=13309,
MASTER_USER='replicacion', MASTER_PASSWORD='****',
MASTER_LOG_FILE='bin.000002',MASTER_LOG_POS=53219;
```

#### 4.- Arrancar la replicación

```
mysql> start slave;
```

## GTIDS

Global Transaction Identifiers (GTIDs) identifica de forma única cada transacción en una arquitectura de réplica.

Cada GTID tiene la forma

```
<source-uuid>.<transaction-id>
```

```
<source-uuid>.<transaction-id-desde>-<transaction-id-hasta>
```

#### .- Configurar con GTIDs

Configura las variables

<b>gtid-mode=ON</b>	Registrará cada transacción con un único GTID
<b>enforce-gtid-consistency</b>	Deshabilita cualquier evento que no se pueda registrar como una transacción segura (Nuevo desde 5.6.9)  <i>Sentencias como CREATE TABLE ..SELECT o CREATE TEMPORARY TABLE no se puedan usar.</i>
<b>log-slave-updates</b>	Registra los eventos replicados en el binary log del slave

```
CHANGE MASTER TO MASTER_AUTO_POSITION=1;
```

## MySQL Tools para la Replicación

```
mysqldbcopy
mysqldbcompare
mysqlrpladmin
mysqlfailover
mysqlrplcheck
mysqlreplicate
mysqlrplshow
```

## Replicación SEMISINCRONA

MySQL por defecto utiliza la replicación **ASINCRONA**.

Replicación **SEMISINCRONA** = el MASTER confirma el COMMIT al menos en un esclavo que también tiene la información de la transacción. OJO, en caso de timeout del slave se comporta como asíncrona.

#### Requisitos

Debe de instalarse el plugin en el MASTER y en al menos 1 de los SLAVES.

MASTER	MASTER	SLAVE
Plugin	<b>rpl_semi_sync_master</b>	<b>rpl_semi_sync_slave</b>
Configuración	rpl_semi_sync_master_enabled	rpl_semi_sync_slave_enabled
	rpl_semi_sync_master_timeout = 10000 (milisegundos por defecto)	

## Cuantos HILOS o THREADS se usan en la Replicación

MASTER	SLAVE
<b>Binlog dump thread</b>	<b>Slave I/O thread</b>
Este hilo envía eventos dentro del log binario al esclavo	Se conecta al MASTER, lee del Thread Binlog Dump las transacciones y las escribe en su local RELAY Log.

mientras que el esclavo esté conectado.

El maestro crea un thread Binlog dump para cada esclavo conectado

#### **Slave SQL thread**

Ejecuta los eventos que lee del Relay Log.

#### **Multithreaded Slaves**

El proceso de replicación usando un único hilo para aplicar los cambios y puede ser un cuello de botella. Las operaciones de escritura (COMMIT) se aplican una a una.

MySQL 5.6 tiene opciones si escribes a distintas bases de datos de aplicar estos cambios en paralelo

**slave\_parallel\_workers**