

## U.T. III Elementos de MySQL.

Manual de referencia MySQL 5.0:

<http://download.nust.na/pub6/mysql/doc/refman/5.0/es/index.html>

### 1. Motores MySQL.

- 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
- Algunos de ellos son:
  - o **MyISAM:** Es la evolución del tipo de tabla original conocida como ISAM (Indexed sequential access method). Son tablas especialmente útiles cuando los datos no cambian con frecuencia y la operación más usada es la de lectura.
  - o **InnoDB:** Proporcionan bloqueo por registro y control de transacciones. Para lograr esto, se sacrifica un poco la velocidad de acceso.
  - o **HEAP:** Se almacenan en memoria RAM, algo que las hace extremadamente rápidas, aunque tienen el inconveniente de que existen mientras el servidor que contiene la base de datos esté funcionando.
  - o **MERGE:** MySQL permite agrupar tablas de idénticas características bajo un mismo nombre.
- Cuando se crea una nueva tabla, puede indicársele al servidor el tipo de tabla a crear añadiendo al final de la sentencia la opción *ENGINE = nombre\_motor;*
  - o La opción por defecto es MyISAM.

- Manual: *14. Motores de almacenamiento de MySQL y tipos de tablas.*

## 2. Diccionario de datos.

- El diccionario de datos es un componente esencial de cualquier SGBD, ya que contiene información sobre todos los objetos de las BBDD existentes en nuestro gestor.
- El diccionario de datos contiene lo que se denomina "Metadatos", que es información acerca de los datos, tales como los nombres de los objetos, tipos de datos de las columnas, permisos de acceso, etc.
- En la mayoría de los casos el diccionario de datos se almacena en forma de BD, que en MySQL se denomina *information\_schema*.
- *information\_schema* es una BD que almacena información acerca de todos los objetos de todas las otras BBDD que mantiene nuestro servidor.
  - o Contiene muchas tablas de sólo lectura que en realidad son vistas, por lo que no se tiene acceso a las tablas base.
- Cada usuario tiene derecho a acceder a estas tablas, pero sólo a los registros que se corresponden con los objetos a los que tiene permiso de acceso.
- Manual: *22 La base de datos de información.*

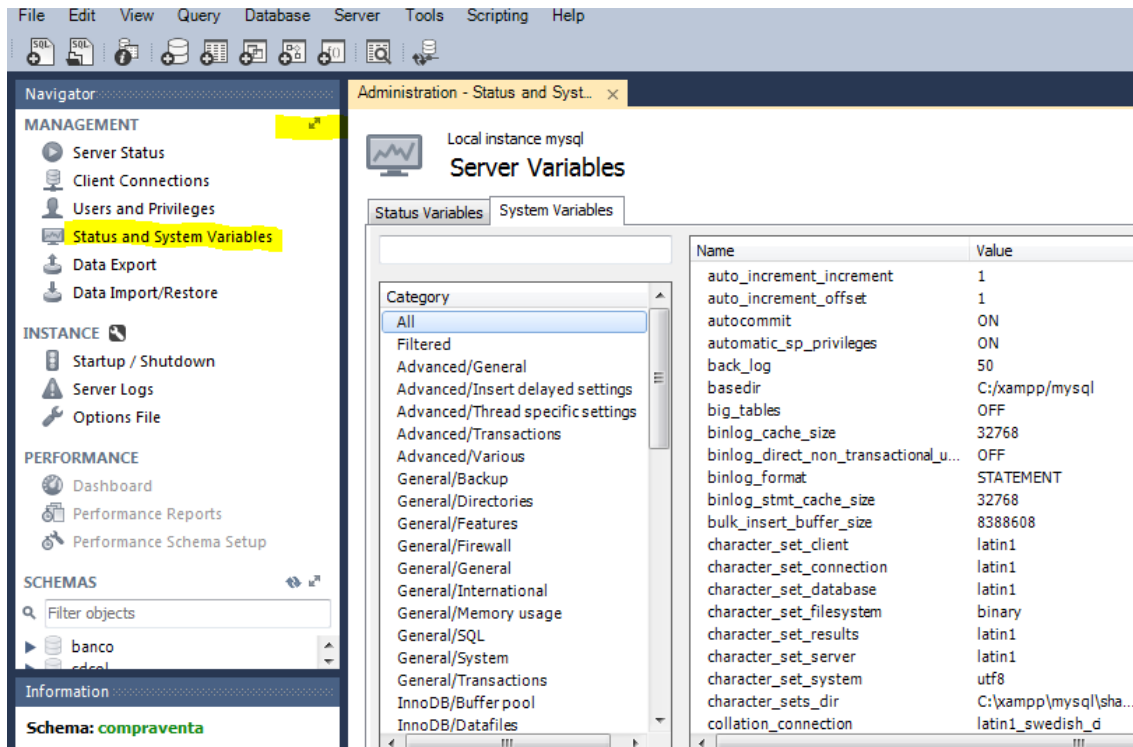
### Ejercicio 301.

1. Obtener información sobre todas las tablas almacenadas:
2. Idem sobre las tablas de compraventa:
3. Idem sobre las columnas de compraventa:
4. Idem sobre las columnas de la tabla Clientes.

5. Idem acerca de las restricciones establecidas en compraventa:
6. Idem acerca de las restricciones establecidas para la tabla Pedidos.

### 3. Variables del sistema

- El servidor MySQL mantiene muchas variables del sistema que indican cómo está configurado.
- Cada una tiene un valor por defecto y la mayoría pueden ser modificadas dinámicamente mientras el servidor está corriendo.
- El administrador puede ajustar los valores de las diferentes variables utilizando ficheros de opciones, incluyendo dichas opciones al arrancar el servidor o modificándolas con el comando SET.
- Manual: *5.3.3. Variables de sistema del servidor.*
- Podemos obtener un listado de las variables del sistema con su contenido mediante la instrucción **show variables;**
- Para ver el estado de las variables del sistema con Worckbench:



## 4. ficheros log.

- El servidor MySQL mantiene varios ficheros de registro diferentes que almacenan información sobre lo que está ocurriendo en el servidor en cada momento, se denominan genéricamente *ficheros LOG*.
- Los principales son:
  - o El registro de errores *ERROR LOG*: contiene información generada cuando se inicia y detiene el servidor, incluyendo los errores producidos mientras el servicio estuvo activo.
  - o El registro general de consultas: Se utiliza para registrar todas las consultas realizadas al servidor.
    - Sirve para obtener datos con fines estadísticos, permitiendo al administrador saber cuáles son las tablas más consultadas con objeto de optimizar su rendimiento.

- o El registro binario: Registra todas las sentencias que han actualizado datos, o podían haberlo hecho si hubieran encontrado filas concordantes, almacenándolas en forma de *eventos* que describen las modificaciones.
    - También contiene información sobre el tiempo que tardó en ejecutarse cada sentencia de actualización de datos.
  - o El registro *InnoDB*: Es un tipo especial de registro donde se almacenan las operaciones realizadas sobre tablas InnoDB.
- Manual: *5.10. Los ficheros de registro (log) de MySQL.*
  - Para ver su contenido con Worckbench:

The screenshot shows the MySQL Workbench interface. On the left is the 'Navigation' pane with categories: MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), STANCE (Startup / Shutdown, **Server Logs**, Options File), PERFORMANCE (Dashboard, Performance Reports, Performance Schema Setup), and REMAS (Filter objects). The main window is titled 'Administration - Server Logs' and shows 'Local instance mysql' with a warning icon. The 'Error Log File' tab is selected, and the 'InnoDB' radio button is chosen. A table displays the log entries:

| Timestamp           | Thread | Type    | Details                          |
|---------------------|--------|---------|----------------------------------|
| 2014-07-17 12:37:51 |        | InnoDB: | Shutdown completed; log s        |
| 2014-07-17 12:37:51 |        | [Note]  | C:\xampp\mysql\bin\mysql         |
| 2014-07-17 12:39:14 |        | [Note]  | Plugin 'FEDERATED' is disab      |
| 2014-07-17 12:39:15 |        | InnoDB: | The InnoDB memory heap is        |
| 2014-07-17 12:39:15 |        | InnoDB: | Mutexes and rw_locks use W       |
| 2014-07-17 12:39:15 |        | InnoDB: | Compressed tables use zlib       |
| 2014-07-17 12:39:15 |        | InnoDB: | Initializing buffer pool, size   |
| 2014-07-17 12:39:15 |        | InnoDB: | Completed initialization of b    |
| 2014-07-17 12:39:16 |        | InnoDB: | highest supported file form:     |
| 2014-07-17 12:39:21 |        | InnoDB: | Waiting for the background i     |
| 2014-07-17 12:39:22 |        | InnoDB: | 5.5.36 started; log sequenc      |
| 2014-07-17 12:39:22 |        | [Note]  | Server hostname (bind-addi       |
| 2014-07-17 12:39:22 |        | [Note]  | - '0.0.0.0' resolves to '0.0.0.0 |
| 2014-07-17 12:39:22 |        | [Note]  | Server socket created on IP:     |

## 5. Tipos de datos.

- MySQL proporciona una gran cantidad de tipos de datos que permiten ajustar el tipo de cada columna a su contenido previsto.
- Manual: [11. Tipos de columna.](#)

## 6.- Operadores.

En una consulta MySQL pueden utilizarse, entre otros, los siguientes operadores:

| Operador               | Significado   |
|------------------------|---|
| <, >, <=, >=, <>(!=)   | Operadores de relación  |
| NOT, AND, OR           | Operadores lógicos  |
| IS NULL                | Cierto si no devuelve nada  |
| BETWEEN ...<br>AND ... | Cierto si el valor del campo, de la variable o el resultado de la consulta está entre los 2 valores especificados.  |
| IN                     | Cierto si el valor del campo, de la variable o el resultado de la consulta está en la lista de valores que viene a continuación.  |
| LIKE                   | Cierto si el valor del campo, de la variable o el resultado de la consulta cumple la máscara que viene a continuación. (% coincidencia con 0 ó más caracteres. _ coincidencia con 1 carácter) |

- Manual: [12.1. Operadores](#)

## Ejercicio 302.

Con la BD Compraventa.

1. Nombre y precio de los productos cuyo nombre empieza por 'a' ó 'b'. Ídem de los que no empiezan por 'a' ni por 'b'.

a) Utilizando un operador de relación.

b) Utilizando el operador LIKE.

c) Utilizando el operador BETWEEN ... AND ...

2. Identificador de los pedidos que no tienen fecha de envío. Ídem de aquellos que sí la tienen.

3. Identificador de los pedidos que incluyen algún producto con identificador 1, 3, 8, 14, 19, 34 ó 49. Ídem que no lo incluyen.

4. Nombre de las compañías cliente que no han hecho ningún pedido.

a) Utilizando el operador IN.

b) Utilizando el operador EXISTS.

## 7. Funciones predefinidas.

- MySQL provee de gran cantidad de funciones predefinidas que pueden incluirse directamente en el lugar adecuado de cualquier sentencia.
- Manual: *12. Funciones y operadores.*
- Algunas de ellas son:

### Funciones de texto

| Función          | Finalidad  |
|------------------|--|
| LENGTH(columna)  | Devuelve la longitud de la cadena almacenada en la columna   |
| LEFT(columna,x)  | Devuelve los x primeros caracteres de una columna            |
| RIGHT(columna,x) | Devuelve los x últimos caracteres de una columna             |
| TRIM(columna)    | Quita los espacios extra al comienzo y al final de la cadena |

|                         |  |
|-------------------------|--|
| UPPER(columna)          | Convierte a mayúsculas la cadena               |
| LOWER(columna)          | Convierte a minúsculas la cadena               |
| SUBSTRING(columna,i ,l) | Devuelve l caracteres de columna a partir de i |
| CONCAT(col1, col2, ..)  | Permite realizar concatenaciones               |

### Funciones numéricas

| Función           | Finalidad   |
|-------------------|---|
| ABS(columna)      | Devuelve el valor absoluto                                  |
| CEILING(columna)  | Devuelve el siguiente entero                                |
| FLOOR(columna)    | Devuelve el valor entero                                    |
| FORMAT(columna,y) | El valor se devuelve con y decimales                        |
| MOD(x,y)          | Devuelve el resto de dividir x entre y                      |
| RAND()            | Devuelve un número aleatorio entre 0 y 1                    |
| ROUND(x,y)        | Devuelve el número x redondeado a y cifras decimales        |
| SIGN(columna)     | Devuelve un valor indicando si es negativo, cero o positivo |
| SQRT(columna)     | Devuelve la raíz cuadrada                                   |

### Funciones de fecha y hora

| Función                           | Finalidad                                    |
|-----------------------------------|--|
| HOURL(columna)                    | Devuelve el valor de la hora                 |
| MINUTE(columna)                   | Devuelve el valor de los minutos             |
| SECOND(columna)                   | Devuelve el valor de los segundos            |
| DAYNAME(columna)                  | Devuelve el nombre del día                   |
| DAYOFMONTH(columna)               | Devuelve el valor numérico del día           |
| MONTHNAME(columna)                | Devuelve el nombre del mes                   |
| MONTH(columna)                    | Devuelve el valor numérico del mes           |
| YEAR(columna)                     | Devuelve el valor del año                    |
| ADDDATE(columna, INTERVAL x tipo) | Añade x unidades a una columna               |
| SUBDATE(columna, INTERVAL x tipo) | Resta x unidades de una columna              |
| CURDATE()                         | Devuelve la fecha actual                     |
| CURTIME()                         | Devuelve la hora actual                      |
| NOW()                             | Devuelve la fecha y hora actuales            |
| UNIX_TIMESTAMP(fecha)             | Devuelve el número de segundos transcurridos |
| DATE_FORMAT(columna, formato)     | Modifica el formato de la fecha y hora       |
| TIME_FORMAT(columna, formato)     | Modifica el valor horario                    |

### Ejercicio 303.

1. Sentencia para visualizar la fecha actual en formato dd/mm/aaaa.´
2. Sentencia para visualizar el nombre y apellido de cada empleado en mayúsculas y su fecha de contratación en formato dd-mm-aa.



3. Sentencia para visualizar el identificador de cada pedido junto con el número de días transcurridos entre su fecha de envío y su fecha de entrega.

4. Sentencia que visualice el nombre de cada producto con una extensión máxima de 15 caracteres y su precio redondeado a 2 decimales.