

MySQL

Introducción

Crescencio Bravo



Contenidos

- **1 ¿Qué es MySQL?**
- **2 El SQL de MySQL**
- **3 Tutorial Básico de MySQL**
- **4 Manual de SQL**
- **5 Características avanzadas**
- **6 Herramientas básicas de administración**
 - **mysqladmin**
 - **mysql**
- **7 Resumen**
- **8 Ejercicios**

1 ¿Qué es MySQL? (i)

- **Un SGBD**
 - Relacional
 - De código abierto
- **Características generales:**
 - Es rápido
 - Fiable
 - Fácil de usar
 - Muy adecuado para aplicaciones en Internet
 - Soporta arquitectura C/S y sistemas *embebidos*
 - Hay disponible gran cantidad de software que soportan MySQL

1 ¿Qué es MySQL? (ii)

■ Principales vendedores y BD:

Microsoft	SQL Server 2000 EE http://www.microsoft.com/spain/servidores/sql
MySQL AB	MySQL Database Server 4 http://www.mysql.com/products/mysql/index.html
PostgreSQL	PostgreSQL 8 http://www.postgresql.com
Oracle	Oracle 9i EE Database http://otn.oracle.com/products/oracle9i/index.html
Sybase Inc	Adaptive Server Enterprise (ASE) 12.5.1 http://www.sybase.com/products/databaseservers/ase
IBM	Informix Standard Engine http://www-3.ibm.com/software/data/informix/se
IBM	DB2 8.1 http://www-5.ibm.com/es/software/db2

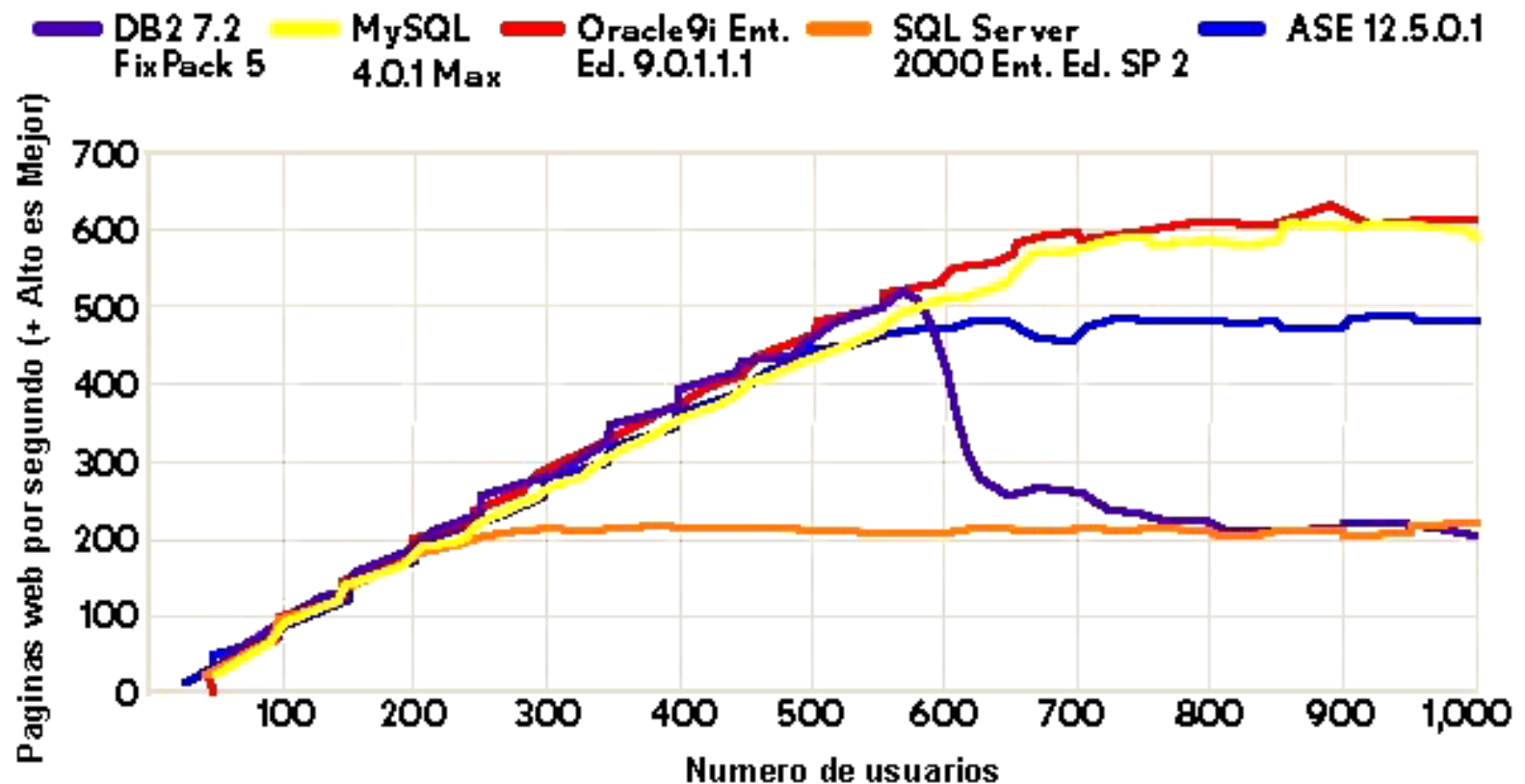
1 ¿Qué es MySQL? (iii)

- **BD Open Source:**

- MySQL es una base de datos Open Source (Código Abierto) y por lo tanto de coste gratuito
- **Código Abierto:** Movimiento que promulga el derecho de los usuarios a tener libre acceso al código fuente con el que se desarrolle cualquier aplicación, sistema operativo o librería
- Ejemplo más conocido: Linux
- Las bases de datos Open Source ya juegan en “Primera División”
- PostgreSQL es también un ejemplo de BD de código abierto

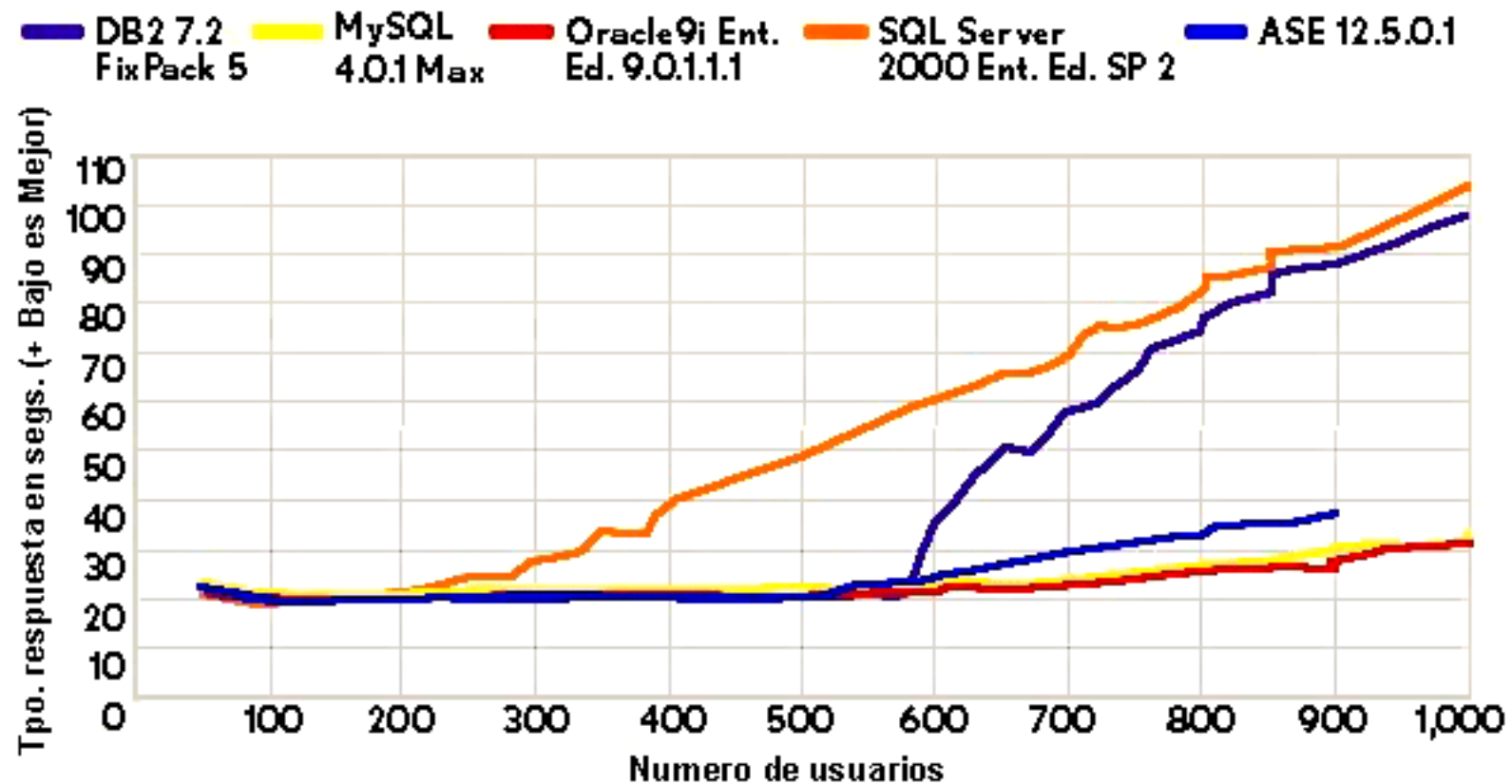
1 ¿Qué es MySQL? (iv)

■ Comparativas – Productividad:



1 ¿Qué es MySQL? (v)

■ Comparativas – Tiempo de respuesta:



1 ¿Qué es MySQL? (vi)

▪ En 2003-2004, MySQL...

- ✓ Velocidad/Rendimiento
- ✓ Bajo consumo
- ✓ Utilidades de administración
- ✓ Probabilidad reducida de corromper los datos
- ✓ Apache+PHP+MySQL en foros/buscadores de aplicaciones
- ✗ Carece de soporte para transacciones, rollbacks y subconsultas
- ✗ No maneja integridad referencial
- ✗ No es viable para grandes bases de datos con accesos continuos (mala escalabilidad)

1 ¿Qué es MySQL? (vii)

▪ Características detalladas (MySQL 5):

- C / C++ → Multiplataforma
- Soporta multi-hilo (hilos del kernel)
- APIs para muchos lenguajes (C++, Java, Perl, PHP, Python, etc.)
- Soporta transacciones
- Árboles B muy rápidos (compresión de índices)
- Sistema de gestión de memoria muy eficiente
- Joins optimizados
- Servidor separado para arquitecturas C/S o como librería *embebida*
- Muchos tipos de columnas; registros de longitud fija y variable
- Amplio abanico de sentencias y funciones
- Posibilidad de mezclar tablas de BD diferentes
- Sistema de privilegios y passwords
- Manejo de BD muy grandes
- 64 índices por tabla; índices de hasta 16 columnas; máxima longitud de índice de 1000 bytes
- Conectores ODBC y Java
- Soporte de internacionalización (mensajes, juego de caracteres y ordenación)

2 El SQL de MySQL

- **MySQL soporta ANSI SQL y otros modos de SQL**
- **Extensiones a SQL estándar:**
 - Los nombres de BD y tablas son sensibles a mayúsculas (de acuerdo al SO)
 - Las cadenas se pueden encerrar entre " o '
 - Se puede acceder a tablas de diferentes BD (`db.table`)
 - Tipos de campos: **MEDIUMINT, SET, ENUM, BLOB, TEXT**
 - Modificadores de tipos: **AUTO_INCREMENT, BINARY, NULL, UNSIGNED, ZEROFILL**
 - Se aceptan los operadores **||** y **&&**
 - Comparaciones entre cadenas: No sensibles a mayúsculas/minúsculas
 - **%** es sinónimo de **MOD()**
 - Cada BD se localiza en un directorio del directorio de datos
 - Cada tabla se localiza en un fichero
 - Muchas sentencias tienen una funcionalidad ampliada; existen nuevas funciones
 - Algunas otras se comportan de forma diferente

3 Tutorial Básico de MySQL (i)

▪ Conexión al servidor:

```
shell> mysql -h host -u user -p
Enter password: *****
```

```
shell> mysql
```

▪ Desconexión:

```
mysql> QUIT
Bye
```

▪ Introducción de consultas:

```
mysql> SELECT VERSION(), CURRENT_DATE;
+-----+-----+
| VERSION() | CURRENT_DATE |
+-----+-----+
| 3.22.20a-log | 1999-03-19 |
+-----+-----+
1 row in set (0.01 sec)
mysql>
```

```
mysql> SELECT
-> USER()
-> ,
-> CURRENT_DATE;
+-----+-----+
| USER() | CURRENT_DATE |
+-----+-----+
| jon@localhost | 2005-07-11 |
+-----+-----+
```

3 Tutorial Básico de MySQL (ii)

■ Utilización de una BD:

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| mysql    |
```

```
mysql> USE test
Database changed
```

```
shell> mysql -h host -u user -p menagerie
Enter password: *****
```

■ Creación de una BD:

```
mysql> CREATE DATABASE menagerie;
```

```
shell> mysql -h host -u user -p menagerie
Enter password: *****
```

■ Creación de una tabla:

```
mysql> CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20),
-> species VARCHAR(20), sex CHAR(1), birth DATE, death DATE);
```

```
mysql> SHOW TABLES;
+-----+
| Tables in menagerie |
+-----+
| pet                  |
+-----+
```

3 Tutorial Básico de MySQL (iii)

▪ Estructura de una tabla:

```
mysql> DESCRIBE pet;
```

Field	Type	Null	Key	Default	Extra
name	varchar(20)	YES		NULL	
owner	varchar(20)	YES		NULL	
species	varchar(20)	YES		NULL	
sex	char(1)	YES		NULL	
birth	date	YES		NULL	
death	date	YES		NULL	

▪ Recuperación de información:

```
SELECT what_to_select
FROM which_table
WHERE conditions_to_satisfy
```

```
mysql> SELECT * FROM pet;
```

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1993-02-04	NULL
Claws	Gwen	cat	m	1994-03-17	NULL
Buffy	Harold	dog	f	1989-05-13	NULL
Fang	Benny	dog	m	1990-08-27	NULL
Bowser	Diane	dog	m	1998-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	NULL
Whistler	Gwen	bird	NULL	1997-12-09	NULL
Slim	Benny	snake	m	1996-04-29	NULL
Puffball	Diane	hamster	f	1999-03-30	NULL

3 Tutorial Básico de MySQL (iv)

▪ Selección de determinadas filas:

```
mysql> SELECT * FROM pet WHERE (species = "cat" AND sex = "m")
-> OR (species = "dog" AND sex = "f");
```

name	owner	species	sex	birth	death
Claws	Gwen	cat	m	1994-03-17	NULL
Buffy	Harold	dog	f	1989-05-13	NULL

▪ Selección de columnas:

```
mysql> SELECT name, species, birth FROM pet
-> WHERE species = "dog" OR species = "cat";
```

name	species	birth
Fluffy	cat	1993-02-04
Claws	cat	1994-03-17

3 Tutorial Básico de MySQL (v)

■ Ordenación de filas:

```
mysql> SELECT name, birth FROM pet ORDER BY birth;
+-----+-----+
| name   | birth   |
+-----+-----+
| Buffy  | 1989-05-13 |
| Bowser | 1989-08-31 |
```

```
mysql> SELECT name, species, birth FROM pet ORDER BY species, birth DESC;
+-----+-----+-----+
| name   | species | birth   |
+-----+-----+-----+
| Chirpy | bird    | 1998-09-11 |
| Whistler | bird    | 1997-12-09 |
| Claws  | cat     | 1994-03-17 |
| Fluffy | cat     | 1993-02-04 |
| Fang   | dog     | 1990-08-27 |
```

■ Contar y agrupar:

```
mysql> SELECT species, COUNT(*) FROM pet GROUP BY species;
+-----+-----+
| species | COUNT(*) |
+-----+-----+
| bird    | 2 |
| cat     | 2 |
| dog     | 3 |
| hamster | 1 |
| snake   | 1 |
```


3 Tutorial Básico de MySQL (vi)

Operaciones con fechas:

```
mysql> SELECT name, birth, CURDATE(),
-> (YEAR(CURDATE())-YEAR(birth))
-> - (RIGHT(CURDATE(),5)<RIGHT(birth,5))
-> AS age
-> FROM pet;
```

```
mysql> SELECT name, birth FROM pet
-> WHERE MONTH(birth) = MOD(MONTH(CURDATE()), 12) + 1;
```

Información de bases de datos y tablas:

```
mysql> SELECT DATABASE();
+-----+
| DATABASE() |
+-----+
| menagerie  |
+-----+
```

```
mysql> SHOW TABLES;
+-----+
| Tables in menagerie |
+-----+
| event               |
| pet                  |
+-----+
```


3 Tutorial Básico de MySQL (vii)

- **Versión de MySQL:**

- `mysql -V`

- **Ejecución en modo *batch*:**

```
shell> mysql -h host -u user -p < batch-file  
Enter password: *****
```

4 Manual de SQL: Estructura del lenguaje (i)

▪ Literales:

- "cadena", 'cadena de caracteres'
- 123.45, 5e10
- Valores lógicos

```
mysql> SELECT TRUE, true, FALSE, false;
-> 1, 1, 0, 0
```

▪ Valor nulo: NULL

▪ Nombres:

	Length (bytes)	
Database	64	Any character that is allowed in a directory name, except '/', '\', or '.'
Table	64	Any character that is allowed in a filename, except '/', '\', or '.'
Column	64	All characters
Index	64	All characters
Alias	255	All characters

4 Manual de SQL: Estructura del lenguaje (ii)

■ Comentarios:

```
mysql> SELECT 1+1;      ¶ This comment continues to the end of line
mysql> SELECT 1+1;      -- This comment continues to the end of line
mysql> SELECT 1 /* this is an in-line comment */ + 1;
mysql> SELECT 1+
/*
this is a
multiple-line comment
*/
1;
```

4 Manual de SQL: Tipos de columnas

▪ Tipos de columnas de ANSI SQL:

Tipo de Datos	Longitud	Descripción
BINARY	1 byte	Para consultas sobre tabla adjunta de productos de bases de datos que definen un tipo de datos Binario.
BIT	1 byte	Valores Si/No ó True/False
BYTE	1 byte	Un valor entero entre 0 y 255.
COUNTER	4 bytes	Un número incrementado automáticamente (de tipo Long)
CURRENCY	8 bytes	Un entero escalable entre 922.337.203.685.477,5808 y 922.337.203.685.477,5807.
DATETIME	8 bytes	Un valor de fecha u hora entre los años 100 y 9999.
SINGLE	4 bytes	Un valor en punto flotante de precisión simple con un rango de $-3.402823 \cdot 10^{38}$ a $-1.401298 \cdot 10^{-45}$ para valores negativos, $1.401298 \cdot 10^{-45}$ a $3.402823 \cdot 10^{38}$ para valores positivos, y 0.
DOUBLE	8 bytes	Un valor en punto flotante de doble precisión con un rango de $-1.79769313486232 \cdot 10^{308}$ a $-4.94065645841247 \cdot 10^{-324}$ para valores negativos, $4.94065645841247 \cdot 10^{-324}$ a $1.79769313486232 \cdot 10^{308}$ para valores positivos, y 0.
SHORT	2 bytes	Un entero corto entre -32,768 y 32,767.
LONG	4 bytes	Un entero largo entre -2,147,483,648 y 2,147,483,647.
LONGTEXT	1 byte por carácter	De cero a un máximo de 1.2 gigabytes.
LOBINARY	Según se necesite	De cero 1 gigabyte. Utilizado para objetos OLE.
TEXT	1 byte por carácter	De cero a 255 caracteres.

4 Manual de SQL: Tipos Numéricos (i)

▪ Notación:

- **M**: Ancho de visualización máximo (nº total de dígitos decimales)
 - **D**: En tipos en coma flotante y de coma fija, el número de dígitos que siguen a la coma
 - **p**: Precisión en bits
 - **[]**: opcional
-
- **BIT[(M)]** : Campo de bits, M (1 a 64) indica el nº de bits, 1 por defecto
 - **TINYINT[(M)] [UNSIGNED] [ZEROFILL]** : -122 a 127 (unsigned: 0 a 255)
 - **BOOL, BOOLEAN** : Sinónimos para TINYINT(1). 0 es falso, cualquier otra cosa es verdadero
 - **SMALLINT[(M)] [UNSIGNED] [ZEROFILL]** : -32768 a 32767 (unsigned: 0 a 65535)
 - **MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL]** : -8388608 a 8388607 (unsigned: 0 a 16777215)
 - **INT[(M)] [UNSIGNED] [ZEROFILL]** : -2147483648 a 2147483647 (unsigned: 0 a 4294967295)
 - **INTEGER[(M)] [UNSIGNED] [ZEROFILL]** : Sinónimo para INT

4 Manual de SQL: Tipos Numéricos (ii)

- **INTEGER[(M)] [UNSIGNED] [ZEROFILL]** : Sinónimo para INT
- **BIGINT[(M)] [UNSIGNED] [ZEROFILL]** : -9223372036854775808 a 9223372036854775807 (unsigned: 0 a 18446744073709551615)
- **FLOAT[(M,D)] [UNSIGNED] [ZEROFILL]** : -3.402823466E+38 a -1.175494351E-38, 0 y 1.175494351E-38 a 3.402823466E+38. El n° aproximado de decimales es 7
- **DOUBLE[(M,D)] [UNSIGNED] [ZEROFILL]** : -1.7976931348623157E+308 a -2.2250738585072014E-308, 0 y 2.2250738585072014E-308 a 1.7976931348623157E+308. El n° aproximado de decimales es 15
- **DOUBLE PRECISION[(M,D)] [UNSIGNED] [ZEROFILL], REAL[(M,D)] [UNSIGNED] [ZEROFILL]** : Sinónimos para DOUBLE
- **FLOAT(p) [UNSIGNED] [ZEROFILL]**
- **DECIMAL[(M[,D])] [UNSIGNED] [ZEROFILL]** : Un n° en coma fija empaquetado. M≤65, D≤30. Por defecto, D=0, M=10.
- **DEC[(M[,D])] [UNSIGNED] [ZEROFILL], NUMERIC[(M[,D])] [UNSIGNED] [ZEROFILL], FIXED[(M[,D])] [UNSIGNED] [ZEROFILL]** : Sinónimos para DECIMAL

4 Manual de SQL: Tipos Numéricos (iii)

▪ Necesidades de almacenamiento:

Column Type	Storage Required
TINYINT	1 byte
SMALLINT	2 bytes
MEDIUMINT	3 bytes
INT, INTEGER	4 bytes
BIGINT	8 bytes
FLOAT (p)	4 bytes if $0 \leq p \leq 24$, 8 bytes if $25 \leq p \leq 53$
FLOAT	4 bytes
DOUBLE [PRECISION], item REAL	8 bytes
DECIMAL (M, D), NUMERIC (M, D)	Varies; see following discussion
BIT (M)	approximately $(M+7)/8$ bytes

4 Manual de SQL: Tipos Fecha y Hora

- **DATE:** Fecha, de '1000-01-01' a '9999-12-31'. MySQL muestra las fechas en el formato 'YYYY-MM-DD'
 - **DATETIME:** Fecha y hora, de '1000-01-01 00:00:00' a '9999-12-31 23:59:59', en formato 'YYYY-MM-DD HH:MM:SS'
 - **TIMESTAMP[(M)]:** Un sello temporal, de '1970-01-01 00:00:00' hasta el año 2037, en formato 'YYYY-MM-DD HH:MM:SS'
 - Útil para recoger la fecha y hora de una operación INSERT o UPDATE
 - **TIME:** Hora, de '-838:59:59' to '838:59:59', en formato 'HH:MM:SS'
 - **YEAR[(2|4)]:** Año en formato de 2 ó 4 dígitos, de 1901 a 2155, incluido el año 0, en formato 'YYYY'
- **Necesidades de almacenamiento:**

Column Type	Storage Required
DATE	3 bytes
DATETIME	8 bytes
TIMESTAMP	4 bytes
TIME	3 bytes
YEAR	1 byte

4 Manual de SQL: Tipos Cadena (i)

- **Notación:**
 - **M:** nº de caracteres
- **[NATIONAL] CHAR(M) [BINARY | ASCII | UNICODE]** : Cadena de longitud fija que se completa con espacios en blanco por la derecha; el rango de M es de 0 a 255
- **[NATIONAL] VARCHAR(M) [BINARY]** : Cadena de longitud variable; el rango de M es de 0 a 65535
- **BINARY** y **VARBINARY** son similares pero contienen cadenas binarias, en lugar de cadenas de caracteres
- **BLOB[(M)]** : Objeto binario con una longitud máxima de 65535 bytes.
 - Permite cuatro subtipos: TINYBLOB, BLOB, MEDIUMBLOB y LONGBLOB
- **TEXT[(M)]** : Secuencia de caracteres con una longitud máxima de 65535 caracteres
 - Permite cuatro subtipos: TINYTEXT, TEXT, MEDIUMTEXT y LONGTEXT
- **ENUM('value1','value2',...)** : Una enumeración. Un objeto de tipo cadena que puede tener un único valor elegido de la lista de valores.
- **SET('value1','value2',...)** : Un conjunto. Un objeto cadena que puede tener 0 o más valores, cada uno elegido de la lista de valores.

4 Manual de SQL: Tipos Cadena (ii)

■ Necesidades de almacenamiento:

Column Type	Storage Required
CHAR (<i>M</i>)	<i>M</i> bytes, $0 \leq M \leq 255$
VARCHAR (<i>M</i>)	<i>L</i> +1 bytes, where $L \leq M$ and $0 \leq M \leq 65535$ (see note below). This was $0 \leq M \leq 255$ before MySQL 5.0.3.
BINARY (<i>M</i>)	<i>M</i> bytes, $0 \leq M \leq 255$
VARBINARY (<i>M</i>)	<i>L</i> +1 bytes, where $L \leq M$ and $0 \leq M \leq 255$
TINYBLOB, TINYTEXT	<i>L</i> +1 byte, where $L < 2^8$
BLOB, TEXT	<i>L</i> +2 bytes, where $L < 2^{16}$
MEDIUMBLOB, MEDIUMTEXT	<i>L</i> +3 bytes, where $L < 2^{24}$
LONGBLOB, LONGTEXT	<i>L</i> +4 bytes, where $L < 2^{32}$
' , ... ENUM ('value1', 'value2')	1 or 2 bytes, depending on the number of enumeration values (65,535 values maximum)
SET ('value1', 'value2', ...)	1, 2, 3, 4, or 8 bytes, depending on the number of set members (64 members maximum)

4 Manual de SQL: Compatibilidad con otros Tipos

▪ Equivalencias con otras BD:

Other Vendor Type	MySQL Type
BOOL,	TINYINT
BOOLEAN	TINYINT
CHAR VARYING (M)	VARCHAR (M)
DEC	DECIMAL
FIXED	DECIMAL
FLOAT4	FLOAT
FLOAT8	DOUBLE
INT1	TINYINT
INT2	SMALLINT
INT3	MEDIUMINT
INT4	INT
INT8	BIGINT
LONG VARBINARY	MEDIUMBLOB
LONG VARCHAR	MEDIUMTEXT
LONG	MEDIUMTEXT
MIDDLEINT	MEDIUMINT
NUMERIC	DECIMAL

4 Manual de SQL: Funciones (i)

- **Operadores relacionales:**

- **>, <, >=, <=, <>, !=,**
- **IS NULL, IS NOT NULL**
 - ISNULL(expr)
- **expr BETWEEN min AND max,**
 - expr NOT BETWEEN min AND max
- ...

- **Operadores lógicos:**

- **NOT, !, OR, ||, AND, &&**

- **Funciones de control del flujo:**

- **IF(expr1, expr2, expr3)**
- ...

4 Manual de SQL: Funciones (ii)

■ Funciones de cadenas de caracteres:

- **CONCAT(str1, str2, ...)**
- **LENGTH(str)**
- **LEFT(str, len)**
- **RIGHT(str, len)**
- **SUBSTRING(str, pos, len)**
- **UCASE(str), UPPER(str), LCASE(str), RCASE(str)**
- ...

- **expr LIKE pat**
 - *pat* es el patrón con el que comparar; puede contener dos tipos de comodines:
 - % coincide con cualquier nº de caracteres, incluido cero caracteres
 - _ coincide exactamente con un caracter

4 Manual de SQL: Funciones (iii)

- **Operaciones aritméticas:**

- $+$, $-$, $*$, $/$

- **Funciones matemáticas:**

- **ABS(x)**, **MOD(n, m)**, **SQRT(x)**, **PI()**, ...

- **Otras funciones:**

- **DATABASE()**, **USER()**, **VERSION()**, ...

```
mysql> SELECT VERSION();  
-> '3.23.13-log'
```

- **Funciones de agregado:**

- **AVG**: Calculo promedio de los valores de una columna
 - **COUNT**: N° de filas de la selección
 - **SUM**: Suma de los valores de una columna
 - **MAX, MIN**: Valor más alto y más bajo de una columna

4 Manual de SQL: Manipulación de Datos (i)

▪ SELECT

```

SELECT
  [ALL | DISTINCT | DISTINCTROW ]
  [HIGH_PRIORITY]
  [STRAIGHT JOIN]
  [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
  [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
  select_expr, ...
  [INTO OUTFILE 'file_name' export_options
   | INTO DUMPFILE 'file_name']
  [FROM table_references
  [WHERE where_definition]
  [GROUP BY {col_name | expr | position}
   [ASC | DESC], ... [WITH ROLLUP]]
  [HAVING where_definition]
  [ORDER BY {col_name | expr | position}
   [ASC | DESC], ...]
  [LIMIT {[offset,] row_count | row_count OFFSET offset}]
  [PROCEDURE procedure_name(argument_list)]
  [FOR UPDATE | LOCK IN SHARE MODE]]

```

- ***select_expression*** indica las columnas que se quieren recuperar
 - Se puede incluir cualquier expresión válida

4 Manual de SQL: Manipulación de Datos (ii)

■ INSERT

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
[INTO] tbl_name [(col_name,...)]
VALUES ({expr | DEFAULT},...), (...), ...
[ ON DUPLICATE KEY UPDATE col_name=expr, ... ]
```

Or:

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
[INTO] tbl_name
SET col_name={expr | DEFAULT}, ...
[ ON DUPLICATE KEY UPDATE col_name=expr, ... ]
```

Or:

```
INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE]
[INTO] tbl_name [(col_name,...)]
SELECT ...
[ ON DUPLICATE KEY UPDATE col_name=expr, ... ]
```


4 Manual de SQL: Manipulación de Datos (iii)

▪ UPDATE

```
UPDATE [LOW_PRIORITY] [IGNORE] tbl_name
  SET col_name1=expr1 [, col_name2=expr2 ...]
  [WHERE where_definition]
  [ORDER BY ...]
  [LIMIT row_count]
```

▪ DELETE

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM tbl_name
  [WHERE where_definition]
  [ORDER BY ...]
  [LIMIT row_count]
```

Multiple-table syntax:

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
  tbl_name[.*] [, tbl_name[.*] ...]
  FROM table_references
  [WHERE where_definition]
```

4 Manual de SQL: Definición de Datos (i)

■ Crear y eliminar BD:

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name
    [create_specification [, create_specification] ...]
```

```
DROP {DATABASE | SCHEMA} [IF EXISTS] db_name
```

■ Creación de tablas:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    [(create_definition,...)]
    [table_options] [select_statement]
```

Or:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    [( ) LIKE old_tbl_name ( )];
```

```
create_definition:
    column_definition
    | [CONSTRAINT [symbol]] PRIMARY KEY [index_type] (index_col_name,...)
    | KEY [index_name] [index_type] (index_col_name,...)
    | INDEX [index_name] [index_type] (index_col_name,...)
    | [CONSTRAINT [symbol]] UNIQUE [INDEX]
      [index_name] [index_type] (index_col_name,...)
    | [FULLTEXT|SPATIAL] [INDEX] [index_name] (index_col_name,...)
    | [CONSTRAINT [symbol]] FOREIGN KEY
      [index_name] (index_col_name,...) [reference_definition]
    | CHECK (expr)
```

4 Manual de SQL: Definición de Datos (ii)

column definition:

```
col_name type [NOT NULL | NULL] [DEFAULT default_value]
[AUTO_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY]
[COMMENT 'string'] [reference_definition]
```

type:

```
TINYINT[(length)] [UNSIGNED] [ZEROFILL]
SMALLINT[(length)] [UNSIGNED] [ZEROFILL]
MEDIUMINT[(length)] [UNSIGNED] [ZEROFILL]
INT[(length)] [UNSIGNED] [ZEROFILL]
INTEGER[(length)] [UNSIGNED] [ZEROFILL]
BIGINT[(length)] [UNSIGNED] [ZEROFILL]
REAL[(length,decimals)] [UNSIGNED] [ZEROFILL]
DOUBLE[(length,decimals)] [UNSIGNED] [ZEROFILL]
FLOAT[(length,decimals)] [UNSIGNED] [ZEROFILL]
DECIMAL(length,decimals) [UNSIGNED] [ZEROFILL]
NUMERIC(length,decimals) [UNSIGNED] [ZEROFILL]
DATE
TIME
TIMESTAMP
DATETIME
CHAR(length) [BINARY | ASCII | UNICODE]
VARCHAR(length) [BINARY]
TINYBLOB
BLOB
MEDIUMBLOB
LONGBLOB
TINYTEXT [BINARY]
TEXT [BINARY]
MEDIUMTEXT [BINARY]
LONGTEXT [BINARY]
ENUM(value1,value2,value3,...)
SET(value1,value2,value3,...)
spatial_type
```

4 Manual de SQL: Definición de Datos (iii)

```
index_col_name:
    col_name [(length)] [ASC | DESC]
```

```
reference definition:
    REFERENCES tbl_name [(index_col_name,...)]
        [MATCH FULL | MATCH PARTIAL | MATCH SIMPLE]
        [ON DELETE reference_option]
        [ON UPDATE reference_option]
```

```
reference_option:
    RESTRICT | CASCADE | SET NULL | NO ACTION
```

```
table_options: table_option [table_option] ...
```

```
table_option:
    {ENGINE|TYPE} = engine_name
    AUTO_INCREMENT = value
    AVG_ROW_LENGTH = value
    [DEFAULT] CHARACTER SET charset_name [COLLATE collation_name]
    CHECKSUM = {0 | 1}
    COMMENT = 'string'
    CONNECTION = 'connect_string'
    MAX_ROWS = value
    MIN_ROWS = value
    PACK_KEYS = {0 | 1 | DEFAULT}
    PASSWORD = 'string'
    DELAY_KEY_WRITE = {0 | 1}
    ROW_FORMAT = {DEFAULT|DYNAMIC|FIXED|COMPRESSED|REDUNDANT|COMPACT}
    UNION = (tbl_name[,tbl_name]...)
    INSERT_METHOD = { NO | FIRST | LAST }
    DATA DIRECTORY = 'absolute path to directory'
    INDEX DIRECTORY = 'absolute path to directory'
```

```
select_statement:
    [IGNORE | REPLACE] [AS] SELECT ... (Some legal select statement)
```

4 Manual de SQL: Definición de Datos (iv)

▪ Modificación de tablas:

```

ALTER [IGNORE] TABLE tbl_name
    alter_specification [, alter_specification] ...

alter_specification:
    ADD [COLUMN] column_definition [FIRST | AFTER col_name ]
    ADD [COLUMN] (column_definition,...)
    ADD INDEX [index_name] [index_type] (index_col_name,...)
    ADD [CONSTRAINT [symbol]]
        PRIMARY KEY [index_type] (index_col_name,...)
    ADD [CONSTRAINT [symbol]]
        UNIQUE [index_name] [index_type] (index_col_name,...)
    ADD [FULLTEXT|SPATIAL] [index_name] (index_col_name,...)
    ADD [CONSTRAINT [symbol]]
        FOREIGN KEY [index_name] (index_col_name,...)
            [reference_definition]
    ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}
    CHANGE [COLUMN] old_col_name column_definition
        [FIRST|AFTER col_name]
    MODIFY [COLUMN] column_definition [FIRST | AFTER col_name]
    DROP [COLUMN] col_name
    DROP PRIMARY KEY
    DROP INDEX index_name
    DROP FOREIGN KEY fk_symbol
    DISABLE KEYS
    ENABLE KEYS
    RENAME [TO] new_tbl_name
    ORDER BY col_name
    CONVERT TO CHARACTER SET charset_name [COLLATE collation_name]
    [DEFAULT] CHARACTER SET charset_name [COLLATE collation_name]
    DISCARD TABLESPACE
    IMPORT TABLESPACE
    table_options
  
```

4 Manual de SQL: Definición de Datos (v)

▪ Cambio del nombre de una tabla:

```
RENAME TABLE tbl_name TO new_tbl_name  
[, tbl_name2 TO new_tbl_name2] ...
```

▪ Eliminación de tablas:

```
DROP [TEMPORARY] TABLE [IF EXISTS]  
tbl_name [, tbl_name] ...  
[RESTRICT | CASCADE]
```

▪ Creación de índice:

```
CREATE [UNIQUE|FULLTEXT|SPATIAL] INDEX index_name  
[USING index_type]  
ON tbl_name (index_col_name,...)
```

```
index_col_name:  
col_name [(length)] [ASC | DESC]
```

▪ Eliminación de índice:

```
DROP INDEX index_name ON tbl_name
```


5 Características Avanzadas

- **Utilidad para chequeo, optimización y reparación de tablas**
- **DELETE, INSERT, REPLACE y UPDATE devuelven el número de filas afectadas**
- **Todos los programas/utilidades permiten la obtención de ayuda en línea (--help o -?)**
- **El comando SHOW muestra información sobre BD, tablas e índices**

6 Herramientas de Administración: **mysqladmin** (i)

- **mysqladmin**: Una utilidad para realizar operaciones administrativas

```
shell> mysqladmin [OPTIONS] command [command-option] command ...
```

```
mysqladmin --help
```

- **Operaciones más habituales:**

- **Crear BD:**

- `mysqladmin create <nombreBD>`

- **Eliminar BD:**

- `mysqladmin drop <nombreBD>`

- **Ver el estado de MySQL:**

- `mysqladmin ping`

- `mysqladmin status`

- **Detener el servidor:**

- `mysqladmin shutdown`

- **Puede ser necesario autenticarse:**

- `mysqladmin create <nombreBD> -u root -p`

6 Herramientas de Administración: mysqladmin (ii)

▪ Todas las operaciones:

create databasename

Create a new database.

drop databasename

Delete a database and all its tables.

extended-status

Gives an extended status message from the server.

flush-hosts

Flush all cached hosts.

flush-logs

Flush all logs.

flush-tables

Flush all tables.

flush-privileges

Reload grant tables (same as reload).

kill id,id,...

Kill mysql threads.

password Set a new password. Change old password to new-password.

ping Check if mysqld is alive.

processlist

Show list of active threads in server.

reload Reload grant tables.

refresh Flush all tables and close and open logfiles.

shutdown Take server down.

slave-start

Start slave replication thread.

slave-stop

Stop slave replication thread.

status Gives a short status message from the server.

variables

Prints variables available.

version Get version info from server.

6 Herramientas de Administración: mysql (i)

- **mysql**: Una consola para ejecutar comandos de SQL
- Permite un uso interactivo y no interactivo

```
shell> mysql database < script.sql > output.tab
```

- **Comandos más habituales:**

- **help, ?**: Ayuda
- **exit, quit**: Salir
- **use <nombreBD>**: Abrir otra BD
- **status**: Estado del servidor
- **source <nombreFich>**: Ejecuta un fichero con comandos SQL

6 Herramientas de Administración: mysql (ii)

▪ Todos los comandos:

help	(\h)	Display this text.
?	(\h)	Synonym for 'help'.
clear	(\c)	Clear command.
connect	(\r)	Reconnect to the server. Optional arguments are db and host.
edit	(\e)	Edit command with \$EDITOR.
ego	(\G)	Send command to mysql server, display result vertically.
exit	(\q)	Exit mysql. Same as quit.
go	(\g)	Send command to mysql server.
nopager	(\n)	Disable pager, print to stdout.
notee	(\t)	Don't write into outfile.
pager	(\P)	Set PAGER [to_pager]. Print the query results via PAGER.
print	(\p)	Print current command.
prompt	(\R)	Change your mysql prompt.
quit	(\q)	Quit mysql.

rehash	(\#)	Rebuild completion hash.
source	(\.)	Execute a SQL script file. Takes a file name as an argument.
status	(\s)	Get status information from the server.
tee	(\T)	Set outfile [to_outfile]. Append everything into given outfile.
use	(\u)	Use another database. Takes database name as argument.

7 Resumen

- **Qué hemos aprendido...**
 - **Sabemos qué es MySQL**
 - **Hemos repasado sus características principales y hemos visto que soporta un SQL extendido**
 - **Hemos realizado un pequeño tutorial de MySQL**
 - **Hemos repasado el SQL que “entiende” MySQL**
 - **Hemos practicado con dos de sus herramientas principales de consola**

8 Ejercicios

- **Con *mysql*:**
 - 1 Crear una BD alumnos con tres tablas: alumno, asignatura, nota
 - 2 Crear una BD personal con la lista de dvds y las cuentas de la casa
 - 3 Una vez creadas, mostrar todas las BD con el comando adecuado
 - 4 Mostrar todas las tablas de cada BD
 - 5 Ver la estructura de las tablas
 - 6 Crear un fichero de texto con extensión .sql que contenga instrucciones SQL para insertar algunos datos de ejemplo y ejecutarlo
 - 7 Practicar con diferentes *select* y *update* desde la línea de órdenes de *mysql*
- **Con *mysqladmin*:**
 - 8 Estudiar la forma equivalente de crear estas dos BD