Conceptos de MySQL

Mayo 2005

Jordi Llonch
jordi@laigu.net

SOME RIGHTS RESERVED

http://creativecommons.org/licenses/by-sa/2.0/

MySQL

- MySQL es un motor de base de datos SQL:
  - Muy rápido.
  - Multi-threat.
  - Multiusuario.
- Disponemos de dos tipos de licéncia:
  - Open Source/Software libre (GNU General Public License).
  - Standard commercial license.

Motores de almacenaje y tipos de tablas (1)

- ISAM
  - Es el motor original. En *deprecated* a partir de la versión 4.1.
- MyISAM
  - Reemplazo de ISAM.
  - Tipo de tabla por defecto.
  - No soporta transacciones.
- HEAP
  - Tablas en memoria.
  - Actualmente se conoce como MEMORY.
  - No soporta transacciones.
- MERGE
  - Permite que una colección de tablas MyISAM sean tratadas como una única tabla.
  - No soporta transacciones.

Motores de almacenaje y tipos de tablas (2)

- BDB (Berkeley DB)
  - Soporta transacciones.
  - No funciona en todas las plataformas.
- InnoDB
  - Soporta transacciones (ACID compliant).
  - Bloqueo a nivel de fila (incrementa la capacidad de concurrencia y rendimiento).
  - Se puede mezclar InnoDB con otros tipos de tablas en la misma bbdd.
- NDB Cluster
  - Soporta tablas que se encuentran particionadas en diferentes ordenadores.

Tipos de campos (1)

- Tipos numéricos:
  - TINYINT[(M)] [UNSIGNED] [ZEROFILL]
    - Entero muy pequeño (-128 a 127 con signo / 0 a 255 sin signo).
  - BIT/BOOL/BOOLEAN
    - Sinónimo de TINYINT(1).
    - Añadido en MySQL 4.1.0.
    - Cero es falso. Valores no zero son considerados verdaderos.
  - SMALLINT[(M)] [UNSIGNED] [ZEROFILL]
    - Entero pequeño (-32768 a 32767 con signo / 0 a 65535 sin signo).
  - MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL]
    - Entero medio (-8388608 a 8388607 con signo / 0 a 16777215 sin signo).
  - INT[(M)] [UNSIGNED] [ZEROFILL]
    - Entero (-2147483648 a 2147483647 con signo / 0 to 4294967295 sin signo).
  - INTEGER[(M)] [UNSIGNED] [ZEROFILL]
    - Sinónimo de INT.

Tipos de campos (2)

- Tipos numéricos:
  - BIGINT[(M)] [UNSIGNED] [ZEROFILL]
    - Entero grande (-9223372036854775808 a 9223372036854775807 con signo / 0 to 18446744073709551615 sin signo)
  - FLOAT[(M,D)] [UNSIGNED] [ZEROFILL]
    - Número coma flotante pequeño.
    - -3.402823466E+38 a -1.175494351E-38, 0, y 1.175494351E-38 a 3.402823466E+38
  - DOUBLE[(M,D)] [UNSIGNED] [ZEROFILL]
    - Número coma flotante normal.
    - -1.7976931348623157E+308 a -2.2250738585072014E-308, 0, y 2.2250738585072014E-308 a 1.7976931348623157E+308
  - DOUBLE PRECISION[(M,D)] [UNSIGNED] [ZEROFILL]
  - REAL[(M,D)] [UNSIGNED] [ZEROFILL]
    - Sinónimos de DOUBLE.

Tipos de campos (3)

- Tipos numéricos:
  - DECIMAL[(M[,D])] [UNSIGNED] [ZEROFILL]
    - Número con decimal fijo.
    - Se comporta com un CHAR.
    - M es el total de dígitos y D es el número de decimales.
  - DEC[(M[,D])] [UNSIGNED] [ZEROFILL]
  - NUMERIC[(M[,D])] [UNSIGNED] [ZEROFILL]
  - FIXED[(M[,D])] [UNSIGNED] [ZEROFILL]
    - Sinónimos de DECIMAL

Tipos de campos (4)

- Tipos fecha y hora:
  - DATE
    - Fecha. Rango de '1000-01-01' a '9999-12-31'.
    - Formato: 'YYYY-MM-DD'.
  - DATETIME
    - Combinación de fecha y hora. Rango de '1000-01-01 00:00:00' a '9999-12-31 23:59:59'.
    - Formato: 'YYYY-MM-DD HH:MM:SS'.
  - TIMESTAMP[(M)]
    - Rango de '1970-01-01 00:00:00' a aproximadamente el año 2037.
  - TIME
    - Tiempo. Rango de '-838:59:59' a '838:59:59'.
    - Formato: 'HH:MM:SS'.
  - YEAR[(2|4)]
    - Año en 2 dígitos o 4 dígitos. Rango de 1901 to 2155, and 0000 para 4 dígitos y de 70 a 69 (de 1970 a 2069) para 2 dígitos.

Tipos de campos (5)

- Tipos string:
  - [NATIONAL] CHAR(M) [BINARY | ASCII | UNICODE]
    - Cadena de carácteres de longitud fija.
    - M de 0 a 255.
  - CHAR
    - Sinónimo de CHAR(1).
  - [NATIONAL] VARCHAR(M) [BINARY]
    - Cadena de carácteres de longitud variable.
    - M de 0 a 255.
  - TINYBLOB/TINYTEXT
    - Una columna BLOB o TEXT de como máximo 255 carácteres.
  - BLOB/TEXT
    - Una columna BLOB o TEXT de como máximo 65,535 carácteres.
  - MEDIUMBLOB/MEDIUMTEXT
    - Una columna BLOB o TEXT de como máximo 16,777,215 carácteres.

Tipos de campos (6)

- Tipos string:
  - LONGBLOB/LONGTEXT
    - Una columna BLOB o TEXT de como máximo 4,294,967,295 carácteres.
  - ENUM('value1','value2',...)
    - Enumeración.
    - Cadena de carácteres que sólo puede tener un valor dado en la lista.
  - SET('value1','value2',...)
    - Lista que puede tener valor cero o más valores. Deben estar en la lista y como máximo pueden haber 64 miembros.

Chequear y reparar tablas (1)

- Algunas veces las tablas tipo MyISAM contienen errores.
- Los síntomas son consultas que abortan inesperadamente y observación de errores como:
  - `tbl_name.frm' is locked against change
  - Can't find file `tbl_name.MYI' (Errcode: ###)
  - Unexpected end of file
  - Record file is crashed
  - Got error ### from table handler

Chequear y reparar tablas (2)

- Chequeo de una tabla:
  - Usaremos el comando *myisamchk*:
    - myisamchk tbl_name
      - Encontrará el 99.99% de los errores. No busca corrupción en los datos (es muy inusual)
    - myisamchk -m tbl_name
      - Encontrará el 99.999% de los errores
    - myisamchk -e tbl_name
      - Chequeo completo.
      - Muy lento para tablas grandes.

Chequear y reparar tablas (3)

- Reparación de una tabla:
  - Debemos parar el servidor MySQL antes de reparar las tablas.
  - Usaremos el comando *myisamchk*:
    - Primero probaremos *myisamchk -r -q tbl_name* (“quick recovery mode”)
      - Intentará reparar el archivo índice sin tocar el archivo de datos.
    - Si no funciona el siguiente paso será:
      - Hacer backup de los archivos de datos.
      - Usar *myisamchk -r tbl_name* (“recovery mode”).
        - Borrará registros incorrectos y borrados del archivo de datos y reconstruirá el archivo índice.
    - El siguiente paso en caso que no funcione el precedente es:
      - Usar *myisamchk --safe-recover tbl_name*
        - Método antiguo de recuperación que en algunos casos recupera lo que le modo normal no puede.
        - Es muy lento.
    - (aún existen más pasos, consultar al manual)

Backup (1)

- Se dispone de dos opciones:
  - Volcar datos (a nivel de SQL o en *csv*)
  - Copiando los archivos del directorio de la base de datos.
- Backups de como mínimo una tabla.
- Antes de iniciar el proceso se debe bloquear la escritura en las tablas y vaciar las cachés.
  - LOCK TABLE
  - FLUSH TABLES

Backup (2)

- Backup volcando datos:
  1) CSV (sólo datos):
     - mysql> **SELECT INTO ... OUTFILE**
       - **Restaurar** con: mysql> *LOAD DATA INFILE 'data.txt' INTO TABLE db2.my_table*
  2) SQL (datos y/o estructura):
     - shell> **mysqldump**
       - mysqldump [options] db_name [tables]
       - mysqldump [options] --databases DB1 [DB2 DB3...]
       - mysqldump [options] –all-databases
         - » Ex: mysqldump --all-databases > all_databases.sql
       - **Restaurar** con: shell> *mysql database_name < database.sql*

Backup (3)

- Backup de archivos del directorio de base de datos:
  - SOLO PARA LOS TIPOS **MYISAM** E **ISAM** (NUNCA InnoDB)
  -

1) Copiar los archivos: *.frm, *.MYD, *.MYI

2) mysql> **BACKUP TABLE** tbl_name [, tbl_name] ... TO '/path/to/backup/directory'

3) Al restaurar se debería parar el servidor MySQL