# MySQL

www.mysql.com

ULB

# MySQL AB

- MySQL AB : swedish company of the MySQL founders and main developers: David Axmark, Allan Larsson, and Michael "Monty" Widenius.
- Develop the MySQL database software, promoting it to new users, sell support and certifications
- MySQL AB owns copyright to the source code, the MySQL logo and (registered) trademark
- "AB" is the acronym for the Swedish "aktiebolag," or "stock company." It translates to "MySQL, Inc."



2

ULB

# MySQL AB Goals

- To be the best and the most widely used database in the world
- To be available and affordable by all
- To be easy to use
- To be continuously improved while remaining fast and safe
- To be fun to use and improve
- To be free from bugs



3

ULB

# MySQL AB Core Values

- We subscribe to the Open Source philosophy and support the Open Source community
- We aim to be good citizens
- We prefer partners that share our values and mindset
- We answer email and provide support
- We are a virtual company, networking with others
- We work against software patents



4

ULB

# MySQL
# What it is

- MySQL is a relational database management system.
- MySQL software is Open Source: GPL, but to embed MySQL code into a commercial application:a commercially licensed version
- The MySQL Database Server is very fast, reliable, and easy to use
- Client/server or embedded systems
- Google, Yahoo, YouTube, Flickr, Second Life, Wikipedia, Craigslist, Slashdot, Blogger.com, Facebook, LiveJournal, Digg, Del.icio.us and Weather.com



5

ULB

# Internals

- Written in C and C++
- Works on many different platforms
- APIs for C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, and Tcl are available
- Fully multi-threaded using kernel threads. It can easily use multiple CPUs if they are available



6

ULB

# Data Types

- Many data types:

  signed/unsigned integers 1, 2, 3, 4, and 8 bytes long, FLOAT, DOUBLE

  CHAR, VARCHAR, TEXT, BLOB

  DATE, TIME, DATETIME, TIMESTAMP, YEAR

  SET, ENUM, and OpenGIS spatial types

- Fixed-length and variable-length records



7

ULB

# Connectivity

- Clients can connect to the MySQL server using TCP/IP sockets on any platform
- Connector/ODBC (MyODBC): MS Access to MySQL server, clients can run on Windows or Unix
- Java: JDBC
- Connector/NET: .NET application, implements ADO.NET interfaces and integrates into ADO.NET aware tools, any .NET languages, fully managed ADO.NET driver written in 100% pure C#



8

ULB

# Storage Engines

- Storage engines: how and when data is stored
- MySQL storage engines:
  - MyISAM: non transactional tables
  - InnoDB: transactional engine
  - MEMORY (aka HEAP): in-memory tables



9

ULB

# Using Storage Engines

- Specify storage engine: `ENGINE` or `TYPE` (old) table option to the `CREATE TABLE` statement:
  - `CREATE TABLE t (i INT) ENGINE = INNODB;`
  - `CREATE TABLE t (i INT) TYPE = MEMORY;`
- `--default-storage-engine` or `--default-table-type` server startup option: set the default engine (default MyISAM)
- Database may contain tables of different types



10

ULB

# Storage Engines

# Transactional or Not

- Transaction-safe tables (InnoDB) have several advantages over non-transaction-safe tables:
  - They are safer. Even if MySQL crashes or you get hardware problems, you can get your data back, either by automatic recovery or from a backup plus the transaction log
  - You can combine many statements and accept them all at the same time with the `COMMIT` statement (if `autocommit` is disabled) or `ROLLBACK`
  - If an update fails, all of your changes are reverted
  - Transaction-safe storage engines can provide better concurrency for tables that get many updates concurrently with reads.

ULB

# Storage Engines

# Transactional or Not

- Non-transaction-safe (MyISAM) tables have several advantages:
  - Much faster
  - Lower disk space requirements
  - Less memory required to perform updates
- Combine transaction-safe and non-transaction-safe tables in the same statements, changes to non-transaction-safe tables still are committed immediately and cannot be rolled back



12

ULB

# MyISAM

- MyISAM is the default storage engine. It is based on the older ISAM code
- Non-transactional very efficient engine
- No check on `FOREIGN KEY` constraints
- Can be corrupted - need repair



13

ULB

# MyISAM Table Corruption

- Corruption can occur when:
  - `mysqld` process is killed in the middle of a write
  - Unexpected computer shutdown (e.g. computer is turned off)
  - Hardware failures.
  - External program (such as `myisamchk`) modify a table that is being modified by the server at the same time



14

ULB

# MyISAM Table Corruption

- Symptoms of a corrupt table are:
  - `Incorrect key file for table: '...'. Try to repair it`
  - Queries don't find rows or return incomplete results
- Use `CHECK TABLE` statement, and `REPAIR TABLE`, when `mysqld` is not running, use `myisamchk` utility



15

ULB

# InnoDB

- Transaction-safe (ACID compliant) storage engine with commit, rollback, and crash recovery capabilities and supports `FOREIGN KEY` constraints
- Designed for maximum performance when processing large data volumes
- CPU efficiency is probably not matched by any other disk-based relational database engine



16

ULB

# InnoDB
# Multi-versioning and ACID

- Old versions of rows are kept in a rollback segment
- Internally, InnoDB adds two fields to each row stored:
  - the transaction identifier for the last transaction that inserted or updated the row
  - the roll pointer pointing to an undo log record written to the rollback segment (contains the information necessary to rebuild the content of the row before it was updated)
- InnoDB uses the rollback segment for undo operations in a transaction rollback and for earlier versions of a row for a consistent read
- Commit transactions regularly, including for consistent reads for InnoDB to purge its rollback segment



17

ULB

# Memory aka HEAP

- Contents that are stored in memory
- One disk file: `table_name.frm`
- Use hash indexes by default thus very fast
- Used for temporary tables
- No persistence: empty when the server restarts.



18

ULB

# Memory

- Use a fixed-length row storage format, no `BLOB` nor `TEXT`
- `max_heap_table_size`: impose a maximum size on MEMORY tables
- Needs sufficient memory to maintain all MEMORY tables that are in use at the same time
- Execute `DELETE` or `TRUNCATE TABLE`, or remove the table altogether using `DROP TABLE` when no more needed



19

ULB

# Storage Engines

- ISAM: deprecated original storage engine
- ARCHIVE: storing large amounts of data without indexes with a very small footprint
- CSV: stores data in text files using comma-separated values format



20

ULB

# Storage Engines

- NDB Cluster: MySQL Cluster Engine, manage tables partitioned over many computers (Only Unix for now)
- EXAMPLE: "stub" engine that does nothing
- BLACKHOLE: accepts but does not store data and retrievals always return an empty set
- FEDERATED: stores data in a remote database, works with MySQL only, in future releases, intends to connect to other data sources using other drivers or client connection methods (transparent gateway?)

# GUI Tools

ULB

# Administrator

- Configuring, monitoring and starting and stopping a MySQL server, managing users and connections, performing backups, and a number of other administrative tasks
- Its graphical user interface makes it more intuitive
- It provides a better overview of the settings that are crucial for the performance, reliability, and security of your MySQL servers
- It displays performance indicators graphically, thus making it easier to determine and tune server settings
- MySQL Administrator is designed to work with MySQL versions 4.0 and higher
- http://dev.mysql.com/doc/administrator/en/index.html



23

# Query Browser

- Creating, executing, and optimizing queries in a graphical environment
- Intuitive GUI
- MySQL versions 4.0 and higher.
- For MySQL Query Browser, there's a dedicated forum available on http://forums.mysql.com/list.php?108.
- http://dev.mysql.com/doc/query-browser/en/index.html

ULB

# PHPMyAdmin

- Admin + Query ! :-)



25

ULB

# User Management

ULB

# Access Control Roles

- Identification: user_name and host
- Authentication: identification and password
- Authenticate users, associate privileges
- Control access:
  - Stage 1: connection access
  - Stage 2: each statement against user privileges



27

ULB

# Connection Privileges Examples

| Host | User | Allowable Connections |
|---|---|---|
| 'thomas.loc.gov' | 'fred' | fred from host thomas.loc.gov |
| 'thomas.loc.gov' | '' | any user from host thomas.loc.gov |
| '%' | 'fred' | fred from anywhere |
| '%' | '' | anyone from anywhere |
| '%.loc.gov' | 'fred' | fred from anything.loc.gov |
| 'x.y.%' | 'fred' | fred from host x.y.net, x.y.org... |
| '144.155.166.%' | 'fred' | fred from 144.555.166.something |
| '144.155.166.0/255/255/255/0' | 'fred' | same as previous |

ULB

# Access Control Tables Used

- `mysql.user`, `mysql.db`, and `mysql.host` tables at both stages of access control
- `mysql.tables_priv` and `mysql.columns_priv` tables provide finer privilege control at the table and column levels
- For verification of requests that involve stored routines, the server may consult the `procs_priv` table.



29

ULB

# Access Control Statements

- Account management statements: `grant`, `revoke`, `create user`, `drop user`...
- Syntax: http://dev.mysql.com/doc/refman/5.0/en/account-management-sql.html
- Or directly with: update, insert, delete in the `mysql.*` tables
- Consult privileges: `mysql> SHOW GRANTS FOR 'bob'@'pc84.example.com`
- mysqlaccess: `shell> mysqlaccess --help`



30

ULB

# Access Control
# Limiting Account Resources

- The number of:
  - queries per hour
  - updates per hour
  - times an account can connect per hour
  - simultaneous connections

```
mysql> GRANT ALL ON customer.* TO
'francis'@'localhost'
    ->     IDENTIFIED BY 'frank'
    ->     WITH MAX_QUERIES_PER_HOUR 20
    ->          MAX_UPDATES_PER_HOUR 10
    ->          MAX_CONNECTIONS_PER_HOUR 5
    ->          MAX_USER_CONNECTIONS 2;
```

```
mysql> GRANT USAGE ON *.* TO 'francis'@'localhost'
    ->     WITH MAX_CONNECTIONS_PER_HOUR 0;
```

ULB

# Access Control Statements MySQL vs Standard SQL

- Identification: hostname and username vs only a username.
- Standard SQL does not have global or database-level privileges, nor does it support all the privilege types that MySQL supports.
- MySQL does not support the standard SQL TRIGGER or UNDER privileges.
- In standard SQL, when you drop a table, all privileges for the table are revoked. In standard SQL, when you revoke a privilege, all privileges that were granted based on that privilege are also revoked. In MySQL, privileges can be dropped only with explicit REVOKE statements or by manipulating values stored in the MySQL grant tables.
- In MySQL, it is possible to have the INSERT privilege for only some of the columns in a table. In this case, you can still execute INSERT statements on the table, provided that you omit those columns for which you do not have the INSERT privilege. The omitted columns are set to their implicit default values if strict SQL mode is not enabled. In strict mode, the statement is rejected if any of the omitted columns have no default value. (Standard SQL requires you to have the INSERT privilege on all columns.)

ULB

# Access Control

## When privileges takes effects

- When `mysqld` starts, it reads all grant tables into memory
- When the server reloads the grant tables it takes effect:
  - table and column privilege: the client's next request
  - database privilege: next `use db_name` statement
  - changes to global privileges and passwords: next time the client connects
- `grant`, `revoke`,... reloads grant tables
- `update`, `insert`,... does not => use `flush privileges` or `mysqladmin reload`



33

ULB

# Security

- General guidelines
- Secure against attackers
- `mysqld` security options
- Security issue with `LOAD Data`



34

ULB

# Security

- MySQL security is based on Access Control Lists (ACLs) for connections, queries, and other operations
- SSL-encrypted connections between MySQL clients and servers

ULB

# Security
# General Guidelines

- Fully protect the entire server host (not just the MySQL server): eavesdropping, altering, playback, and denial of service
- root only has access to user table in the mysql database! This is critical.
- Do not grant more privileges than necessary. Never grant privileges to all hosts.
- Use the `SHOW GRANTS` statement to check which accounts have access to what



36

ULB

# Security
# Password

- Do not choose passwords from dictionaries. Special programs exist to break passwords.
- Check password of root and anonymous users
- Do not store any plain-text passwords in your database, use `MD5()`, `SHA1()`, or some other one-way hashing function and store the hash value



37

ULB

# Security
# Web Development

- Do not trust any data entered by users of your applications: special or escaped character sequences in Web forms, URLs, or whatever application you have built: "`; DROP DATABASE mysql;`".
- A common mistake is to protect only string data values. Remember to check numeric data as well.
- Example of denial of service:

  `SELECT * FROM table WHERE ID=234` when a user enters the value `234`, the user can enter the value `234 OR 1=1`



38

ULB

# Security
# Application Development

- Checklist:
  - Try to enter single and double quote marks (‘’’ and ‘’”’) in all of your Web forms
  - Try to modify dynamic URLs by adding %22 (‘’”’), %23 (‘#’), and %27 (‘’’)
  - Try to enter characters, spaces, and special symbols rather than numbers in numeric fields. Your application should remove them before passing them to MySQL or else generate an error. Passing unchecked values to MySQL is very dangerous!
  - Check the size of data before passing it to MySQL
  - Have your application connect to the database using a username different from the one you use for administrative purposes. Do not give your applications any access privileges they do not need



39

ULB

# Security
# Application Development

- Encrypt data over the Internet, use an encrypted protocol such as SSL or SSH
- Use `tcpdump` and strings utilities:

```
shell> tcpdump -l -i eth0 -w - src or dst port 3306 | strings
```



40

ULB

# Security
# Unix-MySQL

- Never run the MySQL server as the Unix `root` user (user with the `FILE` privilege can create files as root e.g. `~root/.bashrc`
- `mysqld` refuses to run as root unless: --user=root option.
- `mysqld` should be run as an ordinary, unprivileged user instead

```
[mysqld]

user=mysql
```

- Make sure that the only Unix user with read or write privileges in the database directories is the user that `mysqld` runs as

ULB

# Security

- Do not grant the `PROCESS` or `SUPER` privilege to non-administrative users. The output of `mysqladmin processlist` and `SHOW PROCESSLIST` shows the text of any statements currently being executed
- `mysqld` reserves an extra connection for users who have the `SUPER` privilege, so that a MySQL `root` user can log in and check server activity even if all normal connections are in use
- The `SUPER` privilege can terminate client connections, change system variables, control replication servers



42

ULB

# Security

- If you do not trust your DNS, use IP numbers rather than hostnames in the grant tables. Don't grant table entries using hostname values that contain wildcards
- Limit user resources
- `--ssl*`: options that begin with `--ssl` specify whether to allow clients to connect via SSL and indicate where to find SSL keys and certificates



43

ULB

# Security
# mysqld Options

- `--secure-auth`: disallow authentication for accounts that have old (pre-4.1) passwords.
- `--skip-grant-tables`: server don't use the privilege system at all, anyone with access to the server has unrestricted access to all databases. Revert with `mysqladmin flush-privileges`
- `--skip-name-resolve`: hostnames are not resolved. All Host column values in the grant tables must be IP numbers or localhost
- `--skip-networking`: do not allow TCP/IP connections over the network. All connections to `mysqld` must be made via Unix socket files
- `--skip-show-database`: `SHOW DATABASES` only if `SHOW DATABASES` privilege, displays all database names. Without this option, `SHOW DATABASES` is allowed to all users, displays each database name the user has the `SHOW DATABASES` privilege for or some other privilege. Any global privilege is a privilege for the database.



44

# Falcon Storage Engine

- InnoDB and BDB storage engine most used in MySQL, best transactional engine for MySQL today. Both bought by... Oracle!
- True Multi Version Concurrency Control (MVCC) enables records and tables to be updated without the overhead associated with row-level locking mechanisms. The MVCC implementation virtually eliminates the need to lock tables or rows during the update process.
- Flexible locking, including flexible locking levels and smart deadlock detection keep data protected and transactions and operations flowing at full speed.
- Optimized for modern CPUs and environments to support multiple threads allowing multiple transactions and fast transaction handling.

# Falcon Storage Engine

- Transaction-safe (fully ACID-compliant) and able to handle multiple concurrent transactions.
- Serial Log provides high performance and recovery capabilities without sacrificing performance.
- Advanced B-Tree indexes.
- Data compression stores the information on disk in a compressed format, compressing and decompressing data on the fly. The result is in smaller and more efficient physical data sizes.