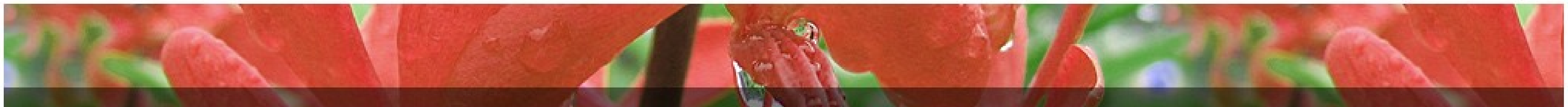
The MySQL Database

Khalid Baheyeldin

KWLUG
May 2010
http://2bits.com

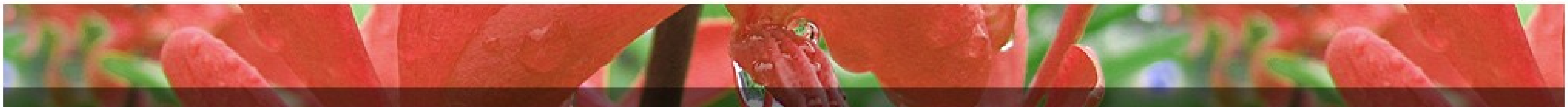
Agenda

- Introduction
- What is MySQL?
- Engines
- Installation
- Repair and Optimize
- Backup and Restore
- Replication
- Discussion



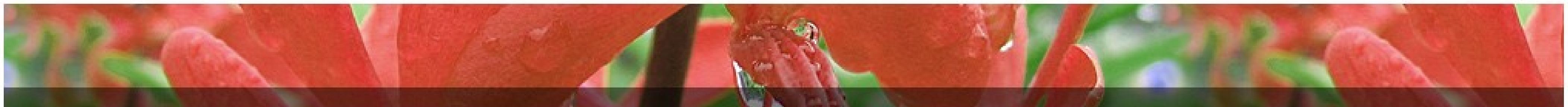
About Khalid

- 25 years in software development and consulting
- Sinclair ZX Spectrum, mainframe, then UNIX since 1987
- Linux discovered 1990, using it regularly since 1995, “LAMP” since 1999
- Open source developer, contributor since 2003
- Full time open source consulting



About 2bits.com

- Founded in 1999
- Drupal CMS/CMF since 2003
- Full time consulting
- Services
  - Drupal development
  - LAMP performance optimization and tuning
  - Server provisioning for performance and uptime
  - Manage huge sites for various clients
- http://2bits.com



Relational Databases

Before relational ...

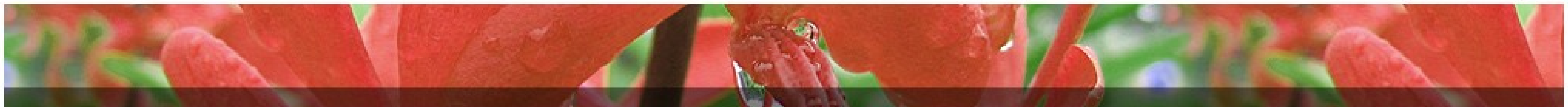
- Indexed files (read, write, re-write)
- Hierarchical databases (IBM IMS)
- Network databases (Cincom TOTAL, master/variable ... read, read next, ...)

Relational Databases

- Normalization (3rd normal form 3NF)
- Codd and Date
- Fields -> Columns
- Records -> Rows
- Set operations
- Joins
- Cartesian products

Relational Databases

- Commercial
  - Oracle
  - IBM DB2
  - Informix
  - Ingres
  - SQL-Server
- Free
  - MySQL
  - PostgreSQL
  - Firebird
  - SQLite



ACID

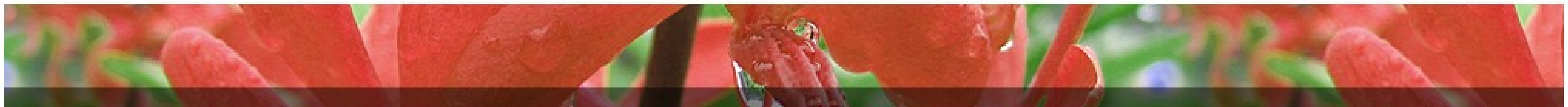
- **A**tomicity: modifications “all or nothing”
- **C**onsistency: database remains in a consistent state at all time (no partial updates)
- **I**solation: operations do not see other transactions' modifications until they are complete
- **D**urability: Once a user is notified that a transaction is complete, it cannot be lost

SQL

- SQL = Structured Query Language
- Used by all relational databases today
- Various levels of standardization (each database has its own specific set)

SQL command types

- DDL (Data Definition Language)
  - CREATE DATABASE, CREATE TABLE, CREATE VIEW, CREATE INDEX, ...
- DML (Data Manipulation Language)
  - SELECT, INSERT, DELETE, UPDATE...



Information Schema

- ANSI standard
- Called information_schema
- A database that holds meta data on other databases, tables, columns, ...etc.

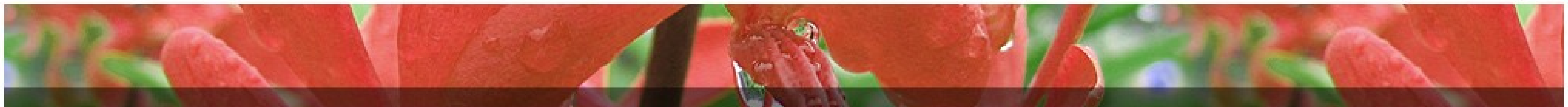
Information Schema

- Number of tables in all databases

```
SELECT COUNT(table_name) FROM
information_schema.tables;
```

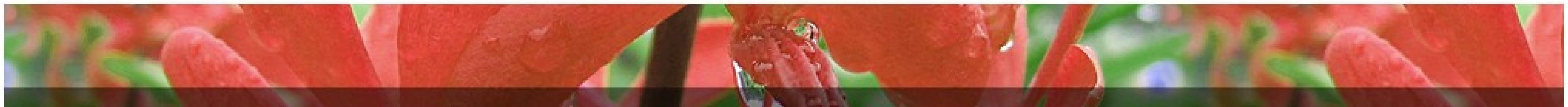
- Table names in a certain database

```
SELECT table_name FROM
information_schema.tables WHERE
table_schema = 'my_database';
```



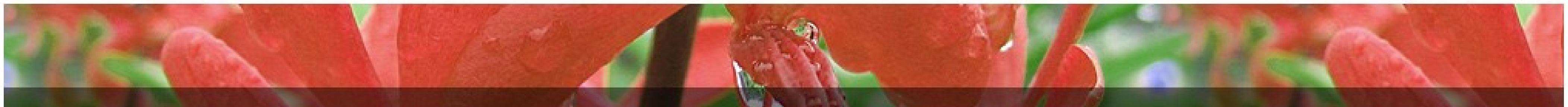
What is MySQL?

- A relational database server
- Open Source (GPL licensed)
- Cross platform
- Version 5.1 is latest stable (ready for production)



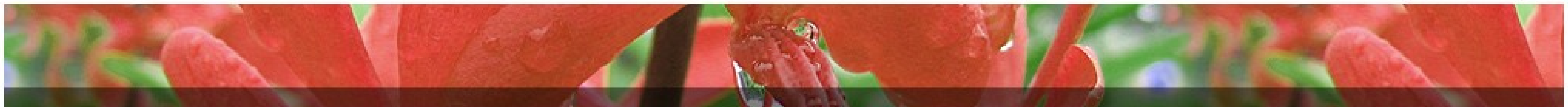
Who uses MySQL?

- Google, Yahoo
- Sears, Symantec, UN FAO, TicketMaster,
- SecondLife
- Web sites: Wikipedia, Facebook, Flickr, Slashdot, Continental Airlines, LinkedIn, craiglist, NeoPets, StumbleUpon, LiveJournal, Drupal.org, WhiteHouse.gov, Wordpress.com, YouTube



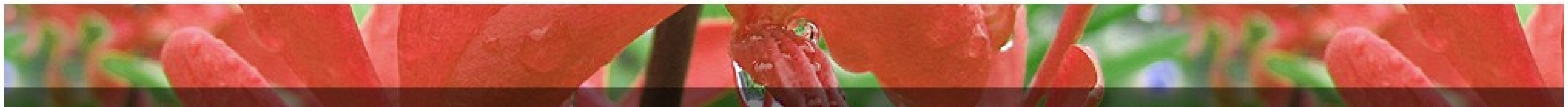
Platform support

- Linux
- FreeBSD
- Various UNIX variants (Solaris, HP/UX, AIX)
- Windows
- MacOS/X



Application support

- Many applications support MySQL as the storage backend.
- More than those that support PostgreSQL
- Anecdote: Ubuntu repository 138 packages vs. 80.



Language support

- MySQL is written in C and C++
- Supported by libraries for most commonly used languages
- C/C++, PHP, Perl, Java, Python, Ruby
- ODBC as well

MySQL Advantages

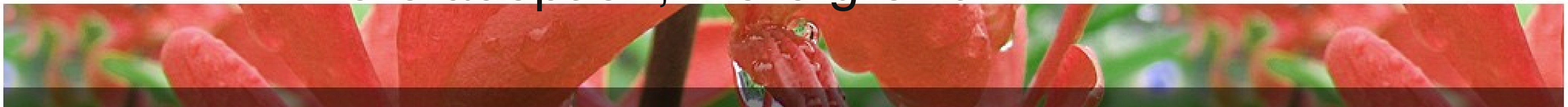
- Easy to use and administer
- Supported by all languages and frameworks
- Small enough
- Powerful enough
- Upgrades are easy (data format remains the same)

Disadvantages

- Ignores statements that the engine does not support:
  - e.g. Non transactional engine (MyISAM) but using BEGIN TRANSACTION, ...etc.
- No sub second measurement
  - SHOW PROCESSLIST
  - 3rd Party patches have it (Percona)

MySQL History

- The beginnings
  - Initially there was mSQL which hosters used
  - MySQL 1994 Founded by Monty Widenius and David Axmark
  - MySQL emulated MiniSQL (mSQL)
  - Offered free to hosting companies
  - “My” is Monty's daughter
  - The company MySQL AB formed in 1995
  - Was used for decision support (read heavy)
  - More adoption, more growth

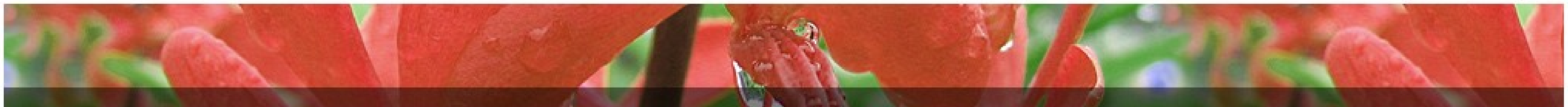


MySQL History

- Licensing
  - Was “previous version GPL'd”
  - Later fully GPL'd
- Very recent
  - Feb 2008: MySQL AB purchased by Sun for $1 billion
  - 2009: Oracle purchasing Sun
  - EU initially objecting to the sale, and Monty Widenius fuelling it

MySQL future

- EU approved the merger (saw PostgreSQL as a viable competitor), Monty Widenius objections not withstanding
- Already there are several efforts
  - Friendly forks, not rival
- MySQL is GPL, so will live on in one form or another
- The current issue is who owns the copyright for the proprietary version, the GPL version is safe



Drizzle

- An attempt to refactor the code base, to make it simpler and more pluggable
- Brian Akers (krow)
- 1.2M loc to 300K loc
- Optimizations (any one use 4 bit integers?)
- Memory and CPU
- Transactional
- Web applications

MariaDB

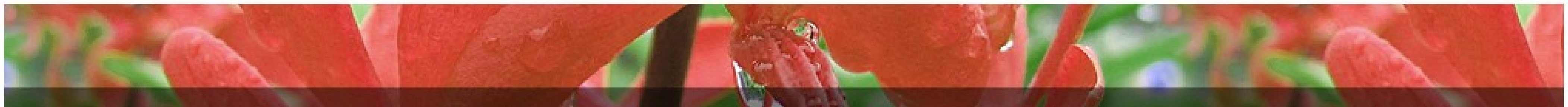
- By co-founder (Monty Widenius)
- Named after his daughter (My, Maria, ...)
- Stared as a reaction to Oracle owning InnoDB, outside of MySQL AB
- Transactional replacement for MyISAM and InnoDB

Our Delta

- Set of community patches
- Existed before MySQL was sold off

Percona

- Set of performance and scalability patches
- XtraDB
- XtraBackup

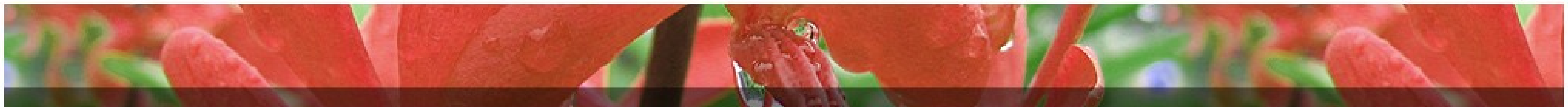


Others ...

- Google's patches
- Facebook patches
- Both available publicly ...

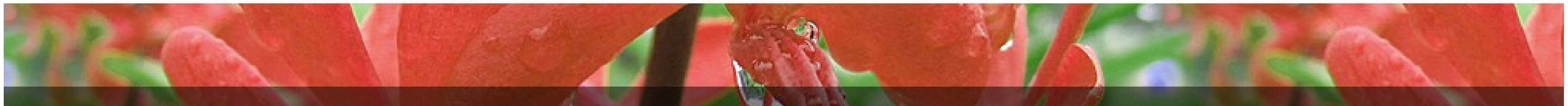
MySQL Licensing

- Dual Licensed
  - Proprietary
  - GPL
- Claimed that protocol is a GPL “conduit”
  - And STDs can be contracted via 1-900 numbers
- Does not apply to proprietary licensed version
- Community Edition is GPL
- Required contributor agreement to assign rights to MySQL



Installation

- Distro binaries
  - Far easier to install and maintain
  - Easier to apply security updates
  - Good for most cases
- Compiling from Source
  - Can customize further
    - e.g. Remove certain engines for example
    - Patches for extra features (e.g. Percona's)
  - You take ownership of security patches and bug fixes



Distro install

- Debian/Ubuntu
  - Minimal install:

```
aptitude install mysql-server
```

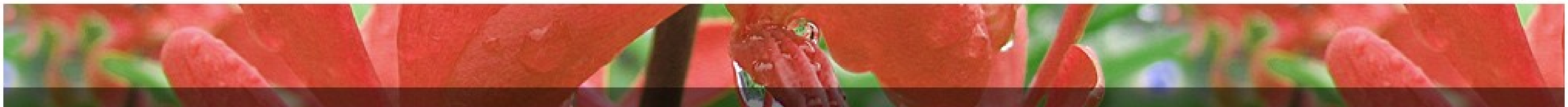
- CentOS
  - Yum

MySQL architecture

- A database server process (mysqld)
  - Many threads inside this process
- Pluggable storage engines
- Serves clients connecting on a specific port (3306)
- Clients can be in various forms (language libraries, command line, GUI, ...etc.)

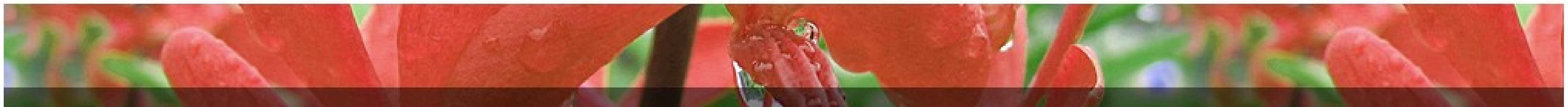
MySQL Engines

- Too many to know them all
- Some general purpose, some very specialized
- Various proliferation/adoption levels
- Most are third party developed, not by MySQL AB, which leads to interesting relationships
- `CREATE TABLE mytable ...` **`ENGINE=InnoDB`**;



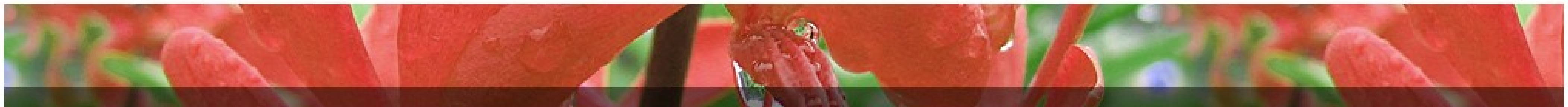
MySQL Engines (cont'd)

- Stub
  - Example (code example for engine developers)
  - Blackhole (/dev/null)
- General purpose
  - **MyISAM:** Non-Transactional. Optimized for read heavy applications (decision support analytics, web sites, ...etc.)
  - **InnoDB:** Transactional, ACID



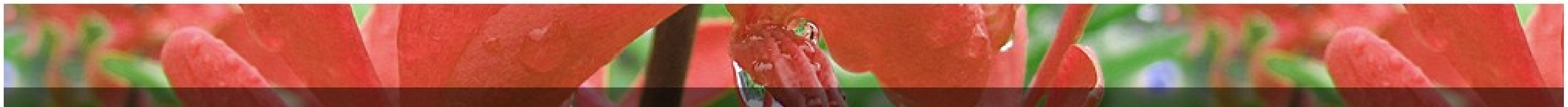
MySQL Engines (cont'd)

- **Memory/HEAP**: useful for temporary tables (CREATE TEMPORARY TABLE ....)
- **Merge/MRG_MyISAM**: a way of partitioning data by value. Identical MyISAM tables “merged” as one, e.g. txn_2009, txn_2010, ...etc..
- **Archive**: Storing large amount of data without indexes in a compact footprint (e.g. Logs)



MySQL Engines (cont'd)

- **NDB**: Network Database, a clustering engine, where the database spans several nodes.
- **BerkleyDB (BDB)**: SleepyCat software, owned by Oracle. Transactional. Not in the default install since 5.1.
- **CSV**: Yes, a comma separate variable engine for a relational database!



MySQL Engines (cont'd)

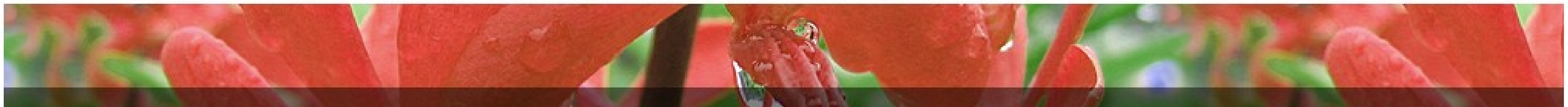
- Emerging
  - **Falcon** (5.2 alpha, transactional, by MySQL itself. Not yet mature)
  - **MariaDB** (by Monty Program AB, transactional. Not yet mature)
- Mature/Legacy
  - SolidDB, PrimeBase XT, ...

MySQL Engines (cont'd)

- Niche engines/Other
  - IBM DB2
  - NitroEDB, BrightHouse, OpenOLAP
  - InfoBright, LucidDB, InfiniDB, MonetDB
  - Speculation: Maybe Oracle in the future?

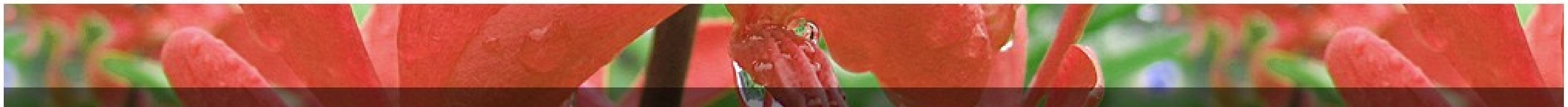
Break?

Need to stretch your legs?



MyISAM Engine

- Non-transactional engine
- Lightweight
- Fast! Optimized for reads
- Table level locking
  - Bad for high traffic sites
- Indexes and data in separate files
  - Easier to recover from a crash



MyISAM Engine

- Each table is composed of the following:
  - .frm file (table definition, columns, ...etc.)
  - .MYI file (index)
  - .MYD file (data)

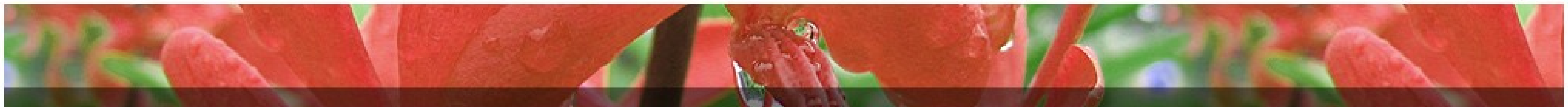
InnoDB Engine

- Developed by InnoBase
- Transactional engine
- Row level locking
- Foreign keys
- Tablespaces (like Oracle), or one file per table
- Index and data stored in the same file
- Each table can be MyISAM or InnoDB
- More resource intensive

InnoDB Engine

Combined table space mode (default)

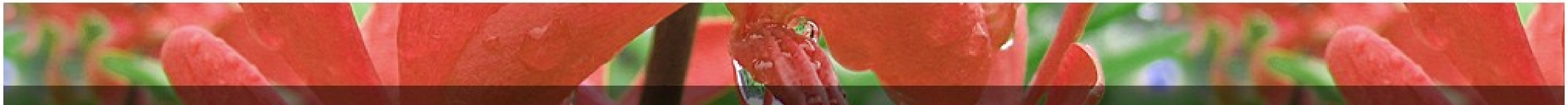
- .frm (table definition, same as MyISAM)
- Combined data for all databases and tables:
  - ibdataN file (data and index combined)
  - ib_logfileN (transaction log files)



InnoDB Engine

Combined table space mode (default)

- Configuration
- innodb_data_home_dir = /ibdata
- innodb_data_file_path=ibdata1:100M;ibdata2:100M:autoextend
- Another example
- innodb_data_file_path=**/data/db/**ibdata1:100M;**/data2/db/**ibdata2:100M:autoextend
- Yet another example
- innodb_data_file_path=ibdata1:100M;ibdata2:100M:autoextend:**max:10G**



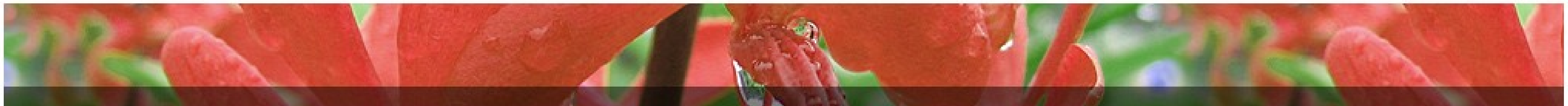
InnoDB Engine

File per table mode

- .frm (same as above)
- One **.ibd** file per table, in the database's directory
- ib_logfileN files

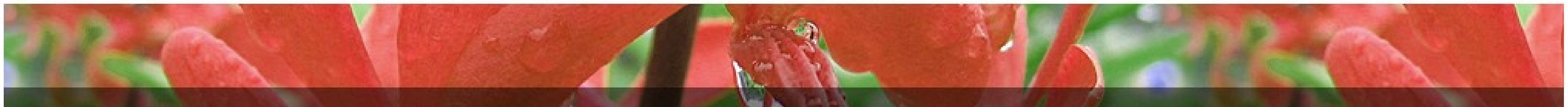
Tip: Convert to InnoDB

- InnoDB is better for not locking the whole table
- Better concurrency
- Simple to convert tables from MyISAM to InnoDB
  - `ALTER TABLE table1 Engine=InnoDB;`
- Be careful with large tables though
  - Took 6 hours to complete for one client!



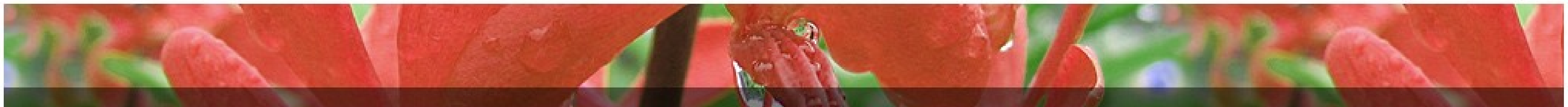
Creating a database

- Easiest way (command line)
  - `mysqladmin create db_name`
- Or from MySQL prompt
  - `CREATE DATABASE db1;`



Creating a table

- Data Definition Language (DDL)
  - `CREATE TABLE table1 (column1 INTEGER, column2 VARCHAR(25));`

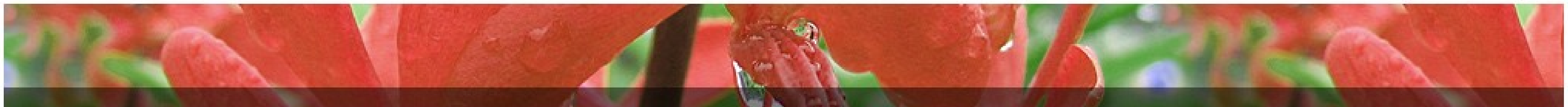


Inserting Data

- As follows
  - `INSERT INTO table1 (column1, column2) VALUES (1, 'first row');`
  - `INSERT INTO table1 (column1, column2) VALUES (2, 'second row');`
  - `INSERT INTO table1 (column1, column2) VALUES (3, 'third row');`

Deleting Data

- As follows
  - `DELETE FROM table1 WHERE column1 = 2;`



Retrieving Data

- As follows:
  - `SELECT * FROM table1; -- All rows`
  - `SELECT * FROM table1 WHERE col1 = 2;`
  - `SELECT col2 FROM table1 WHERE col1 = 2;`

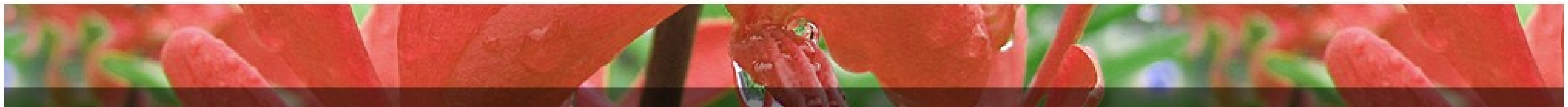
Backup (dumps)

Works for all table types (including InnoDB)

- Not a consistent point in time backup, unless you stop the application (e.g. Apache), or application is mostly read only
- Simplest form

```
mysqldump db_name > file.sql
mysqldump —all-databases
```

- Output is basically MySQL CREATE TABLE and INSERT statements



Hot Backup

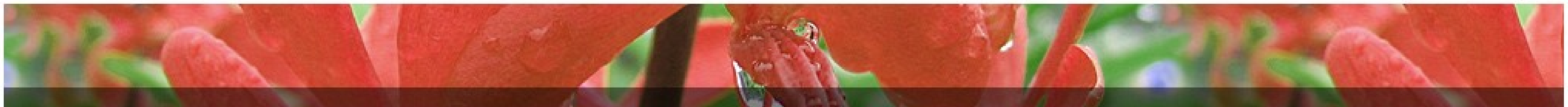
- Also called Online backup
- Attempts a consistent point in time copy of the database
- InnoDB Hot Backup
  - Commerical product (annual license)
- Percona's XtraBackup is the open source equivalent

Replication

- Two (or more) networked servers running continuously updated copies of a database
- Asynchronous (unlike NDB cluster, which are synchronous)
- Benefits
  - For high availability
  - For scalability (scaling out)
  - For reporting/analytics
  - For backup

Replication

- Easy to do under MySQL, has been for many years.
- Only available in recent versions of PostgreSQL
- Incurs some overhead, specially when there are many slaves, and/or many transactions

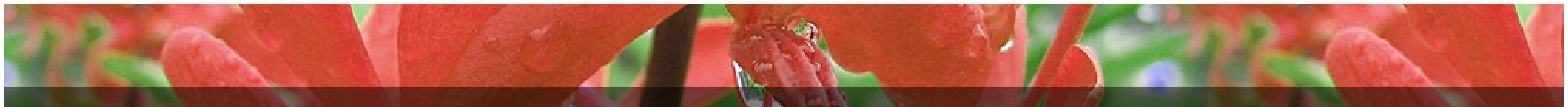


Master/Slave

- One version of the truth (master) for read/write, others are read only
- Binary Log required
- Each server has a server-id in the configuration
- Dump the database from the master
- Recreate the database on the slave
- Sync the binary log offset
- The slave executes the transactions in the same order as the master, with some lag

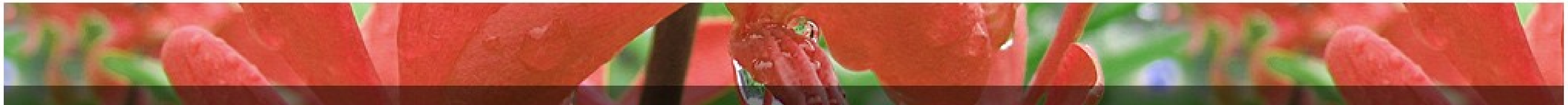
Master/Slave

- Generally application safe, i.e. Applications do not require changes, nor need to be aware of replication
- You can do a CHANGE MASTER on the slave if the master fails
- Details on how to set this up here
  http://www.howtoforge.com/mysql_database_replication



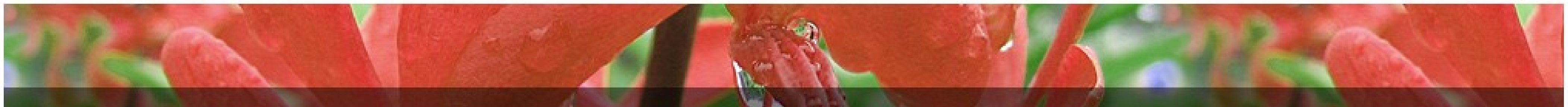
Master/Master

- More than one read/write masters (master pair)
- Redundancy (with Flipper for IP switching)
- Not all applications like this!
- Less used than master/slave
- Details on how to set it up here
- http://www.howtoforge.com/mysql_master_master_replication
- http://mysql-mmm.org/
- http://provenscaling.com/software/flipper
- http://capttofu.livejournal.com/1752.html



Upgrading

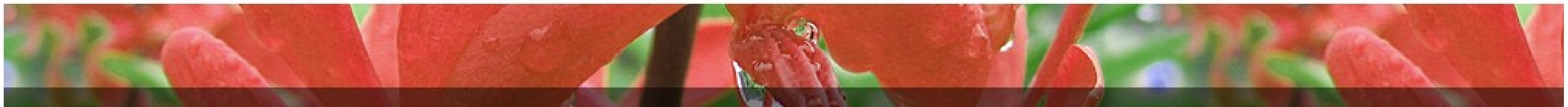
- Nothing special, file formats remain the same across versions
- PostgreSQL used to require an export/upgrade/import exercise on upgrading



Check & Repair Tables

Checking

- Works for MyISAM and InnoDB (sometimes?)
- From Operating System command line
  - `mysqlcheck -c db_name table_name`
- From SQL
  - CHECK TABLE table_name



Check & Repair Tables

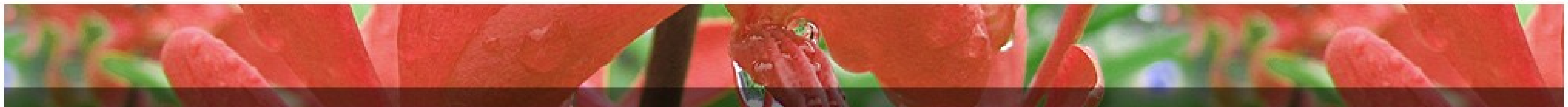
Repairing MyISAM

- From Operating System command line
  - `mysqlcheck -r db_name table_name`
- From SQL
  - REPAIR TABLE table_name

Check & Repair Tables

Repairing InnoDB

- Most of the time, it is automatic, on start, it uses the binlog that logged transactions, and attempts to replay them.
- In some cases, there will be corruption that does not get automatically fixed. The database will refuse to start. Review the logs first.
- What are the options?
- Add this to my.cnf (`innodb_force_recovery=1`)



Check & Repair Tables

Repairing InnoDB (cont'd)

- Add this to your my.cnf file

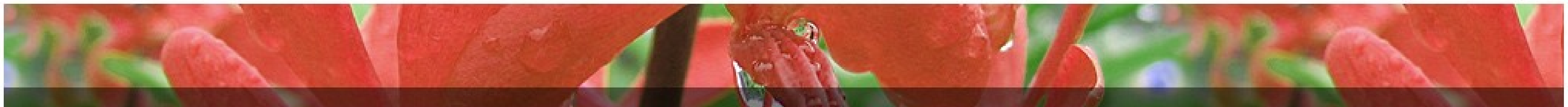
```
innodb_force_recovery=1
```

- Start the database, create a MyISAM table with the same structure, and do an INSERT/SELECT from the corrupted table
- If it does not start, try using 2 or 3 up to 8
- Or use http://code.google.com/p/innodb-tools/

Optimizing

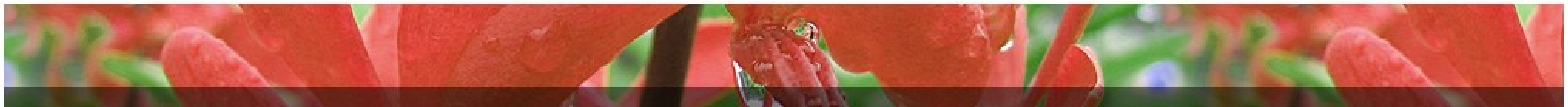
- Table optimization utility
- Makes the indexes optimized, compacts unused space from deleted rows, ... etc.
- Updates index statistics
- Can help performance in certain cases
- Can schedule it to run automatically at quiet times nightly or on weekends

```
mysqlcheck -o db_name
```



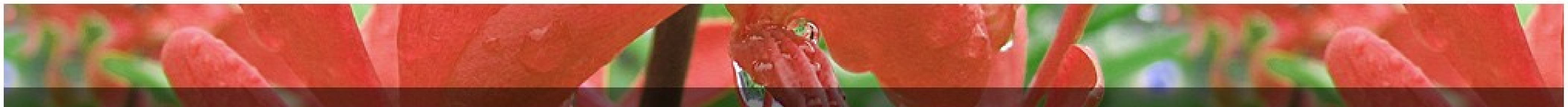
Tuning tools

- Use PROCESSLIST to check for locks or long running queries
- Use EXPLAIN to see which queries are slow
- Use SET PROFILING = 1, then SHOW PROFILES (Community Edition only)



Tuning

- Big topic, often application specific, even site specific (e.g. one Drupal site to the other)
- Avoid MySQL queries:
  - Static caching in the application (e.g. For same page load, store data in variables `static $something;`)
  - Persistent object caching (e.g. memcached, application needs to be modified)
- Give more RAM to MySQL, and it will be happier

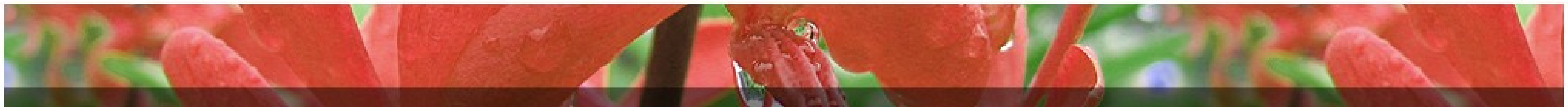


Tuning

- Turn on the MySQL query cache
- Convert tables with locks to InnoDB
- Put stuff on separate disks spindles (e.g. Different table spaces, logs, other application files, such as Apache logs, media files, ...etc.
- Watch for LVM and RAID meta layers)

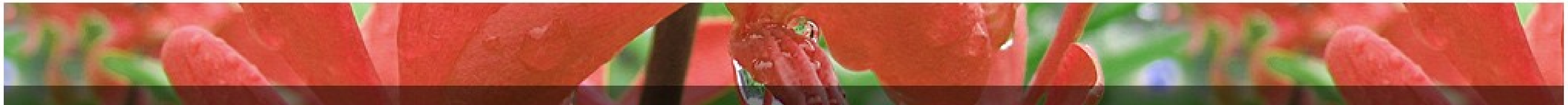
Tuning

- Enable slow query logging
- Watch resource utilization (CPU, memory, disk I/O – recent presentation on tools)
- MySQL Tuning scripts/reports (use with caution)
- Maat Kit



Alternatives: Relational

- PostgreSQL
  - BSD licensed
  - ACID
  - Harder to administer
  - Can be slower due to overhead
- SQLite
  - Single file, no server processes/threads
  - For embedded applications,
  - Less concurrency due to write locks
- FireBird

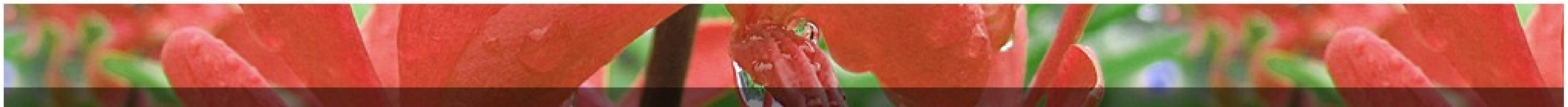


Alternatives: NoSQL

- Trades speed and simplicity for consistency and complexity
- I see it as: complimentary, not replacement (
  - NoSQL = N(ot) O(nly) SQL
- Cassandra. Created by Facebook. Used on Digg, Reddit, Twitter. A BigTable distributed database
- MongoDB. Getting lots of use in Drupal now.
- CouchDB

Conclusion

- Capable
  - On par with PostgreSQL (InnoDB)
- Powers much of the web
- Proven
- Easy to use
- Easy to administer
- Free!



P in LAMP?

Beginner level PHP presentation?

Discussion

Questions?

Comments?