

# *OKC MySQL Users Group*



# *OKC MySQL*

- Discuss topics about MySQL and related open source RDBMS
- Discuss complementary topics (big data, NoSQL, etc)
- Help to grow the local ecosystem through meetups and events

*Sponsored By...*



# *Introduction to MySQL*

# *Some General Database Terms*

- Relational Database
  - Stores data and how it is related
  - Think 2 dimensional “tables” with data and relationships
- ACID (for databases, not the 60s)
  - Atomicity (all or nothing)
  - Consistency (database moves from valid state to valid state)
  - Isolation (transactions operate independently)
  - Durability (once data is committed, it will remain)
- Transaction
  - Set of queries that must succeed (or fail) as a group

## *More Database Terms...*

- Table
  - Collection of fields in several rows
  - Think of a table as a spreadsheet
- Primary Key (PK)
  - Key identifier for a row (non-specific, randomly generated ID)
- Join
  - How tables are related
  - Often times, a table with PKs of other tables

So...  
What is MySQL??

# *MySQL Intro*

- Open Source Relational Database
  - Owned by Oracle
    - Previously MySQL AB, Sun Microsystems
  - Community version free to use in production (restriction free)
  - No licensing fees
  - Paid enterprise version available
- Database Server
  - Allows for remote connections
  - Similar to SQL Server and/or Oracle
  - Very different from the MS Access model (single file)

# *Components of MySQL*

- Server (mysqld)
  - Various storage engines (plugins)
  - Connection handling
  - Diagnostics
- Client (mysql)
  - Standard command line interface
  - Issue queries, etc



## *Components – Cont.*

- Connectors/Drivers
  - Language specific connectors (php, python, etc)
  - Interact with the database from remote application
- Third Party GUIs
  - MySQL Workbench
  - phpmyadmin (web based)

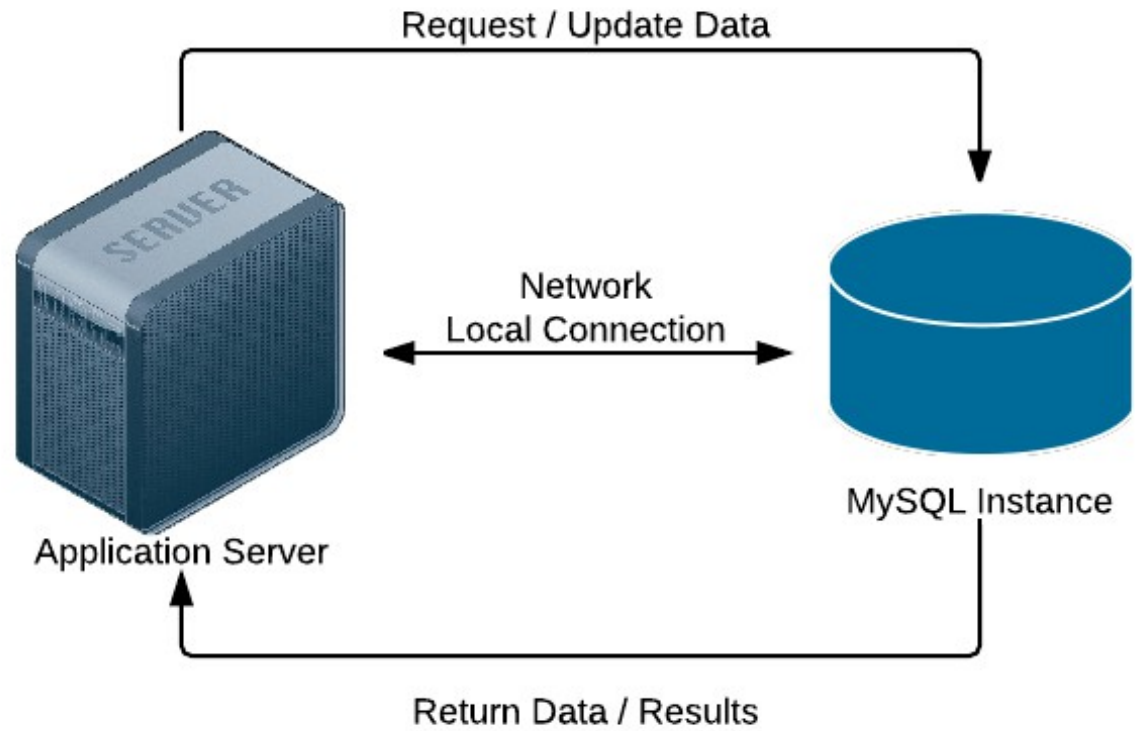
# *MySQL Terms*

- Storage Engine
  - Plugin to the actual mysql server
  - Determines how data is stored (and interacted with)
- InnoDB
  - Transactional, ACID compliant storage engine
  - Generally recommended storage engine
- MyISAM
  - Legacy default storage engine
  - Non-ACID, non-transactional

# *More MySQL Terms*

- Replication
  - Built in mechanism to keep multiple servers in sync
- Master/Slave
  - Hierarchy within replication
  - Master → Slave → Slave ...
- Instance
  - Single mysqld program running, with all data
- Database (or Schema)
  - A collection of tables
- User
  - A system user with certain privileges on certain tables/databases

# *How do I use it?*



# *Pros*

- Open source and free to use
- Horizontally scalable for read traffic
- Relatively lightweight
- Highly configurable (more on the linux side of things)
- Flexibility in setup
  - Both from a per-instance standpoint and full deployment (HA, etc)
- Well vetted in large scale production setups – i.e. battle tested
- Several open source tools available for management
- Available on multiple platforms (Windows, Linux)

# *Cons*

- Some features require outside management (backups, HA, monitoring, scaling to some extent)
- Some metrics not readily available
- Harder to scale write traffic
- Somewhat of a learning curve
- Not “cool” anymore – i.e. it isn't a flashy NoSQL solution that you hear developers asking for
- Requires schema to be strictly defined and adhered to
- Not heavily used in the area (OKC metro)

## *Various Flavors*

- MySQL Community (free, from Oracle)
- MySQL Enterprise (paid, from Oracle)
- Percona Server (free, from Percona, branch of upstream)
- MariaDB (free, from MariaDB Foundation, diverging fork)
- WebScaleSQL (free, Facebook/Twitter/others, branch)
- In general, you can use these versions interchangeably, though MariaDB is diverging more

But it is just a silly, free, open source toy...  
Right?



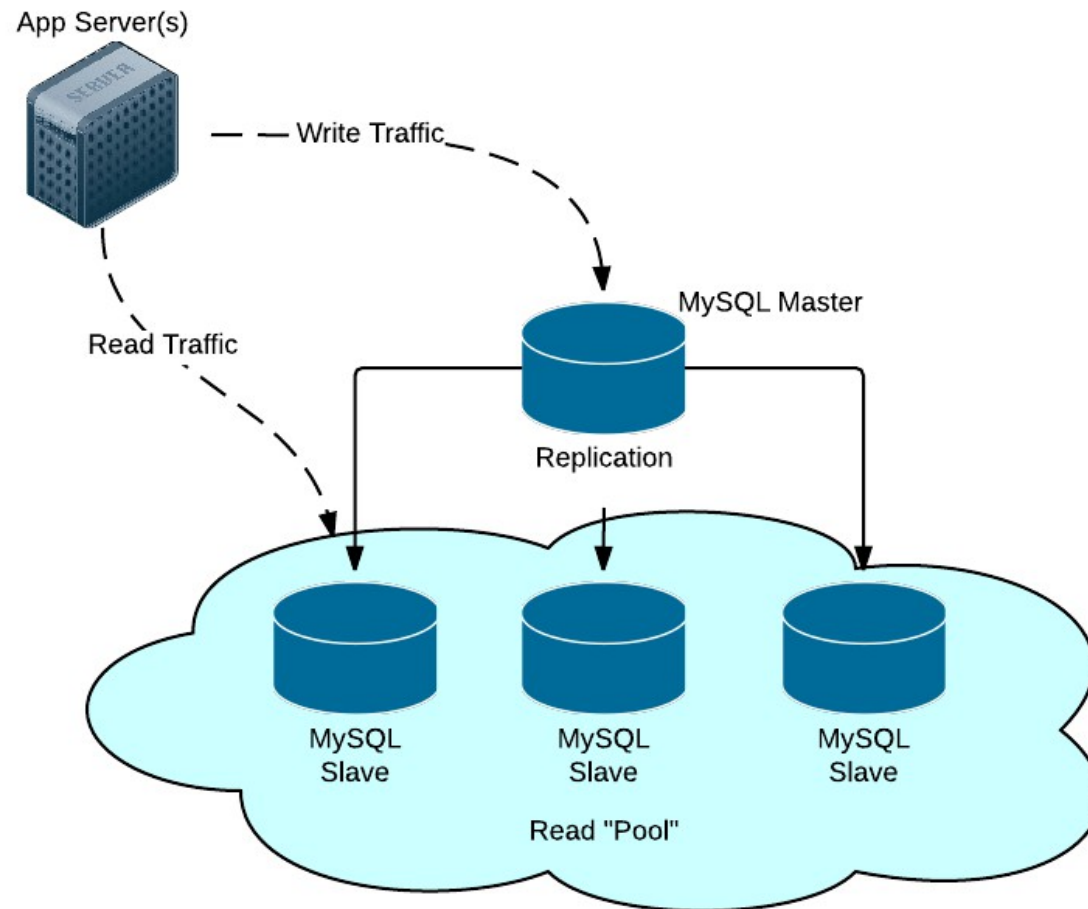
# *Who uses MySQL*

- Some names you may recognize...
  - Facebook (can I stop here?)
  - Twitter
  - Google
  - Amazon
  - Yahoo
  - < Insert other top 100 companies here >

## *Not always a perfect fit*

- Where it works well?
  - OLTP systems
  - Low latency, interactive data stores
  - Relatively small datasets (think 1TB or less)
- Where it struggles?
  - Large, continually growing datasets
  - Full table scan analytics
  - Data warehouse systems (though it is possible)

# *Sample Production Architecture*



# *Questions?*

(I know you have them, that's why you came today...  
so don't be shy)

On to the demo!

# *Demo*

- AWS Windows Instance
- Install MySQL Community Server
- Install MySQL Workbench
- Create a database
- Create a table
- Insert/Select some data
  - From workbench
  - From command line

Open Discussion...

## *Next Meetups*

- Currently scheduled for:
  - Nov 26<sup>th</sup> (day before Thanksgiving)
  - Dec 24<sup>th</sup> (Christmas Eve)
- Possible dates?
  - Nov 17<sup>th</sup> or 18<sup>th</sup> (Mon/Tues) before Thanksgiving
  - Dec 17<sup>th</sup> (Wed), Dec 22<sup>nd</sup> (Mon), Dec 23<sup>rd</sup> (Tues)



*Thanks for coming!*

Website: <http://okcmysql.org>

Twitter: @okcmysql

Email: [mike@okcmysql.org](mailto:mike@okcmysql.org)