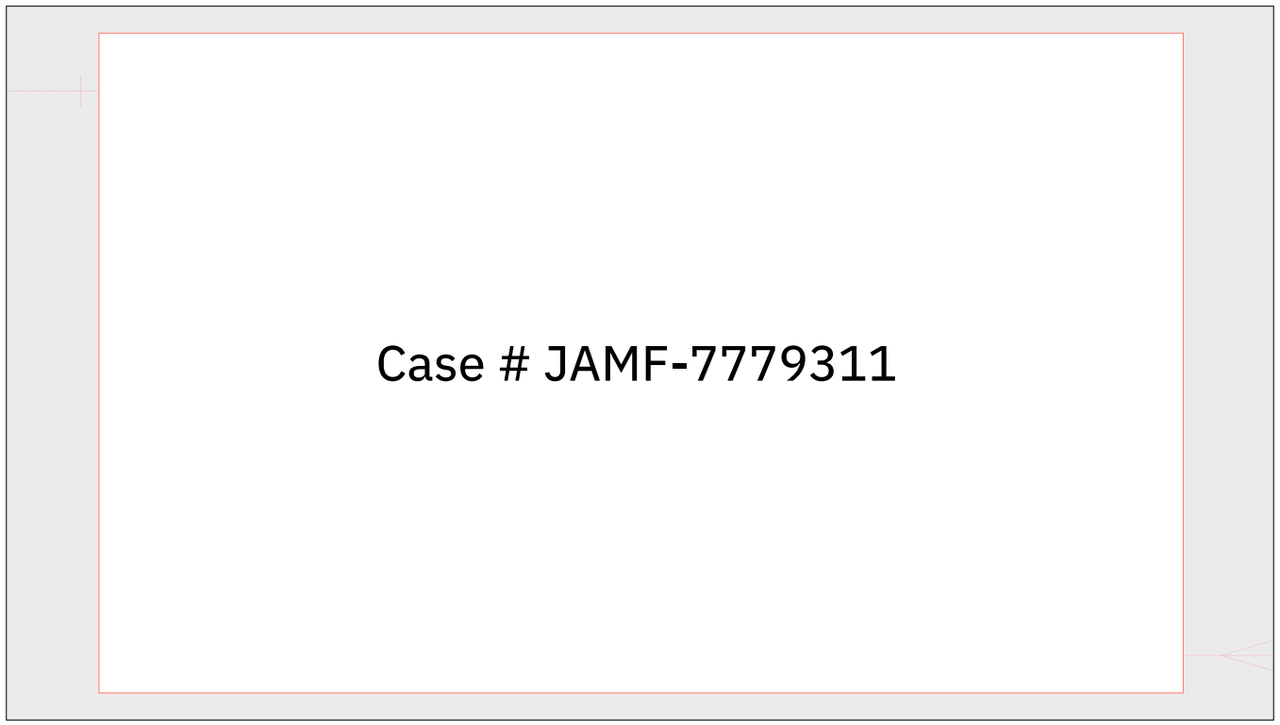
The Joy of Semicolons

Jason Broccardo
Senior Systems Engineer, Glassdoor
@zoocoup on *Twitter*
and *MacAdmins*

- Quick aside
- Come work with my team, our support team or our security team
- Bay Area and Chicago

- This is not a deep dive
- This is a survey or an overview to help you get acclimated with terms & concepts related to SQL and the systems that work with SQL so when you do start your own investigations it’s not all foreign
- I’m a fan of the “See One, Do One, Teach One” maxim



- I got my interest in SQL sparked by running a command that was given to me by Jamf support
- I didn’t understand what it was doing, but it fixed my problem.
- A few articles, books and videos later, here I am

It’s Everywhere

- Like bash, SQL know-how is cross platform.
- Something that speaks SQL exists on the Mac, Linux, Windows, iOS, Android and probably your toaster

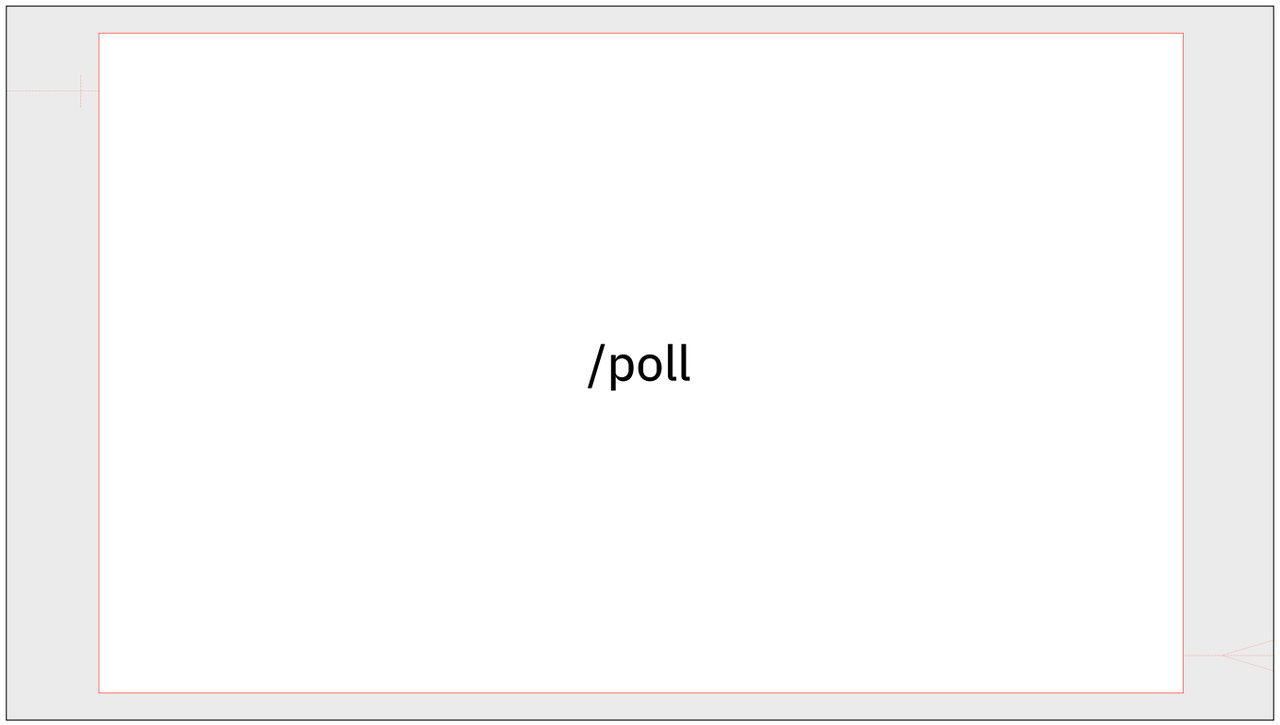
It’s Sort of Timeless

- SQL is a standard (and like many computer things, it was developed at IBM), and though that standard has been updated the past 30 years it’s still largely the same now as it was in 1995
- If you learn to work with it now, that knowledge should last you

~~Database~~ Administrator

- Tinker, Tailor, Admin
- Don’t have to be a DBA to find utility in SQL
- Having it in your toolkit can help you be a better Mac Admin

- Most of the time you will not be configuring SQL or manipulating it. App developers should have done that. But being aware of these items & their impact can help you figure out things you do need to work with like preference settings or gather information about your fleet



- Who here has any experience working with SQL?

It’s Math...

- Under the hood,SQL is relational algerbra and calculus
- Everything is an equation

- You ask the computer a question phrased like a sentence and it will tell you some information

Think HTML, not Python

- SQL is a computer language, but it’s not a programming language
- It is a domain specific language
- SQL’s purpose in life is to talk to databases

- Why databases?
- When you have a large amount of data that you need to access in complex ways, the best thing is a database
- Unlike flat files, a database can hold metadata about it's data, which can contribute to how you use it

Related to what?

- Databases are not spreadsheets
- It’s not 25 columns but five tables, each with multiple columns
- The relationship is how those tables can be connected via common values

Units of Measure

- Databases are made of tables
- Tables are a structure of columns and rows
- Columns are attributes that define a thing
- Rows are occurances of entities
- Entities are person, places, minerals, things about which data exists

ID	TITLE	AUTHOR
1	I Am I Be	K. Mercer
2	The Magic Number	K. Mercer
3	The Bridge is Over	L. Parker

```
select title from books where author = 'mercer'
```

- SQL is sorting through all that structure to get data
- A few concepts before we get into things…

Shenanigans & Schemas

- A schema is the logical structure of the database.
- What’s allowed. Where does it go.
- You’ll likely never touch the schema but just be aware it’s there.

- SQL is built on data types, which are defined when a table is created
- Restriciting data to a particular type helps with sorting, indexing, constructing and verifying
- Let’s database management systems know how to operate on it
- Efficiency

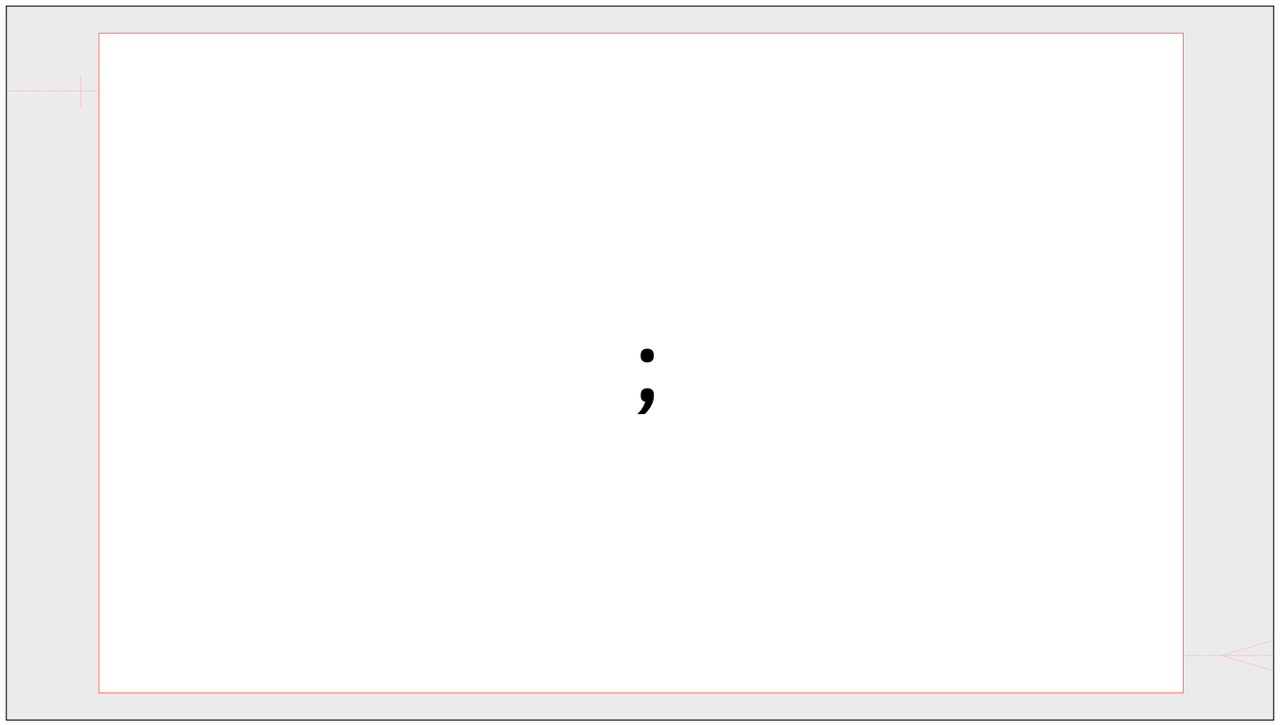
Constraints make the swim lanes

- Constraints are rules in the database that enforce data types.
- No text in your booleans
- Constraints might well be the thing that throws the error when you try to insert items into the database

NULL is not nothing

- Null means nothing found there but it is not the lack of something
- Columns and rows need content of some sort

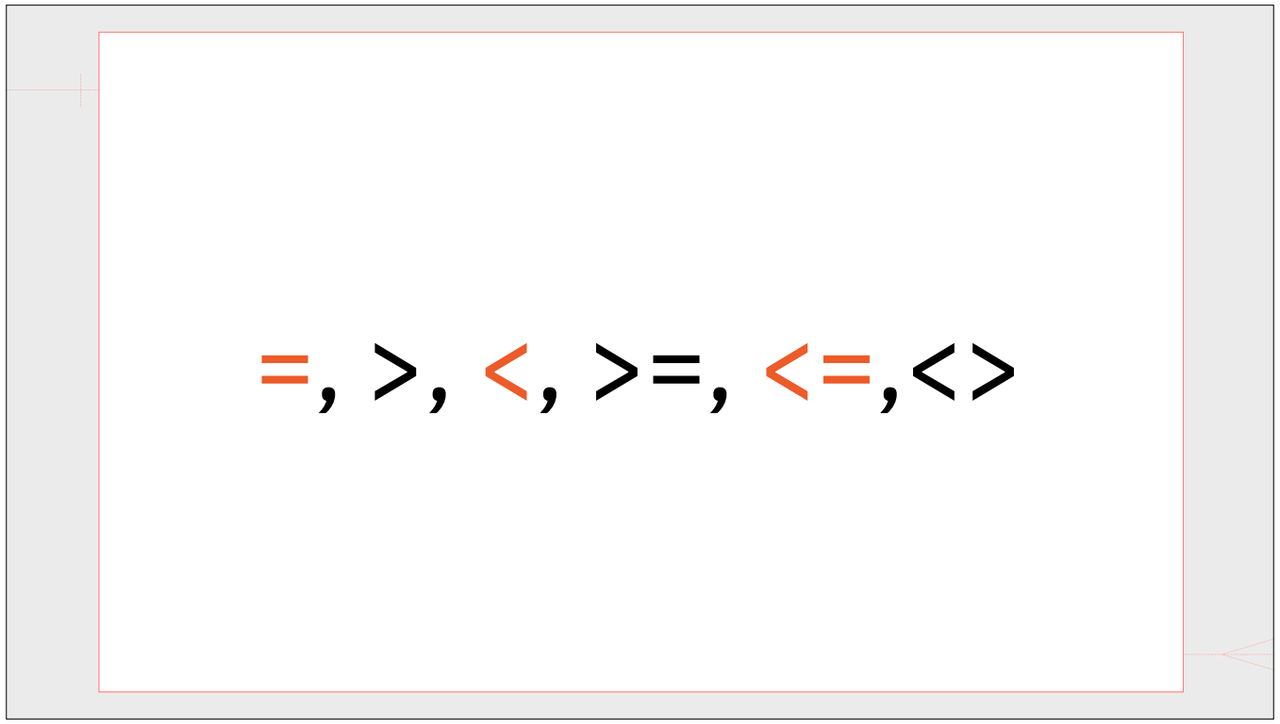
- Many databases contain an Index or multiple indexes
- Indexes take up disk space, because they are a physical file, but they should greatly increase the efficiency of the databases
- Indexes create a lookup of a single column. Then, rather than searching a 1000 rows, the database instead searches just the entries in the index and uses that to quickly find the needed rows



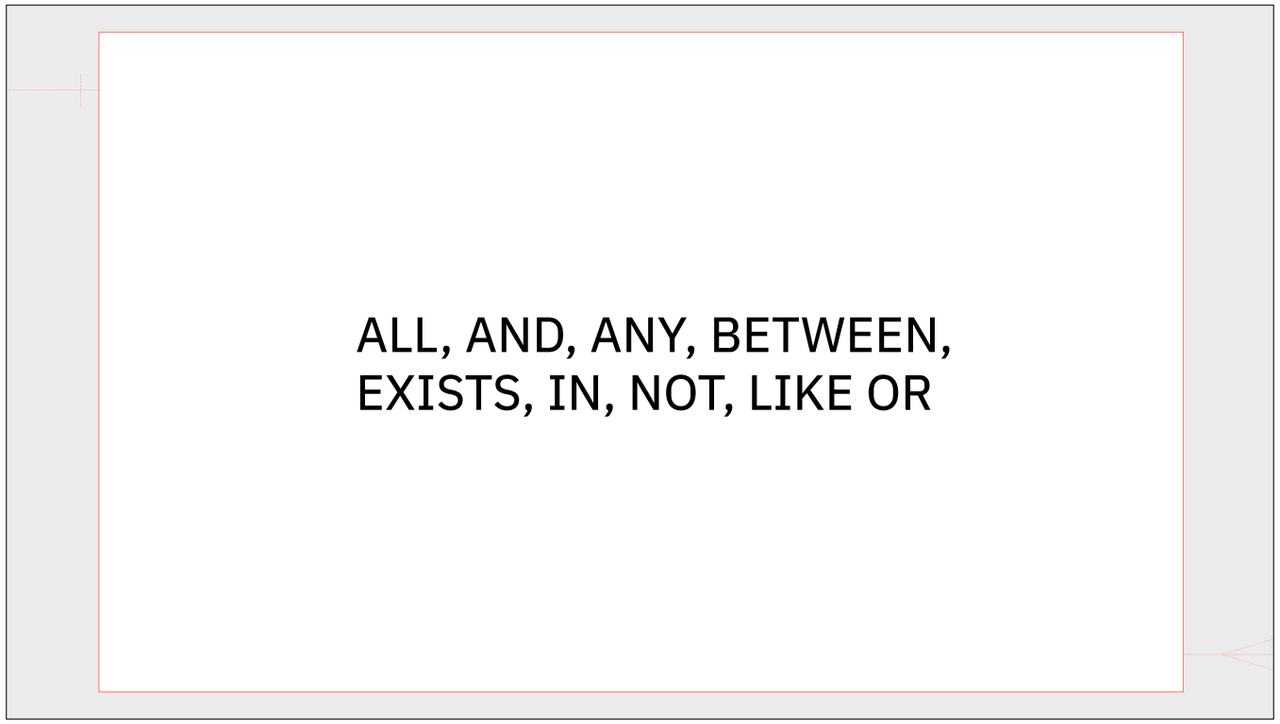
- Maybe the most important thing in SQL
- You can have all the whitespace in the world till this terminator stops the entry and allows execution

- Select, create, insert, update, delete, alter, drop: these are all SQL commands
- Select is the one you will use most often

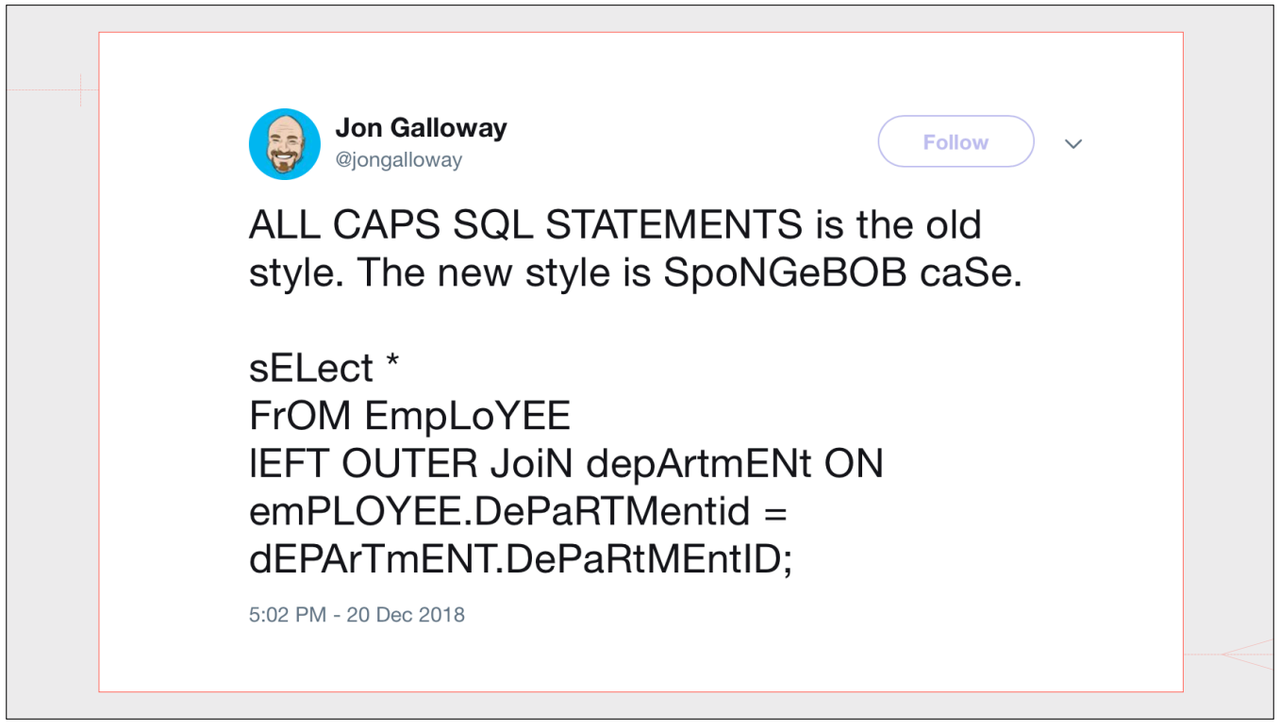
- This is the funnel. Where is the keyword that starts the information sort



- These are the comparison operators. Equal, greater than, less than, greater than or equal to, less than or equal to and not equal
- They aren’t just for numbers. Will work on all data types, not just integers.



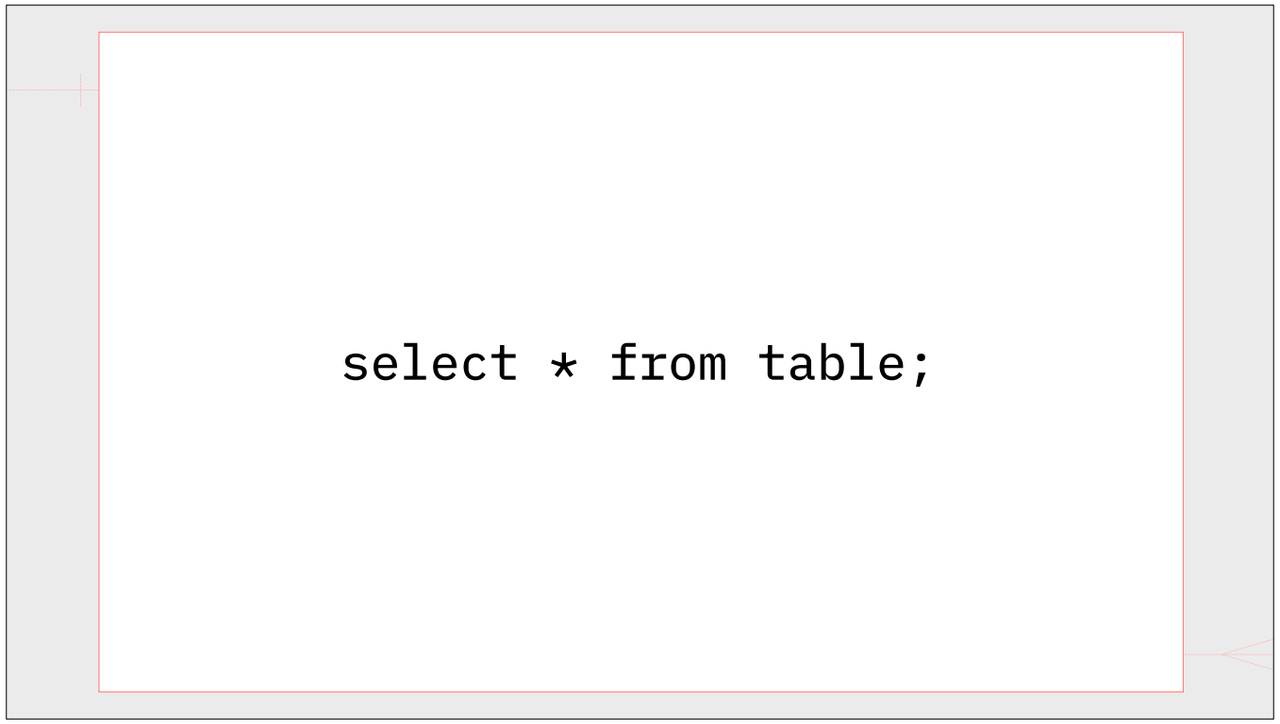
- These are logical operators
- They help narrow down or direct the search



- You don’t need to type everything in ALL Caps when working with SQL
- It’s a convention and can be helpful, but not manadatory

```
command column from table
where column = criteria
```

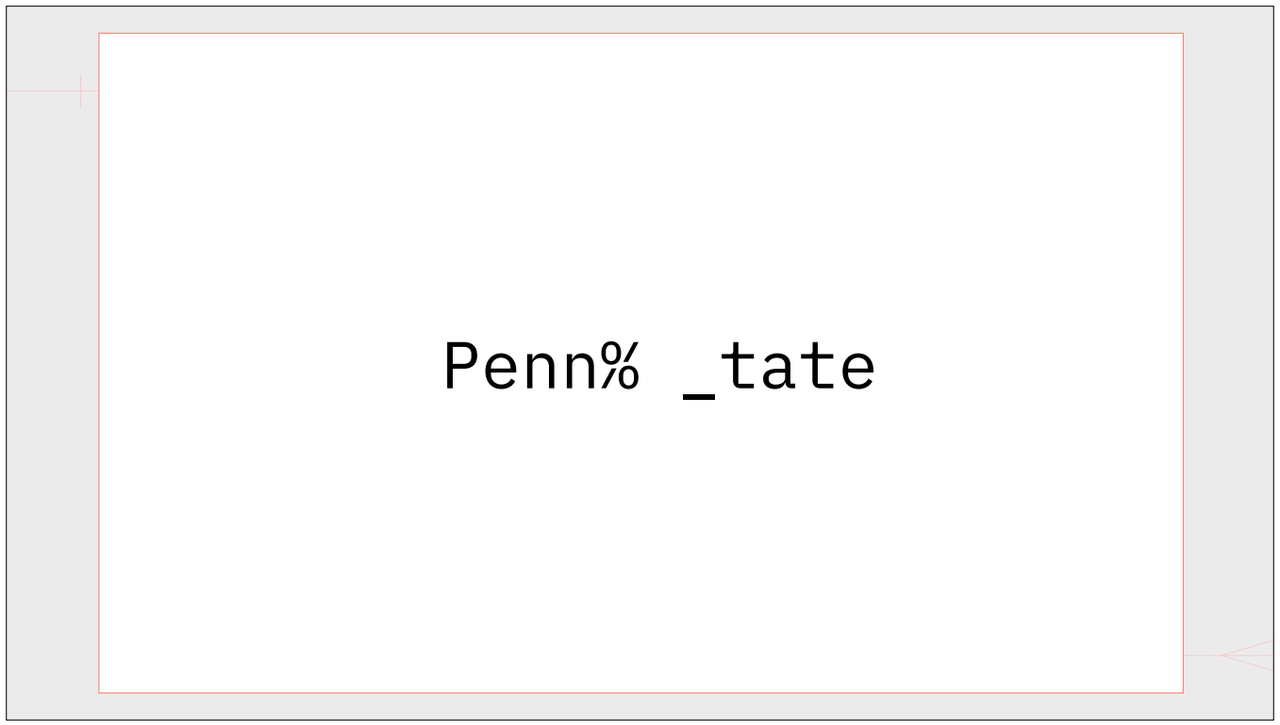
- Most everything you do in SQL will look like this.
- Queries can get complex, but this is the most basic form



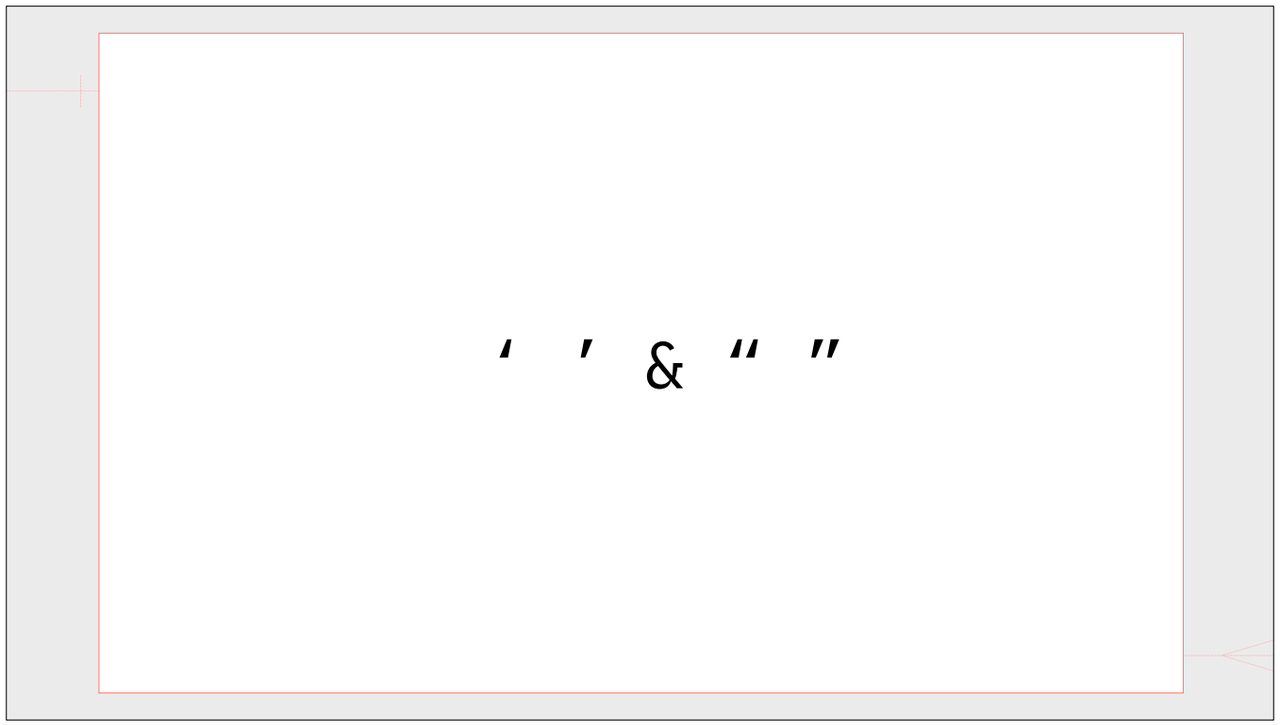
- This is a perfectly valid query. You will get results, but given a database of enough size you’ll never make your way through it
- This is what you can start building with though

```
select first_name,
last_name from table;
```

- SQL will allow you to connect together multiple values using just a comma
- This works with all sorts of commands and actions



- SQL has wildcard characters
- %: any number of characters
- _: just one character



- Punctuation matters. Different characters have meaning
- Single quotes are used to signify a value
- Double quotes can be used to protect things like file paths
- Double Quotes can also be used to insert a blank value (but not null)

- SQL has reserved words and characters. If your data has one of the protected items, such as an forward slash, you’ll need to escape that with quotes or using \

```
select title from bookList
where (publishDate > 2008)
and (authorName like 'Dwyer')
```

- You can combine criteria using parentheses and a conjunction like and or or

```
select title from bookList
where publishDate > 2008
order by authorName
```

- With Order By, DESC and ASC options you can arrange how the returns will be presented.
- Ordering even works with multiple columns (order by this, that
- Ascending order is the default

```
select distinct author
from books where
publishDate > 2008
limit by 10
```

- Distinct will cut out any duplicates that might be present in the return
- Limit will is like head in bash. It takes just the top X amount of the return
- These are filters to help get more concise results

Fully Functional

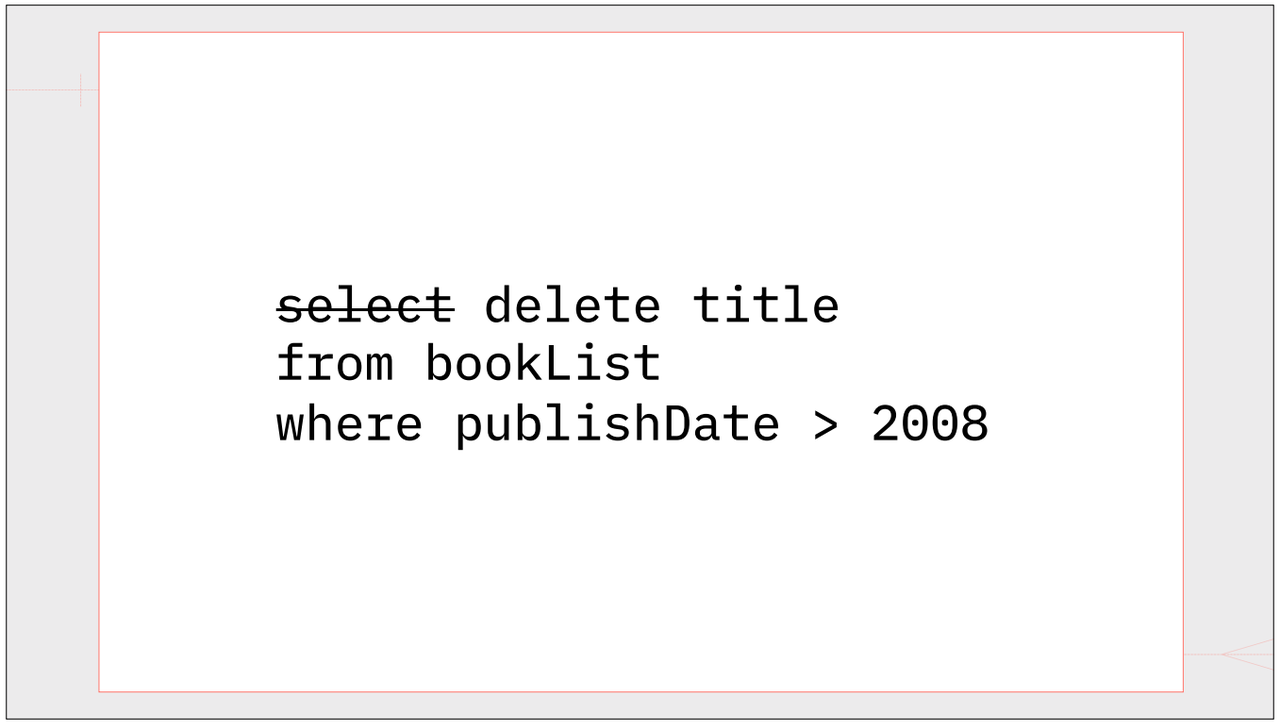
- Aside from returning data results, SQL can also return metadata of sorts
- Counts
- Sums
- Averages
- You can even concatenate columns together to create a combined return

```
update table
set column = ‘value’
where criteria
```

- Update and set always go together
- Update modifies data in a column not across a row.
- This is how you can change data without having to remove it and reenter it

```
insert into
table(columnA,columnB)
values(‘key1’,’key2)
```

- You have nothing to select from if you don’t enter data in the first place
- This is a basic insert statement, but you can have a complex statement construction out of this that inserts hundreds or thousand of rows
- If you don’t list the columns (in parens) then the values you provide must line up in count. Use “” to fill the space
- Single quote all value entries



- As mentioned, select statements will likely be the most used interaction with databases via SQL but sometimes you need to remove data. SQL makes it easy by using the same syntax for a delete statement as you do for a select statement
- You can use a select statement to double-check things before you committing to removing data

- We’re concerned mostly with gathering information. Alter and Drop are ways to make wholesale changes to a database that we’ll not be covering today.

Keys to the kingdom

- Primary key provide a way to uniquely identify a row in a single column and a way to 0identify data from one column to another (think ID number)
- Foreign Key is another table’s primary key
- Keys matter because they are the connective tissue between tables

dot.notation

- When referencing columns from multiple tables in the same statement you will need to fully qualify the name.
- It’s always table dot column

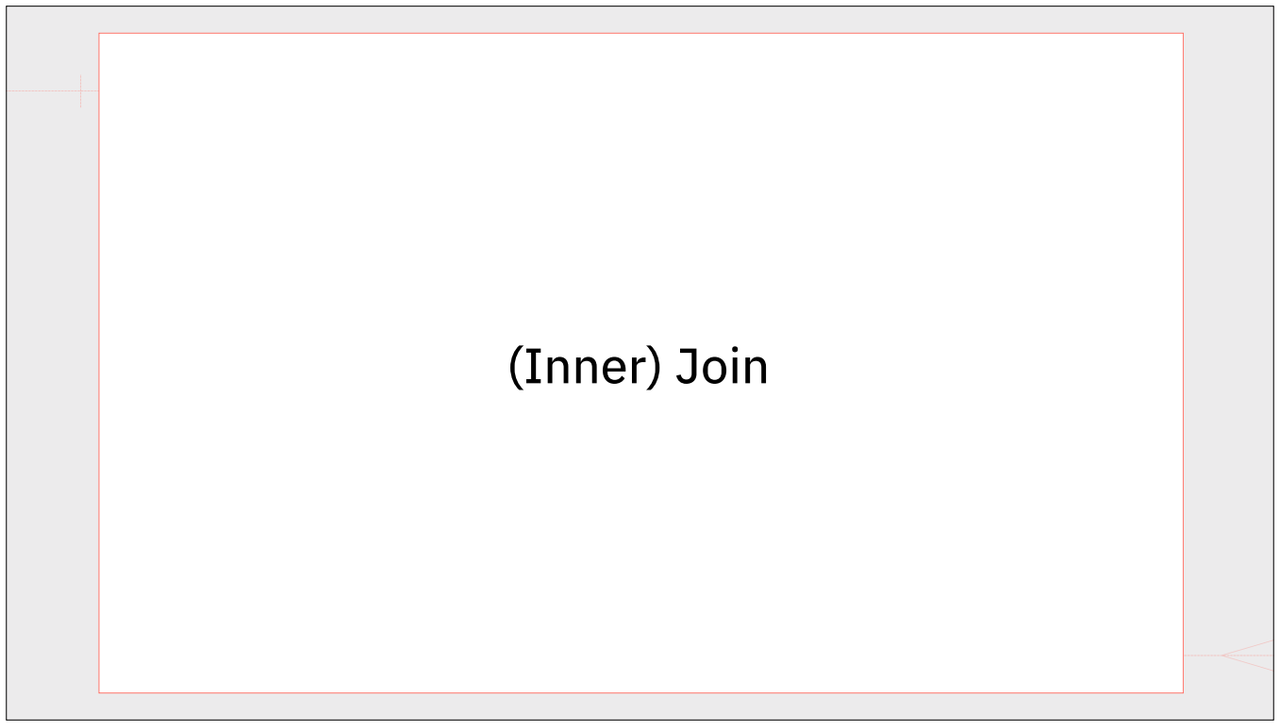
A(lia)s

- AS is Alias
- Allows you a way to manipulate column and table names without making permanent changes
- Ease of use, brevity, obfuscation

Queries all the way down

- Subqueries are like joins
- Order of operations (inside) outside
- Inner query must be valid for the whole query to take place
- The inside query helps define what will be the search criteria for the outside query
- subquery selects can only return a single column

- Joins are the embodiement of “relational” in relational databases
- Joins link together tables in a database a common column
- This connection is made between the primary and foreign keys
- Joins are not permanent. They exist only at the time the query is run.



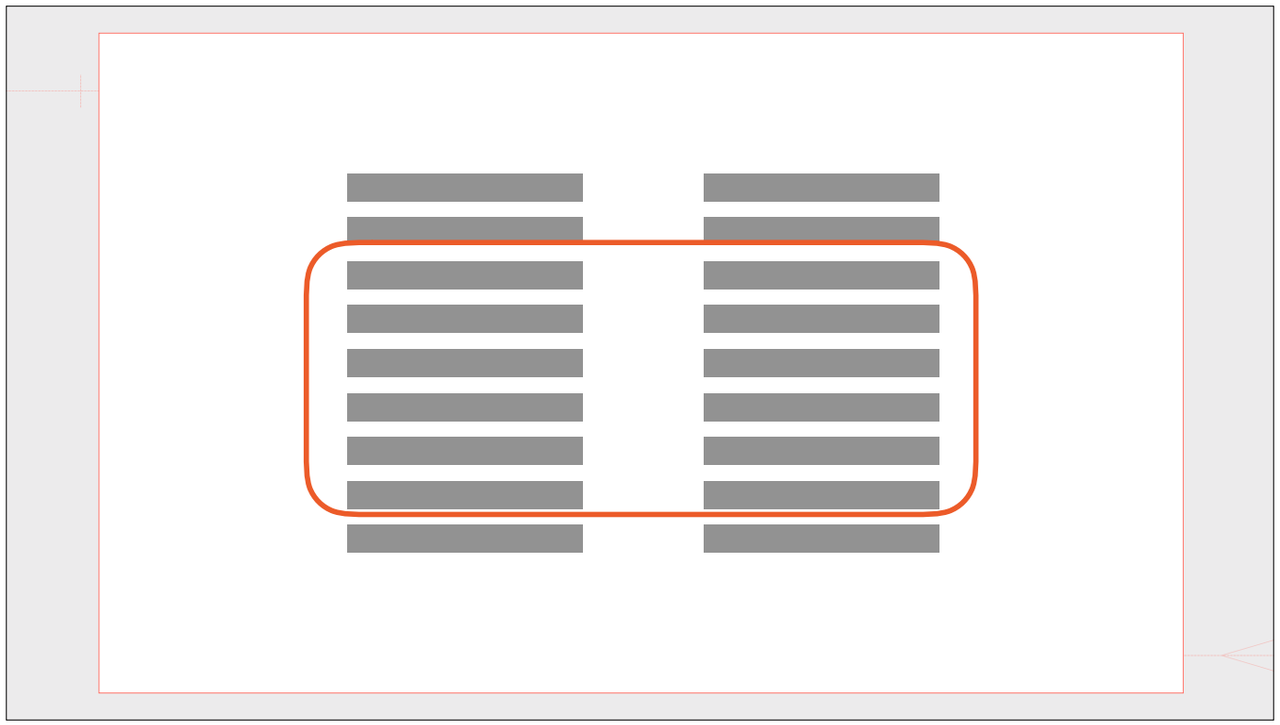
- Join and inner join are the same thing. This is the most common form of a join you’ll likely use

```
select column from tableA
JOIN tableB
on tableA.column = tableB.column
```

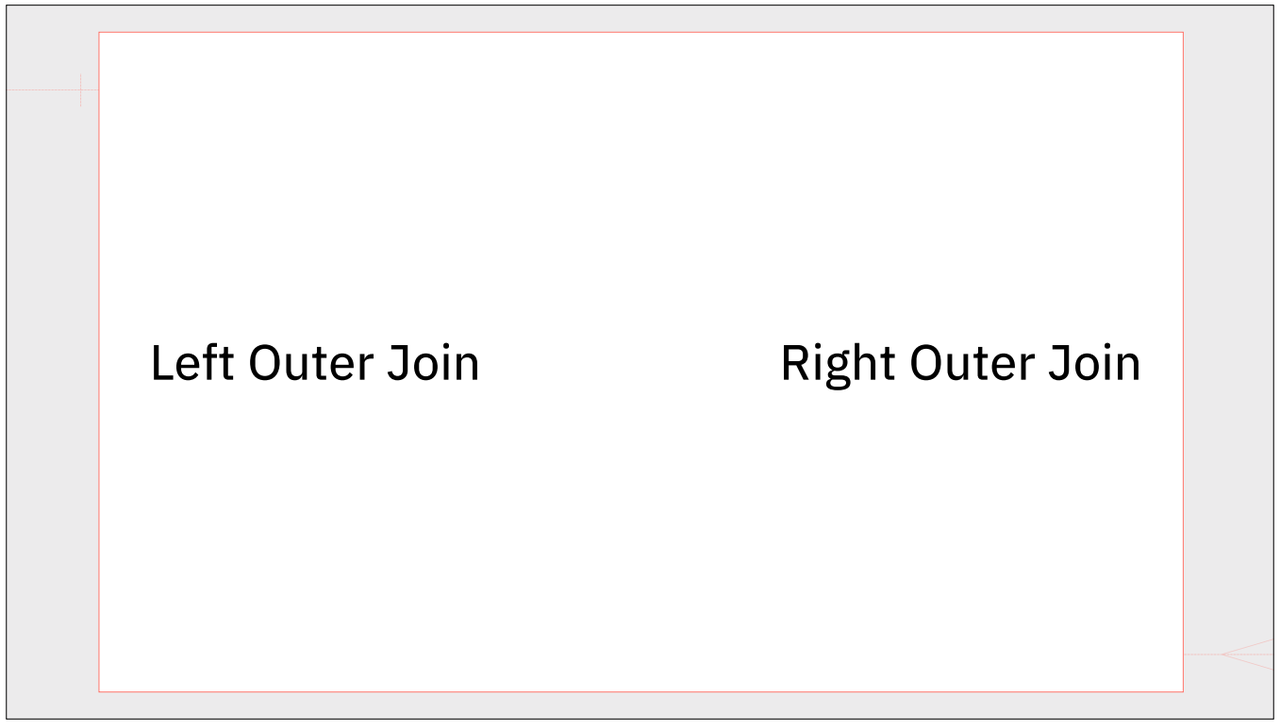
- This is the syntax for an inner join
- It is possible to have multiple joins

```
select admins.name,admins.company from admins
JOIN speakers on admins.id = speakers.id
where admins.name like'%GregEagle%;
```

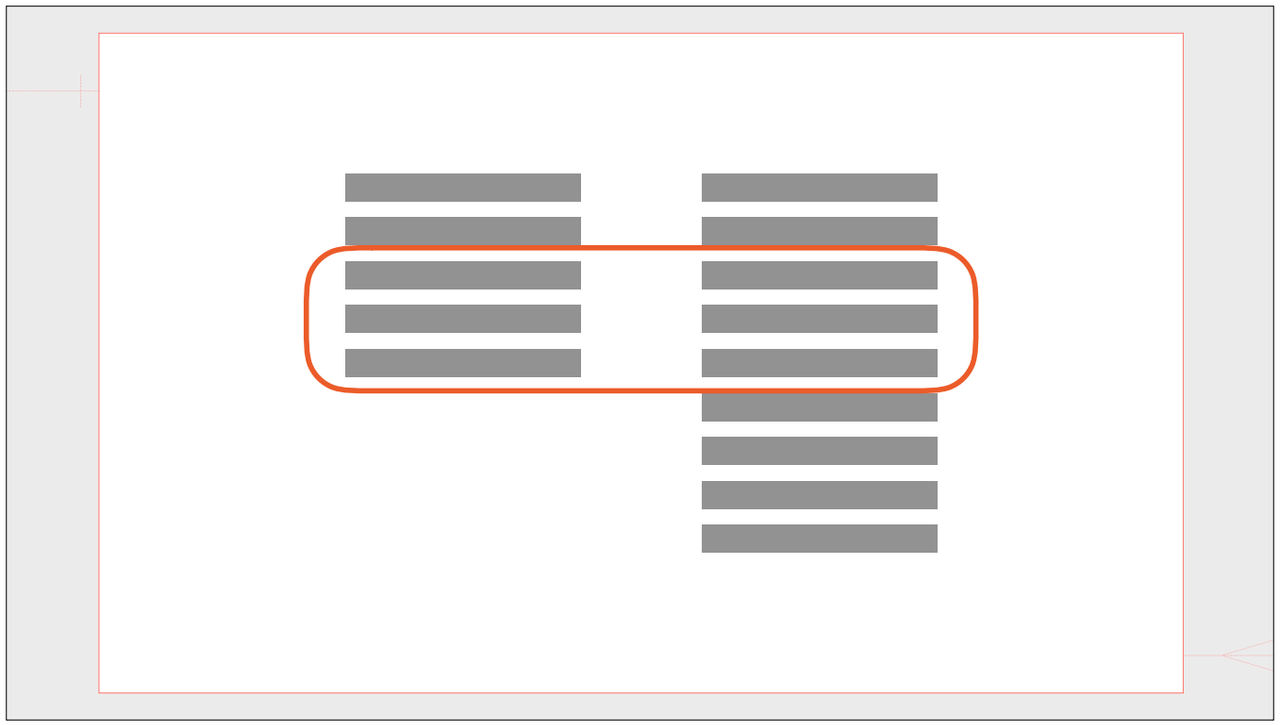
- Here's another view.
- Note the use of the fully qualified names to call the proper tables in the database
-



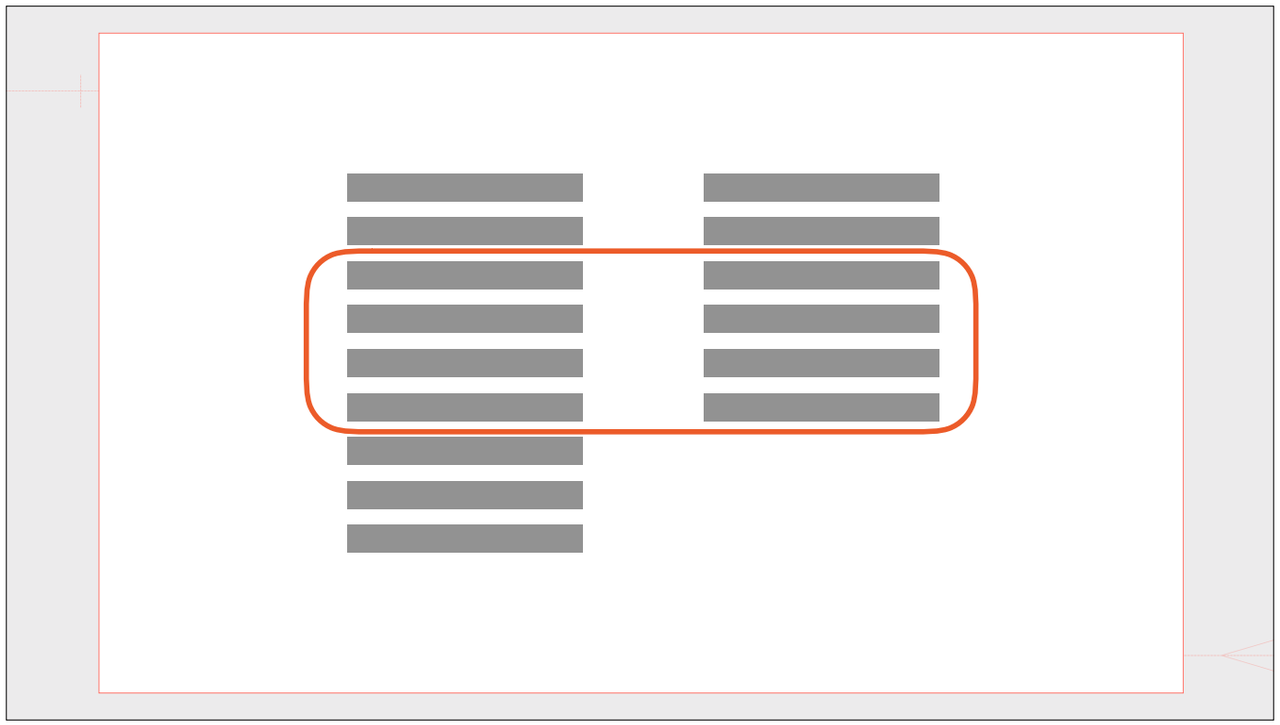
- Roughly what an inner join looks like. All the parts of the first table that match exactly with the second table are included in the returned result



- There are other kinds of joins, including left outer and right outer joins
- These two come into play when you need to get data out of more than two tables
- I consistantly find ways to screw up outer joins and wish you good luck



- Called right for the table named second in the query statement
- Right outer joins should return all rows in the righthand table along with any rows in the lefthand table that match



- Left for the lefthand table, aka the first table named in the join clause
- Should return all rows in the lefthand table along with any rows in the righthand table that match

ACID Test

- This is gets further into the realm of database design than we're going to go but it is worth mentioning the concept of ACID.
- Transactions, for a database, are a set of instructions that all occur together or don't occur at all. You trust that the thing happened in the way attended.
- ACID compliance is a way for a database to say transactions are happening as intended.
- Atomicity. Consistency. Isolation. Durability

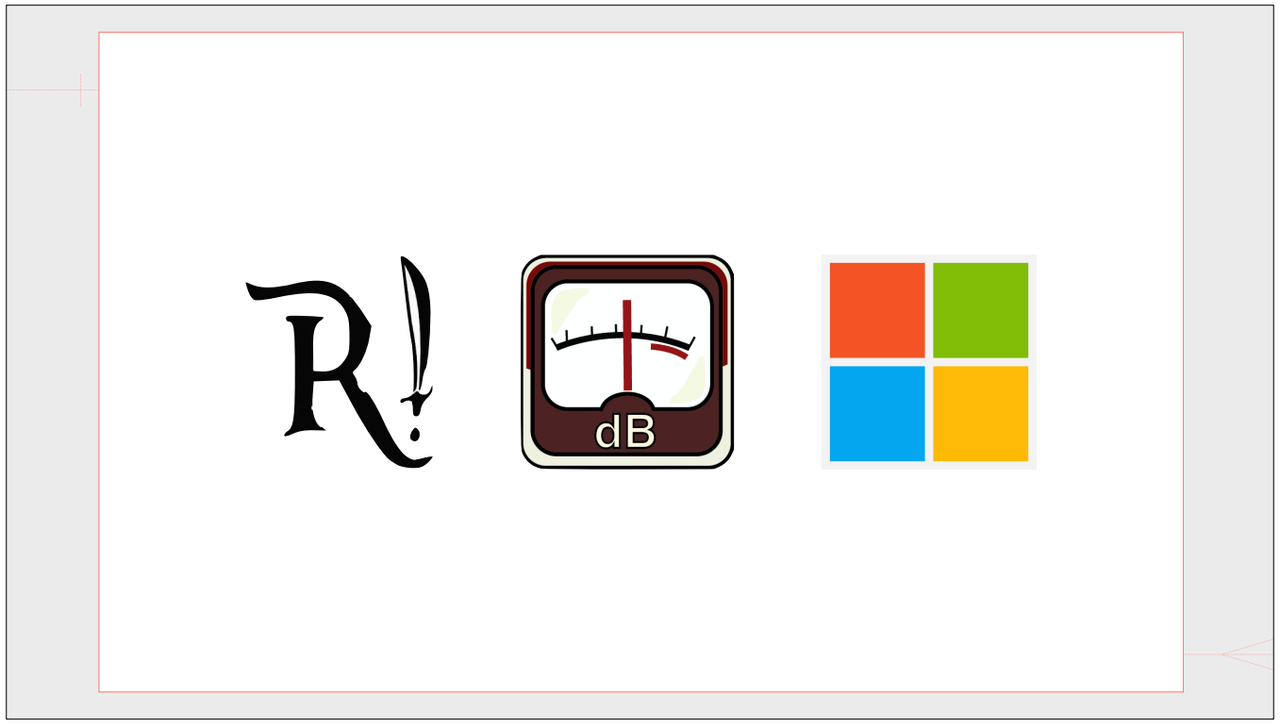
Nothing is Normal

- Also a part of database design is the concept of normalization.
- A database can be in different stages of normaliness, each building off the level that came before it.
- Normalization is about having data organized in the database in the most efficient way possible that optimizes searchability

- 1. SQL Basics Section Demo.txt

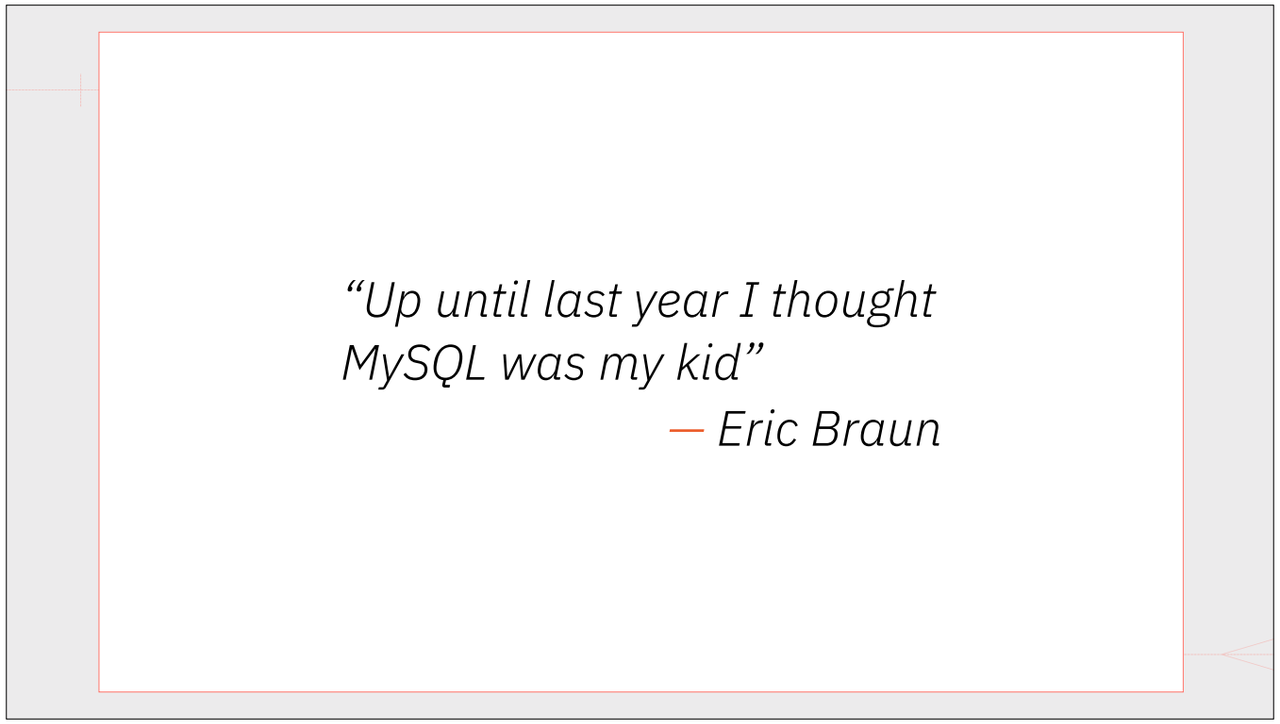
Mid-point Questions?

- Before we proceed, any questions about anything that has come up thus far?



- RDBMS: Relational Database Management System
- RDBMSes define the structure for how data is stored and indexed in a database
- Govern how data is interacted with, including ingest and export
- Control access to the data
- Provide an interface of some sort to the data
- Bit of a stretch but if SQL is a standard like HTML is a standard, RDBMS relationship to SQL is akin to the relationships of web browsers to HTML

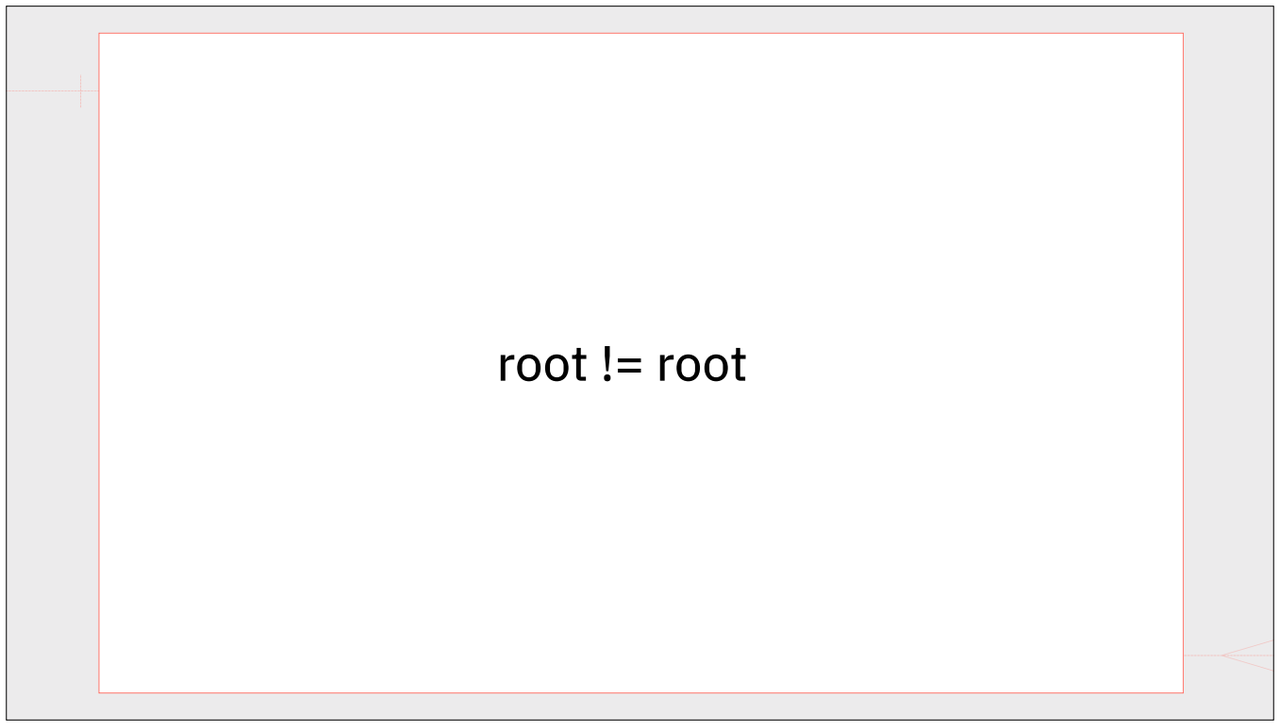
- There are many RDBMSes. We're only going to discuss MySQL and SQlite in this talk
- If you use Universal Type Server or Sal, those are likely using Postgres
- Be aware that MSSQL uses it's own variant of SQL called T-SQL. MSSQL has some different datatypes and commands from other SQL services like MySQL



- MySQL runs on pretty much everything. Linux, Mac, Windows
- Many applications depend on MySQL as their backend
- MySQL, like almost all of the RDBMSes aside from SQlite, operates in a client-server fashion.
- MySQL is still open source despite being purchased several years ago by Oracle. After the purchase several forks were spun off, including MariaDB and Drizzle. Depending on the use case, you should be able to drop one of these forks down in place of MySQL and not need to make any application changes.

```
103   ??  Ss    59:35.13 /usr/local/mysql/bin/
mysqld --user=_mysql --basedir=/usr/local/
mysql --datadir=/usr/local/mysql/data --
plugin-dir=/usr/local/mysql/lib/plugin --log-
e r r o r = / u s r / l o c a l / m y s q l / d a t a /
mysqld.local.err --pid-file=/usr/local/mysql/
data/mysqld.local.pid --keyring-file-data=/
usr/local/mysql/keyring/keyring --early-
plugin-load=keyring_file=keyring_file.so
```

- Though MySQL will run on anything, running well on anything is another matter. Dpending on the number and size of databases being served and the number of current clients connecting to the service), you might need to increase hardware resources
- CPU matters (remember, there's a lot of math involved behind the scenes)
- Memory allocations can make a difference
- Disk read and write speeds can be a significant factor



- MySQL maintains its own users
- MySQL's root user is not the system's root user
- Users can be local (@localhost) or remote (@IP)

- Grants are MySQL's permissions scheme
- Grants define what a MySQL user account has access to and what actions they can perform
- Grants can be broad (access to everything) or granular (select access to a set of defined tables only)

Engine Engine Number 9

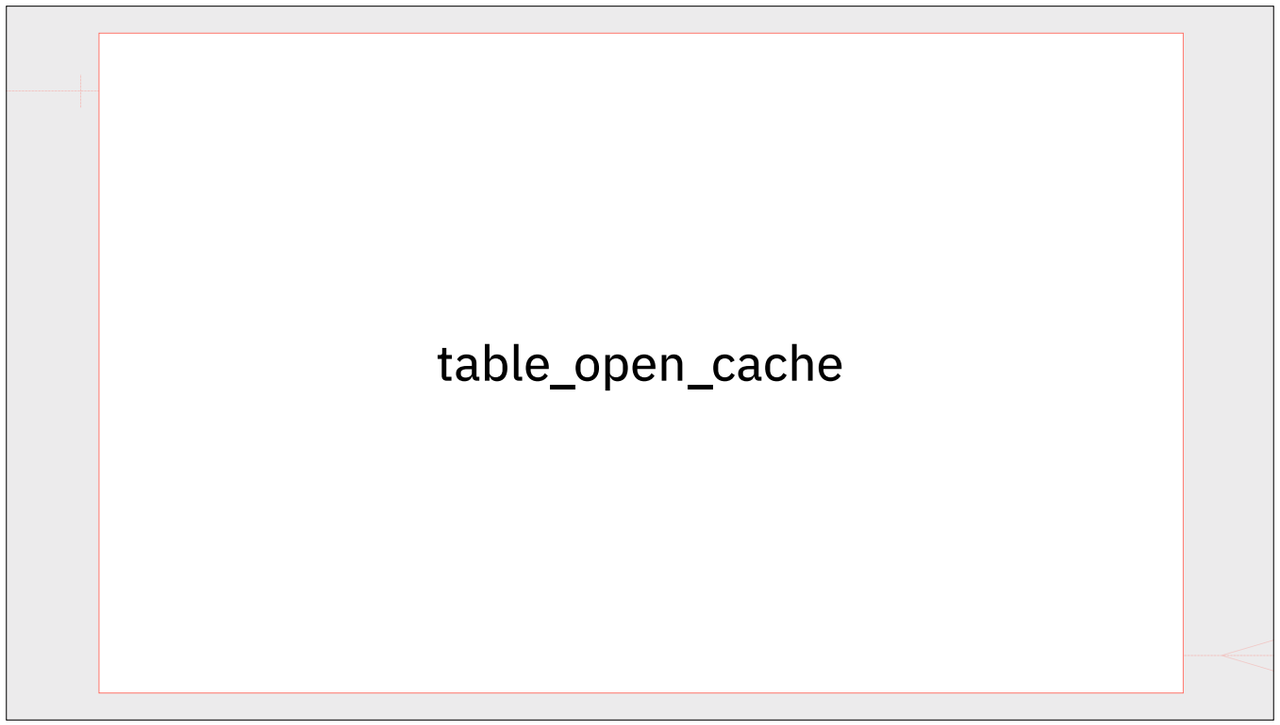
- Storage Engines are a component of MySQL that govern how data is managed and store by the database. Storage Engines are kind of akin to a file system for a database. Sort of.
- Storage engines are responsible for things like file and table locking, how files are physically written to disk and how indexing is handled.
- There are several different storage engines, but the most common are MyiSAM, which was previously the default, and InnoDB, which has been the default for the past few years.
- For the Jamf admins who run on-prem, Jamf Pro 10.6 requiring InnoDB should help database performance going forward. If anything, it gets Jamf's MySQL requirements up-to-date to where MySQL development is at in 2019

/usr/local/mysql/bin/mysqladmin

- MySQL includes an administrative tool called mysqladmin.
- mysqladmin can be used to run CLI instructions against MySQL to administer the service and databases.
- Depending on the circumstances, mysqladmin will likely be involved if you have to troubleshoot an issue with mysql

- my.cnf is mysql's primary configuration file. Depending on which OS you are running it on, the default location will change. /etc is not a bad place to start (note the v8 on the Mac does not enable the traditional my.cnf by default)
- There are numerous options that can be set in the my.cnf file. I wanted to highlight a few that might be relevant to Jamf Admins. These only matter if you are hosting on-prem. Jamf Cloud locks you completely out of any database interactions, let alone the service configuration
-

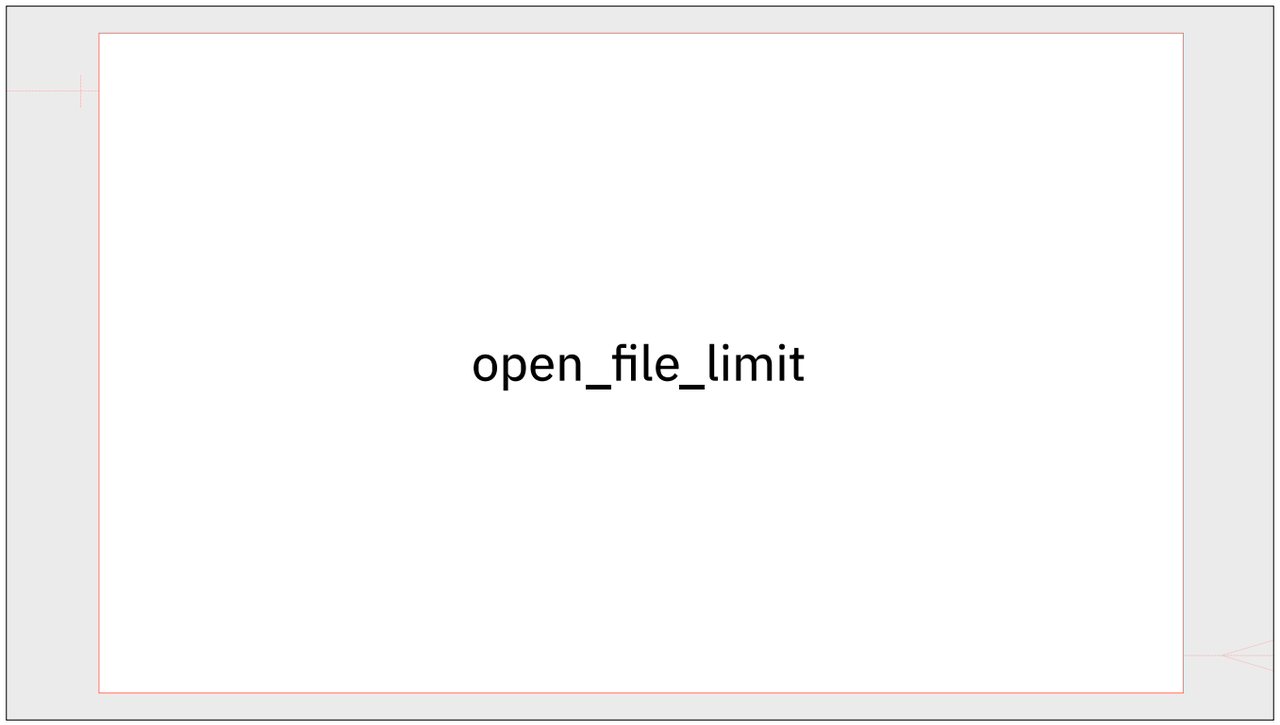
- How many incoming connections to the mysql database will be supported.
- Rule of thumb is set this number to (# of Web Apps x maxpool) +1
- maxpool is a Tomcat variable. The recommended default is 90
- Plus +1 is needed to provide for at least one open connection the event things go haywire and a app or process monopolizes the connections.



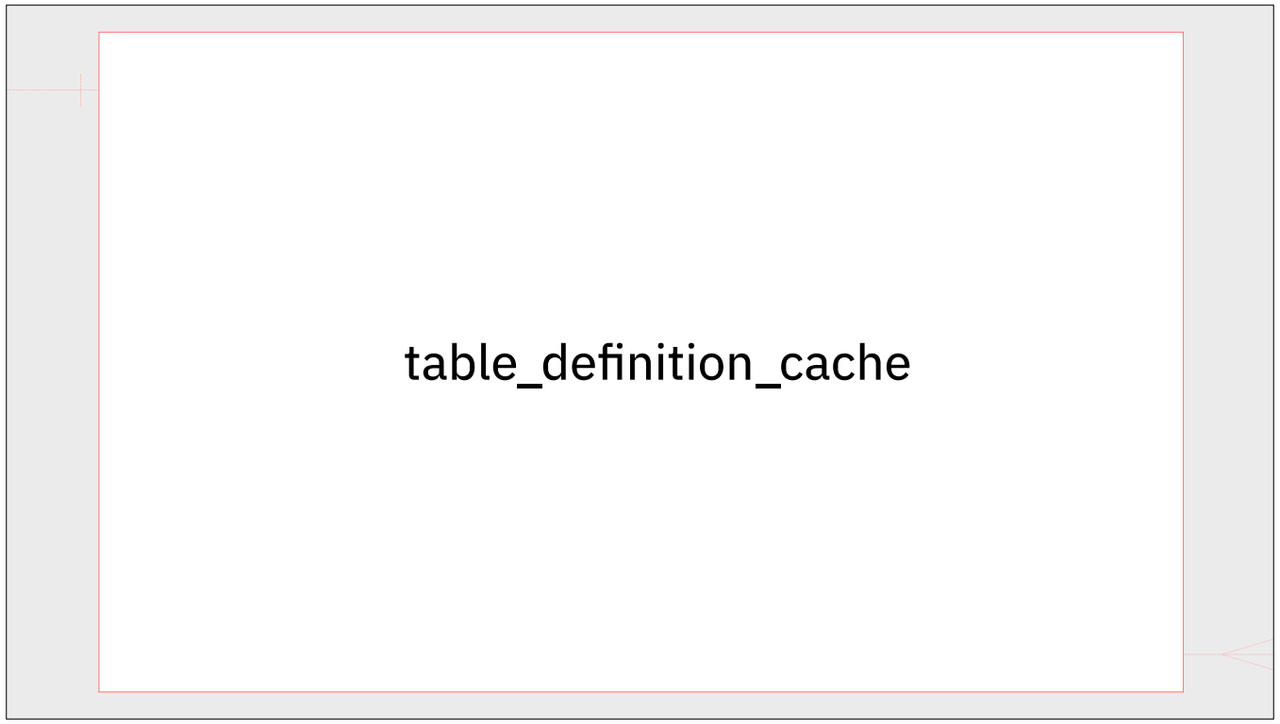
- This is how many open tables – but not OPENED, that's a different measure that increases over time – will be allowed
- I've can't find a hard and fast recommendation for this relative to hosting a Tomcat webapp like Jamf. It's a bit of a guestimate.
- This might be something set after running your server during a “burn in time”; see how things are going, then make adjustments
- The value of table_open_cache impacts several other settings, so you need to start here.

1. *max_connections* x (maximum number of tables per join in any query)
2. *max_connections* x 10 (< 10,000)
3. Calculate “hit rate” for 50% rate: (*open tables* x 100)/*opened tables*

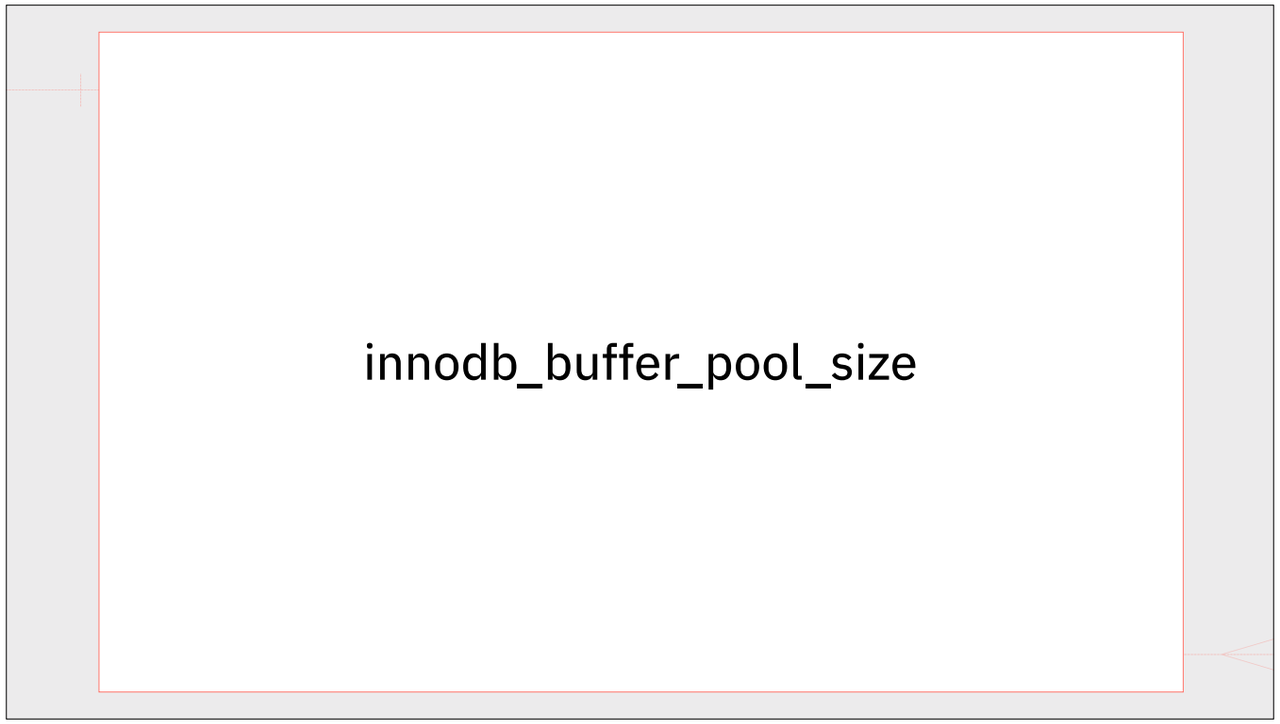
- #1 is from the official docs, but who knows how many joins will be in any query coming out of the JPS, especially when you consider jamf v10 has about 430 tables
- #3 requires observing your database at different of load to get an average. Maybe start with #2 and then figure out #3
- I'll end with a link to a google doc of links, including a cheatsheet I put together that has the formulas mentioned here



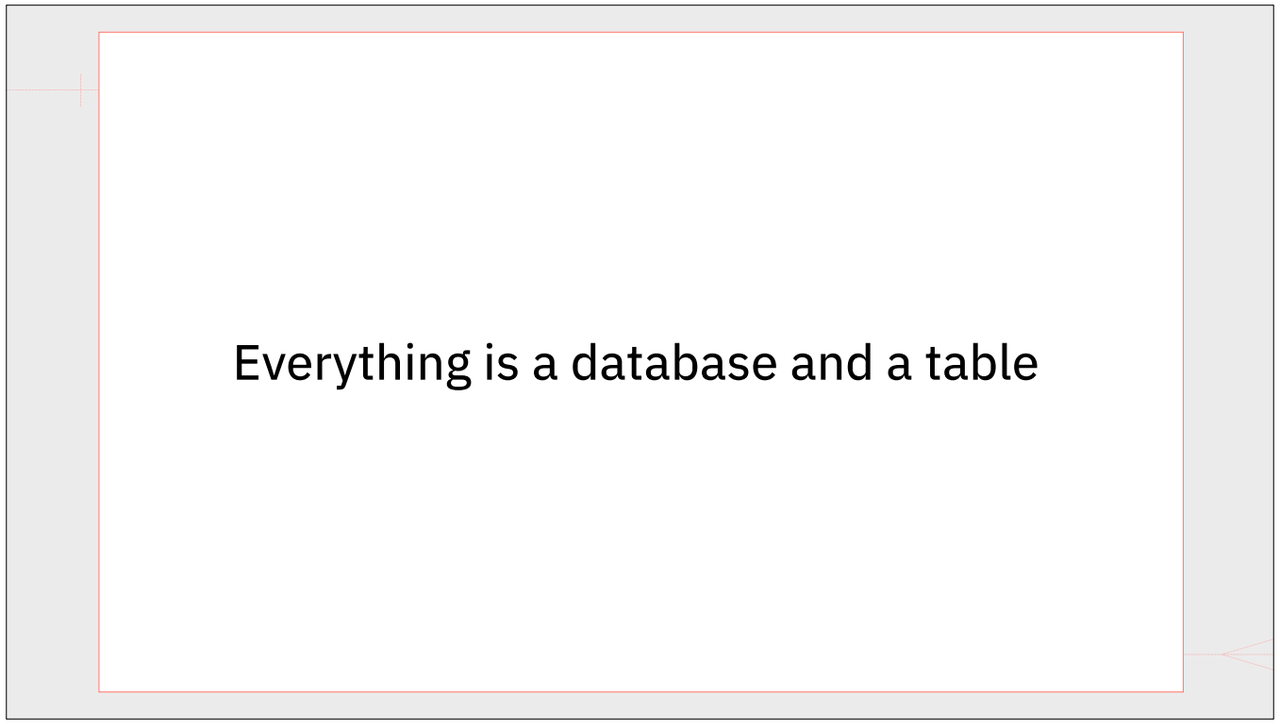
- Multiple your value for table_open_cache x 2



- Setting this can help speed up the opening of tables.
- (table_open_cache/2) + 400



- This sets how much memory InnoDB can use for the in-memory cache
- You want to give InnoDB as much memory as you can within reason
- Rule of thumb is to set this to 80% of available memory AFTER subtracting for memory to be used by the system, memory used by other service running on the box — including mysql itself — and a bit of overhead.



- MySQL really goes all in on this whole database concept. Every MySQL install has databases dedicated to the service itself, and you can query these databases to get performance and configuration data.
- User information
- Running config
- Performance specs
- All this can be harvested from the mysql and sys databases

- 2. MySQL Demo.txt

Apple OSes run on SQlite

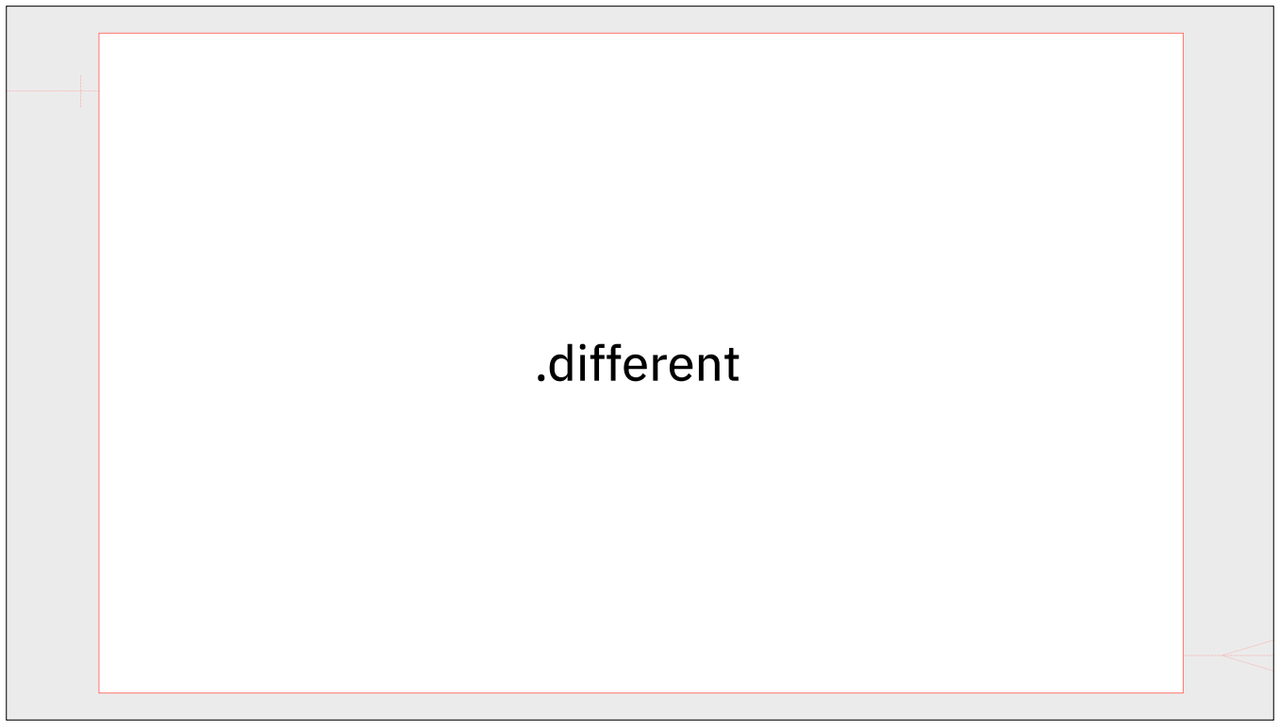
- This is only a slight exaggeration
- sqlite databases are everywhere in macOS and iOS (presumedly the tvOS and watchOS too)
- I count over 1600 in my user home folder alone

Why?

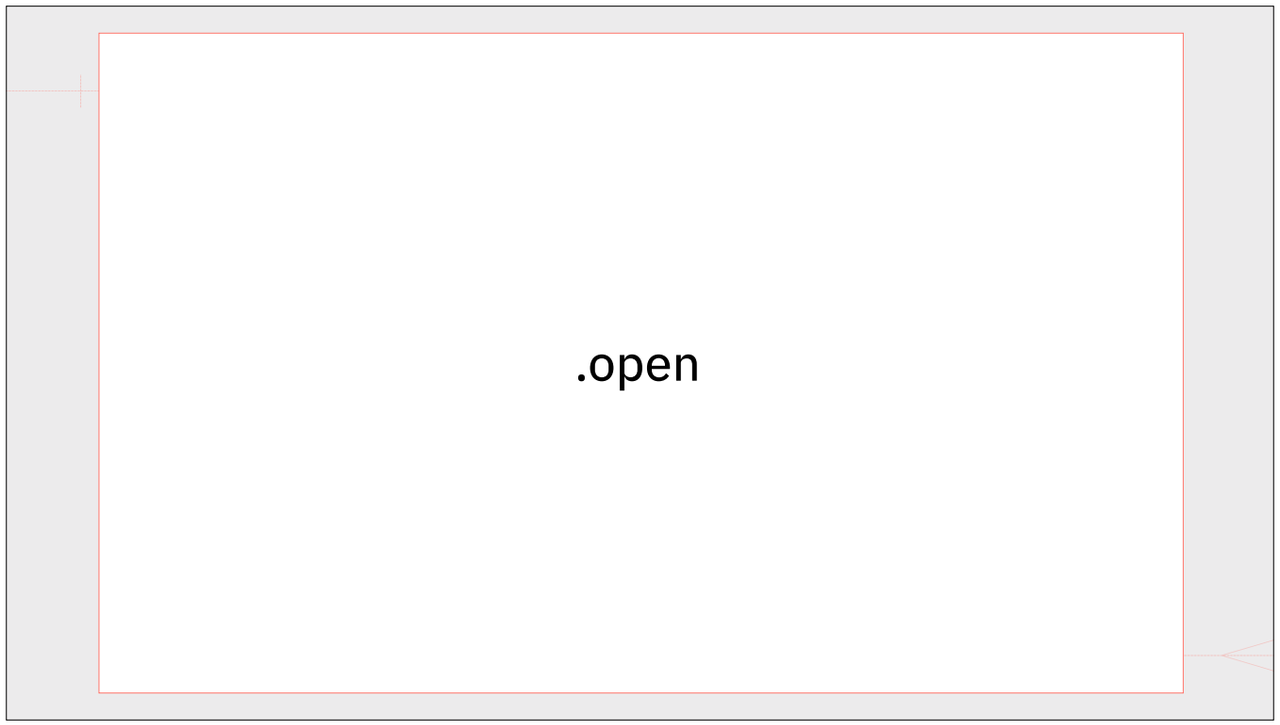
- Recall what databases are good for: When you have a large amount of data that you need to access in complex ways, the best thing is a database
- Plus that data can be indexed for quick returns
- These databases are constantly being updated, hold different data types beyond just key/value pairs. It's more efficient.
- Compare it with a flat file like the PLIST. Think about the system trying to constantly update key-value pairs in plist or quickly look up a needed value out of 200 listed

But why sqlite?

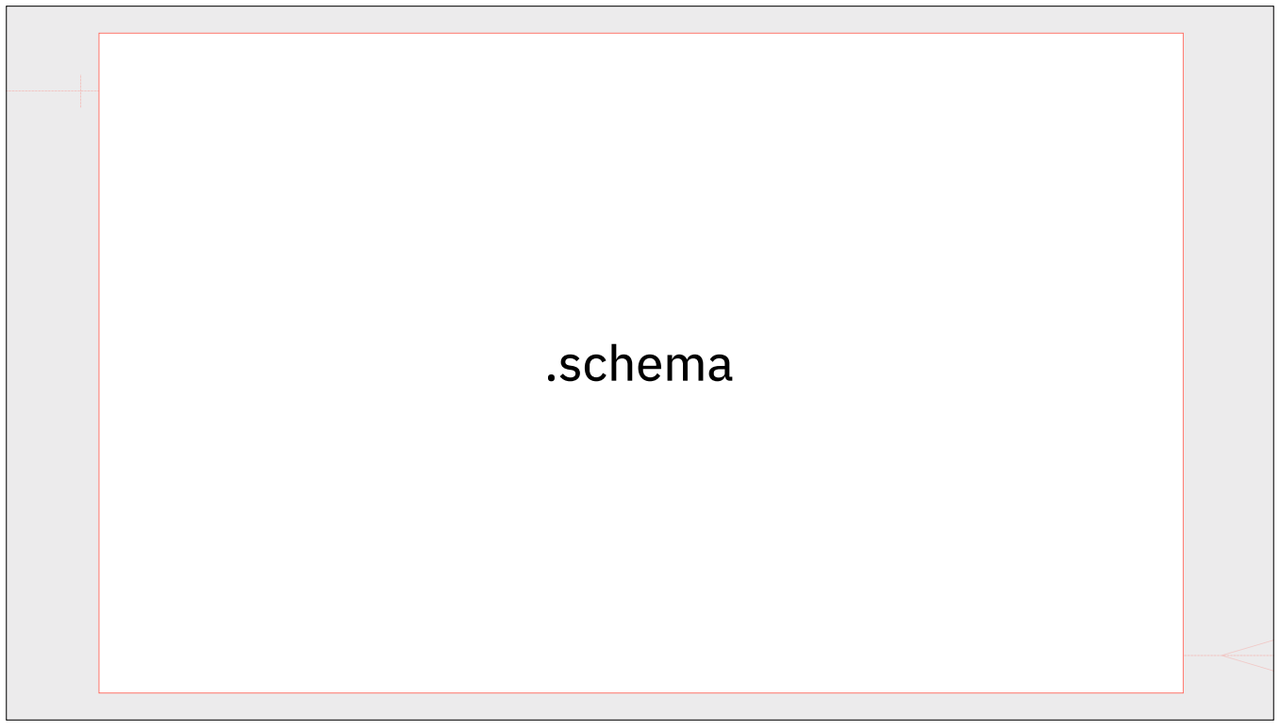
- Short answer: this is what it's made for
- sqlite is a:
    - file format
    - a storage engine
    - a library that can be used by applications
- The sqlite motto: “Small.Fast. Reliable. Choose any three”



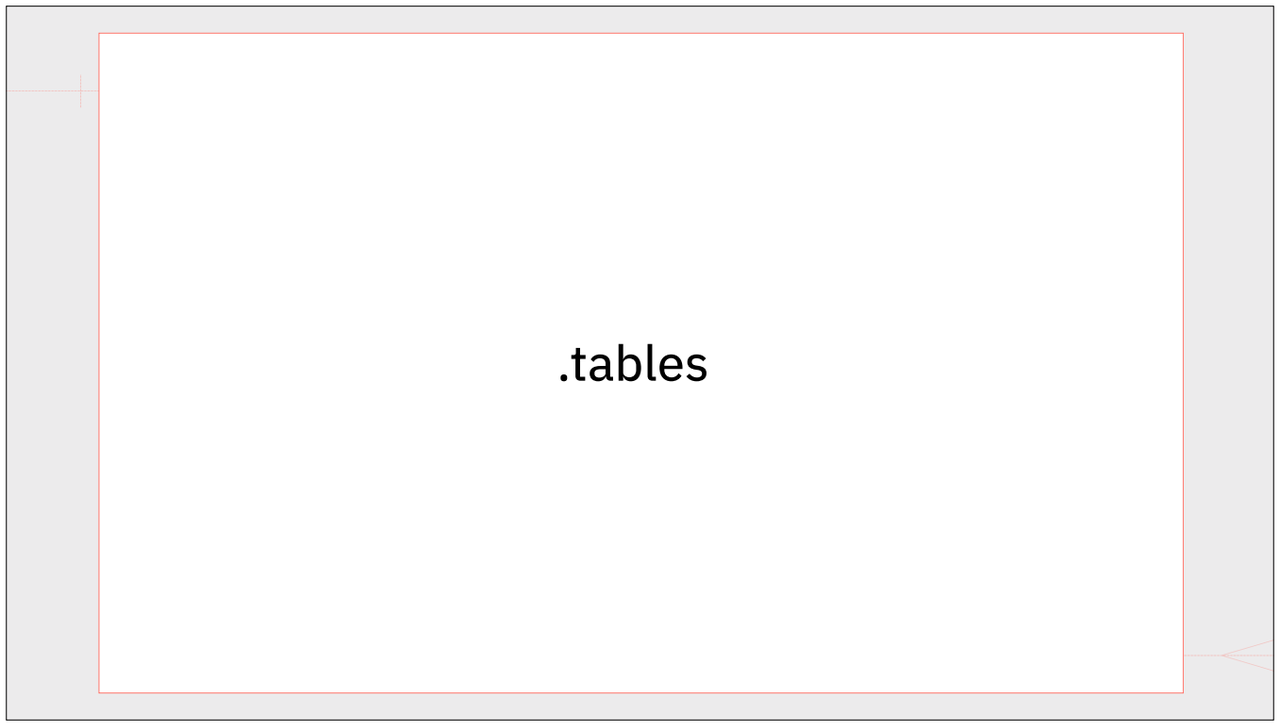
- SQLite is a file-based RDBMS. There is no server
- It uses SQL syntax for queries but it's commands are all in a particular form that start with a dot



- Here are a few to get started with
- You first enter a sqlite session by calling sqlite3 from the prompt
- .open /file/path can be used to open a database.



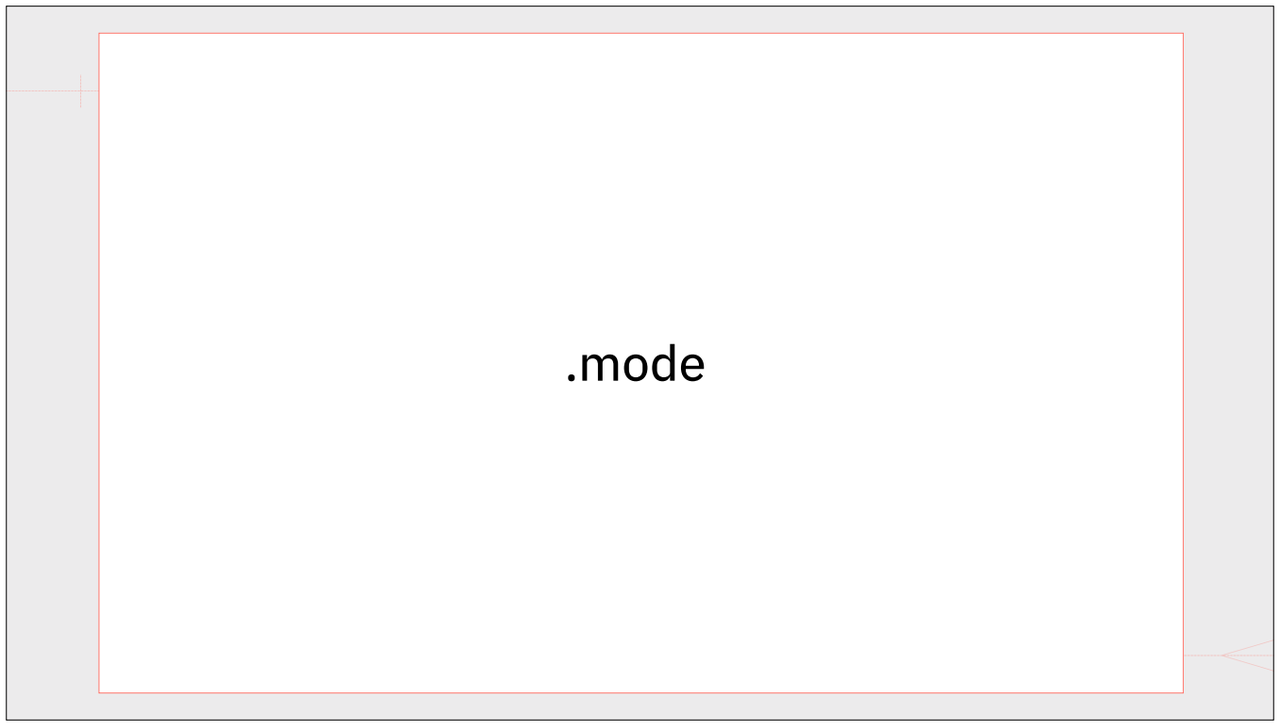
- .schema is the sqlite equivalent of desc



- .tables is the same as show tables

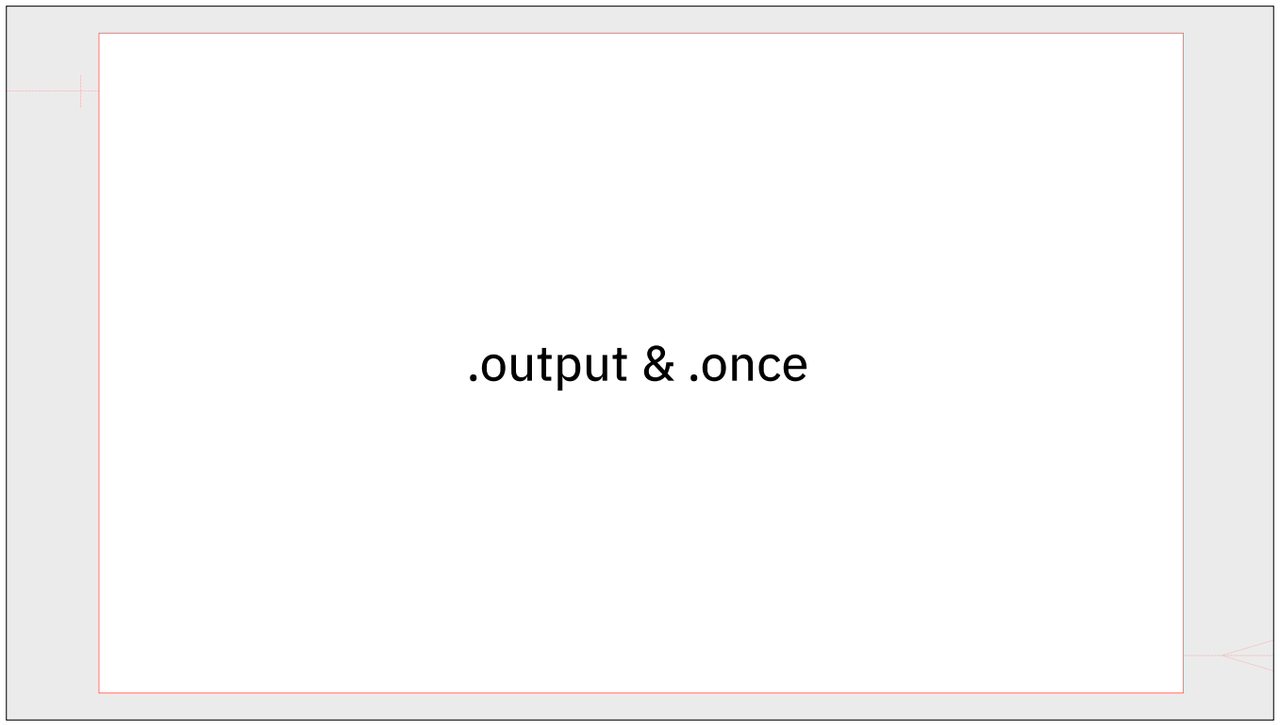
.dump

- .dump will dump the headers of a table _and_ all the contents. It's messy but can be useful when you are first investigating a database



- Sqlite allows you to format your returns
- line, list and csv are likely the most useful

- Opens an in-memory export of queries in your spreadsheet app of choice
- Not just Excel



- Output will queries to a file and keep going
- Once will export a single query's return

.backup

- .backup [table]
- saves to same directory as db file

```
sqlite3 some.db 'select * from table'
```

- What's neat is you can also run queries against a sqlite database without having to open the database
- This makes them scriptable – think EAs
- Need to supply the full path to the DB
- The query will need to be in single quotes with double quote "inside" the query if needed

Going on a database safari

- How do you find these databases?
- Think about the OS. Where would you expect preferences and config data to be?
- Look for obvious names. Lots of them are named .db or .sqldata
- Look for the wal and shm files
-

```
mdfind -name .db -onlyin /some/path
```

- mdfind is your friend

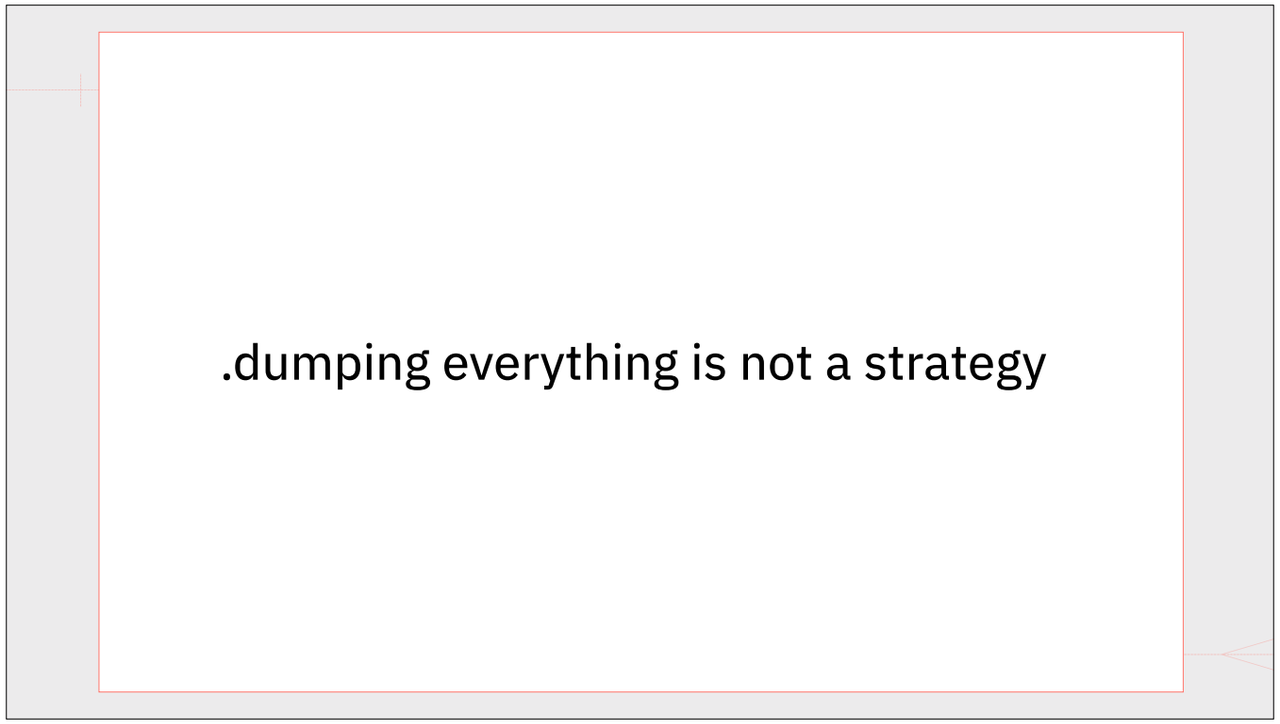
```
mdfind "kMDItemKind == 'Database
Document'" -onlyin /some/path
```

- Especially if you get specific about what you are looking for

- Sometimes these databases aren't obviously named or even tagged with metadata that identifies them as databases.
- File it
- Cat it
- Try it open it. Worse thing that happens is it fails

wal? shm?

- wal = write ahead log
- Basically a safety valve. Changes are written to the WAL file and are then only written to disk after a commit occurs
- shm = shared memory
- Holds indexes

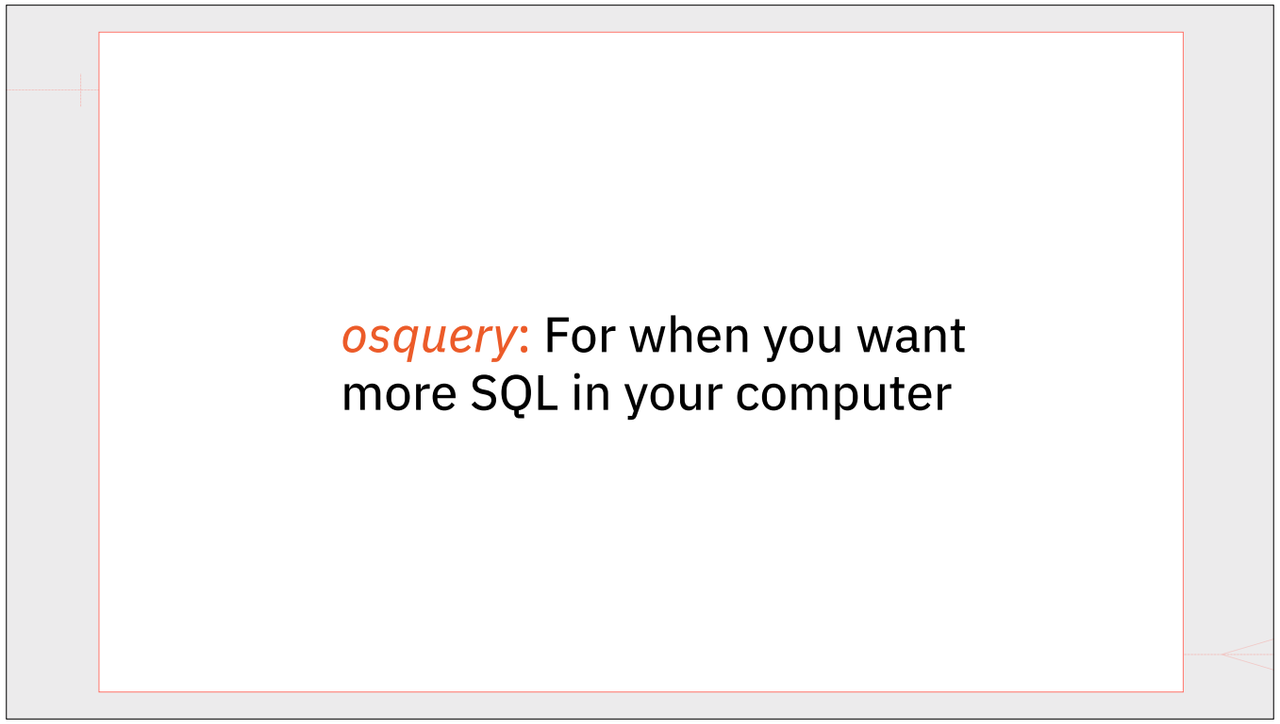


- How do you start investigating one of these databases?
- I recommend:
    - .tables to see what tables are there
    - .schema to see what the structure is
        - look for varchar
    - Use limit

Timestamp Format	Epoch (Zero Time)
Mac OS	January 1, 1904 00:00:00 UTC
Unix	January 1, 1970 00:00:00 UTC
Cocoa/WebKit (reference date*)	January 1, 2001 00:00:00 UTC
Google Chrome	January 1, 1601 00:00:00 UTC
Mozilla Firefox (PRTime)	January 1, 1970 00:00:00 UTC
Microsoft	January 1, 1601 00:00:00 UTC

- No one at Apple can agree on what time it is
- Different databases will use different times. You might need to experiment to figure out which time system is being used in the return you are looking at

- Give me a number between 1 and 26 so we can see what kind of things you can pull out of the system with sqlite



- This is not an osquery talk
- There were several good talks about osquery last year at PSU, including:
    - Exploration, Monitoring and Security with osquery by Zachary Wasserman
    - Using osquery via Fleet for client/server visibility by Lucas Hall
- There are also a couple of osquery sessions _this_ year
- Want to briefly mention it because it is yet another way of “working in SQL” that helps admin Macs

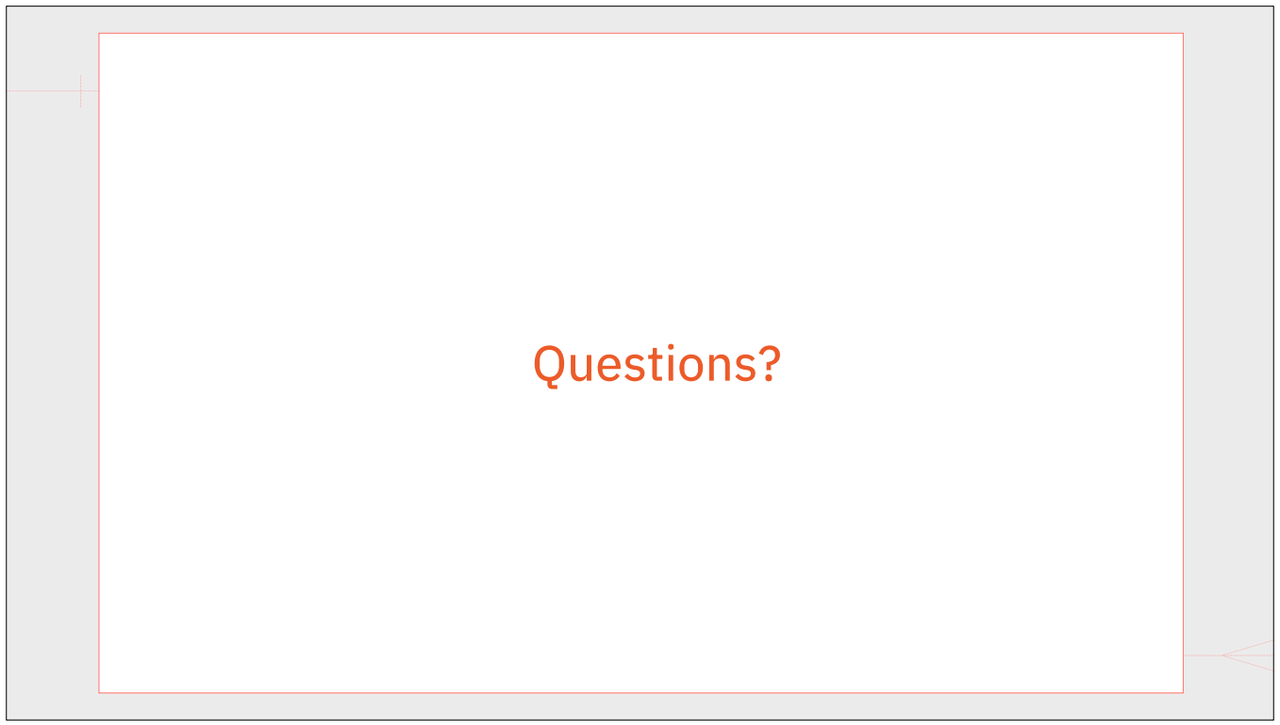
- 5. OSquery Section Demo.txt
- Take note that osquery uses sqlite .dot syntax

How can you practice without blowing up your production?

- SQlite files are all over the place and use SQL syntax
- Use the sakila database and experience time travel before Netflix killed the video store
- Install mysql on your machine and import a JPS nightly backup

http://bit.ly/MoreJoyOfSemicolons

- Links and queries



- Thank you