

DATABASE MANAGEMENT WORKSHOP

DESIGNING & IMPLEMENTING RELATIONAL DATABASES

ARCHAEOLOGICAL INFORMATION INTEREST GROUP
MONDAY JANUARY 28, 2018

ZACK BATIST

SECTION 1

ABOUT THIS WORKSHOP

OUTCOMES

- ★ Understand what a database is, and how it relates to archaeological workflows
- ★ Understand how to design and maintain effective data structures
- ★ Understand how to input, modify and retrieve data
- ★ Create your own research database to suit your specific needs
- ★ Become a proficient user of the command line and develop practical computing skills

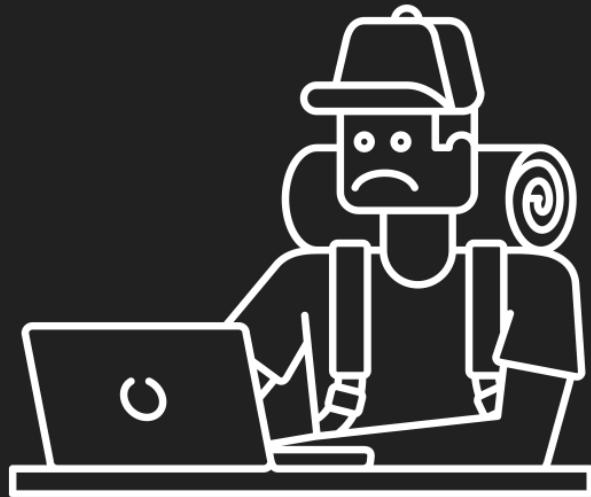
WORKSHOP STRUCTURE / LOGISTICS

- ★ Website: <https://zackbatist.github.io/archaeo-db-workshop/>
- ★ GitHub: <https://github.com/zackbatist/archaeo-db-workshop/>

ABOUT ME

- ★ Background doing network analysis and GIS
- ★ Looking at how archaeological practice is supported by and works alongside information infrastructures
- ★ I like lithics
- ★ I like R
- ★ I like using R to code cool things relating to lithics
- ★ I manage the databases at the Stelida Naxos Archaeological Project

ABOUT YOU



Created by Fernando Affonso
from Noun Project

SECTION 2

WHAT IS A DATABASE?

WHAT IS A DATABASE?

"A database is an organized collection of data, generally stored and accessed electronically from a computer system."

~ Wikipedia:Database

THREE MAIN COMPONENTS:

1. Database Management System [DBMS]
2. Database Schema
3. Data

1. DATABASE MANAGEMENT SYSTEM [DBMS]

**Software that mediates between end users,
applications and the database itself**

1. DATABASE MANAGEMENT SYSTEM [DBMS]

★ **Data definition**

- Creation, modification and removal of definitions that structure the data

★ **Update**

- Insertion, modification and deletion of actual data

★ **Retrieval**

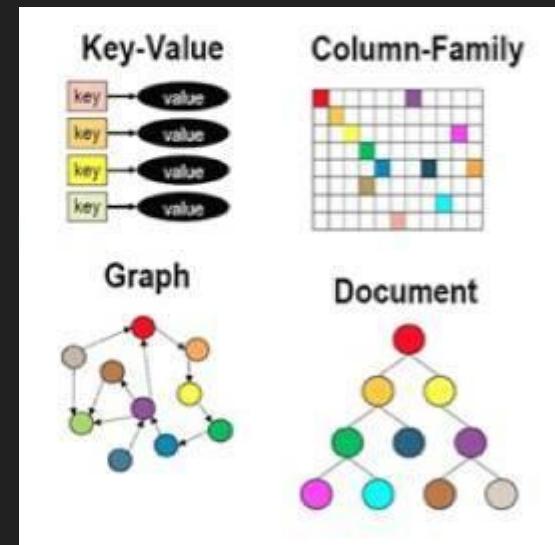
- Obtaining data from the database through a system of machine-readable commands

★ **Administration**

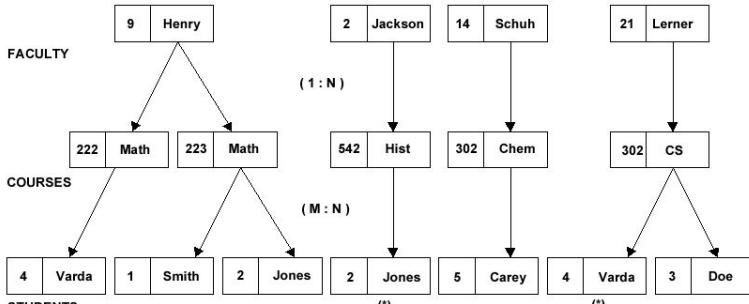
- Registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, etc

2. DATABASE SCHEMA

- ★ Overall architecture or structure of the database
- ★ The explicitly defined set of rules that structure the data and enable it to be input and retrieved in a systematic manner
- ★ Models include:
 - relational
 - graph/network
 - hierarchical/tree
 - key:value pairs/JSON

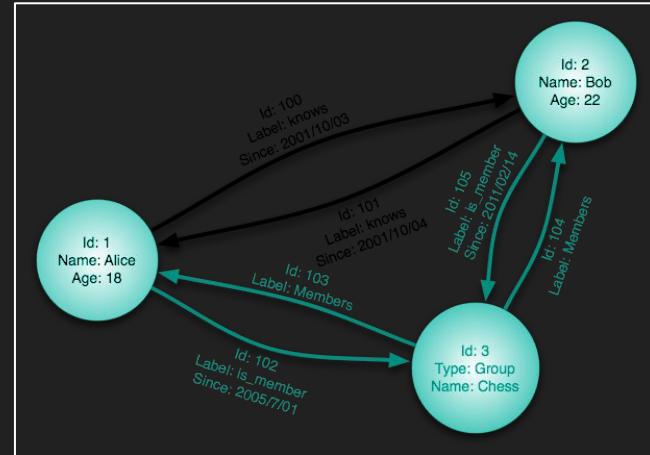


Hierarchical Data Model



Notes:
 (1) Some fields of the records in the database have been left out to simplify the diagram.
 (2) The (*)'s indicate duplicate records. We will talk about alternatives to duplication in class.

key	value
firstName	Bugs
lastName	Bunny
location	Earth



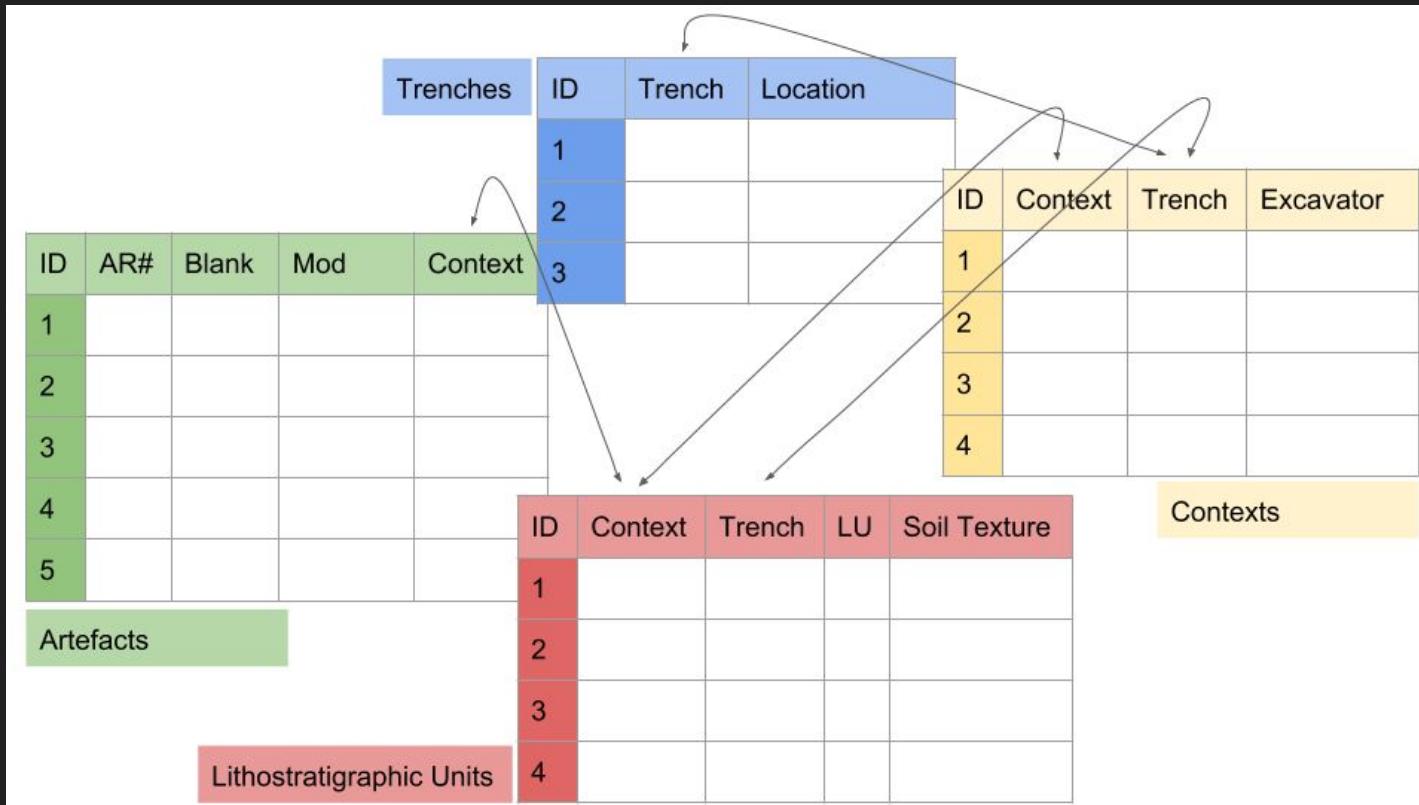
RELATIONAL DATABASES

- ★ Organizes data into a series of thematically-arranged tables of columns and rows

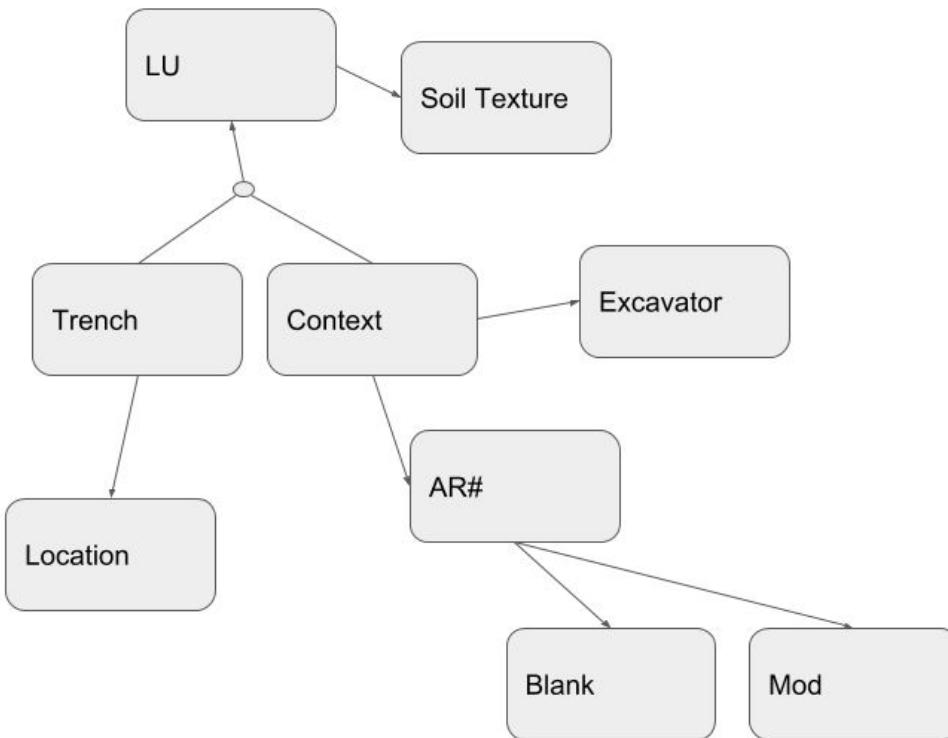
ID	Col1	Col2	Col3	Col4
Row1				
Row2				
Row3				
Row4				
Row5				

Table1

RELATIONAL DATABASES

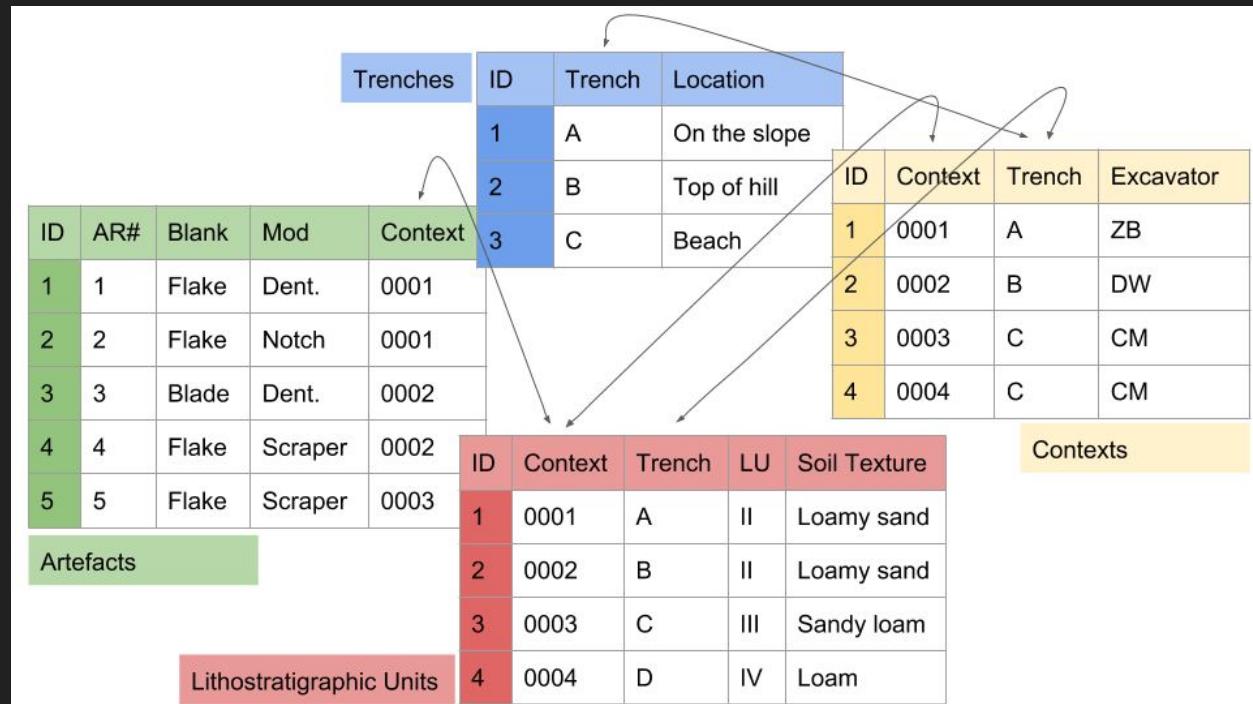


RELATIONAL DATABASES



3. DATA

- ★ The contents of the database
- ★ Must 'fit' the schema



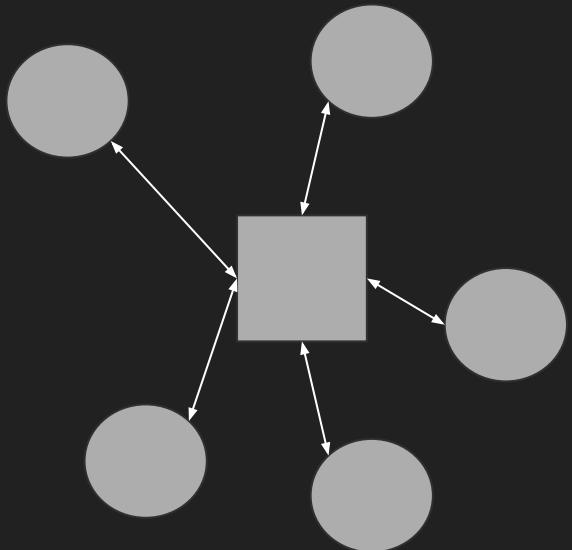
QUESTIONS?

SECTION 3

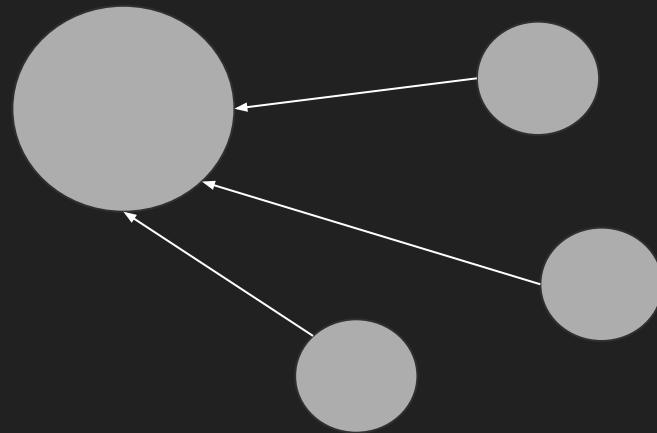
SETTING UP OUR DBMS

COMPARISON OF SOME DBMS

DBMS	Main Aspects
MariaDB/MySQL	multi-user, open source, must build your own UI
Access/FileMaker	single-user, integrated UI, proprietary, single-platform
SQLite	single-user, open source
Postgres	emphasis on standards compliance, data preservation
Neo4j	graph database, open source



Each user is granted access to a central server and can work independently

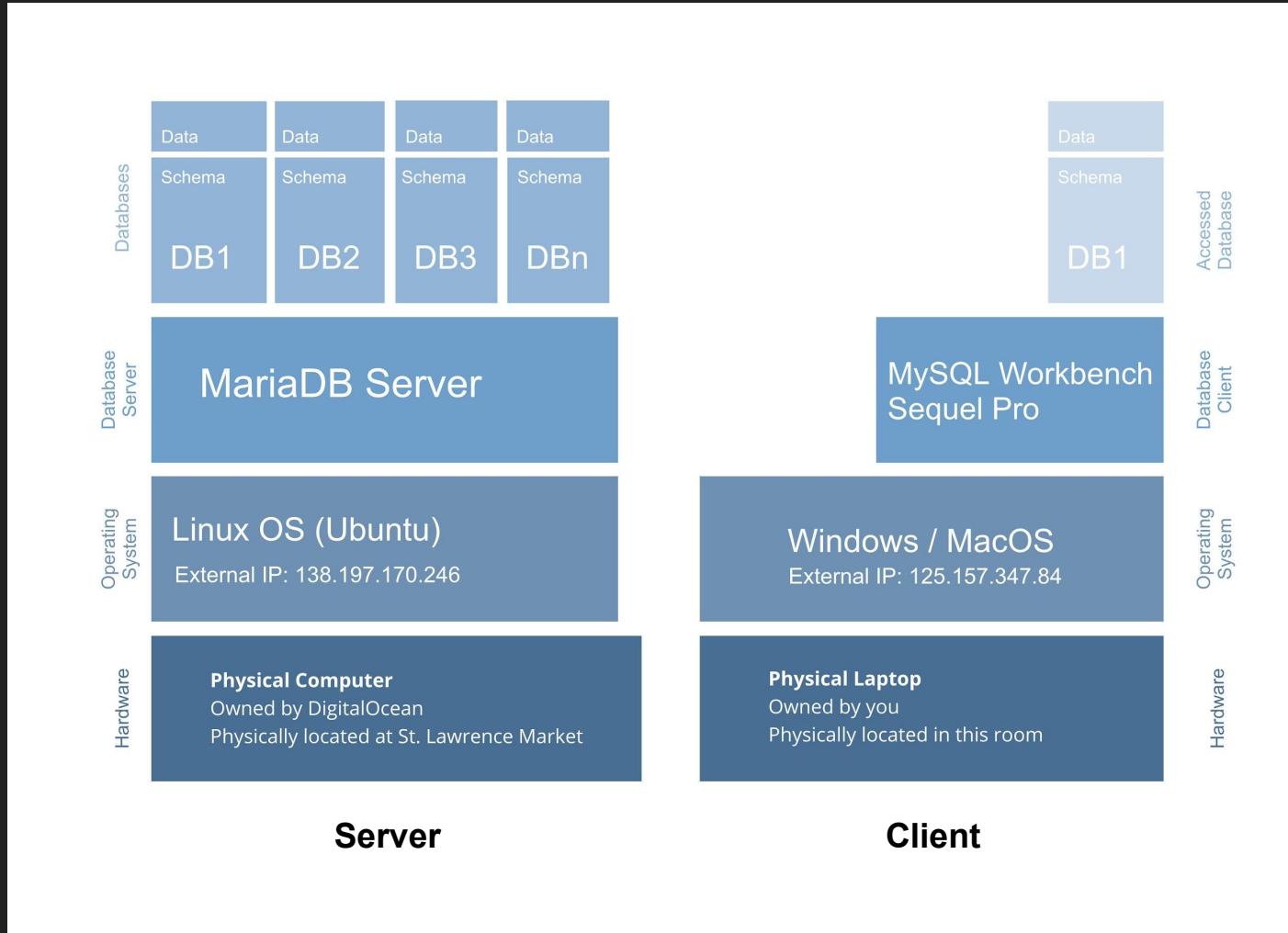


One person wrangles a bunch of emailed excel spreadsheets

MARIADB SERVER SETUP

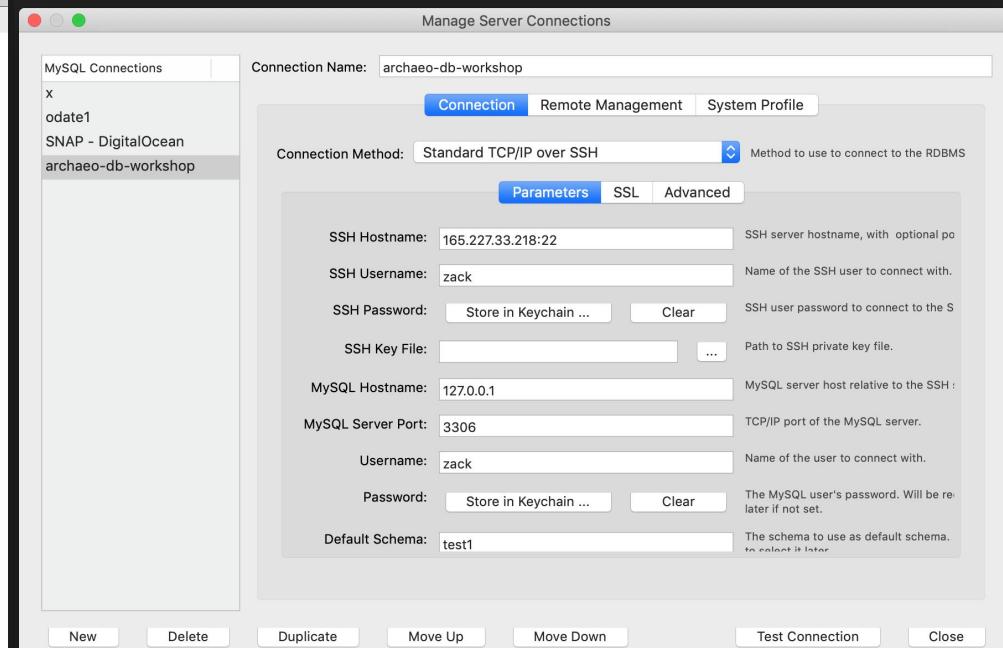
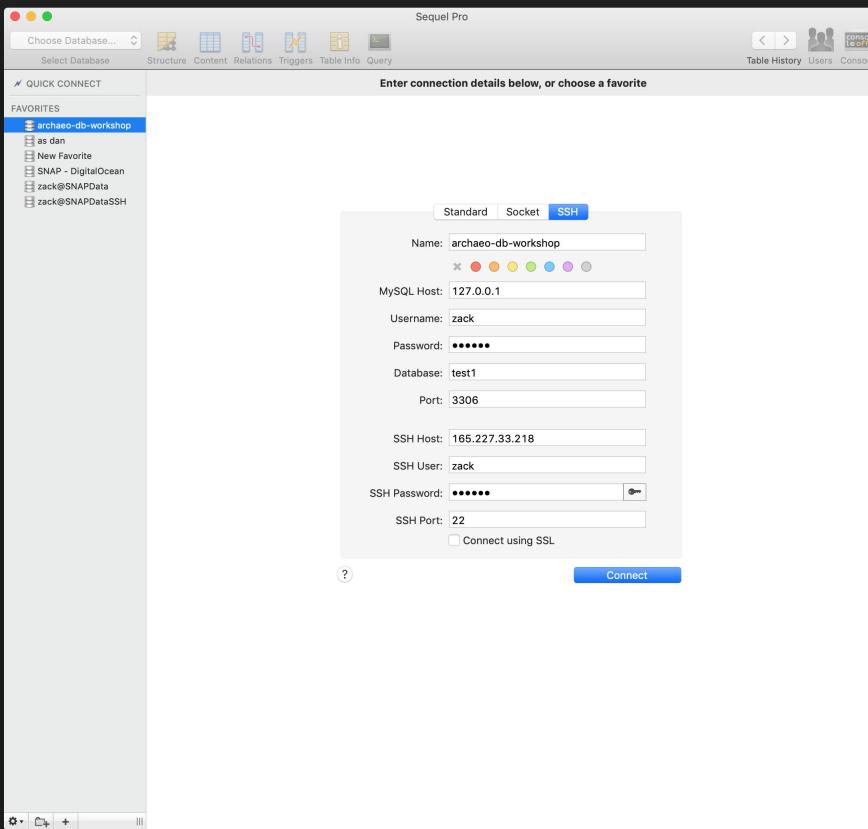
- ★ Database on a server, accessible by multiple clients
- ★ Data is centralized, unified and consistent

- ★ In this case...
 - hosted on a cloud platform [DigitalOcean]
 - free with an educational discount
 - physically situated in St. Lawrence Market

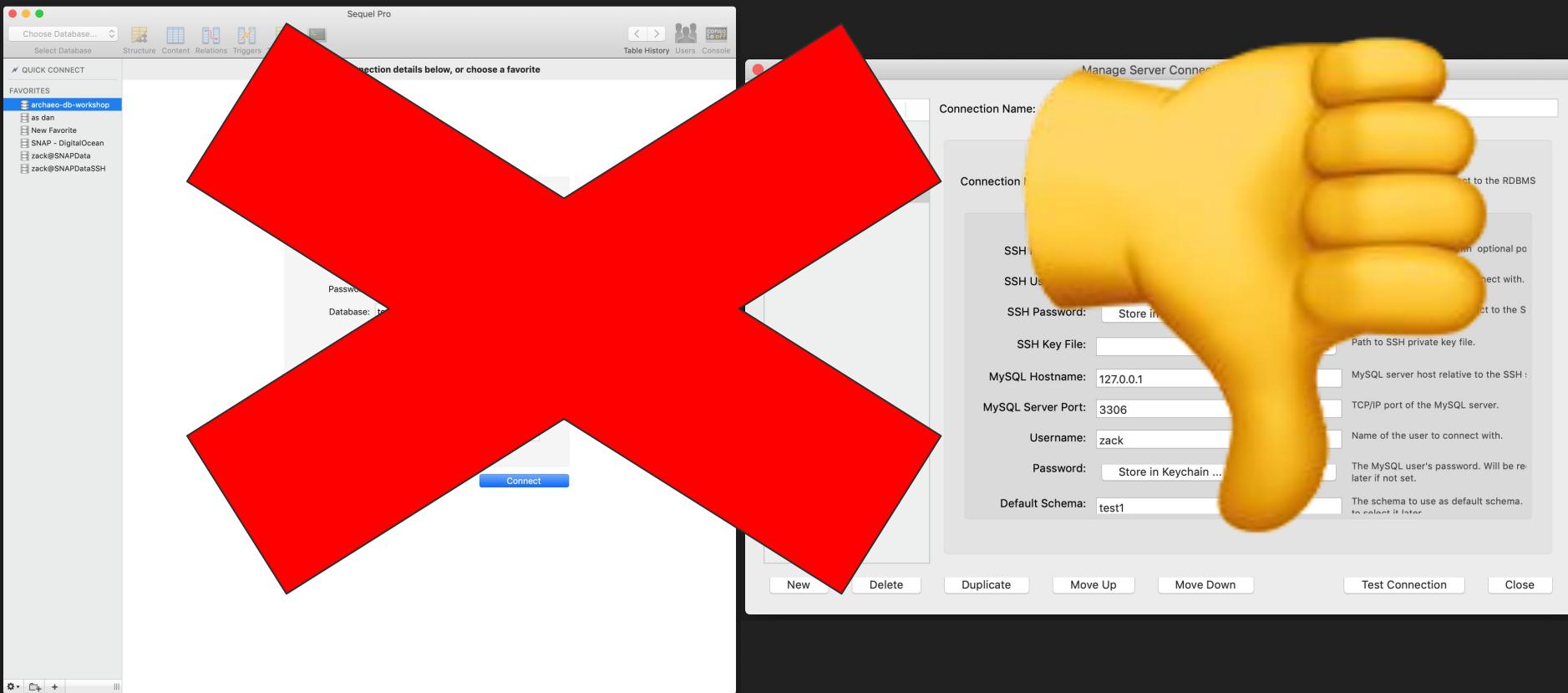


QUESTIONS?

USING A GRAPHICAL CLIENT



USING A GRAPHICAL CLIENT



USING THE COMMAND LINE ~ DON'T PANIC!

~~~

Every action is intentional, direct and self-contained

~~~

It's like getting access to the breakfast buffet instead of ordering off the fixed menu!

~~~



University of Toronto Coders - Introduction to Bash

<https://uoftcoders.github.io/studyGroup/lessons/misc/bash-intro/lesson/>

# BASH

- ★ Command
- ★ Flags / options
- ★ Arguments

CP(1) BSD General Commands Manual CP(1)

**NAME**  
`cp` -- copy files

**SYNOPSIS**  
`cp [-R [-H | -L | -P]] [-fi | -n] [-opvX] source_file target_file`  
`cp [-R [-H | -L | -P]] [-fi | -n] [-opvX] source_file ... target_directory`

**DESCRIPTION**  
In the first synopsis form, the `cp` utility copies the contents of the `source_file` to the `target_file`. In the second synopsis form, the contents of each named `source_file` is copied to the destination `target_directory`. The names of the files themselves are not changed. `cp` detects an attempt to copy a file to itself, the copy will fail.

The following options are available:

- a Same as `-pPR` options. Preserves structure and attributes of files but not directory structure.
- f If the destination file cannot be opened, remove it and create a new file without prompting for confirmation regardless of its mode.

-R If `source_file` designates a directory, `cp` copies the directory and the entire subtree connected at that point. If the `source_file` ends in a `/`, the contents of the directory are copied rather than the directory itself. This option also causes symbolic links to be copied, rather than indirection through, and for `cp` to create special files rather than copying them as normal files. Created directories have the same mode as the corresponding source directory, unmodified by the process' umask.

In `-R` mode, `cp` will continue copying even if errors are detected.

Note that `cp` copies hard-linked files as separate files. If you need to preserve hard links, consider using `tar(1)`, `cpio(1)`, or `pax(1)` instead.

-v Cause `cp` to be verbose, showing files as they are copied.

-X Do not copy Extended Attributes (EAs) or resource forks.

-c copy files using `clonefile(2)`

`cp -R ~/files ~/files-backup`

# BASH

- ★ Command
- ★ Flags / options
- ★ Arguments

```
zackbatist — less + man cp — 82x71
CP(1)          BSD General Commands Manual          CP(1)

NAME
cp -- copy files

SYNOPSIS
cp [-R [-H | -L | -P]] [-fi | -n] [-apvX] source_file target_file
cp [-R [-H | -L | -P]] [-fi | -n] [-apvX] source_file ...
    target_directory

DESCRIPTION
In the first synopsis form, the cp utility copies the contents of the
source_file to the target_file. In the second synopsis form, the con-
tents of each named source_file is copied to the destination
target_directory. The names of the files themselves are not changed.
cp detects an attempt to copy a file to itself, the copy will fail.

The following options are available:

-a   Same as -pPR options. Preserves structure and attributes of files
     but not directory structure.

-f   If the destination file cannot be opened, remove it and create a
     new file without prompting for confirmation regardless of its no-
```

-R      If `source_file` designates a directory, `cp` copies the directory and the entire subtree connected at that point. If the `source_file` ends in a `/`, the contents of the directory are copied rather than the directory itself. This option also causes symbolic links to be copied, rather than indirection through, and for `cp` to create special files rather than copying them as normal files. Created directories have the same mode as the corresponding source directory, unmodified by the process' umask.

In `-R` mode, `cp` will continue copying even if errors are detected.

Note that `cp` copies hard-linked files as separate files. If you need to preserve hard links, consider using `tar(1)`, `cpio(1)`, or `pax(1)` instead.

-v      Cause `cp` to be verbose, showing files as they are copied.

-X      Do not copy Extended Attributes (EAs) or resource forks.

-c      copy files using `clonefile(2)`

Command (copy)

cp -R ~/files ~/files-backup

Flag  
(recursive)

Source directory

Target directory

## BASH

- ★ Command
  - ★ Flags
  - ★ Directories and files

## **Common symbols and conventions**

- ★ avoid spaces in file names
  - ★ capitalization matters
  - ★ 'up' key cycles through your most recent command
  - ★ .. is like going up one directory
  - ★ ~ is home directory
  - ★ \* is wild card

CP(1) BSD General Commands Manual CP(1)

**NAME**

**cp** -- copy files

**SYNOPSIS**

**cp** [-R [-H | -L | -P]] [-fi | -n] [-apvX] source\_file target\_file  
**cp** [-R [-H | -L | -P]] [-fi | -n] [-apvX] source\_file ...  
          target\_directory

**DESCRIPTION**

In the first synopsis form, the **cp** utility copies the contents of the source\_file to the target\_file. In the second synopsis form, the contents of each named source\_file is copied to the destination target\_directory. The names of the files themselves are not changed. **cp** detects an attempt to copy a file to itself, the copy will fail.

The following options are available:

**-a** Same as **-pPR** options. Preserves structure and attributes of files but not directory structure.

**-f** If the destination file cannot be opened, remove it and create a new file without prompting for confirmation regardless of its na

## Command (copy)

```
cp -R ~/files ~/files-backup
```

Flag   
(recursive)

# I

## Source directory

A white arrow pointing upwards and to the left, indicating the target directory.

# SQL [STRUCTURED QUERY LANGUAGE]

- ★ Language used to retrieve, modify, enter or manipulate data stored in a relational database
- ★ Self-contained logical statements that include all parameters needed to do a particular action
- ★ Semicolons [; ] indicate the end of a command
- ★ If something doesn't add up, the terminal will spit out an error and nothing will happen

Example:

```
SELECT column1, column2, ...
FROM table_name;
```



<https://www.w3schools.com/sql/default.asp>

# FANTASTIC RESOURCES:

University of Toronto Coders - Introduction to Bash

<https://uoftcoders.github.io/studyGroup/lessons/misc/bash-intro/lesson/>

Learn Enough Command Line to Be Dangerous by Michael Hartl

<https://www.learnenough.com/command-line-tutorial/basics>

# Let's go!

## Connecting to the remote computer

1. ssh workshop@165.227.33.218
2. Input password: workshop1
3. Accept security certificate, type 'yes'
4. You're in! 

# Let's go!

## Connecting to the remote computer

1. ssh workshop@165.227.33.218
2. Input password: workshop1
3. Accept security certificate, type 'yes'
4. You're in! 🎉

Instructs the computer to use the information in the rest of the line to establish a link over the internet to the computer that hosts the database

The username on that computer

The location of that computer on the internet

Credentials to gain access

Used to ensure that your key fits the lock (or something...)

```
zackbatist — workshop@ubuntu-s-1vcpu-1gb-tor1-01: ~ — ssh...
Last login: Sun Jan 20 14:15:37 on ttys000
[Zacks-MacBook-Pro:~ zackbatist$ ssh workshop@165.227.33.218
[workshop@165.227.33.218's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-43-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sun Jan 20 19:16:07 UTC 2019

System load:  0.1          Processes:      86
Usage of /:   8.9% of 24.06GB  Users logged in:   0
Memory usage: 19%
Swap usage:   0%          IP address for eth0: 165.227.33.218
                           IP address for eth1: 10.137.200.208

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch

27 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Sun Jan 20 19:15:45 2019 from 184.149.38.121
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

workshop@ubuntu-s-1vcpu-1gb-tor1-01:~$
```

# Let's go!

## Connecting to the remote computer

1. ssh workshop@165.227.33.218
2. Input password: workshop1
3. Accept security certificate, type 'yes'
4. You're in! 🎉

Instructs the computer to use the information in the rest of the line to establish a link over the internet to the computer that hosts the database

The username on that computer

The location of that computer on the internet

Credentials to gain access

Used to ensure that your key fits the lock (or something...)

## Connecting to the MariaDB server on that computer

1. Mysql -u <username> -p
2. Enter your password
3. □

# Let's go!

## Connecting to the remote computer

1. ssh workshop@165.227.33.218
2. Input password: workshop1
3. Accept security certificate, type 'yes'
4. You're in! 🎉

Instructs the computer to use the information in the rest of the line to establish a link over the internet to the computer that hosts the database

The username on that computer

The location of that computer on the internet

Credentials to gain access

Used to ensure that your key fits the lock (or something...)

## Connecting to the MariaDB server on that computer

1. Mysql -u <username> -p
2. Enter your password
3. □

Instructs the computer to use the following information to grant access afforded to a user by the database administrator

Indicates that what follows is a username

Ensures that the computer prompts you to input a password

```
zackbatist — workshop@ubuntu-s-1vcpu-1gb-tor1-01: ~ — ssh...
Last login: Sun Jan 20 14:15:37 on ttys000
[Zacks-MacBook-Pro:~ zackbatist$ ssh workshop@165.227.33.218
[workshop@165.227.33.218's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-43-generic x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Sun Jan 20 19:16:07 UTC 2019

System load: 0.1 Processes: 86
Usage of /: 8.9% of 24.06GB Users logged in: 0
Memory usage: 19% IP address for eth0: 165.227.33.218
Swap usage: 0% IP address for eth1: 10.137.200.208

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch

27 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Sun Jan 20 19:15:45 2019 from 184.149.38.121
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

workshop@ubuntu-s-1vcpu-1gb-tor1-01: ~
```

```
zackbatist — workshop@ubuntu-s-1vcpu-1gb-tor1-01: ~ — ssh...
Last login: Sun Jan 20 14:24:35 on ttys001
[Zacks-MacBook-Pro:~ zackbatist$ ssh workshop@165.227.33.218
[workshop@165.227.33.218's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-43-generic x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Sun Jan 20 19:24:40 UTC 2019

System load: 0.0 Processes: 83
Usage of /: 8.9% of 24.06GB Users logged in: 0
Memory usage: 19% IP address for eth0: 165.227.33.218
Swap usage: 0% IP address for eth1: 10.137.200.208

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

* Canonical Livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
https://ubuntu.com/livepatch

27 packages can be updated.
0 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Last login: Sun Jan 20 19:24:04 2019 from 184.149.38.121
[workshop@ubuntu-s-1vcpu-1gb-tor1-01: ~] $ mysql -u zack -p
[Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 39
Server version: 10.1.34-MariaDB-0ubuntu0.18.04.1 Ubuntu 18.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

# SQL [STRUCTURED QUERY LANGUAGE]

- ★ Language used to retrieve, modify, enter or manipulate data stored in a relational database
- ★ Self-contained logical statements that include all parameters needed to do a particular action
- ★ Semicolons [; ] indicate the end of a command
- ★ If something doesn't add up, the terminal will spit out an error and nothing will happen

Example:

```
SELECT column1, column2, ...
FROM table_name;
```



<https://www.w3schools.com/sql/default.asp>

# Show databases

```
MariaDB [(none)]> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| test1 |  
+-----+  
2 rows in set (0.00 sec)
```

```
MariaDB [(none)]>
```

# Show tables

```
MariaDB [test1]> USE test1;  
Database changed  
MariaDB [test1]> SHOW TABLES;  
+-----+  
| Tables_in_test1 |  
+-----+  
| Persons |  
| Places |  
+-----+  
2 rows in set (0.00 sec)
```

```
MariaDB [test1]>
```

# Describe a table

```
MariaDB [test1]> DESC Persons;  
+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+  
| PersonID | int(11) | YES | | NULL | |  
| LastName | varchar(255) | YES | | NULL | |  
| FirstName | varchar(255) | YES | | NULL | |  
| Address | varchar(255) | YES | | NULL | |  
| City | varchar(255) | YES | | NULL | |  
+-----+-----+-----+-----+-----+  
5 rows in set (0.01 sec)
```

```
MariaDB [test1]>
```

# Inspect users (only as administrator)

```
MariaDB [(none)]> SELECT HOST, USER, PASSWORD FROM mysql.user;  
+-----+-----+-----+  
| HOST | USER | PASSWORD |  
+-----+-----+-----+  
| localhost | root | |
| localhost | zack | *1665C2C61D7115C9F8D07333B512F8528EEECA7E |  
| % | zack | *1665C2C61D7115C9F8D07333B512F8528EEECA7E |  
| 127.0.0.1 | zack | *1665C2C61D7115C9F8D07333B512F8528EEECA7E |  
+-----+-----+  
4 rows in set (0.00 sec)
```

```
MariaDB [(none)]>
```

# CREATING A DB ON THE SERVER

## Syntax

```
CREATE DATABASE databasename;
```

## Example

```
[MariaDB [(none)]> CREATE DATABASE D0bsiSS;  
Query OK, 1 row affected (0.00 sec)
```

```
MariaDB [(none)]> █
```

# CREATING A TABLE

## Syntax

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
    ...
);
```

## Example

```
MariaDB [(none)]> CREATE TABLE `DObSiSS`.`sites` (
    -> `id` INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
    -> `site` VARCHAR(255) DEFAULT '',
    -> `lat` VARCHAR(255) DEFAULT '',
    -> `lon` VARCHAR(255) DEFAULT '',
    -> `country` VARCHAR(255) DEFAULT '',
    -> `retired` BIT DEFAULT '',
    -> PRIMARY KEY (`id`)
    -> );
Query OK, 0 rows affected (0.01 sec)
```

```
MariaDB [(none)]>
```

## Verify

```
Database changed
[MariaDB [DObSiSS]]> SHOW TABLES;
+-----+
| Tables_in_DObSiSS |
+-----+
| sites           |
+-----+
1 row in set (0.00 sec)

[MariaDB [DObSiSS]]> DESC sites;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra   |
+-----+-----+-----+-----+-----+-----+
| id         | int(11) unsigned | NO  | PRI | NULL    | auto_increment |
| site       | varchar(255)    | YES |     |          |          |
| lat        | varchar(255)    | YES |     |          |          |
| lon        | varchar(255)    | YES |     |          |          |
| country    | varchar(255)    | YES |     |          |          |
| retired    | bit(1)          | YES |     | b'0'    |          |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)

MariaDB [DObSiSS]]>
```

# DATA TYPES

| Data Type      | Description            |
|----------------|------------------------|
| VARCHAR        | variable-length string |
| CHAR           | fixed-length string    |
| INT            | integer                |
| DEC            | decimal number         |
| TEXT           | long string of text    |
| BLOB           | binary object          |
| DATE/TIME/YEAR | date/time/year         |

~ BREAK ~

# SECTION 4

# STRUCTURING OUR DATABASE

# SCOPING TABLES

- ★ Each table represents one kind of entity
- ★ Rows represent instances of the entity
- ★ Each instance has the same kinds of attributes, but different variations thereof
- ★ Columns are assigned a *data type*

# INDEXES

- ★ Indexes indicate an independent variable or set of variables, in relation to their attributes
  
- ★ Sometimes referred to as keys
- ★ Can be set to require that each instance be unique
- ★ Each table has a primary index, which must be unique

**Pro Tip:** Use sequential integers or time stamps to obtain unique values!

# RELATIONSHIPS

- ★ Relationships indicate that the values in one index are equivalent to the values in another index
- ★ Records must relate variables of the same data type

# SCOPING TABLES

- ★ What kinds of physical stuff are you recording?
- ★ What ‘units’ are you working with?
- ★ Who are the stakeholders? Which are the relevant domains to include?
- ★ How will data be integrated or synthesized at later stages of your project?
- ★ How are you standardizing and documenting your terminology and methods?
- ★ Who will be directly interfacing with the database?
- ★ What relevant kinds of data is not contained in the database?
- ★ How might you leave the door open for expansion?

# SCOPING TABLES (DObsiSS)

## ★ Sites

- names, coordinates

## ★ DataPoints

- sites split according to blank periodization scheme

## ★ References

- bibliographical references to site info, index is the reference not the site

## ★ SourcingStudies

- info about each study that sourced artefacts from each site, index is the attempt

## ★ SourcingResults

- results of sourcing studies

## ★ Sources

- list of obsidian sources, geographic coordinates, notable features (peralkaline, snowflake, etc)

## ★ Periods

- details regarding the blank periodization applied to the region

# SCOPING TABLES (DObsiSS)

## ★ Sites

- names, coordinates, local sequences

## ★ DataPoints

- sites split according to blank periodization scheme

## ★ References

- bibliographical references to site info, index is the reference not the site

## ★ SourcingStudies

- info about each study that sourced artefacts from each site, index is the attempt

## ★ SourcingResults

- results of sourcing studies

## ★ Sources

- list of obsidian sources, geographic coordinates, notable features (peralkaline, snowflake, etc)

## ★ Periods

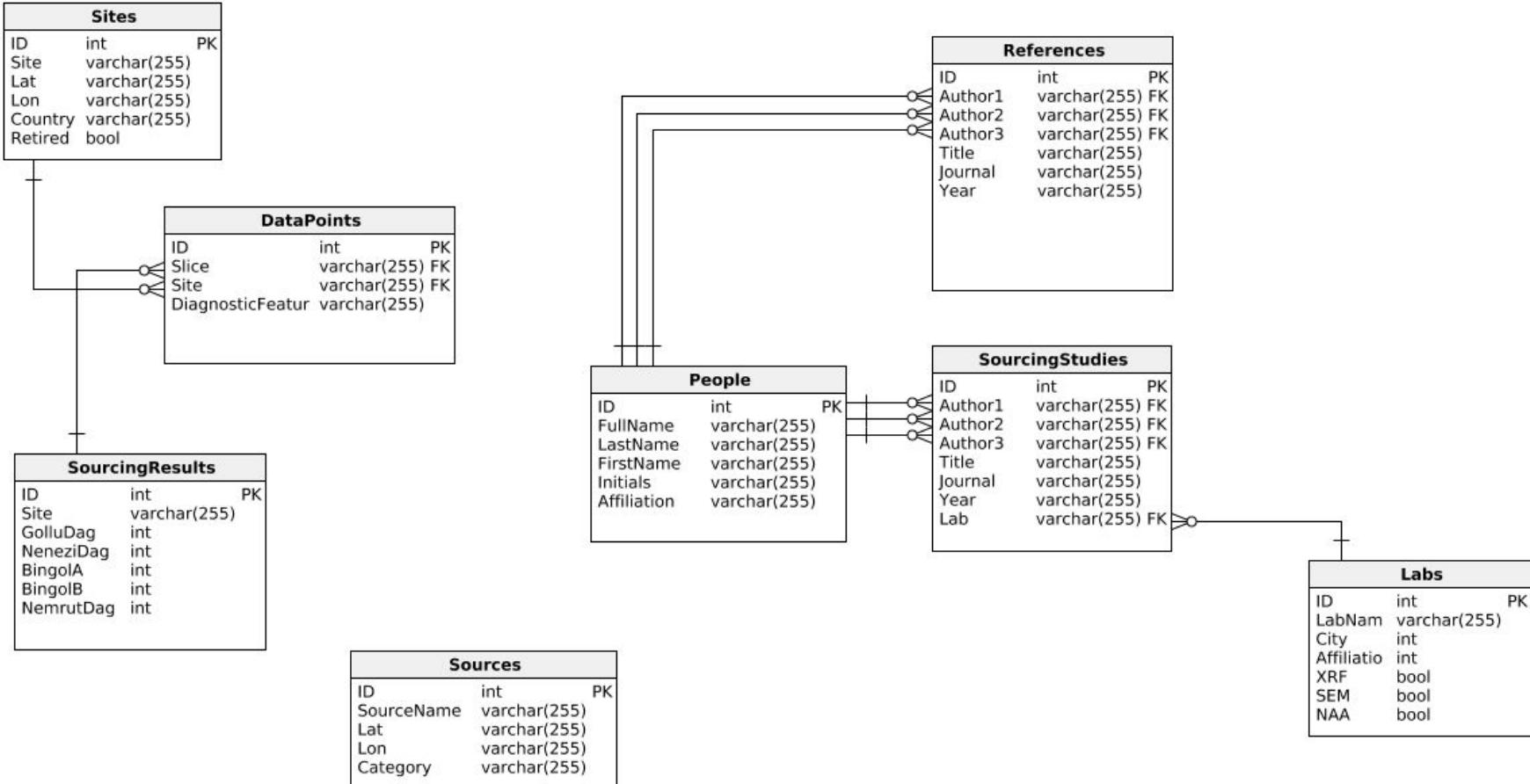
- details regarding the blank periodization applied to the region

### Potential expansions:

- AssemblageInfo
  - ◆ How much is obsidian?
- Artefacts
  - ◆ Listing of artefact-level features from each site
  - ◆ Possibly split into typology/technology and sourcing results
- Rhyolite/Basalt/other materials
- ...

# SCOPING TABLES

- ★ What kinds of physical stuff are you recording?
- ★ What ‘units’ are you working with?
- ★ Who are the stakeholders? Which are the relevant domains to include?
- ★ How will data be integrated or synthesized at later stages of your project?
- ★ How are you standardizing and documenting your terminology and methods?
- ★ Who will be directly interfacing with the database?
- ★ What relevant kinds of data is not contained in the database?
- ★ How might you leave the door open for expansion?



```

4 CREATE DATABASE DObsiSS;
5 CREATE USER 'zack'@'localhost' IDENTIFIED BY 'batist';
6 CREATE USER 'zack'@'12.34.56.78' IDENTIFIED BY 'batist';
7 GRANT ALL PRIVILEGES ON DObsiSS.* TO 'zack'@'%' IDENTIFIED BY 'batist';
8 FLUSH PRIVILEGES;
9 mysql -u zack -p
10
11 CREATE TABLE `DObsiSS`.`sites` (
12   `id` INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
13   `site` VARCHAR(255) DEFAULT '',
14   `lat` VARCHAR(255) DEFAULT '',
15   `lon` VARCHAR(255) DEFAULT '',
16   `country` VARCHAR(255) DEFAULT '',
17   `retired` BIT DEFAULT '',
18   PRIMARY KEY (`id`)
19 );
20
21 INSERT INTO `DObsiSS`.`sites` (site, lat, lon, country, retired) VALUES ("Toronto", "43.7001100", "-79.4163000", "Canada", 0);
22 SELECT `site`, `lat`, `lon` FROM `sites` WHERE `sites`.`site` = "Toronto";
23 UPDATE `DObsiSS`.`sites` SET `retired` = 1 WHERE `site` = "Toronto";
24 DELETE FROM `sites` WHERE `site` = "Toronto";
25
26 CREATE TABLE `DObsiSS`.`datapoints` (
27   `id` INT(11) SIGNED NOT NULL AUTO_INCREMENT,
28   `slice` VARCHAR(255) DEFAULT '',
29   `site` VARCHAR(255) DEFAULT '',
30   `diagnosticfeatures` VARCHAR(255) DEFAULT '',
31   PRIMARY KEY (`id`)
32 );
33
34 INSERT INTO `DObsiSS`.`sites` (site, lat, lon, country, retired)
35 VALUES ("Toronto", "43.7001100", "-79.4163000", "Canada", 0);
36
37 INSERT INTO `DObsiSS`.`datapoints` (slice, site, diagnosticfeatures)
38 VALUES ("GreatestGen", "Toronto", "Horses and buggies, doilies");
39
40 INSERT INTO `DObsiSS`.`datapoints` (slice, site, diagnosticfeatures)
41 VALUES ("Boomers", "Toronto", "Vinyl records, blue jeans");
42
43 INSERT INTO `DObsiSS`.`datapoints` (slice, site, diagnosticfeatures)
44 VALUES ("GenX", "Toronto", "Walkman");
45
46 INSERT INTO `DObsiSS`.`datapoints` (slice, site, diagnosticfeatures)
47 VALUES ("Millennials", "Toronto", "Beanie babies, iPods");
48
49 INSERT INTO `DObsiSS`.`datapoints` (slice, site, diagnosticfeatures)
50 VALUES ("GenZ", "Toronto", "Fidget spinners, bad things");
51
52 INSERT INTO `DObsiSS`.`datapoints` (slice, site, diagnosticfeatures)
53 VALUES ("ApocalypseSurvivors", "Toronto", "sticks and stones");
54
55 ALTER TABLE `DObsiSS`.`datapoints`
56 ADD CONSTRAINT FK_datapoints-site
57 FOREIGN KEY (`site`)
58 REFERENCES `DObsiSS`.`sites` (`site`)
59 ON DELETE CASCADE ON UPDATE CASCADE;
60
61 SELECT `datapoints`.slice, `datapoints`.diagnosticfeatures, `sites`.lat, `sites`.lon
62 FROM `datapoints` INNER JOIN `sites`
63 ON `sites`.`site` = `datapoints`.`site`;

```

CREATE database, GRANT USER privileges

CREATE TABLE called 'sites'

INSERT values for site named Toronto

Demonstrate the SELECT, UPDATE and DELETE queries

CREATE a second table called 'datapoints'

INSERT values into datapoints

Create relationship between 'sites' and 'datapoints' via their mutual field, 'site'

JOIN query

# SECTION 5

# MANIPULATING DATA

# QUERIES

- ★ Everything you do is expressed as a self-contained logical statement
- ★ There are a few fundamental commands:

**INSERT**

**UPDATE**

**SELECT**

**DELETE**

**JOIN**

# INSERT - Inserts records into a table

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

# UPDATE - Modified cells within a table, when they match certain conditions

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE SiteName = "Abu Hureyra"
      Country = "Syria"
      ...
```

# SELECT - Retrieves data from a table

```
SELECT column1, column2, ...  
FROM table_name;
```

```
SELECT * FROM table_name;
```

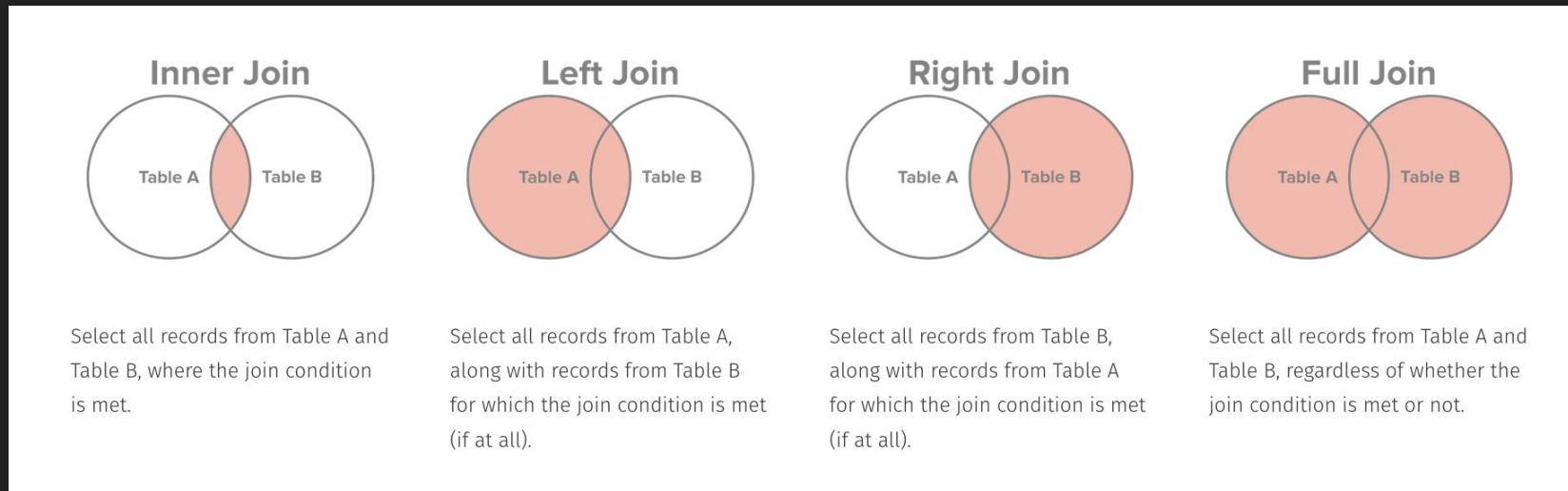
## DELETE - Deletes a record

```
DELETE FROM table_name WHERE condition;
```

```
DELETE FROM table_name WHERE
```

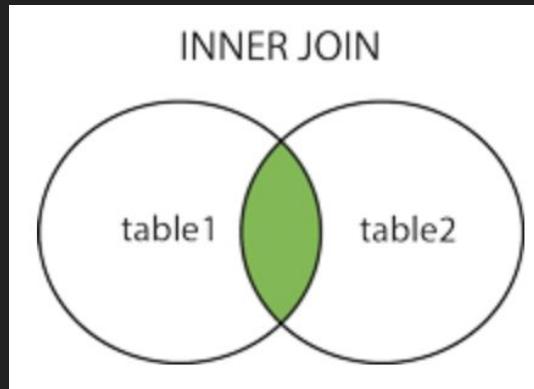
SiteName = "Abu Hureyra"  
Country = "Syria"  
...

# JOIN - Used as optional component of SELECT queries when you want to integrate data spanning multiple tables



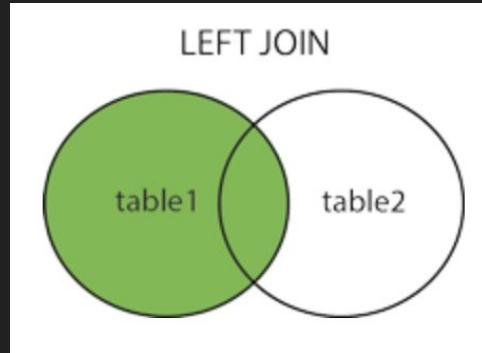
# INNER JOIN - Only retrieves records from both tables that have a certain value in common

```
SELECT column_name(s)  
FROM table1  
INNER JOIN table2 ON table1.column_name = table2.column_name;
```



**LEFT JOIN - Retrieves all records from one table and only records from another table when there is a corresponding record in the first table**

```
SELECT column_name(s)  
FROM table1  
LEFT JOIN table2 ON table1.column_name = table2.column_name;
```



# “Computer -- show me some characteristics of lithics from the Middle Palaeolithic alongside some geoarchaeological characteristics of their contexts”

```
SELECT `level3`.`ArtefactID`, `level3`.`Blank`, `level3`.`Modification`, `level3`.`Period`, `level3`.`WeatheringIndex`, `level3`.`Patination`,  
`level3`.`Locus`, `level3`.`LocusType`, `sediment`.`TextureClass`, `sediment`.`Structure`, `sediment`.`Consistence`, `sediment`.`Notes`  
FROM `level3`  
LEFT JOIN `sediment` ON `level3`.`Locus` = `sediment`.`Context`  
WHERE `level3`.`Period` = 'MP';
```

\*\* beep boop beep beep boop beep \*\*

| ArtefactID | Blank                | Modification | Period | WeatheringIndex | Patination | Locus                     | LocusType                 | TextureClass | Structure | Consistence | Notes |
|------------|----------------------|--------------|--------|-----------------|------------|---------------------------|---------------------------|--------------|-----------|-------------|-------|
| AR00002    | Levallois Flake Core |              | MP     | 2               |            | NULL   G100-020.1060-1145 | Grid Collection Point     | NULL         | NULL      | NULL        | NULL  |
| AR00003    | Blade 2              | backed       | MP     | 2               |            | NULL   G100-020.1060-1145 | Grid Collection Point     | NULL         | NULL      | NULL        | NULL  |
| AR00004    | Levallois Flake 3    | denticulate  | MP     | 2               |            | NULL   G100-020.1060-1145 | Grid Collection Point     | NULL         | NULL      | NULL        | NULL  |
| AR00005    | Levallois Flake 3    | denticulate  | MP     | 2               |            | NULL   G100-020.1060-1145 | Grid Collection Point     | NULL         | NULL      | NULL        | NULL  |
| AR00006    | Levallois Flake 3    |              | MP     | 2               |            | NULL   G100-020.1060-1145 | Grid Collection Point     | NULL         | NULL      | NULL        | NULL  |
| AR00007    | Levallois flake core |              | MP     | 2               |            | NULL   T39-0800           | Transect Collection Point | NULL         | NULL      | NULL        | NULL  |
| AR00024    | Levallois flake core | combined     | MP     | 2               |            | NULL   T20-0850           | Transect Collection Point | NULL         | NULL      | NULL        | NULL  |
| AR00025    | Levallois flake core |              | MP     | 2               |            | NULL   T20-0580           | Transect Collection Point | NULL         | NULL      | NULL        | NULL  |
| AR00026    | Levallois point core |              | MP     | 2               |            | NULL   T39-1130           | Transect Collection Point | NULL         | NULL      | NULL        | NULL  |
| AR00027    | Flake 1              | combined     | MP     | 2               |            | NULL   T05-0920           | Transect Collection Point | NULL         | NULL      | NULL        | NULL  |
| AR00028    | Flake Core           |              | MP     | 2               |            | NULL   T05-0920           | Transect Collection Point | NULL         | NULL      | NULL        | NULL  |
| AR00029    | Blade-like flake 3   | point        | MP     | 2               |            | NULL   T07-0990           | Transect Collection Point | NULL         | NULL      | NULL        | NULL  |
| AR00030    | Flake 3              | unmodified   | MP     | 2               |            | NULL   T07-0990           | Transect Collection Point | NULL         | NULL      | NULL        | NULL  |
| AR00031    | Flake 3              | burin        | MP     | 2               |            | NULL   T07-0990           | Transect Collection Point | NULL         | NULL      | NULL        | NULL  |
| AR00032    | Levallois blade core | combined     | MP     | 2               |            | NULL   T29-0900           | Transect Collection Point | NULL         | NULL      | NULL        | NULL  |
| AR00033    | Flake 3              | point        | MP     | 2               |            | NULL   T39-1130           | Transect Collection Point | NULL         | NULL      | NULL        | NULL  |

# SECTION 6

# FRONT-END DEVELOPMENT

# FLIRT - Flexible Lithic Recording Tool

## SNAP Lithics Processing

|       |            |                    |         |              |                      |
|-------|------------|--------------------|---------|--------------|----------------------|
| Locus | Locus Type | Period             | Blank   | Modification | Quantity             |
| 0000  | Context    | Lower Palaeolithic | Flake 1 | Uniface      | <input type="text"/> |

Submit  New Blank or Modification

|                                     |                                            |                    |
|-------------------------------------|--------------------------------------------|--------------------|
| New Blank                           | New Modification                           | Period             |
| <input type="text"/>                | <input type="text"/>                       | Lower Palaeolithic |
| <input type="button"/> Submit Blank | <input type="button"/> Submit Modification |                    |

All Level 2 All Level 3

Show 10  entries

Search:

| id  | Locus | LocusType | Period         | Blank              | Modification    | Quantity | Notes |
|-----|-------|-----------|----------------|--------------------|-----------------|----------|-------|
| All | All   | All       | All            | All                | All             | All      | All   |
| 1   | 0001  | Context   | Non-Diagnostic | Blade-like Flake 2 | No Modification | 22       |       |
| 2   | 0001  | Context   | Non-Diagnostic | Blade-like Flake 3 | No Modification | 17       |       |
| 3   | 0001  | Context   | Non-Diagnostic | Chips              | No Modification | 4        |       |
| 4   | 0001  | Context   | Non-Diagnostic | Chunk 2            | No Modification | 88       |       |
| 5   | 0001  | Context   | Non-Diagnostic | Chunk 3            | No Modification | 60       |       |
| 6   | 0001  | Context   | Non-Diagnostic | Flake 1            | No Modification | 28       |       |
| 7   | 0001  | Context   | Non-Diagnostic | Flake 2            | No Modification | 188      |       |
| 8   | 0001  | Context   | Non-Diagnostic | Flake 3            | No Modification | 199      |       |
| 9   | 0002  | Context   | Non-Diagnostic | Blade-like Flake 2 | No Modification | 20       |       |
| 10  | 0002  | Context   | Non-Diagnostic | Blade-like Flake 3 | No Modification | 21       |       |

# FLIRT - Flexible Lithic Recording Tool (R Shiny, JavaScript)

Retrieves list of loci that match the specified LocusType (excavation context, transect collection point, aeolianite deposits, etc) in the `AllLoci` table

**SNAP Lithics Processing**

|               |                       |                              |                  |                         |          |
|---------------|-----------------------|------------------------------|------------------|-------------------------|----------|
| Locus<br>0000 | Locus Type<br>Context | Period<br>Lower Palaeolithic | Blank<br>Flake 1 | Modification<br>Uniface | Quantity |
|---------------|-----------------------|------------------------------|------------------|-------------------------|----------|

Submit | New Blank or Modification

|                      |                      |                              |
|----------------------|----------------------|------------------------------|
| New Blank            | New Modification     | Period<br>Lower Palaeolithic |
| <input type="text"/> | <input type="text"/> | <input type="text"/>         |
| Submit Blank         | Submit Modification  |                              |

This info get passed into an INSERT query when submit button is clicked

Submit button searches the respective table for existing records to avoid duplicates, and if it finds none then passes this info to an INSERT query and refreshes the Blank/Modification dropdowns to include the update

Submit button also 'splits' each record into a number of records specified in the Quantity field, and INSERTs those into the `level3` table, which is artefact-level analysis

All Level 2 All Level 3

Show 10 entries Search:

Each cell can be edited, which runs an UPDATE and subsequent SELECT query to make changes and refresh the data

|    | id   | Locus   | LocusType      | Period             | Blank           | Modification | Quantity | Notes |
|----|------|---------|----------------|--------------------|-----------------|--------------|----------|-------|
|    | All  | All     | All            | All                | All             | All          | All      | All   |
| 1  | 0001 | Context | Non-Diagnostic | Blade-like Flake 2 | No Modification |              | 22       |       |
| 2  | 0001 | Context | Non-Diagnostic | Blade-like Flake 3 | No Modification |              | 17       |       |
| 3  | 0001 | Context | Non-Diagnostic | Chips              | No Modification |              | 4        |       |
| 4  | 0001 | Context | Non-Diagnostic | Chunk 2            | No Modification |              | 88       |       |
| 5  | 0001 | Context | Non-Diagnostic | Chunk 3            | No Modification |              | 60       |       |
| 6  | 0001 | Context | Non-Diagnostic | Flake 1            | No Modification |              | 28       |       |
| 7  | 0001 | Context | Non-Diagnostic | Flake 2            | No Modification |              | 188      |       |
| 8  | 0001 | Context | Non-Diagnostic | Flake 3            | No Modification |              | 199      |       |
| 9  | 0002 | Context | Non-Diagnostic | Blade-like Flake 2 | No Modification |              | 20       |       |
| 10 | 0002 | Context | Non-Diagnostic | Blade-like Flake 3 | No Modification |              | 21       |       |

Junk

# CREATING A DATABASE ON THE SERVER

---

```
CREATE DATABASE odate1;
```

## CREATING USERS AND SETTING PERMISSIONS

```
CREATE USER 'lara'@'localhost' IDENTIFIED BY 'croft';
GRANT ALL PRIVILEGES ON odate1.*  
TO 'lara'@'%' IDENTIFIED BY 'croft';
FLUSH PRIVILEGES;
```

# Let's go!

1. ssh workshop@xxx.xxx.xxx.xxx
2. Input password: xxxxxxxx
3. Accept security certificate, type 'yes'
4. You're in! 

1. sudo mysql
  2. CREATE DATABASE '<database>';
  3. CREATE USER '<username>'@'localhost' IDENTIFIED BY '<password>';
  4. GRANT ALL PRIVILEGES ON '<database>'.\* TO '<username>'@'%' IDENTIFIED BY '<password>';
  5. FLUSH PRIVILEGES;
  6. exit
- 
1. Mysql -u <username> -p
  2. Enter your password
  3.

# INSERT

---

Inserts records into a table

```
INSERT INTO `odate1`.`contexts` (Context, Trench, ContextDescr  
VALUES (0001, 001, "The first context of the season. Loamy san
```

# UPDATE

---

Modifies cells within a table

```
UPDATE `odate1`.`contexts`  
SET `SedimentColour` = "Dark Brown"  
WHERE `Munsell` = "5YR 3/3";
```

# SELECT

---

Retrieves data from a table

```
SELECT `Context`, `Munsell`, `SedimentColour`  
FROM `contexts`  
WHERE `contexts`.`Trench` = 001;
```

# DELETE

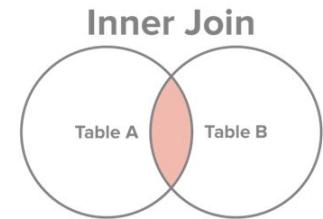
---

Deletes a record

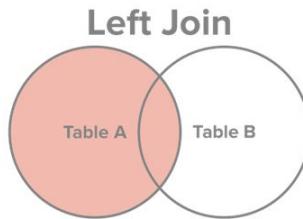
```
DELETE FROM `contexts`  
WHERE `Context` = "0005";
```

# JOIN

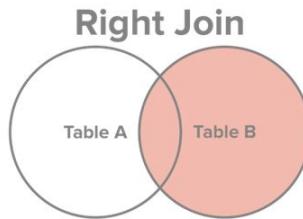
Used as optional component of SELECT queries when you want to integrate data spanning multiple tables



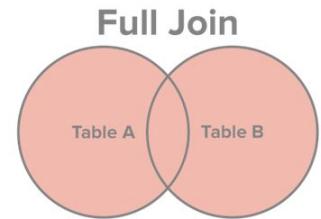
Select all records from Table A and Table B, where the join condition is met.



Select all records from Table A, along with records from Table B for which the join condition is met (if at all).



Select all records from Table B, along with records from Table A for which the join condition is met (if at all).



Select all records from Table A and Table B, regardless of whether the join condition is met or not.

# [INNER] JOIN

---

```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDat  
FROM Orders  
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID
```

# ACCESSING OUR DATABASE SERVER

1. Access the remote computer via SSH

```
ssh pi@<ip address>
```

2. Access the MariaDB database server

```
mysql -u MY_USERNAME -p
```