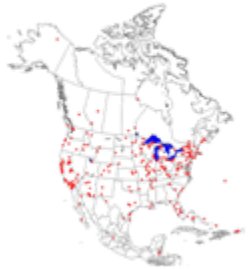
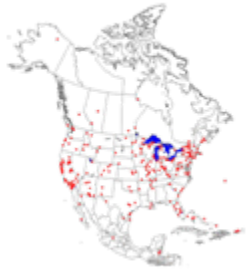
Data Query Using MySQL

John Kim
Field Station Programs
San Diego State University



What is SQL?

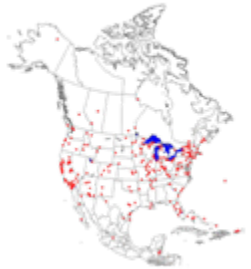
- **S**tructured **Q**uery **L**anguage
- ANSI standard computer language, yet many variants
- Can retrieve data from a database
- Can insert new records in a database
- Can delete records from a database
- Can update records in a database
- SQL is easy to learn, hard to master
- MySQL understands a subset/variant of SQL



What is SQL?

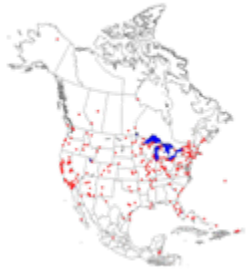
- "Syntax" = grammar for computer commands
- "Keywords" = words reserved & understood by a computer language
- Manual: http://dev.mysql.com/doc/
- A query = well-formed command. Can retrieve or manipulate (e.g., insert, update, delete).
- Query -> result -> rethink -> revise

SEEK KNB NCEAS OBFS LTER NSF



4 Basic Queries:

- **SELECT** - retrieves data from a database table
- **INSERT** - inserts new data into a table
- **UPDATE** - updates data in a database table
- **DELETE** - deletes data from a database table



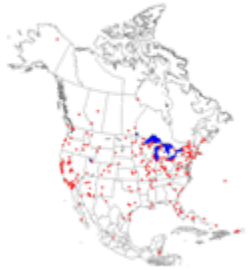
SELECT

1) SELECT cover FROM observation

2) SELECT cover FROM observation LIMIT 5

3) SELECT cover, height, count FROM observation LIMIT 5

4) SELECT * FROM observation LIMIT 5



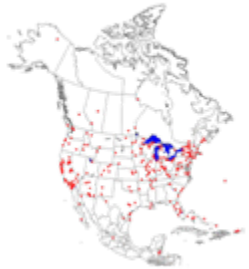
SELECT (continued)

1) **SELECT * FROM observation WHERE height > 10 and height < 20**

2) **SELECT * FROM observation WHERE height > 10 and height < 20 ORDER BY height**

3) **SELECT * FROM species WHERE species = “baaba”**

4) **SELECT * FROM species WHERE species LIKE “bo%”**



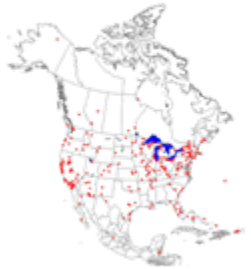
JOINS (finally!)

SELECT * FROM observation

SELECT * FROM *table1* **LEFT JOIN** *table2*
ON *match up two fields*

SELECT * FROM observation LEFT JOIN species
ON species_id = species_id

SELECT * FROM observation LEFT JOIN species
ON observation.species_id = species.species_id

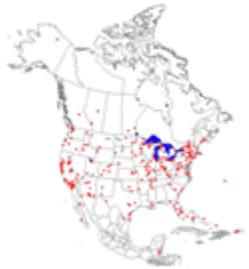


LEFT OUTER JOIN aka LEFT JOIN

SPECIES_ID	OBS	COVER	HEIGHT	COUNT
4	1	0.5	4	13
2	2	0.1	2	16
4	1	0.01	4	2
4	2	0.1	5	1
1	3	0.5	12	1
3	1	0.25	15	1

SPECIES_ID	NAME
0	LATR2
1	ERPU8
3	LEFE
4	GUSA2

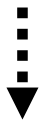
SPECIES_ID	OBS	COVER	HEIGHT	COUNT	SPECIES_ID	NAME
4	1	0.5	4	13	4	GUSA2
2	2	0.1	2	16		
4	1	0.01	4	2	4	GUSA2
4	2	0.1	5	1	4	GUSA2
1	3	0.5	12	1	1	ERPU8
3	1	0.25	15	1	3	LEFE



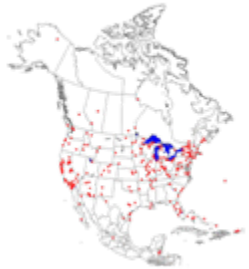
INNER JOIN

SPECIES_ID	OBS	COVER	HEIGHT	COUNT
4	1	0.5	4	13
2	2	0.1	2	16
4	1	0.01	4	2
4	2	0.1	5	1
1	3	0.5	12	1
3	1	0.25	15	1

SPECIES_ID	NAME
0	LATR2
1	ERPU8
3	LEFE
4	GUSA2



SPECIES_ID	OBS	COVER	HEIGHT	COUNT	SPECIES_ID	NAME
4	1	0.5	4	13	4	GUSA2
2	2	0.1	2	16		
4	1	0.01	4	2	4	GUSA2
4	2	0.1	5	1	4	GUSA2
1	3	0.5	12	1	1	ERPU8
3	1	0.25	15	1	3	LEFE



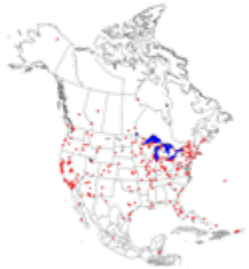
INSERT

Basic syntax:

INSERT INTO *table (list of fields)* **VALUES** *(list of values)*

INSERT INTO species (species) VALUES ('alin')

INSERT INTO location (site, web, plot, quad) VALUES ('P', 2, 'E', 1)



UPDATE

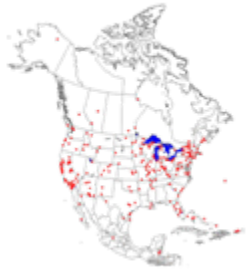
Basic syntax: **UPDATE** *table* **SET** *field* **=** *value*

SELECT * FROM VISIT

UPDATE visit SET date = curdate()

SELECT comments FROM observation
WHERE comments LIKE "never%"

UPDATE observation
SET comments = concat("Cannot identify as of ", curdate())
WHERE comments LIKE "never%"



DELETE

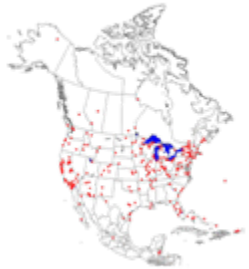
Basic syntax:

DELETE FROM *table* **WHERE** *(condition)*

DELETE FROM species WHERE species = 'acer saccharum'

But good to try a select first, before deleting:

SELECT FROM species WHERE species = 'acer saccharum'



Aggregate Functions

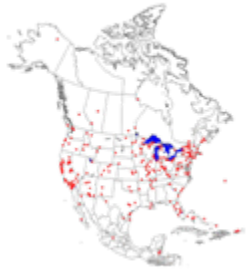
You can use function in place of plain column names.

SELECT height FROM observation

SELECT AVG(height) FROM observation

AVG(), STDDEV(), VARIANCE(), MAX(), MIN()

```
SELECT AVG(height) FROM observation
        LEFT JOIN species ON observation.species_id = species.species_id
        WHERE species.species = 'MUAR2'
```

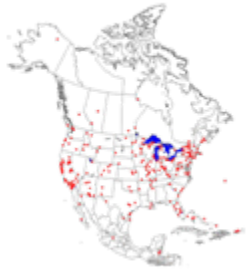


REVIEW

1) SQL basic commands:
   SELECT, INSERT, UPDATE, DELETE

2) LEFT JOIN vs. INNER JOIN

3) Aggregate functions, string comparisons & functions.



EXERCISES - I

1) An observation database is now in your database
2) "link" obs to your plant species table by creating a foreign key:

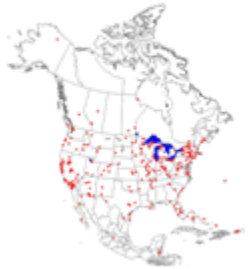
UPDATE obs, plantspecies
SET obs.plantspecies_id =plantspecies.plantspecies_id
WHERE obs.species_code=plantspecies.species_code

obs

observation_id	date	site	web	plot	quad	species_code	plantspecies_id	obs	cover	height	count	phen	comments
2000	1999-02-03	FPC	1	E	1	ERPU8	0	1	0.5	4	13	V	NA
2001	1999-02-03	FPC	1	E	1	ERPU8	0	2	0.1	2	16	V	NA
2002	1999-02-03	FPC	1	E	1	GUSA2	0	1	0.01	4	2	V	NA
2003	1999-02-03	FPC	1	E	1	GUSA2	0	2	0.1	5	1	V	NA

plantspecies

plantspecies_id	species_code	genus_name	species_name	life_form	variety_subspecies	author	family
2	ABFR2	Abronia	fragrans	H	na	Nutt._ex_Hook.	NYCTA
3	ACMIO	Achillea	millefolium	H	var._occidentalis	DC.	ASTER
4	ACNA2	Acourtia	nana	H	na	(Gray)_Reveal_&_King	ASTER
5	ACNE	Acalypha	neomexicana	H	na	Muell.-Arg.	EUPHO



EXERCISES - II

REFERENCE: http://dev.mysql.com/doc/

1. Select all rows with height less than 3.
2. Select all rows with height greater than 2 and phen = V.

3. Insert a new species row into the species database.

4. Find average height of all observations.

5. Find average height of all observations made in Feb 1999.