



Database Queries

John Kim

Field Station Programs
San Diego State University





talk outline

- Queries using SQL
- SQL by example
- Hands-on queries through PhpMyAdmin





SQL

- Structured Query Language
- ANSI standard computer language with many variants (I'll use MySQL)
- Read, write, calculate, modify data
- Easy to learn, hard to master





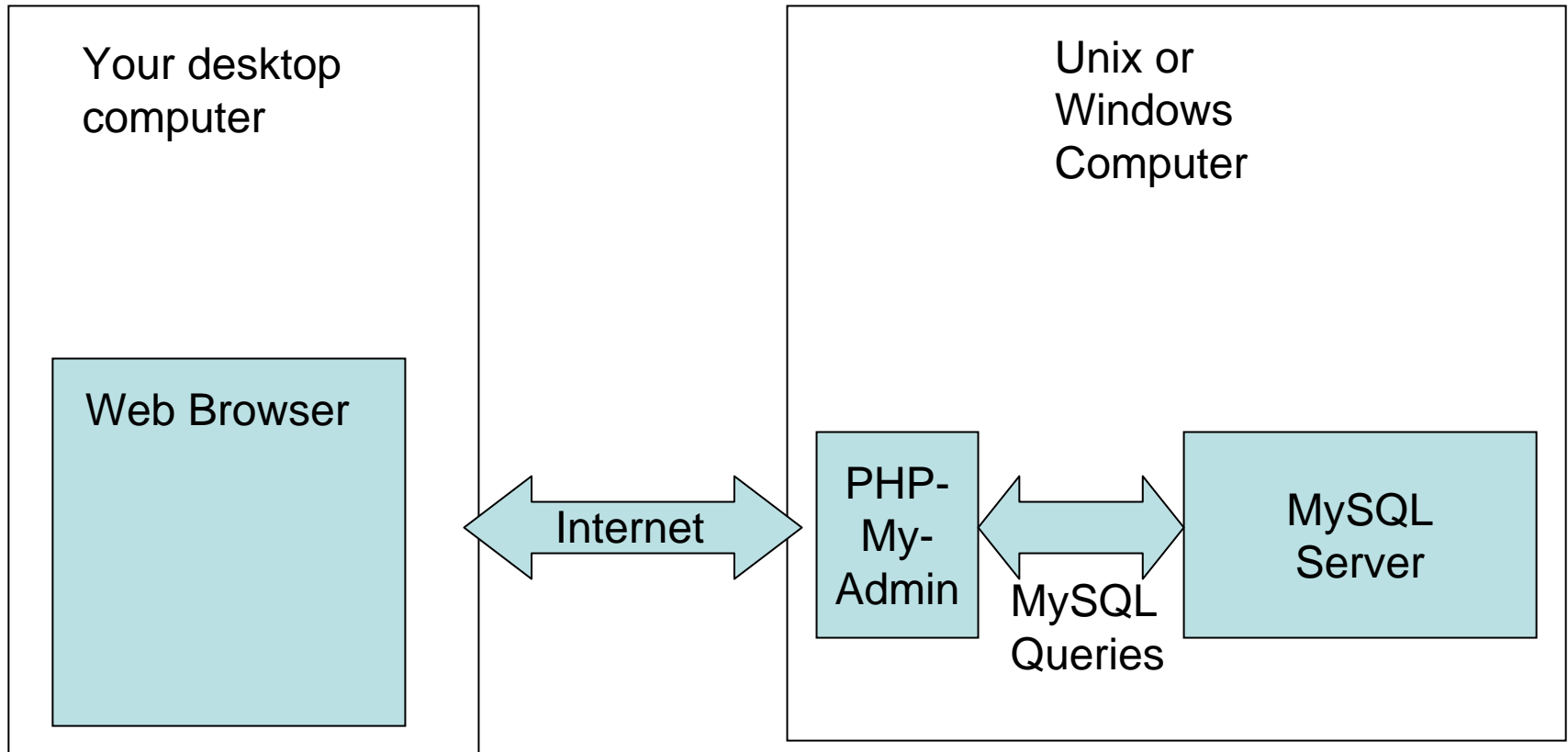
Querying Tips

- Build queries incrementally.
- Make a back-up copy of tables.
- Manual: <http://dev.mysql.com/doc/>





Introduction to PHPMyAdmin





phpMyAdmin 2.6.1 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Reload Links

Address <https://www.lternet.edu/mysql/> Go

Google Search 18 blocked Check

phpMyAdmin

Welcome to phpMyAdmin 2.6.1

Language:

Login

(Cookies must be enabled past this point.)

Server:

Username:

Password:



Server: marsh.lnternet.edu ▶ Database: rcn_admin ▶ Table: mytbl

Structure Browse SQL Search Insert Export Operations Empty Drop

InnoDB free: 10240 kB

	Field	Type	Collation	Attributes	Null	Default	Extra	Action				
<input type="checkbox"/>	sdf	int(10)		UNSIGNED	No		auto_increment					

Check All / Uncheck All With selected:

Print view Relation view Propose table structure?

Add field(s) ☒ At End of Table ☐ At Beginning of Table ☐ After sdf Go

Indexes: ?

Keyname	Type	Cardinality	Action	Field
PRIMARY	PRIMARY	2		sdf

Create an index on columns Go

Space usage:

Type	Usage
Data	16,384 Bytes
Index	0 Bytes
Total	16,384 Bytes

Row Statistic:

Statements	Value
Format	fixed
Collation	latin1_swedish_ci
Next Autoindex	6
Creation	Oct 31, 2005 at 02:29 PM

Run SQL query/queries on database rcn_admin?

Fields:

SELECT * FROM 'mytbl' WHERE 1

sdf

<<

☒ Show this query here again

Go



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

Basic Syntax: **SELECT** *columns* **FROM** *table*

- 1) get specific columns from a table:

```
SELECT cover, height FROM observation
```

- 2) get all columns:

```
SELECT * FROM observation
```

- 3) get specific rows from a table:

```
SELECT cover, height FROM observation  
WHERE cover > 1
```





SELECT (continued)

More complex conditions:

```
SELECT * FROM observation  
WHERE ( height > 10 AND height < 20 ) OR count > 10
```





JOINS

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

SPECIES_ID	NAME
0	LATR2
1	ERPU8
3	LEFE
4	GUSA2



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

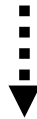




INNER JOIN

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

SPECIES_ID	NAME
0	LATR2
1	ERPU8
3	LEFE
4	GUSA2



SPECIES_ID	COVER	HEIGHT	COUNT	SPECIES_ID	NAME
4	0.5	4	13	4	GUSA2
2	0.1	2	16		
4	0.01	4	2	4	GUSA2
4	0.1	5	1	4	GUSA2
1	0.5	12	1	1	ERPU8
3	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 observation  
WHERE comments = "NA"
```

```
UPDATE observation SET comments = "Not Applicable"  
WHERE comments = "NA"
```

```
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 AVG(height) FROM observation
```

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

Calculate the average height of sand muhly (MUAR2):

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





EXERCISES

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

1. Get all observations with height less than 3.
2. Get all observations for one site.
3. Get all observations for one species at one site.
(hint: do join twice)
4. Insert a new species into the species table.

