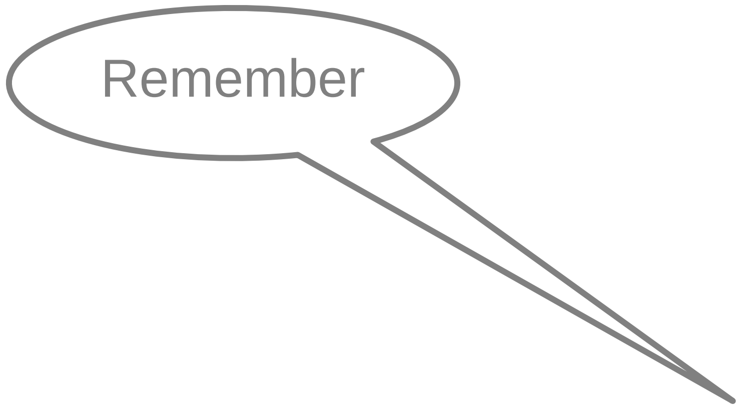
Modifying DBs and using MySQL

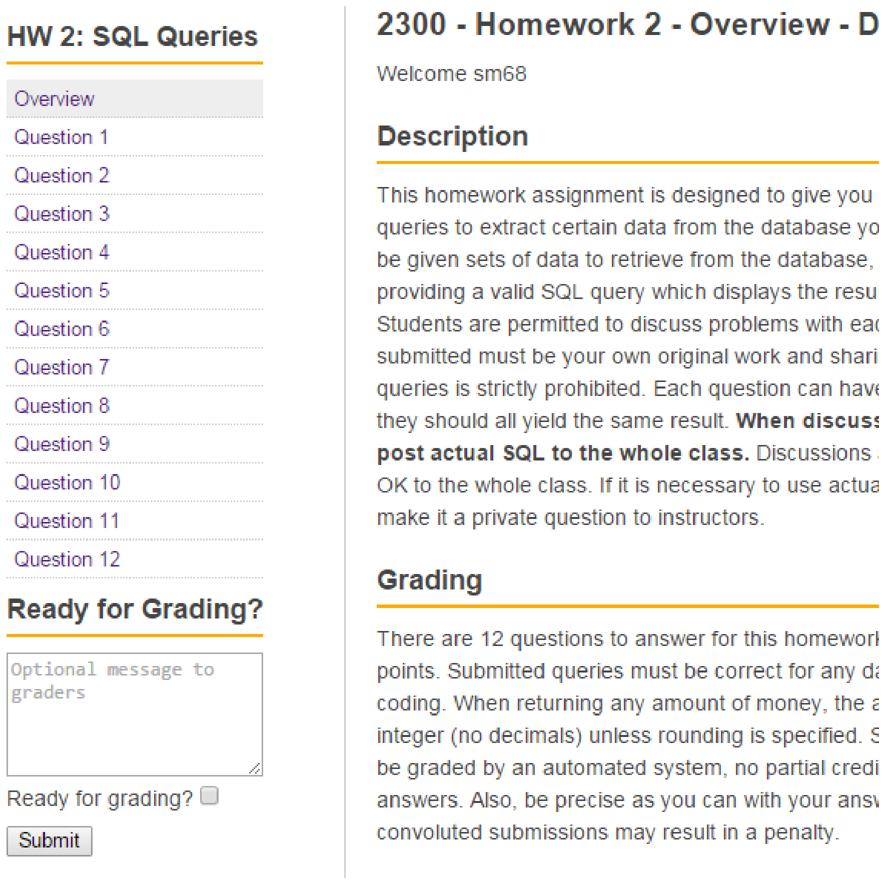
INFO/CS 2300: Intermediate Web Design and Programming

1 sheet on half wall

HW 2

Due Tuesday, March 7 at 5 pm





HW 2: SQL Queries

Overview
Question 1
Question 2
Question 3
Question 4
Question 5
Question 6
Question 7
Question 8
Question 9
Question 10
Question 11
Question 12

Ready for Grading?

Optional message to graders

Ready for grading? ☐

Submit

2300 - Homework 2 - Overview - D

Welcome sm68

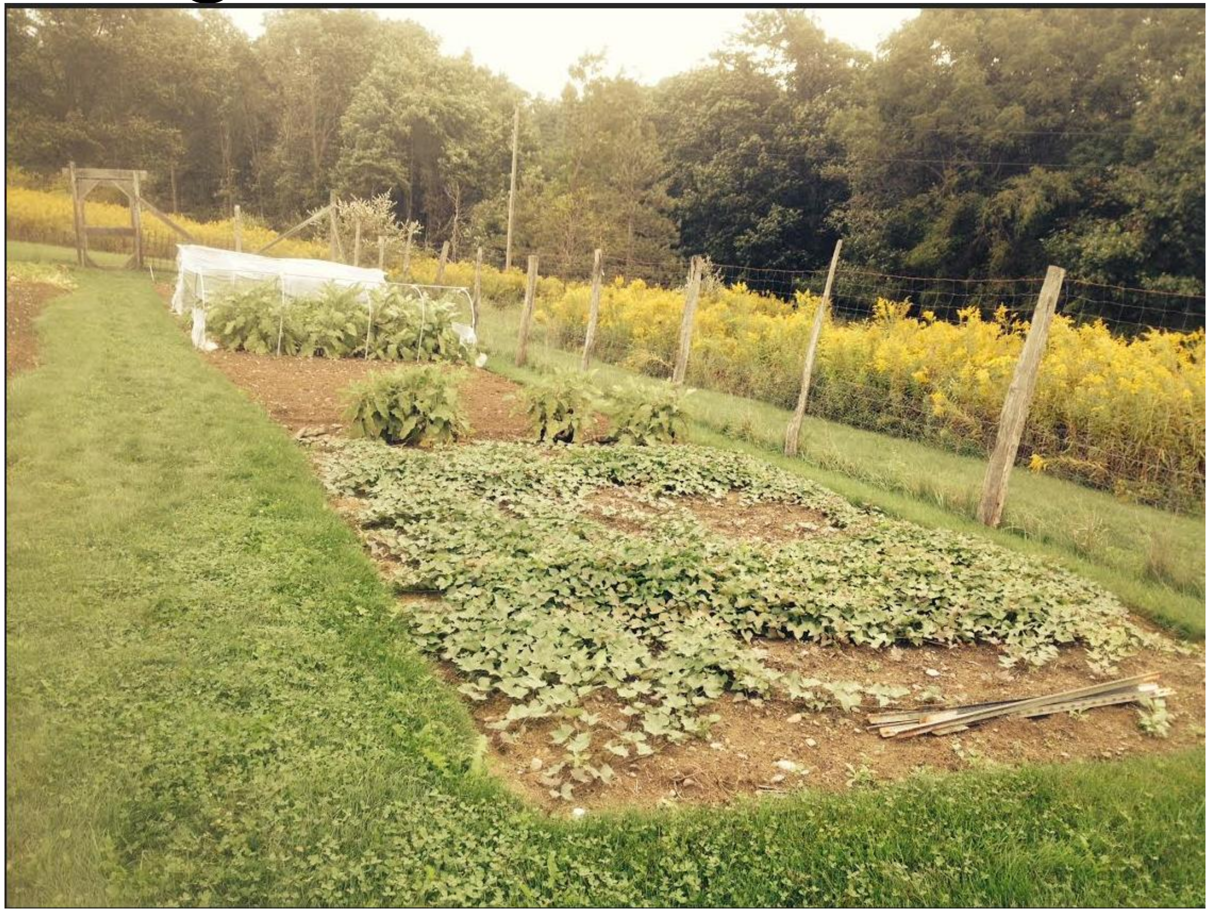
Description

This homework assignment is designed to give you
queries to extract certain data from the database yo
be given sets of data to retrieve from the database,
providing a valid SQL query which displays the resu
Students are permitted to discuss problems with ea
submitted must be your own original work and shari
queries is strictly prohibited. Each question can hav
they should all yield the same result. **When discus
post actual SQL to the whole class.** Discussions
OK to the whole class. If it is necessary to use actua
make it a private question to instructors.

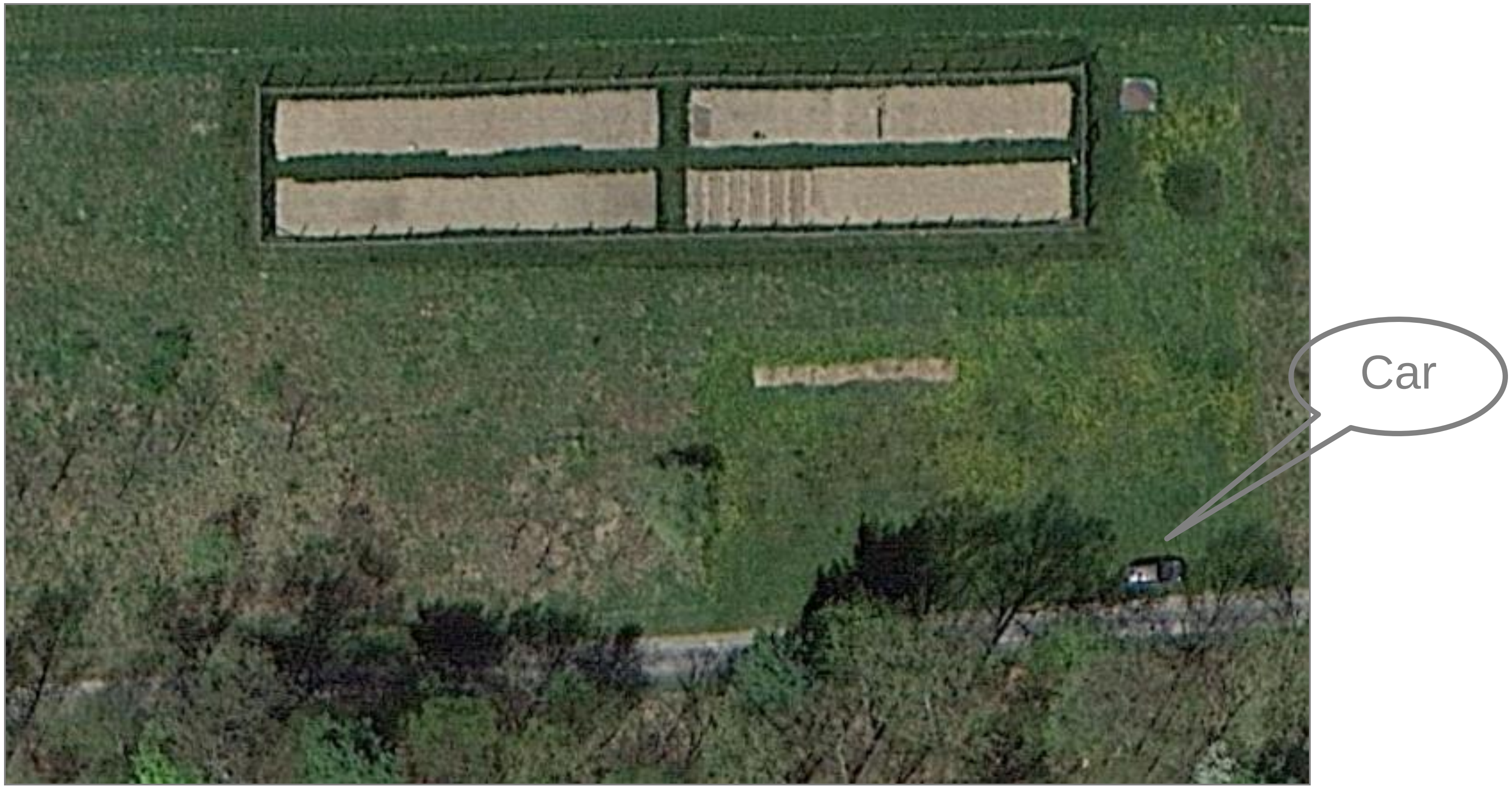
Grading

There are 12 questions to answer for this homewor
points. Submitted queries must be correct for any d
coding. When returning any amount of money, the
integer (no decimals) unless rounding is specified.
be graded by an automated system, no partial credi
answers. Also, be precise as you can with your ans
convoluted submissions may result in a penalty.

Planning a database



Steve's Garden



Garden scenario

Growing 30-40 different kinds (cultivars) of vegetables each year (spinach, tomatoes, etc)

Fill roughly 40 beds (locations) each year

Lots of different actions with each (seeding, transplanting, watering, weeding)

Some actions are measured (quantity)
1, 2, 3, 4…

Quantities have units (plants, inches, feet, pounds, etc)

Sometimes its nice to add a note

Garden scenario reporting

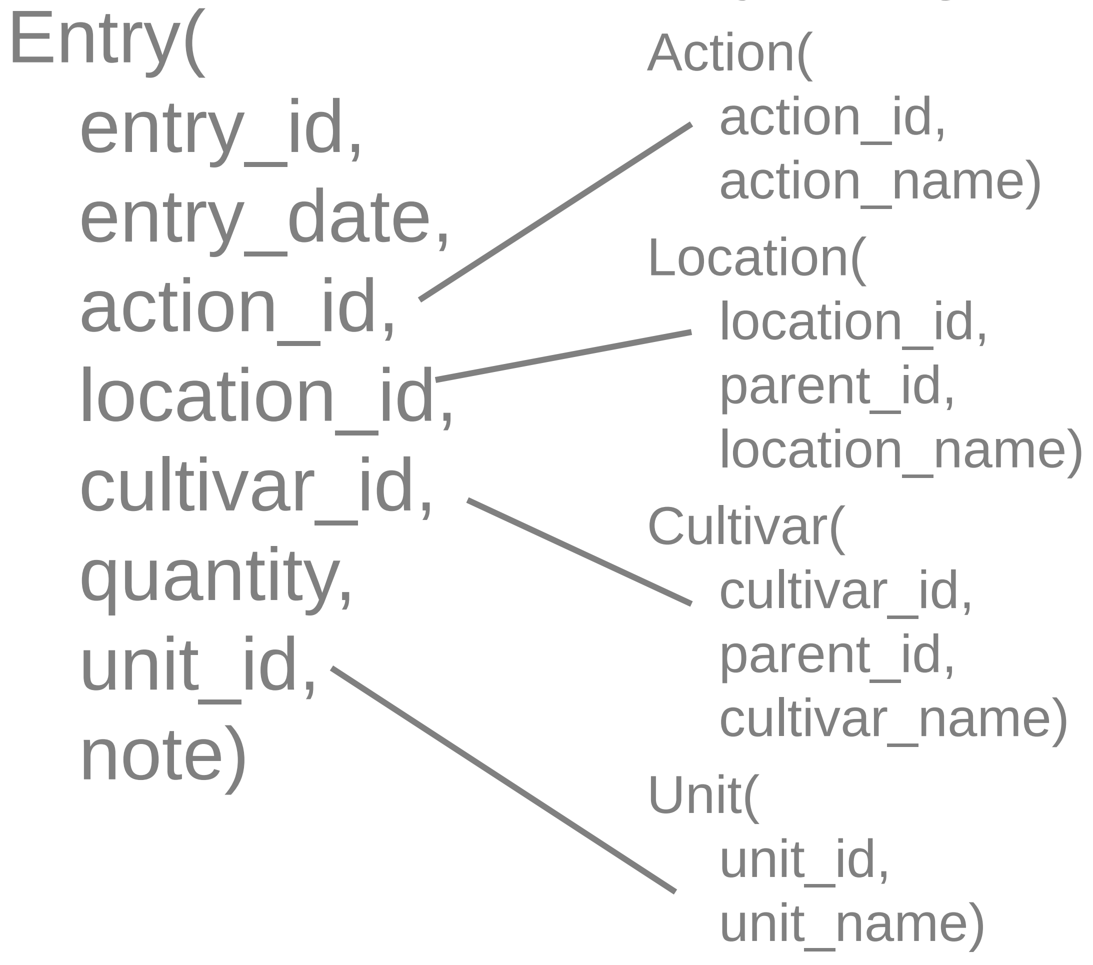
I need to be able to reliably look at what was planted in any given bed (location) so I don't repeat too soon.

I need to be able to look at any given kind of plant (cultivar) to see what actions I took when in previous years.

I like to know how much I harvested of a particular plant or from a particular location

For consistent data entry, I want to pick from lists rather than enter text

Garden activity log schema



```
Entry(
   entry_id,
   entry_date,
   action_id,
   location_id,
   cultivar_id,
   quantity,
   unit_id,
   note)
```

```
Action(
   action_id,
   action_name)
Location(
   location_id,
   parent_id,
   location_name)
Cultivar(
   cultivar_id,
   parent_id,
   cultivar_name)
Unit(
   unit_id,
   unit_name)
```

Click In!

Click In!

Which is equivalent to this statement

SELECT DISTINCT last_name FROM students

SELECT last_name FROM students ______________

A. LIMIT 1
B. WHERE last_name IS DISTINCT
C. GROUP BY last_name HAVING COUNT(last_name) = 1;
D. INNER JOIN students AS students2 ON students.last_name = students2.last_name
E. GROUP BY last_name

Click In!

Which is equivalent to this statement

SELECT DISTINCT last_name FROM students

SELECT last_name FROM students ______________

A. LIMIT 1
B. WHERE last_name IS DISTINCT
C. GROUP BY last_name HAVING COUNT(last_name) = 1;
D. INNER JOIN students AS students2 ON students.last_name = students2.last_name
E. GROUP BY last_name

Modifying databases via SQL

We can `INSERT` records into a table, `UPDATE` records, and `DELETE` records from a table.

Create
Read
Update
Delete

Browse
Read
Edit
Add
Delete

INSERT

Title	Year	Length
Gladiator	2000	155
A Beautiful Mind	2001	135
Chicago	2002	113
The Return of the King	2003	201
Million Dollar Baby	2004	132

What do I need to add a movie?

```
INSERT INTO
VALUES
```

```
INSERT INTO          (Length, Title, Year)
VALUES
```

INSERT generalized

```
INSERT INTO table (field1, field2, …, fieldk);
VALUES
   (value1, value2, …, valuek),
   (value1, value2, …, valuek),
   (value1, value2, …, valuek);
```

Specifying the field names is not required.

If fields are not specified and a field is added to this table, what happens if this query is not rewritten?

INSERT

title	year	length
Gladiator	2000	155
Chicago	2002	113

name	title	year
Russell Crowe	Gladiator	2000
Russell Crowe	A Beautiful Mind	2001
Viggo Mortensen	Return of the King	2003
Hillary Swank	Million Dollar Baby	2004

How do I create movies from the StarsIn table?

```
INSERT INTO Movies(title, year)
  SELECT DISTINCT title, year
  FROM StarsIn
```

What goes here so we don’t add duplicates

INSERT

title	year	length
Gladiator	2000	155
Chicago	2002	113

name	title	year
Russell Crowe	Gladiator	2000
Russell Crowe	A Beautiful Mind	2001
Viggo Mortensen	Return of the King	2003
Hillary Swank	Million Dollar Baby	2004

How do I create movies from the StarsIn table?

```
INSERT INTO Movies(title, year)
  SELECT DISTINCT title, year
  FROM StarsIn
  LEFT OUTER JOIN Movies
      ON Movies.title = StarsIn.title
      AND Movies.year = StarsIn.year
  WHERE Movies.title IS NULL;
```

Don't include movies already in Movies

UPDATE

Title	Year	Length
Gladiator	2000	155
A Beautiful Mind	2001	135
Chicago	2002	113
The Return of the King	2003	201
Million Dollar Baby	2004	132

How do I change the length of all movies to hours not minutes?

```
UPDATE
SET LENGTH = LENGTH / 60
```

UPDATE generalized

```
UPDATE table
SET field = expression
WHERE condition;


-- oops
UPDATE
SET Title = 'Moonlight',

WHERE
```

INSERT / UPDATE

They can be combined – mostly makes sense when using natural keys:

```
INSERT INTO table
   (`netid`, `query_number`, `query`)
   VALUES (?, ?, ?)
ON DUPLICATE KEY UPDATE `query` = ?,
   `timestamp` = ?;
```

Primary key is netid and query_number combined

This is the INSERT query for HW2 submissions

DELETEs

```
DELETE
FROM table
WHERE condition;
```

E.g.

```
DELETE
FROM Movies
WHERE Title IN (SELECT Title
                FROM StarsIn
                WHERE Name='Tom Hanks');
```

Add a new blue boat, id 105, named ‘Clipper’.

Increase the rating of every sailor by 1.

Remove every sailor whose age is over 65.

Add a new blue boat, id 105, named ‘Clipper’.

```
INSERT INTO Boats
  VALUES( 105, ‘Clipper’, ‘blue’);

INSERT INTO Boats (boatID, boatName, color)
  VALUES( 105, ‘Clipper’, ‘blue’);
```

Increase the rating of every sailor by 1.

```
UPDATE Sailors
  SET rating = rating + 1;
```

Remove every sailor whose age is over 65.

```
DELETE FROM Sailors
  WHERE age > 65;
```

Creating tables in SQL

CREATE TABLE

To make a table in a database, we use the CREATE TABLE command.

```
CREATE TABLE table_name (
  field1 type1,
  field2 type2,
  …
  fieldk typek
);
```

MySQL numeric field types

Common numeric types:

- int / integer (size)
- tinyint (size)
- bigint (size)
- float (size,d)
- double (size,d)
- decimal (size,d)

Boolean:

- tinyint(1)
- bit

size = max number of digits
d = digits to the right of the decimal point

More details:
http://www.w3schools.com/sql/sql_datatypes.asp

MySQL text field types

Common text types:

- char(*m*): string of exactly *m* characters (spaces added if necessary)
- varchar(*m*): string of up to *m* characters
- text: string of up to 64K bytes
- blob: a "binary large object" up to 64K bytes long

varchar and text are most common

More fields and details:
http://www.w3schools.com/sql/sql_datatypes.asp

MySQL date field types

Common text types:

- date: a date in YYYY-MM-DD format
- time: a time in HH:MM:SS format
- datetime: date & time in YYYY-MM-DD HH:MM:SS format
- year: year in YY or YYYY format
- timestamp: the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD HH:MM:SS

More details:
http://www.w3schools.com/sql/sql_datatypes.asp

E.g.

```
CREATE TABLE Movies (
  title varchar(150),
  year year,
  length int(5)
);
```

Don't use keywords for field names. Could put the field name in quotes but… just don't do it

Not null

We can impose that certain fields are not null.

```
CREATE TABLE Movies (
  movie_title varchar(150) NOT NULL,
  movie_year year NOT NULL,
  minutes int(5)
);
```

Default values

We can specify the default value for some fields – to be used when no value is given when creating a record.

```
CREATE TABLE Movies (
  movie_title varchar(150) NOT NULL,
  movie_year year NOT NULL DEFAULT 2002,
  minutes int(5) DEFAULT 120,

  PRIMARY KEY (movie_title, movie_year)
);
```

Primary key

What has to be true of a primary key?

Primary key MySQL

```
CREATE TABLE relation (
  field1 type1 NOT NULL,
  field2 type2 NOT NULL,
  field3 type3,
  …
  fieldk typek

  PRIMARY KEY (field1, field2)
);
```

Auto increment

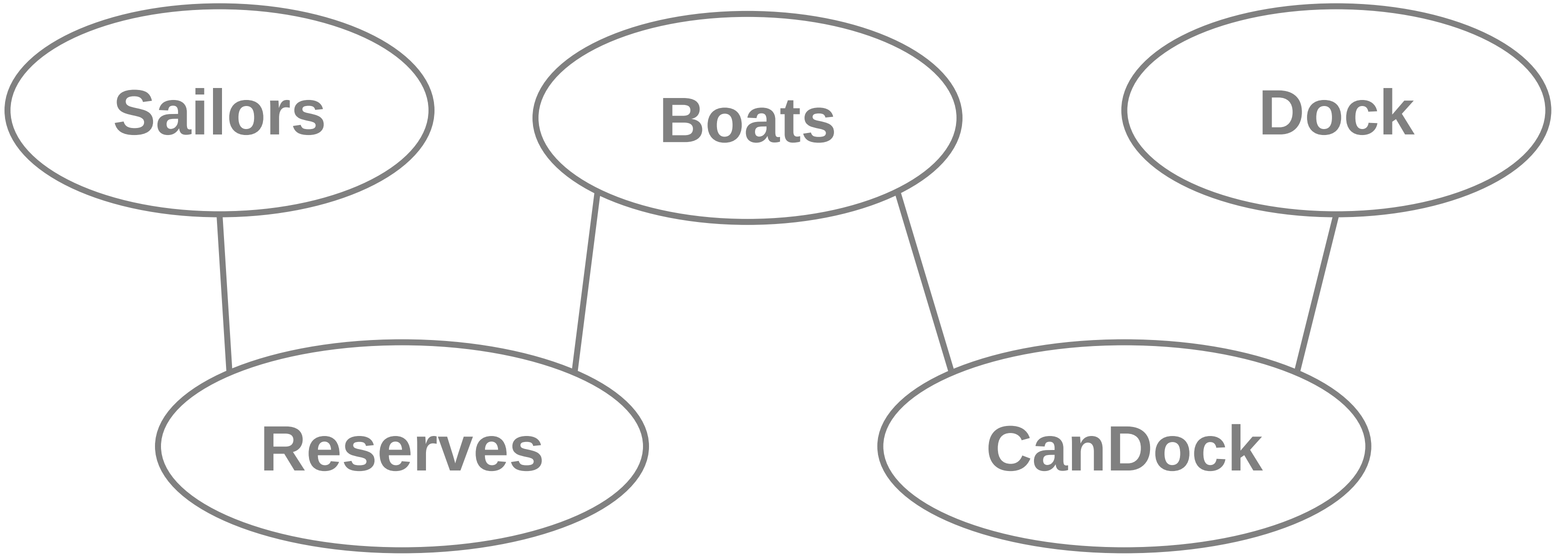
```
CREATE TABLE Students {
  id int(5) auto_increment,
  name varchar(50);
}
```

What does this do and why would it be useful?

Create new tables for the database with the following schema:

Dock(<u>dockId: integer</u>, dockDescription: string)

CanDock(<u>boatId: integer, dockId: integer</u>)



Create new tables for the database with the following schema:

Dock(did: integer, ddescription: string)

CanDock(bid: integer, did: integer)

```
CREATE TABLE Dock (
  dockId int(5) NOT NULL AUTO_INCREMENT,
  dockDescription varchar(255),
  PRIMARY KEY  (`dockId`)
);
CREATE TABLE CanDock (
  boatId int(5) NOT NULL,
  dockId int(5) NOT NULL
  canDock tinyint(1),
  PRIMARY KEY ( boatId, dockId ),
);
```

Quote is required if the field name has spaces but avoid spaces

This field may or may not be necessary depending on whether false is separate from null

Using phpMyAdmin

XAMPP phpMyAdmin

phpMyAdmin is part of the XAMPP install we recommended earlier, so if you installed XAMPP on you own machine, you already have it.

http://localhost/phpmyadmin/

Or

http://localhost/phpMyAdmin

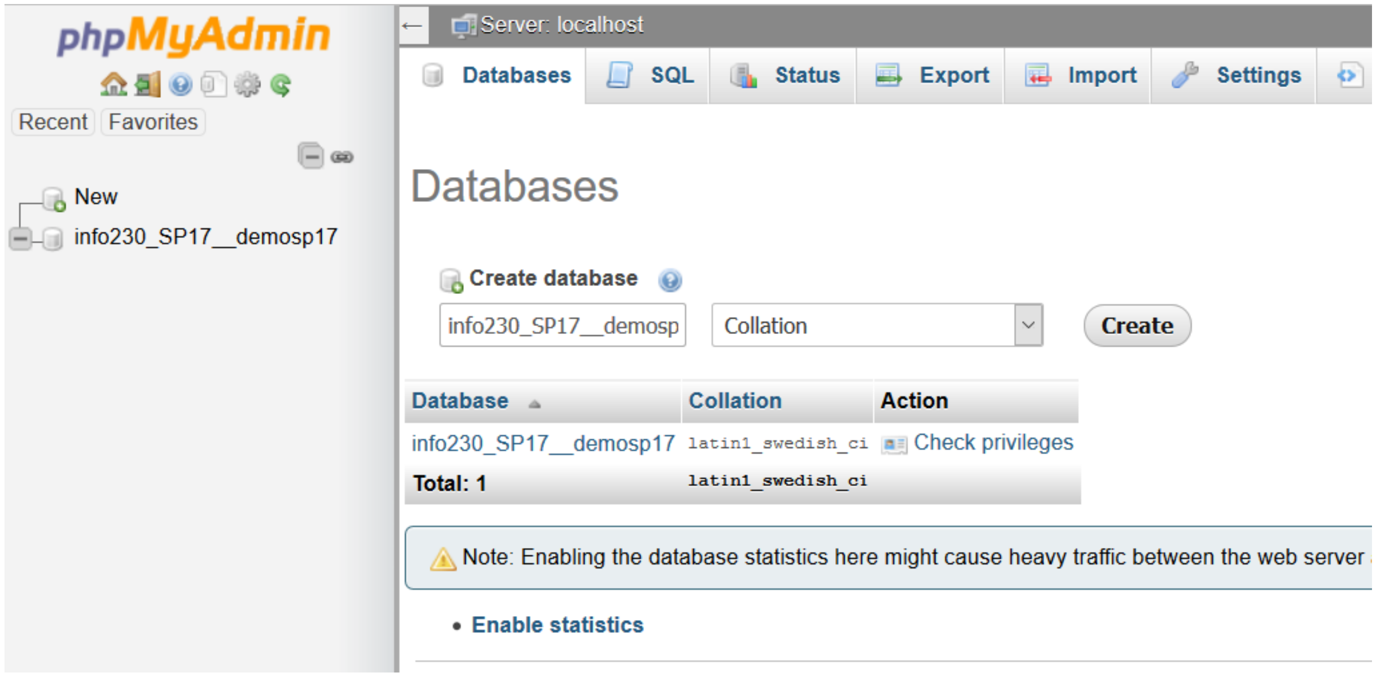
See your documentation

MySQL DBs on the course server

Each user on the 2300 server has a MySQL DB info230_SP17_*username*. You can use it for your upcoming projects and to do whatever experimentation you want (within reason…)

phpMyAdmin

https://info2300.coecis.cornell.edu/phpMyAdmin/



Databases

Create database

info230_SP17__demosp | Collation | Create

Database	Collation	Action
info230_SP17__demosp17	latin1_swedish_ci	Check privileges
Total: 1	latin1_swedish_ci	

Note: Enabling the database statistics here might cause heavy traffic between the web server

- Enable statistics

Log in with your course server username and password to continue.

Create and populate 2 tables

```
CREATE TABLE Movies (
  movie_title VARCHAR(150) NOT NULL,
  movie_year YEAR NOT NULL,
  minutes INT(5),

  PRIMARY KEY (
    movie_title,
    movie_year)
);

CREATE TABLE StarsIn (
  star_name VARCHAR(50) NOT NULL,
  movie_title VARCHAR(150) NOT NULL,
  movie_year YEAR NOT NULL,

  PRIMARY KEY (
    star_name,
    movie_title,
    movie_year)
);
```

```
INSERT INTO `Movies`
(`movie_title`, `movie_year`, `minutes`) VALUES
('Gladiator', 2000, 155),
('Crouching Tiger, Hidden Dragon', 2000, 120),
('Moulin Rouge', 2001, 127),
('A Beautiful Mind', 2001, 135),
('Chicago', 2002, 113),
('Lost in Translation', 2003, 102),
('The Return of the King', 2003, 201),
('Million Dollar Baby', 2004, 132);

INSERT INTO `StarsIn`
(`star_name`, `movie_title`, `movie_year`) VALUES
('Hillary Swank', 'Million Dollar Baby', 2000),
('Russell Crowe', 'A Beautiful Mind', 2001),
('Russell Crowe', 'Gladiator', 2000),
('Viggo Mortensen', 'The Return of the King',
   2003);
```

A familiar query

```
SELECT
  Movies.movie_title,
  StarsIn.movie_name
  Movies.minutes
FROM Movies
INNER JOIN StarsIn
  ON
  Movies.movie_title = StarsIn.movie_title
    AND
    Movies.movie_year = StarsIn.movie_year;
```

More familiar queries

```
SELECT
  movie_title,
  movie_year,
  minutes
FROM Movies
WHERE
  minutes >(SELECT AVG(minutes) FROM Movies);

SELECT
  movie_year,
  AVG(minutes) AS AvgLength
FROM Movies
GROUP BY movie_year;
```

Review

- SQL allows us to state constraints on the data in the CREATE TABLE statement, including domain constraints and key constraints.
- We're now ready to exercise our SQL skills in the MySQL DB, either through installs on own machine or via phpMyAdmin on the INFO 2300 server.