Welcome to SI564

SQL and Databases

SCHOOL OF INFORMATION
UNIVERSITY OF MICHIGAN

No smoking

- Please do not smoke ANYTHING including marijuana before class, during class or after class.
- We have someone in class with a SEVERE allergy.

First homework

Some notes:

1) **Academic integrity**
   I. **Do not copy and paste. Make sure to use your own words.**
2) **If you are missing a database required for a homework, please let us know asap.**
3) **Average grade was 90%.**
   I. **This does not include points taking off for being late.**

Office Hours

- If you would like to come to office hours, please book online.
- If those dates/times conflict with a class, let us know.

Ambiguity

- This class has a decent amount of intentional ambiguity (and some unintentional ambiguity).
- It is always ok to ask clarifying questions as one would do in a job.
- If there is a misunderstanding, you are always welcome to redo an assignment for full credit as you would do in a job.
- Some classes at SI, make the formats explicit as to what they are looking for, this is **not** the way intermediate and senior titles work.

How to decide how to do work?

- Most important thing is to explain your choice and be open to change.
- Unlike other classes, I am not really dictating *how* to do the work.
- You can choose your best approach, if myself or Mariah do not agree, present your case. If you have two paths when you turn something in, explain which route you took and why.
- You can always ask for clarification.

Contacting us

- Please do not use canvas to contact us.
- Slack is the best method.
- Slack DM’s are ok for either of us.

Attendance

- Please log your attendance in canvas.
- Login. Under week 3, choose today's date.
- SELECT SQL
  - Is the code
- (all caps, 1 space)

Slides

- By request, slides will be posted.
- However, attendance may be taken.
- I will do my best to have the slides to you before class so you can take notes on them.

Jupyter has changed

- To open a terminal, click new
  - Terminal

Databases are large

- Almost never do we want ALL the data in a table.

WHERE

- Today we are going to discuss how to filter data from a table using the WHERE part of the select statement.
- The same syntax we learn this week (and last week) is used on SELECT, UPDATE, and DELETE queries.

Diagram

- SELECT fielda, fieldb as B, FROM table where field=1 ORDER BY field DESC limit 5;
- _verb ____what____FROM WHERE __ORDER BY____LIMIT;

WHERE

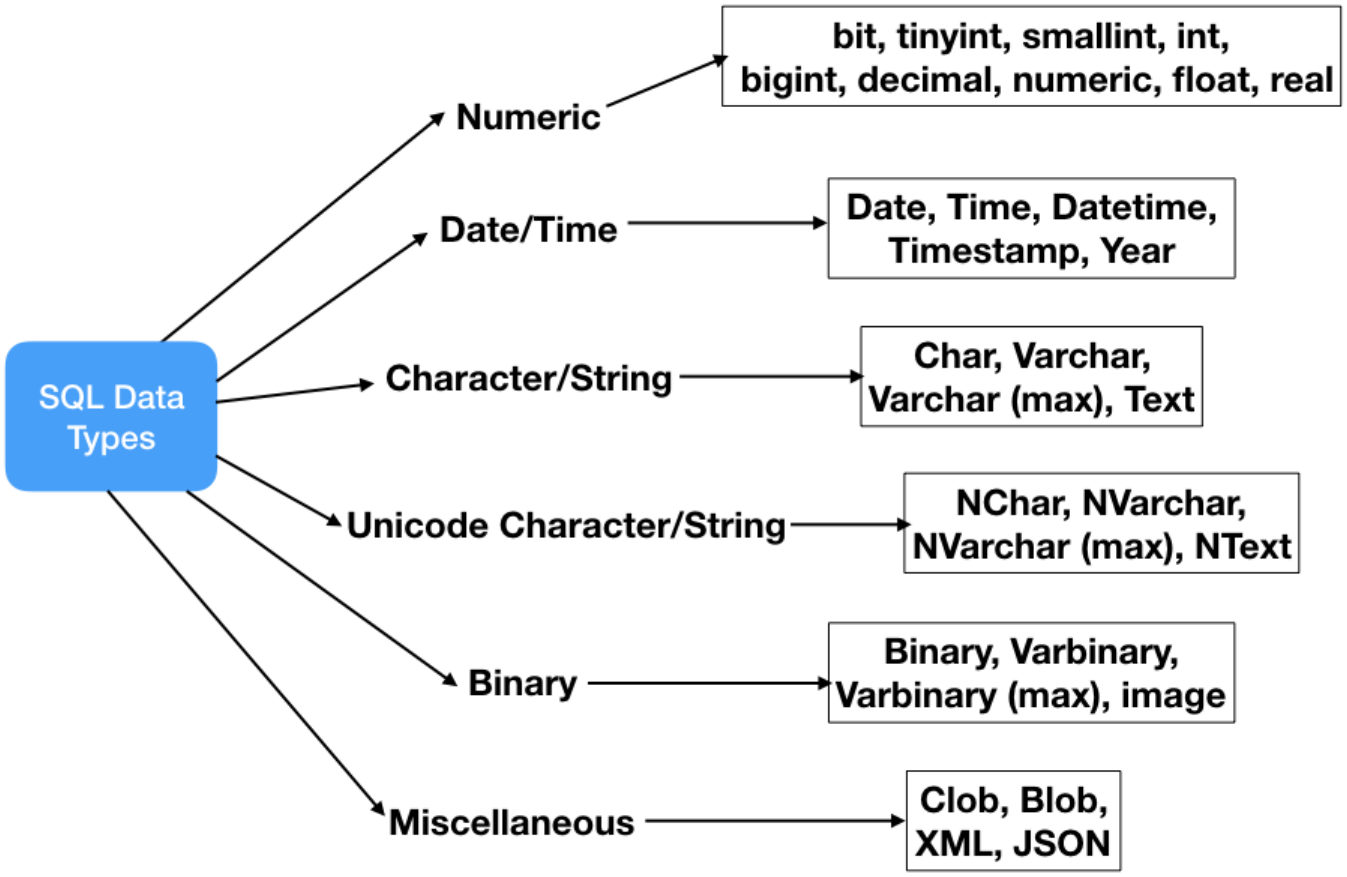
- The simple form of this is
- WHERE field = SOMETHING;

Table Diagrams

Before you start to query any data, make sure you make a table diagram first.

- We started on this last week. This does not end at this week.
- There was some confusion around the word "Diagram" - for this week lets stick to text diagrams/descriptions.
- Next week, we will start with image diagrams.

Field Types


SQL Data Types
Numeric
bit, tinyint, smallint, int, bigint, decimal, numeric, float, real
Date/Time
Date, Time, Datetime, Timestamp, Year
Character/String
Char, Varchar, Varchar (max), Text
Unicode Character/String
NChar, NVarchar, NVarchar (max), NText
Binary
Binary, Varbinary, Varbinary (max), image
Miscellaneous
Clob, Blob, XML, JSON

Field Types

Fall into groups?

- Int
- Text (char, varchar)
- Other

Field Types

- When you query a field, you need to be aware of the type
- somerandom_number = "123"
- somerandom_number = 123
- somerandom_string = "123"
- somerandom_string = 123

To start with Number or not?

- If a number, don't use quotes.
- If not a number use quotes.
- Some datatypes are special.
- MySQL may display warnings if you get this wrong, other RDBMS's won't.

AND

Compound Queries

- Sometime you want to query 2 fields at the same time.
- WHERE field1 = “blah” AND field2=“BLAH”
  - Will only return if both match

OR

- OR queries let you match on either or both parts of a question
- WHERE field1 = "yes" OR field2=1;

Field1	Field2	Match?
Yes	0	✓
No	0	✗
No	1	✓
Yes	1	✓

IN

- Assume you have a list of items you want to match on. Say students in a class. You have their id numbers 1, 5, 9,10;
- You could write
  - WHERE stuid = 1 OR stuid=5 OR stuid=9 OR stuid=10;
  - However, this is hard to read and does not really scale.
- IN lets you write
  - WHERE stuid IN (1,5,9,10);
- With strings
  - WHERE name IN ('Michael','Raj','Kelly');
- Not used with consecutive numbers.
  - ~~WHERE id in (1,2,3,4,5,6)~~

BETWEEN

- WHERE field1 BETWEEN val1 and val2;
- BETWEEN is INCLUSIVE.
- WHERE id BETWEEN 100 AND 200;
  - WHERE id >= 100 AND id <= 200;
- BETWEEN makes it a tad easier to read.
- Often used with dates and number ranges.
  - WHERE SIGNUP between ('2019-02-02') and NOW()
- Not often used with strings.

LIKE

- Like allows for string matching
- WHERE lastname LIKE ‘smi%’;
- % stands as a wild card.
- Let’s you search text fields.
- This can be slow.

Combine like queries

- SELECT * FROM table where first_name like ‘m%’ and last_name like ‘j%’;

LIKE

- SELECT * FROM names WHERE firstname like 't_m';
- The underscore will match 1 character only. In this case the o in Tom.

LIKE

- Escape is \
- SELECT * FROM discountcodes WHERE code LIKE '%umich\%20off';
- Would match
- umsi_umich%20off
- Since % is in the string, we need to escape it with a \ and tell MySQL not to use it as a wild card.

Full Text Searching

- SELECT * FROM taxdata WHERE MATCH (name,purpose) AGAINST ('+good –bad');
  - We can't use this yet, but we will be able to soon. (week 9)

REGEXP

- If LIKE is not enough you can use REGEX searches.
- This only works on string like fields.
- SELECT name FROM primary_name WHERE name REGEXP '^(A|B|C)'

REGEXP Case sensitive

- If LIKE is not enough you can use REGEX searches.
- This only works on string like fields.
- SELECT name FROM primary_name WHERE name REGEXP BINARY '^(A|B|C)'
- Force case sensitive string comparison by doing a binary compare.

subquery

- Assume 2 tables
- Table 1: Orders
- Table 2 Customers
- Assume you want to query orders for active customers.
- SELECT * FROM orders WHERE customerid in (SELECT id FROM customers WHERE active=1);
- This uses the IN syntax but rather than pull from a list, it pulls from a query.

Sub Query

You can also do same table subqueries.

(You can also do this with an AND clause, but later there are reasons to do this)

- select * from taxdata where ein in (select ein from taxdata where year = '2013') and purpose like '%good%';

Sub Query

- select * from account where open_emp_id in (select emp_id from employee where dept_id =3);
- Give me all the accounts that were opened by someone in dept3;

NOT Syntax

- Sometimes you want to get records when something is not true.
- Using the NOT syntax works for SQL that does not have a mathematical function

NOT

- SELECT * FROM table WHERE field NOT LIKE 'blah%';
- SELECT * FROM table WHERE field NOT BETWEEN ….
- SELECT * FROM table WHERE field NOT IN (SELECT id from ….)

String

- SELECT * FROM table WHERE field != “Yes”;
- ~~SELECT * FROM table WHERE field NOT = “YES”;~~

Numeric

```
SELECT * FROM table WHERE field <> 1;
SELECT * FROM table WHERE field != 1;
```

NULL

- Sometimes you want to query empty columns (or NOT empty columns)
- Select * from table where field IS NULL;
- Select * from table where field IS NOT NULL;

Null

- How can you tell if a field can be null?

Functions

Not a full list

- MYSQL has built in functions you can use to help with
  - Display
    - Left side of the FROM
  - Selection
    - In the where clause

Math!

- RAND()
- ABS() - absolute value
- ROUND() – Round a number
- TRUNCATE(x,y) – returns x , rounding after y decimal places.
  - Useful for $12.32

Strings

- LEFT(Michael', 5);
  - Micha
- SELECT LENGTH(Raj’);
  - 3
- SELECT TRIM (“ blah   “);
  - “blah”

Strings

- Upper()
- Lower()
- WATCH OUT
  - SELECT * from table where UPPER(FIELD1) = LOWER(FIELD1)
  - This will return all rows in the table.

Binary Casting

- SELECT 'Michael' = 'michael';
  - TRUE
- SELECT BINARY 'Michael' = 'michael';
  - False

CAST

- Cast converts 1 data type into another.
- Sometimes by force.
- Mysql will help you, others will not always
  - SELECT 1+'1';
    - 2
- Varchar field that contains a number
- Status varchar(25);
- However status always contains 1 or 0
- Select * from events where cast(status as int) = 1;

Dates

- Date
  - 2019-01-03
  - YYYY-MM-DD
- Datetime
  - 2020-01-30 01:37:52
  - YYYY-MM-DD HH:MM:SS
- Time
  - HH:MM:SS
- Ts (int)
  - 1580348326
  - Sometimes called a unix time stamp

Timestamps

- FROM_UNIXTIME();
- UNIX_TIMESTAMP();

Date Parts

- MONTH()
- DAY()
- YEAR()

Date Math

- DATEDIFF(date1,date2)
  - Will subtract the 2 dates
- SELECT DATE_ADD('2020-05-01',INTERVAL 5 DAY);

Format

- DATE_FORMAT(date,"format");

Specifier	Description
%a	Abbreviated weekday name (Sun..Sat)
%b	Abbreviated month name (Jan..Dec)
%c	Month, numeric (0..12)
%D	Day of the month with English suffix (0th, 1st, 2nd, 3rd, ...)
%d	Day of the month, numeric (00..31)
%e	Day of the month, numeric (0..31)
%f	Microseconds (000000..999999)
%H	Hour (00..23)
%h	Hour (01..12)
%I	Hour (01..12)
%i	Minutes, numeric (00..59)
%j	Day of year (001..366)
%k	Hour (0..23)
%l	Hour (1..12)
%M	Month name (January..December)
%m	Month, numeric (00..12)
%p	AM or PM
%r	Time, 12-hour (*hh:mm:ss* followed by AM or PM)
%S	Seconds (00..59)
%s	Seconds (00..59)
%T	Time, 24-hour (*hh:mm:ss*)
%U	Week (00..53), where Sunday is the first day of the week; WEEK() mode 0
%u	Week (00..53), where Monday is the first day of the week; WEEK() mode 1
%V	Week (01..53), where Sunday is the first day of the week; WEEK() mode 2; used with %X
%v	Week (01..53), where Monday is the first day of the week; WEEK() mode 3; used with %x
%W	Weekday name (Sunday..Saturday)
%w	Day of the week (0=Sunday..6=Saturday)
%X	Year for the week where Sunday is the first day of the week, numeric, four digits; used with %V
%x	Year for the week, where Monday is the first day of the week, numeric, four digits; used with %v
%Y	Year, numeric, four digits
%y	Year, numeric (two digits)

Format

- Select GET_FORMAT(DATE,'USA')
- SELECT DATE_FORMAT(date, GET_FORMAT(DATE,'USA'));

Sleep

Can be used for debug.

- Sleep(<seconds>);
  - Don't use this in production;
- Select now(); select sleep(2); select now();

STR_TO_DATE( )

- Converts a string to a datetime

count(1)

- select count(1) from TABLE where X=Y; (That 1 does not change)
- This is an aggregation function. We will cover this in more detail in 2 weeks.
- You can still write a normal query or you use this.

If logic

- SELECT IF(status=1,'on','off') from events where name = 1;

- Interactive Review

Python

For your work this week, we ask that you query mysql using python.

Almost all queries from python to sql follow the same format.

Python

DO NOT COPY CODE FROM POWER POINT.

https://gist.github.com/mlhess/be7841fda8baba5645f6aad807b50083

- import pymysql
- con = pymysql.connect(host=X, port=X,user="X",password=X,database=X,cursorclass=pymysql.cursors.DictCursor) #replace X with your information from canvas. Please note, some fields will need quotes.
- cur = con.cursor()
- cur.execute("SELECT QUERY HERE")
- for row in cur:
- ## do something with data here.

Next Week

- Joins across tables

Information
changes
everything.

THANK YOU

SCHOOL OF
INFORMATION
UNIVERSITY OF MICHIGAN