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.

Plan to drop? please do it by 10pm

Taxdata table

Homework

- If we have not taught how to do something in SQL, please don't skip ahead.

Welcome to Mariah Jacobs

- She is the class GSI.
- She is in slack
- Feel free to direct questions to her or myself.

Slack

- Everyone in this room should be in slack
- Everyone should be in #database_team
- This slack is setup like a cooperate slack

Office Hours

- Office Hours are on Tuesday and Thursday mornings.
- The link to sign up is pined in slack.
- You MUST sign up for office hours before coming.

Class Pacing

Query databases

- There are 4 different basic types of SQL queries
- (There is another entire subset used to create tables, more on that later)

SELECT

- SELECT queries get data from the database for display.
- Think of select queries like a question.
- You want information from the database, so you ask a question.

Stopping a query

- Pressing control-c can stop a query from running in most cases.
- If you are using Jupyter, you can stop it by stopping your server from the control panel if control-c does not work.

Stopping a query.

Most be on your RW account.

- Show processlist;
- Kill ID;

INSERT

- Insert queries put data into the database.

UPDATE

- Update queries modify/change data in the database

DELETE

- Delete queries remove data from the database.

Queries/SELECT

- When we query a database we issue a formatted "sentence" to the database server. The database parses the sentence and preforms the action.
- If the database is not able to parse the sentence we get an error.

CRUD

- Create
- Read
- Update
- Delete

Basic select

- SELECT NOW();
- Not all queries select data from a table, but most do
- SELECT 2+2;
- SELECT RAND();
- SELECT CONCAT('a','b');
- SELECT (2+5*(3/5*3) +1)/6;

- These are examples of functions you can run within mysql. They do not require a table.

SELECT

- Normally takes the form of “select * from <table>”;
- This instructs the database server to query all of the data in the system and return all of the data for all of the fields.

Structure

- _VERB_ _ WHAT_ “FROM” _LOCATION_ _JOINS_ _WHERE_ _GROUP_BY _ORDER BY_
- Select field1 FROM table .....

Break a query down

- SELECT field1,field2,field3 from table1;
- Rather than return ALL fields, just return the fields we want to see data for.
- This is more efficient.

Field aliases

- You might want to rename a field in the query results
- SELECT daystp as daily_steps, dg as daily_goal from fitbit_data;
- daystp is hard for folks to understand, so we want to rename it so that it is more clear.

Archaeologist

- When given a new database, we often have to play archaeologists. We don't always have a clear understanding of the data in it.
- Often times the information behind how a database was created is not present. You will need to use your best guess at times to create the information using the info in the table.

Entity Relationship Diagram

- These are "pictures" that show the tables (and their relationships) in a database.
- We will discuss some format of these later.

Entity Relationship Diagram

- For this class, you MUST create your all your documentation by hand. Please DO NOT use automated software for this. This would be cheating.

ERD diagrams

- If you don't want to "draw" on paper, you can use word, excel, google docs, google sheets, paint, etc.
- Your diagram's don't have to be fancy.

To start:

- When you run desc <table> it is useful to document what you find.
- Create a list of all the fields, the type of data in each field and what you think you might find in there.
- You may not know what type of data is in there so you will have to guess to start.

A very detailed example

Physical Element Name – the field name

Logical Element Name – the human name

Format type – the field type

Format length – the max length of the field

Element Definition – What you can find here

Examples of Valid Values – data that is from the field as an example

Physical Element Name	Logical Element Name	Format Type	Format Length	Element Definition	Examples of Valid Values
PLCMNT_TEST_ID_B_DESCRSHORT	Placement Test Identification B Short Description	VARCHAR	10	An abbreviated textual description of the identification for the 'B' test. Used for placement purposes.	Examples of valid values: UM Plcmnt
PLN_SPLN_RQD_NBR	Plan Sub Plan Required Number	INT	38	The number of plans or sub-plans that must be declared by the student in the field.	Examples of valid values: Functionality not currently used.
POSTSECONDARY	Post-Secondary Code	VARCHAR	1	A code indicating that the extracurricular activity occurred in the academic year identified by the field name. There are similar fields for ninth, tenth, eleventh and twelfth grades.	Examples of valid values: Y = Yes; N = No
PREV_ALT_IDENT	Previous Alternate Identifier	VARCHAR	9	The Student Identification Number Office Assigned (SINOA) associated with an EMPLID. If the Curr_Alt_Ident associated with an EMPLID is Social Security Number (that is if an SSN exists), the Prev_Alt_Ident can contain a SINOA. If the Curr_Alt_Ident contains a SINOA, the Prev_Alt_Ident will be blank.	Examples of valid values: 123456789
PRIMARY_FIRST_NAME	Primary First Name	VARCHAR	30	The first name part of a person's name which is generally used for legal purposes.	Examples of valid values: Barbara; Joseph; William
PRIMARY_LAST_NAME	Primary Last Name	VARCHAR	30	The last or family name part of a person's name which is generally used for legal purposes.	Examples of valid values: de la Garza; Smith Jr; Washington
PRIMARY_MIDDLE_NAME	Primary Middle Name	VARCHAR	30	The middle name part of a person's name which is generally used for legal purposes.	Examples of valid values: Marie; L; Boyd
PRIMARY_NAME	Primary Name	VARCHAR	50	The name by which a person is known. This is a person's primary name which is generally used for legal purposes. Person name format is Last, First Middle	Examples of valid values: de la Garza, Barbara Marie; Smith Jr, Joseph L; Washington, William Boyd

Documentation

- Everything you do should have documentation behind it.
- This will help you now and in the future.
- It will also help ANYONE else who is working with your work.
- You can be an awesome programmer | dba | tech | etc, but unless you can document your work, others are unable to use it.

Sensitive Data

- Some tables might contain sensitive data. If you see sensitive data in a field, you should flag it. You might be asked to provide copies of a table without it. It is much easier to flag it now, then to figure it out later or under deadline.

What is sensitive data?

- Your org will determine this.
- In general:
  - PII (personal identifiable information)
  - Company sensitive data
- Why is this your problem?
  - Dev/stage/prod
  - Testing
  - DevOps

Query data

desc taxdata;

```
mysql> desc taxdata;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| id       | int(11)     | NO   | PRI | NULL    | auto_increment |
| ein      | int(11)     | YES  |     | NULL    |                |
| name     | varchar(255)| YES  |     | NULL    |                |
| year     | int(11)     | YES  |     | NULL    |                |
| revenue  | bigint(20)  | YES  |     | NULL    |                |
| expenses | bigint(20)  | YES  |     | NULL    |                |
| purpose  | text        | YES  |     | NULL    |                |
| ptid     | varchar(255)| YES  |     | NULL    |                |
| ptname   | varchar(255)| YES  |     | NULL    |                |
| city     | varchar(255)| YES  |     | NULL    |                |
| state    | varchar(255)| YES  |     | NULL    |                |
| url      | varchar(255)| YES  |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
```

Always start with documentation

- If you are not given documentation, create your own.

SQL modifiers and functions

- While there many of these, in your toolkit should be DISTINCT

DISTINCT

- SELECT DISTINCT <fieldname> from table;
- Will only return unique entries of fieldname.

DISTINCT

- This can be useful in tables that have multiple rows for the same information.

LIMIT

- Select field1, field2 from table limit 1;
  - Returns only 1 row
- Select field1, field2 from table LIMIT 0,50;
  - This would return the first 50 rows.
- Select field1, field2 from table LIMIT 50,100;
  - This returns rows 50-100;

- Limit is the last part of the query. Why?

ORDER BY

- ORDER BY will sort you query by a value. Normally the value of a field.
- ORDER by is almost always the last part of a select query.
- Select field from table ORDER BY field (desc, asc)
- ASC = ascending order
- Desc = descending order
- SQL functions that we use in the first part of the select statement can also be used here.
  - select field from table order by rand();

WHERE

- You can filter select statements using the "WHERE" syntax. We are going to learn about this next week.
- We will spend all of next week on WHERE queries.
- This week we are going to focus on finding and understanding databases.
- Please do not use WHERE on your homework this week.

Diagram

- SELECT fielda, fieldb as B, FROM table ORDER BY field DESC limit 5;
- _verb____what_____FROM___ORDER BY____LIMIT;

PRIMARY KEY

- The primary key is a field that can uniquely define a row.

Primary Key

- Assume you have a table of students.
- Your table looks like
  - Fullname
  - Year
  - Gender
  - Dob

Full name is not a primary key? Why?

Primary Key

- A Primary key is almost always a number or integer. It does not have to be, but best/standard practice is to make it one.

Primary Keys

- Exposed to you here:
- Class ID

Class	Section	Days & Times	Room	Instructor	Meeting Dates	Units	Instruction Mode	Enrl Restrict	Seats Reserved For	Avail Seats	W L T
34179	001-LEC Regular	Mo 4:00PM - 5:30PM	2245 NQ	Shannon Weber, Michael Hess	01/13/2020 - 04/21/2020	1.5	In Person	Y	N/A	0	
Class	Section	Days & Times	Room	Instructor	Meeting Dates	Units	Instruction Mode	Enrl Restrict	Seats Reserved For	Avail Seats	W L T
36574	002-LEC Regular	Mo 6:30PM - 8:00PM	1255 NQ	Michael Hess, Shannon Weber	01/13/2020 - 04/21/2020	1.5	In Person	Y	N/A	0	

Primary Key

- Some data has a built in primary key. For example, SSN used to be a commonly used primary key.
- Some datasets we have to add our own primary key to. For example, you all have student id numbers. That is the primary key we use to identify you.

Primary Key

* This is true 99.9% of the time.

- On your diagram make your primary key bold.
- Remember you can only have one primary key per table*

Lets look at a table together

- Background:
- The smarthouse database is the data behind a smarthome. This database has been modified for this class, but comes from software called home assistant. (https://www.home-assistant.io/)
- We will look at this again next week.
- Rather than attaching this to a smart house, I have used it to pull data around NQ.

Start to think about

- How you would store this type of information.
- We won't really talk about it for a few weeks, but start thinking about this now.
- With the tables we have, would you change the layout of them.

Ro_query

- The RO query database 2 tables in it. Lets look at the home_value_by_zip table.

home_value_by_zip

```
+---------+-------------+------+-----+---------+----------------+
| Field   | Type        | Null | Key | Default | Extra          |
+---------+-------------+------+-----+---------+----------------+
| id      | int(11)     | NO   | PRI | NULL    | auto_increment |
| regionid | int(11)    | YES  |     | NULL    |                |
| city    | varchar(255) | YES |     | NULL    |                |
| state   | char(2)     | YES  |     | NULL    |                |
| metro   | varchar(255) | YES |     | NULL    |                |
| county  | varchar(255) | YES |     | NULL    |                |
| ym      | varchar(20) | YES  |     | NULL    |                |
| ym_val  | int(11)     | YES  |     | NULL    |                |
+---------+-------------+------+-----+---------+----------------+
8 rows in set (0.03 sec)
```

- When looking at the desc table, we can guess some information.
- City, state, county make sense.
  What about regionid, metro and ym and ym_val?

#?#

- It is ok to have questions, if you can’t figure something out.
- That is part of data science work. However, always try your best to find the data.

Home_value_by_zip

- What is missing from that table?
- Should you bring that up, in an email at some point if asked to deal with that table but not asked that question?

Tips for documenting a table

- This week we have focused on basic select queries and table documentation. Next week we are going to pick things up a tad.
- However, as I discussed before, having a strong basic idea of what we are doing is very helpful.

Tips for documenting a table

- SELECT DISTINCT fieldname FROM table;
  - This is helpful to see unique rows.

Tips for documenting a table

- While DESC tablename provides a basic overview, understanding the real information inside the table is helpful.
- Status could be a 0/1, could be on/off, could be a random number or string. Just because the DESC table gives you the type, you want to know what the data is so you can figure out how to query it.

Tips for documenting a table

- ORDER BY field;
- ORDER BY field desc;
- ORDER BY field asc;
- ORDER BY RAND();
- ORDER BY field1 desc, field2 asc

Tips for documenting a table

- LIMIT 5
- LIMIT 5,10
- Can be used with order by
- ORDER BY RAND() limit 5;
- The last part of a select statement;

Upper and Lower

- You can select a field and change the case
- SELECT UPPER(field1) FROM table;
- Will convert the text in the field to uppercase.
- SELECT LOWER(field2) FROM table;
- Converts the text to lower case.
- DO NOT USE in where clause.

Date Functions

- For a field that is a date time field:
- MONTH()
- YEAR()
- DAY()

Information
changes
everything.

THANK YOU

SCHOOL OF
INFORMATION
UNIVERSITY OF MICHIGAN