

Dear Raj,

Hope this mail finds you well. It is my pleasure to work with you and I am willing to help you gain a basic understanding of the current database Borromean has in place. And here are my answers with your confusions about the database:

Part 1

1. *From the reading what are the 3 types of relationships the author defines.*
one-to-one, one-to-many, and many-to-many.
2. *Can you define a few different entities that could be related. For example, Class<->professor.*
Community<->Building Building<->Room Room<->Resident;
University<->School School<->Department Department<->Division Division<->Faculty.
3. *The author defines 2 different types of databases, in what use case at might you use each type?*

Hierarchical Database:

The university is a kind of hierarchical system. A university have several schools. Each School have several departments. Under each department there are different divisions. And under each division there are corresponding employees.

Relational Database:

In warehouse management database system, we can use customer id as unique id in each tables(order table, payment table, address table) to connect them. And this is a typical one-to-many relationship between customer and other entities since a customer can make several payments, order several times and have multiple addresses.

Part 2

1. *What databases are you able to see?*

There are 14 databases, they are:

bikes, information_schema, kubedb_system, mysql, performance_schema, ro_company1, ro_employees , ro_query, ro_recipes, ro_research1, ro_twitter, sakila, sys, world.

```

mysql> show databases;
+-----+
| Database |
+-----+
| bikes
| information_schema
| kubedb_system
| mysql
| performance_schema
| ro_company1
| ro_employees
| ro_query
| ro_recipes
| ro_research1
| ro_twitter
| sakila
| sys
| world
+-----+
14 rows in set (0.06 sec)

```

2. If you look at the ro_query database, how many tables are in this database? Sales indicates there should be 4. I think there are 5? Help us decide who is correct.

There are two tables(home_value_by_zip, taxdata) in the ro_query database. None of you is correct.

```

mysql> use ro_query;
No connection. Trying to reconnect...
Connection id: 226
Current database: *** NONE ***

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_ro_query |
+-----+
| home_value_by_zip |
| taxdata
+-----+
2 rows in set (0.06 sec)

```

3. What columns are there in the tax table?

There are 12 columns in the tax table, they are: id, ein, name, year, revenue, expenses, purpose, ptid, ptname, city, state, url.

```
mysql> desc taxdata;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int   | NO   | PRI  | NULL   | auto_increment |
| ein   | int   | YES  |      | NULL   |               |
| name  | varchar(255) | YES  |      | NULL   |               |
| year  | int   | YES  |      | NULL   |               |
| revenue | bigint | YES  |      | NULL   |               |
| expenses | bigint | 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   |               |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.20 sec)
```

4. Inside the ro_employees database, how many rows are in the titles table.

There are 443308 rows in the titles table.

```
mysql> use ro_employees;
No connection. Trying to reconnect...
Connection id: 376
Current database: *** NONE ***

Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> desc titles;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_no | int   | NO   | PRI  | NULL   |               |
| title  | varchar(50) | NO   | PRI  | NULL   |               |
| from_date | date  | NO   | PRI  | NULL   |               |
| to_date | date  | YES  |      | NULL   |               |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.06 sec)

mysql> select count(*) from titles;
+-----+
| count(*) |
+-----+
| 443308 |
+-----+
1 row in set (0.11 sec)
```

Thank you!

Best,

Jun

Dear Kelly,

Hope this mail finds you well! It is my pleasure to be a part contributing to the acquisition and the following are my answers with your questions:

Part One

1. *The cost of the company we are thinking about buying is \$30,000,000. However, the owner is on our board and will give a discount of \$15,210,000. We will be buying a share of this company with 6 other investors. The board member is willing to give all of us the discount on the purchase price. What is the price that each investor will pay?*

```
mysql> select (30000000-15210000)/7;
+-----+
| (30000000-15210000)/7 |
+-----+
|          2112857.1429 |
+-----+
1 row in set (0.04 sec)
```

Each investor supposed to pay \$2112857.1429.

Part Two

1. *How many accounts does company1 have? (Optional)*

There are 24 accounts.

```
mysql> select * from account;
+-----+-----+-----+-----+-----+-----+-----+
| account_id | product_cd | cust_id | open_date | close_date | last_activity_date | status |
| open_branch_id | open_emp_id | avail_balance | pending_balance |          |
+-----+-----+-----+-----+-----+-----+-----+
24 rows in set (0.14 sec)
```

2. *What is the primary key of the accounts table?*

'account_id' is the primary key.

```
mysql> desc account;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| account_id | int unsigned | NO | PRI | NULL | auto_increment |
| product_cd | varchar(10) | NO | MUL | NULL | |
| cust_id | int unsigned | NO | MUL | NULL | |
| open_date | date | NO | | NULL | |
| close_date | date | YES | | NULL | |
| last_activity_date | date | YES | | NULL | |
| status | enum('ACTIVE','CLOSED','FROZEN') | YES | | NULL | |
| open_branch_id | smallint unsigned | YES | MUL | NULL | |
| open_emp_id | smallint unsigned | YES | MUL | NULL | |
| avail_balance | float(10,2) | YES | | NULL | |
| pending_balance | float(10,2) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.04 sec)
```

3. What are the possible values that could be in the status field in the account table?

'ACTIVE','CLOSED','FROZEN' are possible values that could be in the status field in the account table.

```
mysql> desc account;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| account_id | int unsigned | NO | PRI | NULL | auto_increment |
| product_cd | varchar(10) | NO | MUL | NULL | |
| cust_id | int unsigned | NO | MUL | NULL | |
| open_date | date | NO | | NULL | |
| close_date | date | YES | | NULL | |
| last_activity_date | date | YES | | NULL | |
| status | enum('ACTIVE','CLOSED','FROZEN') | YES | | NULL | |
| open_branch_id | smallint unsigned | YES | MUL | NULL | |
| open_emp_id | smallint unsigned | YES | MUL | NULL | |
| avail_balance | float(10,2) | YES | | NULL | |
| pending_balance | float(10,2) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.04 sec)
```

However, in current table, there is only 'ACTIVE' in it.

```
mysql> select distinct status from account;
+-----+
| status |
+-----+
| ACTIVE |
+-----+
1 row in set (0.05 sec)
```

Table Documentation

Account

Field Name	Data Type	Values	Note
account_id	int	id of account	primary key
product_id	int	id of product	
open_date	date	account open date	
close_date	date	account close date	
last_activity_date	date	account last active date	
status	enum('ACTIVE'CLOSED'FROZEN')	three kinds of account status	
open_branch_id	int	id of branch where account opened	
open_emp_id	int	id of employee who opened the account	
avail_balance	float	account available balance	
pending_balance	float	account pending balance	

Part Three

1. Which employee ID has opened the most company accounts? (look at the account table for this one)

The employee with id 1 is the person who opened the most company accounts since in the open_emp_id 1 has the most appearance.

account_id	product_cd	cust_id	open_date	close_date	last_activity_date	status	open_branch_id	open_emp_id	available_balance	pending_balance
1057.75	1	CHK	1057.75	1	2000-01-15	NULL	2005-01-04	ACTIVE	2	10
500.00	2	SAV	500.00	1	2000-01-15	NULL	2004-12-19	ACTIVE	2	10
3000.00	3	CD	3000.00	1	2004-06-30	NULL	2004-06-30	ACTIVE	2	10
2258.02	4	CHK	2258.02	2	2001-03-12	NULL	2004-12-27	ACTIVE	2	10
200.00	5	SAV	200.00	2	2001-03-12	NULL	2004-12-11	ACTIVE	2	10
1057.75	7	CHK	1057.75	3	2002-11-23	NULL	2004-11-30	ACTIVE	3	13
2212.50	8	MM	2212.50	3	2002-12-15	NULL	2004-12-05	ACTIVE	3	13
534.12	10	CHK	534.12	4	2003-09-12	NULL	2005-01-03	ACTIVE	1	1
767.77	11	SAV	767.77	4	2000-01-15	NULL	2004-10-24	ACTIVE	1	1
5487.09	12	MM	5487.09	4	2004-09-30	NULL	2004-11-11	ACTIVE	1	1
2237.97	13	CHK	2897.97	5	2004-01-27	NULL	2005-01-05	ACTIVE	4	16
122.37	14	CHK	122.37	6	2002-08-24	NULL	2004-11-29	ACTIVE	1	1
10000.00	15	CD	10000.00	6	2004-12-28	NULL	2004-12-28	ACTIVE	1	1
5000.00	17	CD	5000.00	7	2004-01-12	NULL	2004-01-12	ACTIVE	2	10
3487.19	18	CHK	3487.19	8	2001-05-23	NULL	2005-01-03	ACTIVE	4	16
387.99	19	SAV	387.99	8	2001-05-23	NULL	2004-10-12	ACTIVE	4	16
125.67	21	CHK	125.67	9	2003-07-30	NULL	2004-12-15	ACTIVE	1	1
9345.55	22	MM	9845.55	9	2004-10-28	NULL	2004-10-28	ACTIVE	1	1
1500.00	23	CD	1500.00	9	2004-06-30	NULL	2004-06-30	ACTIVE	1	1
23575.12	24	CHK	23575.12	10	2002-09-30	NULL	2004-12-15	ACTIVE	4	16
0.00	25	BUS	0.00	10	2002-10-01	NULL	2004-08-28	ACTIVE	4	16
9345.55	27	BUS	9345.55	11	2004-03-22	NULL	2004-11-14	ACTIVE	2	10
38552.05	28	CHK	38552.05	12	2003-07-30	NULL	2004-12-15	ACTIVE	4	16
50000.00	29	SBL	50000.00	13	2004-02-22	NULL	2004-12-17	ACTIVE	3	13

2. When did the employee who has opened the most accounts start working at the company? (optional)

The employee1 started to work for the company since June 22nd, 2001.

```
mysql> select emp_id, start_date from employee where emp_id = 1;
+-----+-----+
| emp_id | start_date |
+-----+-----+
|      1 | 2001-06-22 |
+-----+-----+
1 row in set (0.15 sec)
```

Part Four

1. Write a query that returns only the date of the transaction and the account id (in that order) from the transaction table sort this by the most recent transaction.

The following is my query:

```

mysql> select txn_date, account_id from transaction order by txn_date desc;
+-----+-----+
| txn_date        | account_id |
+-----+-----+
| 2004-12-28 00:00:00 |      15 |
| 2004-10-28 00:00:00 |      22 |
| 2004-09-30 00:00:00 |      12 |
| 2004-06-30 00:00:00 |      23 |
| 2004-06-30 00:00:00 |       3 |
| 2004-01-27 00:00:00 |      13 |
| 2004-01-12 00:00:00 |      17 |
| 2003-09-12 00:00:00 |      10 |
| 2003-07-30 00:00:00 |      28 |
| 2003-07-30 00:00:00 |      21 |
| 2002-12-15 00:00:00 |       8 |
| 2002-11-23 00:00:00 |       7 |
| 2002-09-30 00:00:00 |      24 |
| 2002-08-24 00:00:00 |      14 |
| 2001-05-23 00:00:00 |      18 |
| 2001-05-23 00:00:00 |      19 |
| 2001-03-12 00:00:00 |       5 |
| 2001-03-12 00:00:00 |       4 |
| 2000-01-15 00:00:00 |      11 |
| 2000-01-15 00:00:00 |       2 |
| 2000-01-15 00:00:00 |       1 |
+-----+-----+
21 rows in set (0.05 sec)

```

Table Documentation

Transaction

Field Name	Data Type	Values	Note
txn_id	int	id of transaction	primary key
txn_date	datetime	transaction date and specific time	
account_id	int	id of account	
txn_type_cd	enum('DBT','CDT')	transaction via debit card or credit card	
teller_emp_id	int	id of teller employee	
execution_branch_id	int	id of branch where the transaction executed	
funds_avial_date	datetime	time when the fund available	

2. *What is the primary key in the transaction table?*

'txn_id' is the primary key.

```
mysql> desc transaction;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| txn_id | int unsigned | NO | PRI | NULL | auto_increment |
| txn_date | datetime | NO | | NULL | |
| account_id | int unsigned | NO | MUL | NULL | |
| txn_type_cd | enum('DBT','CDT') | YES | | NULL | |
| amount | double(10,2) | NO | | NULL | |
| teller_emp_id | smallint unsigned | YES | MUL | NULL | |
| execution_branch_id | smallint unsigned | YES | MUL | NULL | |
| funds_avail_date | datetime | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.05 sec)
```

3. What are the unique product types that are offered? (optional)

'ACCOUNT', 'INSURANCE', 'LOAN' are the unique product types.('product_type_cd' is the primary key.)

```
mysql> desc product_type;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| product_type_cd | varchar(10) | NO | PRI | NULL | |
| name | varchar(50) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.04 sec)

mysql> select * from product_type;
+-----+-----+
| product_type_cd | name |
+-----+-----+
| ACCOUNT | Customer Accounts |
| INSURANCE | Insurance Offerings |
| LOAN | Individual and Business Loans |
+-----+-----+
3 rows in set (0.11 sec)

mysql> select product_type_cd from product_type;
+-----+
| product_type_cd |
+-----+
| ACCOUNT |
| INSURANCE |
| LOAN |
+-----+
3 rows in set (0.04 sec)
```

Part Five

1. How many branches are located in MA? (optional)

There are 3 branches located in MA.

```

mysql> select state from branch;
+-----+
| state |
+-----+
| MA   |
| MA   |
| MA   |
| NH   |
+-----+
4 rows in set (0.04 sec)

```

2. Can you write a query will list the name of each branch, the address of each branch, the zip code of each branch, and a field that indicates the current date and time you made the query that shows in the query as a field called "querytime"? I want to make sure that when I view the data, I am seeing when (date, time) you made the query to validate if you are correct.

The following is my query:

```

mysql> select name, address, zip , now() as querytime from branch;
+-----+-----+-----+-----+
| name      | address          | zip    | querytime        |
+-----+-----+-----+-----+
| Headquarters | 3882 Main St. | 02451 | 2023-09-16 00:03:11 |
| Woburn Branch | 422 Maple St. | 01801 | 2023-09-16 00:03:11 |
| Quincy Branch | 125 Presidential Way | 02169 | 2023-09-16 00:03:11 |
| So. NH Branch | 378 Maynard Ln. | 03079 | 2023-09-16 00:03:11 |
+-----+-----+-----+-----+
4 rows in set (0.04 sec)

```

Table Documentation

Branch

Field Name	Data Type	Values	Note
branch_id	int	id of branch	primary key
name	varchar	name of branch	
address	varchar	address of branch	
city	varchar	city where branch located	
state	varchar	state where branch in	
zip	varchar	zipcode of branch	

Best regards,

Jun

Dear Raj,

Hope you had a lovely weekend! It is my pleasure to help with decision making in our company's non-profit investment. Here are my answers for your questions:

Part 1 - Revenue

1. *2014 was a key year for generating revenue. How many Ann Arbor-based companies (rows) are listed in the database that year?*

There are 247 Ann Arbor-based companies are listed in the database in 2014.

```
Database changed
mysql> select count(1) from taxdata where year = 2014 and city = 'Ann Arbor';
+-----+
| count(1) |
+-----+
|      247 |
+-----+
1 row in set (4.65 sec)
```

taxdata

Field Name	Data Type	Values	Note
id	int	id of the company	primary key
ein	int	ein assigned to the company	
name	varchar	name of the company	
year	int	taxyear	
revenue	int	revenue of the company	
expenses	int	expense of the company	
purpose	text	purpose of the company	
ptid	varchar	ptid of the company	
ptname	varchar	ptname of the company	
city	varchar	city where company located	
state	varchar	state where company in	
url	varchar	url of company	

2. To get a broader picture, we also need a list of the names of ALL of the companies that generated more than \$10,000,000,000 (ten billion dollars) in 2014.

```
mysql> select name from taxdata where year = 2014 and revenue>10000000000;
+-----+
| name |
+-----+
| IHC HEALTH SERVICES INC
| CENTRAL STATES SE & SW AREAS HEALTH & WELFARE F
| Banner Health
| KAISER FOUNDATION HEALTH PLAN INC
| Thrivent Financial for Lutherans
| UAW RETIREE MEDICAL BENEFITS TRUST
| KAISER FOUNDATION HOSPITALS
| Massachusetts Institute of Technology
| UPMC GROUP
| Cornell University
| Trustees of the University of Pennsylvania
| President and Fellows of Harvard College
| YALE UNIVERSITY
| STATE EMPLOYEES' CREDIT UNION
| DUKE UNIVERSITY
| TRUSTEES OF THE UNIVERSITY OF PENNSYLVANIA
| Howard Hughes Medical Institute
| Partners HealthCare System Inc & AffiliatesGroup Return
| THE BOARD OF TRUSTEES OF THE LEAND STANFORDJUNIOR UNIVERSITY
| DIGNITY HEALTH
+-----+
20 rows in set (2.18 sec)
```

3. How does this compare to how many unique companies generated a revenue of more than \$1,000,000,000 (one billion dollars) in any year?

There are 404 unique companies made revenue more than 1,000,000,000 in any year.

```
mysql> select count(distinct name)    from taxdata where revenue>10000000000;
+-----+
| count(distinct name) |
+-----+
|          404         |
+-----+
1 row in set (0.43 sec)
```

Part 2 - Expenses

1. We are looking for the top 20 unique companies by expenses (as in: who had the most expenses?). Your query should provide the name of the company and their expenses in 2013.

```

mysql> select distinct name, expenses
-> from(select name, expenses , year from taxdata where year = 2013 order by expenses desc limit 20) as subquery;
+-----+-----+
| name | expenses |
+-----+-----+
| KAISER FOUNDATION HEALTH PLAN INC | 4198228055 |
| UNIVERSITY OF ROCHESTER | 2776536619 |
| ST FRANCIS HOSPITAL AND MEDICAL CENTER | 713077460 |
| WELLSTAR HEALTH SYSTEM INC | 660400156 |
| SKY LAKES MEDICAL CENTER | 488333644 |
| NAPLES COMMUNITY HOSPITAL INC | 426168844 |
| YAKIMA VALLEY MEMORIAL HOSPITAL | 401985865 |
| EMBRY-RIDDLE AERONAUTICAL UNIVERSITY INC | 374154000 |
| ST MARY'S HOSPITAL & MEDICAL CENTERINC | 372041256 |
| St Barnabas Hospital | 349321637 |
| HOSPITAL COMMITTEE FOR THE LIVERMOREPLEASANTON AREA | 341257233 |
| VALLEY PRESBYTERIAN HOSPITAL | 273789797 |
| WINCHESTER HOSPITAL | 264605951 |
| TEAMSTERS WESTERN REGION AND NEW JERSEYHEALTH CARE FUND | 231059337 |
| OROVILLE HOSPITAL | 200242937 |
| ST JOSEPH HOSPITAL OF EUREKA | 196999313 |
| SWARTHMORE COLLEGE | 168742729 |
| ST FRANCIS HOSPITAL - POUGHKEEPSIE | 164506124 |
| Vail Clinic Inc | 159617575 |
| TUSKEGEE UNIVERSITY | 152572824 |
+-----+-----+
20 rows in set (0.58 sec)

```

2. We also need to know the EINs and cities for each company that made between \$1-100,000 in revenue and between \$10000-200,000 in expenses. (If this list is long, just give us a random selection of 20 and your query)

```

mysql> select ein,city from(select ein,city,revenue, expenses from taxdata where 1<revenue<10000 and 10000 < expenses< 200000) as subquery limit 20;
+-----+-----+
| ein | city |
+-----+-----+
| 742661023 | SAN ANTONIO |
| 562629114 | BROOKLYN |
| 270678774 | GLENDOORA |
| 464114252 | NEWPORT |
| 510311790 | DOVER |
| 261460932 | WISE |
| 270609504 | DETROIT |
| 205710892 | ELK GROVE |
| 251910030 | SAN FRANCISCO |
| 521548962 | ROCKVILLE |
| 731653383 | FERNDALE |
| 10706544 | Ocala |
| 274563334 | Tucson |
| 262511957 | ADA |
| 272260017 | Cincinnati |
| 271198702 | LEESBURG |
| 841650210 | AUSTIN |
| 760294275 | WASHINGTON |
| 541250706 | WARRENTON |
| 270526903 | North Fort Myers |
+-----+-----+
20 rows in set (0.04 sec)

```

Part 3 - Specific Words

1. First, we are interested in the number of companies with the word "toy" anywhere in the 'purpose' field.

There are 378 companies with the word 'toy' anywhere in the 'purpose' field.

```

mysql> select count(1) from taxdata where purpose like '%toy%';
+-----+
| count(1) |
+-----+
| 378 |
+-----+
1 row in set (2.39 sec)

```

2. Additionally, we need data on how many rows have both the word 'smith' in the 'ptname' field and have reported revenue (e.g. revenue is not empty or 0).

There are 2796 rows meet the above requirements.

```
mysql> select count(1) from taxdata where ptname like '%smith%' and revenue is not null
   and revenue <> 0;
+-----+
| count(1) |
+-----+
|      2796 |
+-----+
1 row in set (0.92 sec)
```

Part 4

- First, we need to know the company name and length of the name for 50 random companies with a ptid of P01345770.

The following is my query and part of the output table;

```
mysql> select name, length(name) from taxdata where ptid= 'P01345770' order by rand() limit 50;
+-----+-----+
| name           | length(name) |
+-----+-----+
| Singhal Family Charitable Foundation          |            36 |
| Xin Cheng Foundation                          |            20 |
| McLamore Family Foundation Inc                |            30 |
| Cultures of Resistance Network Foundation     |            41 |
| The EvansKabush Family Foundation             |            33 |
| The Esperance Family Foundation Inc           |            35 |
| The McLean Family Foundation                  |            28 |
| The Loftus Family Foundation                  |            28 |
| The Obrzut - Ling Foundation Inc             |            32 |
| William T Foley Foundation Inc               |            30 |
| Twomey Family Foundation                     |            24 |
| Lighten Family Foundation                    |            25 |
| The John Murray and Eleanor M StritterFoundation Inc | 52 |
| Joy and Hank Kuchta Foundation              |            30 |
| The Joseph and Lucille Madri FamilyFoundation |            45 |
| The Hartings Family Charitable Foundation    |            41 |
| Dilmaghani Dream Foundation                 |            27 |
| Turkanis Family Foundation                  |            26 |
| Oak Tree Foundation of Colorado AKA OakTree Foundation | 54 |
| The Hart Family Foundation                  |            26 |
| The Leah and Alain Lebec Foundation Inc    |            39 |
| The Cosse-Finney Foundation                 |            27 |
| Sampson Family Foundation                  |            25 |
| JM McDonald Foundation Inc                 |            26 |
| The Sturtevant Family Foundation             |            32 |
| Blue Earth Foundation Inc                  |            25 |
| Kanshi Ram Memorial Arora (KARMA) Foundation | 44 |
| Bhai Jaita Foundation Inc                  |            25 |
| Gary W Dietrich Family Foundation           |            33 |
| Mary Elizabeth Conover Foundation          |            33 |
| William H Wood Family Foundation            |            32 |
| Robert and Ardis James Foundation Inc       |            37 |
+-----+-----+
```

- Second, we need the number of companies that have a 'purpose' field containing less than 10 characters. Keep in mind that the value of the 'purpose' field should never be an empty string.

There are 917 companies meet the requirement.

```
mysql> select count(1) from taxdata where length(purpose) < 10 and purpose <> '';
+-----+
| count(1) |
+-----+
|      917 |
+-----+
1 row in set (1.01 sec)
```

Part 5 - Employee's

1. What shortest WHERE clause can you write to query the number of folks that got hired in 1994, 1995 and 1990. (You can do this using the count(1) syntax as discussed in class, or a query that returns all the records and you just count). I want the shortest WHERE query possible.

The following is my query:

```
mysql> select count(1) from employees where year(hire_date) in (1990, 1994, 1995);
+-----+
| count(1) |
+-----+
|      52560 |
+-----+
1 row in set (0.27 sec)
```

2. What query will provide a count of all 'Senior Engineer' that were at the company on 1986-06-26. (Hint, this is not a simple where date=1)

The following is my query:

```
mysql> select count(1) from titles where title = 'Senior Engineer'
    -> and from_date <= '1986-06-26'
    -> and (to_date > '1986-06-26' or to_date is null);
+-----+
| count(1) |
+-----+
|      2795 |
+-----+
1 row in set (0.79 sec)
```

employees

Field Name	Data Type	Values	Note
emp_no	int	employee number	primary key
birth_date	date	birthdate of employee	
first_name	varchar	first name of employee	
last_name	varchar	last name of employee	
gender	enum('M', 'F')	gender of employee	
hire_date	date	employee hired date	

3. Can you give me a list of unique names of folks that have had a title of "Engineer". We have not covered joins yet, so please make sure to do this without a join and in a single query.

There are 11930 names meet the requirements showed in the following picture, to give a short preview I added 'limit 10' command at the last:

```
+-----+
| SachinTukuda |
+-----+
111930 rows in set (1.59 sec)

mysql> select distinct concat(first_name, ' ', last_name) as full_name from employees where emp_no in (select distinct emp_no from titles where title = 'Engineer') limit 10;
+-----+
| full_name |
+-----+
| Chirstian Koblick
| Suman Peac
| Duangkaew Piveteau
| Patricio Bridgland
| Berni Genin
| Kazuhide Peha
| Mayuko Warwick
| Shahaf Famili
| Bojan Montemayor
| Yongqiao Berztiss |
+-----+
10 rows in set (0.04 sec)
```

Title

Field Name	Data Type	Values	Note
emp_no	int	employee number	primary key
title	varchar	title of employee	
from_date	date	employee hired from date	
to_date	date	employee left company date	

Hopefully my analytics will be helpful! Thank you!

Best regards,

Jun

Hi Sarah,

Hope this mail finds you well. It has been a long time not to see you. Here are my answers with your questions.

Part 1

I have a database of recipes, and I could really use your help in finding out some information. This is called **ro_recipes**.

1. What are the recipe titles that are either a vegetable or a salad? I want to provide this information to my friend; please give me a SINGLE query that does this without any numbers in the query.

```
mysql> select R.RecipeTitle from Recipes R join Recipe_Classes RC on R.RecipeClassID = R.C.RecipeClassID where RC.RecipeClassDescription in ('Salad', 'Vegetable');
+-----+
| RecipeTitle |
+-----+
| Garlic Green Beans |
| Asparagus |
| Mike's Summer Salad |
+-----+
3 rows in set (0.05 sec)
```

Table Documentation:

Recipes

Field Name	Data Type	Values	Note
RecipeID	int	ID for the Recipe	primary key
RecipeTitle	varchar	Title for the Recipe	
RecipeClassID	int	Class ID the recipe belongs	foreign key with Recipe_Class table
Preparation	text	Preparation Process for the Recipe	
Notes	text	Tips for the recipe	

Recipe_Classes

Field Name	Data Type	Values	Note
RecipeClassID	int	Recipe Class ID category	primary key
RecipeClassDescription	varchar	Corresponding text description with number ID	

2. I have someone that is allergic to seafood coming over; please list all the dishes that contain seafood. Once again, one query with no numbers in it, please. (optional, this could be hard depending on how you try to do it)

```
mysql> select R.RecipeTitle
    -> from Recipes R
    -> join Recipe_Ingredients RI on R.RecipeID = RI.RecipeID
    -> join Ingredient_Classes IC on RI.IngredientID = IC.IngredientClassID
    -> where IC.IngredientClassDescription = 'Seafood';
+-----+
| RecipeTitle
+-----+
| Salsa Buena
| Pollo Picoso
| Roast Beef
| Huachinango Veracruzana (Red Snapper, Veracruz style)
| Tourtière (French-Canadian Pork Pie)
| Salmon Filets in Parchment Paper
+-----+
6 rows in set (0.04 sec)
```

Table Documentation:

Recipe_Ingredients

Field Name	Data Type	Values	Note
RecipeID	int	ID for the Recipe	primary key
RecipeSeqNo	int	sequence number of recipe	
IngredientID	int	id of the ingredient	
MeasureAmountID	int	id of measurement amount	
Amount	float	amount of the recipe	

Ingredient_Classes

Field Name	Data Type	Values	Note
IngredientClassID	int	Ingredient Class ID category	primary key
IngredientClassDescription	varchar	Corresponding text description with number ID	

3. Raj and the rest of the database team are coming over next week, and they really like beef and garlic; what can I make for everyone that contains beef and garlic? (If you're not a fan of Raj, you don't have to answer this question). (Please note, this is the most challenging question on this homework and will require some time and complexity. We will be happy to discuss in a future week of office hours. Once again, this is a very very very hard question. You do not have to answer it.)

```

mysql> select R.RecipeTitle
-> from Recipes R
-> join Recipe_Ingredients RI on R.RecipeID = RI.RecipeID
-> join Ingredients I on RI.IngredientID = I.IngredientID
-> where IngredientName in ('Beef','Garlic');
+-----+
| RecipeTitle |
+-----+
| Irish Stew |
| Roast Beef |
| Garlic Green Beans |
| Pollo Picoso |
| Roast Beef |
| Asparagus |
+-----+
6 rows in set (0.05 sec)

```

Table Documentation:

Ingredients

Field Name	Data Type	Values	Note
IngredientID	int	id of the ingredient	primary key
IngredientName	varchar	name of ingredient	
IngredientClassID	int	class ID of ingredient	
MeasureAmountID	int	id of measurement amount	

Part 2

While you are helping out, I have a few more questions... I want to send cards to a few of my co-workers.

1. In the **ro_company1** database, can you give me the names of those employees who work in the Operations department? Once again, please don't put any numbers in your query.

```

mysql> select concat(e.fname,' ',e.lname) as name from employee e join department d on e.dept_id = d.dept_id where d.name = 'Operations';
+-----+
| name |
+-----+
| Susan Hawthorne
| Helen Fleming
| Chris Tucker
| Sarah Parker
| Jane Grossman
| Paula Roberts
| Thomas Ziegler
| Samantha Jameson
| John Blake
| Cindy Mason
| Frank Portman
| Theresa Markham
| Beth Fowler
| Rick Tulman
+-----+
14 rows in set (0.19 sec)

```

Table Documentation:

employee

Field Name	Data Type	Values	Note
emp_id	int	id of the employee	primary key
fname	varchar	first name	
lname	varchar	last name	
start_date	date	start date of the employment	
end_date	date	end_date of the employment	
superior_emp_id	int	employment id of superior	
dept_id	int	id of the department	
title	varchar	title in the company	
assigned_branch_id	int	id of assigned branch	

department

Field Name	Data Type	Values	Note
dept_id	int	id of the department	primary key
name	varchar	name of the department	

2. Also, employee Paula Roberts has opened a large number of accounts. Can you let me know on what dates she opened accounts on and what the address of those customers are? (one query)

```
mysql> select A.open_date AccountOpenDate, C.address As CustomerAddress
-> from employee E
-> join account A on E.emp_id = A.open_emp_id
-> join customer C on A.cust_id = C.cust_id
-> where E.fname = 'Paula' And E.lname = 'Roberts';
+-----+-----+
| AccountOpenDate | CustomerAddress |
+-----+-----+
| 2000-01-15      | 47 Mockingbird Ln |
| 2000-01-15      | 47 Mockingbird Ln |
| 2004-06-30      | 47 Mockingbird Ln |
| 2001-03-12      | 372 Clearwater Blvd |
| 2001-03-12      | 372 Clearwater Blvd |
| 2004-01-12      | 29 Admiral Ln |
| 2004-03-22      | 287A Corporate Ave |
+-----+-----+
7 rows in set (0.05 sec)
```

Table Documentation:

customer

Field Name	Data Type	Values	Note
cust_id	int	id of customer	primary key
fed_id	varchar	fed id of customer	
cust_type_id	enum('I','B')	type id of customer	
address	varchar	address of customer	
city	varchar	city where customer located	
state	varchar	state where customer located	
Postal_code	varchar	zipcode of customer address	

account

Field Name	Data Type	Values	Note
account_id	int	id of the account	Primary key
product_id	int	id of product	
cust_id	int	id of customer	
open_date	date	account open date	
close_date	date	account close date	
last_activity_date	date	last account activity date	
status	enum('ACTIVE','CLOSED','FROZEN')	account status	
open_branch_id	int	id of branch where account opened	
open_emp_id	int	id of employee who opened the account	
avail_balance	int	available balance for the account	
pending_balance	int	pending balance for the account	

3. Can you provide me a list of *all* our employees and their manager's ID?

```
mysql> select E.emp_id as EmployeeID, M.emp_id as ManagerID
    -> from employee E
    -> left join employee M on E.superior_emp_id = M.emp_id;
+-----+-----+
| EmployeeID | ManagerID |
+-----+-----+
|      1      |     NULL   |
|      2      |        1   |
|      3      |        1   |
|      4      |        3   |
|      5      |        4   |
|      6      |        4   |
|     10      |        4   |
|     13      |        4   |
|     16      |        4   |
|      7      |        6   |
|      8      |        6   |
|      9      |        6   |
|     11      |       10  |
|     12      |       10  |
|     14      |       13  |
|     15      |       13  |
|     17      |       16  |
|     18      |       16  |
+-----+-----+
18 rows in set (0.16 sec)
```

4. Can you give me a list of all accounts that have not been closed, the available balance, the state the customer is located in, and the name of the customer's business if they have one?

```

mysql> select A.avail_balance, C.state, B.name
-> from account A
-> join customer C on A.cust_id = C.cust_id
-> left join business B on C.cust_id = B.cust_id where A.status = 'ACTIVE';
+-----+-----+-----+
| avail_balance | state | name      |
+-----+-----+-----+
| 1057.75      | MA    | NULL     |
| 500.00        | MA    | NULL     |
| 3000.00       | MA    | NULL     |
| 2258.02       | MA    | NULL     |
| 200.00        | MA    | NULL     |
| 1057.75       | MA    | NULL     |
| 2212.50       | MA    | NULL     |
| 534.12        | MA    | NULL     |
| 767.77        | MA    | NULL     |
| 5487.09       | MA    | NULL     |
| 2237.97       | NH    | NULL     |
| 122.37        | MA    | NULL     |
| 10000.00      | MA    | NULL     |
| 5000.00       | MA    | NULL     |
| 3487.19       | NH    | NULL     |
| 387.99        | NH    | NULL     |
| 125.67        | MA    | NULL     |
| 9345.55       | MA    | NULL     |
| 1500.00       | MA    | NULL     |
| 23575.12      | NH    | Chilton Engineering |
| 0.00           | NH    | Chilton Engineering |
| 9345.55       | MA    | Northeast Cooling Inc. |
| 38552.05      | NH    | Superior Auto Body |
| 50000.00      | MA    | AAA Insurance Inc. |
+-----+-----+-----+
24 rows in set (0.05 sec)

```

Table Documentation:

business

Field Name	Data Type	Values	Note
cust_id	int	id of customer	primary key
name	varchar	name of business	
state_id	varchar	id of state	
incorp_date	date	date of incorporation	

5. Can you provide a list of employees that work at a branch with the address of "422 Maple St."?

```

mysql> select concat(E.fname, ' ', E.lname) as name
      -> from employee E
      -> join branch B on E.assigned_branch_id = B.branch_id
      -> where address = '422 Maple St.';
+-----+
| name |
+-----+
| Paula Roberts |
| Thomas Ziegler |
| Samantha Jameson |
+-----+
3 rows in set (0.17 sec)

```

Table Documentation:

branch

Field Name	Data Type	Values	Note
branch_id	int	id of customer	primary key
name	varchar	name of branch	
address	varchar	address of branch	
city	varchar	city where branch located	
state	varchar	state of the branch	
zip	varchar	zip code of branch address	

Part 3

My partner and I are trying to find some exciting places to visit for our upcoming vacation. Using the **world** database, can you find:

1. 5 random cities that have a population above 13,000 and below 500,000 and are not located in North America?

```

mysql> select C.name
      -> from city C
      -> join country Co on C.CountryCode = Co.code
      -> where 13000 < C.population < 500000 and Co.Continent = 'North America'
      -> order by rand() limit 5;
+-----+
| name |
+-----+
| Reno |
| George Town |
| Corona |
| Provo |
| Raleigh |
+-----+
5 rows in set (0.05 sec)

```

Table Documentation:

city

Field Name	Data Type	Values	Note
ID	int	id of the city	primary key
name	char	name of the city	
CountryCode	char	code of the country where city located	
District	char	district where the city belongs	
Population	int	population of the city	

country

Field Name	Data Type	Values	Note
Code	char	code of the country	primary key
Name	char	name of the country	
Continent	enum('Asia','Europe','North America','Africa','Oceania','Antarctica','South America')	continent the country located	
Region	char	region where country located	
SurfaceArea	decimal	surface area of the country	
IndepYear	int	Independence year	
Population	int	Population	
LifeExpectancy	decimal	life expectancy	
GNP	decimal	GNP	
GNPOld	decimal	GNPOld	
LocalName	char	local name of the country	
GovernmentForm	char	form the government in	
HeadOfState	char	president of the country	
Capital	int	capital	
Code2	char	code2 of the country	

2. How many cities are in a Constitutional Monarchy?

There are 611 cities meet the requirement.

```
mysql> select count(1) from city C join country CO on C.Countrycode = CO.code where CO.GovernmentForm like '%Constitutional Monarchy%';
+-----+
| count(1) |
+-----+
|      611 |
+-----+
```

3. 5 random cities that a population above 13000 and below 500,000 that don't speak English as an official language and are not a Republic?

```
mysql> select C.name
->   from city C
->   join country C0 on C.CountryCode = C0.code
->   join countrylanguage CL on C0.code = CL.CountryCode
->   where 13000 < C.population < 500000 and C0.GovernmentForm <> 'Republic'
->   and CL.Language <> 'English'
->   order by rand() limit 5;
+-----+
| name |
+-----+
| Gütersloh |
| Baotou |
| Odessa |
| Yuyao |
| Boise City |
+-----+
5 rows in set (0.09 sec)
```

Table Documentation:

country_language

Field Name	Data Type	Values	Note
CountryCode	char	code of the country	
Language	char	language of the country	
IsOfficial	enum('T','F')	whether language official	
Percantage	decimal	percent of language use	

Part 4

Using the **ro_employee** database:

1. From 1985-01-01 to 1986-01-01, how many people were on payroll with a title that contained the word "Engineer"? (Optional)

There are 16749 people meet the requirement.

```

mysql> select count(1) from salaries S
    -> join titles T on S.emp_no = T.emp_no
    -> where T.title like '%Engineer%'
    -> and S.from_date <= '1986-01-01' and S.to_date >= '1985-01-01';
+-----+
| count(1) |
+-----+
|     16749 |
+-----+
1 row in set (3.56 sec)

```

2. How many people work in the Production department anytime between 1985-01-01 to 1992-01-01?
(Optional)

There are 31700 people meet the requirement.

```

mysql> select count(1) from employees E
    -> join dept_emp DE on E.emp_no = DE.emp_no
    -> join departments D on DE.dept_no = D.dept_no
    -> where D.dept_name = 'Production' and
    -> DE.to_date >= '1985-01-01' and DE.from_date <= '1992-01-01';
+-----+
| count(1) |
+-----+
|     31700 |
+-----+
1 row in set (0.58 sec)

```

3. Can you give a list of the names of 20 random employees, their current salary, and their current title?
(Optional)

```

mysql> select concat(E.first_name, ' ', E.last_name) as name, S.salary, T.title  from employees E join salaries S on E.emp_no = S.emp_no  join titles T on E.emp_no = T.emp_no  order by rand() limit 20;
+-----+-----+-----+
| name          | salary | title        |
+-----+-----+-----+
| Yishai Buescher | 52362 | Senior Engineer
| Lubomir Camurati | 57483 | Staff
| Menkae Krzyzanowski | 65785 | Engineer
| Ulf Junot | 78270 | Senior Staff
| Mary Katalagarianos | 49498 | Engineer
| Martien Pokrovskii | 104537 | Senior Staff
| Boguslaw Matteis | 58670 | Senior Staff
| Kagan Busillo | 64747 | Senior Staff
| Geoffrey Koprowski | 47883 | Staff
| Adas Hofmeyr | 123536 | Senior Staff
| Jinya Tramer | 69170 | Senior Staff
| Adam Sndden | 82296 | Staff
| Tokuyasu Sevcikova | 83276 | Senior Engineer
| Fai Hiyoshi | 55463 | Engineer
| Lihong Thiria | 42940 | Senior Staff
| Marsja Eiron | 54374 | Senior Engineer
| Candido Gaughan | 46265 | Senior Engineer
| Rosli Champarnaud | 72765 | Engineer
| Utz Matzat | 41437 | Senior Staff
| Bowen Rissland | 63262 | Engineer
+-----+-----+-----+
20 rows in set (16.53 sec)

```

Can't wait to see you and talk with you! See you next week!

Best,

Jun

Dear Lawrence,

Hope this mail finds you well. It is my pleasure to make contributions with our acquisition plan and I will obey the NDA rule not to talk this with anyone. Here are my answers with your questions:

1. What is the average customer's credit limit?

```
mysql> select avg(creditLimit) from customers;
+-----+
| avg(creditLimit) |
+-----+
| 67659.016393 |
+-----+
1 row in set (0.09 sec)
```

Table Documentation

customers

Field Name	Data Type	Values	Note
customerNumber	int	number of customer	primary key
customerName	varchar	name of customer	
contactLastName	varchar	last name of contact	
contactFirstName	varchar	first name of contact	
phone	varchar	phone number of customer	
addressLine1	varchar	first line of address	
addressLine2	varchar	second line of address	
city	varchar	city of customer	
state	varchar	state where customer live	
postalCode	varchar	zipcode of customer address	
country	varchar	country of customer	
salesRepEmployeeNumber	int	number of employee sales representative	
creditLimit	decimal	credit limit of customer	

2. Please provide a list of each customer, their phone number and the payments they have made.

There are 273 rows, so I screenshot part of it.

```
mysql> select C.customerName, C.phone, P.paymentDate, P.amount from customers C join payments P on C.customerNumber = P.customerNumber;
+-----+-----+-----+-----+
| customerName | phone | paymentDate | amount |
+-----+-----+-----+-----+
| Atelier graphique | 40.32.2555 | 2004-10-19 | 6066.78 |
| Atelier graphique | 40.32.2555 | 2003-06-05 | 14571.44 |
| Atelier graphique | 40.32.2555 | 2004-12-18 | 1676.14 |
| Signal Gift Stores | 7025551838 | 2004-12-17 | 14191.12 |
| Signal Gift Stores | 7025551838 | 2003-06-06 | 32641.98 |
| Signal Gift Stores | 7025551838 | 2004-08-20 | 33347.88 |
| Australian Collectors, Co. | 03 9520 4555 | 2003-05-20 | 45864.03 |
| Australian Collectors, Co. | 03 9520 4555 | 2004-12-15 | 82261.22 |
| Australian Collectors, Co. | 03 9520 4555 | 2003-05-31 | 7565.08 |
| Australian Collectors, Co. | 03 9520 4555 | 2004-03-10 | 44894.74 |

```

Table Documentation

Payments

Field Name	Data Type	Values	Note
customerNumber	int	number of customer	primary key
checkNumber	varchar	number of chcking account	
paymentDate	pate	date of payment	
ammount	decimal	amount of payment	

3. We have a customer that has a phone number of 2125558493. Please provide the total this customer has paid us. Write your query so that you don't have to do any math. (You will need to use joins and an aggregate function.)

The total amount is \$69214.33.

```
mysql> select sum(P.amount) as total from customers C
    -> join payments P on C.customerNumber = P.customerNumber
    -> where C.phone = '2125558493';
+-----+
| total |
+-----+
| 69214.33 |
+-----+
1 row in set (0.04 sec)
```

4. Please provide the average payment for customers who are in the state of New York and customers who are NOT in the state of New York. You will need 2 queries to do this.

Average Payments in NY:

```
mysql> select avg(P.amount) as average from customers C join payments P on C.customerNumber = P.customerNumber where C.state = 'NY';
+-----+
| average |
+-----+
| 31466.379412 |
+-----+
1 row in set (0.05 sec)
```

Average Payments NOT in NY:

```
mysql> select avg(P.amount) as average from customers C join payments P on C.customerNumber = P.customerNumber where C.state <> 'NY' and C.state
is not NULL;
+-----+
| average |
+-----+
| 32337.312617 |
+-----+
1 row in set (0.04 sec)
```

5. For orders that have shipped, what is the average spent per item.

Average Spent Per Item: \$91.03358.

```
mysql> select avg(OD.priceEach) as averagePerItem from orders O
      -> join orderdetails OD on O.orderNumber = OD.orderNumber
      -> where O.status = 'Shipped';
+-----+
| averagePerItem |
+-----+
|      91.033558 |
+-----+
1 row in set (0.05 sec)
```

Table Documentation

orders

Field Name	Data Type	Values	Note
orderNumber	int	number of order	primary key
orderDate	date	date of order	
requiredDate	date	date required for order	
shipped date	date	date when order shipped	
status	varchar	current status for the order	
comments	text	comments left for the order	
customerNumber	int	corresponding customer number	

orderdetails

Field Name	Data Type	Values	Note
orderNumber	int	number of order	primary key
productCode	varchar	code of product	
quantityOrdered	int	number of items ordered	
priceEach	decimal	price of each product	
orderLineNumber	int	number of order line	

6. Please provide a list of all employees and who they report to (I want names). I am told there are 23 people that work at this company.

```

mysql> select concat(E1.firstName, ' ', E1.lastName) as employee, concat(E2.firstName, ' ', E2.lastName) as manager from employees E1
-> left join employees E2 on E1.reportsTo = E2.employeeNumber;
+-----+-----+
| employee | manager |
+-----+-----+
| Diane Murphy | NULL |
| Mary Patterson | Diane Murphy |
| Jeff Firrelli | Diane Murphy |
| William Patterson | Mary Patterson |
| Gerard Bondur | Mary Patterson |
| Anthony Bow | Mary Patterson |
| Leslie Jennings | Anthony Bow |
| Leslie Thompson | Anthony Bow |
| Julie Firrelli | Anthony Bow |
| Steve Patterson | Anthony Bow |
| Foon Yue Tseng | Anthony Bow |
| George Vanauf | Anthony Bow |
| Loui Bondur | Gerard Bondur |
| Gerard Hernandez | Gerard Bondur |
| Pamela Castillo | Gerard Bondur |
| Larry Bott | Gerard Bondur |
| Barry Jones | Gerard Bondur |
| Andy Fixter | William Patterson |
| Peter Marsh | William Patterson |
| Tom King | William Patterson |
| Mami Nishi | Mary Patterson |
| Yoshimi Kato | Mami Nishi |
| Martin Gerard | Gerard Bondur |
+-----+-----+
23 rows in set (0.04 sec)

```

Table Documentation

employees

Field Name	Data Type	Values	Note
employeeNumber	int	number of employee	primary key
lastName	varchar	last name of employee	
firstName	varchar	first name of employee	
extension	varchar	extension of employee	
email	varcahr	email of employee	
officeCode	varchar	office code of employee	
reportsTo	int	if of person employee reports to	
jobTitle	varchar	jobtitle of employee	

7. *MSRP is the manufacturer's suggested retail price. Buy price is the price the classic models is charged. Our markup on a product is the difference. Please create a query that shows the buyprice, the MSRP, the markup (labeled markup) and the first 5 characters of the product line textDescription. You should order this by the markup and limit it to 5, so that I can show our board the 5 most marked up products.*

```

mysql> select P.buyPrice, P.MSRP, (P.MSRP-P.buyPrice) as markup, left(PL.textDescription, 5) as first5 from products P
-> join productlines PL on P.productLine = PL.productLine order by markup desc limit 5;
+-----+-----+-----+-----+
| buyPrice | MSRP | markup | first5 |
+-----+-----+-----+-----+
| 98.58 | 214.30 | 115.72 | Atten |
| 95.59 | 207.80 | 112.21 | Atten |
| 91.02 | 193.66 | 102.64 | Our m |
| 95.34 | 194.57 | 99.23 | Atten |
| 72.56 | 168.75 | 96.19 | Our V |
+-----+-----+-----+-----+
5 rows in set (0.05 sec)

```

Table Documentation

Products

Field Name	Data Type	Values	Note
productCode	varchar	code of product	primary key
productName	varchar	name of product	
productScale	varchar	scale of product	
productVendor	varchar	vendor of product	
productDescription	text	product description	
quantityInStock	int	stock of product	
buyPrice	decimal	buy price of product	
MSRP	decimal	MSRP or product	

ProductLines

Field Name	Data Type	Values	Note
productLine	varchar	product line	primary key
textDescription	varchar	description of product line	
htmlDescription	text	html description of product line	
image	blob	img of product line	

8. Please provide a list of employees that had an order in 2005.

```

mysql> select distinct concat(E.firstName, ' ', E.lastName) as Name from employees E
    -> join customers C on E.employeeNumber = C.salesRepEmployeeNumber
    -> join orders O on C.customerNumber = O.customerNumber where year(O.orderDate) = 2005;
+-----+
| Name |
+-----+
| Leslie Jennings |
| Leslie Thompson |
| Julie Firrelli |
| Steve Patterson |
| Foon Yue Tseng |
| George Vanauf |
| Loui Bondur |
| Gerard Hernandez |
| Pamela Castillo |
| Larry Bott |
| Barry Jones |
| Andy Fixter |
| Peter Marsh |
| Mami Nishi |
+-----+
14 rows in set (0.04 sec)

```

9. Please provide a total of all orders with customers who are in postalCodes of 97562, 80686 and 44000

```

mysql> select sum(OD.quantityOrdered * OD.priceEach) as total from customers C
    -> join orders O on C.customerNumber = O.customerNumber
    -> join orderdetails OD on O.orderNumber = OD.orderNumber
    -> where C.postalCode in ('97562', '80686', '44000');
+-----+
| total |
+-----+
| 899449.54 |
+-----+
1 row in set (0.04 sec)

```

Also, I have been trying to teach myself SQL. Can you help me with joins. If I want records from 2 tables where there is a match on the left table and the right table, what type of join do I want.

INNER JOIN

If I want records from 2 tables where I get ALL the records on the right table and only matching records from the left table, what type of Join do I want?

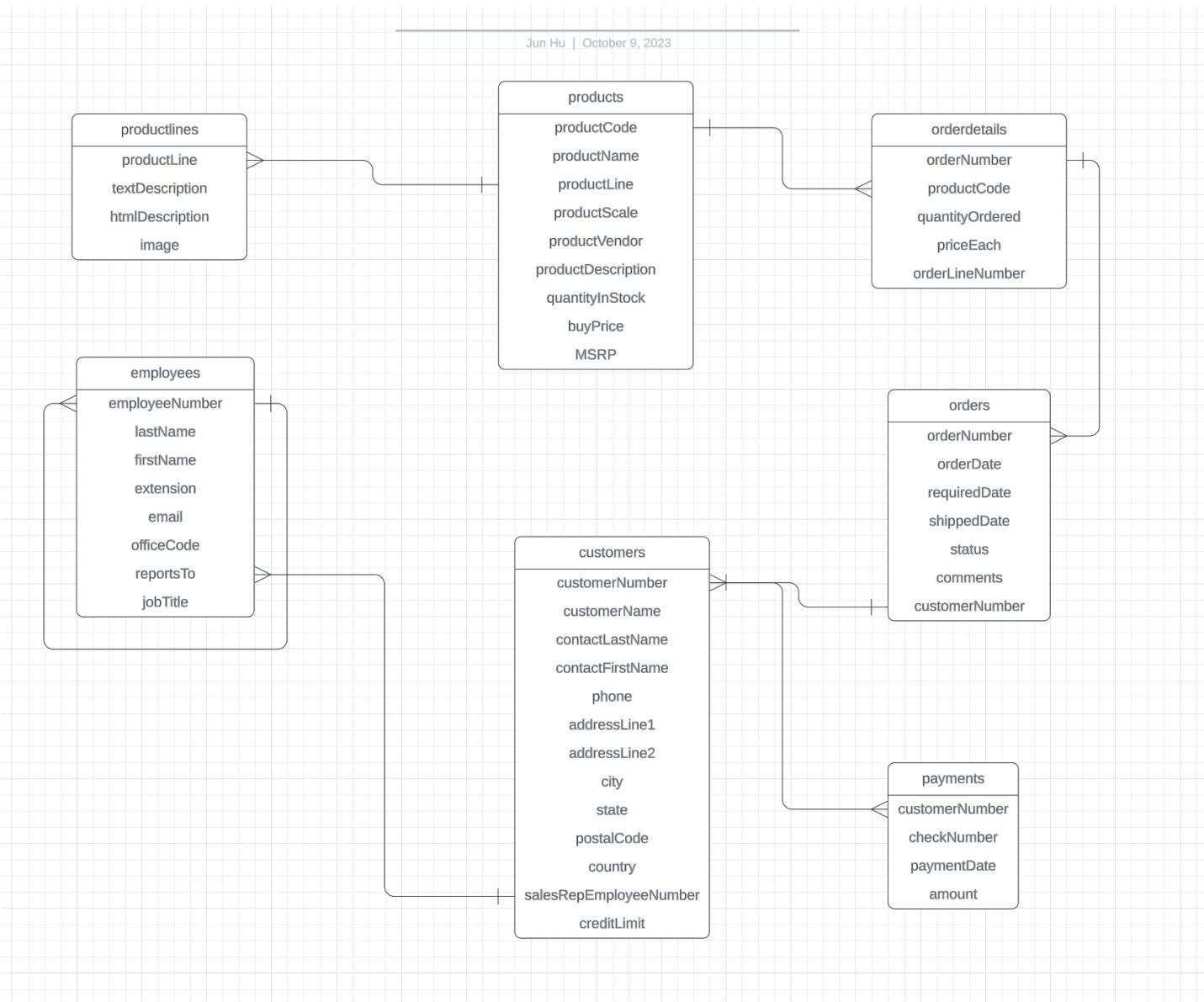
RIGHT JOIN

Also I have attached my ERD for the classicmodels database at the end. Hopfully that will be helpful.

Regards,

Jun

ERD



Dear Lawrence,

Hope you had a lovely honeymoon! It is my pleasure to answer your questions and here are my answers with your questions:

Part 1: From the employee database

1. How many employees do we have born in each month? *

```
mysql> select month(birth_date) as month, count(*) as number from employees
-> group by month(birth_date) order by month(birth_date);
+-----+-----+
| month | number |
+-----+-----+
|     1 |   25412 |
|     2 |   23483 |
|     3 |   25649 |
|     4 |   24631 |
|     5 |   25113 |
|     6 |   24712 |
|     7 |   25698 |
|     8 |   25262 |
|     9 |   24720 |
|    10 |   25518 |
|    11 |   24500 |
|    12 |   25326 |
+-----+
12 rows in set (0.33 sec)
```

[employees](#)

Field Name	Data Type	Values	Note
emp_no	int	employee number	primary key
birth_date	date	birthdate of employee	
first_name	varchar	first name of employee	
last_name	varchar	last name of employee	
gender	enum('M', 'F')	gender of employee	
hire_date	date	employee hired date	

2. What day(as in day of the month) on average do we hire the most people* ?

3 is the day in a month hired most people.

```

mysql> select day(hire_date) as day,
    -> count(*)/(select count(distinct date_format(hire_date, '%Y-%m')) from employees) as hires
    -> from employees group by day(hire_date) order by hires desc;
+-----+-----+
| day | hires |
+-----+-----+
|   3 | 55.9006 |
|  28 | 55.5635 |
|  16 | 55.2928 |
|  25 | 55.1602 |
|  14 | 55.0718 |
|  24 | 55.0331 |
|   7 | 55.0276 |
|   4 | 54.9613 |
|  13 | 54.8840 |
|  21 | 54.6851 |
|  22 | 54.6796 |
|   6 | 54.5912 |
|   8 | 54.5856 |
|  12 | 54.5856 |
|  11 | 54.5359 |
|  20 | 54.4807 |
|  17 | 54.3923 |
|   2 | 54.3923 |
|  18 | 54.2486 |
|  23 | 54.2431 |
|   9 | 54.2210 |
|  10 | 54.1160 |
|   1 | 54.0773 |
|  15 | 53.9061 |
|  26 | 53.7348 |
|  27 | 53.6851 |
|  19 | 53.4586 |
|   5 | 53.2486 |
|  29 | 50.4365 |
|  30 | 49.5138 |
|  31 | 30.8785 |
+-----+-----+
31 rows in set (1.21 sec)

```

3. What is the average salary by job title currently for all staff? *

```

mysql> select t.title, avg(s.salary) as avg_salary from titles t
    -> join salaries s on t.emp_no = s.emp_no
    -> where t.from_date <= now() and (t.to_date > now() or t.to_date is null)
    -> and s.from_date <= now() and (s. to_date > now() or s.to_date is null)
    -> group by t.title;
+-----+-----+
| title          | avg_salary |
+-----+-----+
| Senior Engineer | 70823.4376 |
| Staff           | 67330.6652 |
| Senior Staff    | 80706.4959 |
| Engineer        | 59602.7378 |
| Assistant Engineer | 57317.5736 |
| Technique Leader | 67506.5903 |
| Manager          | 77723.6667 |
+-----+-----+
7 rows in set (5.61 sec)

```

Title

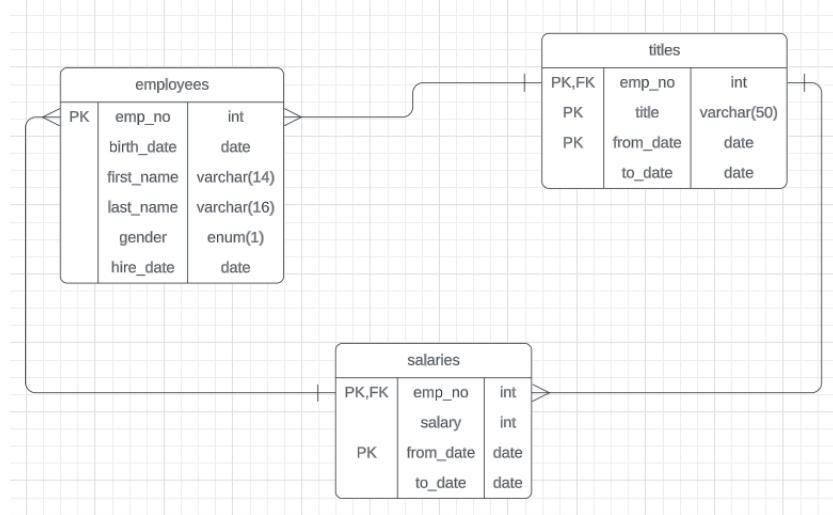
Field Name	Data Type	Values	Note
emp_no	int	employee number	primary key
title	varchar	title of employee	
from_date	date	employee hired from date	
to_date	date	employee left company date	

Salaries

Field Name	Data Type	Values	Note
emp_no	int	employee number	primary key
salary	int	salary of employee	
from_date	date	employee hired from date	
to_date	date	employee left company date	

4. What is the average salary for all folks that are currently employed by year of hire? *

```
mysql> select year(e.hire_date) as year, avg(s.salary) as avg_salary
-> from employees e join salaries s on e.emp_no = s.emp_no
-> where s.from_date <= now() and (s.to_date > now() or s.to_date is null)
-> group by year(e.hire_date) order by year;
+-----+-----+
| year | avg_salary |
+-----+-----+
| 1985 | 78870.3162 |
| 1986 | 77411.4463 |
| 1987 | 75927.5882 |
| 1988 | 74201.5604 |
| 1989 | 73053.4454 |
| 1990 | 71483.8574 |
| 1991 | 69812.8034 |
| 1992 | 68286.0711 |
| 1993 | 67090.8002 |
| 1994 | 65332.5509 |
| 1995 | 63705.1261 |
| 1996 | 62424.6746 |
| 1997 | 60794.5994 |
| 1998 | 59673.0602 |
| 1999 | 58199.3812 |
| 2000 | 58192.1111 |
+-----+-----+
16 rows in set (4.74 sec)
```



Part 2: From the research1 database

- By username what is the average number of steps for July. *

```

mysql> select u.name as username, avg(f.fitbit_steps) as steps7
-> from fitbit_day_detail f join users_field_data u on f.user_id = u.uid
-> where month(f.fitbit_date) = 7 group by u.name;
+-----+-----+
| username | steps7 |
+-----+-----+
| 1de2e393b047677dcf7cf5f729c3afc4 | 7.0396 |
| 82c8ca7904fea3535400823529ade611 | 6.0800 |
| c95edebebbb7ffac997419157cd0e4e9 | 3.0955 |
| 00e873bcbfa8c6171db3d1afbf6bf0cf | 7.4410 |
| 44a688027cc06a0ad4f399e3b7a1cc87 | 1.6860 |
| a1ad3be33cf61d95d8f21a93a094c747 | 3.6324 |
+-----+-----+
6 rows in set (3.65 sec)
    
```

fitbit_day_detail

Field Name	Data Type	Values	Note
id	int	id of detail	primary key
user_id	int	id of corresponding user	
fitbit_steps	int	steps of detail	
created	int	person who created the api	
changed	int	person who changed the api	
fitbit_date	date	date of the detail	

users_field_data

Field Name	Data Type	Values	Note
uid	int	id of user	primary key
langcode	varchar	corresponding langcode of user	
name	varchar	name of user	
mail	varchar	mail of user	
timezone	varcahr	timezone of user	

2. For the user with the name 'f9f67f5beddc05e72d4c1715c26df95d', what is the average number of minutes they sleep each month. Remember to write this as one query. *

```
mysql> select month(f.fitbit_date) as month, avg(f.fitbit_timeinbed) as sleep7
    -> from fitbit_sleep f join users_field_data u on f.user_id = u.uid
    -> where u.name = 'f9f67f5beddc05e72d4c1715c26df95d'
    -> group by month(f.fitbit_date);
Empty set (0.04 sec)
```

fitbit_sleep

Field Name	Data Type	Values	Note
id	int	id of detail	primary key
user_id	int	id of corresponding user	
name	varchar	name of user	
fitbit_date	date	date of active	
fitbit_duration	int	duration of fitbit	
fitbit_efficiency	int	efficiency of fitbit	
fitbit_timeinbed	int	sleep of fitbit	
fitbit_ismainsleep	int	whether fitbit main sleep	
created	int	user who created the api	

3. Please write a query that lists each user_id and the max steps they walked in a single day for each month.

```

mysql> select f.user_id, month(f.fitbit_date) as month, max(f.fitbit_steps) as steps
-> from fitbit_day_detail f group by f.user_id, month(f.fitbit_date) order by user_id, month;
+-----+-----+-----+
| user_id | month | steps |
+-----+-----+-----+
|    148 |     5 |      0 |
|    148 |     6 |    132 |
|    148 |     7 |    176 |
|    148 |     8 |    140 |
|    148 |     9 |    151 |
|    148 |    10 |    139 |
|    148 |    11 |    142 |
|    207 |     7 |    146 |
|    207 |     8 |    152 |
|    207 |     9 |    174 |
|    207 |    10 |    165 |
|    475 |     7 |    117 |
|    475 |     8 |    155 |
|    475 |     9 |    151 |
|    475 |    10 |    117 |
|    592 |     5 |    125 |
|    592 |     6 |    160 |
|    592 |     7 |    167 |
|    592 |     8 |    152 |
|    592 |     9 |    137 |
|    592 |    10 |    138 |
|    592 |    11 |    146 |
|    601 |     7 |    184 |
|    601 |     8 |    144 |
|    601 |     9 |      0 |
|    601 |    10 |      0 |
|    648 |     6 |    153 |
|    648 |     7 |    168 |
|    648 |     8 |    176 |
|    648 |     9 |    149 |
|    648 |    10 |    160 |
|    648 |    11 |    133 |
+-----+-----+-----+
32 rows in set (1.54 sec)

```

4. Please provide a listing of each user_id and the if they met their goal on average per month (This is a quote from one of our researchers, figure out how to answer it and describe how you answered it)

```

mysql> select g.user_id, sum(f.fitbit_steps)/count(distinct date_format(f.fitbit_date, '%Y-%m')) as avg_steps
-> from goal_entity g join fitbit_day_detail f on g.user_id = f.user_id
-> group by g.user_id having avg(g.goal) < avg_steps;
+-----+-----+
| user_id | avg_steps |
+-----+-----+
|    148 | 4392671.4286 |
|    207 | 2017531.2500 |
|    601 | 963651.5000 |
|    648 | 2299503.5000 |
+-----+-----+
4 rows in set (2 min 7.38 sec)

```

goal_entity

Field Name	Data Type	Values	Note
id	int	id of detail	primary key
user_id	int	id of corresponding user	
goal	int	goal of steps	
date	varchar	date of goal set	

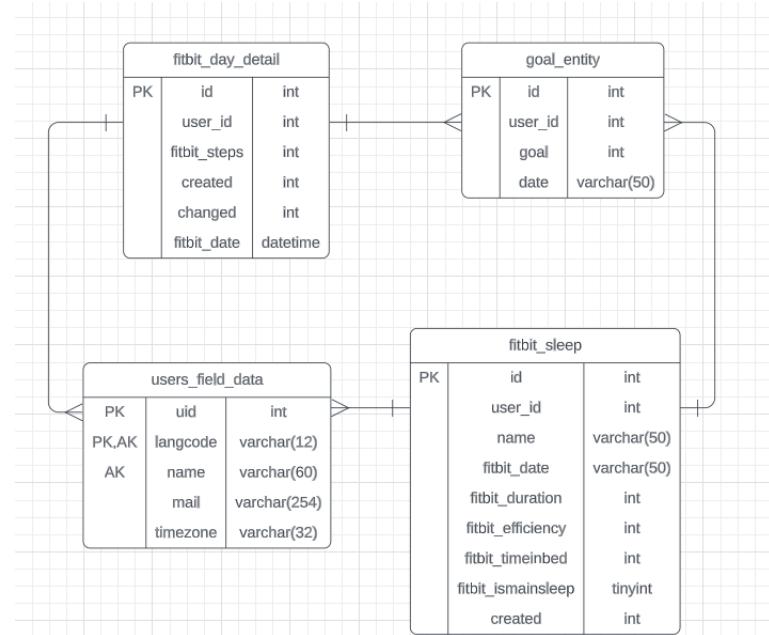
5. What days did users get more than 8 hours of sleep? Please provide me a sample of the user_id's, the names and the days if the list is too long. Your query should NOT contain any numbers above 100. *

The output table is too long so I showed randomly 10 rows.

```
mysql> select f.user_id, u.name, f.fitbit_date from fitbit_sleep f join users_field_data u on f.user_id = u.uid
-> where f.fitbit_timeinbed >= 8*60 limit 10;
+-----+-----+-----+
| user_id | name      | fitbit_date |
+-----+-----+-----+
| 148    | 1de2e393b047677dcf7cf5f729c3afc4 | 2019-07-06 |
| 148    | 1de2e393b047677dcf7cf5f729c3afc4 | 2019-07-04 |
| 148    | 1de2e393b047677dcf7cf5f729c3afc4 | 2019-07-03 |
| 148    | 1de2e393b047677dcf7cf5f729c3afc4 | 2019-06-30 |
| 148    | 1de2e393b047677dcf7cf5f729c3afc4 | 2019-06-29 |
| 148    | 1de2e393b047677dcf7cf5f729c3afc4 | 2019-06-27 |
| 148    | 1de2e393b047677dcf7cf5f729c3afc4 | 2019-06-25 |
| 592    | 82c8ca7904fea3535400823529ade611 | 2019-07-07 |
| 592    | 82c8ca7904fea3535400823529ade611 | 2019-07-06 |
| 592    | 82c8ca7904fea3535400823529ade611 | 2019-07-04 |
+-----+-----+-----+
10 rows in set (0.04 sec)
```

6. For user with the name of 82c8ca7904fea3535400823529ade611, what were the average number of steps taken per month per min?

```
mysql> select date_format(fitbit_date, '%Y-%m') as month, sum(fitbit_steps)/count(distinct fitbit_date) as avg_steps_min
-> from fitbit_day_detail f join users_field_data u on f.user_id = u.uid
-> where u.name = '82c8ca7904fea3535400823529ade611' group by date_format(fitbit_date, '%Y-%m') order by month;
+-----+-----+
| month | avg_steps_min |
+-----+-----+
| 2019-05 | 9.2243 |
| 2019-06 | 7.1256 |
| 2019-07 | 6.0800 |
| 2019-08 | 6.8825 |
| 2019-09 | 7.1243 |
| 2019-10 | 7.2644 |
| 2019-11 | 7.2535 |
+-----+-----+
7 rows in set (2.68 sec)
```



I hope these insights are of value to you! On a side note, I haven't been informed about our leader's promotion. If this is accurate, do you think there will be any changes within our team?

Regards,

Jun

Welcome to SI564

SQL and Databases



Plan to drop?
Why are you
here then 😊

If you plan to drop, please do it now. You don't have to leave the room, but it will open the wait list up.

Did you just add the class?

- Feel free to take an extra week on this weeks homework.
- Indicate when you turn in your homework, that is why it is late.
- If your database is not working by Wed, message us.

Audits

- I have been informed by the RO, that class audits are not allowed. I don't have the authority to authorize them and the RO won't do them.

Optional Homework Questions

Taxdata table

Large table, use limits (we will cover today)

- DO NOT WRITE A QUERY LIKE
- SELECT * FROM TAXDATA

Homework

You may lose points.

- If we have not taught how to do something in SQL, please don't skip ahead.
- You will lose points on your homework if you use SQL syntax that we have not covered. We have not covered where yet. Don't use where. We have not covered count yet, don't use count.
- If you have questions, ask

Home work Ambiguity

- This is by design. The teaching team does not have an answer key.
- The real world does not have an answer key.
- Just explain



Ask for help!

Office Hours

- Office hours link
- umsi.in/564office
- This includes my time and the GSI's times.
- WHEN BOOKING PLEASE INDICATE WHAT CLASS YOU ARE IN.

Slack

- Please join slack.
- Go to slack.umich.edu
- An hour after you join, it should add you to channels.
- Feel free to ask questions in slack.
- DO NOT POST HOMEWORK ANSWERS OR HINTS.

Missing databases or other infra

- Reach out via slack or email

Class Pacing

- Slow and steady
- Ask questions
- We will pick up the pace in 2-3 weeks

Mental Health

- Mental health is important.
- If you are having issues, with this course, or life, or anything, please reach out to the support systems we have in place.
- If you need an extension on something, given the current world state. Let me know.
- As we move forward, please have understanding. Call out issues early. Let us know what I can do to support you.



How are you doing this week?

ⓘ Start presenting to display the poll results on this slide.



Python

Query databases

- There are 4 different basic types of SQL queries
- (There is another entire subset used to create/drop 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

Side Note

- Pressing control-c can stop a query from running in most cases.

Stopping a query.

Most be on your RW account.

You most likely won't need this.

- 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 performs 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.

Prep for Homework

Name : _____ Score : _____
Teacher : _____ Date : _____

$$\begin{array}{r} 81 \\ \times 76 \\ \hline \end{array} \quad \begin{array}{r} 15 \\ \times 12 \\ \hline \end{array} \quad \begin{array}{r} 78 \\ \times 25 \\ \hline \end{array} \quad \begin{array}{r} 8 \\ \times 22 \\ \hline \end{array} \quad \begin{array}{r} 48 \\ \times 28 \\ \hline \end{array} \quad \begin{array}{r} 19 \\ \times 9 \\ \hline \end{array}$$

$$\begin{array}{r} 91 \\ \times 11 \\ \hline \end{array} \quad \begin{array}{r} 98 \\ \times 8 \\ \hline \end{array} \quad \begin{array}{r} 4 \\ \times 26 \\ \hline \end{array} \quad \begin{array}{r} 18 \\ \times 15 \\ \hline \end{array} \quad \begin{array}{r} 37 \\ \times 17 \\ \hline \end{array} \quad \begin{array}{r} 95 \\ \times 29 \\ \hline \end{array}$$

$$\begin{array}{r} 54 \\ \times 66 \\ \hline \end{array} \quad \begin{array}{r} 97 \\ \times 13 \\ \hline \end{array} \quad \begin{array}{r} 94 \\ \times 1 \\ \hline \end{array} \quad \begin{array}{r} 83 \\ \times 7 \\ \hline \end{array} \quad \begin{array}{r} 25 \\ \times 5 \\ \hline \end{array} \quad \begin{array}{r} 9 \\ \times 61 \\ \hline \end{array}$$

$$\begin{array}{r} 44 \\ \times 24 \\ \hline \end{array} \quad \begin{array}{r} 17 \\ \times 23 \\ \hline \end{array} \quad \begin{array}{r} 85 \\ \times 6 \\ \hline \end{array} \quad \begin{array}{r} 96 \\ \times 41 \\ \hline \end{array} \quad \begin{array}{r} 58 \\ \times 20 \\ \hline \end{array} \quad \begin{array}{r} 45 \\ \times 0 \\ \hline \end{array}$$

$$\begin{array}{r} 47 \\ \times 27 \\ \hline \end{array} \quad \begin{array}{r} 38 \\ \times 4 \\ \hline \end{array} \quad \begin{array}{r} 84 \\ \times 14 \\ \hline \end{array} \quad \begin{array}{r} 77 \\ \times 36 \\ \hline \end{array} \quad \begin{array}{r} 27 \\ \times 10 \\ \hline \end{array} \quad \begin{array}{r} 5 \\ \times 56 \\ \hline \end{array}$$

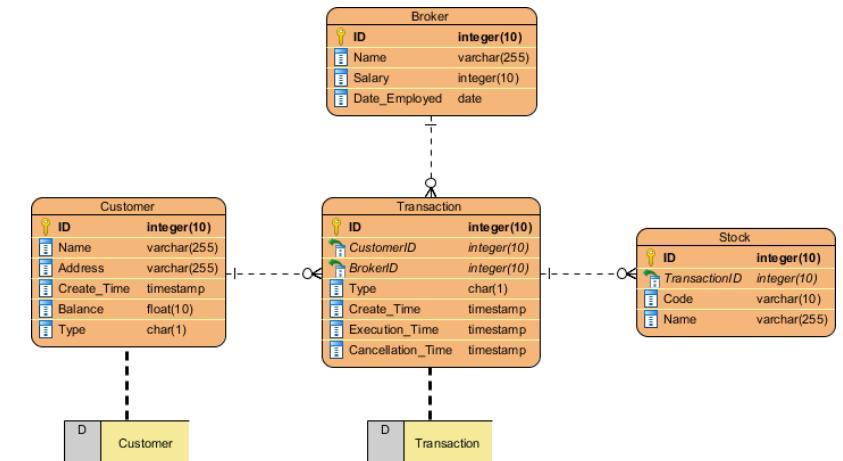
Archaeologist

- 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 and data dictionary

- 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, stone and chisel, etc.
- Your diagram's don't have to be fancy.

For NOW

- Do not worry about the “picture” aspect of this. Just write up what you find in each table.
- Don’t over think this, but it can take some time to do.

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 and then write queries to validate.
- If you really can't figure something out, indicate that.

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_DESC_RSHORT	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.

Documentation

- ONLY document the tables for now, NOT what the relationships. We have not covered that yet.

Homework Example

Table = People

Field Name	Data type	Values	Notes
ID	number	numbers	Primary Key
Name	varchar	Names of people	
City	varchar	The city where the person lives	
Age	int	Range from 1-130	
Skill	longtext	What the person's skill(s) are	

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 your 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. You will lose points.**

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	VLT
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	
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*

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 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 DISTINCTfieldname FROMtable;
 - 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;

DO NOT DOCUMENT A TABLE

- Unless there is a question about it.
- If we ask you to look at a database that has 5 tables it, but only use one of the tables to answer the question, then only document that table.

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. (Next week ☺)

Date Functions

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

- Questions

Information
changes
everything.

THANK YOU



Welcome to SI564

SQL and Databases





Favorite Ann Arbor restaurant?

ⓘ Start presenting to display the poll results on this slide.

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 me know asap.**
- 3) You must use a terminal to access data. Do not use a GUI client running on your computer (Questions? Feel free to ask in slack) After this week, if you use a GUI you will lose points.**
- 4) If you dropped one category on either scale, it could be formatting or a simple error.**

Show your work

- Screenshots
- Query log
- Anything that works for you

Office Hours

Link in canvas

- If you would like to come to office hours, please book online.
- If all the dates/times conflict with a class or other commitment , let me know.

Ambiguity

- This class has some 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 the GSI 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.
- As we move forward the ways in which to answer the questions will become more blurred.

Contacting us

- Please do not use canvas to contact us.
(Yes this is a thing)
- Book office hours umsi.in/564office
- Slack

Attendance

- Please log your attendance in canvas.
- Login. Under week 3, choose today's date.
- SELECT SQL
 - Is the code
- (all caps, 1 space)
- Please note, if you are not in the classroom, and complete this, it is an academic integrity violation. You will get a zero and it will go to "HR" for review.

Mental Health Reminder

Calendar

- Today – limit data pulls
- Joins (Video will be posted before class, please watch)
- Joins again
- Group by
- Fall break
- Midterm!

Midterm

- We lost a Monday due to Labor day.
- Midterm is the 23rd of OCT.
- Monday 9th is new content (that will be on the midterm)
- Wednesday 11th at 2:30 is a review sessions (in Lindsey's lab)
- Sunday the 22nd there _maybe_ a review session
- Monday 23rd is the midterm.

Policy questions

Databases are large

- Almost never do we want ALL the data in a table.
- Sometimes we just want a subset of all the data.
- Sometimes we want a subset and a single field.

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

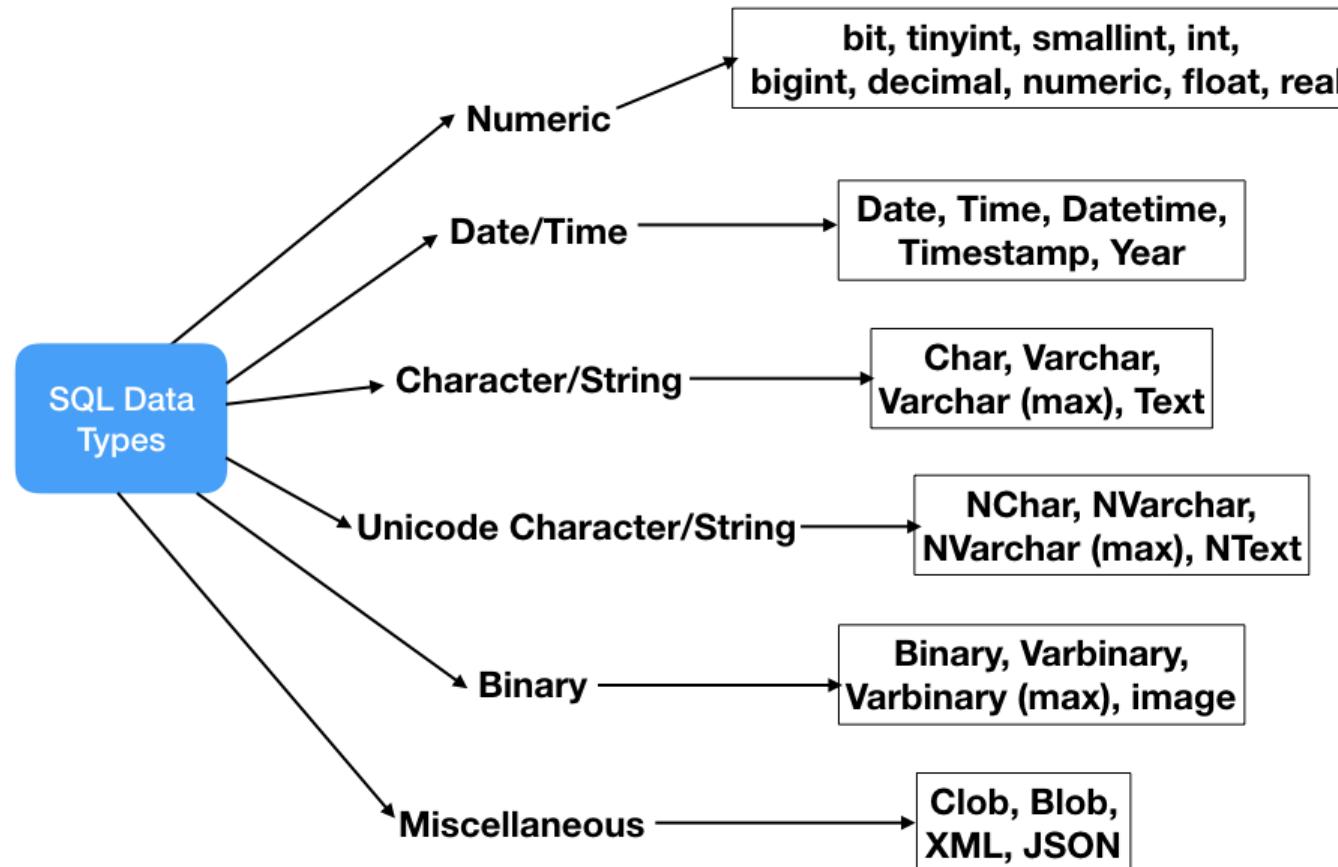
- The simple form of this is
- WHERE field = SOMETHING;

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 making “pictures”

Field Types



Field Types

Fall into groups?

- Int/number
- 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.
 - Even if it displays a warning, it might work, but fix it don't depend on it fixing things for you!

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.

FullText 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

NULL IS SPECIAL

- 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
- WHERE DOES YOUR LOGIC GO?

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)
%l	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;`

Thank you

Next Week

- Joins across tables
- Please watch the video before class.
- GSI's will be doing the live demo

Information
changes
everything.

THANK YOU



Welcome to SI564

SQL and Databases



New Terminal

- Shift - insert

Regrade Policy

Any assignment can be resubmitted for credit as long as it meets the following conditions: The assignment lost **more than 9 points** in the professionalism category.

The assignment lost **more than 8 points** in the accuracy category.

(or) The assignment lost **more than 15 points** over both categories.

The assignment is resubmitted **within 7 days** of the assignment being released.

Additional Information: After your work has been resubmitted, regrading will take an average of **three weeks** to complete.

If you have extenuating circumstances, please contact the entire teaching team via Slack or email, and we can address any concerns you may have.

If you have lost a total of 15 or more points across more than three assignments, you can contact the teaching team for extra credit.

Attendance

- The GSI's will post a question in slack, please answer it by the end of class.

Calendar

- Last week – limit data pulls
- Today Joins (Video will be posted before class, please watch)
- Joins again
- Group by
- Fall break
- Midterm!

Midterm

- We lost a Monday due to Labor day.
- Midterm is the 23rd of OCT.
- Monday 9th is new content (that will be on the midterm)
- Wednesday 11th at 2:30 is a review sessions (in Lindsey's lab)
- Sunday the 22nd there _maybe_ a review session
- Monday 23rd is the midterm.

Midterm Questions

Please wait till next week or slack us.

Python and SQL

- Optional Homework being released.
- We will provide support in office hours
- We won't grade this unless you ask us to.

Python work and WHERE part 2

- I am not going to require the python part of the home work
- If you are going into data science, you should do this work.
- Pros:
 - No python code
- Cons:
 - We jump into some harder queries today

Group By

- We have not covered group by in class. If you use group by on your homework, you will lose points.

Moving forward!

- JOINS today
- Review of things next week
- Maybe some of the home work questions will come back.
- Midterm Review

See issues?

- Some of you have had issues with tables missing, if you think you are missing a table let us know so we can get IT to fix the issue.

Office Hours

- If you would like to come to office hours, please book online.
- Umsi.in/564office
- Or ask on slack

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.
- 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.

Ambiguity

- Hello Raj,
- In trying to answer your question, it seems that the dataset contains 3201 items.
- Here is an example 5 lines
 - Blah, blah ,blah blah
- Rather than send this via email, I would like to send it to you using a file server, where would you like me to put it?

- Do not use automated **database** tools to generate your diagrams. This is cheating.
- You should only use the terminal to access your database. Do not use any graphical program.
- Anyone not using terminal (or other approved software) will get 0 points.

Attendance

- A decent amount of work goes into prepping class. Taking attendance saves everyone time. If you are in class, you are less likely to ask questions that are covered in class.
- In a company, you will have meetings, that you can not skip.

Asking questions

Describe the goal, not the step

Way one:

How do I get the color-picker on the FooDraw program to take a hexadecimal RGB value?

Way two:

I'm trying to replace the color table on an image with values of my choosing. Right now the only way I can see to do this is by editing each table slot, but I can't get FooDraw's color picker to take a hexadecimal RGB value.

Describe the symptoms of your problem or bug carefully and clearly.

Describe the research you did to try and understand the problem before you asked the question.

Describe the diagnostic steps you took to try and pin down the problem yourself before you asked the question.

Describe any possibly relevant recent changes in your computer or software configuration.

If at all possible, provide a way to reproduce the problem in a controlled environment.

*Source: Esr
Asking questions
the smart way.*

Homework

- If the CIO sends you an email.
 - And the CIO is your boss's boss
 - Then do not respond with "Yo Kelly"
-
- If the CIO says "Can you write a query that does XYZ"
 - The answer is not Yes or No.
 - You need to answer the question.
 - What does the CIO want?

Subqueries

- Select * from table where **ID** in (select **ID** from table2);
- Question: Who has been at any point a dept manager.
- Select first_name, last_name from employee where emp_no IN (select emp_no from dept_manager);

Like

- When making a query using like, you need to have at least one %;

Join's today

- This week's homework is hard. Joins take some time to wrap your head around.
- Come to office hours if you are stuck
- Ask about concepts on slack (Do not ask about specific questions)
- GSI's and myself are not online 24/7, book office hours.
- Do the readings!

Order of operations

Julia Evans

The query's steps don't happen in the order they're written:

ATION

how the query
is written

how you should
think about it

SELECT ...

FROM + JOIN

WHERE ...

GROUP BY ...

HAVING ...

ORDER BY ...

LIMIT ...

FROM + JOIN

WHERE

GROUP BY

HAVING

SELECT

ORDER BY

LIMIT

(In reality query execution is much more complicated than this.
There are a lot of optimizations.)

SQL is about relationships

Data is related to other data in almost all systems.

- Rdbms
- Data should be *normalized*.
- We will cover normalization later in class.
- But in short, don't duplicate data.
- For example ptid and ptname in tax data.

ERD diagrams

- From this point forward, please make sure your diagrams show
 - The text descriptions you have had up till now.
 - **How data between fields relates.**
- The format of this does not matter as long as you show the information you have been showing and the relationships.
- I show another method in the readings to do this, and I use yet another one in class. Anything is fine.
- You can draw simple lines.

FOREIGN KEY

A Foreign key is used to relate a table to another table.

Normally links to a primary key on another table (but does not have to)

Dataset

This is not a good way to handle this.

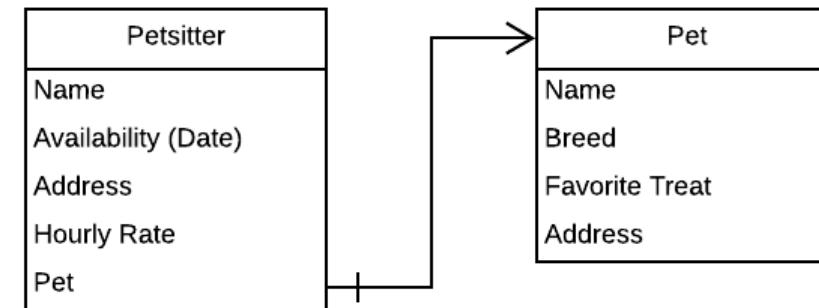
- Classes
 - ID
 - Name
 - Sectionid
 - Credit hours
 - Instructor name
 - Instructor email
 - Instructor office number

Dataset

- Classes
 - ID (PK)
 - Name
 - Sectionid
 - Credit hours
 - InstructorID
 - Instructor
 - ID (PK)
 - name
 - Email
 - Office number
 - Formal title
 -
- 
- FK

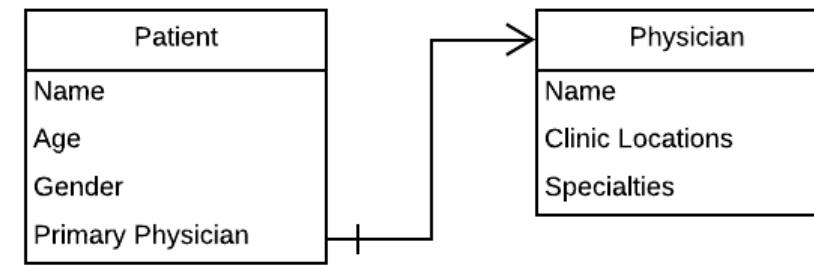
Dataset

What is missing?



Data Sets

What is missing?



PK FK

- Every relationship needs a way to match to other records. The field in the first table is used as a foreign key.
- Every table should have at least 1 primary key.
- Tables can have 0, 1, N foreign keys.

We often need
to query more
than one table.

- We need to be able to present data from more than one table.
- We need to be able to query from both tables when presenting results.

Syntax Rule

- So far we have referenced table fields by field name within a database and in the same table.
- However, all fields can be referred to by tablename.fieldname
- Select name from table1;
- Select table1.name from table1;
- We will need this syntax when dealing with joins.

RO_COMPANY

Give me all fields from both the accounts table and the employee table. Linking on the employee who created the account.

Step 1:

How do these fields link to each other. What field table in our first table references the second table.

Our first table is account.

account.open_emp_id references the employee.emp_id

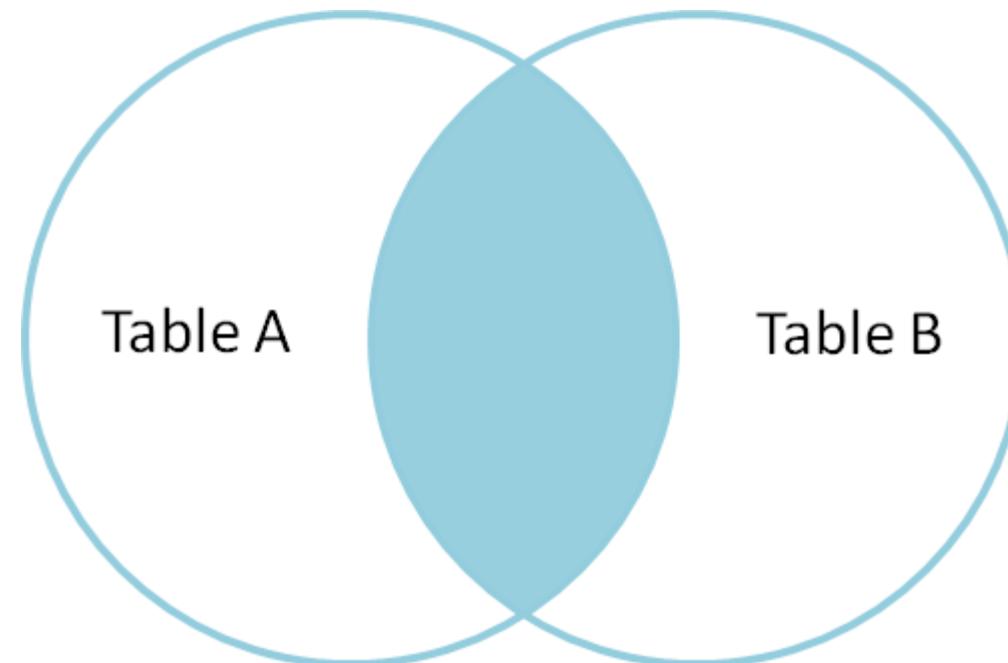
How do we know? In theory we have a proper ERD diagram left by someone who built this. Sometimes you have to guess.

Types of Joins

- There are different types of joins. I will cover these with the readings (and they are long this week), but in short we have
 - (inner) JOIN – matches rows from both tables.
 - Left outer join -- matches ALL rows from the left table and only matching rows from the right table.
 - Right outer join – matches all rows from the right table and only matching rows from the left table. (I don't use this often)
 - Full outer join returns all rows from all tables matching where a match can be made.

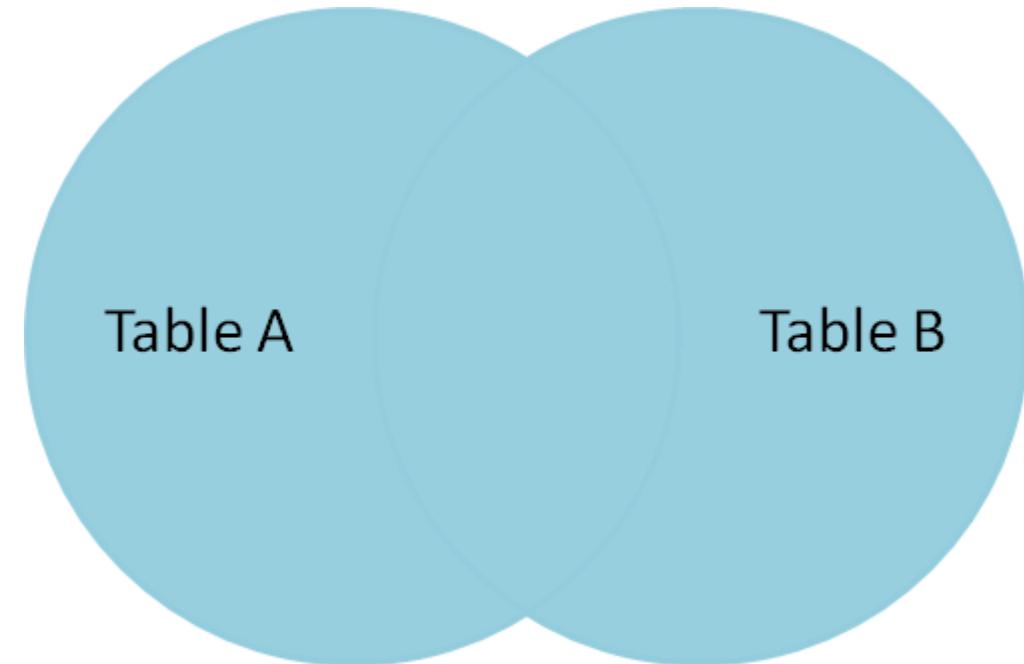
Inner Join

```
SELECT * FROM TableA  
INNER JOIN TableB  
ON TableA.fid = TableB.id
```



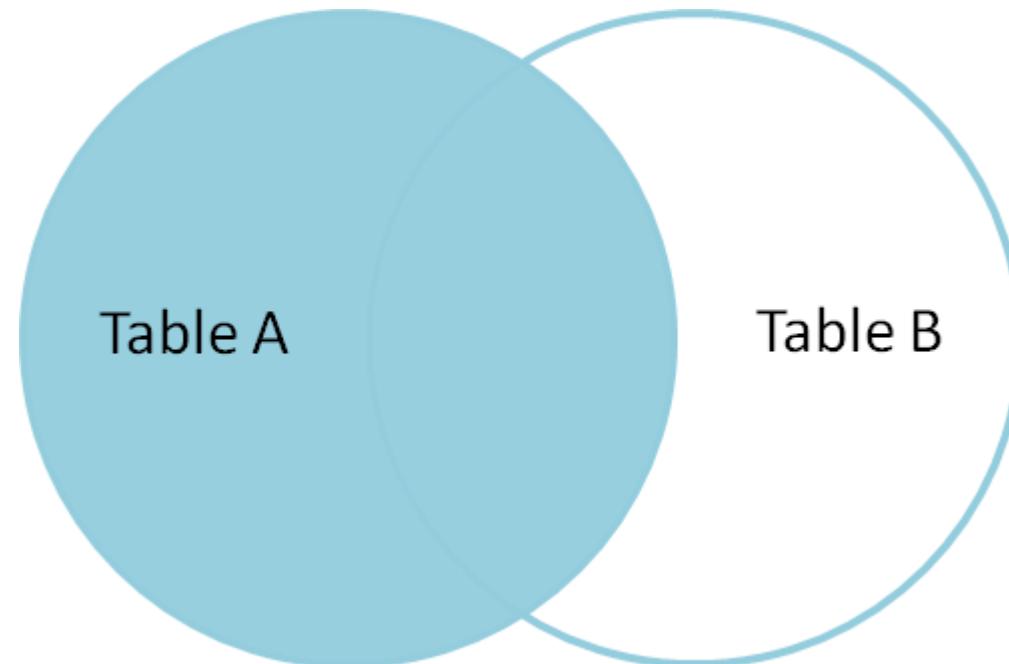
Full Outer Join

```
SELECT * FROM TableA  
FULL OUTER JOIN TableB  
ON TableA.name =  
TableB.name
```



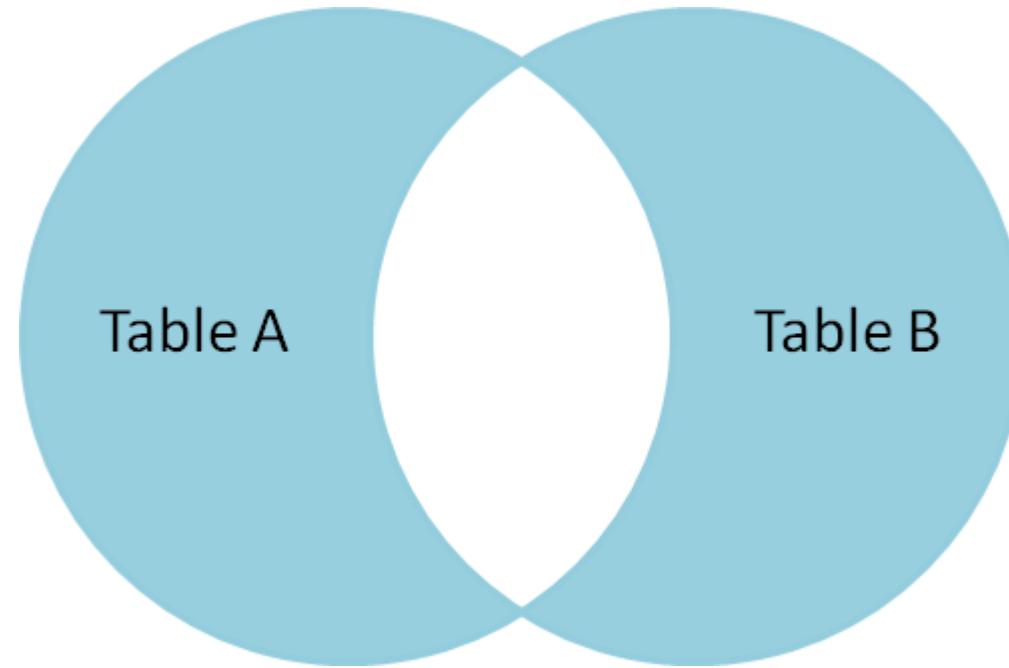
Left/Right outer join

```
SELECT * FROM TableA  
LEFT OUTER JOIN TableB  
ON TableA.name =  
TableB.name
```



Inverse full outer join

```
SELECT * FROM TableA  
FULL OUTER JOIN TableB  
ON TableA.name =  
TableB.name  
WHERE TableA.id IS null  
OR TableB.id IS null
```



- This is not a formal join type

Illustrated

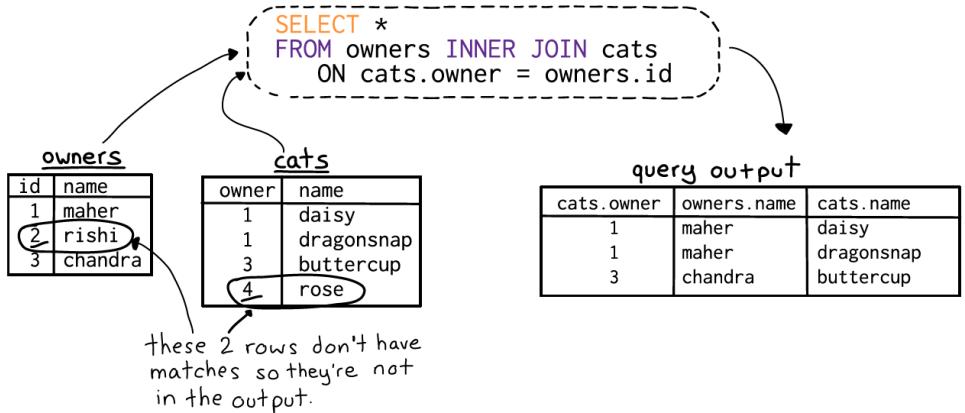
Julia Evans

INNER JOIN and LEFT JOIN

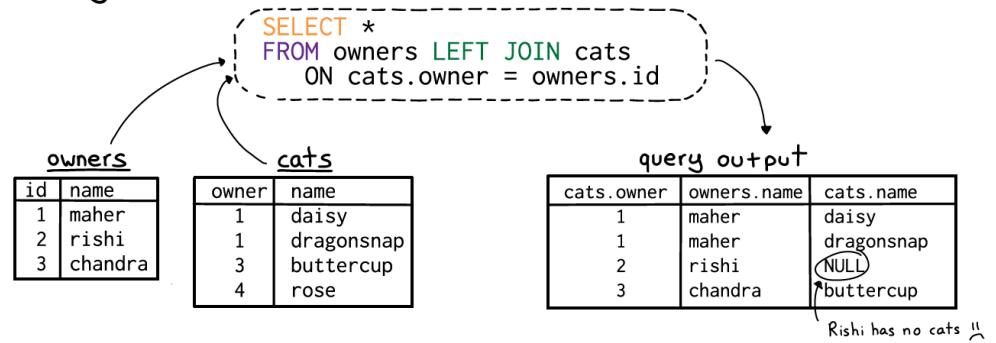
ITION

Here are examples of how INNER JOIN and LEFT JOIN work:

INNER JOIN only includes rows that match the ON condition.
This query combines the cats and owners tables:



LEFT JOIN includes every row from the left table (owners in this example), even if it's not in the right table. Rows not in the right table will be set to NULL.



This is a classic example of a join that follows my 4 guidelines from the previous page:

- 1) it's an INNER JOIN / LEFT JOIN
- 2) it includes the table name in `cats.owner` and `owners.id`
- 3) the condition `ON cats.owner = owners.id` is simple
- 4) it joins on a unique column (the `id` column in the `owners` table)

Types of relationships

- 1-1 JOIN
 - Exactly one row on a match
- 1-many JOIN
 - Always at least 1 row, but might be more
- (0-1) OUTER JOIN OR JOIN
 - 0 or 1
- (0-many) OUTER JOIN OR JOIN
 - 0 or many rows.

Query

- Give me all fields from both the accounts table and the employee table, linking on the employee who created the account.
- Select * from account inner **JOIN** employee **ON** account.open_emp_id=employee.emp_id;
- Select * from account **JOIN** employee **ON** account.open_emp_id=employee.emp_id;

Diagram

- verb what FROM (JOINS) WHERE ORDER
BY LIMIT;

Joins!

- FROM **LEFT_TABLE_JOIN_TYPE_RIGHT_TABLE** ON
lefttable.fieldfromlefttable=righttable.fieldfromrighttable

Show me all
customers and
their officers

- Show me all customers and their officers
- Is this a 1-1, 1-many 0-1 etc
- select count(1) from customer; -> 13
- select count(1) from officer -> 4

Show me all
customers and
their officers

- Both ways
- `select * from customer JOIN officer on customer.cust_id = officer.cust_id;`
- Only returns 4 rows, where there is a union.
- `select * from customer LEFT OUTER JOIN officer on customer.cust_id = officer.cust_id;`
 - `select * from customer LEFT OUTER JOIN officer on customer.cust_id = officer.cust_id;`
- Returns 13 rows

Table Alias

- `select * from customer LEFT OUTER JOIN officer on customer.cust_id = officer.cust_id;`
- This is long. Like fields we can alias a table
- `select * from customer c LEFT OUTER JOIN officer o on c.cust_id = o.cust_id;`

Table Alias

- Show me the title and the federal id from officers and customers;
- Select fed_id, title from customer LEFT OUTER JOIN officer on customer.cust_id = officer.cust_id; <- Don't do this.
- Writing queries like this is confusing later on!
 - What if 2 tables have the same field name?

Table alias

- Always specify where the data comes from
- `select customer.fed_id, officer.title from customer LEFT OUTER JOIN officer on customer.cust_id = officer.cust_id;`
- Order it better
- `select customer.fed_id, officer.title from customer LEFT OUTER JOIN officer on customer.cust_id = officer.cust_id order by officer.title desc;`
- Long query

Lets Table alias it

- select c.fed_id, o.title from customer c LEFT OUTER JOIN officer o on c.cust_id = o.cust_id order by o.title desc;

Table alias notes

Always use table aliases

Always specify the table the data comes from with a table alias

Query to write

- How many of our customers live in the same zipcode as one of our branches.
- There are many ways to write this query

Join on zipcode

- Join on a non defined key
- Zipcode is not a key. However, you can still join on it.
- `select * from customer c join branch b on b.zip=c.postal_code;`

Subquery

- `select * from customer where postal_code in (select zip from branch);`
- Upside: Faster query
- Downside, can't pull branch information in
- Also what is wrong with this table design?

Self join

- For 50 rows, show me the 2013 and the 2014 revenue for the same companies
- What is the unique company identifier here?

Taxdata – self join

- select ein, t2.ein, t1.year, t1.revenue, t2.year,t2.revenue from taxdata t1 join taxdata t2 on t1.ein=t2.ein where t1.year = 2014 and t2.year=2013 limit 5;
- What is wrong with this? What did I forget to do?

ambiguous

- Column 'ein' in field list is ambiguous
- This column is in both t1 and t2. SQL can't figure out which one I want to query.
- This is why getting in the habit of always saying what field you want is helpful.

Debug query

- ```
select t1.ein, t2.ein, t1.year, t1.revenue, t2.year,t2.revenue from taxdata t1 join taxdata t2 on t1.ein=t2.ein where t1.year = 2014 and t2.year=2013 limit 5;
```
- We can confirm that the ein's and the years make sense.

## Final query

- mysql> select t1.ein as ein, t1.revenue as 2014rev,t2.revenue as 2013rev from taxdata t1 join taxdata t2 on t1.ein=t2.ein where t1.year = 2014 and t2.year=2013 limit 5;
- +-----+-----+-----+
- | ein | 2014rev | 2013rev |
- +-----+-----+-----+
- | 462212474 | 95852 | 182611 |
- | 42985201 | 43275 | 20050 |
- | 510143732 | 86715 | 74195 |
- | 463065523 | 46350 | 600 |
- | 521119677 | 592132 | 428436 |
- +-----+-----+-----+
- 5 rows in set (0.15 sec)

# Can we do better?

- It is our job to understand the business need behind the query. If someone is asking for this, we can assume that we need to compare 2013 to 2014.
- ```
mysql> select t1.ein as ein, t1.revenue as 2014rev,t2.revenue as 2013rev, (t1.revenue- t2.revenue) as diff from taxdata t1 join taxdata t2 on t1.ein=t2.ein where t1.year = 2014 and t2.year=2013 limit 5;
```
- We can calculate the difference.

Maybe you get curious

- Now you can query the largest difference between years;
- select t1.ein as ein, t1.revenue as 2014rev,t2.revenue as 2013rev, (t1.revenue- t2.revenue) as diff from taxdata t1 join taxdata t2 on t1.ein=t2.ein where t1.year = 2014 and t2.year=2013 order by (t1.revenue- t2.revenue) limit 5;
- -> Please note, without proper indices (week 11) this query will take about an 5 min to run.
- select t1.ein as ein, t1.revenue as 2014rev,t2.revenue as 2013rev, (t1.revenue- t2.revenue) as diff from taxdata t1 join taxdata t2 on t1.ein=t2.ein where t1.year = 2014 and t2.year=2013 order by (t1.revenue- t2.revenue) limit 5 desc;

Another example

- Give me a list of the department number, name, and manager.

Date Query required

- Lets ignore that for now.

Query

- `select * from dept_manager m join departments d on d.dept_no = m.dept_no;`
- All the data

Fix the date issue

- We want the **current** dept manager
- `select * from dept_manager m join departments d on d.dept_no = m.dept_no where to_date > now() and from_date < now();`

Provide a list employee's and their dept manager

Debug Query

- ```
select * from dept_manager m join departments d on d.dept_no = m.dept_no join dept_emp de on de.dept_no=m.dept_no join employees e on e.emp_no = de.emp_no join employees es on es.emp_no = m.emp_no where m.to_date > now() and m.from_date < now() and de.to_date > now() and de.from_date < now() limit 1;
```
- To answer the question, we would remove the select \* and replace with field names.
- What does this do?

# Explain the query

- select \* from dept\_manager m
- JOIN departments d ON d.dept\_no = m.dept\_no JOIN dept\_emp de ON de.dept\_no=m.dept\_no JOIN employees e ON e.emp\_no = de.emp\_no JOIN employees es ON es.emp\_no = m.emp\_no
- WHERE m.to\_date > now() and m.from\_date < now() and de.to\_date > now() and de.from\_date < now()
- LIMIT 1;

## Back to diagrams

- Your database diagrams need both
  - A textual description of the data (what you have been doing now)
  - A graphical view of the data showing interconnections between tables.
  - We don't care about the format, but make sure not to use an automated tool.

| departments |           |                     |
|-------------|-----------|---------------------|
|             | dept_no   | char(4)             |
|             | dept_name | <u>varchar</u> (40) |

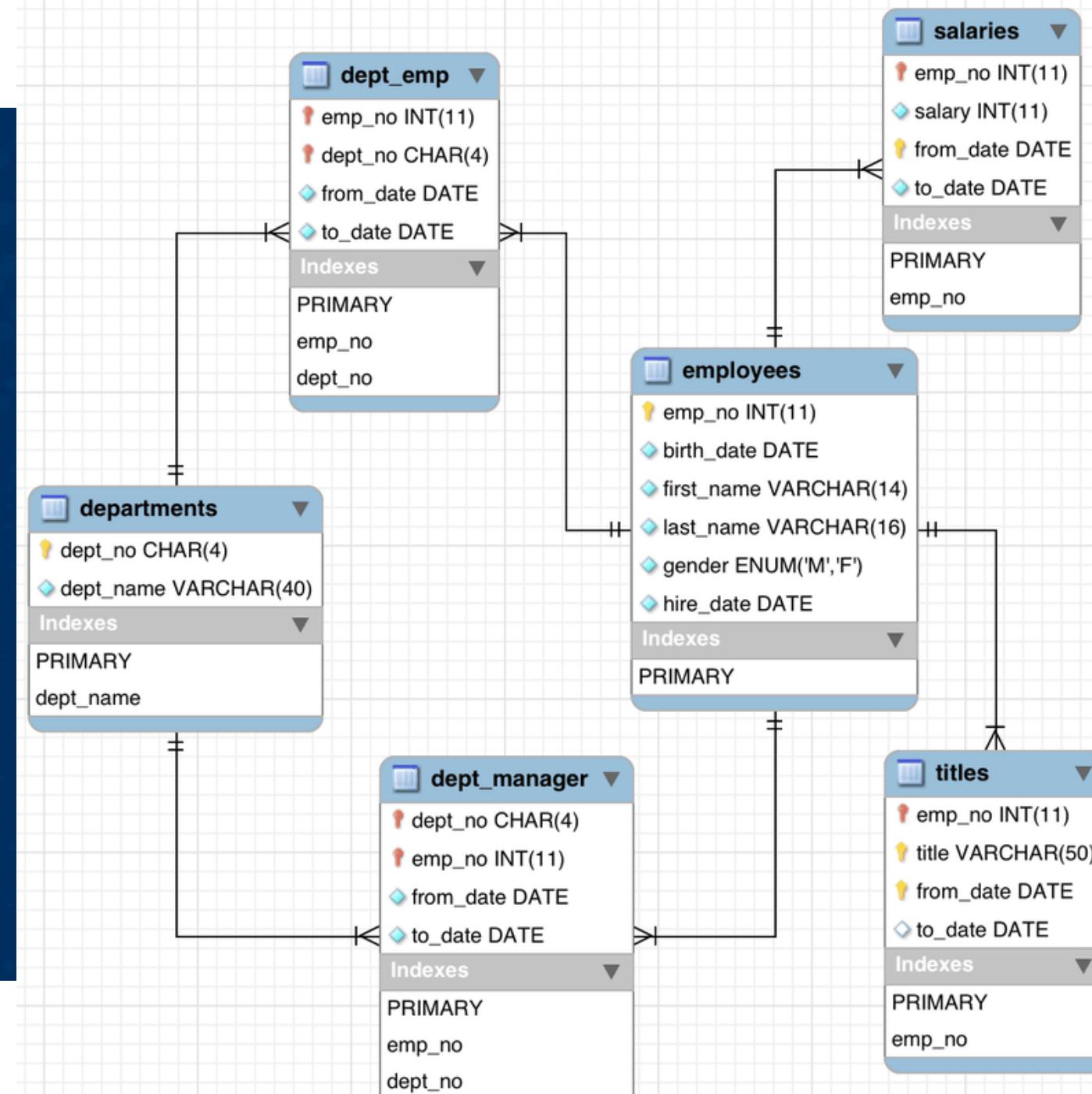
| dept_manager |               |         |
|--------------|---------------|---------|
|              | <u>emp_no</u> | int     |
|              | dept_no       | char(4) |
|              | from_date     | date    |
|              | to_date       | date    |

| dept_emp |               |         |
|----------|---------------|---------|
|          | <u>emp_no</u> | int     |
|          | dept_no       | char(4) |
|          | from_date     | date    |
|          | to_date       | date    |

| salaries |               |      |
|----------|---------------|------|
|          | <u>emp_no</u> | int  |
|          | salary        | int  |
|          | from_date     | date |
|          | to_date       | date |

| employees |               |                     |
|-----------|---------------|---------------------|
|           | <u>emp_no</u> | int                 |
|           | birth_date    | date                |
|           | first_name    | <u>varchar</u> (14) |
|           | last_name     | <u>varchar</u> (16) |
|           | gender        | <u>enum</u> (1)     |
|           | hire_date     | date                |

| titles |               |                     |
|--------|---------------|---------------------|
|        | <u>emp_no</u> | int                 |
|        | title         | <u>varchar</u> (50) |
|        | from_date     | date                |
|        | to_date       | date                |



# Stepping up to Health

SUTH

# Looking at this data.

What query would I write to know how much someone slept and in what time zone they are in?

STOP,

Don't answer the question until you complete the ERD.

| day_entity |                                 |
|------------|---------------------------------|
|            |                                 |
| •          | <b>id</b> int(10) unsigned      |
| •          | <b>user_id</b> int(10) unsigned |
| ■          | <b>created</b> int(11)          |
| ■          | <b>changed</b> int(11)          |
| ■          | <b>fitbit_date</b> date         |
| ■          | <b>fitbit_steps</b> int(11)     |
| ■          | <b>fitbit_goal</b> int(11)      |
| ■          | <b>fitbit_veryacm</b> int(11)   |
| ■          | <b>fitbit_fairacm</b> int(11)   |
| ■          | <b>fitbit_ligham</b> int(11)    |
| ■          | <b>fitbit_sedmin</b> int(11)    |

| fitbit_sleep |                                      |
|--------------|--------------------------------------|
|              |                                      |
| •            | <b>id</b> int(10) unsigned           |
| •            | <b>user_id</b> int(10) unsigned      |
| ■            | <b>name</b> varchar(50)              |
| ■            | <b>fitbit_date</b> varchar(50)       |
| ■            | <b>fitbit_duration</b> int(11)       |
| ■            | <b>fitbit_efficiency</b> int(11)     |
| ■            | <b>fitbit_timeinbed</b> int(11)      |
| ■            | <b>fitbit_ismainsleep</b> tinyint(4) |
| ■            | <b>created</b> int(11)               |

| fitbit_hr |                                 |
|-----------|---------------------------------|
|           |                                 |
| •         | <b>id</b> int(10) unsigned      |
| •         | <b>user_id</b> int(10) unsigned |
| ■         | <b>fitbit_date</b> datetime     |
| ■         | <b>fitbit_hr</b> int(11)        |
| ■         | <b>fitbit_time</b> varchar(255) |
| ■         | <b>created</b> int(11)          |

| fitbit_day_detail |                                 |
|-------------------|---------------------------------|
|                   |                                 |
| •                 | <b>id</b> int(10) unsigned      |
| •                 | <b>user_id</b> int(10) unsigned |
| ■                 | <b>fitbit_steps</b> int(11)     |
| ■                 | <b>created</b> int(11)          |
| ■                 | <b>changed</b> int(11)          |
| ■                 | <b>fitbit_date</b> datetime     |

| track_entity |                                 |
|--------------|---------------------------------|
|              |                                 |
| •            | <b>id</b> int(10) unsigned      |
| •            | <b>user_id</b> int(10) unsigned |
| ■            | <b>time</b> varchar(50)         |
| ■            | <b>playerstatus</b> varchar(50) |
| ■            | <b>video</b> varchar(50)        |

| users_field_data |                             |
|------------------|-----------------------------|
|                  |                             |
| •                | <b>uid</b> int(10) unsigned |
| •                | <b>langcode</b> varchar(12) |
| ■                | <b>name</b> varchar(60)     |
| ■                | <b>mail</b> varchar(254)    |
| ■                | <b>timezone</b> varchar(32) |

| suth_user |                                 |
|-----------|---------------------------------|
|           |                                 |
| •         | <b>id</b> int(10) unsigned      |
| •         | <b>user_id</b> int(10) unsigned |
| ■         | <b>startdate</b> varchar(50)    |
| ■         | <b>ptstatus</b> varchar(50)     |

| goal_entity |                                 |
|-------------|---------------------------------|
|             |                                 |
| •           | <b>id</b> int(10) unsigned      |
| •           | <b>user_id</b> int(10) unsigned |
| ■           | <b>goal</b> int(11)             |
| ■           | <b>date</b> varchar(50)         |

# Day Entity

Fitbit\_goal and fitbit\_date and fitbit\_steps

The goal is the users goal

The date is the date from the hardware

The steps are the number of steps walked.

User\_id is the user id of the research PT.

| day_entity     |                  |
|----------------|------------------|
| id             | int(10) unsigned |
| user_id        | int(10) unsigned |
| created        | int(11)          |
| changed        | int(11)          |
| fitbit_date    | date             |
| fitbit_steps   | int(11)          |
| fitbit_goal    | int(11)          |
| fitbit_veryacm | int(11)          |
| fitbit_fairacm | int(11)          |
| fitbit_ligham  | int(11)          |
| fitbit_sedmin  | int(11)          |

# User Field Data

| users_field_data |                  |
|------------------|------------------|
| uid              | int(10) unsigned |
| langcode         | varchar(12)      |
| name             | varchar(60)      |
| mail             | varchar(254)     |
| timezone         | varchar(32)      |

- Uid is the users study id.
- This is the same as user\_id in other tables.

# Suth\_user

Study data table

| suth_user |                                 |
|-----------|---------------------------------|
| •         | <b>id</b> int(10) unsigned      |
| •         | <b>user_id</b> int(10) unsigned |
| •         | <b>startdate</b> varchar(50)    |
| •         | <b>ptstatus</b> varchar(50)     |

- Ptstatus is the status of the pt in the study.

# fitbit\_hr

Heart rate data

- For every min, fitbit\_hr is the heart rate of the user.

| fitbit_hr   |                  |
|-------------|------------------|
| id          | int(10) unsigned |
| user_id     | int(10) unsigned |
| fitbit_date | datetime         |
| fitbit_hr   | int(11)          |
| fitbit_time | varchar(255)     |
| created     | int(11)          |

# More than one join

This is rare

- Badly designed tables might require you to have more than one join rule
  - Select \* from tablea a join tableb b on (b.id1 = a.id1 and b.id2=a.id2)
  - You can also write
    - Select \* from tablea a join tableb b on b.id1 = a.id1 where b.id2=a.id2

Information  
changes  
everything.

# THANK YOU



# Welcome to SI564

SQL and Databases



Will try to answer questions sent via DM's. Please DM all teaching team members.

# Left/Right Join

# Right Join

# Left Join

- A,<sub>1</sub>
- B,<sub>2</sub>
- C,<sub>3</sub>
- Z,<sub>9</sub>
- A,<sub>7</sub>
- B,<sub>8</sub>
- C,<sub>9</sub>
- D,<sub>2</sub>
- E,<sub>1</sub>
- A,<sub>1,7</sub>
- B,<sub>2,8</sub>
- C,<sub>3,9</sub>
- Z,<sub>9,null</sub>

# Left join

- All records from one, only matching records from another
- ALL students and their classes .
- All Employees and their satisfaction scores

# Inner Join

# Inner Join

- A,<sub>1</sub>
  - B,<sub>2</sub>
  - C,<sub>3</sub>
  - Z,<sub>9</sub>
  - A,<sub>6</sub>
  - B,<sub>7</sub>
  - C,<sub>8</sub>
  - D,<sub>4</sub>
  - E,<sub>5</sub>
- A,<sub>1,6</sub>
  - B,<sub>2,7</sub>
  - C,<sub>3,8</sub>

## Inner Join

- Records that match on both sides.
- Give me students and their classes
- What office people are located in
- Classroom and the classes assigned to each

# Questions?

# Why use Python?

This was optional

# Homework that was not done.

- `con = pymysql.connect(host='1.2.3.4', port=NUM,user='user',password='pass',database='ro_tweet',cursorclass=pymysql.cursors.DictCursor)`
- `cur = con.cursor()` < get cursor object.
- `cur.execute("select * from tweets ORDER BY RAND() limit 5")` < QUERY must be in quotes. It is a string.

- for row in cur: < loop over all the returned results.
- print((row['id']))
- row['tweet'] has a STRING with all the data in it. NOT a dictionary, not a json object. A string.
- print(type(row['tweet'])) <- proof
- Convert to json
- retweet\_count is sometimes null;  
tc = rowjson['retweet\_count'] will fail.
- How to fix?
  - Try catch
  - Check if *None* with .get method.

# Check your types

- A result set from a DB can be a dictionary. (row is a dictionary)
- Data inside a result set (like a column) is almost always a string.
- In this case it was a string of json.
- Printing it looks like a dictionary.
- However, it is a string.
- Check types with `print(type(var))`

# Dictionary

- `dict['value']` will work, but fails on null keys
- `dict.get('value')` will return `None` for null keys
- Try/catch is another way to handle this.
  - But don't do this.

You can do this  
in native  
MySQL

However, you have to be  
able to parse json strings.

- select
- JSON\_EXTRACT(tweet,"\$.text"),
- JSON\_EXTRACT(tweet,"\$.retweet\_count")
- from tweets;

# Derived tables

Beef and Garlic

- Select id from (select \* from table) exp
- This creates a derived table called exp, which is the content of select \* from table.

# Office Hours

- If you would like to come to office hours, please book online.
- [Umsi.in/564office](http://Umsi.in/564office)

# Asking questions in slack

- When using a DM
- Be detailed with what you need.
- Copy and paste the homework problem you are working on
- What exact steps have you taken? What have they produced?  
What do you think is wrong?
- For some issues, we may request you book office hours (or ask someone else on the teaching team)



Welcome!

# AveryX

- AveryX has joined our company.

# Mid Term

- It will cover everything we have covered so far in class.
- Including next week's lecture on group by.

# Midterm Logistics

- You MUST sign up for a session for the midterm.
- You will join a zoom call.
- You will share your screen and video and record both using zoom to your local computer.
- We will not look at the recording unless there is an issue, but I will validate that it is there and you shared your screen and camera. It will be part of your grade.
- **Please test screen sharing in zoom before the exam.**
- The entire midterm will be given on a single day. Slots will start at 8:30am and go till 7pm.
- If you don't see a slot that works for you, let us know.

## Mid Term rules

- Use of Canvas is allowed.
- Internet (stack overflow/ google/etc) is not allowed.

# Mid Term

- You will be asked to query a database like you do now for homework.
- This will cover *order by*, *group by*, *where*, and *table joins*.
- Grading will be somewhat like homework.
- I will provide the diagram

## Midterm help

- 1) No required documentation.
- 2) You will be asked to answer questions and format them into a report. We are less concerned about the formatting of the report and more concerned about the questions being answered.
- 3) This will require some common sense to provide direction for yourself. We will ask questions, but not tell you where the information is stored.
- 4) New database will be provided.

# Mid Term grading

- You are graded on the number of queries that it takes you to solve a problem.
- For example, queries that require a join, could be written as 2 queries rather than 1. I have said in the assignments to not put numbers in your queries.
- As long as you are writing full queries that do the joins and the group by's, this won't impact your grade.
- You can issue all the queries you want to get the job done. We are grading on the queries you turn in as answers.

# Midterm

- No python on exam.

# Database usage

- Use of your database after 5pm Thursday will result in a zero on your midterm.

## Review sessions

- The class before the midterm will be a review session, bring questions.

# Midterm Hours

- We will start midterms at 8:30am.
- We will offer 4-5 slots for people spaced in the day. (8:30am, 10:30am, etc)
- You will sign up online for a slot.
- You will join a zoom.

## Zoom invite

- When you sign up for your midterm, you will get a calendar invite. **At least 24 hours before the midterm, this invite will get updated with your zoom link.**
- If you don't see the zoom link within 24 hours of your midterm time, message the entire teaching team.

# Midterm

- Help
- If you need help on the midterm there are 2 ways of getting it
- If your question is a policy question, you can use slack and message the ENTIRE teaching team.
- If your question is a technical question, you can join the “office hours queue”

# Office Hours Queue

We will not be meeting in person for assistance, a teaching team member will join your zoom call.

## Welcome to the Midterm Exam Help meeting queue.

You're up next, but the host isn't quite ready for you. Pay attention to this page -- we will tell you when the host is ready. Make sure you are nearby the in-person meeting location.

You are currently in line.

Your Number in Line: 1

Did you know? You can receive an SMS (text) message when it's your turn by adding your cell phone number and enabling attendee notifications in your [User Preferences](#).

**Time Joined:** Wednesday, September 21, 2022 at 7:40:29 PM

**Meeting Via:** In Person (A Host has been assigned to this meeting. Meeting Type can no longer be changed.)

**Meet At:** We will join your exam zoom.

**Meeting Agenda (Optional):**

Let the host(s) know the topic you wish to discuss.

**Update**

[Leave the Line](#)

# Attendance points

- Given that we are using this tool for help, it is important that everyone know how to use it.
- Please join the mid term assistance queue.
- Do not leave the queue, we will measure attendance with it.
- If you are in 564 and 504, you should only join once.

# Attendance points

- <https://officehours.it.umich.edu/queue/1242>
- Make the “Topic” Attendance

# Mid Term

- No class on the midterm date

# 24 hours and day of midterm

- You will get a zoom link added to your calendar 24 hours BEFORE the midterm starts.

# Midterm Questions

## After midterm

- After the midterm we will be looking at building databases.
- We are 1 week behind, so I will be reducing the complexity of the final project.
- **We will be adding a relaxation event before a class.**
- **This will be a field trip in person only.**
- **This will be 100% optional but I promise it to be a very unique experience.**
- **You will choose your time slot.**

# Bulgan Jugderkhuu



- Pronounced similarly to "Logan"
- ~~2nd year MSI student on the Data track~~
  - Data Privacy Analyst @ Lyft
- Mongolian
- From Chicago suburbs
- Love sushi, hiking, music, movies
- Ask me anything about Ann Arbor and recommendations, the MSI program, or anything else you want to chat about :)
- **She/her/hers**

# Review Today

And some new concepts

# New Functions

- We will learn about these again in different detail next week.

# Aggregation

- Next week we will cover group by
- Today we are going to start with some Aggregation functions

# Avg(X)

- In a table with 100 rows
- Select avg(val) from table;
- Will return the mean of val for each row.

# Count(x)

- Almost always use `count(1)`

# Sum(x)

- In a table with 100 rows
- Select sum(val) from table;
- Will return the total of val for each row.

No Group by  
this week

# Live Demo

- Today is a live demo
- Live coding is live coding, it is not “prepped”

# Today

- Please think of today as a discussion.
- We will start with documentation.
  - Understanding business
  - Document table layout
  - Think about what might cause headaches.
- **This is an online store.**
  - What do you think about first
- **Your homework is based on the classic models dataset**

Information  
changes  
everything.

# THANK YOU



# Welcome to SI564

SQL and Databases



Welcome!

# Mid Term

- It will cover everything we have covered so far in class.

# Midterm Review

- Accept or deny the calendar invite.

# Midterm Logistics

- You MUST sign up for a session for the midterm.
  - DO THIS NOW!
- You will join a zoom call.
- You will share your screen and video and record both using zoom to your local computer.
- We will not look at the recording unless there is an issue, but I will validate that it is there and you shared your screen and camera. It will be part of your grade.
- **Please test screen sharing in zoom before the exam.**
- The entire midterm will be given on a single day. Slots will start at 8:30am and go till 7pm.
- If you don't see a slot that works for you, let us know.

## Mid Term rules

- Use of Canvas is allowed.
- Internet (stack overflow/ google/etc) is not allowed.

# Mid Term

- You will be asked to query a database like you do now for homework.
- This will cover *order by*, *group by*, *where*, and *table joins*.
- Grading will be somewhat like homework.
- I will provide the diagram

# Midterm help

- 1) No required documentation.
- 2) You will be asked to answer questions and format them into a report. We are less concerned about the formatting of the report and more concerned about the questions being answered.
- 3) This will require some common sense to provide direction for yourself. We will ask questions, but not tell you where the information is stored.
- 4) New database will be provided.

# Mid Term grading

- You are graded on the number of queries that it takes you to solve a problem.
- For example, queries that require a join, could be written as 2 queries rather than 1. I have said in the assignments to not put numbers in your queries.
- As long as you are writing full queries that do the joins and the group by's, this won't impact your grade.
- You can issue all the queries you want to get the job done. We are grading on the queries you turn in as answers.

# Midterm

- No python on exam.

# Database usage

- Do not use your database from Thursday to Monday (the day of the midterm)

## Review sessions

- Accept the calendar invite!
- Bring your questions!

# Midterm Hours

- We will start midterms at 8:30am.
- We will offer 4-5 slots for people spaced in the day. (8:30am, 10:30am, etc)
- You will sign up online for a slot.
- You will join a zoom.

## Zoom invite

- When you sign up for your midterm, you will get a calendar invite. At least 24 hours before the midterm, this invite will get updated with your zoom link.
- If you don't see the zoom link within 24 hours of your midterm time, message the entire teaching team.

# Midterm

- Help
- If you need help on the midterm there are 2 ways of getting it
- If your question is a policy question, you can use slack and message the ENTIRE teaching team.
- If your question is a technical question, you can join the “office hours queue”

# Office Hours Queue

We will not be meeting in person for assistance, a teaching team member will join your zoom call.

## Welcome to the Midterm Exam Help meeting queue.

You're up next, but the host isn't quite ready for you. Pay attention to this page -- we will tell you when the host is ready. Make sure you are nearby the in-person meeting location.

You are currently in line.

Your Number in Line: 1

Did you know? You can receive an SMS (text) message when it's your turn by adding your cell phone number and enabling attendee notifications in your [User Preferences](#).

**Time Joined:** Wednesday, September 21, 2022 at 7:40:29 PM

**Meeting Via:** In Person (A Host has been assigned to this meeting. Meeting Type can no longer be changed.)

**Meet At:** We will join your exam zoom.

**Meeting Agenda (Optional):**

Let the host(s) know the topic you wish to discuss.

**Update**

[Leave the Line](#)

# Mid Term

- No class on the midterm date

# Midterm Questions

## After midterm

- After the midterm we will be looking at building databases.
- We are 2 weeks behind, so I will be reducing the complexity of the final project.
- **We will be adding a relaxation event before a class.**
- **This will be a field trip in person only.**
- **This will be 100% optional but I promise it to be a very unique experience.**
- **You will choose your time slot.**

## Note on this weeks homework

- If your results don't make sense, write the simplest query to debug. If you are asking for help, please let us know the debug query you have written.

# Aggregation

- What is the average population BY Continent?
- What is the total number of countries BY each type of government?
- What continent has the largest surface area?

# Country table

```
• mysql> desc country;
• +-----+-----+-----+-----+
• | Field | Type | Null | Key | Default | Extra |
• +-----+-----+-----+-----+
• | Code | char(3) | NO | PRI | |
• | Name | char(52) | NO | | |
• | Continent | enum('Asia','Europe','North America','Africa','Oceania','Antarctica','South America') | NO | | Asia |
• | Region | char(26) | NO | | |
• | SurfaceArea | decimal(10,2) | NO | | 0.00 |
• | IndepYear | smallint(6) | YES | NULL | |
• | Population | int(11) | NO | | 0 |
• | LifeExpectancy | decimal(3,1) | YES | NULL | |
• | GNP | decimal(10,2) | YES | NULL | |
• | GNPOld | decimal(10,2) | YES | NULL | |
• | LocalName | char(45) | NO | | |
• | GovernmentForm | char(45) | NO | | |
• | HeadOfState | char(60) | YES | NULL | |
• | Capital | int(11) | YES | NULL | |
• | Code2 | char(2) | NO | | |
• +-----+-----+-----+-----+
• 15 rows in set (0.03 sec)
```



By hand?

- Sort the table
- Select Continent , Population from Country order by Continent;
- Add each row, start over when Continent changes.

# Group By

- SQL to the rescue
- New Functions today and review of some today!

# Aggregation Functions

- count(<what>)
- sum(<what>)
- Average(<what>) // Mean
- Maximum(<what>)
- Minimum(<what>)

# Group by base

- Loop and collapse on the group by element.

- SELECT field from table group by field;
  
- What will this do?
- The same as distinct.
- WHY?
- For each row we will select field and we will collapse this by the field.

# GROUP BY

- We want to SUMMARIZE by the fields after GROUP BY
- We can summarize by more than one field.

# Group by

- Select group\_by\_field, agg\_func from table (JOINS) GROUP by group\_by\_field;
- 1) sort by group\_by\_field
- 2) run agg function
- 3) When group\_by\_field changes, reset agg total

select state,  
count(state)  
group by state

| state    | count |
|----------|-------|
| Ohio     | 3     |
| Ohio     | 2     |
| Ohio     | 1     |
| Michigan | 2     |
| Michigan | 1     |

- Ohio,3
- Michigan,2

select state,  
sum(company)  
group by state

State,company

Ohio,3

Ohio,3

Ohio,1

Michigan,10

Michigan,20

- Ohio,7
- Michigan,30

# What is the average population BY Continent?

- mysql> select Continent, avg(Population) from country;
- ERROR 1140 (42000): In aggregated query without GROUP BY, expression #1 of SELECT list contains nonaggregated column 'world.country.Continent' this is incompatible with sql\_mode=only\_full\_group\_by
- We are trying to use an aggregation function without a group by clause. This is not really valid. (count is an exception)
- Running SET sql\_mode=(SELECT REPLACE(@@sql\_mode, 'ONLY\_FULL\_GROUP\_BY', ''));
- Can shut that off, but don't unless you have a good reason to do it. Normally, this indicates you have something wrong with your query.

# What is the average population BY Continent?

How can we make this better?

- mysql> select Continent, avg(Population) from country group by Continent;
- +-----+-----+
- |Continent | avg(Population)|
- +-----+-----+
- | North America | 13053864.8649 |
- | Asia | 72647562.7451 |
- | Africa | 13525431.0345 |
- | Europe | 15871186.9565 |
- | South America | 24698571.4286 |
- | Oceania | 1085755.3571 |
- | Antarctica | 0.0000 |
- +-----+-----+

# What is the average population BY Continent?

Better!

- mysql> select Continent, round(avg(Population)) as pop from country group by Continent;

# What is the total number of countries BY each type of government?

How do we make better

- mysql> select count(Code), GovernmentForm from country group by GovernmentForm;
- +-----+-----+
- | count(Code) | GovernmentForm |
- +-----+-----+
- | 2 | Nonmetropolitan Territory of The Netherlands |
- | 1 | Islamic Emirate |
- | 122 | Republic |
- | 12 | Dependent Territory of the UK |
- | 1 | Parliamentary Coprincipality |
- | 1 | Emirate Federation |
- | 15 | Federal Republic |
- | 3 | US Territory |
- | 1 | Co-administrated |
- | 4 | Nonmetropolitan Territory of France |
- | 29 | Constitutional Monarchy |

# Cleaner

- select count(Code) **as c**, GovernmentForm from country group by GovernmentForm order **by c**;
- We can order by the result of the count. Notice I use the field alias.

# What continent has the largest surface area?

However, this does not answer my question? Why not?

- mysql> select Continent, sum(SurfaceArea) from country group by Continent;
- +-----+-----+
- | Continent | sum(SurfaceArea) |
- +-----+-----+
- | North America | 24214470.00 |
- | Asia | 31881005.00 |
- | Africa | 30250377.00 |
- | Europe | 23049133.90 |
- | South America | 17864926.00 |
- | Oceania | 8564294.00 |
- | Antarctica | 13132101.00 |
- +-----+-----+

# ANSWER THE QUESTION ASKED

SA is not a good alias? I  
don't know what it means

- mysql> select Continent, sum(SurfaceArea) as sa from country group by Continent order by sa desc limit 1;
- +-----+-----+
- | Continent | sa |
- +-----+-----+
- | Asia | 31881005.00 |
- +-----+-----+
- 1 row in set (0.02 sec)

DB

- Sakila

# Query

- What is the total that each customer has paid?
- What tables do we need?

# Query

- What is the average number of rentals per customer?
- WHAT DOES AVERAGE MEAN HERE?

# Query

- What item gets rented the most?
- What items have been rented more than 5 times.\*\*\*

# More than one group by

- Show me the payments by staff and date?
- select month(payment\_date),staff\_id, sum(amount) from payment group by month(payment\_date), staff\_id;

# More than one group by

- Ro\_employees;
- What is the gender breakdown by year?
- select count(gender), gender, year(hire\_date) from employees  
group by year(hire\_date), gender;

# Filter on the group by

- Select count(id) from table where count(id) > 10; <- Works
  - No group by
- select count(email) from users where count(email) > 1;
  - Did anyone sign up with a duplicate email?
- This will not work!

The where  
clause is RUN  
BEFORE group  
by

The query's steps don't happen in the order they're written:

ATION

how the query  
is written

how you should  
think about it

SELECT ...

FROM + JOIN

WHERE ...

GROUP BY ...

HAVING ...

ORDER BY ...

LIMIT ...

FROM + JOIN

WHERE

GROUP BY

HAVING

SELECT

ORDER BY

LIMIT

(In reality query execution is much more complicated than this.  
There are a lot of optimizations.)

# Having

- HAVING is a way to filter AFTER group by.
- select count(email) from table HAVING count(email) > 1;

# Having VS where

Reminder

- Having is used to filter AFTER GROUP by. You normally use it on an aggregation function. Note that the aggregation function does not need to be in the SELECT part.

# WHERE

- If you want to remove items BEFORE aggregation, use the WHERE clause.
- `SELECT count(email) from users where admin =0 group by email HAVING count(email) > 1;`

# More than one group by

- Show me the top rentals by store.
- Lots of joins.

# Stuck

- Write the full query first to return all the data
- Check the data
- Then modify to add in aggregation

# End of new topics in SQL

Select queries

- We will start creating our own tables.

# Home work complex this week

- If you are working on homework over spring break (please don't but) but if you do....please only message myself, not the teaching team.

# Reminders

- Sign up for exam.
- Study sessions

Information  
changes  
everything.

# THANK YOU

