SQL for KB

December 17

/Chaomander2018
5 Hours of SQL

Overall

- Database
- **Query language (data manipulation)**
- Importing csv to database
- Creating database
- Learning from the internet (resources)

Rules

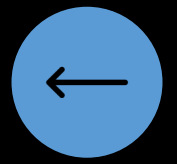
- Judgement free zone
- Participation
- Stop anytime
- 30 min- 5 min- 30 min

Agenda for the 1st hour

DS flow & SQL languages(3 mins)

Set up SQLite Studio and drink db (5mins)

Schema



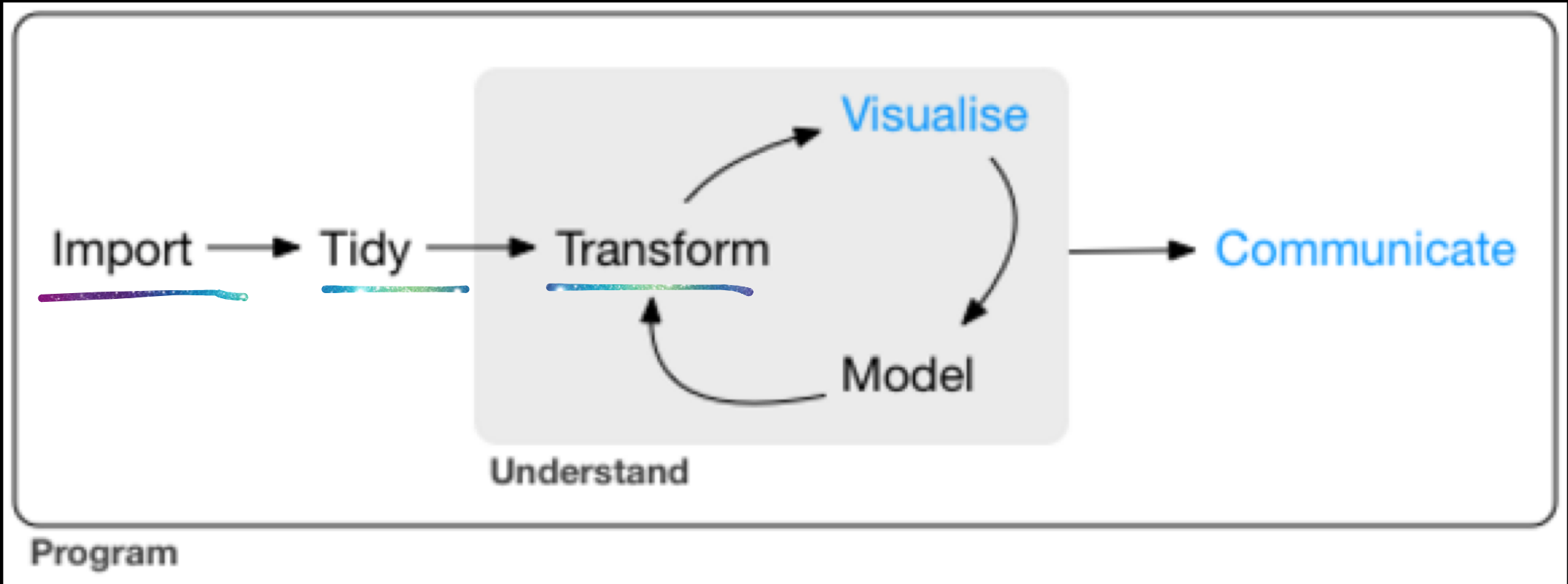
Keys (10min)

Command/statements (30 mins)

Data Type (5 mins)

Use GWM data to practice writing a schema (10 min)

Wrap up (2mins)



- https://www.youtube.com/watch?time_continue=112&v=uCNOtUht2Xc

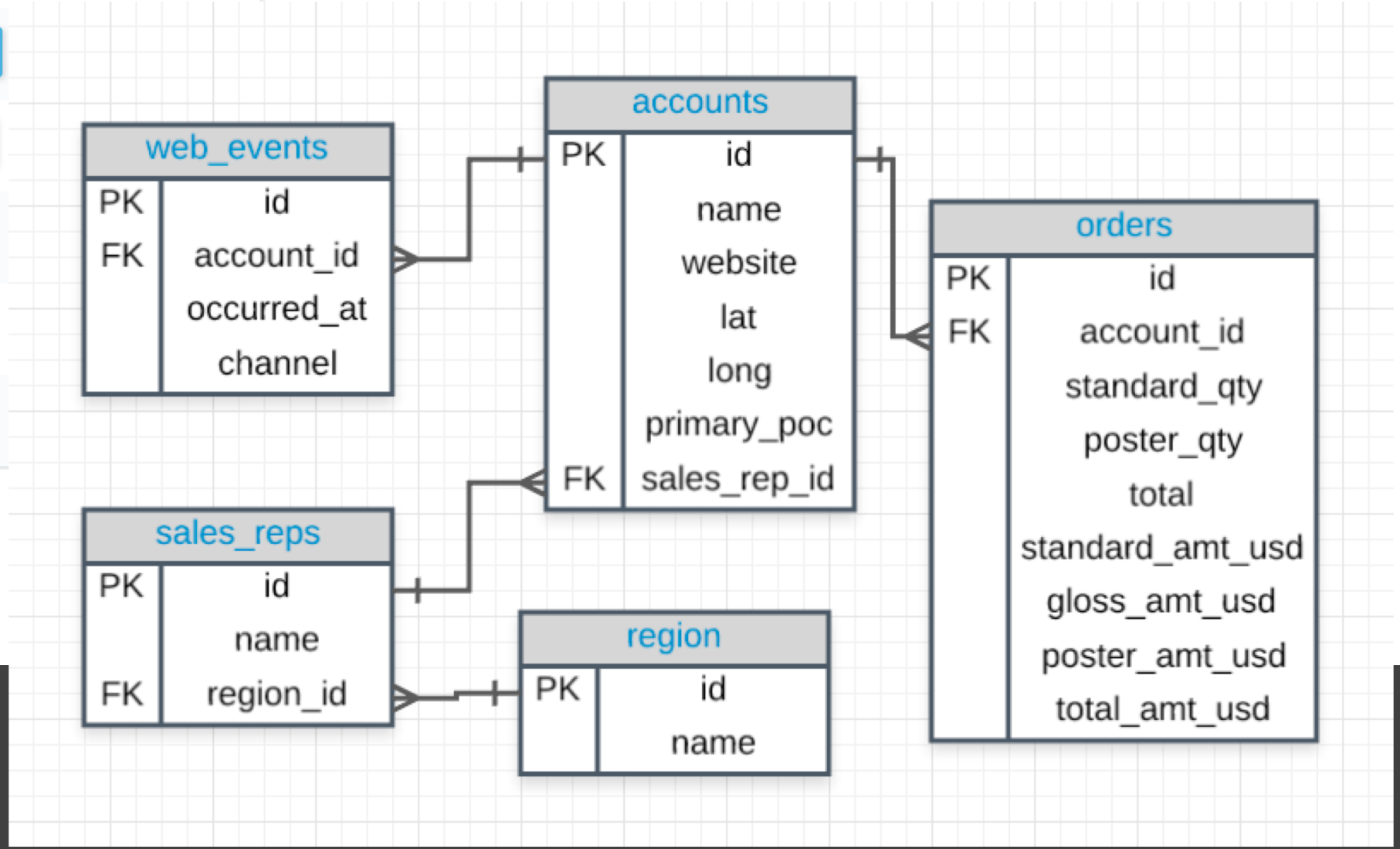
Database and SQL

- A database is a collection of information that is organized so that it can be easily accessed, managed and updated.

- Relational database (other databases)

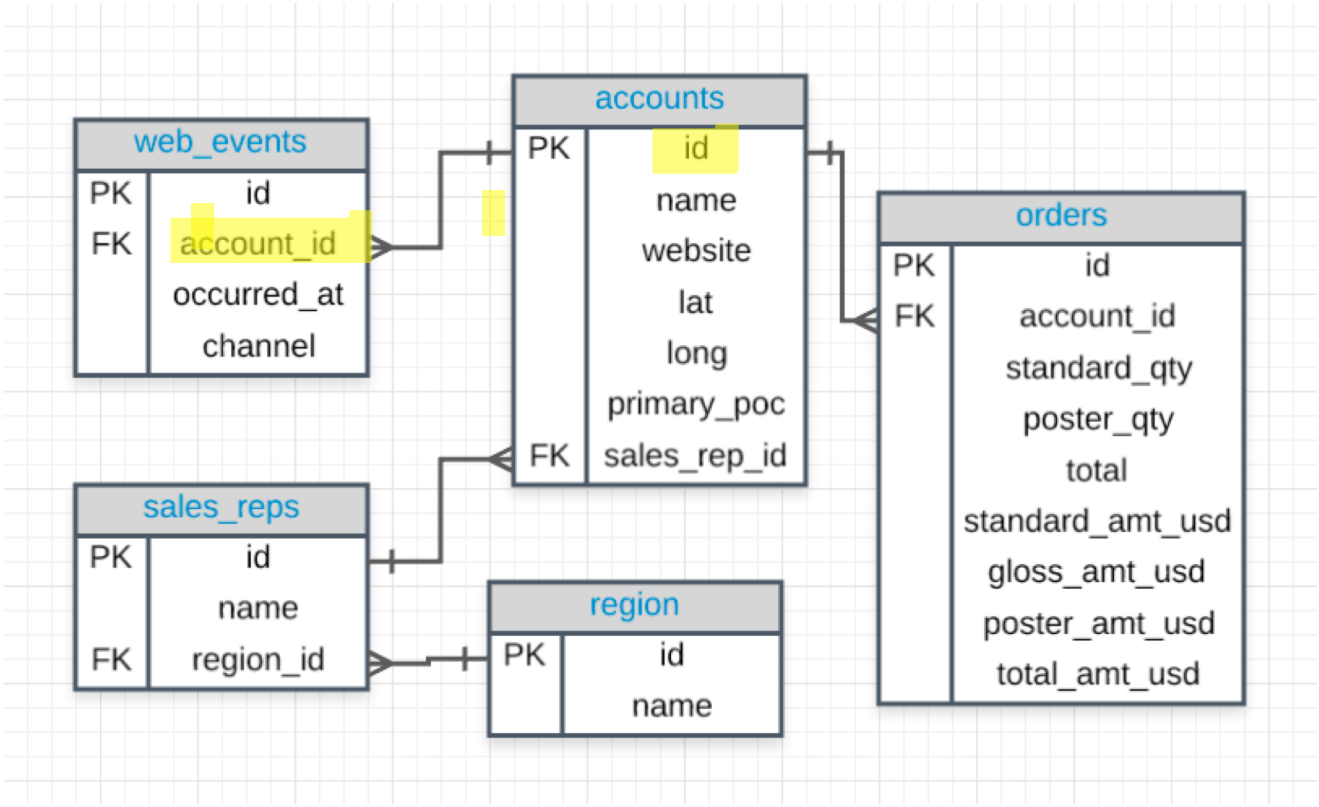
https://searchsqlserver.techtarget.com/definition/database

DEFINITION	TERM
A column name in the Parch & Posey database.	primary_poc
A table name in the Parch & Posey database.	web_events
A collection of tables that share connected data stored in a computer.	Database
A diagram that shows how data is structured in a database.	ERD
A language that allows us to access data stored in a database.	SQL

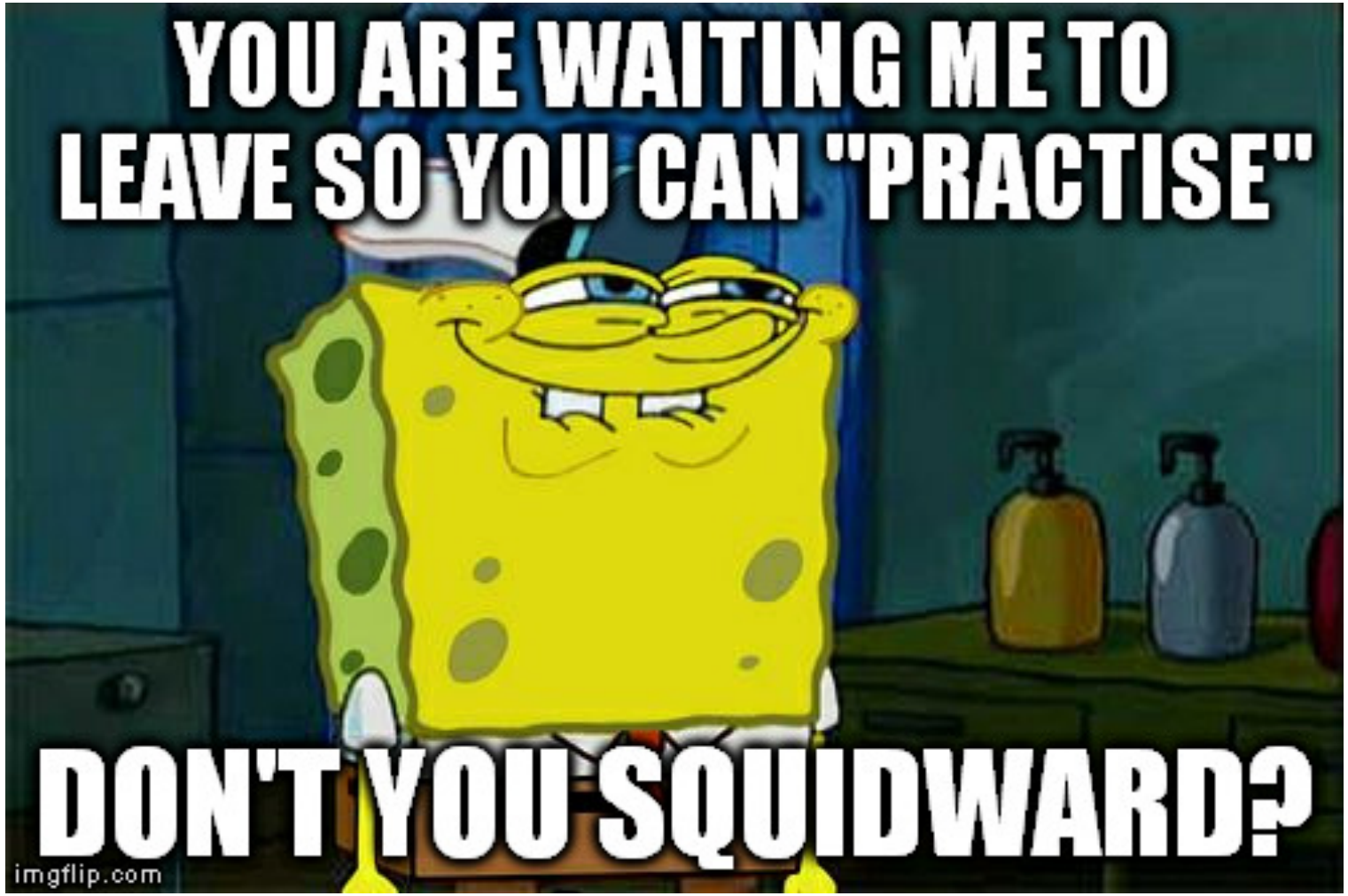


Keys & Entity Relationship Diagrams

- Relation as tables or sheets in excel
- Primary key (Unique Identifier, not NULL)
- Foreign key
- Understand the db
- Write out drink db on board
- Super key(affiliate name+ AM)



Drink db practice



Data type

SQL Domain Type	SQLite	Description
CHAR(N)	**TEXT**	Fixed length character string with length n
VARCHAR(N)		Variable length character strings, with max length n
INT	**INTEGER**	Integer (of some default length, machine-dependent)
BIGINT, SMALLINT, TINYINT		Integer of longer/shorter lengths
NUMERIC(P,D)	**NUMERIC**	A fixed-point number with defined precision of *p* digits, with *d* digits to the right of the decimal point
FLOAT(N), REAL	**REAL**	Floating point number, with defined precision of at least *n* digits
DATE	**TEXT, REAL, or INTEGER**	A calendar date (eg. '2018-07-27')
TIME		Time (eg. '09:00:30')
TIMESTAMP		Includes both date and time (eg. '2018-07-27 09:00:30.75')

http://www.cs.toronto.edu/~nn/csc309-20085/guide/pointbase/docs/html/htmlfiles/dev_datatypesandconversionsFIN.html

Statement/ Commands

- Capital letters
- SELECT: read& display
- FROM: where the columns are located
- WHERE :conditions
- ORDER BY
- GROUP BY
- HAVING: conditions
- LIMIT

Find all the infomation about wines in drinkInfo

```
SELECT *
FROM drinkInfo
WHERE type = "wine";
```

	drinkId	type
1	drink 1	cocktail
2	drink 2	wine
3	drink 3	rum
4	drink 4	cocktail
5	drink 5	cocktail
6	drink 6	wine
7	drink 7	cocktail
8	drink 8	whisky
9	drink 9	rum
10	drink 10	wine
11	drink 11	wine
12	drink 12	vodka
13	drink 13	whisky
14	drink 14	soda
15	drink 15	vodka
16	drink 16	wine

	drinkId	type
1	drink 2	wine
2	drink 6	wine
3	drink 10	wine
4	drink 11	wine
5	drink 16	wine
6	drink 17	wine
7	drink 30	wine
8	drink 37	wine
9	drink 48	wine

Find all the drinkIds of beer

```
SELECT drinkId
FROM drinkInfo
WHERE type = “beer”;
```

LIKE, IN, NOT IN, !=, =

	drinkId	type
1	drink 1	cocktail
2	drink 2	wine
3	drink 3	rum
4	drink 4	cocktail
5	drink 5	cocktail
6	drink 6	wine
7	drink 7	cocktail
8	drink 8	whisky
9	drink 9	rum
10	drink 10	wine
11	drink 11	wine
12	drink 12	vodka
13	drink 13	whisky
14	drink 14	soda
15	drink 15	vodka
16	drink 16	wine

	drinkId
1	drink 2
2	drink 6
3	drink 10
4	drink 11
5	drink 16
6	drink 17
7	drink 30
8	drink 37
9	drink 48

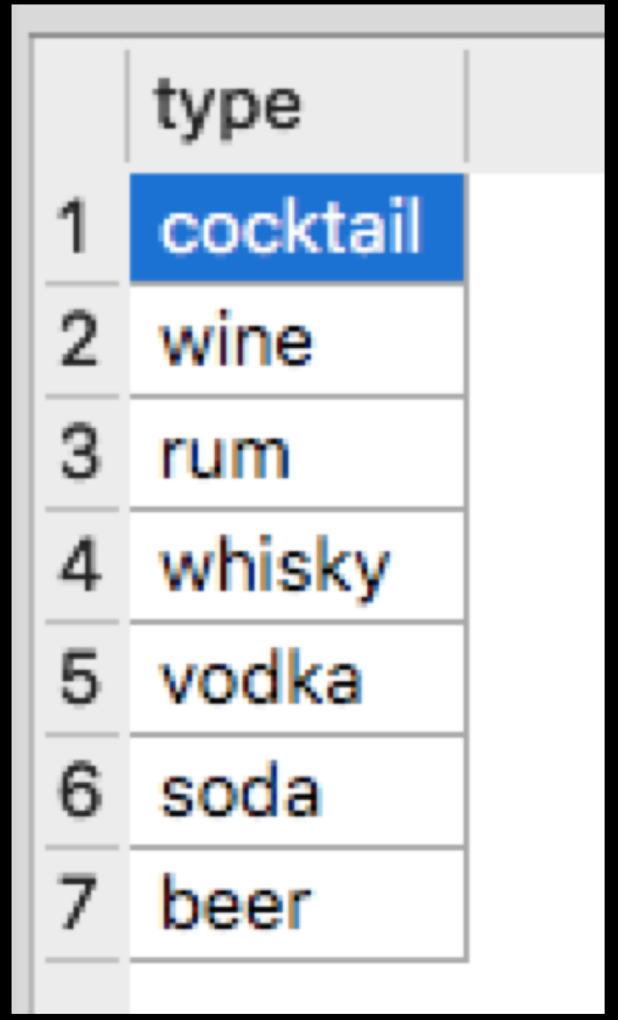
Find what types of drinks are on the menu
(Distinct)

SELECT …

FROM …;

SELECT DISTINCT…

FROM…;



	type
1	cocktail
2	wine
3	rum
4	whisky
5	vodka
6	soda
7	beer

Find all the orders from person 2

	person	date	bar	drinkId	quantity
1	person 2	2016-10-29	bar 11	drink 38	1
2	person 2	2016-10-29	bar 11	drink 4	19
3	person 2	2016-10-29	bar 11	drink 42	1
4	person 2	2016-10-24	bar 13	drink 10	4
5	person 2	2016-10-24	bar 13	drink 32	5
6	person 2	2016-10-24	bar 13	drink 37	3
7	person 2	2016-10-05	bar 17	drink 2	4
8	person 2	2016-10-05	bar 17	drink 23	2
9	person 2	2016-10-21	bar 19	drink 1	2
10	person 2	2016-10-21	bar 19	drink 42	1
11	person 2	2016-10-21	bar 19	drink 9	2
12	person 2	2016-10-26	bar 19	drink 42	1
13	person 2	2016-10-26	bar 19	drink 9	1
14	person 2	2016-10-06	bar 5	drink 25	10

Find All orders from person 2, arrange by date

The **ORDER BY** statement always comes in a query after the **SELECT** and **FROM** statements, but before the **LIMIT** statement. If you are using the **LIMIT** statement, it will always appear last. As you learn additional commands, the order of these statements will matter more.

DESC/ ASC

ORDER BY multiple values

	person	date	bar	drinkId	quantity
1	person 2	2016-10-05	bar 17	drink 2	4
2	person 2	2016-10-05	bar 17	drink 23	2
3	person 2	2016-10-06	bar 5	drink 25	10
4	person 2	2016-10-06	bar 5	drink 38	5
5	person 2	2016-10-21	bar 19	drink 1	2
6	person 2	2016-10-21	bar 19	drink 42	1
7	person 2	2016-10-21	bar 19	drink 9	2
8	person 2	2016-10-24	bar 13	drink 10	4
9	person 2	2016-10-24	bar 13	drink 32	5
10	person 2	2016-10-24	bar 13	drink 37	3
11	person 2	2016-10-26	bar 19	drink 42	1
12	person 2	2016-10-26	bar 19	drink 9	1
13	person 2	2016-10-29	bar 11	drink 38	1
14	person 2	2016-10-29	bar 11	drink 4	19

Aggregations

- MIN()
- MAX()
- AVG()
- SUM()
- COUNT()

- Can you think of the difference between SUM() and COUNT()?

Find the cheapest drink

	drinkId	the_cheapest_drink
1	drink 43	3.4778854849282 7

LIMIT

- Only display the first a couple of results.
- Too long to return

- Find the first 5 rows of `HasOnMenu`

	bar	drinkId	price
1	bar 1	drink 40	60.82020287984051
2	bar 1	drink 4	17.11281040217727
3	bar 1	drink 6	313.7675603996031
4	bar 1	drink 44	65.82548062549904
5	bar 1	drink 47	61.2035750946961 3

GROUP BY

- Find the average price by bar, order it from high to low.

	bar	AVG(price)
1	bar 12	152.0710318342317
2	bar 13	150.56043863608502
3	bar 3	129.8095159989316
4	bar 16	121.36436782585224
5	bar 17	107.32970053202007
6	bar 18	105.21477991195
7	bar 5	100.6038574514
8	bar 14	93.4388559625484
9	bar 15	90.99379045958631
10	bar 20	86.51315756270196
11	bar 1	83.6785209467 3019
12	bar 8	67.91904428955168
13	bar 7	61.56748556054663
14	bar 10	60.57263578820442

Find how many drinks do each person ordered (alias)

Aliases for Columns in Resulting Table
Aliases for table name
Lower case and underscore

	person	total_q
1	person 1	13
2	person 10	68
3	person 100	18
4	person 11	26
5	person 12	61
6	person 13	129
7	person 14	15
8	person 15	61
9	person 16	54
10	person 17	25
11	person 18	30
12	person 19	45

GWM idea

- Think about two relations you have and what are the primary keys and what are the foreign keys.
- Identify some of the column names, lets try write some queries using SELECT, FROM, WHERE, GROUP BY and ORDER BY

WRAP UP

- Database
- Keys, ER Diagrams
- Data types
- The Most Important SQL statements ()
- Aliases
- Logical Operators
- Aggregations functions
- GWM ideas

Agenda for 2-3 hours

Arithmetic Operators

More statement

DATE

Schema

CREATE, DROP, UPDATE tables

Join

Case when

VIEW

Review(3 min)

- CSV: comma-separated values
- Database
- Keys, ER Diagrams
- Data types
- The Most Important SQL statements (6)
- Aliases
- Logical Operators
- Aggregations functions

Review Exercise

- Retrieve the bar name and the average price of each bar. Order the results from the most expensive to the cheapest bar.

	bar	avg_price
1	bar 12	152.0710318342317
2	bar 13	150.56043863608502
3	bar 3	129.8095159989316
4	bar 16	121.36436782585224
5	bar 17	107.32970053202007
6	bar 18	105.21477991195086
7	bar 5	100.60385745142896
8	bar 14	93.4388559625484
9	bar 15	90.99379045958631
10	bar 20	86.51315756270196
11	bar 1	83.67852094673019
12	bar 8	67.91904428955168
13	bar 7	61.56748556054663
14	bar 10	60.57263578820442

Review Exercise

- Retrieve the bar name and the average price of each bar. Order the results from the most expensive to the cheapest bar.

- SELECT bar, AVG(price) as avg_price
- FROM hasonmenu
- GROUP BY bar
- ORDER BY avg_price DESC;

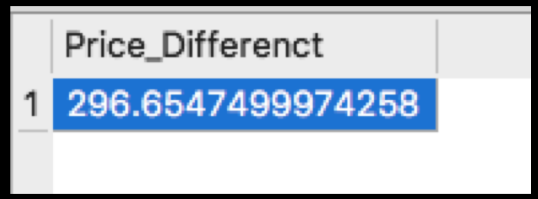
Arithmetic Operators

Operator	Meaning	Operates on
+ (Add)	Addition	Numeric value
- (Subtract)	Subtraction	Numeric value
* (Multiply)	Multiplication	Numeric value
/ (Divide)	Division	Numeric value
% (Modulo)	Returns the integer remainder of a division. For example, 17 % 5 = 2 because the remainder of 17 divided by 5 is 2.	Numeric value

Use them with SELECT, WHERE and HAVING statement

Find the price difference between the most expensive drink and the cheapest drink in bar 1 using HasOnMenu table

- SELECT MAX(price)-MIN(price)AS Price_Differenct
- FROM HasOnMenu
- WHERE bar = "bar 1";



	Price_Differenct
1	296.6547499974258

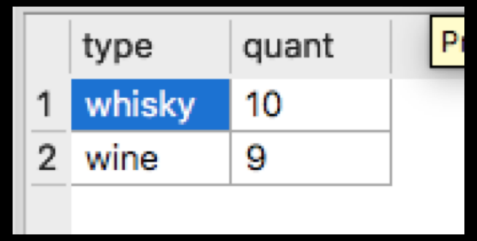
HAVING

- The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

WHERE subsets the returned data based on a logical condition.
WHERE appears after the **FROM**, **JOIN**, and **ON** clauses, but before **GROUP BY**.
HAVING appears after the **GROUP BY** clause, but before the **ORDER BY** clause.
HAVING is like **WHERE**, but it works on logical statements involving aggregations.

Find the type of drinks that have more than 8 different drinkIds in drinkInfo(HAVING)

- SELECT type, count(drinkId) as quant
- FROM drinkInfo
- GROUP BY type
- HAVING quant > 8



	type	quant
1	whisky	10
2	wine	9

Logical operators

- AND
- BETWEEN
- OR
- IN
- LIKE
- NOT

Like Operator (regular expressions)

WHERE affiliate_name LIKE 'a%'	Finds any values that start with "a"
WHERE affiliate_name LIKE '%a'	Finds any values that end with "a"
WHERE affiliate_name LIKE '%or%'	Finds any values that have "or" in any position
WHERE affiliate_name LIKE '_r%'	Finds any values that have "r" in the second position
WHERE affiliate_name LIKE 'a_%_%'	Finds any values that start with "a" and are at least 3 characters in length
WHERE affiliate_name LIKE 'a%o'	Finds any values that start with "a" and ends with "o"

Find drink types with an ‘a’

Find all the drinks that are either rum of whisky

```
SELECT *
FROM drinkInfo
WHERE type = “rum” OR type= “whisky”
```

	drinkId	type
1	drink 3	rum
2	drink 8	whisky
3	drink 9	rum
4	drink 13	whisky
5	drink 20	whisky
6	drink 21	whisky
7	drink 22	rum
8	drink 28	whisky
9	drink 29	rum
10	drink 31	rum
11	drink 34	whisky
12	drink 36	whisky
13	drink 39	rum
14	drink 40	whisky

DATE

- Most time stamps are unique
- Order from most least granular to the most.
- SQLite doesn’t have a date type so it would be complicated to deal with data.
- In other languages (posgresql) you can use DATE_TRUNC()

Date truncations

2017–04-01 12:15:01

RESULT	INPUT
2017–04-01 12:15:01	DATE_TRUNC ('second', 2017-04-01 12:15:01)
2017–04-01 00:00:00	DATE_TRUNC ('day', 2017-04-01 12:15:01)
2017–04-01 00:00:00	DATE_TRUNC ('month', 2017-04-01 12:15:01)
2017–01-01 00:00:00	DATE_TRUNC ('year', 2017-04-01 12:15:01)

Select specific part from the date

2017–04-01 12:15:01

RESULT	INPUT
1	DATE_PART ('second', 2017-04-01 12:15:01)
1	DATE_PART ('day', 2017-04-01 12:15:01)
4	DATE_PART ('month', 2017-04-01 12:15:01)
2017	DATE_PART ('year', 2017-04-01 12:15:01)

Subtle differences

TRUNC - Truncate Datetime - Oracle to MySQL Migration

In Oracle, TRUNC function, when applied for a datetime value, truncates it to the specified part (to day, by default). In MySQL, you can use DATE or DATE_FORMAT functions.

Oracle:

```
-- Truncate the current date and time (time part will be set to 00:00:00)
SELECT TRUNC(SYSDATE) FROM dual;
# 2016-04-07

SELECT TRUNC(SYSDATE, 'DD') FROM dual;
# 2016-04-07
```

MySQL:

```
-- Truncate the current date and time (convert to DATE value)
SELECT DATE(SYSDATE());
# 2016-04-07
```

http://www.sqlines.com/oracle-to-mysql/trunc_datetime

CASE WHEN act as IF ELSE

- It is like another column with given conditions

- The CASE statement always goes in the SELECT clause.

- CASE must include the following components: WHEN, THEN, and END. ELSE is an optional component to catch cases that didn't meet any of the other previous CASE conditions.

- You can make any conditional statement using any conditional operator (like WHERE) between WHEN and THEN. This includes stringing together multiple conditional statements using AND and OR.

	person	COUNT(quantity)	status
1	person 1	8	normal
2	person 10	23	normal
3	person 100	8	normal
4	person 11	9	normal
5	person 12	21	normal
6	person 13	39	alcoholic
7	person 14	7	normal
8	person 15	18	normal
9	person 16	19	normal
10	person 17	13	normal
11	person 18	15	normal
12	person 19	15	normal
13	person 2	15	normal
14	person 20	34	alcoholic