It's pronounced 'Ess-que-ell'

An Introduction to Coding for Databases

Tech Lady Hackathon 2016

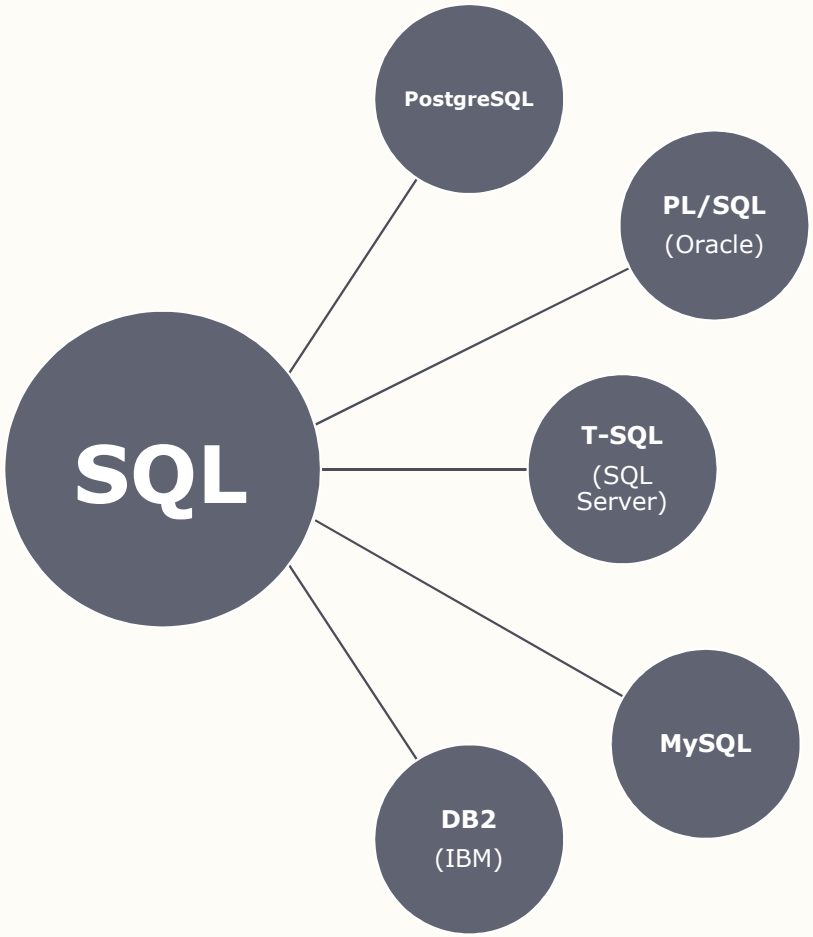
Alexis Johnson-Gresham

@ayrenay

What is SQL?

What is SQL?

SQL is the standard language for interacting with databases



4

SQL is a standard

This standard has various **implementations** which are run on different **platforms**

You can use SQL to do simple things like:

- add a user to a website
- enroll a student in a class
- remove an employee

SQL can also complete more complex tasks:

- determine which ads performed the best during a year
- show different employees different views of a source
- report quarterly sales for each salesperson by region

What kinds of questions do you need to answer?

5

SQL is for interacting with databases

These interactions include **defining**, **manipulating**, and **controlling** data and data structures

What is a database?

What is a database?

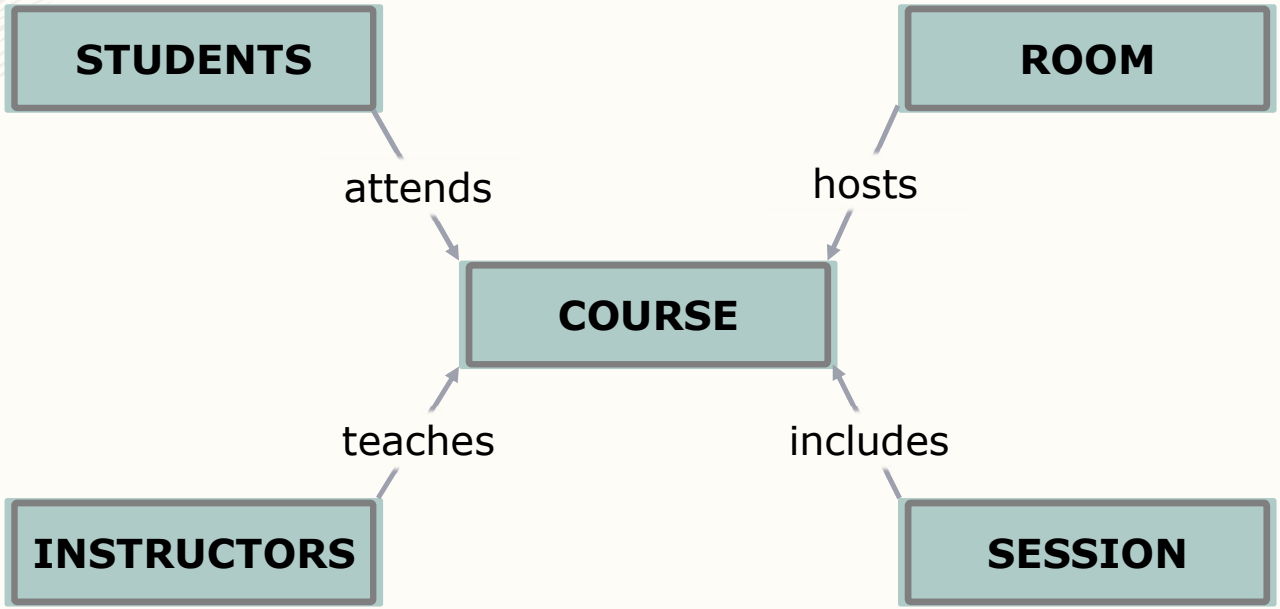
A database is a logically ordered collection of related data

8 Databases contain logically ordered data

- This data is typically stored in tables: rows and columns

userID	displayName	profileType	lastLogin
230985763	Elizabeth	pro	1/1/16 00:15
506839104	Camila	free	9/25/16 16:23
019598738	Kali	pro	4/10/16 23:23

9 Databases contain related data



Let’s make some tables!

11 Create a table using SQL

- Basic syntax:

```
create table <tablename> ( <columnname> <datatype> );
      1          2              3           4
```

Hold on – what's a datatype?

Hold on – what's a datatype?

Datatype describes the kind of data that should be in a column

- Some examples of datatypes:

Datatype	Description	Example
char	Fixed length string	DC MD VA
varchar	Variable length string	Alexis Eric Maryan
integer	Whole number	15 45
decimal	Number with value after decimal point	3.4 2.56
date	Date	1998-07-01 January 19, 1973 12/19/1992
timestamp	Date with a time value	2004-10-19 10:23:54

Datatype describes data's format

A single column can only be formally assigned a single datatype

15 Create a table using SQL

- **Try it**:

1. Go to **sqlfiddle.com** and set the editor to **PostgreSQL**
2. Add this line to the Schema (left) panel:

```
create schema techladyhackathon_init;
```

replace 'init' with your initials

3. Add the following to the Schema panel:

```
create table student
(
   first_name varchar(50)
   , last_initial char(1)
   , track varchar(25)
);
```

16 Check the table structure

- **Try it**:
  1. Add the following to the Query (right) panel:

```
select      column_name
            , data_type
from        information_schema.columns
where       table_name = 'student'
;
```

Other databases allow you use the following syntax instead of accessing the system tables:

```
describe table <tablename>;
```

17 Look at what's stored in the table

- Basic syntax:

```
select * from <tablename> ;
```

- **Try it**:

1. Add the following to the *right* panel:

```
select * from student ;
```

No rows returned! Why's that?

Let’s add some data!

19 Add some records to your table

- Frequently, you will add data from external files (i.e. CSV) or sources (i.e. API)
- Since our fiddles are all in the browser, we'll use INSERT statements for now
- **Try it**:
  1. Add the following to the *right* panel:

```
INSERT INTO student ("first_name", "last_initial", "track")
VALUES ('Alexis', 'J', 'Data'), ('Lin', 'Y', 'Design'),
('Elise', 'Z', 'Data'), ('Nellie', 'F', 'General'), ('Mel',
'B', 'Design'), ('Liza', 'G', 'General'), ('Beatrice', 'M',
'Data') ;
```

What else can we do with a SELECT statement?

What else can we do with a SELECT statement?

SELECT statements and their clauses can be used to filter, sort, and transform data

- WHERE is used to add limitations or specifications to your query
- You can use it to answer questions like:
  - What are sales like in the North region?
  - Who was born after 1989?
  - Which posts have content containing 'woman' or 'women'?
- Basic syntax:

```
select    <expressions>
from      <tablename>
where     <condition(s)> ;
```


SQL can be used to filter data

The WHERE clause controls which records appear in a SELECT statement's output

- Some SQL comparison operators:

Operator	Description	Example
Equality	Is exactly the same as (=)	name = 'Melissa' Manager <> 'Jones'
Inequality	Is not the same as (<>, !=); also <, >, <=, >=	year < 2010 age >= 21
LIKE	Is similar to	LIKE 't%mp%' LIKE 'pe_l'
IS NULL / IS NOT NULL	Has a value / does not have a value	published IS NOT NULL
IN / NOT IN	Is present / is not present in a list	type IN ('A', 'B', 'C')
NOT	Negates a condition	See above examples
AND	Two conditions are true	statement = 'balance sheet' AND year = '2015'
OR	At least one condition is true	product = 'water' OR product IS NULL

* EXISTS/NOT EXISTS and BETWEEN are two other helpful operators you may want to explore

SQL can be used to filter data

The WHERE clause requires some sort of conditional comparison

Try it:

Write a query to show rows for students on the ‘Data’ track only.

- ORDER BY runs after all other parts of a SQL query, causing the results to be displayed using the field you specify
- In this clause, columns can be referenced by
  - name : ORDER BY students_with_average_gpa
  - alias : ORDER BY avg_gpa
  - position : ORDER BY 2 [the second column in SELECT list]
- Basic syntax:

```
select      <expressions>
from        <tablename>
order by    <expressions>;
```

25

SQL can be used to sort data

The ORDER BY clause is used to display rows based on a specific column or columns

Try it:

Write a query to sort the students by first name.

- There are many different types of functions:
  - Single-row (scalar)
  - Multi-row (aggregate)
  - Analytic (advanced - not covered here)
- Functions take in no or a specific data type and return the same or a different data type
- Database documentation and other online resources are very helpful for learning about available functions and their parameters

27

SQL can be used to transform data

SQL contains many built-in **functions** for interacting and changing the appearance of data

- You can use single-row functions in the SELECT list and the WHERE clause
- Types of single-row functions:
  - numeric
  - character
  - datetime
  - general
  - conversion
  - encoding
  - hierarchical
- Example:

```
select      lower(first_name)
from        student;
```


SQL can be used to transform data

Single-row, or scalar, functions return one row of output per row of input

Try it:

Write a query to display first name in upper case.

- You can use aggregate functions in the SELECT list
- Some aggregate functions:
  - SUM
  - MIN
  - MAX
  - COUNT
- Example

```
select      sum(years_experience)
from        instructor;
```


SQL can be used to transform data

Multi-row, or aggregate, functions return one row of output per multiple rows of input

Try it:

Write a query to count the number of students.

- You can use both aggregate functions and individual expressions in the SELECT list
- When you do this, you **must** add the GROUP BY clause
- GROUP BY tells the engine to cluster the aggregations by the unique values for the column you specify
- Example

```
select      first_name
          , sum(years_experience)
from        instructor
group by    first_name;
```

32

SQL can be used to transform data

Aggregate functions can be combined with individual columns

Try it:

Write a query to show the number of students by track.

- HAVING restricts the records to be displayed based on an aggregate function (like WHERE for non-aggregate expressions)
- HAVING requires a condition, just like the WHERE clause
- Example

```
select      first_name
            , sum(years_experience)
from        instructor
group by    first_name
having      sum(years_experience)>=3;
```


SQL can be used to transform and filter data

Aggregate functions can be filtered using the HAVING clause

Try it:

Write a query to show the tracks with fewer than 3 students.

So, can you combine all of these clauses?

So, can you combine all of these clauses?

Yes!

38 SQL statements are ordered in a specific way

- You're required to use SELECT and FROM in all SQL statements
- When combined, the clauses must appear in the following order:

```
select      <expression>
from        <tablename>
where       <condition>
group by    <expression>
having      <condition>
order by    <expression>
;
```

You did it!