

Computational Structures in Data Science

Databases & SQL

UC Berkeley EECS
Lecturer
Michael Ball

UC Berkeley | Computer Science 88 | Michael Ball | <http://cs88.org>

1

Computing In the News

- Cybathlon tournament showcases life-changing tech for people with disabilities
 - By Aaliyah Harris, CNN ([Link](#))
- Maneuvering around household furniture or slicing bread for breakfast are tasks that most of us do without thinking.
- But many people with physical disabilities rely on technological help with essential activities like eating, washing laundry or moving from one room to another.
- [Cybathlon](#) is a global championship, organized by ETH Zurich, which brings together people with physical disabilities to compete in performing everyday tasks using state-of-the-art assistance technologies.



UC Berkeley | Computer Science 88 | Michael Ball | <http://cs88.org>


2

Announcements

- Once again: No labs this week.
 - This means the lab couple assignments will be pushed into deadweek a bit. We'll try to have them out early.
- HW12 and Lab 12 dates will be updated later today.

UC Berkeley | Computer Science 88 | Michael Ball | <http://cs88.org>

3



Computational Structures in Data Science

Databases & SQL

UC Berkeley EECS
Lecturer
Michael Ball

UC Berkeley | Computer Science 88 | Michael Ball | <http://cs88.org>

4

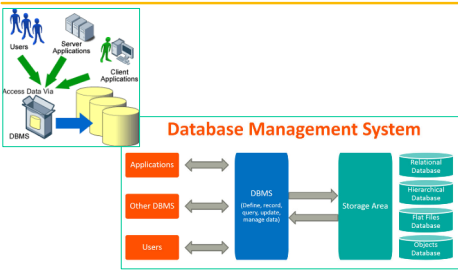
Why Databases?

- Data lives in files: website access logs, in images, in CSVs and so on...
 - Useful, but hard to access, aggregate and compute results with.
- Databases provide a mechanism to store vast amounts of data in an *organized* manner.
 - The (often) rely on "tables" as an abstraction. We
 - There are other kinds of databases, that store "documents" or other forms of data.
- This stuff is the topic of CS186

UC Berkeley | Computer Science 88 | Michael Ball | <http://cs88.org>

5

Database Management Systems

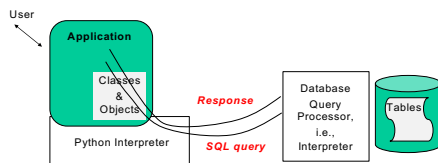


The diagram illustrates the architecture of Database Management Systems. On the left, 'Users' and 'Client Applications' interact with 'Server Applications' and 'Access Control' components. These components connect to a 'Database Management System' (DBMS) box, which is labeled 'Database Management System (DBMS)'. The DBMS box is connected to a 'Storage Area' box. The Storage Area is further divided into 'Relational Database', 'Hierarchical Database', 'Text File Database', and 'Object Database'. The DBMS box also has connections to 'Applications', 'Other DBMS', and 'Users'.

UC Berkeley | Computer Science 88 | Michael Ball | <http://cs88.org>

6

Applications Issue Queries to a Database

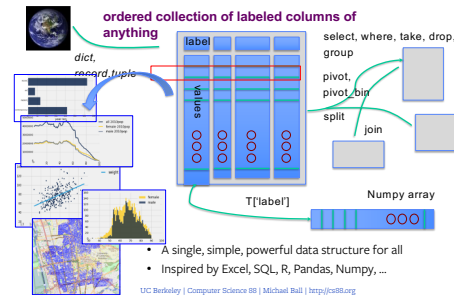


- The SQL language is represented in query strings delivered to a DB backend.
- Use the techniques learned here to build clean abstractions.
- You have already learned the relational operators!

UC Berkeley | Computer Science 88 | Michael Ball | <http://cs88.org>

7

Data 8 Tables



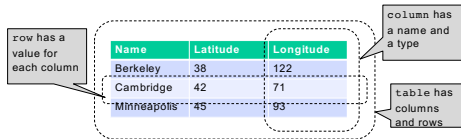
- A single, simple, powerful data structure for all
- Inspired by Excel, SQL, R, Pandas, Numpy, ...

UC Berkeley | Computer Science 88 | Michael Ball | <http://cs88.org>

8

Database Management Systems

- DBMS are persistent tables with powerful relational operators
 - Important, heavily used, interesting!
- A **table** is a collection of **records**, which are **rows** that have a value for each **column**



- Structure Query Language (SQL) is a declarative programming language describing operations on tables

UC Berkeley | Computer Science 88 | Michael Ball | <http://cs88.org>

9

Computational Structures in Data Science



UC Berkeley E&CS
Lecturer
Michael Ball

Introduction to SQL

UC Berkeley | Computer Science 88 | Michael Ball | <http://cs88.org>

10

Why SQL?

- SQL is a *declarative* programming language for accessing and modifying data in a relational database.
- It is an entirely new way of thinking ("new" in 1970, and new to you now!) that specifies *what* should happen, but not *how* it should happen.
- One of a few major programming paradigms
 - Imperative/Procedural
 - Object Oriented
 - Functional
 - Declarative

UC Berkeley | Computer Science 88 | Michael Ball | <http://cs88.org>

11

What is SQL?

- A declarative language
 - Described *what* to compute
 - Imperative languages, like python, describe *how* to compute it
 - Query processor (interpreter) chooses which of many equivalent query plans to execute to perform the SQL statements
- ANSI and ISO standard, but many variants
 - This SQL will work on nearly all "relational" databases.
- **SELECT** statement creates a new table, either from scratch or by projecting a table
- **create table** statement gives a global name to a table
- Lots of other statements
 - analyze, delete, explain, insert, replace, update, ...

UC Berkeley | Computer Science 88 | Michael Ball | <http://cs88.org>

12

SQL example

- SQL statements create tables
 - Give it a try with sqlite3 or <http://kripken.github.io/sql.js/CI/W/>
 - Each statement ends with ‘

```
culder$ sqlite3
SQLite version 3.9.2 2015-11-02 18:31:45
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> select 38 as latitude, 122 as longitude, "Berkeley" as
name;
38|122|Berkeley
sqlite>
```

UC Berkeley | Computer Science 88 | Michael Hall | <http://cs88.org>

13

A Running example from Data 8 Lec 10

```
# An example of creating a Table from a list of rows.
Table(['Flavor','Color','Price']).with_rows([
  ('strawberry','pink',3.55),
  ('chocolate','light brown',4.75),
  ('chocolate','dark brown',5.25),
  ('strawberry','pink',5.25),
  ('bubblegum','pink',4.75)])
```

Flavor	Color	Price
strawberry	pink	3.55
chocolate	light brown	4.75
chocolate	dark brown	5.25
strawberry	pink	5.25
bubblegum	pink	4.75



```
culder@CulderMac ~/Classes/CS88-Fall18/ideas/sql> sqlite3 icecream.db
SQLite version 3.13.0 2016-05-18 10:57:30
Enter ".help" for usage hints.
sqlite>
```

UC Berkeley | Computer Science 88 | Michael Hall | <http://cs88.org>

14

select

- Comma-separated list of *column descriptions*
- Column description is an expression, optionally followed by **as** and a **column name**
- Selecting literals creates a one-row table
- select** "strawberry" as Flavor, "pink" as Color, 3.55 as Price;
- union** of select statements is a table containing the union of the rows

```
select "strawberry" as Flavor, "pink" as Color, 3.55 as Price union
select "chocolate", "light brown", 4.75 union
select "chocolate", "dark brown", 5.25 union
select "strawberry", "pink", 5.25 union
select "bubblegum", "pink", 4.75
```

UC Berkeley | Computer Science 88 | Michael Hall | <http://cs88.org>

15

Select ...

```
culder@CulderMac ~/Classes/CS88-Fall18/ideas/sql> sqlite3 icecream.db
SQLite version 3.13.0 2016-05-18 10:57:30
Enter ".help" for usage hints.
sqlite> create table cones as
...> select 1 as ID, "strawberry" as Flavor, "pink" as Color, 3.55 as Pri
ce union
...> select 2, "chocolate", "light brown", 4.75 union
...> select 3, "chocolate", "dark brown", 5.25 union
...> select 4, "strawberry", "pink", 5.25 union
...> select 5, "bubblegum", "pink", 4.75 union
...> select 6, "chocolate", "dark brown", 5.25;
sqlite> select * from cones;
1|strawberry|pink|3.55
2|chocolate|light brown|4.75
3|chocolate|dark brown|5.25
4|strawberry|pink|5.25
5|bubblegum|pink|4.75
6|chocolate|dark brown|5.25
sqlite>
```

```
cones = Table(['ID', 'Flavor', 'Color', 'Price']).with_rows([
  (1, 'strawberry', 'pink', 3.55),
  (2, 'chocolate', 'light brown', 4.75),
  (3, 'chocolate', 'dark brown', 5.25),
  (4, 'strawberry', 'pink', 5.25),
  (5, 'bubblegum', 'pink', 4.75),
  (6, 'chocolate', 'dark brown', 5.25)])
```

ID	Flavor	Color	Price
1	strawberry	pink	3.55
2	chocolate	light brown	4.75
3	chocolate	dark brown	5.25
4	strawberry	pink	5.25
5	bubblegum	pink	4.75
6	chocolate	dark brown	5.25

UC Berkeley | Computer Science 88 | Michael Hall | <http://cs88.org>

16

Projecting existing tables

- Input table specified by **from** clause
- Subset of rows selected using a **where** clause
- Ordering of the selected rows declared using an **order by** clause

```
select [columns] from [table] where [condition] order by [order];
```

```
select * from cones order by Price;
```

ID	Flavor	Color	Price
1	strawberry	pink	3.55
2	chocolate	light brown	4.75
5	bubblegum	pink	4.75
3	chocolate	dark brown	5.25
4	strawberry	pink	5.25
6	chocolate	dark brown	5.25

UC Berkeley | Computer Science 88 | Michael Hall | <http://cs88.org>

17

Projection

```
In [5]: cones.select(['Flavor', 'Price'])
Out[5]:
```

Flavor	Price
strawberry	3.55
chocolate	4.75
chocolate	5.25
strawberry	5.25
bubblegum	4.75
chocolate	5.25

```
sqlite> select Flavor, Price from cones;
Flavor|Price
strawberry|3.55
chocolate|4.75
chocolate|5.25
strawberry|5.25
bubblegum|4.75
chocolate|5.25
```

- A "projection" is a view of a table, it doesn't alter the state of the table.

UC Berkeley | Computer Science 88 | Michael Hall | <http://cs88.org>

18

Permanent Data Storage

```
sqlite> .quit
culler@cullerMac ~/Classes/CS88-Fall18/ideas/sql> sqlite3 icecream.db
SQLite version 3.13.0 2016-05-18 10:57:38
Enter ".help" for usage hints.
sqlite> .tables
cones
sqlite> select * from cones where Color is "dark brown";
3|chocolate|dark brown|5.25
6|chocolate|dark brown|5.25
sqlite> |
```

ID	Flavor	Color	Price
1	strawberry	pink	3.50
2	chocolate	light brown	4.75
5	bubbligum	pink	4.75
3	chocolate	dark brown	5.25
4	strawberry	pink	5.25
6	chocolate	dark brown	5.25

UC Berkeley | Computer Science 88 | Michael Hall | <http://cs88.org>

19

Computational Structures in Data Science



UC Berkeley EECS
Lecturer
Michael Hall

Filtering in SQL

UC Berkeley | Computer Science 88 | Michael Hall | <http://cs88.org>

20

Filtering rows - where

- Set of Table records (rows) that satisfy a condition

```
select [columns] from [table] where [condition] order by [order];
```

```
sqlite> select * from cones where Flavor = "chocolate";
3|chocolate|dark brown|5.25
6|chocolate|dark brown|5.25
sqlite> |
```

```
sqlite> select * from cones where Flavor = "chocolate";
3|chocolate|dark brown|5.25
6|chocolate|dark brown|5.25
sqlite> |
```

```
sqlite> select * from cones where Price > 5;
3|chocolate|dark brown|5.25
6|chocolate|dark brown|5.25
sqlite> |
```

```
SQL:
sqlite> select * from cones where Price > 5;
ID|Flavor|Color|Price
3|chocolate|dark brown|5.25
6|chocolate|dark brown|5.25
sqlite> |
```

UC Berkeley | Computer Science 88 | Michael Hall | <http://cs88.org>

21

SQL Operators for predicate

- use the WHERE clause in the SQL statements such as `SELECT`, `UPDATE`, and `DELETE`, to filter rows that do not meet a specified condition

SQLite understands the following binary operators, in order from highest to lowest precedence:

```
||
+ / %
* -
<< >> & |
< <= > >=
= == != <> IS IS NOT IN LIKE GLOB MATCH REGEXP
AND
OR
```

Supported unary prefix operators are these:

```
- + ~ NOT
```

UC Berkeley | Computer Science 88 | Michael Hall | <http://cs88.org>

22

Summary

- SQL a declarative programming language on relational tables
 - largely familiar to you from data8
 - create, select, where, order, group by, join
- Databases are accessed through Applications
 - e.g., all modern web apps have Database backend
 - Queries are issued through API
 - » Be careful about app corrupting the database
- Data analytics tend to draw database into memory and operate on it as a data structure
 - e.g., Tables
- More in lab

UC Berkeley | Computer Science 88 | Michael Hall | <http://cs88.org>

23

create table

- SQL often used interactively
 - Result of select displayed to the user, but not stored
- Create table statement gives the result a name
 - Like a variable, but for a permanent object

```
create table [name] as [select statement];
```

UC Berkeley | Computer Science 88 | Michael Hall | <http://cs88.org>

24

SQL: creating a named table

```
create table cones as
select 1 as ID, "strawberry" as Flavor, "pink" as Color,
3.55 as Price union
select 2, "chocolate", "light brown", 4.75 union
select 3, "chocolate", "dark brown", 5.25 union
select 4, "strawberry", "pink", 5.25 union
select 5, "bubblegum", "pink", 4.75 union
select 6, "chocolate", "dark brown", 5.25;
```

Notice how column names are introduced and implicit later on.

UC Berkeley | Computer Science 88 | Michael Hall | <http://cs88.org>

25

Summary – Part 1

```
SELECT <col spec> FROM <table spec> WHERE <cond spec>
GROUP BY <group spec> ORDER BY <order spec>;
```

```
INSERT INTO table(column1, column2,...)
VALUES (value1, value2,...);
```

```
CREATE TABLE name ( <columns> );
```

```
CREATE TABLE name AS <select statement>;
```

```
DROP TABLE name;
```

UC Berkeley | Computer Science 88 | Michael Hall | <http://cs88.org>

38

Summary

- SQL a declarative programming language on relational tables
 - largely familiar to you from data8
 - create, select, where, order, group by, join
- Databases are accessed through Applications
 - e.g., all modern web apps have Database backend
 - Queries are issued through API
 - » Be careful about app corrupting the database
- Data analytics tend to draw database into memory and operate on it as a data structure
 - e.g., Tables
- More in lab

UC Berkeley | Computer Science 88 | Michael Hall | <http://cs88.org>

39