

# SQL

---

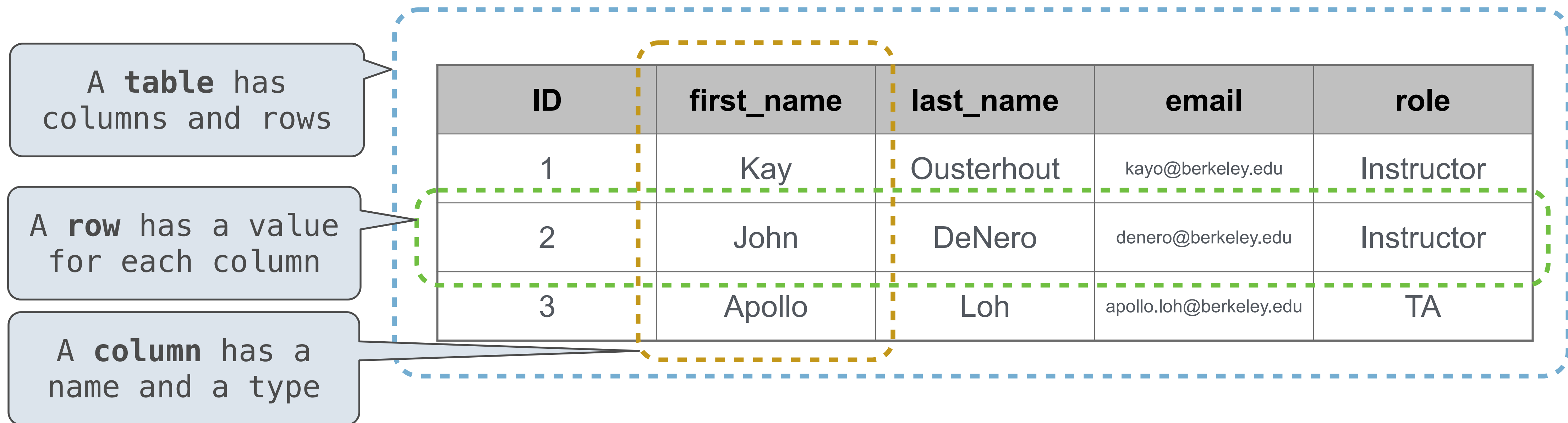
# Announcements

# Databases

# Database Management Systems

Database management systems (DBMS) are important, heavily used, and interesting!

A table is a collection of records, which are rows that have a value for each column



The Structured Query Language (SQL) is perhaps the most widely used programming language

# Structured Query Language (SQL)

# Tables and Queries

Give the result a name

```
CREATE TABLE [name] AS
```

```
SELECT [columns]  
FROM [table]  
WHERE [condition]  
ORDER BY [order];
```



Data represented  
as tables

Select statements  
transform one  
table to another



Output table may  
have different  
rows and columns  
than input

# SELECT Statements

**SELECT** [columns] **FROM** [table] **WHERE** [condition] **ORDER BY** [order];

ID	first_name	last_name	email	role
1	Kay	Ousterhout	kayo@berkeley.edu	Instructor
2	John	DeNero	denero@berkeley.edu	Instructor
3	Apollo	Loh	apollo.loh@berkeley.edu	TA

**SELECT** email, first\_name **FROM** users **WHERE** role = "Instructor";

email	first_name
kayo@berkeley.edu	Kay
denero@berkeley.edu	John

column name

(Demo)

# SELECT Statements

`SELECT [columns] FROM [table] WHERE [condition] ORDER BY [order];`

ID	first_name	last_name	email	role
1	Kay	Ousterhout	kayo@berkeley.edu	Instructor
2	John	DeNero	denero@berkeley.edu	Instructor
3	Apollo	Loh	apollo.loh@berkeley.edu	TA

Make a new table with the TA's emails:

`CREATE TABLE ta_emails AS SELECT email FROM users WHERE role = "TA";`

Make a new table **ta\_emails** with the result

email
apollo.loh@berkeley.edu

(Demo)

[pollev.com/cs61a](https://pollev.com/cs61a)



# Command overview

---

SQLite shell commands:

**.tables:** List all tables

**.mode box** (or **.mode column**): Make the display pretty

SQL:

**SELECT \* FROM [table] LIMIT 5;** Show the first 5 rows of table

**[expression] AS [name], [expression] AS [name], ...**

**SELECT [columns] FROM [table] WHERE [condition] ORDER BY [order];**

Which columns

Which rows

# Internet Movie Database

<https://developer.imdb.com/non-commercial-datasets/>

( Demo )

# Example titles queries

tconst	title	year	runtime	genres
tt0012349	The Kid	1921	68	Comedy,Drama,Family
...				

Create a new table **oldest** with the 5 oldest movie titles and years in the dataset

```
CREATE TABLE oldest as SELECT title, year FROM titles ORDER BY year LIMIT 5;
```

[pollev.com/cs61a](http://pollev.com/cs61a)

Select the **title** and **runtime in hours** of the 20 longest movies

```
SELECT title, runtime / 60.0 FROM titles ORDER BY runtime DESC LIMIT 20;
```

[pollev.com/cs61a](http://pollev.com/cs61a)

# Example names queries

nconst	name	birth	death	profession	knownForTitles
nm0000002	Lauren Bacall	1924	2014	actress,miscellaneous,soundtrack	tt0037382,tt0075213,tt0038355,tt0117057
...					

Select the names and ages of all people:

```
SELECT name, death - birth FROM names;
```

[pollev.com/cs61a](http://pollev.com/cs61a)

Select the names and ages of all alive people, where birth is not ‘\N’:

```
SELECT name, death - birth FROM names WHERE death = '\N' and birth != '\N' ;
```

[pollev.com/cs61a](http://pollev.com/cs61a)

# Declarative Programming

In **declarative programming**:

- A "program" is a description of the desired result
- The interpreter figures out how to generate the result

Database is responsible for determining how to run this quickly

**SELECT** name, death - birth **FROM** names **WHERE** death = '\N' **and** birth != '\N';

nconst	name	birth	death	profession	knownForTitles
nm0000002	Lauren Bacall	1924	2014	actress,miscellaneous,soundtrack	tt0037382,tt0075213,tt0038355,tt0117057
...					
nm0000250	Renée Zellweger	1969	\N	actress,producer,writer	tt0299658,tt0159365,tt0116695,tt0243155
nm0000255	Ben Affleck	1972	\N	producer,actor,writer	tt1024648,tt2267998,tt0840361,tt0119217

**Possible process:**

Scan all rows for death = '\N' **and** birth != '\N'

# Declarative Programming

```
SELECT name, death - birth FROM names WHERE death = '\N' and birth != '\N';
```

nconst	name	birth	death	profession	knownForTitles
nm0000002	Lauren Bacall	1924	2014	actress,miscellaneous,soundtrack	tt0037382,tt0075213,tt0038355,tt0117057
...					
nm0000250	Renée Zellweger	1969	\N	actress,producer,writer	tt0299658,tt0159365,tt0116695,tt0243155
nm0000255	Ben Affleck	1972	\N	producer,actor,writer	tt1024648,tt2267998,tt0840361,tt0119217
...					

**Index:** Separate data structure that keeps track of where certain values are in the table

death index: {2014: • , '\N': • }

**Possible process:**

Use death index to lookup only entries with death = '\N'