SQL

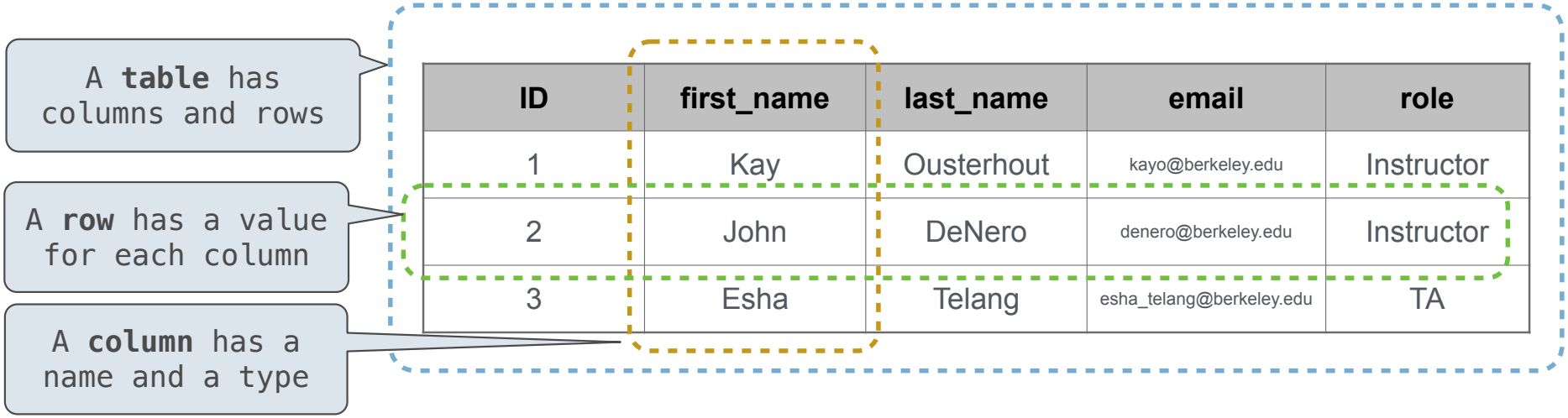
Announcements

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

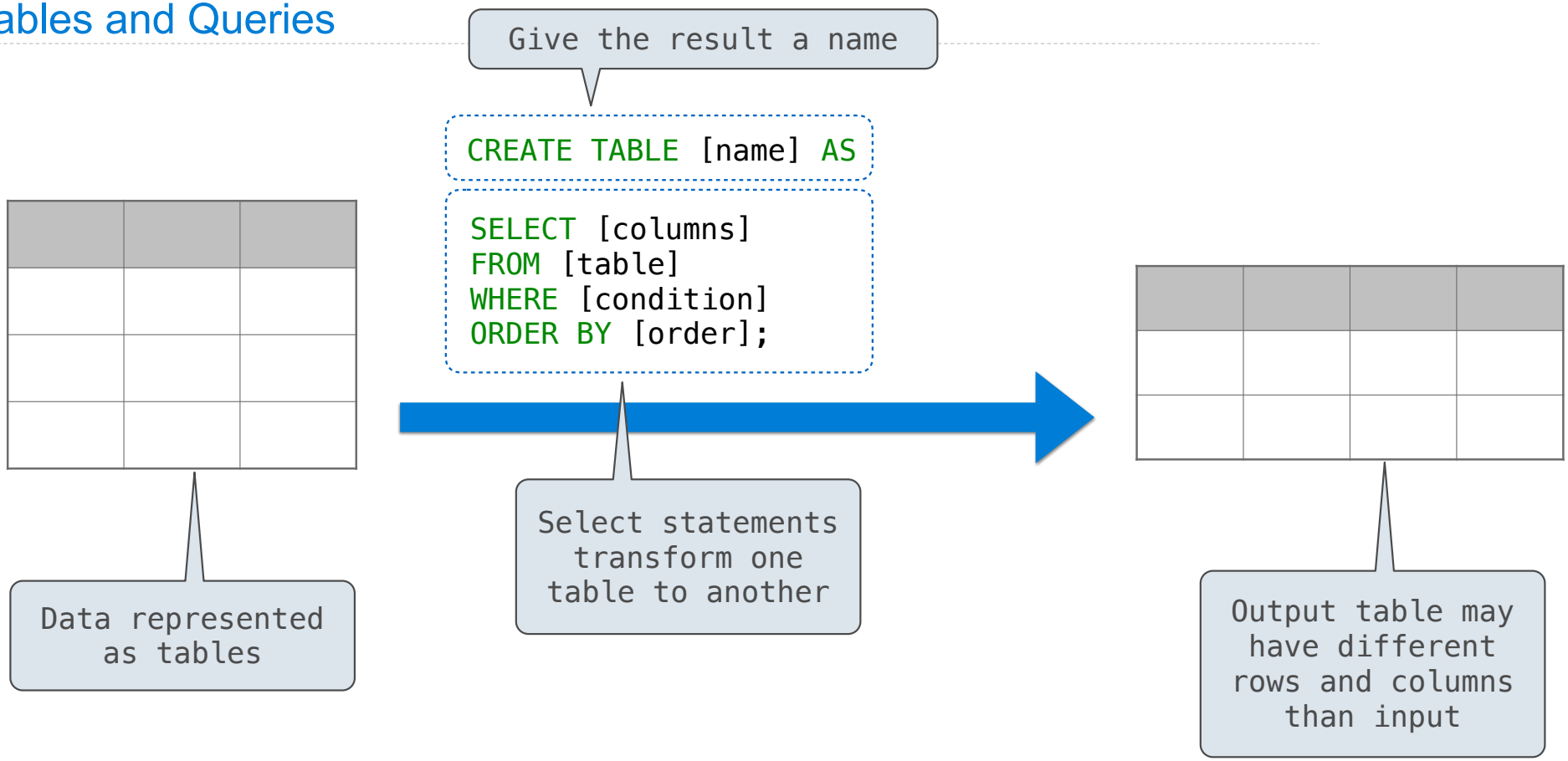
Databases in Applications

Databases are used to provide a **single source of truth** and ensure **data validity**

- Validity might require multiple updates to all happen together or not at all (**A**tomicity)
- Validity might also require that a set of constraints are always met (**C**onsistency)
- An application could run concurrently on many machines with a shared database (**I**solation)
- Programs/machines could crash, but databases ensure updates are persistent (**D**urability)

Structured Query Language (SQL)

Tables and Queries



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	Esha	Telang	esha_telang@berkeley.edu	TA

Which columns

```
SELECT email, first_name FROM users WHERE role = "Instructor";
```

column name

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

(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	Esha	Telang	esha_telang@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
esha_telang@berkeley.edu

(Demo)

Internet Movie Database

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

(Demo)

Command overview

SQLite shell commands:

.tables: List all tables

.mode box: 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

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;
```

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;
```

Example names queries

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

Select the name of each person and how long they lived:

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

Select the names and ages of all alive people (death is '\N') who have known birth years:

```
SELECT name, 2026 - birth FROM names WHERE death = '\N' and birth != '\N' ;
```

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, 2026 - 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, 2026 - 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'