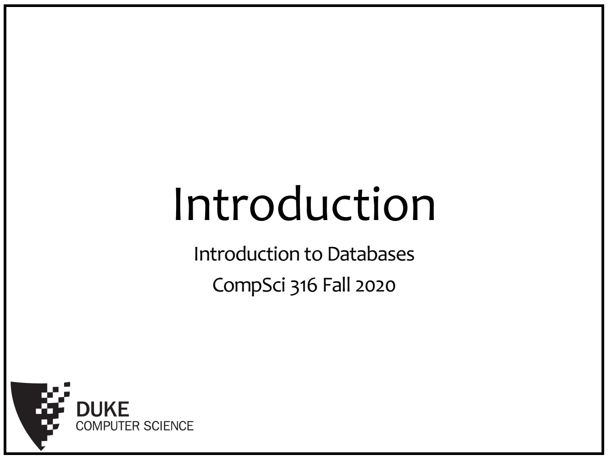


Introduction

Introduction to Databases

CompSci 316 Fall 2020

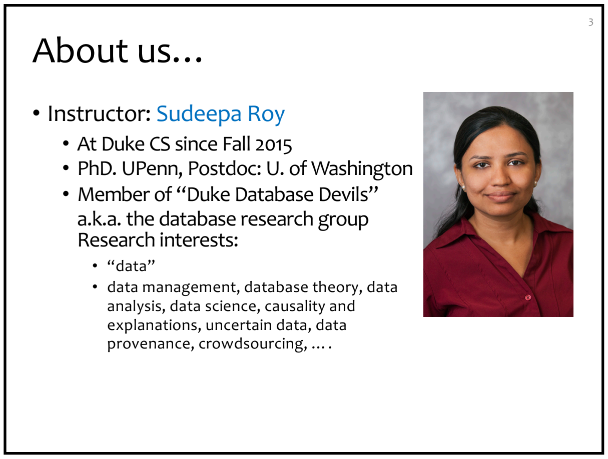
DUKE COMPUTER SCIENCE

1

Welcome to

CompSci 316: Introduction to Database Systems!!
Fall 2020

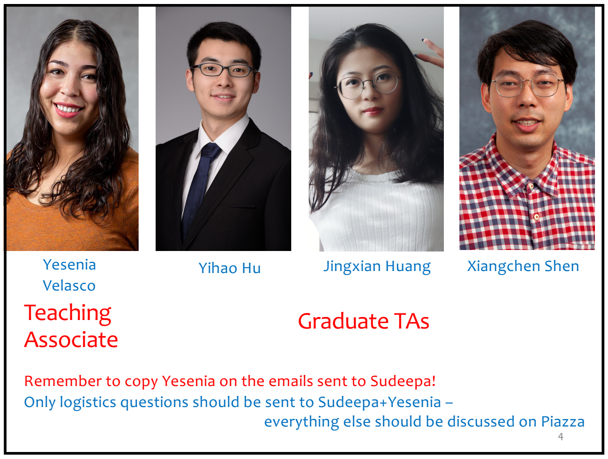
2



About us…

- Instructor: Sudeepa Roy
  - At Duke CS since Fall 2015
  - PhD. UPenn, Postdoc: U. of Washington
  - Member of "Duke Database Devils" a.k.a. the database research group
    Research interests:
    - "data"
    - data management, database theory, data analysis, data science, causality and explanations, uncertain data, data provenance, crowdsourcing, … .

3



Yesenia Velasco

Yihao Hu

Jingxian Huang

Xiangchen Shen

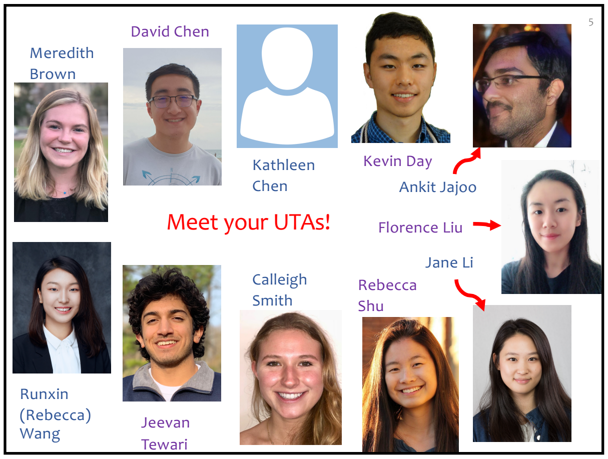
Teaching Associate

Graduate TAs

Remember to copy Yesenia on the emails sent to Sudeepa!

Only logistics questions should be sent to Sudeepa+Yesenia – everything else should be discussed on Piazza

4



Meredith Brown

David Chen

Kathleen Chen

Kevin Day

Ankit Jajoo

Meet your UTAs!

Florence Liu

Jane Li

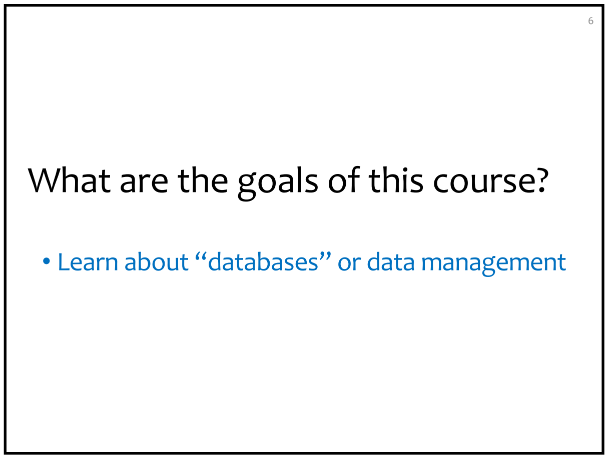
Calleigh Smith

Rebecca Shu

Runxin (Rebecca) Wang

Jeevan Tewari

5



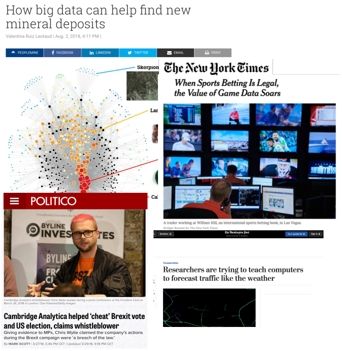
What are the goals of this course?

- Learn about "databases" or data management

6

Why do we care about data? (easy)

7



How big data can help find new mineral deposits

Valentina Ruiz Leotaud | Aug. 2, 2018, 4:11 PM |

The New York Times

When Sports Betting Is Legal, the Value of Game Data Soars

A trader working at William Hill, an international sports betting book, in Las Vegas.

Bridget Bennett for The New York Times

The Washington Post

Transportation

Researchers are trying to teach computers to forecast traffic like the weather

POLITICO

Cambridge Analytica whistleblower Chris Wylie speaks during a press conference at the Frontline Club on March 26, 2018 in London | Dan Kitwood/Getty Images

Cambridge Analytica helped 'cheat' Brexit vote and US election, claims whistleblower

Giving evidence to MPs, Chris Wylie claimed the company's actions during the Brexit campaign were 'a breach of the law.'

By MARK SCOTT | 3/27/18, 5:46 PM CET | Updated 3/29/18, 9:18 PM CET

… The three years of gathering and analyzing data culminated in what U.S. Sailing calls their "Rio Weather Playbook," a body of critical information about each of the seven courses only available to the U.S. team…

— FiveThirtyEight, **"Will Data Help U.S. Sailing Get Back On The Olympic Podium?"** Aug 15, 2016

Data =
Money
Information
Power
Fun
in
Science, Business,
Politics, Security
Sports, Education, ….

7

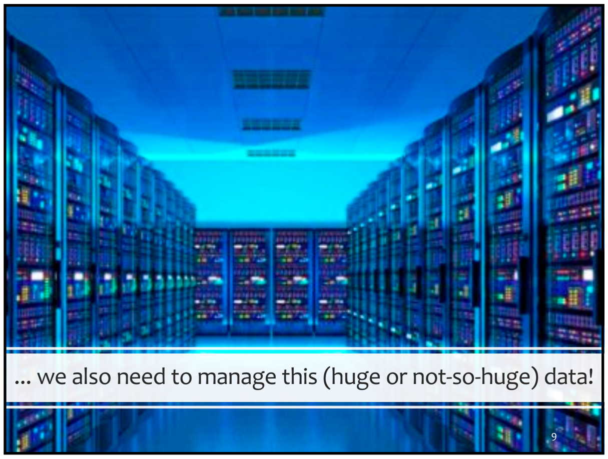
8

Wait.. don't we need to take a Machine Learning or Statistics course for those things?

Yes, but..

Pic: https://www.technobuffalo.com/sites/technobuffalo.com/files/styles/xlarge/public/wp/2012/05/confused-student.jpg

8



… we also need to manage this (huge or not-so-huge) data!

9

9

Also think about building a new App or website based on data from scratch

10

- E.g., your own version of mini-Amazon* or a Book Selling Platform
- Large data! (think about all books in the world or even in English)

- How do we start?

* Many of you are going to do this in the course projects!

10

Who are the key people? (book-selling website)

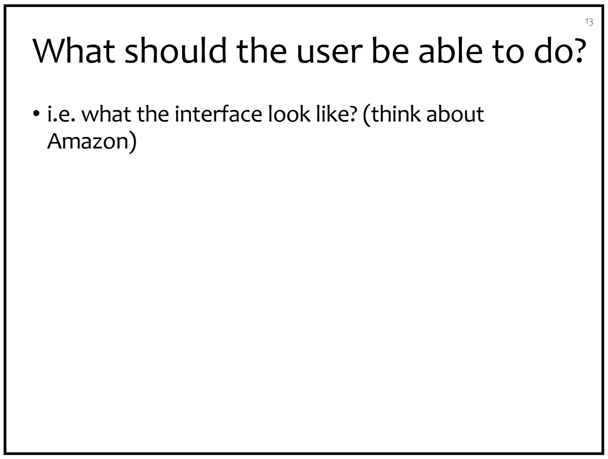
11

11

Who are the key people? (book-selling website)

12

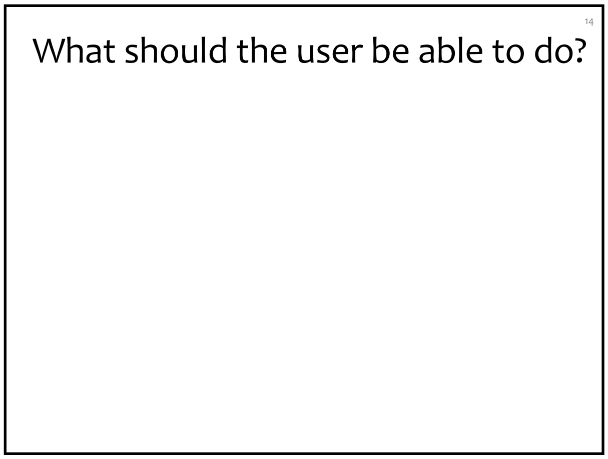
12



What should the user be able to do?

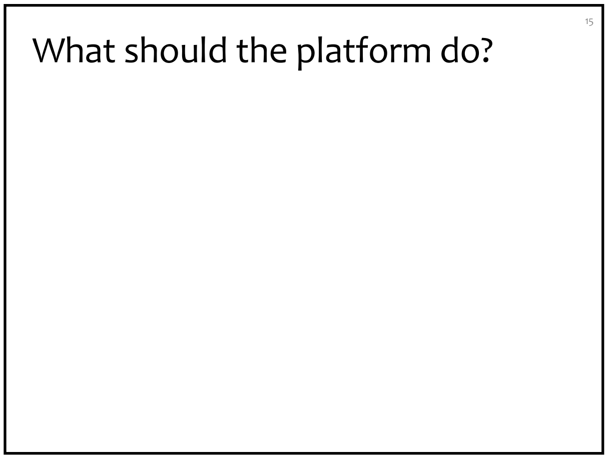
- i.e. what the interface look like? (think about Amazon)

13



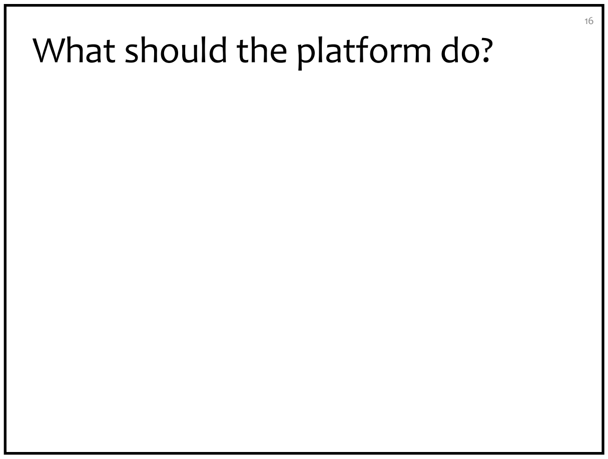
What should the user be able to do?

14



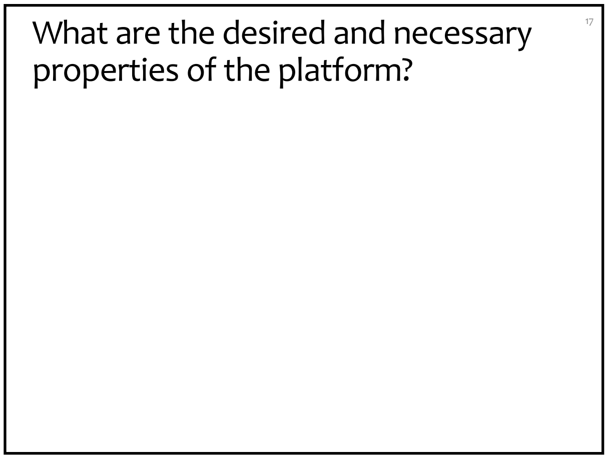
What should the platform do?

15



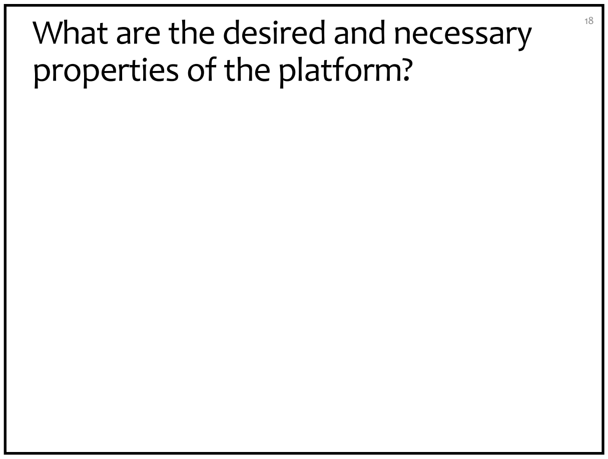
What should the platform do?

16



What are the desired and necessary properties of the platform?

17

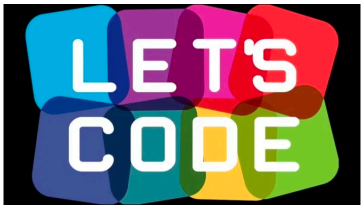


What are the desired and necessary properties of the platform?

18

19

That was the design phase (a basic one though)



How about C++, Java, or Python?
On data stored in large files

https://i1.wp.com/dynamiclandscapes.vita-learn.org/wp-content/uploads/2019/05/Lets-code.jpg?resize=768%2C432&ssl=1

19

20

Sounds simple!

James Morgan#Durham, NC

… …
A Tale of Two Cities#Charles Dickens#3.50#7
To Kill a Mockingbird#Harper Lee#7.20#1
Les Miserables#Victor Hugo#12.80#2
… …

- Text files – for books, customer, …
- Books listed with title, author, price, and no. of copies
- Fields separated by #'s

20

21

Query by programming

James Morgan#Durham, NC

… …
A Tale of Two Cities#Charles Dickens#3.50#7
To Kill a Mockingbird#Harper Lee#7.20#1
Les Miserables#Victor Hugo#12.80#2
… …

- James Morgan wants to buy "To Kill a Mockingbird"
- A simple script
  - Scan through the books file
  - Look for the line containing "To Kill a Mockingbird"
  - Check if the no. of copies is >= 1
  - Bill James $7.20 and reduce the no. of copies by 1

Better idea than scanning?

What if he changes the "query" and wants to buy a book by Victor Hugo?

21

22

Revisit: What are the desired and necessary properties of the platform?

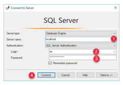
- Should be able to handle a large amount of data
- Should be efficient and easy to use (e.g., search with authors as well as title)
- If there is a crash or loss of power, information should not be lost or inconsistent
  - Imagine a user was in the middle of a transaction when a crash happened, paid the money, but the book has not been purchased
- No surprises with multiple users logged in at the same time
  - Imagine one last copy of a book that two users are trying to purchase at the same time
- Easy to update and program
  - For the admin

22

23

Solution?

- DBMS = Database Management System


Connect to Server

SQL Server

Server type: Database Engine
Server name: localhost 1
Authentication: SQL Server Authentication
Login: sa 2
Password: ************** 3
☑ Remember password

4 Connect | Cancel | Help | Options >>

PostgreSQL MySQL ORACLE DATABASE IBM DB2 mongoDB

23

24

A DBMS takes care of all of the following (and more):

In an easy-to-code, efficient, and robust way

- Sho… to handle a large amount of data ✓
- …d be efficient and easy to use (e.g., search with authors as well as title) ✓
- If there is a crash o… power, information should not be lost o… inconsistent ✓
  - …user was in the middle of a transaction when a c… …ppened, paid the money, but the book has not be…
- No surprises with multiple users logged i… ✓
  - Imagine one last copy of a book that two users … purchase at the same time
- Easy to update and program ✓
  - For the admin

Optimization

Index

Recovery

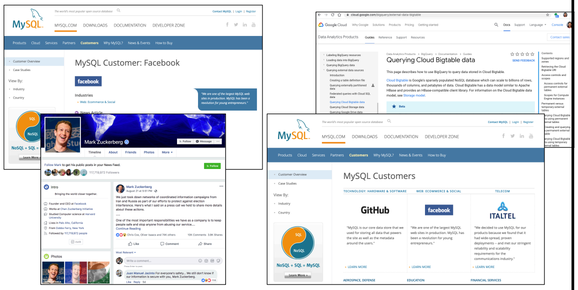
Concurrency Control

Declarative

* We will learn these in the course!

24

DBMS helps the big ones!



MySQL Customer: Facebook

"We are one of the largest MySQL web sites in production. MySQL has been a revolution for young entrepreneurs."

Querying Cloud Bigtable data

This page describes how to use BigQuery to query data stored in Cloud Bigtable.

Cloud Bigtable is Google's sparsely populated NoSQL database which can scale to billions of rows, thousands of columns, and petabytes of data. Cloud Bigtable has a data model similar to Apache HBase and provides an HBase-compatible client library. For information on the Cloud Bigtable data model, see Storage model.

Mark Zuckerberg

We just took down networks of coordinated information campaigns from Iran and Russia as part of our efforts to protect against election interference. Here's what I said on a press call we held to share more details about these actions.

One of the most important responsibilities we have as a company is to keep people safe and stop anyone from abusing our service....

MySQL Customers

GitHub: "MySQL is our core data store that we used for storing all data that powers the site as well as the metadata around the users."

facebook: "We are one of the largest MySQL web sites in production. MySQL has been a revolution for young entrepreneurs."

ITALTEL: "We decided to use MySQL for our products because we found that it had wide-spread, proven deployments -- and met our stringent reliability and scalability requirements for the communications industry."

Note: Not always the "standard" DBMS (called Relational DBMS), but we need to know pros and cons of all alternatives

25

CompSci 316 gives an intro to DBMS

- How can a user use a DBMS (programmer's/designer's perspective)
  - Run queries, update data (SQL, Relational Algebra)
  - Design a good database (ER diagram, normalization)
  - Use different types of data (Mostly relational, also XML/JSON)
- How does a DBMS work (system's or admin's perspective, also for programmers for writing better queries)
  - Storage, index
  - Query processing, join algorithms, query optimizations
  - Transactions: recovery and concurrency control
- Glimpse of advanced topics and other DBMS
  - NOSQL, Spark (big data)
  - Data mining, Parallel DBMS
- Hands-on experience in class projects by building an end-to-end website or an app that runs on a database

26

Misc. course info

- All information available on the Course Website: https://www2.cs.duke.edu/courses/fall20/compsci316/
  - Course info; tentative schedule and reference sections in the book; lecture slides, assignments, help docs, …

27

Projects

- **Fixed project Option:** Mini-amazon
- **Open project Option:** Your own idea! (More work, more fun)
  - From previous years:
  - RA: next-generation relational algebra interpreter
    - You may get to try it out for Homework #1!
  - *Managing tent shifts and schedules!*
  - *Tutor-tutee matching*
  - *What's in my fridge and what can I cook?*
  - *Hearsay: manage your own musics*
  - *Dining at Duke (and deliver meals to students)*
  - *National Parklopedia*: a website to find information about national parks
- *Project-details doc will be posted soon*
- *More examples later - but we expect you to be creative with a new idea!*

28

Let's get started!

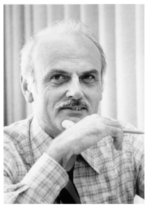
Relational Data Model

What is a good model to store data?
Tree? Nested data? Graph?

(just) Tables!

29

Edgar F. Codd (1923-2003)

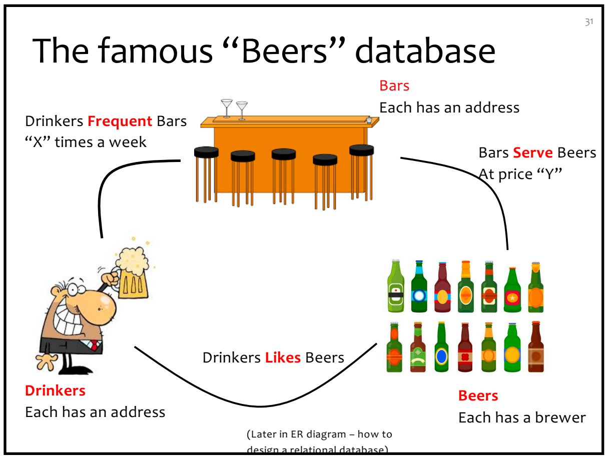


- Pilot in the Royal Air Force in WW2
- Inventor of the relational model and algebra while at IBM
- Turing Award, 1981

RDBMS = Relational DBMS

http://en.wikipedia.org/wiki/File:Edgar_F_Codd.jpg

30



31

The famous "Beers" database

Bars
Each has an address

Drinkers **Frequent** Bars
"X" times a week

Bars **Serve** Beers
At price "Y"

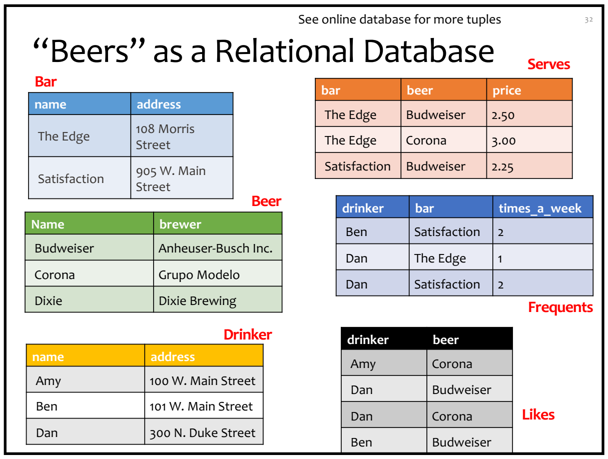
Drinkers **Likes** Beers

Drinkers
Each has an address

Beers
Each has a brewer

(Later in ER diagram – how to design a relational database)

31



See online database for more tuples

32

"Beers" as a Relational Database

Bar

name	address
The Edge	108 Morris Street
Satisfaction	905 W. Main Street

Serves

bar	beer	price
The Edge	Budweiser	2.50
The Edge	Corona	3.00
Satisfaction	Budweiser	2.25

Beer

Name	brewer
Budweiser	Anheuser-Busch Inc.
Corona	Grupo Modelo
Dixie	Dixie Brewing

drinker	bar	times_a_week
Ben	Satisfaction	2
Dan	The Edge	1
Dan	Satisfaction	2

Frequents

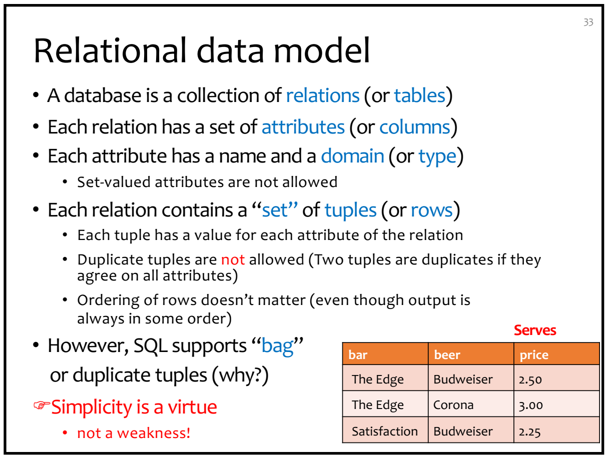
Drinker

name	address
Amy	100 W. Main Street
Ben	101 W. Main Street
Dan	300 N. Duke Street

drinker	beer
Amy	Corona
Dan	Budweiser
Dan	Corona
Ben	Budweiser

Likes

32



33

Relational data model

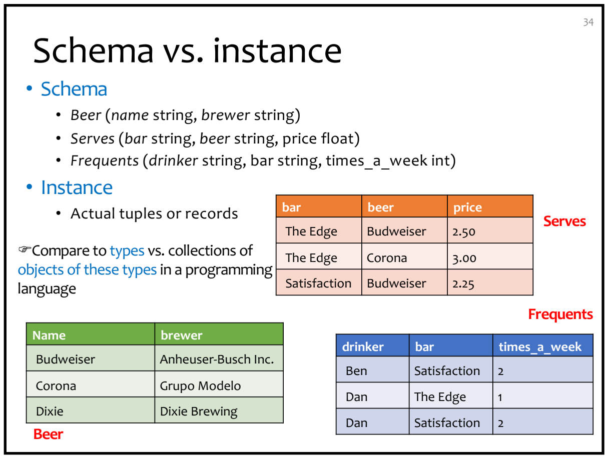
- A database is a collection of relations (or tables)
- Each relation has a set of attributes (or columns)
- Each attribute has a name and a domain (or type)
  - Set-valued attributes are not allowed
- Each relation contains a "set" of tuples (or rows)
  - Each tuple has a value for each attribute of the relation
  - Duplicate tuples are not allowed (Two tuples are duplicates if they agree on all attributes)
  - Ordering of rows doesn't matter (even though output is always in some order)
- However, SQL supports "bag" or duplicate tuples (why?)

☞Simplicity is a virtue
- not a weakness!

Serves

bar	beer	price
The Edge	Budweiser	2.50
The Edge	Corona	3.00
Satisfaction	Budweiser	2.25

33



34

Schema vs. instance

- Schema
  - *Beer* (*name* string, *brewer* string)
  - *Serves* (*bar* string, *beer* string, price float)
  - *Frequents* (*drinker* string, bar string, times_a_week int)
- Instance
  - Actual tuples or records

☞Compare to types vs. collections of objects of these types in a programming language

bar	beer	price
The Edge	Budweiser	2.50
The Edge	Corona	3.00
Satisfaction	Budweiser	2.25

Serves

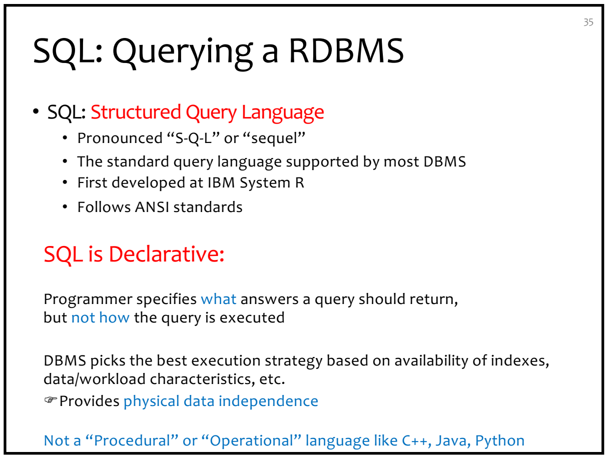
Frequents

Name	brewer
Budweiser	Anheuser-Busch Inc.
Corona	Grupo Modelo
Dixie	Dixie Brewing

Beer

drinker	bar	times_a_week
Ben	Satisfaction	2
Dan	The Edge	1
Dan	Satisfaction	2

34



35

SQL: Querying a RDBMS

- SQL: Structured Query Language
  - Pronounced "S-Q-L" or "sequel"
  - The standard query language supported by most DBMS
  - First developed at IBM System R
  - Follows ANSI standards

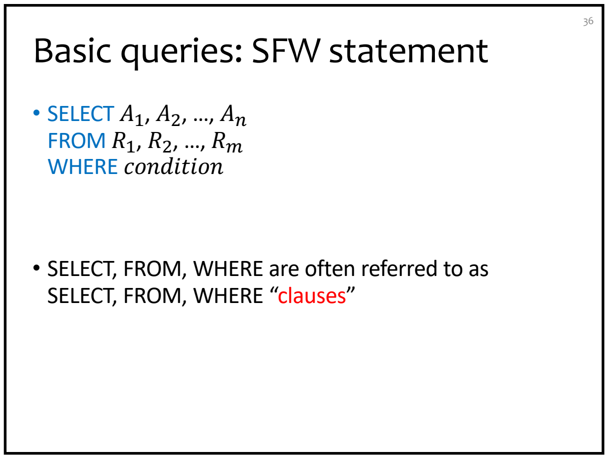
SQL is Declarative:

Programmer specifies what answers a query should return,
but not how the query is executed

DBMS picks the best execution strategy based on availability of indexes, data/workload characteristics, etc.
☞Provides physical data independence

Not a "Procedural" or "Operational" language like C++, Java, Python

35



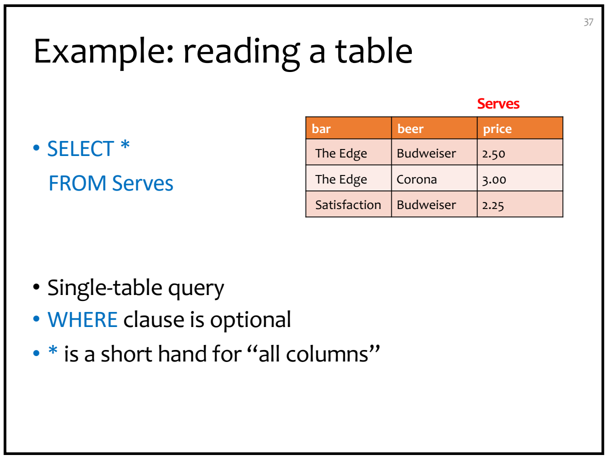
36

Basic queries: SFW statement

- SELECT $A_1, A_2, \dots, A_n$
  FROM $R_1, R_2, \dots, R_m$
  WHERE *condition*

- SELECT, FROM, WHERE are often referred to as SELECT, FROM, WHERE "clauses"

36



Example: reading a table

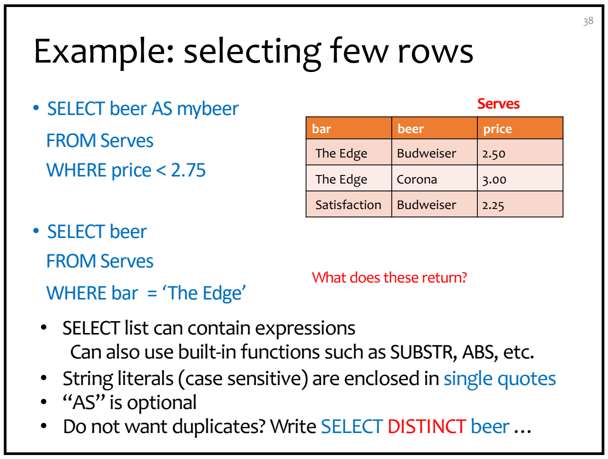
- SELECT *
  FROM Serves

Serves

bar	beer	price
The Edge	Budweiser	2.50
The Edge	Corona	3.00
Satisfaction	Budweiser	2.25

- Single-table query
- WHERE clause is optional
- * is a short hand for “all columns”

37



Example: selecting few rows

- SELECT beer AS mybeer
  FROM Serves
  WHERE price < 2.75

- SELECT beer
  FROM Serves
  WHERE bar = ‘The Edge’

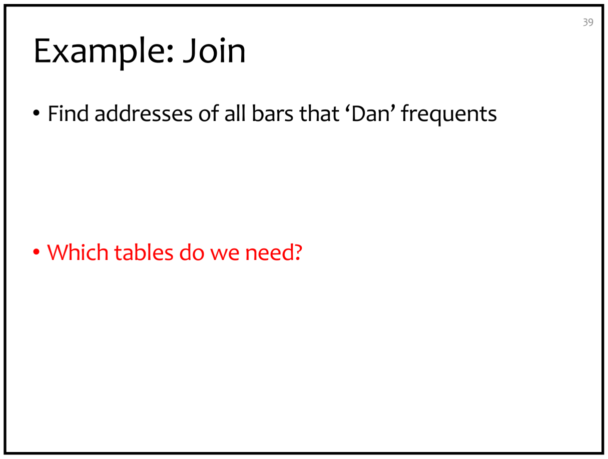
Serves

bar	beer	price
The Edge	Budweiser	2.50
The Edge	Corona	3.00
Satisfaction	Budweiser	2.25

What does these return?

- SELECT list can contain expressions
  Can also use built-in functions such as SUBSTR, ABS, etc.
- String literals (case sensitive) are enclosed in single quotes
- “AS” is optional
- Do not want duplicates? Write SELECT DISTINCT beer …

38

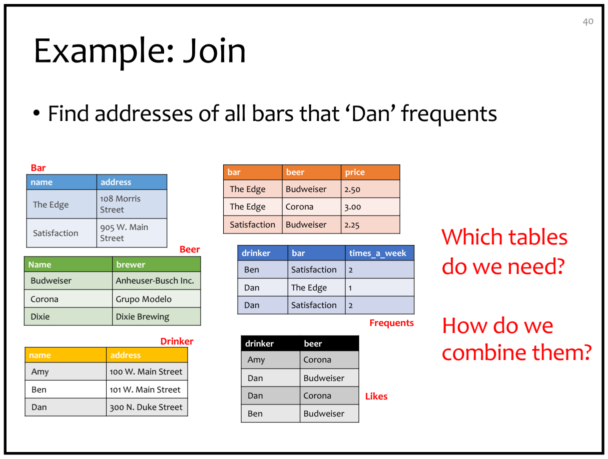


Example: Join

- Find addresses of all bars that ‘Dan’ frequents

- Which tables do we need?

39



Example: Join

- Find addresses of all bars that ‘Dan’ frequents

Bar

name	address
The Edge	108 Morris Street
Satisfaction	905 W. Main Street

bar	beer	price
The Edge	Budweiser	2.50
The Edge	Corona	3.00
Satisfaction	Budweiser	2.25

Beer

Name	brewer
Budweiser	Anheuser-Busch Inc.
Corona	Grupo Modelo
Dixie	Dixie Brewing

drinker	bar	times_a_week
Ben	Satisfaction	2
Dan	The Edge	1
Dan	Satisfaction	2

Frequents

Drinker

name	address
Amy	100 W. Main Street
Ben	101 W. Main Street
Dan	300 N. Duke Street

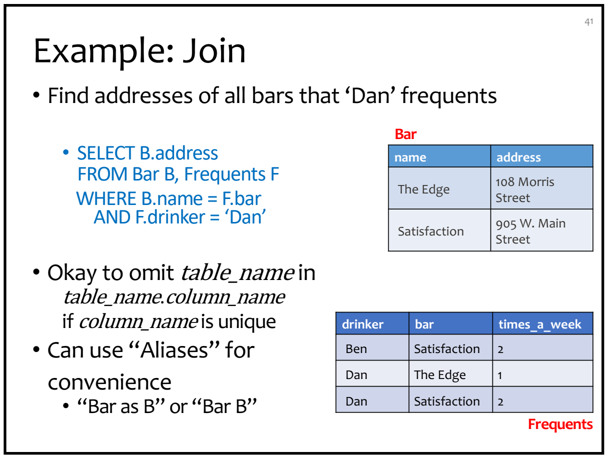
drinker	beer
Amy	Corona
Dan	Budweiser
Dan	Corona
Ben	Budweiser

Likes

Which tables do we need?

How do we combine them?

40



Example: Join

- Find addresses of all bars that ‘Dan’ frequents
  - SELECT B.address
    FROM Bar B, Frequents F
    WHERE B.name = F.bar
    AND F.drinker = ‘Dan’

Bar

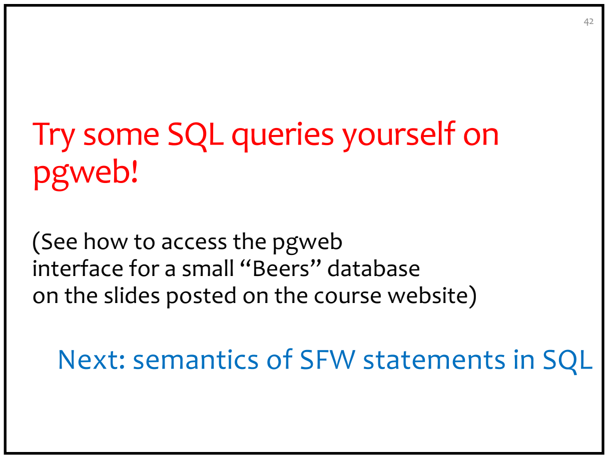
name	address
The Edge	108 Morris Street
Satisfaction	905 W. Main Street

- Okay to omit *table_name* in *table_name.column_name* if *column_name* is unique
- Can use “Aliases” for convenience
  - “Bar as B” or “Bar B”

drinker	bar	times_a_week
Ben	Satisfaction	2
Dan	The Edge	1
Dan	Satisfaction	2

Frequents

41

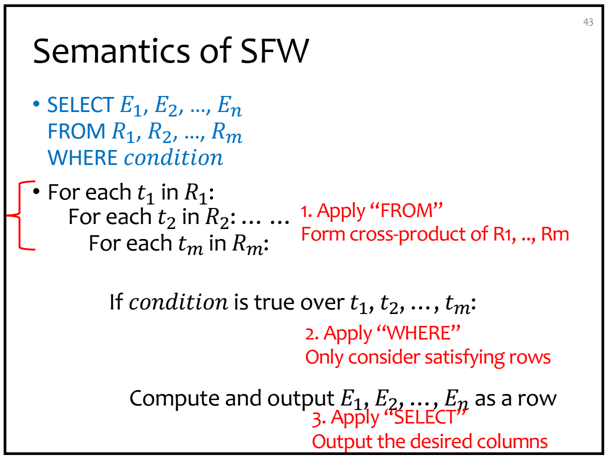


Try some SQL queries yourself on pgweb!

(See how to access the pgweb interface for a small “Beers” database on the slides posted on the course website)

Next: semantics of SFW statements in SQL

42



Semantics of SFW

- SELECT $E_1, E_2, \ldots, E_n$
  FROM $R_1, R_2, \ldots, R_m$
  WHERE *condition*
- For each t_1 in R_1:
  For each t_2 in R_2: … …
  For each t_m in R_m:

1. Apply “FROM”
Form cross-product of R1, .., Rm

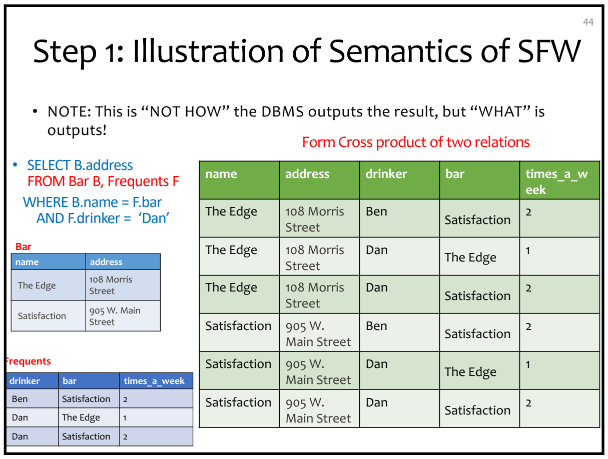
If *condition* is true over $t_1, t_2, \ldots, t_m$:

2. Apply “WHERE”
Only consider satisfying rows

Compute and output $E_1, E_2, \ldots, E_n$ as a row

3. Apply “SELECT”
Output the desired columns

43



Step 1: Illustration of Semantics of SFW

- NOTE: This is “NOT HOW” the DBMS outputs the result, but “WHAT” is outputs!

- SELECT B.address
  FROM Bar B, Frequents F
  WHERE B.name = F.bar
  AND F.drinker = ‘Dan’

Bar

name	address
The Edge	108 Morris Street
Satisfaction	905 W. Main Street

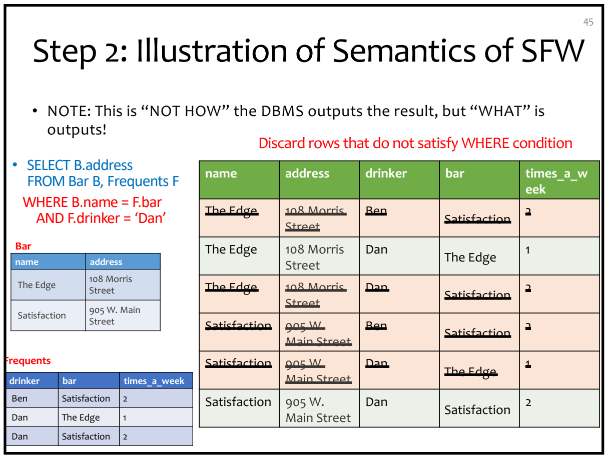
Frequents

drinker	bar	times_a_week
Ben	Satisfaction	2
Dan	The Edge	1
Dan	Satisfaction	2

Form Cross product of two relations

name	address	drinker	bar	times_a_week
The Edge	108 Morris Street	Ben	Satisfaction	2
The Edge	108 Morris Street	Dan	The Edge	1
The Edge	108 Morris Street	Dan	Satisfaction	2
Satisfaction	905 W. Main Street	Ben	Satisfaction	2
Satisfaction	905 W. Main Street	Dan	The Edge	1
Satisfaction	905 W. Main Street	Dan	Satisfaction	2

44



Step 2: Illustration of Semantics of SFW

- NOTE: This is “NOT HOW” the DBMS outputs the result, but “WHAT” is outputs!

- SELECT B.address
  FROM Bar B, Frequents F
  WHERE B.name = F.bar
  AND F.drinker = ‘Dan’

Bar

name	address
The Edge	108 Morris Street
Satisfaction	905 W. Main Street

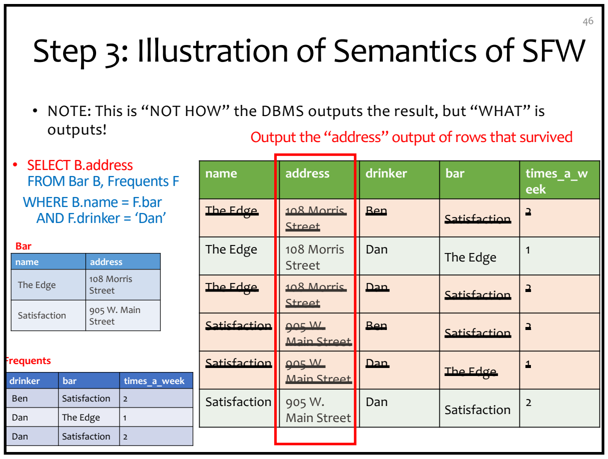
Frequents

drinker	bar	times_a_week
Ben	Satisfaction	2
Dan	The Edge	1
Dan	Satisfaction	2

Discard rows that do not satisfy WHERE condition

name	address	drinker	bar	times_a_week
~~The Edge~~	~~108 Morris Street~~	~~Ben~~	~~Satisfaction~~	~~2~~
The Edge	108 Morris Street	Dan	The Edge	1
~~The Edge~~	~~108 Morris Street~~	~~Dan~~	~~Satisfaction~~	~~2~~
~~Satisfaction~~	~~905 W. Main Street~~	~~Ben~~	~~Satisfaction~~	~~2~~
~~Satisfaction~~	~~905 W. Main Street~~	~~Dan~~	~~The Edge~~	~~1~~
Satisfaction	905 W. Main Street	Dan	Satisfaction	2

45



Step 3: Illustration of Semantics of SFW

- NOTE: This is “NOT HOW” the DBMS outputs the result, but “WHAT” is outputs!

- SELECT B.address
  FROM Bar B, Frequents F
  WHERE B.name = F.bar
  AND F.drinker = ‘Dan’

Bar

name	address
The Edge	108 Morris Street
Satisfaction	905 W. Main Street

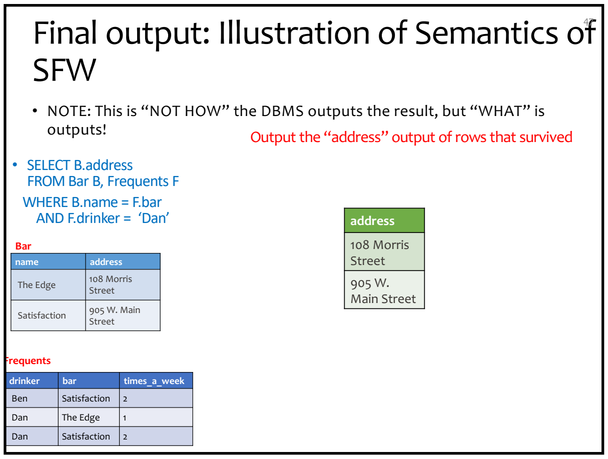
Frequents

drinker	bar	times_a_week
Ben	Satisfaction	2
Dan	The Edge	1
Dan	Satisfaction	2

Output the “address” output of rows that survived

name	address	drinker	bar	times_a_week
~~The Edge~~	~~108 Morris Street~~	~~Ben~~	~~Satisfaction~~	~~2~~
The Edge	108 Morris Street	Dan	The Edge	1
~~The Edge~~	~~108 Morris Street~~	~~Dan~~	~~Satisfaction~~	~~2~~
~~Satisfaction~~	~~905 W. Main Street~~	~~Ben~~	~~Satisfaction~~	~~2~~
~~Satisfaction~~	~~905 W. Main Street~~	~~Dan~~	~~The Edge~~	~~1~~
Satisfaction	905 W. Main Street	Dan	Satisfaction	2

46



Final output: Illustration of Semantics of SFW

- NOTE: This is “NOT HOW” the DBMS outputs the result, but “WHAT” is outputs!

- SELECT B.address
  FROM Bar B, Frequents F
  WHERE B.name = F.bar
  AND F.drinker = ‘Dan’

Bar

name	address
The Edge	108 Morris Street
Satisfaction	905 W. Main Street

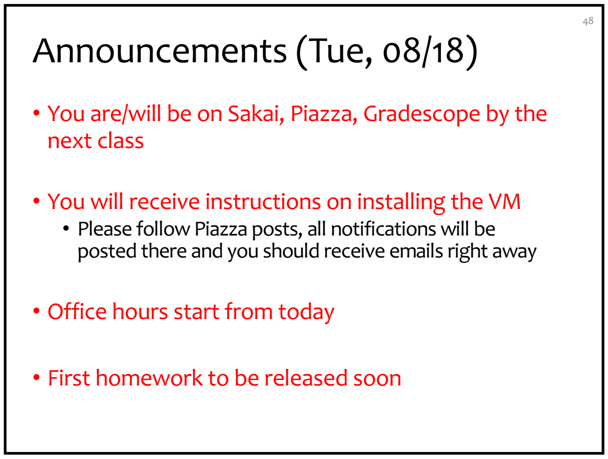
Frequents

drinker	bar	times_a_week
Ben	Satisfaction	2
Dan	The Edge	1
Dan	Satisfaction	2

Output the “address” output of rows that survived

address
108 Morris Street
905 W. Main Street

47



Announcements (Tue, 08/18)

- You are/will be on Sakai, Piazza, Gradescope by the next class
- You will receive instructions on installing the VM
  - Please follow Piazza posts, all notifications will be posted there and you should receive emails right away
- Office hours start from today
- First homework to be released soon

48