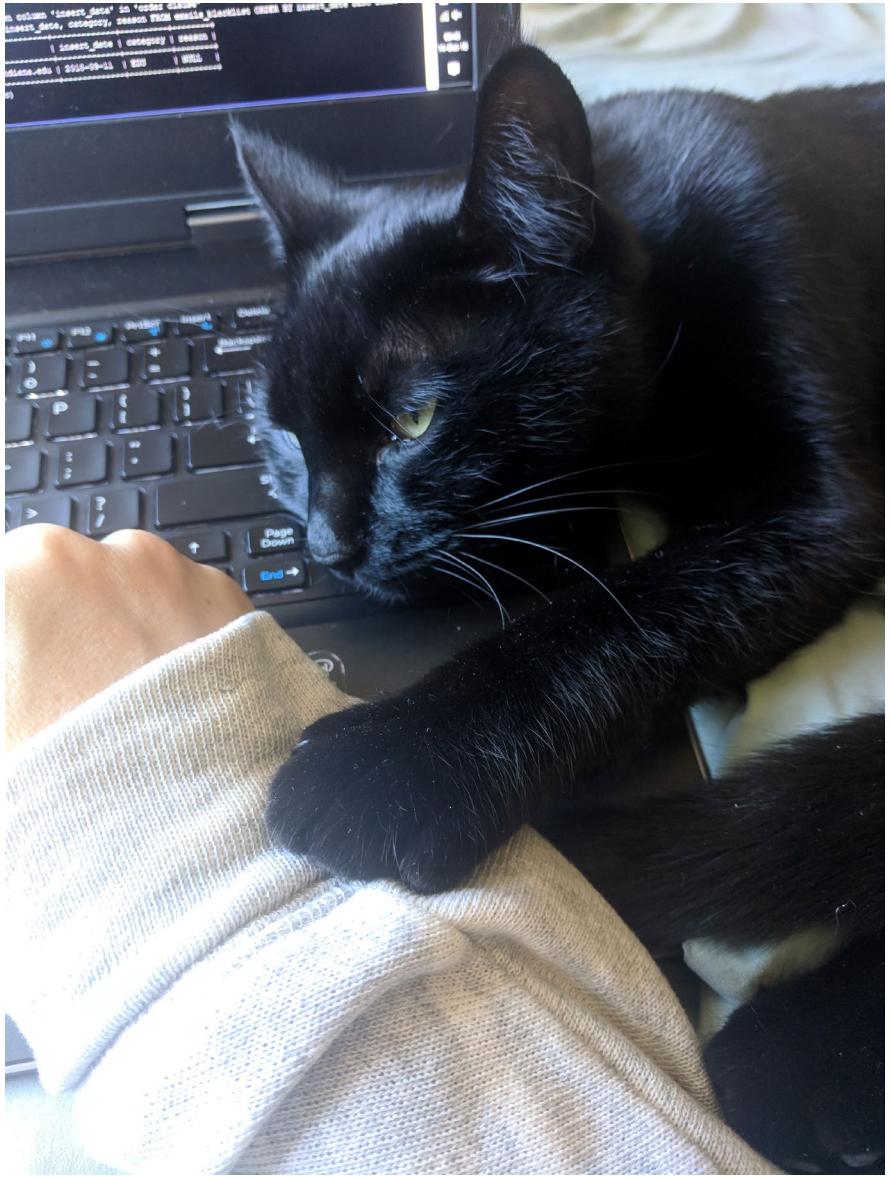


Databases Overview

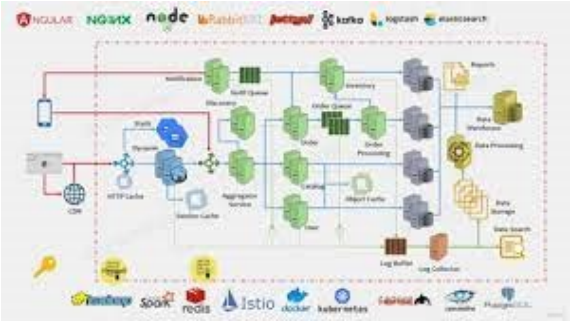
ERIN KEITH

Goals

1. Database Background
2. Types of Databases
3. Database Components

Examples

- web applications
- installed software
- larger scale applications

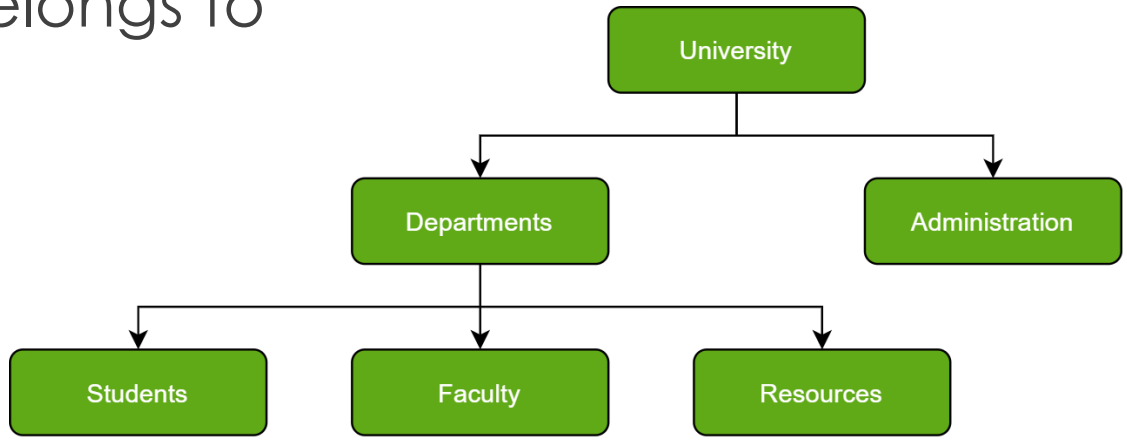


Types of Databases

- Hierarchical databases
- Network databases
- Object-oriented databases
- Relational databases
- Non-relational databases

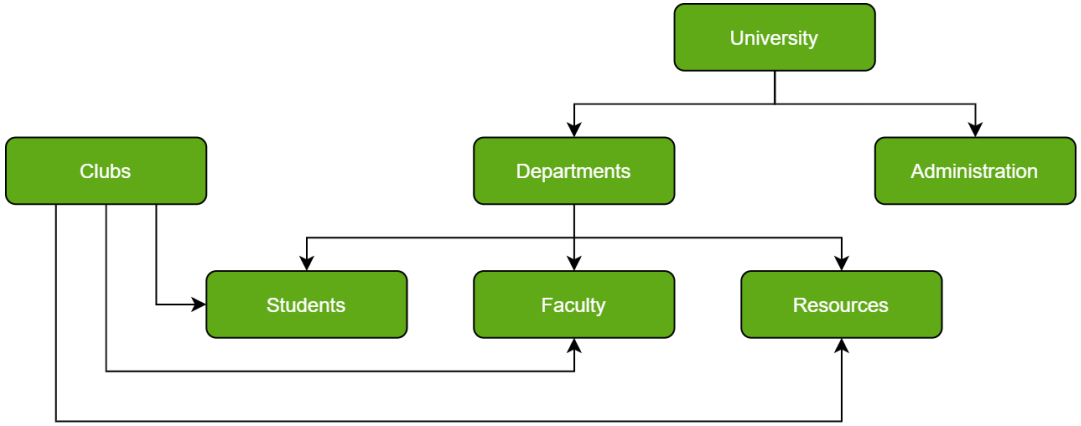
Hierarchical

- Data is categorized as ranks
  - increased commonality has a higher rank
  - “belongs to”



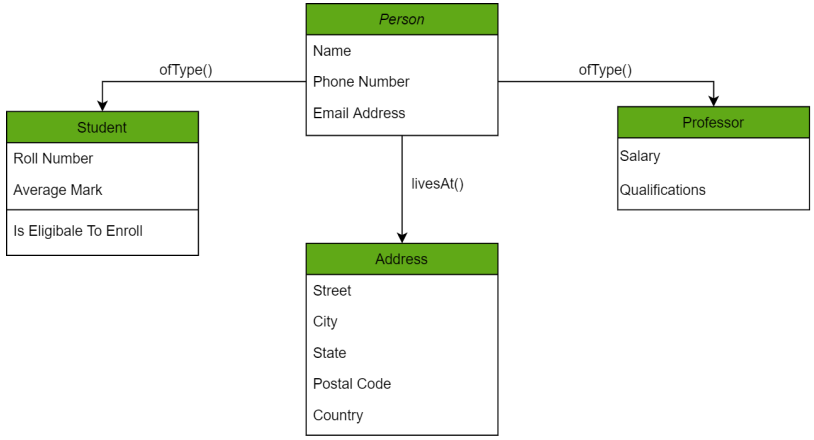
Network

- Hierarchical but with the possibility of multiple parents



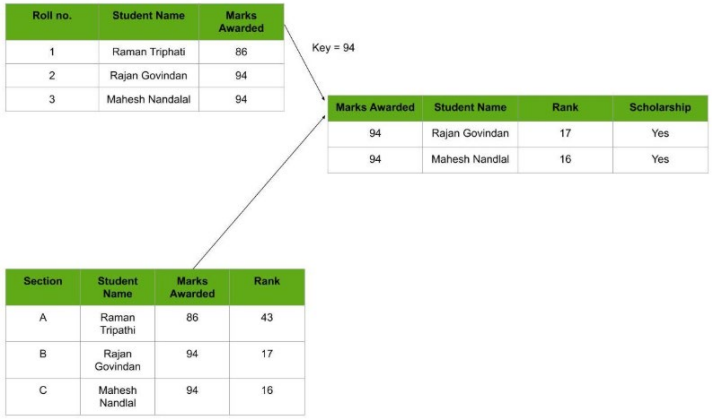
Object-Oriented

- Stores objects with attributes
- Objects have relationships



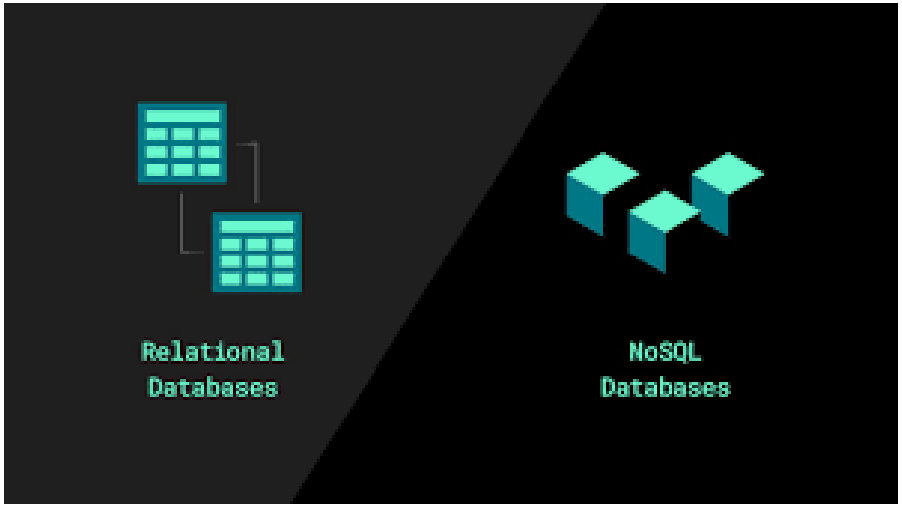
Relational

- Every piece of information has a relationship with every other piece of information
- So much more powerful!



Non-relational

- No relations
- No hierarchy



Database Players

ADMINISTRATORS

Operational

- DBAs
- IT

IS 475

DEVELOPERS

Functional

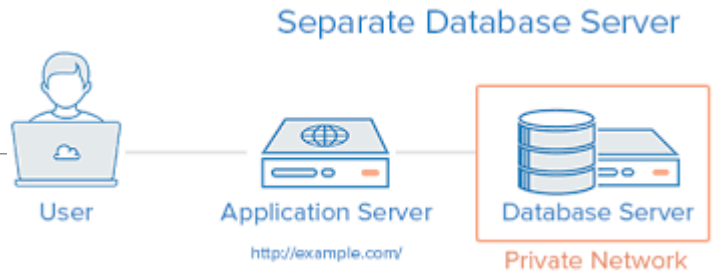
- Software Engineers
- Researchers
- Data Scientists

CS 457

Server / Client

- databases generally run as a separate service
- it's generally referred to as the "server"
  - although it doesn't have to be on a separate machine
- can interact with it directly through
  - the command line
  - a GUI
- write SQL queries or stored procedures to get data from the database
- **this implies you have to "connect" to it**

Connecting

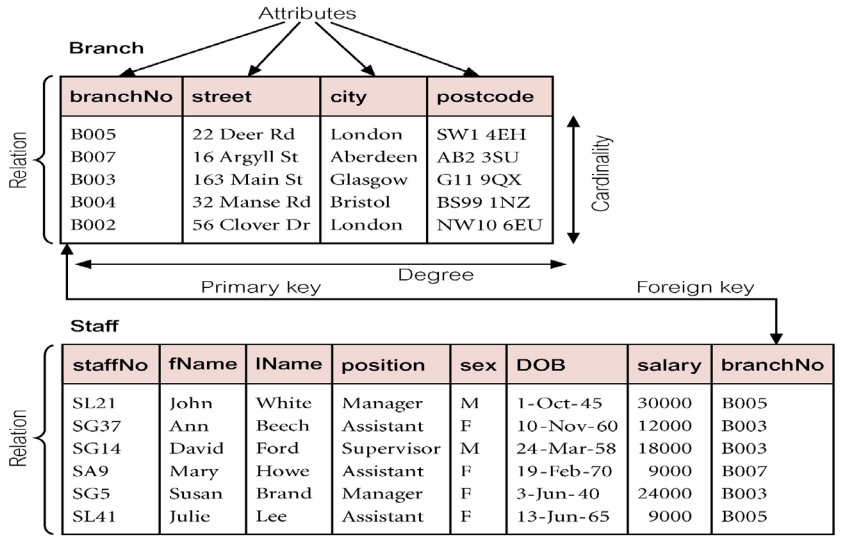


- ODBC
  - Open DataBase Connectivity
  - a standard interface between a SQL database and an application that accesses the data in the database
- ORM
  - Object Relational Mapping
  - creates a "bridge" between object-oriented programs and relational databases

Schema

for relational databases

- map connections between data
- A **schema** is the structure that we define for our data. It defines
  - tables
  - fields
  - relationships between tables
  - indexes



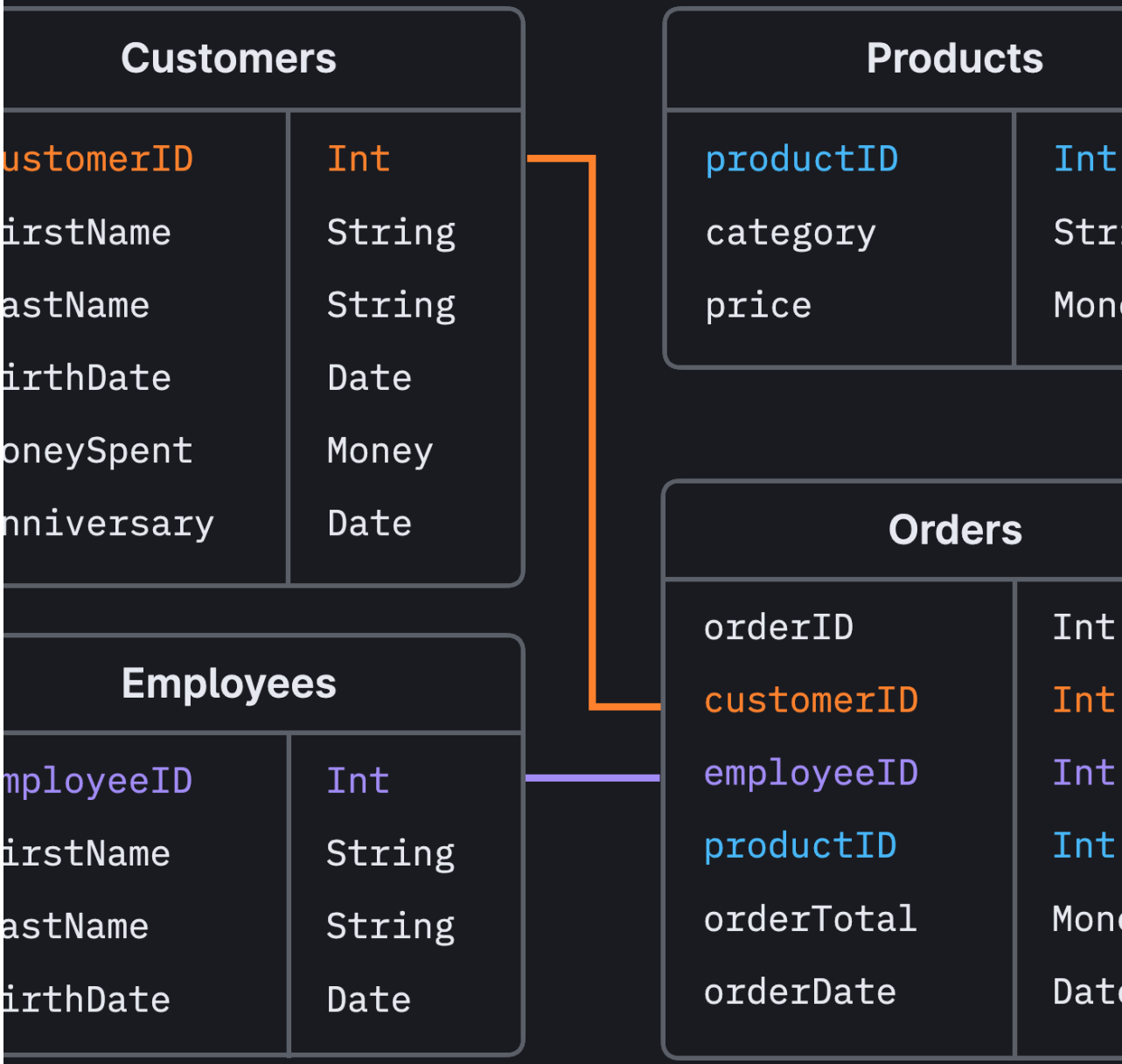
Tables

- central to databases
- data representing an entity organized into columns and rows
  - columns are properties
  - rows are entries in the table

Relationships

between tables

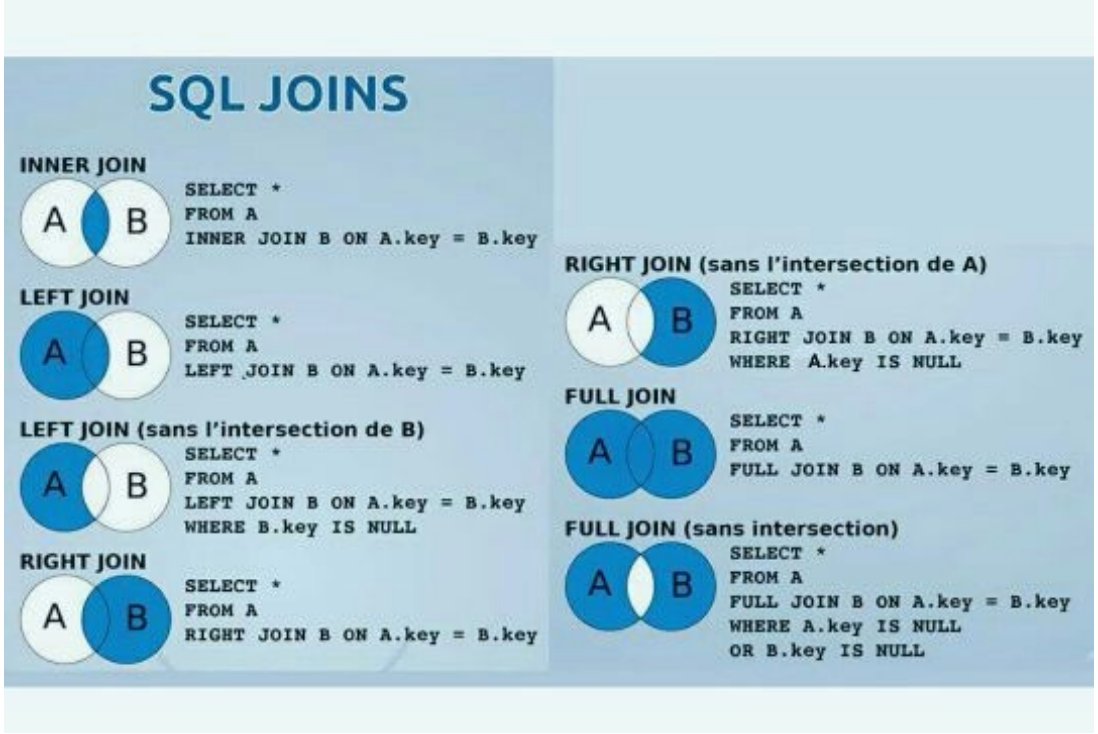
- how the tables "hook" into each other
- columns shared between tables
- normalization



Fields

Fields

- column, property, data field
- each table should have one ***unique identifying property***
  - primary key
    - how the tables "hook" into each other



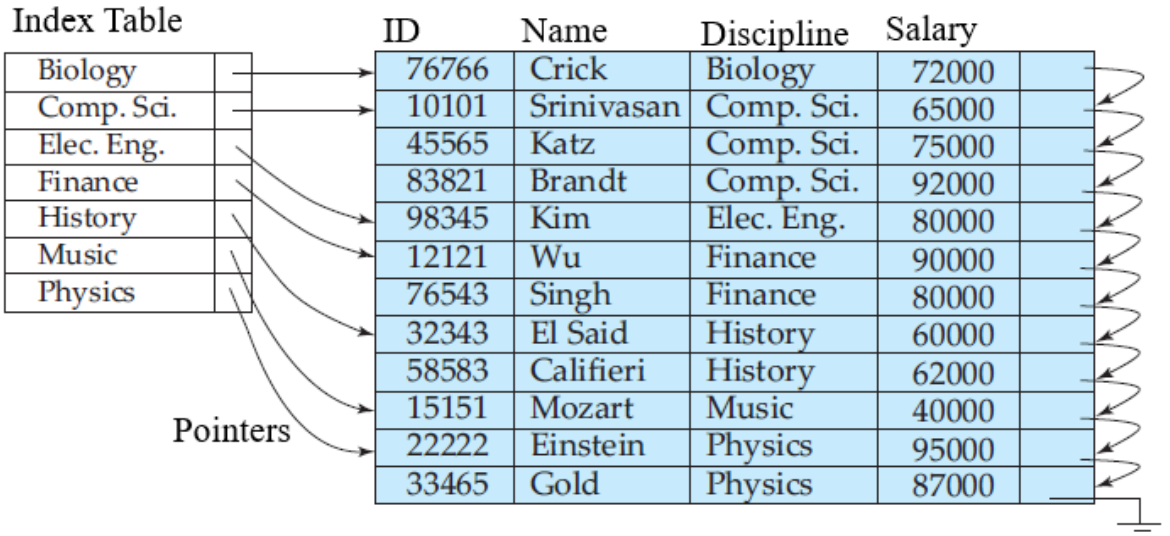
Queries

SQL

- Structured Query Language
- based on relational algebra
- how we get data out of the tables

Indexes

- used behind the scenes to connect tables
- optimizes data retrieval



Indexes

When relations are very large, it becomes expensive to scan all the tuples of a relation to find those (perhaps very few) tuples that match a given condition.

- An index on an attribute **`A`** of a relation is a data structure that makes it efficient to find those tuples that have a fixed value for attribute **`A`**.

Views

Relations that are defined with a CREATE TABLE statement actually "exist" in the database.

Views are relations that are defined by a query over other relations.

- these ***do not*** "exist"
- they ***can be*** queried
- (it's kind of like a temporary table)

Next Class

Module:

Week 2: Background, Ch 2

Topic:

Storage

The Relational Model