Introduction to Database Systems: CS312
Summary

Oliver Bonham-Carter

22 April 2018

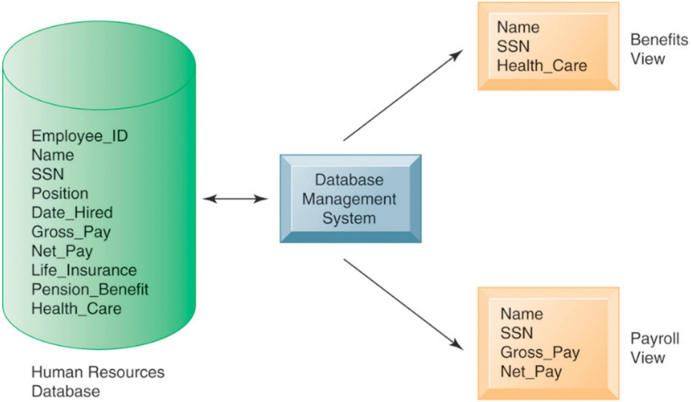
What has this class covered?

- Some of the fundamental theory and methods behind modern databases systems
  - SQL: *Sqlite3*, *DBBrowser*
  - NoSQL: *Mongo*, *Cassandra*
  - Graphical: *Neo4J*
- Building schemas with integrity constrains for data management
- Manipulating data, populating bases and extracted out filtered information
- Programming queries across all DB systems
- Management and Automation: Programming for abstraction
- How to pull information (knowledge) from raw data

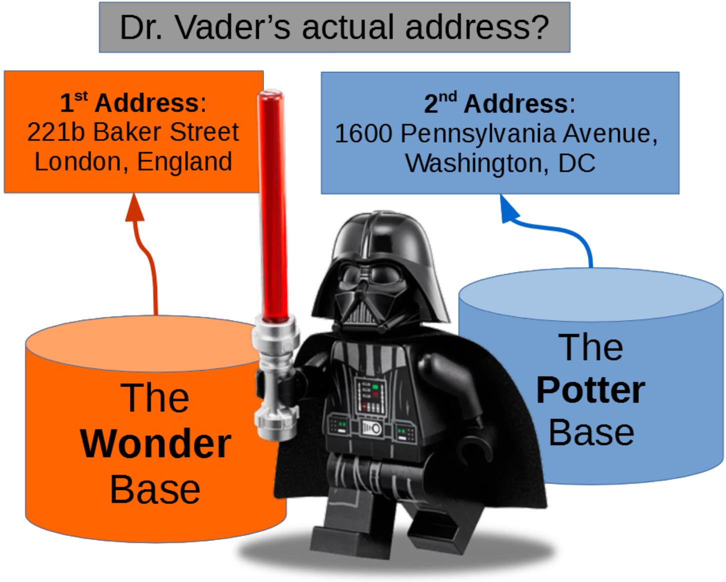
SQL Databases

What is the function of a database?

To Connect Data



Data Disconnections at *Batman and Associates*
Two different address? :-(



Dr. Vader’s actual address?

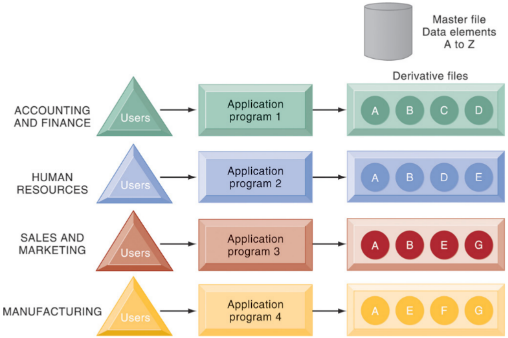
1st Address:
221b Baker Street
London, England

2nd Address:
1600 Pennsylvania Avenue,
Washington, DC

The **Wonder** Base

The **Potter** Base

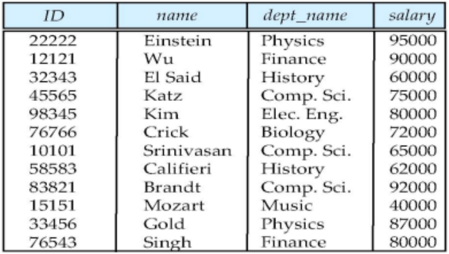
How Many Databases Do You Need??



- A firm may have managed several information sources at the same time
- Anything wrong with disconnecting databases?

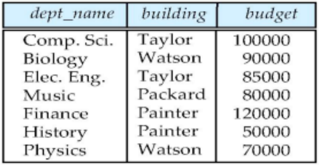
Specific Information For Each Table

Combine and connect the data in one base



ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table



dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

- Specific tables for types of data

SQL Is So Very ...

- Pronounced "ess-que-el" stands for *Structured Query Language*.
- Used to communicate with a database.
- According to ANSI (American National Standards Institute), it is the standard language for relational database management systems.
- The standard computer language for relational database management and data manipulation.
  - Used to query, insert, update and modify data

SQLite3

A practical Open Source Database

Command

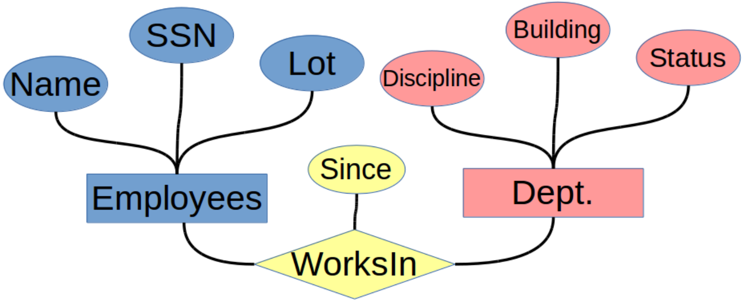
```
$sqlite3
```

You should see this

```
SQLite version 3.11.0 2016-02-15 17:29:24
Enter ”.help” for usage hints.
Connected to a transient in-memory database.
Use ”.open FILENAME” to reopen on a persistent database.
sqlite>
```

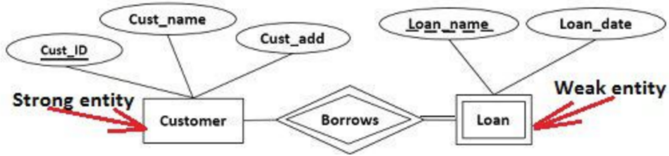
ER Model Basics

Schemas and Relationships



- A schema resembles a subroutine and describes the table and the data that it contains.
- Relationship: An association among two of more entities
- Relationship Set: A collection of similar relationships for entities
- Relationship sets can also have *descriptive attributes* (i.e., the “since” attribute of *WorksIn*)

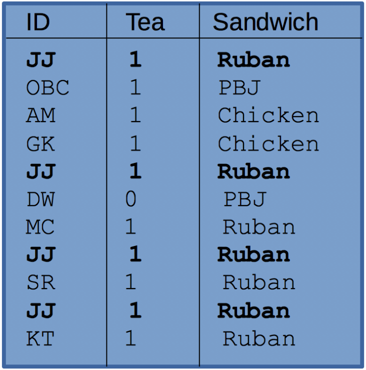
Entity Sets



Entity Sets

- An **entity set** is a set of entities of the same type (e.g., all persons having an account at a bank). Entity sets need not be disjoint. For example, the entity set employee (all employees of a bank) and the entity set customer (all customers of the bank) may have members in common.
- *Weak entities* are those that cannot be uniquely identified by its attributes alone

Entity sets



ID	Tea	Sandwich
JJ	**1**	**Ruban**
OBC	1	PBJ
AM	1	Chicken
GK	1	Chicken
JJ	**1**	**Ruban**
DW	0	PBJ
MC	1	Ruban
JJ	**1**	**Ruban**
SR	1	Ruban
JJ	**1**	**Ruban**
KT	1	Ruban

- **Entity set**: a collection of entities of the same kind
  - (i.e., the preferred sandwiches.)
- Strong Entity sets: Each row is unique in the table.

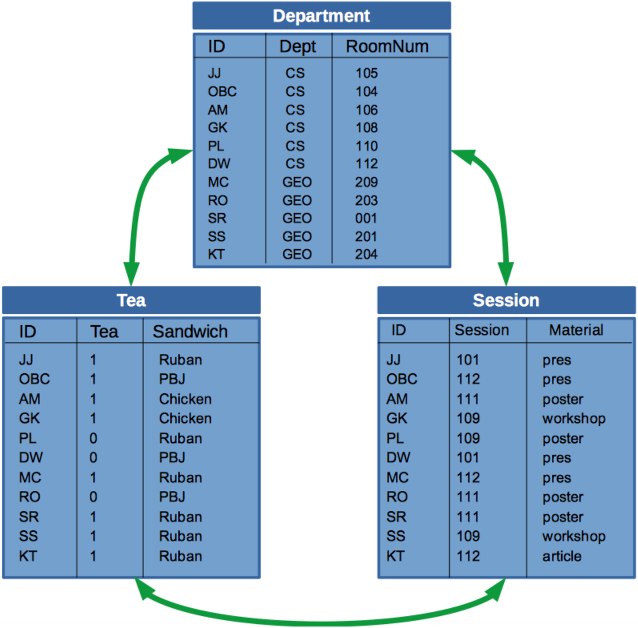
Keys for SQL

- **Primary keys**: Unique identifiers for the row of information sharing a relation (n-tuple).
- **Super keys**: A superkey is a set of attributes within a table whose values can be used to uniquely identify a n-tuple.
- **Candidate keys**: is a minimal set of attributes necessary to identify a n-tuple.
- **SuperKeys**: a set of attributes within a table whose values can be used to uniquely identify a tuple (each row is unique from the other rows)

Keys

You will note the importance of keys once you start storing your data in your own databases!

Linking the tables by queries



Department

ID	Dept	RoomNum
JJ	CS	105
OBC	CS	104
AM	CS	106
GK	CS	108
PL	CS	110
DW	CS	112
MC	GEO	209
RO	GEO	203
SR	GEO	001
SS	GEO	201
KT	GEO	204

Tea

ID	Tea	Sandwich
JJ	1	Ruban
OBC	1	PBJ
AM	1	Chicken
GK	1	Chicken
PL	0	Ruban
DW	0	PBJ
MC	1	Ruban
RO	0	PBJ
SR	1	Ruban
SS	1	Ruban
KT	1	Ruban

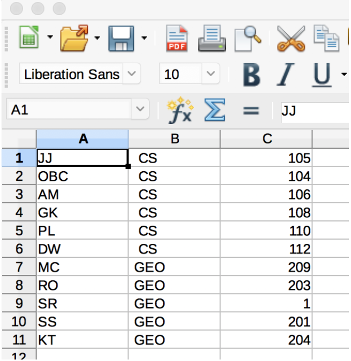
Session

ID	Session	Material
JJ	101	pres
OBC	112	pres
AM	111	poster
GK	109	workshop
PL	109	poster
DW	101	pres
MC	112	pres
RO	111	poster
SR	111	poster
SS	109	workshop
KT	112	article

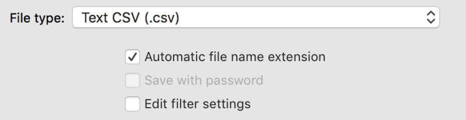
Putting Data into CSV format

- Data as Comma-Separated Values

Making files of CSV's (Comma-separated values)...?



	A	B	C
1	JJ	CS	105
2	OBC	CS	104
3	AM	CS	106
4	GK	CS	108
5	PL	CS	110
6	DW	CS	112
7	MC	GEO	209
8	RO	GEO	203
9	SR	GEO	1
10	SS	GEO	201
11	KT	GEO	204
12			



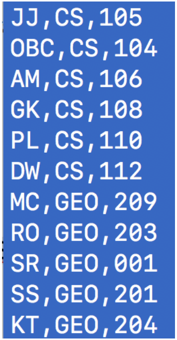
File type: Text CSV (.csv)

☑ Automatic file name extension

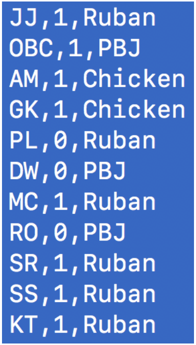
□ Save with password

□ Edit filter settings

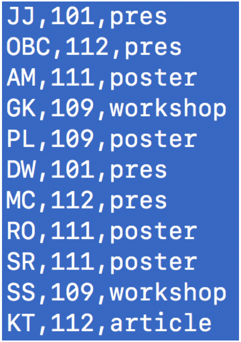
Making files of CSV’s (Comma-separated values)...?



```
JJ,CS,105
OBC,CS,104
AM,CS,106
GK,CS,108
PL,CS,110
DW,CS,112
MC,GEO,209
RO,GEO,203
SR,GEO,001
SS,GEO,201
KT,GEO,204
```



```
JJ,1,Ruban
OBC,1,PBJ
AM,1,Chicken
GK,1,Chicken
PL,0,Ruban
DW,0,PBJ
MC,1,Ruban
RO,0,PBJ
SR,1,Ruban
SS,1,Ruban
KT,1,Ruban
```



```
JJ,101,pres
OBC,112,pres
AM,111,poster
GK,109,workshop
PL,109,poster
DW,101,pres
MC,112,pres
RO,111,poster
SR,111,poster
SS,109,workshop
KT,112,article
```

- Tables: *department*, *tea*, *session*
- Once your file is in this CSV format, it can be easily loaded into the database

The **select** Clause

The SELECT clause filters out particular data from a table.

- SQL allows duplicates in relations as well as in query results.
- The SELECT statement has many optional clauses:
  - WHERE specifies which rows to retrieve.
  - GROUP BY groups rows sharing a property so that an aggregate function can be applied to each group.
  - HAVING selects among the groups defined by the GROUP BY clause.
  - ORDER BY specifies an order in which to return the rows.
  - AS provides an alias which can be used to temporarily rename tables or columns..

Given table 'T'

SELECT

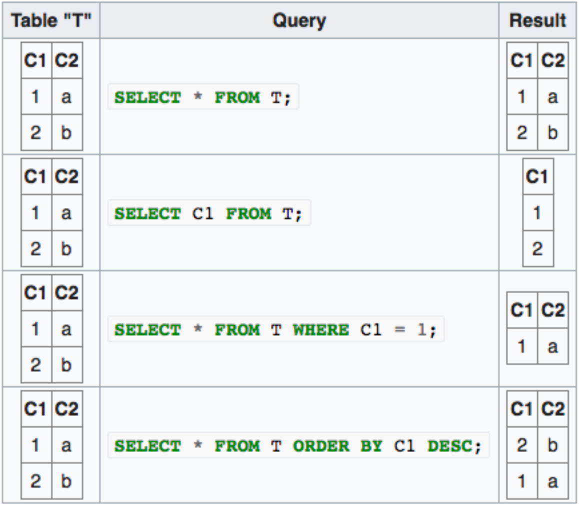


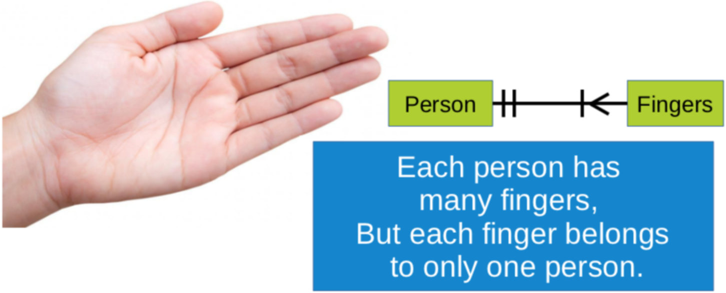
Table "T"	Query	Result
C1 C2 1 a 2 b	`SELECT * FROM T;`	C1 C2 1 a 2 b
C1 C2 1 a 2 b	`SELECT C1 FROM T;`	C1 1 2
C1 C2 1 a 2 b	`SELECT * FROM T WHERE C1 = 1;`	C1 C2 1 a
C1 C2 1 a 2 b	`SELECT * FROM T ORDER BY C1 DESC;`	C1 C2 2 b 1 a

Integrity Constraints

- The CONSTRAINTS are an integrity which defines some conditions that restrict the column to contain the true data while inserting or updating or deleting.
- Integrity constraints provide a mechanism for ensuring that data conforms to guidelines specified by the database administrator. The most common types of constraints include:
  - UNIQUE constraints: To ensure that a given column is unique
  - NOT NULL constraints: To ensure that no null values are allowed
  - FOREIGN KEY constraints: To ensure that two keys share a primary key to foreign key relationship
    - Ensure that a link exists between two tables.

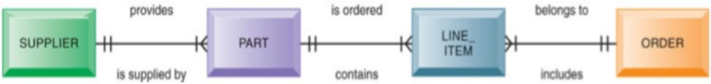
One to many relationships

The *One to Many* Relationship

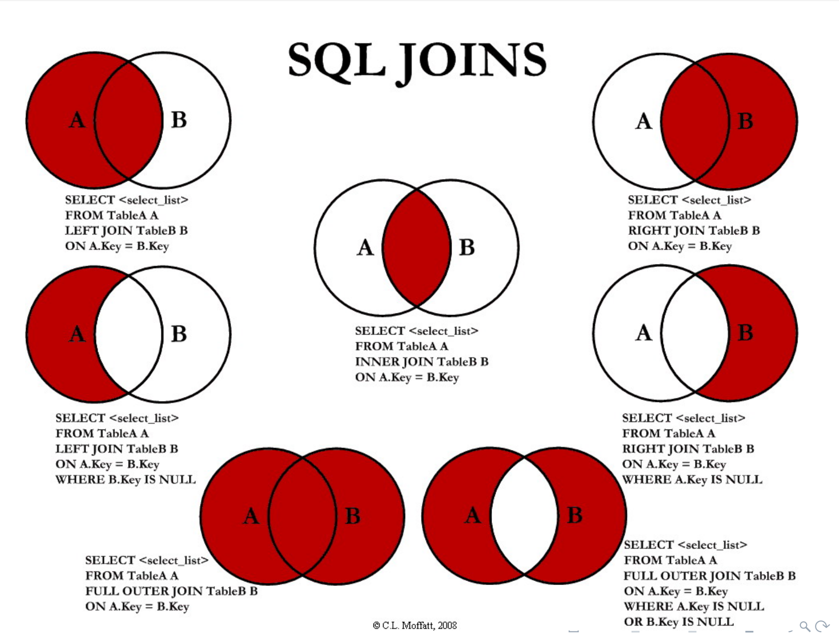


Each person has many fingers,
But each finger belongs to only one person.

FIGURE 6-11 AN ENTITY-RELATIONSHIP DIAGRAM



Joins with SQL code



```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```

```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```

```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```

```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

© C.L. Moffatt, 2008

Django

An easy-to-create web site and online database server

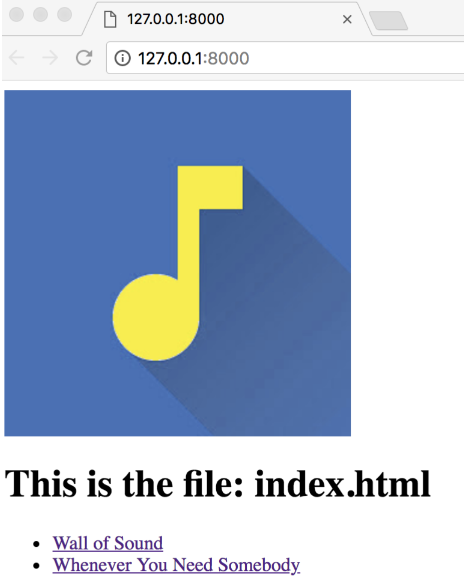
- https://www.djangoproject.com/

The Files of Your App

Notable Files

- **apps.py**: The main file for the *hello* App
- **models.py**: A blueprint for how data will be used in the site
- **tests.py**: For adding tests for bug checking the *hello* part of the project
- **views.py**: A request-handler for connecting the URL to the displayed website
- **mysite/mysite/urls.py**: Requests for apps are all directed using this file.
- **mysite/hello/urls.py**: Requests for the *hello* apps are all directed using this file.

To Make: An online music album viewer



This is the file: index.html

- Wall of Sound
- Whenever You Need Somebody

An Important Song To Add

“Never Gonna Give You Up”

An important link

`https://www.youtube.com/watch?v=dQw4w9WgXcQ`

What is XML?

Extendable Markup Language: A vehicle to move data around

- Unstructured versus structured database
  - Unlike SQL Data held in one major table
  - XML data may be spread-out across a large document
- XML language is flexible
  - Applied to diverse data types, constructions and forms

What Does XML Code Look Like?

Partial XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
   <to>Class</to>
   <from>Community</from>
   <heading>Reminder</heading>
   <body>Don't forget to vote today!</body>
</note>
```

Displaying...

Note

To: Class

From: Community

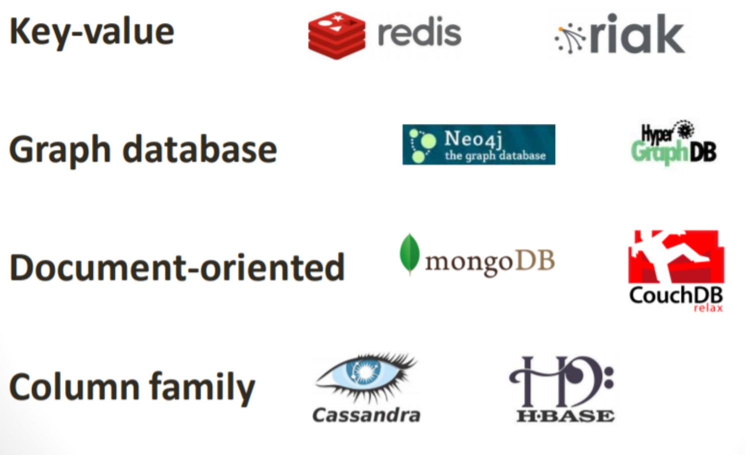
Don’t forget to vote today!

Declaring an Element

- Elements are declared using: <xs:element> tags in the schema file
  - <xs:element name="x" type="y" />
  - <xs:element name="Birthday" type="xs:date" />
- Can be declared as having a simple or complex type
- Elements can have mixed, empty or element context
- Elements can be given a minimum or maximum number of times that they are allowed occur
- Elements restricted to having specific values (similar to *integrity constraints*)

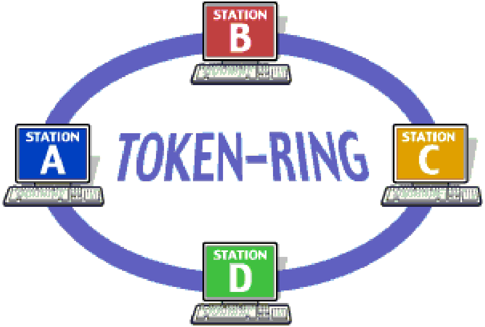
NoSQL: Another Type of Database

"Not only SQL" (so much more to offer!)



- Different types of NoSQL databases

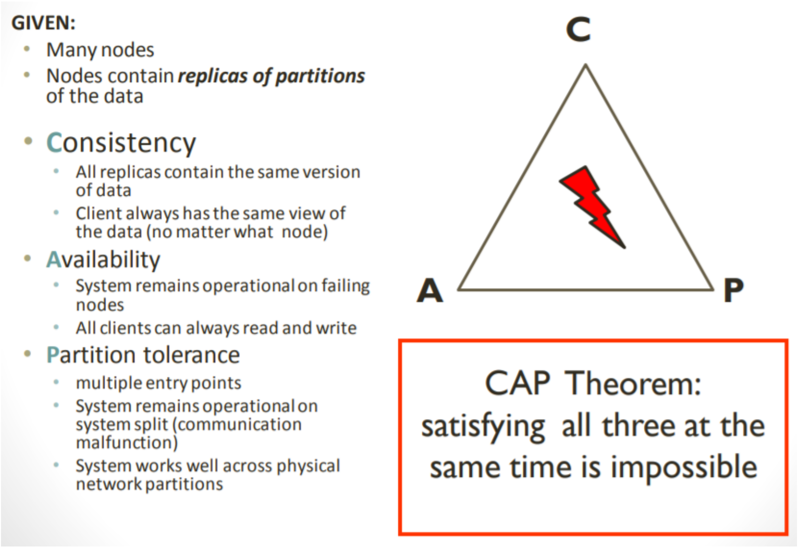
An Architecture For NoSQL Database Connectivity



Token Ring Networks

- Early networks established foundational database configurations for wide-spread usage
- All devices on the ring share data and update each other
- Hashing function maps each key to a server (node)

CAP Theory



GIVEN:

- Many nodes
- Nodes contain ***replicas of partitions*** of the data

- Consistency
  - All replicas contain the same version of data
  - Client always has the same view of the data (no matter what node)
- Availability
  - System remains operational on failing nodes
  - All clients can always read and write
- Partition tolerance
  - multiple entry points
  - System remains operational on system split (communication malfunction)
  - System works well across physical network partitions

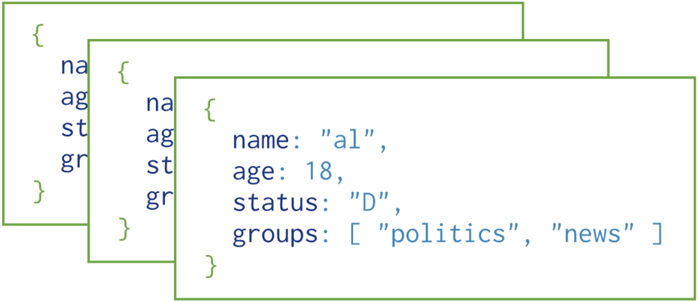
CAP Theorem: satisfying all three at the same time is impossible

A NoSQL Database Management System

(SQLite3 cannot operate here.)

- https://www.mongodb.com/

Schema Free



```
{
   name: "al",
   age: 18,
   status: "D",
   groups: [ "politics", "news" ]
}
```

Collection

- No pre-defined data schema
  - Data may be entered at in absence of a defined schema
- Every document in a collection could have different data

A NoSQL Database Management System

- Apache Cassandra is a massively scalable open source non-relational database
- Offers continuous availability, linear scale performance, operational simplicity and easy data distribution across multiple data centers and cloud availability zones.
- Founded at Facebook 2008, developed at Apache in 2010

- `http://cassandra.apache.org/`
- `https://academy.datastax.com/planet-cassandra/cassandra`

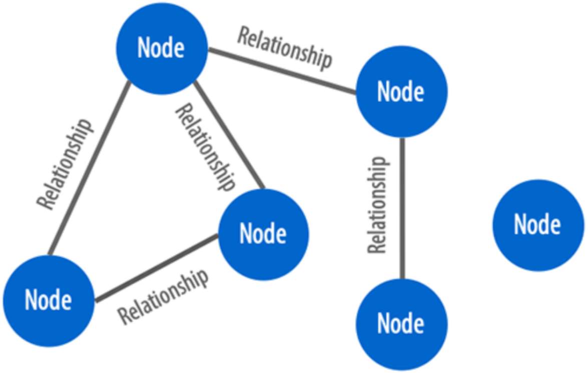
Databases, Visually

- A visual database system using methods from graph theory to use networks to determine relationships (edges) and discover meaning from connected data-points (nodes). Users are able to interact with the data in a network.

- `https://neo4j.com/`
- Graphgists Projects: `https://neo4j.com/graphgists/`

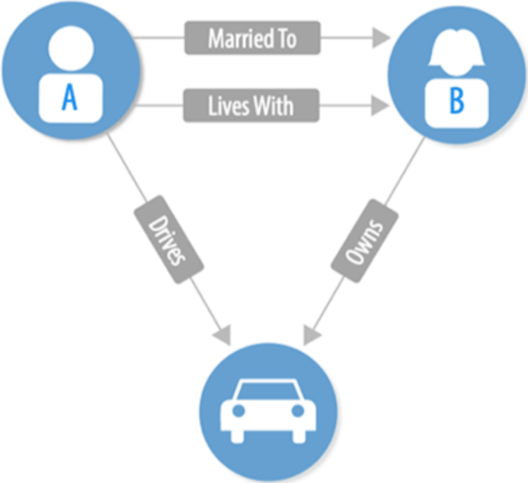
Networks Of Data

Relationships exist by connectivity



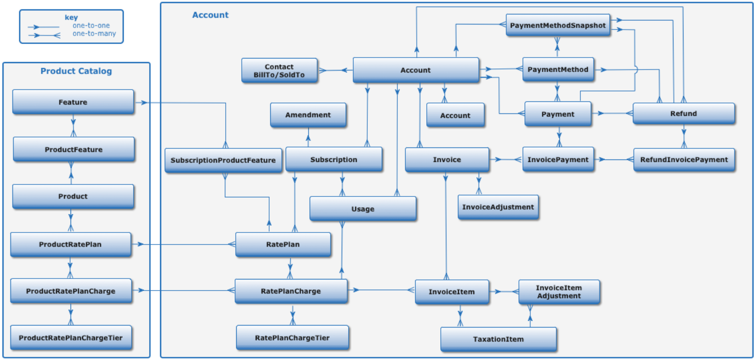
- Nodes and edges represent inter-relationships
- Relationships are described by connections between nodes
- Single nodes have no immediate relationships with the others

Networks In Neo4J



- An acting schema: The relationships between nodes are built into the network

What Has This Class Covered?



What has this class *not* covered?!

(Now go update the skills section of your resumé!)