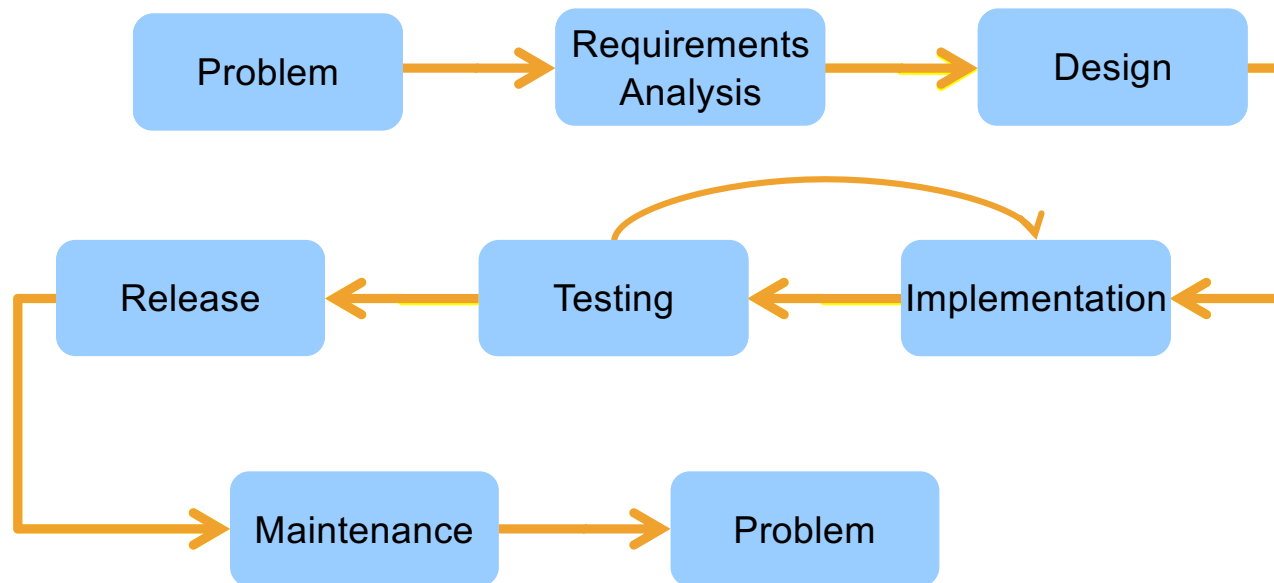# CS 4320 / 7320 Advanced Database Concepts

Maintenance

What is the SDLC?
Where does Maintenance fit?

Data Lakes and Data Warehouses

PSDS 3100
Database Analytics

Fifty Years of Database Technology in a Nutshell

Topics

Relational Databases
    Entities
    Relationships
OLAP Versus OLTP
    Transactions
    Analytics

Cluster Computing
    Big Data
    Spark
NOSQL
Graph Databases
    Aggregation is Out
    Connection is In

What is a database?

Wikipedia says:
> "A database is an organized collection of data. The data are typically organized to model relevant aspects of reality in a way that supports processes requiring this information."

Database Design

- What are the data entities of the system?
- What are the attributes of each entity?
- What are the constraints on the attributes of the entities?
- What can be used to uniquely identify entities in the system?
- How are the different entities related?

Entities

A person, place or thing about which we want to collect and store multiple instances of data.
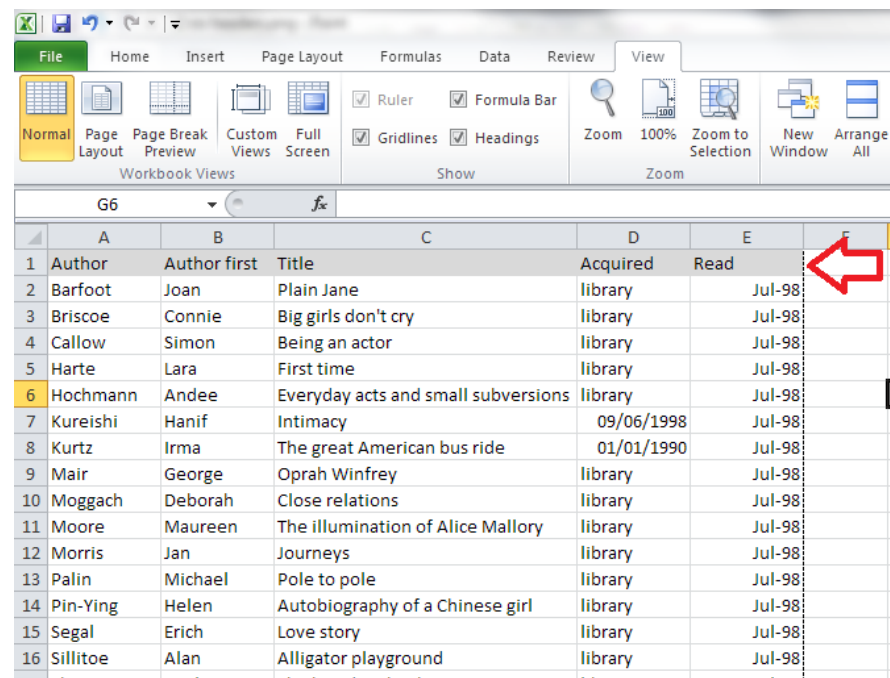Similar to an Object in Object Oriented Design
Think of Entities as nouns
These will be the tables in your database

Attributes

Data that describes the Entities.
These will be the columns of each table in
your database.

Familiar concept?

Constraints

- Specific rules for the Attributes.
- Make sure that the data is consistent.
- In SQL:
  - NOT NULL
  - UNIQUE
  - CHECK
  - DEFAULT

How to access a record?

Primary Key: attribute or combination of attributes that uniquely identify each row in the table.

Relationships

- Entities have some relationship to other entities in the system
- Illustrates an association between two entities.
- Cardinality Constraints:
  - Zero or More
  - One or More
  - One and only One
  - Zero or One

ERD: Entity Relationship Diagram

Entity-Relationship Diagram
- Data Model to describe a database in an abstract way
- Visually shows the relationships of data within the system
- We will use Crow's Foot Notation, because that is what real-world professionals use

ERD Example

STUDENT
*student_id
student_name
student_address

SEAT
*seat_no
seat position

fills

takes

INSTRUCTOR
*instructor_no
instructor_name
instructor_faculty

teaches

COURSE
*course_name
*course_number

has

SECTION
*section_number

teaches

PROFESSOR
*professor_id
professor_name
professor_faculty

Can I code this exactly as is??
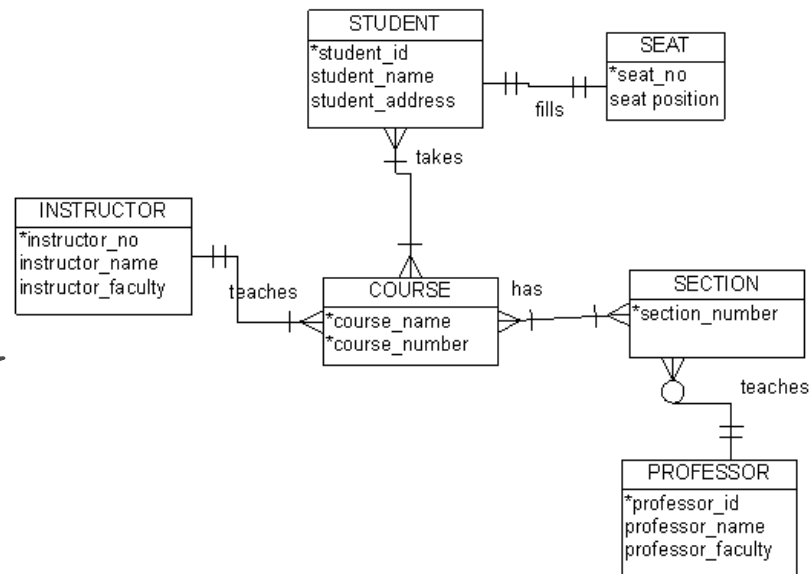
```
CREATE TABLE Student (
    student_id INT,
    student_name VARCHAR(30),
    student_address VARCHAR(40),
    PRIMARY KEY (student_id)
)


CREATE TABLE Course (
    course_name VARCHAR(20),
    course_number INT,
    PRIMARY KEY (course_name, course_number)
)
```
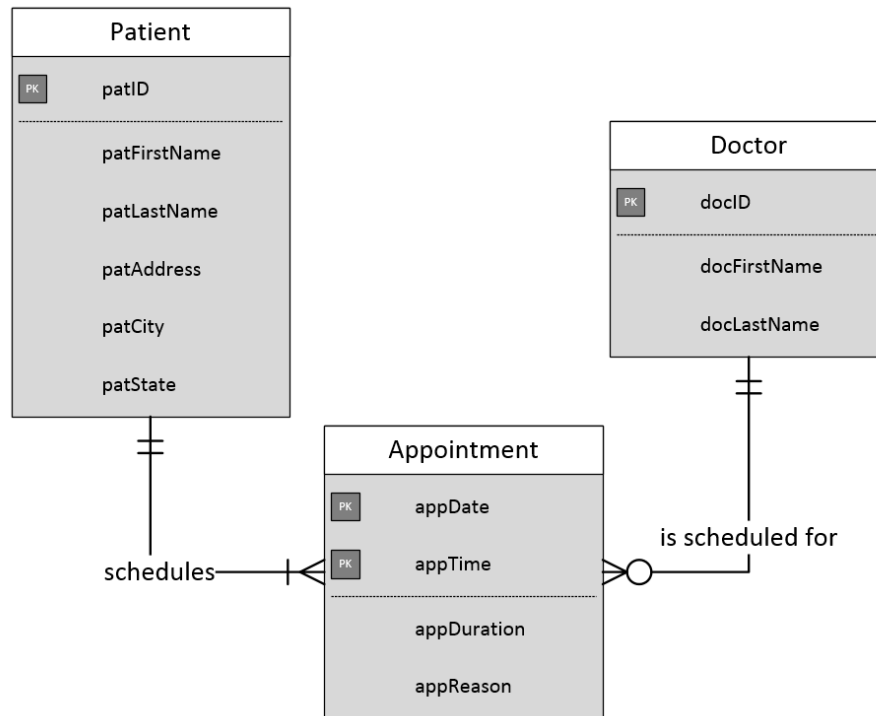
```sql
CREATE TABLE Enrollment (
    student_id INT,
    course_name VARCHAR(20),
    course_number INT,
    PRIMARY KEY
      (student_id, course_name, course_number),
    FOREIGN KEY (student_id)
     REFERENCES Student(student_id),
    FOREIGN KEY (course_name, course_number)
     REFERENCES Course(course_name, course_number)
)
```

Example

Think about a system that keeps track of doctor appointments scheduled by patients.

- What would the entities (tables) be?
- What would be the attributes (fields/columns) of each entity?
- What uniquely identifies the entities? (primary keys)
- How are the entities related?

Patient

PK	patID

patFirstName

patLastName

patAddress

patCity

patState

Doctor

PK	docID

docFirstName

docLastName

Appointment

PK	appDate
PK	appTime

appDuration

appReason

schedules

is scheduled for

Data Warehousing

Phase Two of Database Technology:

From Transactions to Analysis of Structured Data

Data Warehouses

Relational : Often Online Transaction Processing (OLTP)
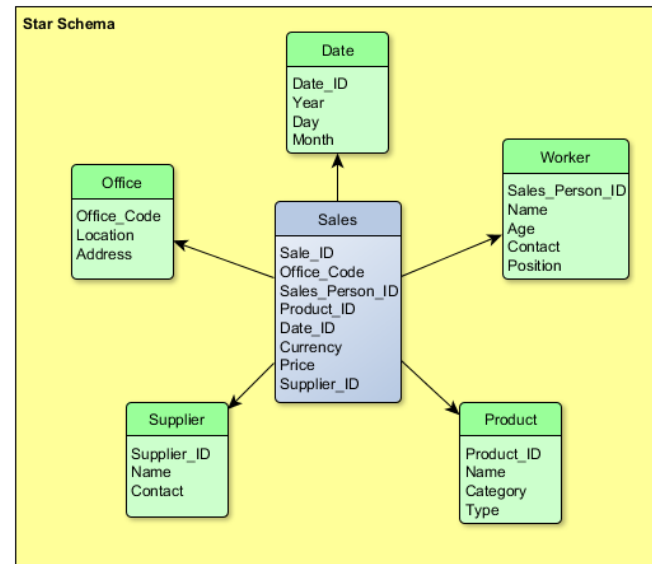
High volume inputs
Data organized to avoid redundancy

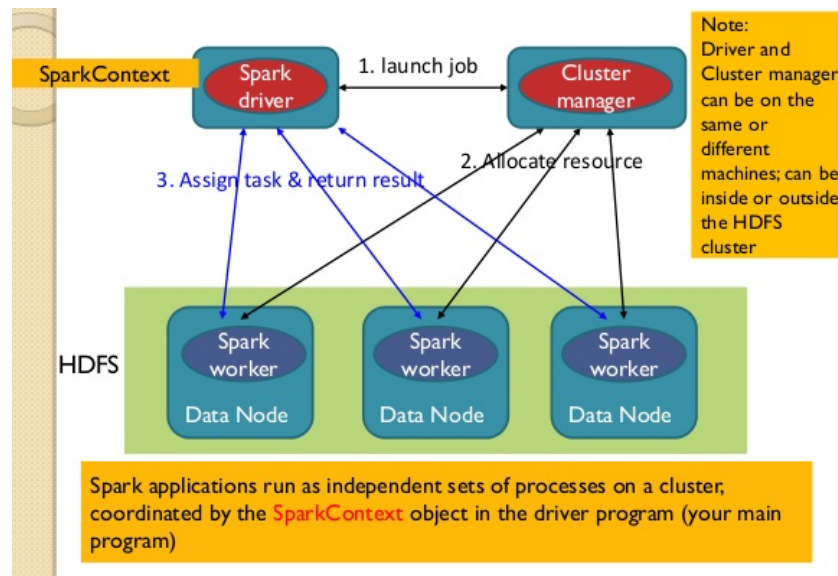Data Warehouses: Often Online Analytical Processing (OLAP)

Decision Support
Designed for Analytical Speed

OLAP Database Design: Star Schema

Cluster Database Computing

Phase Three of Database Technology:

From Reorganizing and Moving Data to Massive Parallel Processing

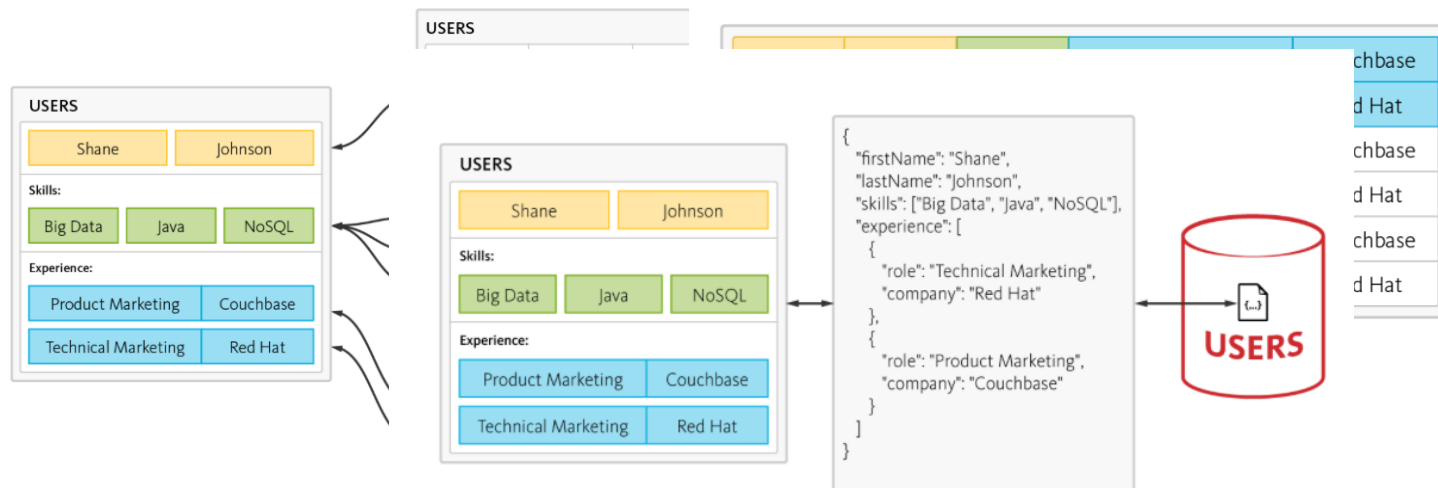# Big Data: Distributed Computing on Clusters

Big Data: Distributed Computing on Clusters

	OPERATIONAL		ANALYTICAL
	Couchbase	Spark	Hadoop
Use cases	• Operational • Web / Mobile	• Analytics • Machine Learning	• Analytics • Machine Learning
Processing mode	• Online • Ad Hoc	• Ad Hoc • Batch • Streaming (+/-)	• Batch • Ad Hoc (+/-)
Low latency =	< 1 ms ops	Seconds	Minutes
Performance	Highly predictable	Variable	Variable
Users are typically…	Millions of customers	100's of analysts or data scientists	100's of analysts or data scientists
	Memory-centric	Memory-centric	Disk-centric
Big data =	10s of Terabytes	Petabytes	Petabytes

NOSQL

USERS

USERS	
Shane	Johnson

Skills:

Big Data	Java	NoSQL

Experience:

Product Marketing	Couchbase
Technical Marketing	Red Hat

USERS

USERS	
Shane	Johnson

Skills:

Big Data	Java	NoSQL

Experience:

Product Marketing	Couchbase
Technical Marketing	Red Hat

```
{
    "firstName": "Shane",
    "lastName": "Johnson",
    "skills": ["Big Data", "Java", "NoSQL"],
    "experience": [
        {
            "role": "Technical Marketing",
            "company": "Red Hat"
        },
        {
            "role": "Product Marketing",
            "company": "Couchbase"
        }
    ]
}
```

USERS

Graph Databases

Representing Everything as a Graph
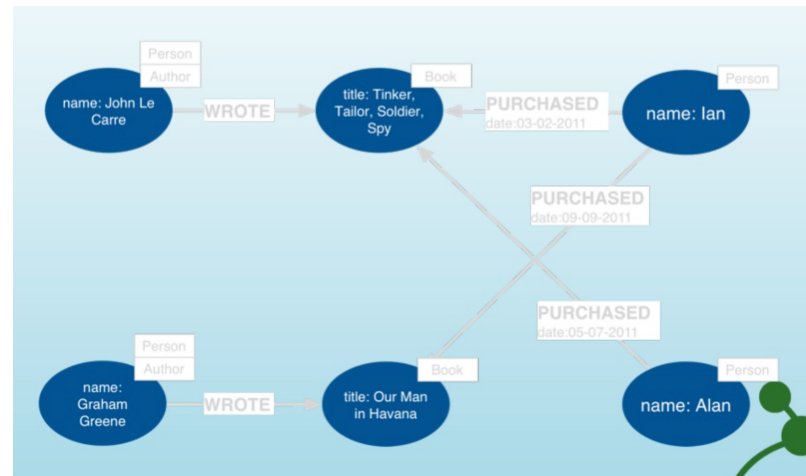
Fifty Years of Database Technology

- OLTP / OLAP: Long History, Built on a relational model

- NOSQL / Spark: High Volume Inputs, Distributed Processing

- Graph Databases: Combining OLTP & OLAP Performance Characteristics in One Technology

Graph Databases

Aggregate v Connected Models: OLAP & OLTP are both "Aggregate" focused. Graph Databases are "Connection Focused"

Graph Building Blocks
- Nodes
  - Entities (tables)
- Relationships
  - Connect to Structure
- Properties
  - Attributes and metadata
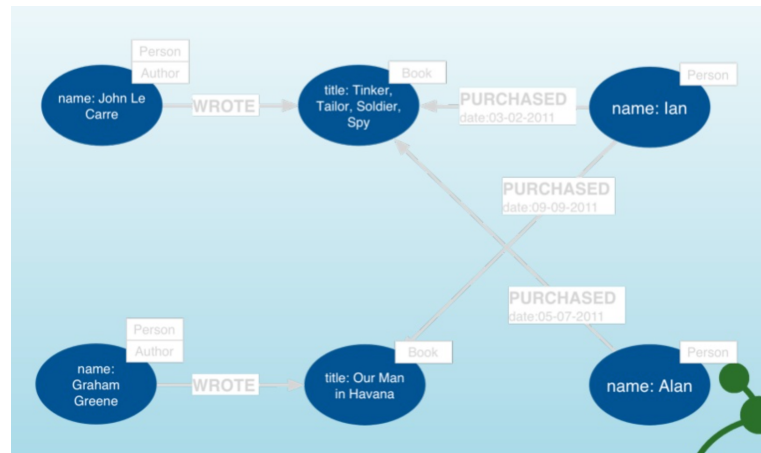- Labels
  - Group Nodes by Role

Graph Databases

Aggregate v Connected Models: OLAP & OLTP are both "Aggregate" focused. Graph Databases are "Connection Focused"

Graph Building Blocks
- Nodes
  - Entities (tables)
- Relationships
  - Connect to Structure
- Properties
  - Attributes and metadata
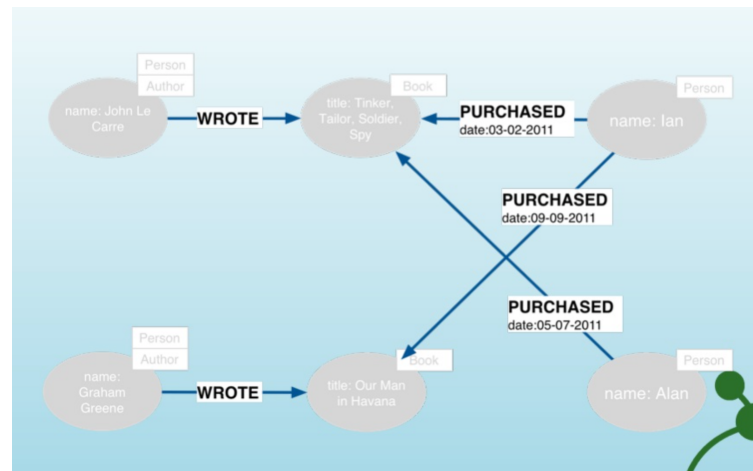- Labels
  - Group Nodes by Role

Graph Databases

Aggregate v Connected Models: OLAP & OLTP are both "Aggregate" focused. Graph Databases are "Connection Focused"

Graph Building Blocks
- Nodes
  - Entities (tables)
- Relationships
  - Connect to Structure
- Properties
  - Attributes and metadata
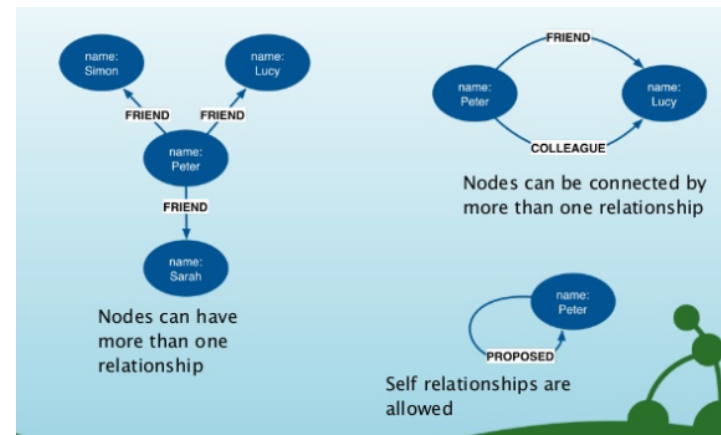- Labels
  - Group Nodes by Role

Graph Databases

Aggregate v Connected Models: OLAP & OLTP are both "Aggregate" focused. Graph Databases are "Connection Focused"

Graph Building Blocks
- Nodes
  - Entities (tables)
- Relationships
  - Connect to Structure
- Properties
  - Attributes and metadata
- Labels
  - Group Nodes by Role

Graph Databases

Aggregate v Connected Models: OLAP & OLTP are both "Aggregate" focused. Graph Databases are "Connection Focused"

Graph Building Blocks
- Nodes
    - Entities (tables)
- Relationships
    - Connect to Structure
- Properties
    - Attributes and metadata
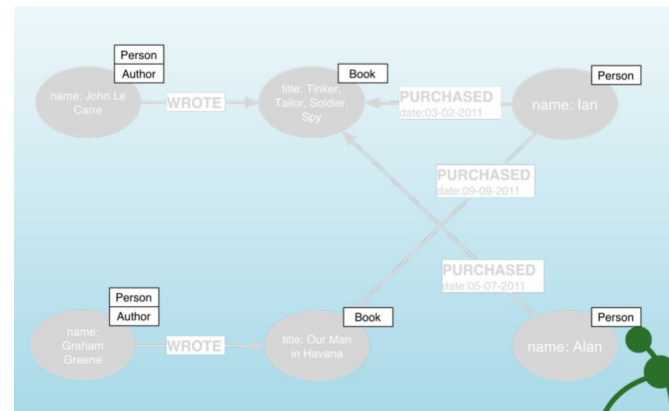- Labels
    - Group Nodes by Role

Questions?