CS 6016

Database Systems
Summer 2024

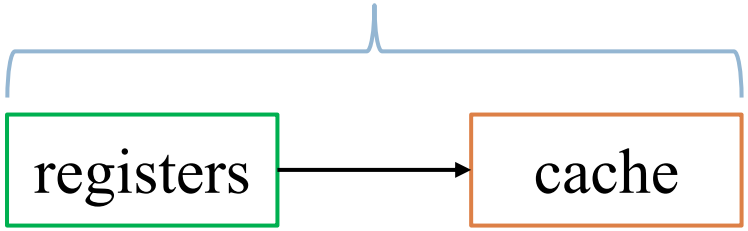
Welcome!

Course Overview

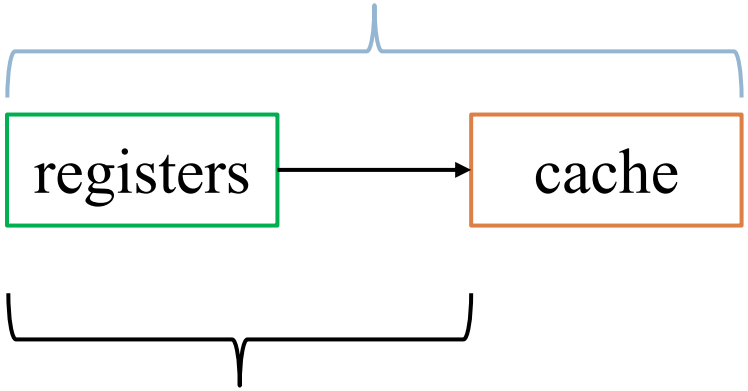
Intro to Data Storage

Domains of Data

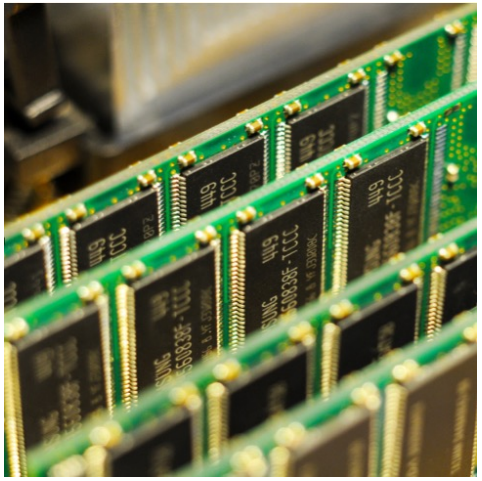
Domains of Data

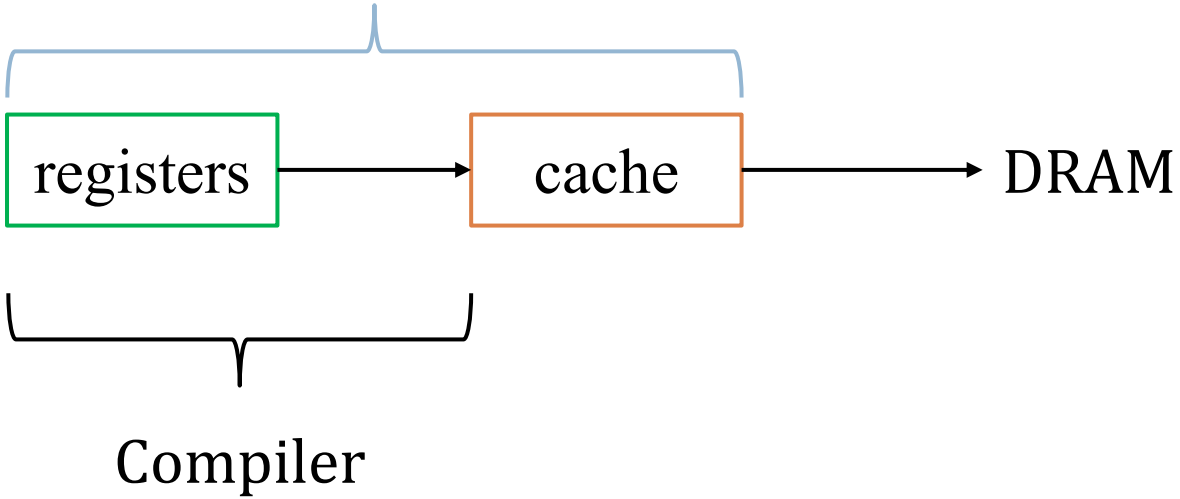


Domains of Data

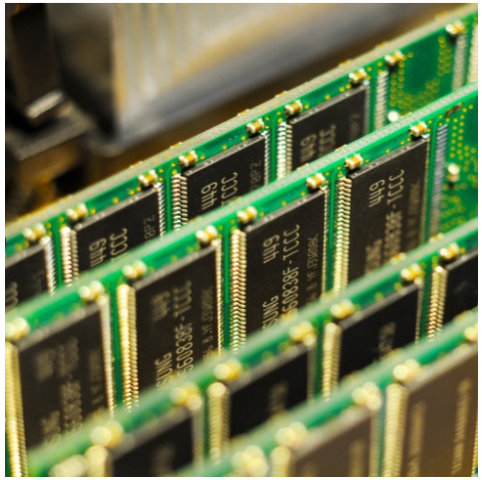


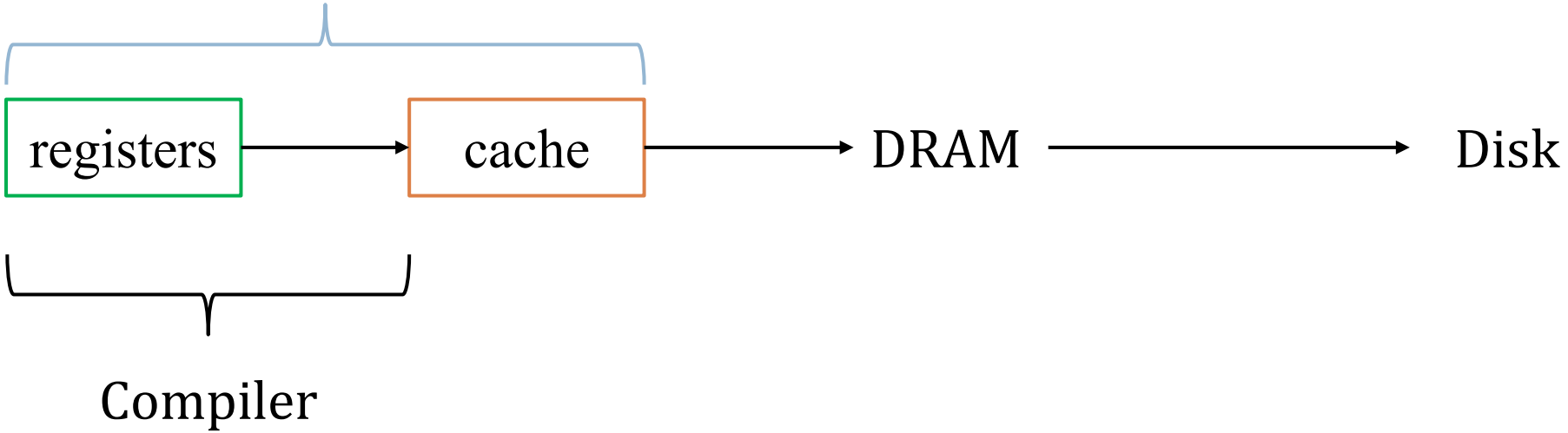
Domains of Data



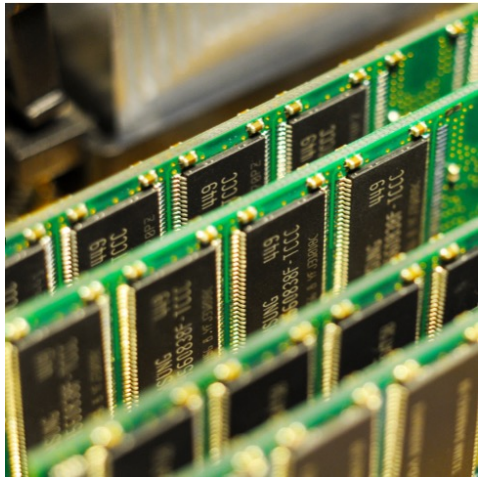


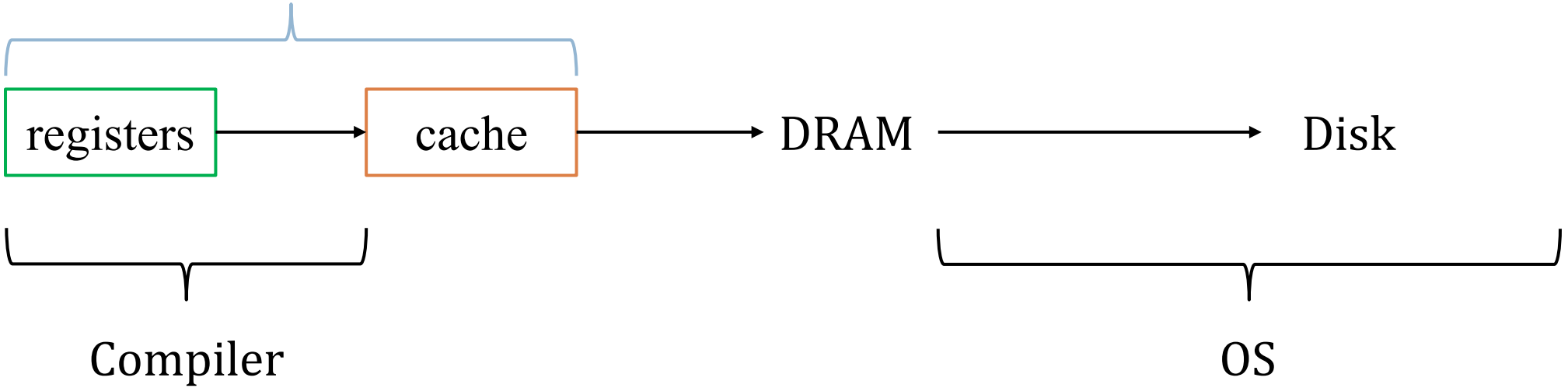
Domains of Data



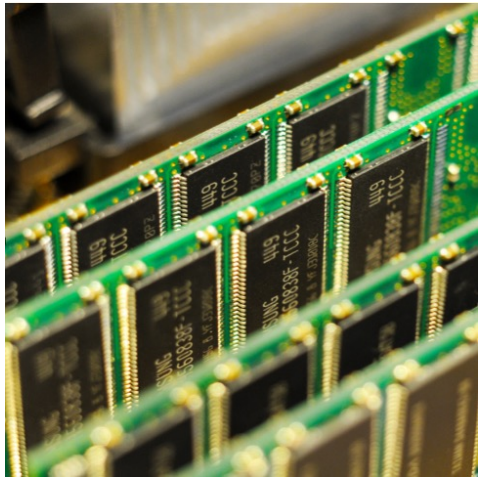


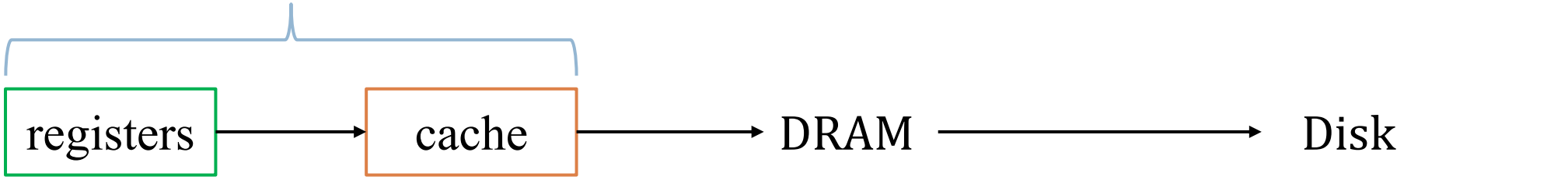
Domains of Data





Domains of Data



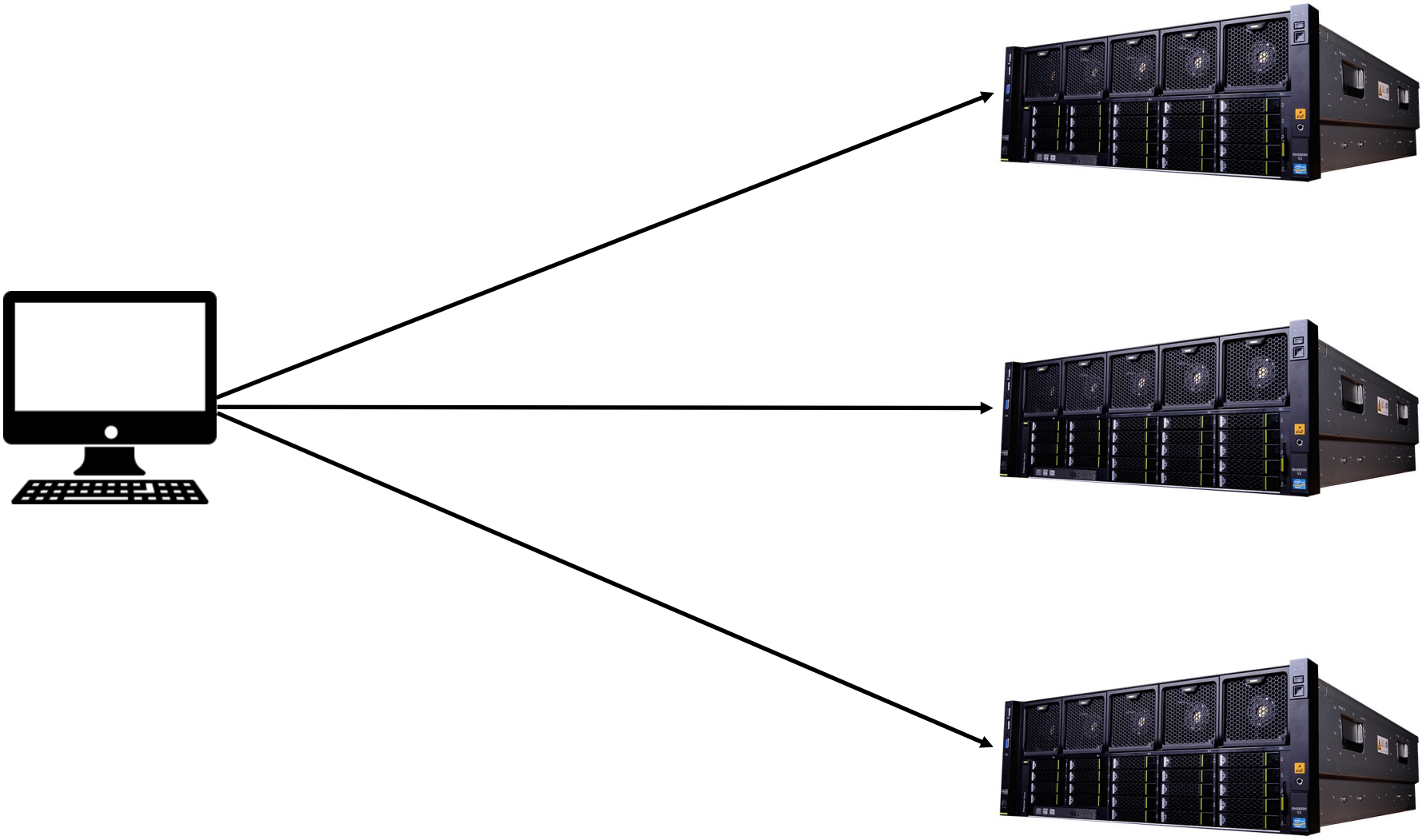


Compiler

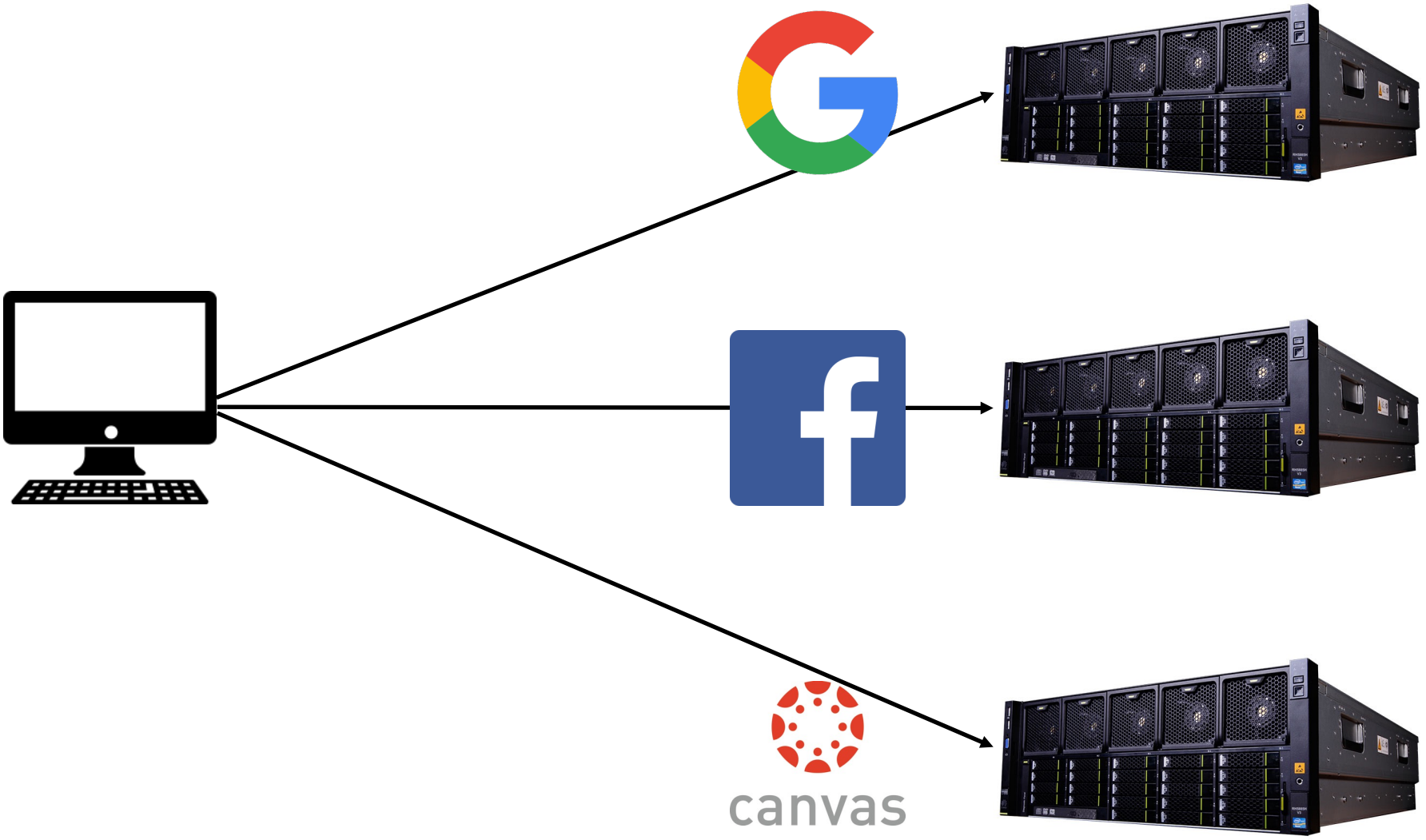
OS

Programmer

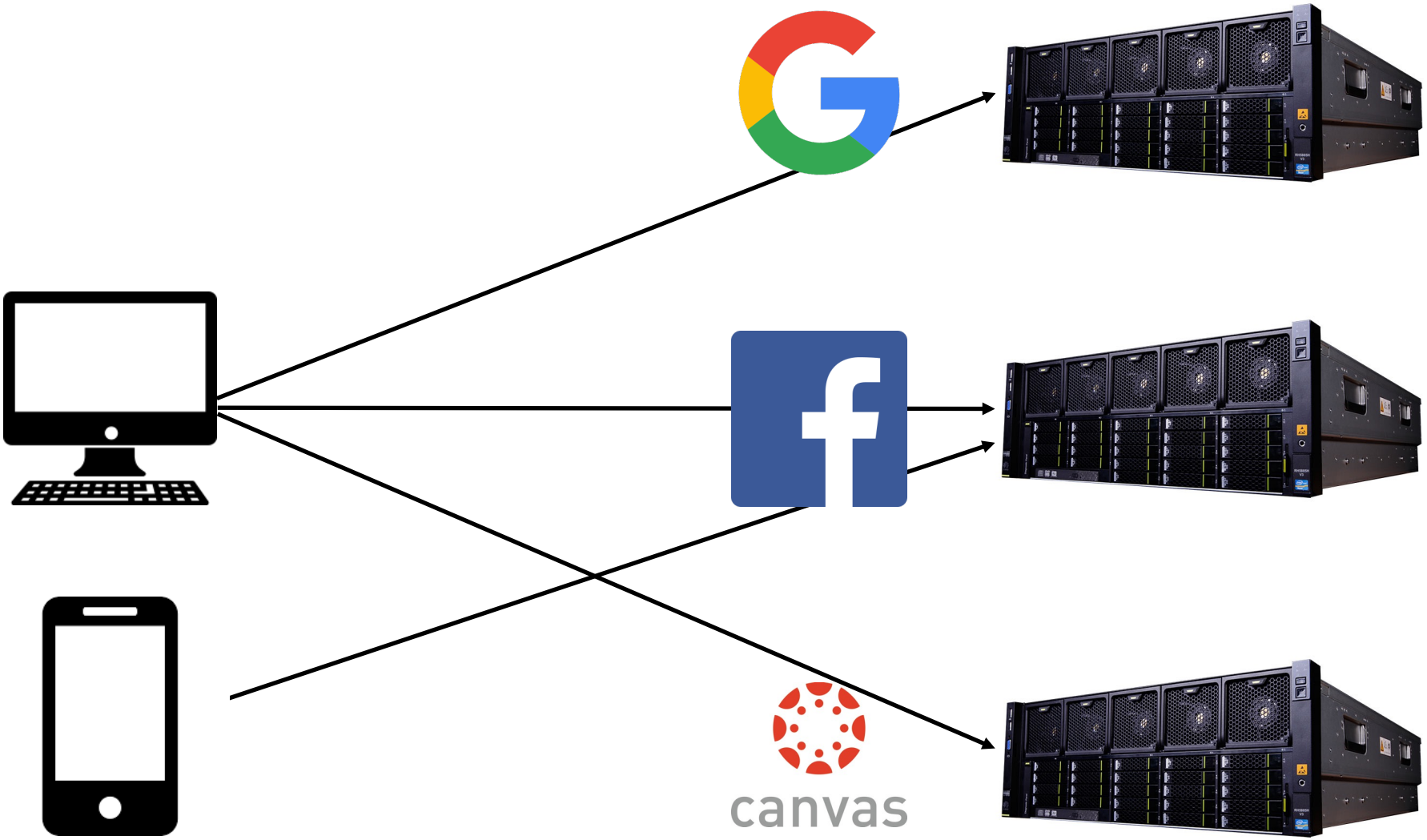
Domains of Data



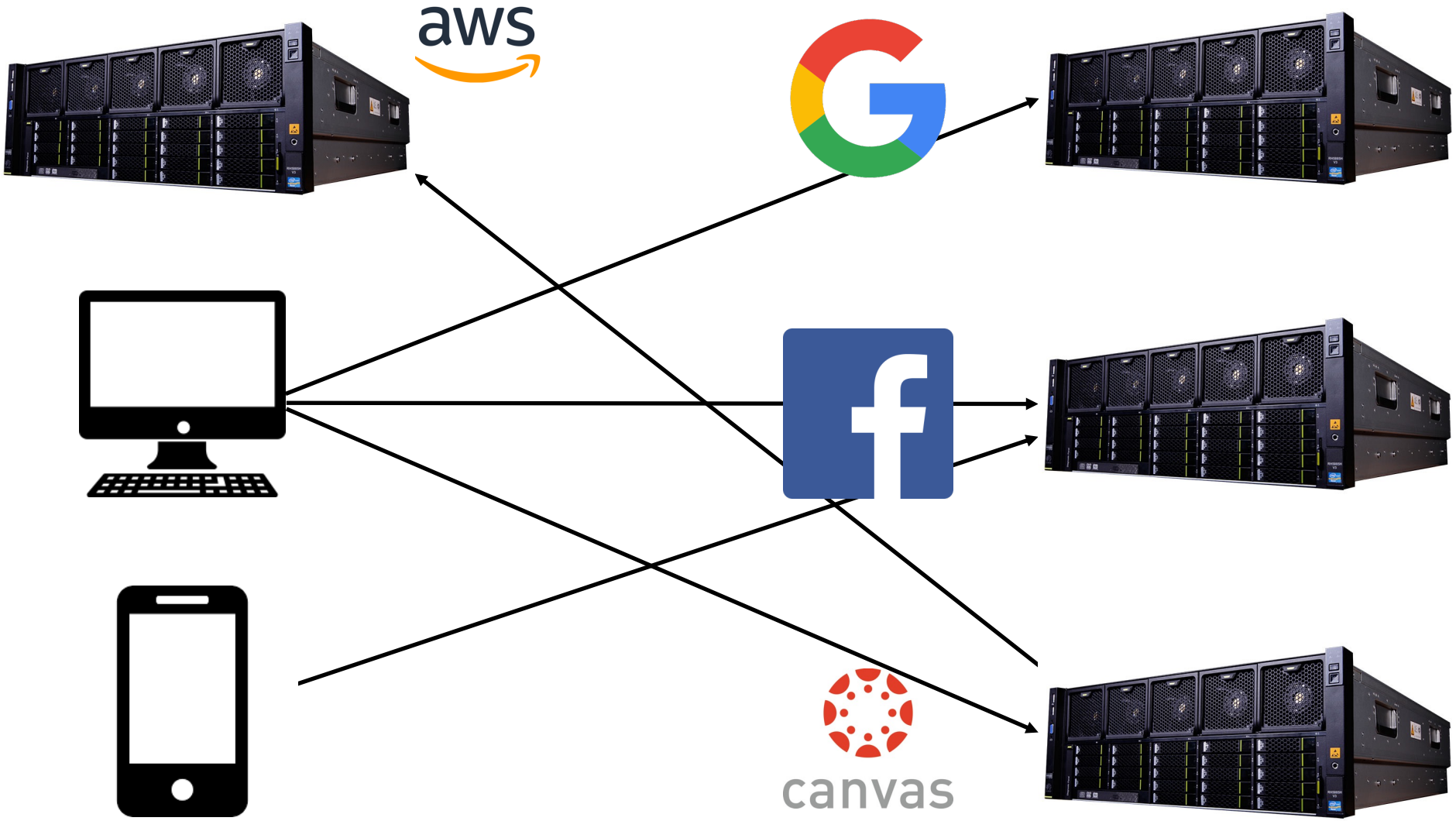
Domains of Data



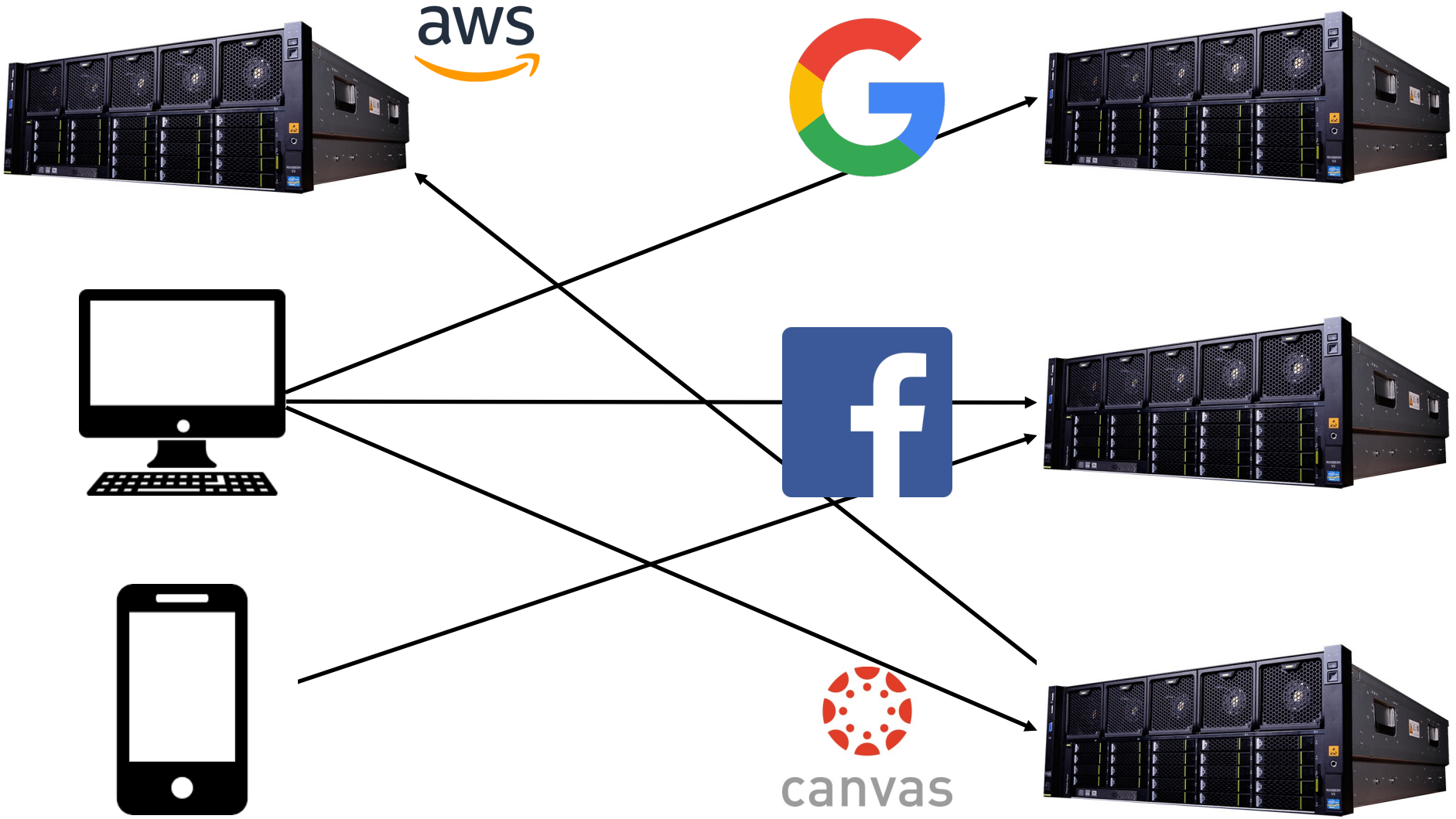
Domains of Data



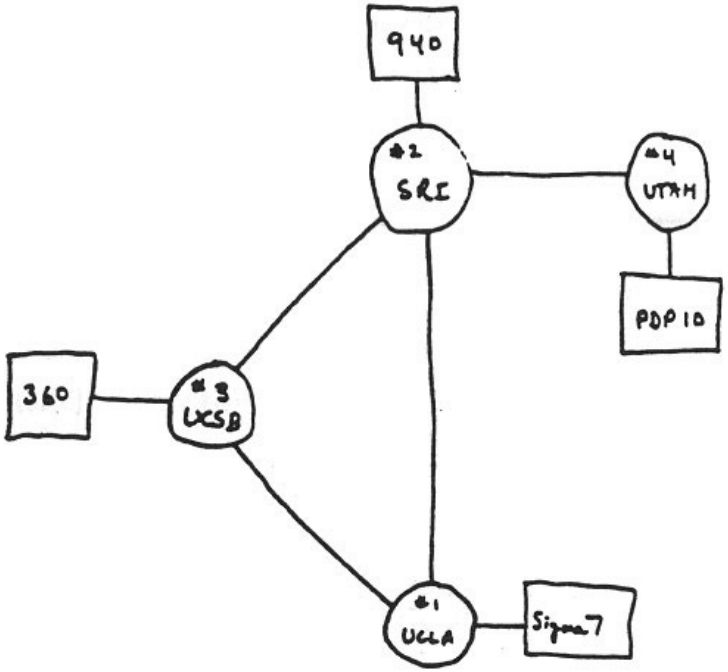
Domains of Data



Domains of Data



History



THE ARPA NETWORK

DEC 1969

4 NODES

“Big” Data

- This end of the spectrum has different challenges:
  - **Vast amounts of data**
  - **Fast access/combination/filtering**
  - **Must be available**
    - Online
    - Securely
    - To simultaneous users

Ponder…

- Suppose you want to save a bunch of Students to a file

Ponder…

- Suppose you want to save a bunch of Students to a file

- Option 1:

"Jane Doe is a Film major with a GPA of 3.7, and is enrolled in CS2420, and her ID is 12345

John Smith is …"

Ponder…

- Suppose you want to save a bunch of Students to a file

- Option 1:

"Jane Doe is a Film major with a GPA of 3.7, and is enrolled in CS2420, and her ID is 12345

John Smith is …"

- How do we search for a student?

Ponder…

- Suppose you want to save a bunch of Students to a file

- Option 1:

"Jane Doe is a Film major with a GPA of 3.7, and is enrolled in CS2420, and her ID is 12345

John Smith is …"

- How do we search for a student?
  - First we have to know the data's format
  - O(N) scan of entire file

Representing Data

- Option 2 (JSON-like)

Major: Film
Class: CS2420
Name: Jane Doe
GPA: 3.7
ID: 12345

Representing Data

- Option 2 (JSON-like)

Major: Film
Class: CS2420
Name: Jane Doe
GPA: 3.7
ID: 12345

- How do we find all students enrolled in 2420?
  - Still a linear scan

How About XML?

```
<Course>
  <Name>CS2420</Name>
  <Students>
    <Student>
      <Name>Jane Doe</Name>
      <Major>Film</Major>
    </Student>
    <Student>
      <Name>John Smith</Name>
      <Major>CS</Major>
    </Student>
  </Students>
</Course>
```

How About XML?

```
<Course>
  <Name>CS2420</Name>
  <Students>
    <Student>
      <Name>Jane Doe</Name>
      <Major>Film</Major>
    </Student>
    <Student>
      <Name>John Smith</Name>
      <Major>CS</Major>
    </Student>
  </Students>
</Course>
```

Still not scalable!

Exercise

- Store a bunch of student records by name, and quickly
  - Add
  - Remove
  - Search
  - Enumerate

Exercise

- Store a bunch of student records by name, and quickly
  - Add
  - Remove
  - Search
  - Enumerate
- Binary search tree

Exercise

- Store a bunch of:
  - students
  - courses
  - professors

Exercise

Professors

Teaching: CS5530, CS4400
Name: Daniel Kopta
ID: 55555

Teaching: CS3500, CS4150
Name: Joe Zachary
ID: 44444

Students

Classes: CS5530, Phys2010
Name: Jane Doe
GPA: 3.7
ID: 12345

Classes: CS3500, FILM1010
Name: Jon Smith
GPA: 3.4
ID: 12421

Courses

Name: Database Systems
Num: 5530
Dept. CS

Name: Software Practice
Num: 3500
Dept. CS

Exercise

- All courses student *Y* is enrolled in?
- All teachers of student *Z*?
- Order courses by enrollment number?

Students

Classes: CS5530, Phys2010
Name: Jane Doe
GPA: 3.7
ID: 12345

Classes: CS3500, FILM1010
Name: Jon Smith
GPA: 3.4
ID: 12421

Professors

Teaching: CS5530, CS4400
Name: Daniel Kopta
ID: 55555

Teaching: CS3500, CS4150
Name: Joe Zachary
ID: 44444

Courses

Name: Database Systems
Num: 5530
Dept. CS

Name: Software Practice
Num: 3500
Dept. CS

Exercise

Professors

Teaching: CS5530, CS4400
Name: Daniel Kopta
ID: 55555

Teaching: CS3500, CS4150
Name: Joe Zachary
ID: 44444

Students

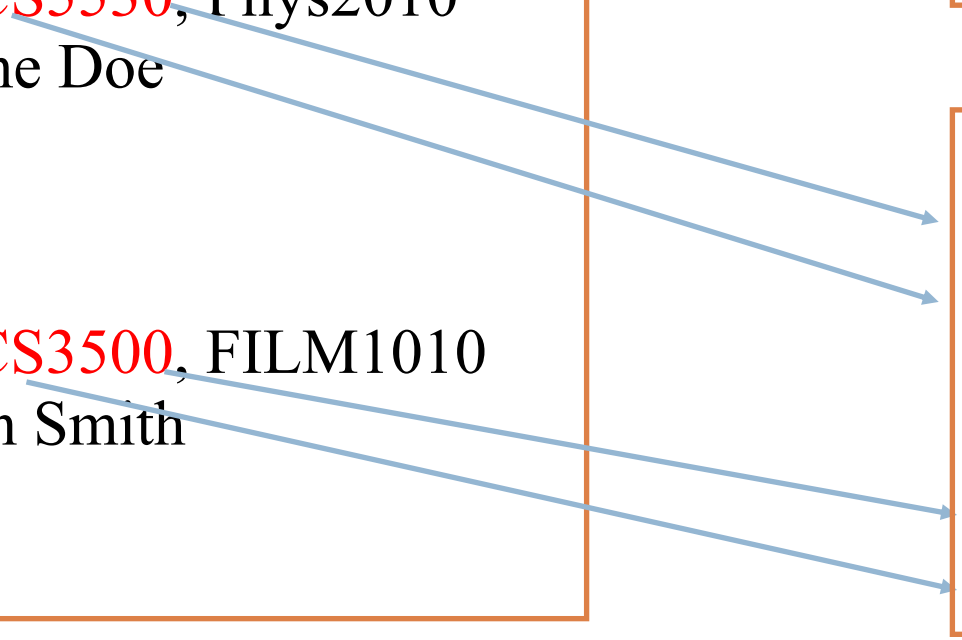
Classes: CS5530, Phys2010
Name: Jane Doe
GPA: 3.7
ID: 12345

Classes: CS3500, FILM1010
Name: Jon Smith
GPA: 3.4
ID: 12421

Courses

Name: Database Systems
Num: 5530
Dept. CS

Name: Software Practice
Num: 3500
Dept. CS



Exercise

Professors

Teaching: CS5530, CS4400
Name: Daniel Kopta
ID: 55555

Teaching: CS3500, CS4150
Name: Joe Zachary
ID: 44444

Students

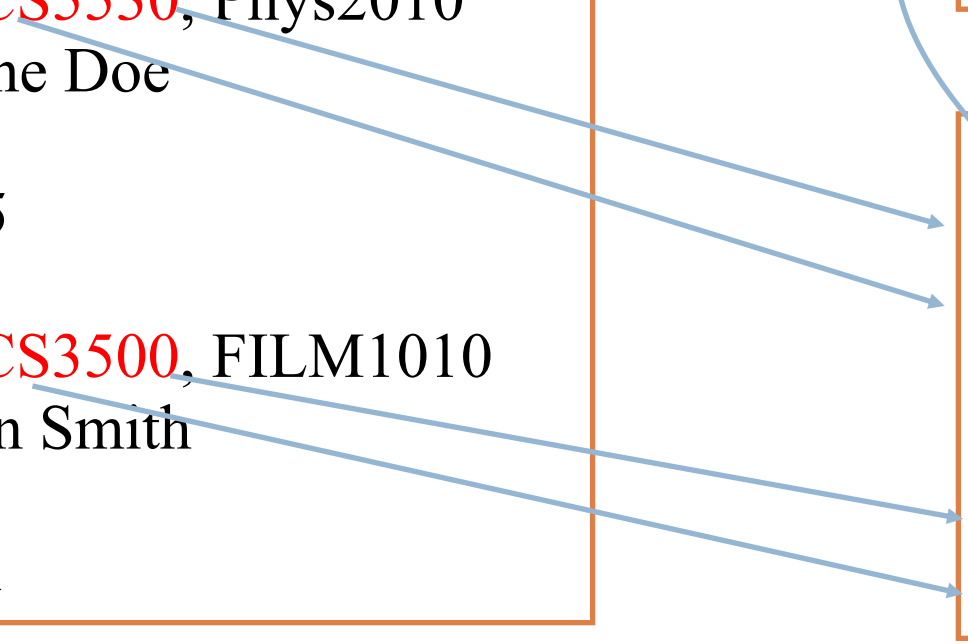
Classes: CS5530, Phys2010
Name: Jane Doe
GPA: 3.7
ID: 12345

Classes: CS3500, FILM1010
Name: Jon Smith
GPA: 3.4
ID: 12421

Courses

Name: Database Systems
Num: 5530
Dept. CS

Name: Software Practice
Num: 3500
Dept. CS



Exercise

- How can we quickly
  - Find all students in course *X*?
  - Find all course(s) student *Y* is enrolled in?
  - Find all teachers of student *Z*?
  - Order students by GPA?
  - Order courses by enrollment number?
  - …
- Now imagine there are **millions** of each
  - And these operations happen frequently

Solution

- *Structured data*
  - Records can not have arbitrary/unpredictable fields/values
  - e.g. courses have: dept, num, and name
    (string) (int) (string)

Structured Data

- Unstructured

  Jane Doe is a Film major with a GPA of 3.7, and her ID is 12345

- Structured

  Name (string): “Jane Doe”
  Major (string): “Film”
  Classes: CS5530, CS4150
  GPA (float): 3.7
  ID (uint): 12345

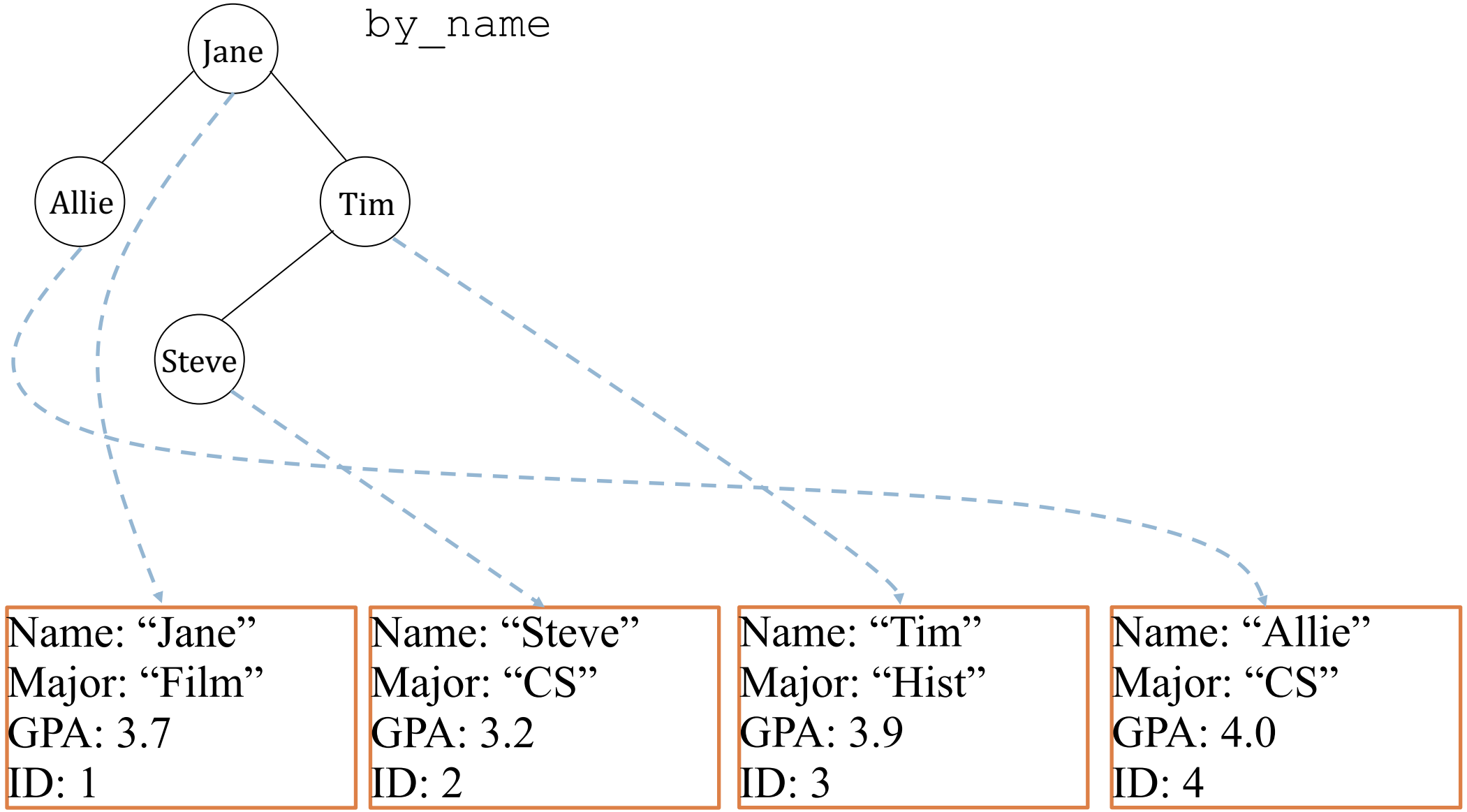
Data Storage

- Save data itself + data structures
  - Trees, hash tables, etc…

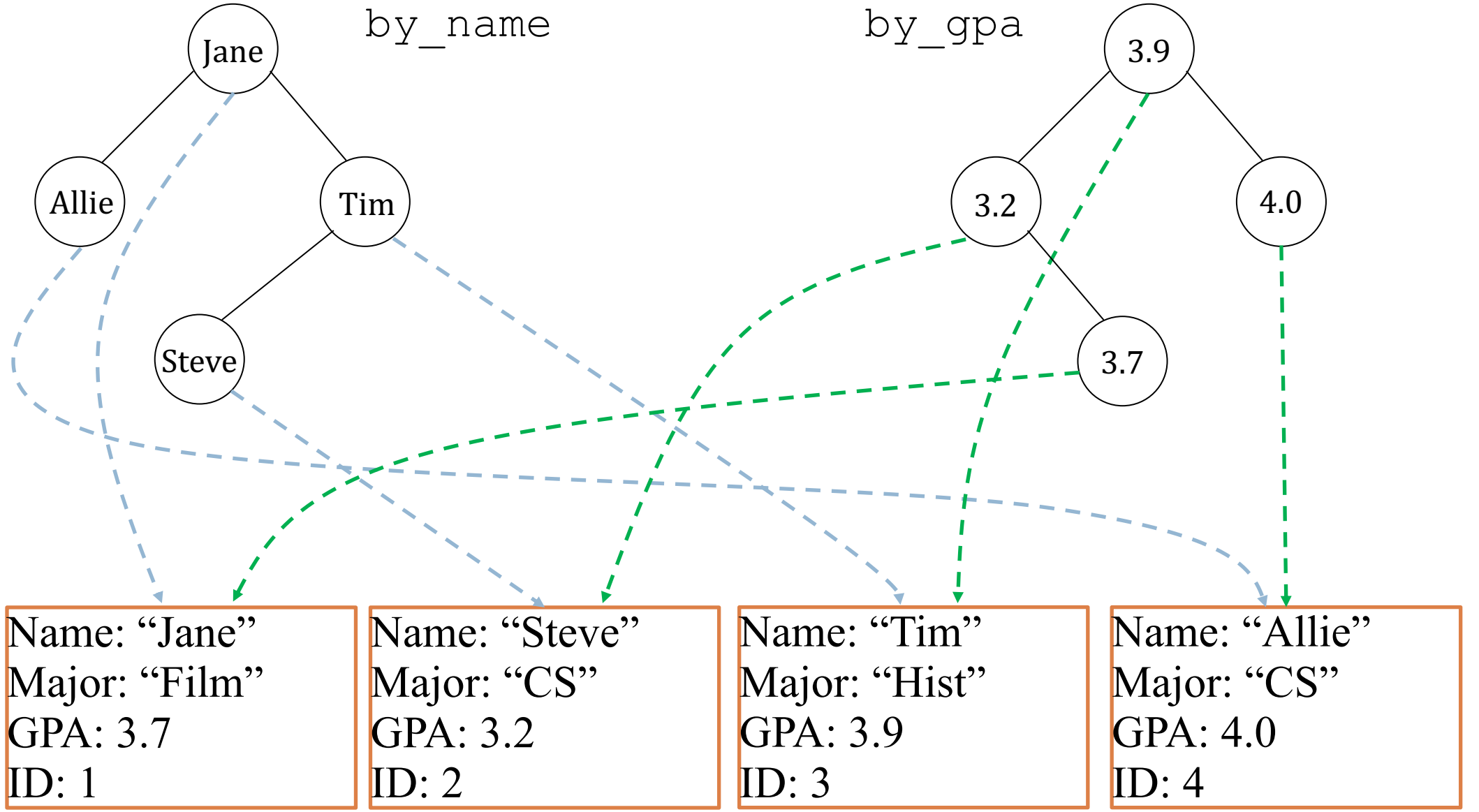
Structured Data

Name: “Jane” Major: “Film” GPA: 3.7 ID: 1	Name: “Steve” Major: “CS” GPA: 3.2 ID: 2	Name: “Tim” Major: “Hist” GPA: 3.9 ID: 3	Name: “Allie” Major: “CS” GPA: 4.0 ID: 4

Structured Data + Data Structure



Structured Data + Data Structure



Exercise

- Language for expressing:
  - Find all students in course *X*?
  - Find all course(s) student *Y* is enrolled in?
  - Find all teachers of student *Z*?
  - Order students by GPA?
  - Order courses by enrollment number?
  - …

- C++, Java, C# etc…?

Solution

- Devise language for combining/filtering data

```
SELECT Name FROM Students
WHERE GPA > 3.5;
```

Solution

- Data + data structures saved on disk
- Devise language for combining/filtering data

- … but this is exactly what a *database* does for you
  - Plus much more!

Why Databases?

- Take advantage of decades of research
  - Availability
  - Reliability
  - Performance
  - Concurrency
  - Interface

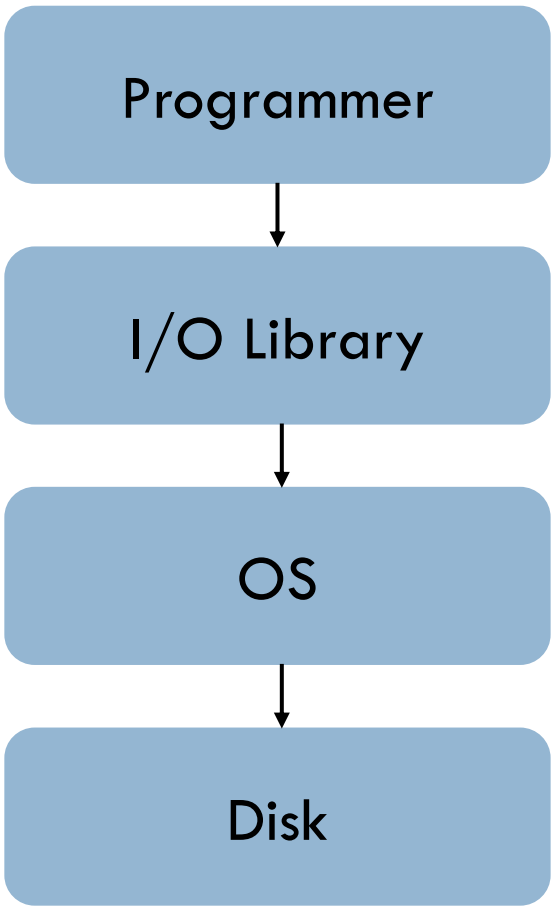
- Don’t reinvent the wheel

Database System

- Two major components:
  - **Database Management System (DBMS)**
    - Underlying machinery
  - **Query Language**
    - Common interface

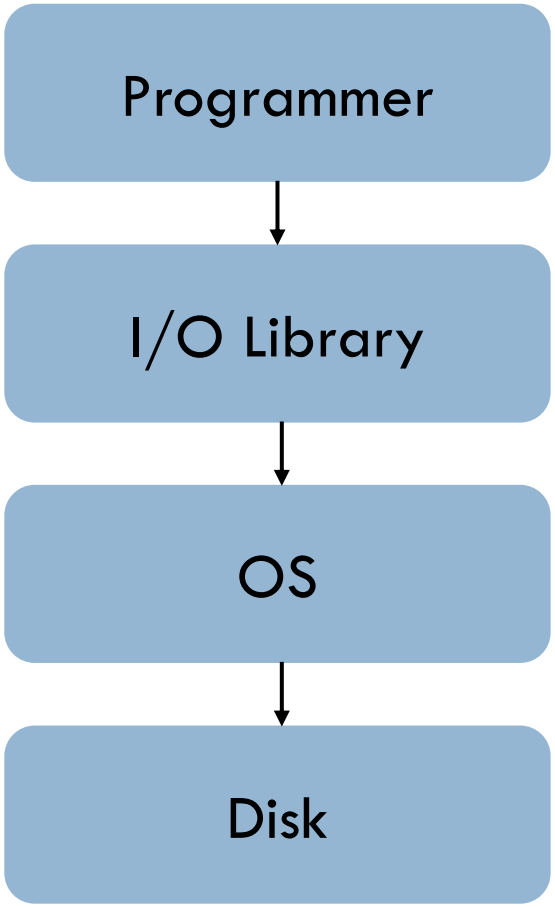
Data Storage

Option 1

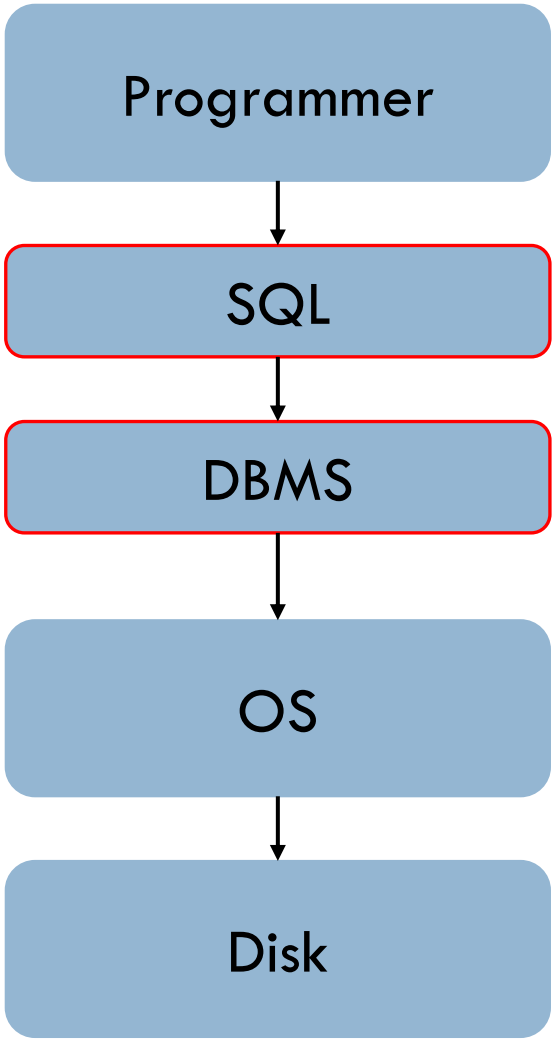


Data Storage

Option 1



Option 2



This Class

- Principles of structured information storage
- Application of databases
- Software interfacing with databases
- Understand DBMS enough to use it effectively

Assignments

- A mixture of:
  - Programming
  - Written/diagram
  - Database manipulation
  - Quizzes

Server

- `cs-db.eng.utah.edu`
    - You will all have your own MySQL database on this server

- As students in the college of engineering (CoE), you also have remote access to the CADE lab machines.
    - cade.utah.edu

Project

- Implement your own Canvas-like system
  - UI is provided for you
  - You implement the back-end
- Multiple phases throughout semester

Grading

- Project: 50%
- Assignments: 35%
- Midterm: 15%

Content

- Structured information storage
  - Relational model, relational databases
  - Relational algebra
  - Entity-Relationship model
  - Database schema design

Content

- Database applications
  - SQL
  - SQL via C# and LINQ
  - Web servers and “cloud” deploy
  - Proper software engineering techniques

Content

- Database engines
  - Data storage challenges
  - B+ Trees
  - R-Trees

Tools

- C# (Using VS Code IDE)
- MySQL
- Linux

Class Resources

- Canvas
- Course announcements via Slack + email
- Grades on Canvas

Important dates

- Midterm: Thursday June 20, 2024 (Week 6)

Getting Help

- Please ask for help!
- Office hours and TA hours are on Canvas

TAs

- Avishek Choudhury
  - TRW 12:00 – 2:00 PM
- Khushal Dodeja
  - TRW 2:00 – 4:00 PM

Academic Misconduct

- Taken very seriously by the SoC
- Cheating punishment ranges up to automatically failing the course!
- Cheating policy is linked in syllabus

Relational Databases

- Structured data storage

- Related data are stored "next to" each other
  - e.g. in a table

ID	Name	DOB

Relational Databases

- Structured data storage
- Related data are stored "next to" each other
  - e.g. in a table

ID	Name	DOB

- Non-relational databases exist too

Tables

- Database comprised of one or more tables (schema)
- One table represents pieces of directly-related data

ID	Name	DOB
1	Harry	31 JUL 1980
2	Hermione	19 SEP 1979
3	Ron	01 MAR 1980
4	Malfoy	05 JUN 1980

Relation (table)

- Each row is a *tuple* – a set of data units

ID	Name	DOB	GPA
1	Harry	31 JUL 1980	3.5
2	Hermione	19 SEP 1979	3.5
3	Ron	01 MAR 1980	4.0
4	Malfoy	05 JUN 1980	3.9

Multiple Tables

- Non directly-related data are separated

Students

ID	Name	DOB
1	Harry	31 JUL 1980
2	Hermione	19 SEP 1979
3	Ron	01 MAR 1980
4	Malfoy	05 JUN 1980

Courses

Course Num	Name
2420	Alg. and DS
3500	SW Practice
3810	Architecture
4400	Systems
5530	Databases

- Each table is a "relation"

Relation (table)

- Each row is a *tuple* – a set of data units
  - Does every cell need to be unique?

ID	Name	DOB	GPA
1	Harry	31 JUL 1980	3.5
2	Hermione	19 SEP 1979	3.5
3	Ron	01 MAR 1980	4.0
4	Malfoy	05 JUN 1980	3.9

Relation (table)

- Each row is a *tuple* – a set of data units
  - Does every cell need to be unique? No

ID	Name	DOB	GPA
1	Harry	31 JUL 1980	3.5
2	Hermione	19 SEP 1979	3.5
3	Ron	01 MAR 1980	4.0
4	Malfoy	05 JUN 1980	3.9

Relation (table)

- Each row is a *tuple* – a set of data units
  - Does every *row* need to be unique?

ID	Name	DOB	GPA
1	Harry	31 JUL 1980	3.5
2	Hermione	19 SEP 1979	3.5
3	Ron	01 MAR 1980	4.0
4	Malfoy	05 JUN 1980	3.9

Relation (table)

- Each row is a *tuple* – a set of data units
  - Does every *row* need to be unique? Yes

ID	Name	DOB	GPA
1	Harry	31 JUL 1980	3.5
2	Hermione	19 SEP 1979	3.5
3	Ron	01 MAR 1980	4.0
4	Malfoy	05 JUN 1980	3.9

Library Example

Name	Phone	CardNum	ISBN	Book
Harry	123-1123	123	978-000	Harry Potter
Hermione	555-1234	124	978-001	A Tale of Two Cities
Ron	123-4567	228	978-002	Last of Us
Malfoy	765-4321	623	978-003	Dune

Library Example

- Is Malfoy directly-related to Dune?

Name	Phone	CardNum	ISBN	Book
Harry	123-1123	123	978-000	Harry Potter
Hermione	555-1234	124	978-001	A Tale of Two Cities
Ron	123-4567	228	978-002	Last of Us
Malfoy	765-4321	623	978-003	Dune

Library Example

- Is Malfoy directly-related to Dune?

Name	Phone	CardNum	ISBN	Book
Harry	123-1123	123	978-000	Harry Potter
Hermione	555-1234	124	978-001	A Tale of Two Cities
Ron	123-4567	228	978-002	Last of Us
Malfoy	765-4321	623	978-003	Dune

- No; only indirectly (he has it checked out)

Library Example

- What if one person checks out multiple books?

Name	Phone	CardNum	ISBN	Book
Harry	123-1123	123	978-000	Harry Potter
Hermione	555-1234	124	978-001	A Tale of Two Cities
Ron	123-4567	228	978-002	Last of Us
Malfoy	765-4321	623	978-003	Dune

Library Example

- What if one person checks out multiple books?

Name	Phone	CardNum	ISBN	Book
Harry	123-1123	123	978-000	Harry Potter
Hermione	555-1234	124	978-001	A Tale of Two Cities
Ron	123-4567	228	978-002	Last of Us
Malfoy	765-4321	623	978-003	Dune
Malfoy	765-4321	623	978-004	Hyperion
Malfoy	765-4321	623	978-005	Bunny Meadows

Library Example

- What if one person checks out multiple books?
  - Duplicate data

Name	Phone	CardNum	ISBN	Book
Harry	123-1123	123	978-000	Harry Potter
Hermione	555-1234	124	978-001	A Tale of Two Cities
Ron	123-4567	228	978-002	Last of Us
Malfoy	765-4321	623	978-003	Dune
Malfoy	765-4321	623	978-004	Hyperion
Malfoy	765-4321	623	978-005	Bunny Meadows

Library Example

- What if one person checks out multiple books?
  - Make a list?

Name	Phone	CardNum	ISBN	Book
Harry	123-1123	123	978-000	Harry Potter
Hermione	555-1234	124	978-001	A Tale of Two Cities
Ron	123-4567	228	978-002	Last of Us
Malfoy	765-4321	623	978-003, 978-004, 978-005	Dune, Hyperion, …

Library Example

- What if one person checks out multiple books?
  - How do we count number of books?
  - How do we get all books that start with 'H'?

Name	Phone	CardNum	ISBN	Book
Harry	123-1123	123	978-000	Harry Potter
Hermione	555-1234	124	978-001	A Tale of Two Cities
Ron	123-4567	228	978-002	Last of Us
Malfoy	765-4321	623	978-003, 978-004, 978-005	Dune, Hyperion, …

Library Example

- DBMS builds "index" data structures around many of the columns
  - Can't do this if some cells are lists

tree

Name	Phone	CardNum	ISBN	Book
Harry	123-1123	123	978-000	Harry Potter
Hermione	555-1234	124	978-001	A Tale of Two Cities
Ron	123-4567	228	978-002	Last of Us
Malfoy	765-4321	623	978-003	Dune

Library Example

- What if one person checks out no books?

Name	Phone	CardNum	ISBN	Book
Harry	123-1123	123	978-000	Harry Potter
Malfoy	765-4321	623	???	???

Even Worse

- Multiple phone numbers, multiple checkouts

Name	Phone	CardNum	ISBN	Title
Dan	888-8888	4	1005	Profiles in Courage
Dan	999-9999	4	1005	Profiles in Courage
Dan	888-8888	4	1006	The Good Soldier
Dan	999-9999	4	1006	The Good Soldier

Solution

- First, let's fix the unrelated-data problem

Patrons

Name	Phone	CardNum
Harry	123-1123	123
Malfoy	765-4321	623

Inventory

ISBN	Book
978-002	Last of Us
978-003	Dune
978-007	Annihilation

Solution

- First, let's fix the unrelated-data problem

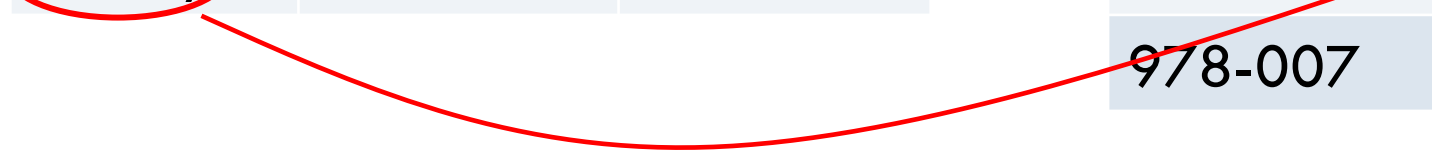
- But what about indirect relationships?
  - How do we specify Malfoy checked out Dune?

Patrons

Name	Phone	CardNum
Harry	123-1123	123
Malfoy	765-4321	623

Inventory

ISBN	Book
978-002	Last of Us
978-003	Dune
978-007	Annihilation



Solution

- Add a table that relates the two

Patrons

Name	Phone	CardNum
Harry	123-1123	123
Malfoy	765-4321	623

Inventory

ISBN	Book
978-002	Last of Us
978-003	Dune
978-007	Annihilation

CheckedOut

CardNum	ISBN
123	978-002
623	978-003

Solution

- Add a table that relates the two

Patrons

Name	Phone	CardNum
Harry	123-1123	123
Malfoy	765-4321	623

Inventory

ISBN	Book
978-002	Last of Us
978-003	Dune
978-007	Annihilation

CheckedOut

CardNum	ISBN
123	978-002
623	978-003

Solution

- Multiple checkouts
  - Duplicate data minimized

Patrons

Name	Phone	CardNum
Harry	123-1123	123
Malfoy	765-4321	623

Inventory

ISBN	Book
978-002	Last of Us
978-003	Dune
978-007	Annihilation

CheckedOut

CardNum	ISBN
123	978-002
123	987-007
623	978-003

Solution

- No checkouts

Patrons

Name	Phone	CardNum
Harry	123-1123	123
Malfoy	765-4321	623

Inventory

ISBN	Book
978-002	Last of Us
978-003	Dune
978-007	Annihilation

CheckedOut

CardNum	ISBN
623	978-003

Solution

- What about multiple phone numbers?

Patrons

Name	Phone	CardNum
Harry	123-1123	123
Malfoy	765-4321	623

Inventory

ISBN	Book
978-002	Last of Us
978-003	Dune
978-007	Annihilation

CheckedOut

CardNum	ISBN
623	978-003

Solution

- What about multiple phone numbers?

Patrons

Name	Phone	CardNum
Harry	123-1123	123
Malfoy	765-4321	623

Inventory

ISBN	Book
978-002	Last of Us
978-003	Dune
978-007	Annihilation

Phones

CardNum	Phone
123	123-1123
123	555-5555
623	765-4321

CheckedOut

CardNum	ISBN
623	978-003

Solution

- Pick some unique ID-like field to relate tables (key)
  - CardNum and ISBN

Patrons

Name	CardNum
Harry	123
Malfoy	623

Inventory

ISBN	Book
978-002	Last of Us
978-003	Dune
978-007	Annihilation

Phones

CardNum	Phone
123	123-1123
123	555-5555
623	765-4321

CheckedOut

CardNum	ISBN
623	978-003

Solution

- Pick some unique ID-like field to relate tables (key)
  - CardNum and ISBN

Patrons

Name	CardNum
Harry	123
Malfoy	623

Phones

CardNum	Phone
123	123-1123
123	555-5555
623	765-4321

Inventory

ISBN	Book
978-002	Last of Us
978-003	Dune
978-007	Annihilation

CheckedOut

CardNum	ISBN
623	978-003

Solution

- What if we have multiple copies of the same book?

Patrons

Name	CardNum
Harry	123
Malfoy	623

Phones

CardNum	Phone
123	123-1123
123	555-5555
623	765-4321

Inventory

ISBN	Book
978-002	Last of Us
978-003	Dune
978-007	Annihilation

CheckedOut

CardNum	ISBN
623	978-003

Solution

- What if we have multiple copies of the same book?
  - Make another table!

Patrons

Name	CardNum
Harry	123
Malfoy	623

Inventory

ISBN	Book
978-002	Last of Us
978-003	Dune
978-007	Annihilation

Phones

CardNum	Phone
123	123-1123
123	555-5555
623	765-4321

CheckedOut

CardNum	ISBN
623	978-003

Library

Patrons

Name	CardNum
Joe	1
Ann	2
Ben	3
Dan	4

Inventory

Serial	ISBN
1001	978-0590353427
1002	978-0590353427
1003	978-0679732242
1004	978-0394823379
1005	978-0394823379
1006	978-0062278791

CheckedOut

CardNum	Serial
1	1001
1	1004
4	1005
4	1006

Phones

CardNum	Phone
1	555-5555
2	666-6666
3	777-7777
4	888-8888
4	999-9999

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Library

Patrons

Name	CardNum
Joe	1
Ann	2
Ben	3
Dan	4

Inventory

Serial	ISBN
1001	978-0590353427
1002	978-0590353427
1003	978-0679732242
1004	978-0394823379

CheckedOut

CardNum	Serial
1	1001
1	1004
4	1005

Phones

CardNum	Phone
1	555-5555
2	666-6666

How do we find all the books checked out by Joe?

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Library

Patrons

Name	CardNum
Joe	1
Ann	2
Ben	3
Dan	4

Inventory

Serial	ISBN
1001	978-0590353427
1002	978-0590353427
1003	978-0679732242
1004	978-0394823379
1005	978-0394823379
1006	978-0062278791

CheckedOut

CardNum	Serial
1	1001
1	1004
4	1005
4	1006

Phones

CardNum	Phone
1	555-5555
2	666-6666
3	777-7777
4	888-8888
4	999-9999

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Library

Patrons

Name	CardNum
Joe	1
Ann	2
Ben	3
Dan	4

Inventory

Serial	ISBN
1001	978-0590353427
1002	978-0590353427
1003	978-0679732242
1004	978-0394823379
1005	978-0394823379
1006	978-0062278791

CheckedOut

CardNum	Serial
1	1001
1	1004
4	1005
4	1006

Phones

CardNum	Phone
1	555-5555
2	666-6666
3	777-7777
4	888-8888
4	999-9999

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Library

Patrons

Name	CardNum
Joe	1
Ann	2
Ben	3
Dan	4

Inventory

Serial	ISBN
1001	978-0590353427
1002	978-0590353427
1003	978-0679732242
1004	978-0394823379
1005	978-0394823379
1006	978-0062278791

CheckedOut

CardNum	Serial
1	1001
1	1004
4	1005
4	1006

Phones

CardNum	Phone
1	555-5555
2	666-6666
3	777-7777
4	888-8888
4	999-9999

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Library

Patrons

Name	CardNum
Joe	1
Ann	2
Ben	3
Dan	4

Inventory

Serial	ISBN
1001	978-0590353427
1002	978-0590353427
1003	978-0679732242
1004	978-0394823379
1005	978-0394823379
1006	978-0062278791

CheckedOut

CardNum	Serial
1	1001
1	1004
4	1005
4	1006

Phones

CardNum	Phone
1	555-5555
2	666-6666
3	777-7777
4	888-8888
4	999-9999

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Library

Patrons

Name	CardNum
Joe	1
Ann	2
Ben	3
Dan	4

Inventory

Serial	ISBN
1001	978-0590353427
1002	978-0590353427
1003	978-0679732242
1004	978-0394823379
1005	978-0394823379
1006	978-0062278791

CheckedOut

CardNum	Serial
1	1001
1	1004
4	1005
4	1006

Phones

CardNum	Phone
1	555-5555
2	666-6666
3	777-7777
4	888-8888
4	999-9999

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Library

Patrons

Name	CardNum
Joe	1
Ann	2
Ben	3
Dan	4

Inventory

Serial	ISBN
1001	978-0590353427
1002	978-0590353427
1003	978-0679732242
1004	978-0394823379
1005	978-0394823379
1006	978-0062278791

CheckedOut

CardNum	Serial
1	1001
1	1004
4	1005
4	1006

Phones

CardNum	Phone
1	555-5555
2	666-6666
3	777-7777
4	888-8888
4	999-9999

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Library

Patrons

Name	CardNum
Joe	1
Ann	2
Ben	3
Dan	4

Inventory

Serial	ISBN
1001	978-0590353427
1002	978-0590353427
1003	978-0679732242
1004	978-0394823379
1005	978-0394823379
1006	978-0062278791

CheckedOut

CardNum	Serial
1	1001
1	1004
4	1005
4	1006

Phones

CardNum	Phone
1	555-5555
2	666-6666
3	777-7777
4	888-8888
4	999-9999

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Library

Patrons

Name	CardNum
Joe	1
Ann	2
Ben	3
Dan	4

Inventory

Serial	ISBN
1001	978-0590353427
1002	978-0590353427
1003	978-0679732242
1004	978-0394823379
1005	978-0394823379
1006	978-0062278791

CheckedOut

CardNum	Serial
1	1001
1	1004
4	1005
4	1006

Phones

CardNum	Phone
1	555-5555
2	666-6666
3	777-7777
4	888-8888
4	999-9999

Titles

ISBN	Title	Author
978-0590353427	Harry Potter	Rowling
978-0679732242	The Sound and the Fury	Faulkner
978-0394823379	The Lorax	Seuss
978-0062278791	Profiles in Courage	Kennedy
978-0441172719	Dune	Herbert

Basic Design Goals

- Entries should be atoms (not complex)
  - Don’t store lists/arrays

Basic Design Goals

- Entries should be atoms (not complex)
  - Don’t store lists/arrays
  - Build compound information by referencing other tables

Basic Design Goals

- Entries should be atoms (not complex)
  - Don’t store lists/arrays
  - Build compound information by referencing other tables
  - Enables powerful reasoning about data and relationships, cleaner design

Basic Design Goals

- Entries should be atoms (not complex)
  - Don’t store lists/arrays
  - Build compound information by referencing other tables
  - Enables powerful reasoning about data and relationships, cleaner design
  - Enable DBMS to optimize

Basic Design Goals

- Bad news: SQL will let you violate good design rules
- Thus, we design the tables without even thinking about SQL