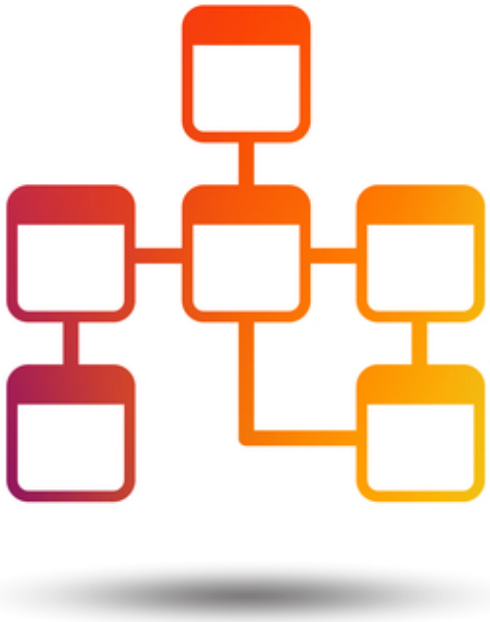
Full-Stack Web Development



FS1030 – **Database Design and Principles**

school of continuing studies | YORK UNIVERSITÉ UNIVERSITY

About Tarun

- B.Tech in Computer Science – I.P. University
- MS in Computer Science – Georgia Tech.
- MBA – Schulich School of Business
- Entrepreneur for 6 years – Web Development Startup
- Full Stack Developer – LAMP & MEAN
- Solutions Architect – Blockchain, IBM Canada
- Serious Hiker – Hiked to Annapurna Base Camp in Nepal (~5000 meters) & climbed Cerro Chirripo, highest peak in Central America in Costa Rica.

Introductions

Tell the class about:

- Brief introduction about yourself
- Your reason to take this certification
- Your goal after this certification
- Your one hobby/achievement/interest/passion

Recommended tools for this course

- Draw.io - to draw EER and IFD diagrams
- MS Word – to write pseudo code and task decomposition
- MySQL Server – to run MySQL locally
- Workbench – MySQL Editor/IDE for managing MySQL databases
- MongoDB – A NoSQL database
- CompassDB – A mongoDB IDE to manage the mongoDB
- IDE of your choice for writing code (Visual Studio Code, Atom, Sublime, Notepad++, etc.)
- Server side language of your choice for programming (PHP, Node.js, Python, etc.)
- Front end language of your choice for UI development (Angular, React, HTML5, etc.)

Agenda

- What is a database?
- What are the different types of databases?
- How are they different from each other?
- Why and when do we pick one over other?
- How are databases structured?
- What role can a database play in any technical-business use case?
- Intro to Information Flow Diagrams
- Intro to Entity Relationship Diagrams

What is the database?

"A database is an organized collection of data, generally stored and accessed electronically from a computer system."

- Wikipedia

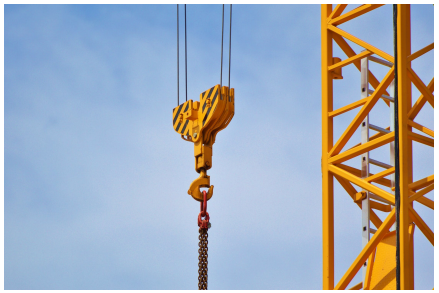
Why do we need DBs?

In recent history, DBs were files.

Issues:

- Data redundancy – duplication of data in multiple files
- Difficulty in accessing data – programs for accessing data
- Isolation – multiple formats and files
- Data Integrity issues – hard to add constraints
- Atomicity – failures leave data in inconsistent state
- Concurrency – access by multiple users at the same time
- Security – protection of data becomes an overhead

Some use cases?

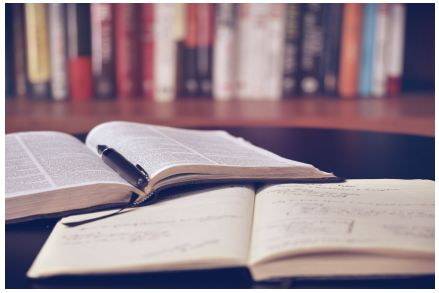


Supply Chain

Banking & Finance

Healthcare

Retail



Education

Telecom & IT

and many more…

RDBMS vs. NoSQL

RDBMS

- ACID (Atomicity, Consistency, Isolation, Durability)
- Meant for data integrity
- Data is structured and unhanging
- Table based relational structure
- High investment of resources for scale.

NoSQL

- Schema less so no ACID properties
- Meant for performance
- Dynamic structure
- Can scale horizontally
- Document based

A.C.I.D

Atomicity

Either the transaction happens or not it does not i.e. transactions do not occur partially.

Consistency

The data integrity must be maintained before and after the transaction.

Isolation

Multiple transactions can occur at the same time without leading to inconsistencies in the database. All transactions remain isolated.

Durability

The data is written on the disk and is persistent. The changes whether insert, update or delete are permanent.

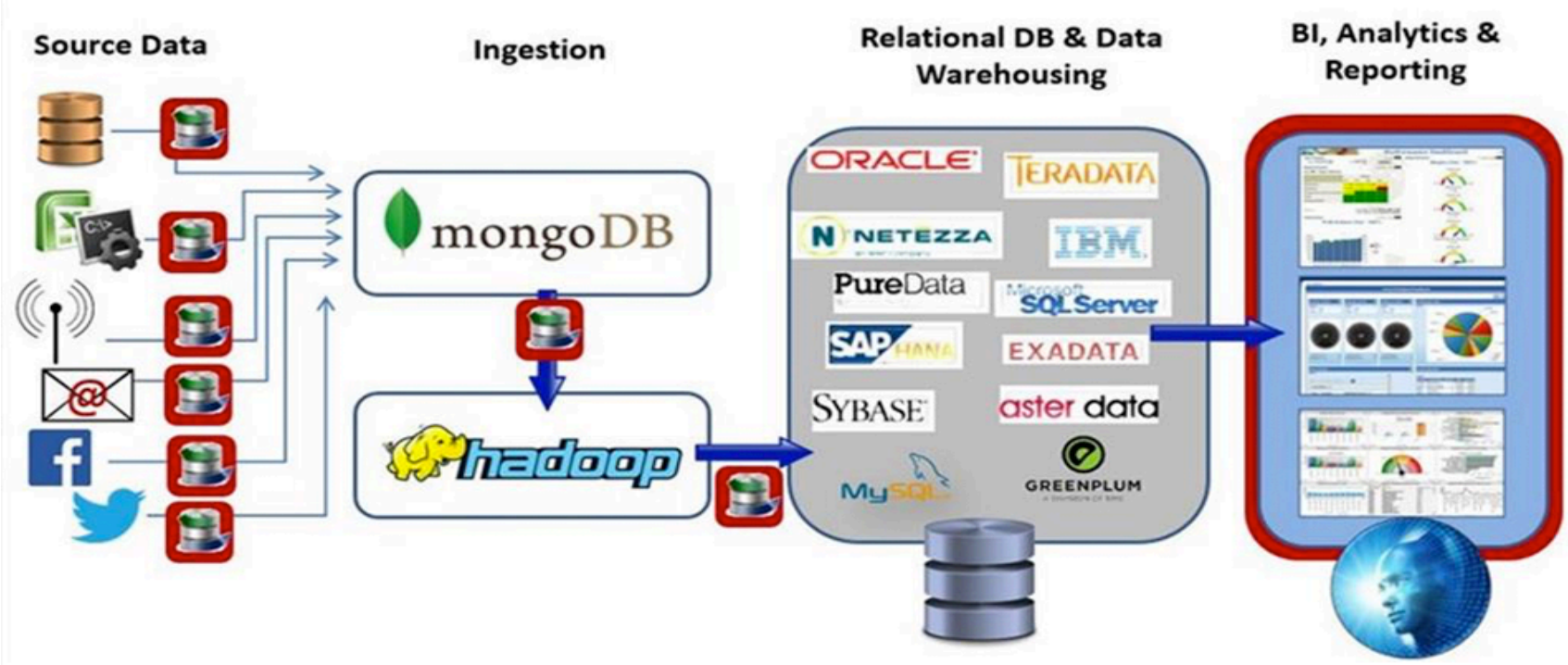
Use cases

- Adam has $600
- Curtis has $400
- $100 deducted from Adam's total
- $100 added to Curtis's total

- Adam is paying the invoice online
- Curtis also paying the same invoice at the same time.
- What happens now?

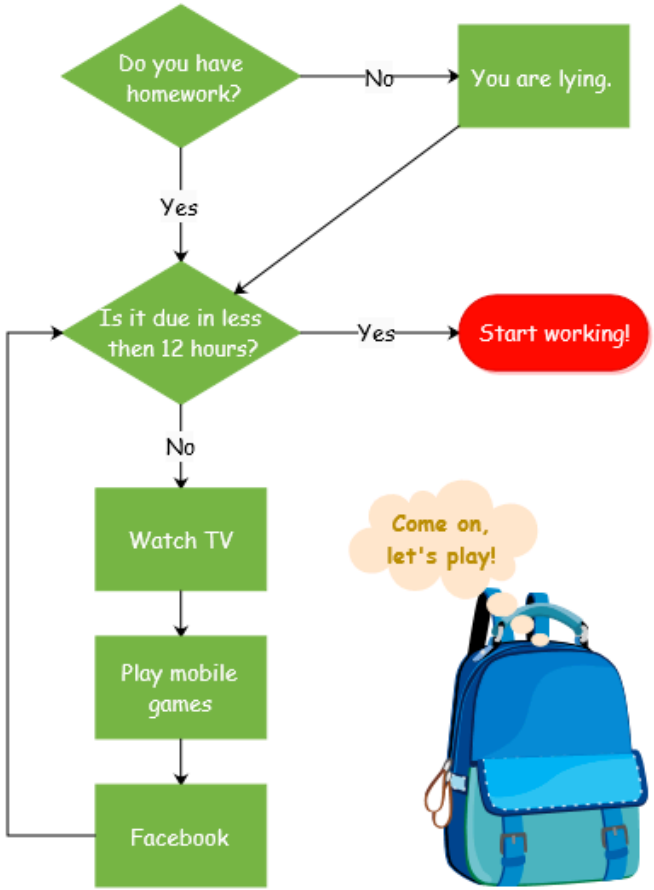
Data organization & architecture

Data organization is a set of rules, models, policies and standards that define the type of data collected and how it is managed and stored within an organization and database systems.



BPM & DM

Business process modelling is the illustration of organization's business processes.



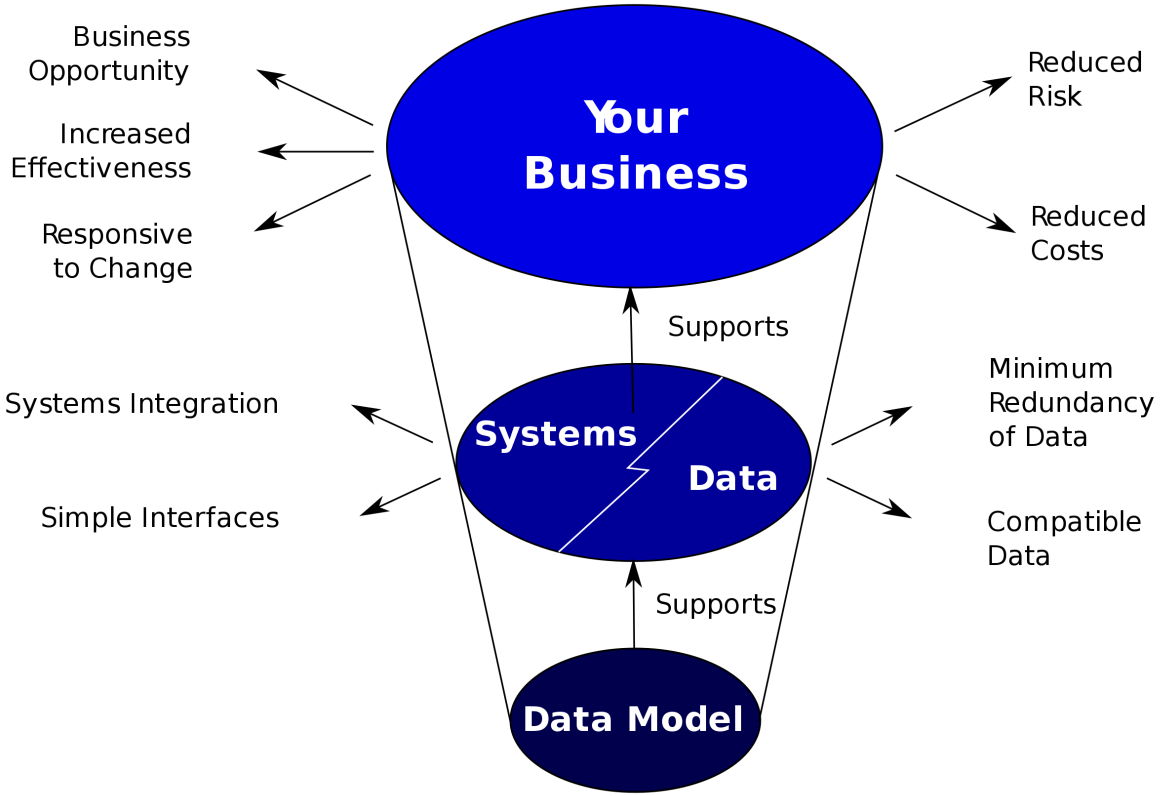
Source: https://www.edrawsoft.com/fun-flowchart.php

Many techniques for BPM such as:

- Flowcharts
- BPM Notation
- Universal Process Notation
- Gantt Charts
- Petri-nets

BPM & DM

Data modeling is the process of defining the conceptual representation of data to be stored in the database.



Source: Wikipedia

BPM & DM

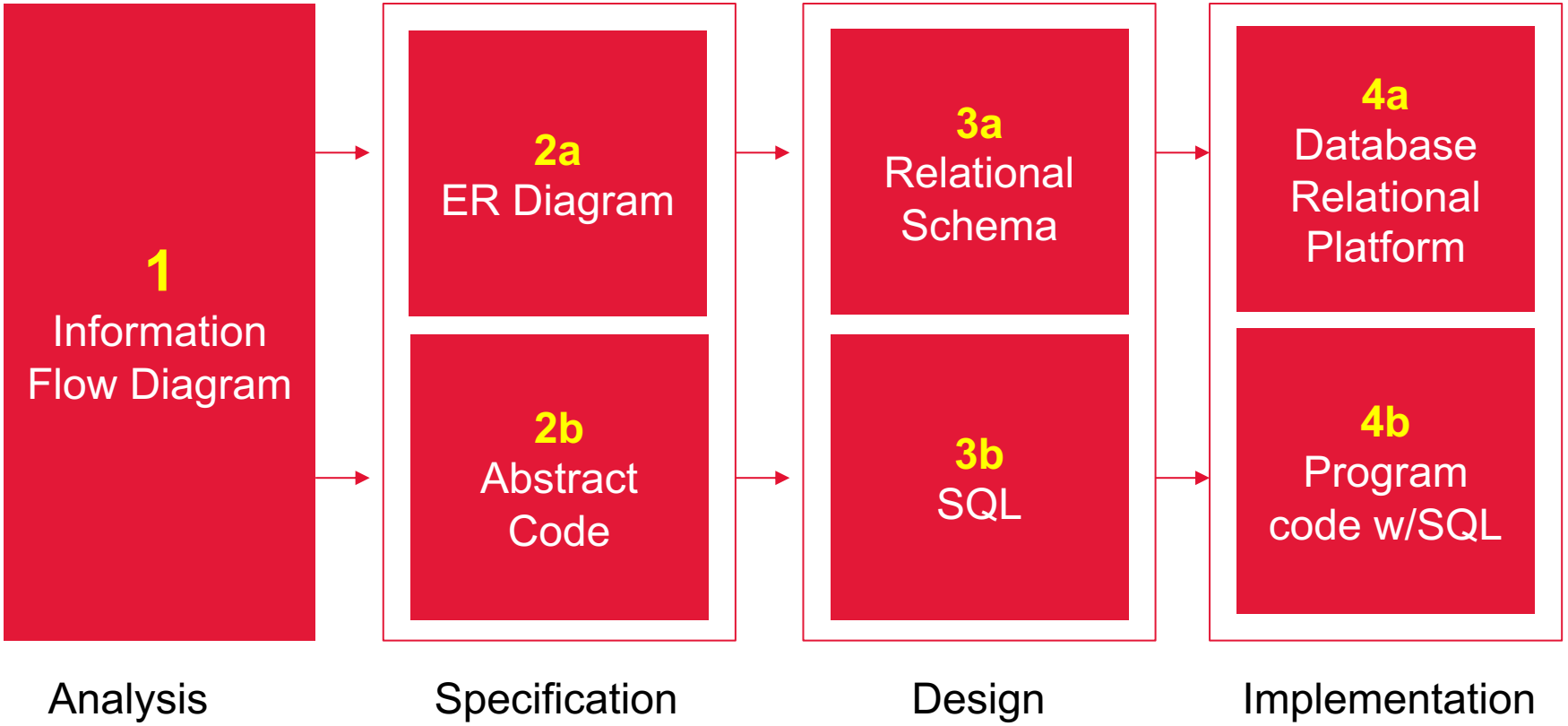
The two major techniques for DM are:

1. Entity Relationship Model (ER Model) – ER Diagram
2. UML (Unified Modelling Language) – IFD Diagram

Why and when do we use them?

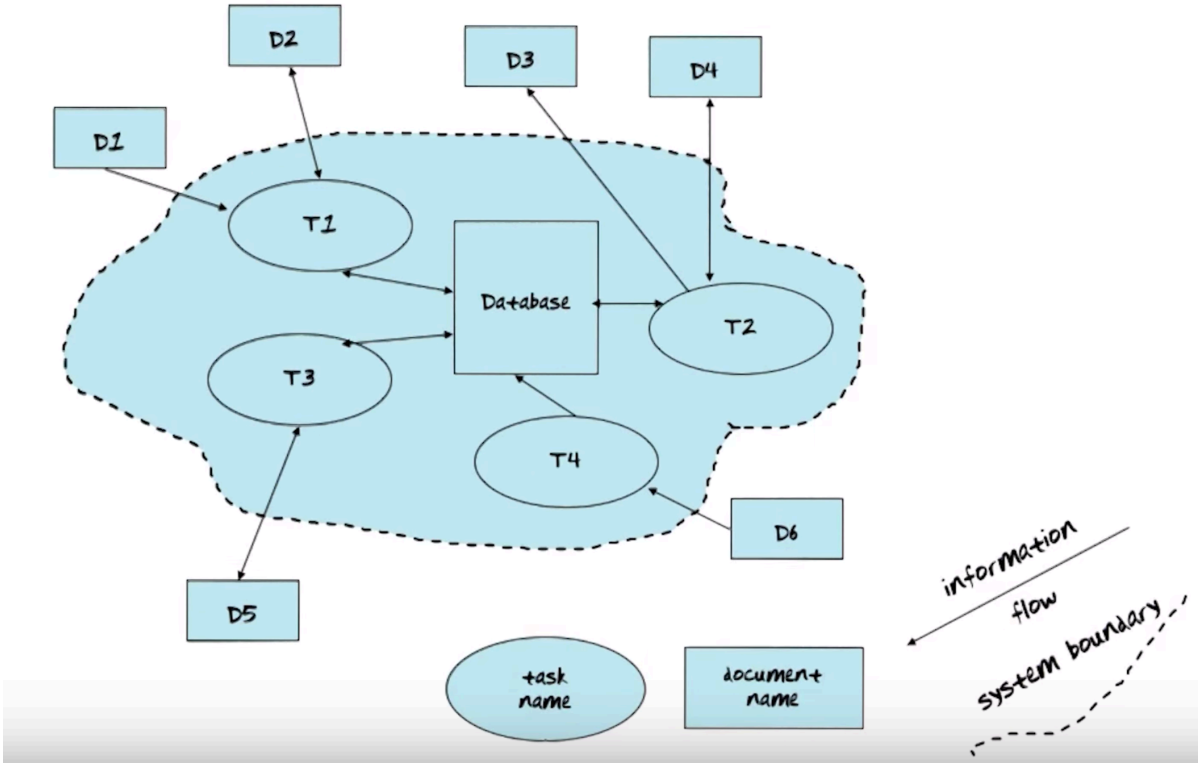
- Database design
- Database debugging
- Database creation and patching
- Requirements gathering

Methodology – Data First



IFD

The diagram shows the flow of information from source to destination. Its generally constructed as the UML diagram.



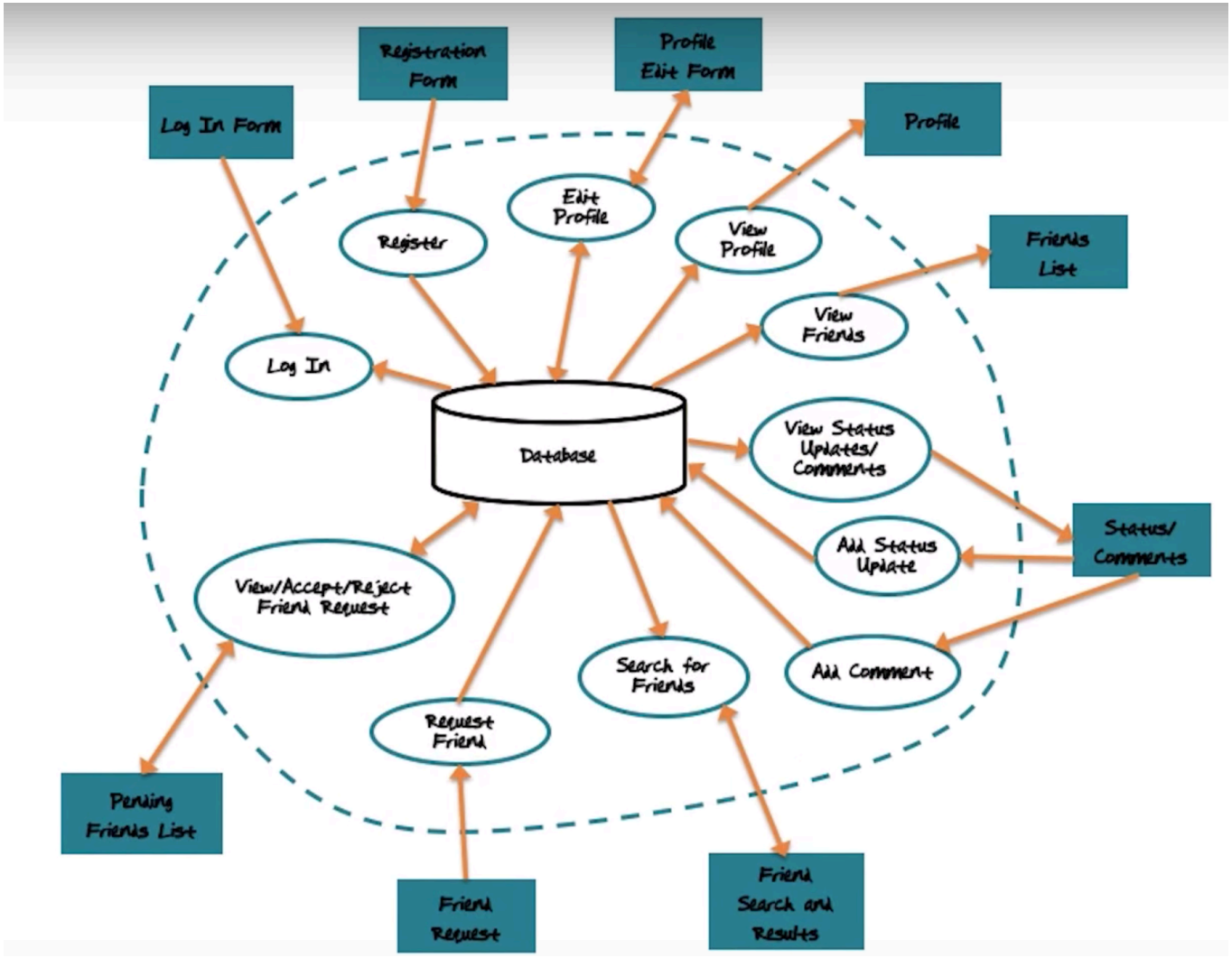
Group Activity

Using the notation described in the diagram construct the IFD for following use case:

YU Social Network

1. Login to YU Social with email address and password combination.
2. Users who are new must register first.
3. Everyone has a user profile containing basic information about them, once user logs in, the user is taken to the edit profile right away and that may include some basic information like sex, dob, city, town and interests.
4. The profile also has employment information (like LinkedIn)
5. The profile also has education information (like LinkedIn)
6. Search for friends and send request to connect to them.
7. View Friends show connected friends.
8. View Pending Requests show all pending requests.

Looks similar?



DB Components

Entity

An entity is a real world identifiable object. It can be a place, person, organization etc. For e.g. Student

Attribute

Attributes are properties that are used to describe an entity. For e.g. Roll number, Class, Marks etc.

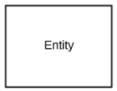
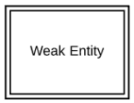
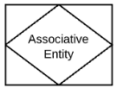
Relationship

The association among entities is called relationship. For e.g. A student enrolls in a course.

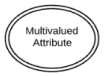
Domain

A domain is a set of values allowed for an attribute. For e.g. Marks in course is restricted to 1-100.

ER – Entity Relationship Diagrams

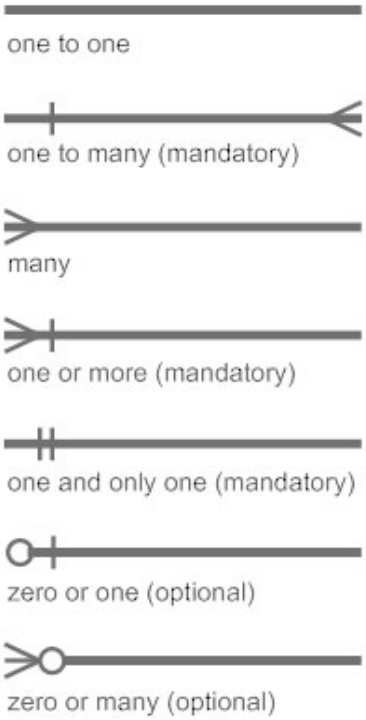
Entity Symbol	Name	Description
	Strong entity	These shapes are independent from other entities, and are often called parent entities, since they will often have weak entities that depend on them. They will also have a primary key, distinguishing each occurrence of the entity.
	Weak entity	Weak entities depend on some other entity type. They don't have primary keys, and have no meaning in the diagram without their parent entity.
	Associative entity	Associative entities relate the instances of several entity types. They also contain attributes specific to the relationship between those entity instances.

Relationship Symbol	Name	Description
	Relationship	Relationships are associations between or among entities.
	Weak relationship	Weak Relationships are connections between a weak entity and its owner.

Attribute Symbol	Name	Description
	Attribute	Attributes are characteristics of an entity, a many-to-many relationship, or a one-to-one relationship.
	Multivalued attribute	Multivalued attributes are those that are can take on more than one value.
	Derived attribute	Derived attributes are attributes whose value can be calculated from related attribute values.
	Relationship	Relationships are associations between or among entities.

ER – Entity Relationship Diagrams

Information Engineering Style



Martin Style

1 - one, and only one (mandatory)

* - many (zero or more - optional)

1...* - one or more (mandatory)

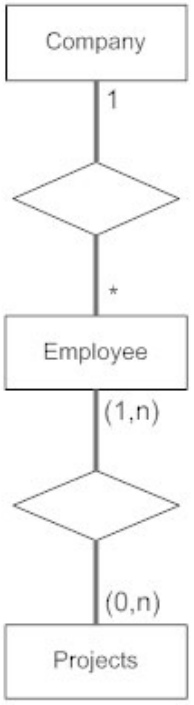
0...1 - zero or one (optional)

(0,1) - zero or one (optional)

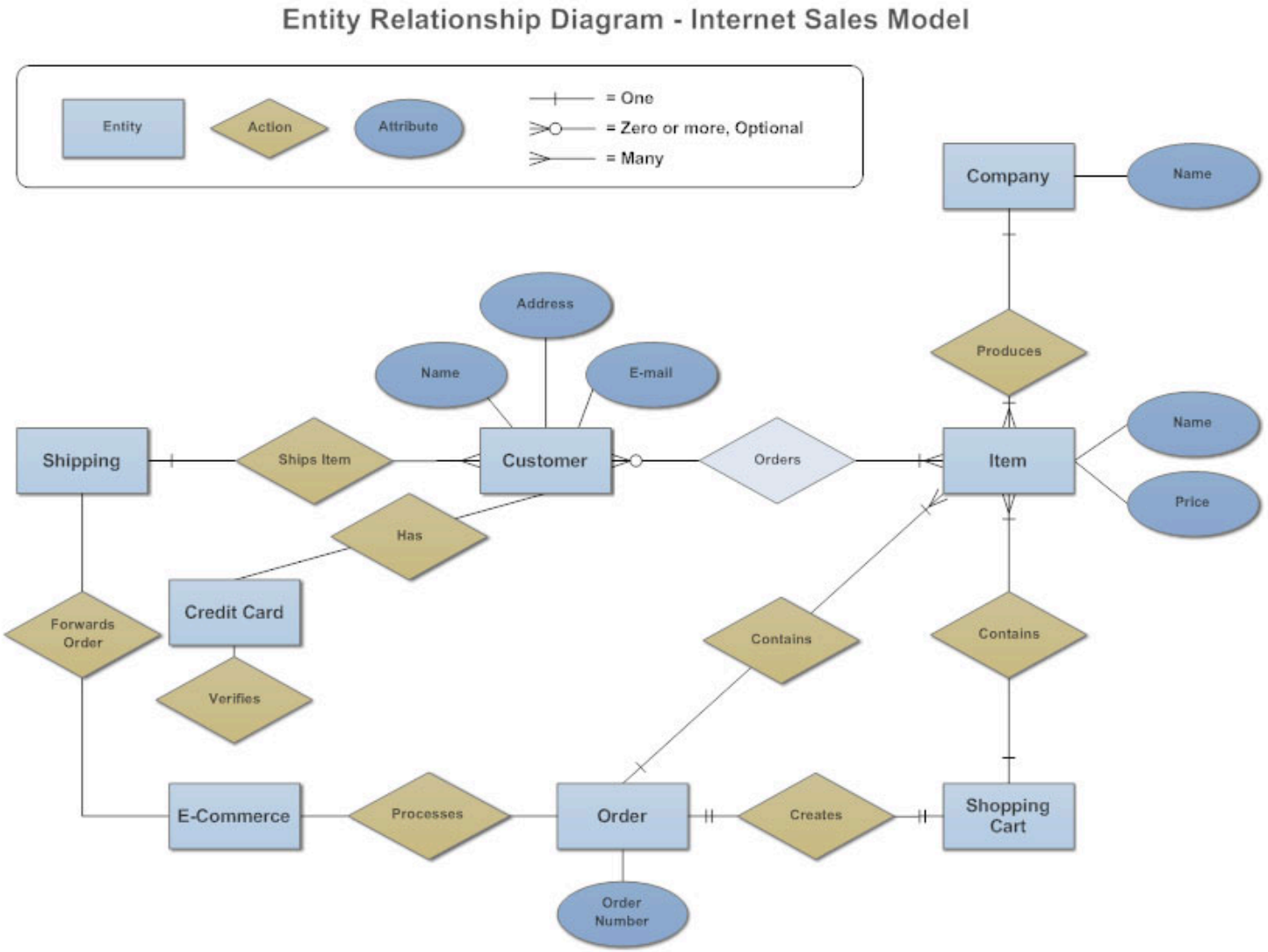
(1,n) - one or more (mandatory)

(0,n) - zero or more (optional)

(1,1) - one and only one (mandatory)



ER Diagram Example

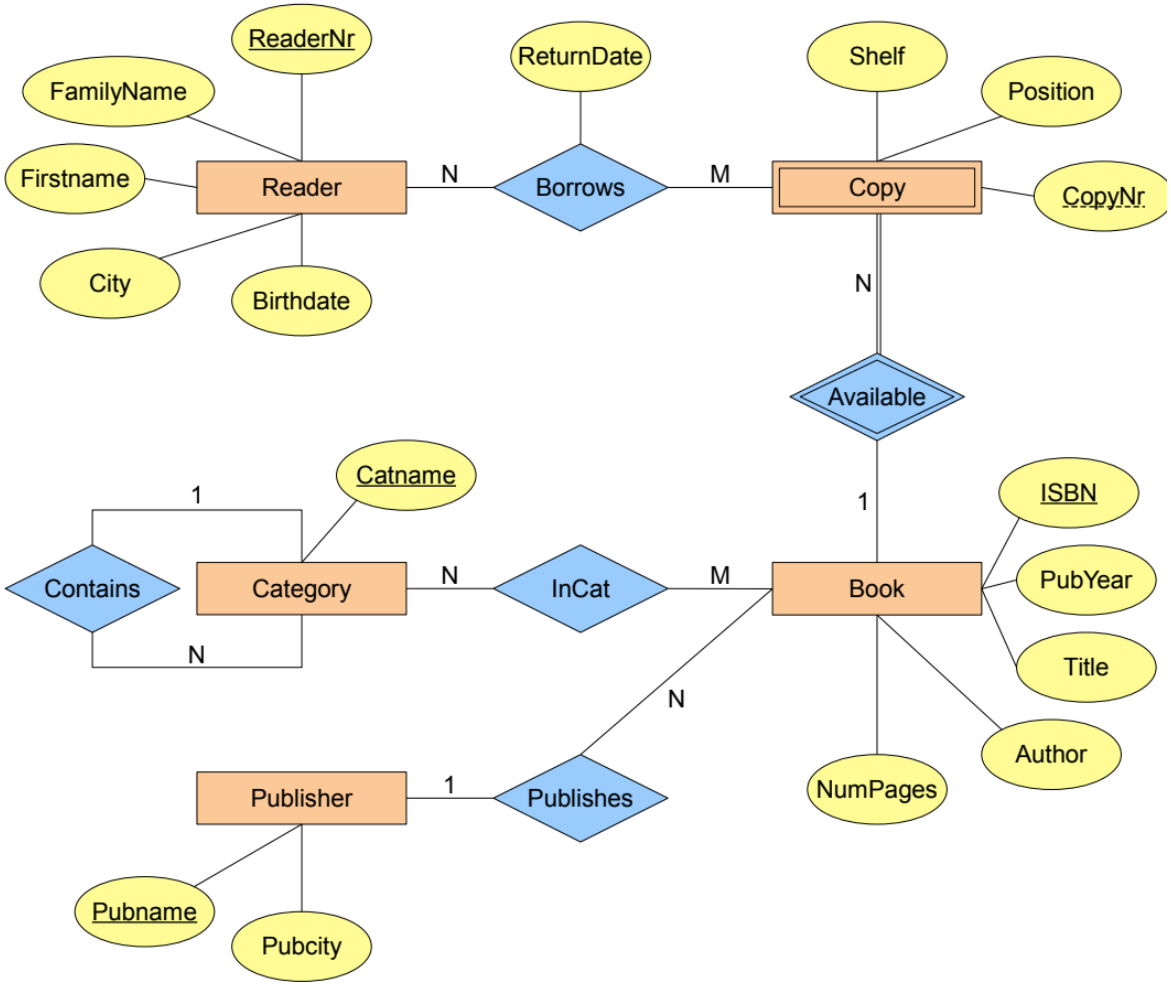


Source: smartdraw.com

Group Activity

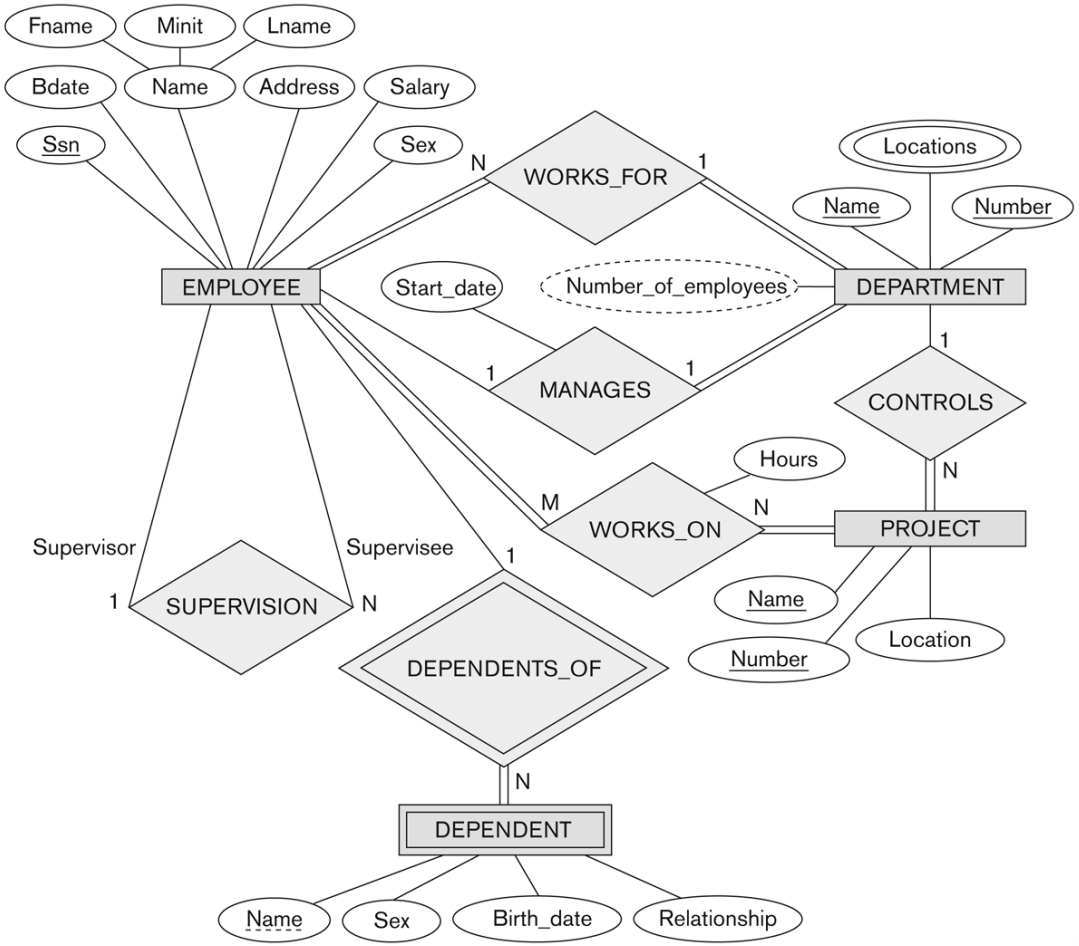
Assume there is a library system with the following properties. The library contains one or several copies of the same book. Every copy of a book has a copy number and is located at a specific location in a shelf. A copy is identified by the copy number and the ISBN number of the book. Every book has a unique ISBN, a publication year, a title, an author, and a number of pages. Books are published by publishers. A publisher has a name as well as a location. Within the library system, books are assigned to one or several categories. A category can be a subcategory of exactly one other category. A category has a name and no further properties. Each reader needs to provide his/her family name, his/her first name, his/her city, and his/her date of birth to register at the library. Each reader gets a unique reader number. Readers borrow copies of books. Upon borrowing the return date is stored. Create an ER diagram of this library system.

ER Solution


ReaderNr
FamilyName
Firstname
City
Birthdate
Reader
N
Borrows
M
ReturnDate
Shelf
Position
Copy
CopyNr
N
Available
1
Catname
1
Contains
Category
N
N
InCat
M
Book
ISBN
PubYear
Title
Author
NumPages
N
Publisher
1
Publishes
Pubname
Pubcity

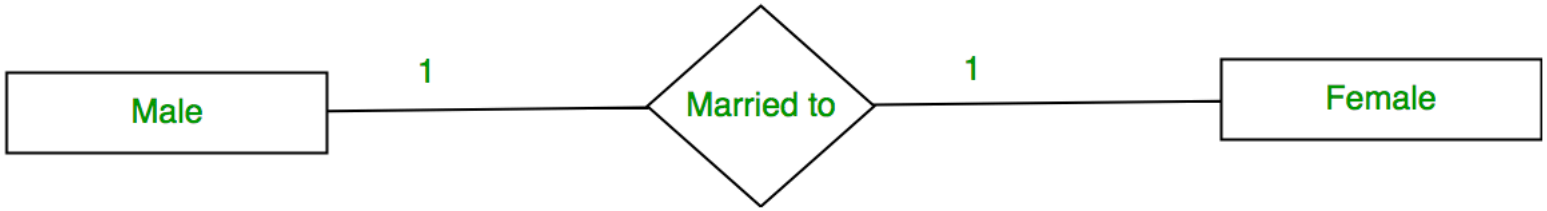
Group Activity

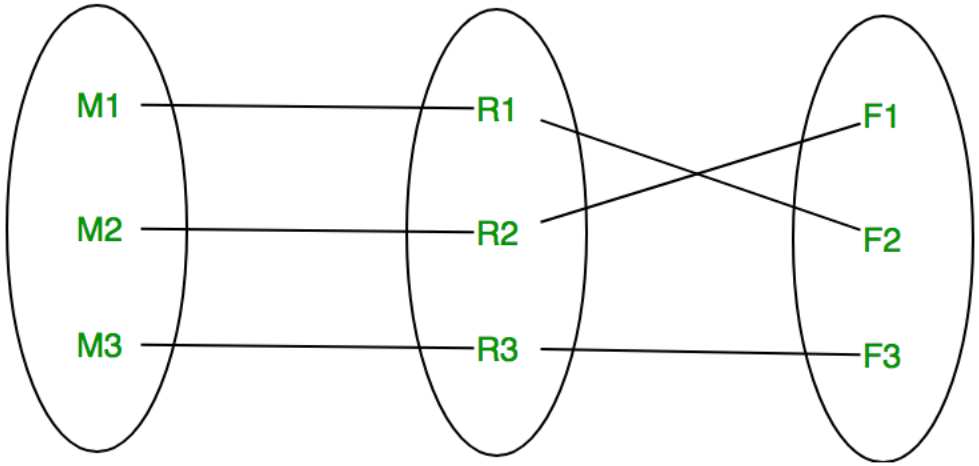
Check the diagram below and describe it in bullet points:



Relationships

One to one (1 to 1)

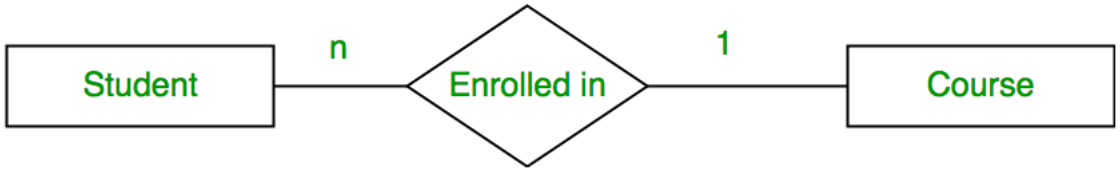


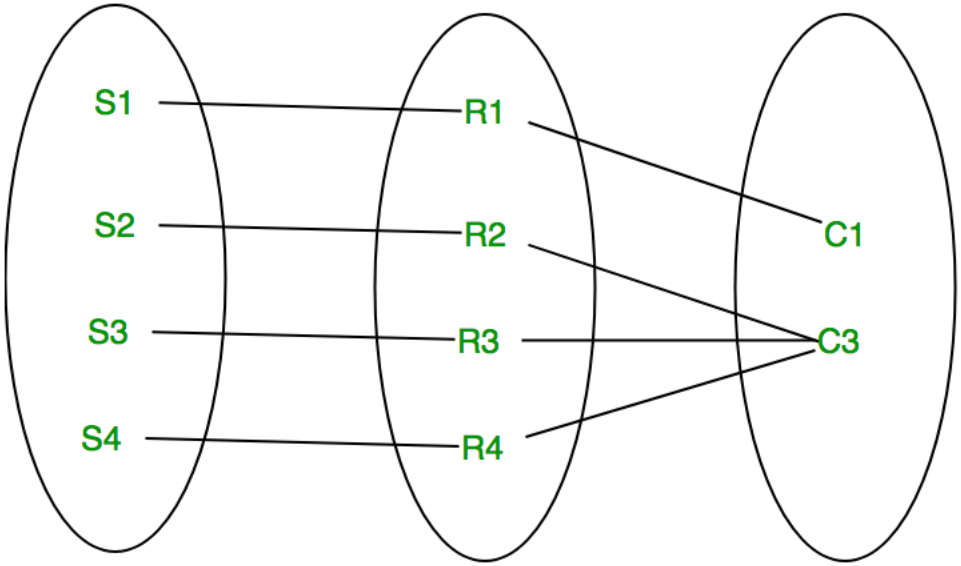


Source: Wikipedia

Relationships

Many to one (n to 1)

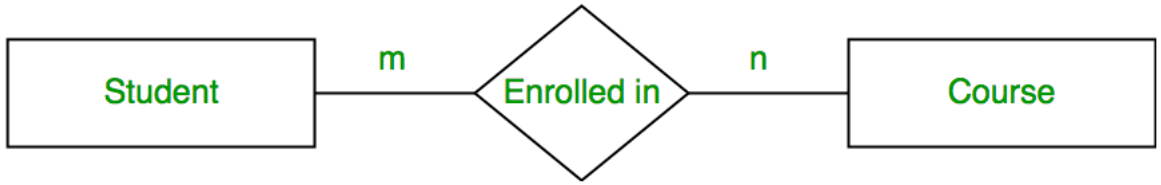


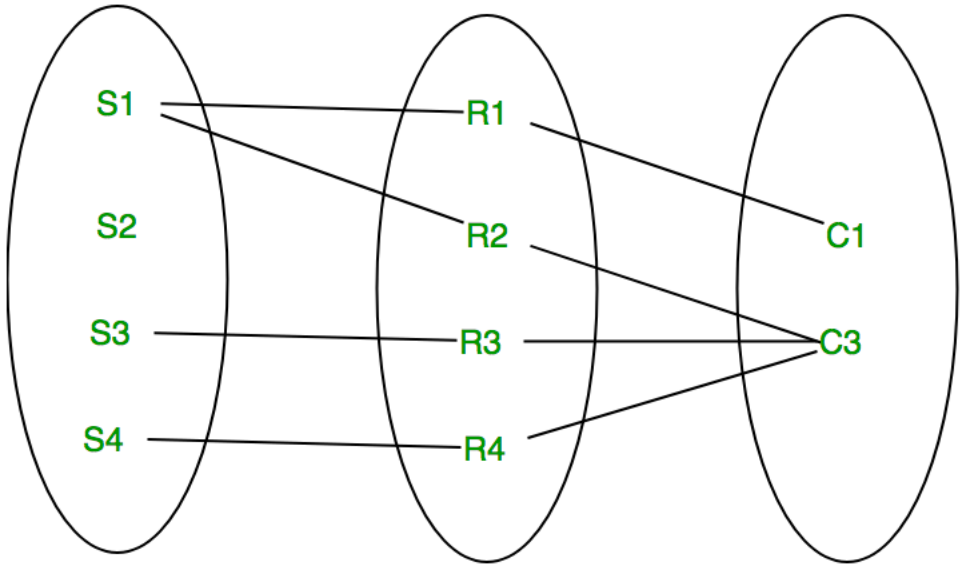


Source: Wikipedia

Relationships

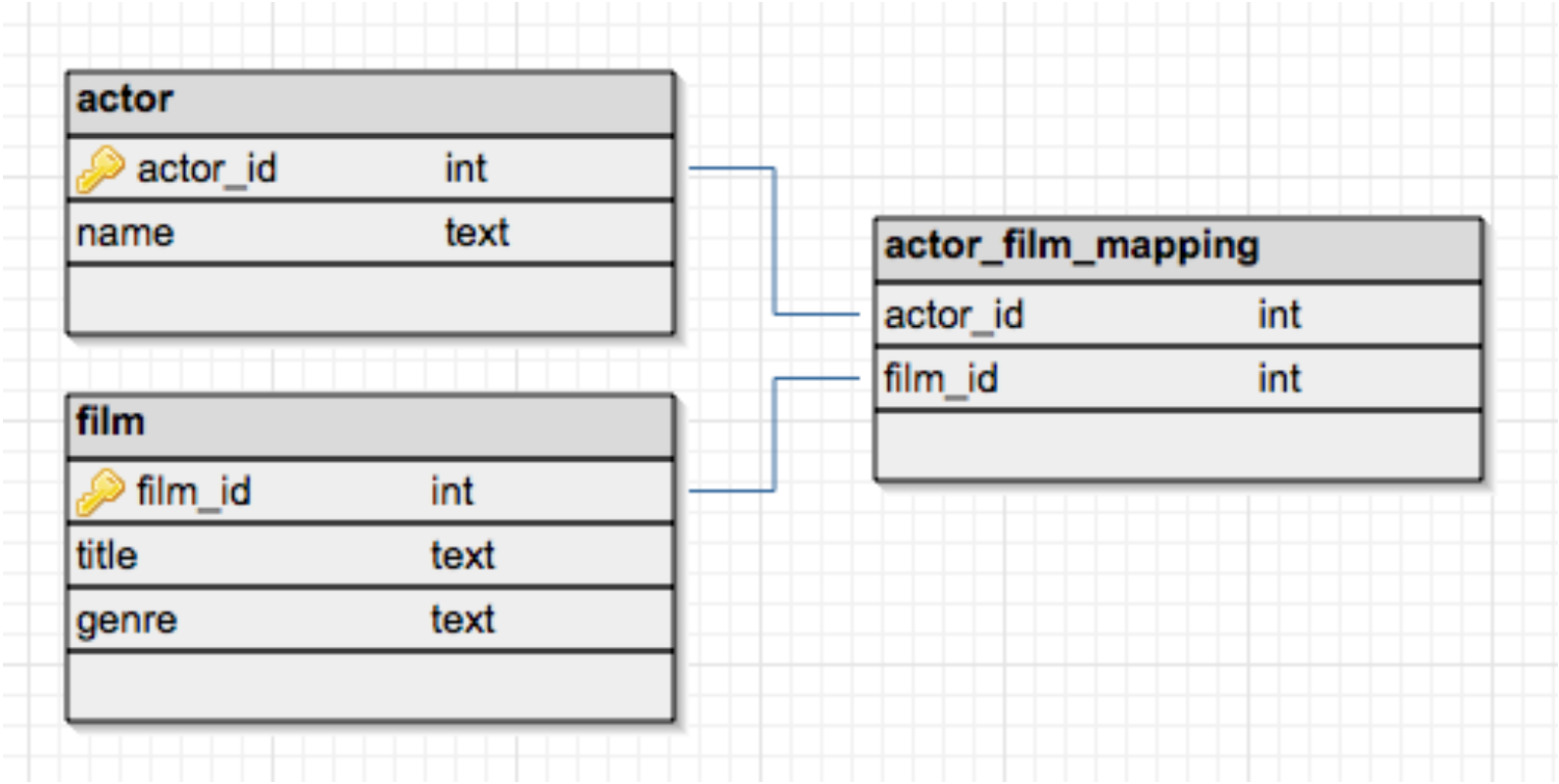
Many to Many (m to n)





Source: Wikipedia

Many to many – associative entity



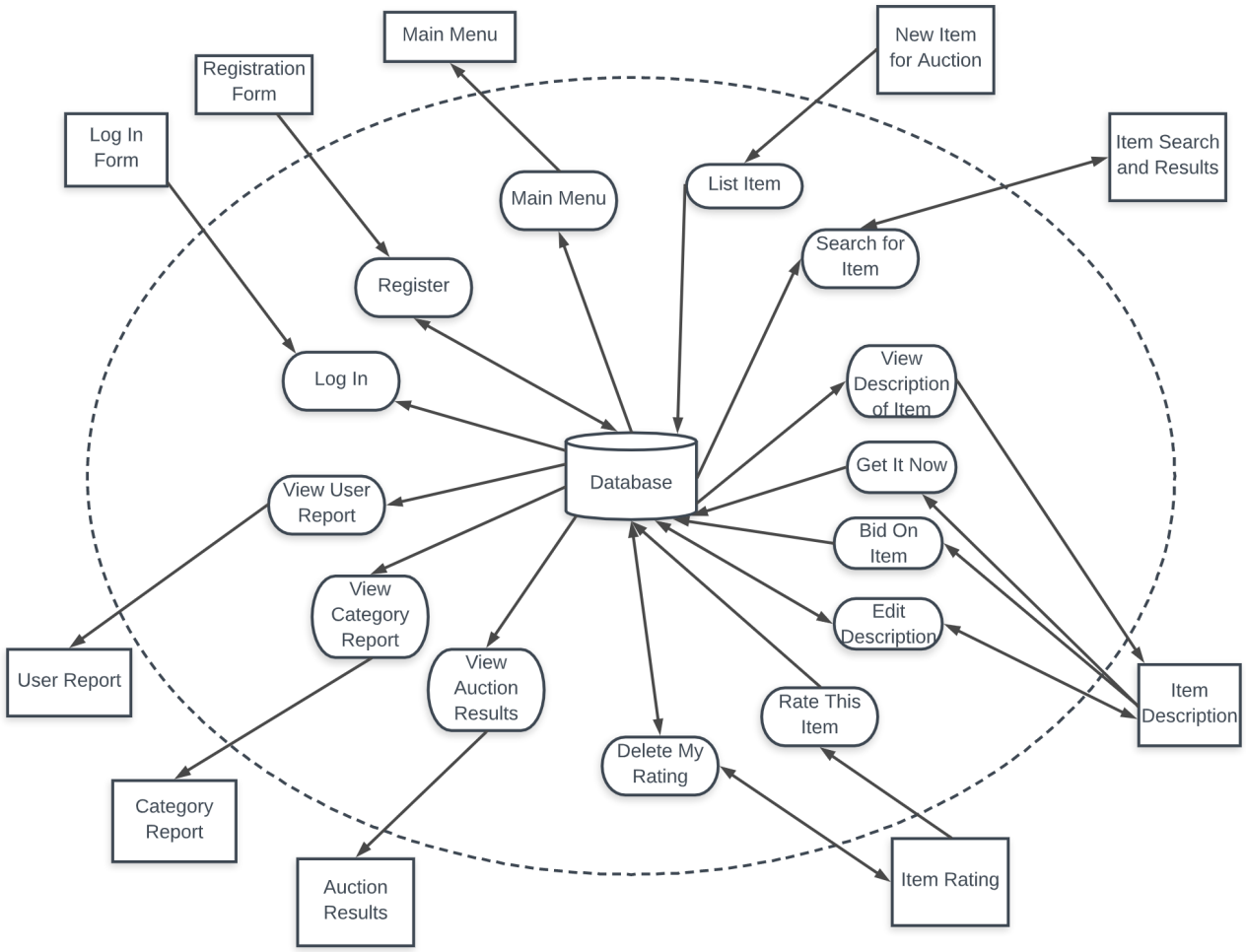
Source: Wikipedia

ER to Relational Schema

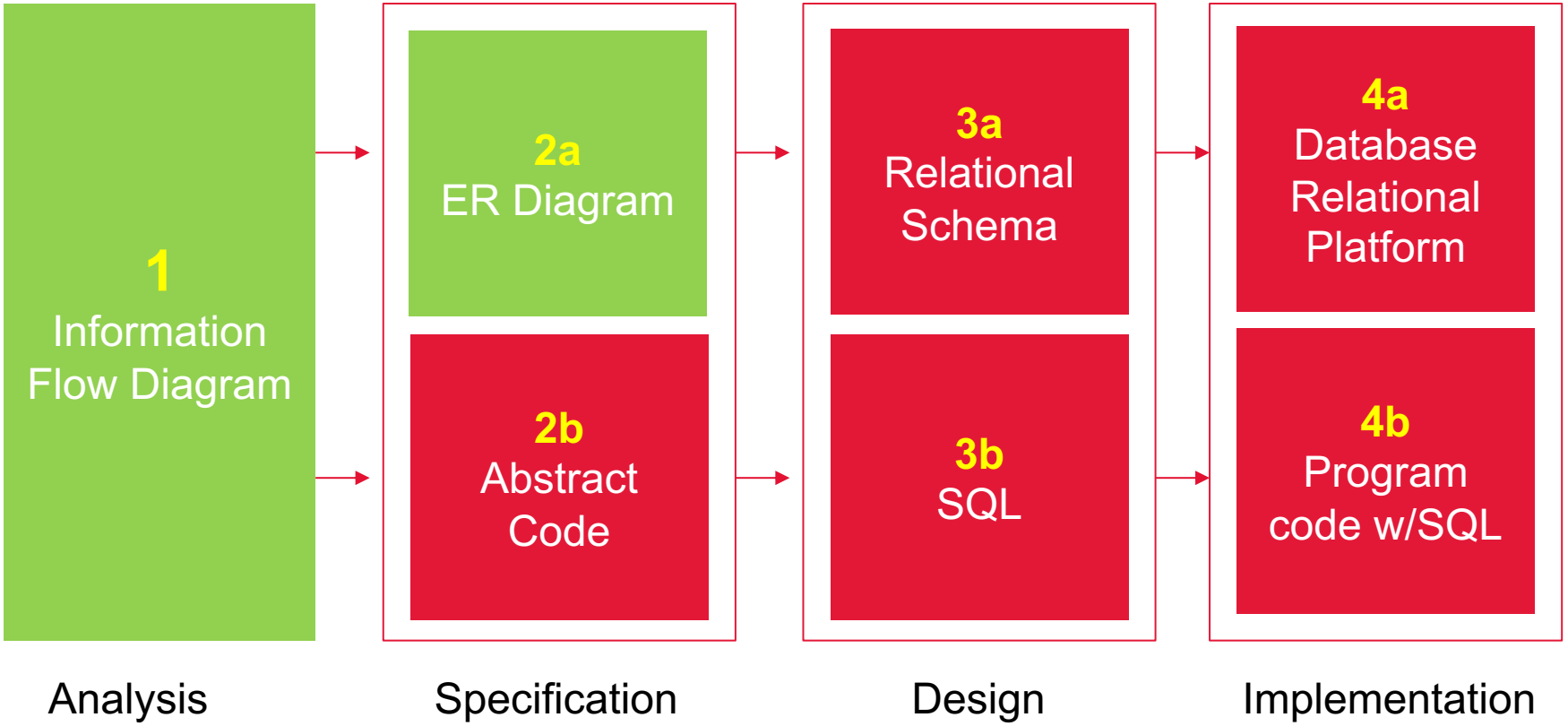
ER Element	Relational Schema Element(s)
Entity Set	Table
Simple Attribute	Attribute
Multivalued Attribute	Table and foreign key
1:1 or 1:N Relationship	Foreign key (or relationship table)
N:M Relationship	Relationship table and two foreign keys

Group Activity

Check the diagram below and describe it in bullet points:



Methodology – Data First



Resources

Create ER diagrams on draw.io – https://about.draw.io/entity-relationship-diagrams-with-draw-io/