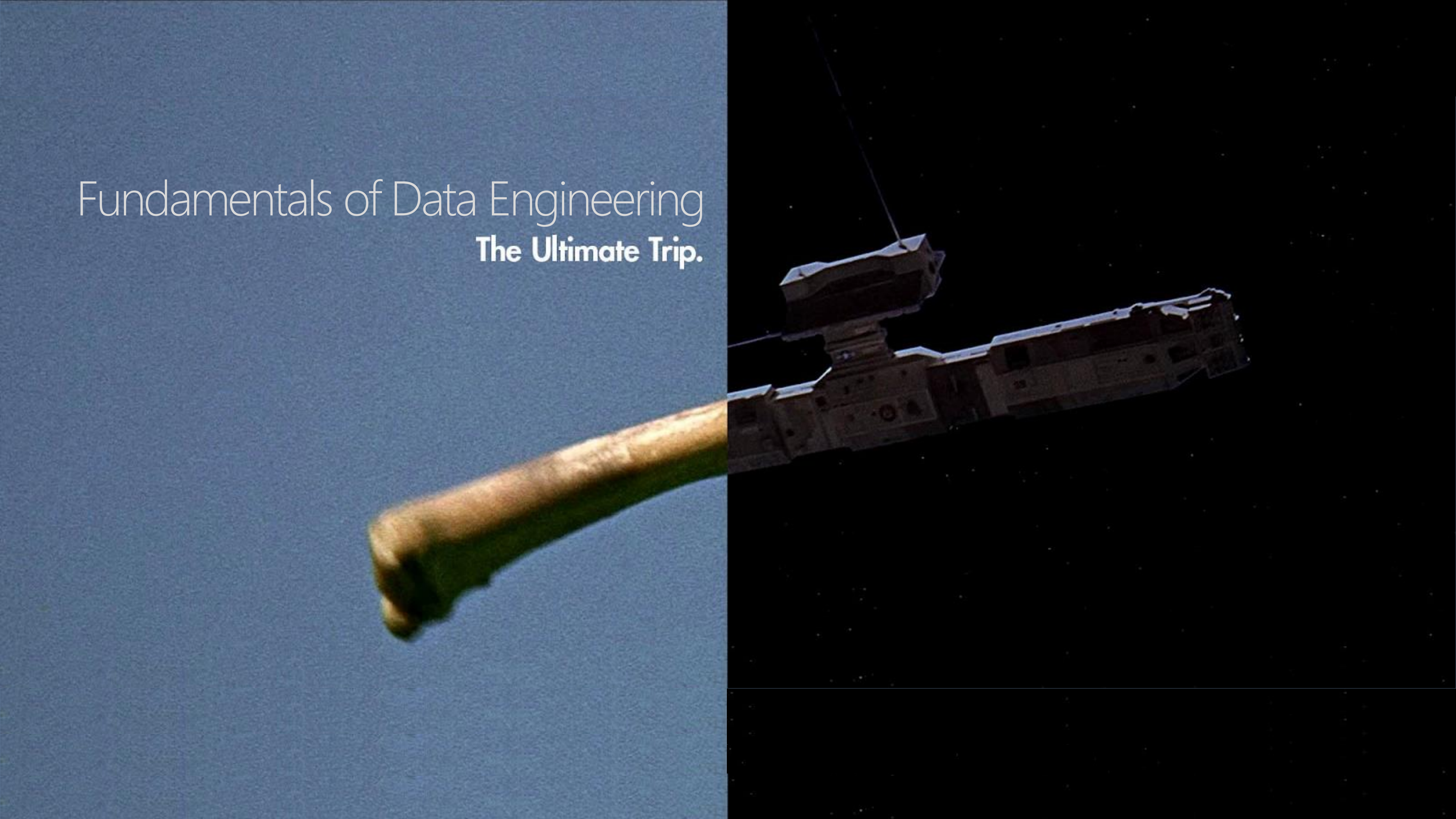
Fundamentals of Data Engineering



The Ultimate Trip.

Today

Welcome |

| Course Information |

| Data Modeling |

| Memory |

| File |

| Database

Course Information

Title	Fundamentals of Data Engineering
Code	COE848
Term	Winter 2019
Time	Tuesday 09AM-12PM
Location	KHE225

Lecturer

Name	Hossein Fani
Education	Ph.D. Candidate Faculty of Computer Science University of New Brunswick, NB., Canada
Office	ENG324
Hour	Tuesday 12PM-01PM
Email	hossein.fani@ryerson.ca

Grading

Labs and Final Project	35%
Midterm Exam (W08: Feb. 26, 2019)	25%
Final Exam	40%

Laboratory

Location	ENG411
Assignments	Steps of one final project
Deadlines	7 days <u>Late submission receives none!</u>

Laboratory × Deadline

Lab#	No. of Weeks	Due Date
Lab 1	1	Jan. 23
Lab 2	1	Jan. 30
Lab 3	2	Feb. 13
Lab 4	2	Mar. 06
Lab 5	2	Mar. 20
Lab 6	2	Apr. 03
All sections have same submission deadline, i.e., Thursdays Midnight		

Laboratory × TAs

Chengliang Huang
c28huang@ryerson.ca

Raman Deep Singh
raman.singh@ryerson.ca

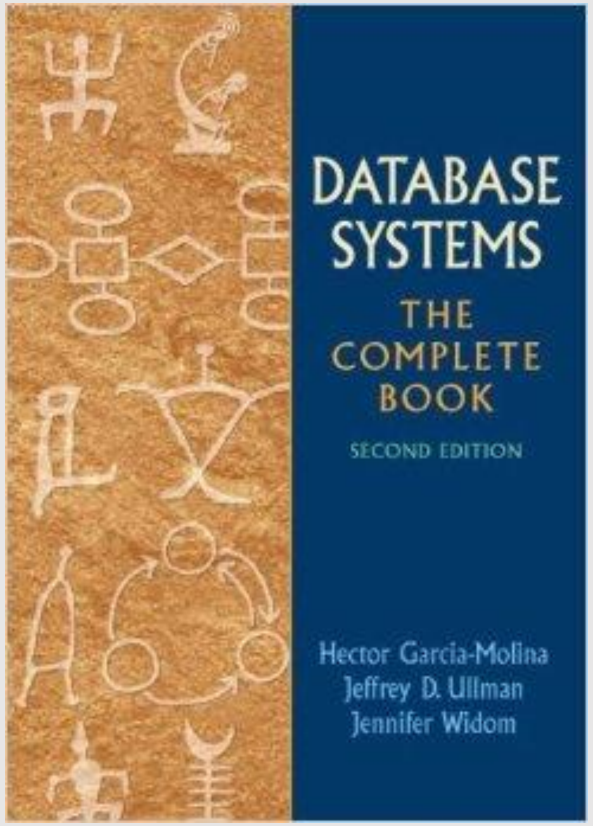
Laboratory × Time

	Monday	Tuesday	Wednesday	Thursday
09:00		C: Hossein KHE225		
10:00			L3: Raman ENG411	
11:00	L4: Chengliang ENG411			
12:00		O: Hossein ENG324		L2: Raman ENG411
13:00				
14:00	L1: Chengliang ENG411			
15:00				

L: Lab | C: Class | O: Office Hour

Book

Title	Database Systems: The Complete Book
Authors	Hector Garcia-Molina Jeffrey D. Ullman Jennifer D. Widom
Publisher	Prentice Hall
Edition	2nd Edition, 2008
ISBN	0131873253



DATABASE SYSTEMS

THE COMPLETE BOOK

SECOND EDITION

Hector Garcia-Molina
Jeffrey D. Ullman
Jennifer Widom

Homepage (D2L)

Course Outline	Done
Lab Guide	Done
Lab Deadlines	Done
Slides	End of Each Week
Announcements	Check D2L for course updates & information!

Homepage (D2L) Tour

Running Example

Title	Moviesion
Use Case	Similar Movie Recommendation
Test Case	Input: “2001: A Space Odyssey” Output: 1. “Interstellar” 2. “Her” 3. “Metropolis” 4. …

Programming Language

Name	Java (why?)
IDE	NetBeans
Code Convention	www.oracle.com/technetwork/java/codeconventions-150003.pdf
Naming Convention	www.oracle.com/technetwork/java/codeconventions-135099.html

Programming Language × Naming

Package	all.lower.case `package ca.ryerson.ee.coe848`

Class	Singular Noun, PascalCase
	`public class Movie{}`
	`public class MovieProductionCompany{}`

Variable	Noun, camelCase
	`Movie oscarBestPicture = new Movie();`
	`ArrayList<Movie> oscarWinners = new ArrayList<>();`

Method	Verb, camelCase `public void addMovie(Movie newMovie);`
Constant	ALL UPPER CASE `final int MAX_MOVIE_TITLE_WIDTH = 255;`

Attendance

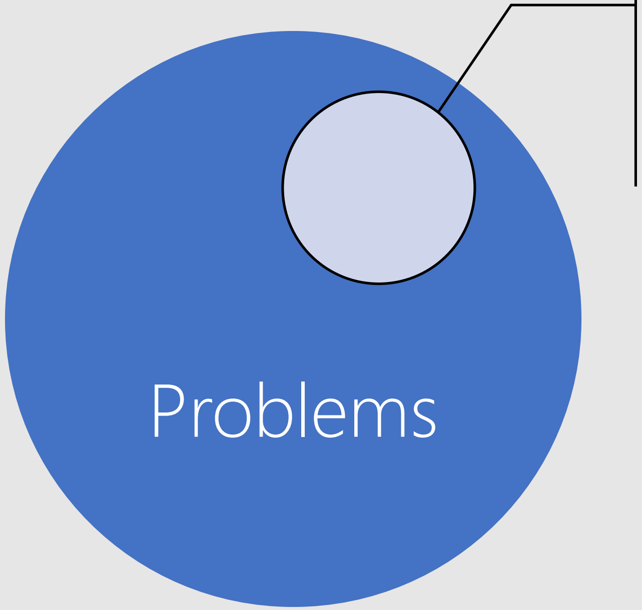
Policy	KINDLY DO NOT DISTRACT OTHERS!
Phone	Off Silent
Laptop	Closed Far End of Each Row
Break	10:30 - 10:45
Others	www.ryerson.ca/studentguide/learning_experience/

2019: A DATA ODYSSEY

2019: A Data Odyssey

Problems

2019: A Data Odyssey

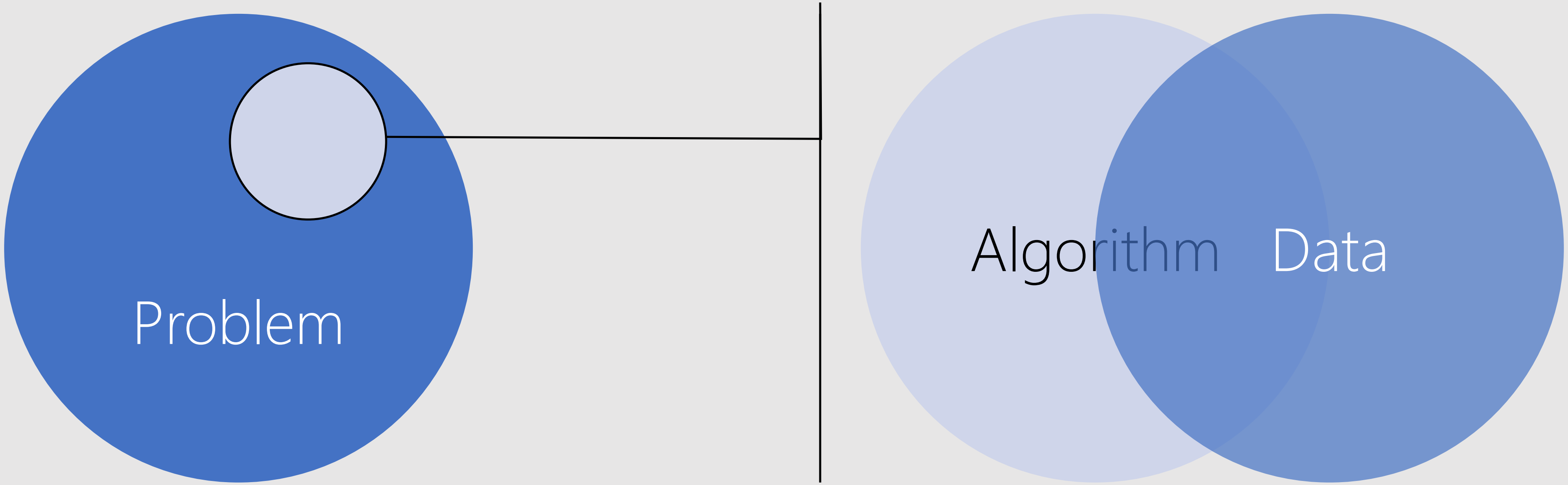


Computable

Theory of Computation

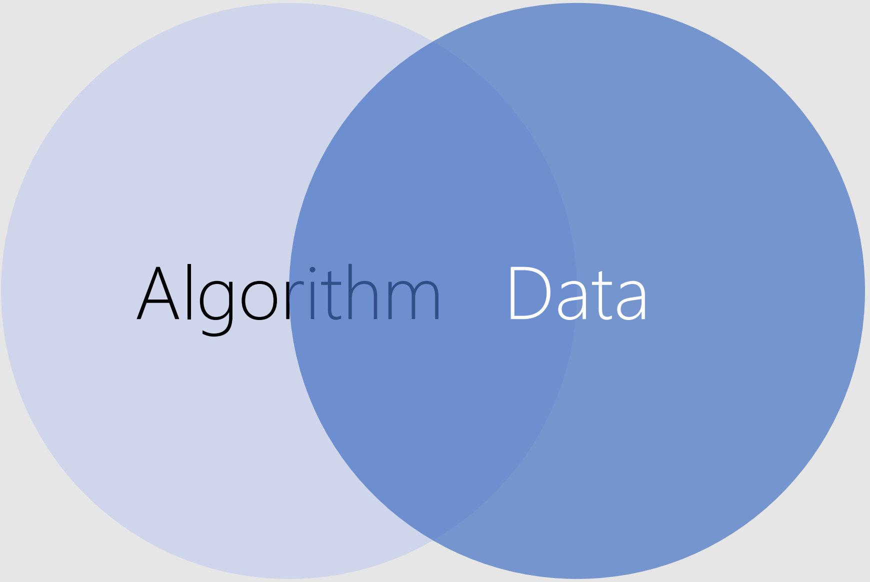
Theory of Computer Science

2019: A Data Odyssey



2019: A Data Odyssey

There is interaction between data engineers and algorithm designers.

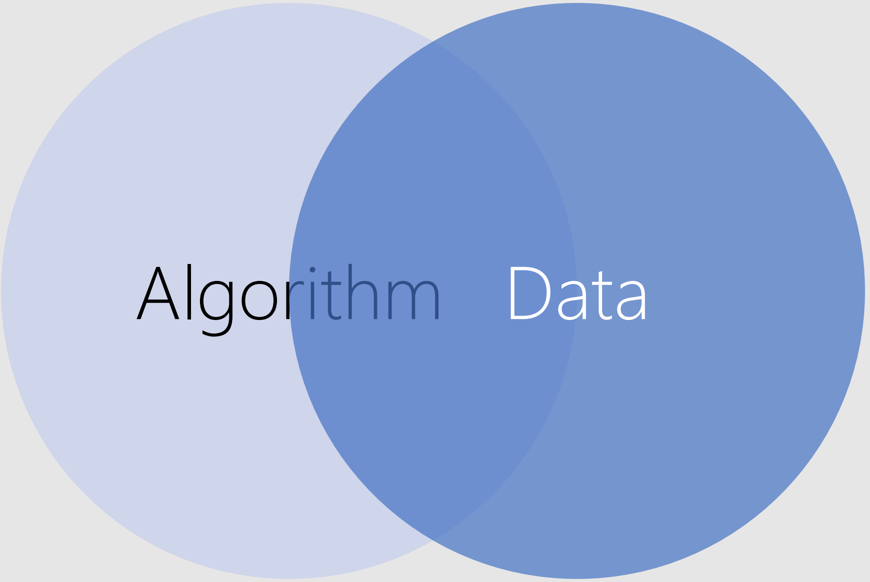


Data gets bigger!
Algorithms get slower!

Better Data Modeling → Faster Algorithm
e.g., ?

2019: A Data Odyssey

There is interaction between data engineers and algorithm designers.

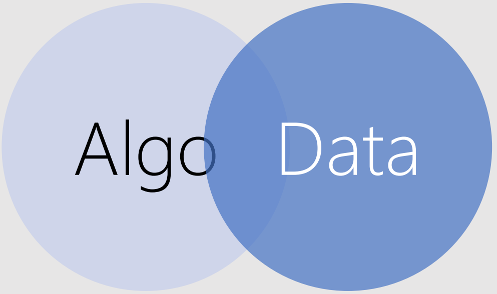


Data gets bigger!
Algorithms get slower!

Better Data Modeling → Faster Algorithm
e.g., Sorting in Arrays vs. Trees
e.g., Searching in Arrays vs. Trees
e.g., Sparse Matrix in Arrays vs. Linked Lists

2019: A Data Odyssey × Academy

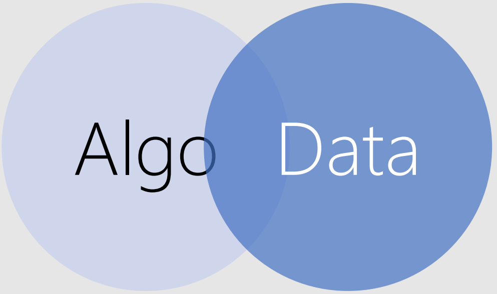
Algorithm Design
Algorithm Analysis
Artificial Intelligence (AI)
Machine Learning
Data Mining



Data Structure (Memory)
File Structure (File Systems)
Database Management Systems
Data Warehouse
Big Data
Cloud

2019: A Data Odyssey × Academy

Algorithm Design
Algorithm Analysis
Artificial Intelligence (AI)
Machine Learning
Data Mining



Data Structure (Memory)
File Structure (File Systems)
Database Management Systems
Data Warehouse
Big Data
Cloud

2019: A Data Odyssey × Real World

Data Modeling: Real World Entity

Conceptual Level |

| Logical Level |

| Physical Level

Computable Entity

Data Modeling × Conceptual Level

1. Identify real world entities, attributes, relationships
2. Create Schema

Data Modeling × Conceptual Level

Entities for a movie recommender system:

Movie

2001: a space odyssey
MGM PRESENTS A STANLEY KUBRICK PRODUCTION
Super Panavision® and Metrocolor

Director

STANLEY KUBRICK

Company

MGM

An epic drama of adventure and exploration
Space Station One: your first step in an Odyssey that will take you to the Moon, the planets and the distant stars.
2001: a space odyssey
MGM PRESENTS A STANLEY KUBRICK PRODUCTION
Super Panavision® and Metrocolor

Data Modeling × Conceptual Level

Movie Attributes:

- Title
- Genre
- Language
- RunningTime
- Poster
- …

An epic drama of adventure and exploration
Space Station One: your first step in an Odyssey that will take you to the Moon, the planets and the distant stars.
2001: a space odyssey
MGM PRESENTS A STANLEY KUBRICK PRODUCTION
Super Panavision® and Metrocolor
MGM

Data Modeling × Conceptual Level

Movie Attributes’ Type (More Optional Details):

- Title (`string`)
- Genre (`string`)
- Language (`string`)
- RunningTime (`decimal`)
- Poster (?)
- …

An epic drama of adventure and exploration
Space Station One: your first step in an Odyssey that will take you to the Moon, the planets and the distant stars.
MGM PRESENTS A STANLEY KUBRICK PRODUCTION
2001: a space odyssey
Super Panavision® and Metrocolor

Data Modeling × Conceptual Level

Movie Attributes’ Type (More Optional Details):

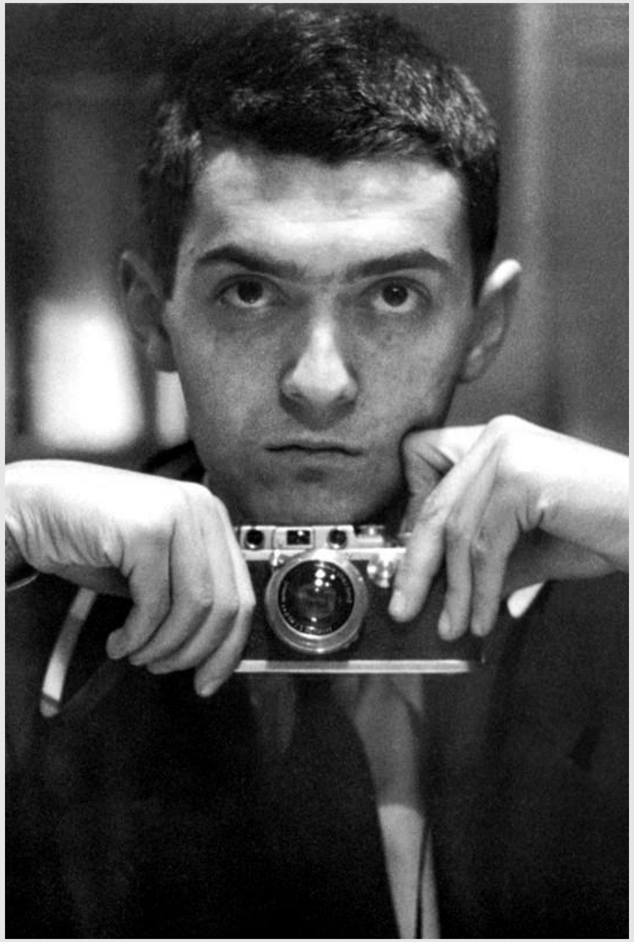
- Title (`string`)
- Genre (`string`)
- Language (`string`)
- RunningTime (`decimal`)
- Poster (`byte[]`)
- …

An epic drama of adventure and exploration
Space Station One: your first step in an Odyssey that will take you to the Moon, the planets and the distant stars.
2001: a space odyssey
MGM PRESENTS A STANLEY KUBRICK PRODUCTION
Super Panavision® and Metrocolor

Data Modeling × Conceptual Level

Director Attributes:

- Name (string)
- DateOfBirth (date)
- PlaceOfBirth (string)
- Photo (byte[])
- …



Data Modeling × Conceptual Level

Company Attributes:

- Name (`string`)
- Address (`string`)
- DateOfEstablishment (`date`)
- Logo (`byte[]`)
- …

Data Modeling × Conceptual Level

Movie Schema	Movie Data <u>Instance</u>
Title (string)	*2001: A Space Odyssey*
Genre (string)	*Sci-fi*
Language (string)	*English*
RunningTime (decimal)	*142*
Poster (byte[])	An epic drama of adventure and exploration 2001: a space odyssey

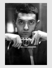
Data Modeling × Conceptual Level

Movie Schema	Movie Data Instance
Title (`string`)	*Rosemary's Baby*
Genre (`string`)	*Horror*
Language (`string`)	*English*
RunningTime (`decimal`)	*136*
Poster (`byte[]`)	

Data Modeling × Conceptual Level

Movie Schema	Title (`string`)	Genre (`string`)	Language (`string`)	RunningTime (`decimal`)	Poster (`byte[]`)
Movie Instance 1	*2001: A Space Odyssey*	*Sci-fi*	*English*	*142*	
Movie Instance 2	*Rosemary's Baby*	*Horror*	*English*	*136*	

Data Modeling × Conceptual Level

Director Schema	Name (`string`)	DateOfBirth (`date`)	PlaceOfBirth (`date`)	Photo (`byte[]`)
Director Instance 1	*Stanley Kubrick*	*1949*	*USA*	
Director Instance 2	*Roman Polanski*	*1933*	*Poland*	

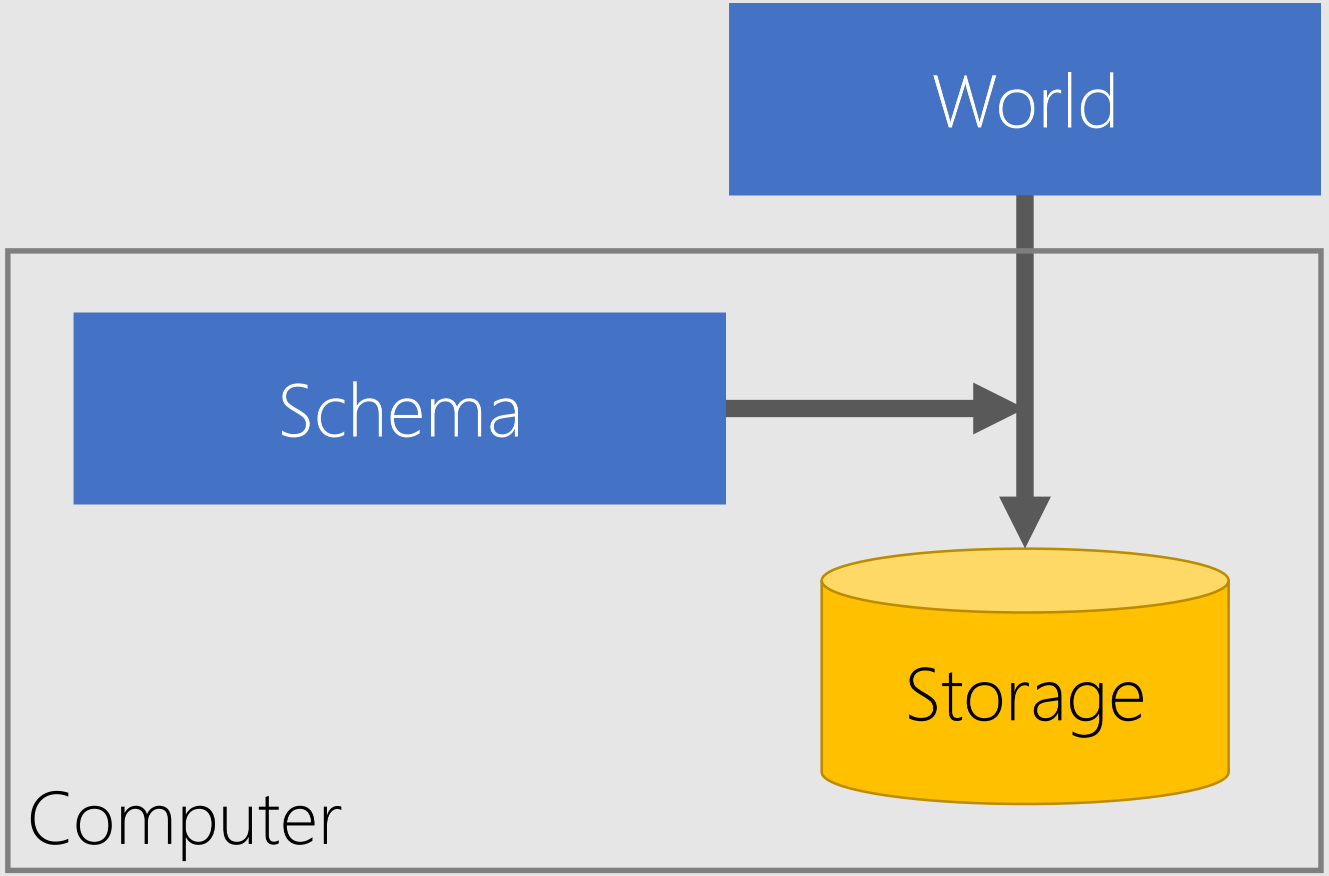
Data Modeling × Conceptual Level

Company Schema	Name (`string`)	Address (`string`)	DateOfEstablishment (`date`)	Logo (`byte[]`)
Company Instance 1	*Metro-Goldwyn-Mayer Studios Inc.*	*Beverly Hills, CA, USA*	*1924*	
Company Instance 2	*Warner Bros*	`null`	*1923*	

Data Modeling × Conceptual Level

Data Schema	Data Instance
Data about Data: Meta-data	*Actual Data*
Defined at Setup Time	*Inserted at Running Time*
Rarely Change	*Rapidly Change*
	Must Conform to Schema

Data Modeling × Conceptual Level



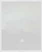
Data Modeling × Conceptual Level

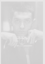
Movie Schema	Title (string)	Genre (string)	Language (string)	RunningTime (decimal)	Poster (byte[])
Movie Instance 1	*2001: A Space Odyssey*	*Sci-fi*	*English*	*142*	
Movie Instance 2	*Rosemary's Baby*	*Horror*	*English*	*136*	

Director Schema	Name (string)	DateOfBirth (date)	PlaceOfBirth (date)	Photo (byte[])
Director Instance 1	*Stanley Kubrick*	*1949*	*USA*	
Director Instance 2	*Roman Polanski*	*1933*	*Poland*	

Company Schema	Name (string)	Address (string)	DateOfEstablishment (date)	Logo (byte[])
Company Instance 1	*Metro-Goldwyn-Mayer Studios Inc.*	*Beverly Hills, CA, USA*	*1924*	
Company Instance 2	*Warner Bros*	null	*1923*	

Data Modeling × Conceptual Level

Movie Schema	Title (string)	Genre (string)	Language (string)	RunningTime (decimal)	Poster (byte[])
Movie Instance 1	*2001: A Space Odyssey*	*Sci-fi*	*English*	*142*	
Movie Instance 2	*Rosemary's Baby*	*Horror*	*English*	*136*	

Director Schema	Name (string)	DateOfBirth (date)	PlaceOfBirth (date)	Photo (byte[])
Director Instance 1	*Stanley Kubrick*	*1949*	*USA*	
Director Instance 2	*Roman Polanski*	*1933*	*Poland*	

Company Schema	Name (string)	Address (string)	DateOfEstablishment (date)	Logo (byte[])
Company Instance 1	*Metro-Goldwyn-Mayer Studios Inc.*	*Beverly Hills, CA, USA*	*1924*	
Company Instance 2	*Warner Bros*	null	*1923*	

No Relationship!

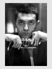
Movie × Director × Company

Data Modeling × Conceptual Level

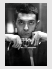
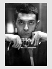
Movie × Director × Company Relationships:

Director makes Movie	Movie is made by Director
Company distributes Movie	Movie is distributed by Company
Company budgets Movie	Movie is budgeted by Company

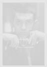
Data Modeling × Conceptual Level

Movie Schema	Title (string)	Genre (string)	Language (string)	RunningTime (decimal)	Poster (byte[])	DirectedBy (Director Schema)			
						Name (string)	DateOfBirth (date)	PlaceOfBirth (date)	Photo (byte[])
Movie Instance 1	*2001: A Space Odyssey*	*Sci-fi*	*English*	*142*		*Stanley Kubrick*	*1949*	*USA*	
Movie Instance 2	*Rosemary's Baby*	*Horror*	*English*	*136*		*Roman Polanski*	*1933*	*Poland*	

Data Modeling × Conceptual Level

Movie Schema	Title (string)	Genre (string)	Language (string)	RunningTime (decimal)	Poster (byte[])	DirectedBy (Director Schema)			
						Name (string)	DateOfBirth (date)	PlaceOfBirth (date)	Photo (byte[])
Movie Instance 1	*2001: A Space Odyssey*	*Sci-fi*	*English*	*142*		***Stanley Kubrick***	***1949***	***USA***	
Movie Instance 2	*Rosemary's Baby*	*Horror*	*English*	*136*		*Roman Polanski*	*1933*	*Poland*	
Movie Instance 3	*A Clockwork Orange*	*Sci-fi*	*English*	*136*		***Stanley Kubrick***	***1949***	***USA***	

Data Modeling × Conceptual Level

Movie Schema	Title (string)	Genre (string)	Language (string)	RunningTime (decimal)	Poster (byte[])	DirectedBy (Director Schema)			
						Name (string)	DateOfBirth (date)	PlaceOfBirth (date)	Photo (byte[])
Movie Instance 1	*2001: A Space Odyssey*	*Sci-fi*	*English*	*142*		*Stanley Kubrick*	*1949*	*USA*	
Movie Instance 2	*Rosemary's Baby*	*Horror*	*English*	*136*		*Roman Polanski*	*1933*	*Poland*	
Movie Instance 3	*A Clockwork Orange*	*Sci-fi*	*English*	*136*		*Stanley Kubrick*	*1949*	*USA*	

Redundancy!

Not All Data Models at Conceptual Level are Efficient.

Not All Conceptual Models are Efficient.

Not All Data Models are Efficient.

Data Modeling × Conceptual Level

Practice1: Data Modeling for Ryerson Library at Conceptual Level

Entities: ?
Attributes: ?
Relationships: ?

Data Modeling × Conceptual Level

Practice1: Data Modeling for Ryerson <u>Library</u> at Conceptual Level

Entities: Book, Staff, Person, …
Attributes:
Book: ISBN, Title, …
Staff: Name, Salary, Rank, …
Student: StudentId, Name, …
Relationships: Student <u>borrows</u> Book from Staff

Data Modeling × Conceptual Level

Practice2: Data Modeling for Ryerson <u>University</u> at Conceptual Level

Entities: ?
Attributes: ?
Relationships: ?

2019: A Data Odyssey × Real World

Data Modeling: Real World Entity

Conceptual Level |

| Logical Level |

| Physical Level

Computable Entity

Data Modeling × Logical Level

1. How entities, attributes, relationships should be <u>represented</u>.
2. <u>Update</u> Schema

Data Modeling × Logical Level

We already have seen one representation, haven't we?

Data Modeling × Logical Level

We already have seen one representation, haven’t we? Tabular!

Movie Schema	Title (`string`)	Genre (`string`)	Language (`string`)	RunningTime (`decimal`)	Poster (`byte[]`)
Movie Instance 1	*2001: A Space Odyssey*	*Sci-fi*	*English*	*142*	
Movie Instance 2	*Rosemary's Baby*	*Horror*	*English*	*136*	

Data Modeling × Logical Level

There are other representations as well.
There are other logical models as well.
There are other data models at logical level as well.

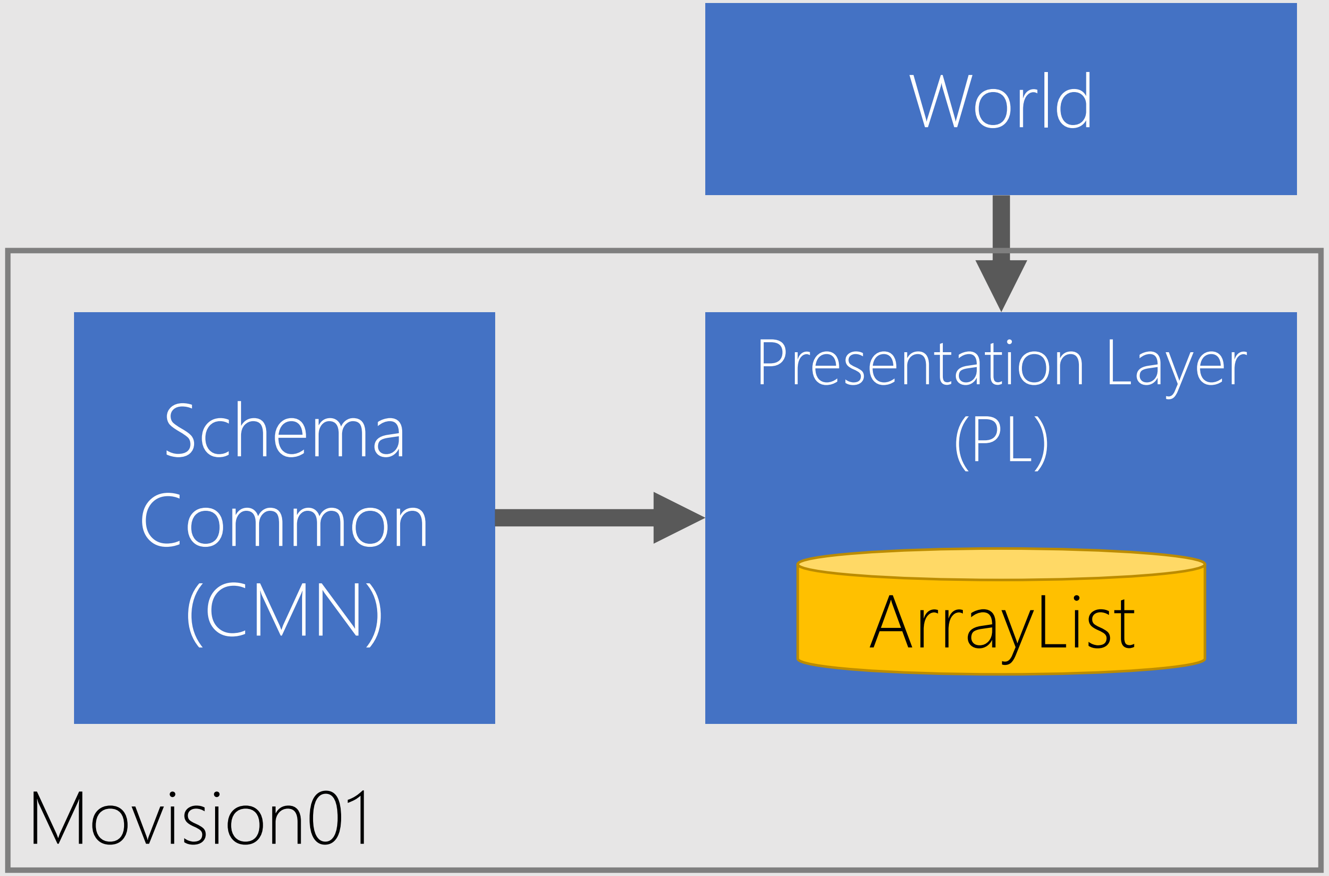
- ~1960: Object Oriented
- 1969: Relational: Mathematical Relation~~ships~~
- 1996: XML

Data Modeling × Logical Level

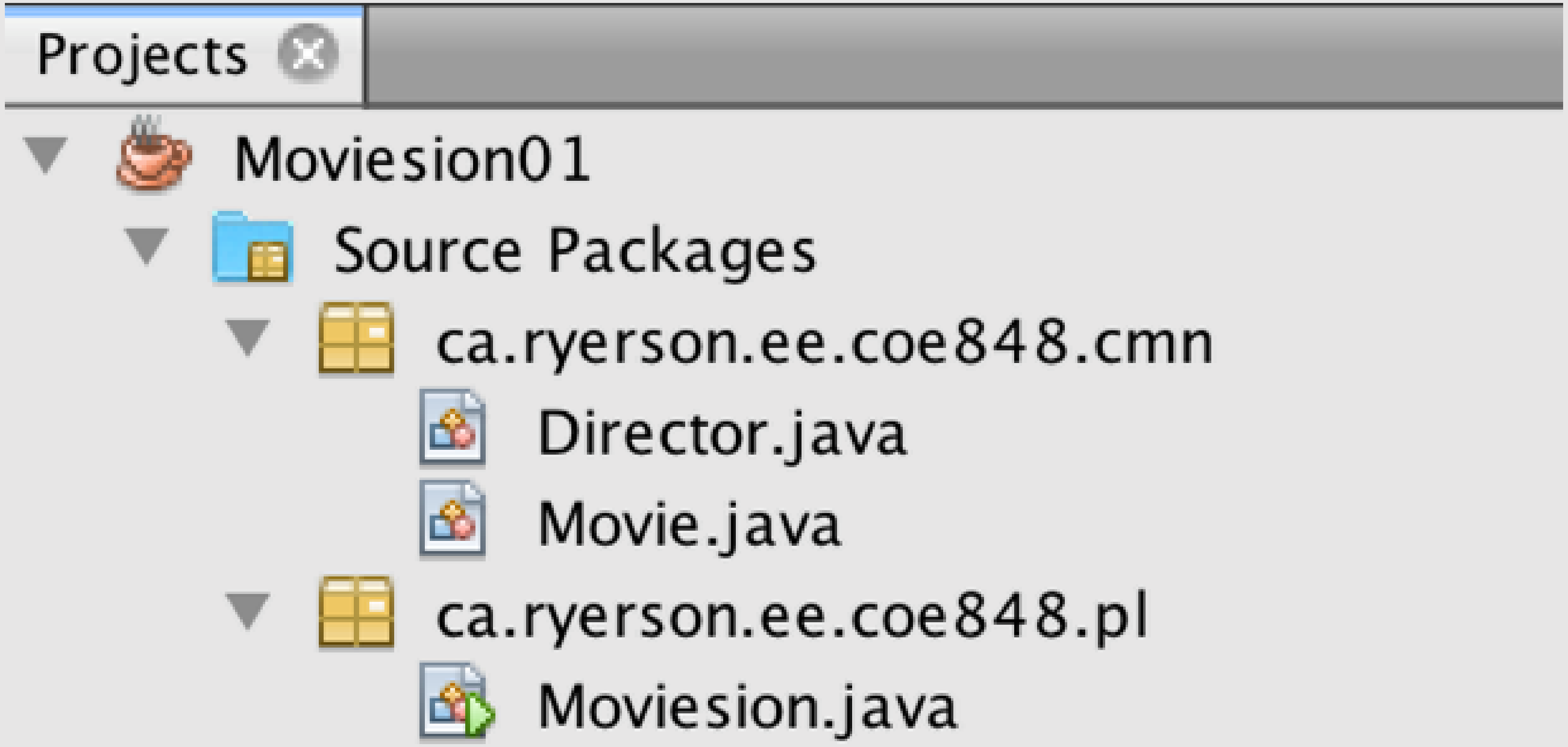
There are other representations as well.
There are other logical models as well.
There are other data models at logical level as well.

~1960: Object Oriented
1969: Relational: Mathematical Relation
1996: XML

Logical Level × Object Oriented



Logical Level × Object Oriented



- Moviesion01
  - Source Packages
    - ca.ryerson.ee.coe848.cmn
      - Director.java
      - Movie.java
    - ca.ryerson.ee.coe848.pl
      - Moviesion.java

Logical Level × Object Oriented

Conceptual Model	Transformation	Logical Model by OO
Data Schema	DDL	*Class*
Data Instance	DML	*Object*

Data Definition Language (DDL): Commands for setting up schema of data
Data Manipulation Language (DML): Commands to manipulate data, also called “query language”

Logical Level × Object Oriented

Conceptual Model

Movie Schema

- Title
- Genre
- Language
- RunningTime
- Poster
- …

Logical Model by OO

```
public class Movie {
    String title;
    String genre;
    String language;
    Decimal runningTime;
    Byte[] poster;
}
```

Logical Level × Object Oriented

Conceptual Model | Logical Model by OO

Movie Instance

Movie Object

Title	Genre	Language	RunningTime	Poster
2001: A Space Odyssey	*Sci-fi*	*English*	*142*	

```
Movie newMovie = new Movie();
newMovie.setName("2001: A Space Odyssey");
newMovie.setGenre("Sci-fi");
newMovie.setLanguage(“English");
newMovie.setRunningTime(142);
newMovie.setPoster(?);
```

Logical Level × Object Oriented

Pros

Time | Random Access Memory (RAM) | Fast

Variety of Data Structures

Array, Linked List, Stack, Queue, Tree, HashTable, …

Easy DML in Memory

SELECT: `E element = ArrayList<E>.get(int index);`

INSERT: `ArrayList<E>.add(int index, E element);`

DELETE: `ArrayList<E>.remove(int index);`

UPDATE: `ArrayList<E>.set(int index, E element);`

Any Computation

Sorting, Searching, …

Cons

Space Far Too Small

Expensive

NOT DURABLE

Volatility | Transient | Short-term Retention

Portability

2019: A Data Odyssey × Real World

Data Modeling: Real World Entity

Conceptual Level |

| Logical Level |

| Physical Level

Computable Entity

Data Modeling × Physical Level

1. Where and how are the data instances stored in physical storage.

Data Modeling × Physical Level

We already have seen one physical storage, haven’t we?

Data Modeling × Physical Level

There are other physical storages as well.
There are other physical models as well.
There are other data models at physical level as well.

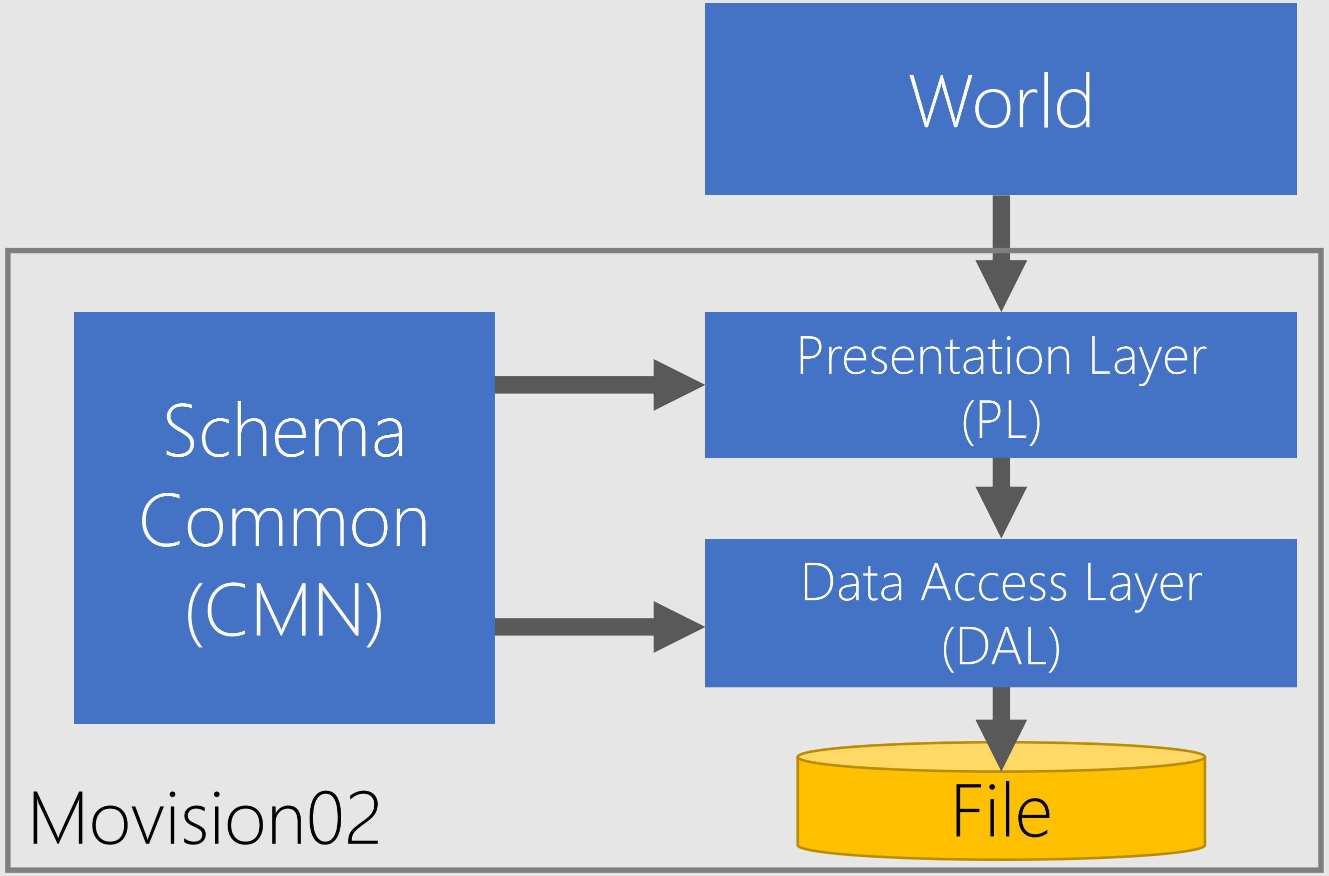
File
Database
Cloud

Data Modeling × Physical Level

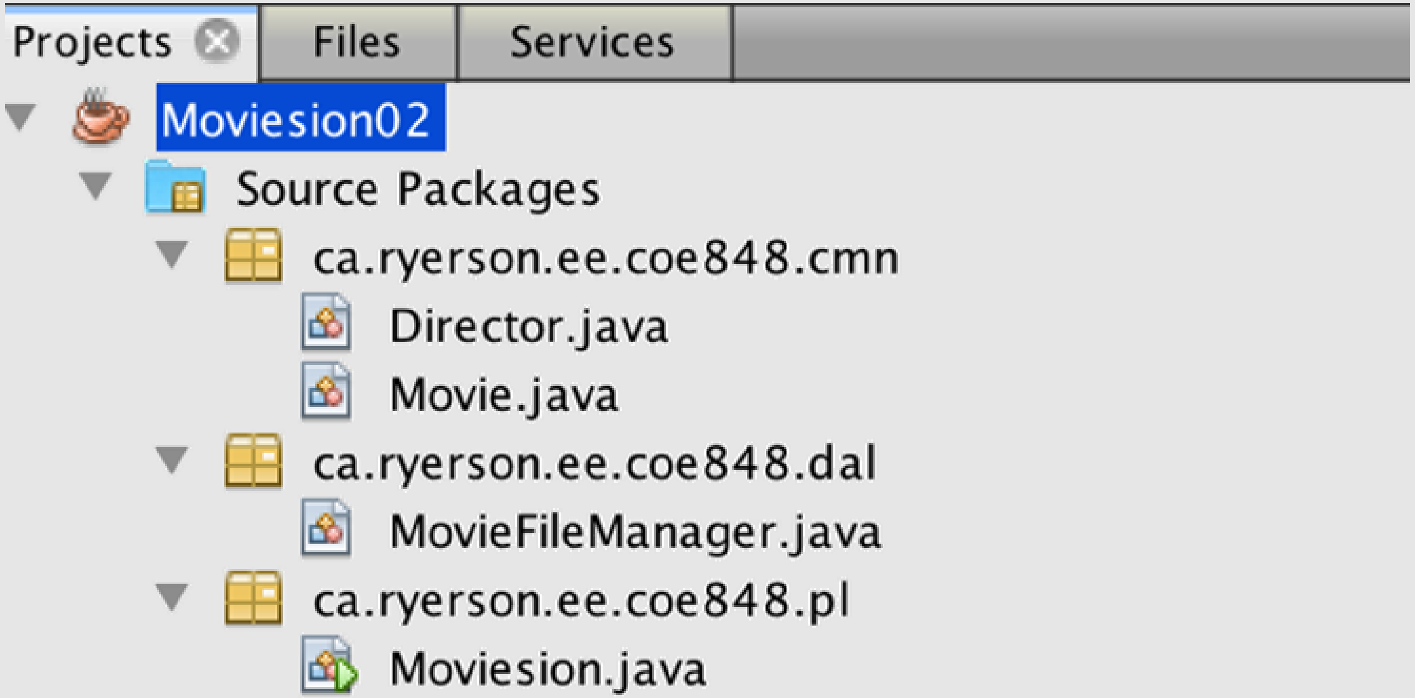
There are other physical storages as well.
There are other physical models as well.
There are other data models at physical level as well.

File
Database
Cloud

Physical Level × File



Physical Level × File



- Moviesion02
  - Source Packages
    - ca.ryerson.ee.coe848.cmn
      - Director.java
      - Movie.java
    - ca.ryerson.ee.coe848.dal
      - MovieFileManager.java
    - ca.ryerson.ee.coe848.pl
      - Moviesion.java

Physical Level × File

Conceptual Model	Logical Model by OO	Physical Model by File
Data Schema	Class	*CSV File: nothing* *JSON File: nothing* *XML File: XSD*
Data Instance	Object	*Record*

2019: A Data Odyssey × Real World

Real World Entity

Conceptual Level | Instance

| Logical Level | Object

| Physical Level | Record

Computable Entity

Physical Level × File

Pros

Space

DURABLE

Persistency | Long-term Retention

Portable

Cons

Time

Hard Disk Drive (HDD)

Sequential Access Method

Electromechanical

Not Easy DML

DELETE | UPDATE

No ACID Properties

Solid State Drive (SSD) | Random Access Memory | Expensive

Physical Level × File × Cons

Not Easy DML

DELETE

- Strategy#1: Create a new file without the deleted record.
- Strategy#2: Shift all following over the deleted record.
- Strategy#3: Logical Deletion | Invalidate Record

UPDATE

- Strategy#1: Replace bytes of old value with bytes of new value
- Strategy#2: Delete Record + Insert Record

Physical Level × File × ACID

Transaction

A group one or more operations (DDL|DML) into a single unit of work.

```
BEGIN TRANSACTION
    INSERT Movie
    INSERT Movie's Director IF NOT EXIST
    INSERT Movie's distributor Company IF NOT EXIST
END TRANSACTION
```

Physical Level × File × ACID

Atomicity

All-or-nothing execution of transaction

```
BEGIN TRANSACTION
    INSERT Movie
    INSERT Movie's Director IF NOT EXIST
    INSERT Movie's distributor Company IF NOT EXIST
END TRANSACTION
Transaction Committed.
3 records affected.
```

Physical Level × File × ACID

Atomicity

All-or-nothing execution of transaction

BEGIN TRANSACTION

INSERT *Movie*

Error!

Partial Execution of Transaction!

ROLLBACK any changes, i.e., Movie record!

Physical Level × File × ACID

Atomicity

- All-or-nothing execution of transaction

- Another example in online banking?

Physical Level × File × ACID

Atomicity

All-or-nothing execution of transaction

```
BEGIN TRANSACTION
    Withdraw    Money from Hossein’s Saving Account
    Deposit     Money to Hossein’s Checking Account
END TRANSACTION
```

Physical Level × File × ACID

Consistency

Respect constraints or expectations among data instances

CONSTRAINT#1:	All Movies Must Have ReleaseDate ≥ 1890
CONSTRAINT#2:	All Movies Must Have at Least One Director
CONSTRAINT#3:	All Movies Must Have UNIQUE Title

Physical Level × File × ACID

Consistency

Respect constraints or expectations among data instances

```
BEGIN TRANSACTION
    INSERT Movie
    INSERT Movie's distributor Company IF NOT EXIST
END TRANSACTION
```

Transaction Conflicts with CONSTRAINT#2.

ROLLBACK any changes.

Physical Level × File × ACID

Consistency

Respect constraints or expectations among data instances

```
BEGIN TRANSACTION
    INSERT Movie
    Error: INSERT Movie's Director IF NOT EXIST
    INSERT Movie's distributor Company IF NOT EXIST
END TRANSACTION
Transaction Committed.
2 records affected.
Partial Execution of Transaction → Transaction Conflicts with CONSTRAINT#2
```

Inconsistency was due to lack of atomicity.
ACID properties are not orthogonal (independent).

Physical Level × File × ACID

Isolation

Transaction appear to be executed as if no other transaction is executing at the same time

Multi-user environment!

Physical Level × File × ACID

Isolation

Transaction appear to be executed as if no other transaction is executing at the same time

Transaction Queue (Sequential Execution, One-at-a-time)		
T1	W1	T1.1. INSERT *2001: A Space Odyssey*
	W2	T1.2. INSERT *Stanley Kubrick* IF NOT EXIST
	W3	T1.3. INSERT *MGM* IF NOT EXIST
T2	W4	T2.1. INSERT *A Clockwork Orange*
	W5	T2.2. INSERT *Stanley Kubrick* IF NOT EXIST
	W6	T2.3. INSERT *Warner Bros* IF NOT EXIST
T3	W7	T3.1. UPDATE Director SET PlaceOfBirth = "United States of America" WHERE PlaceOfBirth = "USA"
T4	W8	…

Physical Level × File × ACID

Isolation

Transaction appear to be executed as if no other transaction is executing at the same time

Transaction Queue (Parallel Execution)			
W1	T1.1. INSERT *2001: A Space Odyssey*	T2.1. INSERT *A Clockwork Orange*	
W2	T1.2. INSERT *Stanley Kubrick* IF NOT EXIST	T1.3. INSERT *MGM* IF NOT EXIST	T2.3. INSERT *Warner Bros* IF NOT EXIST
W3	T2.2. INSERT *Stanley Kubrick* IF NOT EXIST		
W4	T3.1. UPDATE Director SET PlaceOfBirth = "United States of America" WHERE PlaceOfBirth = "USA"		

Physical Level × File × ACID

Isolation

Transaction appear to be executed as if no other transaction is executing at the same time

Transaction Queue (Parallel Execution)		
W1	T1.1. INSERT *2001: A Space Odyssey*	T2.1. INSERT *A Clockwork Orange*
W2	T1.2. INSERT *Stanley Kubrick* IF NOT EXIST	T2.2. INSERT *Stanley Kubrick* IF NOT EXIST
W3	T1.3. INSERT *MGM* IF NOT EXIST	T2.3. INSERT *Warner Bros* IF NOT EXIST
W4	T3.1. UPDATE Director SET PlaceOfBirth = "United States of America" WHERE PlaceOfBirth = "USA"	

What's the Result? Duplicate *Stanley Kubrick*!

Physical Level × File × ACID

Durability

Once a transaction has committed, the effect must never be lost

Function Exception

Program Crash

System Crash: files might not be durable at this level.

"Apocalypse Now"

Starchild: Database Management System

Database Management Systems (DBMS)

A Software System

- Provides General Purpose, Efficient, Convenient, & Safe Multi-user Storage of and Access to Massive Amounts of Persistent Data

ACID Properties Guaranteed!

Database Management Systems (DBMS)

General Purpose	Different Applications: Moviesion, Library,
Safe	From System Failures, From Malicious Users
Convenient	Simple DDL & DML
Efficient	Don’t Search All Files in Order to Get a Record
ACID	Atomicity, Consistency, Isolation, Durability

Database Management Systems (DBMS)

	People (Role)
Is this course about building a DBMS?	No! DBMS Implementors
Is this course about maintaining a DBMS?	No! DB Administers (DBA)
Is this course about using a DBMS?	Yes. DB Designers & User

DBMS Products

Company	DBMS
Oracle	Oracle, MySQL
Microsoft	MySQLServer
IBM	DB2
OpenLink Software	Virtuoso Universal Server
Apache Software Foundation	CouchDB