A Strawman Database System

RegistrarApp

Adds and edits students details

Tracks courses taken by each student

Supports a search feature to look up students

Let’s use a csv file!

```
+++++ students.csv +++++++


name, major, pic, joined, course
”Baby Groot”,”Biology”,”images/groot.jpg”,”Sept 1, 2018”,”Bio-01 The Animal
Kingdom”
”Baby Yoda”,”Philosophy”,”images/yoda.jpg”,”Sept 1, 2019”,”PHIL-01 Meaning of
Life”
”Baby Yoda”,”Philosophy”,”images/yoda.jpg”,”Sept 1, 2019”,”ENG-01 Logic”
”Baby Yoda”,”Philosophy”,”images/yoda.jpg”,”Sept 1, 2019”,”CS-20 Database Systems”
```

You want to get Baby Yoda’s major:

```
For line in students:
      record=parse(line)
      If “Baby Yoda”==record[0]:
           print record[1]
```

But what if ..

put some maps and arrays in your csv?

you want to update a student's major?

the university closed the Phil major and moved all Phil students to CS?

the different threads try to access and update the file at the same time?

you decide to include more student information (e.g. DOB)?

the generators fail and the application crashes?

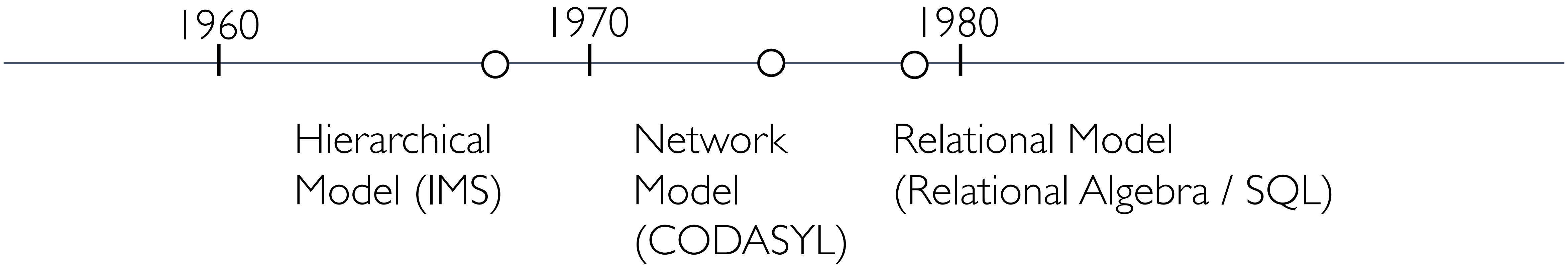
“Once you start down the dark path, forever will it dominate your destiny.
Consume you, it will.”

– Yoda

Data Models

What is a data model?

A collection of tools/concepts for describing data, including its semantics, relationships and constraints.



Hierarchical
Model (IMS)

Network
Model
(CODASYL)

Relational Model
(Relational Algebra / SQL)

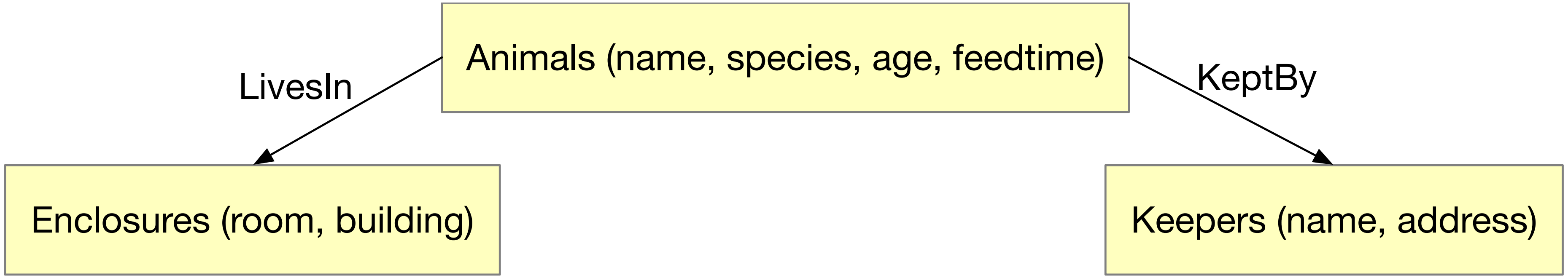
Three Models

Criteria for evaluating a model

1. Data Repetition
2. Physical Data Independence - Protection from changes to the physical structure
3. Logical Data Independence - Protection from changes to the logical structure
4. Support for a high level language

The NYZAD Database

In an alternative universe, a series of events caused NYUAD to be replaced by NYZAD:



Animals (name, species, age, feedtime)

LivesIn

KeptBy

Enclosures (room, building)

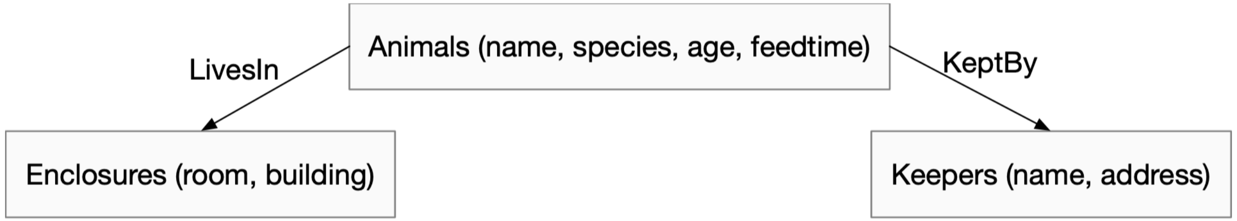
Keepers (name, address)

Happy, monkey, 1yr, 8:30
Gym, CC
Azza, A2-177

Angry, hen, 4yr, 5:30
Library, CC
Miro, A1-1102G

Squeaky, dolphin, 6yr, 10:30
Pool, CC
Batu, UnixLab

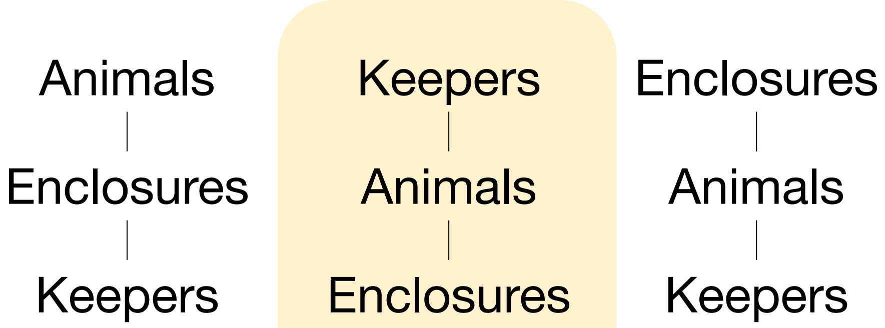
The Hierarchical Model


Animals (name, species, age, feedtime)
LivesIn
KeptBy
Enclosures (room, building)
Keepers (name, address)

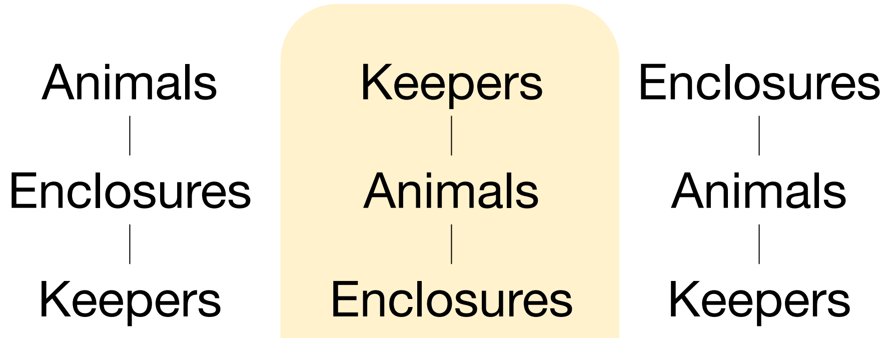
Segment type: or (record type): a collection of named fields with their associated data types.

Segments: collections of records of a type

Tree: Segments are arranged in a tree of segment types



```
Azza (Keepers), A2-177
        Happy, monkey, 1yr, ...
                Gym, CC
        Moma, orangutan, 10yr, ...
                Gym, CC
        Gloomy, sloth, 8yr, ...
                LightCone, ERB
        ...
Miro (Keepers), A1-1102G
        Angry, hen, 4yr, ...
                Library, CC
        ...
```



Animals
Enclosures
Keepers

Keepers
Animals
Enclosures

Enclosures
Animals
Keepers

```
Azza (Keepers), A2-177
        Happy, monkey, 1yr, ...
                Gym, CC
        Moma, orangutan, 10yr, ...
                Gym, CC
        Gloomy, sloth, 8yr, ...
                LightCone, ERB
        ...
Miro (Keepers), A1-1102G
        Angry, hen, 4yr, ...
                Library, CC
        ...
```

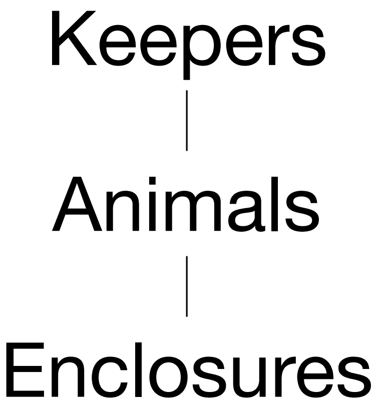
Repeated information on Enclosures.

All representations will repeat information!

The IMS Operators:

```
GetUnique (GU)
GetNext (GN)
GetNext within Parent (GNP)
Insert (INST)
Delete (DEL)
```

Find the enclosure’s that Azza visits?

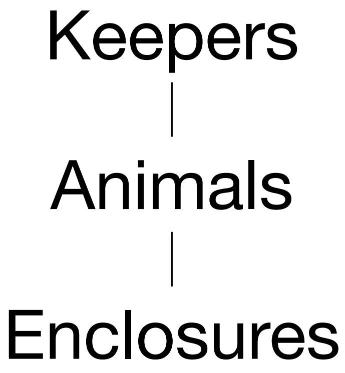

Keepers

Animals

Enclosures

```
GU (Keepers, name = "Azza")
Until done:
    e = GNP (Enclosures)
    print e
```

Physical Independence



What if keepers (roots) are stored:

Sequentially by name?
Hashed by name?
Indexed by key?

```
Azza (Keepers), A2-177
        Happy, monkey, 1yr, ...
                Gym, CC
        Moma, orangutan, 10yr, ...
                Gym, CC
        Gloomy, sloth, 8yr, ...
                LightCone, ERB
        ...
Miro (Keepers), A1-1102G
        Angry, hen, 4yr, ...
                Library, CC
        ...
```

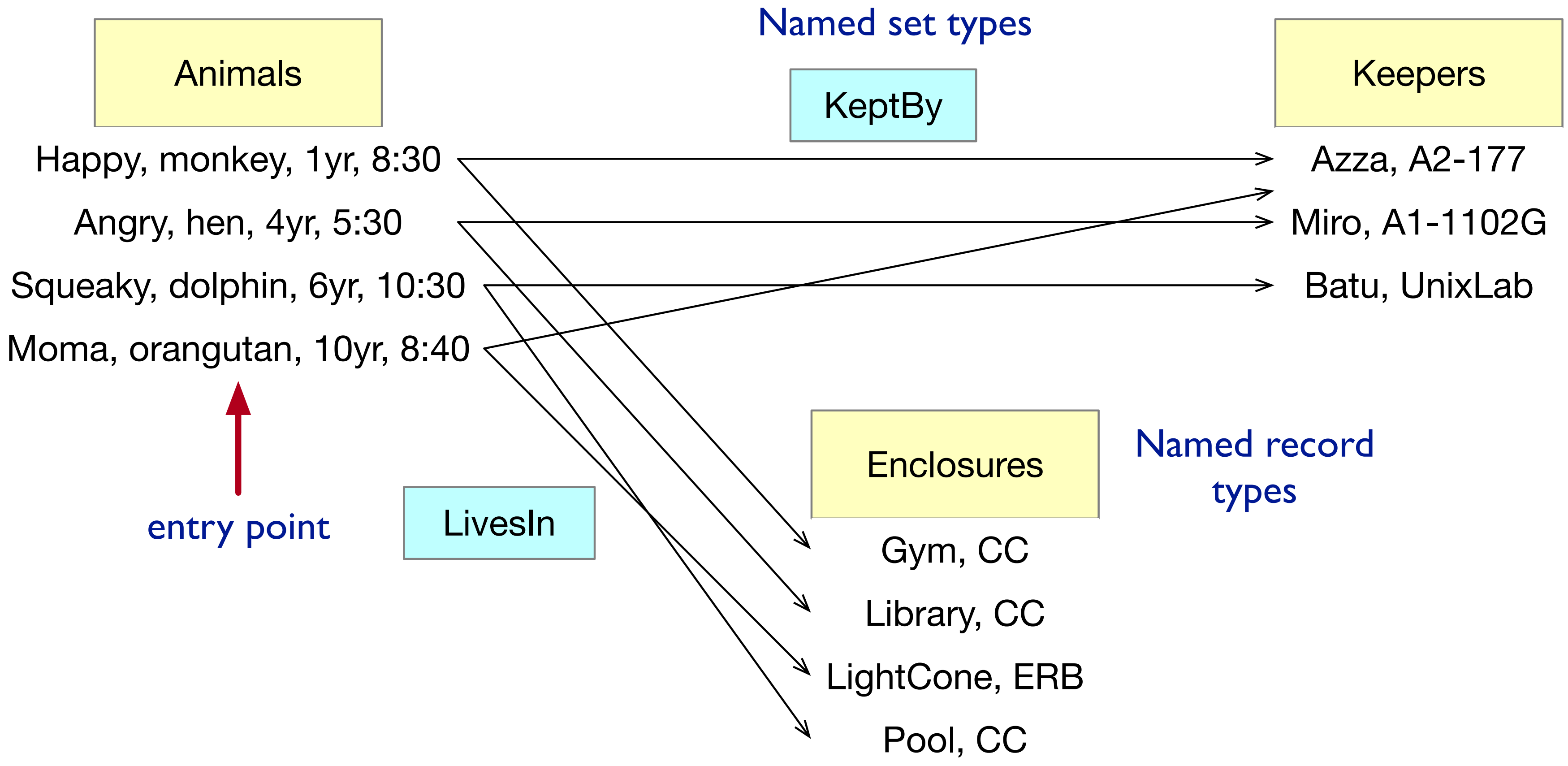
What if dependents are stored:

Physically Sequentially?
Pointers to the Segments?

	Hierarchical Model (IMS)
Data Repetition	☹
Physical Data Independence	☹
Logical Data Independence	😐
Higher Level language	😐

Model Score Card

The Network Model



Named set types

KeptBy

LivesIn

Named record types

Animals

Happy, monkey, 1yr, 8:30

Angry, hen, 4yr, 5:30

Squeaky, dolphin, 6yr, 10:30

Moma, orangutan, 10yr, 8:40

entry point

Keepers

Azza, A2-177

Miro, A1-1102G

Batu, UnixLab

Enclosures

Gym, CC

Library, CC

LightCone, ERB

Pool, CC

Find the enclosure’s that Azza visits?

```
Until done:
       Find next animal a record in Animals
       Find child keeper k record in KeptBy
       Get current record k
       if(k.name == "Azza"):
           print a
```

Model Score Card

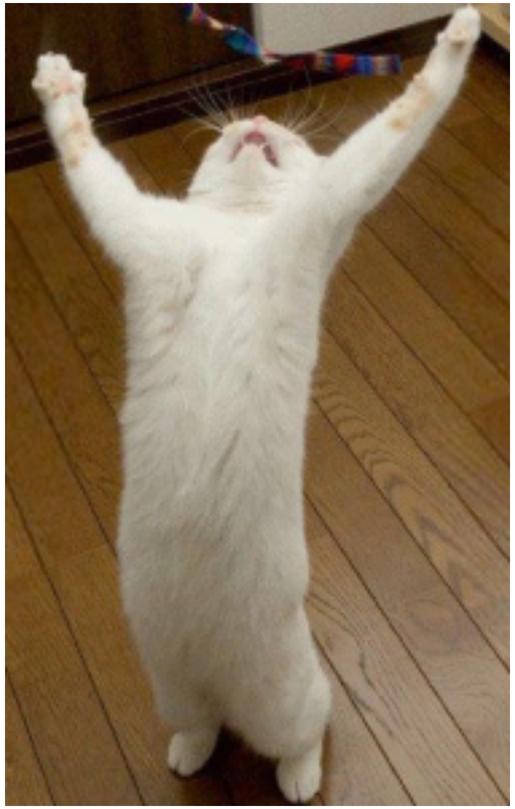
	Hierarchical Model (IMS)	Network Model (CODASYL)
Data Repetition	🙁	🙂
Physical Data Independence	🙁	🙁
Logical Data Independence	😐	🙁
Higher Level language	😐	😐

The Relational Model

Model Score Card

	Hierarchical Model (IMS)	Network Model (CODASYL)
Data Repetition	🙁	🙂
Physical Data Independence	🙁	🙁
Logical Data Independence	😐	🙁
Higher Level language	😐	😐

Model Score Card

	Hierarchical Model (IMS)	Network Model (CODASYL)	Relational Model
Data Repetition	🙁	🙂	
Physical Data Independence	🙁	🙁	
Logical Data Independence	😐	🙁	
Higher Level language	😐	😐	

Ted Codd's Vision

Store the data in simple tables

Access through a high level *set-at-a-time* language (no record-at-a-time operations)

Leave the details of the physical storage open

The Model

Database

A set of *Relations*

Relation

A *schema:* name of the relation, name and type of each column
An *instance*: a table, with rows and columns

Keys

Keys manage relationships between records
A *Primary key* uniquely identifies a record
A *Foreign key* refers to a particular key in another table

Restrictions:

All attributes are *atomic*, *primitive types*, no nested tables
A relation is a *set* of tuples: no tuple can occur more than once

	PK			Column, field, attribute		FK	FK
	aid	**name**	**species**	**age**	**feedtime**	**eid**	**kid**
	325	Happy	monkey	1yr	8:30	72	007
	678	Squeaky	dolphin	6yr	10:30	89	123
Row, Tuple	874	Angry	hen	4yr	5:30	90	555
	921	Moma	orangutan	10yr	8:40	92	007

PK		
eid	**room**	**building**
72	Gym	CC
89	Pool	CC
90	Library	CC
92	LightCone	ERB

PK		
kid	**name**	**address**
007	Azza	A2-177
123	Batu	UnixLab
555	Miro	A1-1102G

Relational Algebra

Algebra

A mathematical system consisting of
Operands: variables or values from which new values are constructed
Operators: symbols denoting procedures that construct new values given existing values

Relational Algebra

Operands are relations
Operators take one or two relation instances as arguments and return one relation instance as result

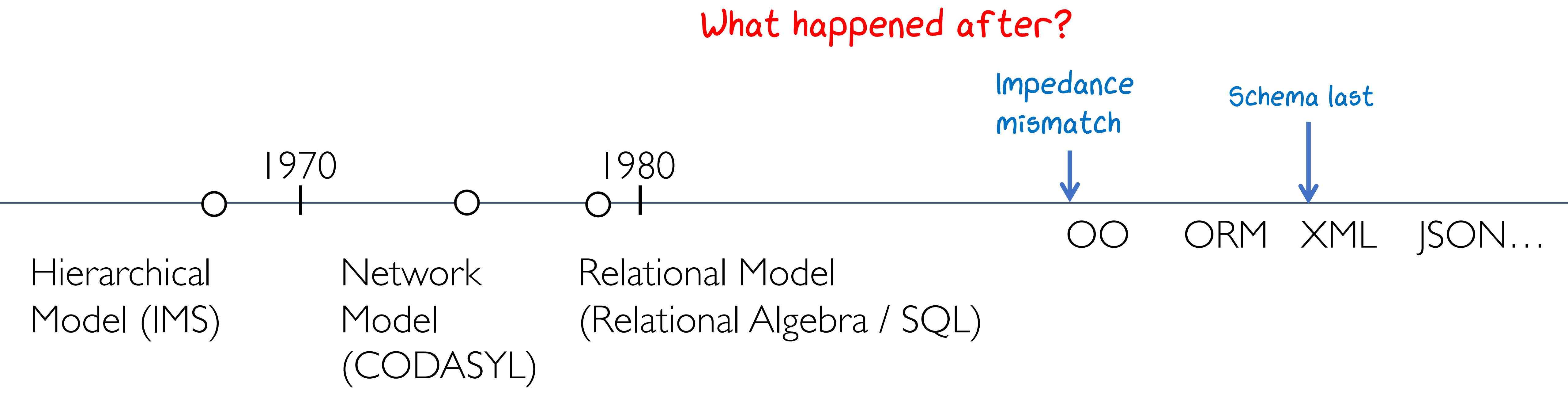
Queries

Relational algebra expression - a composition of relational algebra operators that form a plan of step-by-step procedures to process the data

Model Score Card

	Hierarchical Model (IMS)	Network Model (CODASYL)	Relational Model
Data Repetition	🙁	🙂	🙂
Physical Data Independence	🙁	🙁	🙂
Logical Data Independence	😐	🙁	🙂
Higher Level language	😐	😐	🙂

More Models



What happened after?

1970

1980

Hierarchical Model (IMS)

Network Model (CODASYL)

Relational Model (Relational Algebra / SQL)

Impedance mismatch

Schema last

OO

ORM

XML

JSON…

There's an old saying
about those who forgot
history.
I don't remember it but
it's good.

Stephen Colbert