CSC 430/530 : DATABASE MANAGEMENT SYSTEMS/ DATABASE THEORY

Lecture 1 - continued

16

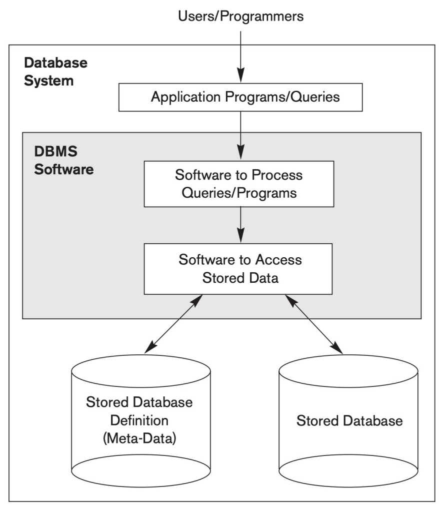
Review

- A database is a collection of related data.
  - Eg. Microsoft Excel, Microsoft Access
- Properties:
  - A database represents some aspects of the real world (aka mini-world).
  - Any changes in the mini-world are reflected in the database.
  - A database is a logically coherent collection of data with some inherent meaning
  - A database is designed, built, and populated with data for a specific purpose.
- A database management system (DBMS) is a collection of programs that enable users to create and maintain a database.

17

What is a DBMS?

- **Definition:** A DBMS is a **general-purpose software system** that facilitates the process of *defining, constructing, manipulating, and searching* databases among various users and applications.



18

Characteristics of a Database Approach

- File processing, is the older approach to storing data.
  - Each user defines and implements the files needed for a specific software application as part of programming the application.
  - Here each application is free to name data elements independently.
- The main characteristics of a database approach versus a file-processing approach are the following:
  - Self-describing nature of a database system --- using **Meta data**.
  - Insulation between programs and data, and data abstraction ---using **data modeling.**
  - Support of multiple views of the data --- using **queries and views**.
  - Sharing of data and multi-user transaction processing --- using **access control and concurrency control.**

19

History Repeats itself

Old database issues are still relevant today.

The **SQL Vs NoSQL** debate is reminiscent of **Relational Vs CODASYL** debate from the 1970s.

May of the ideas in today's database systems are not new

Reference: ADS 2020-Carnegie Mellon University

20

1960s - IDS

- **I**ntegrated **D**ata **S**tore
- Developed internally at GE in the early 1960s.
- GE sold their computing division to Honeywell in 1969.

- One of the first DBMSs:
  - Network data model.
  - Tuple-at-a-time queries.

Reference: ADS 2020-Carnegie Mellon University

21

1960s - **CODASYL**

- COBOL people got together and proposed a standard for how programs will access a database. Lead by Charles Bachman.
  - Network data model
  - Tuple-at-a-time queries.
- Bachman also worked at Culliance Database Systems in the 1970s to help build IDMS.

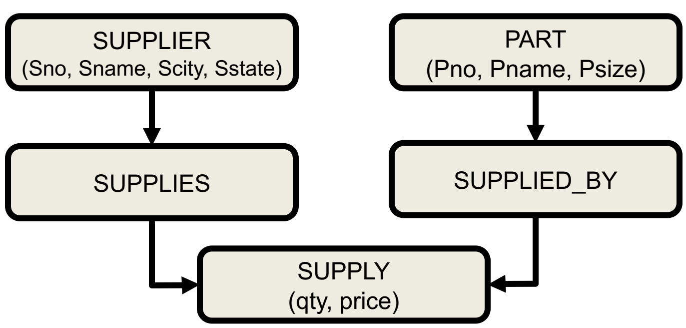
Charles Bachman, 1973 Turing Award recipient

Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

22

Network Data Model

Schema

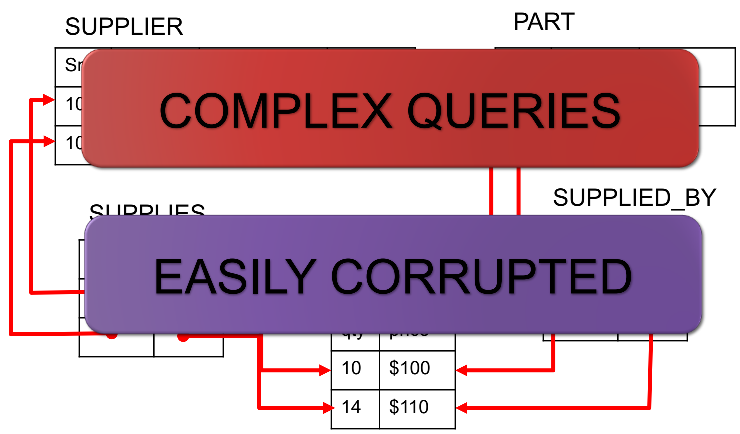


Reference: ADS 2020-Carnegie Mellon University

23

Network Data Model

Instance



Reference: ADS 2020-Carnegie Mellon University

24

1960s – IBM IMS

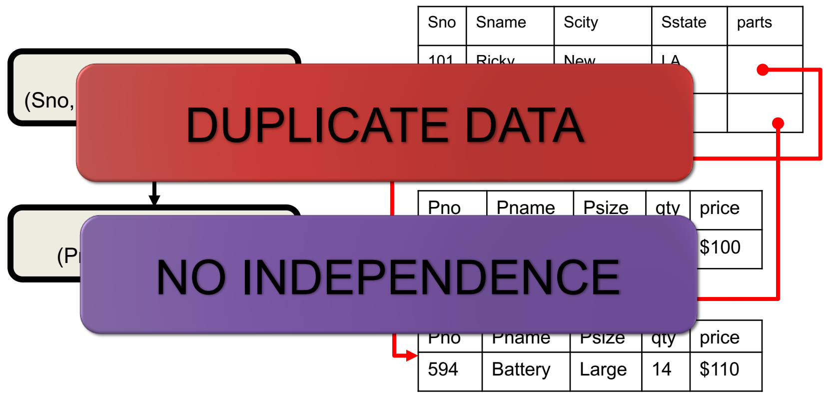
- **I**nformation **M**anagement **S**ystem
- Early database system developed to keep track of purchase orders for **Apollo moon** mission.
  - **Hierarchical data model**
  - Programmer-defined physical storage format
  - Tuple-at-a-time queries

Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

25

Hierarchical Data Model

Schema

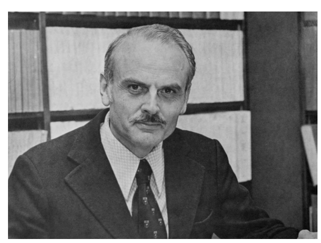


Reference: ADS 2020-Carnegie Mellon University

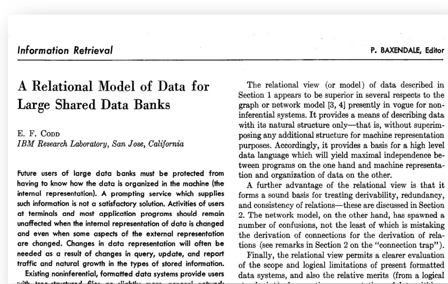
26

1970s – Relational Model

- Edgar "Ted" Codd was a mathematician working at IBM Research.
- He saw developers spending their time rewriting IMS and CODASYL programs every time the database's schema or layout changed.
- **Database abstraction** to avoid this maintenance:
  - Store database in **simple data structures.**
  - Access data **through high-level language.**
  - Physical **storage left up to implementation.**



Edgar Codd, 1981 Turing Award recipient



Information Retrieval P. BAXENDALE, Editor

A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users

The relational view (or model) of data described in Section 1 appears to be superior in several respects to the graph or network model [3, 4] presently in vogue for non-inferential systems. It provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes. Accordingly, it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and machine representation and organization of data on the other.

A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations—these are discussed in Section 2. The network model, on the other hand, has spawned a number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the "connection trap").

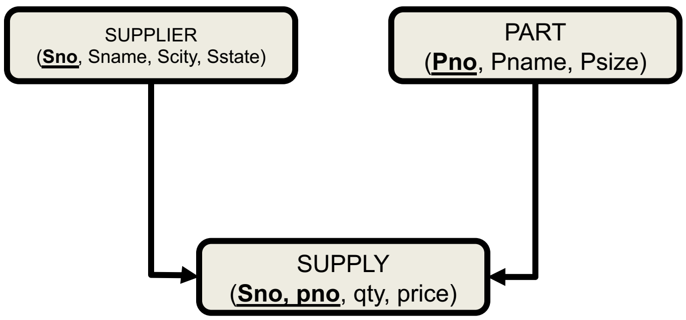
Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical

Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

27

Relational Data Model

Schema



Reference: ADS 2020-Carnegie Mellon University

28

Relational Data Model

Instance

SUPPLIER

Sno	Sname	Scity	Sstate
101	Ricky	New Orleans	LA
102	Deers	Dallas	TX

PART

Pno	Pname	Psize
594	Battery	Large

SUPPLY

Sno	Pno	qty	price
101	594	10	$100
102	594	14	$110

Reference: ADS 2020-Carnegie Mellon University

29

Relational Data Model

Instance

SUPPLIER

Sno	Sname	Scity	Sstate
101	Ricky	New Orleans	LA
102	Deers	Dallas	TX

PART

Pno	Pname	Psize
594	Battery	Large

SUPPLY

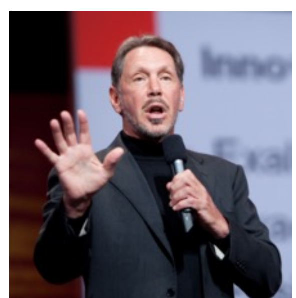
Sno	Pno	qty	price
101	594	10	$100
102	594	14	$110

Reference: ADS 2020-Carnegie Mellon University

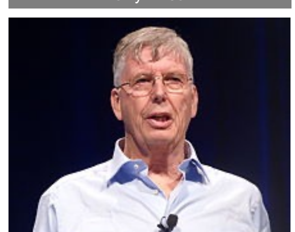
30

1970s – Relational Model

- Early implementations of relational DBMS:
  - System R – IBM Research
  - INGRES – Stonebraker (U.C. Berkeley)
  - Oracle – Larry Ellison



Larry Ellison



Michael Stonebraker received the 2014 Turing Award

Reference: ADS 2020-Carnegie Mellon University

31

1980s – Relational Model

- The Relational model wins.
  - IBM comes out with DB2 in 1983
  - “SEQUEL” becomes the standard (SQL)
- Many new “enterprise” DBMSs but Oracle wins marketplace.
- Stonebraker creates Postgres

Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

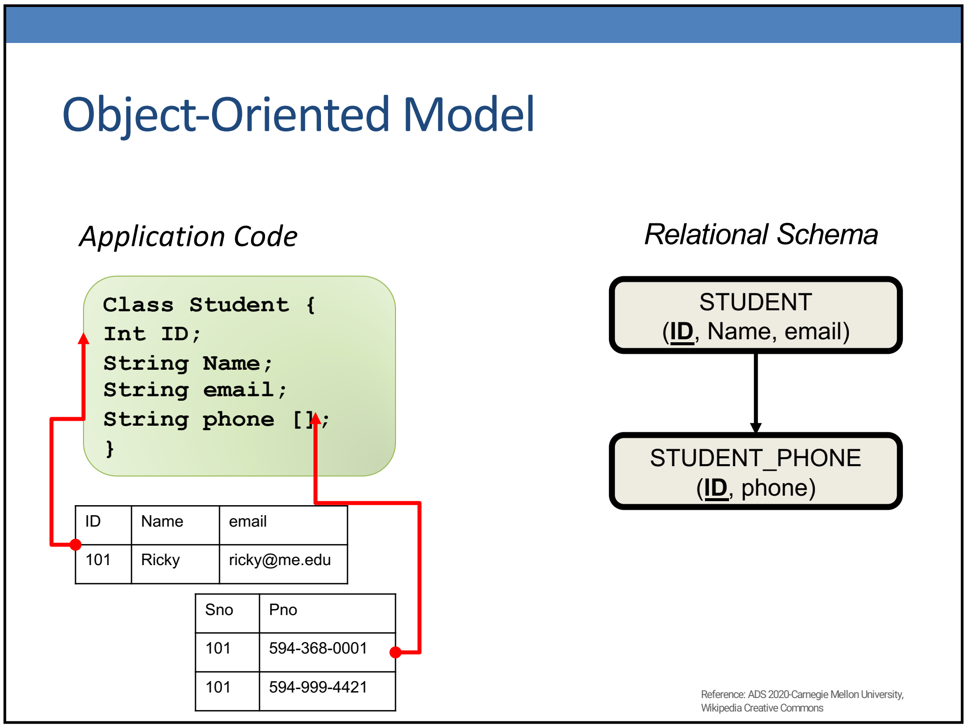
32

1980s – Object-Oriented Databases

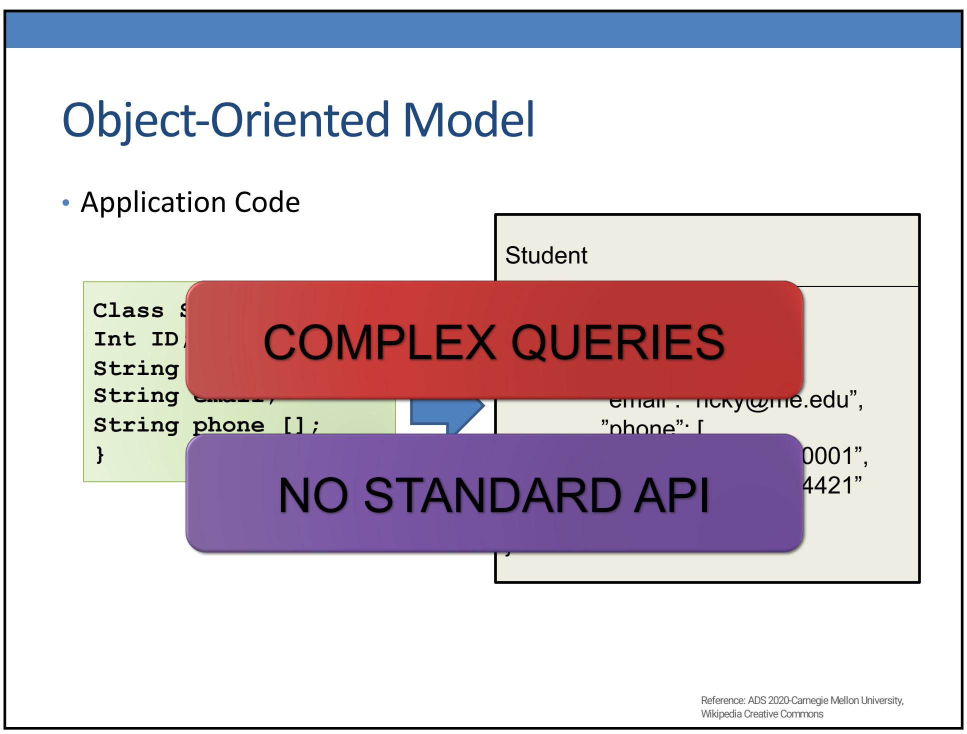
- Avoid **“relational-object impedance mismatch”** by tightly coupling objects and databases.
- Few of these original DBMSs from the 1980s still exist today but many of these technologies exist in other forms (JSON, XML)

Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

33


Object-Oriented Model
Application Code
Class Student {
Int ID;
String Name;
String email;
String phone [];
}
ID
Name
email
101
Ricky
ricky@me.edu
Sno
Pno
101
594-368-0001
101
594-999-4421
Relational Schema
STUDENT
(ID, Name, email)
STUDENT_PHONE
(ID, phone)
Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

34


Object-Oriented Model
Application Code
Student
Class
Int ID
String
String
String phone [];
}
COMPLEX QUERIES
NO STANDARD API
Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

35

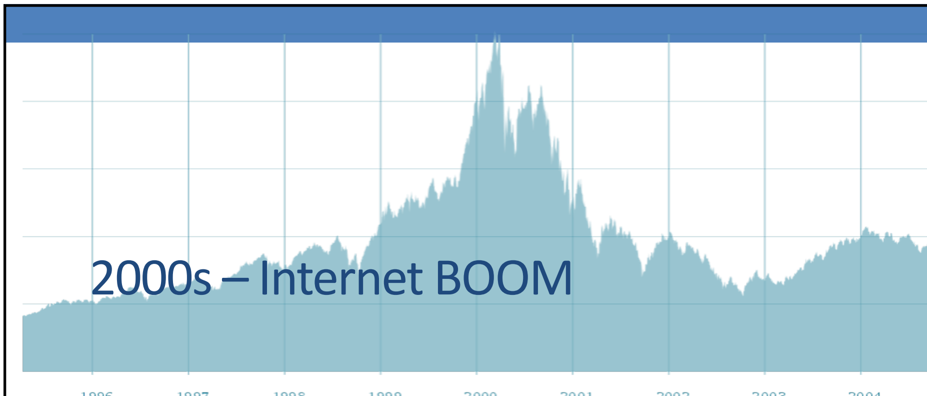
1990s

- No major advancements in database systems or application workloads
  - Microsoft creates SQL Server.
  - MySQL is written as a replacement of mSQL
  - Postgres gets SQL support
  - SQLite started in early 2000s.

Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

36

2000s – Internet BOOM



- All the big players were heavyweight and expensive.
- Open-source databases were missing important features
- Many companies wrote their own custom middleware to scale out database across single-node DBMS instances.

Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

37

2000s – Data Warehouses

RowId	EmpId	Lastname	Firstname	Salary
001	10	Smith	Joe	60000
002	12	Jones	Mary	80000
003	11	Johnson	Cathy	94000
004	22	Jones	Bob	55000

- Rise of the special purpose OLAP DBMSs
  - Distributed / Shared – Nothing
  - Relational / SQL
  - Usually closed-source.

```
001:10,Smith,Joe,60000;
002:12,Jones,Mary,80000;
003:11,Johnson,Cathy,94000;
004:22,Jones,Bob,55000;
```

- Significant performance benefits from using columnar data storage model

```
10:001,12:002,11:003,22:004;
Smith:001,Jones:002,Johnson:003,Jones:004;
Joe:001,Mary:002,Cathy:003,Bob:004;
60000:001,80000:002,94000:003,55000:004;
```

Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

38

2000s – NoSQL Systems

- Focus on high-availability & high scalability:
  - Schemaless
  - Non-relational data models
  - No ACID transactions
  - Custom APIs instead of SQL
  - Usually open-source.

mongoDB

redis

cassandra

APACHE HBASE

CouchDB

RethinkDB

neo4j

Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

39

2010s - NewSQL

- Provide same performance for OLTP workloads as NoSQL DBMSs without giving up ACID:
  - Relational / SQL
  - Distributed
  - Usually closed-source

SAP HANA · TiDB · Google Cloud Spanner · VOLTDB · NUODB · Cockroach DB · MEMSQL

Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

40

2010s – Cloud Systems

- First database-as-a-service (DBaaS) offerings were "containerized" versions of existing DBMSs.

- There are new DBMSs that are designed from scratch explicitly for running in the cloud environment.

SlicingDice · Amazon Aurora · Google Cloud Spanner · amazon REDSHIFT · xeround The Cloud Database · amazon DynamoDB · FAUNA · snowflake

Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

41

2010s – Shared-Disk Engines

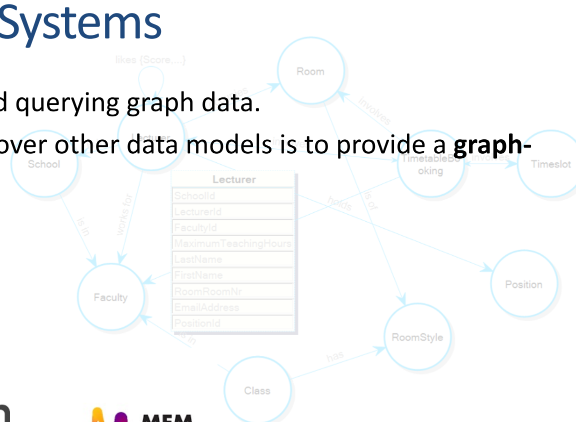
- Instead of writing a custom storage manager, the DBMS leverages **distributed storage**
  - Scale execution layer independently of storage
  - Favors log-structured approaches
- This is what most people think of when they talk about a **data lake.**

Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

42

2010s- Graph Systems

- Systems for storing and querying graph data.
- Their main advantage over other data models is to provide a **graph-centric query API**



Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

43

2010s – Timeseries Systems

- Specialized systems that are designed to store timeseries / event data.
- The design of these systems make deep assumptions about the distribution of data and workload query patterns.

influxdb

Timescale

M3

ClickHouse

VICTORIA METRICS

Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

44

Future of Database Management

- **Newer DBMSs**
  - Embedded DBMSs
  - Multi-Model DBMSs
  - Blockchain DBMSs
- **Harness the potential of AI and Machine Learning**
  - Machine learning will power a diverse array of data management capabilities, including data cataloging, metadata management, data mappings, anomaly detection, etc.
  - AI will enable recommended actions, auto-discovery of metadata, and auto-monitoring of governance controls.
- **DataOps**
  - DataOps combines agile development, technologies, processes, and practices such as statistical process control to deliver data and analytics.

Reference: ADS 2020-Carnegie Mellon University, Wikipedia Creative Commons

45

Why Study Databases?

Academic
Databases involve many aspects of computer science.

Active area of research

Three Turing awards in databases

Programmer
A wide array of applications involve using or accessing databases.

Business
Every organization needs databases

Student
Easier to get hired!!!!!

46

What is the goal of a DBMS?

Electronic record-keeping

Fast and *convenient* ***access*** to information

DBMS = Database Management System

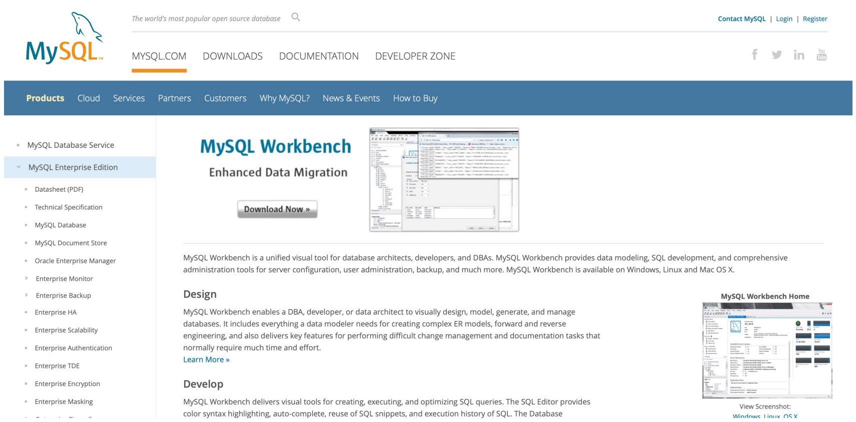
"**Relational**" in this course

Data + set of instructions to access / manipulate data

47

MySQL Workbench

- Install both MySQL workbench and Server (https://www.mysql.com/products/workbench/)



48

Questions?

49