Data Models and Database Management Systems (DBMSs)
Data Models in the 1960s, 1970s, and 1980s

Hierarchical

Network (Graph)

Relational – Schema (Model) - first

1956 Vern Watts was IMS's chief architect for many years. Watts joined IBM in 1956 and worked at IBM's Silicon Valley development labs until his death on April 4, 2009.

1966 IBM began the design of IMS with Rockwell and Caterpillar for the Apollo program, where it was used to inventory the very large bill of materials (BOM) for the Saturn V moon rocket and Apollo space vehicle.

1968 The first "IMS READY" message appeared on an IBM 2740 terminal in Downey, California, on 14 August 1968.

1965 CODASYL (Conference / Committee on Data Systems Languages) formed a List Processing Task Force.

1969 the network CODASYL database model was published.

1980s interest in CODASYL gradually faded due to growing interest in relational databases.

Year Event
1970 Dr. E. F. Codd develops the relational database model.
1978 IBM develops the predecessor to SQL.
1979 Relational Software, Inc. (later renamed Oracle) releases the first relational DBMS, Oracle.
1982 IBM releases their first RDBMS, SQL/DS (SQL/Data System).
1985 IBM released DB2 (Database 2).
1987 Microsoft releases SQL Server.
1989 ANSI publishes first SQL standards (ANSI/ISO SQL-89, or SQL1).
Dr. E. F. Codd

1. Humans only know how to store data on Disk or in Memory in the form of Mathematical Relations (Tables).
2. So, the Data Model should be Relational (i.e., Tables).
3. Also, DBMSs should have an ad hoc query language.

A mathematical Relation is a subset of the cross-product of a set of Domains. e.g.,

| <  | 0   | 1   | 0   | 2   | 0   | 3   | ...
|----|-----|-----|-----|-----|-----|-----|-----
|    | 1   |     | 2   |     | 3   |     |...

<table>
<thead>
<tr>
<th>D1: People</th>
<th>D2: Ages</th>
<th>D3: Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phil</td>
<td>60</td>
<td>M</td>
</tr>
<tr>
<td>Rita</td>
<td>65</td>
<td>F</td>
</tr>
</tbody>
</table>

Example 2

<table>
<thead>
<tr>
<th>People</th>
<th>Ages</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phil</td>
<td>60</td>
<td>M</td>
</tr>
<tr>
<td>Phil</td>
<td>60</td>
<td>F</td>
</tr>
<tr>
<td>Phil</td>
<td>65</td>
<td>M</td>
</tr>
<tr>
<td>Phil</td>
<td>65</td>
<td>F</td>
</tr>
<tr>
<td>Rita</td>
<td>60</td>
<td>M</td>
</tr>
<tr>
<td>Rita</td>
<td>60</td>
<td>F</td>
</tr>
<tr>
<td>Rita</td>
<td>65</td>
<td>M</td>
</tr>
<tr>
<td>Rita</td>
<td>65</td>
<td>F</td>
</tr>
</tbody>
</table>
Dr. Philip Cannata

## Traditional Data Management in the 1980s, 1990s, and 2000s

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1956</td>
<td>Vern Watts was IMS's chief architect for many years. Watts joined IBM in 1956 and worked at IBM's San Valley development labs until his death on April 4, 2009.</td>
</tr>
<tr>
<td>1966</td>
<td>IBM began the design of IMS with Rockwell and Caterpillar for the Apollo program. It was used to inventory the very large bill of materials (BOM) for the Saturn V rocket and Apollo space vehicle.</td>
</tr>
<tr>
<td>1968</td>
<td>The first &quot;IMS READY&quot; message appeared on an IBM 2740 terminal in Downey, California, on 14 August 1968.</td>
</tr>
<tr>
<td>1965</td>
<td>CODASYL (Conference Committee on Data Systems Languages) formed the Processing Task Force.</td>
</tr>
<tr>
<td>1969</td>
<td>CODASYL's network CODASYL database model was published.</td>
</tr>
<tr>
<td>1980s</td>
<td>Interest in CODASYL gradually faded due to growing interest in relational databases.</td>
</tr>
<tr>
<td>1970</td>
<td>Dr. E. F. Codd develops the relational database model.</td>
</tr>
<tr>
<td>1978</td>
<td>IBM develops the predecessor to SQL.</td>
</tr>
<tr>
<td>1979</td>
<td>Relational Software, Inc. (later renamed Oracle) releases the first relational DBMS, Oracle.</td>
</tr>
<tr>
<td>1982</td>
<td>IBM releases their first RDBMS, SQL/DS (SQL/Data System).</td>
</tr>
<tr>
<td>1985</td>
<td>IBM released DB2 (Database 2).</td>
</tr>
<tr>
<td>1987</td>
<td>Microsoft releases SQL Server.</td>
</tr>
<tr>
<td>1989</td>
<td>ANSI publishes first SQL standards (ANSI/ISO SQL-89, or SQL1).</td>
</tr>
</tbody>
</table>

### Hierarchical

- Sales Representative
  - Customer Contact #1
  - Customer Contact #2

### Network (Graph)

- Sales Representative
  - Market Research
    - Name, Address, Telephone
    - Social Security Info.
  - Human Resources
    - Job Function
      - Manager, Group
      - Human Resources

### Relational – Schema (Model) - first

- First Name
- Last Name
- Social Security No.
- Date of Birth
- Address
- Social Security No.
- 1980s interest in CODASYL gradually faded due to growing interest in relational databases.
cs347 Data Management

Relational (YeSQL) – Schema-first

The Complexity goes Somewhere!

NoSQL – Schema(Model)-later

Big Data – Schema(Model)-never

Dr. Philip Cannata
Relational, Graph, and Property Graph Data Models

Relational Model

<table>
<thead>
<tr>
<th>Subject</th>
<th>Predicate</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>emp class</td>
<td></td>
<td>RowGUID1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>empno 7369</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hiredate 17-DEC-80</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ename SMITH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>job CLERK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mgr 7902</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hiredate 17-DEC-80</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sal 800</td>
</tr>
<tr>
<td></td>
<td></td>
<td>deptno 20</td>
</tr>
</tbody>
</table>

Graph Model

Property Graph Model

<table>
<thead>
<tr>
<th>Node</th>
<th>Relationship</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>emp node 1</td>
<td>department</td>
<td>dept node 1</td>
</tr>
<tr>
<td>empno = 7369</td>
<td></td>
<td>deptno= 7369</td>
</tr>
<tr>
<td>ename = SMITH</td>
<td></td>
<td>dname= SMITH</td>
</tr>
<tr>
<td>job = CLERK</td>
<td></td>
<td>loc= CLERK</td>
</tr>
<tr>
<td>mgr = 7902</td>
<td></td>
<td>employee</td>
</tr>
<tr>
<td>hiredate = 17-DEC-80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sal = 800</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Relational, Graph, and Property Graph Data Models

**Relational Model**

```
CREATE TABLE EMP
(EMPNO NUMBER(7) NOT NULL,
CONSTRAINT emp_pkey PRIMARY KEY (empno),
ENAME VARCHAR2(10),
JOB VARCHAR2(9),
MGR NUMBER(4),
HIREDATE DATE,
SAL NUMBER(7, 2),
COMM NUMBER(7, 2),
DEPTNO NUMBER(2));
```

```
CREATE TABLE DEPT
(DEPTNO NUMBER(2),
DNAME VARCHAR2(14),
LOC VARCHAR2(13));
```

**Graph Model**

Not needed but allowed and standardized by W3C. We’ll see this later with rdf, rdfs, and owl.

**Property Graph Model**

Not allowed.
Dr. Philip Cannata

Relational, Graph, and Property Graph Data Models

Carnot Technology Stack (January 2016)

- Tableau
- R
- Oracle Apex
- Oracle applications
- Other Rest Clients

Carnot RE (Rest Engine)

Carnot KE (Knowledge Engine)

SQL

Neo4j
Gremlin
Others

SPARQL

Other Languages

Oracle RDF Graph for NoSQL

Oracle Graph

Other Clients (e.g., StarDog, Open Cyc)

Oracle NoSQL

Oracle RDBS EE

Other Servers (e.g., StarDog, Cyc Server)

Carnot DE (Data Engine)

SIM

Oracle NoSQL and Titan

Shared Parser

Dr. Philip Cannata
Relational, Graph, and Property Graph Data Models

Relational Model

SQL:
- SQL on conn "INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO) VALUES (7369, 'SMITH', 'CLERK', 7902, TO_DATE('17-DEC-1980', 'DD-MON-YYYY'), 800, NULL, 20) ;"
- SQL on conn "INSERT INTO DEPT (DEPTNO, DNAME, LOC) VALUES (20, 'RESEARCH', 'DALLAS') ;"

RDF Java API:
In addQuad, stmt is: :dept, :7580ebed-e755-4de1-a490-f37c32542f0b, :deptno, :20
In addQuad, stmt is: :dept, :7580ebed-e755-4de1-a490-f37c32542f0b, :dname, :RESEARCH
In addQuad, stmt is: :dept, :7580ebed-e755-4de1-a490-f37c32542f0b, :loc, :DALLAS
In addQuad, stmt is: :dept, :7580ebed-e755-4de1-a490-f37c32542f0b, :employees,
    :dfb8b0a9-4dca-4ecc-922e-d4b515043010

Graph Model

Property Graph Model

Neo4j OpenCypher:
- Neo4j on conn "CREATE (:dept { DEPTNO : 20, DNAME : 'RESEARCH', LOC : 'DALLAS' });"
- Neo4j on conn "MATCH (a:emp),(b:dept) WHERE a.deptno = 20 AND b.deptno = 20 CREATE (a)<-[employees]->(b) ;"
- Neo4j on conn "MATCH (a:emp),(b:dept) WHERE a.deptno = 20 AND b.deptno = 20 CREATE (a)-[dept]->(b) ;"
Relational, Graph, and Property Graph Data Models

**Relational Model**

SQL:
- SQL on conn "select * from emp";
- SQL on conn "select * from dept"
- SQL on conn "select e.ename, d.dname from emp e join dept d on(e.deptno = d.deptno)"

**Graph Model**

**Property Graph Model**

Neo4j OpenCypher:
- Neo4j on conn "MATCH(a:emp) RETURN a.ename; "
- Neo4j on conn "MATCH(a:dept) RETURN a.dname; "
- Neo4j on conn "MATCH(a:emp)<-[employees]-(b:dept) RETURN a.ename, b.dname; "

sal = 800
deptno = 20
Relational, Graph, and Property Graph Data Models

### Relational Model

<table>
<thead>
<tr>
<th>EMPNO</th>
<th>ENAME</th>
<th>JOB</th>
<th>MGR</th>
<th>HIREDATE</th>
<th>SAL</th>
<th>COMM</th>
<th>DEPTNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>7369</td>
<td>SMITH</td>
<td>CLERK</td>
<td>7902</td>
<td>17-DEC-80</td>
<td>800</td>
<td>null</td>
<td>20</td>
</tr>
</tbody>
</table>

**SPARQL:**

```python
>>> results = SPARQL on conn "select ?g ?x ?y ?z where { GRAPH ?g { ?x ?y ?z } }
>>> for r in sorted(results):
    print r
```

See the results on the next 3 pages.

### Graph Model

**Node:**
- emp node 1
  - empno = 7369
  - ename = SMITH
  - job = CLERK
  - mgr = 7902
  - hiredate = 17-DEC-80
  - sal = 800
  - deptno = 20

**Relationship:**
- department

**Node:**
- dept node 1
  - deptno = 7369
  - dname = SMITH
  - loc = CLERK

### Property Graph Model

**Node:**
- RESARCH
  - RowGUID1

**Node:**
- CLERK
  - RowGUID2

**Node:**
- dept class

**Node:**
- RowGUID1

**Relationship:**
mgr

**Relationship:**
emp
Relational, Graph, and Property Graph Data Models

('SCHEMA', 'dept', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type', 'http://www.w3.org/2000/01/rdf-schema#Class')
('SCHEMA', 'emp', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type', 'http://www.w3.org/2000/01/rdf-schema#Class')

('dept', '94f693ee-c71d-4caa-8f76-0d51152635af', 'deptno', 20)
('dept', '94f693ee-c71d-4caa-8f76-0d51152635af', 'dname', 'RESEARCH')
('dept', '94f693ee-c71d-4caa-8f76-0d51152635af', 'employees', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89')
('dept', '94f693ee-c71d-4caa-8f76-0d51152635af', 'loc', 'DALLAS')

('dept_SCHEMA', '94f693ee-c71d-4caa-8f76-0d51152635af', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type', 'dept')
('dept_SCHEMA', 'deptno', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type', 'http://www.w3.org/2002/07/owl#DatatypeProperty')
('dept_SCHEMA', 'deptno', 'http://www.w3.org/2000/01/rdf-schema#domain', 'dept')
('dept_SCHEMA', 'deptno', 'http://www.w3.org/2000/01/rdf-schema#range', 'http://www.w3.org/2001/XMLSchema#string')
('dept_SCHEMA', 'dname', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type', 'http://www.w3.org/2002/07/owl#DatatypeProperty')
('dept_SCHEMA', 'dname', 'http://www.w3.org/2000/01/rdf-schema#domain', 'dept')
('dept_SCHEMA', 'dname', 'http://www.w3.org/2000/01/rdf-schema#range', 'http://www.w3.org/2001/XMLSchema#string')
('dept_SCHEMA', 'loc', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type', 'http://www.w3.org/2002/07/owl#DatatypeProperty')
('dept_SCHEMA', 'loc', 'http://www.w3.org/2000/01/rdf-schema#domain', 'dept')
('dept_SCHEMA', 'loc', 'http://www.w3.org/2000/01/rdf-schema#range', 'http://www.w3.org/2001/XMLSchema#string')
Relational, Graph, and Property Graph Data Models

Relational Model

<table>
<thead>
<tr>
<th>EMPNO</th>
<th>ENAME</th>
<th>JOB</th>
<th>MGR</th>
<th>HIREDATE</th>
<th>SAL</th>
<th>COMM</th>
<th>DEPTNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>7369</td>
<td>SMITH</td>
<td>CLERK</td>
<td>7902</td>
<td>17-DEC-80</td>
<td>800</td>
<td>(null)</td>
<td>20</td>
</tr>
</tbody>
</table>

('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'comm', 0)
('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'dept', '94f693ee-c71d-4caa-8f76-0d51152635af')
('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'deptno', 20)
('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'empno', 7369)
('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'ename', 'SMITH')
('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'hiredate', '17-DEC-80')
('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'job', 'CLERK')
('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'mgr', 7902)
('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'sal', 800).org/2001/XMLSchema#string')

('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'comm', 0)
('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'dept', '94f693ee-c71d-4caa-8f76-0d51152635af')
('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'deptno', 20)
('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'empno', 7369)
('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'ename', 'SMITH')
('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'hiredate', '17-DEC-80')
('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'job', 'CLERK')
('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'mgr', 7902)
('emp', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'sal', 800).org/2001/XMLSchema#string')

dname = SMITH
dloc = CLERK

employee

date = 17-DEC-80
sal = 800
depnto = 20
Relational, Graph, and Property Graph Data Models

SCHEMA

('emp_SCHEMA', 'a9e1615f-c8df-478d-8c48-ea4b2ccd1a89', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type', 'emp')
('emp_SCHEMA', 'comm', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type', 'http://www.w3.org/2002/07/owl#DatatypeProperty')
('emp_SCHEMA', 'comm', 'http://www.w3.org/2000/01/rdf-schema#domain', 'emp')
('emp_SCHEMA', 'comm', 'http://www.w3.org/2000/01/rdf-schema#range', 'http://www.w3.org/2001/XMLSchema#string')
('emp_SCHEMA', 'deptno', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type', 'http://www.w3.org/2002/07/ owl#DatatypeProperty')
('emp_SCHEMA', 'deptno', 'http://www.w3.org/2000/01/rdf-schema#domain', 'emp')
('emp_SCHEMA', 'deptno', 'http://www.w3.org/2000/01/rdf-schema#range', 'http://www.w3.org/2001/XMLSchema#string')
('emp_SCHEMA', 'empno', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type', 'http://www.w3.org/2002/07/ owl#DatatypeProperty')
('emp_SCHEMA', 'empno', 'http://www.w3.org/2000/01/rdf-schema#domain', 'emp')
('emp_SCHEMA', 'empno', 'http://www.w3.org/2000/01/rdf-schema#range', 'http://www.w3.org/2001/XMLSchema#string')
('emp_SCHEMA', 'ename', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type', 'http://www.w3.org/2002/07/ owl#DatatypeProperty')
('emp_SCHEMA', 'ename', 'http://www.w3.org/2000/01/rdf-schema#domain', 'emp')
('emp_SCHEMA', 'ename', 'http://www.w3.org/2000/01/rdf-schema#range', 'http://www.w3.org/2001/XMLSchema#string')
('emp_SCHEMA', 'hiredate', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type', 'http://www.w3.org/2002/07/ owl#DatatypeProperty')
('emp_SCHEMA', 'hiredate', 'http://www.w3.org/2000/01/rdf-schema#domain', 'emp')
('emp_SCHEMA', 'hiredate', 'http://www.w3.org/2000/01/rdf-schema#range', 'http://www.w3.org/2001/XMLSchema#string')
('emp_SCHEMA', 'job', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type', 'http://www.w3.org/2002/07/owl#DatatypeProperty')
('emp_SCHEMA', 'job', 'http://www.w3.org/2000/01/rdf-schema#domain', 'emp')
('emp_SCHEMA', 'job', 'http://www.w3.org/2000/01/rdf-schema#range', 'http://www.w3.org/2001/XMLSchema#string')
('emp_SCHEMA', 'mgr', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type', 'http://www.w3.org/2002/07/ owl#DatatypeProperty')
('emp_SCHEMA', 'mgr', 'http://www.w3.org/2000/01/rdf-schema#domain', 'emp')
('emp_SCHEMA', 'mgr', 'http://www.w3.org/2000/01/rdf-schema#range', 'http://www.w3.org/2001/XMLSchema#string')
('emp_SCHEMA', 'sal', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type', 'http://www.w3.org/2002/07/owl#DatatypeProperty')
('emp_SCHEMA', 'sal', 'http://www.w3.org/2000/01/rdf-schema#domain', 'emp')
('emp_SCHEMA', 'sal', 'http://www.w3.org/2000/01/rdf-schema#range', 'http://www.w3.org/2001/XMLSchema#string')
class/subclass Inferencing is just the beginning

For more information on Inferencing, see Chapter 9 of the Learning SPARQL book (see the Books and Papers Tab on the class website).
All Data Models are Ignoring the First Data Model

2 Visualizing Aristotle’s 10-fold Division & His Varieties of Attribution

Figure 1 allows us to get a visual feel for Aristotle’s more fine-grained, 10-fold division, and the ways in which he thinks attributions work (and don’t work). Here, we are drawing on both the *Categories* readings and the *Topics* readings.

Figure 1: Picturing Aristotle’s 10-fold division & varieties of attribution — using trees.

Table 2 describes four different varieties of Aristotelian attribution which we can use to classify the attributions in Figure 1.

<table>
<thead>
<tr>
<th>Attribute is substance</th>
<th>Attribute is non-substance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thing is substance</td>
<td>Synonymous attribution (①, ②) (says what thing is)</td>
</tr>
<tr>
<td>Thing is non-substance</td>
<td>Not a genuine attribution (⑥)</td>
</tr>
</tbody>
</table>

Table 2: Four varieties of Aristotelian attribution.

---

1 In Figure 1, the arrows represent attributions. For instance, “Human —> Socrates” (i.e., attribution ② in Figure 1) is to be understood as the attribution “Socrates is [a] Human.” Not all of the arrows drawn in Figure 1 represent genuine attributions within Aristotle’s theory. For instance, arrows ⑤ and ⑥ are not genuine attributions, as far as Aristotle is concerned. Why not? Can you think of other non-genuine attributions?

2 Can you find relevant passages in the *Categories* and/or the *Topics* which justify (or refute!) each of the claims made in this handout?
Except for SIM, which was implemented in the 1980s and still exists at UT with Me

Based on the 1981 paper “Database Description with SDM: A Semantic Database Model” by Michael Hammer and Dennis McLeod, see class calendar for a pdf version of the paper.
Except for SIM, which was implemented in the 1980s and still exists at UT with Me