History and Types of Databases

Alan G. Labouseur, Ph.D.
Alan.Labouseur@Marist.edu
Data?

What is data?
What is data?
Data?

What is data?

What does it mean?

1  007  42  21  12  90  125  86  75  30  9
Data?

What is data?

What does it mean?

1 007 42 21 12 90 125 86 75 30 9

We need context.
Data?

What is data?

What does it mean?

With context we can draw conclusions from data.

With context we have information.
Data is Dangerous

What is data?

What does it mean?

What if we’re wrong?
Data is Dangerous, Information is Valuable

Data + Context = Information

Information is valuable.
Information is difficult to obtain.
Information is what we want.

And to get it, we need to impose structure for context.
Evolution

Consider the evolution of Data Management

- stone tablets
- punched cards
- flat files on tape
- hierarchical databases on DASD
- network databases on disk
- relational databases
- object stores
- object-relational databases (Third Manifesto?)
- graph databases
Evolution

Consider the evolution of Data Management

• stone tablets
• punched cards
• flat files on tape
• hierarchical databases on DASD
• network databases on disk
• relational databases
• object stores
• object-relational databases (Third Manifesto?)
• graph databases

Heavy data
Evolution

Consider the evolution of Data Management

- stone tablets
- punched cards
- flat files on tape
- hierarchical databases on DASD
- network databases on disk
- relational databases
- object stores
- object-relational databases (Third Manifesto?)
- graph databases

1890 Census

Big data

(Still heavy.)
Evolution

Consider the evolution of Data Management
- stone tablets
- punched cards
- flat files on tape
- hierarchical databases on DASD
- network databases on disk
- relational databases
- object stores
- object-relational databases (Third Manifesto?)
- graph databases

• Files of **Records** of **Fields** for a D&D-type game

<table>
<thead>
<tr>
<th>Players File</th>
<th>Items File</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Player 1 Record</strong>&lt;br&gt;<strong>Player 1 Fields</strong>&lt;br&gt;id : 1&lt;br&gt;name : James&lt;br&gt;rank : Captain&lt;br&gt;items: wand, gem</td>
<td><strong>Item 1 Record</strong>&lt;br&gt;<strong>Item 1 Fields</strong>&lt;br&gt;id : A&lt;br&gt;name : wand&lt;br&gt;desc : ...</td>
</tr>
<tr>
<td><strong>Player 2 Record</strong>&lt;br&gt;<strong>Player 2 Fields</strong>&lt;br&gt;id : 2&lt;br&gt;name : Leonard&lt;br&gt;rank : Admiral&lt;br&gt;items: gem, mace</td>
<td><strong>Item 2 Record</strong>&lt;br&gt;<strong>Item 2 Fields</strong>&lt;br&gt;id : B&lt;br&gt;name : gem&lt;br&gt;desc : ...</td>
</tr>
<tr>
<td><strong>Item 3 Record</strong>&lt;br&gt;<strong>Item 3 Fields</strong>&lt;br&gt;id : C&lt;br&gt;name : mace&lt;br&gt;desc : ...</td>
<td><strong>Item 4 Record</strong>&lt;br&gt;<strong>Item 4 Fields</strong>&lt;br&gt;id : D&lt;br&gt;name : sword&lt;br&gt;desc : ...</td>
</tr>
</tbody>
</table>
Evolution

Consider the evolution of Data Management:

- stone tablets
- punched cards
- flat files on tape
- hierarchical databases on DASD
- network databases on disk
- relational databases
- object stores
- object-relational databases (Third Manifesto?)
- graph databases

D&D Game

Player 1
James

Item A
wand

Item B
gem

Item B
gem

Item C
mace

Player 2
Leonard

Item D
sword
Consider the evolution of Data Management

- stone tablets
- punched cards
- flat files on tape
- hierarchical databases on DASD
- network databases on disk
- relational databases
- object stores
- object-relational databases (Third Manifesto?)
- graph databases
Evolution

Consider the evolution of Data Management

- stone tablets
- punched cards
- flat files on tape
- hierarchical databases on DASD
- network databases on disk
- relational databases
- object stores
- object-relational databases (Third Manifesto?)
- graph databases
Evolution

Consider the evolution of Data Management

- stone tablets
- punched cards
- flat files on tape
- hierarchical databases on DASD
- network databases on disk
- relational databases
- object stores
- object-relational databases (Third Manifesto?)
- graph databases

```
DB=# -- Players and their Items
DB=# select Players.name, Items.name
DB=# from Players inner join Inventory on Players.pid = Inventory.pid
DB=# inner join Items on Inventory.iid = Items.iid
DB=# ;

<table>
<thead>
<tr>
<th>name</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>James</td>
<td>wand</td>
</tr>
<tr>
<td>James</td>
<td>gem</td>
</tr>
<tr>
<td>Leonard</td>
<td>gem</td>
</tr>
<tr>
<td>Leonard</td>
<td>mace</td>
</tr>
</tbody>
</table>
(4 rows)
```
Evolution

Consider the evolution of Data Management

- stone tablets
- punched cards
- flat files on tape
- hierarchical databases on DASD
- network databases on disk
- relational databases
- object stores
- object-relational databases (Third Manifesto?)
- graph databases

```
DB=# -- Unused Items
DB=# select *
DB=# from Items
DB=# where iid not in (select iid
DB=# from Inventory);

 iid | name  | descr
-----+-------+-------
    | sword | ...
(1 row)
```
Evolution

Consider the evolution of Data Management

- stone tablets
- punched cards
- flat files on tape
- hierarchical databases on DASD
- network databases on disk
- relational databases
- object stores
- object-relational databases (Third Manifesto?)
- graph databases

```sql
DB=# -- Item use count v1
DB=# select iid, count(iid)
DB=# from Inventory
DB=# group by iid
DB=# order by count(iid) DESC;

<table>
<thead>
<tr>
<th>iid</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
</tr>
</tbody>
</table>

(3 rows)
```
Consider the evolution of Data Management

- stone tablets
- punched cards
- flat files on tape
- hierarchical databases on DASD
- network databases on disk
- relational databases
- object stores
- object-relational databases (Third Manifesto?)
- graph databases

```
DB=# -- Item use count v2, now with item names!
DB=# select Inventory.iid, Items.name, count(Inventory.iid)
DB=# from Inventory inner join Items on Inventory.iid = Items.iid
DB=# group by Inventory.iid, Items.name
DB=# order by count(Inventory.iid) DESC
DB=#

<table>
<thead>
<tr>
<th>iid</th>
<th>name</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>gem</td>
<td>2</td>
</tr>
<tr>
<td>A</td>
<td>wand</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>mace</td>
<td>1</td>
</tr>
</tbody>
</table>

(3 rows)
```
Evolution

SQL Script for Player, Items, and Inventory tables and a few queries

create table Players (  
    pid   int not null,  
    name  text,  
    rank text,  
    primary key (pid)  
);  

insert into Players(pid, name, rank)  
values (1, 'James', 'Captain'),  
(2, 'Leonard', 'Admiral');  

select *  
from Players;

create table Items (  
    iid   char(1) not null,  
    name  text,  
    descr text,  
    primary key (iid)  
);  

insert into Items (iid, name, descr)  
values ('A', 'wand', '...'),  
('B', 'gem', '...'),  
('C', 'mace', '...'),  
('D', 'sword', '...');  

select *  
from Items;

create table Inventory (  
    pid          int     not null references Players(pid),  
    iid          char(1) not null references Items(iid),  
    dateAcquired date,  
    primary key(pid, iid)  
);  

insert into Inventory (pid, iid, dateAcquired)  
values (1, 'A', '2020-01-23'),  
(1, 'B', '2020-01-23'),  
(2, 'B', '2020-01-23'),  
(2, 'C', '2020-01-23');  

select *  
from Inventory;

-- Players and their Items  
select Players.name, Items.name  
from Players inner join Inventory on Players.pid = Inventory.pid  
inner join Items on Inventory.iid = Items.iid;

-- Unused Items  
select *  
from Items  
where iid not in (select iid  
    from Inventory);

-- Item use count v1  
select iid, count(iid)  
from Inventory  
group by iid  
order by count(iid) DESC;

-- Item use count v2, now with item names!  
select Inventory.iid, Items.name, count(Inventory.iid)  
from Inventory inner join Items on Inventory.iid = Items.iid  
group by Inventory.iid, Items.name  
order by count(Inventory.iid) DESC;
Evolution

Consider the evolution of Data Management

- stone tablets
- punched cards
- flat files on tape
- hierarchical databases on DASD
- network databases on disk
- relational databases
- object stores
- object-relational databases (Third Manifesto?)
- graph databases

Players:
{
    {1, {James, Captain, {A, B}}},
    {2, {Leonard, Admiral, {B, C}}}
}

Items:
{
    {A, {wand, ..., {1}}},
    {B, {gem, ..., {1, 2}}},
    {C, {mace, ..., {3}}},
    {D, {sword, ..., {}}}
}
Consider the evolution of Data Management

- stone tablets
- punched cards
- flat files on tape
- hierarchical databases on DASD
- network databases on disk
- relational databases
- object stores
- object-relational databases (Third Manifesto?)
- graph databases
Evolution

Consider the evolution of Data Management

- stone tablets
- punched cards
- flat files on tape
- hierarchical databases on DASD
- network databases on disk
- relational databases
- object stores
- object-relational databases (Third Manifesto?)
- graph databases

A graph is like a network in most ways.

But graph databases are modern tools for managing them and gaining insight from the data pile.
Consider the evolution of Data Management

Graph . . .

as Matrix

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.</td>
<td>1</td>
<td>.</td>
<td>.</td>
<td>1</td>
<td>1</td>
<td>.</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>.</td>
<td>1</td>
<td>.</td>
<td>1</td>
<td>1</td>
<td>.</td>
</tr>
<tr>
<td>3</td>
<td>.</td>
<td>1</td>
<td>.</td>
<td>1</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>4</td>
<td>.</td>
<td>.</td>
<td>1</td>
<td>.</td>
<td>1</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>.</td>
<td>1</td>
<td>.</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>.</td>
<td>.</td>
<td>1</td>
<td>.</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>1</td>
<td>1</td>
<td>.</td>
</tr>
</tbody>
</table>
Evolution

Consider the evolution of Data Management

Graph . . .

as Adjacency List

[1]  2  5  6
[2]  1  3  5  6
[3]  2  4
[4]  3  5
[5]  1  2  4  6  7
[6]  1  2  5  7
[7]  5  6
Evolution

Consider the evolution of Data Management

- stone tablets
- punched cards
- flat files on tape
- hierarchical databases on DASD
- network databases on disk
- relational databases
- object stores
- object-relational databases (Third Manifesto?)
- graph databases
Evolution

G*The Dynamic Graph Database

Browser Application
Evolution

The Dynamic Graph Database

Graph Editor

```
  130  -- Evolution: 4 Incremental Graphs (with cloning)
  131  create graph 10.0
  132  add vertex 1 with attributes (color=black)
  133  add vertex 2 with attributes (color=black)
  134  add vertex 3 with attributes (color=black)
  135  add edge 1-2
  136  add edge 2-3
  137  clone graph 11.0 from 10.0
  138  add vertex a with attributes (color=white)
  139  add vertex b with attributes (color=white)
  140  add vertex c with attributes (color=white)
  141  add edge 1-a
  142  add edge 1-b
  143  add edge 1-c
  144  clone graph 12.0 from 11.0
  145  add vertex d with attributes (color=white)
  146  add vertex e with attributes (color=white)
  147  add vertex f with attributes (color=white)
  148  add edge 2-d
  149  add edge 2-e

```

---

8-vertex Full
32-vertex Ring
32-vertex Bipartite (16 pairs)
63-vertex Tree (branch factor = 2)
64-vertex Star
64-vertex 72-edge Erdos-Renyi Random
Other
Evolution

Interactive Console

```
Console

5  Graph 2.0 :
6    Vertices: 63
7    Edges : 62
8  Graph 1.0 :
9    Vertices: 4
10   Edges : 2
11  Graph 0.0 :
12    Vertices: 2
13    Edges : 1
14  G* create graph 4
15  New graph 4.0 was created.
16  G* add vertex Kirk
17  Vertex Kirk added to graph 4.0.
18  G* add vertex Spock
19  Vertex Spock added to graph 4.0.
20  G* add vertex McCoy
21  Vertex McCoy added to graph 4.0.
22  G* add edge Kirk-Spock
23  Edge from Kirk to Spock added to graph 4.0.
24  G* add edge Kirk-McCoy
25  Edge from Kirk to McCoy added to graph 4.0.
26  G* draw
27  Drawing complete.
28  G*
```
Evolution
Evolution

Top-k Query

Visualizer

Data

top 5 vertices by total degree: 10,11,12,13,14
3-hop neighbors of vertex 1: 
2,3,4,5,6,7,8,9,10,11,12,13,14,15
3-hop neighbors of vertex 18:
20,21,5,40,41,42,43,2,11,4,1,22,23
Evolution

Degree Distribution Query

Erdős-Rényi random graph

Degree Distribution
Evolution

top 20 vertices with the largest change in degree over consecutive graph snapshot pairs from 6 to 8:

<table>
<thead>
<tr>
<th>snapshotPairs</th>
<th>vertexID</th>
<th>change</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-&gt;6</td>
<td>1</td>
<td>+3</td>
</tr>
<tr>
<td>6-&gt;7</td>
<td>2</td>
<td>+5</td>
</tr>
<tr>
<td>7-&gt;8</td>
<td>3</td>
<td>+3</td>
</tr>
<tr>
<td>5-&gt;6</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>5-&gt;6</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>6-&gt;7</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6-&gt;7</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>6-&gt;7</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>7-&gt;8</td>
<td>2</td>
<td>-2</td>
</tr>
</tbody>
</table>

G5
G6
G7
G8
Evolution

Consider the evolution of Data Management

- stone tablets
- punched cards
- flat files on tape
- hierarchical databases on DASD
- network databases on disk
- relational databases
- object stores
- object-relational databases (Third Manifesto?)
- graph databases

What should we concentrate on?
Where should we spend our time?
Consider the evolution of Data Management

- stone tablets
- punched cards
- flat files on tape
- hierarchical databases on DASD
- network databases on disk
- **relational databases**
- object stores
- object-relational databases (Third Manifesto?)
- **graph databases**

We will spend most of our time on the Relational model and relational databases. And a little time on graphs.