Data Integration: From The Beginning to Today

Mary Roth
IBM Research – Almaden
January 2018
In the beginning, Codd made the relational database...

- In the world before Codd (~1960s), these data models prevailed:
  
  **Hierarchical (e.g. IMS)**
  
  - Library
    - Fiction
    - Non-Fiction
      - British
      - American
  
  **Network Model (e.g. CODASYL)**
  
  - Library
    - Non-fiction
      - American
      - British
  - American
    - Title: Fences
      - Author: August Wilson
    - Title: Team of Rivals
      - Author: Doris Kearns Goodwin

- But these models did not provide *data independence*;
  - If the data layout changed, the application had to change
  - If you wanted to change the layout, you often had to bring the whole system down
  - Changes had to occur over scheduled system down time.
    - Slow! Annoying! Expensive!

The relational database *revolutionized* how we electronically store and retrieve data.
And Codd saw that the relational database was good.

- It provided a structured, organized and application independent way to store data. [Codd 1970]

**Books in a Library...**

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Dewey</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fences</td>
<td>Wilson, August</td>
<td>812.54</td>
<td>Drama</td>
</tr>
<tr>
<td>The Hobbit</td>
<td>Tolkien, J. R. R.</td>
<td>823.912</td>
<td>Fiction</td>
</tr>
<tr>
<td>Team of Rivals</td>
<td>Goodwin, Doris Kearns</td>
<td>973.7092 B 22</td>
<td>Non-fiction</td>
</tr>
</tbody>
</table>

**Or books in a Bookstore...**

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>ISBN</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fences</td>
<td>Wilson, August</td>
<td>0452264014</td>
<td>8.47</td>
</tr>
<tr>
<td>The Hobbit</td>
<td>Tolkien, J. R. R.</td>
<td>9780007440849</td>
<td>10.55</td>
</tr>
<tr>
<td>Team of Rivals</td>
<td>Goodwin, Doris Kearns</td>
<td>0743270754</td>
<td>13.42</td>
</tr>
</tbody>
</table>

**Or even groceries in a Grocery Store!**

<table>
<thead>
<tr>
<th>Name</th>
<th>Aisle</th>
<th>UPC</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chobani Yogurt</td>
<td>Dairy</td>
<td>894700010014</td>
<td>1.00</td>
</tr>
<tr>
<td>Jiffy Peanut Butter</td>
<td>12</td>
<td>051500255162</td>
<td>5.49</td>
</tr>
<tr>
<td>Welch’s Fruit Snacks</td>
<td>14</td>
<td>034856050926</td>
<td>4.29</td>
</tr>
</tbody>
</table>

**Key Advantages:**

- Data could be restructured without affecting applications
- Data could be quickly and efficiently retrieved via arbitrarily complex queries
Over time, commercial enterprises began to rely on relational databases to run all aspects of their business.

Eventually, this led to new desires to combine data from multiple different places, e.g.,:

- How can I run analysis across all of the bank accounts at all of my branches?
- How can I find out if a book is at any library or book store near me?
These desires led to new challenges in data integration.

1. How do I query and combine data that is physically stored in different places?

2. How do I query and combine data that is described differently in different places?

3. How do I query and combine data that is represented differently in different places?
Challenge 1: How do I query data physically stored in different places?

Approach 1: Data Federation
leave the data in place and move the query to the data. [Roth, et. al, 1996, Tomasic, et. al, 1996]

Approach 2: Data Warehousing
Move the data to a single place and query it there. [Chaudhuri, et. al, 1997]
Data Federation: Move the query to the data.

*Mediation, Virtual Integration, Data Integration, Lazy Integration*

SELECT title, first+last
FROM UCLABooks
WHERE title = ‘Fences’

SELECT title, author, ISBN
FROM AllBooks
WHERE title = ‘Fences’

GET http://bookstore.com/api/books
/book?title='Fences'

<table>
<thead>
<tr>
<th>Title</th>
<th>Last name</th>
<th>First name</th>
<th>Dewey</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fences</td>
<td>Wilson</td>
<td>August</td>
<td>812.54</td>
<td>Drama</td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Dewey</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fences</td>
<td>Wilson, August</td>
<td>812.54</td>
<td>Drama</td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>ISBN</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fences</td>
<td>Wilson, August</td>
<td>0452264014</td>
<td>8.47</td>
</tr>
<tr>
<td>The Hobbit</td>
<td>Tolkien, J. R. R.</td>
<td>9780007440849</td>
<td>10.5</td>
</tr>
<tr>
<td>Team of Rivals</td>
<td>Goodwin, Doris Kearns</td>
<td>0743270754</td>
<td>13.4</td>
</tr>
</tbody>
</table>

SELECT title, author
FROM UCSBBooks
WHERE title = ‘Fences’
Data Warehousing: consolidate the data in one central location.

Materialization, Data Exchange, ETL (Extract, Transform, Load), Eager Integration

SELECT title, author, ISBN, location
FROM AllBooks A, Locations L
WHERE title = 'Fences'

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>ISBN</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fences</td>
<td>Wilson, August</td>
<td>0452264014</td>
<td>8.47</td>
</tr>
<tr>
<td>The Hobbit</td>
<td>Tolkien, J. R.</td>
<td>9780007440489</td>
<td>10.5</td>
</tr>
<tr>
<td>Team of Rivals</td>
<td>Goodwin, Doris</td>
<td>0743270754</td>
<td>13.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ISBN</th>
<th>Location</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>0452264014</td>
<td>UCLA</td>
<td>1</td>
</tr>
<tr>
<td>0452264014</td>
<td>UCSB</td>
<td>2</td>
</tr>
<tr>
<td>0452264014</td>
<td>UCLABookStore</td>
<td>2</td>
</tr>
</tbody>
</table>
Challenge 2: How do I query data that is described differently in different sources?

Declarative schema mappings describe the relationships between the schemas of heterogeneous data sources [Miller, et. al, 2000]

\[ \text{Source schemas} \rightarrow \text{Common target schema} \]

\[ \text{UCLA Library} \rightarrow \text{AllBooks} \rightarrow \text{UCLA Bookstore} \]

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>ISBN</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllBooks</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Schema mappings can be compiled into code to restructure the source data to a unified target schema

```
SELECT TITLE, CONCAT(FIRSTNAME, LASTNAME), DEWEY2ISBN(DEWEY), 0.00
FROM UCLALIBRARY
```

Challenge 3: How do I query data that is \textit{represented} differently in different sources?

Data can be dirty and may require \textit{cleaning}; correcting errors, removing duplicates and resolving inconsistencies [Hernandez, et. al, 1998]

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>ZIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charles E Young Research Library</td>
<td>290 Charles E Young Dr N</td>
<td>LA</td>
<td>California</td>
<td>90095</td>
</tr>
<tr>
<td>Research Library</td>
<td>North 280 Charles Young Drive</td>
<td>Los Angeles</td>
<td>CA</td>
<td></td>
</tr>
<tr>
<td>UCLA Charles Young Research Library</td>
<td>290 Charles Young Drive</td>
<td>Los Angeles</td>
<td>90059</td>
<td></td>
</tr>
</tbody>
</table>

This is especially challenging if data that identifies entities across sources is inconsistent; in this case, \textit{entity resolution} is needed [Hernandez, et. al, 2013]
...And then Tim Berners-Lee brought us the world wide web, introducing a new generation of integration challenges.

**Veracity:** Data is no longer tightly controlled by DBAs and governed applications

**Volume:** There is a lot of it!

**Velocity:** It is being generated at a faster than we can consume!

**Variety:** Data is no longer structured
Relevant Publications

Relational and Distributed Databases


Data federation


• Anthony Tomasic, Louiqa Raschid, Patrick Valduriez: Scaling Heterogeneous Databases and the Design of Disco. ICDCS 1996: 449-457

• Laura M. Haas, Donald Kossmann, Edward L. Wimmers, Jun Yang: Optimizing Queries Across Diverse Data Sources. VLDB 1997: 276-285

• Mary Tork Roth, Peter M. Schwarz: Don't Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources. VLDB 1997: 266-275
Relevant Publications, cont’d

Data Warehousing


Schema Mapping

• Renée J. Miller, Laura M. Haas, Mauricio A. Hernández, Schema Mapping as Query Discovery, Proceedings of the 26th International Conference on Very Large Data Bases, p.77-88, September 10-14, 2000

• Sergey Melnik, Atul Adya, Philip A. Bernstein, Compiling mappings to bridge applications and databases, Proceedings of the 2007 ACM SIGMOD international conference on Management of data, June 11-14, 2007, Beijing, China

Relevant Publications, cont’d

Data Cleansing
• Vijayshankar Raman, Joseph M. Hellerstein: Potter's Wheel: An Interactive Data Cleaning System. VLDB 2001: 381-390

Entity Resolution

New Challenges in Data Integration: Volume, Variety, Volume and Veracity