Data Engineering for Data Science

M. Tamer Özsu

University of Waterloo

University of Waterloo | DSg Data Systems Group

Context – Two Main Concerns

Big data management

Data preparation

- Data enrichment, integration and storage
  - ETL/ELT process (?)
  - Data lakes
- Storage and management of big datasets
- Data processing platforms

Context – Two Main Concerns

Big data management

- Data enrichment, integration and storage
  - ETL/ELT process (?)
  - Data lakes
- Storage and management of big datasets
- Data processing platforms

Data preparation

- Data acquisition/gathering
- Data cleaning
- Data provenance & lineage

Wealth of Data

Traditional database applications	• Store numeric short textual information • Typically for managing enterprises
Text and multimedia databases	• Store documents, digital images, audio, and video streams
Geographic information systems (GIS)	• Store maps, weather data, and satellite images • Route-finding, agriculture, and natural resource management
Data warehouses & analytics systems (OLAP)	• Store historical business information • For business analytics and decision support
Real-time and active database technology	• Store process models, constraints, and key performance indicators • Control industrial and manufacturing processes

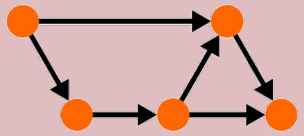
Plan

Data Management Basics

Big Data Concerns

Data Integration

Data Quality/Cleaning



Data Provenance

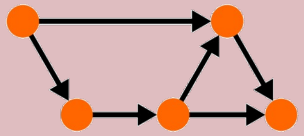
Plan


Data Management Basics

Big Data Concerns


Data Integration


Data Quality/Cleaning


Data Provenance

Big Data Management (Data Management Basics, Big Data Concerns, Data Integration)

Data Preparation (Data Quality/Cleaning, Data Provenance)

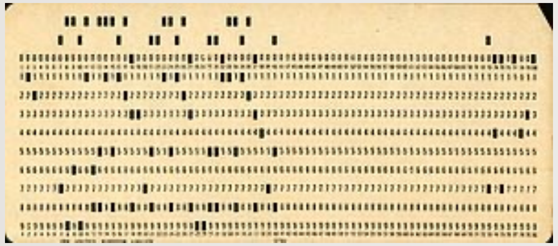
PLAN

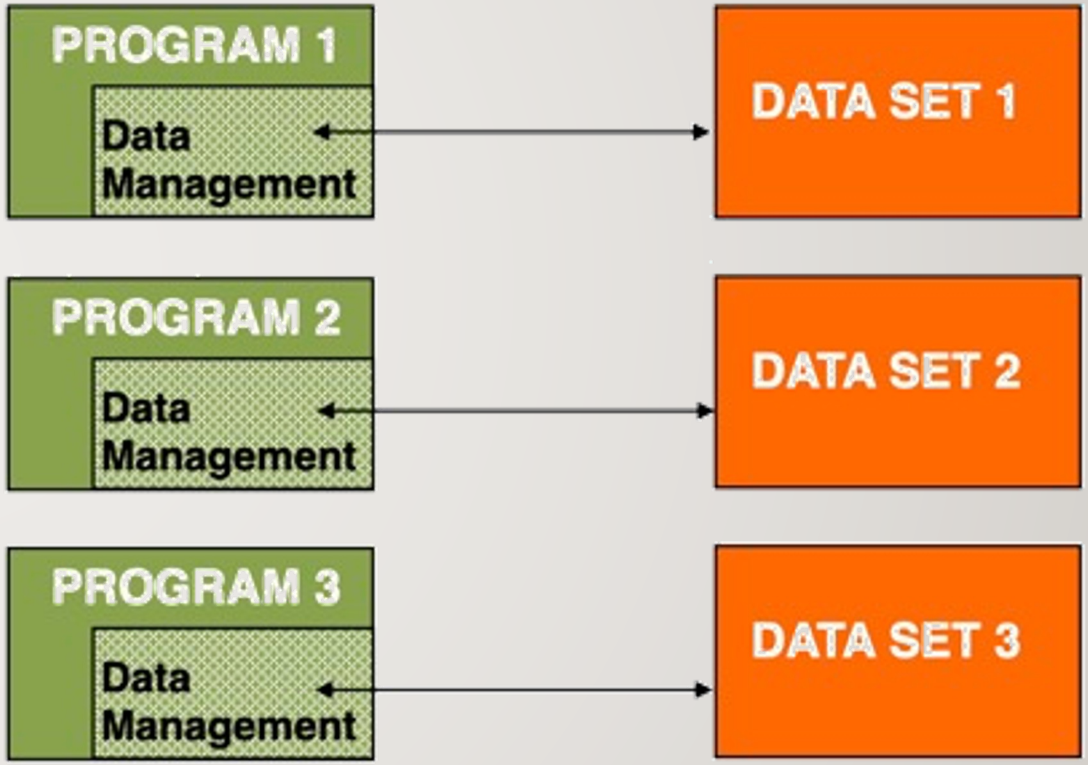
Data Management Basics

Early data management – ancient history

1950s and moving into 1960s

- Data are not stored on disk
- One data set per program
- High data redundancy

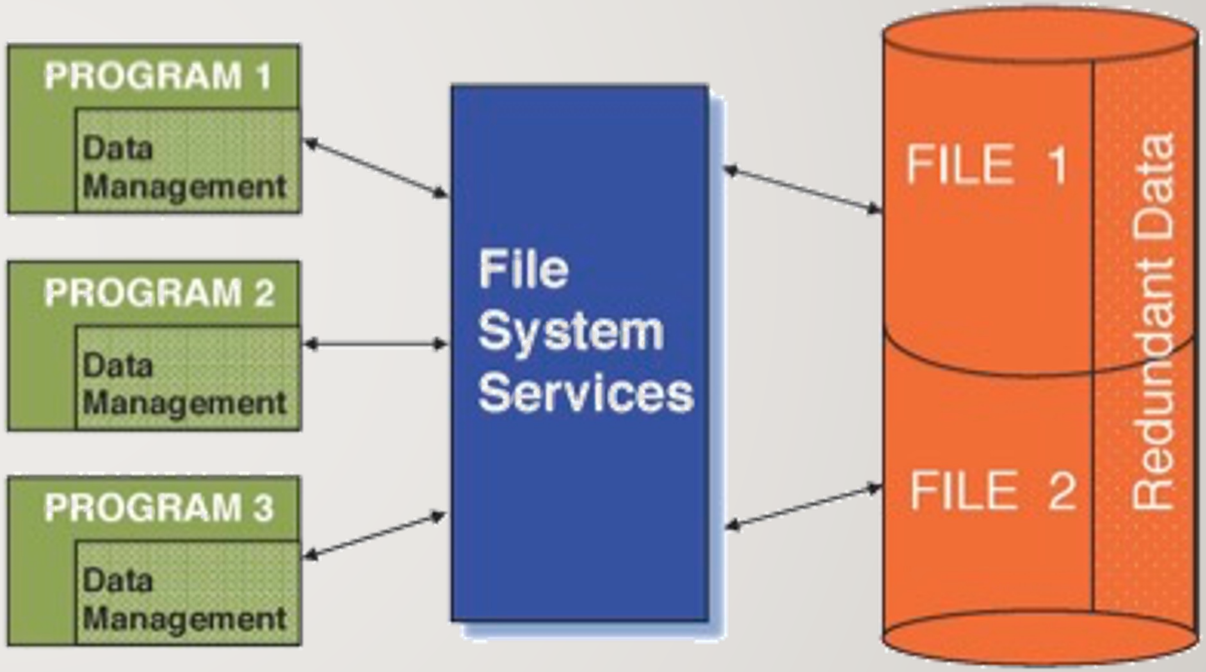




File Processing – More Recent History

Starting 1960s

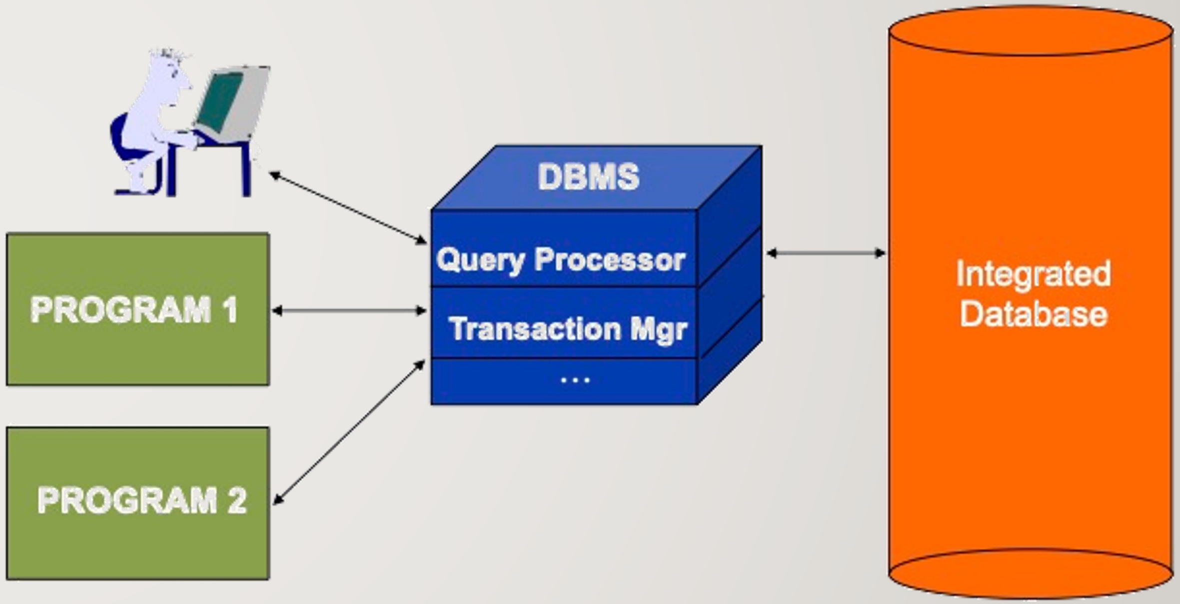
- File systems introduced
- Various access methods
- Disk drives are introduced
- One file shared by several programs



Database Approach

Starting mid-1960s

- A more integrated approach to organizing, managing, and accessing data
- Avoids uncontrolled duplication
- Better integrity, security and privacy control
- Databases really rely on disk storage



WHAT IS A DATABASE MANAGEMENT SYSTEM

- Database Management System (DBMS): A program (or set of programs) that manages details related to storage and access for a database.
- Examples of database management systems
  - IBM's DB2, Microsoft's Access and SQL Server, Oracle, MySQL, SAP Hana, …

Data Model

- Every DBMS has a data model
  - Description of the structure of
- Logical data model
  - Representation scheme within a database
- Physical model
  - Definition of how the data is stored and the access paths to reach data

PATIENT

PID	PName	InsNo	Address

ROOM

RoomNo	RoomType	Capacity

MEDICINE

MedCode	Name	Administration

SUPPLIER

SID	Name	Address

EMPLOYEE

EID	EName	Address	Sex	Salary

Relational Model

- Invented by E.F. Codd in 1970
- British; fighter pilot in WW2
- IBM 1948-53
- Was in Canada 1953-57
- Worked at IBM San Jose Research Lab. 1957-84
- Died in 2003

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 with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on *n*-ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

KEY WORDS AND PHRASES: data bank, data base, data structure, data organization, hierarchies of data, networks of data, relations, derivability, redundancy, consistency, composition, join, retrieval language, predicate calculus, security, data integrity
CR CATEGORIES: 3.70, 3.73, 3.75, 4.20, 4.22, 4.29

1. Relational Model and Normal Form

1.1. Introduction

This paper is concerned with the application of elementary relation theory to systems which provide shared access to large banks of formatted data. Except for a paper by Childs [1], the principal application of relations to data systems has been to deductive question-answering systems. Levein and Maron [2] provide numerous references to work in this area.

In contrast, the problems treated here are those of *data independence*—the independence of application programs and terminal activities from growth in data types and changes in data representation—and certain kinds of *data inconsistency* which are expected to become troublesome even in nondeductive systems.

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.

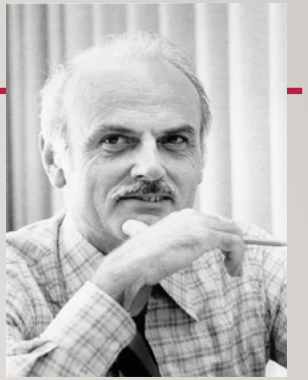
A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations—these are discussed in Section 2. The network model, on the other hand, has spawned a number of confusions, not the least of which is mistaking the derivation 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 standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of systems to support the relational model are not discussed.

1.2. Data Dependencies in Present Systems

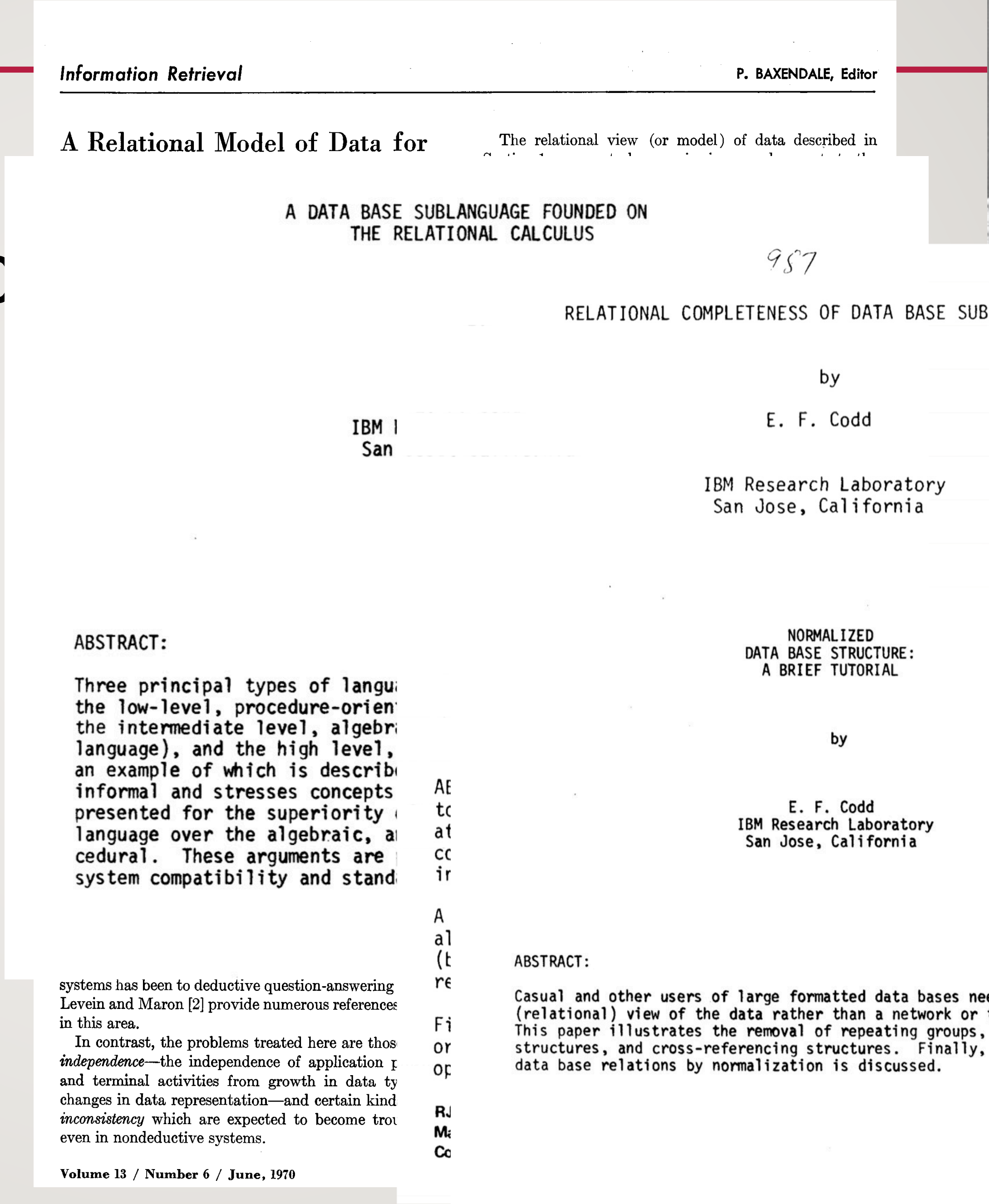
The provision of data description tables in recently developed information systems represents a major advance toward the goal of data independence [5, 6, 7]. Such tables facilitate changing certain characteristics of the data representation stored in a data bank. However, the variety of data representation characteristics which can be changed *without logically impairing some application programs* is still quite limited. Further, the model of data with which users interact is still cluttered with representational properties, particularly in regard to the representation of collections of data (as opposed to individual items). Three of the principal kinds of data dependencies which still need to be removed are: ordering dependence, indexing dependence, and access path dependence. In some systems these dependencies are not clearly separable from one another.

1.2.1. *Ordering Dependence.* Elements of data in a data bank may be stored in a variety of ways, some involving no concern for ordering, some permitting each element to participate in one ordering only, others permitting each element to participate in several orderings. Let us consider those existing systems which either require or permit data elements to be stored in at least one total ordering which is closely associated with the hardware-determined ordering of addresses. For example, the records of a file concerning parts might be stored in ascending order by part serial number. Such systems normally permit application programs to assume that the order of presentation of records from such a file is identical to (or is a subordering of) the



Relational Model

- Invented by E.F. Codd in 1970
- British; fighter pilot in WW2
- IBM 1948-53
- Was in Canada 1953-57
- Worked at IBM San Jose Research Lab. 1957-84
- Died in 2003



Information Retrieval — P. BAXENDALE, Editor

A Relational Model of Data for

The relational view (or model) of data described in

systems has been to deductive question-answering
Levein and Maron [2] provide numerous references
in this area.

In contrast, the problems treated here are thos
independence—the independence of application p
and terminal activities from growth in data ty
changes in data representation—and certain kind
inconsistency which are expected to become trou
even in nondeductive systems.

Volume 13 / Number 6 / June, 1970

A DATA BASE SUBLANGUAGE FOUNDED ON
THE RELATIONAL CALCULUS

IBM
San

ABSTRACT:

Three principal types of langu
the low-level, procedure-orien
the intermediate level, algebr
language), and the high level,
an example of which is describ
informal and stresses concepts
presented for the superiority
language over the algebraic, a
cedural. These arguments are
system compatibility and stand

957

RELATIONAL COMPLETENESS OF DATA BASE SUBLANGUAGES

by

E. F. Codd

IBM Research Laboratory
San Jose, California

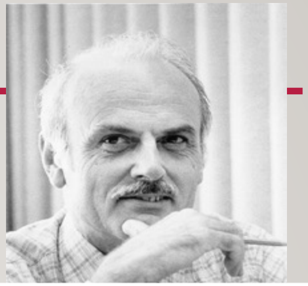
NORMALIZED
DATA BASE STRUCTURE:
A BRIEF TUTORIAL

by

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

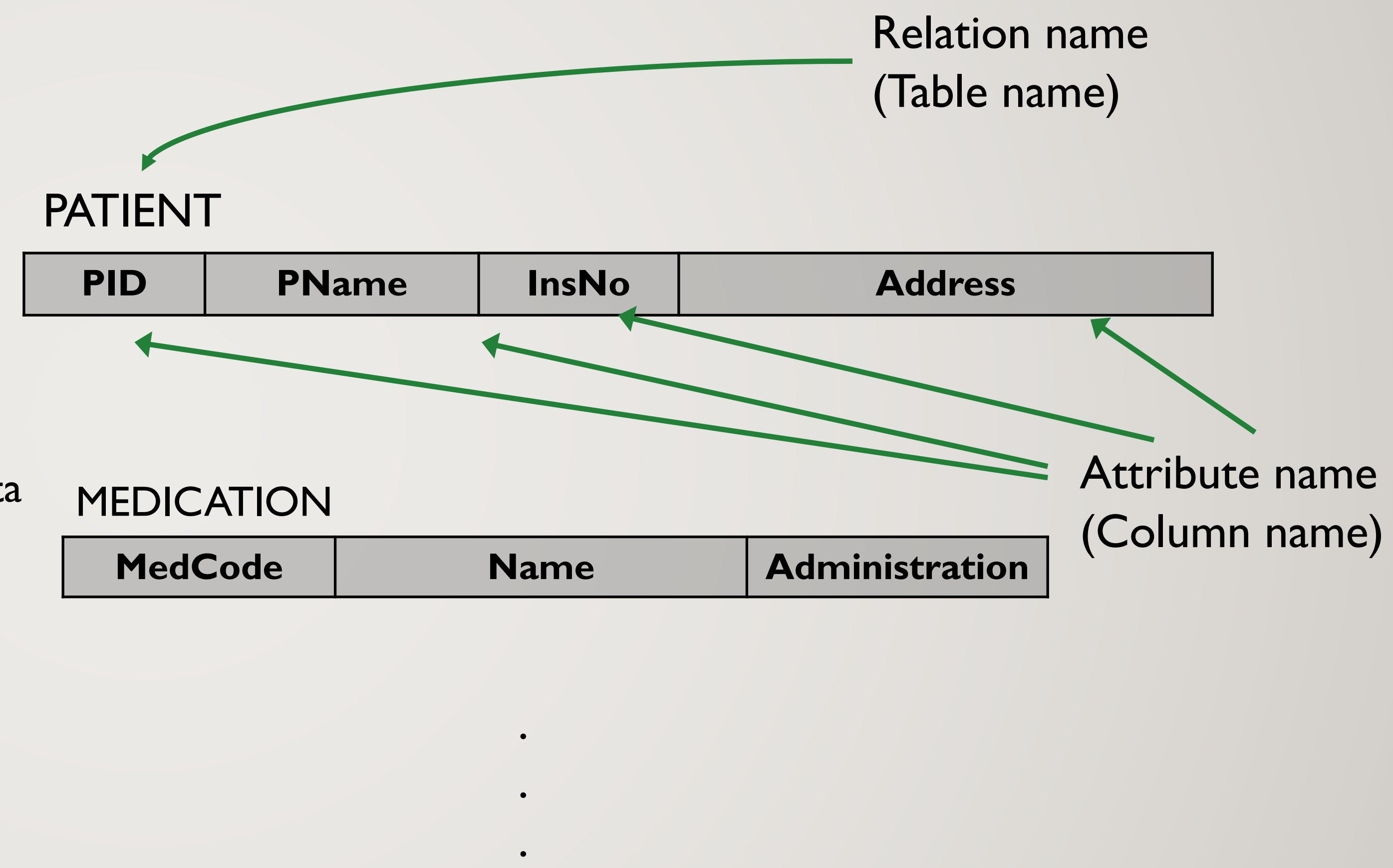
ABSTRACT:

Casual and other users of large formatted data bases need a simple tabular (relational) view of the data rather than a network or tree-structured view. This paper illustrates the removal of repeating groups, hierarchic and plex structures, and cross-referencing structures. Finally, the simplification of data base relations by normalization is discussed.



Relational Model Concepts

- Represent the miniworld as a collection of relations
- Each relation holds facts about a particular entity/concept
  - Each relation has a number of attributes
    - Store the property values of that entity/concept
  - Each relation consists of a set of tuples
    - Each tuple represents a record of related data values
    - Facts that typically correspond to an entity/concept
- A relation can be represented as a table
  - Each row corresponds to a tuple
  - Each column corresponds to an attribute



Relational Model Concepts

- Represent the miniworld as a collection of relations
- Each relation holds facts about a particular entity/concept
  - Each relation has a number of attributes
    - Store the property values of that entity/concept
  - Each relation consists of a set of tuples
    - Each tuple represents a record of related data values
    - Facts that typically correspond to an entity/concept
- A relation can be represented as a table
  - Each row corresponds to a tuple
  - Each column corresponds to an attribute

PATIENT(PID, Pname, InsNo, Address)

ROOM(RoomNo, RoomType, Capacity)

MEDICINE(MedCode, Name, Administration)

SUPPLIER(SID, Name, Address)

EMPLOYEE(EID, EName, Address, Sex, Salary)

NURSE(EID, EName, Address, Sex, Salary, Position)

DOCTOR(EID, EName, Address, Sex, Salary, Specialization)

TREATMENT(PID, EID, Tbegin, Tend, Cost)

ASSIGNED(PatientNo, RoomNo, Admitted, Discharged)

PRESCRIBED(PID, MedCode)

SUPPLIED_BY(MedCode, SID, Price)

GOVERNS(NurseID, RNo, Begin, End)

CONTACT(PID, PContactInfo)

One Possible (Partial) Database Instance

For a relation schema *R*, instance *r*(*R*)

PATIENT

PID	PName	InsNo	Address
49875	Jane Smith	ON8677	54 Beachwood St, Waterloo, Ontario
15948	Ali Nadir	ON7740	583 College St., Toronto, Ontario
98143	Jiang Ni	AB39658	189-95 Ave., Edmonton, Alberta

ROOM

RoomNo	RoomType	Capacity
DC212	Private	1
DC259	Semi-private	2
MC187	Ward	4
MC489	Ward	8
RD552	Semi-private	2

MEDICINE

MedCode	Name	Administration
00216666	Novasen	Oral
02439816	Ramipril	Oral
02220261	Penicillin G	Intramuscular
02245385	Symbicort	Inhale
02339501	Afinitor	Oral

GOVERNS

RNo	NurseID	Begin	End
DC259	94729532	25-Jan-2017	31-Dec-2020
RD552	69367883	20-Apr-2017	31-Dec-2020
DC212	94729532	28-Jul-2016	20-May-2021
MC489	33857201	1-Sept-2020	5-Mar-2022
MC187	33857201	27-Oct-2019	30-Nov-2020

One Possible (Partial) Database Instance

For a relation schema *R*, instance *r*(*R*)

PATIENT

PID	PName	InsNo	Address
49875	Jane Smith	ON8677	54 Beachwood St, Waterloo, Ontario
15948	Ali Nadir	ON7740	583 College St., Toronto, Ontario
98143	Jiang Ni	AB39658	189-95 Ave., Edmonton, Alberta

ROOM

RoomNo	RoomType	Capacity
DC212	Private	1
DC259	Semi-private	2
MC187	Ward	4
MC489	Ward	8
RD552	Semi-private	2

Attributes (columns)

MEDICINE

MedCode	Name	Administration
00216666	Novasen	Oral
02439816	Ramipril	Oral
02220261	Penicillin G	Intramuscular
02245385	Symbicort	Inhale
02339501	Afinitor	Oral

Tuples (rows)

GOVERNS

RNo	NurseID	Begin	End
DC259	94729532	25-Jan-2017	31-Dec-2020
RD552	69367883	20-Apr-2017	31-Dec-2020
DC212	94729532	28-Jul-2016	20-May-2021
MC489	33857201	1-Sept-2020	5-Mar-2022
MC187	33857201	27-Oct-2019	30-Nov-2020

Relational Database Properties

- Atomic values
  - Composite and multivalued attributes not allowed

PATIENT

PID	PName	InsNo	Contact	Address
49875	Jane Smith	ON8677	5195552389	54 Beachwood St, Waterloo, Ontario
15948	Ali Nadir	ON7740	22698[illegible]6 51999[illegible]5	583 College St., Toronto, Ontario
98143	Jiang Ni	AB39658	7809495678	189-95 Ave., Edmonton, Alberta

This is a single string value

Relational Database Properties

- Atomic values
  - Composite and multivalued attributes not allowed
- Normalized
  - Each fact in its own table

PATIENT(PID, Pname, InsNo, Address)

ROOM(RoomNo, RoomType, Capacity)

MEDICINE(MedCode, Name, Administration)

SUPPLIER(SID, Name, Address)

EMPLOYEE(EID, EName, Address, Sex, Salary)

NURSE(EID, EName, Address, Sex, Salary, Position)

DOCTOR(EID, EName, Address, Sex, Salary, Specialization)

TREATMENT(PID, EID, Tbegin, Tend, Cost)

ASSIGNED(PatientNo, RoomNo, Admitted, Discharged)

PRESCRIBED(PID, MedCode)

SUPPLIED_BY(MedCode, SID, Price)

GOVERNS(NurseID, RNo, Begin, End)

CONTACT(PID, PContactInfo)

Relational Database Properties

- Atomic values
  - Composite and multivalued attributes not allowed
- Normalized
  - Each fact in its own table
- Time-varying
  - Updates occur to data
  - Relation represents the state at a given time *t*

PATIENT

PID	PName	InsNo	Contact	Address
49875	Jane Smith	ON8677	5195552389	54 Beachwood St, Waterloo, Ontario
15948	Ali Nadir	ON7740	2269873456 5199905555	583 College St., Toronto, Ontario
98143	Jiang Ni	AB39658	7809495678	189-95 Ave., Edmonton, Alberta

PID	PName	InsNo	Address
49875	Jane Smith	ON8677	54 Beachwood St, Waterloo, Ontario
15948	Ali Nadir	ON7740	583 College St., Toronto, Ontario
98143	Jiang Ni	AB39658	189-95 Ave., Edmonton, Alberta
75880	Tom White	ON6409	884 Water St., Burlington, Ontario

CONSTRAINTS

- A relational schema captures only the structure of relations
- Can be extended by rules called constraints
- Examples:
  - Each relation has to have a key attribute (PID)
  - Functional dependency (FD)
  - Foreign key (FK)

PATIENT

PID	PName	InsNo	Address
49875	Jane Smith	ON8677	54 Beachwood St, Waterloo, Ontario
15948	Ali Nadir	ON7740	583 College St., Toronto, Ontario
98143	Jiang Ni	AB39658	189-95 Ave., Edmonton, Alberta

PRESCRIBED

PID	MedCode
49875	00216666
15948	02439786
74956	02521156

CONSTRAINTS

- A relational schema captures only the structure of relations
- Can be extended by rules called constraints
- Examples:
  - Each relation has to have a key attribute (PID)
  - Functional dependency (FD)
  - Foreign key (FK)

Key

PATIENT

PID	**PName**	**InsNo**	**Address**
49875	Jane Smith	ON8677	54 Beachwood St, Waterloo, Ontario
15948	Ali Nadir	ON7740	583 College St., Toronto, Ontario
98143	Jiang Ni	AB39658	189-95 Ave., Edmonton, Alberta

PRESCRIBED

PID	**MedCode**
49875	00216666
15948	02439786
74956	02521156

CONSTRAINTS

- A relational schema captures only the structure of relations
- Can be extended by rules called constraints
- Examples:
  - Each relation has to have a key attribute (PID)
  - Functional dependency (FD)
  - Foreign key (FK)

Key

PATIENT

PID	**PName**	**InsNo**	**Address**
49875	Jane Smith	ON8677	54 Beachwood St, Waterloo, Ontario
15948	Ali Nadir	ON7740	583 College St., Toronto, Ontario
98143	Jiang Ni	AB39658	189-95 Ave., Edmonton, Alberta

Key

PRESCRIBED

PID	**MedCode**
49875	00216666
15948	02439786
74956	02521156

CONSTRAINTS

- A relational schema captures only the structure of relations
- Can be extended by rules called constraints
- Examples:
  - Each relation has to have a key attribute (PID)
  - Functional dependency (FD)
  - Foreign key (FK)

Key

FD

PATIENT

PID	**PName**	**InsNo**	**Address**
49875	Jane Smith	ON8677	54 Beachwood St, Waterloo, Ontario
15948	Ali Nadir	ON7740	583 College St., Toronto, Ontario
98143	Jiang Ni	AB39658	189-95 Ave., Edmonton, Alberta

Key

PRESCRIBED

PID	**MedCode**
49875	00216666
15948	02439786
74956	02521156

CONSTRAINTS

- A relational schema captures only the structure of relations
- Can be extended by rules called constraints
- Examples:
  - Each relation has to have a key attribute (PID)
  - Functional dependency (FD)
  - Foreign key (FK)

Key

FD

PATIENT

PID	PName	InsNo	Address
49875	Jane Smith	ON8677	54 Beachwood St, Waterloo, Ontario
15948	Ali Nadir	ON7740	583 College St., Toronto, Ontario
98143	Jiang Ni	AB39658	189-95 Ave., Edmonton, Alberta

FK

Key

PRESCRIBED

PID	MedCode
49875	00216666
15948	02439786
74956	02521156

How to Access the Database

- SQL – Declarative Query Language
  - State what you want in the result, not how to compute it

PATIENT

PID	PName	InsNo	Street	City	Province
49875	Jane Smith	ON8677	54 Beachwood St.	Waterloo	Ontario
15948	Ali Nadir	ON7740	583 College St.	Toronto	Ontario
98143	Jiang Ni	AB39658	189-95 Ave.	Edmonton	Alberta
75880	Tom White	ON6409	884 Water St.	Burlington	Ontario
13086	Mark Smith	ON7843	54 King St.	Waterloo	Ontario

How to Access the Database

- SQL – Declarative Query Language
  - State what you want in the result, not how to compute it
- Example 1 : *Find the names and insurance numbers of patients in Ontario*

PATIENT

PID	PName	InsNo	Street	City	Province
49875	Jane Smith	ON8677	54 Beachwood St.	Waterloo	Ontario
15948	Ali Nadir	ON7740	583 College St.	Toronto	Ontario
98143	Jiang Ni	AB39658	189-95 Ave.	Edmonton	Alberta
75880	Tom White	ON6409	884 Water St.	Burlington	Ontario
13086	Mark Smith	ON7843	54 King St.	Waterloo	Ontario

```
SELECT   PName, InsNo
FROM     PATIENT
WHERE    Province = 'Ontario';
```

PName	InsNo
Jane Smith	ON8677
Ali Nadir	ON7740
Tom White	ON6409
Mark Smith	ON7843

How to Access the Database

- SQL – Declarative Query Language
  - State what you want in the result, not how to compute it
- Example 1 : *Find the names and insurance numbers of patients in Ontario*
- Example 2: *Retrieve the name, city and treatment cost of all patients whose treatment cost more than $15,000*

PATIENT

PID	PName	InsNo	Street	City	Province
49875	Jane Smith	ON8677	54 Beachwood St.	Waterloo	Ontario
15948	Ali Nadir	ON7740	583 College St.	Toronto	Ontario
98143	Jiang Ni	AB39658	189-95 Ave.	Edmonton	Alberta
75880	Tom White	ON6409	884 Water St.	Burlington	Ontario
13086	Mark Smith	ON7843	54 King St.	Waterloo	Ontario

TREATMENT

PID	Begin	End	EID	Cost
49875	25-Jan-2017	31-Dec-2020	34757200	11500
15948	20-Apr-2017	31-Dec-2020	85993920	37300
49875	1-Sept-2020	5-Mar-2022	34757200	25000

```
SELECT  P.PName, P.City, T.Cost
FROM    PATIENT P, TREATMENT T
WHERE   Cost > 15000
AND     P.PID = T.PID;
```

PName	City
Jane Smith	Waterloo
Ali Nadir	Toronto
Jiang Ni	Edmonton

Plan

Data Management Basics

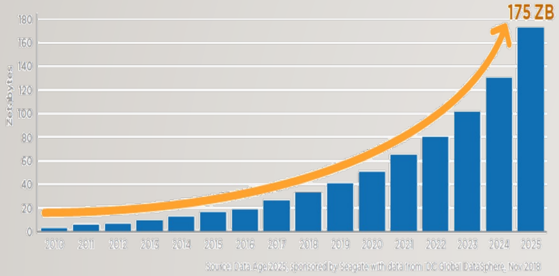
Big Data Concerns

Big Data – Four Vs

"refers to large, diverse, complex, longitudinal, and/or distributed data sets generated from instruments, sensors, Internet transactions, email, video, click streams, and/or all other digital sources available today and in the future."

NSF BIGDATA Solicitation

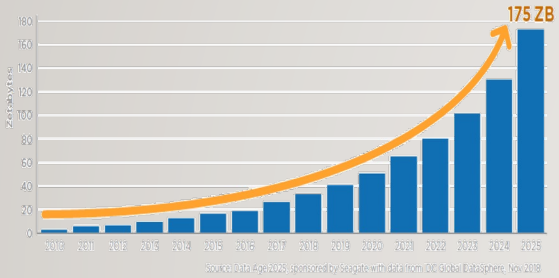
Big Data – Four Vs



Volume

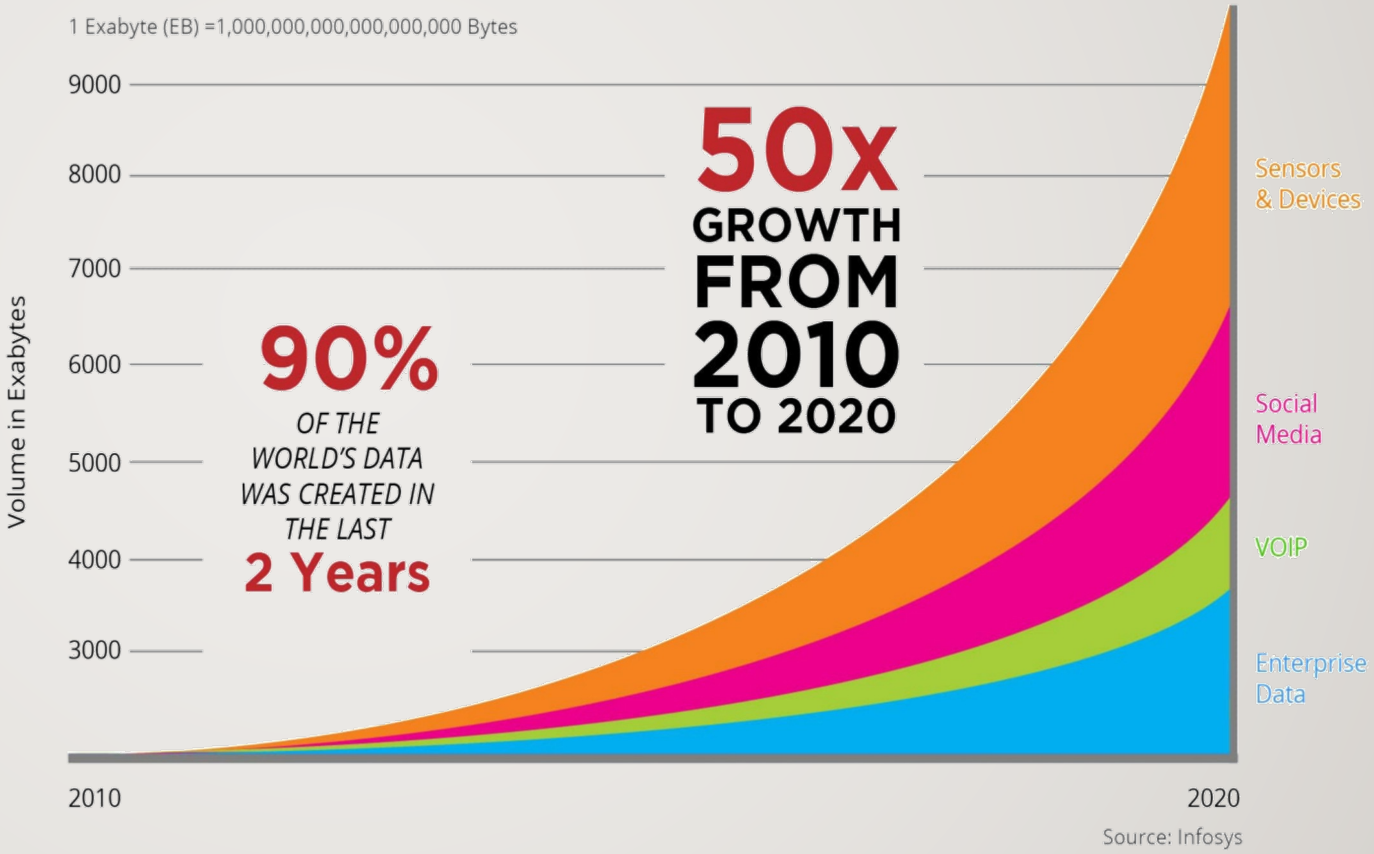
- Scale of data
- Data at rest

BIG DATA – FOUR VS

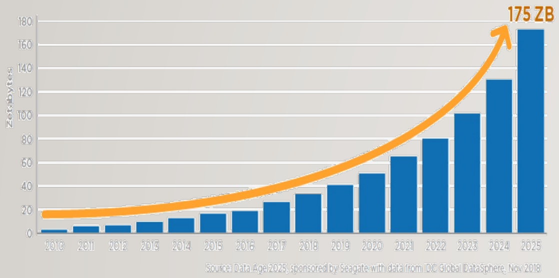


Volume

- Scale of data
- Data at rest



Big Data – Four Vs



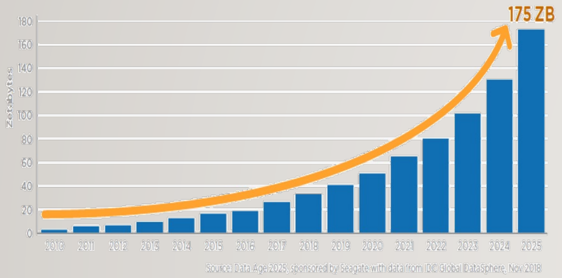
Volume

- Scale of data
- Data at rest

Variety

- Forms of data
- Unstructured challenges

Big Data – Four Vs

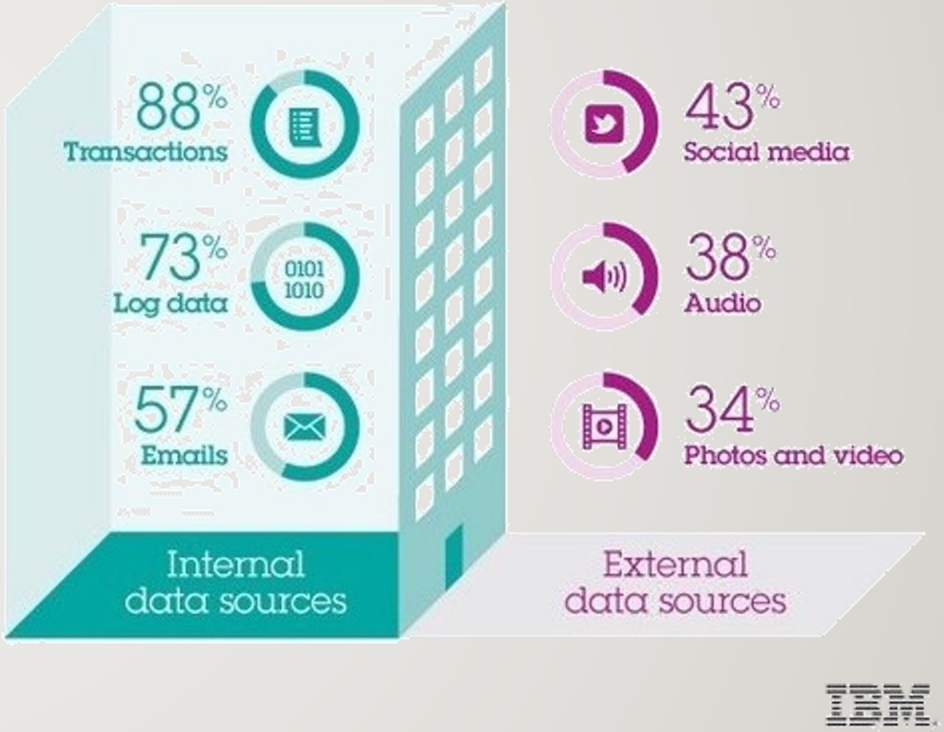


Volume

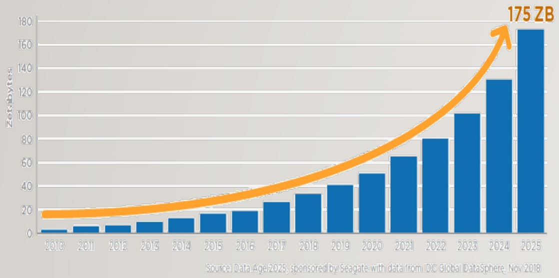
- Scale of data
- Data at rest

Variety

- Forms of data
- Unstructured challenges



Big Data – Four Vs

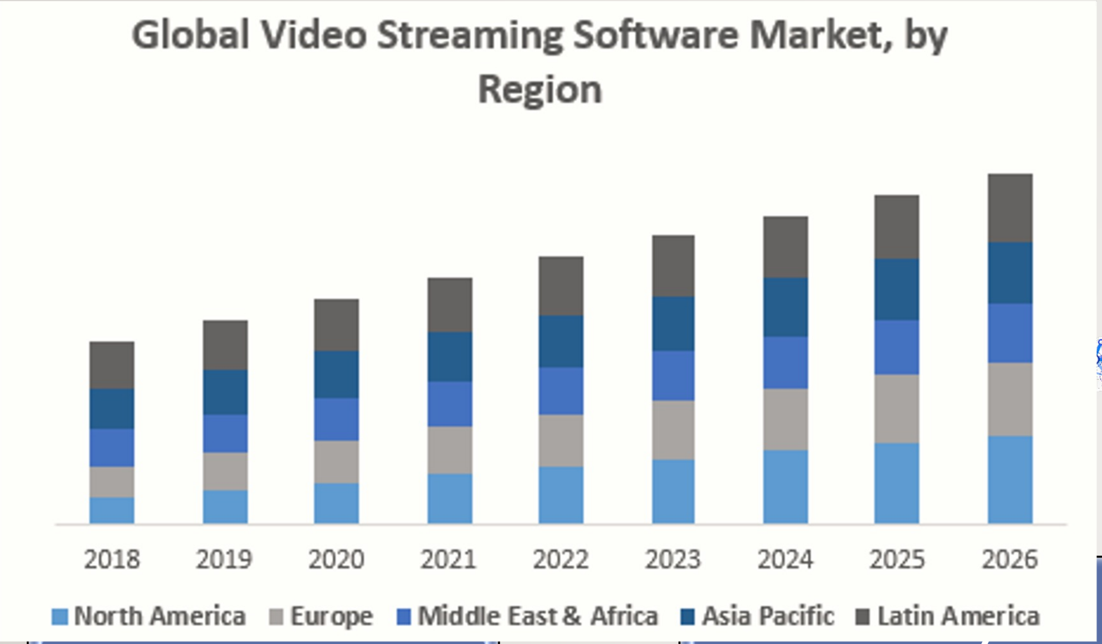


Volume
• Scale of data • Data at rest

Variety
• Forms of data • Unstructured challenges

Velocity
• Streaming data • Data in motion

Big Data – Four Vs



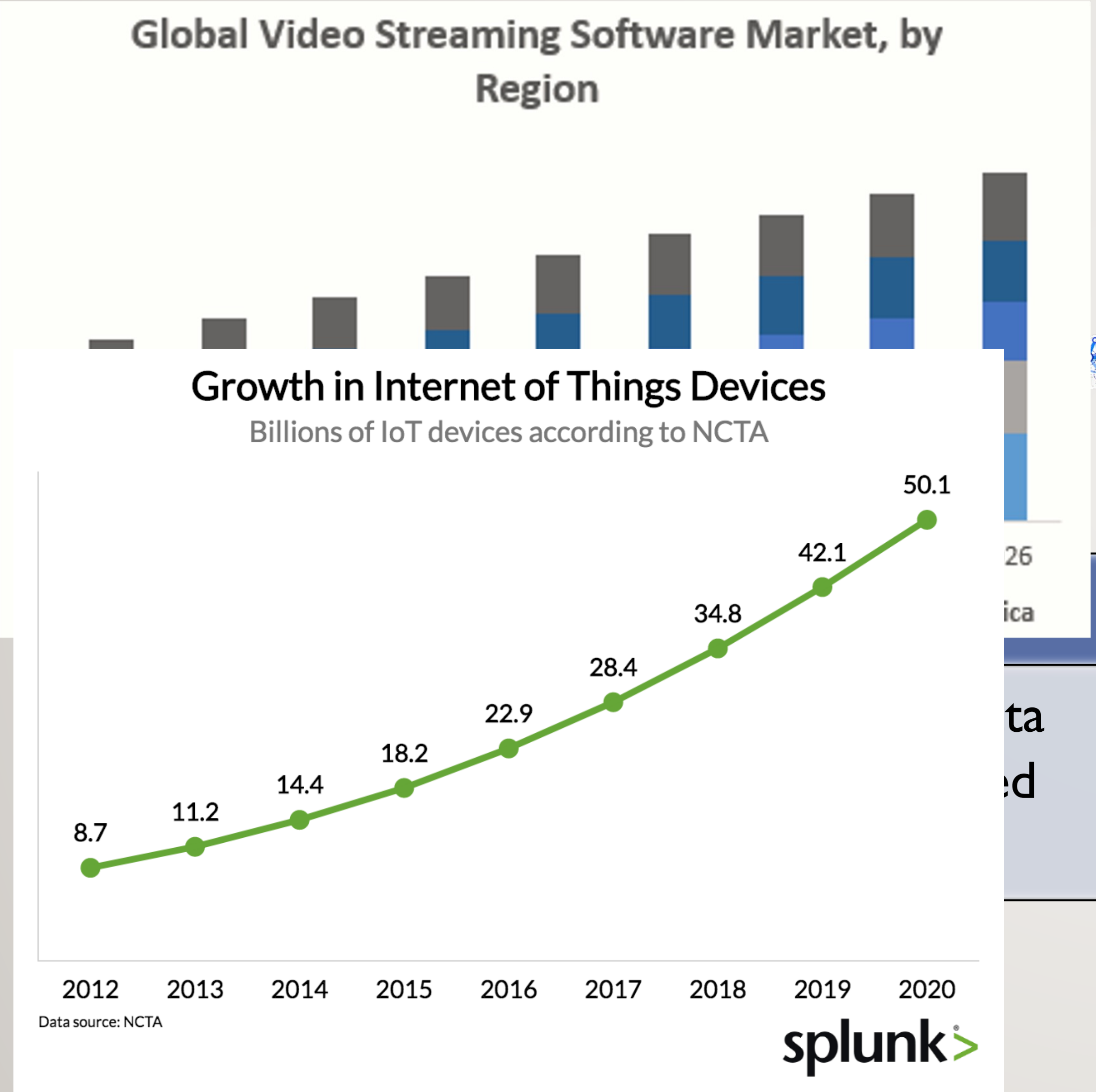
- Scale of data
- Data at rest

- Forms of data
- Unstructured challenges

Velocity

- Streaming data
- Data in motion

Big Data – Four Vs

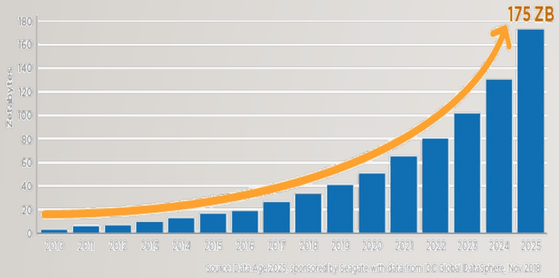


ta
ed

Velocity

- Streaming data
- Data in motion

Big Data – Four Vs



Volume
• Scale of data • Data at rest

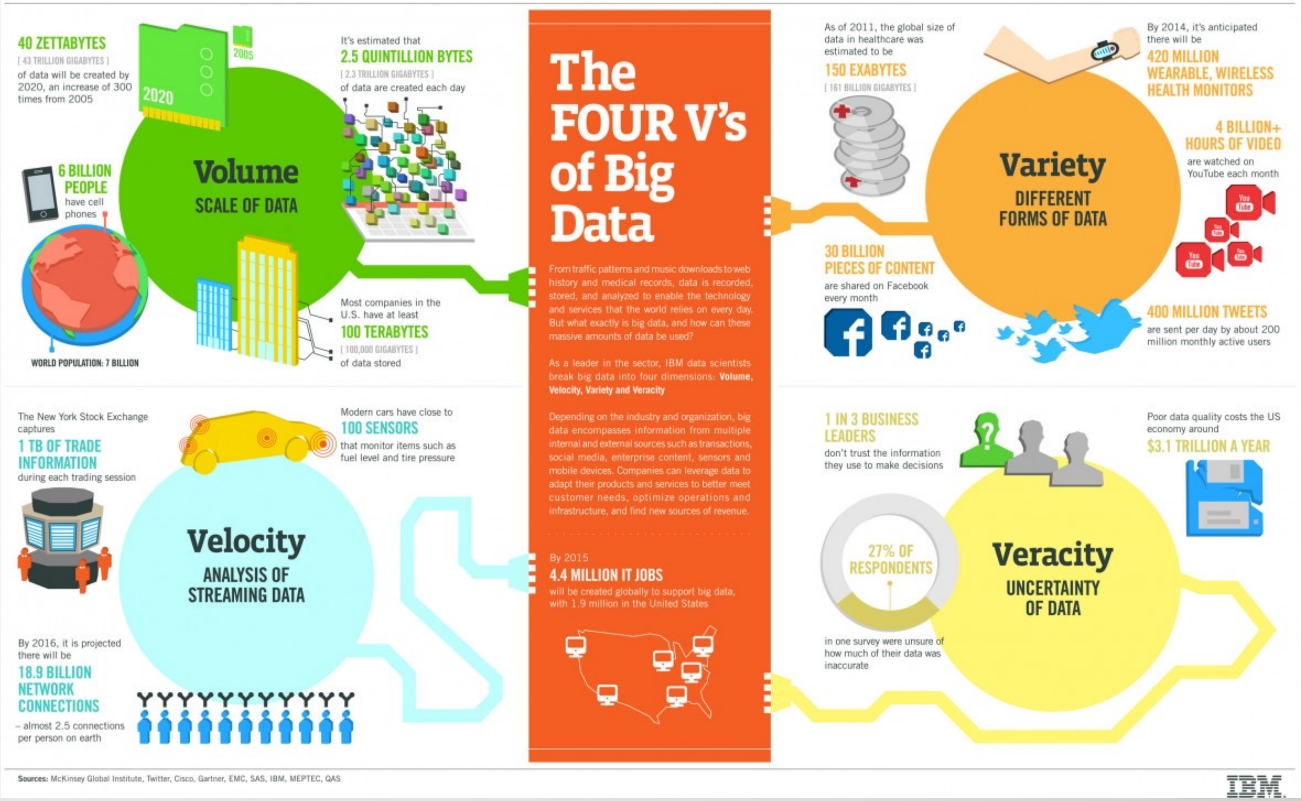
Variety
• Forms of data • Unstructured challenges

Velocity
• Streaming data • Data in motion

Veracity
• Uncertainty/ incorrectness in data • Data quality

Big Data Management

- Can you manage "big data" using relational DBMS?
  - Yes, but problematic
- Characteristics of 4Vs demand more flexibility and more scalability
  - May not have schema
  - Scale-out solutions
  - May not need SQL



NoSQL Database Systems – Dealing with Variety

Any DBMS that is not relational and does not have the relational restrictions

PATIENT

PID: 49875	PName: Jane Smith	InsNo: ON8677	Street: 54 Beachwood St.	City: Waterloo	Province: Ontario
PID: 15948	FName: Ali	LName: Nadir	Street: 583 College St.	City: Toronto	Province: Ontario
PID: 98143	PName: Jiang Ni	Address: 583 College St., Toronto, Ontario			
PID: 75880	FName: Tom	Lname: White	Contact: 〈2269873456, 5199905555〉	Address: 189-95 Ave., Edmonton, Alberta	

TREATMENT

PID: 49875	TBegin: 25-Jan-2017	TEnd: 31-Dec-2020	Cost: 11500	
PID: 15948	TBegin: 20-Apr-2017	Doctor: 31-Dec-2020		
PID: 98143	TBegin: 1-Sept-2020	TEnd: 5-Mar-2022	Invoice$: 25000	Insurance: Manulife

NoSQL Database Systems – Dealing with Variety

Any DBMS that is not relational and does not have the relational restrictions

PATIENT

PID: 49875	PName: Jane Smith	InsNo: ON8677	Street: 54 Beachwood St.	City: Waterloo	Province: Ontario
PID: 15948	FName: Ali	LName: Nadir	Street: 583 College St.	City: Toronto	Province: Ontario
PID: 98143	PName: Jiang Ni	Address: 583 College St., Toronto, Ontario			
PID: 75880	FName: Tom	Lname: White	Contact: 〈2269873456, 5199905555〉	Address: 189-95 Ave., Edmonton, Alberta	

Varying structure

TREATMENT

PID: 49875	TBegin: 25-Jan-2017	TEnd: 31-Dec-2020	Cost: 11500	
PID: 15948	TBegin: 20-Apr-2017	Doctor: 31-Dec-2020		
PID: 98143	TBegin: 1-Sept-2020	TEnd: 5-Mar-2022	Invoice$: 25000	Insurance: Manulife

Varying structure

NoSQL Database Systems – Dealing with Variety

Any DBMS that is not relational and does not have the relational restrictions

PATIENT

PID: 49875	PName: Jane Smith	InsNo: ON8677	Str 54	: terloo	Province: Ontario
PID: 15948	FName: Ali	LName: Nadir	Street: 583 College St.	City: Toronto	Province: Ontario
PID: 98143	PName: Jiang Ni	Address: 583 College St., Toronto, Ontario			
PID: 75880	FName: Tom	Lname: White	Contact: 〈2269873456, 5199905555〉	Address: 189-95 Ave., Edmonton, Alberta	

No connection

TREATMENT

PID: 49875	TBegin: 25-Jan-2017	TEnd: 31-Dec-2020	Cost: 11500	
PID: 15948	TBegin: 20-Apr-2017	Doctor: 31-Dec-2020		
PID: 98143	TBegin: 1-Sept-2020	TEnd: 5-Mar-2022	Invoice$: 25000	Insurance: Manulife

Varying structure

Varying structure

Relational vs NoSQL Systems

Relational

- Strict schema
- Strict consistency
- High-level query language
  - Complex query support
- Scalability possible, but limited

NoSQL

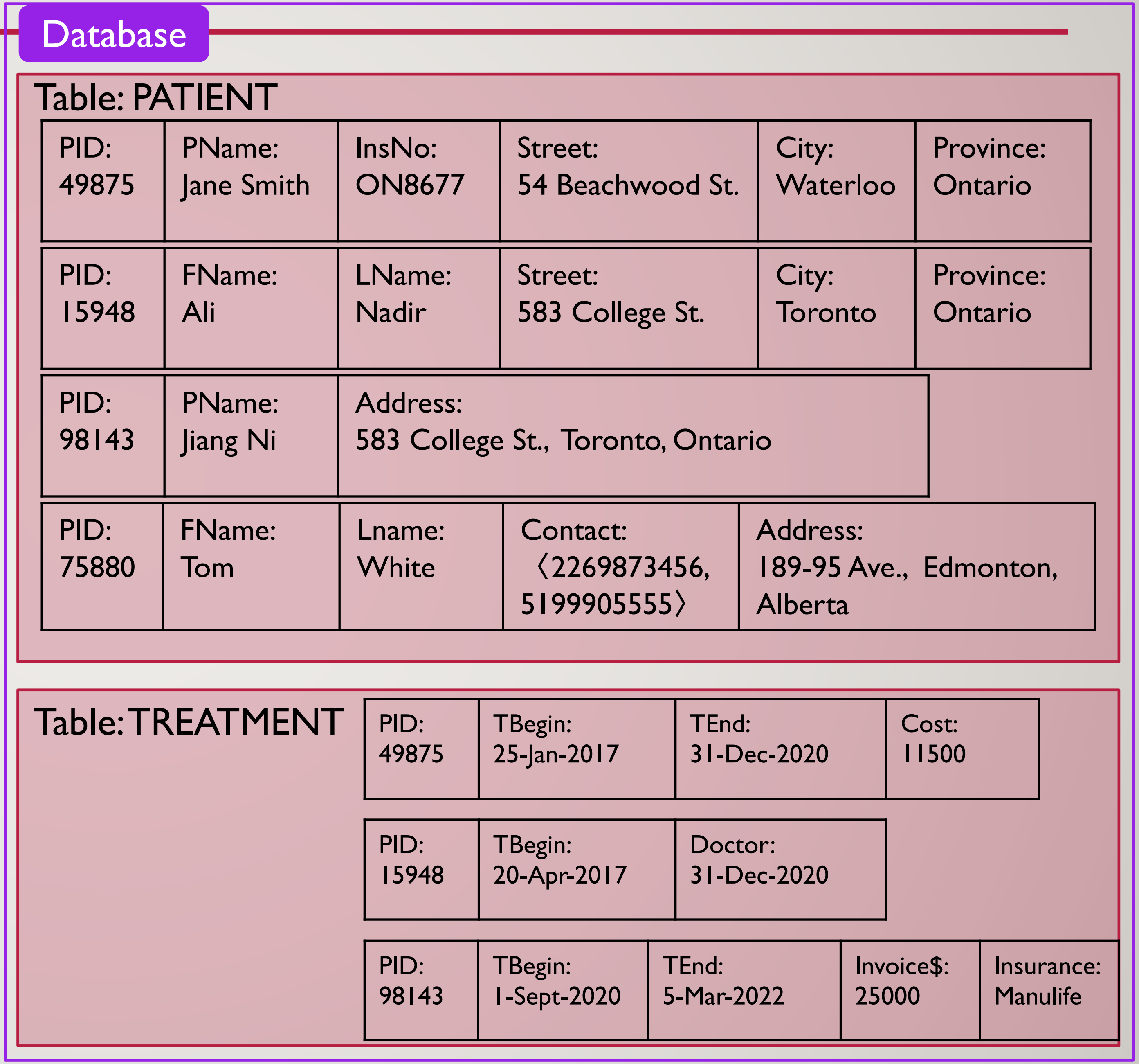
- Flexible schema
- Some inconsistency is OK
  - Eventual consistency
- Highly scalable
- Fast access
- No complex query support
  - Give a key, retrieve the value

NoSQL Types

- Key-value
- Wide-column (column-oriented; big table)
- Document
- Graph
- Multimodel

Key-Value Stores

- Simple (key, value) data model
  - Key = unique id
  - Value = a text, a binary data, structured data, etc.
- Simple queries
  - put(key, value)
    - Inserts a (key, value) pair
  - value = get(key)
    - Returns the value associated with key
  - {(key, value)} = get_range(key1, key2)
    - Returns the data whose key is in interval [key1, key2]

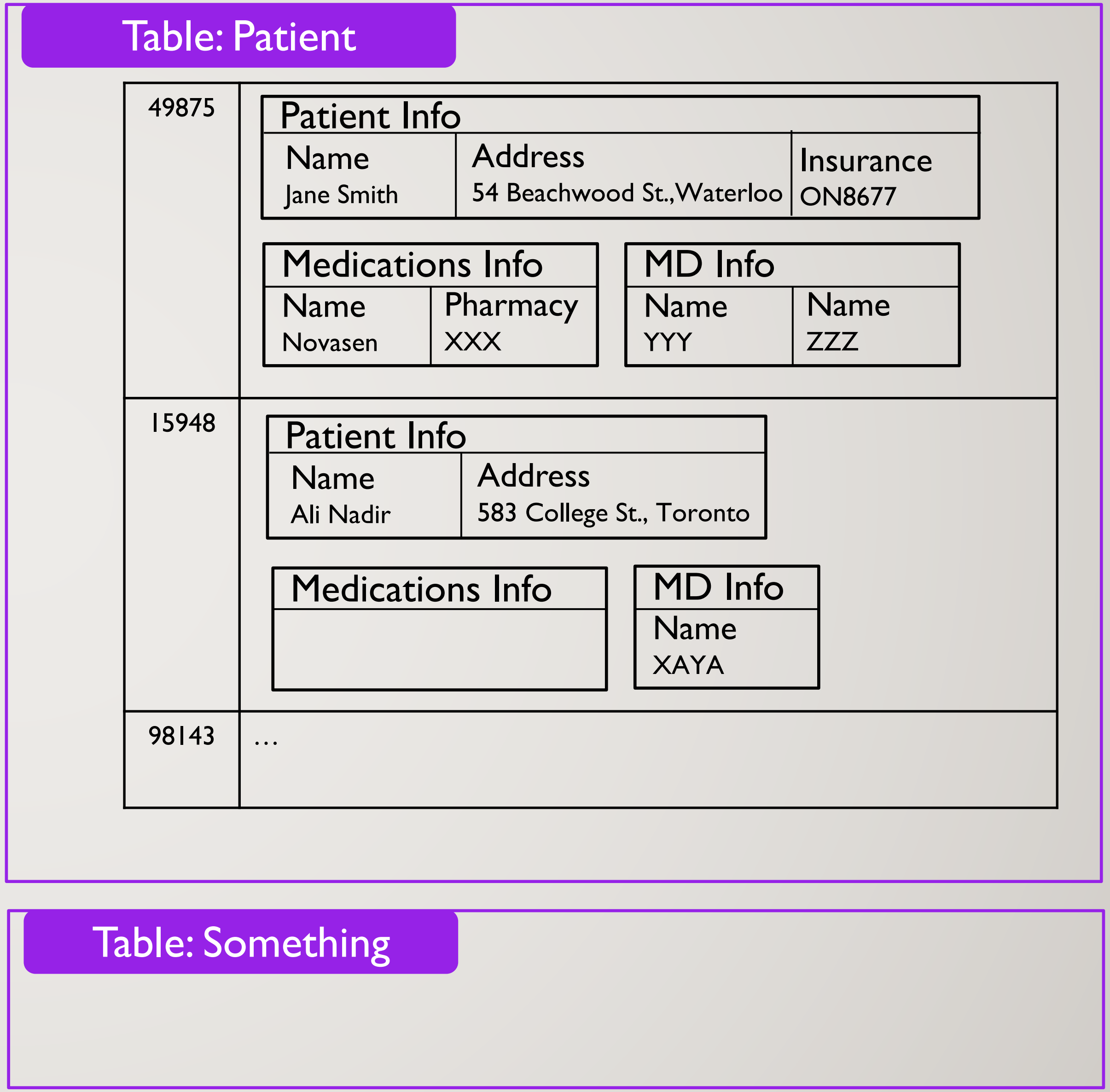


Key-Value Stores

- Simple (key, value) data model
  - Key = unique id
  - Value = a text, a binary data, structured data, etc.
- Simple queries
  - put(key, value)
    - Inserts a (key, value) pair
  - value = get(key)
    - Returns the value associated with key
  - {(key, value)} = get_range(key1, key2)
    - Returns the data whose key is in interval [key1, key2]

WIDE-COLUMN SYSTEMS

- Reverse relational maxim of "each fact in its own table"
  - "everything about an entity are together"
- Loose schema
  - Define *column family*
  - Column families exist in each row, but not columns
- Operators for creation, insertion, retrieval, and deletion

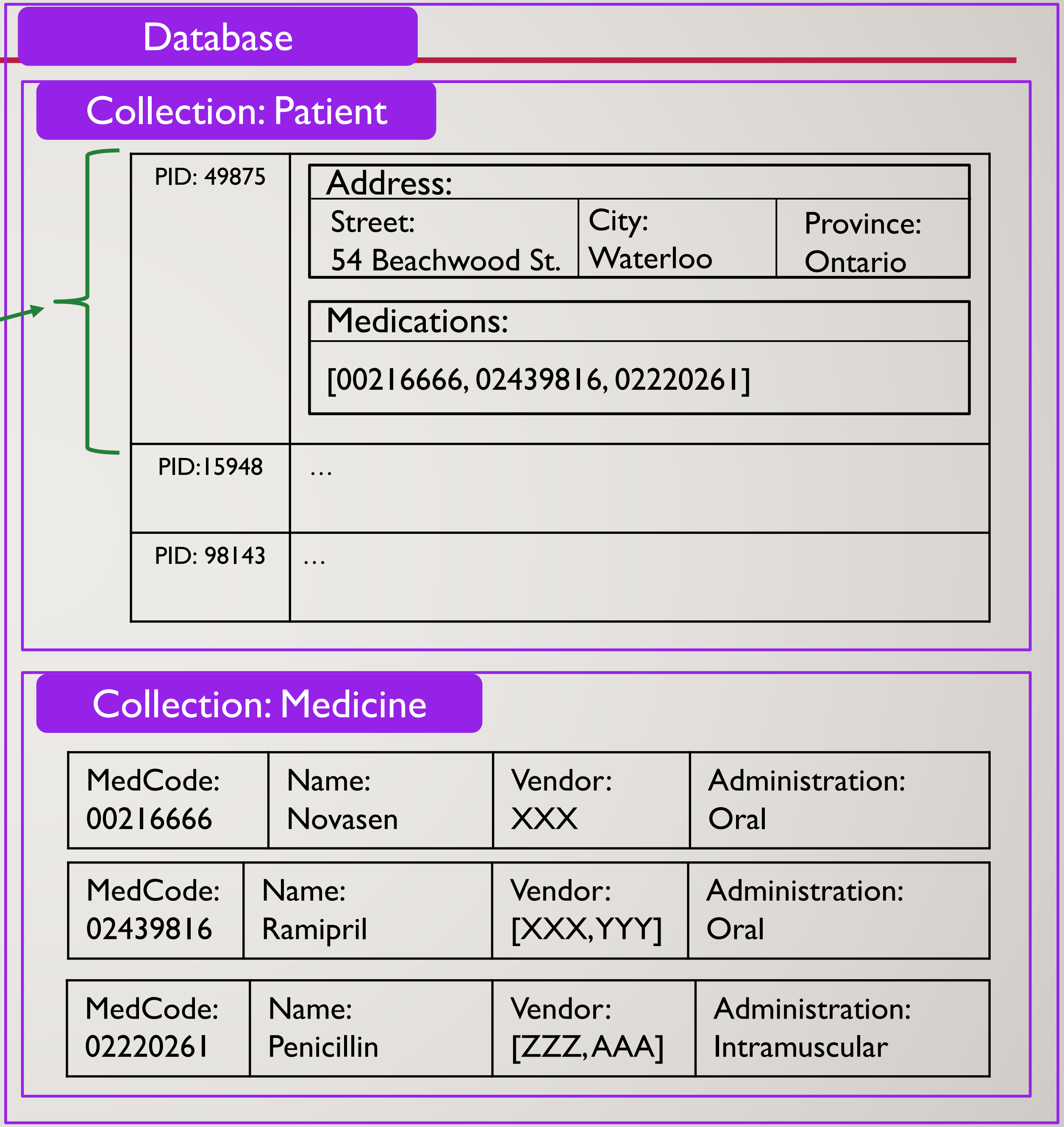


Wide-Column Systems

- Reverse relational maxim of "each fact in its own table"
  - "everything about an entity are together"
- Loose schema
  - Define *column family*
  - Column families exist in each row, but not columns
- Operators for creation, insertion, retrieval, and deletion

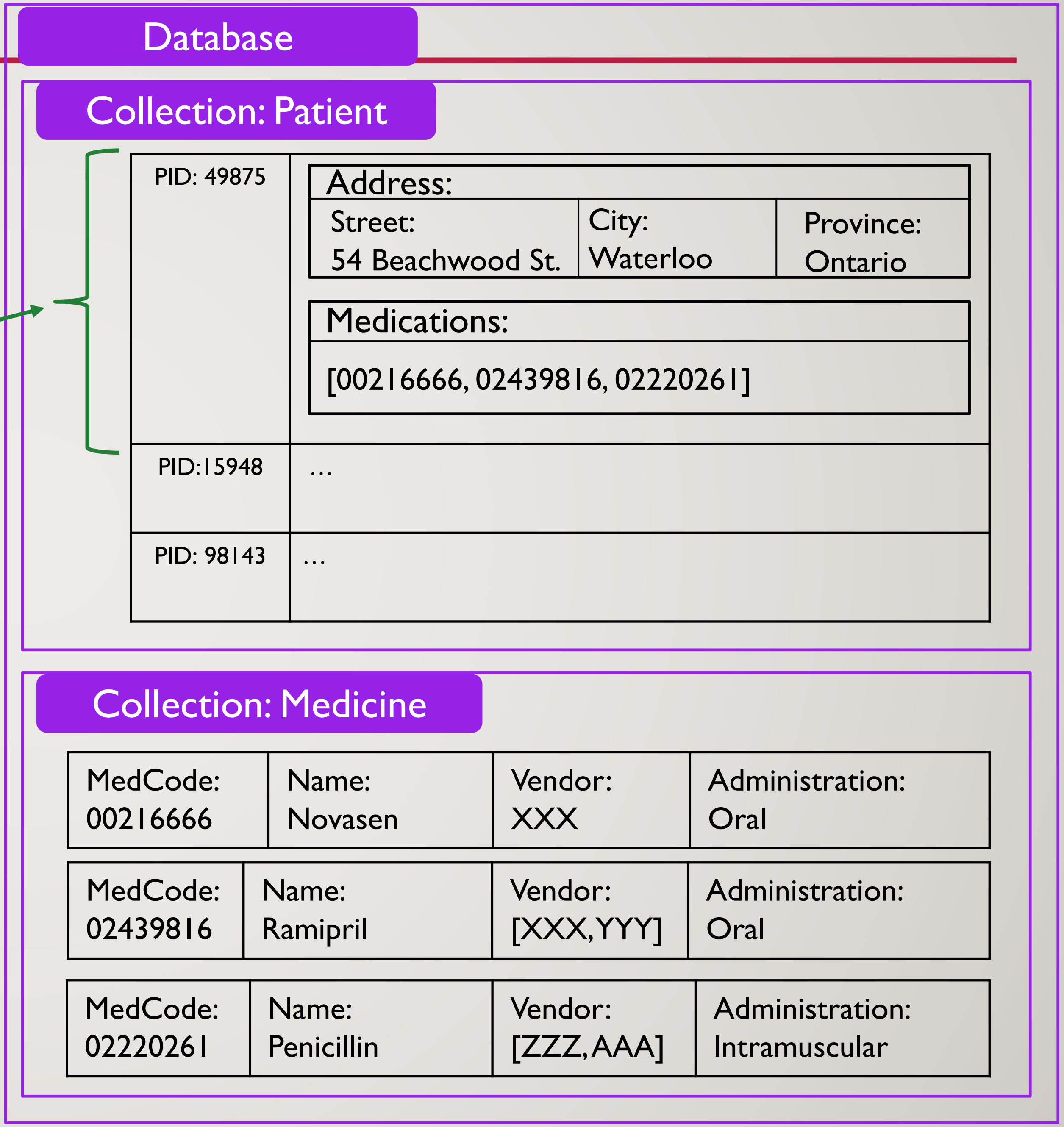
Document Stores

- Documents
  - Hierarchical structure, with nesting of elements
  - Value is a JSON object



Document Stores

- Documents
  - Hierarchical structure, with nesting of elements
  - Value is a JSON object
- Simple queries
  - db.Medicine.find({Vendor:"XXX"})
    - Find all medicines supplied by XXX
  - db.Patient.find({Address.Province:"Ontario"})
    - Find the records of patients who live in Ontario



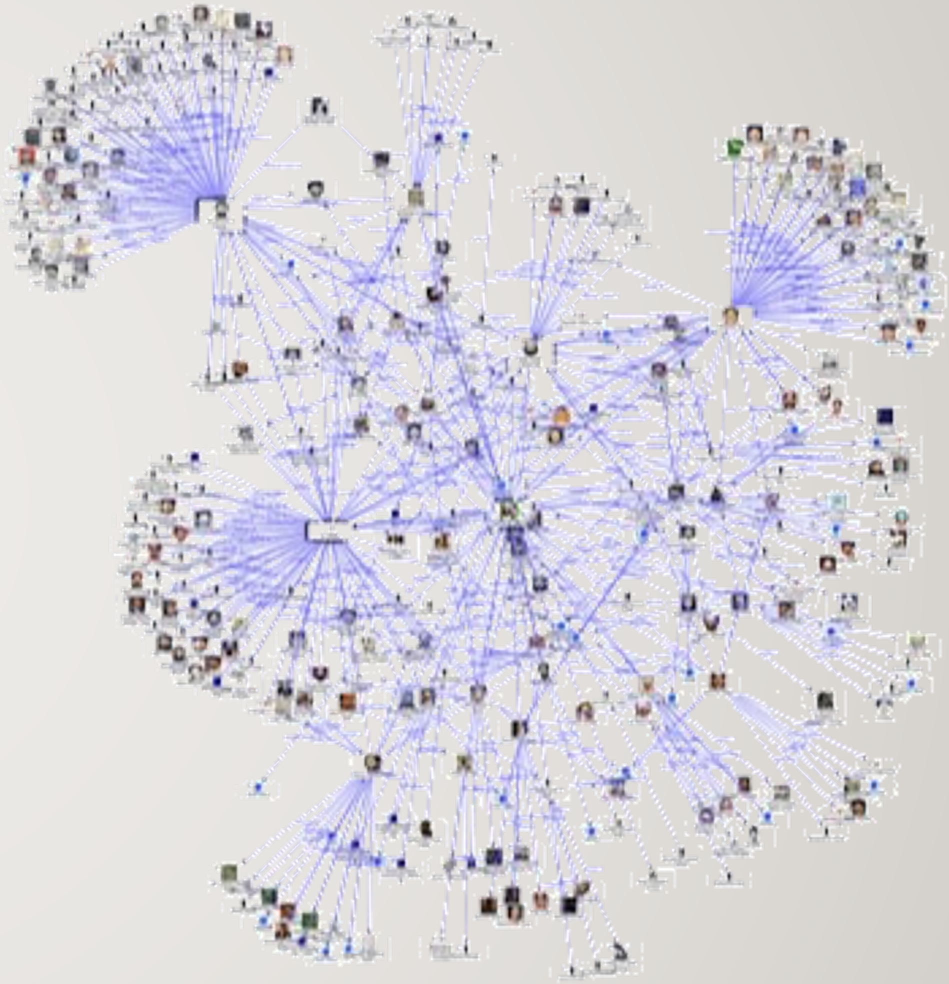
Document Stores

- Documents
  - Hierarchical structure, with nesting of elements
  - Value is a JSON object
- Simple queries
  - db.Medicine.find({Vendor:"XXX"})
    - Find all medicines supplied by XXX
  - db.Patient.find({Address.Province:"Ontario"})
    - Find the records of patients who live in Ontario

GRAPH DATABASES

- When *relationships* are important
- Model:
  - *Vertices* represent entities
  - *Edges* represent relationships
- Consider Facebook graph and find friends of my friends

```
MATCH pMutualFriends=(me { name: 'Tamer' })-
[:FRIEND*2..2]->(foaf)
WHERE NOT (me)-[:FRIEND]-(foaf) AND NOT me=foaf
RETURN foaf.name
```



Social network

Graph Databases

- When *relationships* are important
- Model:
  - *Vertices* represent entities
  - *Edges* represent relationships
- Consider Facebook graph and find friends of my friends

```
MATCH pMutualFriends=(me { name: ‘Tamer' })-
[:FRIEND*2..2]->(foaf)
WHERE NOT (me)-[:FRIEND]-(foaf) AND NOT me=foaf
RETURN foaf.name
```

neo4j Amazon Neptune OrientDB JanusGraph TigerGraph

Scale-Out Solutions – Dealing with Volume

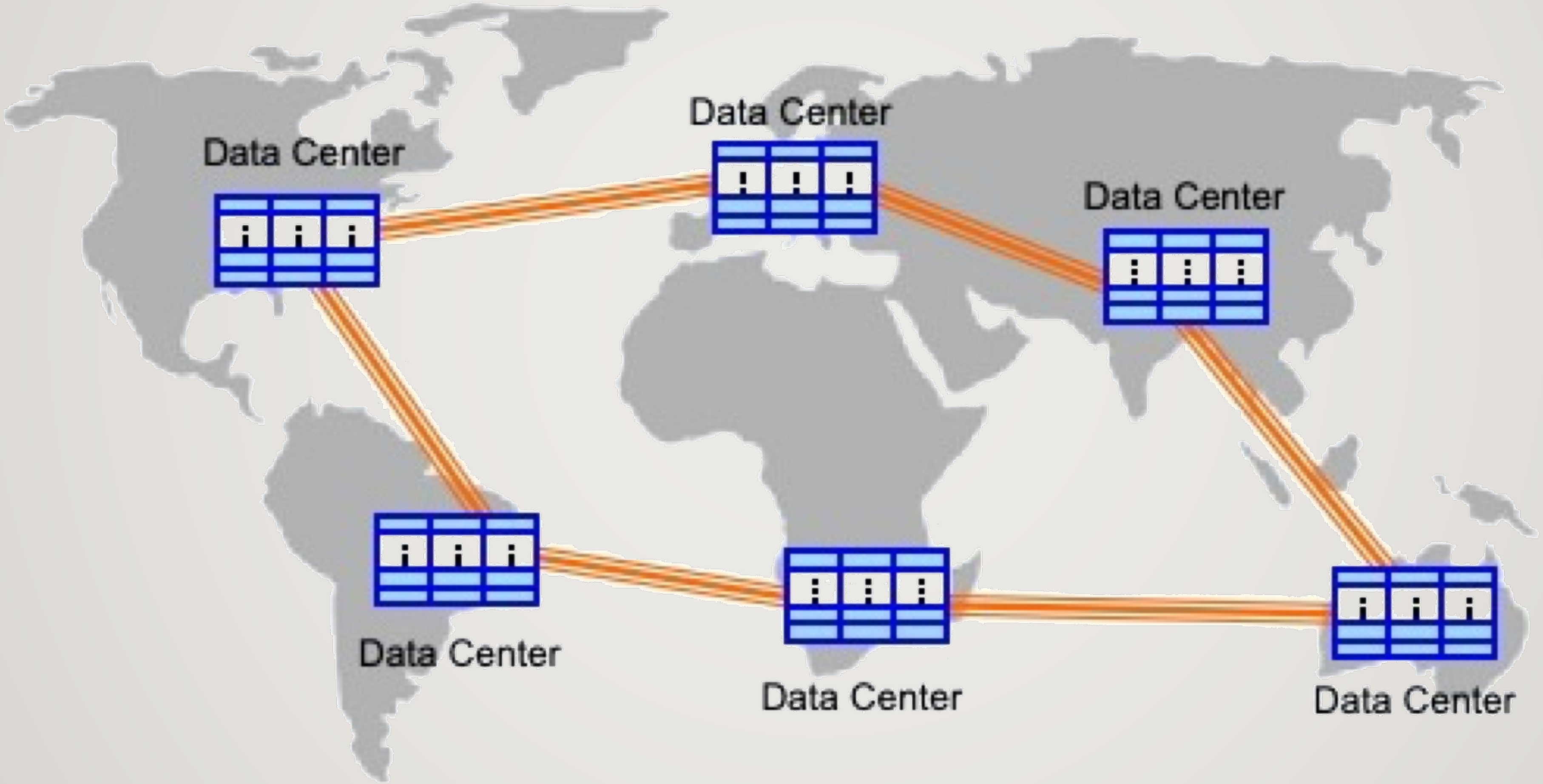
Data may be too big to fit in one machine

Computation may be too heavy to be done by one machine

↓

Employ a number of machines, distribute data and distribute computation

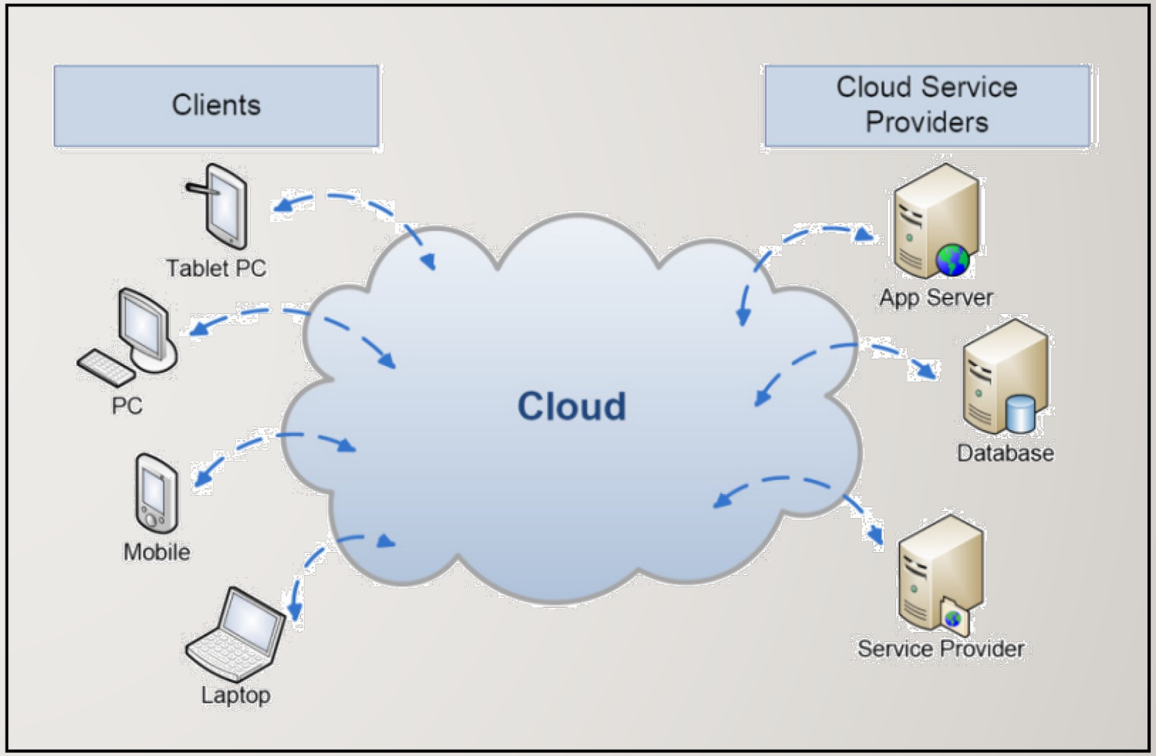
Geographically Distributed Data Centres



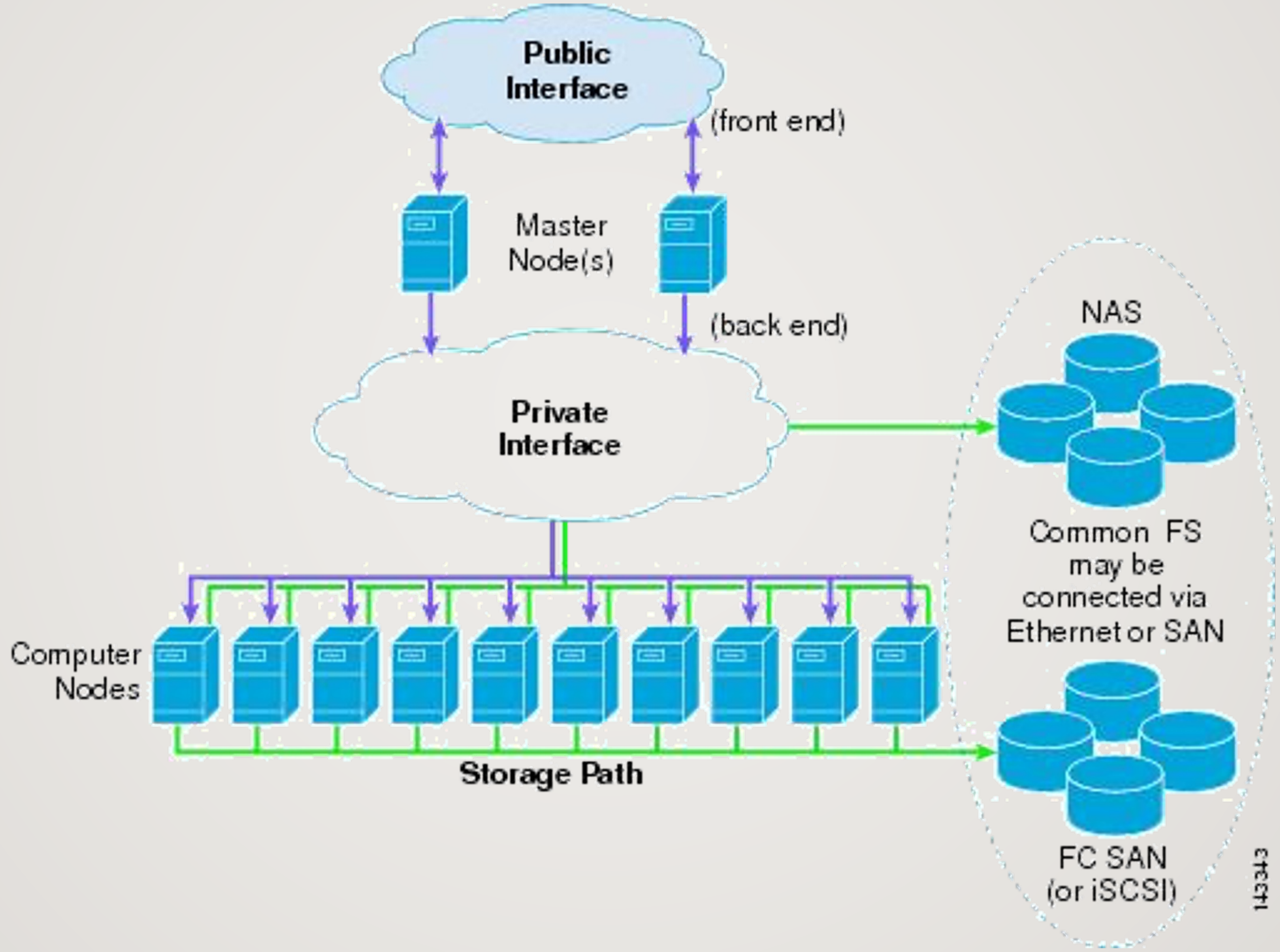
Cloud Computing

On-demand, reliable services provided over the Internet in a cost-efficient manner

- Cost savings: no need to maintain dedicated compute power
- Elasticity: better adaptivity to changing workload
- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)



Inside a Data Centre – Parallel DBMS



Source: Cisco Data Center Infrastructure Design Guide, 2011

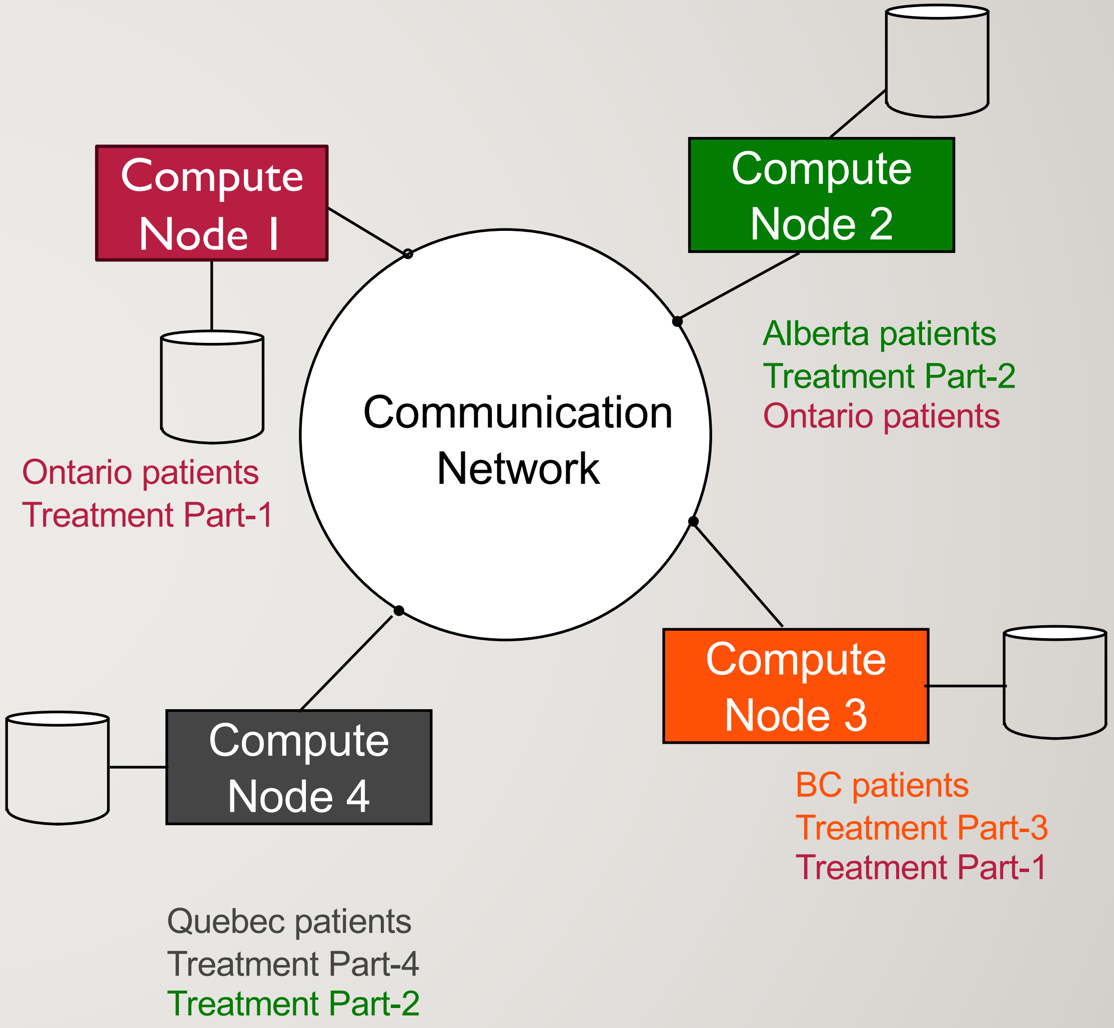
Data Partitioning & Querying

PATIENT

PID	PName	InsNo	Street	City	Province
49875	Jane Smith	ON8677	54 Beachwood St.	Waterloo	Ontario
15948	Ali Nadir	ON7740	583 College St.	Toronto	Ontario
98143	Jiang Ni	AB39658	189-95 Ave.	Edmonton	Alberta
75880	Tom White	ON6409	884 Water St.	Burlington	Ontario
13086	Mark Smith	ON7843	54 King St.	Vancouver	BC

TREATMENT

PID	Begin	End	EID	Cost
49875	25-Jan-2017	31-Dec-2020	34757200	11500
15948	20-Apr-2017	31-Dec-2020	85993920	37300
49875	1-Sept-2020	5-Mar-2022	34757200	25000



Data Partitioning & Querying

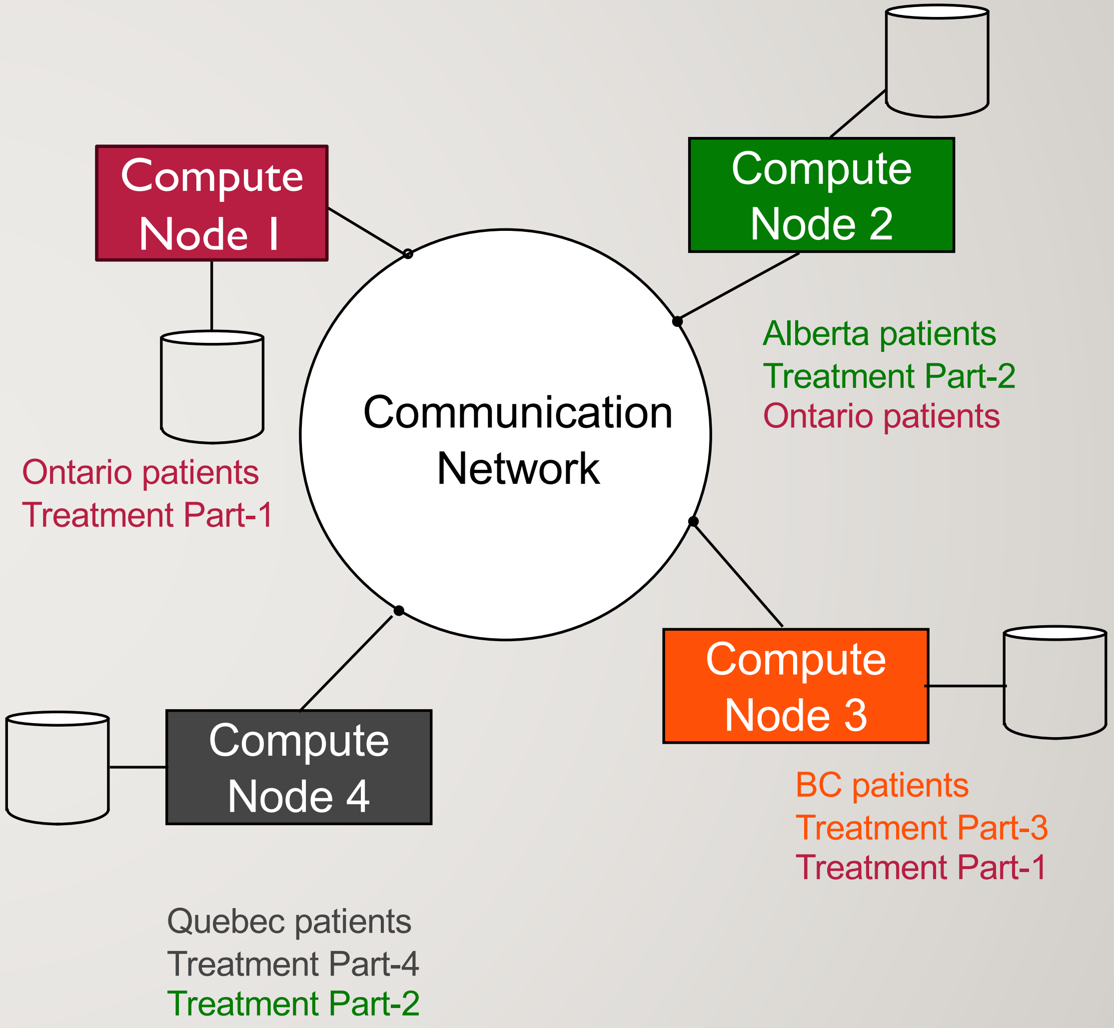
PATIENT

PID	PName	InsNo	Street	City	Province
49875	Jane Smith	ON8677	54 Beachwood St.	Waterloo	Ontario
15948	Ali Nadir	ON7740	583 College St.	Toronto	Ontario
98143	Jiang Ni	AB39658	189-95 Ave.	Edmonton	Alberta
75880	Tom White	ON6409	884 Water St.	Burlington	Ontario
13086	Mark Smith	ON7843	54 King St.	Vancouver	BC

TREATMENT

PID	Begin	End	EID	Cost
49875	25-Jan-2017	31-Dec-2020	34757200	11500
15948	20-Apr-2017	31-Dec-2020	85993920	37300
49875	1-Sept-2020	5-Mar-2022	34757200	25000

```
SELECT  P.PName, P.City, T.Cost
FROM    PATIENT P, TREATMENT T
WHERE   Cost > 15000
AND     P.PID = T.PID;
```



Big Data Scalable Execution – MapReduce

Data-parallel execution → All compute nodes do the same thing on some small portion of data

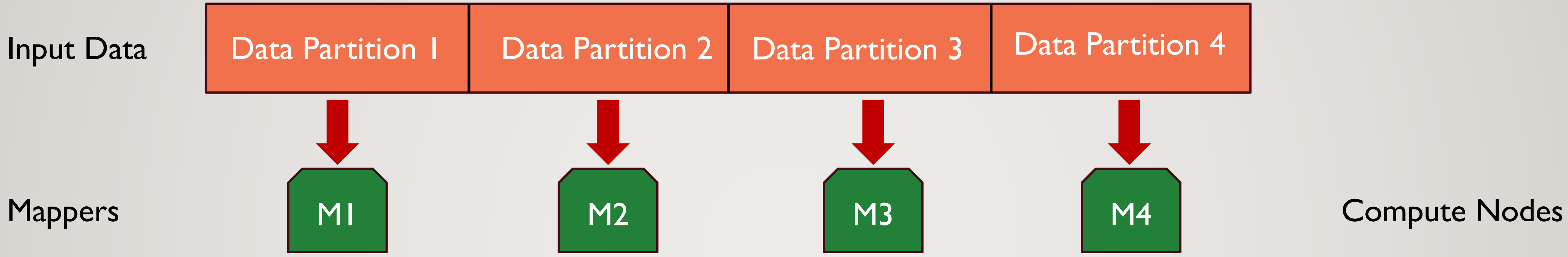
Big Data Scalable Execution – MapReduce

Data-parallel execution → All compute nodes do the same thing on some small portion of data

Input Data	Data Partition 1	Data Partition 2	Data Partition 3	Data Partition 4

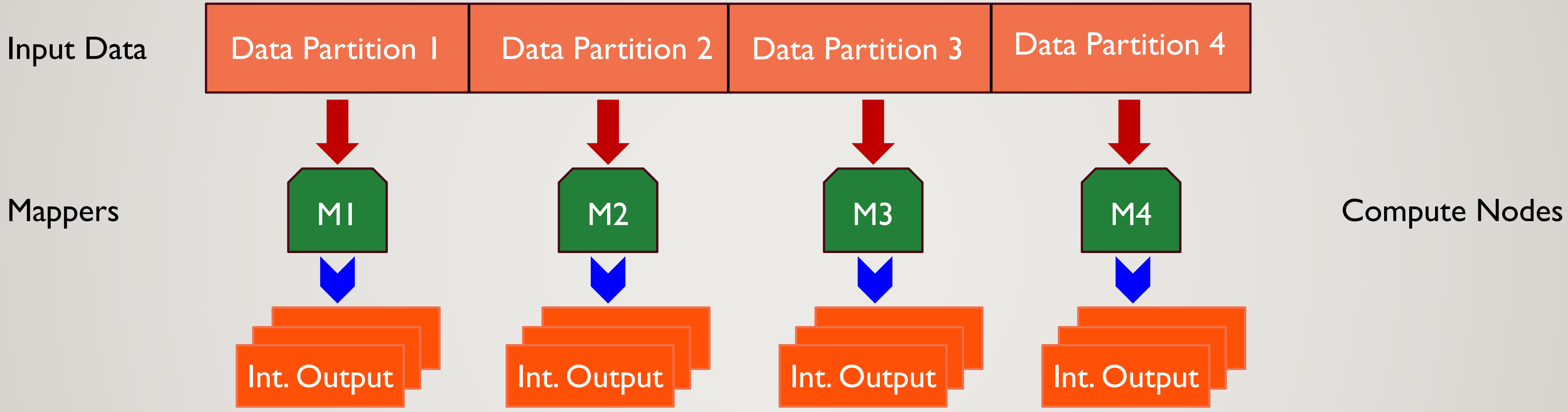
Big Data Scalable Execution – MapReduce

Data-parallel execution → All compute nodes do the same thing on some small portion of data



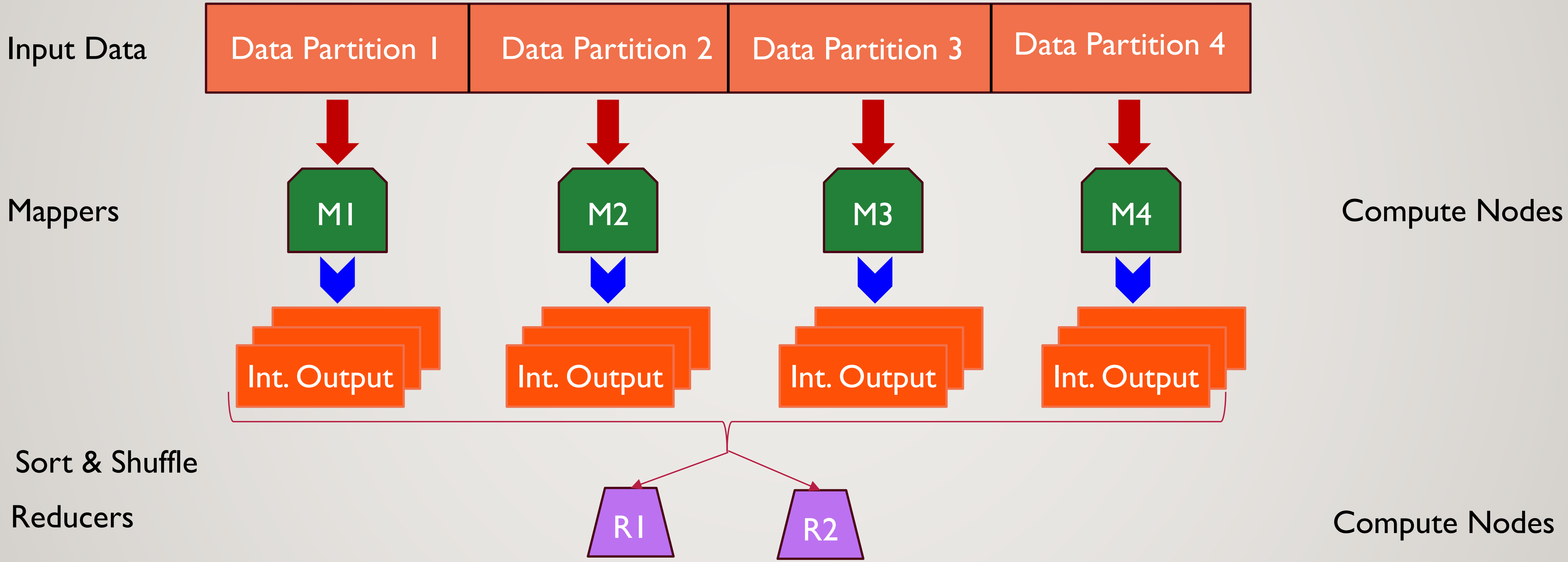
BIG DATA SCALABLE EXECUTION – MAPREDUCE

Data-parallel execution → All compute nodes do the same thing on some small portion of data



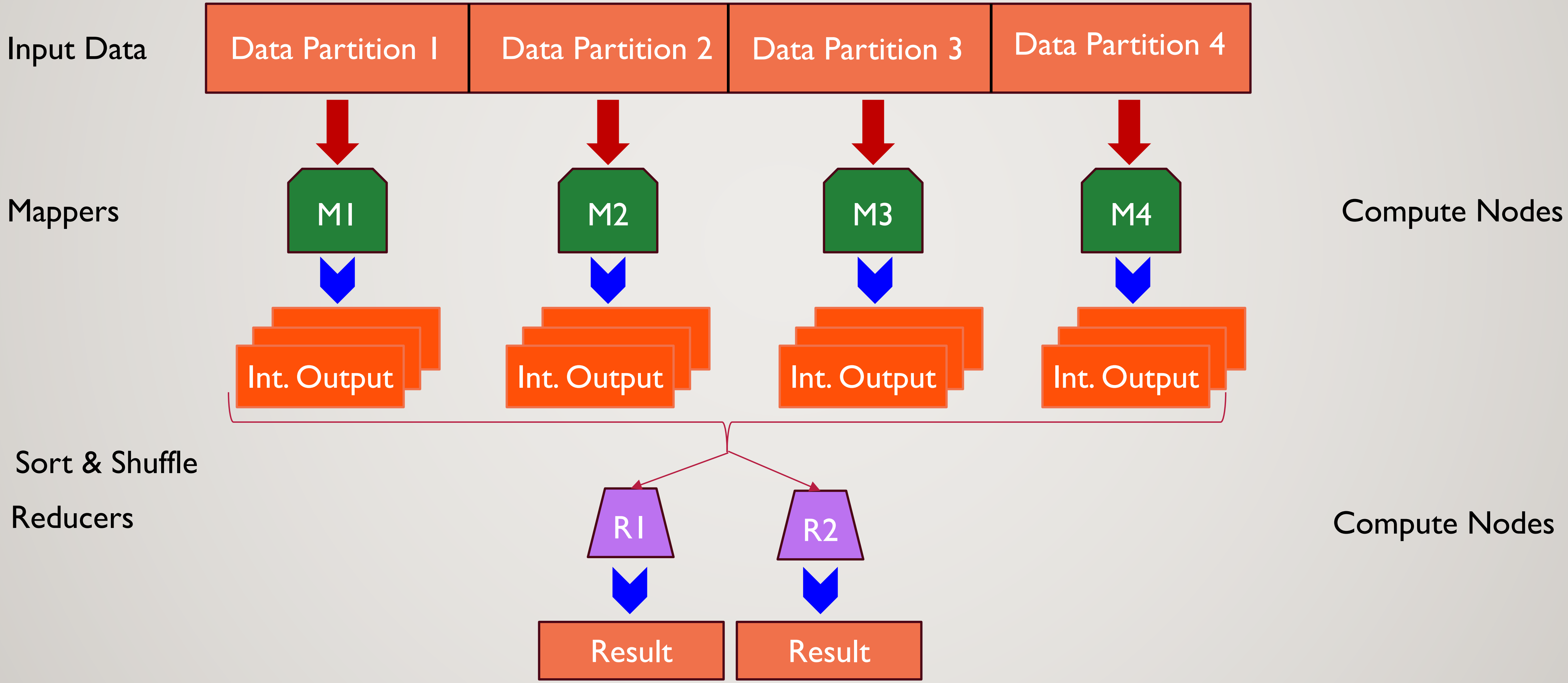
Big Data Scalable Execution – MapReduce

Data-parallel execution → All compute nodes do the same thing on some small portion of data



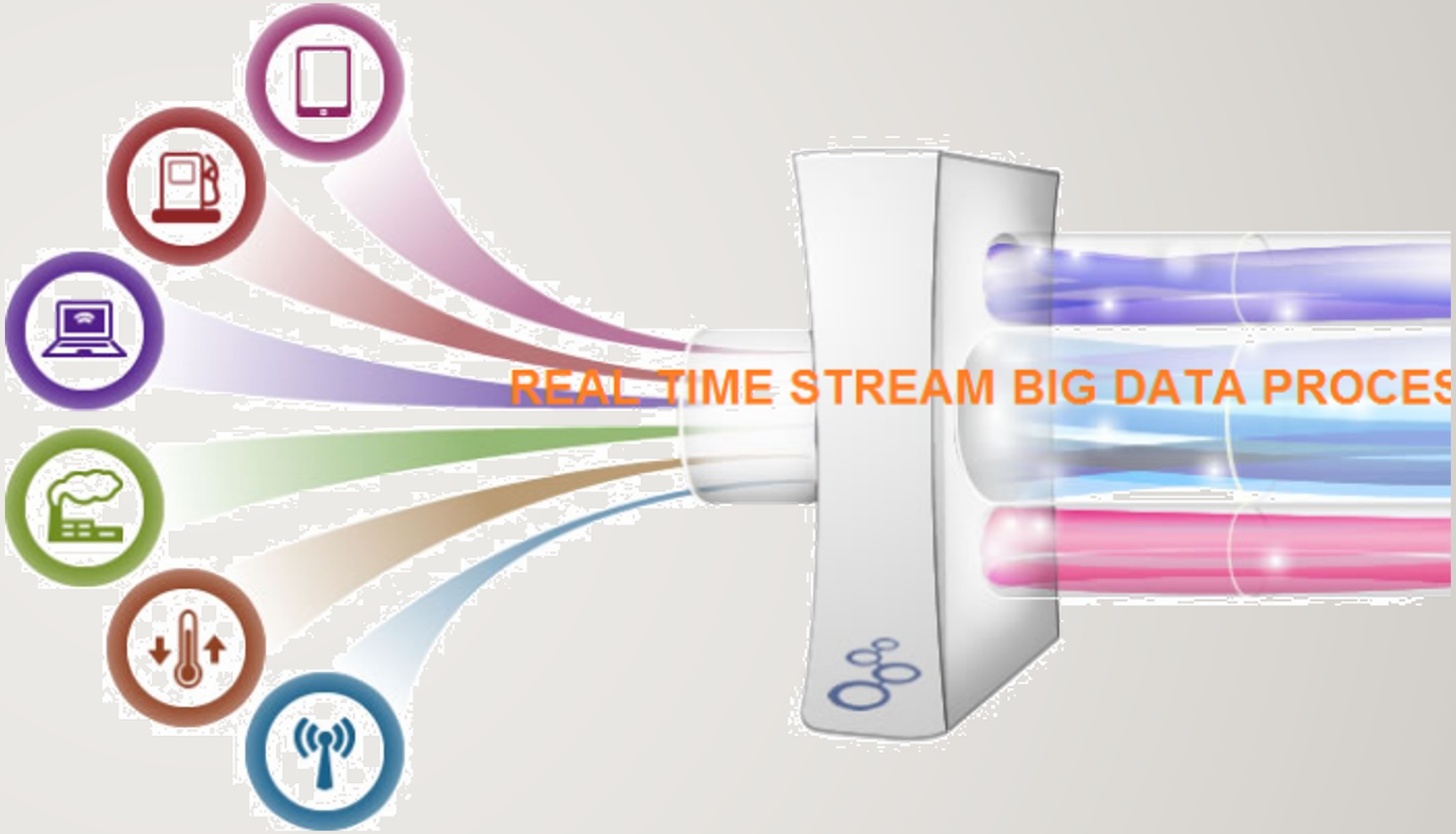
Big Data Scalable Execution – MapReduce

Data-parallel execution → All compute nodes do the same thing on some small portion of data



STREAMING DATA – DEALING WITH VELOCITY

- Data is not static, but “streams” into the system
- Unbounded
- Examples
  - Streaming video/music
  - Sensor data
  - Financial ticker data

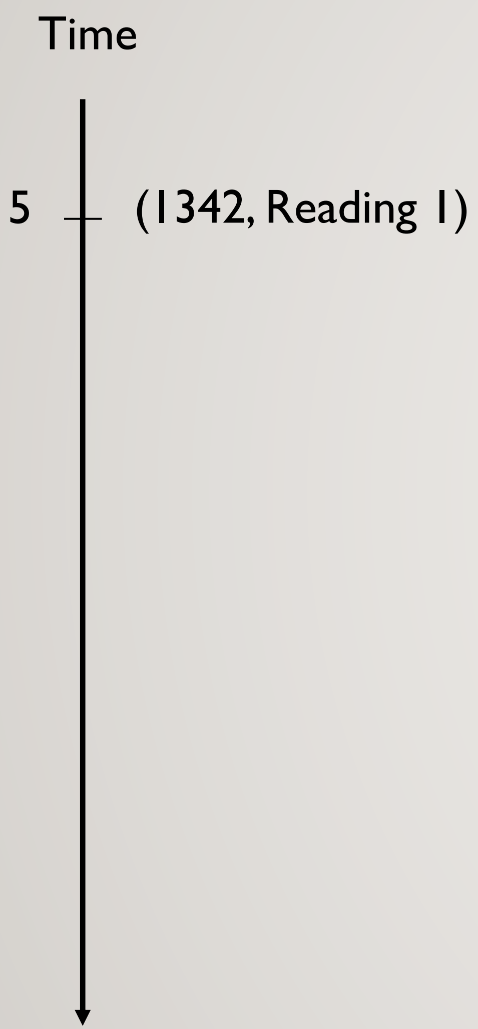


How Does Data Stream

Time

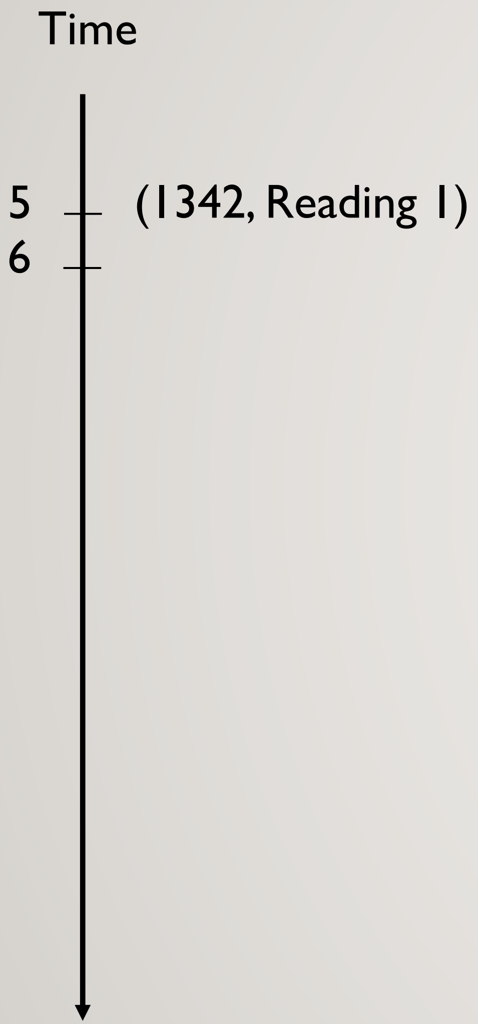
PID	Reading	Timestamp

How Does Data Stream



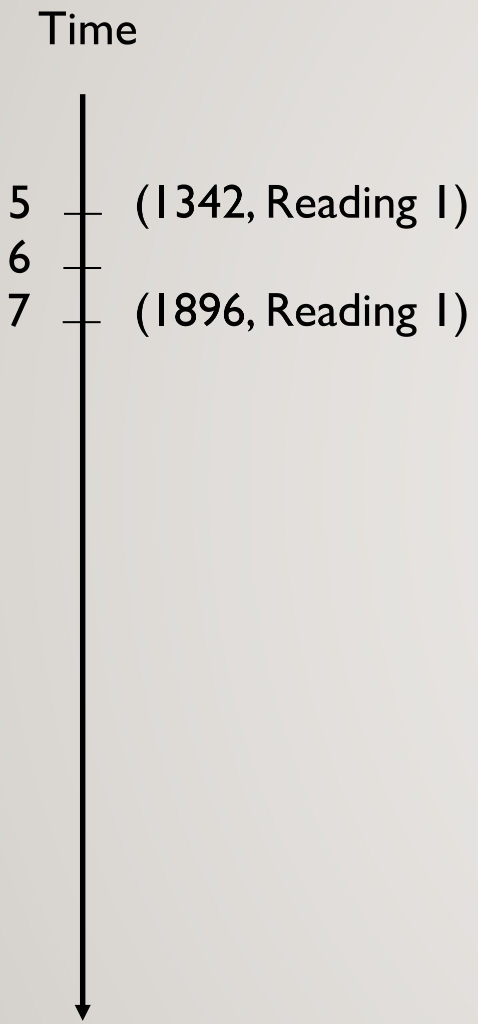
PID	Reading	Timestamp
1342	Reading 1	5

How Does Data Stream



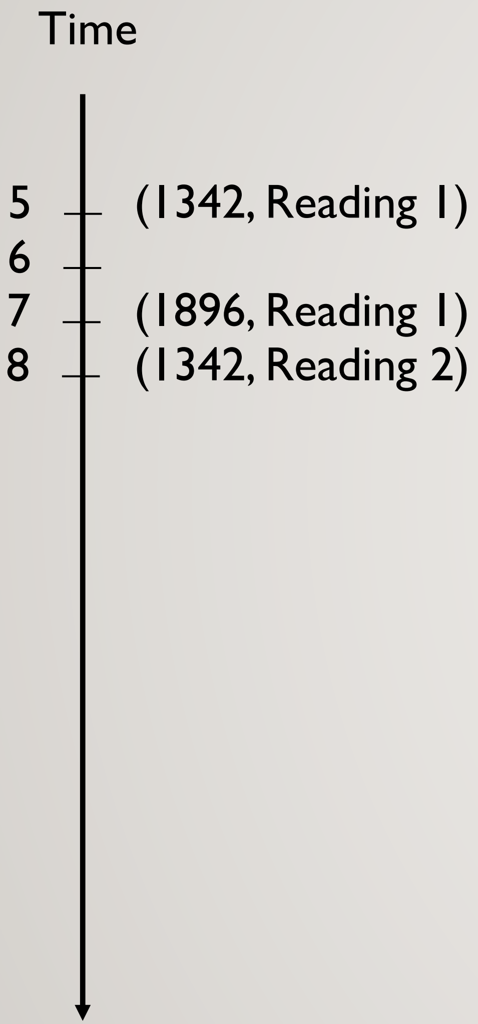
PID	Reading	Timestamp
1342	Reading 1	5

How Does Data Stream



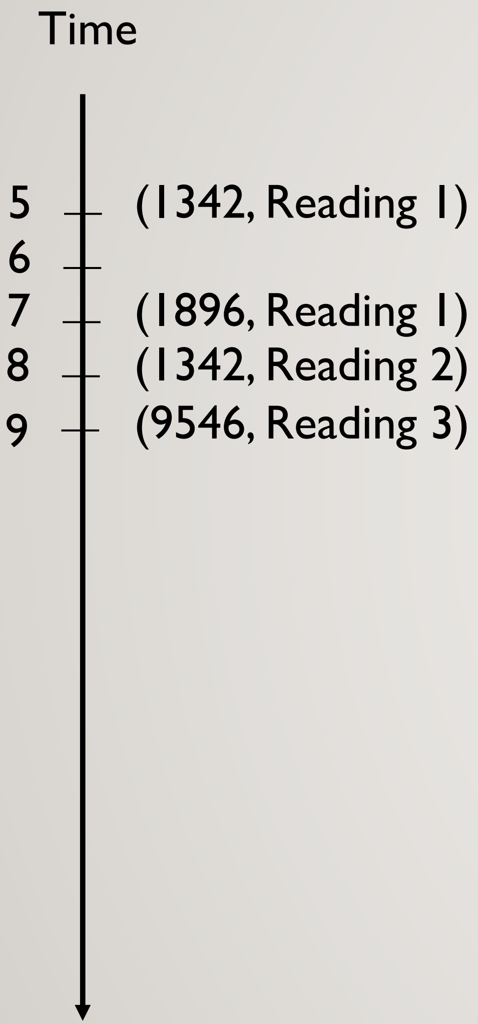
PID	Reading	Timestamp
1342	Reading 1	5
1896	Reading 1	7

How Does Data Stream



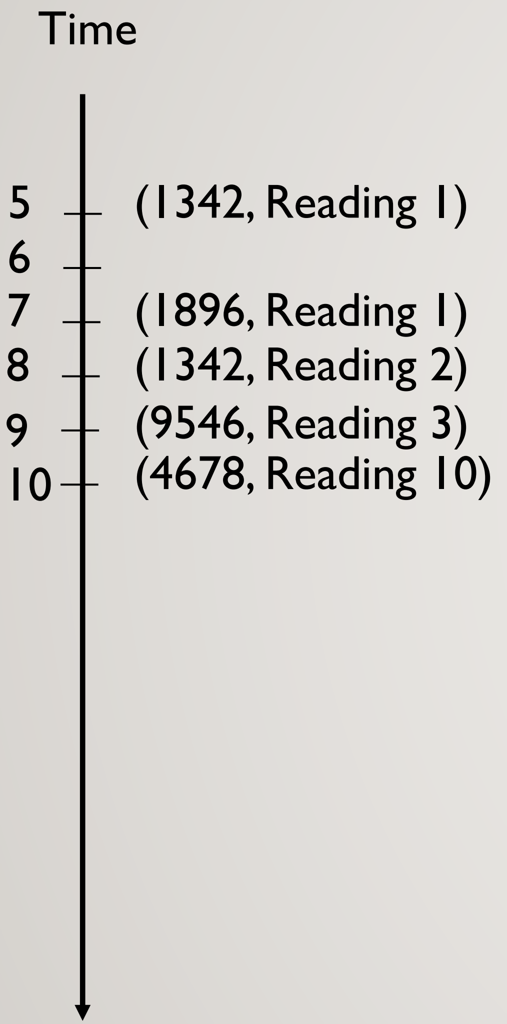
PID	Reading	Timestamp
1342	Reading 1	5
1896	Reading 1	7
1342	Reading 2	8

How Does Data Stream



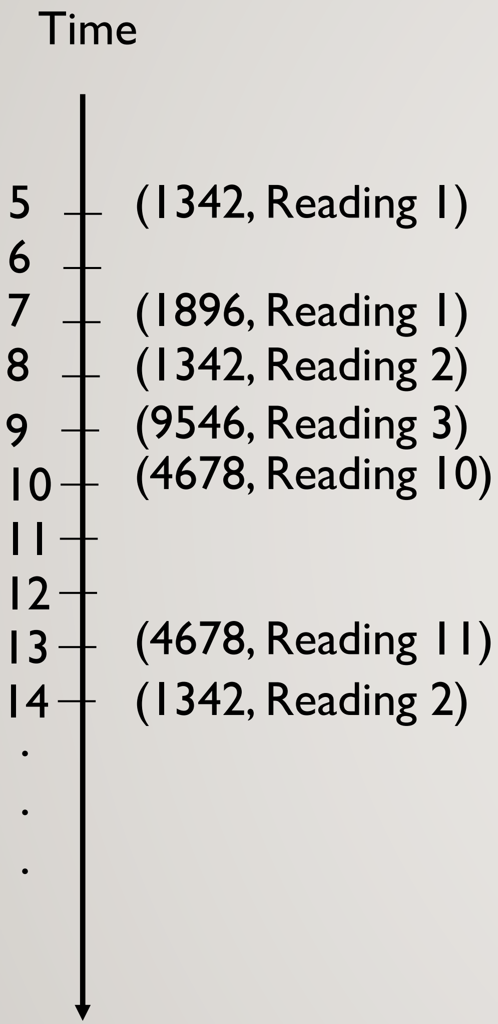
PID	Reading	Timestamp
1342	Reading 1	5
1896	Reading 1	7
1342	Reading 2	8
9546	Reading 3	9

How Does Data Stream



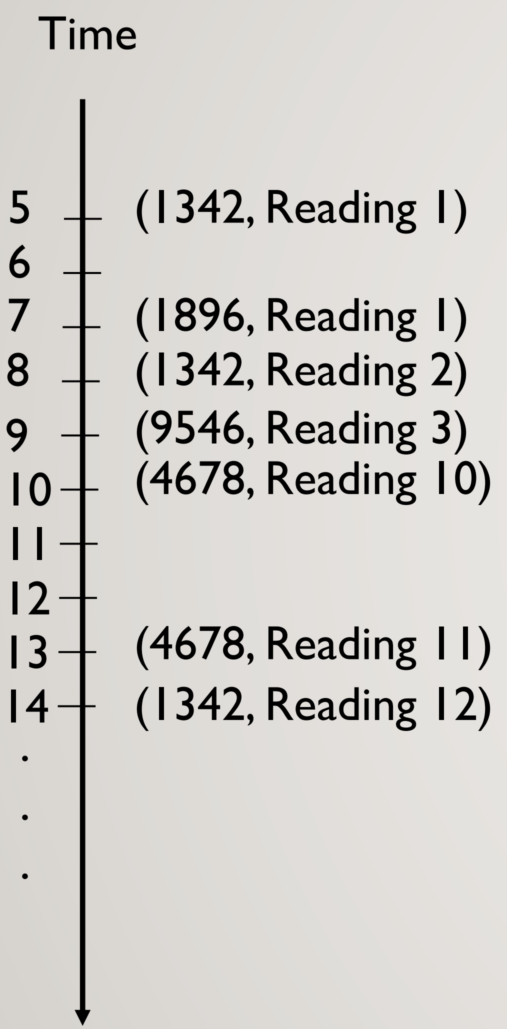
PID	Reading	Timestamp
1342	Reading 1	5
1896	Reading 1	7
1342	Reading 2	8
9546	Reading 3	9
4678	Reading 10	10

How Does Data Stream



PID	Reading	Timestamp
1342	Reading 1	5
1896	Reading 1	7
1342	Reading 2	8
9546	Reading 3	9
4678	Reading 10	10
4678	Reading 11	13
1342	Reading 2	14

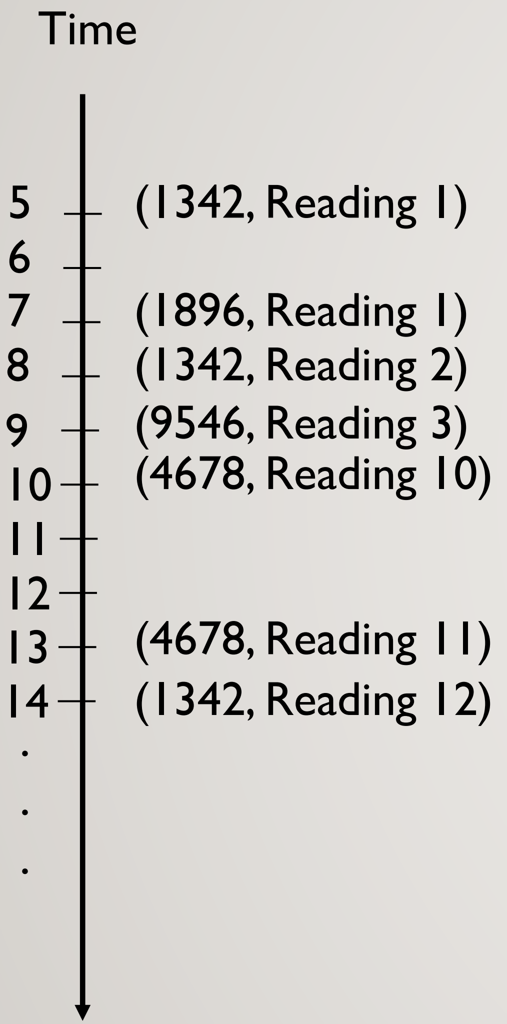
How Does Data Stream



How to process streaming data

PID	Reading	Timestamp
1342	Reading 1	5
1896	Reading 1	7
1342	Reading 2	8
9546	Reading 3	9
4678	Reading 10	10
4678	Reading 11	13
1342	Reading 12	14

How Does Data Stream

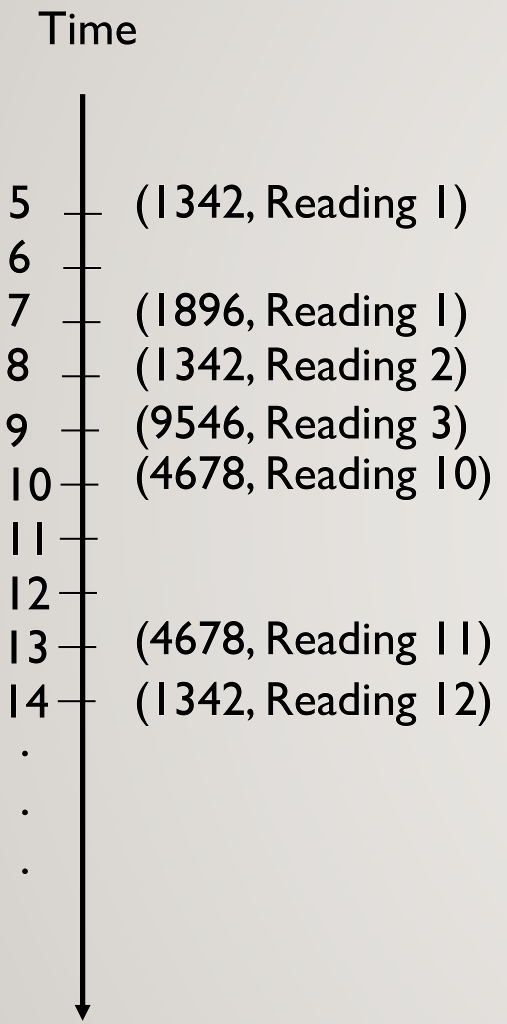


How to process streaming data

- As data arrives
  - Simple operations (e.g., filtering)

PID	Reading	Timestamp
1342	Reading 1	5 ←
1896	Reading 1	7
1342	Reading 2	8
9546	Reading 3	9
4678	Reading 10	10
4678	Reading 11	13
1342	Reading 12	14

How Does Data Stream



How to process streaming data
- As data arrives
  - Simple operations (e.g., filtering)

PID	Reading	Timestamp
1342	Reading 1	5
1896	Reading 1	7
1342	Reading 2	8
9546	Reading 3	9
4678	Reading 10	10
4678	Reading 11	13
1342	Reading 12	14

How Does Data Stream

Time

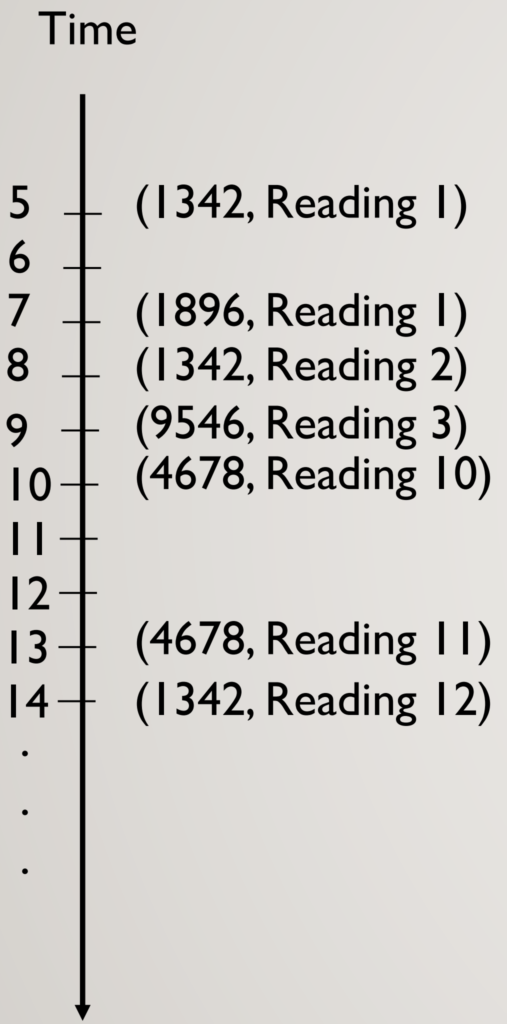
- 5 (1342, Reading 1)
- 6
- 7 (1896, Reading 1)
- 8 (1342, Reading 2)
- 9 (9546, Reading 3)
- 10 (4678, Reading 10)
- 11
- 12
- 13 (4678, Reading 11)
- 14 (1342, Reading 12)
- .
- .
- .

How to process streaming data

- As data arrives
  - Simple operations (e.g., filtering)

PID	Reading	Timestamp
1342	Reading 1	5
1896	Reading 1	7
1342	Reading 2	8 ←
9546	Reading 3	9
4678	Reading 10	10
4678	Reading 11	13
1342	Reading 12	14

How Does Data Stream

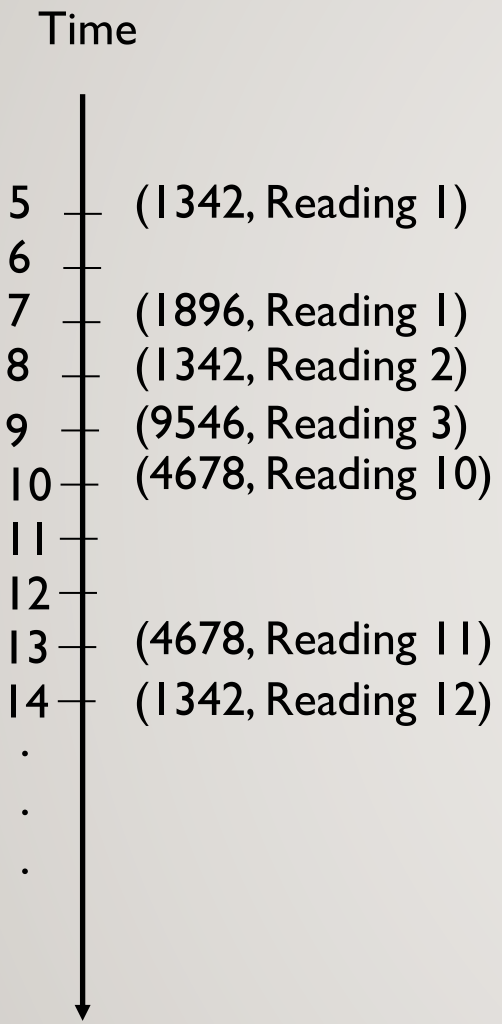


How to process streaming data

- As data arrives
  - Simple operations (e.g., filtering)
- Batching
  - Analytics
  - Windowing

PID	Reading	Timestamp
1342	Reading 1	5
1896	Reading 1	7
1342	Reading 2	8
9546	Reading 3	9
4678	Reading 10	10
4678	Reading 11	13
1342	Reading 12	14

How Does Data Stream



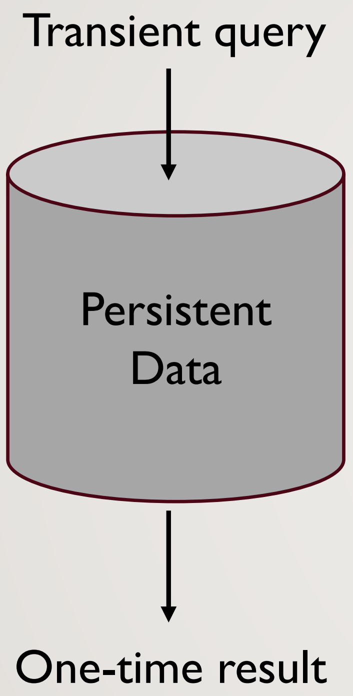
How to process streaming data

- As data arrives
  - Simple operations (e.g., filtering)
- Batching
  - Analytics
  - Windowing

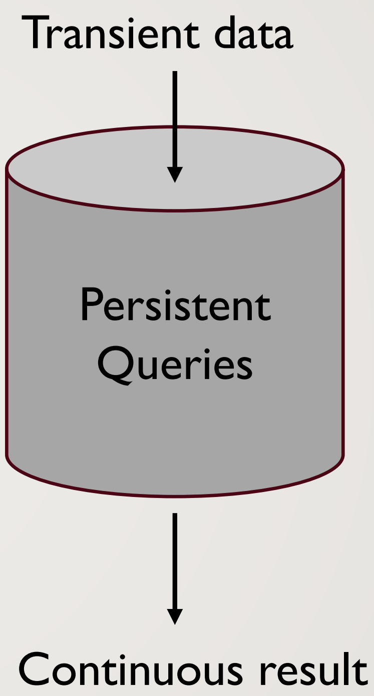
PID	Reading	Timestamp
1342	Reading 1	5
1896	Reading 1	7
1342	Reading 2	8
9546	Reading 3	9
4678	Reading 10	10
4678	Reading 11	13
1342	Reading 12	14

TRADITIONAL DBMS VS STREAMING

DBMS



Streaming



- Other differences
  - Push-based (data-driven)
  - Persistent queries

- Unbounded stream
- System conditions may not be stable

PLAN

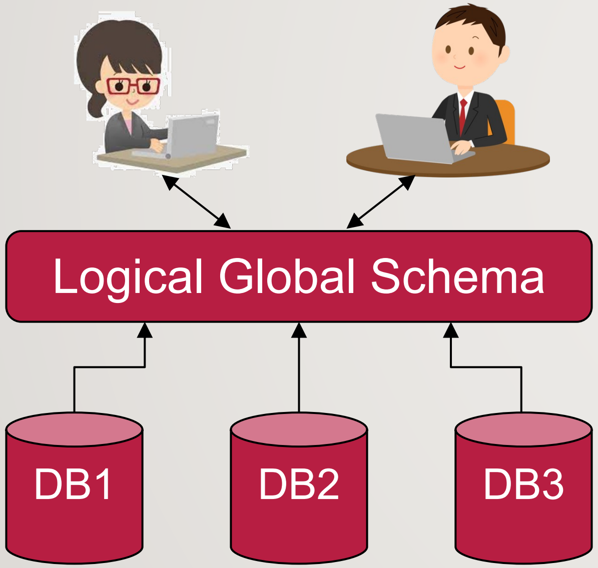
Data Management Basics

Big Data Concerns

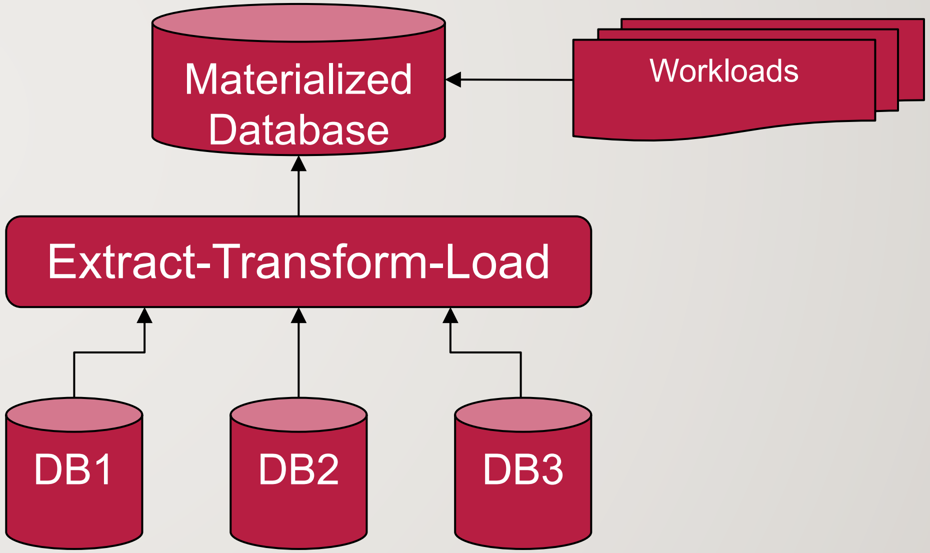
Data Integration

DATA INTEGRATION

Logical Integration



Physical Integration = Data Warehouse



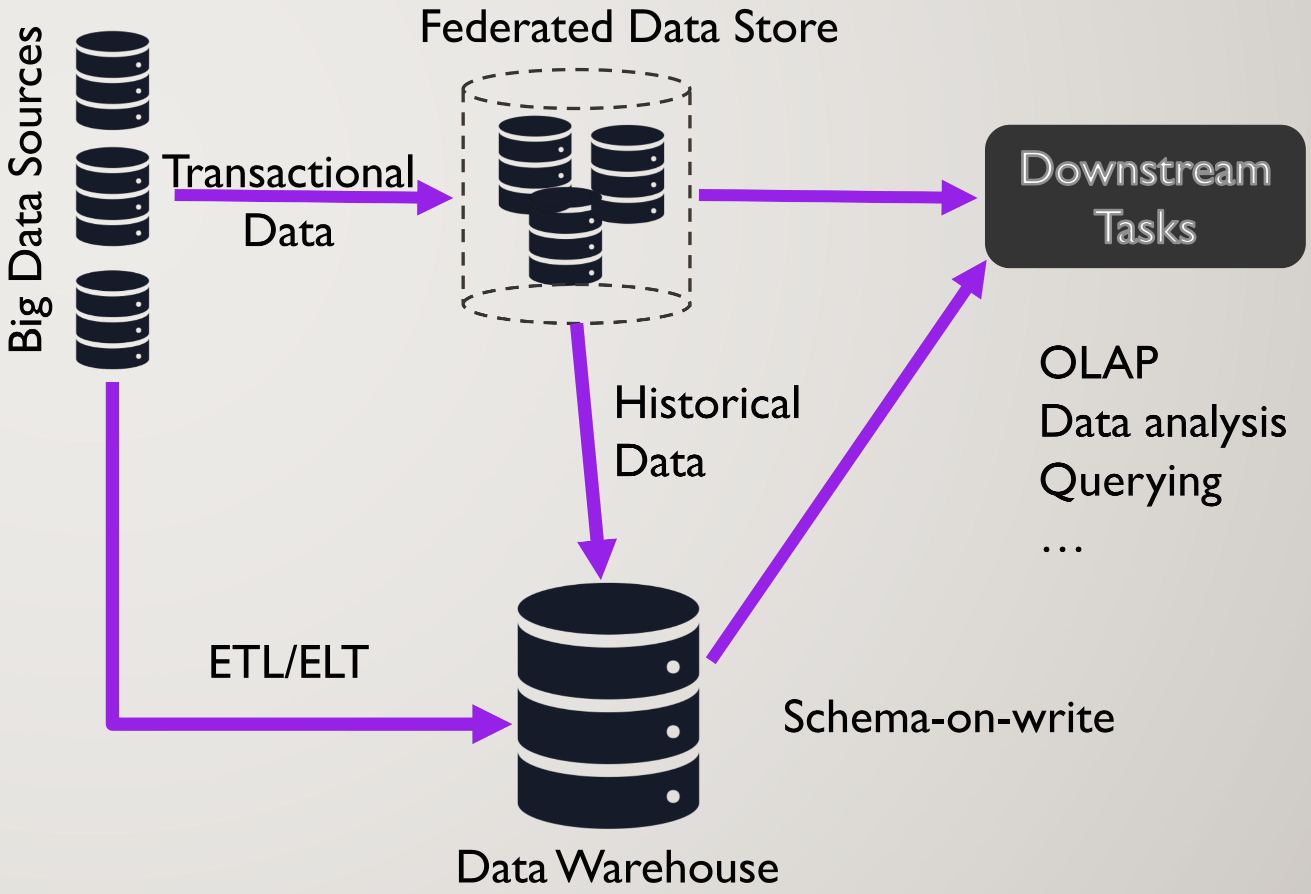
Data Warehouse

"A subject-oriented, integrated, nonvolatile, time-variant collection of data in support of management's decisions." [W.H.Inmon]

Data comes from multiple databases

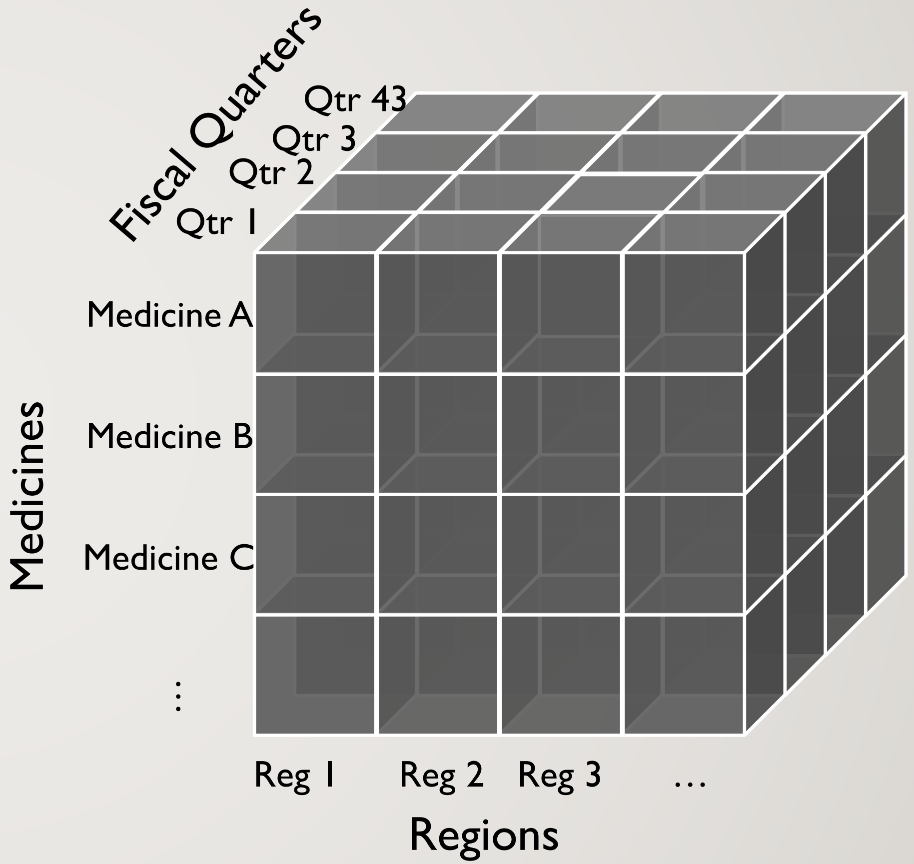
Tools to make business decisions quickly and reliably based on historical data

- **Extract, Transform, and Load** (ETL)
  - Extracted from multiple, heterogeneous sources
  - Includes data cleaning to ensure validity and consistency
- Analyzed data fed back into operating DB and data management



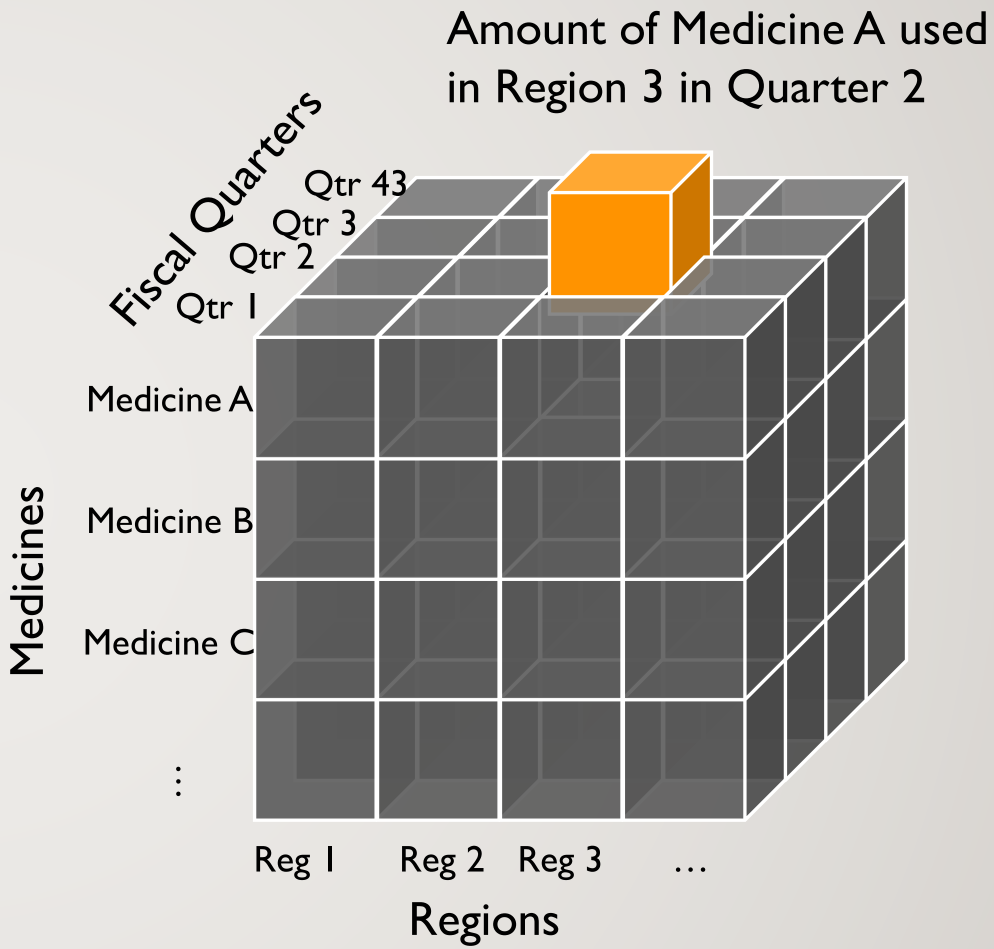
Data Modeling for Data Warehouses

- Usually multi-dimensional
- Advantages
  - Hierarchical views
    - Roll-up/drill-down
  - Querying directly in any combination of dimensions
- In relational implementations, dimensions mapped to tables



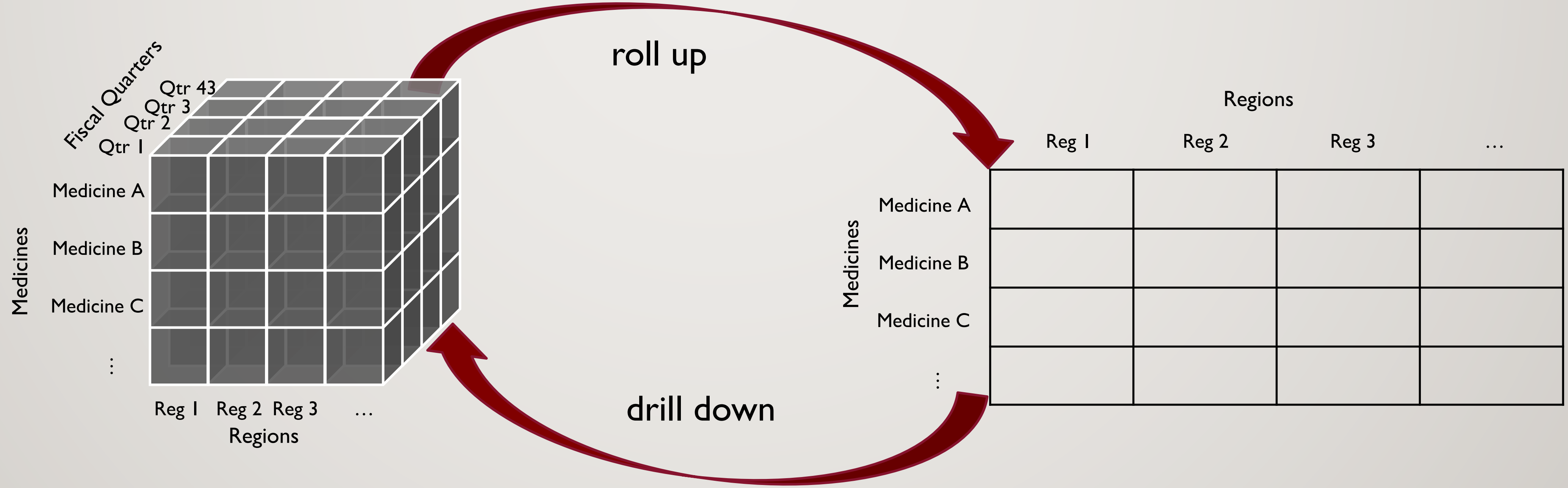
Data Modeling for Data Warehouses

- Usually multi-dimensional
- Advantages
  - Hierarchical views
    - Roll-up/drill-down
  - Querying directly in any combination of dimensions
- In relational implementations, dimensions mapped to tables



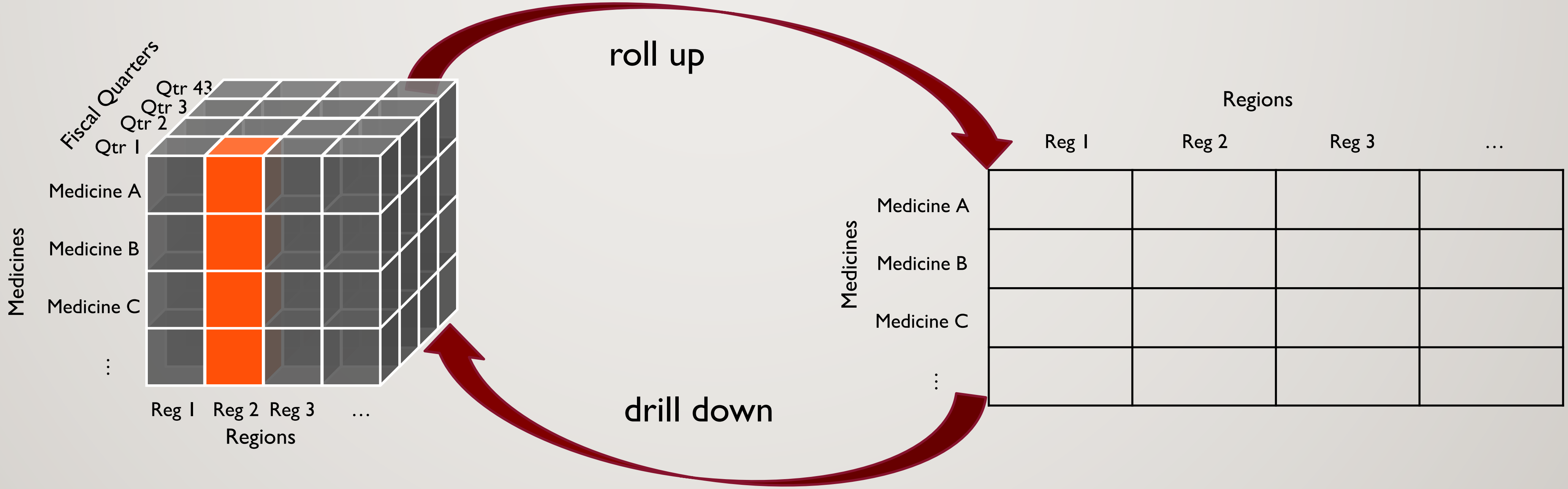
Warehouse Functionality

- Roll-up: Data is summarized with increasing generalization
  - E.g., going from daily or weekly reports to annual aggregations
- Drill-Down: Increasing levels of detail are revealed
  - E.g., going from national sales to sales from a particular region



Warehouse Functionality

- Roll-up: Data is summarized with increasing generalization
  - E.g., going from daily or weekly reports to annual aggregations
- Drill-Down: Increasing levels of detail are revealed
  - E.g., going from national sales to sales from a particular region
- Slice-and-dice: Select and project data with respect to some dimensions
  - E.g., finding sales in a given region in a given quarter



Data Integration – Data Lakes

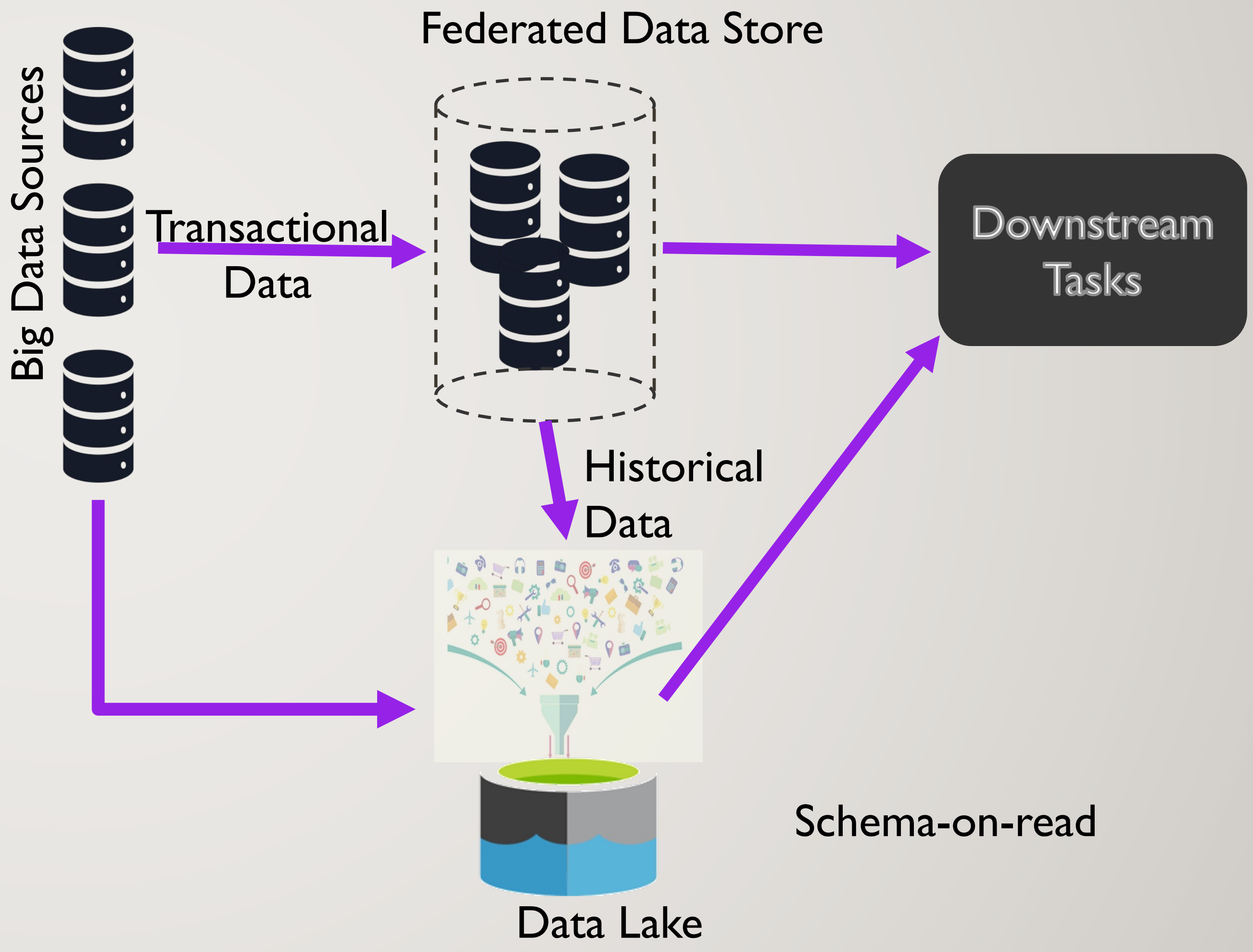
"massive collection of datasets that:

- may be hosted in different storage systems;
- may vary in their formats;
- may not be accompanied by any useful metadata or may use different formats to describe their metadata; and
- may change autonomously over time."

Nargesian et al, Data Lake Management: Challenges and Opportunities, *PVLDB*, 2019.

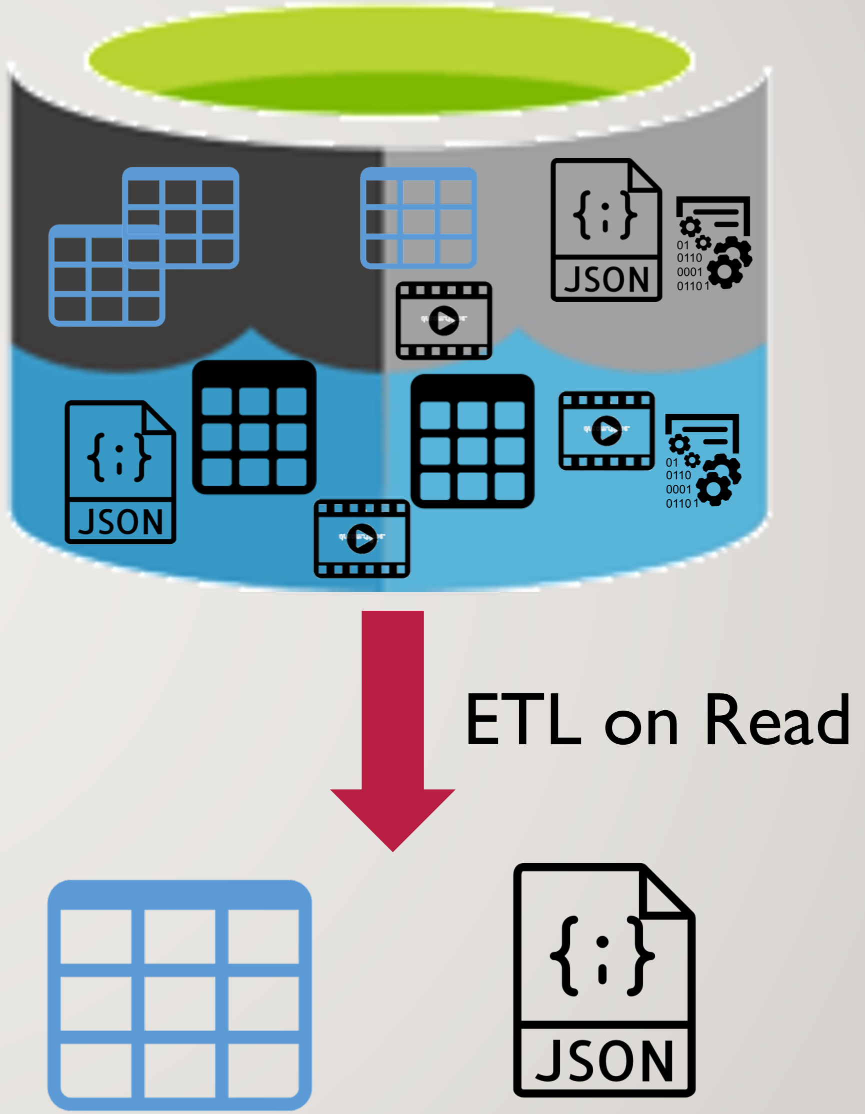
Data Lake

- Capture data in its raw form
- No schema
  - No ETL/ELT
  - Schema-on-read
- Do not know what the data will be used for
- Let downstream tasks (applications) provide schema
- Easier to participate, harder to use

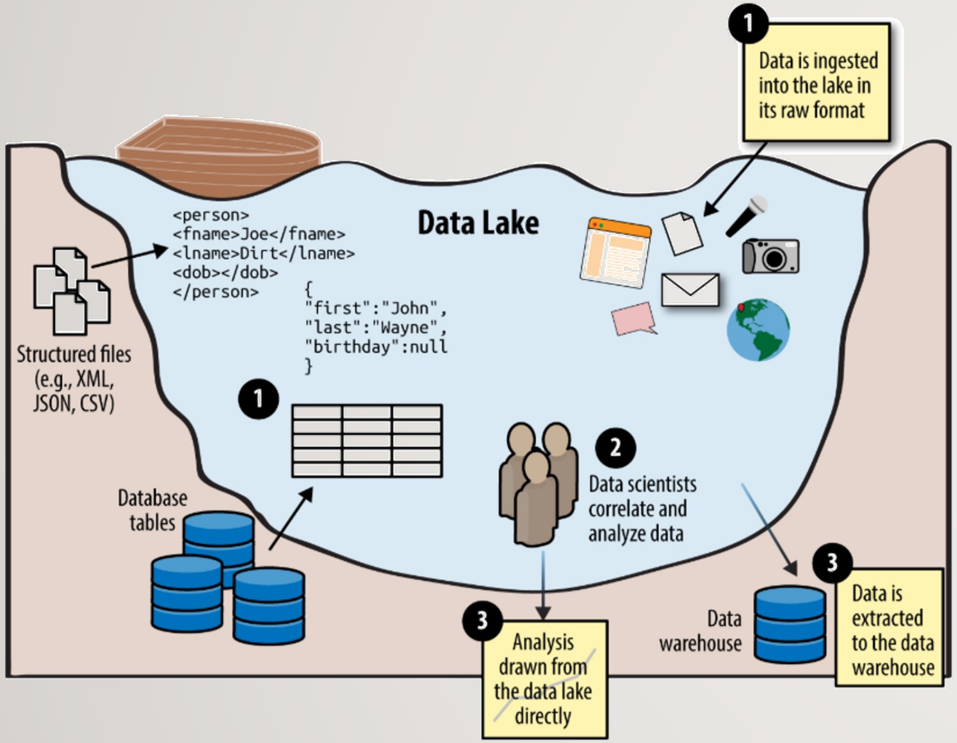


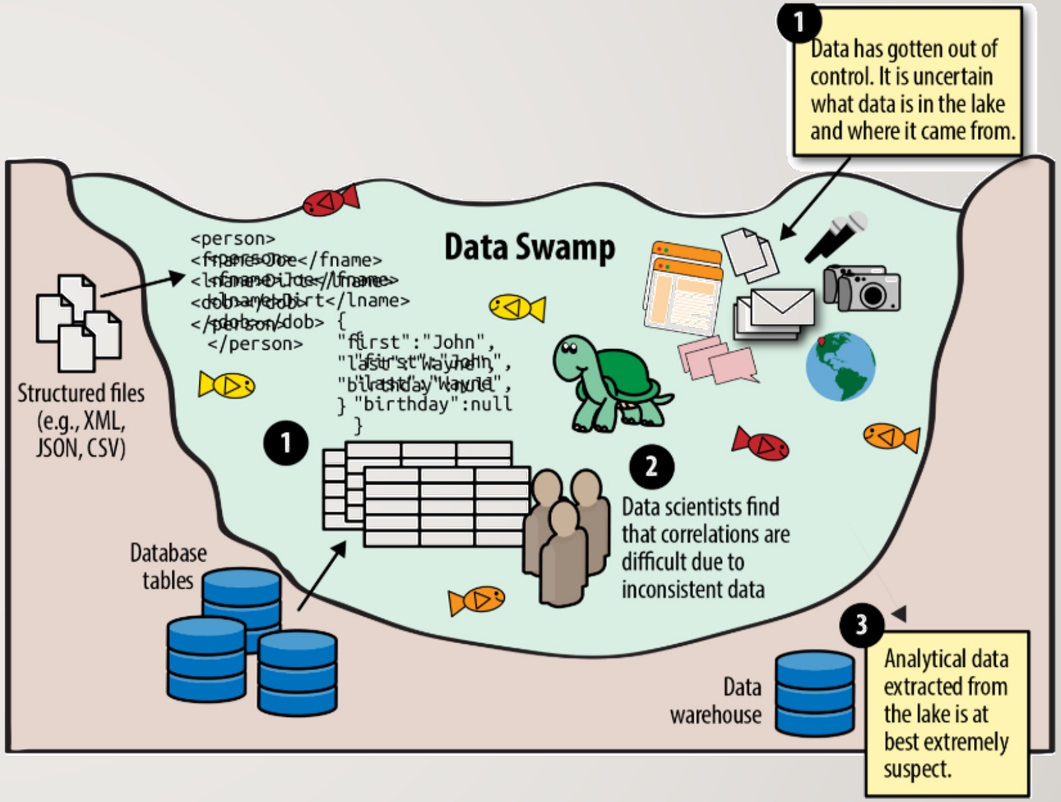
Contents of Data Lake & Use

- Raw data
  - Structured (Tabular)
  - Semi-structured (JSON, log files)
  - Unstructured (videos, images, binary files)
- Schema-on-Read
  - Application finds data
    - How?
  - Application formats data
    - ETL on Read

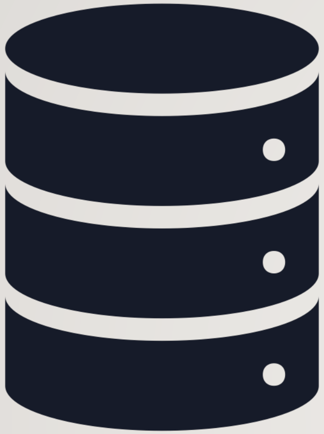


BEWARE OF DATA SWAMPS





Data Warehouses vs Data Lakes



- Simpler to architect
- Single store
- Centralized analytics
- Privacy concerns

- Complexity of dealing with autonomous systems
- Distributed
- Federated/distributed analytics
- Maintain original ownership of data

Plan

Data Management Basics

Big Data Concerns

Data Integration

Data Quality/Cleaning

Data Preparation

1. **Data Acquisition**

   Find data sources appropriate for the problem

2. **Dataset Selection**

   Determine which datasets are most useful and appropriate

3. **Data Integration**

   Integrate multi-modal data from multiple sources

4. **Data Quality**

   Address all impurities and errors in the integrated data

Data Preparation

Data Acquisition	Data Integration	Dataset Integration	Data Quality
Find data sources appropriate for the problem	Integrate multi-modal data from multiple sources	Determine which datasets are most useful and appropriate	Address all impurities and errors in the integrated data

DATA INTEGRATION ⇒ DATA QUALITY ISSUES

89% of executives believe that data quality issues impact the quality of customer service they provide (2017)

experian

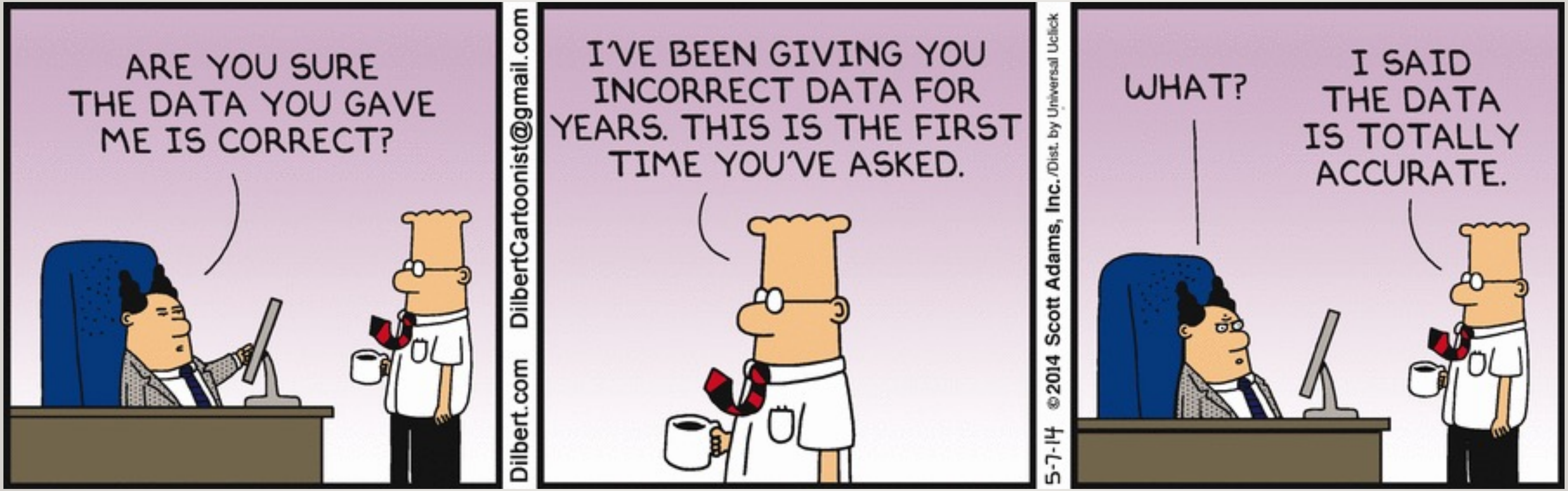
Only 33% of senior executives have a high level of trust in the accuracy of their big data analytics (2016)

KPMG

59% of executives do not believe their company has capabilities to generate business insights from their data (2016)

BAIN & COMPANY

DATA INTEGRATION ⇒ DATA QUALITY ISSUES



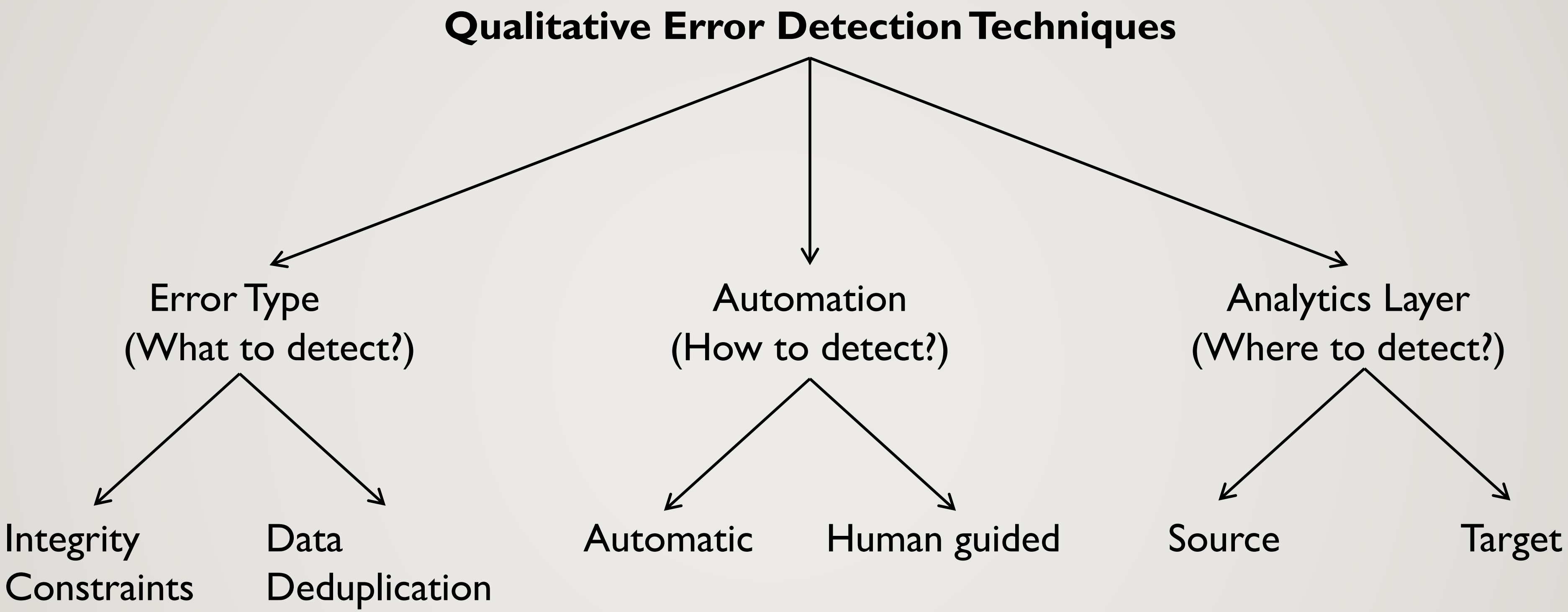
DATA QUALITY – DEALING WITH VERACITY

DAMA UK Working Group, 2013

Data Cleaning

- What do we want to do?
  - Remove errors
  - Fill missing values
  - Transform units and formats
  - Map and align columns
  - Remove duplicate records
  - Fix integrity constraints violations
- Data unification
- Data repair

ERROR DETECTION

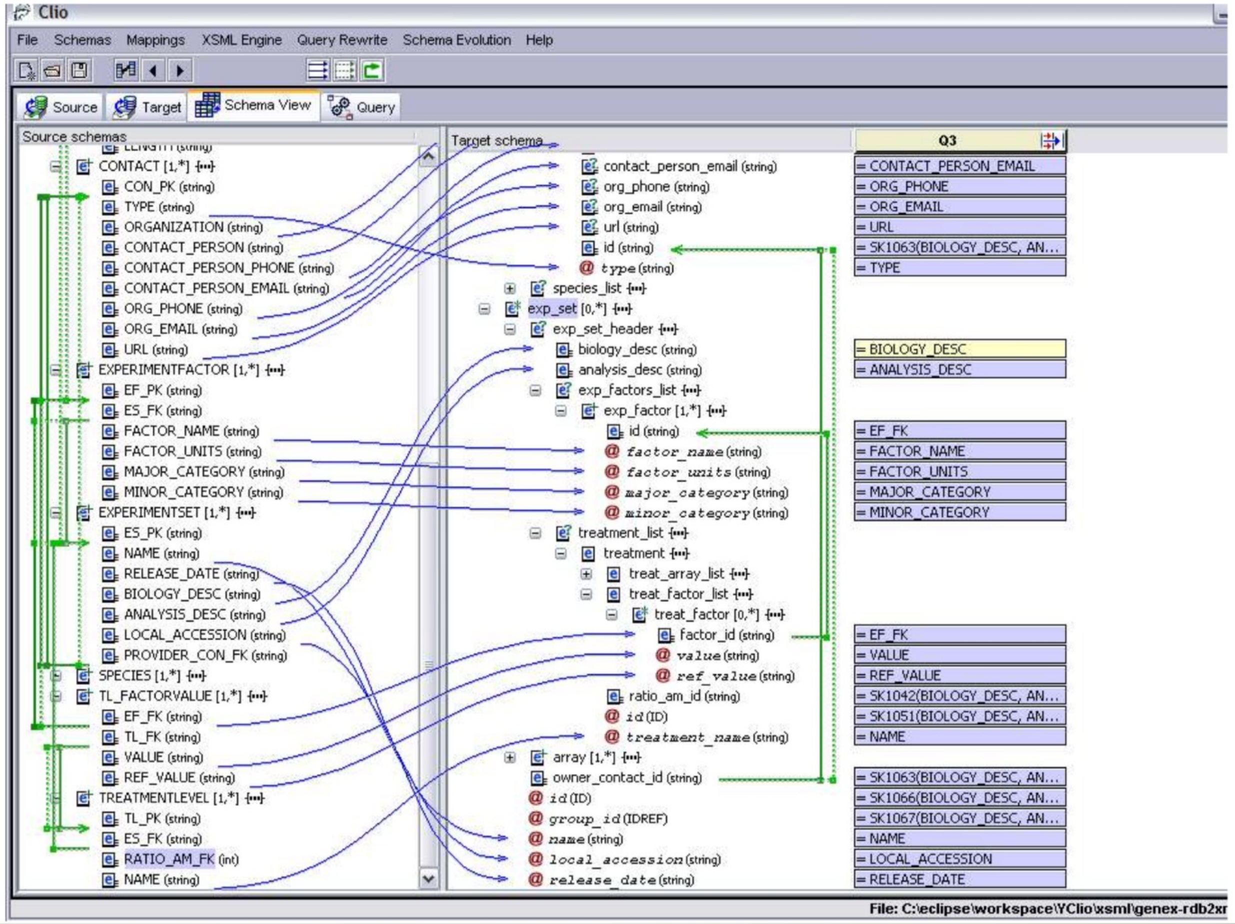


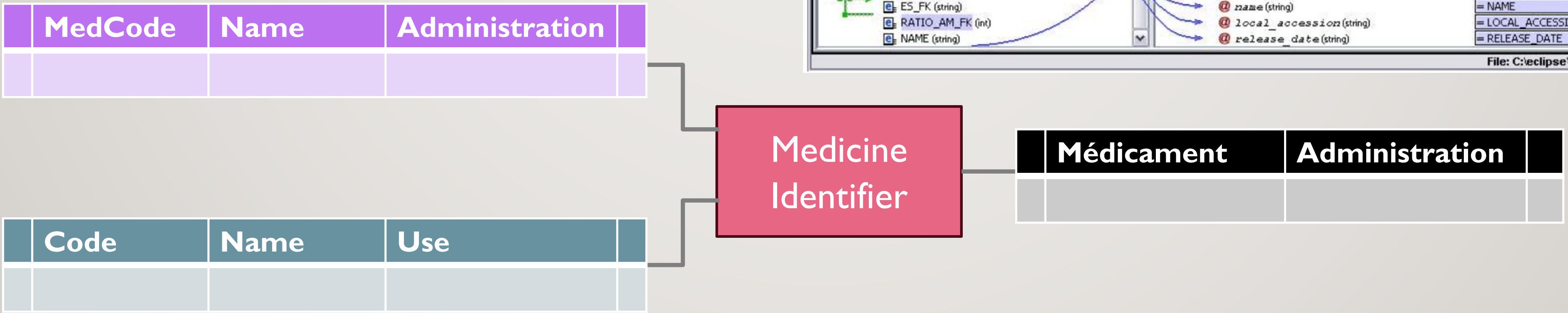
Ilyas and Chu, **Trends in Cleaning Relational Data: Consistency and Deduplication**
Foundations and Trends in Database Systems, 2015

Data Unification

Schema Mapping

- How entities in different data repositories map to each other
- Part of ETL/ELT process

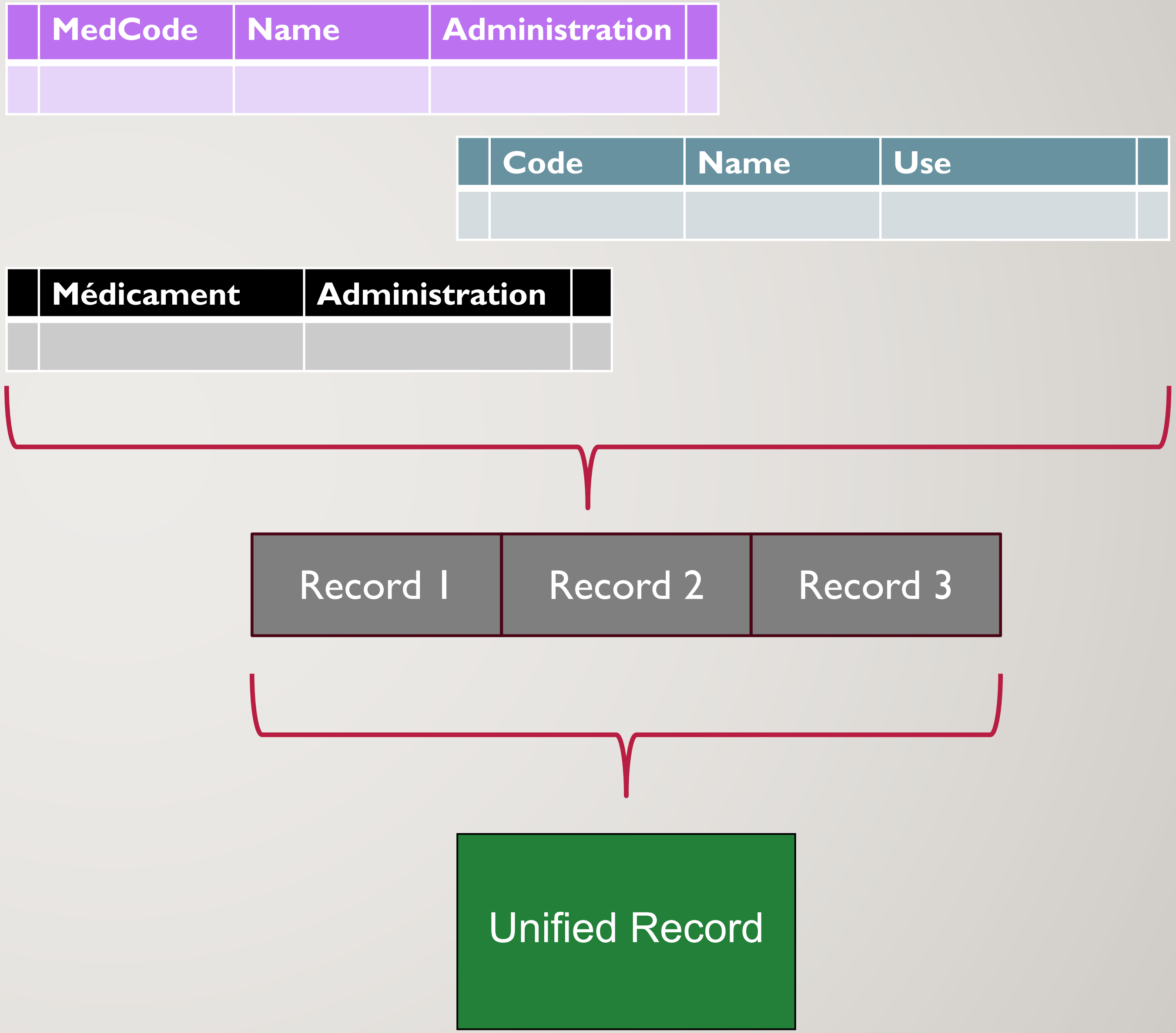




Data Unification

Data Deduplication

- Eliminating duplicate records
- Comparison → similarity measure
- Machine learning for classifying items as duplicates



DATA REPAIR

MISSING VALUES

- Real data is full of Nulls, and special values (99999)
- Curse of Nulls
  - N/A
  - Don't know
  - Not entered
- Make query answering difficult

PATIENT

PID	PName	InsNo	Street	City	Province
49875	Jane Smith	ON8677	54 Beachwood St.	Waterloo	Ontario
15948	Ali Nadir	ON7740	583 College St.	Toronto	Ontario
98143	Jiang Ni		189-95 Ave.	Edmonton	Alberta
75880	Tom White	ON6409	884 Water St.	Burlington	
13086	Mark Smith	ON7843	54 King St.	Waterloo	Ontario

Data Repair

Rule Violations

- Consider integrity constraints

EMPLOYEE

ID	FNname	LName	ROLE	CITY	PROV	SALARY
105	Anne	Nash	Mgr	Toronto	ON	110
211	Mark	White	Eng	Vancouver	BC	80
386	Mark	Lee	Eng	Edmonton	AB	75
235	John	Smith	Mgr	Toronto	ON	1200

Data Repair

Rule Violations

- Consider integrity constraints
  - Describe business rules
  - (ROLE,CITY) → SALARY

EMPLOYEE

ID	FNname	LName	ROLE	CITY	PROV	SALARY
105	Anne	Nash	Mgr	Toronto	ON	110
211	Mark	White	Eng	Vancouver	BC	80
386	Mark	Lee	Eng	Edmonton	AB	75
235	John	Smith	Mgr	Toronto	ON	1200

Two employees of the same role, the one who lives in Toronto cannot earn less than the one who does not live in Toronto

Data Repair

Rule Violations

- Consider integrity constraints
  - Describe business rules
  - (ROLE,CITY) → SALARY

EMPLOYEE

ID	FNname	LName	ROLE	CITY	PROV	SALARY
105	Anne	Nash	Mgr	Toronto	ON	110
211	Mark	White	Eng	Vancouver	BC	80
386	Mark	Lee	Eng	Edmonton	AB	75
235	John	Smith	Mgr	Toronto	ON	1200

Two employees of the same role, the one who lives in Toronto cannot earn less than the one who does not live in Toronto

Data Repair

Rule Violations

- Consider integrity constraints
  - Describe business rules
  - (ROLE,CITY) → SALARY
- Rule violations need to be repaired

EMPLOYEE

ID	FNname	LName	ROLE	CITY	PROV	SALARY
105	Anne	Nash	Mgr	Toronto	ON	110
211	Mark	White	Eng	Vancouver	BC	80
386	Mark	Lee	Eng	Edmonton	AB	75
235	John	Smith	Mgr	Toronto	ON	1200

Two employees of the same role, the one who lives in Toronto cannot earn less than the one who does not live in Toronto

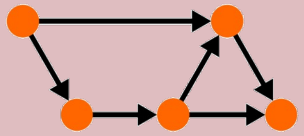
Plan

Data Management Basics

Big Data Concerns

Data Integration

Data Preparation/Cleaning

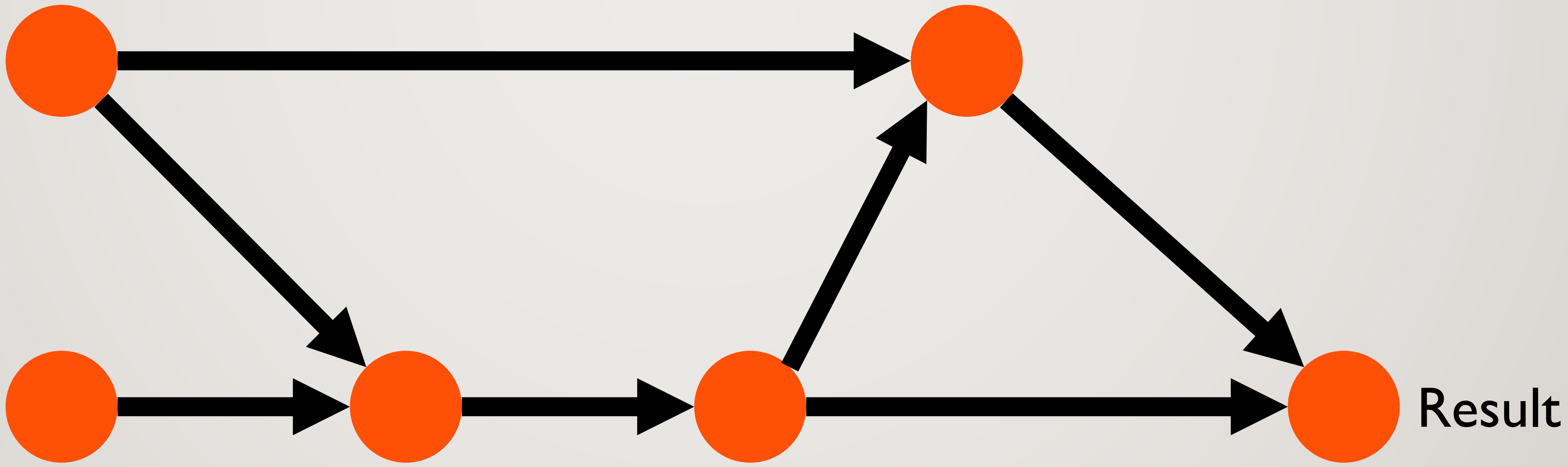


Data Provenance

DATA PROVENANCE

" 'Data provenance' refers to a record trail that accounts for the origin of a piece of data (in a database, document or repository) together with an explanation of how and why it got to the present place. "

[Encyclopedia of Database Systems]

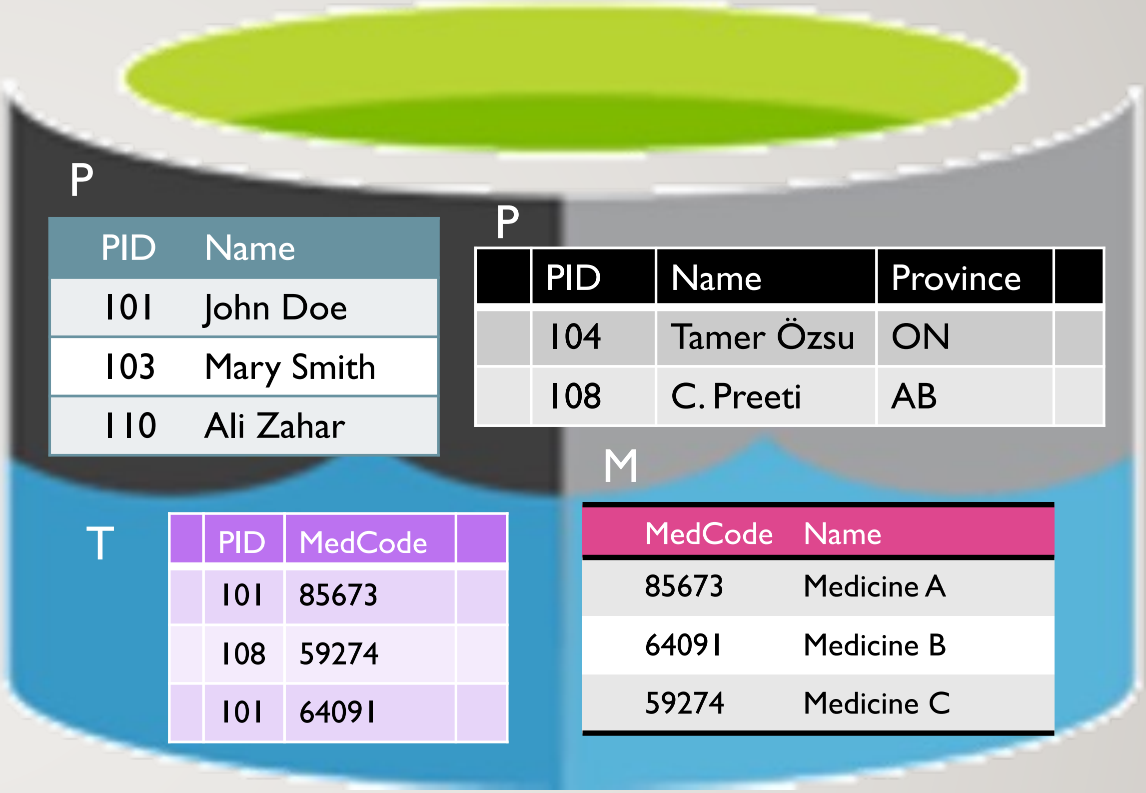


Data Provenance Concerns

- How was this result generated?
- What mappings produced it?
- How trustable is it?

Data Provenance Concerns

- How was this result generated?
- What mappings produced it?
- How trustable is it?



P

PID	Name
101	John Doe
103	Mary Smith
110	Ali Zahar

P

PID	Name	Province
104	Tamer Özsu	ON
108	C. Preeti	AB

T

PID	MedCode
101	85673
108	59274
101	64091

M

MedCode	Name
85673	Medicine A
64091	Medicine B
59274	Medicine C

Data Provenance Concerns

- How was this result generated?
- What mappings produced it?
- How trustable is it?

P

PID	Name
101	John Doe
103	Mary Smith
110	Ali Zahar

P

PID	Name	Province
104	Tamer Özsu	ON
108	C. Preeti	AB

T

PID	MedCode
101	85673
108	59274
101	64091

M

MedCode	Name
85673	Medicine A
64091	Medicine B
59274	Medicine C

Find names of patients and the names of medications that they take.

Data Provenance Concerns

- How was this result generated?
- What mappings produced it?
- How trustable is it?

P

PID	Name
101	John Doe
103	Mary Smith
110	Ali Zahar

P

PID	Name	Province
104	Tamer Özsu	ON
108	C. Preeti	AB

T

PID	MedCode
101	85673
108	59274
101	64091

M

MedCode	Name
85673	Medicine A
64091	Medicine B
59274	Medicine C

Find names of patients and the names of medications that they take.

```
SELECT  P.Pname AS PatientName, M.Name AS MedName
FROM    P, T, M
WHERE   P.PID = T.PID
AND     T.MedCode = M.MedCode;
```

Data Provenance Concerns

- How was this result generated?
- What mappings produced it?
- How trustable is it?

P

PID	Name
101	John Doe
103	Mary Smith
110	Ali Zahar

P

PID	Name	Province
104	Tamer Özsu	ON
108	C. Preeti	AB

T

PID	MedCode
101	85673
108	59274
101	64091

M

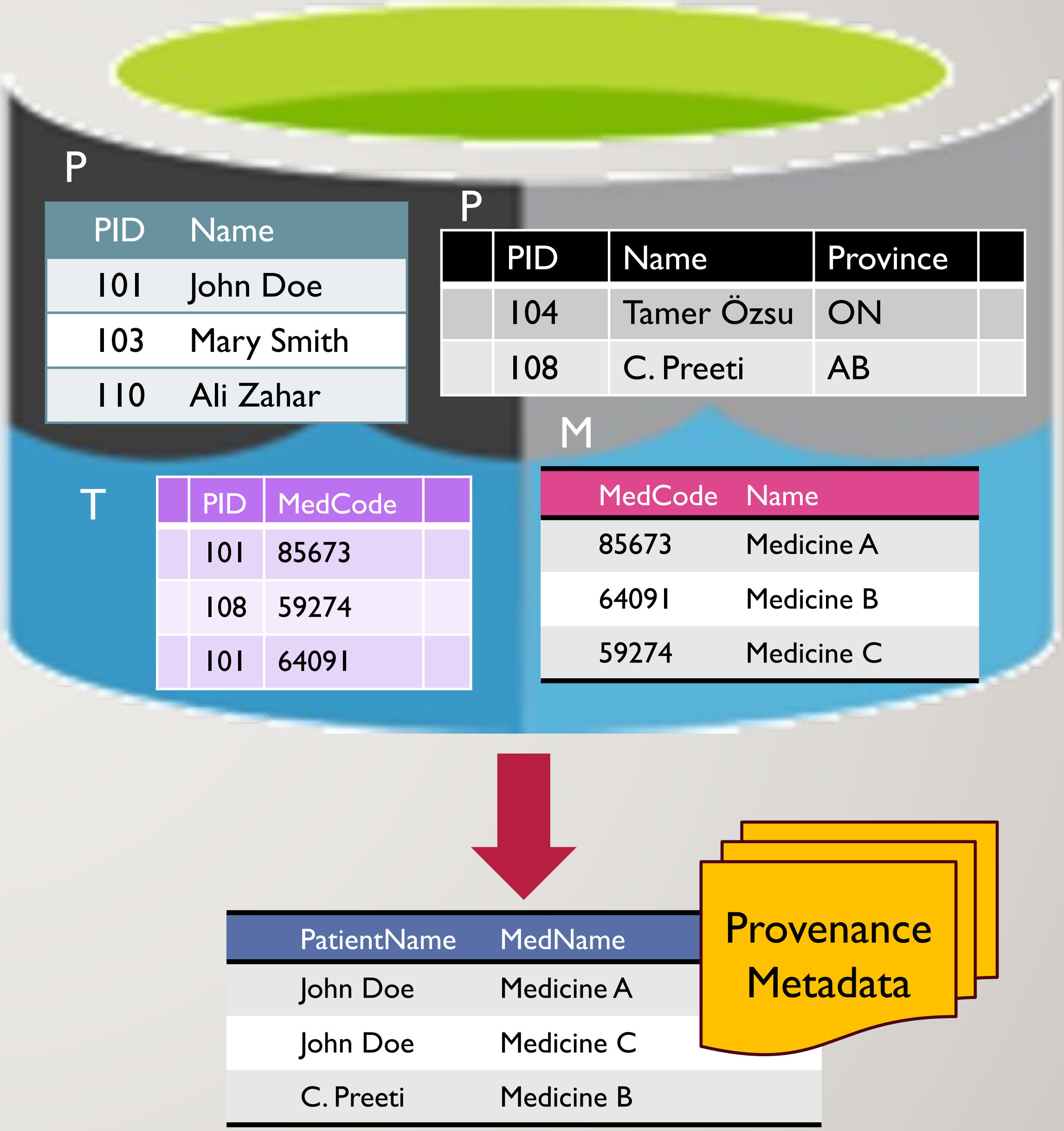
MedCode	Name
85673	Medicine A
64091	Medicine B
59274	Medicine C

Find names of patients and the names of medications that they take.

PatientName	MedName
John Doe	Medicine A
John Doe	Medicine C
C. Preeti	Medicine B

Management Strategies

- Lazy
  - Compute provenance when needed
  - Access to source is needed
- Eager
  - Keep an annotation with the result
  - Provenance info can be looked up
- Annotation
  - Why – what data contributed
  - How – what process created result
  - Where – What column of what row does each result row value come from



Provenance – Final Words

- Provenance is critical to understanding and assessing the believability of data
- Provenance can be helpful in
  - Explainability
    - Why an item exists
  - Scoring
    - Ranked list of results in terms of relevance
  - Reasoning about interactions
    - Demonstrate data relationships
- Careful: I have only covered the basics of provenance
  - Deriving mapping
  - Representing mapping
  - Storing annotations
  - How to query provenance

All of this gets data ready for analysis!

All of this gets data ready for analysis!
Remember …