Data Warehouse

What is a Data Warehouse?

- A data repository where all relevant enterprise data is stored
- Provides online analytical processing (OLAP)
  - Multidimensional data analysis techniques
  - Advanced database support
  - Easy-to-use end-user interfaces
- Acts as a single source of truth for:
  - Reports
  - Analytics
  - Presentation
  - Portals
  - Dashboards

Why Do We Need Data Warehouse?

- Operational & transactional databases (OLTP) aren't suitable for reporting etc:
  - Data is scattered and don't provide accessible insight
  - Transaction processing is designed for single record accesss
  - Data is normalized ➔ generating report data can involve many joins
  - Report generation is slow ➔ impact on OLTP performance
- Combine data from multiple systems in a central repository
- Resolve inconsistencies among systems
- Reduce loads on operational systems in production
- Provide long term data storage

Goals of a Data Warehouse

- Makes an organization’s information easy to access (for reports, analytics etc)
- Presents the organization’s information in a consistent manner
- Be adaptive and resilient to change

- Meets the requirements of the business
- Business Intelligence → foundation for decision making

Data Warehouse Methodologies

- Corporate Information Factory (CIF) – proposed by Bill Inmon, father of the data warehouse (1994)
- Kimball Method (Star Schema) – proposed by Ralph Kimball

- Kimball method is the most widely used by far.

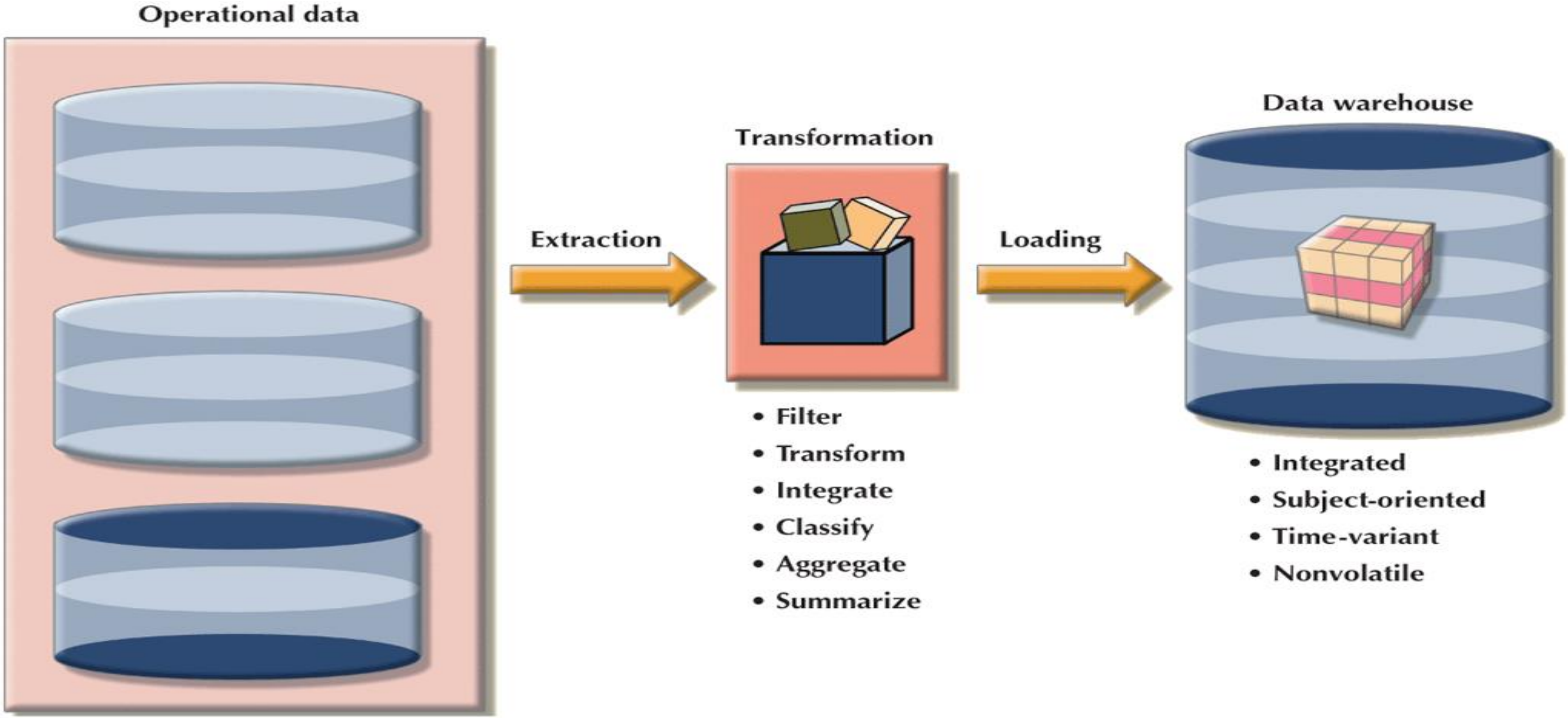
- Snowflake schema if normalizing the dimension tables in a star schema

Characteristics of Data Warehouse

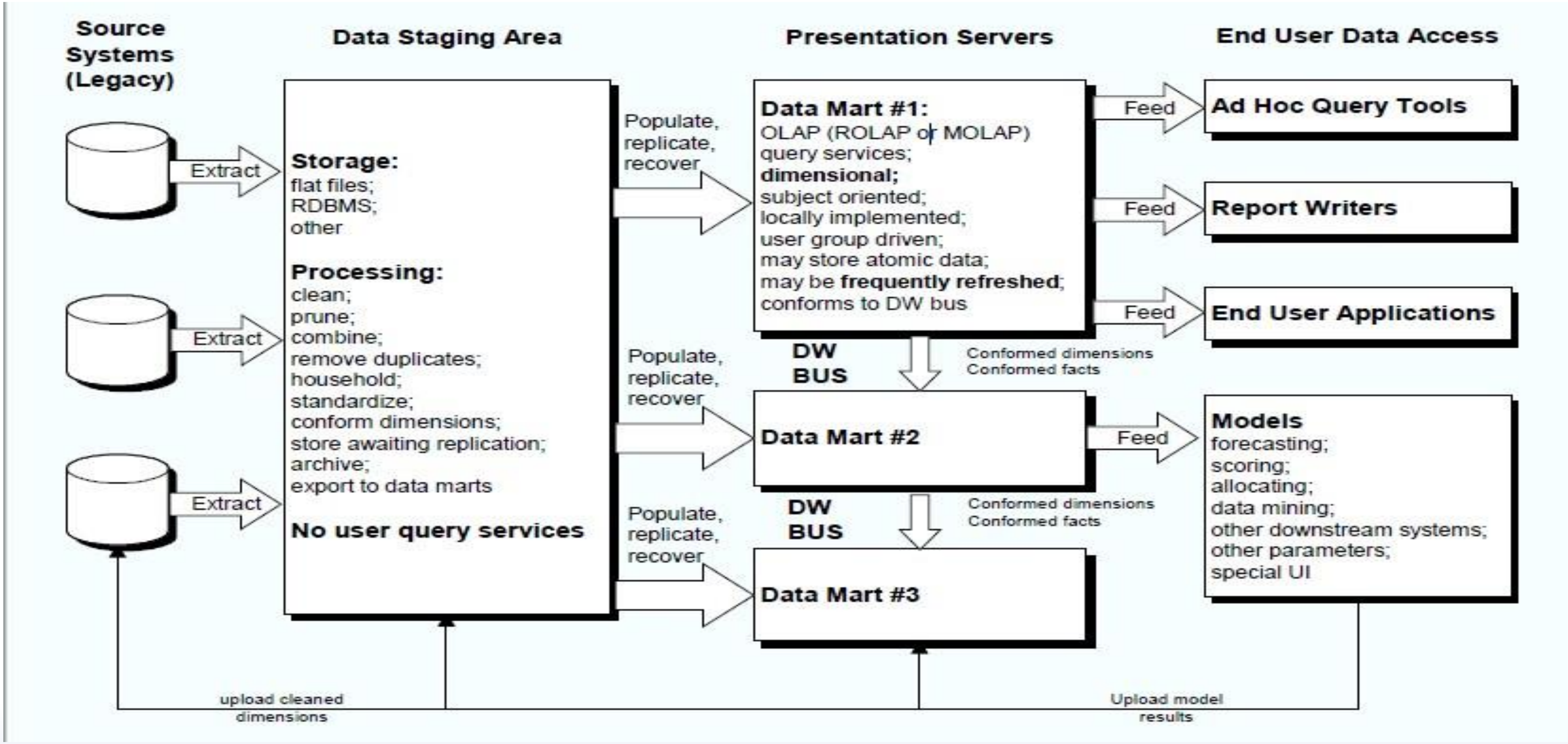
- "an *integrated*, *subject-oriented*, *time-variant*, *nonvolatile* collection of data that provides support for decision making." - Bill Inmon (1994)

CHARACTERISTIC	OPERATIONAL DATABASE DATA	DATA WAREHOUSE DATA
Integrated	Similar data can have different representations or meanings. For example, Social Security numbers may be stored as ###-##-#### or as #########, and a given condition may be labeled as T/F or 0/1 or Y/N. A sales value may be shown in thousands or in millions.	Provide a unified view of all data elements with a common definition and representation for all business units.
Subject-oriented	Data is stored with a functional, or process, orientation. For example, data may be stored for invoices, payments, and credit amounts.	Data is stored with a subject orientation that facilitates multiple views of the data and decision making. For example, sales may be recorded by product, division, manager, or region.
Time-variant	Data is recorded as current transactions. For example, the sales data may be the sale of a product on a given date, such as $342.78 on 12-MAY-2016.	Data is recorded with a historical perspective in mind. Therefore, a time dimension is added to facilitate data analysis and various time comparisons.
Nonvolatile	Data updates are frequent and common. For example, an inventory amount changes with each sale. Therefore, the data environment is fluid.	Data cannot be changed. Data is added only periodically from historical systems. Once the data is properly stored, no changes are allowed. Therefore, the data environment is relatively static.

Data Warehousing



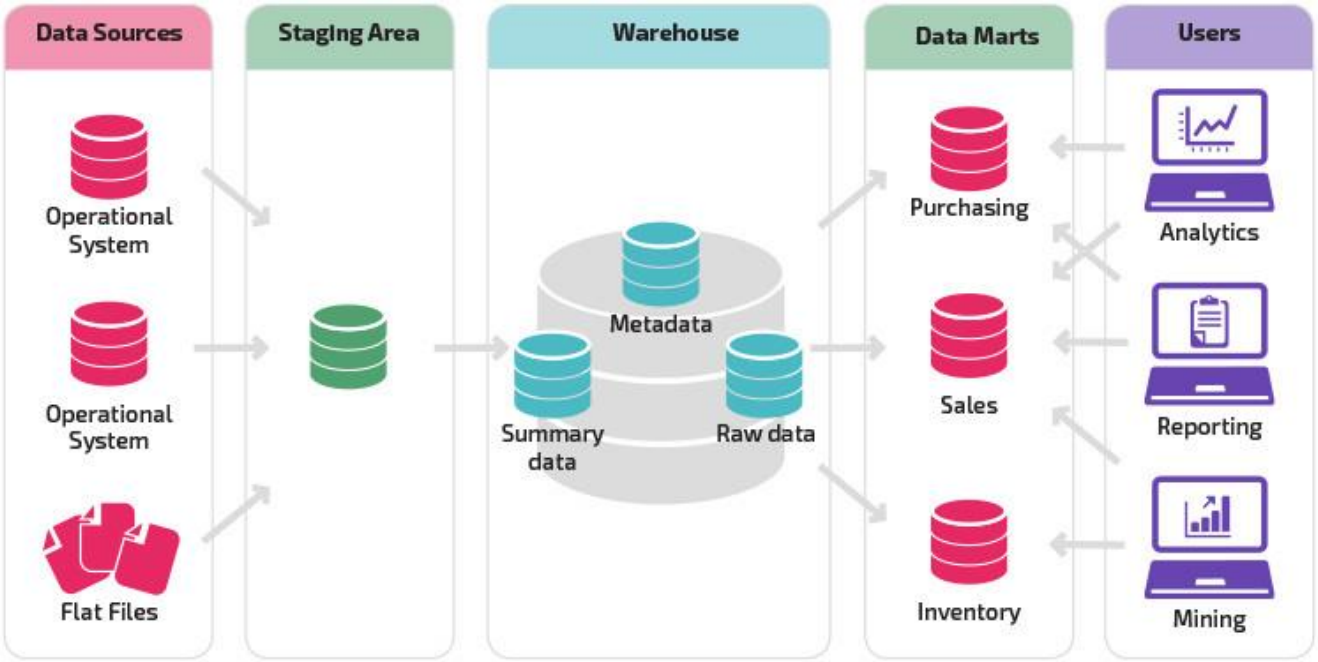
Basic Components of Data Warehouse



Data Marts

- Small, single-subject logical subset of a data warehouse
  - Contains not only the summary data but also atomic data
  - Provides decision support to a small group of people (e.g. for a department)

- Benefits over data warehouses
  - Lower cost and shorter implementation time

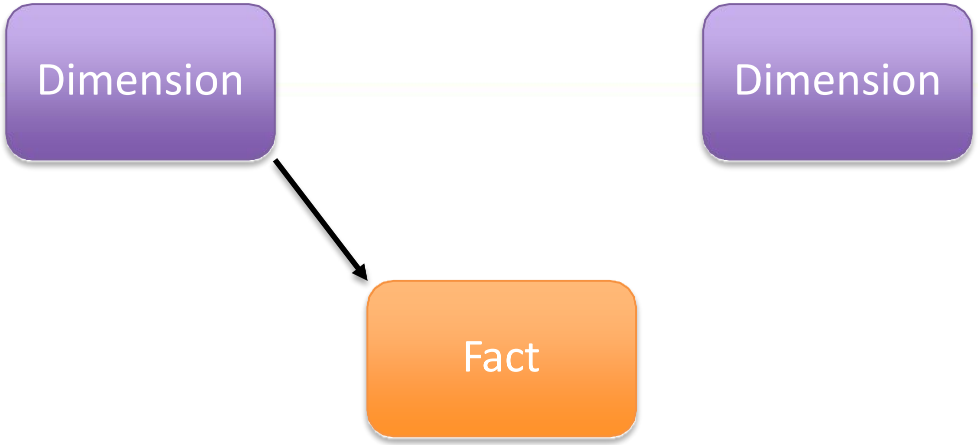


Databases vs Data Warehouses vs Data Marts

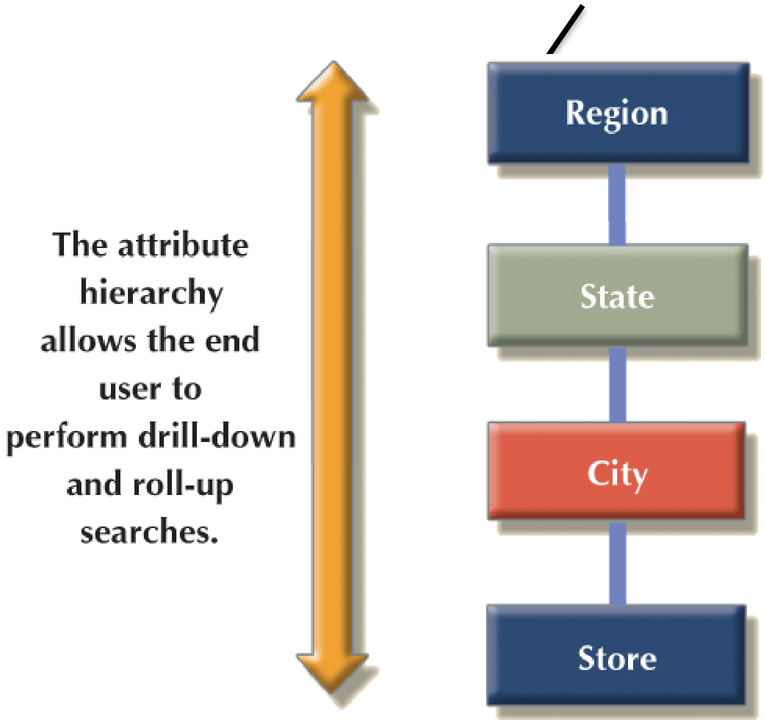
Data Modeling in Data Warehouse

- The Kimball method with **star schema** is used.
- Multidimensional decision support data is mapped into a relational database
- A near equivalent of multidimensional database schema is created from existing relational database
- Yields an easily implemented model for multidimensional data analysis

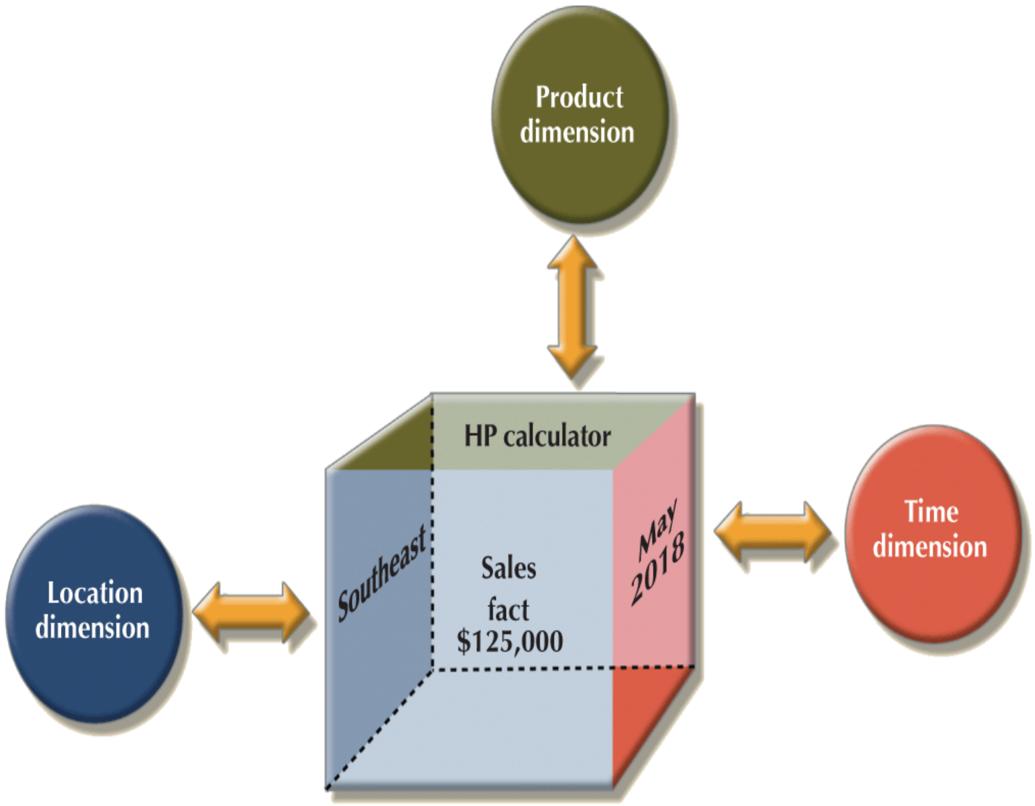
Star Schema Components

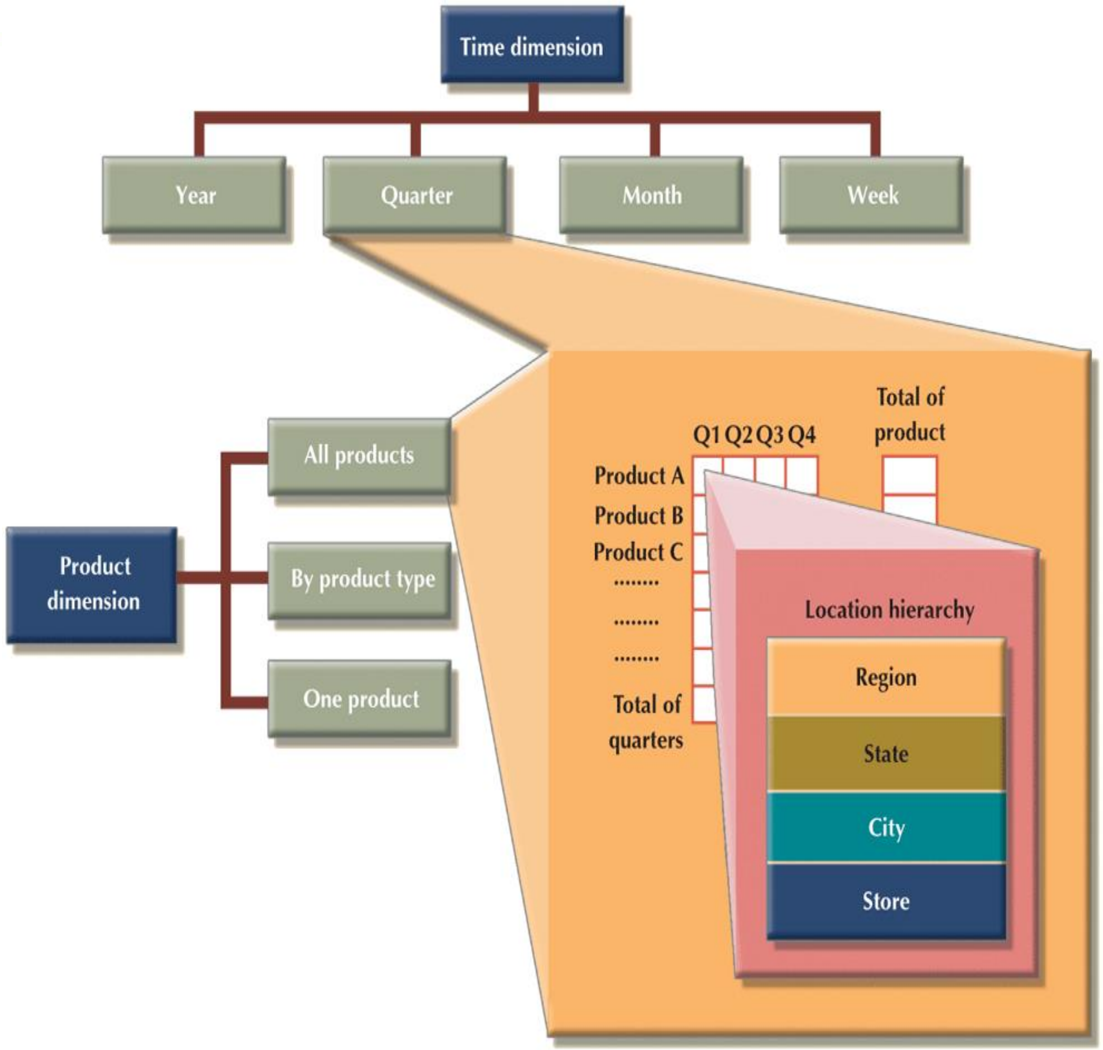


- **Facts**
  - hold numeric values that represent a specific business aspect/event
  - join dimensions such as who's, when's etc of event
- **Dimensions**
  - contain the values that describe facts
  - contains **attributes** that are used to search, filter, and classify facts
- **Attribute hierarchies**
  - provide a top-down data organization
  - Aggregation and drill-down/roll-up data analysis

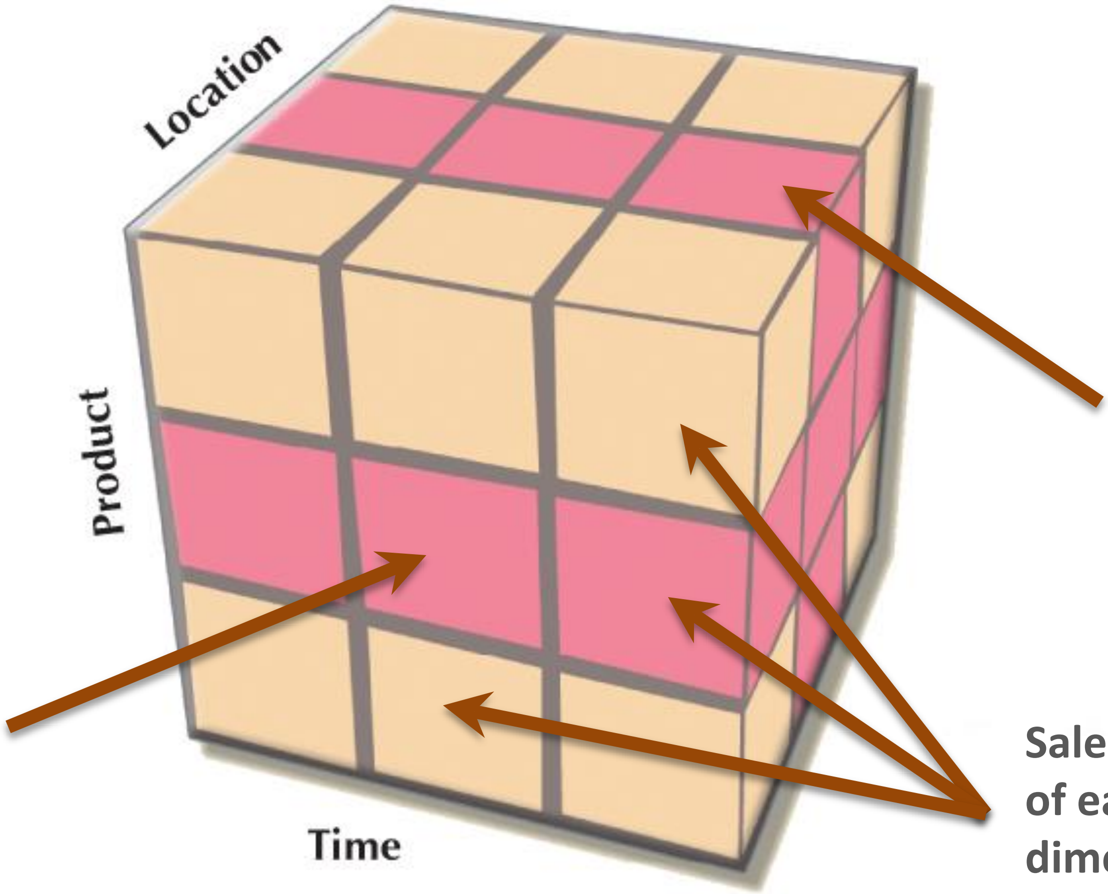


Data Modeling Example





Data Modeling Example – Data Cube

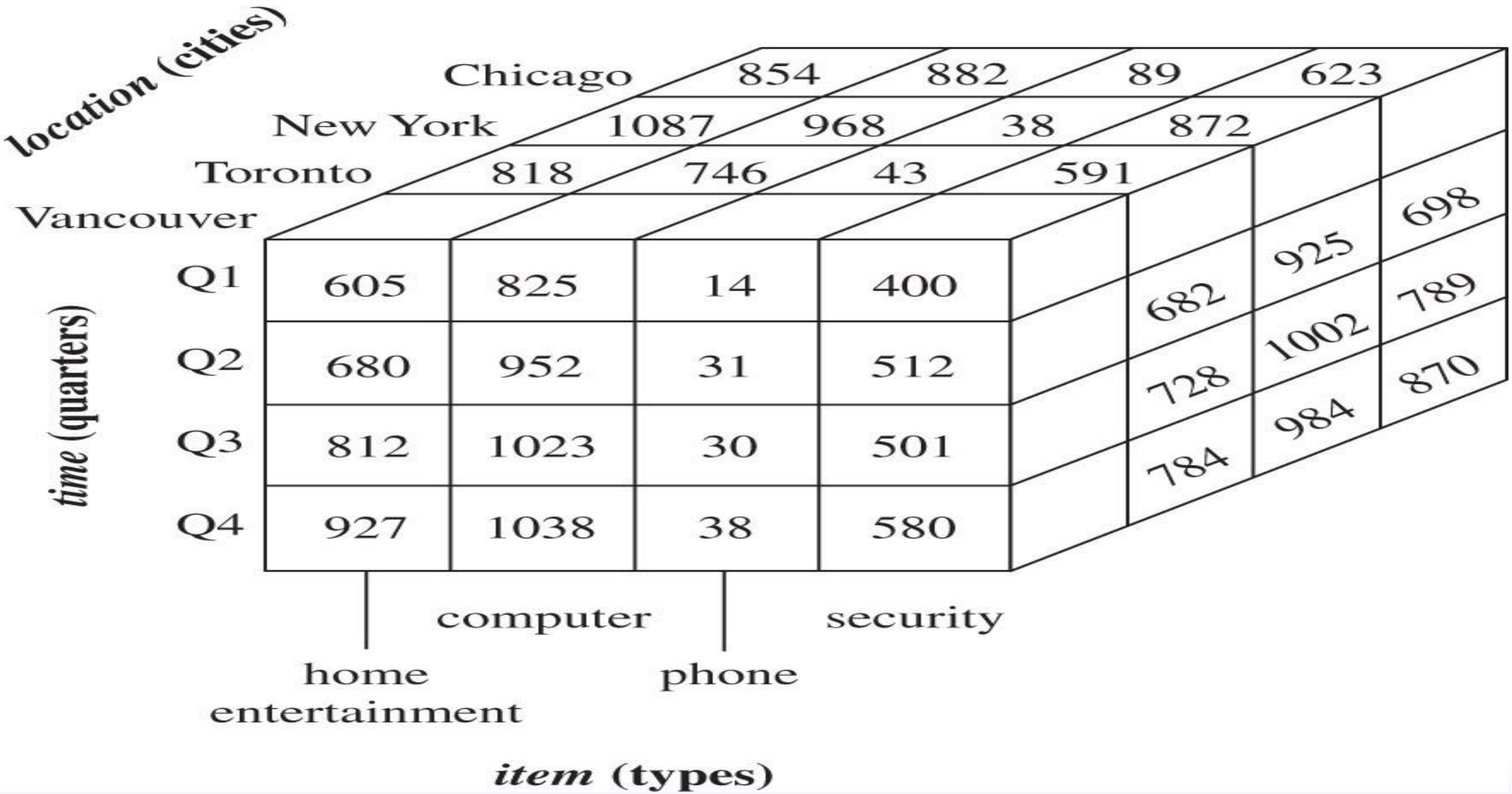


Sales manager's view of sales data

Product manager's view of sales data

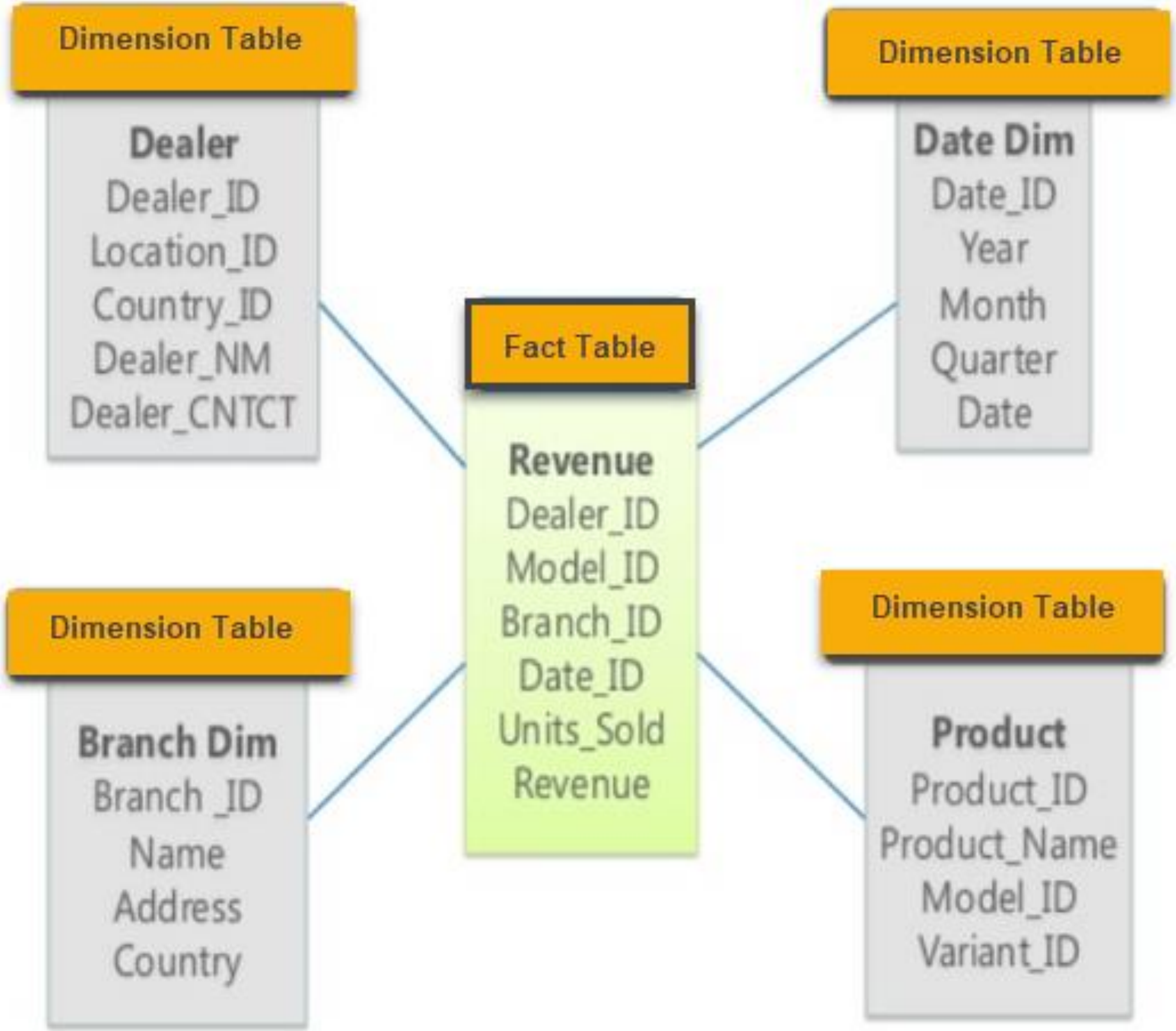
Sales facts are stored in the intersection of each product, time and location dimension.

conceptual three-dimensional cube of sales by product, location, and time


location (cities)
Chicago
854
882
89
623
New York
1087
968
38
872
Toronto
818
746
43
591
Vancouver
698
925
789
682
1002
728
870
984
784
time (quarters)
Q1
605
825
14
400
Q2
680
952
31
512
Q3
812
1023
30
501
Q4
927
1038
38
580
computer
security
home
entertainment
phone
item (types)

Star Schema Representation

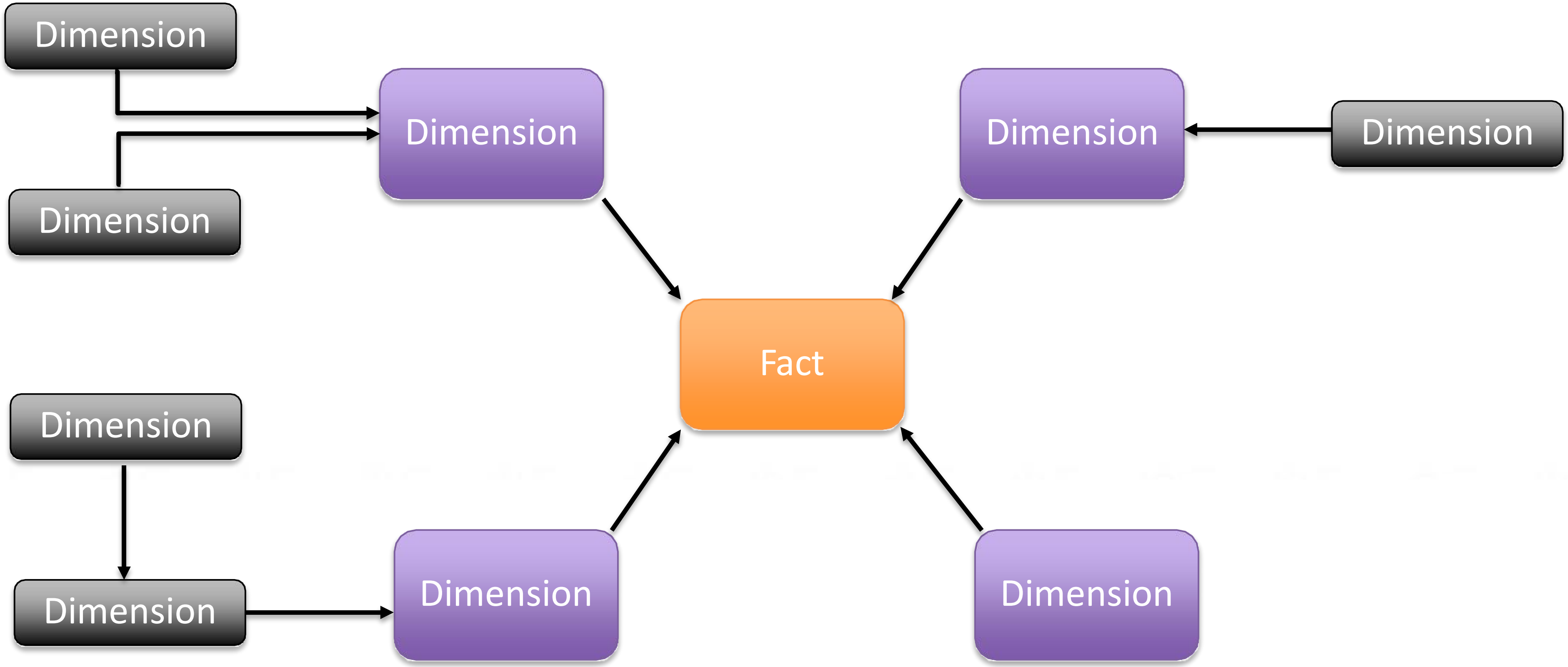
- Facts and dimensions are represented by physical tables in data warehouse
  - many-to-one (M:1) relationship between fact table and each dimension table
- Fact and dimension tables
  - related by foreign keys
  - subject to primary and foreign key constraints
  - primary key of a fact table
    - composite primary key because the fact table is related to many dimension tables
    - always formed by combining the foreign keys pointing to the related dimension tables


Dimension Table
Dealer
Dealer_ID
Location_ID
Country_ID
Dealer_NM
Dealer_CNTCT
Dimension Table
Date Dim
Date_ID
Year
Month
Quarter
Date
Fact Table
Revenue
Dealer_ID
Model_ID
Branch_ID
Date_ID
Units_Sold
Revenue
Dimension Table
Branch Dim
Branch_ID
Name
Address
Country
Dimension Table
Product
Product_ID
Product_Name
Model_ID
Variant_ID

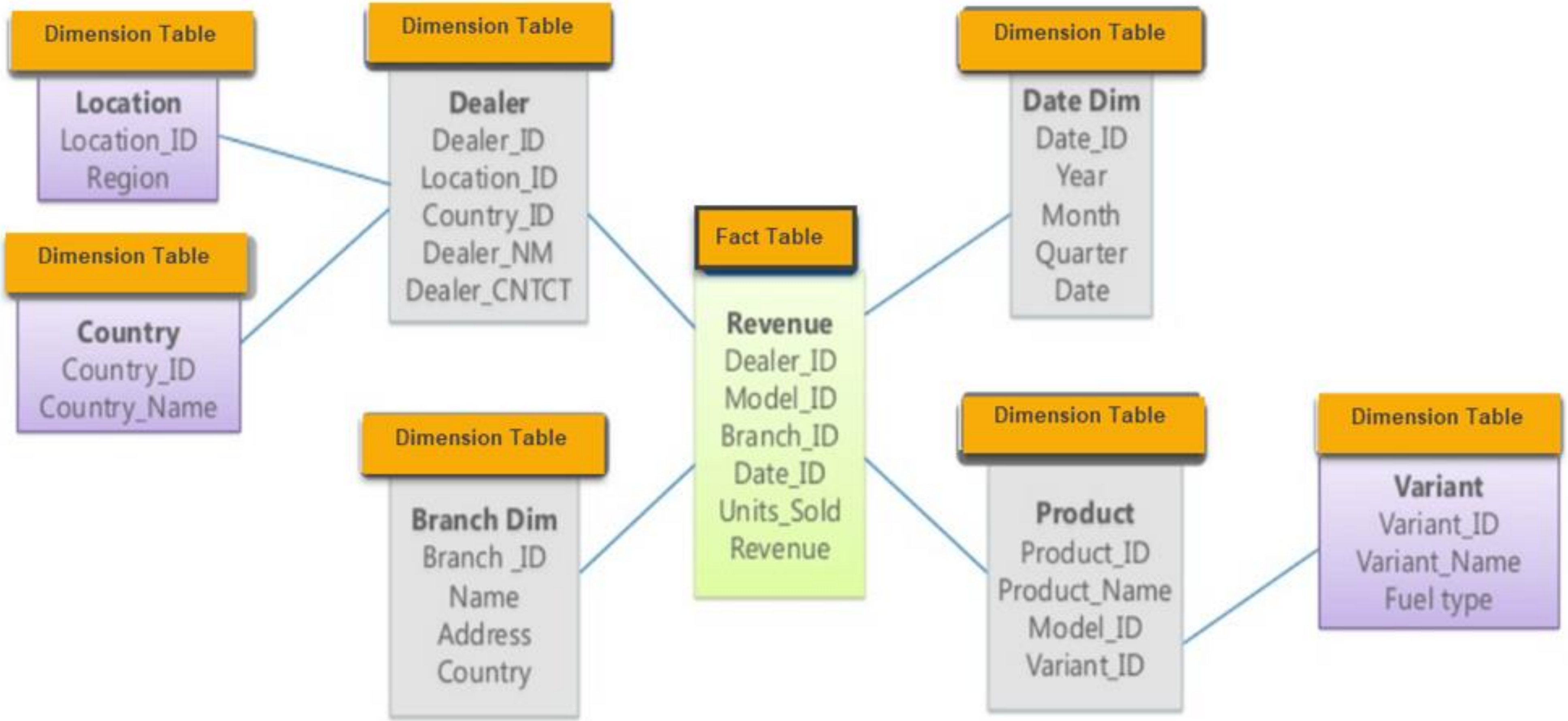
Performance Considerations for Star Schema

- Normalizing dimensional tables
  - Snowflake schema: dimension tables can have their own dimension tables
- Maintaining multiple fact tables to represent different aggregation levels
  - Save processor cycles at run time, thereby speeding up data analysis
- Denormalizing fact tables
  - Improves data access performance and saves data storage space
- Partitioning and replicating tables
  - Partitioning: splits tables into subsets of rows or columns and places them close to the client computer
  - Replication: makes copy of table and places it in a different location
  - Periodicity: provides information about the time span of the data stored in the table

Snowflake Schema



https://www.guru99.com/star-snowflake-data-warehousing.html



OLAP - Multidimensional Data Analysis Techniques

- Data are processed and viewed as part of a multidimensional structure
  - Particularly attractive to business decision makers who tend to view business data as being related to other business data
- Augmented advanced functions
  - Data presentation
  - Data aggregation, consolidation & classification
  - Computational
  - Data-modeling

Table name: DW_INVOICE

Database name: Ch13_Text

INV_NUM	INV_DATE	CUS_NAME	INV_TOTAL
2034	15-May-18	Dartonik	1400.00
2035	15-May-18	Summer Lake	1200.00
2036	16-May-18	Dartonik	1350.00
2037	16-May-18	Summer lake	3100.00
2038	16-May-18	Trydon	400.00

Operational Data

Table name: DW_LINE

INV_NUM	LINE_NUM	PROD_DESCRIPTION	LINE_PRICE	LINE_QUANTITY	LINE_AMOUNT
2034	1	Optical Mouse	45.00	20	900.00
2034	2	Wireless RF remote and laser pointer	50.00	10	500.00
2035	1	Everlast Hard Drive, 60 GB	200.00	6	1200.00
2036	1	Optical Mouse	45.00	30	1350.00
2037	1	Optical Mouse	45.00	10	450.00
2037	2	Roadster 56KB Ext. Modem	120.00	5	600.00
2037	3	Everlast Hard Drive, 60 GB	205.00	10	2050.00
2038	1	NoTech Speaker Set	50.00	8	400.00

Multidimensional View of Sales
(using MS Excel PivotTable)

Time Dimension

Multidimensional View of Sales	INV_DATE		
CUS_NAME	15-May-18	16-May-18	Grand Total
Dartonik	$ 1,400.00	$ 1,350.00	$ 2,750.00
Summer Lake	$ 1,200.00	$ 3,100.00	$ 4,300.00
Trydon		$ 400.00	$ 400.00
Grand Total	$ 2,600.00	$ 4,850.00	$ 7,450.00

Customer Dimension

Sales are located in the intersection of a customer row and date (time) column

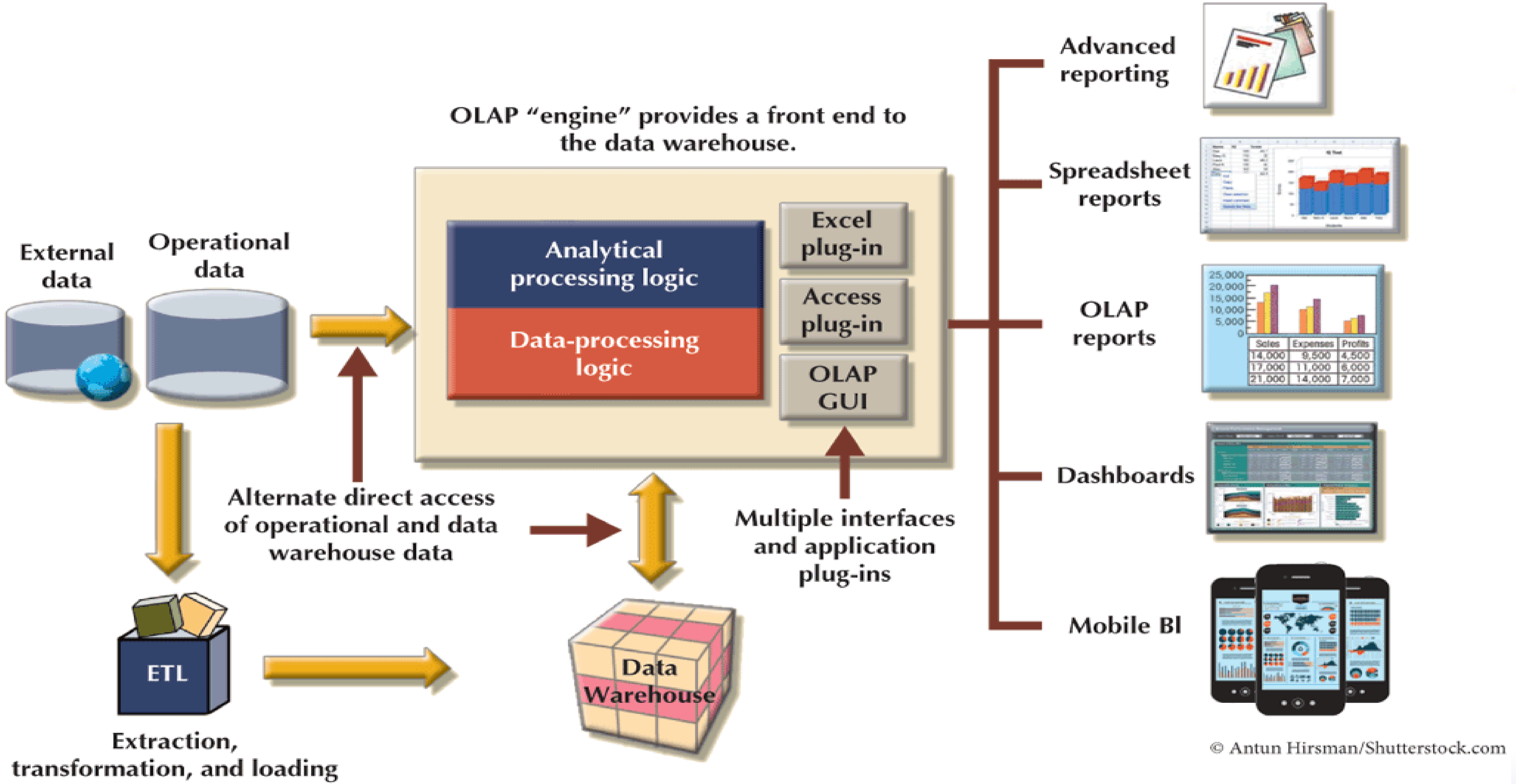
Aggregations (grand total sales) are provided for both dimensions (time and customer)

OLAP - Advanced Database Support

- OLAP tools must have the following features to deliver efficient decision support:
  - Access to many different kinds of DBMSs, flat files, and internal and external data sources
  - Access to aggregated data warehouse data and operational database detail data
  - Advanced data navigation features
  - Rapid and consistent query response times
  - Ability to map end-user requests
  - Support for very large databases

OLAP Architecture

- Designed to meet ease-of-use requirements while keeping the system flexible
- Main architectural components
  - Graphical user interface (GUI)
  - Analytical processing logic
  - Data-processing logic


OLAP "engine" provides a front end to the data warehouse.
External data
Operational data
Analytical processing logic
Data-processing logic
Excel plug-in
Access plug-in
OLAP GUI
Alternate direct access of operational and data warehouse data
Multiple interfaces and application plug-ins
ETL
Extraction, transformation, and loading
Data Warehouse
Advanced reporting
Spreadsheet reports
OLAP reports
25,000
20,000
15,000
10,000
5,000
0
Sales
Expenses
Profits
14,000
9,500
4,500
17,000
11,000
6,000
21,000
14,000
7,000
Dashboards
Mobile BI

Which database is best for OLTP and OLAP queries

Oracle Exadata: Exadata is an integrated Oracle Hardware & Software solution that integrates the Oracle Database into a customized hardware solution that indeed handles OLTP and OLAP well at the same time. How? Tons of flash, dedicated storage servers, dynamic indexes, ridiculous amounts of memory, etc.

SAP Hana: entirely memory based, hyper expensive solution that essentially provides terabytes of in-memory to allow for lightning fast OLAP queries driven through SAP Business Objects front ends. Because of the memory-resident database, OLTP works pretty fast too, and even those queries that are not housed in-memory get written to Sybase IQ columnar databases.

MemSQL: MemSQL (SingleStore) runs a good chunk of its operations in memory, accelerating typical OLTP solutions. But those data that are aged out then get written to columnar data stores optimized for OLAP.

OLAP for NoSQL DB?

1. there is no SUM() function in Neo4j, Cassandra, or hBase.

2. MongoDB released the MongoDB connector for BI, which acts as a MySQL server on top of MongoDB data

3. Work in progress

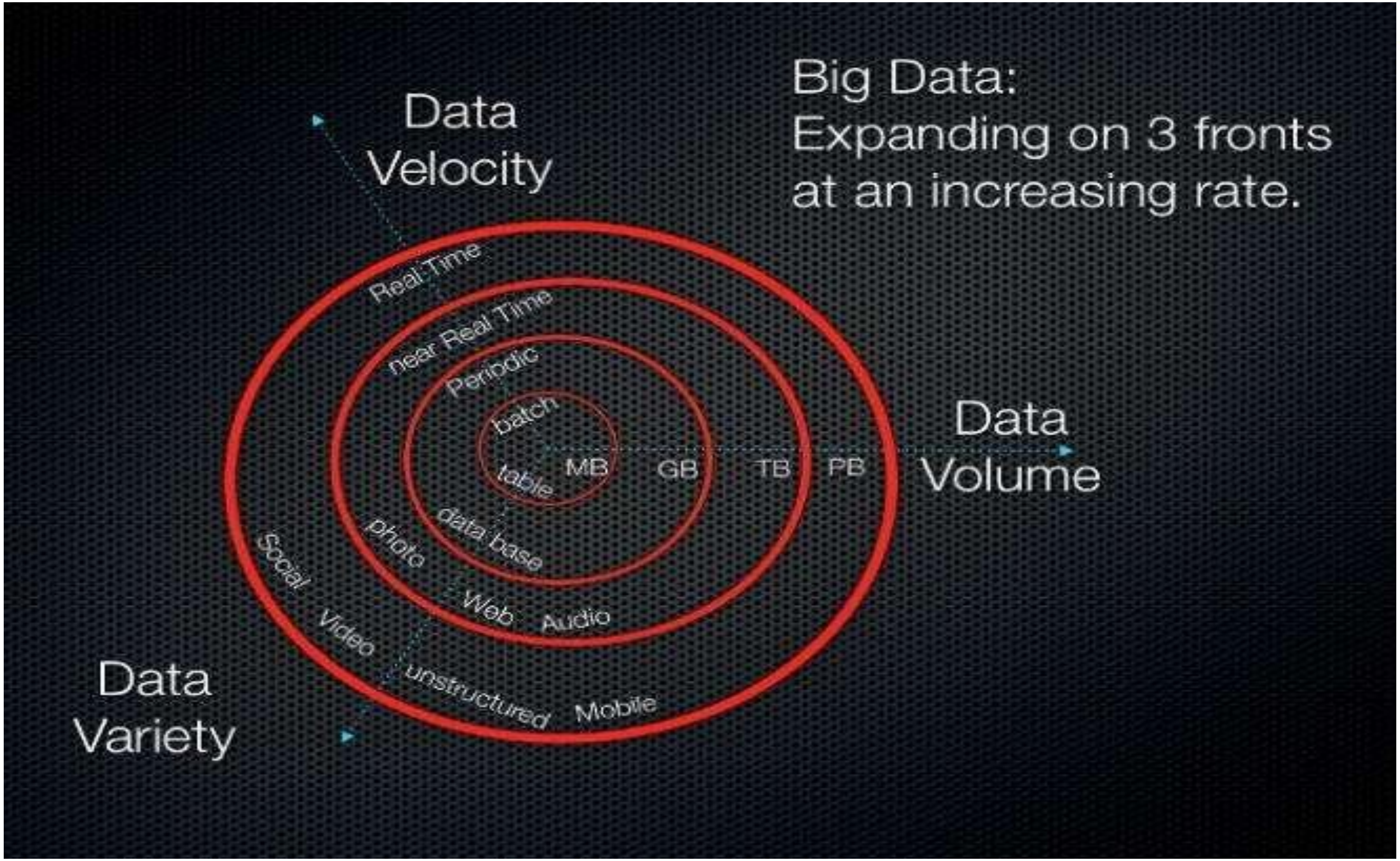
Need to collect any data

Harness the growing and changing nature of data

- Challenge is combining transactional data stored in relational databases with less structured data
- *Big Data = All Data*
- Get the right information to the right people at the right time in the right format

1 NEW DEFINITION IS ADDED ON urban
1,600+ READS ON Scribd.
13,000+ HOURS MUSIC STREAMING ON PANDORA
12,000+ NEW ADS POSTED ON craigslist
370,000+ MINUTES VOICE CALLS ON skype
98,000+ TWEETS
20,000+ NEW POSTS ON tumblr.
THE LARGEST SOCIAL READING PUBLISHING COMPANY!!
New Craigslist Ads
320+ NEW twitter ACCOUNTS
100+ NEW LinkedIn ACCOUNTS
13,000+ iPhone APPLICATIONS DOWNLOADED
IN 60 SECONDS...
1 associatedcontent NEW ARTICLE IS PUBLISHED
THE WORLD'S LARGEST COMMUNITY CREATED CONTENT!!
100+ Answers.com
40+ YAHOO! ANSWERS
QUESTIONS ASKED ON THE INTERNET...
6,600+ NEW PICTURES ARE UPLOADED ON flickr
600+ NEW VIDEOS
You Tube
50+ WordPress DOWNLOADS
25+ HOURS TOTAL DURATION
70+ DOMAINS REGISTERED
60+ NEW BLOGS
168 MILLION EMAILS ARE SENT
694,445 SEARCH QUERIES
1,700+ Firefox DOWNLOADS
695,000+ facebook STATUS UPDATES
125+ PLUGIN DOWNLOADS
1,500+ BLOG POSTS
79,364 WALL POSTS
510,040 COMMENTS
Google
Google Search
GO-Globe.com

The three V's



James Serra: Big Data Evangelist at Microsoft

New big data thinking: All data has value

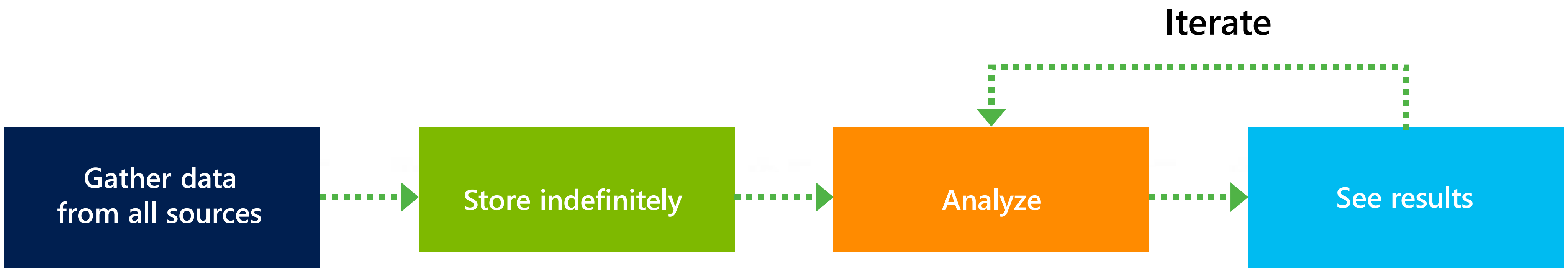
Use a data lake:

All data has potential value Data hoarding

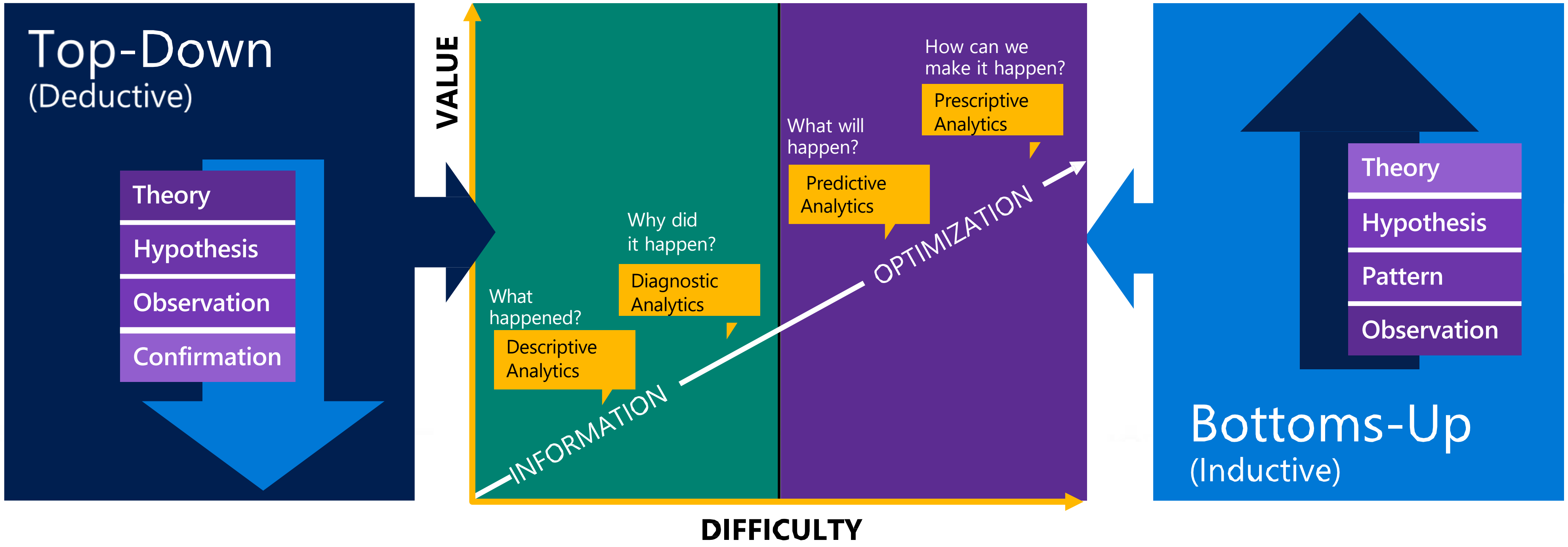
No defined schema—stored in native format

Schema is imposed and transformations are done at query time *(schema-on-read)*.

Apps and users interpret the data as they see fit



Two Approaches to getting value out of data: Top-Down + Bottoms-Up

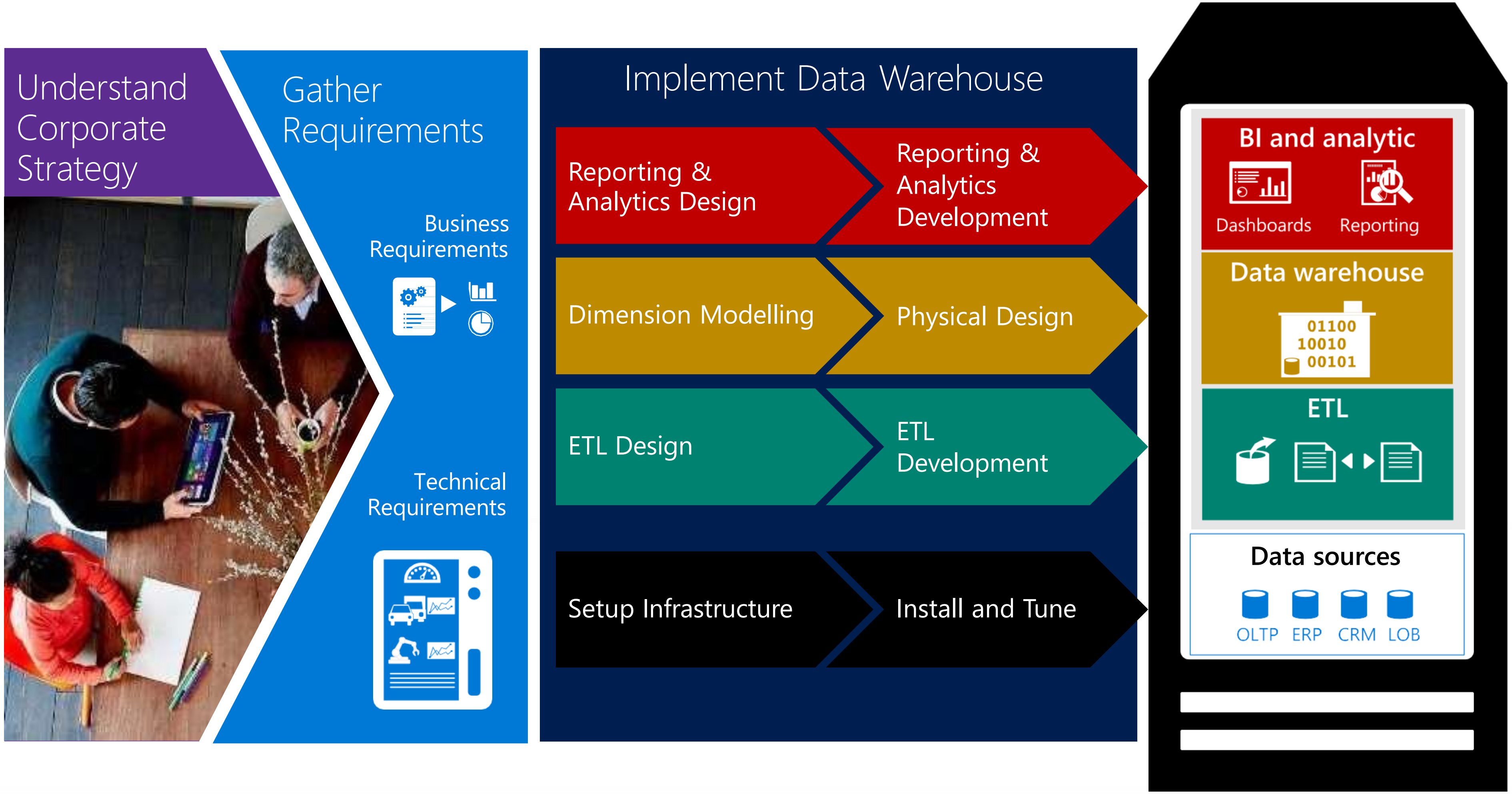


- Know the questions to ask
- Lot's of upfront work to get the data to where you can use it
- Model first

- Don't know the questions to ask
- Little upfront work needs to be done to start using data
- Model later

James Serra: Big Data Evangelist at Microsoft

Data Warehousing Uses A Top-Down Approach



James Serra: Big Data Evangelist at Microsoft

The “data lake” Uses A Bottoms-Up Approach

Ingest all data
regardless of requirements

Store all data
in native format without schema definition

Do analysis
Using analytic engines like Hadoop

James Serra: Big Data Evangelist at Microsoft

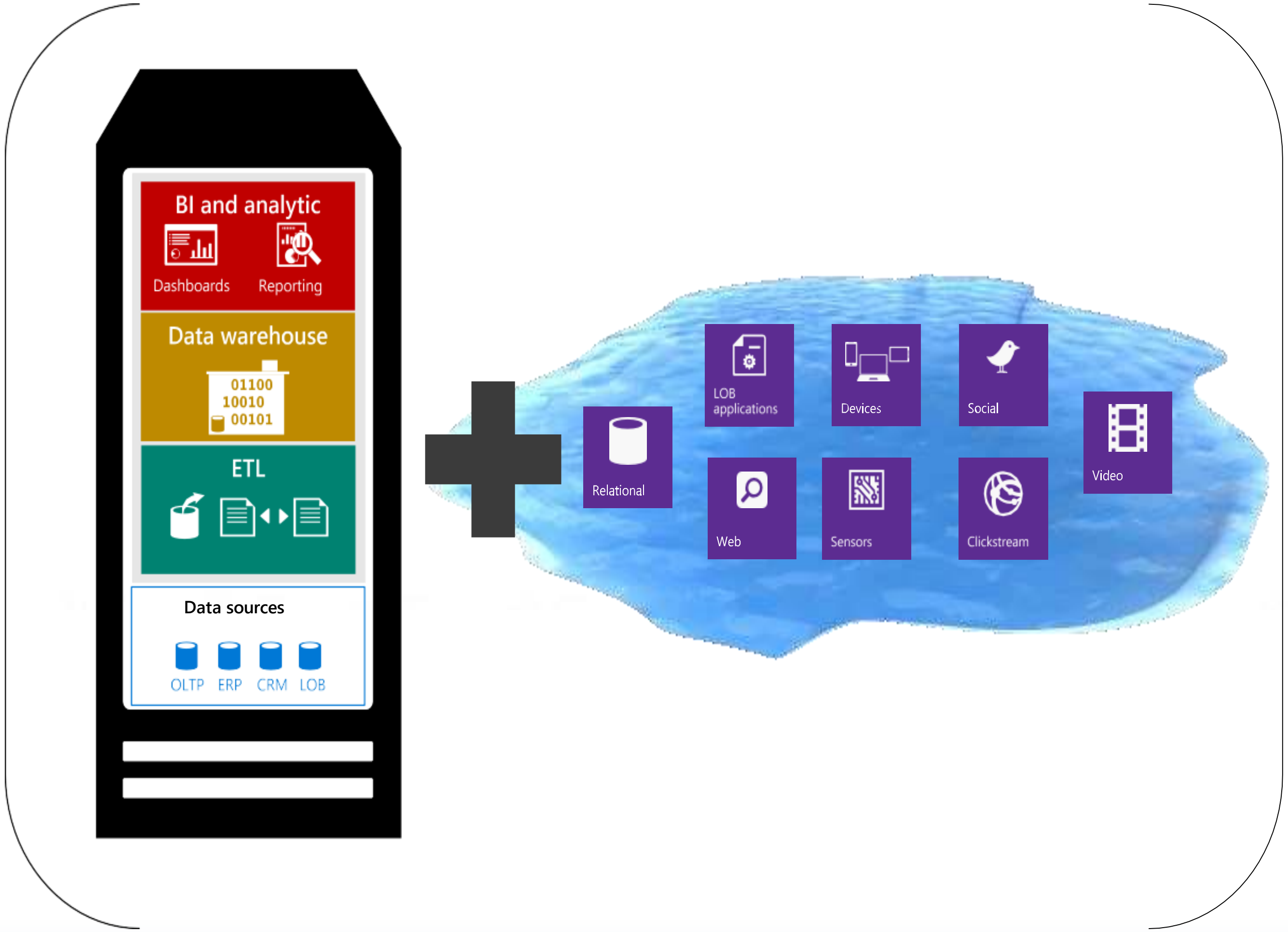
Data Lake + Data Warehouse Better Together

What happened?

Descriptive Analytics

Why did it happen?

Diagnostic Analytics



What will happen?

Predictive Analytics

How can we make it happen

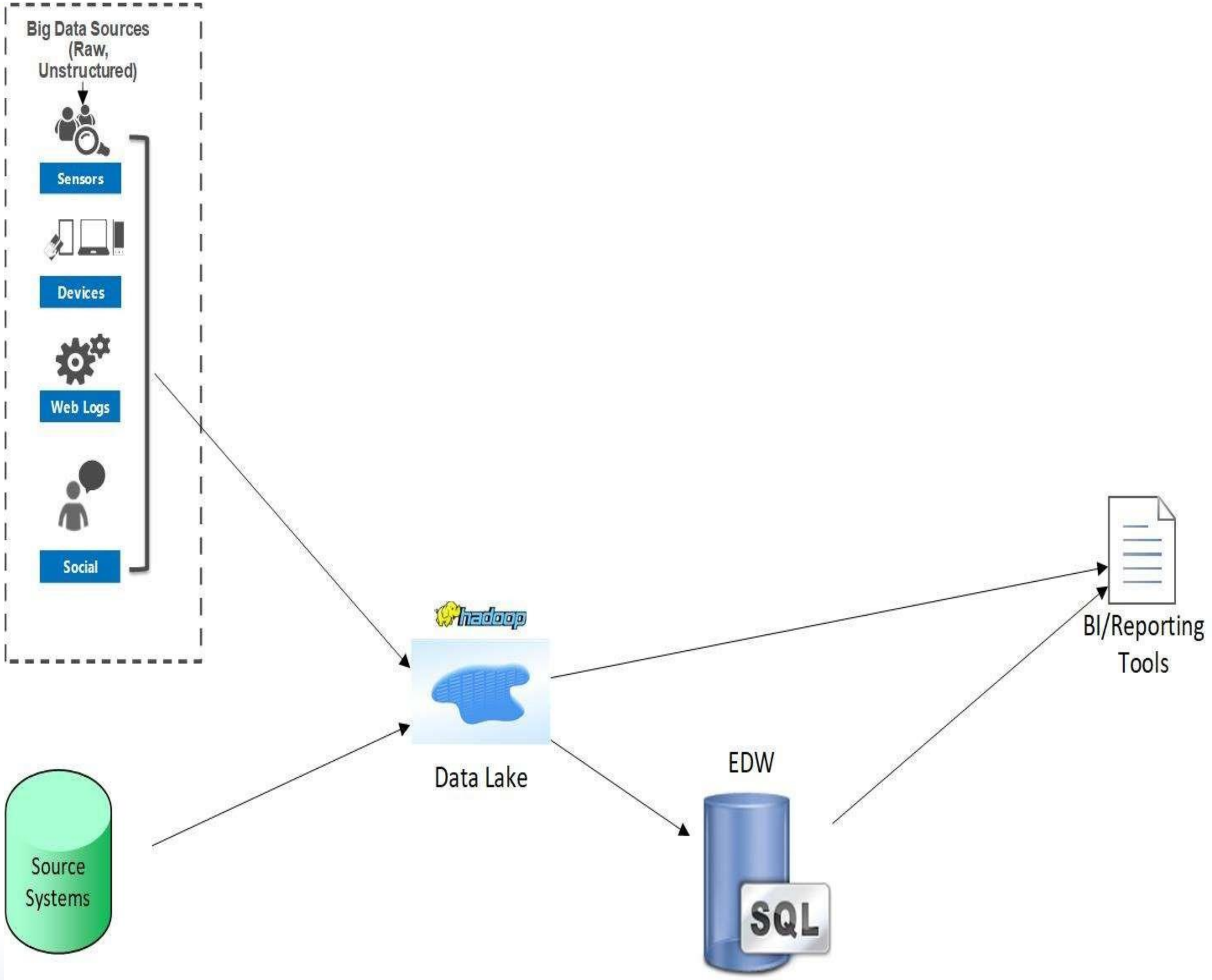
Prescriptive Analytics

James Serra: Big Data Evangelist at Microsoft

Modern Data Warehouse

•Supports future data needs
•Data harmonized and analyzed in the data lake or moved to EDW for more quality and performance

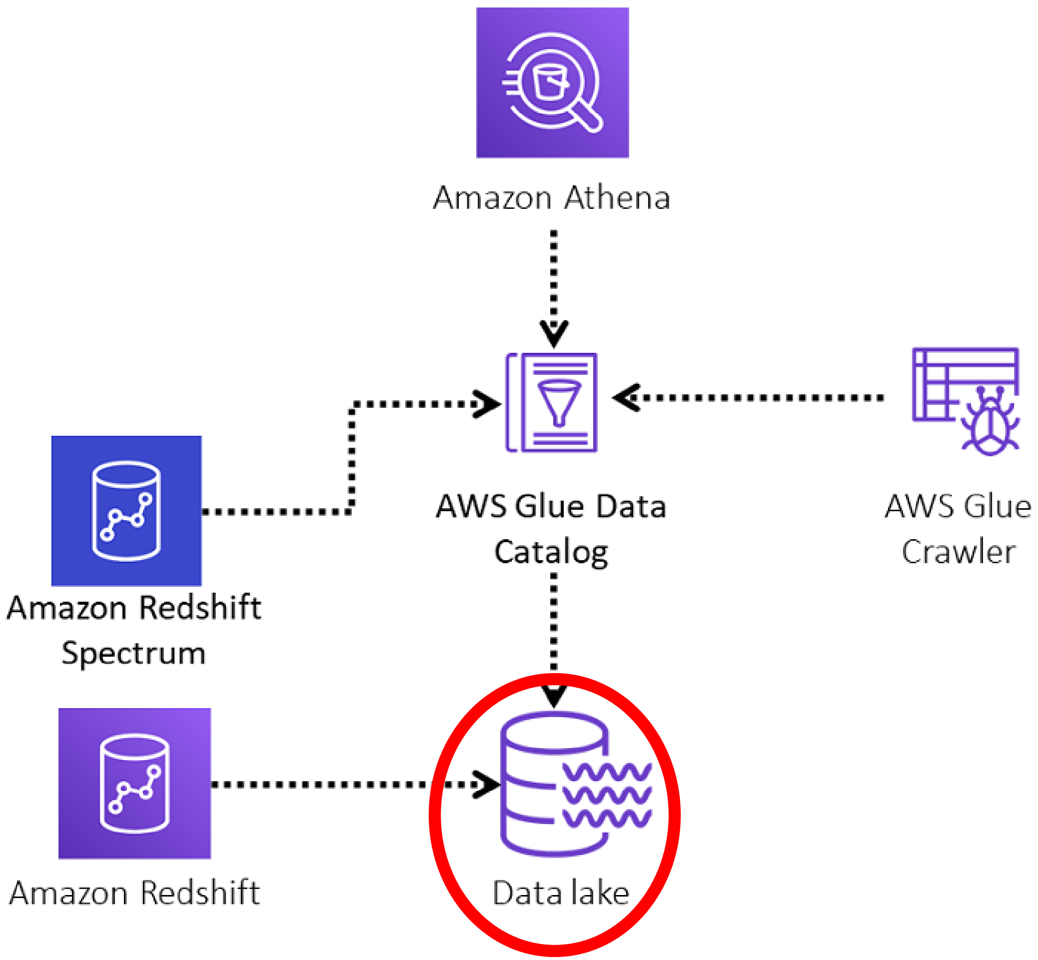
An enterprise data warehouse (EDW)



Data Lake

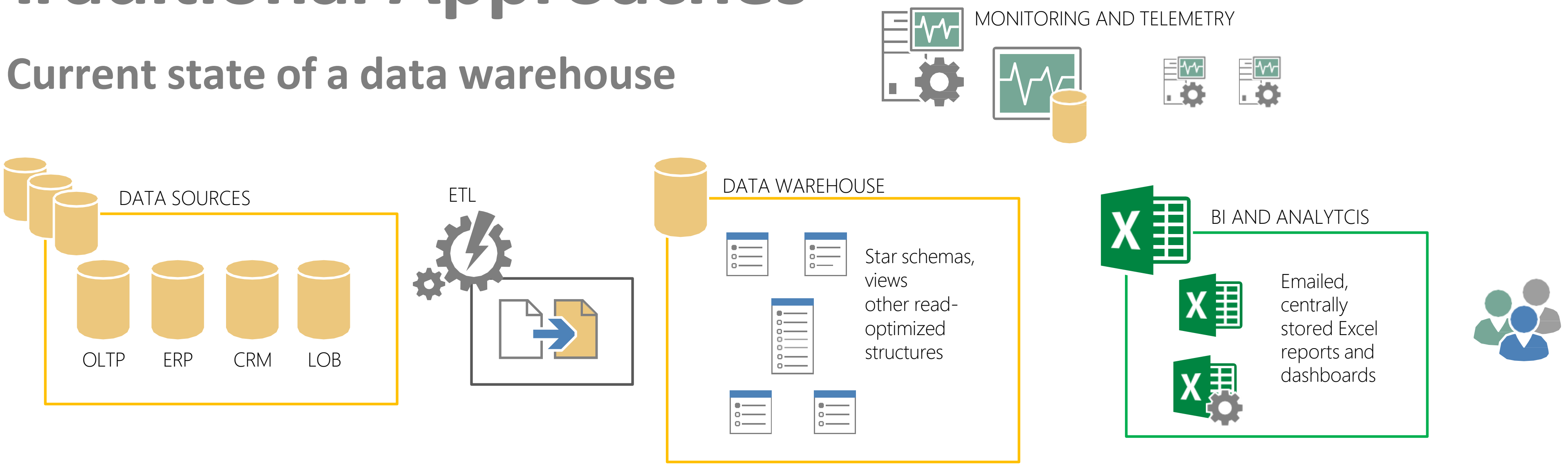
A storage repository, usually Hadoop, that holds a vast amount of raw data in its native format until it is needed.

- A data lake is a repository of data from disparate sources that is stored in its original, **raw format**
- data lakes store large amounts of current and historical data.
- What sets data lakes apart is their ability to store data in a variety of formats including JSON, BSON, CSV, TSV, Avro, ORC, and Parquet.
- A data lake is a repository for data stored in a variety of ways including databases
- Starburst, Presto, Dremio, and Atlas Data Lake can give a database-like view into the data stored in your data lake. In many cases, these tools can power the same analytical workloads as a data warehouse.
- **Amazon Redshift allows you to unload your data using a data lake export to an Apache Parquet file format**



Traditional Approaches

Current state of a data warehouse



Well manicured, often relational sources

Known and expected data volume and formats

Little to no change

Complex, rigid transformations

Required extensive monitoring

Transformed historical into read structures

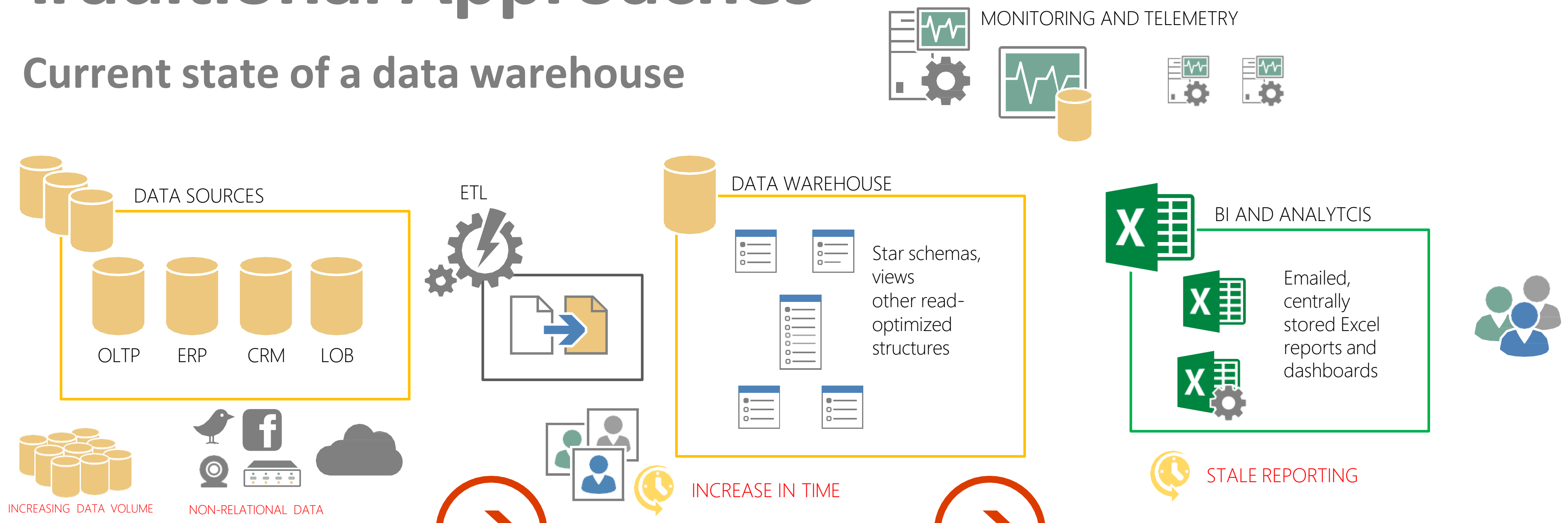
Flat, canned or multi-dimensional access to historical data

Many reports, multiple versions of the truth

24 to 48h delay

Traditional Approaches

Current state of a data warehouse



Increase in variety of data sources

Increase in data volume

Increase in types of data

Pressure on the ingestion engine

Complex, rigid transformations can't longer keep pace

Monitoring is abandoned

Delay in data, inability to transform volumes, or react to new sources

Repair, adjust and redesign ETL

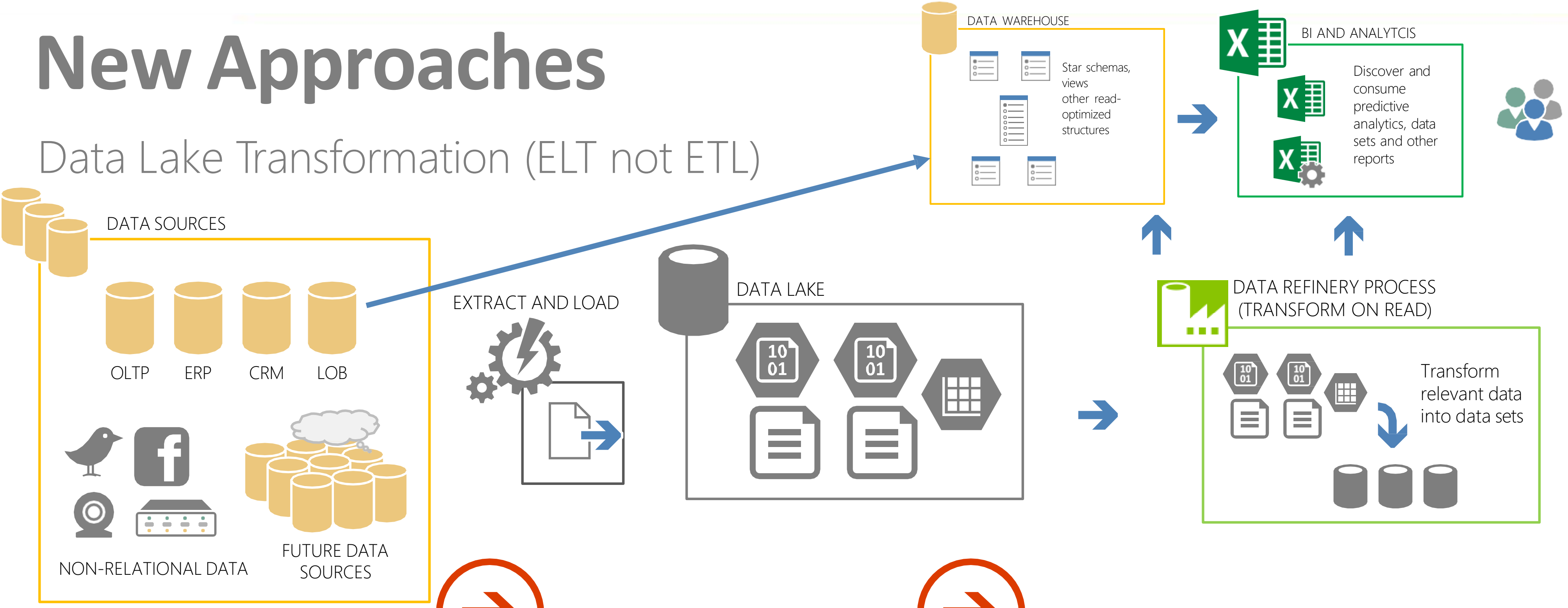
Reports become invalid or unusable

Delay in preserved reports increases

Users begin to “innovate” to relieve starvation

New Approaches

Data Lake Transformation (ELT not ETL)



All data sources are considered

Leverages the power of on-prem technologies and the cloud for storage and capture

Native formats, streaming data, big data

Extract and load, no/minimal transform

Storage of data in near-native format

Orchestration becomes possible

Streaming data accommodation becomes possible

Refineries transform data on read

Produce curated data sets to integrate with traditional warehouses

Users discover published data sets/services using familiar tools

Organizing a Data Lake

Raw Data Zone

- Subject Area
  - Data Source
    - Object
      - Date Loaded
        - File(s)

- Sales
  - Salesforce
    - CustomerContacts
      - 2016
        - 12
          - 01
            - CustContact_2016_12_01.txt

Example 1

Pros: Subject area at top level, organizationwide
Partitioned by time

Cons: No obvious security or organizational boundaries

Curated Data Zone

- Purpose
  - Type
    - Snapshot Date
      - File(s)

- Sales Trending Analysis
  - Summarized
    - 2016_12_01
      - SalesTrend_2016_12_01.txt

Thanks to Melissa Coates,
www.CoatesDataStrategies.com

CLOUD OLAP (BigQuery, Redshift, Snowflake)

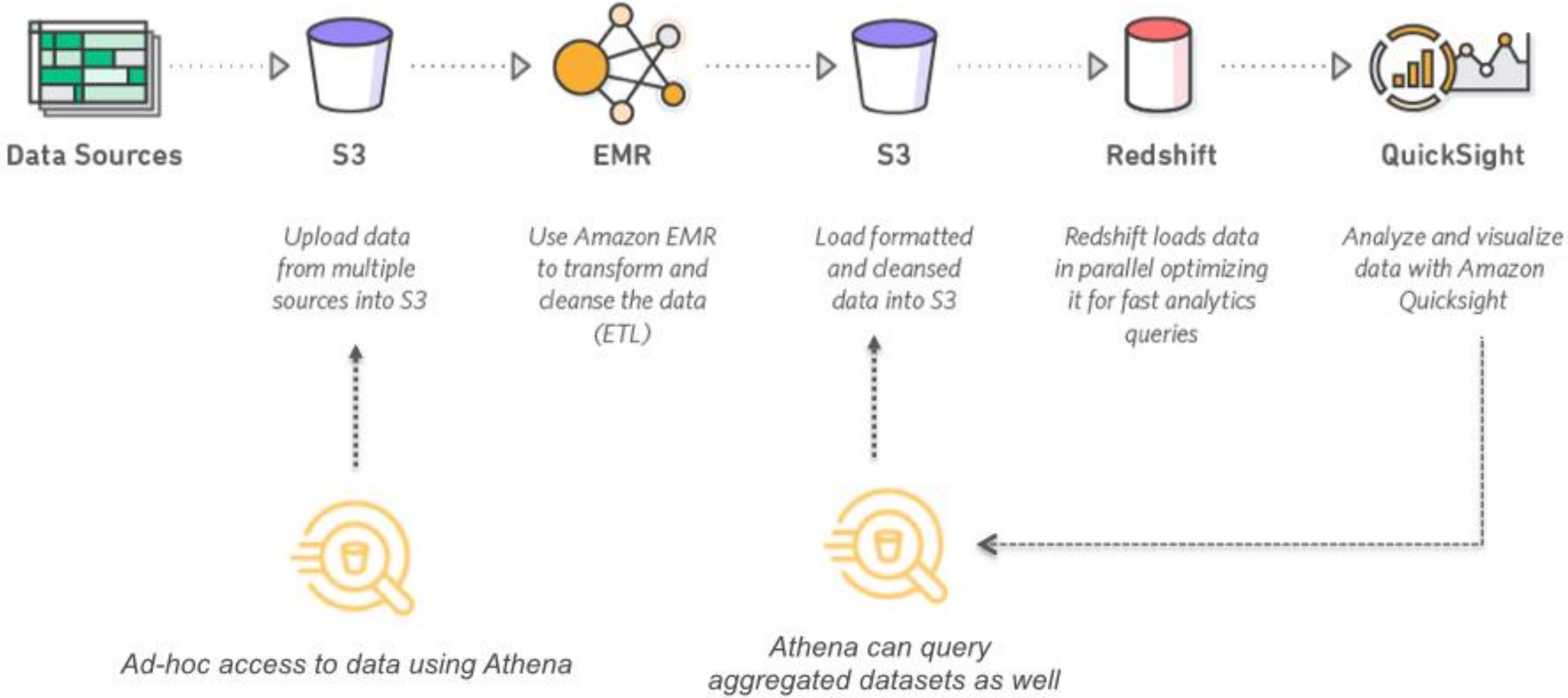
https://medium.com/2359media/redshift-vs-bigquery-vs-snowflake-a-comparison-of-the-most-popular-data-warehouse-for-data-driven-cb1c10ac8555

When should you use Athena?

Amazon Athena should be used to run **ad-hoc queries on Amazon S3** data sets using ANSI SQL. It can process structured, unstructured, and semi-structured data formats. It can also have data integration with BI tools or SQL clients using JDBC, or with QuickSight for easy visualizations.

When should you use Redshift?

It is recommended to use Amazon Redshift on large sets of structured data. Because it contains a number of replicas, it interacts with other nodes and rebuilds the drive. Redshift can be integrated with Tableau, Informatica, Microstrategy, Pentaho, SAS, and other BI Tools. It can be used for log analysis, clickstream events, and real-time data sets.



Redshift requires framework management and data preparation while Athena bypasses that and gets straight to querying data from Amazon S3

Comparing Athena to Redshift is not simple. Athena has an edge in terms of portability and cost, whereas Redshift stands tall in terms of performance and scale

Fast

Cost Efficient

Simple

Elastic

Secure

Compatible

Amazon Redshift

Fast, simple, cost-effective data warehousing.

Managed Massively Parallel Petabyte Scale Data Warehouse

Streaming Backup/Restore to S3

Load data from S3, DynamoDB and EMR

Extensive Security Features

Scale from 160 GB -> 2 PB Online

Amazon Redshift Cluster Architecture

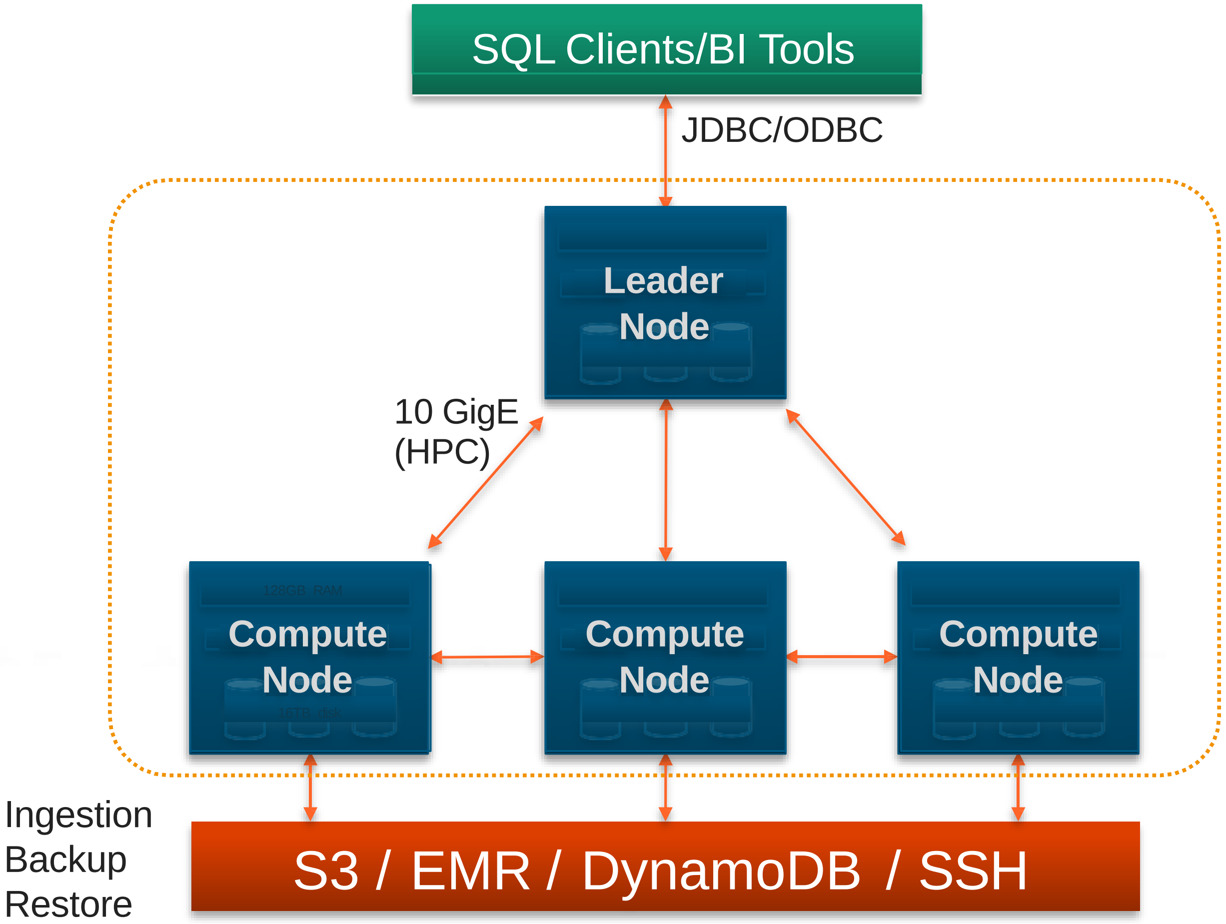
Massively parallel, shared nothing

Leader node

- SQL endpoint
- Stores metadata
- Coordinates parallel SQL processing

Compute nodes

- Local, columnar storage
- Executes queries in parallel
- Load, backup, restore
- 2, 16 or 32 slices



Reduced I/O = Enhanced Performance

Columnar storage

+

Large data block sizes

+

Data compression

+

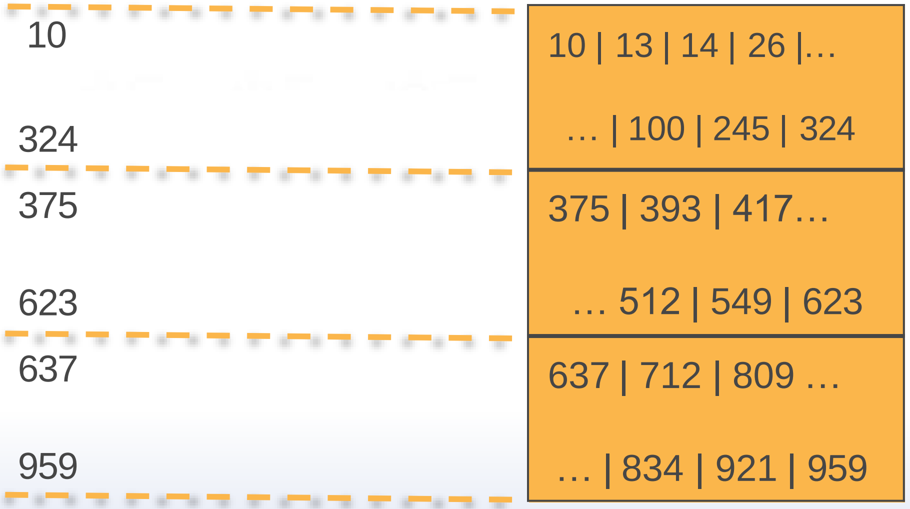
Zone maps (meta-data)

+

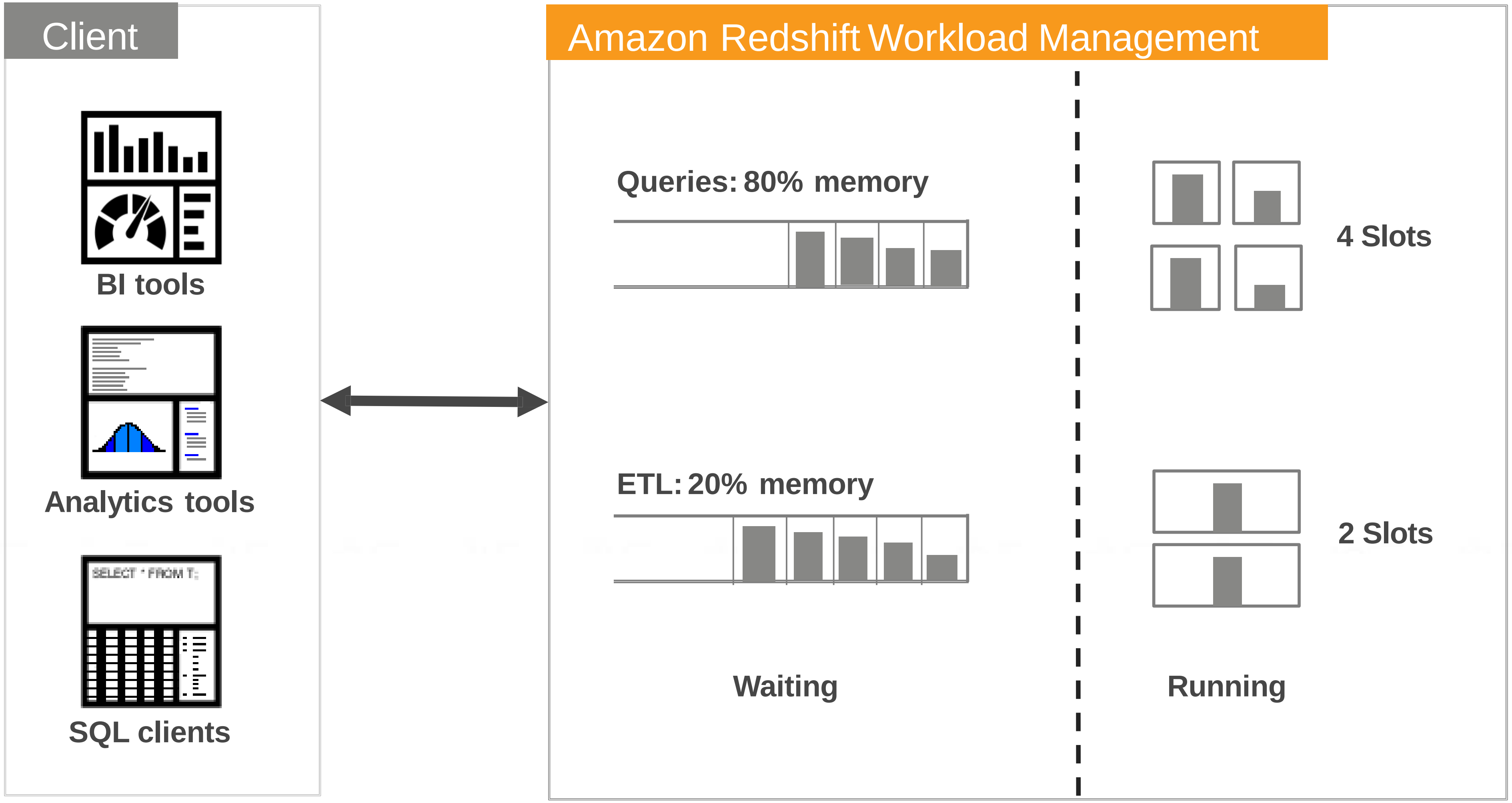
Direct-attached storage

```
analyze compression listing;

  Table   |        Column        | Encoding
----------+----------------------+----------
 listing  | listid               | delta
 listing  | sellerid             | delta32k
 listing  | eventid              | delta32k
 listing  | dateid               | bytedict
 listing  | numtickets           | bytedict
 listing  | priceperticket       | delta32k
 listing  | totalprice           | mostly32
 listing  | listtime             | raw
```



Workload Management



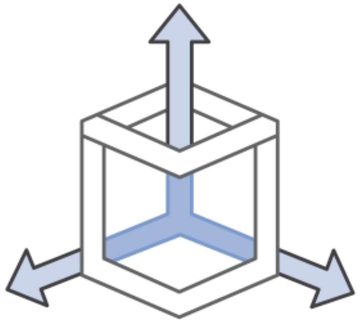
80/4 = **20% per slot**

20/2 = **10% per slot**

Amazon Redshift Spectrum

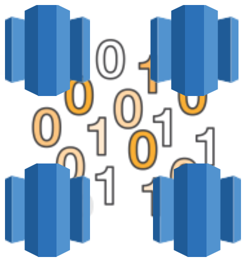
Run SQL queries directly against data in S3 using thousands of nodes

Fast @ exabyte scale



Elastic & highly available

On-demand, pay-per-query

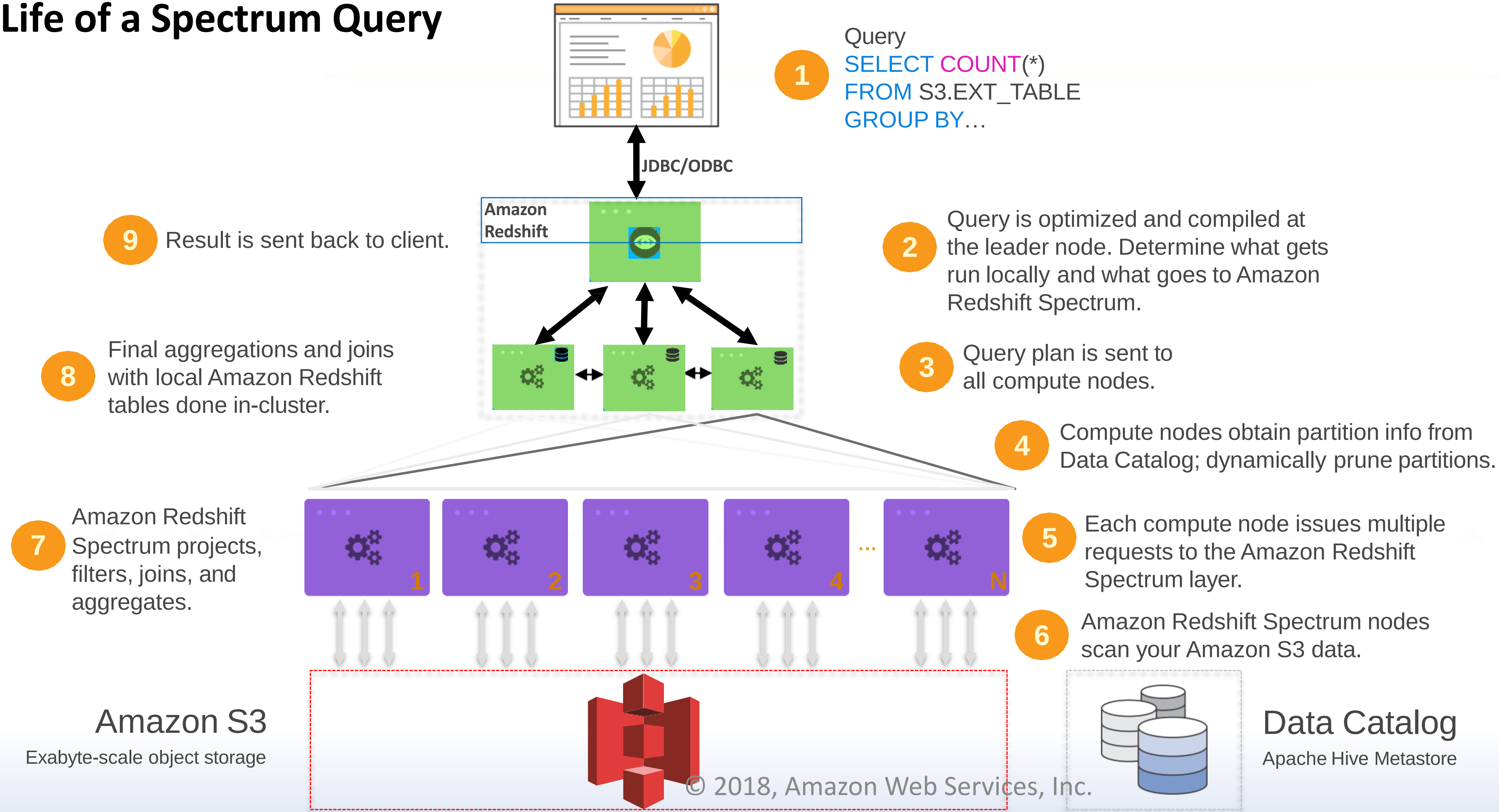


High concurrency: Multiple clusters access same data

No ETL: Query data in-place using open file formats

Full Amazon Redshift SQL support

Life of a Spectrum Query



Is Amazon Redshift Spectrum useful if I don't have an exabyte?

Your data will get bigger

- On average, data warehousing volumes grow 10x every 5 years
- The average Amazon Redshift customer doubles data each year

Amazon Redshift Spectrum makes data analysis simpler

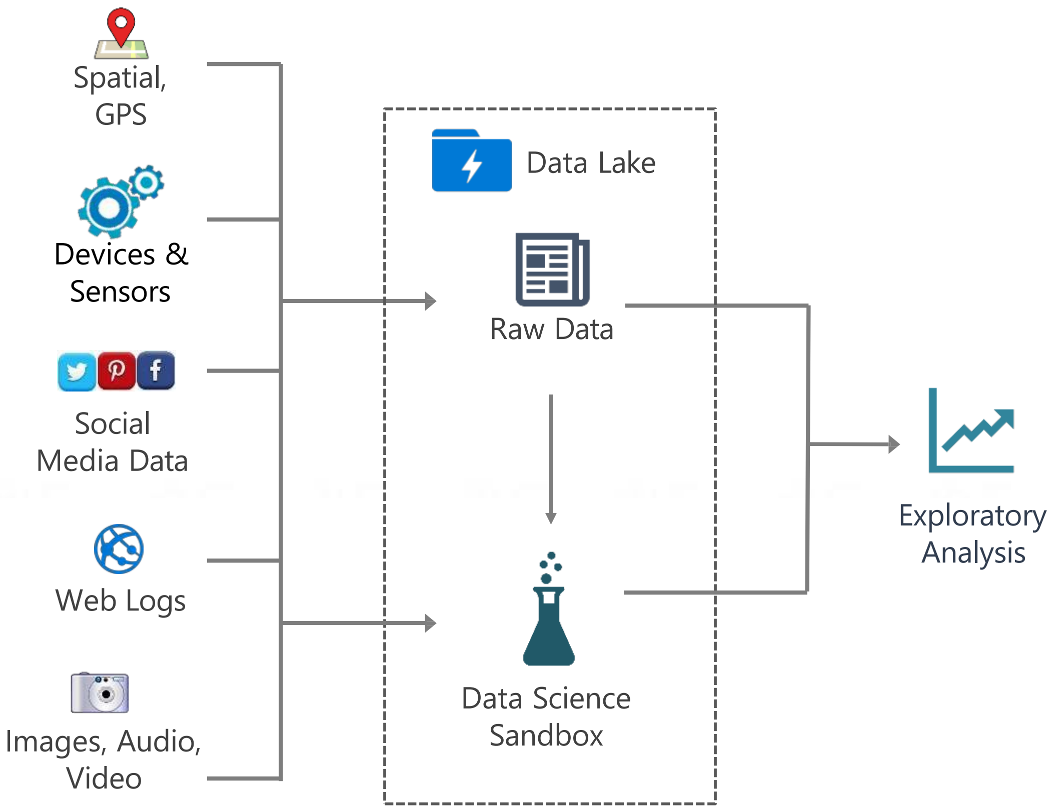
- Access your data without ETL pipelines
- Teams using Amazon EMR, Athena & Redshift can collaborate using the same data lake

Amazon Redshift Spectrum improves availability and concurrency

- Run multiple Amazon Redshift clusters against common data
- Isolate jobs with tight SLAs from ad hoc analysis

Data Lake Use Cases

Ingestion of New File Types

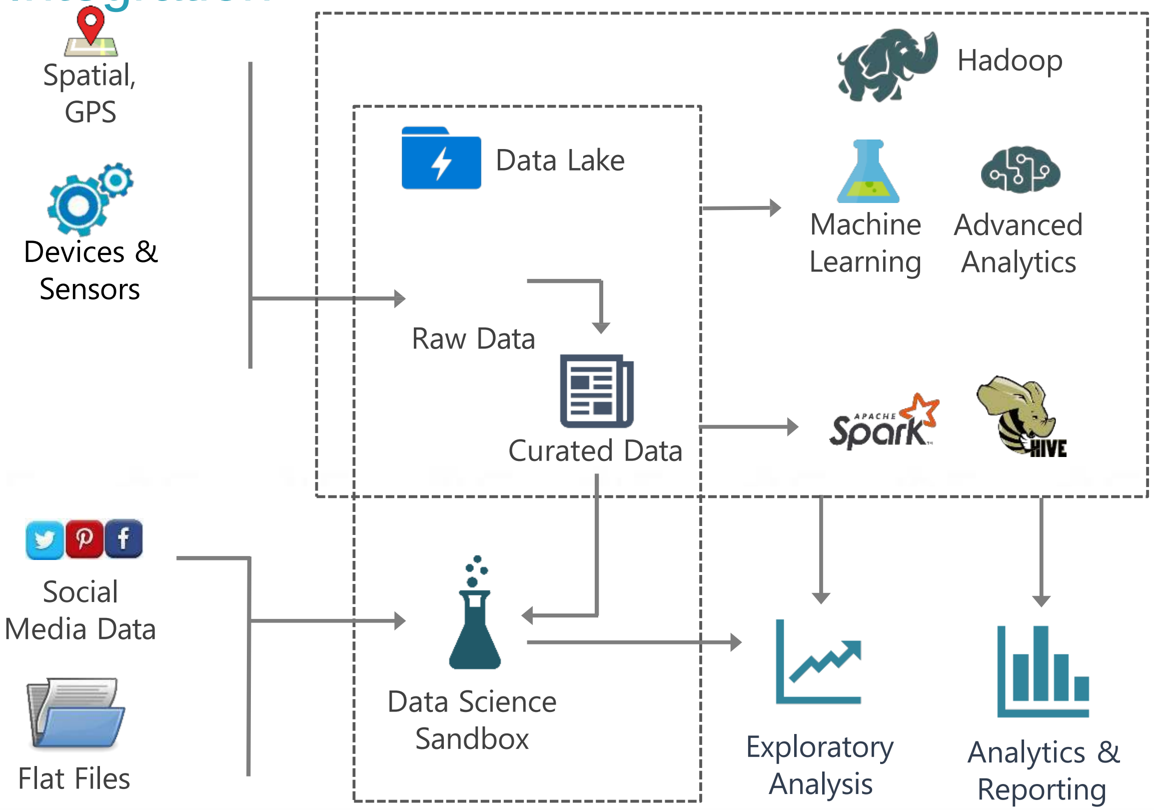


- ✓ Preparatory file storage for multi-structured data
- ✓ Exploratory analysis + POCs to determine value of new data types & sources
- ✓ Affords additional time for longer-term planning while accumulating data or handling an influx of data

Thanks to Melissa Coates, www.CoatesDataStrategies.com

Data Lake Use Cases

Data Science Experimentation | Hadoop Integration

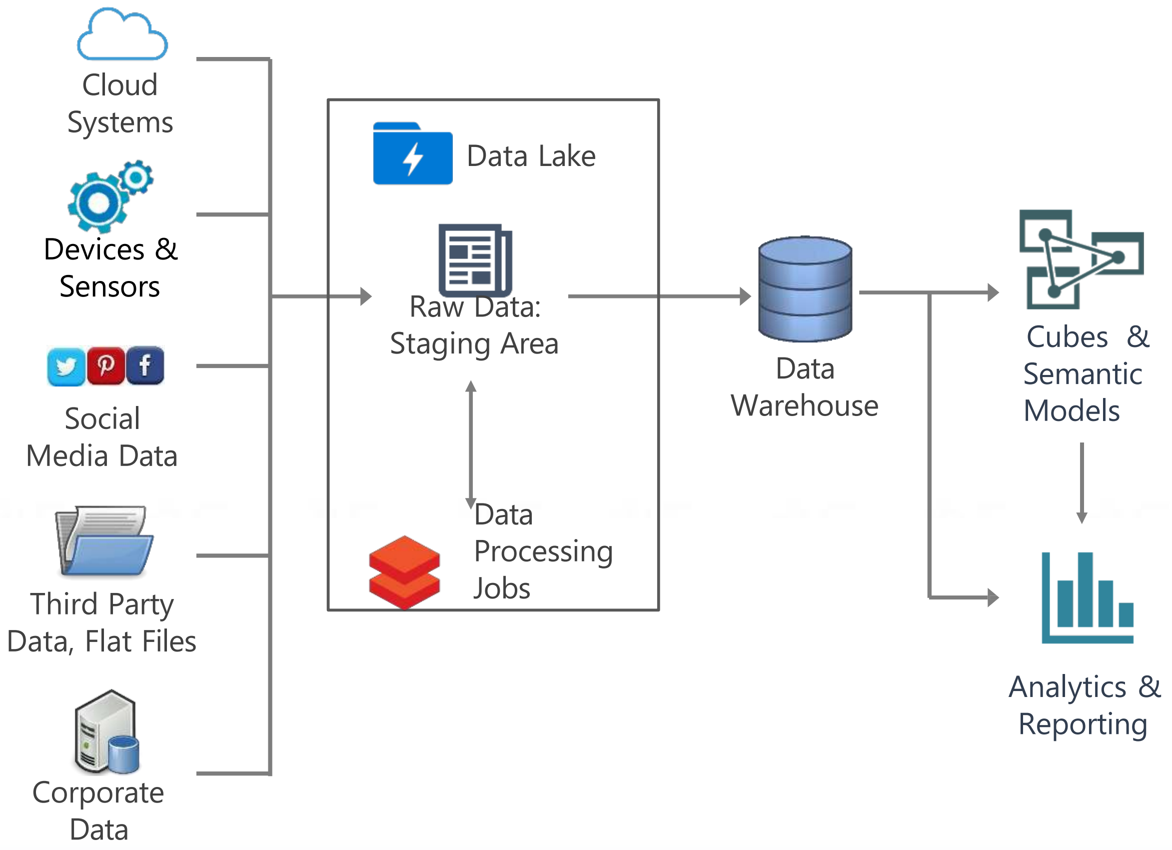


- ✓ Sandbox solutions for initial data prep, experimentation, and analysis
- ✓ Migrate from proof of concept to operationalized solution
- ✓ Integrate with open source projects such as Hive, Pig, Spark, Storm, etc.
- ✓ Big data clusters
- ✓ SQL-on-Hadoop solutions

Thanks to Melissa Coates, www.CoatesDataStrategies.com

Data Lake Use Cases

Data Warehouse Staging Area

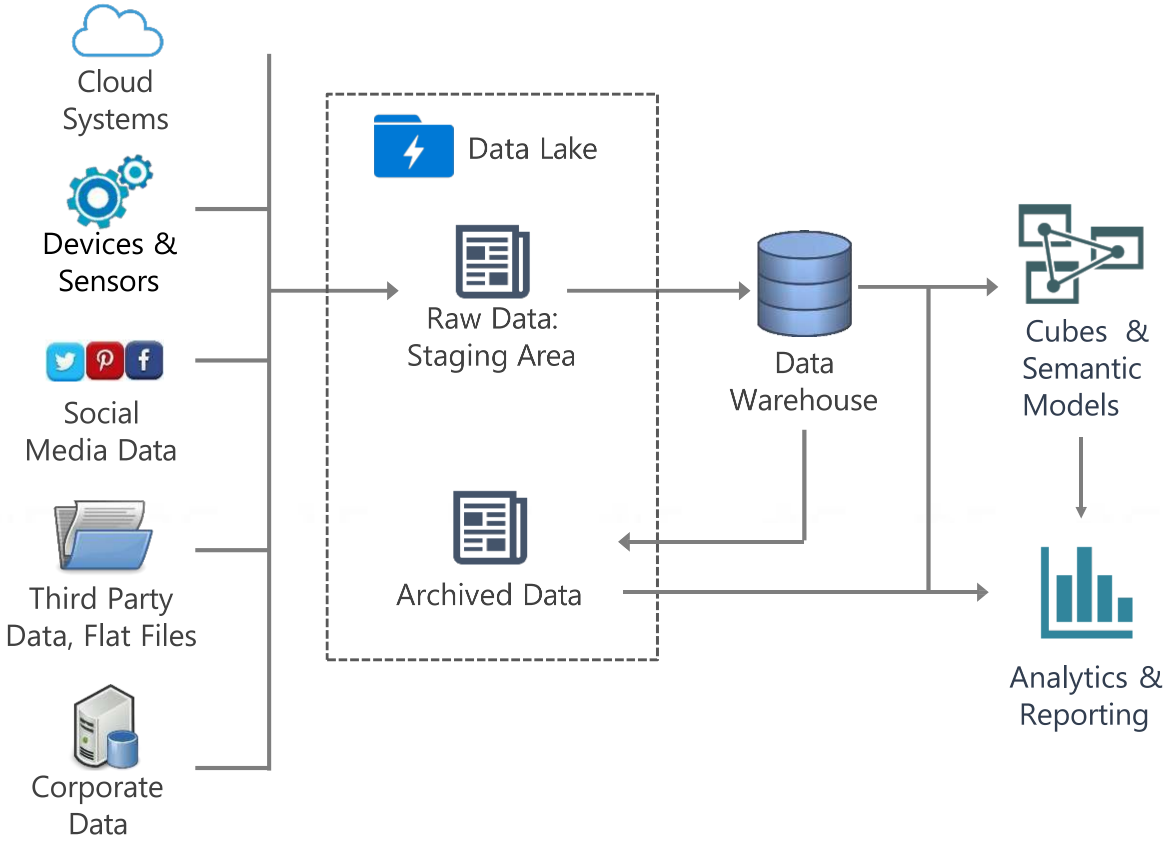


- ✓ELT strategy
- ✓Reduce storage needs in relational platform by using the data lake as landing area
- ✓Practical use for data stored in the data lake
- ✓Potentially also handle transformations in the data lake

Thanks to Melissa Coates, www.CoatesDataStrategies.com

Data Lake Use Cases

Integration with DW | Data Archival | Centralization

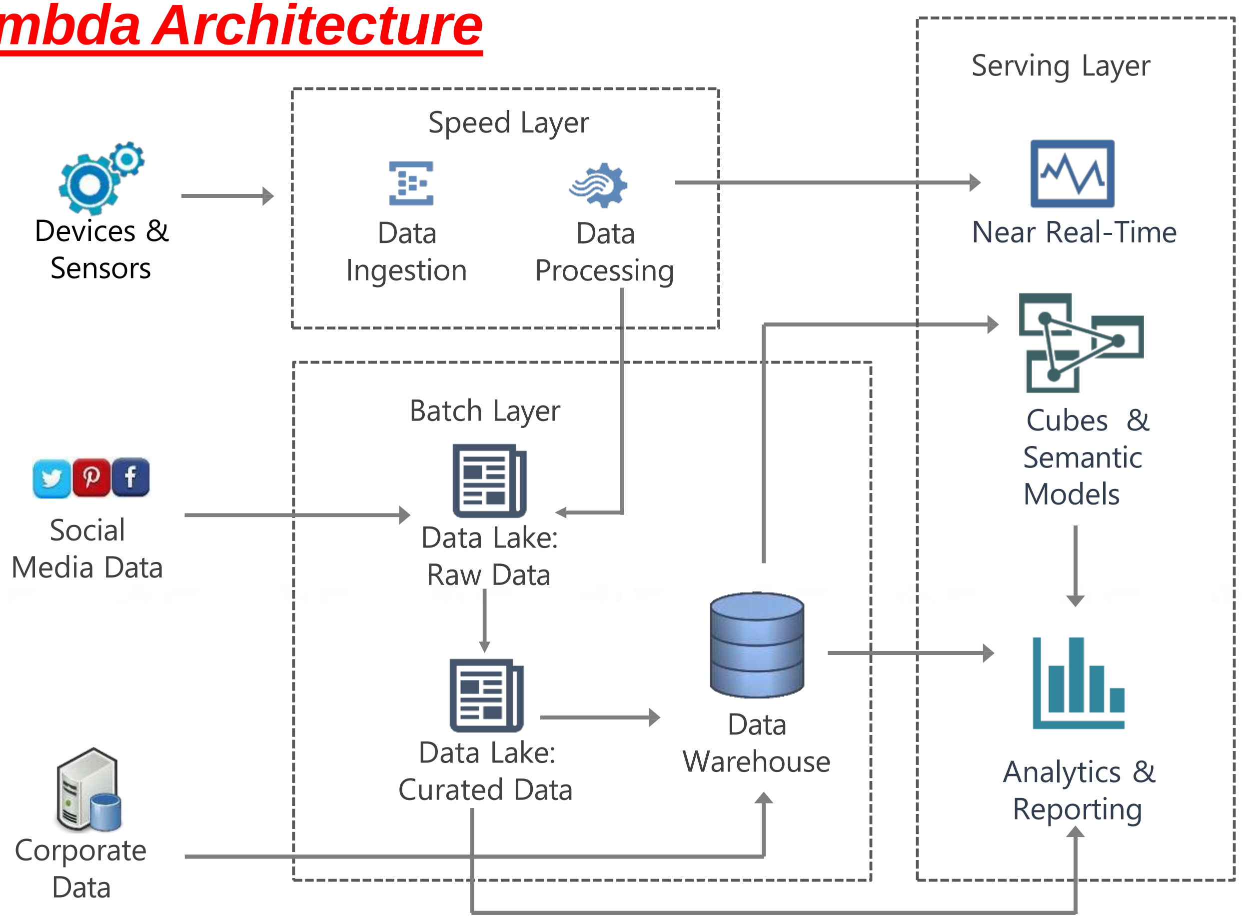


- ✓ Grow around existing DW
- ✓ Aged data available for querying when needed
- ✓ Complement to the DW via data virtualization
- ✓ Federated queries to access current data (relational DB) + archive (data lake)

Thanks to Melissa Coates, www.CoatesDataStrategies.com

Data Lake Use Cases

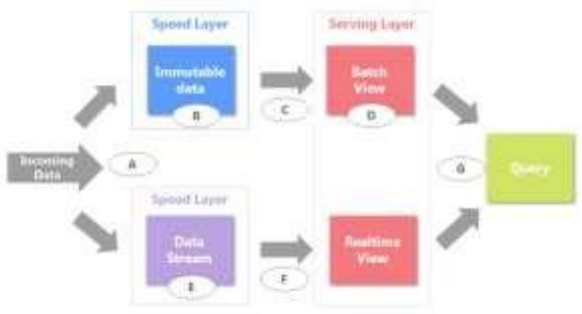
Lambda Architecture



- ✓ Support for low-latency, high-velocity data in near real time
- ✓ Support for batch-oriented operations

Thanks to Melissa Coates, www.CoatesDataStrategies.com

Lambda Architecture



- All data is sent to **both** the **batch and speed layer**
- Master data set is an **immutable, append-only** set of data
- Batch layer **pre-computes** query functions from scratch, result is called Batch Views. Batch layer **constantly re-computes** the batch views.
- Batch views are **indexed** and **stored** in a **scalable database** to get particular values very quickly. Swaps in new batch views when they are available
- Speed layer **compensates** for the high latency of updates to the Batch Views
- Uses fast **incremental algorithms** and read/write databases to produce real- time views
- Queries are resolved by getting results from **both** batch and real-time views