DATABASE TECHNOLOGIES

Design and Implementation of Databases Systems

Suresh Jamadagni

Department of Computer Science and Engineering

DATABASE TECHNOLOGIES

Overview of Data Mining Technology

Suresh Jamadagni
Department of Computer Science and Engineering

DATABASE TECHNOLOGIES

Overview of Data Mining Technology

- Data mining is the process of discovering meaningful correlations, patterns and trends by sifting through large amounts of data stored in repositories.

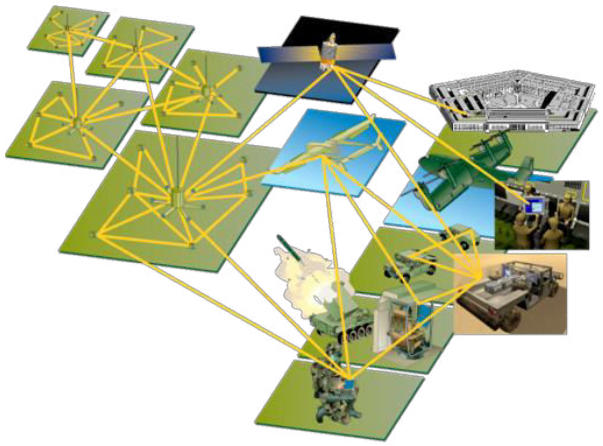
- Data mining employs pattern recognition technologies as well as statistical and mathematical techniques.

DATABASE TECHNOLOGIES

Overview of Data Mining Technology

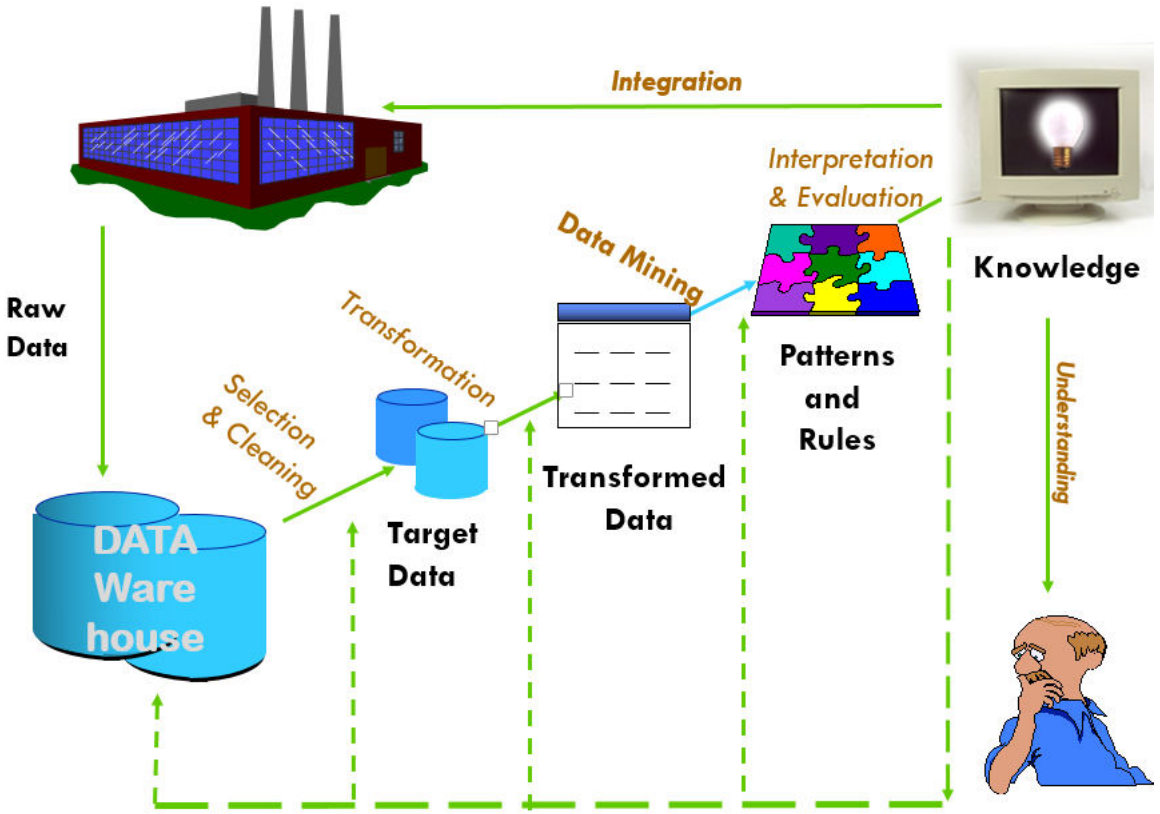
Why Mine Data ?????

1. Data collected and stored at enormous speeds (GB/hour)
    - Remote Sensors on a satellite
    - Telescopes scanning the skies
    - Microarray's generating gene expression data
    - Scientific simulations generating terabytes of data
2. Traditional Techniques infeasible for raw data
3. Data Mining may help scientists in classifying and segmenting data and in Hypothesis formulation.



DATABASE TECHNOLOGIES

Data Mining – Knowledge Discovery Process



Data Mining in Knowledge Discovery Process

Result of mining may be to discover the following types of new information:

- **Association Rules:** customer buys video equipment, he/she also buys another electronic gadget
- **Sequential Patterns :** customer buys a camera, and within three months he/she buys photographic supplies, then within six months he/she is likely to buy an accessory item
- **Classification Trees :** Customers may be classified by frequency of visits, types of financing used, amount of purchase, or affinity for types of items

Goals of Data Mining in Knowledge Discovery

- **Prediction:** Analysis of buying transactions to predict what consumers will buy under certain discounts.
- **Identification:** Intruders trying to break a system may be identified by the programs executed, files accessed, and CPU time per session
- **Classification:** Customers in a supermarket can be categorized into discount- seeking shoppers, shoppers in a rush, loyal regular shoppers, shoppers attached to name brands, and infrequent shoppers.
- **Optimization:** Optimize the use of limited resources such as time, space, money, or materials and to maximize output variables such as sales or profits under a given set of constraints.

DATABASE TECHNOLOGIES

Types of Knowledge Discovery

Association Rules: These rules correlate the presence of a set of items with another range of values for another set of variables

Example:

- When a female retail shopper buys a handbag, she is likely to buy shoes.
- An X-ray image containing characteristics ‘a’ and ‘b’ is likely to also exhibit characteristics ‘c’.

Classification Hierarchies: The goal is to work from an existing set of events or transactions to create a hierarchy of classes.

Example:

- A population may be divided into five ranges to credit Worthiness based on a history of previous credit transactions.

- A model may be developed for the factors that determine the desirability of location of a store on a 1-10 scale.

Types of Knowledge Discovery Contd..

Sequential patterns: A sequence of actions or events is sought.

Example:

- It a patient underwent cardiac by pass surgery for blocked attires and an aneurysm and later developed high blood urea within a year of surgery, he or she is likely to suffer from kidney failure within the next 18 months.

Patterns with time series: Similarities can be detected within position of the time series.

Examples:

Two products show the same selling pattern in summer but a different one in winter.

Types of Knowledge Discovery Contd..

Categorization and segmentation: A given population of events or items can be partitioned into sets of “similar” events.

Examples:

- An entire population of treatment data on a disease may be divided into groups based on the similarity of side effects produced.
- The web accesses made by a collection of users against a set of documents may be analyzed in term of the keywords of documents to reveal clusters or categories of users.

DATABASE TECHNOLOGIES

Frequent Patterns

Frequent patterns are itemsets, subsequences or substructures that appear frequently in a data set.

Example:

- Itemset like milk and bread appearing frequently in a data set
- Sequence of PC -> Digital camera -> memory card appearing in shopping history

DATABASE TECHNOLOGIES

Market-Basket Analysis

- Process analyses buying habits of customers and identifies the association between items bought.
- Support is the percentage of transactions that contain A U B. It refers to how frequently a specific itemset occurs in the database

$$\text{Support } (A \Longrightarrow B) = p(A \cup B)$$

- Confidence is the percentage of transactions containing A that also contains B. This is considered as conditional probability

$$\text{Confidence } (A \Longrightarrow B) = \text{Support } (A \cup B) / \text{Support } (A)$$

Example:

Customer 1: milk, bread, cereal
Customer 2: milk, bread, sugar, eggs
Customer 3: milk, bread, butter, juice
Customer 4: milk, sugar, eggs
Customer 5: sugar, eggs

- Support (milk -> bread) = 3/5 = 0.6
- Confidence (milk -> bread) = 0.6/0.8 = 0.75
- Support (milk, bread -> juice) = 1/5 = 0.2
- Confidence (milk, bread -> juice) = 0.2/0.6 = 0.33

DATABASE TECHNOLOGIES

Apriori Property

All nonempty subsets of a frequent itemset must also be frequent

Transaction Id	Customer Id	Transaction Date	Item	Quantity
1	Cust1	1/1/2010	Pen	2
1	Cust1	1/1/2010	Ink	1
1	Cust1	1/1/2010	Milk	3
1	Cust1	1/1/2010	Juice	6
2	Cust2	1/1/2010	Pen	1
2	Cust2	1/1/2010	Ink	1
2	Cust2	1/1/2010	Milk	1
3	Cust3	1/1/2010	Pen	1
3	Cust3	1/1/2010	Milk	1
4	Cust1	15/1/2010	Pen	2
4	Cust1	15/1/2010	Ink	2
4	Cust1	15/1/2010	Juice	4
4	Cust1	15/1/2010	water	1

- {pen}, {ink}, {milk}, {juice}, {water}
- Frequent item set 1 with a minimum support of 0.7 – {pen}, {ink}, {milk}
- Frequent item set 2 with a minimum support of 0.7 – {pen, ink}, {pen, milk}

DATABASE TECHNOLOGIES

Apriori algorithm for finding Frequent Itemsets

Input: Database of m transactions, D, and a minimum support, mins, represented as a fraction of m.

Output: Frequent itemsets, L_1, L_2, … , L_k

Begin /* steps or statements are numbered for better readability */

1. Compute support(i_j) = count(i_j)/m for each individual item, i_1, i_2, …, i_n by scanning the database once and counting the number of transactions that item i_j appears in (that is, count(i_j));
2. The candidate frequent 1-itemset, C_1, will be the set of items i_1, i_2, …, i_n;
3. The subset of items containing i_j from C_1 where support(i_j) >= mins becomes the frequent 1-itemset, L_1;
4. k = 1; termination = false;

   repeat 1

   1. L_{k+1} = (empty set);
   2. Create the candidate frequent (k+1)-itemset, C_{k+1}, by combining members of L_k that have k–1 items in common (this forms candidate frequent (k+1)-itemsets by selectively extending frequent k-itemsets by one item);
   3. In addition, only consider as elements of C_{k+1} those k+1 items such that every subset of size k appears in L_k;
   4. Scan the database once and compute the support for each member of C_{k+1}; if the support for a member of Ck+1 >= mins then add that member to L_{k+1};
   5. 5. If Lk+1 is empty then termination = true else k = k + 1;

   until termination;

End

Recommendation Systems

A **recommendation** engine filters the data using different algorithms and recommends the most relevant items to users.

It first captures the past behavior of a customer and based on that recommends products which the users might likely buy.

Three major methods in designing a recommendation system:

1. Content-based method
2. Collaborative filtering method
3. Hybrid method

DATABASE TECHNOLOGIES

Recommendation Systems

A **recommendation** engine filters the data using different algorithms and recommends the most relevant items to users.

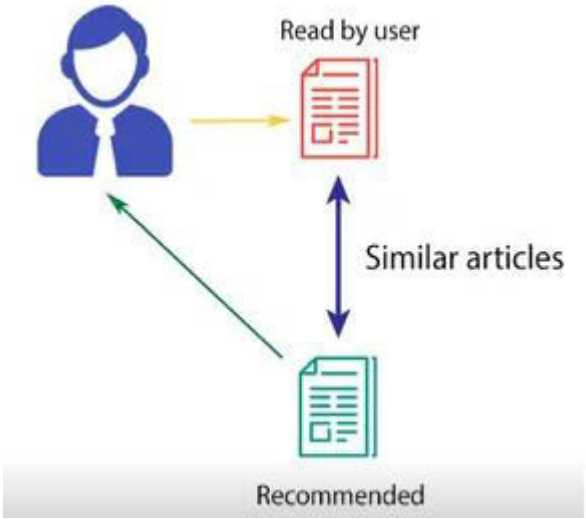
It first captures the past behavior of a customer and based on that, recommends products which the users might likely buy.

Three major methods in designing a recommendation system:

1. Content-based method
2. Collaborative filtering method
3. Hybrid method

Content based filtering

- Content-based approach recommends items that are similar to items the user preferred or queried in the past.
- It relies on product features and textual item descriptions



Content based filtering – Netflix example

- All information related to each user is stored in a vector form. This vector contains the past behavior of the user, i.e. the movies liked/disliked by the user and the ratings given by them. This vector is known as the ***profile*** *vector*.
- All information related to movies is stored in another vector called the ***item*** *vector* which contains details of each movie like genre, cast, director, etc.
- The content-based filtering algorithm finds the cosine of the angle between the profile vector and item vector, i.e. **cosine similarity**.
- Suppose A is the profile vector and B is the item vector, then the similarity between them can be calculated as:

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

- Based on the cosine value, which ranges between -1 to 1, the movies are arranged in descending order and one of the two below approaches is used for recommendations:
  - **Top-n approach**: where the top n movies are recommended (Here n can be decided by the business)
  - **Rating scale approach**: Where a threshold is set and all the movies above that threshold are recommended
- Other methods like **Euclidean Distance** and **Pearson's Correlation** can be used to calculate the similarity

Collaborative filtering

- Collaborative filtering approach considers a user's social environment. It recommends items based on the opinions of other customers who have similar tastes or preferences as the user
- **Example**: If person A likes three movies, say Interstellar, Inception and Predestination and person B likes Inception, Predestination and The Prestige, then they have almost similar interests. We can say with some certainty that **A would like The Prestige** and **B would like Interstellar**.
- The collaborative filtering algorithm uses "User Behavior" for recommending items. This is one of the most commonly used algorithms in the industry as it is not dependent on any additional information.
- There are different types of collaborating filtering techniques
  - User-User collaborative filtering
  - Item-Item collaborative filtering

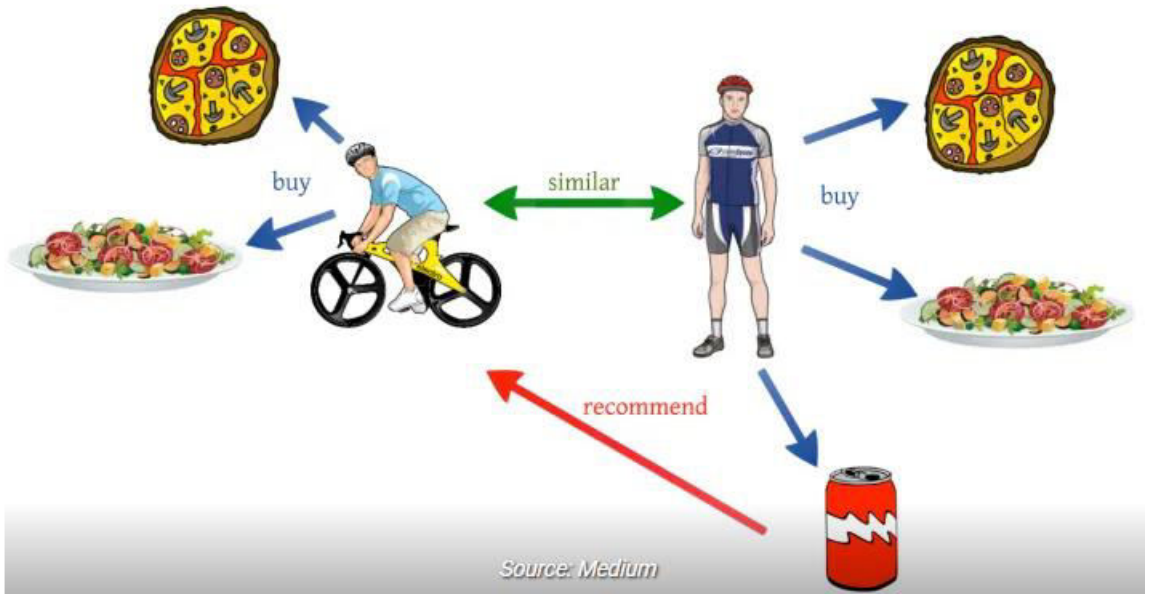
User-User collaborative filtering

- This algorithm first finds the similarity score between users.
- Based on this similarity score, it then picks out the most similar users and recommends products which these similar users have liked or bought previously
- The prediction of an item for a user *u* is calculated by computing the weighted sum of the user ratings given by other users to an item *i*.
- The prediction $P_{u,i}$ is given by:

$$P_{u,i} = \frac{\sum_v (r_{v,i} * s_{u,v})}{\sum_v s_{u,v}}$$

Where

$P_{u,i}$ is the prediction of an item
$R_{v,i}$ is the rating given by a user v to a movie i
$S_{u,v}$ is the similarity between users



User-User collaborative filtering Example:

Consider user-movie rating matrix :

User/Movie	x1	x2	x3	x4	x5	Mean User Rating
A	4	1	–	4	–	3
B	–	4	–	2	3	3
C	–	1	–	4	4	3

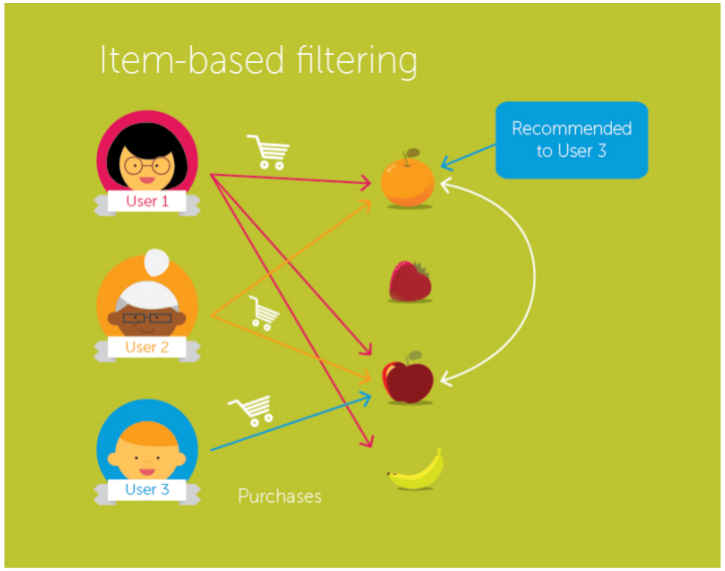
$r_{AC} = [(1-3)*(1-3) + (4-3)*(4-3)]/[((1-3)^2 + (4-3)^2)^{1/2} * ((1-3)^2 + (4-3)^2)^{1/2}] = 1$

$r_{BC} = [(4-3)*(1-3) + (2-3)*(4-3) + (3-3)*(4-3)]/[((4-3)^2 + (2-3)^2 + (3-3)^2)^{1/2} * ((1-3)^2+(4-3)^2 + (4-3)^2)^{1/2}] = -0.866$

- The correlation between user A and C is more than the correlation between B and C. Hence users A and C have more similarity and the movies liked by user A will be recommended to user C and vice versa

Item-Item collaborative filtering

- In this algorithm, we compute the similarity between each pair of items
- We need to find the similarity between each movie pair and based on that, we can recommend similar movies which are liked by the users in the past
- We compute the weighted sum of ratings of “item-neighbors”.
- The prediction is given by: $$P_{u,i} = \frac{\sum_N (s_{i,N} * R_{u,N})}{\sum_N (|s_{i,N}|)}$$



- Similarity between items $$sim(i,j) = cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2}$$
- Using similarity between each movie and the ratings, predictions are made and based on those predictions, similar movies are recommended

Item-Item collaborative filtering example

Consider user-movie rating matrix:

User/Movie	x1	x2	x3	x4	x5
A	4	1	2	4	4
B	2	4	4	2	1
C	–	1	–	3	4
Mean Item Rating	3	2	3	3	3

$$C_{14} = [(4-3)*(4-3) + (2-3)*(2-3)] / [((4-3)^2 + (2-3)^2)^{1/2} * ((4-3)^2 + (2-3)^2)^{1/2}] = 1$$

$$C_{15} = [(4-3)*(4-3) + (2-3)*(1-3)] / [((4-3)^2 + (2-3)^2)^{1/2} * ((4-3)^2 + (1-3)^2)^{1/2}] = 0.94$$

- The similarity between movie x1 and x4 is more than the similarity between movie x1 and x5. So based on these similarity values, if any user searches for movie x1, they will be recommended movie x4 and vice versa.

Hybrid filtering

- Hybrid filtering combines content-based filtering and collaborative filtering
- One approach to combine collaborative and content-based filtering is to make predictions based on a weighted average of the content-based recommendations and the collaborative recommendations
  - **Combining item scores:** In this approach, the ratings are combined by taking the average of the ratings
  - **Combining item ranks:** Suppose collaborative filtering recommended 5 movies A, B, C, D and E in the following order: A, B, C, D, E while content based filtering recommended them in the following order: B, D, A, C, E

Collaborative

Movie	Rank
A	1
B	0.8
C	0.6
D	0.4
E	0.2

Content

Movie	Rank
B	1
D	0.8
A	0.6
C	0.4
E	0.2

Hybrid

Movie	New Rank
A	1 + 0.6 = 1.6
B	0.8 + 1 = 1.8
C	0.6 + 0.4 = 1
D	0.4 + 0.8 = 1.2
E	0.2 + 0.2 = 0.4

Final recommendation: B, A, D, C, E

THANK YOU

Suresh Jamadagni
Department of Computer Science and Engineering

sureshjamadagni@pes.edu

DATABASE TECHNOLOGIES

Design and Implementation of Databases Systems

Suresh Jamadagni

Department of Computer Science and Engineering

DATABASE TECHNOLOGIES

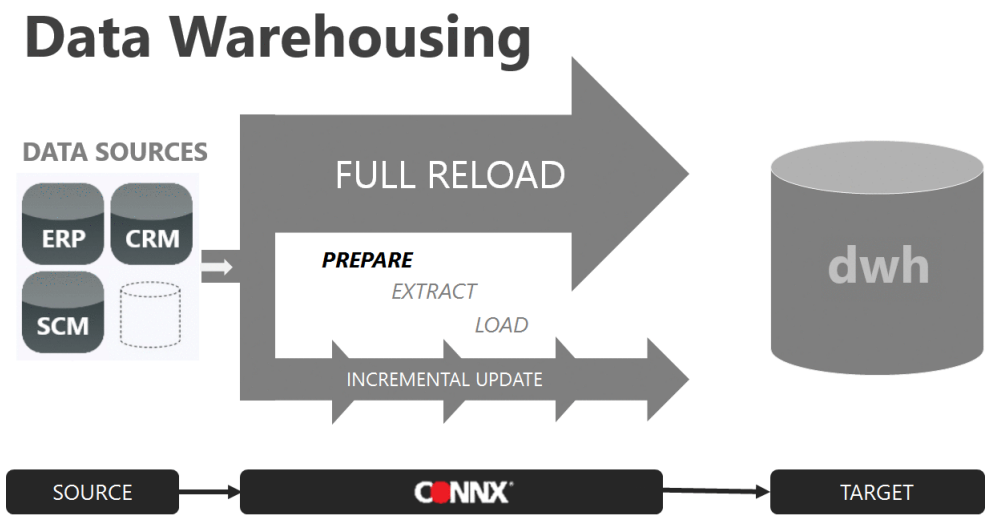
Overview of Data Warehousing Technology

Suresh Jamadagni
Department of Computer Science and Engineering

DATABASE TECHNOLOGIES

Data Warehousing

- A data warehouse is a storage architecture designed to hold data extracted from transaction systems, operational data stores and external sources.
- The warehouse then combines that data in an aggregate, summary form suitable for enterprise wide data analysis and reporting for predefined business needs.
- OLAP (online analytical processing) is a term used to describe the analysis of complex data from the data warehouse.
- A data warehouse is a subject-oriented, integrated, nonvolatile, time-variant collection of data in support of management's decisions.



DATABASE TECHNOLOGIES

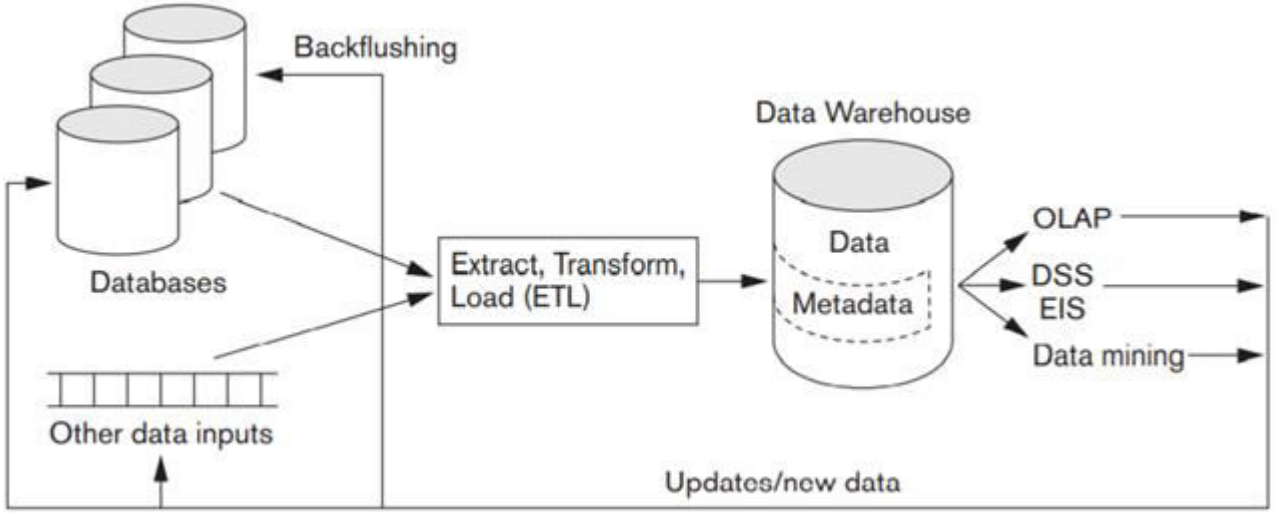
Characteristics of a Data Warehouse

- Multidimensional conceptual view
- Unlimited dimensions and aggregation levels
- Unrestricted cross-dimensional operations
- Dynamic sparse matrix handling
- Client/server architecture
- Multiuser support
- Accessibility
- Transparency
- Intuitive data manipulation
- Inductive and deductive analysis
- Flexible distributed reporting

DATABASE TECHNOLOGIES

Building a Data Warehouse

- Acquisition of Data for the Warehouse involves the following steps:
- Extraction of data from multiple heterogeneous sources
- Data formatting for consistency within a warehouse
- Data cleaning to ensure validity
- Fit data into data model of data warehouse
- Load data into data warehouse



Building a Data Warehousing

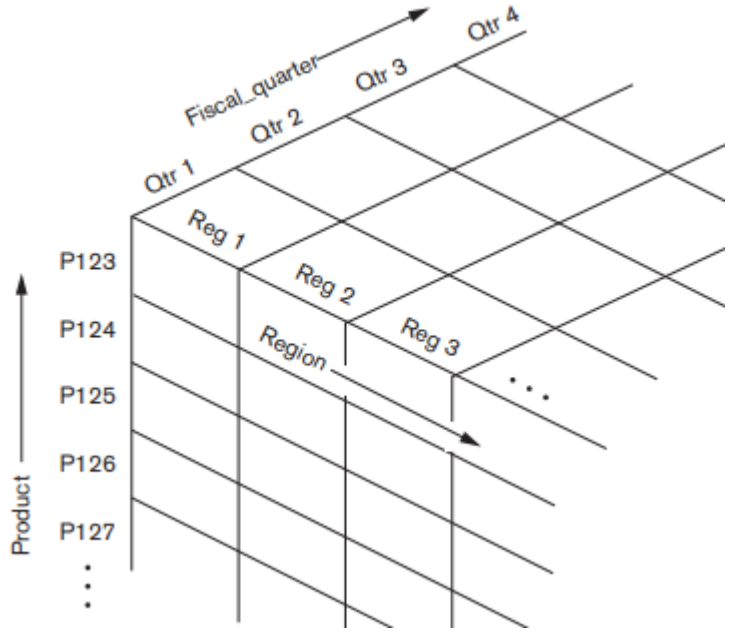
Design Considerations

- Usage projections
- The fit of the data model
- Characteristics of available sources
- Design of the meta-data component
- Modular component design
- Design for manageability and change
- Considerations of distributed and parallel architecture

Multi-Dimensional modeling

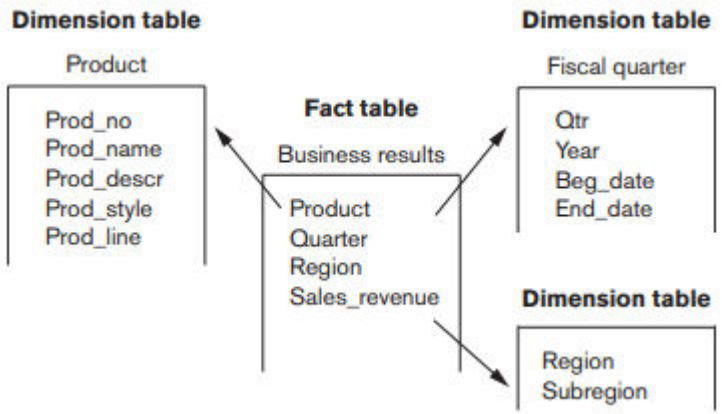
- Multidimensional models take advantage of inherent relationships in data to populate data in multidimensional matrices called **data cubes**
- The multidimensional model involves two types of tables:

  dimension tables and **fact tables**.
- A dimension table consists of tuples of attributes of the dimension.
- A fact table can be thought of as having tuples, one per a recorded fact. This fact contains some measured or observed variable(s) and identifies it (them) with pointers to dimension tables.
- The fact table contains the data and the dimensions identify each tuple in that data.



Multi-Dimensional modeling – Star schema

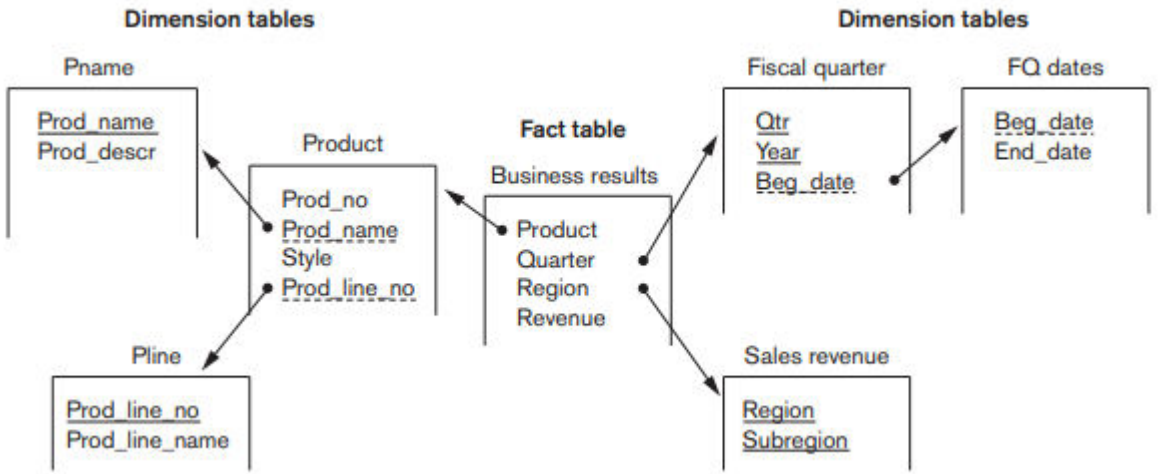
- The star schema consists of a fact table with a single table for each dimension



- A fact constellation is a set of fact tables that share some dimension tables.
- Data warehouse storage also utilizes bitmap indexing techniques to support high performance access

Multi-Dimensional modeling – Snowflake schema

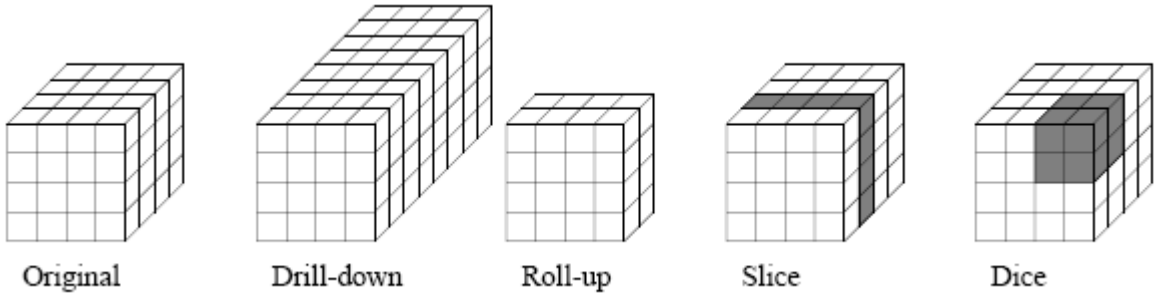
- Snowflake schema is a variation on the star schema in which the dimensional tables from a star schema are organized into a hierarchy by normalizing them



DATABASE TECHNOLOGIES

Online Analytical Processing (OLAP)

- **Roll-up**: Data is summarized with increasing generalization
- **Drill-down**: Increasing levels of detail are revealed (the complement of roll-up).
- **Pivot**: Cross tabulation (also referred to as rotation) is performed.
- **Slice and dice**: Projection operations are performed on the dimensions.
- **Sorting**: Data is sorted by ordinal value.
- **Selection**: Data is filtered by value or range.
- **Derived attributes**: Attributes are computed by operations on stored and derived values.



THANK YOU

Suresh Jamadagni
Department of Computer Science and Engineering

sureshjamadagni@pes.edu

DATABASE TECHNOLOGIES

Design and Implementation of Database Systems

Suresh Jamadagni

Department of Computer Science and Engineering

DATABASE TECHNOLOGIES

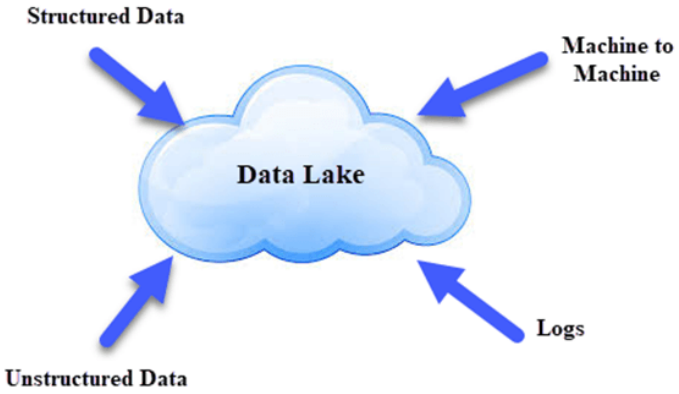
Overview of Data Lakes

Suresh Jamadagni
Department of Computer Science and Engineering

DATABASE TECHNOLOGIES

Data Lake

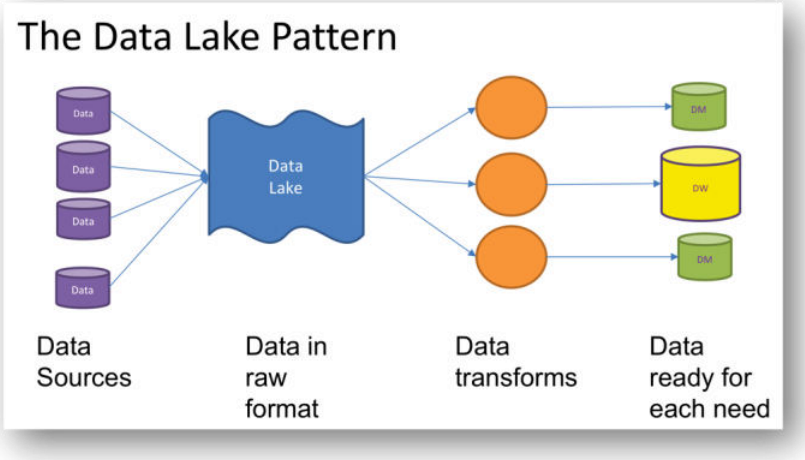
- A data lake is a centralized data repository that is capable of storing both traditional structured (row and column) data as well as unstructured, non-tabular raw data in its native format (like videos, images, binary files and more.)
- Data Lakes leverage inexpensive object storage and open formats to enable many applications to take advantage of the data.
- A data lake is optimized for the quick ingestion of raw, detailed source data plus on-the-fly processing of such data for exploration, analytics, and operations



DATABASE TECHNOLOGIES

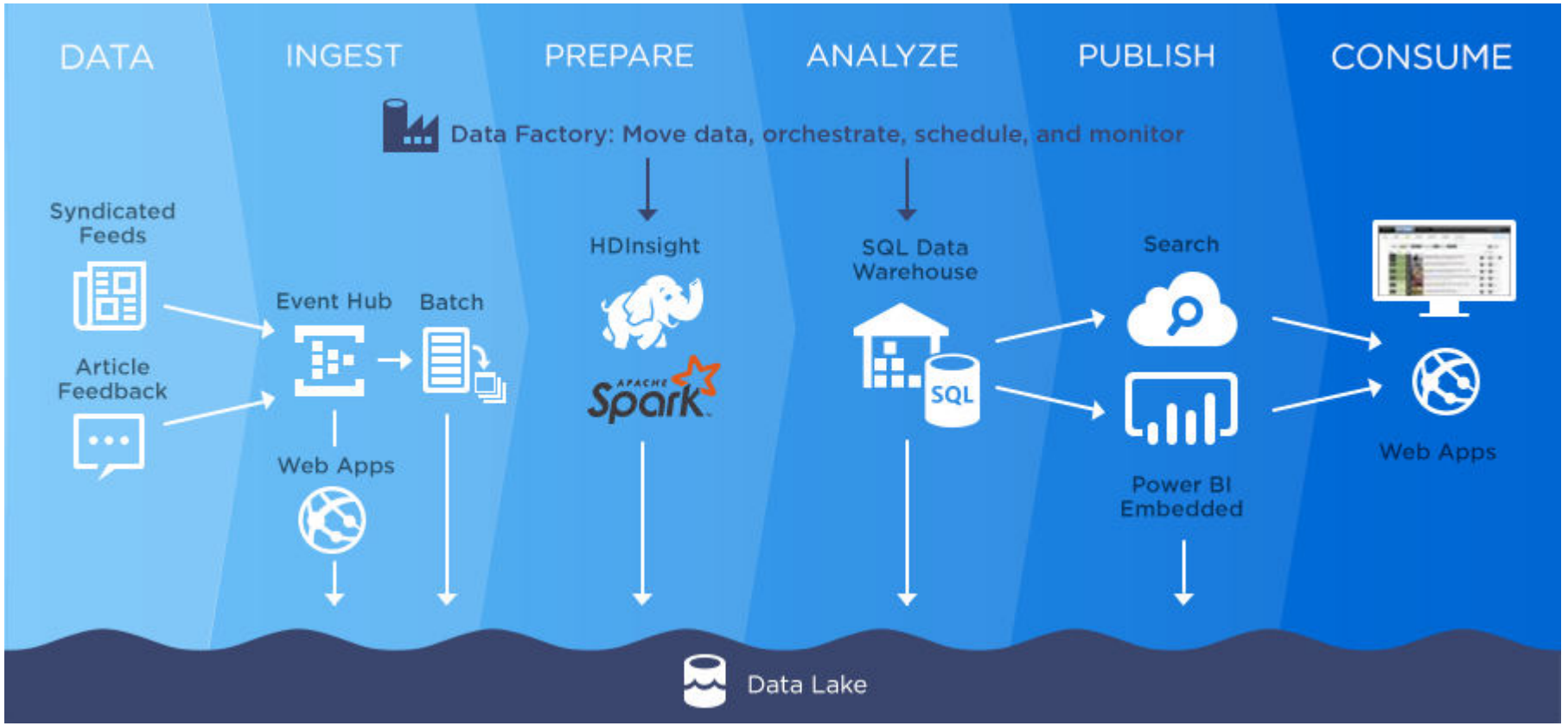
Data Lake

- A design pattern is a generalized, repeatable approach to commonly occurring situations in information technology solutions.
- Data-driven design patterns tend to be localized data constructs such as data schema, models, tables, and record structures
- A data lake is a collection of data organized by user-designed patterns
- A data lake is an effective data-driven design pattern for capturing a wide range of data types, both old and new, at large scale.



DATABASE TECHNOLOGIES

Data Lake



DATABASE TECHNOLOGIES

Comparison of Data Warehouse and Data Lake characteristics

	Data lake	Data warehouse
Primary types of data	All types: Structured data, semi-structured data, unstructured (raw) data	Structured data only
Cost	$	$$$
Scalability	Scales to hold any amount of data at low cost, regardless of type	Scaling up becomes exponentially more expensive due to vendor costs
Intended users	Data analysts, data scientists	Data analysts
Vendor lock-in	No	Yes
Advantages	Low cost, flexibility, scalability, allows storage of the raw data needed for machine learning	User interface is familiar to users of traditional databases
Disadvantages	Exploring large amounts of raw data can be difficult without tools to organize and catalog the data	Expensive, always-on architecture, proprietary software, cannot hold unstructured (raw) data needed for machine learning

DATABASE TECHNOLOGIES

Compelling, Real-World Use Cases for Data Lakes

- Discovery of new insights and opportunities.
- **Self-service** data exploration, data preparation and analytics
- Competing on analytics.
- Multichannel marketing.
- Old and new data draw a more complete view of the customer
- Analytics with all the data
- **Real-time operations**

Apache Hadoop™ and Spark™ enable unstructured data analysis and set the stage for modern data lakes

DATABASE TECHNOLOGIES

Data Lake tools and frameworks

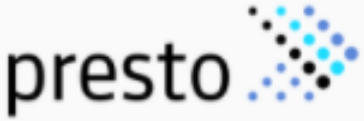
Apache Spark™ is the de facto open source big data processing engine, enabling SQL queries and rapid distributed processing of the data in your data lake. Learn more about running Spark on Databricks.

The Databricks Unified Data Analytics Platform makes it easy to run SQL queries on your data lake, do massive scale data engineering and collaborative data science. Try Databricks for free today.

Simplify and strengthen your data architecture by using Delta Lake to ensure data validity and consistent views at petabyte scale. Learn more about Delta Lake.

Amazon Web Services' Simple Storage Service (S3) provides cost effective object storage for data lakes. Learn more about building a data lake using Amazon Web Services and S3.

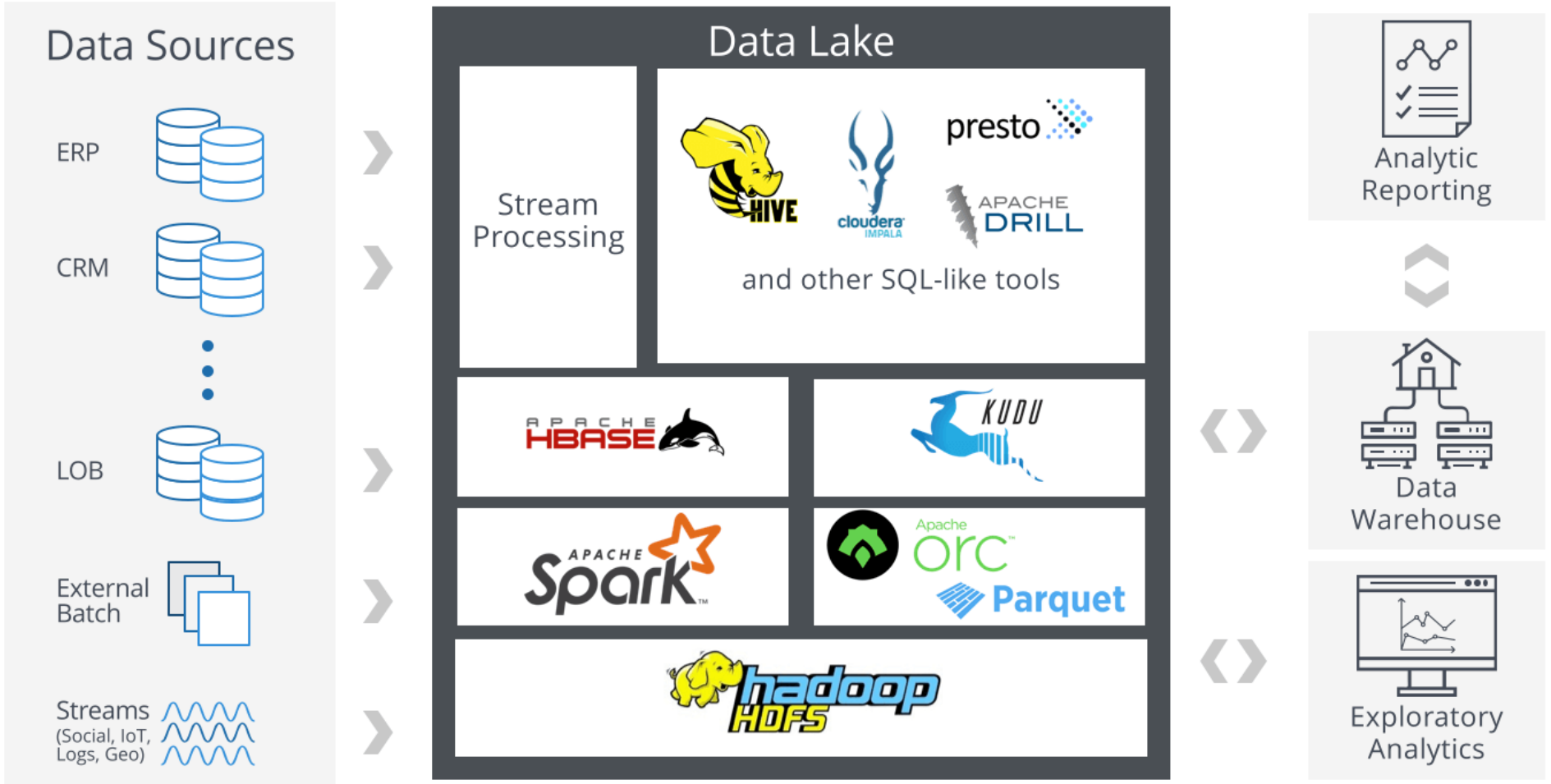
Learn more about building a data lake using Microsoft Azure Data Lake Storage Gen2, and about Azure Databricks.



Presto was originally created by Facebook to run queries on Hadoop data warehouses. It can be used to run SQL queries on data lakes at scale. Learn more about Presto.

DATABASE TECHNOLOGIES

Apache Data Lake tools



DATABASE TECHNOLOGIES

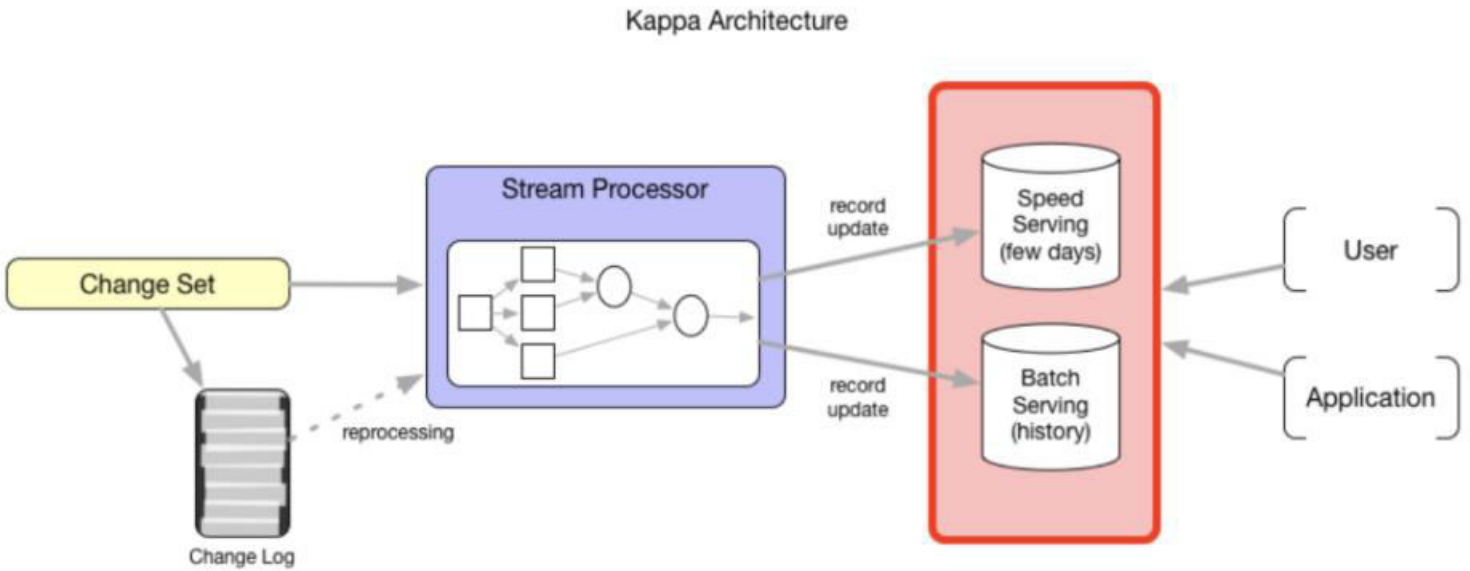
Data Lakes – Apache Kafka

- Apache Kafka is an open-source distributed **event streaming** platform used for high-performance data pipelines, streaming analytics and data integration
- **Event streaming** is the practice of capturing data in real-time from event sources like databases, sensors, mobile devices, cloud services and software applications in the form of streams of events; storing these event streams durably for later retrieval; manipulating, processing, and reacting to the event streams in real-time
- Kafka combines three key capabilities for event streaming end-to-end solution:
    1. To **publish** (write) and **subscribe to** (read) streams of events, including continuous import/export of your data from other systems.
    2. To **store** streams of events durably and reliably for as long as you want.
    3. To **process** streams of events as they occur or retrospectively.
- It is optimized for ingesting and processing streaming data in real-time.

DATABASE TECHNOLOGIES

Data Lakes – Apache Hudi

- Apache Hudi is a data lake framework which provides the ability to ingest, manage and query large analytical data sets on a distributed file system/cloud stores.
- Hudi facilitates “incremental processing” of the data lakes
- Hudi joined the Apache incubator for incubation in January 2019, and was promoted to the top Apache project in May 2020.



DATABASE TECHNOLOGIES

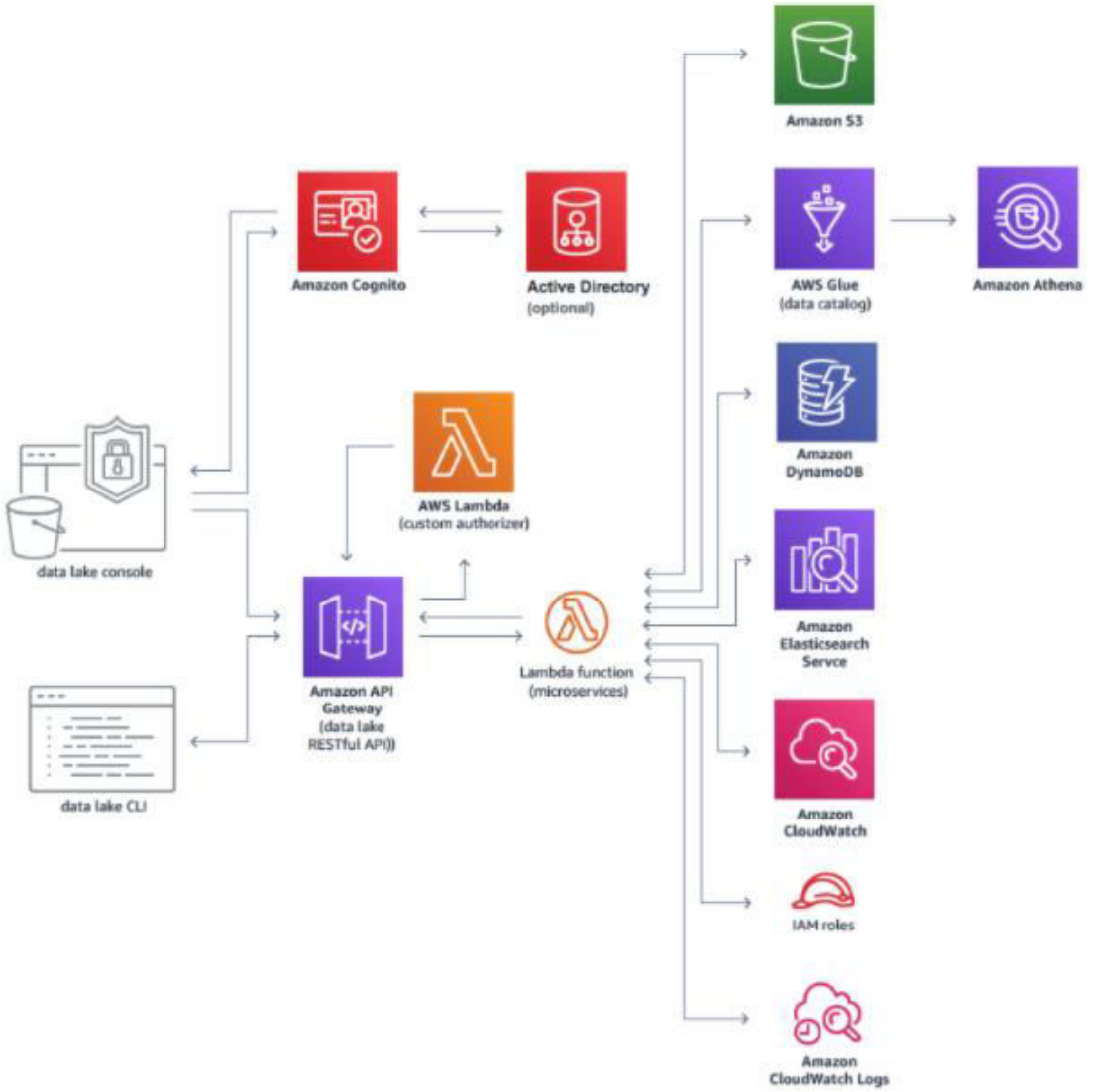
Data Lakes – AWS Services

Category	Use cases	AWS service
Analytics	Interactive analytics	Amazon Athena
	Big data processing	Amazon EMR
	Data warehousing	Amazon Redshift
	Real-time analytics	Amazon Kinesis
	Operational analytics	Amazon Elasticsearch Service
	Dashboards and visualizations	Amazon Quicksight
Data movement	Real-time data movement	Amazon Managed Streaming for Apache Kafka (MSK) Amazon Kinesis Data Streams Amazon Kinesis Data Firehose Amazon Kinesis Data Analytics Amazon Kinesis Video Streams AWS Glue
Data lake	Object storage	Amazon S3 AWS Lake Formation
	Backup and archive	Amazon S3 Glacier AWS Backup
	Data catalog	AWS Glue AWS Lake Formation
	Third-party data	AWS Data Exchange
Predictive analytics and machine learning	Frameworks and interfaces	AWS Deep Learning AMIs
	Platform services	Amazon SageMaker

DATABASE TECHNOLOGIES

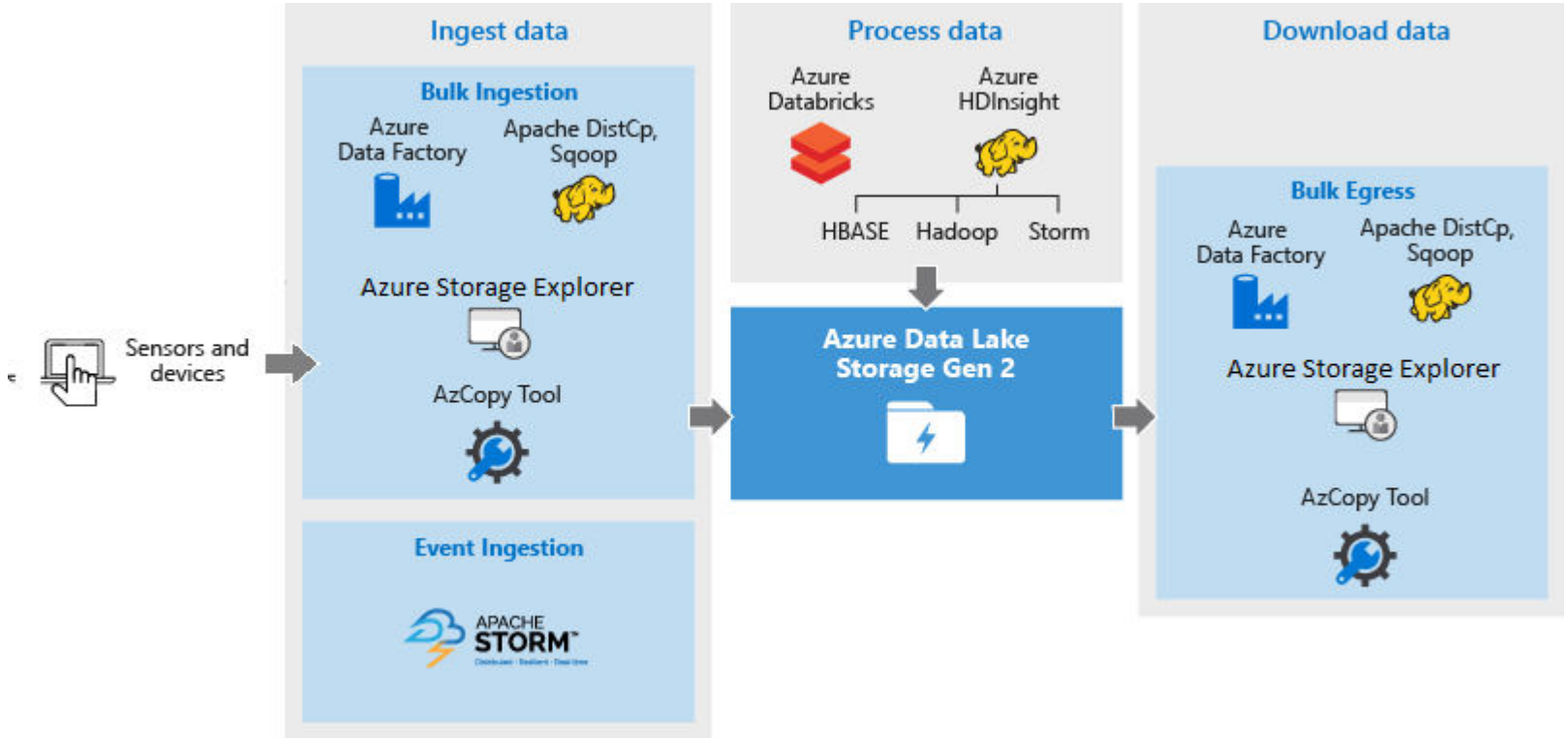
Data Lakes – AWS Services

- Amazon S3 is the core service at the heart of the modern data lake architecture
- Amazon Athena allows users to use S3 data for ad hoc queries using SQL
- Amazon Redshift is the AWS data warehousing service.
- Amazon CloudWatch is a monitoring and observability service. CloudWatch provides data and actionable insights to monitor applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health.
- Amazon Cognito manages user sign-up, sign-in, and access control



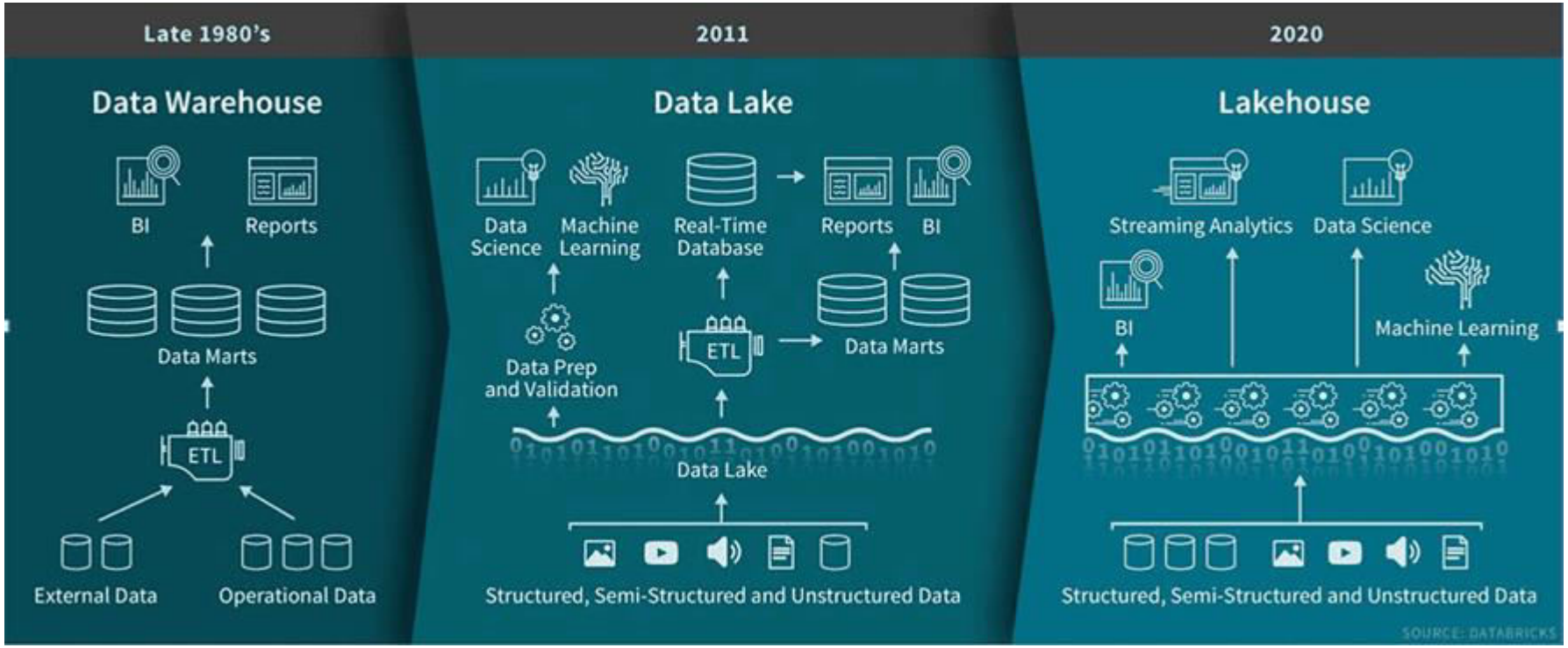
DATABASE TECHNOLOGIES

Data Lakes – Microsoft Azure Services



DATABASE TECHNOLOGIES

Data Lakehouse



A lakehouse is a new paradigm that combines the best elements of data lakes and data warehouses. Lakehouses are enabled by a new system design: implementing similar data structures and data management features to those in a data warehouse, directly on the kind of low cost storage used for data lakes.

THANK YOU

Suresh Jamadagni
Department of Computer Science and Engineering

sureshjamadagni@pes.edu

DATABASE TECHNOLOGIES

Design and Implementation of Databases Systems

Suresh Jamadagni

Department of Computer Science and Engineering

DATABASE TECHNOLOGIES

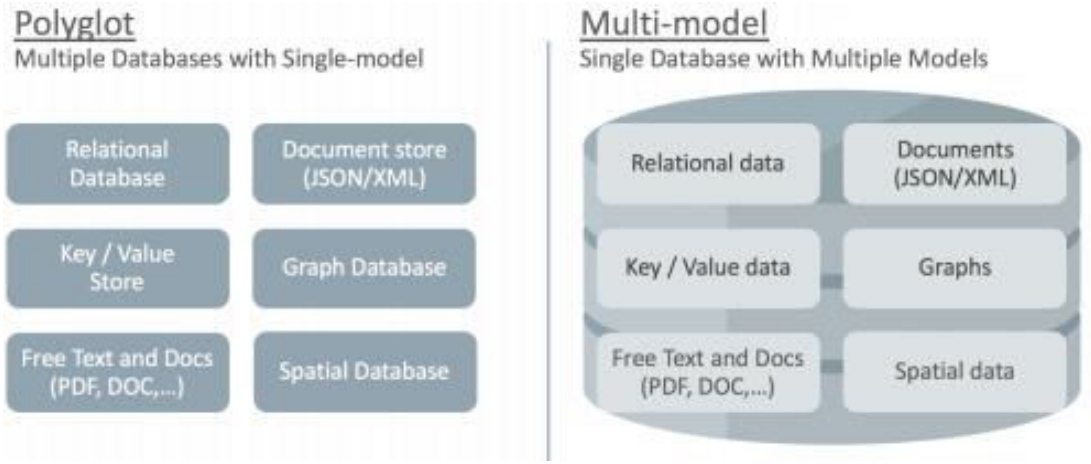
Overview of Multi-model Databases

Suresh Jamadagni
Department of Computer Science and Engineering

DATABASE TECHNOLOGIES

Multi-model Databases

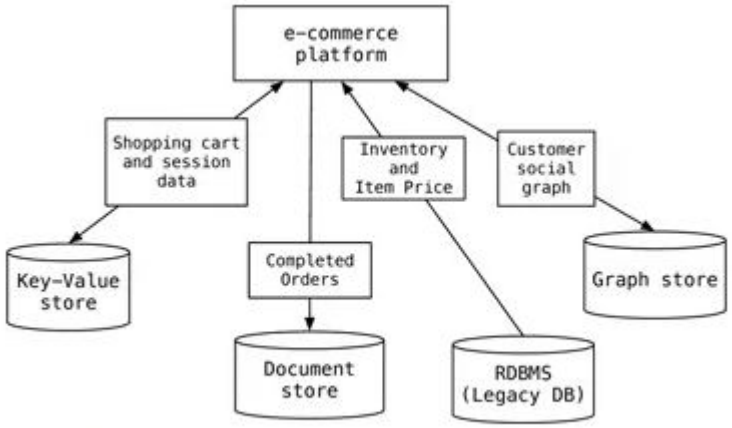
- A Multi-model database is a database that can store, index and query data in more than one model. For some time, databases have primarily supported only one model, such as: relational database, document-oriented database, graph database or Key-value database.
- A database that combines many of these is multi-model.



DATABASE TECHNOLOGIES

Multi-model Databases – Polyglot persistence

- **Polyglot persistence** is a term that refers to using multiple data storage technologies for varying data storage needs across an application.
- When storing data, it is best to use multiple data storage technologies, chosen based upon the way data is being used by individual applications or components of a single application.
- Different kinds of data are best dealt with different data stores. It means picking the right tool for the right use case.



Example implementation of polyglot persistence

DATABASE TECHNOLOGIES

Multi-model Databases - Comparison

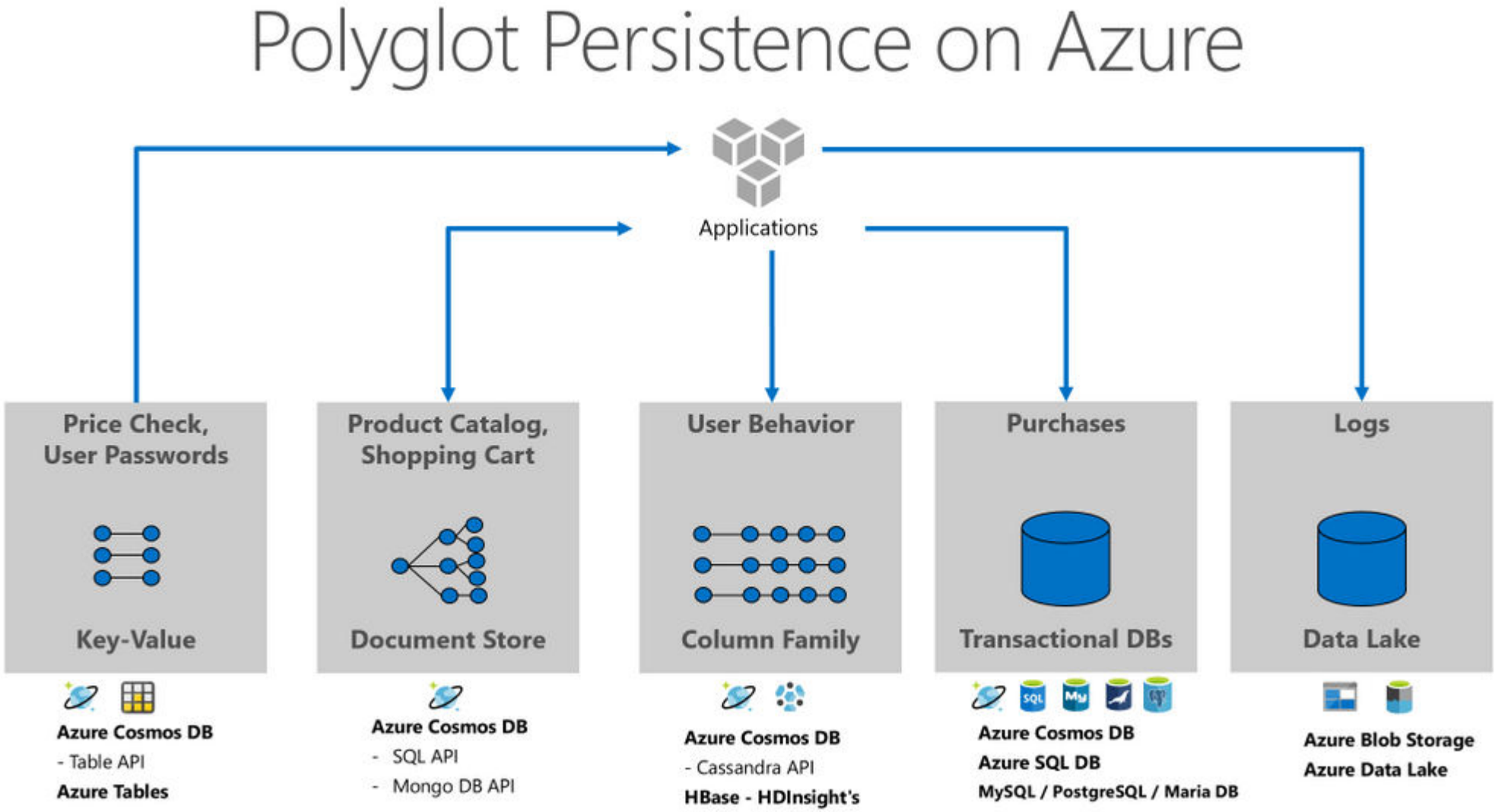
Database	Relational	Document	Graph	Object	License	Transactions
ArangoDB	No	Yes	Yes	No	Apache 2 License	Full ACID, pessimistic locking, configurable durability
Azure Cosmos DB	Yes	Yes	Yes	Yes	Proprietary	Full ACID within a partition, multiple consistency models
CrateIO	Yes	Yes	No	Yes	Apache 2 License	Eventual consistency, Optimistic concurrency control
EnterpriseDB	Yes	Yes	No	Yes	Proprietary	Full ACID
MarkLogic	Yes	Yes	Yes	No	Proprietary	Full ACID
Oracle	Yes	Yes	Yes	Yes	Proprietary	Full ACID
OrientDB	Yes	Yes	Yes	Yes	Apache 2 License	Full ACID, even distributed
SAP HANA	Yes	Yes	Yes	No	Proprietary	Full ACID
Virtuoso	Yes	Yes	Yes	Yes	Proprietary or GNU GPL v2	Full ACID

DATABASE TECHNOLOGIES

Multi-model Databases – OrientDB

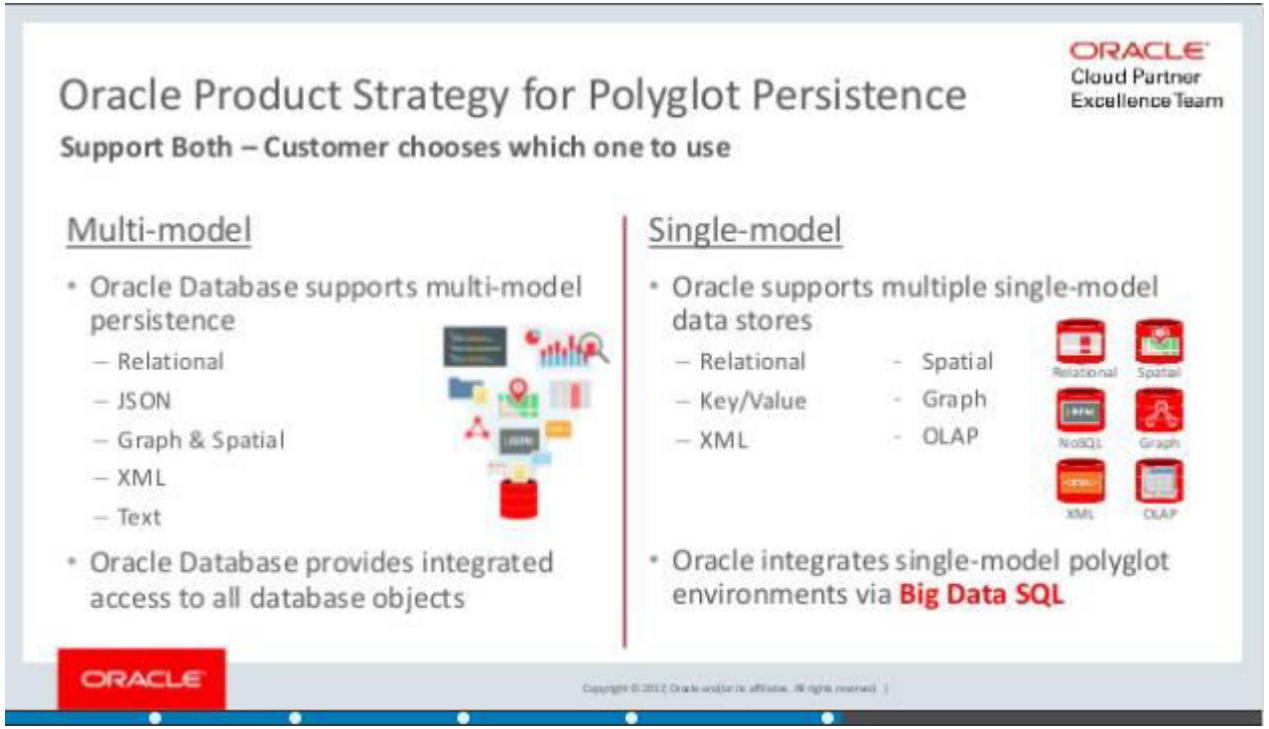
- **OrientDB** is an open source NoSQL database management system written in Java.
- It is a Multi-model database, supporting graph, document, key/value and object models.
- The relationships are managed as in graph databases with direct connections between records.
- It supports schema-less, schema-full and schema-mixed modes.
- OrientDB uses several indexing mechanisms based on B-tree and Extendible hashing.
- Each record has Surrogate key which indicates position of record inside of Array list, links between records are stored either as single value of record's position stored inside of referrer or as B-tree of record positions (so-called record IDs or RIDs) which allows fast traversal (with O(1) complexity) of one-to-many relationships and fast addition/removal of new links

Multi-model Databases – MS Azure CosmosDB



DATABASE TECHNOLOGIES

Multi-model Databases – Oracle



DATABASE TECHNOLOGIES

Multi-model Database - Oracle

```
create table Customers (
  id integer,
  name string,
  address string,
  orders array (
    record (
      order_no string,
      orderlines array (
        record (
          product_no string,
          product_name string,
          price integer ) ) )
    ),
  primary key (id)
);

import -table Customers -file customer.json
```

customer.json:

```
{   "id":1,
    "name":"Mary",
    "address":"Prague",
    "orders" : [
      { "order_no":"0c6df508",
        "orderlines":[
          { "product_no" : "2724f",
            "product_name" : "Toy",
            "price" : 66 },
          { "product_no" : "3424g",
            "product_name"  :"Book",
            "price" : 40 } ] } ]
}
{   "id":2,
    "name":"John",
    "address":"Helsinki",
    "orders" : [
      {"order_no":"0c6df511",
       "orderlines":[
        { "product_no" : "2454f",
          "product_name" : "Computer",
          "price" : 34   } ] } ]
}
```

DATABASE TECHNOLOGIES

Multi-model Database - Oracle

```
sql-> SELECT c.name, c.orders.order_no, c.orders.orderlines[0].product_name
   -> FROM customers c
   -> where c.orders.orderlines[0].price > 50;
 +------+----------+--------------+
 | name | order_no | product_name |
 +------+----------+--------------+
 | Mary | 0c6df508 | Toy          |
 +------+----------+--------------+
```

```
sql-> SELECT c.name, c.orders.order_no,
   -> [c.orders.orderlines[$element.price >35]]
   -> FROM customers c;
 +------+----------+-------------------------+
 | name | order_no |        Column_3         |
 +------+----------+-------------------------+
 | Mary | 0c6df508 | product_no   | 2724f    |
 |      |          | product_name | Toy      |
 |      |          | price        | 66       |
 |      |          |                         |
 |      |          | product_no   | 3424g    |
 |      |          | product_name | Book     |
 |      |          | price        | 40       |
 +------+----------+-------------------------+
 | John | 0c6df511 |                         |
 +------+----------+-------------------------+
```

```
sql-> select * from Customers
   -> ;
 +----+------+----------+----------------------------+
 | id | name | address  |           orders           |
 +----+------+----------+----------------------------+
 |  2 | John | Helsinki | order_no          | 0c6df511 |
 |    |      |          | orderlines                 |
 |    |      |          |     product_no    | 2454f  |
 |    |      |          |     product_name  | Computer |
 |    |      |          |     price         | 34     |
 +----+------+----------+----------------------------+
 |  1 | Mary | Prague   | order_no          | 0c6df508 |
 |    |      |          | orderlines                 |
 |    |      |          |     product_no    | 2724f  |
 |    |      |          |     product_name  | Toy    |
 |    |      |          |     price         | 66     |
 |    |      |          |                            |
 |    |      |          |     product_no    | 3424g  |
 |    |      |          |     product_name  | Book   |
 |    |      |          |     price         | 40     |
 +----+------+----------+----------------------------+
```

DATABASE TECHNOLOGIES

Multi-model Database - PostgreSQL

```
CREATE TABLE customer (
  id       INTEGER PRIMARY KEY,
  name     VARCHAR(50),
  address  VARCHAR(50),
  orders   JSONB
);
```

```
INSERT INTO customer
VALUES (1, 'Mary', 'Prague',
 '{"Order_no":"0c6df508",
   "Orderlines":[
    {"Product_no":"2724f", "Product_Name":"Toy",   "Price":66 },
    {"Product_no":"3424g", "Product_Name":"Book", "Price":40}]
  }');

INSERT INTO customer
VALUES (2, 'John', 'Helsinki',
 '{"Order_no":"0c6df511",
   "Orderlines":[
    { "Product_no":"2454f", "Product_Name":"Computer", "Price":34
}]
  }');
```

id integer	name character varying (50)	address character varying (50)	orders jsonb
1	Mary	Prague	{"Orderlines":[{"Price":66,"Product_Name":"Toy","Product_no":"2724f"},{"Price":40,"Product_Name":...
2	John	Helsinki	{"Orderlines":[{"Price":34,"Product_Name":"Computer","Product_no":"2454f"}],"Order_no":"0c6df511"}

DATABASE TECHNOLOGIES

Multi-model Database - PostgreSQL

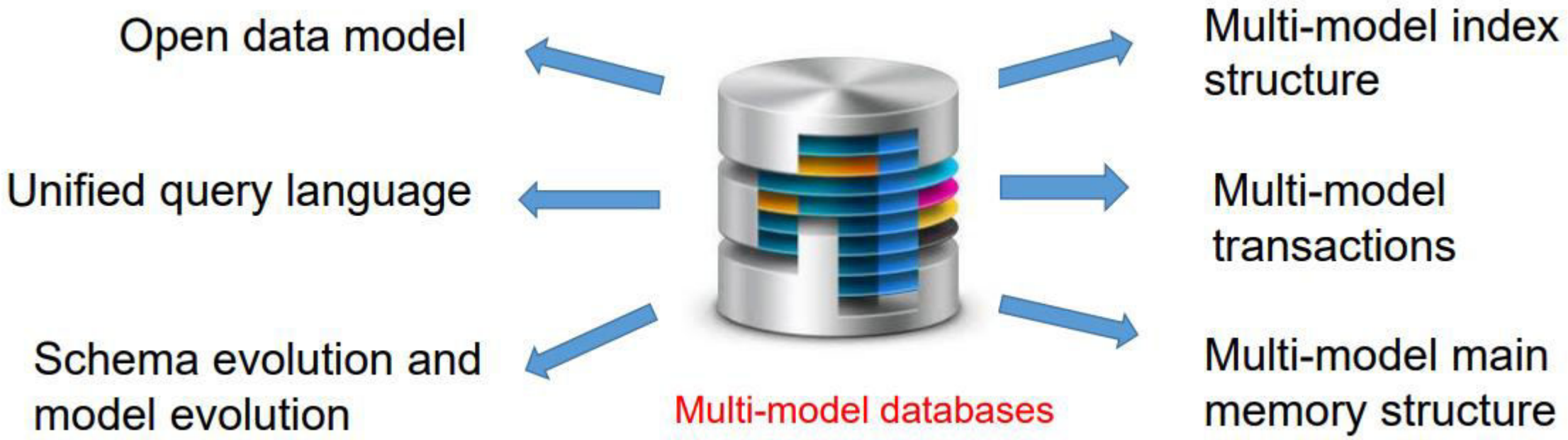
id integer	name character varying (50)	address character varying (50)	orders jsonb
1	Mary	Prague	{"Orderlines":[{"Price":66,"Product_Name":"Toy","Product_no":"2724f"},{"Price":40,"Product_Name":...
2	John	Helsinki	{"Orderlines":[{"Price":34,"Product_Name":"Computer","Product_no":"2454f"}],"Order_no":"0c6df511"}

```
{"Order_no":"0c6df508",
   "Orderlines":[
    { "Product_no":"2724f",
      "Product_Name":"Toy",
      "Price":66 },
    {"Product_no":"3424g",
     "Product_Name":"Book",
     "Price":40}]
}
```

```
SELECT name,
       orders->>'Order_no' as Order_no,
       orders#>'{Orderlines,1}'->>'Product_Name' as Product_Name
FROM customer
where orders->>'Order_no' <> '0c6df511';
```

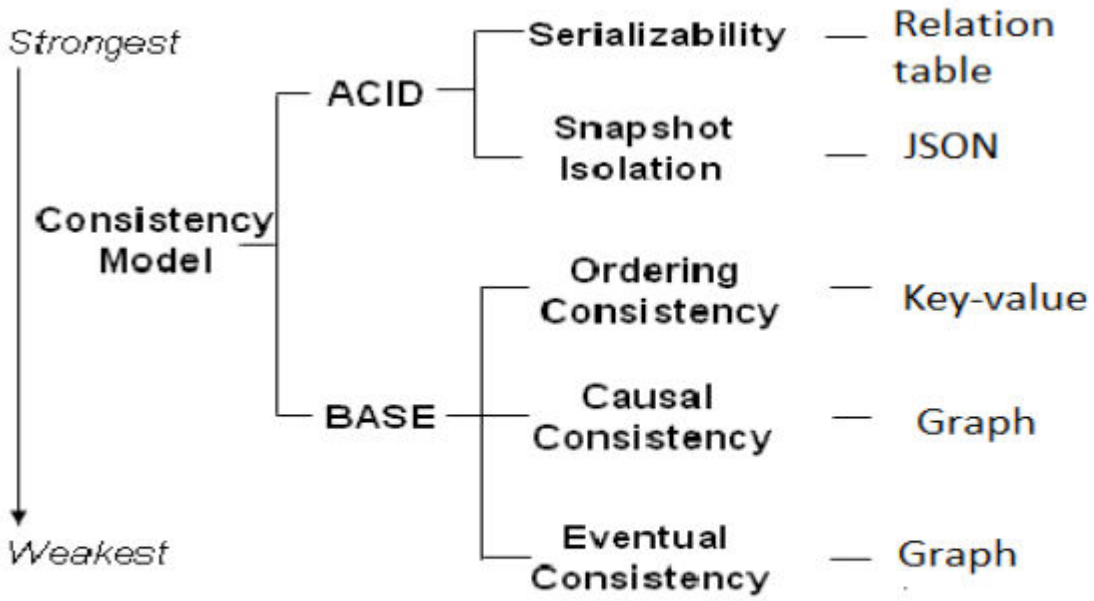
name character varying (50)	order_no text	product_name text
Mary	0c6df508	Book

Multi-model Databases - Challenges



DATABASE TECHNOLOGIES

Multi-model Database Transactions



An example of multi-model data hybrid consistency models

THANK YOU

Suresh Jamadagni
Department of Computer Science and Engineering

sureshjamadagni@pes.edu

DATABASE TECHNOLOGIES

Design and Implementation of Databases Systems

Suresh Jamadagni, Raghu B. A.

Department of Computer Science and Engineering

DATABASE TECHNOLOGIES

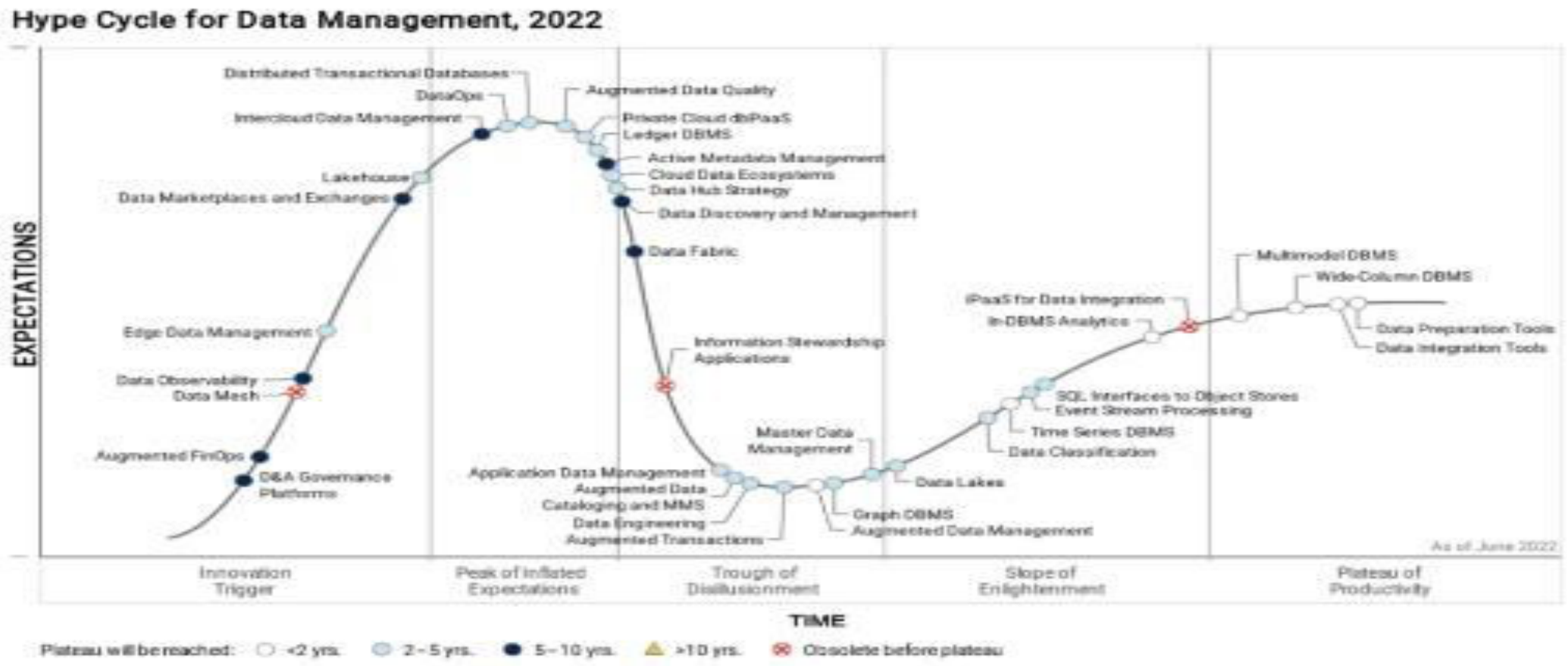
Technology Trends in Data Management

Suresh Jamadagni, Raghu B. A.
Department of Computer Science and Engineering

DATABASE TECHNOLOGIES

Technology Trends

Figure 1: Hype Cycle for Data Management, 2022



Source: Gartner (June 2022)

DATABASE TECHNOLOGIES

Gartner Top 10 Trends in Data and Analytics for 2020

1. Smarter, faster, more responsible AI

- By the end of 2024, 75% of enterprises will shift from piloting to operationalizing AI, driving a 5X increase in *streaming data and analytics* infrastructures

2. Decline of the dashboard

- Dynamic data stories with more automated and consumerized experiences will replace visual, point-and-click authoring and exploration

3. Decision intelligence

- By 2023, more than 33% of large organizations will have analysts practicing decision intelligence, including decision modeling
- Decision intelligence brings together a number of disciplines, including decision management and decision support

4. X analytics

- "X analytics" is an umbrella term, where X is the data variable for a range of different structured and unstructured content such as text analytics, video analytics, audio analytics, etc.

5. Augmented data management

- Augmented data management uses ML and AI techniques to optimize and improve operations. It also converts metadata from being used in auditing, lineage and reporting to powering dynamic systems

6. Cloud is a given

- By 2022, public cloud services will be essential for 90% of data and analytics innovation.

7. Data and analytics worlds collide

- Vendors offering end-to-end workflows enabled by augmented analytics blur the distinction between once separate markets

8. Data marketplaces and exchanges

- Data marketplaces and exchanges provide single platforms to consolidate third-party data offerings. These marketplaces and exchanges provide centralized availability and access (to X analytics and other unique data sets, for example) that create economies of scale to reduce costs for third-party data

9. Blockchain in data and analytics

- Blockchain technologies address two challenges in data and analytics. First, blockchain provides the full lineage of assets and transactions. Second, blockchain provides transparency for complex networks of participants.

10. Relationships form the foundation of data and analytics value

- By 2023, graph technologies will facilitate rapid contextualization for decision making in 30% of organizations worldwide. Graph analytics is a set of analytic techniques that allows for the exploration of relationships between entities of interest such as organizations, people and transactions.

References:

https://www.gartner.com/en/documents/4004072
https://www.montecarlodata.com/new-gartner-report-hype-cycle-for-data-management/

THANK YOU

Suresh Jamadagni
Department of Computer Science and Engineering

sureshjamadagni@pes.edu