Unit – 7

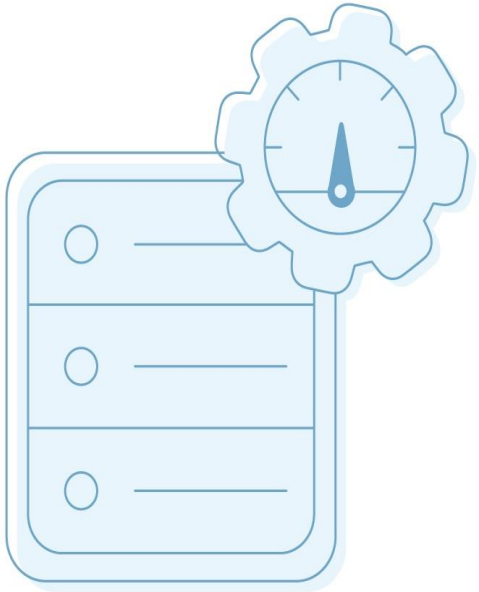
Advanced Topics

Database Performance Tuning

- Database performance tuning refers to a group of activities DBAs perform to ensure databases operate smoothly and efficiently. It helps re-optimize a database system from top to bottom, from software to hardware, to improve overall performance.
- Database performance tuning is the process of optimizing a database system to improve its efficiency, response time and overall performance.
- It involves various techniques and strategies for enhancing the database's ability to handle increasing workloads and deliver faster query processing. Some of them are:
    - ✓ Query Optimization
    - ✓ Data Defragmentation
    - ✓ Increase Memory
    - ✓ Improve Indexes
    - ✓ Update Database Versions

Database Performance Tuning Techniques

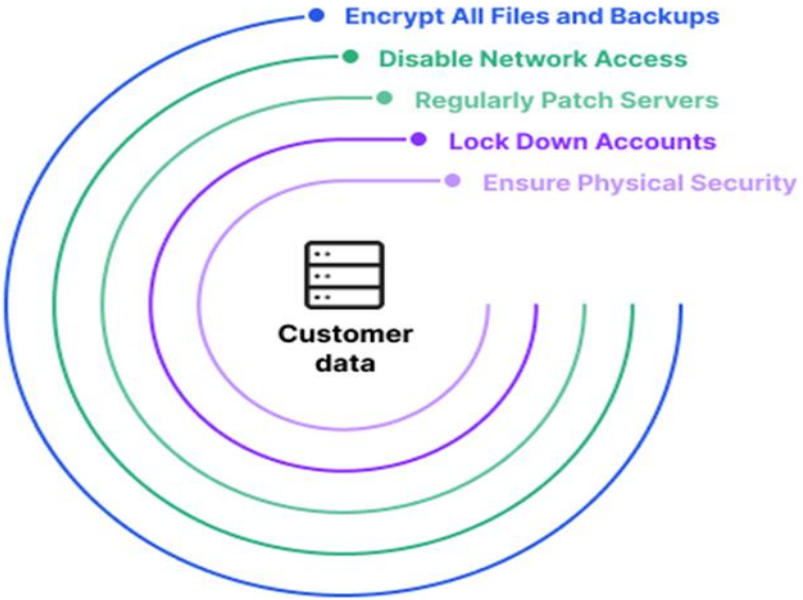
- Implement Baselines
- Use the Automatic Workload Repository (AWR) Tool
- Create Optimal Indexes
- Undertake SQL Tuning
- Check Your Statistics
- Optimize CPU Utilization
- Update Your Database Version



- Consider Upgrading CPU
- Allocate More Memory
- Consider Defragmentation
- Use the Right Disk Types
- Review Access
- Understand Load and Response Time for Service Instances
- Don't Overlook the Network
- Use Professional Software

Database Security

- Database security is the processes, tools, and controls that secure and protect databases against accidental and intentional threats.
- The objective of database security is to secure sensitive data and maintain the confidentiality, availability, and integrity of the database.



Database Security Measures

- **Access Control:**
  - ✓ Implement robust access controls to restrict access to the database to authorized users only.
  - ✓ This includes using strong authentication mechanisms such as passwords, multi-factor authentication (MFA), and role-based access control (RBAC) to manage user privileges effectively.

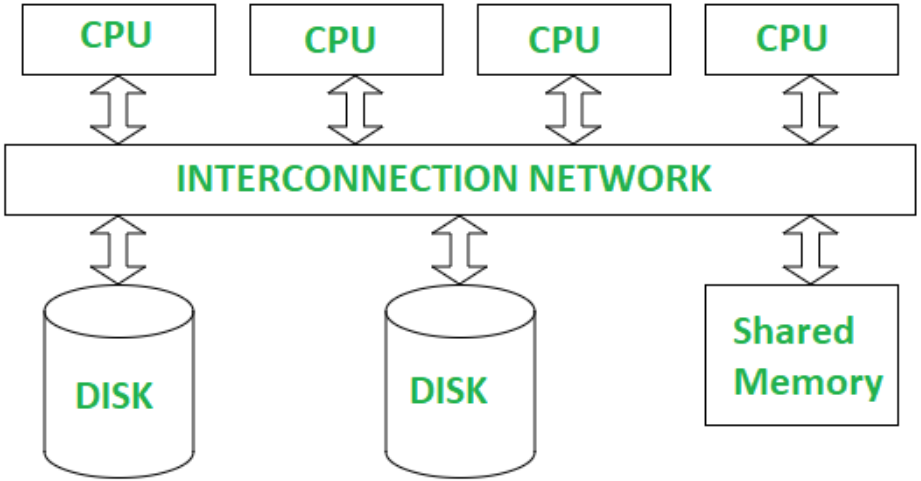
- **Encryption:**
  - ✓ Encrypt sensitive data both at rest and in transit to protect it from unauthorized access.
  - ✓ Use encryption techniques such as Transparent Data Encryption (TDE) for data-at-rest encryption and Secure Socket Layer (SSL) or Transport Layer Security (TLS) for data transmission encryption.

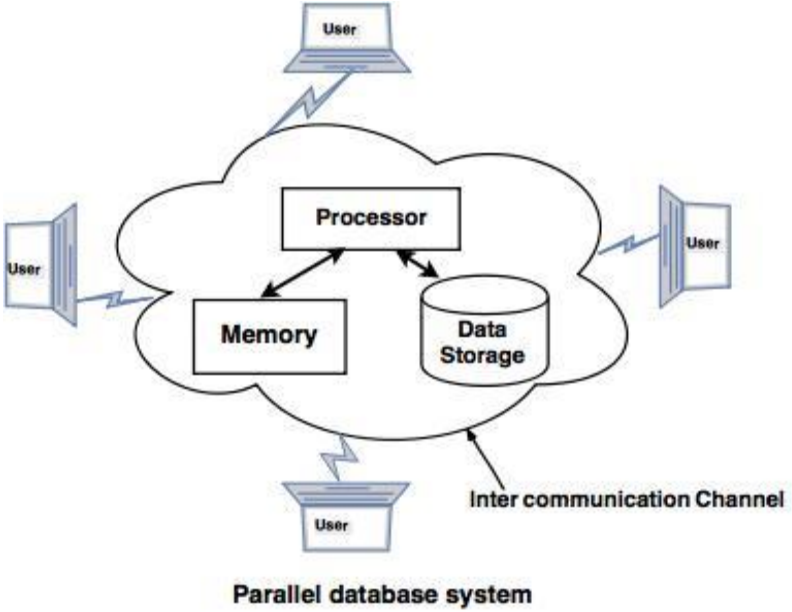
Database Security Measures

- **Database Activity Monitoring (DAM):**
  - ✓ Deploy DAM solutions to monitor real-time database activity and detect suspicious or unauthorized behavior.
  - ✓ DAM tools can help identify potential security incidents, insider threats, and unauthorized access attempts.
- **Backup and Recovery:**
  - ✓ Implement regular data backup and recovery processes to ensure data availability and resilience in the event of data loss, corruption, or security incidents.
  - ✓ Store backup copies securely and test the recovery procedures time to time to validate their effectiveness.
- **Database Firewall:**
  - ✓ Deploy database firewalls to enforce access controls, monitor database traffic, and prevent unauthorized access.
  - ✓ Database firewalls can help detect and block malicious activities targeting the database system.

Parallel Databases

- Parallel DBMS is a Database Management System that runs through multiple processors and disks. They combine two or more processors also disk storage that helps make operations and executions easier and faster. They are designed to execute concurrent operations.





Parallel database system

Advantages of Parallel Databases

- **Speed**
  - ✓ The first advantage of parallel DBMS is speed.
  - ✓ The servers from parallel DBMS are able to break up user database request into parts and it dispatches each of the parts to separate computers.

- **Reliability**
  - ✓ A parallel database when properly configured, can continue to work despite the failure of any computer in the cluster.
  - ✓ The database server can sense that a specific computer is not responding, and it can reroute its work to the remaining computers.

- **Capacity:**
  - ✓ As more and more users request access to the database, the computer administrators add more computers to the parallel server, boosting its overall capacity to the max, databases are more likely to slow down and have slower system.

Disadvantages of Parallel Databases

- **Cost**
  - ✓ The first disadvantage of parallel DBMS is cost. As the need of quicker processing and efficient searches increase, there need to be more and more disks and processors that simultaneously work together to achieve the best and quickest results.
  - ✓ To do that, Parallel DBMS needs lots of processors and disks in the first place. At the end, it is never cheap to implement parallel DBMSs.
- **Resources**
  - ✓ The second disadvantage of parallel DBMS is the huge amount of resources.
  - ✓ It is not and will never be easy to keep up with modern, cheap, and efficient resources.
- **Difficulty of Managing Systems**
  - ✓ The third disadvantage of parallel DBMS is the difficulty in systems managing.
  - ✓ When having lots of systems, lots of resources, and lots of systems running it is never easy to manage them.
  - ✓ When there needs to be a software update, a replacement, or maintenance that all of the system needs to do, it will be time consuming and resource consuming.

Distributed Databases

- A distributed database is basically a database that is not limited to one system, it is spread over different sites, i.e, on multiple computers or over a network of computers.
- A distributed database system is located on various sites that don't share physical components.
- This may be required when a particular database needs to be accessed by various users globally.
- It needs to be managed such that for the users it looks like one single database.

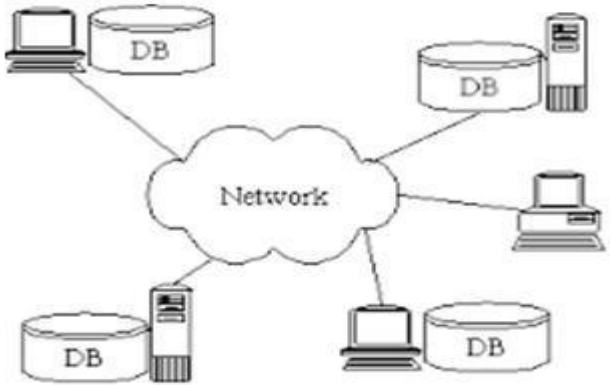
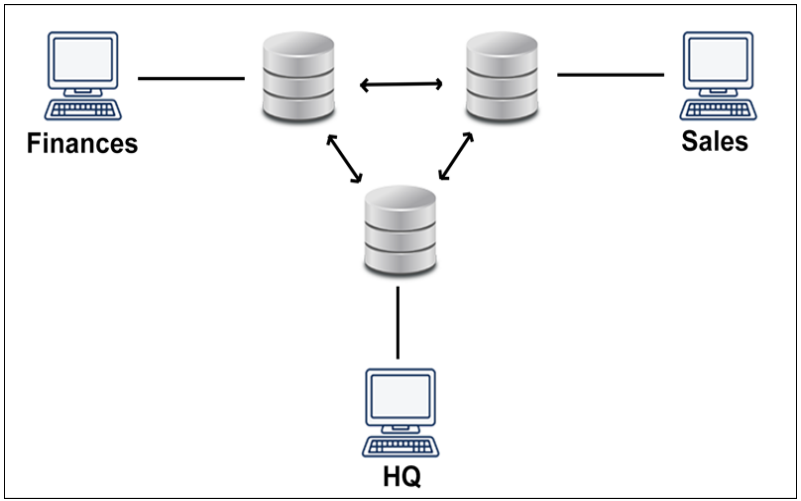


Figure. Distributed database System



Advantages of Distributed Database System

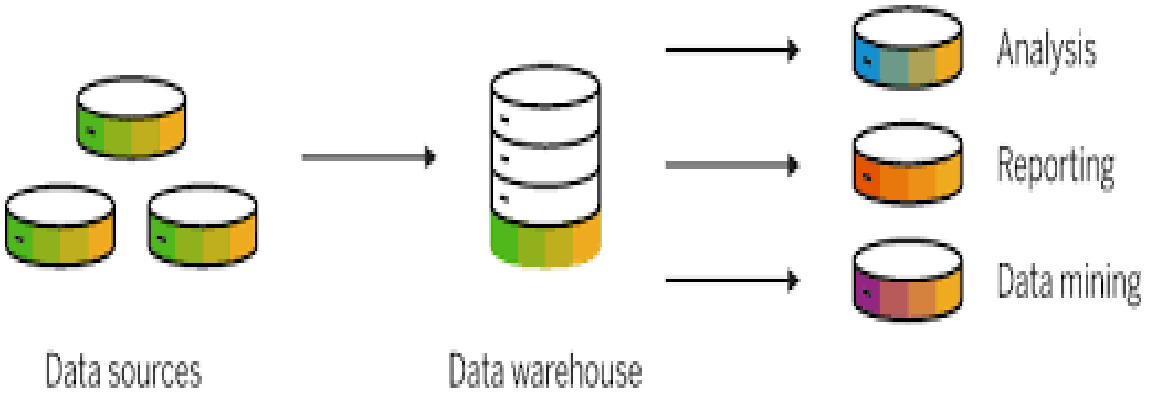
- There is fast data processing as several sites participate in request processing.
- Reliability and availability of this system is high.
- It possess reduced operating cost.
- It is easier to expand the system by adding more sites.
- It has improved sharing ability.

Disadvantages of Distributed Database System

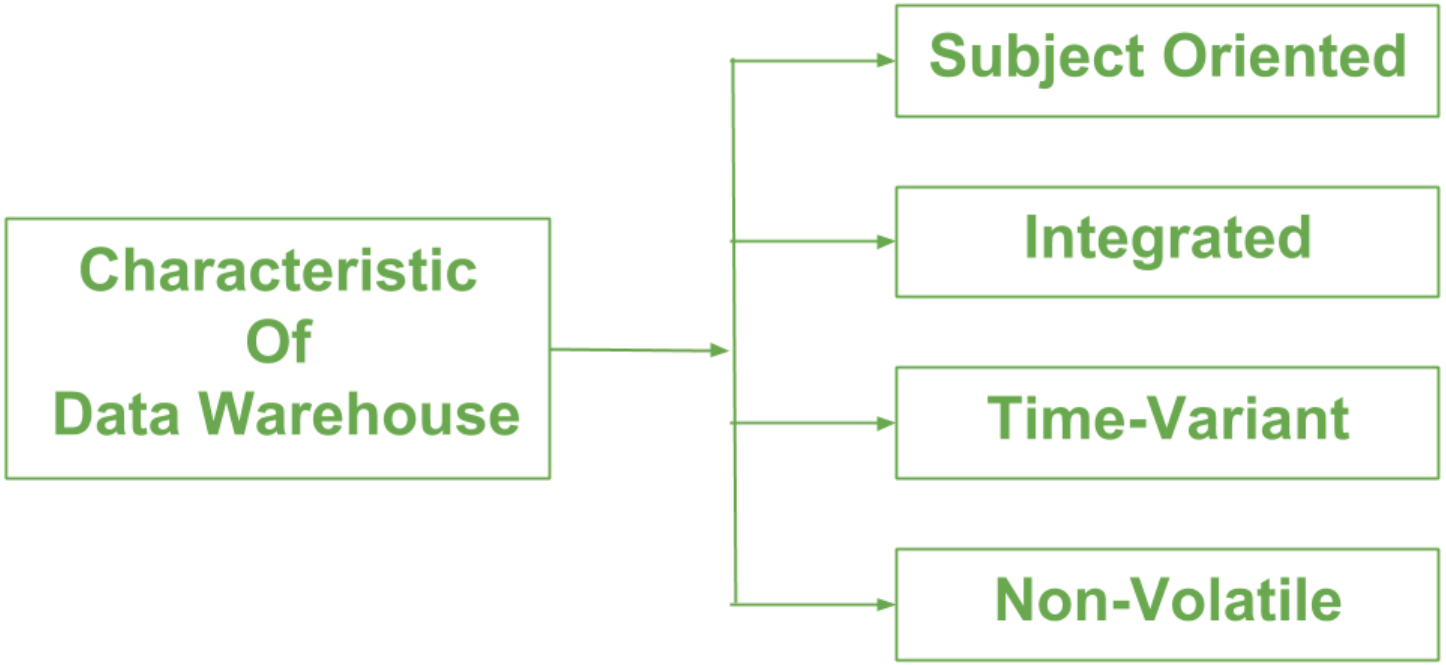
- The system becomes complex to manage and control.
- The security issues must be carefully managed.
- The system require deadlock handling during the transaction processing otherwise the entire system may be in inconsistent state.

Data Warehouse

- A data warehouse is a large collection of business data used to help an organization make decisions.
- Its purpose is to feed business intelligence, reporting, and analytics, and support regulatory requirements.
- Data warehouse store current and historical data in one place and act as the single source of truth for an organization.

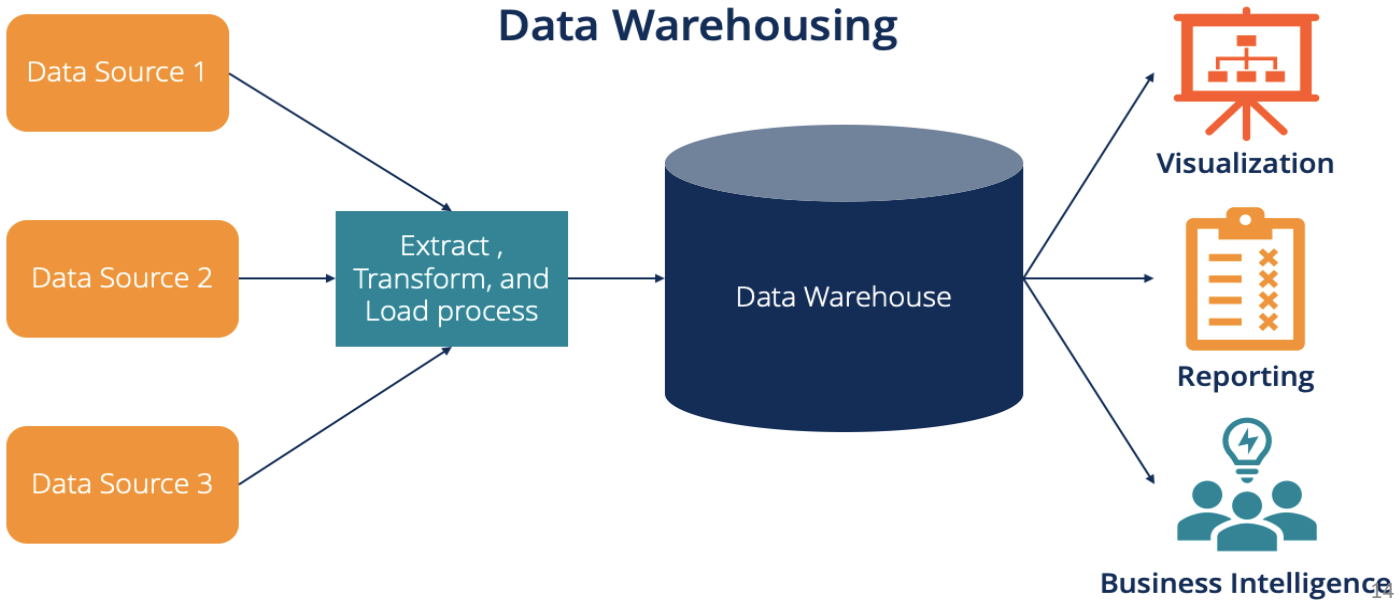


Features of Data Warehouse



Data Warehousing

- Data warehousing is the process of constructing and using the data warehouses.
- It is the process of extracting and transferring operational data into informational data and loading it into a certain data store.



Benefits of data warehousing

- Better business analytics
- Faster queries
- Improved data quality
- Historical insight

Data Mining

The process of extracting information to identify pattern, trends, and useful data that would allow the business to take the data-driven decision from huge sets of data is called data mining.

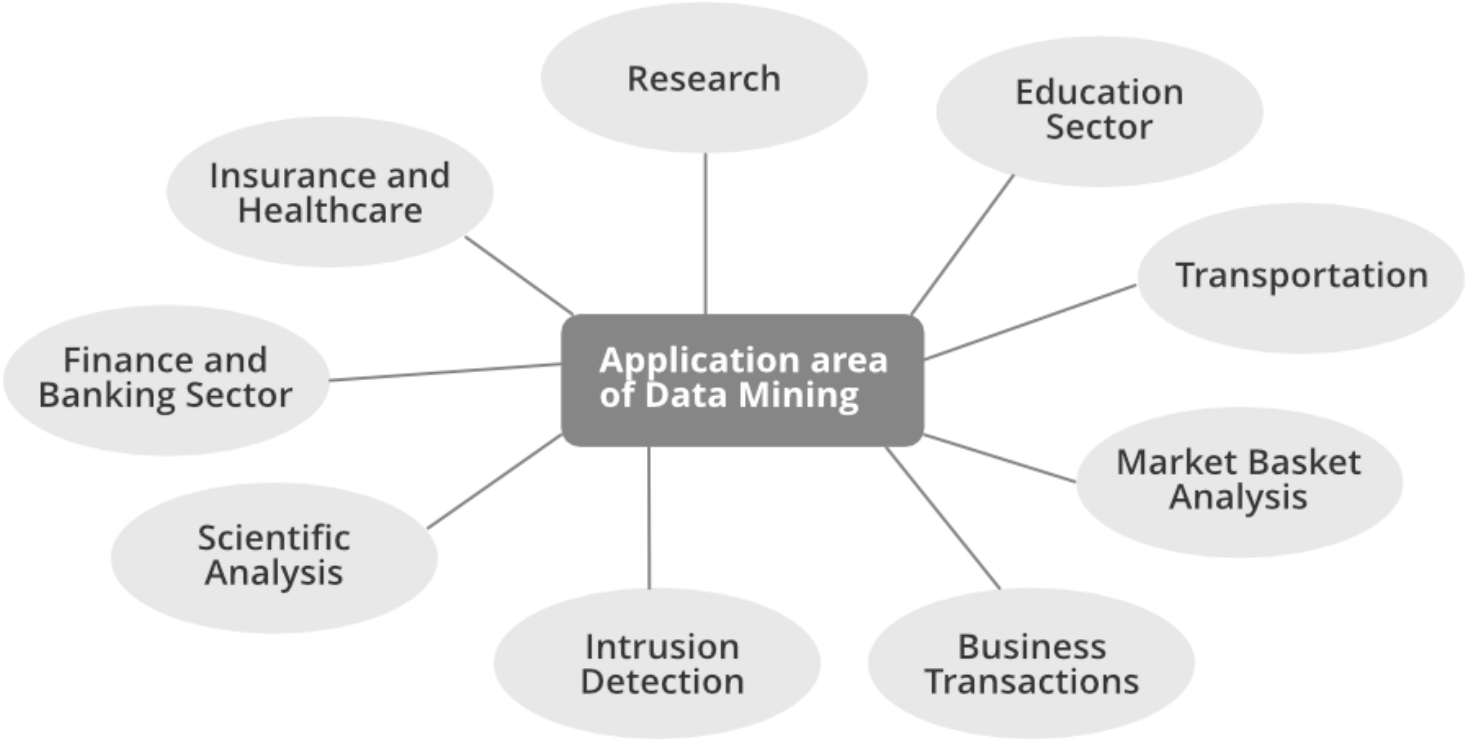
Advantages of Data Mining

- It helps companies gather reliable information
- Helps to analyze huge data sets.
- It helps businesses make profitable production and operational adjustments
- Contributes to the making of the important decision
- Compresses data into valuable information.
- Aids companies to find, attract and retain customers.

Drawbacks of Data Mining:

- Many data analytics tools are complex and challenging to use. Data scientists need the right training to use the tools effectively.
- The requirement of large investments can also be considered a problem as sometimes data collection consumes many resources that suppose a high cost.
- Data mining techniques are not infallible, so there’s always the risk that the information isn’t entirely accurate, and those inaccurate data may lead to the wrong output.
- Companies can potentially sell the customer’s data they have extracted to the other businesses and organizations, raising privacy concerns.
- Huge databases are quite difficult to manage.

Application areas of Data Mining:

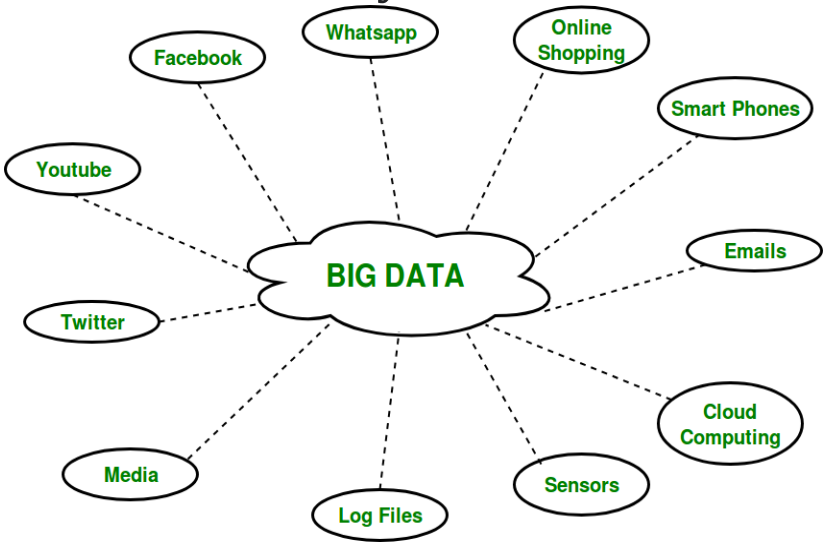


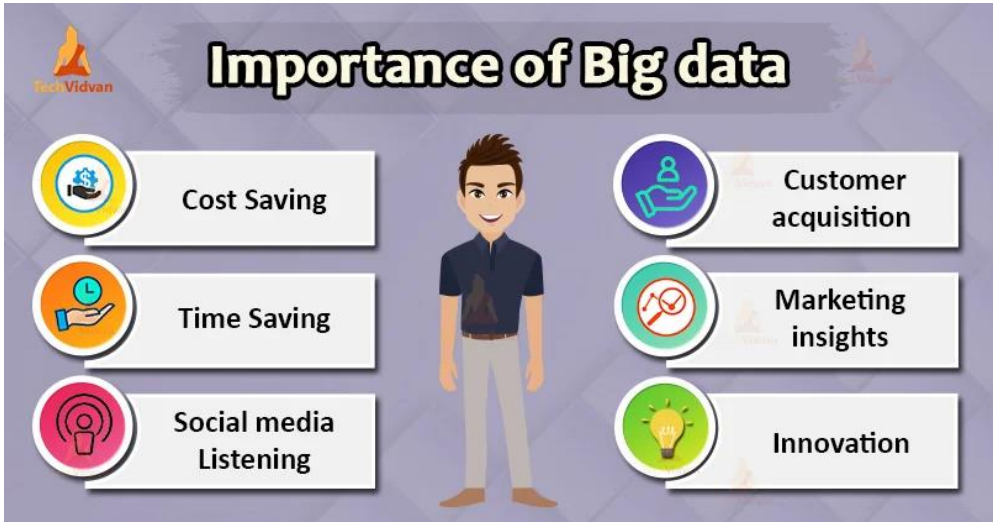
DBMS v/s Data Warehouse

DBMS	Data Warehouse
• Operational systems are designed to support high volume Online Transaction Processing(OLTP).	• Data warehousing systems are typically designed to support high volume Online Analytical Processing(OLAP).
• Operational systems are usually concerned with current data.	• Data warehousing systems are usually concerned with historical data.
• Less number of data accessed.	• Large number of data is accessed.
• It is used to perform day to day operations	• It is used for decision support.
• ER model is used for designing	• Star, Snowflake or Fact Constellation model is used for designing.

Big Data

- Big data refers to extremely large and complex datasets that are difficult to process and analyze using traditional data processing methods and tools.
- Big data is often characterized by the three V's:
  - ✓ the large **volume** of data in many environments;
  - ✓ the wide **variety** of data types frequently stored in big data systems; and
  - ✓ the **velocity** at which much of the data is generated, collected and processed.





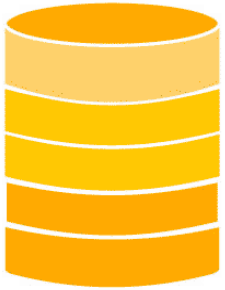
What is Big Data?

Big Data=Enormous amounts and various types of data that is collected for a certain purpose

Volume

Enormous amounts of data reaches a few terabytes to dozens of terabytes or more

Enormous capacity



Variety

Various data formats, such as documents, images, audio, videos, and search and browsing history

Velocity

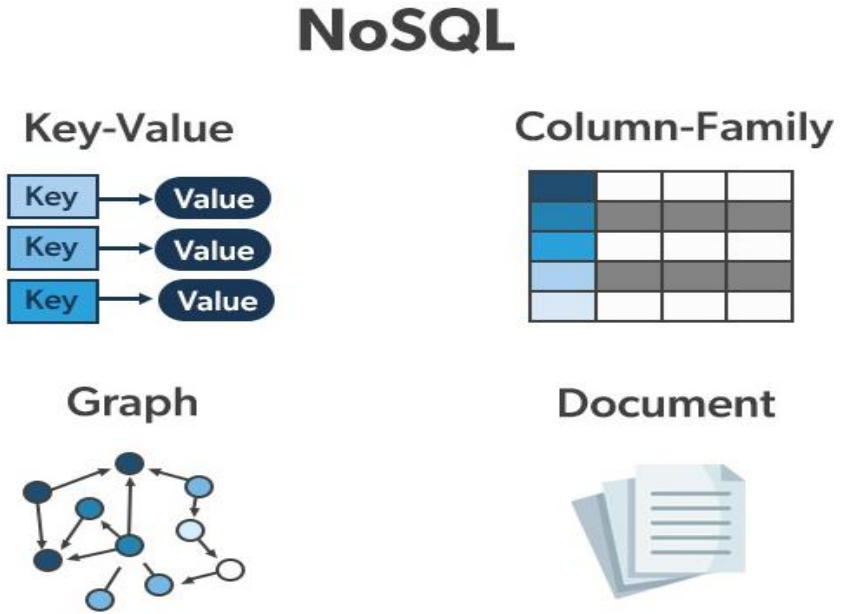
Real-time updates

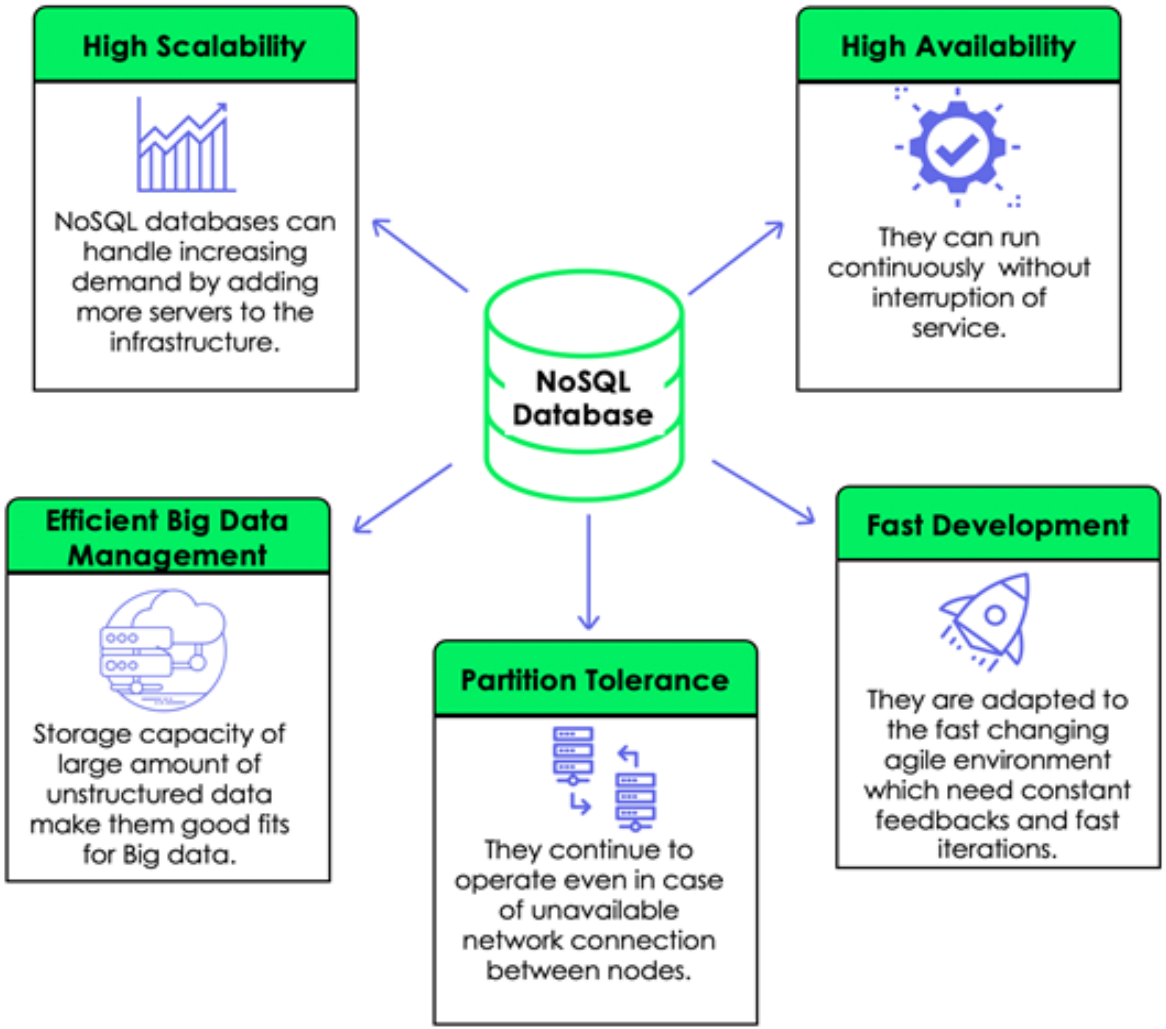
6 big data benefits for businesses

1. Better customer insight
2. Improved operations
3. More insightful market intelligence
4. Agile supply chain management
5. Data-driven innovation
6. Smarter recommendations and targeting

NoSQL Databases

- NoSQL is a type of database management system (DBMS) that is designed to handle and store large volumes of unstructured and semi-structured data.
- NoSQL databases use flexible data models that can adapt to changes in data structures and are capable of scaling horizontally to handle growing amounts of data.




High Scalability
NoSQL databases can handle increasing demand by adding more servers to the infrastructure.
High Availability
They can run continuously without interruption of service.
NoSQL Database
Efficient Big Data Management
Storage capacity of large amount of unstructured data make them good fits for Big data.
Partition Tolerance
They continue to operate even in case of unavailable network connection between nodes.
Fast Development
They are adapted to the fast changing agile environment which need constant feedbacks and fast iterations.

Types of NoSQL Databases

- **Document databases** store data in documents similar to JSON (JavaScript Object Notation) objects or XML documents. Each document contains pairs of fields and values. The values can typically be a variety of types including things like strings, numbers, booleans, arrays, or objects. **Examples**: MongoDB, CouchDB, Cloudant
- **Key-value databases** are a simpler type of database where each item contains keys and values. **Examples**: Memcached, Redis, Coherence
- **Wide-column** stores store data in tables, rows, and dynamic columns. **Examples**: Hbase, Big Table, Accumulo
- **Graph databases** store data in nodes and edges. Nodes typically store information about people, places, and things, while edges store information about the relationships between the nodes. **Examples**: Amazon Neptune, Neo4j

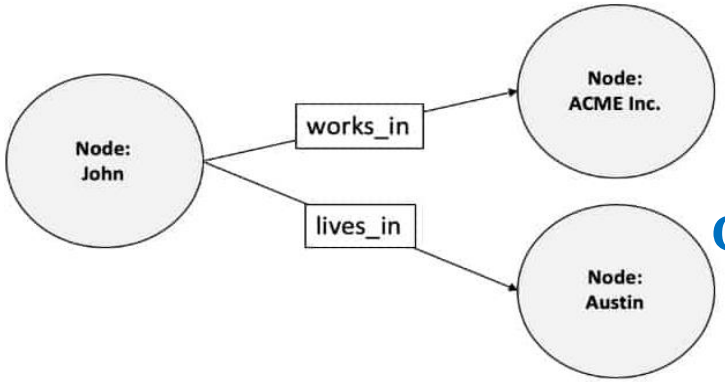
Examples of NoSQL Databases


```
{
  "_id": 1,
  "first_name": "Leslie",
  "last_name": "Yepp",
  "cell": "8125552344",
  "city": "Pawnee",
  "hobbies": ["scrapbooking", "eating waffles", "working"]
}
```

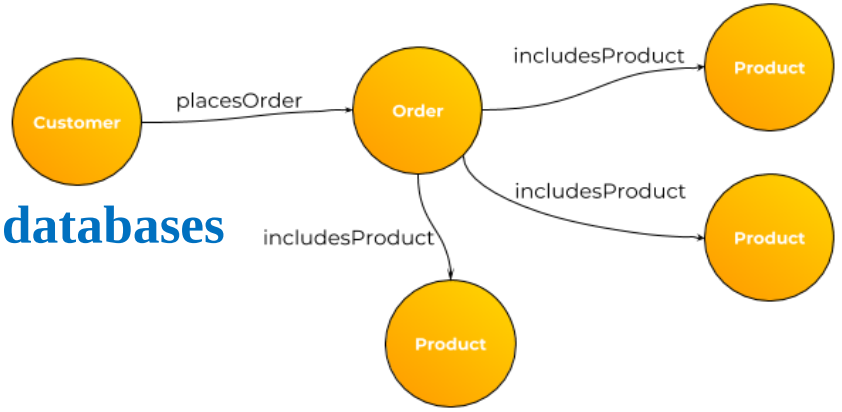

Document Database

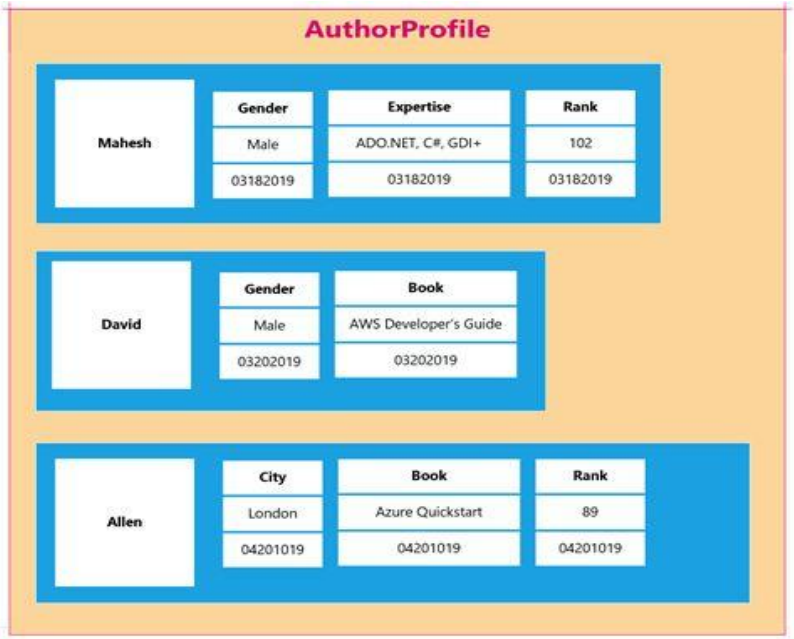
Key	Value
Name	Raman Sharma
Account Number	21657243
Amount	567543

Key-value databases



Graph Based databases





Wide Column Based databases

101	email	name	tel	
	ab@c.to	otto	12345	
103	email	name	tel	tel2
	karl@a.b	karl	6789	12233
104	name			
	linda			

Features of NoSQL Databases

- Each NoSQL database has its own unique features. At a high level, many NoSQL databases have the following features:
  - ✓ Flexible schemas
  - ✓ Horizontal scaling
  - ✓ Fast queries due to the data model
  - ✓ Ease of use for developers

When should NoSQL be used?

- When a huge amount of data needs to be stored and retrieved.
- The relationship between the data stored is not that important.
- The data changes over time and is not structured.
- Support of Constraints and Joins is not required at the database level.
- The data is growing continuously and we need to scale the database regularly to handle the data.

End!