

**SIX WEEKS SUMMER TRAINING REPORT ON  
MODERN BIG DATA ANALYSIS WITH SQL**

**Under the Guidance of**

**Mr. Glynn Durham, Mr. Lan Cook**

**(Coursera, Cloudera)**

**A Summer Training report**

Submitted in partial fulfillment of the requirements for the award of degree of

**B.Tech. (CSE)**

**Submitted to**

**School of Computer Science & Engineering**

**LOVELY PROFESSIONAL UNIVERSITY**



**PHAGWARA, PUNJAB**

**From May 2021 to July 2021**

**SUBMITTED BY**

**Name of student:**

Gaganvir Kaur

**Registration Number:**

11910883

## **Table of Contents**

<b>S. No.</b>	<b>Title</b>	<b>Page</b>
1	Declaration	3
2	Training Certification from organization	4
3	Acknowledgement	8
4	Diagrams/Figures	9
5	Introduction of the Course Undertaken	12
6	Brief description of the work done	14
7	Learning Outcomes	51
8	Conclusion	52
9	References	53

## **Declaration**

I, **Gaganvir Kaur, 11910883** hereby declare that I have completed my six weeks summer training at **Coursera, Cloudera** platform on **Modern Big Data Analysis with SQL** from **May 5, 2021** to **July 10, 2021** under the guidance of Mr. Glynn Durham and Mr. Lan Cook. I have declare that I have worked with full dedication during these 6 weeks of training and my learning outcomes fulfill the requirements of training for the award of degree of B.Tech. CSE , Lovely Proffesional University, Phagwara.

Gaganvir Kaur (11910883)

Dated: 26 August, 2021

## Summer Training Certificates from Coursera, Cloudera



3 Courses

Foundations for Big Data Analysis with SQL

Analyzing Big Data with SQL

Managing Big Data in Clusters and Cloud Storage



Jul 10, 2021

**Gaganvir Kaur**

has successfully completed the online, non-credit Specialization

### Modern Big Data Analysis with SQL

In this Specialization, learners acquired essential knowledge and skills for data analysis with SQL using open source distributed big data systems. Through a sequence of three courses, learners gained knowledge of the fundamental concepts behind relational databases, SQL, and big data; learned how to write and run SQL queries using query engines including Apache Hive and Apache Impala; and learned how to manage large-scale data in clusters and cloud storage using the Hadoop Distributed File System (HDFS) and Amazon Simple Storage Service (S3).

The online specialization named in this certificate may draw on material from courses taught on-campus, but the included courses are not equivalent to on-campus courses. Participation in this online specialization does not constitute enrollment at this university. This certificate does not confer a university grade, course credit or degree, and it does not verify the identity of the learner.



Glynn Durham  
Senior Instructor  
Cloudera



Ian Cook  
Staff Curriculum  
Developer  
Cloudera

Verify this certificate at:  
[coursera.org/verify/specialization/Q2GGJZL7BLK3](https://coursera.org/verify/specialization/Q2GGJZL7BLK3)

# CLOUDERA

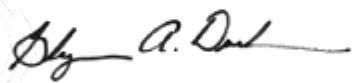
May 26, 2021

**Gaganvir Kaur**

has successfully completed with honors

**Foundations for Big Data Analysis with SQL**

an online non-credit course authorized by Cloudera and offered through Coursera



Glynn Durham  
Senior Instructor  
Cloudera

COURSE  
CERTIFICATE

WITH HONORS



Verify at [coursera.org/verify/S3A8DSXY8GUH](https://coursera.org/verify/S3A8DSXY8GUH)

Coursera has confirmed the identity of this individual and their participation in the course.

# CLOUDERA

Jun 24, 2021

**Gaganvir Kaur**

has successfully completed

**Analyzing Big Data with SQL**

an online non-credit course authorized by Cloudera and offered through Coursera



Ian Cook  
Staff Curriculum Developer  
Cloudera

COURSE  
CERTIFICATE



Verify at [coursera.org/verify/KE9YNSGU2BRA](https://coursera.org/verify/KE9YNSGU2BRA)

Coursera has confirmed the identity of this individual and their participation in the course.

# CLOUDERA

Jul 10, 2021

**Gaganvir Kaur**

has successfully completed with honors

**Managing Big Data in Clusters and Cloud Storage**

an online non-credit course authorized by Cloudera and offered through Coursera

COURSE  
CERTIFICATE

WITH HONORS



A handwritten signature in black ink, reading "Jan Cook" followed by a stylized flourish.

Jan Cook  
Staff Curriculum Developer  
Cloudera

Glynn Durham  
Senior Instructor  
Cloudera

Verify at [coursera.org/verify/BXWBWYFJ6ARG](https://coursera.org/verify/BXWBWYFJ6ARG)

Coursera has confirmed the identity of this individual and their participation in the course.

## **Acknowledgement**

I would like to express my gratitude towards my University as well as Coursera and Cloudera for providing me the golden opportunity to do this wonderful summer training regarding Big Data and SQL, which also helped me in doing a lot of homework and learning. As a result, I came to know about so many new things. So, I am really thank full to them.

Moreover I would like to thank my friends who helped me a lot whenever I got stuck in some problem related to my course. I am really thankfull to have such a good support of them as they always have my back whenever I need.

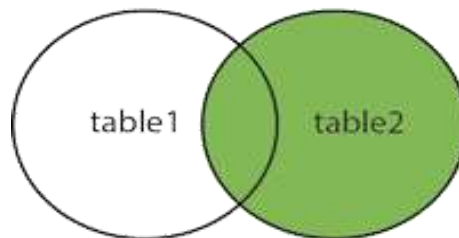
Also,I would like to mention the support system and consideration of my parents who have always been there in my life to make me choose right thing and oppose the wrong. Without them I could never had learned and became a person who I am now.

I have taken efforts in this course. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

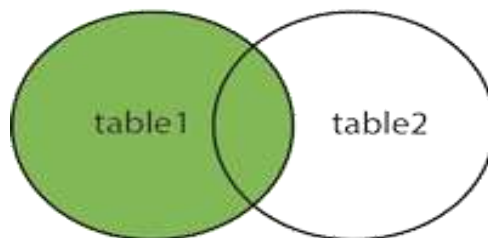


## List of Diagrams/Figures

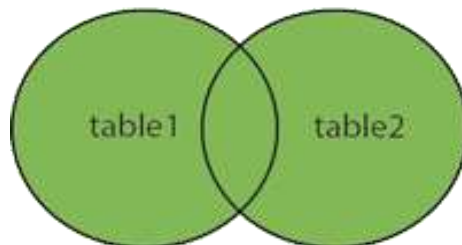
RIGHT JOIN



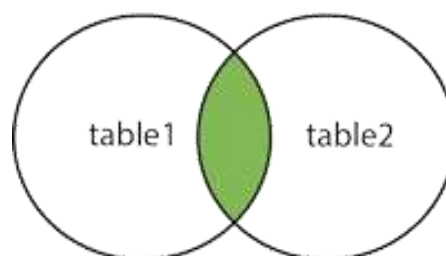
LEFT JOIN

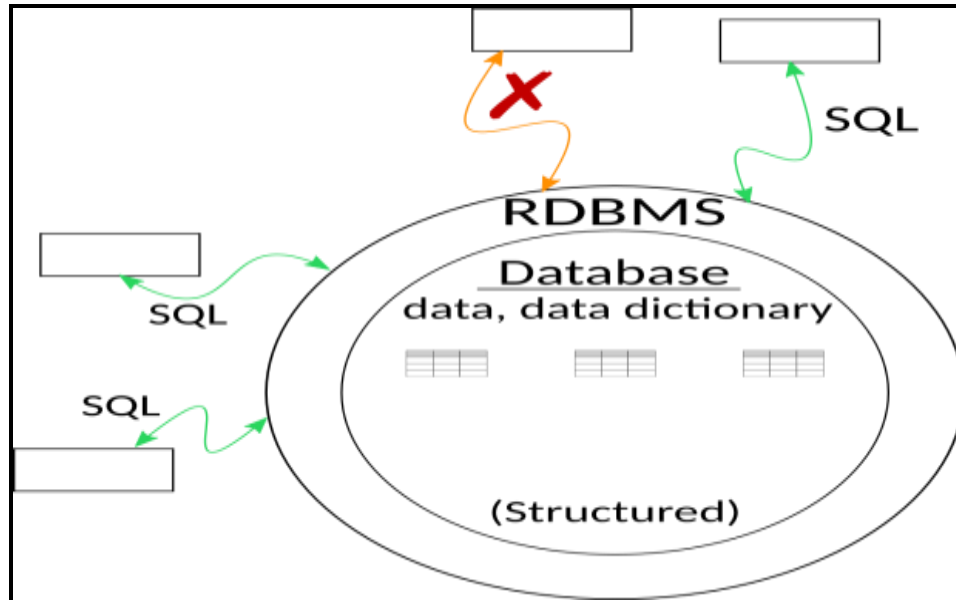


FULL OUTER JOIN

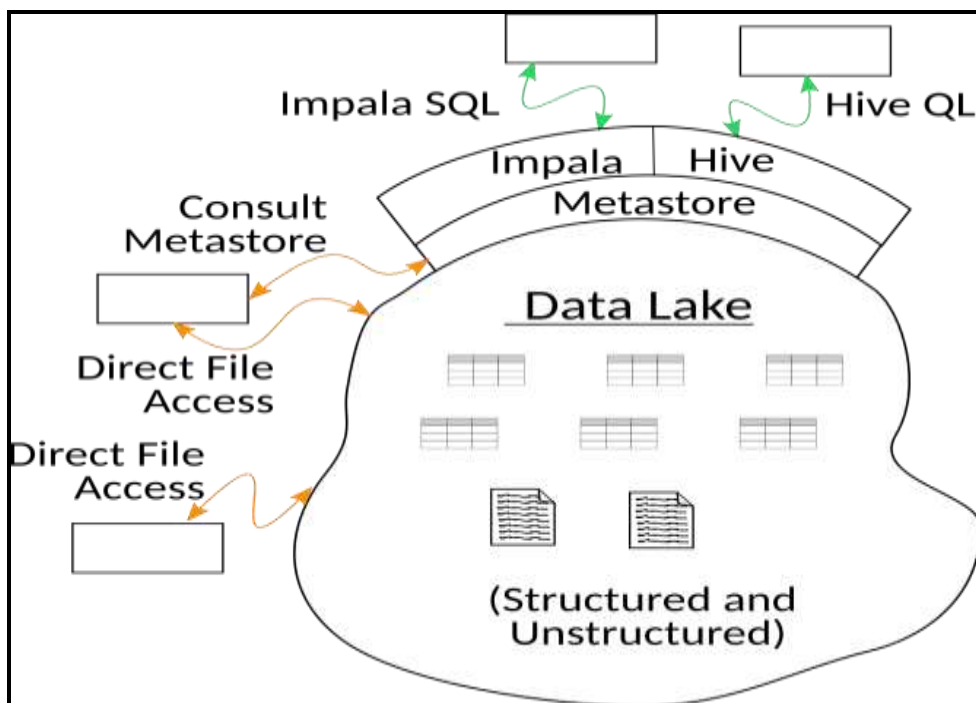


INNER JOIN

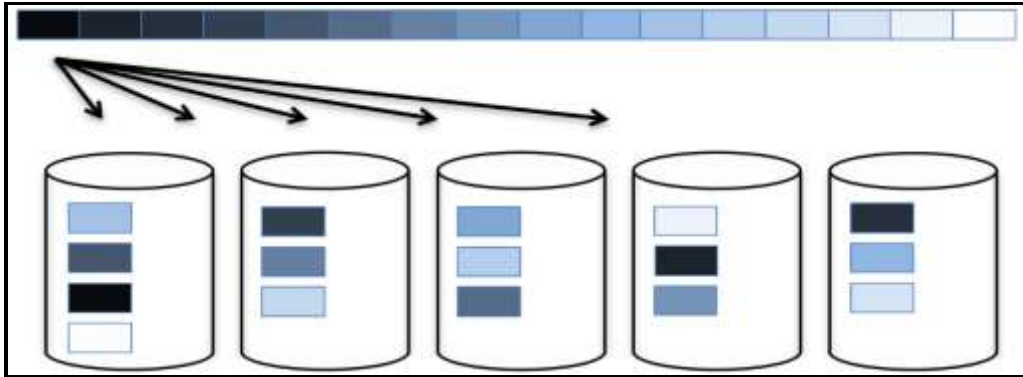




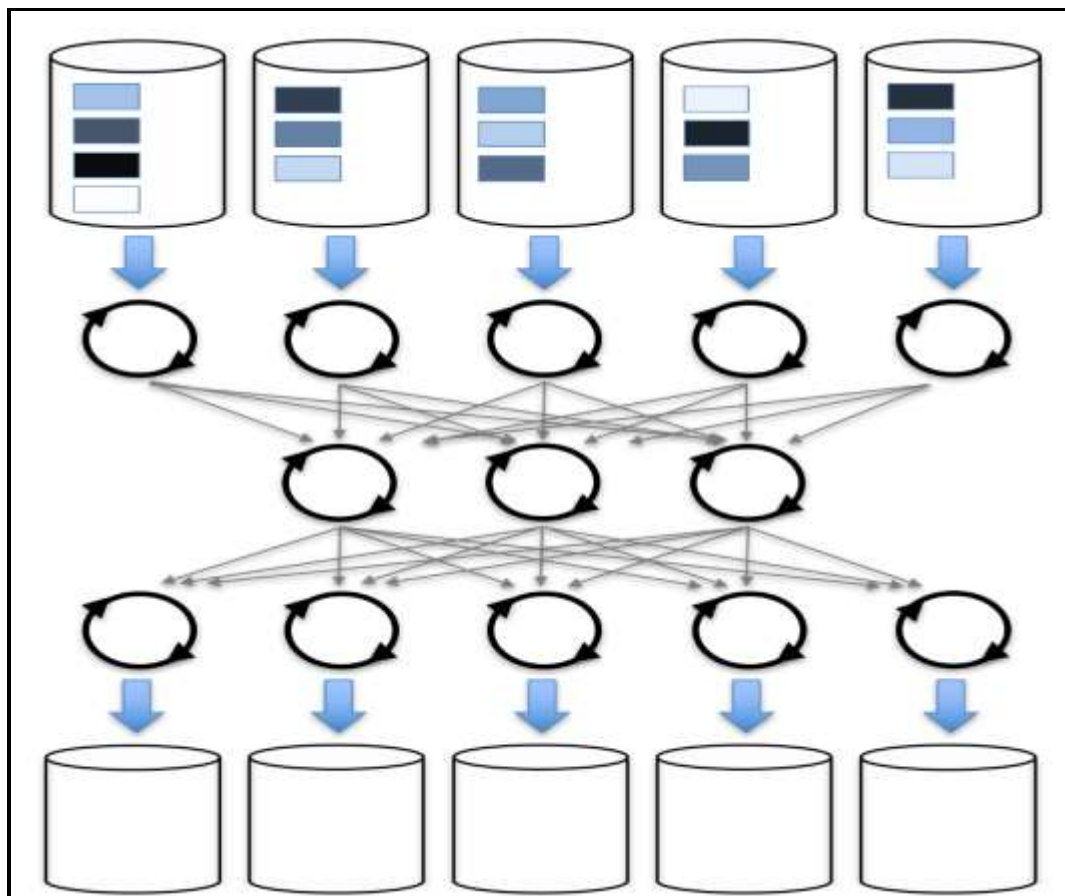
**Structured Data (Storing and Accessing Data, Comparison)**



**Structured and Unstructured Data**



**Distributed storage and processing(file split into pieces)**



**Distributed Storage and processing System**

## Introduction of the Course Undertaken

“**Modern Big Data Analysis with SQL Specialization**” course is a complete package of three courses that helped me to learn Big Data from Basic to an Advance level. All three courses further divided into six-six weeks, where I practiced questions and I have attempted the assessment tests, quizzes accordingly. The modules of this course consist of the following elements:-Video-quizzes, Graded quizzes, Peer-graded Assignments, Discussion forums. The course contains eight graded quizzes, which will count towards my course score. I must have to answer at least 60% of the quiz questions in each graded quiz correctly to pass this course. The course was Self placed means I could join the course anytime and all the content will be available to me once I get enrolled.

After completing each course, I have earned course completion certificate. There were video lectures to learn and multiple-choice questions and quizzes to practice. In this course, I have used **new SQL engine**. These are open-source SQL engines capable of querying enormous datasets. This course focuses on **SQL engine** the most widely deployed of these query engines. This course is designed to provide excellent preparation for the **Cloudera Certified Associate(CCA) Data Analyst** certification exam.

In this course of Specialization first course is “**Foundations for Big Data Analysis with SQL**”. In this course, I have learned to use **SQL** for big data, starting with an overview of data, database systems, and the common querying language (SQL). Then I have learned the characteristics of big data and SQL tools for working on big data platforms. I'll also install an exercise environment (virtual machine) to be used through the specialization courses, and I'll have an opportunity to do some initial exploration of databases and tables in that environment. In this course, I have learned operational from analytic databases and understand how these are applied in big data; understand how database and table design provides structures for working with data; appreciate how differences in volume and variety of data affects your choice of an appropriate database system; recognize the features and benefits of SQL dialects designed to work with big data systems for storage and

analysis and explore databases and tables in a big data platform. To use the hands-on environment for this course, I have download and install a virtual machine and the software on which to run it.

In this course of Specialization second course is **“Analyzing Big Data with SQL”**. In this course, I have learned the SQL SELECT statement and its main clauses. The course focuses on **big data SQL engines** but most of the information is applicable to SQL with traditional RDBMs as well, the instructor explicitly addresses differences for MySQL and PostgreSQL. In this course I have learn to explore and navigate databases and tables using different tools; understand the basics of SELECT statements; understand how and why to filter results; work with sorting and limiting results and combine multiple tables in different ways. The sequence of courses in this specialization is designed to provide excellent preparation for the Cloudera Certified Associate Data Analyst certification exam. This certification was created to identify qualified data analysts, with a talent for using SQL to analyse big data. It's a great way to stand out and be recognized by potential employers.

In this course of Specialization third course is **“Managing Big Data in Clusters and Cloud Storage”**. In this course, I have learned how to manage big datasets, how to load them into clusters and cloud storage, and how to apply structure to the data so that we can run queries on it using distributed SQL engines. I have learned how to choose the right data types, storage systems, and file formats. I have learned to use different tools to browse existing databases and tables in big data systems, to use different tools to explore files in distributed big data filesystems and cloud storage, to create and manage big data databases and tables using Apache Hive and Apache Impala to describe and choose among different data types and file formats for big data system.

**First Course: - Foundations for Big Data Analysis with SQL**

**Data:** In this specialization, data means digital data. Information that can be transmitted, stored, and processed using modern digital technologies, like the Internet, disk drives, and modern computer. Data is a representation of something that captures some features or ignore others. Data is a collection of facts and it is unorganized or unprocessed. It generally includes the raw forms of numbers, statements and characters. It can be anything like name, place or number etc. It does not help in decision making. Data is a representation of something that captures some features and ignores others.

**Examples of Data:**

- Number of Questions Answered in a Paper.
- Total Number of Upvotes.
- Student's name in a class are Data.
- Student's subject marks are Data.
- Total Number of Views Received.
- Information collected for writing a research paper is data.

**Data is of two types:**

- Analog Data
- Digital Data

**Analog Data:** Analog Data is a data represented in a physical way. Analog data may also be known as Organic data or real-world data.

For example: Physical movements of objects can be modeled in a spatial simulation and real-time audio and video can be captured using a range of systems and devices.

**Digital Data:** Digital data is the data that is recorded and sent to another device. It is a set of individual symbols.

For example: All devices using digital data like Computers, Laptops, IPads, Mobile Phone, MP3 Player, Digital Camera etc.

**Need of Organize Data:** The need of Organize Data is necessary because unorganized information has no meaning. There are a lot of benefits to organizing data. The first and foremost benefit are it decreases the time consumed to search for data. Disorganized data has many bottlenecks in terms of data structuring. Suppose you have a data of the results of 1000 students in a school and you need to find out how many students scored a percentage greater than 90. If your data is unorganized, it will take a lot of time and resource to gather the required information, but suppose you have organized the data in descending order of percentages, and then it will be very quick and easy to sort out the required information. Organizing data also helps in reducing data loss and reduces errors. Suppose you have confusion in different sets of data, then the only solution to such problems is organizing the data properly. Data organization also helps you to understand why the data was collected and what the proper use of it is. Once the data is organized, it gives you the validity of the work undertaken. A sequential view of the data is always accepted as compared to abrupt and disorganized view. Data organization can be of various types, depending on the requirement of the user. Sometimes, the repeated values in the data are collected together to know the mode of the data or sometimes the data is organized in increasing or decreasing order, to find the median of the given set of data.

**Data Store:** Data Store is a collection of data.

Examples: Collection of photos, videos, texts in cloud storage.

**Data Base:** Data base is an organized data store. A database is a shared collection of logically related data designed to meet the information needs of an organization. The related information when placed in an organized form makes a database.

Example: Spreadsheet.

**Database Management System (DBMS):** It is a software which helps in organizing a data.

It is a software system that allows users to define, create and maintain a database and provides controlled access to the data. It is basically a collection of programs that enables users to store, modify and extract information from a database as per the requirements. DBMS can solve all your data organizing problems. It is a software that allows systematic organization of data in one or more databases.

### **Working with Database Management System:**

#### **Operations on Databases:**

- To add new information.
- To view or retrieve the stored information.
- To modify or edit the existing.
- To remove or delete the unwanted information.
- Arranging the information in a desired ordered etc.

**Components of Database:** There are five major components in database system environment are:

- Hardware
- Software
- Data
- Users
- Procedure

**Hardware:** It is the actual computer system used for keeping and accessing the database. DBMS hardware consists of secondary storage dev like hard disks.



**Software:** It is the actual DBMS. Between the physical database itself and the users of system is a layer of software, called DBMS.

**Data:** Data acts as the bridge between the machine components and user components.

**Users:** There are number of users who can access or retrieve data on demand using the applications and the interfaces provided by DBMS. The users can be:

- Naive users
- Online users
- Application Programmers
- Sophisticated Users
- Data base Administrator (DBA)

**Procedures:** It refers to the instructions and rules that govern the design and the use of the database. The users of the system and the staff that manage the database requires documented procedures on how to use or run the system.

**Four general activities that can be performed by DBMS:**

- Design
- Update
- Retrieve
- Manage

**Design:** Designing a database includes where and how the things must be setup.

**Update:** Updating includes Adding data, removing data, deleting data, changing data.

**Retrieve:** Retrieving data includes finding answers to many questions.

**Manage:** Managing a data needs a control access to your data.

## **DBMS Functions:**

A DBMS performs several important functions that guarantee the integrity and consistency of the data in the database. Most of those functions are transparent to end users, and most can be achieved only through the use of a DBMS. They include data dictionary management, data storage management, data transformation and presentation, security management, multiuser access control, backup and recovery management, data integrity management, database access languages and application programming interfaces and database communication interfaces.

Each of these functions is explained below.

### **1.Data dictionary management:**

The DBMS stores definitions of the data elements and their relationships (metadata) in a data dictionary. In turn, all programs that access the data in the database work through the DBMS. The DBMS uses the data dictionary to look up the required data component structures and relationships, thus relieving you from having to code such complex relationships in each program. Additionally, any changes made in a database structure are automatically recorded in the data dictionary, thereby freeing you from having to modify all of the programs that access the changed structure. In other words, the DBMS provides data abstraction, and it removes structural and data dependence from the system.

### **2. Data transformations and presentation:**

The DBMS transforms entered data to conform to required data structures. The DBMS relieves you of the chore of making a distinction between the logical data format and the physical data format. That is, the DBMS formats the physically retrieved data to make it conform to the user's logical expectations. For example, imagine an enterprise database used by a multinational company. An end user in England would expect to enter data such as July 11, 2010, as "11/07/2010." In contrast, the same date would be entered in the United States as "07/11/2010." Regardless of the data presentation format, the DBMS must manage the date in the proper format for each country.

### **3. Security Management:**

The DBMS creates a security system that enforces user security and data privacy. Security rules determine which users can access the database, which data items each user can access, and which data operations (read, add, delete, or modify) the user can perform. This is especially important in multiuser database systems.

### **4. Multiuser access control:**

To provide data integrity and data consistency, the DBMS uses sophisticated algorithms to ensure that multiple users can access the database concurrently without compromising the integrity of the database.

### **5. Backup and recovery management:**

The DBMS provides backup and data recovery to ensure data safety and integrity. Current DBMS systems provide special utilities that allow the DBA to perform routine and special backup and restore procedures. Recovery management deals with the recovery of the database after a failure, such as a bad sector in the disk or a power failure. Such capability is critical to preserving the database's integrity.

### **6. Data integrity management:**

The DBMS promotes and enforces integrity rules, thus minimizing data redundancy and maximizing data consistency. The data relationships stored in the data dictionary are used to enforce data integrity. Ensuring data integrity is especially important in transaction-oriented database systems.

### **7. Database access languages and application programming interfaces:**

The DBMS provides data access through a query language. A query language is a nonprocedural language one that lets the user specify what must be done without having to specify how it is to be done. Structured Query Language (SQL) is the de facto query language and data access standard supported by the majority of DBMS vendors.

## **8. Database communication interfaces:**

Current-generation DBMSs accept end-user requests via multiple, different network environments.

For example, the DBMS might provide access to the database via the Internet through the use of Web browsers such as Mozilla Firefox or Microsoft Internet Explorer. In this environment, communications can be accomplished in several ways:

- End users can generate answers to queries by filling in screen forms through their preferred Web browser.
- The DBMS can automatically publish predefined reports on a website.
- The DBMS can connect to third-party systems to distribute information via e-mail or other productivity applications.

## **Applications of DBMS**

- **Banking:** all transactions
- **Airlines:** reservations, schedules
- **Universities:** registration, grades
- **Sales:** customers, products, purchase
- **Online retailers:** order tracking, customized recommendations
- **Manufacturing:** production, inventory, orders, supply chain
- **Human resources:** employee records, salaries, tax deductions

## **Data models, Schemas, and Instances**

**Data model:** A set of concepts to describe the structure of a database, and certain constraints that the database should obey.

**Schema:** The overall description of the database is called the Database Schema. A schema is defined as an outline or a plan that describes the records and relationships existing at the particular level.

**Instance:** Data in the database at a particular moment in time.

**Table:** a table is a collection of data elements organized in terms of rows and columns. A table is also considered as a convenient representation of relations. But a table can have duplicate row of data while a true relation cannot have duplicate data. Table is the simplest form of data storage.

**Null values:** SQL supports a special value known as NULL which is used to represent the values of attributes that may be unknown or not apply to a tuple.

For example: The Apartment number attribute of an address applies only to address that are in apartment buildings and not to other types of residences. It is important to understand that a NULL value is different from zero value.

A NULL value is used to represent a missing value, but that it usually has one of three different interpretations

- Value unknown
- Value not available
- Attribute not applicable

**Primary Key:** A primary key is used to ensure data in the specific column is unique. It is a column cannot have NULL values. It is either an existing table column or a column that is specifically generated by the database according to a defined sequence.

**Foreign Key:** A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. It is a column (or columns) that references a column (most often the primary key) of another table.

**Normalization:** Normalization is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion and updating anomalies. So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables. Types of Normal Forms:

### **1. First Normal Form:**

If a relation contains composite or multi-valued attribute, it violates first normal form or a relation is in first normal form if it does not contain any composite or multi-valued attribute. A relation is in first normal form if every attribute in that relation is singled valued attribute.

### **2. Second Normal Form:**

To be in second normal form, a relation must be in first normal form and relation must not contain any partial dependency. A relation is in 2NF if it has No Partial Dependency, i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.

**Partial Dependency:** If the proper subset of candidate key determines non-prime attribute, it is called partial dependency.

**3. Third Normal Form:** A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form. A relation is in 3NF if at least one of the following condition holds in every non-trivial function dependency  $X \rightarrow Y$

- X is a super key.
- Y is a prime attribute (each element of Y is part of some candidate key).

**Denormalization:** Denormalization is a database optimization technique in which we add redundant data to one or more tables. This can help us avoid costly joins in a relational database. Note that denormalization does not mean not doing normalization. It is an optimization technique that is applied after doing normalization.

## **Advantages and Disadvantages of a DBMS:**

### **Advantages:**

- 1. Reduction of Redundancy:** This is perhaps the most significant advantage of using DBMS. Redundancy is the problem of storing the same data item in more one place. Redundancy creates several problems like requiring extra storage space, entering same data more than once during data insertion, and deleting data from more than one place during deletion.
- 2. Sharing of Data:** In a paper-based record keeping, data cannot be shared among many users. But in computerized DBMS, many users can share the same database if they are connected via a network.
- 3. Data Integrity:** We can maintain data integrity by specifying integrity constrains, which are rules and restrictions about what kind of data may be entered or manipulated within the database. This increases the reliability of the database as it can be guaranteed that no wrong data can exist within the database at any point of time.
- 4. Data independence:** Application programs should be as independent as possible from details of data representation and storage. The DBMS can provide an abstract view of the data to Simulate application code from such details.
- 5. Efficient data access:** A DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently. This feature is especially important if the data is stored on external storage devices.
- 6. Data integrity and security:** If data is always accessed through the DBMS, the DBMS can enforce integrity constraints on the data. For example, before inserting salary information for an employee, the DBMS can check that the department budget is not exceeded. Also, the DBMS can enforce access controls that govern what data is visible to different classes of users.
- 7. Data administration:** When several users share the data, centralizing the administration of data can offer significant improvements. Experienced professionals who understand the nature of the data being managed, and how different groups of users use it, can be responsible for organizing the data representation to minimize redundancy and fine-tuning the storage of the data to make retrieval efficient.

**8. Reduced application development time:** Clearly, the DBMS supports many important functions that are common to many applications accessing data stored in the DBMS. This, in conjunction with the high-level interface to the data, facilitates quick development of applications. Such applications are also likely to be more robust than applications developed from scratch because many important tasks are handled by the DBMS instead of being implemented by the application.

### **Disadvantages:**

- 1. Danger of an Overkill:** For small and simple applications for single users a database system is often not advisable.
- 2. Complexity:** A database system creates additional complexity and requirements. The supply and operation of a database management system with several users and databases is quite costly and demanding.
- 3. Qualified Personnel:** The professional operation of a database system requires appropriately trained staff. Without a qualified database administrator nothing will work for long.
- 4. Costs:** Through the use of a database system new costs are generated for the system itself but also for additional hardware and the more complex handling of the system.
- 5. Lower Efficiency:** A database system is a multi-use software which is often less efficient than specialized software which is produced and optimized exactly for one problem.

### **Relational Database Management System (RDBMS):**

A relational database is a digital database based on the relational model of data, as proposed by E. F. Codd in 1970. A system used to maintain relational databases is a relational database management system. Many relational database systems have an option of using the SQL for querying and maintaining the database. A relational database management system (RDBMS) is a collection of programs and capabilities that enable IT teams and others to create, update, administer and otherwise interact with a relational database. RDBMS's store data in the form of tables, with most commercial relational database management systems using Structured Query Language (SQL) to access the database. However, since SQL was invented after the initial development of the relational



model, it is not necessary for RDBMS use. The RDBMS is the most popular database system among organizations across the world. It provides a dependable method of storing and retrieving large amounts of data while offering a combination of system performance and ease of implementation. All modern Database management systems like SQL, MYSQL Server, IBM DB2, ORACLE, My-SQL and Microsoft Access are based on RDBMS.

### **Structured Query Language (SQL):**

SQL is a Structured Query Language which is a computer language for storing, manipulating and retrieving data stored in relational database. SQL is the standard language for Relation Database System. All relational database management systems like MySQL, MS Access, Oracle, Sybase, Informix and SQL Server uses SQL as standard database language. Also, they are using different dialects, such as:

- MS SQL Server using T-SQL
- Oracle using PL/SQL
- MS Access version of SQL is called JET SQL (native format) etc.

### **Qualities of SQL:**

- Allow users to access data in relational database management systems.
- Allow users to describe the data.
- Allow users to define the data in database and manipulate that data.
- Allow to embed within other languages using SQL modules, libraries & pre-compilers.
- Allow users to create view, stored procedure, functions in a database.
- Allow users to set permissions on tables, procedures, and views SQL Process.

When you are executing an SQL command for any RDBMS, the system determines the best way to carry out your request and SQL engine figures out how to interpret the task.

**SQL Commands:** There are four types of SQL Commands.

1. Data Definition Language (DDL)
2. Data Manipulation Language (DML)
3. Data Query Language (DQL)
4. Data Control Language (DCL)

### **1. Data Definition Language (DDL):**

Data Definition Language can be defined as a standard for commands through which data structures are defined. It is a computer language that is used for creating and modifying structure of the database objects, such as schemas, tables, views, indexes etc. DDL commands work on the structure of a relation only.

It includes these commands:

- **CREATE:** Used to define a new table.
- **ALTER:** Used to change the types of records for a table.
- **DROP:** Used to remove a table.

### **2. Data Manipulation Language (DML):**

Data Manipulation Language (DML) can be defined as a set of syntax elements that are used to manage the data in the database. The commands of DML are not auto-committed and modification made by them are not permanent to the database. It is a computer programming language that is used to perform select, insert, delete and update data in a database. The user requests are assisted by Data Manipulation Language. This language is responsible for all forms of data modification in a database.

It includes these commands:

- **INSERT:** Used to add records to tables.
- **UPDATE:** Used to change a part of record.
- **DELETE:** Used to remove records from a table.

### **3. Data Query Language (DQL):**

DQL statements are used for performing queries on the data within schema objects. The purpose of the DQL Command is to get some schema relation based on the query passed to it.

- **SELECT:** used for retrieving data from table.

### **4. Data Control Language (DCL):**

DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions and other controls of the database system.

It includes these commands:

- **GRANT:** Used to give data privileges.
- **REVOKE:** Used to take away privileges.

### **Success of RDBMS and SQL:**

- Relational model is clean and rigorous.
- SQL is easy. It has simple and coherent language.
- Both SQL and RDBMS have separation from implementation details.
- RDBMS has connectivity to other programming languages.
- RDBMS can have explosion of apps and tools built on SQL.
- SQL can even use on non-relational systems.

**Transaction Processing Systems:** Transaction Processing Systems are the systems with large databases and hundreds of concurrent users executing database transactions.

For example: Airline reservations, banking, stock markets, etc.

- A transaction is an executing program that forms a logical unit of database processing.
- A transaction includes one or more database access operations- these can include insertion, deletion, modification, or retrieval operations.
- Transaction is executed as a single unit. It is a program unit whose execution may or may not change the contents of a database.

**Example:** A transfer of money from one bank account to another requires two changes to the database both must succeed or fail together.

- Subtracting the money from the savings account balance.
- Adding the money to the checking account balance.

### **Processes of Transaction:**

**Read Operation:** To read a database object, it is first brought into main memory from disk and then its value is copied into a program variable.

**Write Operation:** To write a database object, an in-memory copy of the object is first modified and then written back to disk.

### **Desirable Properties of Transactions:**

#### **ACID Properties:**

**Atomicity:** A transaction is an atomic unit of processing; it is either performed in its entirety or not performed at all.

**Consistency preservation:** A correct execution of the transaction must take the database from one consistent state to another.

**Isolation:** A transaction should appear as though it is being executed in isolation from other transactions, even if many transactions are executing concurrently. That is, the execution of a transaction should not be interfered with any other executing transactions.

**Durability or permanency:** Once a transaction changes the database and the changes are committed, these changes must never be lost because of any failure.

#### **Transaction States:**

- **Active state-** the initial state; the transaction stays in this state while it is executing.
- **Partially committed state-** after the final statement has been executed.

- **Failed state** - after the discovery that normal execution can no longer proceed.
- **Aborted state**- after the transaction has been rolled back and the database restored to its state prior to the start of the transaction. Two options after it has been aborted:
  - restart the transaction can be done only if no internal logical error
  - kill the transaction.
- **Committed state** - after successful completion

### **Big Data:**

Big Data is a collection of data that is huge in volume, yet growing exponentially with time. It is a data with so large size and complexity that none of traditional data management tools can store it or process it efficiently. Big data is also a data but with huge size. big data is data that contains greater variety, arriving in increasing volumes and with more velocity. This is also known as the three Vs. Put simply, big data is larger, more complex data sets, especially from new data sources.

Examples of Big Data:

- Live road mapping for autonomous vehicles
- Streamlined media streaming
- Predictive inventory ordering
- Personalized health plans for cancer patients
- Real-time data monitoring and cybersecurity protocols

### **Importance of Big Data:**

Big data analytics helps organizations harness their data and use it to identify new opportunities. That, in turn, leads to smarter business moves, more efficient operations, higher profits and happier customers. In his report Big Data in Big Companies, IIA Director of Research Tom Davenport interviewed more than 50 businesses to understand how they used big data. He found they got value in the following ways: Cost reduction, Faster, better decision making, new products and services.

## **Types of Big Data**

Following are the types of Big Data:

- Structured
- Unstructured
- Semi-structured

**Structured:** Any data that can be stored, accessed and processed in the form of fixed format is termed as ‘structured’ data. Over the period of time, talent in computer science has achieved greater success in developing techniques for working with such kind of data (where the format is well known in advance) and also deriving value out of it. However, nowadays, we are foreseeing issues when a size of such data grows to a huge extent, typical sizes are being in the range of multiple zettabytes.

Data stored in a relational database management system is one example of a ‘**structured**’ data.

**“ $10^{21}$  bytes equal to 1 zettabyte or one billion terabytes forms a zettabyte.”**

## **Benefits of Structured Data:**

- 1. Easily used by machine learning algorithms:** The largest benefit of structured data is how easily it can be used by machine learning. The specific and organized nature of structured data allows for easy manipulation and querying of that data.
- 2. Easily used by business users:** Another benefit of structured data is that it can be used by an average business user with an understanding of the topic to which the data relates. There is no need to have an in-depth understanding of various different types of data or the relationships of that data. It opens up self-service data access to the business user.
- 3. Increased access to more tools:** Structured data also has the benefit of having for far longer, as historically it was the only option. This means that there are more tools that have been tried and tested in using and analyzing structured data. Data managers have more product choices when using structured data.

**Unstructured:** Any data with unknown form or the structure is classified as unstructured data. In addition to the size being huge, un-structured data poses multiple challenges in terms of its processing for deriving value out of it. A typical example of unstructured data is a heterogeneous data source containing a combination of simple text files, images, videos etc. Now day organizations have wealth of data available with them but unfortunately, they don't know how to derive value out of it since this data is in its raw form or unstructured format. Example: The Output returned by google search.

**Semi-structured:** Semi-structured data can contain both the forms of data. We can see semi-structured data as a structured in form but it is actually not defined with e.g., a table definition in relational DBMS. It is also known as self-describing structure. Third category exists (between structured and unstructured data) is because semi-structured data is considerably easier to analyse than unstructured data. Many Big Data solutions and tools have the ability to 'read' and process either JSON or XML. This reduces the complexity to analyse structured data, compared to unstructured data. Example of semi-structured data is a data represented in an XML file.

**Characteristics Of Big Data:** Big data can be described by the following characteristics:

- Volume
- Variety
- Velocity
- Variability

**1. Volume** – The name Big Data itself is related to a size which is enormous. Size of data plays a very crucial role in determining value out of data. Also, whether a particular data can actually be considered as a Big Data or not, is dependent upon the volume of data. Hence, '**Volume**' is one characteristic which needs to be considered while dealing with Big Data solutions.

**2. Variety** – The next aspect of Big Data is its **variety**. Variety refers to heterogeneous sources and the nature of data, both structured and unstructured. During earlier days, spreadsheets and databases were the only sources of data considered by most of the applications. Nowadays, data in the form of emails, photos, videos, monitoring devices, PDFs, audio, etc. are also being considered

in the analysis applications. This variety of unstructured data poses certain issues for storage, mining and analyzing data.

**3. Velocity** – The term ‘velocity’ refers to the speed of generation of data. How fast the data is generated and processed to meet the demands, determines real potential in the data. Big Data Velocity deals with the speed at which data flows in from sources like business processes, application logs, networks, and social media sites, sensors, Mobile devices, etc. The flow of data is massive and continuous.

**4. Variability** – This refers to the inconsistency which can be shown by the data at times, thus hampering the process of being able to handle and manage the data effectively. Big data is also variable because of the multitude of data dimensions resulting from multiple disparate data types and sources. Variability can also refer to the inconsistent speed at which big data is loaded into your database.

### **Advantages Of Big Data Processing:**

Ability to process Big Data in DBMS brings in multiple benefits, such as-

- Businesses can utilize outside intelligence while taking decisions: Access to social data from search engines and sites like Facebook, twitter are enabling organizations to fine tune their business strategies.
- Improved customer service: Traditional customer feedback systems are getting replaced by new systems designed with Big Data technologies. In these new systems, Big Data and natural language processing technologies are being used to read and evaluate consumer responses.
- Early identification of risk to the product/services
- Better operational efficiency

Big Data technologies can be used for creating a staging area or landing zone for new data before identifying what data should be moved to the **data warehouse**. In addition, such integration of Big Data technologies and data warehouse helps an organization to offload infrequently accessed data.



## **Data V/s Traditional System (RDBMSs):**

Actually, big data is different from data that was used traditionally, as the objectives, plans, processes, and tools are very different. Let's now turn to the characteristics that differentiate big data from traditional data.

**1. Flexibility:** A traditional database is based on a fixed schema that is static in nature. It could only work with structured data that fit effortlessly into relational databases or tables. In reality, most data are unstructured. The extensive variety of unstructured data requires new methods to store and process. Some examples include movies and sound files, images, documents, geolocation data, text, weblogs, strings, and web content. Big data uses a dynamic schema that can include structured as well as unstructured data. The data is stored in a raw form and the schema is applied only when accessing it. For big data analytics, datasets from diverse sources are appended, then functions such as storing, cleansing, distributing, indexing, transforming, searching, accessing, Analyzing, and visualizing are performed.

**2. Real-time analytics:** Traditionally, analytics always took place after the event or time period that was being analyzed. With big data, analytics takes place in real-time as the data is being gathered and findings are presented practically instantaneously. This capability enables breakthroughs in medical, safety, smart cities, manufacturing and transportation domains.

**3. Distributed architecture:** While traditional data is based on a centralized database architecture, big data uses a distributed architecture. Computation is distributed among several computers in a network. This makes big data far more scalable than traditional data, in addition to delivering better performance and cost benefits. The use of commodity hardware, open-source software, and cloud storage makes big data storage even more economical. Once data quality checks and the data is modelled so that it can be stored in a data warehouse.

**4. Multitude of sources:** Traditionally, the sources of data were fairly limited. Today there is a data explosion thanks to a multitude of sources that capture data practically every moment. Readings from medical equipment, air particle counters, crowd density calculators, and embedded devices in vehicles are only a few examples that show the huge volume, as well as variety, of big data from different source.

**5.Enables exploratory analysis:** In the traditional approach to data analytics, users had to determine their questions at the start. Data was structured in order to find the answers to their questions and then reports would be generated. Big data, however, enables a more iterative and exploratory approach. The focus is to develop a platform for creative discovery so that users can explore what questions can be asked. In a business scenario, the traditional approach led to the creation of monthly reports, productivity analysis, customer survey findings, etc. Big data provides insights into sentiment analysis, product strategy, asset utilization, preventive maintenance of equipment, etc.

**Important Parameters:**

1 KILOBYTE(KB) = 1000 BYTES

1 MEGA BYTE(MB) = 1000 KB

1 GIGA BYTE(GB) = 1000 MB

1 TERA BYTE(TB) = 1000 GB

1 PETA BYTE(PB) = 1000 TB

1 EXABYTE (EB) = 1000 PB

**“Minimum volume of big data is around 30 terabytes”.**

**Distributed storage:** In modern technology there's no choice but to store your data across multiple disk drives, and the largest data stores must necessarily span thousands of disk drives. So, a big data relies on "**DISTRIBUTED STORAGE**". For distributed storage, instead of storing a large file sequentially, you can split into pieces and scatter those pieces across many disks. This illustration shows a file split into pieces, sometimes called blocks, with those blocks distributed across multiple disks for storage of a file. The big data platform Apache Hadoop includes a file system called a Hadoop Distributed File System or HDFS. In HDFS, a single block is usually of size 128megabytes. So, a one-gigabyte file would consist of 8 blocks, and a one-terabyte file would consist of 8000 blocks.

**Distributed Processing:** Processing is typically distributed across multiple computers, or it will take too long to be practical. Complex tasks may require multiple processing stages, including shuffling data among computers.

## Second Course: Analyzing Big Data with SQL

**SQL:** Structured Query Language (SQL) has been around for decades. It is a programming language used for managing the data held in relational databases. SQL is used all around the world by a majority of big companies. A data analyst can use SQL to access, read, manipulate, and analyze the data stored in a database and generate useful insights to drive an informed decision-making process. SQL is often used to connect code to a variety of data sources.

**Spark SQL:** Spark SQL is a **Spark** module for structured data processing. It provides a programming abstraction called Data Frames and can also act as a distributed SQL query engine. It enables unmodified Hadoop Hive queries to run up to 100x faster on existing deployments and data. It also provides powerful integration with the rest of the Spark ecosystem (e.g., integrating SQL query processing with machine learning). Spark introduces a programming module for structured data processing called Spark SQL. It provides a programming abstraction called Data Frame and can act as distributed SQL query engine.

**Features of Spark SQL:** The following are the features of Spark SQL –

- **Integrated** – Seamlessly mix SQL queries with Spark programs. Spark SQL lets you query structured data as a distributed dataset (RDD) in Spark, with integrated APIs in Python, Scala and Java. This tight integration makes it easy to run SQL queries alongside complex analytic algorithms.
- **Unified Data Access** – Load and query data from a variety of sources. Schema-RDDs provide a single interface for efficiently working with structured data, including Apache Hive tables, parquet files and JSON files.
- **Hive Compatibility** – Run unmodified Hive queries on existing warehouses. Spark SQL reuses the Hive frontend and Meta Store, giving you full compatibility with existing Hive data, queries, and UDFs. Simply install it alongside Hive.

- **Standard Connectivity** – Connect through JDBC or ODBC. Spark SQL includes a server mode with industry standard JDBC and ODBC connectivity.
- **Scalability** – Use the same engine for both interactive and long queries. Spark SQL takes advantage of the RDD model to support mid-query fault tolerance, letting it scale to large jobs too. Do not worry about using a different engine for historical data.

### **Apache Hive:**

Hive QL, the SQL dialect of Apache Hive is not really SQL, but is MapReduce for people who know SQL. MapReduce programs read and process data using multiple distributed tasks that run in parallel across mini computers. Hive automatically translates any SQL SELECT into a suitable Map Reduce program. Using Hive, you cannot process all possible data that has suitable structure. Later, since 2015, Hive had the option to produce Apache spark programs instead of Map Reduce. Spark programs use more memory and reduce the use of disk drive for temporary storage and so can improve response times for long running programs compared to the equivalent Map Reduce programs. Hive is very much useful for large data to store in cluster. Hive is a good choice for processing large amounts of data as a part of an ETL pipeline. Hive translates SQL statements into other programs for actual execution. Hive distributed programs are fault tolerant. When you need to produce a new large data setting or cluster. Hive is a good choice for its reliability and fault tolerance.

**Apache Impala:** Apache Impala is built from the ground up as a distributed SQL engine for big data. Impala runs as a collection of Impala daemons running in a cluster. A daemon is a continuously running server program that awaits and server requests as they appear.

**Web Server:** The impala query can run 10 or even 50 times faster than the same query in Hive. Master node: 1. Catalog service 2. State store.

They run in support of impala daemons. Impala is the better high-speed choice. Impala is also good for business intelligence programmers or dash boards that query your cluster.

## **Running SQL statements using the Hue Query editors:**

Hue is a web browser-based analytics works bench that provides a user interface to Hive and Impala. Hue includes a number of different interfaces: There are just a few interfaces that you will use. To run SQL statements to query the tables, Hue has a different interface. Hue has SQL editors for both Hive and Impala. The Impala editor is the default one. Any organized collection of data can be called a database. SQL engines in general are often called databases and one specific instance of a SQL engine is often called a database.

## **Running SQL utility statements:**

- 1. SHOW DATABASES:** It tells you what databases exist
- 2. USE database name:** To set which database is current database. When you're using a SQL engine, there is always one particular database you're connected to. This is called current database or the active database. Hue doesn't support USE statement. In Hue, you always use point and click actions to set the current database.
- 3. SHOW TABLES:** To see what tables exists in current database.
- 4. DESCRIBE table name:** To see what columns are there in a table.

## **Running SQL select statements:**

**Query:** A SELECT statement in the SQL query.

The SELECT statement is the most important part of the SQL language. The order of the columns in a result set is deterministic but the order of rows is not. When you run a SELECT statement using a distributed SQL engine, the order of the rows in the results set is arbitrary and predictable.

## **SQL interfaces:**

There are two different interface standards that virtually any software can use to connect to virtually any SQL engine.

- ODBC
- JDBC

Both Hive and Impala support both ODBC and JDBC.

## **Where Clause:**

The WHERE clause filters the rows of data based on one or more conditions. In other words, the WHERE clause takes all the data in the table, tests which rows meet some criteria, and returns only those rows. The WHERE clause is optional. Using expressions in the WHERE clause:

### **Expressions in SELECT list:**

- Becomes column in result
- Multiple expressions
- Different data types allowed

### **Expressions in WHERE list:**

- Filters rows in result
- Only one expression
- Expression must be Boolean

## **Big SQL features:**

Big SQL features include easy-to-use tools, flexible security options, strong federation and performance capabilities, and a massively parallel processing (MPP) SQL engine that provides powerful SQL processing features.

- **Administration tools:**

With Big SQL, you can manage your system with the Ambari dashboard, and you can manage your databases with Data Server Manager (DSM).

- **Visualization tools:**

Big SQL includes a powerful SQL processing engine, works in a platform with a data warehouse based on Apache Hive, accesses the DSM console using Apache Knox, and can perform authorization and auditing using the Apache Ranger framework.

- **Federation capabilities:**

With Big SQL you can efficiently query data on Hadoop and also combine data spread around different enterprise data warehouses. The **federation capability** of Big SQL lets you

query against and combine with Hadoop data, as well as letting you push down predicates. Not all data moves back and forth between the systems; only the results of the predicates are sent back to combine with Hadoop data.

- **Performance capabilities:**

Big SQL provides superior SQL-on-Hadoop performance to optimize data ingestion and query performance for your enterprise. Big SQL is not only fast and efficient, but more importantly can successfully execute the most demanding queries on big data. Big SQL provides a robust runtime environment and is compliant with SQL standards.

- **SQL processing features:**

The Big SQL massively parallel processing (MPP) SQL engine provides you with several SQL processing features.

### **Union and Union All in SQL:**

**Union:** Union means joining two or more data sets into a single set. In SQL Server, Union is used to combine two queries into a single result set using the select statements. Union extracts all the rows that are described in the query.

**Syntax –**

query1 UNION query2

Union holds a few conditions before being used in a query. One such condition is that the rows to be extracted must come from the same columns from the tables.

**Union All:** A union is used for extracting rows using the conditions specified in the query while Union All is used for extracting all the rows from a set of two tables.

**Syntax –**

query1 UNION ALL query2

The same conditions are applicable to Union All. The only difference between Union and Union All is that Union extracts the rows that are being specified in the query while Union All extracts all the rows including the duplicates (repeated values) from both the queries.

## **Joins in SQL:**

**Join:** A SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are:

- Inner Join
- Left Join
- Right Join
- Full Join

### **Inner Join:**

The INNER JOIN keyword selects all rows from both the tables as long as the condition satisfies. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e., value of the common field will be same.

#### **Syntax:**

```
SELECT table1.column1,table1.column2,table2.column1,...
```

```
FROM table1
```

```
INNER JOIN table2
```

```
ON table1.matching_column = table2.matching_column;
```

**Note:** We can also write JOIN instead of INNER JOIN. JOIN is same as INNER JOIN.

### **Left Join:**

This join returns all the rows of the table on the left side of the join and matching rows for the table on the right side of join. The rows for which there is no matching row on right side, the result-set will contain null. LEFT JOIN is also known as LEFT OUTER JOIN.

#### **Syntax:**

```
SELECT table1.column1,table1.column2,table2.column1,...
```

```
FROM table1
```

```
LEFT JOIN table2
```



ON table1.matching\_column = table2.matching\_column;

**Note:** We can also use LEFT OUTER JOIN instead of LEFT JOIN, both are same.

### **RIGHT JOIN:**

RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of join. The rows for which there is no matching row on left side, the result-set will contain null. RIGHT JOIN is also known as RIGHT OUTER JOIN.

#### **Syntax:**

SELECT table1.column1, table1.column2, table2.column1, ...

FROM table1

RIGHT JOIN table2

ON table1.matching\_column = table2.matching\_column;

**Note:** We can also use RIGHT OUTER JOIN instead of RIGHT JOIN, both are same.

### **FULL JOIN:**

FULL JOIN creates the result-set by combining result of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both the tables. The rows for which there is no matching, the result-set will contain NULL values.

#### **Syntax:**

SELECT table1.column1,table1.column2,table2.column1....

FROM table1

FULL JOIN table2

ON table1.matching\_column = table2.matching\_column;



### **Third Course:**

### **Managing Big Data in Clusters and Cloud Storage**

#### **Clustering:**

Clustering is a type of unsupervised learning method of machine learning. In the unsupervised learning method, the inferences are drawn from the data sets which do not contain labelled output variable. It is an exploratory data analysis technique that allows us to analyze the multivariate data sets. Clustering is a task of dividing the data sets into a certain number of clusters in such a manner that the data points belonging to a cluster have similar characteristics. Clusters are nothing but the grouping of data points such that the distance between the data points within the clusters is minimal. In other words, the clusters are regions where the density of similar data points is high. It is generally used for the analysis of the data set, to find insightful data among huge data sets and draw inferences from it. Generally, the clusters are seen in a spherical shape, but it is not necessary as the clusters can be of any shape. Learn about clustering and more data science concepts in our data science online course. It depends on the type of algorithm we use which decides how the clusters will be created. The inferences that need to be drawn from the data sets also depend upon the user as there is no criterion for good clustering

#### **Types of Clustering Methods:**

Clustering itself can be categorized into two types viz. Hard Clustering and Soft Clustering. In hard clustering, one data point can belong to one cluster only. But in soft clustering, the output provided is a probability likelihood of a data point belonging to each of the pre-defined numbers of clusters.

#### **Density-Based Clustering:**

In this method, the clusters are created based upon the density of the data points which are represented in the data space. The regions that become dense due to the huge number of data points residing in that region are considered as clusters. The data points in the sparse region (the region where the data points are very less) are considered as noise or outliers. The clusters created in these methods can be of arbitrary shape. Following are the examples of Density-based clustering algorithms:

### **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):**

DBSCAN groups data points together based on the distance metric and criterion for a minimum number of data points. It takes two parameters – eps and minimum points. Eps indicates how close the data points should be to be considered as neighbors. The criterion for minimum points should be completed to consider that region as a dense region.

### **OPTICS (Ordering Points to Identify Clustering Structure):**

It is similar in process to DBSCAN, but it attends to one of the drawbacks of the former algorithm i.e., inability to form clusters from data of arbitrary density. It considers two more parameters which are core distance and reachability distance. Core distance indicates whether the data point being considered is core or not by setting a minimum value for it. Reachability distance is the maximum of core distance and the value of distance metric that is used for calculating the distance among two data points. One thing to consider about reachability distance is that its value remains not defined if one of the data points is a core point.

### **HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise):**

HDBSCAN is a density-based clustering method that extends the DBSCAN methodology by converting it to a hierarchical clustering algorithm.

### **Hierarchical Clustering:**

Hierarchical Clustering groups (Agglomerative or also called as Bottom-Up Approach) or divides (Divisive or also called as Top-Down Approach) the clusters based on the distance metrics. In Agglomerative clustering, each data point acts as a cluster initially, and then it groups the clusters one by one. Divisive is the opposite of Agglomerative, it starts off with all the points into one cluster and divides them to create more clusters. These algorithms create a distance matrix of all the existing clusters and perform the linkage between the clusters depending on the criteria of the linkage. The clustering of the data points is represented by using a dendrogram.

**There are different types of linkages: –**

- **Single Linkage:** In single linkage the distance between the two clusters is the shortest distance between points in those two clusters.
- **Complete Linkage:** In complete linkage, the distance between the two clusters is the farthest distance between points in those two clusters.
- **Average Linkage:** In average linkage the distance between the two clusters is the average distance of every point in the cluster with every point in another cluster.

### **Fuzzy Clustering:**

In fuzzy clustering, the assignment of the data points in any of the clusters is not decisive. Here, one data point can belong to more than one cluster. It provides the outcome as the probability of the data point belonging to each of the clusters. One of the algorithms used in fuzzy clustering is Fuzzy c-means clustering. This algorithm is similar in process to the K-Means clustering and it differs in the parameters that are involved in the computation like fuzzifier and membership values.

### **Partitioning Clustering:**

This method is one of the most popular choices for analysts to create clusters. In partitioning clustering, the clusters are partitioned based upon the characteristics of the data points. We need to specify the number of clusters to be created for this clustering method. These clustering algorithms follow an iterative process to reassign the data points between clusters based upon the distance. The algorithms that fall into this category are as follows: –

### **K-Means Clustering:**

K-Means clustering is one of the most widely used algorithms. It partitions the data points into k clusters based upon the distance metric used for the clustering. The value of 'k' is to be defined by the user. The distance is calculated between the data points and the centroids of the clusters. The data point which is closest to the centroid of the cluster gets assigned to that cluster. After an iteration, it computes the centroids of those clusters again and the process continues until a pre-defined number of iterations are completed or when the centroids of the clusters do not change after an iteration. It is a very computationally expensive algorithm as it computes the distance of every

data point with the centroids of all the clusters at each iteration. This makes it difficult for implementing the same for huge data sets.

### **PAM (Partitioning Around Medoids):**

This algorithm is also called as k-medoid algorithm. It is also similar in process to the K-means clustering algorithm with the difference being in the assignment of the center of the cluster. In PAM, the medoid of the cluster has to be an input data point while this is not true for K-means clustering as the average of all the data points in a cluster may not belong to an input data point.

### **CLARA (Clustering Large Applications):**

CLARA is an extension to the PAM algorithm where the computation time has been reduced to make it perform better for large data sets. To accomplish this, it selects a certain portion of data arbitrarily among the whole data set as a representative of the actual data. It applies the PAM algorithm to multiple samples of the data and chooses the best clusters from a number of iterations.

### **Grid-Based Clustering:**

In grid-based clustering, the data set is represented into a grid structure which comprises of grids (also called cells). The overall approach in the algorithms of this method differs from the rest of the algorithms. They are more concerned with the value space surrounding the data points rather than the data points themselves. One of the greatest advantages of these algorithms is its reduction in computational complexity. This makes it appropriate for dealing with humongous data sets.

After partitioning the data sets into cells, it computes the density of the cells which helps in identifying the clusters. A few algorithms based on grid-based clustering are as follows: –

### **STING (Statistical Information Grid Approach):**

In STING, the data set is divided recursively in a hierarchical manner. Each cell is further sub-divided into a different number of cells. It captures the statistical measures of the cells which helps in answering the queries in a small amount of time.

### **Wave Cluster:**

In this algorithm, the data space is represented in form of wavelets. The data space composes an n-dimensional signal which helps in identifying the clusters. The parts of the signal with a lower frequency and high amplitude indicate that the data points are concentrated. These regions are identified as clusters by the algorithm. The parts of the signal where the frequency high represents the boundaries of the clusters. For more details, you can refer to this paper.

### **CLIQUE (Clustering in Quest):**

CLIQUE is a combination of density-based and grid-based clustering algorithm. It partitions the data space and identifies the sub-spaces using the Apriori principle. It identifies the clusters by calculating the densities of the cells.

### **Hadloop Cluster:**

Apache Hadoop is an open source, Java-based, software framework and parallel data processing engine. It enables big data analytics processing tasks to be broken down into smaller tasks that can be performed in parallel by using an algorithm (like the MapReduce algorithm), and distributing them across a Hadoop cluster. A Hadoop cluster is a collection of computers, known as nodes, that are networked together to perform these kinds of parallel computations on big data sets. Unlike other computer clusters, Hadoop clusters are designed specifically to store and analyze mass amounts of structured and unstructured data in a distributed computing environment. Further distinguishing **Hadoop ecosystems** from other computer clusters are their unique structure and architecture. Hadoop clusters consist of a network of connected master and slave nodes that utilize high availability, low-cost commodity hardware. The ability to linearly scale and quickly add or subtract nodes as volume demands makes them well-suited to **big data analytics** jobs with data sets highly variable in size.

### **Data Mining:**

Data mining is the process of Analyzing large volumes of data so as to discover business intelligence which helps companies to solve problems, seize new opportunities, and mitigate risks. Some data mining **tools** used in the industry are Rapid Miner, oracle data mining, IBM SPSS

Modeler, KNIME, Python Orange, Kaggle, Rattle, Weka, and Teradata. Data mining examples: include Groupon- Data mining allows Groupon alignment of marketing activities closely to customer preferences which analyse just 1 terabyte of real-time customer data that helps to identify the emerging trends. Data mining technique is used by Air France. Strip searches, bookings, social media, flight operations, call centres, and interactions in the airport lounge are analysed and a 360-degree customer view is created. Grocery stores use data mining by giving loyalty cards to customers that make it easy for the cardholders to avail of special prices that are not made available to non-cardholders. The above are a few examples of data mining helping companies to increase efficiency, streamline operations, cost reduction, and improve profits.

**Uses of Data Mining:** Some of the most common Data mining concepts are:

- **Data cleansing and preparation-** in this step transformation of data into a suitable form required for further processing and analysis such as identification and error removal and missing data.
- **Artificial intelligence (AI)-** analytical activities that are associated with human intelligence like reasoning, planning, learning, and problem-solving are performed by these systems.
- **Association rule learning-** also known as market basket analysis, these tools look in the dataset, for the relationship between variables such as concluding which products are purchased by the customers together.
- **Clustering-** is a process in which the dataset is partitioned into sets of relevant divisions called clusters, that would help the users to understand the structure or natural groups in the data.
- **Classification-** with the goal of predicting the target class for each and every case in the data, items are assigned by this technique in the dataset.
- **Data analytics-** data analytics is the process of evaluating digital information and converting it into information useful for business.

- **Data warehousing**- is the component of the foundational importance of most huge-scale data mining efforts with a large collection of data, that is used for decision making in organizations.
- **Machine learning**- is a computer programmed technique, that makes use of statistical probabilities that gives the computer the capacity to 'learn' even without being clearly programmed.
- **Regression**- is a technique that is made use of to predict a variety of numeric values, including sales, price of a stock, temperatures, that are based on a precise dataset.

### **Cloud Storage:**

Cloud storage is a way for businesses and consumers to save data securely online so that it can be accessed anytime from any location and easily shared with those who are granted permission. Cloud storage also offers a way to back up data to facilitate recovery off-site. Today, individuals have access to several free cloud computing services such as Google drive, Dropbox, and Box, which all come with upgraded subscription packages that offer larger storage sizes and additional cloud services. Cloud storage allows individuals and businesses to store and retrieve computer files via an internet-connected device.

Cloud storage has grown increasingly popular among individuals who need larger storage space and for businesses seeking an efficient off-site data back-up solution. Because of cloud storage's increasing popularity and use, cloud security has become a major concern to protect data integrity, prevent hacking attempts, and avoid file or identity theft.

Cloud storage works by allowing a client computer, tablet, or smartphone to send and retrieve files online to and from a remote data server. The same data is usually stored on more than one server simultaneously so that clients can always access their data even if one server is down or loses data.

For example, a laptop computer owner might store personal photos both on her hard drive and in the cloud in case the laptop is stolen. Cloud storage is believed to have been invented by computer scientist Dr. Joseph Carl Ronette Lickliter in the 1960s. About two decades later, CompuServe began to offer its customers small amounts of disk space in order to store some of their files. In the mid-1990s, AT&T launched the first all web-based storage service for personal and business



communication. Since then, a number of different services have become gained traction. Some of the most popular cloud storage providers are Apple (iCloud), Amazon (Amazon Web Services ), Dropbox, and Google.

### **Hadoop Cluster Architecture:**

Hadoop clusters are composed of a network of master and worker nodes that orchestrate and execute the various jobs across the Hadoop distributed file system. The master nodes typically utilize higher quality hardware and include a Name Node, Secondary Name Node, and Job Tracker, with each running on a separate machine. The workers consist of virtual machines, running both Data Node and Task Tracker services on commodity hardware, and do the actual work of storing and processing the jobs as directed by the master nodes. The final part of the system is the Client Nodes, which are responsible for loading the data and fetching the results.

### **Advantages of a Hadoop Cluster:**

- Hadoop clusters can boost the processing speed of many big data analytics jobs, given their ability to break down large computational tasks into smaller tasks that can be run in a parallel, distributed fashion.
- Hadoop clusters are easily scalable and can quickly add nodes to increase throughput, and maintain processing speed, when faced with increasing data blocks.
- The use of low cost, high availability commodity hardware makes Hadoop clusters relatively easy and inexpensive to set up and maintain.
- Hadoop clusters replicate a data set across the distributed file system, making them resilient to data loss and cluster failure.
- Hadoop clusters make it possible to integrate and leverage data from multiple different source systems and data formats.
- It is possible to deploy Hadoop using a single-node installation, for evaluation purposes.

## **Challenges of a Hadoop Cluster:**

- **Issue with small files** - Hadoop struggles with large volumes of small files - smaller than the Hadoop block size of 128MB or 256MB by default. It wasn't designed to support big data in a scalable way. Instead, Hadoop works well when there are a small number of large files. Ultimately when you increase the volume of small files, it overloads the Namenode as it stores namespace for the system.
- **High processing overhead** - reading and writing operations in Hadoop can get very expensive quickly especially when processing large amounts of data. This all comes down to Hadoop's inability to do in-memory processing and instead data is read and written from and to the disk.
- **Only batch processing is supported** - Hadoop is built for small volumes of large files in batches. This goes back to the way data is collected and stored which all has to be done before processing starts. What this ultimately means is that streaming data is not supported and it cannot do real-time processing with low latency.
- **Iterative Processing** - Hadoop has a data flow structure is set-up in sequential stages which makes it impossible to do iterative processing or use for ML.

## **Learning Outcomes**

A lot of beginners and experienced programmers avoid learning Big Data and SQL because it's complicated and they think that there is no use of all the above stuff in real life but there is a lot of implementations of Big Data in daily life.

The topics that I have learned:

- I have learned Modern Big Data and SQL from Basic Level to Advance Level.
- I learned about Data, Database, Database Management System, Tables, Schemas, Normalization etc.
- I have learned SQL, SQL Commands, Operations, SQL Statements etc.
- I learned the concept of Relational Database.
- I learned to manage Big Data in Clusters and Cloud Storage.
- I learned to use SQL Server, Managing Datasets in Clusters.
- I also learn that using SQL to make Projects for a Big Data Analysis.
- This course helps me in identify the most interesting or relevant aspects of Big Data that help me in engineering to be pursued in my future studies.



## Conclusion

Data is an information that can be transmitted, stored, and processed using modern digital technologies like internet. It is of two types analogy data and digital data. For the smooth processing of the data, it needs to be stored efficiently. For that we are having DBMS, which provides us with certain ways of storing the data efficiently. For storing the data into the database and then extracting it efficiently we are having SQL. A database is a table that consists of rows and columns. SQL is the language of databases. It facilitates retrieving specific information from databases that are further used for analysis. It enables a user to create, read, update, and delete relational databases and tables. The Comes Hadoop User Experience (HUE) which is an open-source interface which makes Apache Hadoop's use easier. It is a web-based application. It has a job designer for MapReduce, a file browser for HDFS, an Oozie application for making workflows and coordinators, an Impala, a shell, a Hive UI, and a group of Hadoop APIs. Conclusion Points:

- From this course, I have good hands-on SQL and Big Data.
- The course gives me ample knowledge of Modern Big Data for placement related things, for GATE preparation, for future study and many things.
- As the course was very well designed and provides sufficient knowledge of each and every topic. From my perspective the overall course was good and I completely learnt whatever I read.
- The course will help me to prepare for provide excellent preparation for the **Cloudera Certified Associate (CCA) Data Analyst** certification exam.
- The course knowledge about SQL and Big Data helps me to become Data Analyst.
- The course will also help me in future to get the job.



**L** OVELY  
**P** ROFESSIONAL  
**U** NIVERSITY

## Reference

- <https://www.coursera.org/specializations/cloudera-big-data-analysis-sql>
- <https://www.coursera.org/learn/foundations-big-data-analysis-sql>
- <https://www.coursera.org/learn/cloudera-big-data-analysis-sql-queries>
- <https://www.coursera.org/learn/cloud-storage-big-data-analysis-sql>