Syllabus

Course : USCS606	TOPICS (Credits : 03 Lectures/Week : 03) **Data Science**

Objectives :

Understanding basic data science concepts. Learning to detect and diagnose common data issues, such as missing values, special values, outliers, inconsistencies, and localization. Making aware of how to address advanced statistical situations, Modeling and Machine Learning.

Expected Learning Outcomes :

After completion of this course, the students should be able to understand & comprehend the problem; and should be able to define suitable statistical method to be adopted.

Unit I	**Introduction to Data Science :** What is Data? Different kinds of data, Introduction to high level programming language + Integrated Development Environment (IDE), Exploratory Data Analysis (EDA) + Data Visualization, Different types of data sources, **Data Management :** Data Collection, Data cleaning/extraction, Data analysis & Modeling **(Chapters 1 and 2)**	15L
Unit II	**Data Curation :** Query languages and Operations to specify and transform data, Structured/schema based systems as users and acquirers of data Semi-structured systems as users and acquirers of data, Unstructured systems in the acquisition and structuring of data, Security and ethical considerations in relation to authenticating and authorizing access to data on remote systems, Software development tools, Large scale data systems, Amazon Web Services (AWS) **(Chapter 3)**	15L
Unit III	**Statistical Modelling and Machine Learning :** Introduction to model selection: Regularization, bias/variance tradeoff e.g. parsimony, AIC, BIC, Cross validation, Ridge regressions and penalized regression e.g. LASSO **Data transformations :** Dimension reduction, Feature extraction, Smoothing and aggregating **Supervised Learning :** Regression, linear models, Regression trees, Time-series Analysis, Forecasting, Classification: classification trees, Logistic regression, separating hyperplanes, k-NN **Unsupervised Learning :** Principal Components Analysis (PCA), k-means clustering, Hierarchical clustering, Ensemble methods. **(Chapters 4, 5, 6 and 7)**	15L

❑❑❑

Unit II

CHAPTER 3

Data Curation

3.1 Introduction

Q. 3.1.1 What is Data Curation? (Ref. Sec. 3.1) (5 Marks)

- Data curation defined as the process of collecting data from various sources and integrating it into various repositories that are many more times more valuable than the independent parts (by "techrepublic").
- Data curation is an active and ongoing management of the data through its life cycle of interest and usefulness. It is mainly concern with managing and maintaining the metadata. It is an iterative process which includes three main stages :

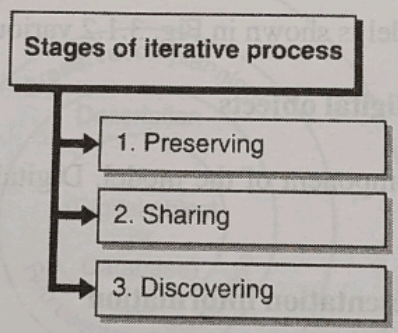

Fig. 3.1.1 : Stages of iterative process

➜ **1. Preserving**

Preserving means collecting of data and then store and mange it.

➜ **2. Sharing**

Sharing deals with making data available according to need of authenticated users.

3. Discovering

Discovering deals with reusing of data with different combination and generating some new data.

- Few tools which are most commonly used for data curation are :
  1. Bitbucket
  2. Colectica
  3. CSV fingerprints
  4. Dash etc.
- Data curation is a term which is used to indicate various processes and activities related to the organization an integration of data collected from sources. It includes various processes needed for controlling data creation, maintenance and management. Data curation process is a user initiated and it is active and on-going process.

3.1.1 Data Curation Lifecycle

Q. 3.1.2 Explain Data Curation Lifecycle in detail. (Ref. Sec. 3.1.1) **(5 Marks)**

Data curation Lifecycle model is shown in Fig. 3.1.2 various components are as follows :

1. Data or database or digital objects

They are the key or core component of the model. Digital objects can be any type of file or complex digital object.

2. Description and representation information

This is the second layer and in this phase assignment of administrative, descriptive, technical, structural and preservation of the metadata is done depending on the standards defined.

3. Preservation planning

Here the planning for the preservation of digital material for throughout cycle is done.

4. Community watch and participation

Here, the tracking of various community activities is done using various standards and tools.

5. Curate and preserve

In this phase the action plans are promoted to curation and are preserved throughout the lifecycle.

6. Conceptualise

Here, how to take data or create data is conceptualize, various methods for generating, storing and capturing of data are being thought of.

7. Create or receive

Creation of various types of administrative, descriptive, structural and technical metadata is done and also the receiving of data in various formats is done.

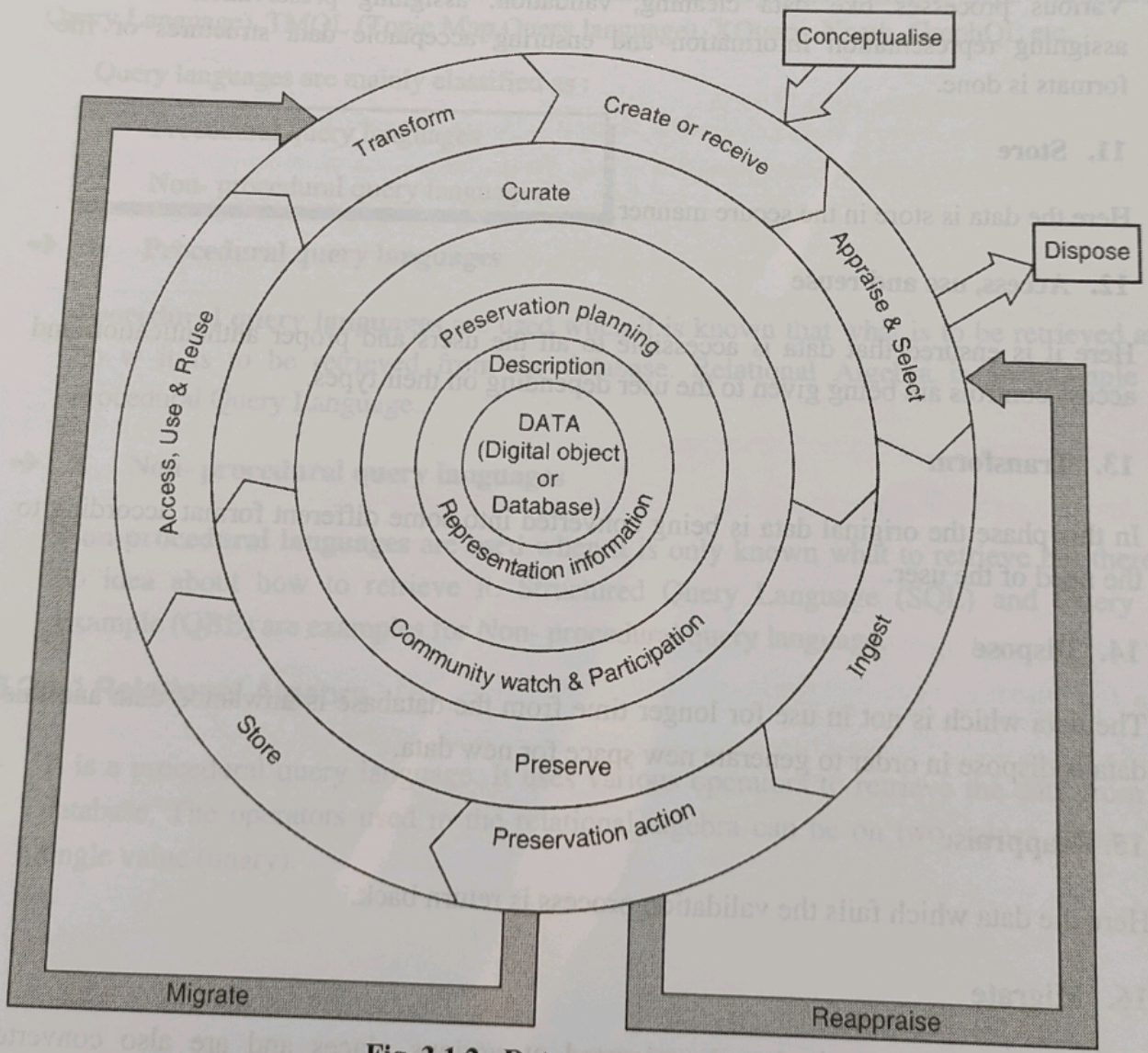

Fig. 3.1.2 : Data curation lifecycle

➜ **8. Appraise and select**

In this phase the evaluation of data and its selection is done which is to be preserved for long term.

➜ **9. Ingest**

In this phase data is transferred to an archive, repository or data centre.

➜ **10. Preservation action**

- Here various actions are ensured for long-term preservation and retention of the data. Preservation actions includes for data to remain authentic, reliable and usable.
- Various processes like data cleaning, validation, assigning preservation metadata, assigning representation information and ensuring acceptable data structures or file formats is done.

➜ **11. Store**

Here the data is store in the secure manner.

➜ **12. Access, use and reuse**

Here it is ensured that data is accessible to all the users and proper authentication and access controls are being given to the user depending on their types.

➜ **13. Transform**

In this phase the original data is being converted into some different format according to the need of the user.

➜ **14. Dispose**

The data which is not in use for longer time from the database is unwanted data and this data is dispose in order to generate new space for new data.

➜ **15. Reappraise**

Here the data which fails the validation process is return back.

➜ **16. Migrate**

Depending on the need, data is migrated at various places and are also converted according to the new environment.

Syllabus Topic : Query Languages and Operations to Specify and Transform Data

3.2 Query Languages and Operations to Specify and Transform Data

Q. 3.2.1 Explain Query languages and their operations. (Ref. Sec. 3.2) **(5 Marks)**

Following section provides various query languages and certain operations they can perform in order to transform the data.

3.2.1 Query Languages

Query languages are special type of languages used for retrieving required data or information from the database or data set / data repository. For example, SQL (Structured Query Language), TMQL (Topic Map Query language), XQuery, Xpath, GraphQL etc.

Query languages are mainly classified as :

1. Procedural query languages
2. Non- procedural query languages

➜ **1. Procedural query languages**

Procedural query languages are used when it is known that what is to be retrieved and ho w it is to be retrieved from the database. Relational Algebra is an example of Procedural Query Language.

➜ **2. Non- procedural query languages**

Non-procedural languages are used when it is only known what to retrieve but there is no idea about how to retrieve it. Structured Query Language (SQL) and Query By Example (QBE) are examples for Non- procedural query language.

3.2.1.1 Relational Algebra

- It is a procedural query language. It uses various operators to retrieve the data from the database. The operators used in the relational algebra can be on two values (binary) or single value (unary).

Various operators used in the relational algebra are as follow :

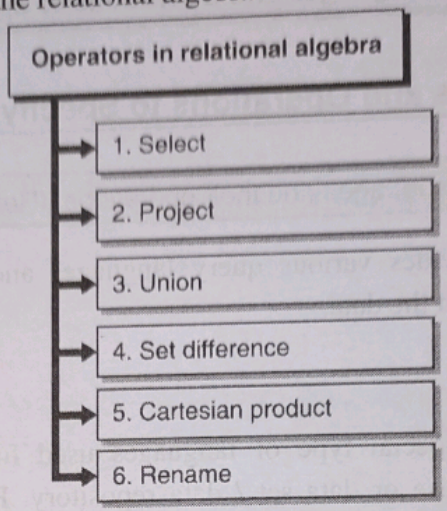

Fig. 3.2.1 : Operators in relational algebra

➜ 1. Select operator (σ) :

- It is use to select particular tuples (row) from the database satisfying the given condition. The symbol **σ** is used to denote the selection operation.
- The operation performed by select operation is represented as **σp(r),** where **r** represents the relation and **p** is the prepositional logical formula. The symbol **p** can use various connectors like "and", "or" and "not", it can also use the relational operators like "=","<",">","=",">=","<=","!=" etc.
- **For example :** σ subject="DataScience" (Books).

  Above query will select tuples from Books where name of subject is "Data Science".

➜ 2. Project operator (Π) :

It is used to project the column(s) that satisfy the given condition. It is denoted as $\Pi a_1, a_2 \ldots a_N(r)$, where $a_1, a_2 \ldots a_N$ are the name of attributes in the relation or table r.

For example : Π subject, author (Books).

Above query will select and project the columns subject and author from Books table or relation.

➜ 3. Union operator (∪) :

- It is use to perform binary union between two relations or tables.

- **For example** : r ∪ s, where r and s are the relations or tables, there are few constraints like r and s should be of same number of attributes, the data type of the attributes should match.

➜ **4. Set difference (–) :**

It is used for removing the tuples which are present in one relation and absent in another. **For example:** r - s, it will remove all the values which are in s but not in r.

➜ **5. Cartesian product (×) :**

- It is use to combine two different relations or tables and makes it as a single relation.
- **For example : r** × s, here r and s are two relations or tables and their output will be values in a single relation or table and it will have all the values of r and s.

➜ **6. Rename operator (ρ) :**

- When queries fired on the table results to a new table to rename that table rename operator is used.
- **For example :** ρ (X, E), where E and X are the name of the table or relation and here E is renamed by X.

3.2.1.2 Joins

Joins are used for combining the Cartesian product followed by a selection process. There are various types of joins which are as follows :

1. Theta (θ) join
2. Natural join (⋈)

➜ **1. Theta (θ) join**

Theta join combines two different relations if the condition satisfies following is the notation of it $R_1 \bowtie_\theta R_2$, where R_1 and R_2 are two different relations which have some attributes, the resulting relation will not consider any attribute which is common in R_1 and R_2.

➜ **2. Natural join (⋈)**

- It does not use any comparison operator. It does not concatenate as done in cartesian product. Natural Join is used when at least one common attribute in the relations on which join is to be done.

- Theta join and natural join are also known as the inner joins. Inner joins considers only those tuples which have matching attributes rest tuples are discarded. To deal with the unmatched attributes outer join concepts are used.

☞ **Outer Joins are of three types**

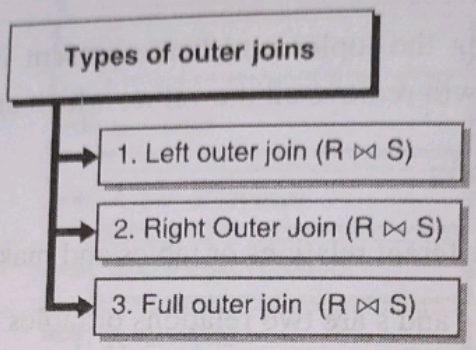

Fig. 3.2.2 : Types of outer joins

➜ **1. Left outer join (R ⋈ S)**

In left outer join all the tuples from the relation left that is R in this case are considered in the output. The tuples which do not have any matching tuples in S from R then that tuples of S are made NULL.

➜ **2. Right Outer Join (R ⋈ S)**

In right outer join all the tuples from the relation right that is S in this case is considered in the output. The tuples which do not have any matching tuples in R form S then that tuples of R are made NULL.

➜ **3. Full outer join (R ⋈ S)**

In full outer join all the tuples of left and right relations are considered and if there are no matching found in the tuples then both relations are made NULL for non matching attributes.

3.2.1.3 Aggregate / Group Functions

These are the functions in which the values of multiple rows are grouped together as input. Various aggregate functions are as follows :

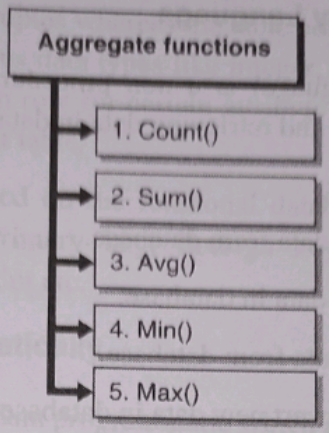

Fig. 3.2.3 : Aggregate functions

➜ 1. Count

It returns the total number of record present For example, Count (*) will give number of rows / tuples in that relation.

➜ 2. Sum

It returns the sum of values for a particular attribute. For example, Sum (salary) will give the sum of salary for all the employee in that relation.

➜ 3. Avg

Avg provides the average of the value for a particular attribute. For example, Avg (salary) will give the average value of salary that is total or sum of all salary divided by total count and returns its value.

➜ 4. Min

It will return minimum value of a particular attribute. For example, Min (salary) it will return the minimum salary value from the salary attribute.

➜ 5. Max

It will return maximum value of a particular attribute. For example, Max (salary) it will return the maximum salary value from the salary attribute.

3.2.1.4 SQL Structured Query Language

SQL (Structured Query Language) is a non procedural language and it is a standa language for storing, manipulating and retrieving data in database.

It has various commands like :

1. **SELECT :** It is use to extract data from database.
2. **UPDATE :** It is use to update data in database.
3. **DELETE:** It is use to delete data from database.
4. **INSERT INTO :** It is use to insert new data in database.
5. **CREATE DATABASE :** It is use to create a new database.
6. **ALTER DATABASE :** It is use to modify database.
7. **CREATE TABLE :** It is use to create a new table.
8. **ALTER TABLE :** It is used to modify table.
9. **DROP TABLE :** It is use to delete table.

Basic syntax for SQL is,

```
SELECT data FROM table_name;
```

Syllabus Topic : Structured/Schema Based Systems as Users and Acquirers of Data

3.3 Structured/Schema Based Systems as Users and Acquirers of Data

Q. 3.3.1 Explain what is structured, semi-structured and unstructured data in detail. (Ref. Sec. 3.3) **(10 Marks)**

- Systems that use structured data are known as structured systems, the data in this type of system are well organized in the form of tables, and these systems are easy to operate that is searching; sorting, accessing and retrieving of data in this system can be easily done.
- Data stored in the relational databases are in the form of tables and can be easily operated on and hence it is a structure system, spreadsheets are also structured system.
- Relational databases also known as RDBMS is the basis for SQL system. It is a database management system which uses table to store the data and hence it is under the structured system.

- The table rows represent the tuples whereas the table columns represent the attributes. The values in the table have various data types like integer, float, string, character etc. In case if there is no value for certain row for certain attribute then in such case NULL value is filled in the blank space in that table.
- Various constraints are applied on the relational databases depending on the condition these constraints are like Primary key, Foreign key, CHECK constraint, UNIQUE constraint, DEFAULT constraint etc.

3.3.1 Access Data Programmatically

There are two ways to access data programmatic which are :

1. Direct downloads/import of data.
2. Applied /Application Programming Interfaces.

1. Direct downloads/import of data

In direct downloads the dataset are downloaded and embedded with the application which someone is developing whereas through API's there is no need to download the data.

Programmatic data access is divided in three categories :

1. Data that you download directly by calling a specific URL.
2. Data that can be directly import from various packages.
3. Data can be downloaded through API where request and returns commands are used to access data.

The data which is accessed programmatically comes in two formats :

1. Tabular Human - readable file
2. Structured Machine - readable files

➜ **1. Tabular human - readable file**

Here the files are in tabular format which are for example csv files.

➜ **2. Structured machine - readable files**

These are in text format but stored in hierarchical or structured way.

3.3.2 Indexing

- It is a way to optimize performance of a database by minimizing the number of disk accesses required whenever a query is processed. Index is a data structure which is used for faster access and locates the data in the database.
- Index have two field first field is search key and second field is data reference. Indexes are of two types :

1. Ordered index
2. Hashing index

➜ **1. Ordered index**

These are based on a sorted ordering of the values.

➜ **2. Hashing index**

These are based on the values which are distributed uniformly across a range of buckets.

Syllabus Topic : Semi-structured Systems as Users and Acquirers of Data

3.4 Semi-structured Systems as Users and Acquirers of Data

Q. 3.4.1 Explain what is structured, semi-structured and unstructured data in detail. (Ref. Sec. 3.4) **(10 Marks)**

- Semi structured data lies in between structured and unstructured data, it is data that has not been organized into a specialized repository, such as database, but that nevertheless has associated information, such as metadata.
- Semi structured data is a type of data that contains semantic tags but they do not have any specific format as it is in structured data.

  Examples include emails, XML, JSON, No SQL and other markup languages.

3.4.1 XML

- HTML (Hypertext Markup Language) is an language which is used to give the structure to the web page, whereas the XML (Extensible Markup Language) is an language which has the capacity to store the data in semi structured format. XML uses the tags in order to store the data in the hierarchical manner.

- These tags are called as the Meaningful tags as this tags are normally defined by keywords which generally tells the meaning what they stand for.
- For example, <deposit> .....</deposit>.The word extensible in the XML means that we can extend the ability to describe the document by increasing the number of meaningful tags.
- XML allows transferring the data between the heterogeneous systems. In various Business models we need to transfer the data in the different format, XML provides a common platform in order to transfer the data using the common means. XML Document is a normal text file with the tags and data inside it and it is saved on the drive by using the extension "filename.xml".
- XML file is seen by using the web browser where it shows the tree hierarchy structure of the data. Similar to HTML, XML is also recommended by the W3C. XML tags are self descriptive and are meaningful.
- There are two main differences between XML and HTML and that is XML is used to present and transfer the data and also it is used to store the data whereas the HTML is used to design the web page.

Advantages of XML

The advantages of the XML are listed as below :

1. Platform independency.
2. Simple and Easy to learn and use.
3. It provides compatibility with various other programming languages like java etc.
4. It provides portability.
5. It is extensible and hence it provides an environment where we can scale the XML document.
6. IT is vendor independent.

Disadvantages of XML

1. The syntax of XML is very large as well as it is redundant.
2. In XML namespaces are used which can create problem while using them.
3. Due to redundancy in the XML code the efficiency is decreased.
4. XML does not support intrinsic data type i.e. they do not have any specification for integer, string, date, float etc.

5. As compared to the relational model or an object oriented graph the hierarchical model for representation is limited in XML.
6. In XML we have overlapping relationships to maintain this we require extra efforts.

☞ Characteristic of XML

1. It is just plain text. Software that can handle plain text can also handle XML.
2. With XML you can create your own tags.
3. XML was created to structure, store and transport information.
4. XML is not a replacement for HTML XML is a complement to HTML.
5. XML is a software and hardware independent tool for carrying information.
6. It is the most common tool for data transmissions between all sorts of applications and is becoming more and more popular in the area of storing and describing information.
7. With XML, data can be stored in separate XML files. This way you can concentrate on using HTML for layout and display and be sure that changes in the underlying data will not require any changes to the HTML.
8. XML simplifies data sharing.

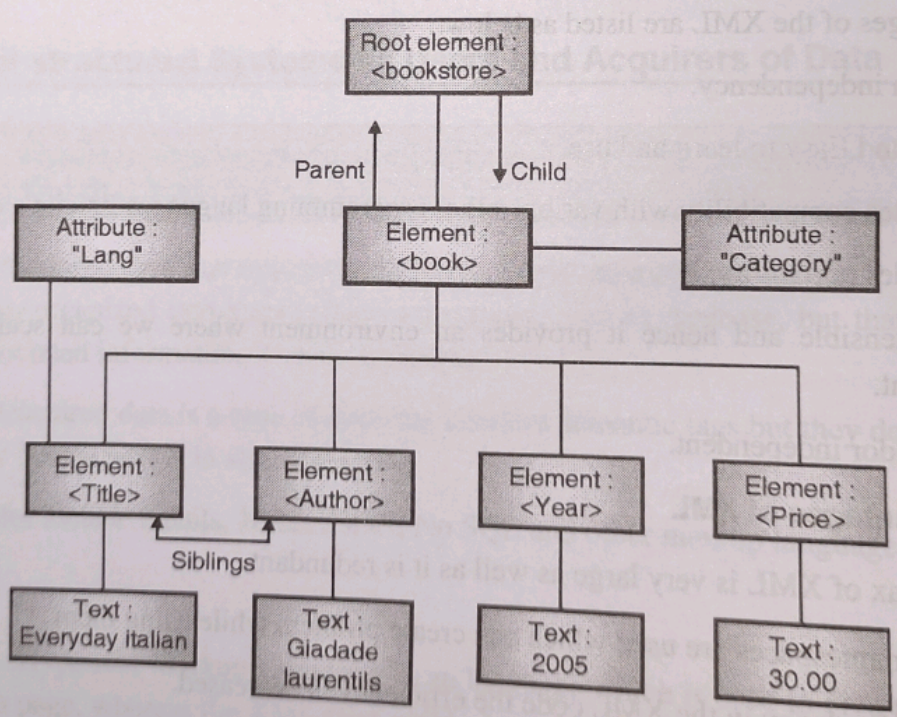

Fig. 3.4.1 : XML structure

9. XML simplifies data transport.
10. XML simplifies platform changes.
11. XML makes your data more available.
12. XML is used to create new internet languages.
13. XML documents form a tree structure.
14. All XML elements must have a closing tag.
15. XML tags are case sensitive.

3.4.1.1 XQuery

Q. 3.4.2 Explain the Querying in XML. (Ref. Sec. 3.4.1.1) (5 Marks)

XQuery in XML is similar to SQL is to database tables. XQuery is designed to query XML data present anywhere on the xml file. XQuery is a language for finding and extracting elements and attributes from XML documents.

To use XQuery following things should be kept in mind.

- In XQuery if the characters are replaced by small or big letters then it may not execute as it is case-sensitive.
- All the variables, data types, elements and attributes must be valid XML Names.
- The values of the attributes in the XQuery should be place in either single Quotes or double.
- In order to define an variable in the XQuery "$" symbol should preceded before the variable name for example: - $employee.

```
for $x in doc("books.xml") / bookstore/book
where $x/price>120
order by $x/title
return $x/title
```

- FLWOR is an acronym for "For, Let, Where, Order by, Return".
- The **for** clause selects all book elements under the bookstore element into a variable called $x.
- The **where** clause selects only book elements with a price element with a value greater than 30.

- The **order by** clause defines the sort-order. Will be sort by the title element.
- The **return** clause specifies what should be returned. Here it returns the title elements.

3.4.1.2 XPath

- XPath stands for XML Path Language and it is a query language in the XML, which is used for selecting a particular child or parent node from the given XML document. The syntax of the XPath is very much different than that of the XQuery.
- In an XML document XPath is also used for computing various. XPath was defined by the World Wide Web Consortium (W3C). Sometimes, the data in an XML document needs to search before it can be formatted and renders using a style sheet at that times XSL provides the XPath language. XPath is used to search and retrieve information from an XML file.

```
<?xml version="1.0" ?>

<bookstore>
< book>
< bookname language="English">XML</ bookname>
<author>Vijay</author>
<year>2010</year>
<cost>250</cost>
</book>
</bookstore>
```

- The structure depicts the different type of nodes in an XML document. For example bookstore, bookname, author, year and cost are element nodes. Language is attribute node. Values XML, Vijay, 2010, 250 represents text node.
- XPath provides a set of expressions and function that can be used to match nodes in an XML document. XPath expressions can be used to retrieve data based on specific conditions you can apply constraints by adding a filter clause also known as a filter pattern. Using XPath, you can create expressions that identify the nodes in an XML document based on their names and values.

Operator/Special character	Example	Description
/	/bookstore	The document element (<bookstore>) of this document select the immediate child elements of /bookstore.

Operator/Special character	Example	Description
//	//author	All <author> elements in the document. Searches for the specified element at any node level.
.	.first-name	All <first-name> elements in the current context node.
..	../author	Select the author element, which exists within the parent of current elements.
*	*	Select all node.
@	@style	The style attribute of the current context.
@*	@*	All attributes of the current element context.
:	:	Separates the namespace prefix from the element or attributes.
()	(height*width)	Group operation.
[]	book[@color="red"]	Apply filter pattern.
+	N1+N2	Return sum of two numbers.
–	N1–N2	Return difference of two numbers.
*	N1*N2	Return product of two numbers.
Div	N1 div N2	Return quotient of two numbers.
Mod	N1 mod N2	Return modules.

- XML document has tree like structure and XPath language is best query language for tree like structure. The XPath language travels through various nodes in order to retrieve the data; a particular node is selected based on some criteria.
- The XPath language actually specifies complete path through which data can be retrieved.

3.4.2 JSON

- It stands for JavaScript Object Notation and it is used for storing and exchanging data, it is a text written using the JavaScript Object Notation.

- JSON can be converted to JavaScript object and can be send to server while exchanging the data or for communication purpose. While storing the data it has to be of certain format if there is no specific format then text is one of the legal forms to store data, JSON can convert the JavaScript object in text. JSON is lightweight data interchange format, easy to understand, self describing and is language independent.
- JSON data is written as name/ value pairs which consists of the field name (attribute) and followed by its value

  For example, "subject": "Data Science".
- JSON supports various data types such as string, number, array, boolean, null and objects. JSON files are stored with extension .json and MIME type for JSON text is "application/json".

Syllabus Topic : Unstructured Systems in the Acquisition and Structuring of Data

3.5 Unstructured Systems in the Acquisition and Structuring of Data

Q. 3.5.1 Explain what is structured, semi-structured and unstructured data in detail. (Ref. Sec. 3.5) **(10 Marks)**

- Data which is not structured and is available in unformatted way is unstructured data. Unstructured data is present on the large scale and to handle this data is a major challenge.
- Unstructured data includes data like text, multimedia, word documents, videos, photos, sensors data, web pages and many others. The source of unstructured data are broadly classified as Machine - Generated unstructured data and Human Generated Unstructured data.
- Machine Generated Unstructured data are Satellite images, Scientific data, Photographs, and Radar or Sonar data.
- Human Generated Data are Social media data such as on Facebook, Mobile data, Text internal to your company and Website content.

3.5.1 Web Scraping

Q. 3.5.2 What is Web Scraping? (Ref. Sec. 3.5.1) **(5 Marks)**

- Web scraping has lot many synonyms like Screen Scraping, Web Data Extraction, and Web Harvesting etc. It is technique used for extracting large amount of data from websites.

- The data extracted through web scraping can be stored in the local file in the computer system. Web scraping software's or tools automatically copies the complete website data and saves it on the machine in the fractional time.
- Depending on the requirement web scraping tools can copy all the pages of the website with the content and the data and links and can load it locally on the machine in offline mode.
- Before the page is copied or extracted it has to be fetch and hence before web scraping, web crawling is done. Web crawling is the process of selecting a particular web page.
- Many websites which compares the products uses web crawling and web scraping process in order to get the information from other sites. Web scraping can also be considered as data mining tool.

Syllabus Topic : Security and Ethical Considerations in Relation to Authenticating and Authorizing Access to Data on Remote Systems

3.6 Security and Ethical Considerations in Relation to Authenticating and Authorizing Access to Data on Remote Systems

Q. 3.6.1 Explain Authentication and Authorization for Storage system. (Ref. Sec. 3.6) **(5 Marks)**

Security is an important parameter for any data storage system. Various security attacks that can be faced in any system can be :

1. Password guessing attack
2. Replay attack
3. Man-in-the-middle attack
4. Masquerade attack
5. Insider attack
6. Phishing attack
7. Shoulder surfing attack.

Authentication and Authorization are two major process used for security of data on the remote system.

☞ Authentication

- It is a process for confirming the identity of the user. The basic way of providing authentication is through username and password, but many a time this approach fails due to hackers or attackers if some hacker will able to crack the password and username than even the hacker will able to use the system.

- Various user authentication methods like x.509 certificates, one - time passwords (OTP), Device finger printing can be used for authentication and these methods are powerful then the traditional approach of username and password.

☞ **Authorization**

- It follows the authentication step which means that once the authentication of a particular user is done the next step is authorization which is to check what rights are given to that user.
- During the process of authentication policies are made which define the authorities of that user.

☞ **Various algorithms used for authentication and authorization are :**

1. RSA algorithm.
2. AES algorithm and MD5 hashing algorithm.
3. OTP password algorithm.
4. Data encryption standard algorithm.
5. Rijndael encryption algorithm.

Syllabus Topic : Software Development Tools

3.7 Software Development Tools

- Software or application development tools are the set of platform, environment and software's needed in order to develop an application.
- Usually an application needs more than one tool and a team to develop it, many time the changes has to be done in the previously developed and deployed application or system in such cases software tools for version control or environment like GitHub is required. Following section explains commonly used version and source control software.

3.7.1 Version Control / Source Control

Q. 3.7.1 Explain in brief Version Control (Ref. Sec. 3.7.1) **(5 Marks)**

- Source control is part of a larger strategy to take one-off, hodge-podge code and make it well documented and ready for larger scale development. With source control, keeping

track of your code doesn't curtail development. Source control is also called as version control or revision control.

- Revision control system maintains the set of all versions of files which are organised in a particular manner along the time. Furthermore, you allow yourself the freedom to experiment, remove or make drastic changes to your code base knowing full well that you can rewind in time and compare what was changed or even revert back if need be.
- You can add the following files to visual studio source control :
  - Solution files (*.sln).
  - Project files, for example, *.csproj, *.vbproj files.
  - Application configuration files, based on XML, used to control run-time behavior of a Visual Studio project.
- Files that you cannot add to source control include the following:
  - Solution user option files (*.suo).
  - Project user option files, for example, *.csproj.user, *.vbproj.user files.
  - Web information files, for example, *.csproj.webinfo, *.vbproj.webinfo, that control the virtual root location of a web project.
  - Build output files, for example, *.dll and *.exe files.

3.7.1.1 Version Control Terminology / Functionalities

The visual studio documentation uses a number of terms to describe source control features and concepts.

Common terms used in the version control are as follow:

- **Basis version :** The server version of a file from which a local version is derived.
- **Binding :** Information that correlates a working folder for a solution or project on disk to its folder in the database.
- **Branching :** Process of creating a new version, or branch of a shared file or project under source control. Once a branch has been created, the two versions under source control will have a shared history up to a certain point and divergent histories after that point.
- **Conflict :** Two or more different changes to the same line of code in situations where two or more developers have checked out and edited the same file.
- **Connection :** A live data link between a source control client (for example, Visual Studio) and a source control database server.

- **Database :** Location where all master copies, history, project structures, and user information are stored. A project is always contained within one database. Multiple projects can be stored in one database, and multiple databases can be used. Other terms commonly used for a database are repository and store.
- **History :** Record of changes to a file since it was initially added to source control. With version control, you can return to any point in the file history and recover the file as it existed at that point.
- **Label/Tag :** User-defined name attached to a specific version of a source-controlled item.
- **Local copy :** File in a user's working folder to which changes are saved until a check-in occurs. A local copy is sometimes referred to as a working copy,
- **Master copy :** The most recently checked-in version of a source-controlled file, as opposed to the local copy of a file in your working folder. Other terms for master copy are server version and database version.
- **Merging :** Process of combining differences in two or more modified versions of a file into a new file version. Merging can affect different versions of the same file or changes made to the same file version.
- **Shared file :** A file having versions that reside in more than one source control location. Other terms for a shared file are copy and shortcut.
- **Solution root :** An empty folder in a database that contains all items in a source-controlled solution. By default, this folder is <solutionname>.root.
- **Super-unified root :** A virtual container beneath which all projects and files in a source-controlled solution are located. For example [SUR]:\ is the super-unified root of a source-controlled solution containing projects that are located in [SUR]:\C:\Solution\ProjOne and [SUR]:\D:\ProjTwo.
- **Unified root :** A path to the parent directory for all working folders and files in a source-controlled solution or project. For example, C:\Solution is the unified root of a source-controlled solution containing files that are located in C:\Solution, C:\Solution\ProjOne and C:\Solution\ProjTwo.
- **Working folder :** Location where your local copies of source-controlled items are stored, usually on your own computer. Another term for a working folder is workspace.
- **Baseline :** It is a version of a document on which we can make the changes. This document is also called as source file.

- **Branch :** It is a point from where the source file is branched and after this two copies of this file will be kept. The branches are also called as forks.
- **Checkout :** It is a process of creating a local working copy from the repository. At any instance of time a user can obtain the working local copy by using the checkout.
- **Commit :** It is an opposite process of checkout and hence can be also called as check-in. In this process the changes which are made in the working local directory are saved back into the Repository.
- **Trunk :** It is called as a mainline or the unique line of the development which is not actually a branch.

The overall diagram of source control or version control can be shown as in Fig. 3.7.1.

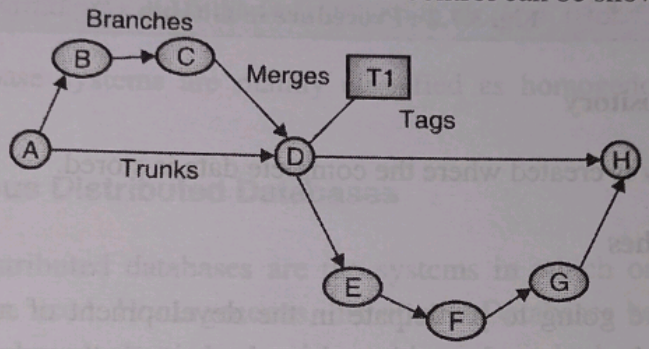

Fig. 3.7.1 : An example of version control flow

3.7.2 GitHub

Q. 3.7.2 Explain in brief GitHub. (Ref. Sec. 3.7.2) **(5 Marks)**

- GitHub is a platform for code hosting and collaborative development for systems. GitHub removes the location barrier; any member of the team can work on a particular project from any location and can upload or download the files or folders any time.
- Recently GitHub is used by many people for sharing the codes and the files.
- GitHub is used for doing version control, as simultaneously number of people can upload their part of code and then merge all the sub parts of the code.
- GitHub has a central repository where the data for a particular project is stored and using branches many people can link to that project and update their part as and when done.
- Git is a version control tool used in GitHub to perform various jobs like fetching data pushing and pulling request to and from the central repository to the user and vice versa.

Following is the procedure followed in GitHub.

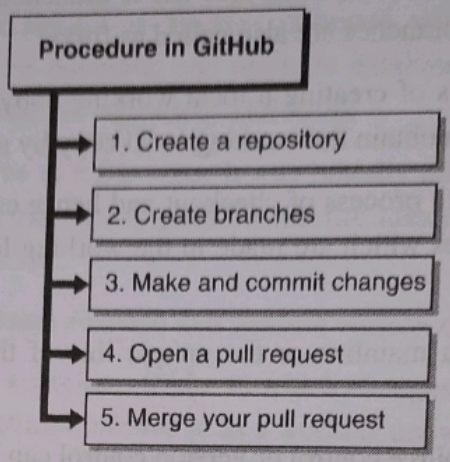

Fig. 3.7.2 : Procedure in GitHub

➜ **1. Create a repository**

A central repository is created where the complete data is stored.

➜ **2. Create branches**

Users those who are going to participate in the development of application are two be connected to the main or central repository branches acts as the connectivity link between central repository and the user.

➜ **3. Make and commit changes**

Once the changes are done they have to updated on the main repository commit is used to make that changes and update the central repository.

➜ **4. Open a pull request**

If any changes have to be reflected on particular user's module then he can make a pull request which is used to get the update from various other members of the team.

➜ **5. Merge your pull request**

Once the development part of every branch is done completely or partially it is merged and stages of the projects are developed.

Syllabus Topic : Large Scale Data Systems

3.8 Large Scale Data Systems

To store the large data normal databases cannot be used and hence data bases like NoSQL, MongoDB and HBase etc are good option for large scale data systems. Large scale systems do not always have centralized data storage. Distributed database approach is widely used in many applications

3.8.1 Paradigms of Distributed Database Storage

Q. 3.8.1 Explain Paradigms of Distributed Database storage. (Ref. Sec. 3.8.1) (5 Marks)

Distributed database systems are mainly classified as homogenous and heterogeneous database.

3.8.1.1 Homogeneous Distributed Databases

- Homogeneous distributed databases are the systems in which on all the sites identical DBMS and OS are used. Homogeneous, distributed Databases have identical software's and here every site know what is happening at other site and where it is located.
- Homogeneous distributed databases are further classified as autonomous and non-autonomous. In autonomous database each site is independent for processing only the integration is done using some controlling application.
- In non-autonomous databases data is distributed across the various nodes or sites and one node manages all the other nodes as if like client server model.

3.8.1.2 Heterogeneous Distributed Database

- In Heterogeneous distributed database every site has different database, OS and different software's. In such system querying is complex as the environment and all other tools are different.
- Heterogeneous distributed database is further classified as federated and un-federated databases. In federated database system every site is independent of each other and hence it acts as a single database system individually. In un-federated database system there is a single central coordinator module through which all the sites communicates.

3.8.2 NoSQL

Q. 3.8.2 Write a short note on NoSQL. (Ref. Sec. 3.8.2) **(5 Marks)**

- NoSQL also known as Not only SQL is an alternative to relational database system. It refers to all the databases which are not based on RDBMS principle, and is used to access and manipulate the data on the large scale.
- It is an non- relational data model which is open source, runs well on clusters, schema less and is built for new generation web applications. As the sources of data are increasing and hence the data is also generated on large scale and in different formats to handle this type of data traditional databases are not capable and hence a concept of NoSQL is developed.

Advantages of using NoSQL

1. It provides support to query languages.
2. It optimizes the performance of the system.
3. It provides horizontally scalability.
4. It supports dynamic schemas.
5. It supports sharding, a process in which the large datasets or databases are partitioned into smaller and faster and easily manageable databases.
6. It does auto replication of the data, which helps in case of data loss.
7. It supports the feature of integrated caching due to which frequently required data are cached and can be accessed faster whenever needed.

NoSQL Database Types

NoSQL support various types of databases which are as follows :

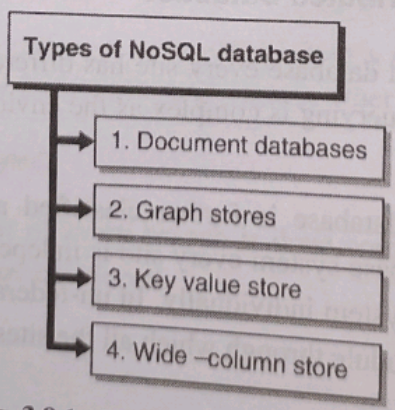

Fig. 3.8.1 : Types of NoSQL database

➜ **1. Document databases**

Here there is key which is paired with document for e.g. MongoDB.

➜ **2. Graph stores**

This type of database is usually used to store networked data where nodes and links between nodes are done. Nodes represent the data and links represents the connection.

➜ **3. Key value store**

It is the simplest NoSQL database in which every key represents some identification and the data here is store in key-value pair.

➜ **4. Wide - column store**

It is used to store large data sets for example Facebook data is stored on Cassandra which is of wide-column store database type.

3.8.3 MongoDB

Q. 3.8.3 Write a short note on Mongo DB (Ref. Sec. 3.8.3) **(5 Marks)**

It is a NoSQL database and is open source and document oriented database. It support cross platform and is written in C++ language. MongoDB provides high performance, high availability and automatic scaling to the database.

There are various features of MongoDB which are as follow :

1. It supports ad hoc queries, which mean that queries do not have any specific way to be fired.
2. It supports indexing, so that the data can be indexed properly.
3. It supports replication, which help us to scale the data as and where needed, this is done using master and slave configuration.
4. Data duplication is done in MongoDB so as to scale the database.
5. It does automatic load balancing so that performance can be optimized.
6. It supports map reduce and aggregation tools.
7. It uses JavaScript and it is not procedure oriented.
8. It provides high performance

Data types supported in MongoDB includes String, integer, boolean, dozuble, min/max keys, arrays, objects, null, symbol, date etc.

3.8.4 HBase

Q. 3.8.4 Write a short note on HBase. (Ref. Sec. 3.8.4) **(5 Marks)**

- It is a data model and it is similar to Google's big table. HBase provides quick access to large amount of structured data. It has a distributed column - oriented database structure and is placed on the Hadoop File System (HDFS). To store, read and access the data in HDFS HBase is used.
- HBase provides fast lookups for larger tables and low latency access to single rows from billions of records. Hashing technique is used in HBase. It is used in companies like Facebook, Twitter, Yahoo etc where large amount of the data storage and fast and continuous accessing of data is done.
- Tables in HBase are sorted by row, and as HBase has column - oriented database the table schema defines only column families.

HBase Features

1. It is linearly scalable and can grow upto any size.
2. It provides automatic feature to recover from failure.
3. Read and write operations can be done consistently.
4. It provides easy interfaces for communication with different systems.
5. It supports clustering and replication.

Syllabus Topic : Amazon Web Services (AWS)

3.9 Amazon Web Services (AWS)

Q. 3.9.1 Explain AWS in brief. (Ref. Sec. 3.9) **(5 Marks)**

- Amazon Web Services (AWS) is Amazon's Cloud web hosting platform that offers flexible, reliable, scalable, easy-to-use, and cost effective solutions. Amazon started providing web and other IT services from 2006, these services started covering IT market very fast and hence to handle demand Amazon started using cloud.

- The cloud environment started by Amazon provides various types of IT services to many costumers at lower cost and software and hardware components. It is a pay per use type of services.
- Cloud environment is an internet based computing environment which consists large groups of remote servers connected together and a centralized system is developed. Cloud environment also known as cloud computing provides lots of features like :
  1. Pay per use method for using resources.
  2. Resources can make available and released whenever needed.
  3. Scaling of number of resources can be done dynamically.
  4. Resources can be accessed from any place wherever internet facility is available.

Depending on the type of the services and the need of users clouds are classified into three types :

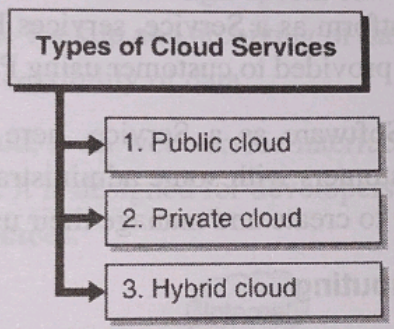

Fig. 3.9.1 : Types of cloud services

➜ 1. Public cloud

In Public cloud, service provider of resources and services use internet as the communication and service link. Anyone can ask for service here that is everyone can be customer.

➜ 2. Private cloud

In Private Cloud the services are provided and managed by some organization or third party but these services are limited to specific customer of that organization only, any customer out of the organization cannot use the services provided by this cloud.

➜ 3. Hybrid cloud

A hybrid cloud is the combination of both private and public cloud. Here both the facilities of public and private cloud can be used.

3.9.1 Cloud Services

Cloud environment mainly provides three types of services :

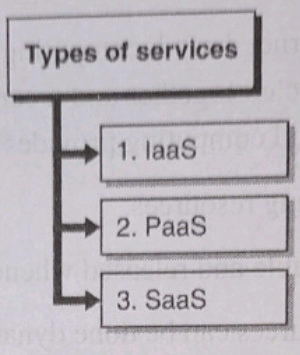

Fig. 3.9.2 : Types of services

➜ 1. **IaaS :** It stands for Infrastructure as a Service, services such as provision processing, storage and network connectivity on demand are provided in IaaS.

➜ 2. **PaaS :** It stands for Platform as a Service, services like databases, queues, workflow engines, emails, etc are provided to customer using PaaS.

➜ 3. **SaaS :** It stands for Software as a Service, here third party provides end- user applications to their customers with some administrative capability at the application level, such as the ability to create and manage their users.

☞ **Advantages of cloud computing**

1. **Cost efficient :** Pay for use facility of cloud computing provides cost efficiency.
2. **Reliability :** It provides reliable and consistent service than an in house IT infrastructure, the availability of this services are 24X7 and 365 days of services.
3. **Unlimited storage :** Cloud computing provides unlimited storage capacity.
4. **Backup and recovery :** Backup and storage of the data and also recovering it back becomes easy in cloud.
5. **Easy access to information :** Users of the cloud have to just make a login once logged in the user can easily access the services or information any time it is needed.

☞ **Disadvantages of cloud computing**

1. **Security issues :** As the cloud is common setup for number of users and hence there can be more security issues, regarding authentication and common storage spaces and software used.
2. **Technical issues :** As number of users on the cloud increase day by day with their different needs and hence to make available and manage various software and hardware is challenge.

3.9.2 AWS Basic Architecture

Q. 3.9.2 Explain the basic architecture of AWS. (Ref. Sec. 3.9.2) (5 Marks)

- AWS basic architecture is shown in the Fig. 3.9.3 it is an AWS EC2 architecture where EC2 stands for Elastic Compute cloud. EC2 provides a user to choose virtual machine with any configuration as per their requirement. EC2 provides wide variety of services and storage with various configurations and price.
- EBS stands for Elastic Block Store; it provides persistent block storage volumes for use in Amazon EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated within its availability zone to protect you from component failure. It also offers high availability and durability, consistent and low-latency performance needed to run workloads. Amazon EBS avails scalability within no time.
- S3 stands for Simple Storage Service. It allows the users to store and retrieve various types of data using API calls. It is an high speed, low-cost, scalable web based service which is designed for online backup and archiving of data and application programs. User here has control over the accessibility of data.
- EC2 stands for Elastic Cloud, is an web service interface that provides resizable compute capacity in the AWS cloud. It is designed for developers in order to have control over web scaling and computing resources.

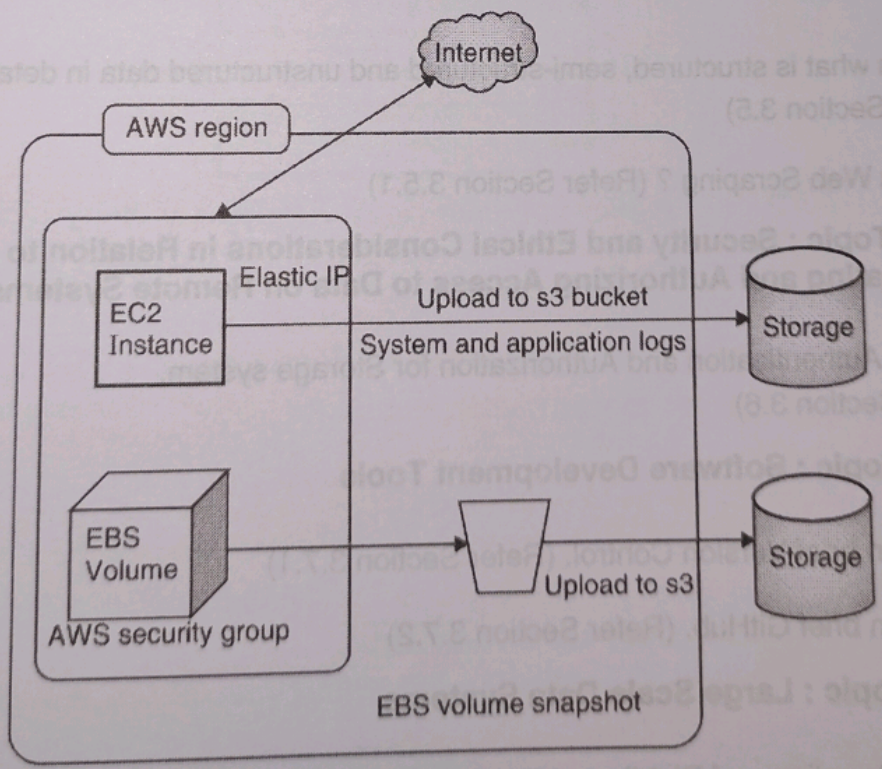

Fig. 3.9.3 : Basic AWS EC2 Architecture

3.10 Exam Pack (Review Questions)

Q. 1 What is Data Curation ? (Refer Section 3.1) **(5 Marks)**

Q. 2 Explain Data Curation Lifecycle in detail. (Refer Section 3.1.1) **(5 Marks)**

☞ **Syllabus Topic : Query Languages and Operations to Specify and Transform Data**

Q. 3 Explain Query languages and their operations. (Refer section 3.2) **(5 Marks)**

☞ **Syllabus Topic : Structured/Schema Based Systems as Users and Acquirers of Data**

Q. 4 Explain what is structured, semi-structured and unstructured data in detail. (Refer section 3.3) **(10 Marks)**

☞ **Syllabus Topic : Semi-structured Systems as Users and Acquirers of Data**

Q. 5 Explain what is structured, semi-structured and unstructured data in detail. (Refer section.3.4) **(10 Marks)**

Q. 6 Explain the Querying in XML. (Refer section 3.4.1.1) **(5 Marks)**

☞ **Syllabus Topic : Unstructured Systems in the Acquisition and Structuring of Data**

Q. 7 Explain what is structured, semi-structured and unstructured data in detail. (Refer Section 3.5) **(10 Marks)**

Q. 8 What is Web Scraping ? (Refer Section 3.5.1) **(5 Marks)**

☞ **Syllabus Topic : Security and Ethical Considerations in Relation to Authenticating and Authorizing Access to Data on Remote Systems**

Q. 9 Explain Authentication and Authorization for Storage system. (Refer Section 3.6) **(5 Marks)**

☞ **Syllabus Topic : Software Development Tools**

Q. 10 Explain in brief Version Control. (Refer Section 3.7.1) **(5 Marks)**

Q. 11 Explain in brief GitHub. (Refer Section 3.7.2) **(5 Marks)**

☞ **Syllabus Topic : Large Scale Data Systems**

Q. 12 Explain Paradigms of Distributed Database storage. (Refer Section 3.8.1) **(5 Marks)**

Q. 13 Write a short note on NoSQL. (Refer Section 3.8.2) **(5 Marks)**

Q. 14 Write a short note on Mongo DB. (Refer Section 3.8.3) **(5 Marks)**

Q. 15 Write a short note on HBase. (Refer Section 3.8.4) **(5 Marks)**

☞ **Syllabus Topic : Amazon Web Services (AWS)**

Q. 16 Explain AWS in brief. (Refer Section 3.9) **(5 Marks)**

Q. 17 Explain the basic Architecture of AWS. (Refer Section 3.9.2) **(5 Marks)**

Chapter Ends...

❑❑❑