click for notes

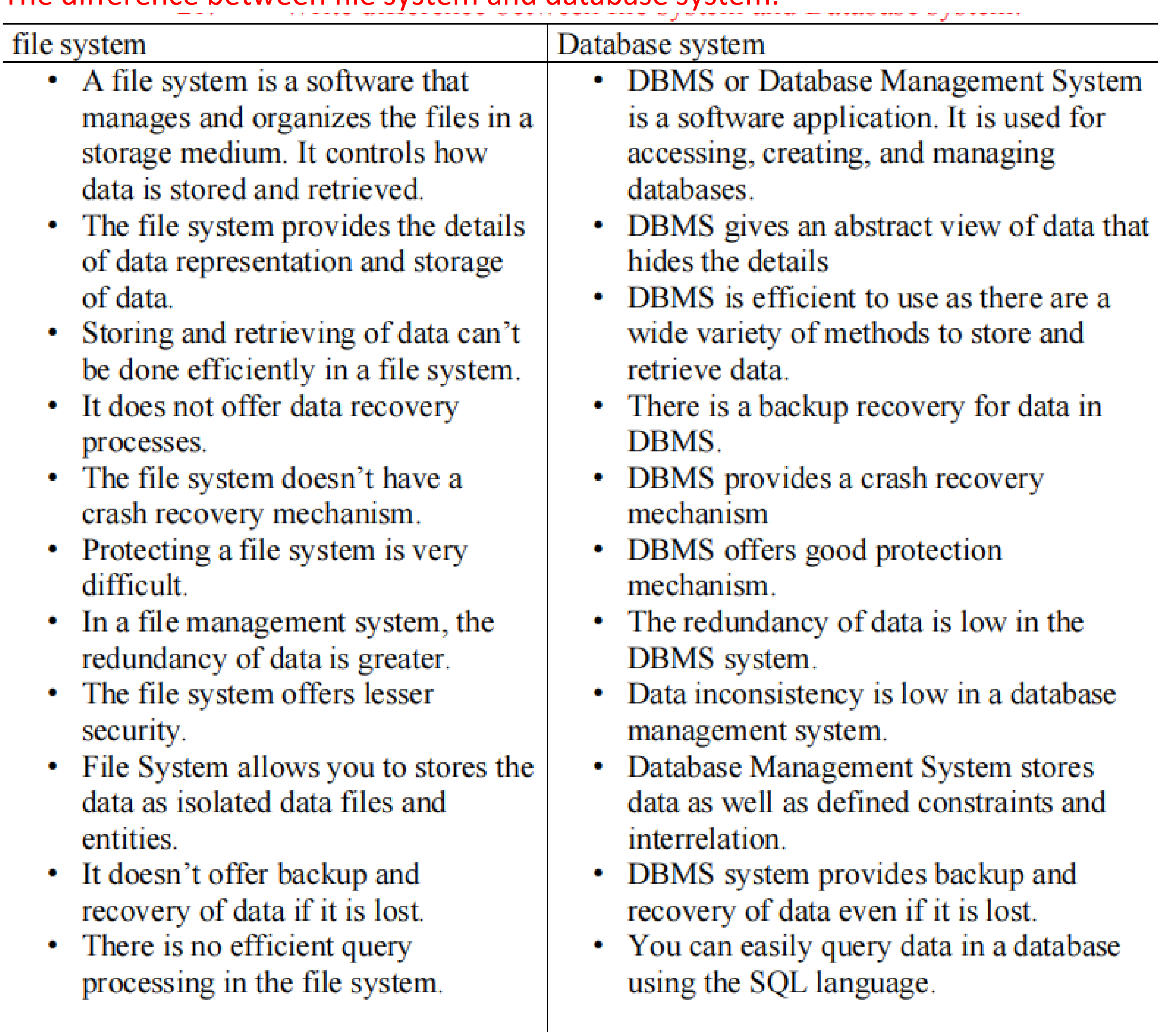
Unit 1

1. What is the objective to study DBMS?

The objectives of DBMS can be narrated as follows:

- Eliminate redundant data.
- Make access to the data easy for the user.
- Provide for mass storage of relevant data.
- Protect the data from physical harm and un-authorised systems.
- Allow for growth in the data base system.
- Make the latest modifications to the data base available immediately.
- Allow for multiple users to be active at one time.
- Provide prompt response to user requests for data.

2. The difference between file system and database system.



file system	Database system
• A file system is a software that manages and organizes the files in a storage medium. It controls how data is stored and retrieved.	• DBMS or Database Management System is a software application. It is used for accessing, creating, and managing databases.
• The file system provides the details of data representation and storage of data.	• DBMS gives an abstract view of data that hides the details
• Storing and retrieving of data can't be done efficiently in a file system.	• DBMS is efficient to use as there are a wide variety of methods to store and retrieve data.
• It does not offer data recovery processes.	• There is a backup recovery for data in DBMS.
• The file system doesn't have a crash recovery mechanism.	• DBMS provides a crash recovery mechanism
• Protecting a file system is very difficult.	• DBMS offers good protection mechanism.
• In a file management system, the redundancy of data is greater.	• The redundancy of data is low in the DBMS system.
• The file system offers lesser security.	• Data inconsistency is low in a database management system.
• File System allows you to stores the data as isolated data files and entities.	• Database Management System stores data as well as defined constraints and interrelation.
• It doesn't offer backup and recovery of data if it is lost.	• DBMS system provides backup and recovery of data even if it is lost.
• There is no efficient query processing in the file system.	• You can easily query data in a database using the SQL language.

3. Give examples DBMS?

Examples of DBMS's include **MySQL, PostgreSQL, Microsoft SQL Server, Oracle Database, and Microsoft Access.**

click for notes

4. Three-tier architecture

   of DBMS.

   Two-tier architecture

   of DBMS

   One-tier architecture

   of DBMS

A **Database Architecture** is a representation of DBMS design. It helps to design, develop, implement, and maintain the database management system. A DBMS architecture allows dividing the database system into individual components that can be independently modified, changed, replaced, and altered. It also helps to understand the components of a database.

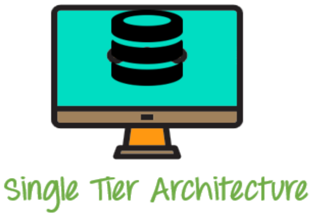
Types of DBMS Architecture

There are mainly three types of DBMS architecture:

- One Tier Architecture (Single Tier Architecture)
- Two Tier Architecture
- Three Tier Architecture

1-Tier Architecture

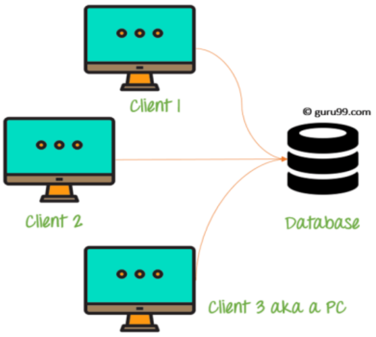
1 Tier Architecture in DBMS is the simplest architecture of Database in which the client, server, and Database all reside on the same machine. A simple one tier architecture example would be anytime you install a Database in your system and access it to practice SQL queries. But such architecture is rarely used in production.



1 Tier Architecture Diagram

2-Tier Architecture

A **2 Tier Architecture** in DBMS is a Database architecture where the presentation layer runs on a client (PC, Mobile, Tablet, etc.), and data is stored on a server called the second tier. Two tier architecture provides added security to the DBMS as it is not exposed to the end-user directly. It also provides direct and faster communication.



2 Tier Architecture Diagram

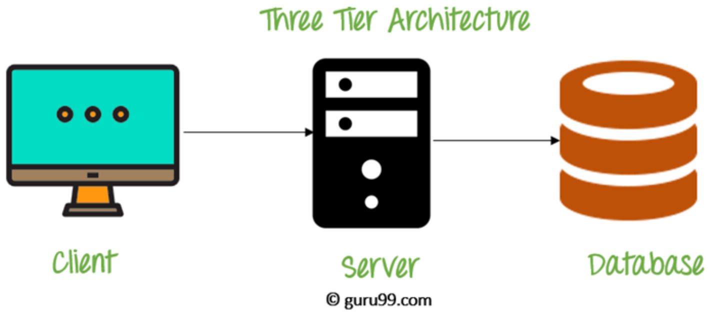
click for notes

3-Tier Architecture

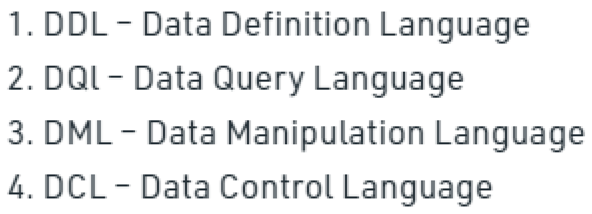
A **3 Tier Architecture** in DBMS is the most popular client server architecture in DBMS in which the development and maintenance of functional processes, logic, data access, data storage, and user interface is done independently as separate modules. Three Tier architecture contains a presentation layer, an application layer, and a database server.

3-Tier database Architecture design is an extension of the 2-tier client-server architecture. A 3-tier architecture has the following layers:

1. Presentation layer (your PC, Tablet, Mobile, etc.)
2. Application layer (server)
3. Database Server



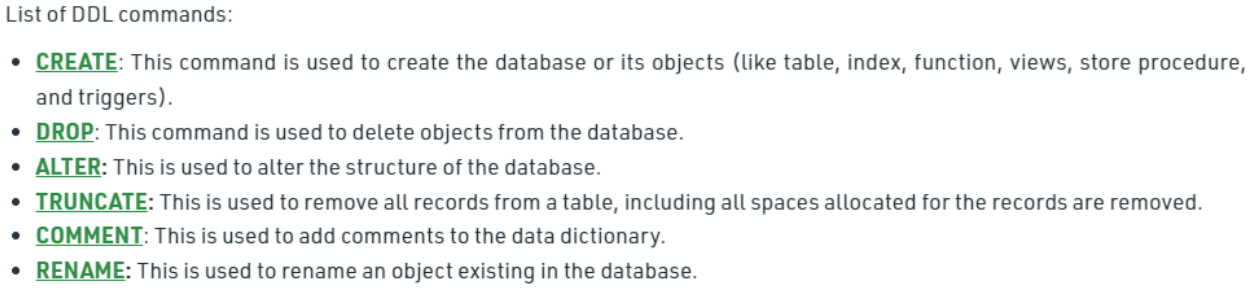
5. Write are the Full form of DDL, DML and DCL?



1. DDL – Data Definition Language
2. DQl – Data Query Language
3. DML – Data Manipulation Language
4. DCL – Data Control Language

6. Explain DDL, DML, DCL and TCL.

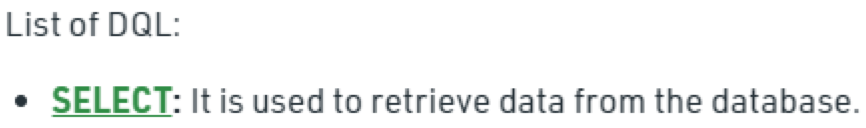
DDL



List of DDL commands:

- **CREATE**: This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).
- **DROP**: This command is used to delete objects from the database.
- **ALTER**: This is used to alter the structure of the database.
- **TRUNCATE**: This is used to remove all records from a table, including all spaces allocated for the records are removed.
- **COMMENT**: This is used to add comments to the data dictionary.
- **RENAME**: This is used to rename an object existing in the database.

DQL

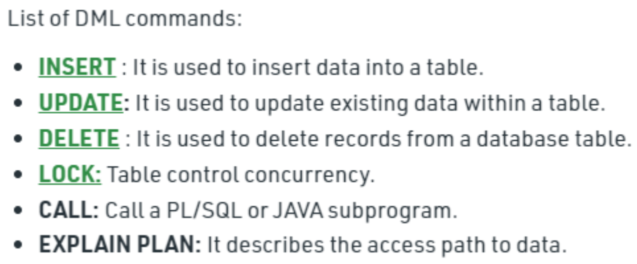


List of DQL:

- **SELECT**: It is used to retrieve data from the database.

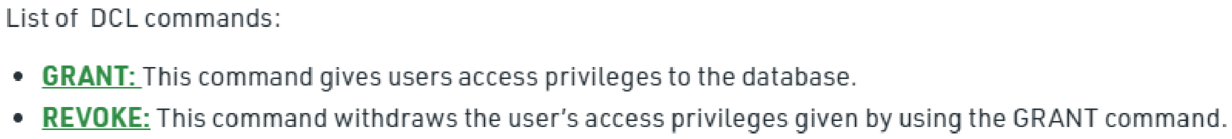
[click for notes]

DML


List of DML commands:

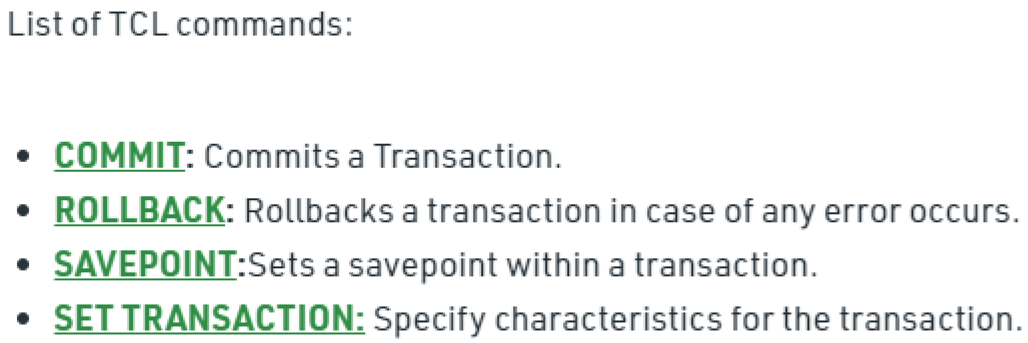
- **INSERT** : It is used to insert data into a table.
- **UPDATE**: It is used to update existing data within a table.
- **DELETE** : It is used to delete records from a database table.
- **LOCK:** Table control concurrency.
- **CALL:** Call a PL/SQL or JAVA subprogram.
- **EXPLAIN PLAN:** It describes the access path to data.

DCL


List of DCL commands:

- **GRANT:** This command gives users access privileges to the database.
- **REVOKE:** This command withdraws the user's access privileges given by using the GRANT command.

TCL


List of TCL commands:

- **COMMIT**: Commits a Transaction.
- **ROLLBACK**: Rollbacks a transaction in case of any error occurs.
- **SAVEPOINT**:Sets a savepoint within a transaction.
- **SET TRANSACTION:** Specify characteristics for the transaction.

Unit 2

1. Write SQL query to get the Student name whose salary is between 1000 and 2000.

   select name from Student where salary between 1000 and 2000;

2. Write SQL query to get the Students Name and address from Student table.

   select Name,address from Student;
3. Write SQL query to get the Students Name who have got marks greater than 90 from Student table
   select Name from Student where marks>90;
4. Explain Union and Intersection in DBMS.

UNION

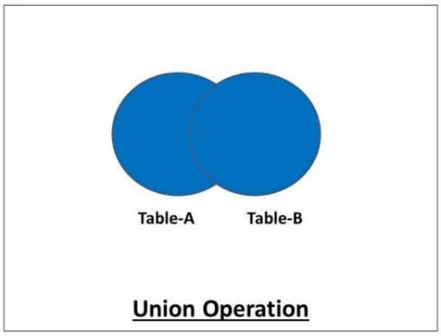
The Union is a binary set operator in DBMS. It is used to combine the result set of two select queries. Thus, It combines two result sets into one. In other words, the result set obtained after union operation is the collection of the result set of both the tables.

But two necessary conditions need to be fulfilled when we use the union command. These are:

click for notes

1. Both SELECT statements should have an equal number of fields in the same order.
2. The data types of these fields should either be the same or compatible with each other.

The Union operation can be demonstrated as follows:



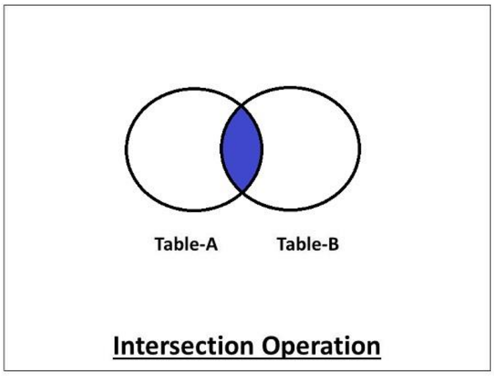
```
SELECT color_name FROM colors_a UNION SELECT color_name FROM colors_b;
```

INTERSECT

Intersect is a binary set operator in DBMS. The intersection operation between two selections returns only the common data sets or rows between them. It should be noted that the intersection operation always returns the distinct rows. The duplicate rows will not be returned by the intersect operator.

Here also, the above conditions of the union and minus are followed, i.e., the number of fields in both the SELECT statements should be the same, with the same data type, and in the same order for the intersection.

The intersection operation can be demonstrated as follows:



```
SELECT color_name FROM colors_a WHERE color_name IN(SELECT color_name
FROM colors_b);
```

click for notes

5. Explain join in DBMS.

Join

In DBMS, a join statement is mainly used to combine two tables based on a specified common field between them. If we talk in terms of Relational algebra, it is the cartesian product of two tables followed by the selection operation. Thus, we can execute the product and selection process on two tables using a single join statement. We can use either 'on' or 'using' clause in MySQL to apply predicates to the join queries.

A Join can be broadly divided into two types:

1. **Inner Join**
2. **Outer Join**

Inner Join

Inner Join is a join that can be used to return all the values that have matching values in both the tables. Inner Join can be depicted using the below diagram.

The inner join can be further divided into the following types:

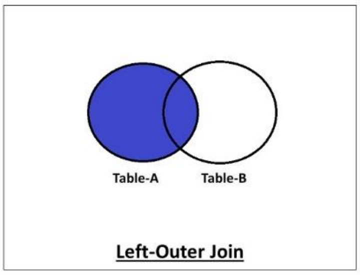
1. **Equi Join**
2. **Natural Join**

Outer Join

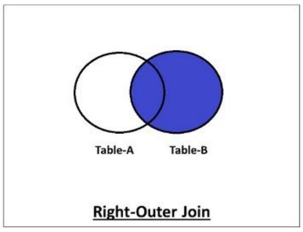
Outer Join is a join that can be used to return the records in both the tables whether it has matching records in both the tables or not.

The outer join can be further divided into three types:

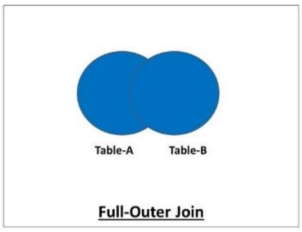
1. **Left-Outer Join**
2. **Right-Outer Join**
3. **Full-Outer Join**



Left-Outer Join



Right-Outer Join



Full-Outer Join

click for notes

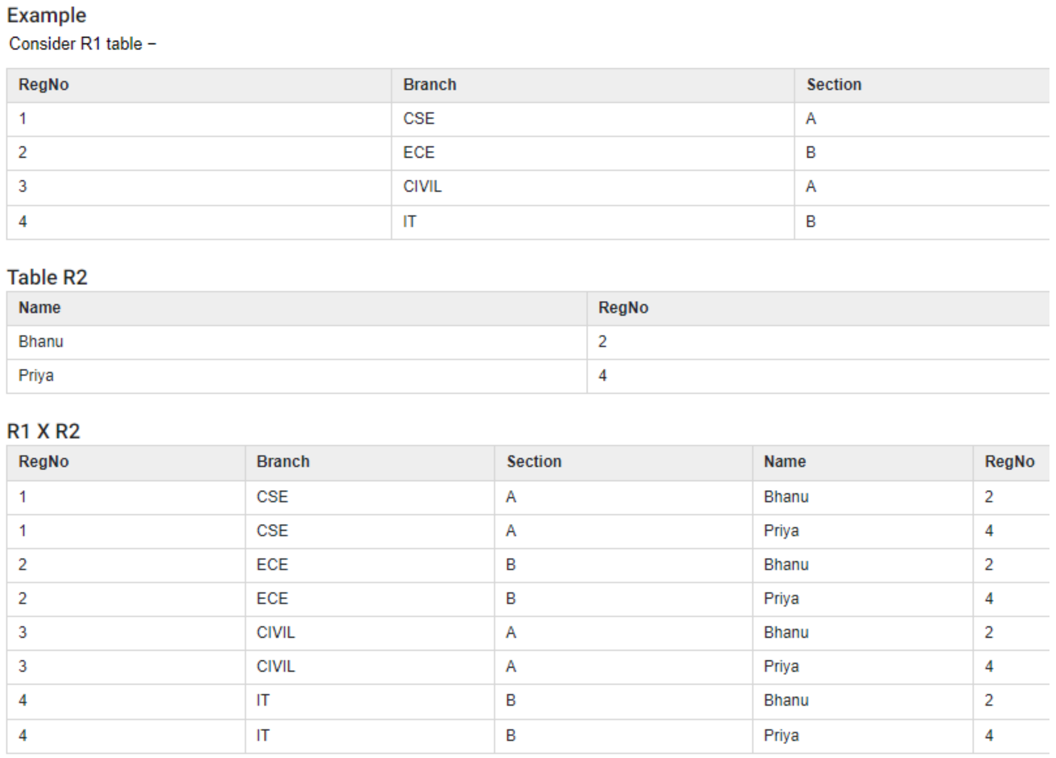
6. Explain Cartesian Product in DBMS with example.Explain Select and Project.

Cartesian product operation

It combines R1 and R2 without any condition. It is denoted by X.

Degree of R1 XR2 = degree of R1 + degree of R2

{degree = total no of columns}



Example

Consider R1 table −

RegNo	Branch	Section
1	CSE	A
2	ECE	B
3	CIVIL	A
4	IT	B

Table R2

Name	RegNo
Bhanu	2
Priya	4

R1 X R2

RegNo	Branch	Section	Name	RegNo
1	CSE	A	Bhanu	2
1	CSE	A	Priya	4
2	ECE	B	Bhanu	2
2	ECE	B	Priya	4
3	CIVIL	A	Bhanu	2
3	CIVIL	A	Priya	4
4	IT	B	Bhanu	2
4	IT	B	Priya	4

Select operation

It displays the records that satisfy a condition. It is denoted by sigma (σ) and is a horizontal subset of the original relation.

Syntax

Its syntax is as follows −

$\sigma_{condition}$(table name)

Projection operation

It displays the specific column of a table. It is denoted by pie (∏). It is a vertical subset of the original relation. It eliminates duplicate tuples.

Syntax

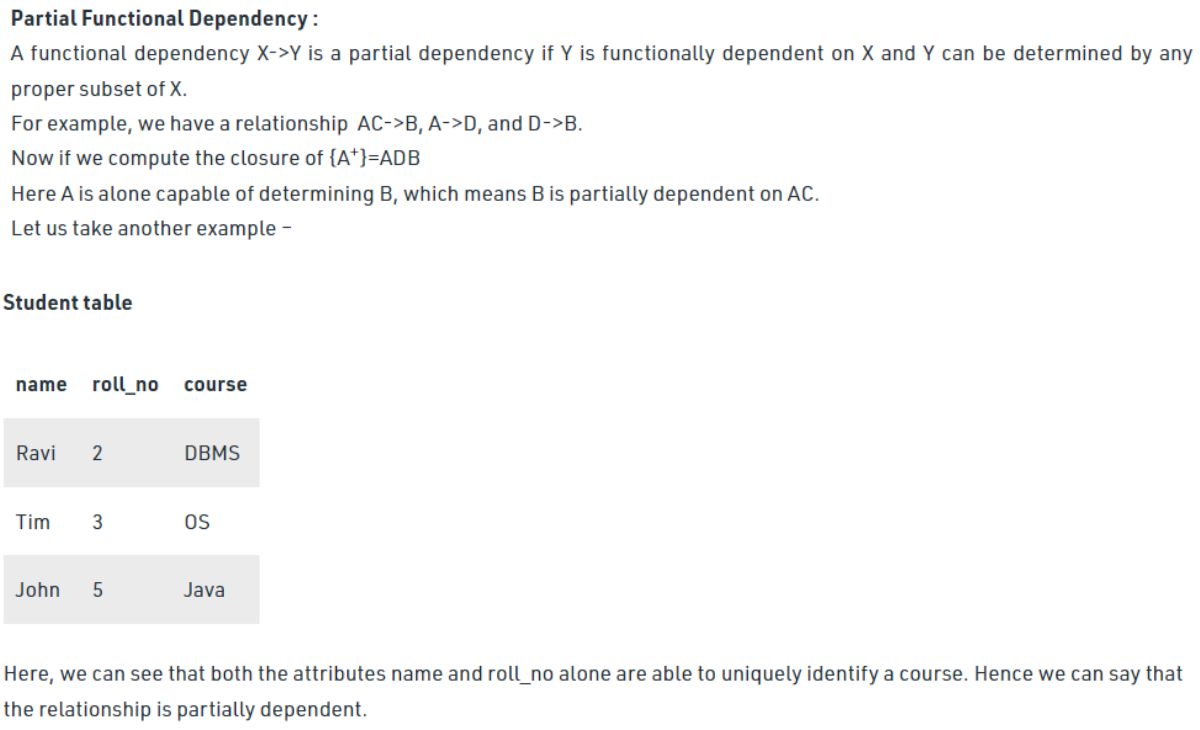
The syntax is as follows −

$\prod_{regno}$(student)

[click for notes]

Unit 3

1. What is partial functional dependency?



Partial Functional Dependency :

A functional dependency X->Y is a partial dependency if Y is functionally dependent on X and Y can be determined by any proper subset of X.

For example, we have a relationship AC->B, A->D, and D->B.

Now if we compute the closure of $\{A^+\}$=ADB

Here A is alone capable of determining B, which means B is partially dependent on AC.

Let us take another example –

Student table

name	roll_no	course
Ravi	2	DBMS
Tim	3	OS
John	5	Java

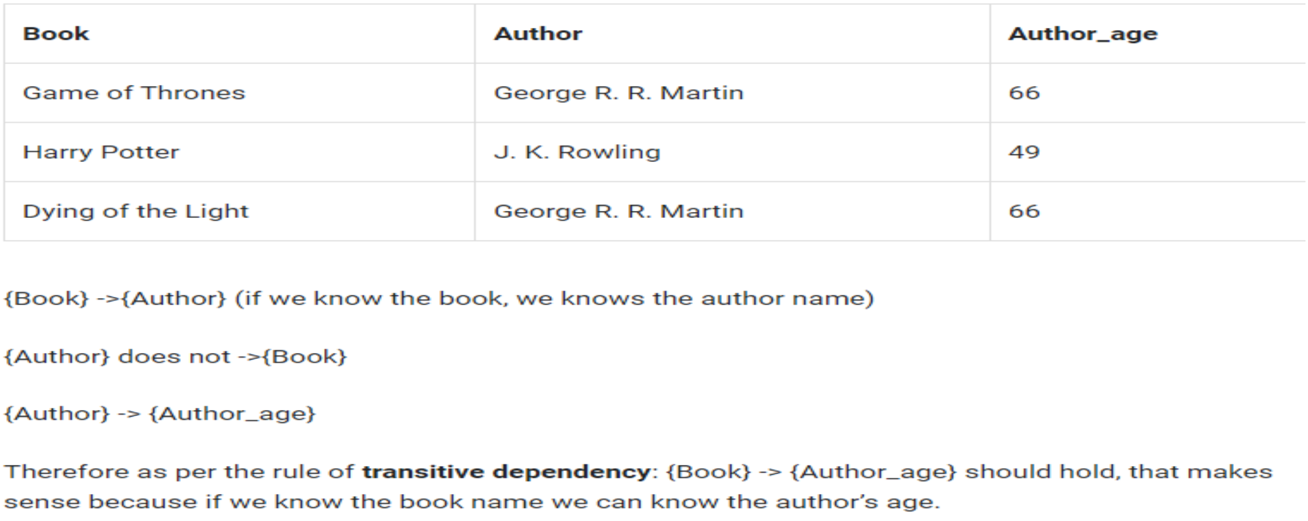
Here, we can see that both the attributes name and roll_no alone are able to uniquely identify a course. Hence we can say that the relationship is partially dependent.

2 What is transitive functional dependency?

A functional dependency is said to be transitive if it is indirectly formed by two functional dependencies. For e.g.

X -> Z is a transitive dependency if the following three functional dependencies hold true:

- X->Y
- Y does not ->X
- Y->Z



Book	Author	Author_age
Game of Thrones	George R. R. Martin	66
Harry Potter	J. K. Rowling	49
Dying of the Light	George R. R. Martin	66

{Book} ->{Author} (if we know the book, we knows the author name)

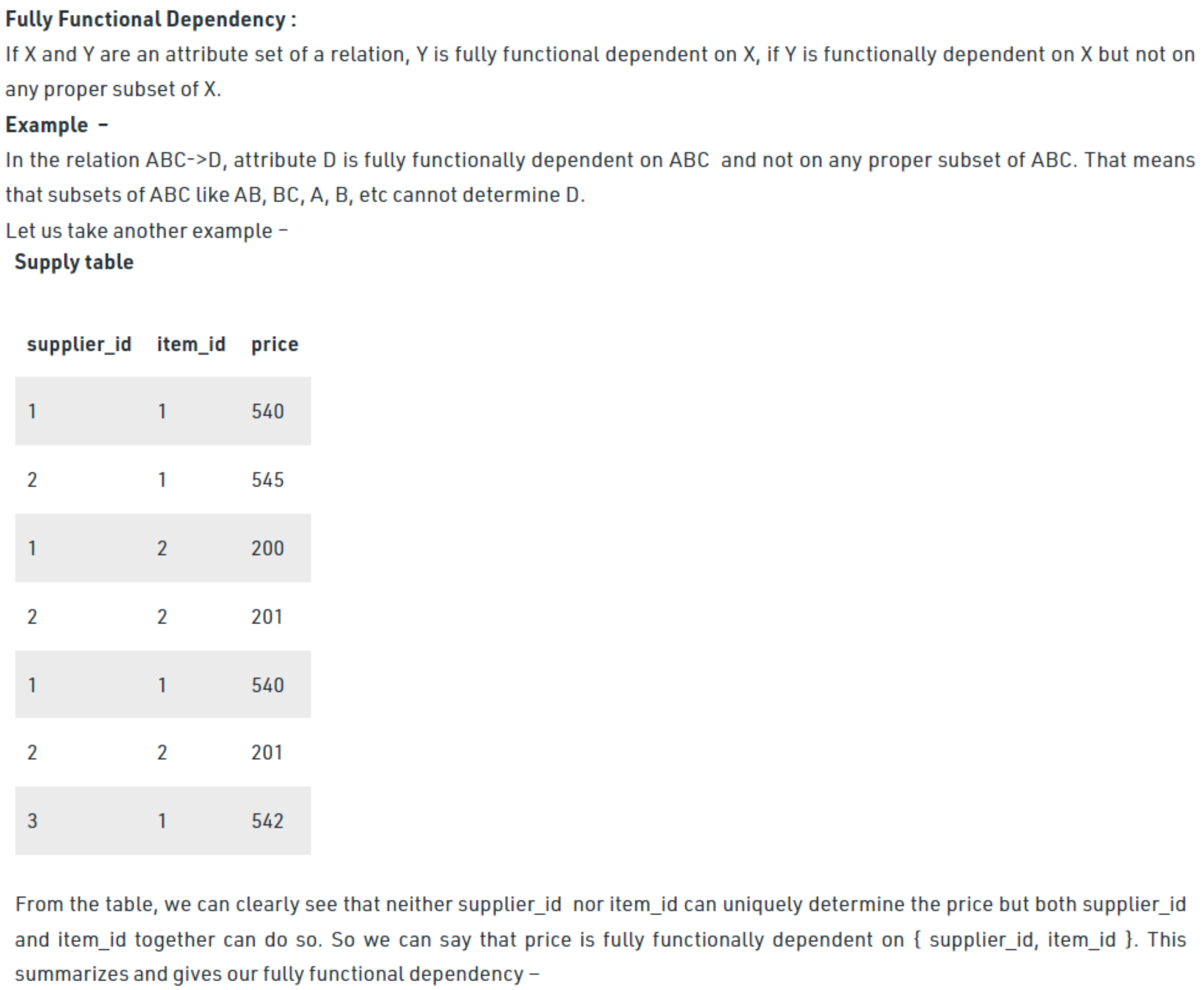
{Author} does not ->{Book}

{Author} -> {Author_age}

Therefore as per the rule of **transitive dependency**: {Book} -> {Author_age} should hold, that makes sense because if we know the book name we can know the author's age.

[click for notes]

3. What is full functional dependency?



Fully Functional Dependency :

If X and Y are an attribute set of a relation, Y is fully functional dependent on X, if Y is functionally dependent on X but not on any proper subset of X.

Example –

In the relation ABC->D, attribute D is fully functionally dependent on ABC and not on any proper subset of ABC. That means that subsets of ABC like AB, BC, A, B, etc cannot determine D.

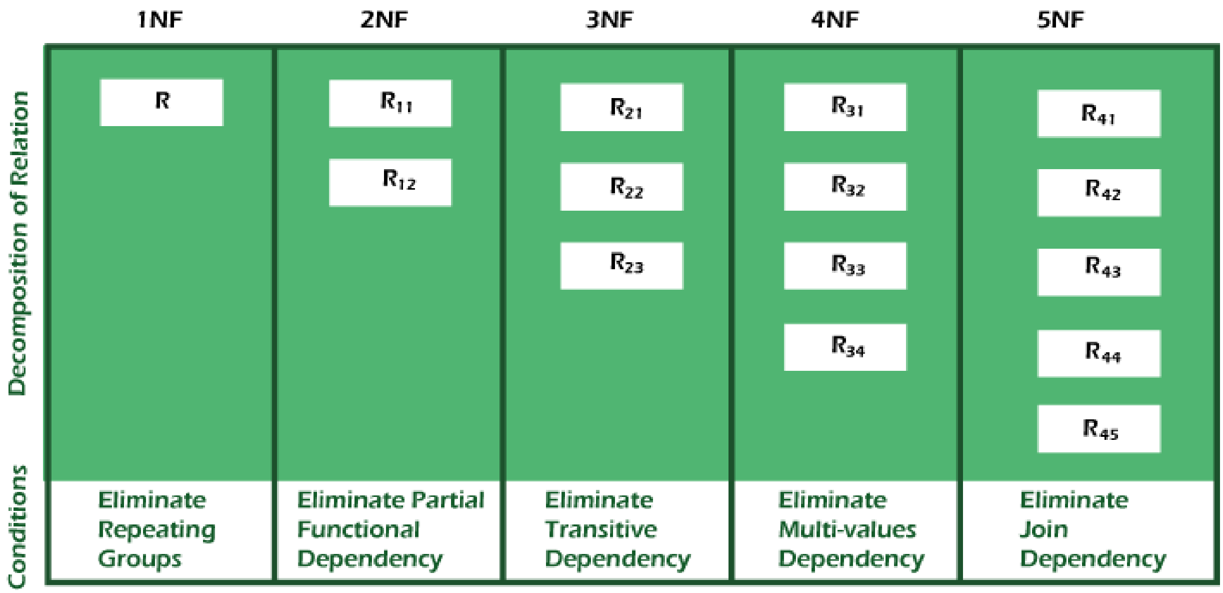
Let us take another example –

Supply table

supplier_id	item_id	price
1	1	540
2	1	545
1	2	200
2	2	201
1	1	540
2	2	201
3	1	542

From the table, we can clearly see that neither supplier_id nor item_id can uniquely determine the price but both supplier_id and item_id together can do so. So we can say that price is fully functionally dependent on { supplier_id, item_id }. This summarizes and gives our fully functional dependency –

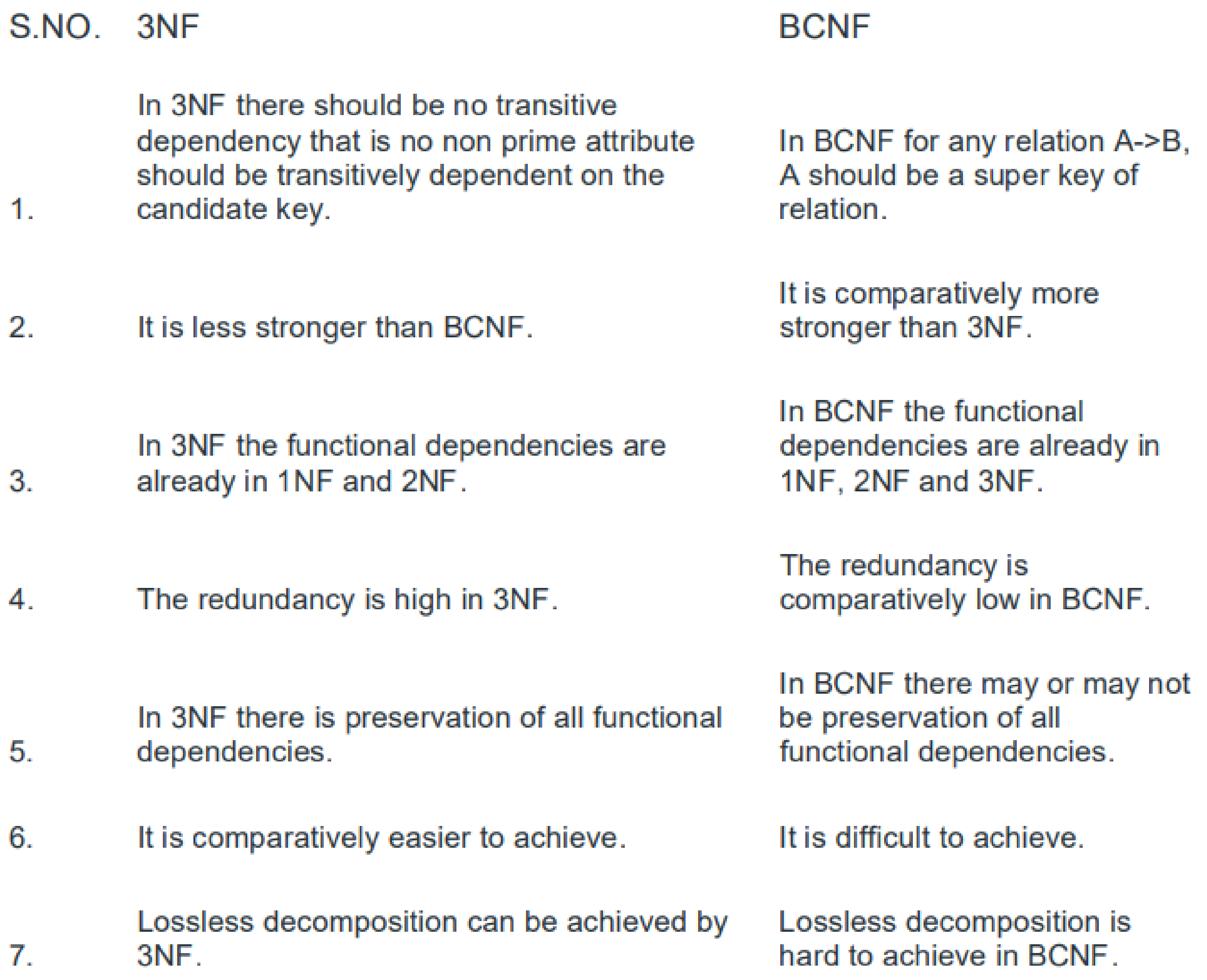
4 Explain 1NF, 2NF, 3NF, BCNF, 4NF with Example.


1NF
2NF
3NF
4NF
5NF
Decomposition of Relation
Conditions
R
R_{11}
R_{12}
R_{21}
R_{22}
R_{23}
R_{31}
R_{32}
R_{33}
R_{34}
R_{41}
R_{42}
R_{43}
R_{44}
R_{45}
Eliminate Repeating Groups
Eliminate Partial Functional Dependency
Eliminate Transitive Dependency
Eliminate Multi-values Dependency
Eliminate Join Dependency

click for notes

Normal Form	Description
1NF	A relation is in 1NF if it contains an atomic value.
2NF	A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key.
3NF	A relation will be in 3NF if it is in 2NF and no transition dependency exists.
BCNF	A stronger definition of 3NF is known as Boyce Codd's normal form.
4NF	A relation will be in 4NF if it is in Boyce Codd's normal form and has no multi-valued dependency.
5NF	A relation is in 5NF. If it is in 4NF and does not contain any join dependency, joining should be lossless.

5. What is the difference between 3NF and BCNF.



S.NO.	3NF	BCNF
1.	In 3NF there should be no transitive dependency that is no non prime attribute should be transitively dependent on the candidate key.	In BCNF for any relation A->B, A should be a super key of relation.
2.	It is less stronger than BCNF.	It is comparatively more stronger than 3NF.
3.	In 3NF the functional dependencies are already in 1NF and 2NF.	In BCNF the functional dependencies are already in 1NF, 2NF and 3NF.
4.	The redundancy is high in 3NF.	The redundancy is comparatively low in BCNF.
5.	In 3NF there is preservation of all functional dependencies.	In BCNF there may or may not be preservation of all functional dependencies.
6.	It is comparatively easier to achieve.	It is difficult to achieve.
7.	Lossless decomposition can be achieved by 3NF.	Lossless decomposition is hard to achieve in BCNF.

click for notes

6.
   R(C, S, Z)
   CS→Z
   Z→C
   Find the list of candidate key.

   video link

7. Find Normal Form

Emp(ename, did, empid,

salary)

ename, did->salary

   did, empid->

   salaryename ->

   empid empid-

   >ename

   video link

8. Find the Normal Form

   R(A,B,C,D,E)

   A->BC

   BC->E

   E->DA

video link

click for notes

Unit 4

1. What is concurrent schedule?

Concurrent Schedules:

A schedule is said to be concurrent in case the instructions of the transactions get executed preemptively.

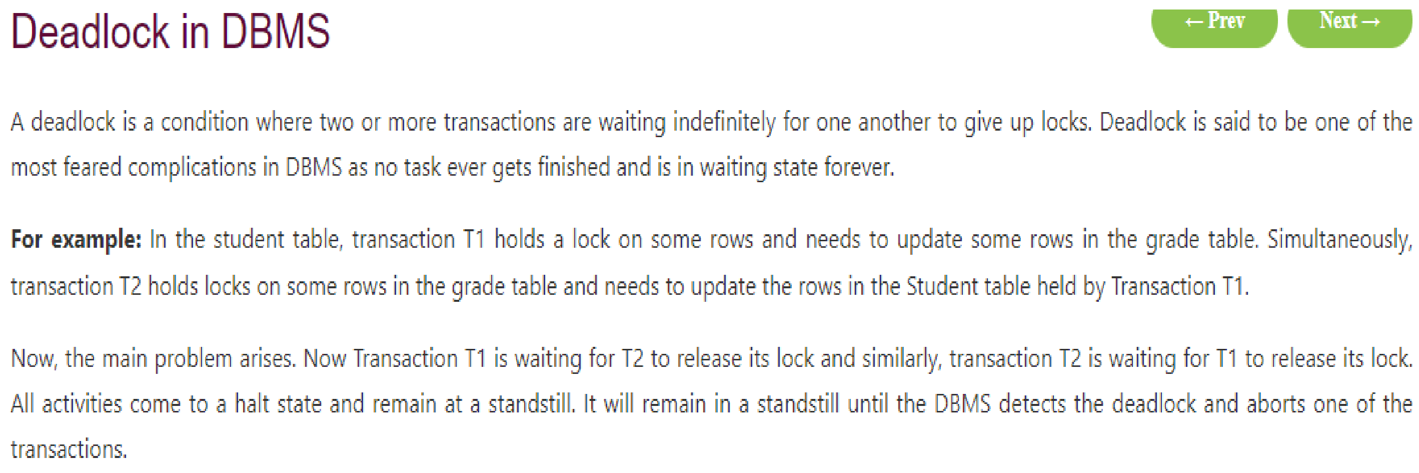
When the database system executes several transactions concurrently, the corresponding schedule no longer needs to be serial.

If two transactions are running concurrently, the operating system may execute one transaction for a little while, then perform a context switch, execute the second transaction for some time, and then switch back to the first transaction for some time, and so on. With multiple transactions, the CPU time is shared among all the transactions.

Several execution sequences are possible, since the various instructions from both transactions may now be interleaved.

In general, it is not possible to predict exactly how many instructions of a transaction will be executed before the CPU switches to another transaction.

2. What is deadlock?

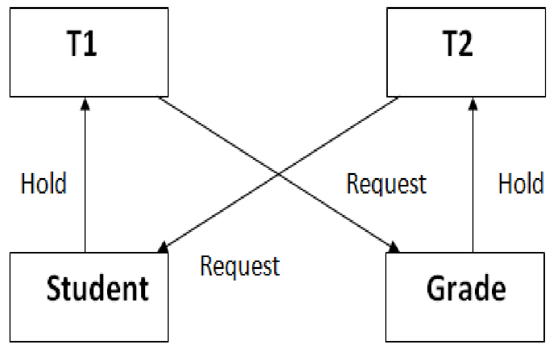


Deadlock in DBMS

A deadlock is a condition where two or more transactions are waiting indefinitely for one another to give up locks. Deadlock is said to be one of the most feared complications in DBMS as no task ever gets finished and is in waiting state forever.

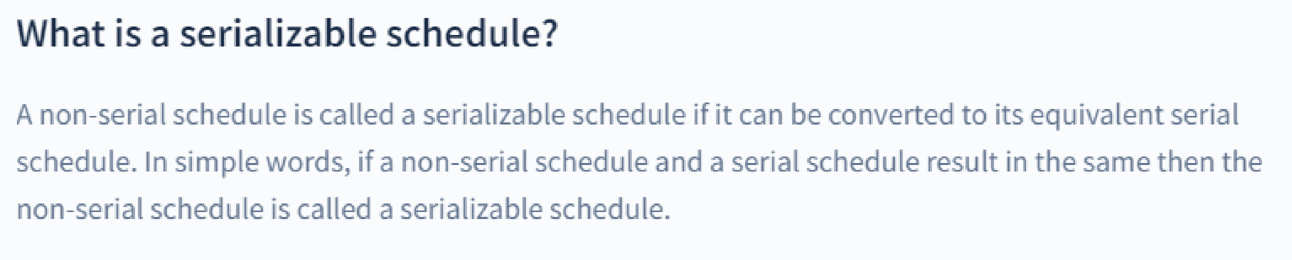
For example: In the student table, transaction T1 holds a lock on some rows and needs to update some rows in the grade table. Simultaneously, transaction T2 holds locks on some rows in the grade table and needs to update the rows in the Student table held by Transaction T1.

Now, the main problem arises. Now Transaction T1 is waiting for T2 to release its lock and similarly, transaction T2 is waiting for T1 to release its lock. All activities come to a halt state and remain at a standstill. It will remain in a standstill until the DBMS detects the deadlock and aborts one of the transactions.



click for notes

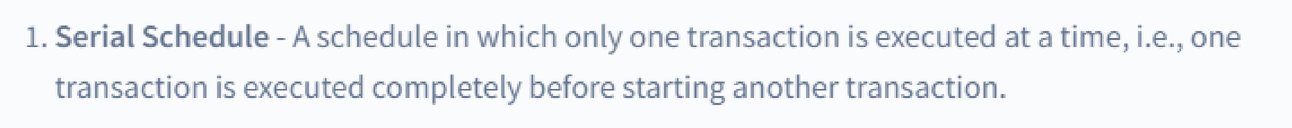
3. What do you mean by serializable schedule.



What is a serializable schedule?

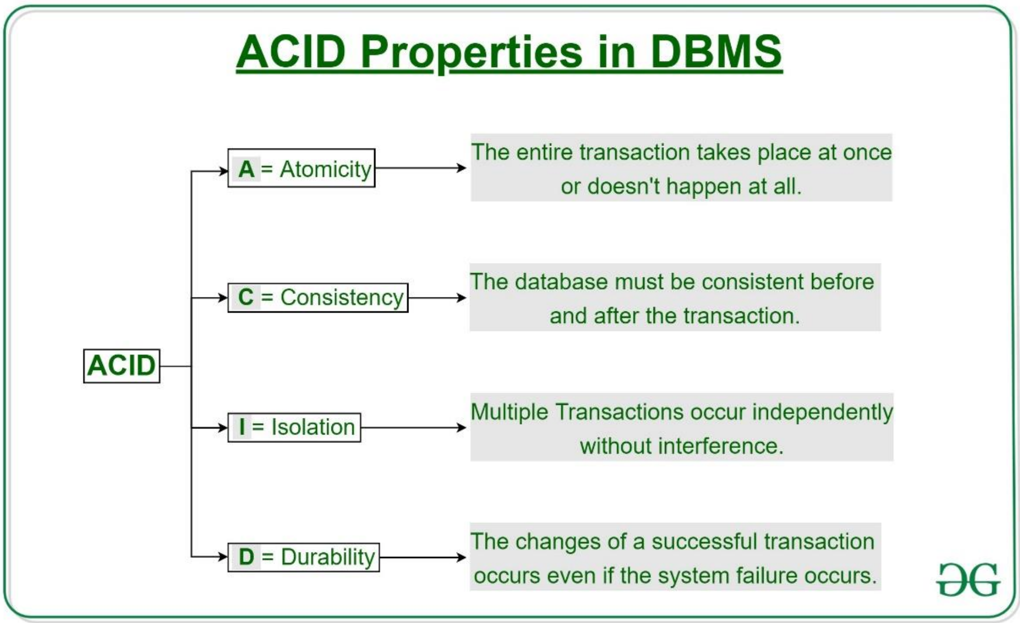
A non-serial schedule is called a serializable schedule if it can be converted to its equivalent serial schedule. In simple words, if a non-serial schedule and a serial schedule result in the same then the non-serial schedule is called a serializable schedule.

4. What is serial schedule?



1. **Serial Schedule** - A schedule in which only one transaction is executed at a time, i.e., one transaction is executed completely before starting another transaction.

5. What do you mean by ACID property?



ACID Properties in DBMS

ACID

A = Atomicity — The entire transaction takes place at once or doesn't happen at all.

C = Consistency — The database must be consistent before and after the transaction.

I = Isolation — Multiple Transactions occur independently without interference.

D = Durability — The changes of a successful transaction occurs even if the system failure occurs.

6. check the given schedule is serializable or not.
Consider the following schedule for transaction T1, T2, T3

T1	T2	T3
r(x)		
	r(y)	
		r(y)
	w(y)	
w(x)		
		w(x)
	r(x)	

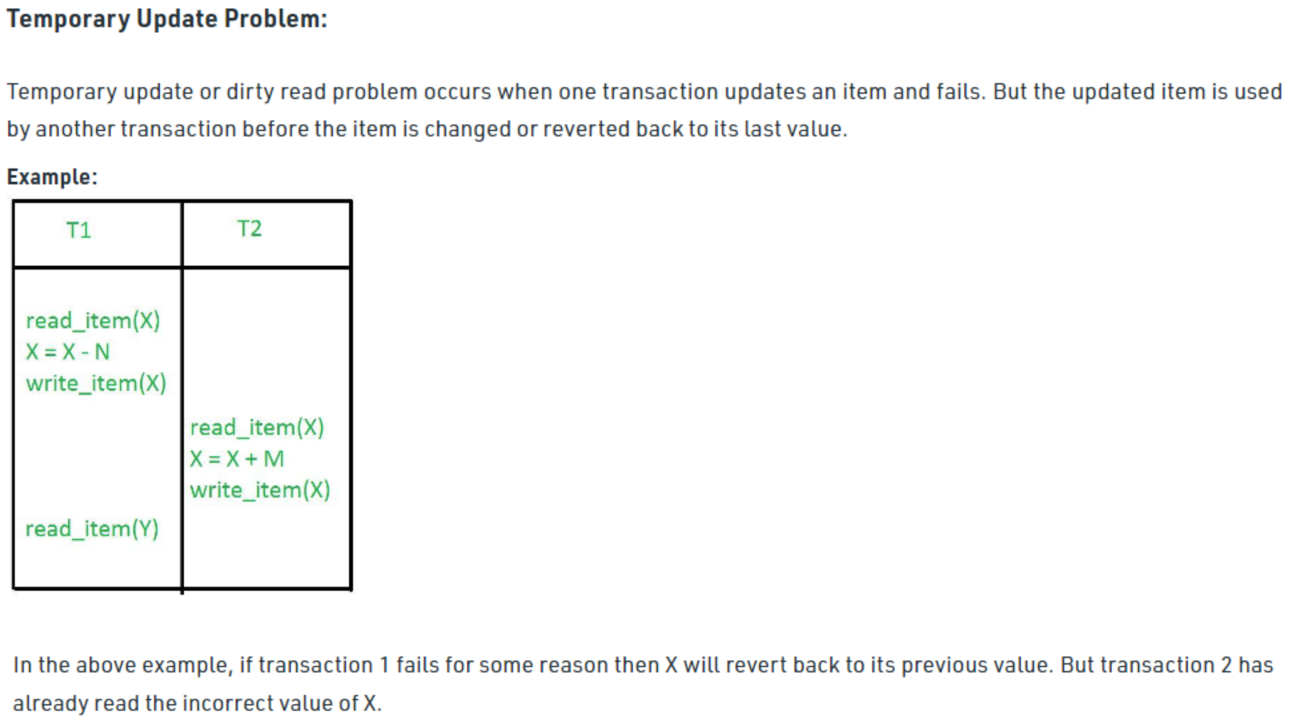
click for notes

	w(x)	

video link

Unit 5

1. What is temporary update problem?



Temporary Update Problem:

Temporary update or dirty read problem occurs when one transaction updates an item and fails. But the updated item is used by another transaction before the item is changed or reverted back to its last value.

Example:

T1	T2

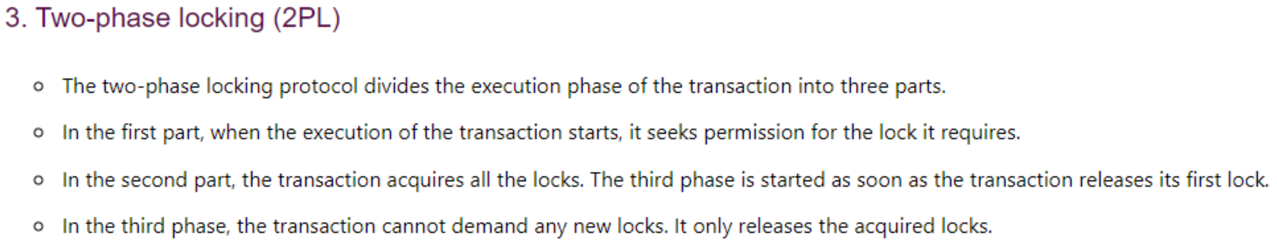
read_item(X)
X = X - N
write_item(X)	

	read_item(X)
	X = X + M
	write_item(X)

read_item(Y)	

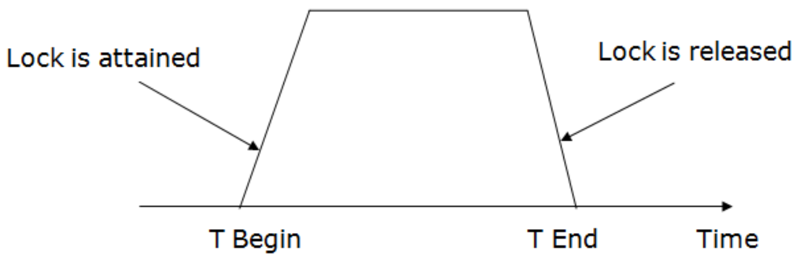
In the above example, if transaction 1 fails for some reason then X will revert back to its previous value. But transaction 2 has already read the incorrect value of X.

2. Explain Two Phase Locking Protocol .



3. Two-phase locking (2PL)

- The two-phase locking protocol divides the execution phase of the transaction into three parts.
- In the first part, when the execution of the transaction starts, it seeks permission for the lock it requires.
- In the second part, the transaction acquires all the locks. The third phase is started as soon as the transaction releases its first lock.
- In the third phase, the transaction cannot demand any new locks. It only releases the acquired locks.



There are two phases of 2PL:

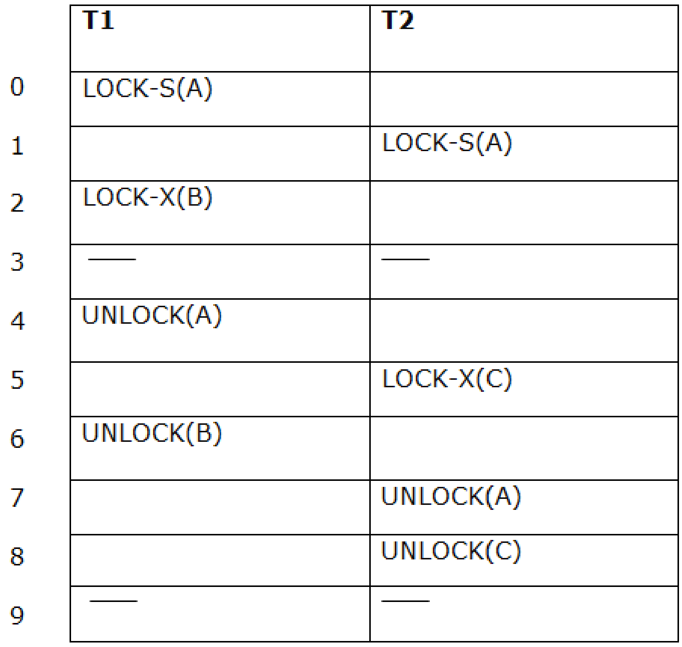
Growing phase: In the growing phase, a new lock on the data item may be acquired by the transaction, but none can be released.

Shrinking phase: In the shrinking phase, existing lock held by the transaction may be released, but no new locks can be acquired.

In the below example, if lock conversion is allowed then the following phase can happen:

1. Upgrading of lock (from S(a) to X (a)) is allowed in growing phase.
2. Downgrading of lock (from X(a) to S(a)) must be done in shrinking phase.

Example:



	T1	T2
0	LOCK-S(A)	
1		LOCK-S(A)
2	LOCK-X(B)	
3	——	——
4	UNLOCK(A)	
5		LOCK-X(C)
6	UNLOCK(B)	
7		UNLOCK(A)
8		UNLOCK(C)
9	——	——

The following way shows how unlocking and locking work with 2-PL.

Transaction T1:

- **Growing phase:** from step 1-3
- **Shrinking phase:** from step 5-7
- **Lock point:** at 3

Transaction T2:

- **Growing phase:** from step 2-6
- **Shrinking phase:** from step 8-9
- **Lock point:** at 6

click for notes

3. What is the objective to do concurrency control?

Concurrency control is provided in a database to:

- (i) enforce isolation among transactions.
- (ii) preserve database consistency through consistency preserving execution of transactions.
- (iii) resolve read-write and write-read conflicts.

Various concurrency control techniques are:

1. Two-phase locking Protocol
2. Time stamp ordering Protocol
3. Multi version concurrency control
4. Validation concurrency control

4. Explain Dirty read problem.

Dirty Read

Dirty read is a read of uncommitted data. If a particular row is modified by another running application and not yet committed, we also run an application to read the same row with the same uncommitted data. This is the state we say it as a dirty read.

The one main thing is that the dirty reader has to stop reading dirty.

We can try to use the shared locks to prevent other transactions to modify the row, if one is carried out here.

Example of dirty read problem

Example 1

- **Step 1** − Consider we have an online shopping system where users can buy and view the buyer products at the same time.
- **Step 2** − Let us suppose a case in which a user tries to buy a product , and as soon as the user buys the product then the count value in update stock will change immediately.
- **Step 3** − Let us consider that there were 10 items in stock, but now they are 9.
- **Step 4** − Moreover due to this transaction, there will also be communication with the billing gateway.
- **Step 5** − Meanwhile, if there is any other user who has also done a transaction at the same time, the new user will be able to see 9 items in the stock.
- **Step 6** − But, let us suppose that the first user was unable to complete his/her transactions due to some error or insufficient funds.
- **Step 7** − Then, in this case the transaction done by the first user will roll back and now the value in stock will be 10 again.
- **Step 8** − But, when the 2nd user was doing a transaction the no items in stocks were 9.
- **Step 9** − This is called DIRTY DATA and this whole problem is called the Dirty Problem.

click for notes

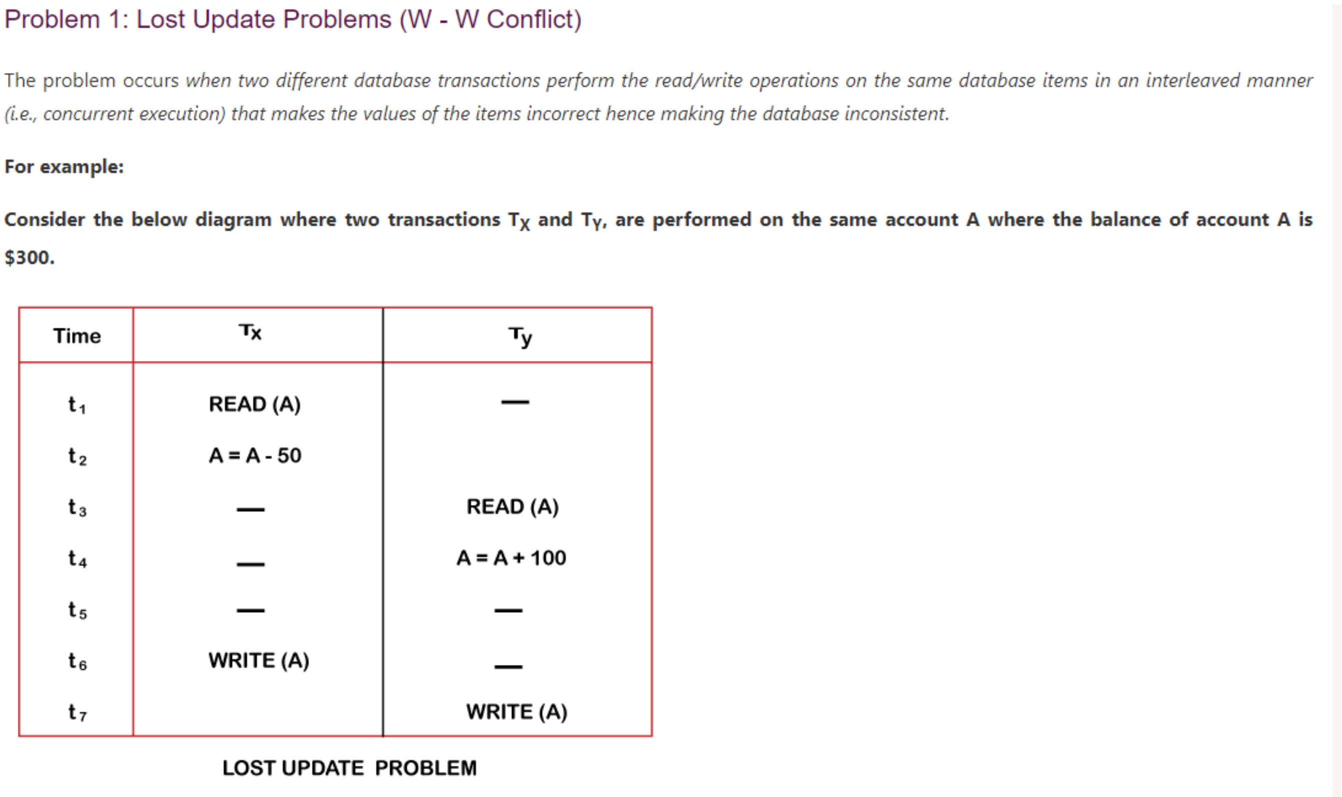
5. What is concurrency control?

What is concurrency control in DBMS

A transaction is a single reasonable unit of work that can retrieve or may change the data of a database. Executing each transaction individually increases the *waiting time* for the other transactions and the overall execution also gets delayed. Hence, to increase the throughput and to reduce the waiting time, transactions are executed concurrently.

Example: Suppose, between two railway stations, **A** and **B**, 5 trains have to travel, if all the trains are set in a row and only one train is allowed to move from station **A** to **B** and others have to wait for the first train to reach its destination then it will take a lot of time for all the trains to travel from station **A** to **B**. To reduce time all the trains should be allowed to move concurrently from station **A** to **B** ensuring no risk of collision between them.

6. What is Lost Update problem?



Problem 1: Lost Update Problems (W - W Conflict)

The problem occurs *when two different database transactions perform the read/write operations on the same database items in an interleaved manner (i.e., concurrent execution) that makes the values of the items incorrect hence making the database inconsistent.*

For example:

Consider the below diagram where two transactions T_X and T_Y, are performed on the same account A where the balance of account A is $300.

Time	T_X	T_Y
t_1	READ (A)	—
t_2	A = A - 50	
t_3	—	READ (A)
t_4	—	A = A + 100
t_5	—	—
t_6	WRITE (A)	—
t_7		WRITE (A)

LOST UPDATE PROBLEM

7. Explain validation-Based Protocol.

Validation Based Protocol is also called Optimistic Concurrency Control Technique. This protocol is used in DBMS (Database Management System) for avoiding concurrency in transactions. It is called optimistic because of the assumption it makes, i.e. very less interference occurs, therefore, there is no need for checking while the transaction is executed.

In this technique, no checking is done while the transaction is been executed. Until the transaction end is reached updates in the transaction are not applied directly to the database. All updates are applied to local copies of data items kept for the transaction. At the end of transaction execution, while execution of the transaction, a **validation phase** checks whether any of transaction updates violate serializability. If there is no violation of serializability the transaction is committed

and the database is updated; or else, the transaction is updated and then restarted.
Optimistic Concurrency Control is a three-phase protocol. The three phases for validation based protocol:

1. **Read Phase:**
   Values of committed data items from the database can be read by a transaction. Updates are only applied to local data versions.

2. **Validation Phase:**
   Checking is performed to make sure that there is no violation of serializability when the transaction updates are applied to the database.

3. **Write Phase:**
   On the success of the validation phase, the transaction updates are applied to the database, otherwise, the updates are discarded and the transaction is slowed down.

Unit 6

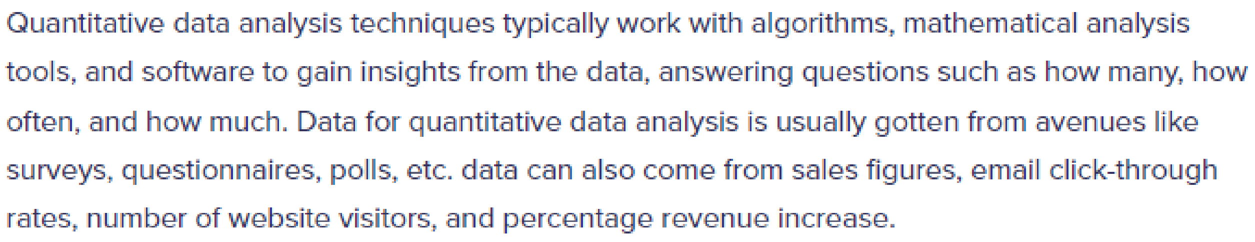
1. What do you mean by research?

Research is a process of systematic inquiry that entails collection of data; documentation of critical information; and analysis and interpretation of that data/information, in accordance with suitable methodologies set by specific professional fields and academic disciplines.

Research is conducted to...

- Evaluate the validity of a hypothesis or an interpretive framework.
- To assemble a body of substantive knowledge and findings for sharing them in appropriate manners.
- To help generate questions for further inquiries.

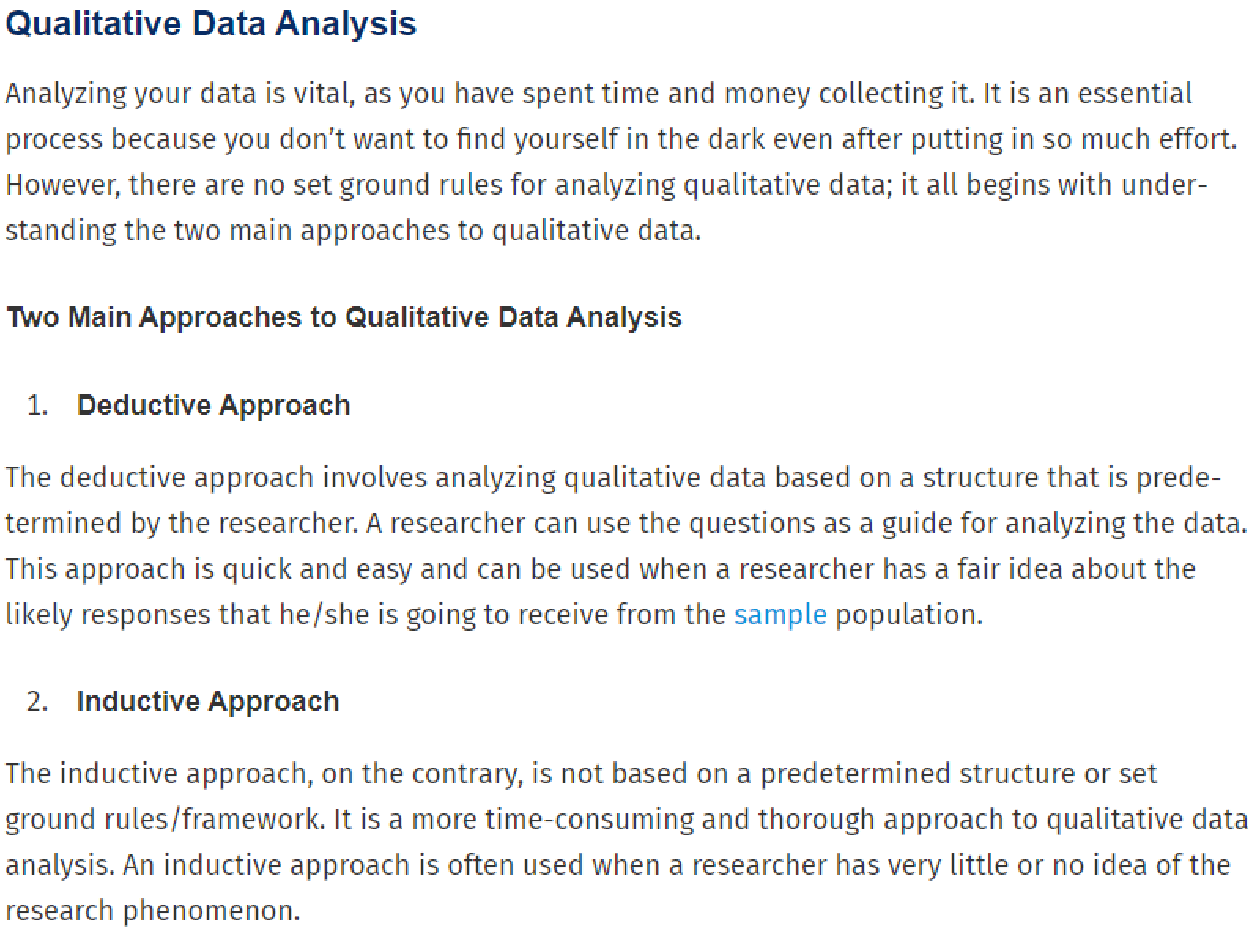
2. Explain Quantitative data Analysis



Quantitative data analysis techniques typically work with algorithms, mathematical analysis tools, and software to gain insights from the data, answering questions such as how many, how often, and how much. Data for quantitative data analysis is usually gotten from avenues like surveys, questionnaires, polls, etc. data can also come from sales figures, email click-through rates, number of website visitors, and percentage revenue increase.

3. Explain Qualitative data Analysis

click for notes



Qualitative Data Analysis

Analyzing your data is vital, as you have spent time and money collecting it. It is an essential process because you don't want to find yourself in the dark even after putting in so much effort. However, there are no set ground rules for analyzing qualitative data; it all begins with understanding the two main approaches to qualitative data.

Two Main Approaches to Qualitative Data Analysis

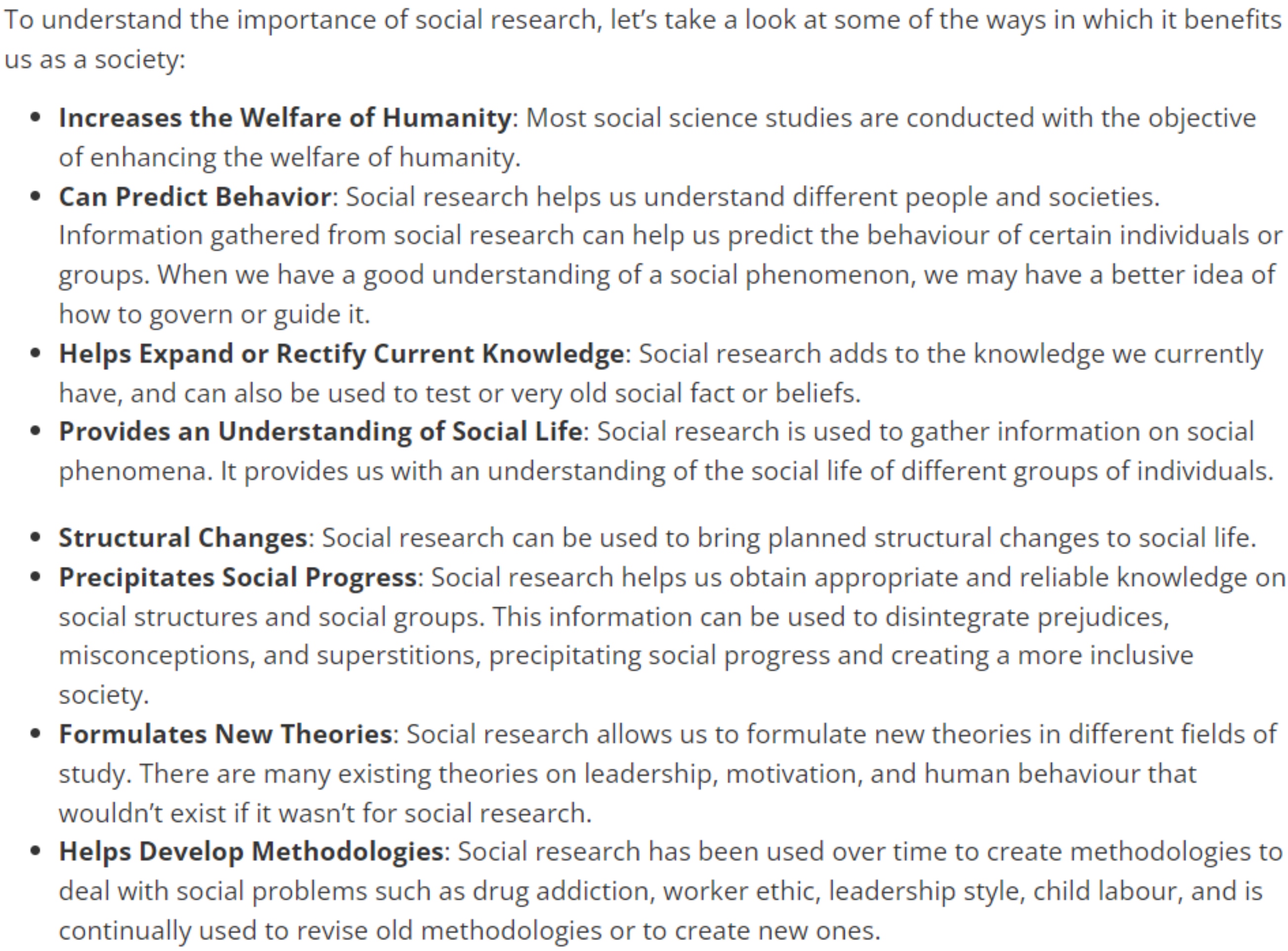
1. **Deductive Approach**

The deductive approach involves analyzing qualitative data based on a structure that is predetermined by the researcher. A researcher can use the questions as a guide for analyzing the data. This approach is quick and easy and can be used when a researcher has a fair idea about the likely responses that he/she is going to receive from the sample population.

2. **Inductive Approach**

The inductive approach, on the contrary, is not based on a predetermined structure or set ground rules/framework. It is a more time-consuming and thorough approach to qualitative data analysis. An inductive approach is often used when a researcher has very little or no idea of the research phenomenon.

4. Importance of research in social aspect.



To understand the importance of social research, let's take a look at some of the ways in which it benefits us as a society:

- **Increases the Welfare of Humanity**: Most social science studies are conducted with the objective of enhancing the welfare of humanity.
- **Can Predict Behavior**: Social research helps us understand different people and societies. Information gathered from social research can help us predict the behaviour of certain individuals or groups. When we have a good understanding of a social phenomenon, we may have a better idea of how to govern or guide it.
- **Helps Expand or Rectify Current Knowledge**: Social research adds to the knowledge we currently have, and can also be used to test or very old social fact or beliefs.
- **Provides an Understanding of Social Life**: Social research is used to gather information on social phenomena. It provides us with an understanding of the social life of different groups of individuals.

- **Structural Changes**: Social research can be used to bring planned structural changes to social life.
- **Precipitates Social Progress**: Social research helps us obtain appropriate and reliable knowledge on social structures and social groups. This information can be used to disintegrate prejudices, misconceptions, and superstitions, precipitating social progress and creating a more inclusive society.
- **Formulates New Theories**: Social research allows us to formulate new theories in different fields of study. There are many existing theories on leadership, motivation, and human behaviour that wouldn't exist if it wasn't for social research.
- **Helps Develop Methodologies**: Social research has been used over time to create methodologies to deal with social problems such as drug addiction, worker ethic, leadership style, child labour, and is continually used to revise old methodologies or to create new ones.

click for notes