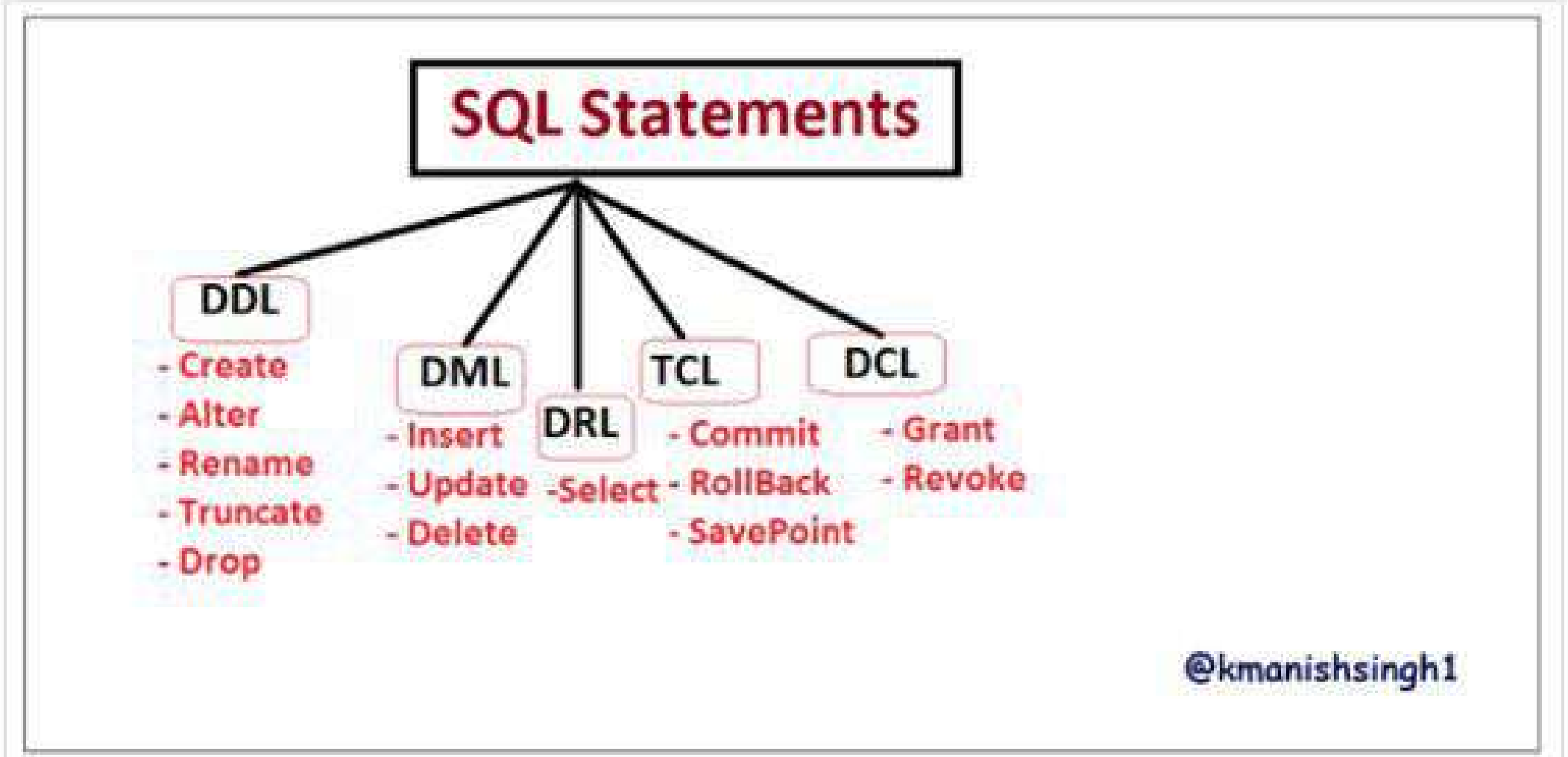
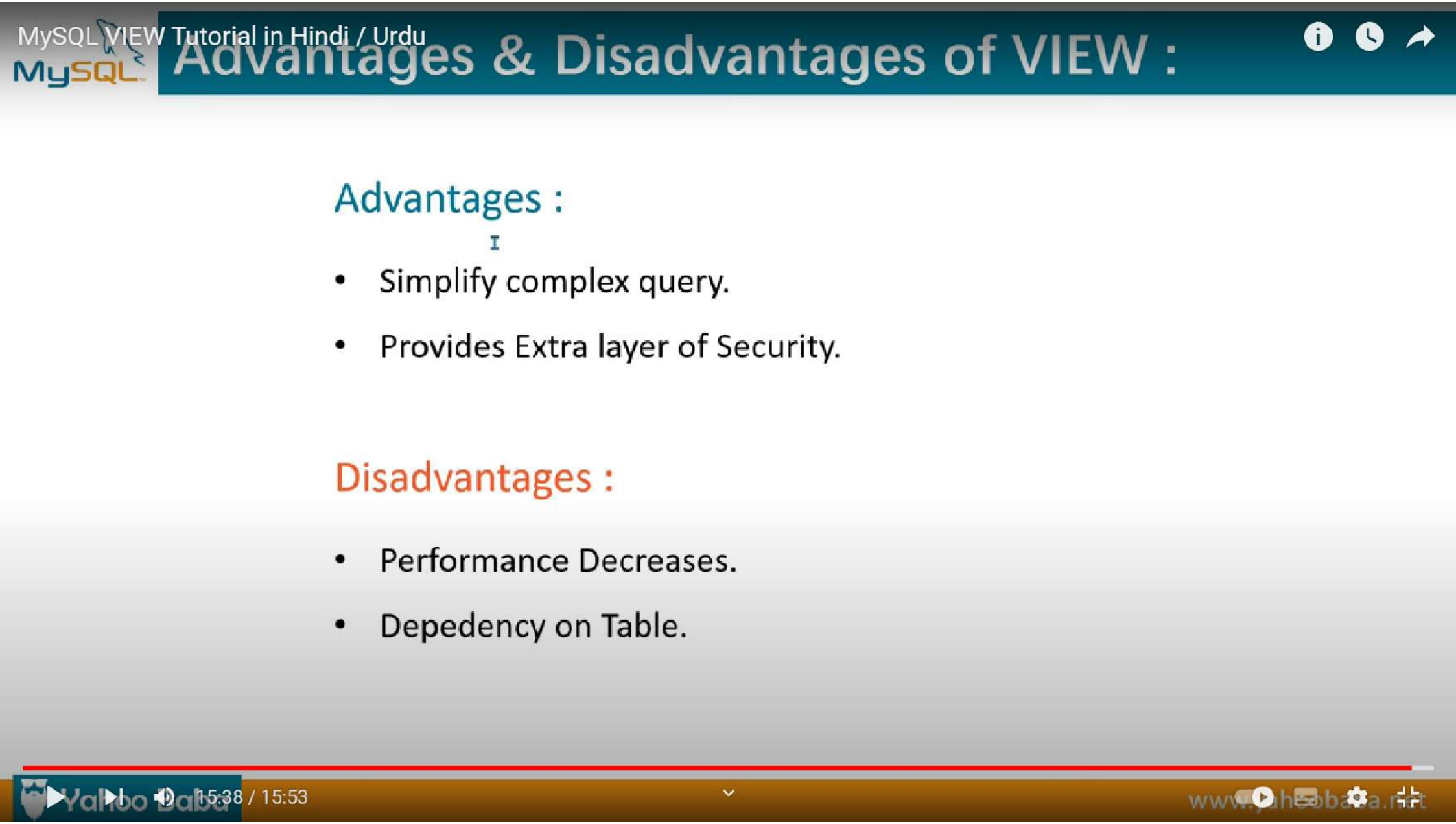

SQL Statements
DDL
- Create
- Alter
- Rename
- Truncate
- Drop
DML
- Insert
- Update
- Delete
DRL
-Select
TCL
- Commit
- RollBack
- SavePoint
DCL
- Grant
- Revoke
@kmanishsingh1


MySQL VIEW Tutorial in Hindi / Urdu
MySQL
Advantages & Disadvantages of VIEW :
Advantages :
Simplify complex query.
Provides Extra layer of Security.
Disadvantages :
Performance Decreases.
Depedency on Table.
15:38 / 15:53

Indexes

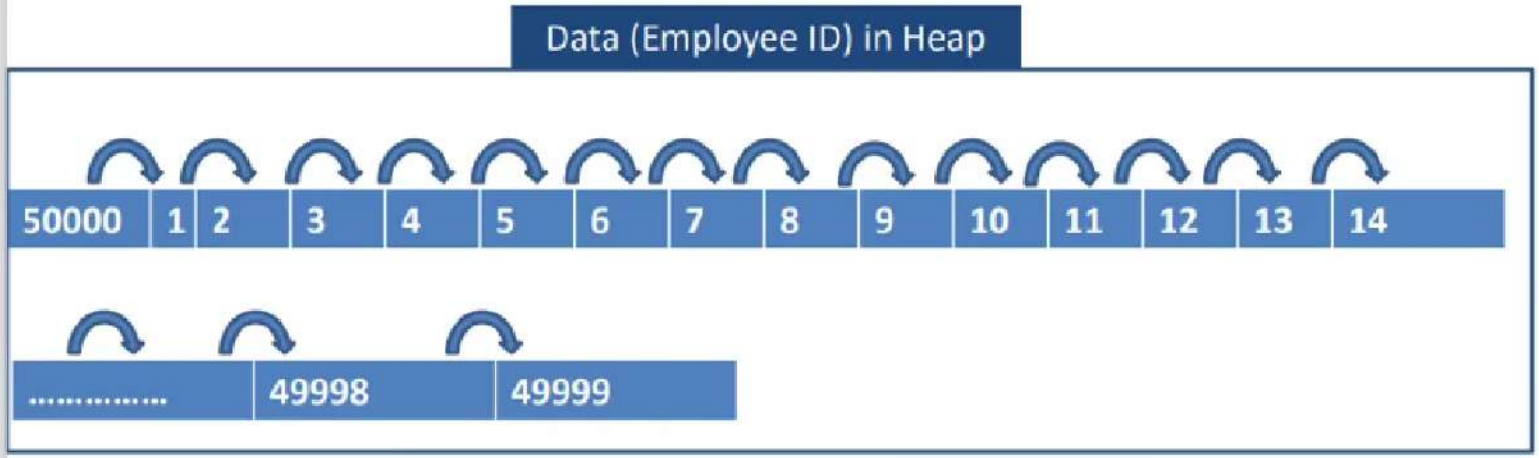
Indexes are special data structures associated with tables or views that help speed up the query.

➢ **What is Index?**

- An index is a pointer to data in a table.
- An index in a database is very similar to an index in the back of a book.
- An index helps to speed up SELECT queries and WHERE clauses.
- Indexes can be created or dropped with no effect on the data.

How does Indexes works?

- **Sequential Search**



Search 50000 in table ← one scan in heap

Search 49999 in table ← 499999 Scans in heap

Types of Indexes

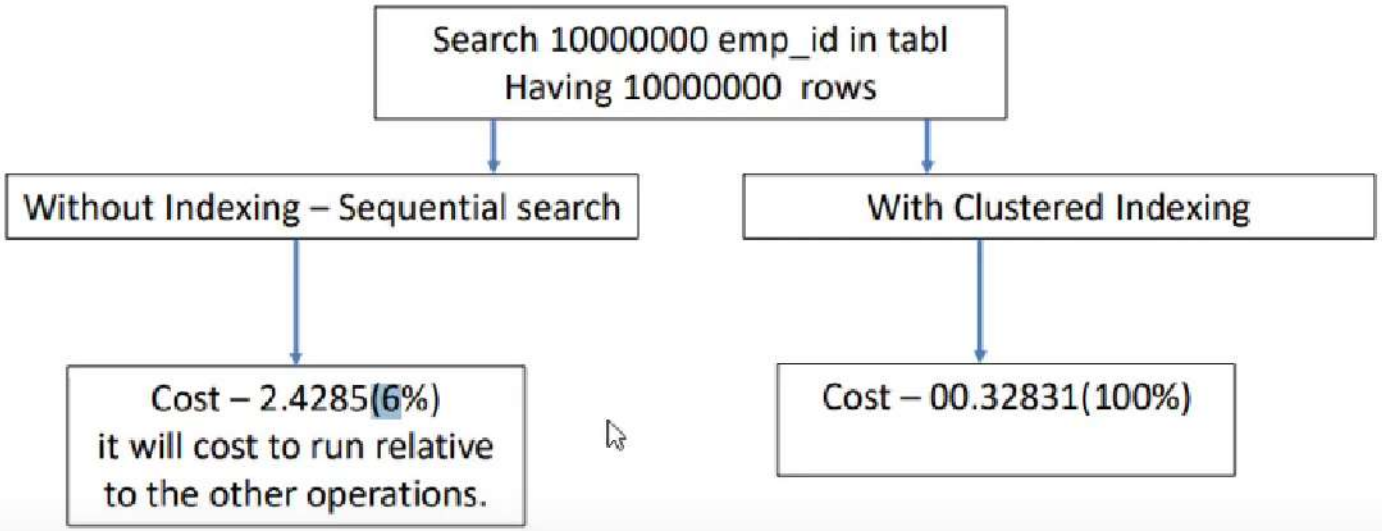
Clustered index

- Each table has **only one clustered index** because data rows can be only sorted in one order.
- A clustered index is a special index which physically orders the data according to the indexed columns.
- The leaf nodes of the index store the data for the rest of the columns in the table.

Non-clustered index

- A table may have one or more non clustered.
- A Non Clustered index is just like the index of a book.
- It points back to the actual page that contains the data. (In other words, it points back to the clustered index)

➢ Search Statistics



Non Clustered index

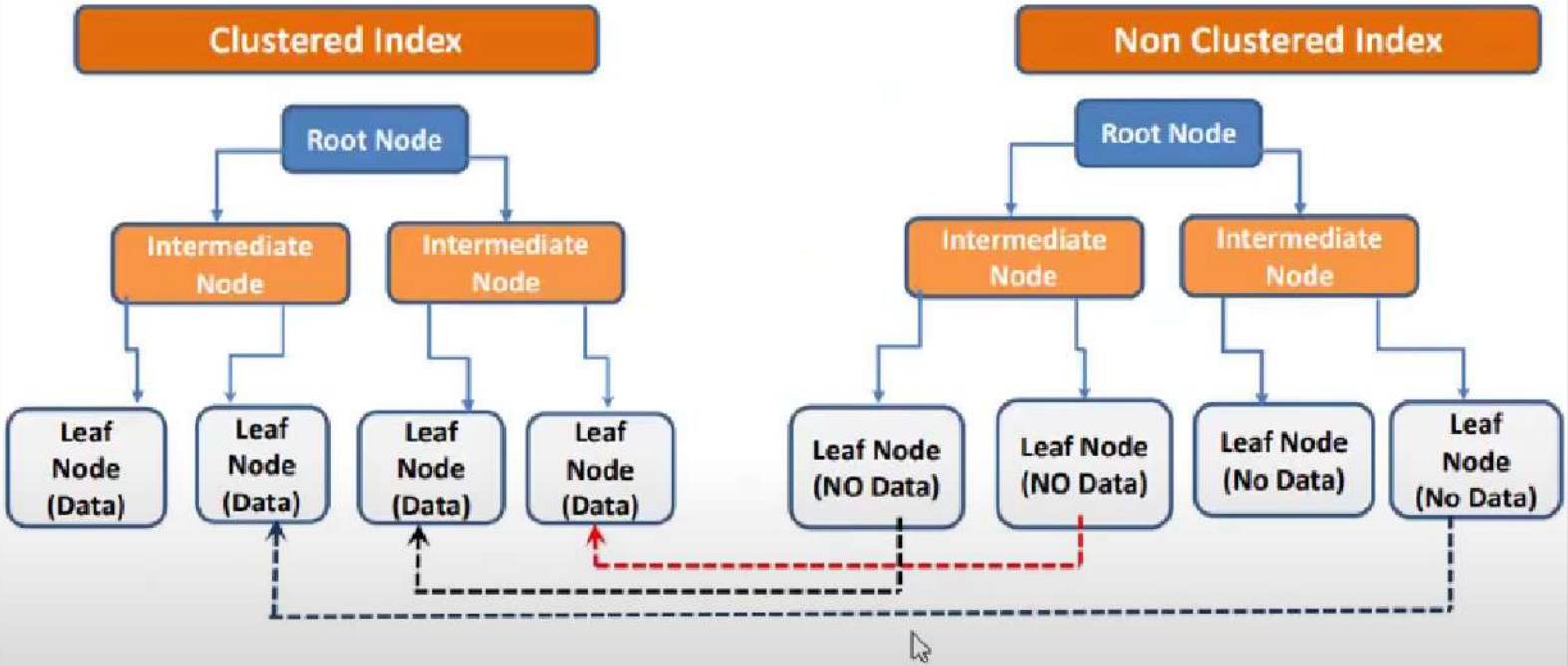
- A non-clustered index doesn't sort the physical data inside the table.
- In fact, a non-clustered index is stored at one place and table data is stored in another place.
- This is similar to a textbook where the book content is located in one place and the index is located in another.
- This allows for more than one non-clustered index per table.

SQL-24-Indexes.pdf

Sql Indexes in Hindi | Indexes in sql server | best practice sql index

File | C:/Users/admin/Documents/Tutorials_All/SQL-YouTube-Video/SQL-24-Indexes.pdf

Sign in



10:32 / 12:39

➢ SQL Server unique index

- A **unique index** ensures the index key columns do not contain any duplicate values.
- A unique index may consist of one or many columns.
- A unique index can be **clustered** or **non-clustered**.

Where to Apply Index

- Indexes are meant to speed up the performance of a database, so use indexing whenever it significantly improves the performance of your database.
- Check query and find reason for slow performance.
- Find column in query which is used frequently for searching.

Disadvantages of Indexing

- In case of update(change in indexed column) and delete a record, the database might need to move the entire row into a new position to keep the rows in sorted order.

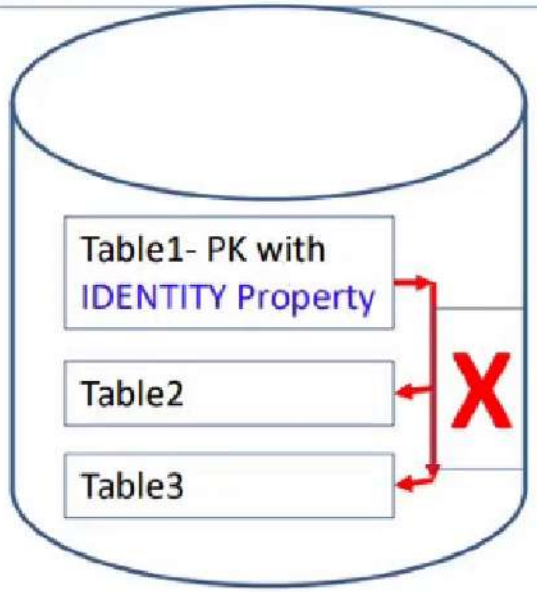
Guidelines of Index :

- Automatically creates the indexes for PRIMARY KEY and UNIQUE columns.
- Index columns that you frequently use to retrieve the data.
- Index columns that are used for joins to improve join performance.
- Avoid columns that contain too many NULL values.
- Small tables do not require indexes.

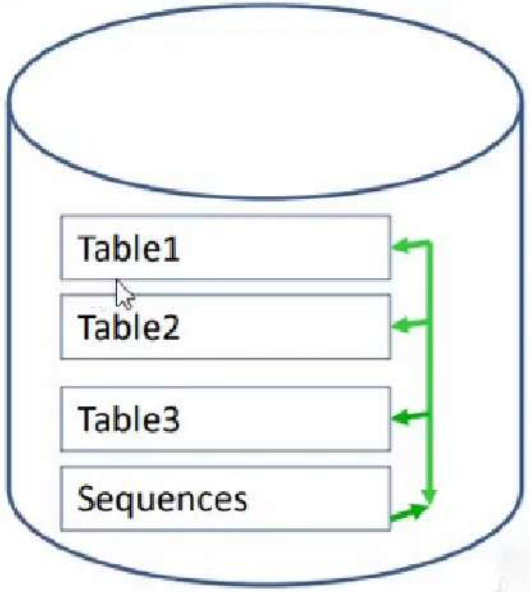
Difference between Identity & Sequence in SQL Server

Point-1:

The IDENTITY property is tied to a particular table and cannot be shared among multiple tables.



SEQUENCE object is defined by the user and can be shared by multiple tables since is it is not tied to any table.



Difference between Identity & Sequence in SQL Server

Point 2:

- To generate the next IDENTITY value, a new row has to be inserted into the table.
- On the other hand, the next VALUE for a SEQUENCE object can simply be generated using **the NEXT VALUE** FOR clause with the sequence object.

- **Identity Column**

```
Create Table test1
(
    ID          INT         IDENTITY(1,1),
    FName       Varchar(100)
)

    Insert Into Test(FName) Values('Smith')
```

- **Sequence**

```
INSERT into students VALUES( NEXT VALUE FOR sequence_1,'Smith');
```

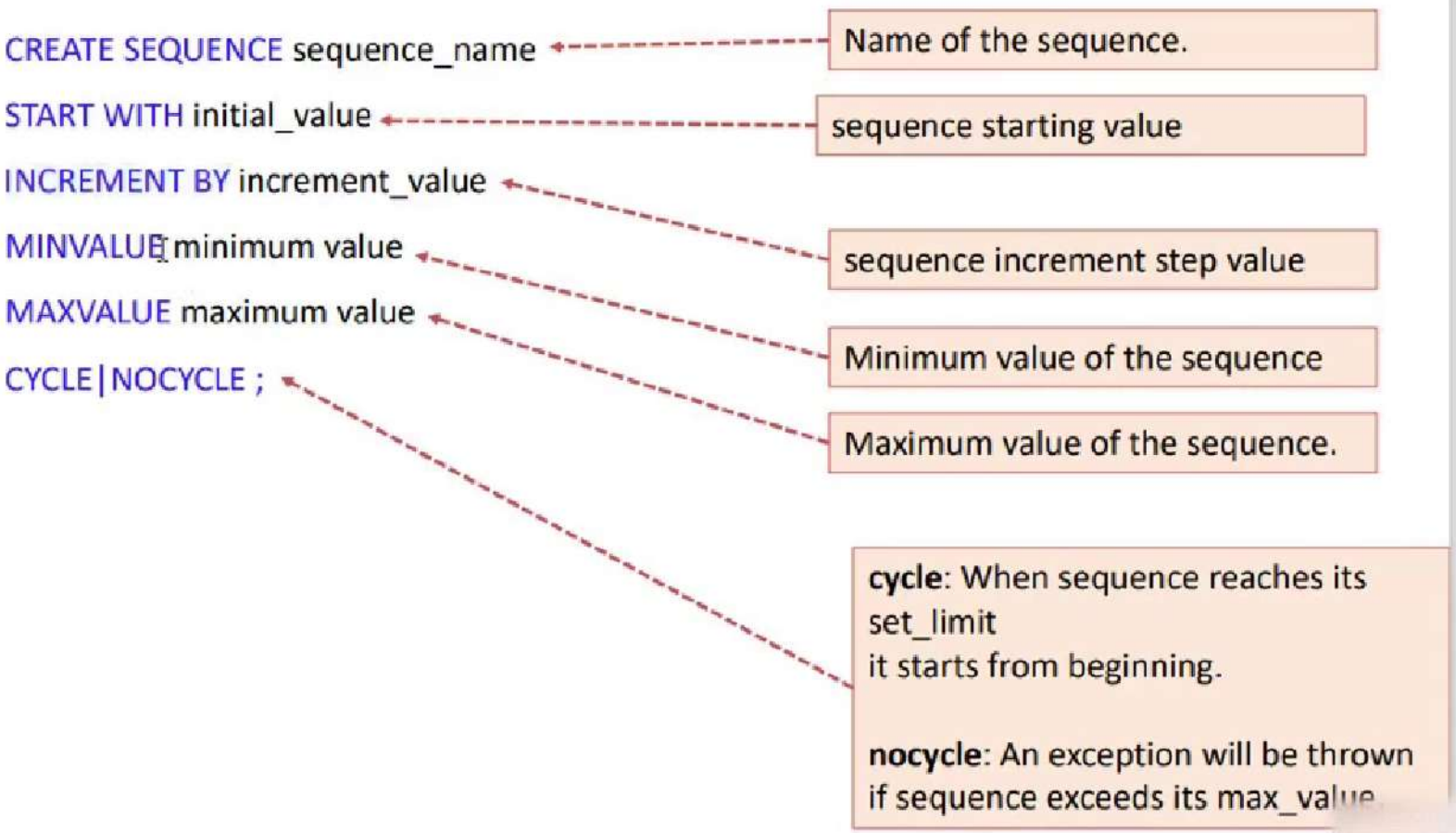
Difference between Identity & Sequence in SQL Server

- **Point 3**
  - The value for the **IDENTITY** property cannot be reset to its initial value.
  - the value for the **SEQUENCE** object can be **reset**.
- **Point 4**
  - A maximum value cannot be set for the **IDENTITY** property.
  - the maximum value for a SEQUENCE object can be defined.

Difference between Identity & Sequence in SQL Server

- **Point 3**
  - The value for the **IDENTITY** property cannot be reset to its initial value.
  - the value for the **SEQUENCE** object can be **reset**.
- **Point 4**
  - A maximum value cannot be set for the **IDENTITY** property.
  - the maximum value for a SEQUENCE object can be defined.

Anatomy of Sequqnece



Difference between Identity & Sequence in SQL Server

Point 2:

- To generate the next IDENTITY value, a new row has to be inserted into the table.
- On the other hand, the next VALUE for a SEQUENCE object can simply be generated using **the NEXT VALUE** FOR clause with the sequence object.

- **Identity Column**

```
Create Table test1
(
    ID      INT     IDENTITY(1,1),
    FName   Varchar(100)
)

    Insert Into Test(FName) Values('Smith')
```

- **Sequence**

```
INSERT into students VALUES( NEXT VALUE FOR sequence_1,'Smith');
```

Creating a Synonym for an Object

Simplify access to objects by creating a synonym (another name for an object). With synonyms, you can:

- Create an easier reference to a table that is owned by another user
- Shorten lengthy object names

CREATE [PUBLIC] SYNONYM synonym
For Object

Creating and Removing Synonyms

Create a shortened name for the Order_current_date View

```
CREATE SYNONYM  O_date
For Order_current_dat;
```

op a synonym

```
ROP SYNONYM  O_date;
```