***Summary***

In this video, Shreyas briefly revisited the topics that were covered in this session, which can be summarised as follows:

**Best practices:**Some of the best practices that you should remember while writing an SQL query are as follows:

* Comment your code using a hyphen **'-'**for a single line and **'/\* ... \*/'**for multiple lines of code.
* Always use table aliases when your query involves more than one source table.
* Assign simple and descriptive names to columns and tables.
* Write SQL keywords in upper case and the names of columns, tables and variables in lower case.
* Always use column names in the 'order by' clause instead of numbers.
* Maintain the right indentation for different sections of a query.
* Use new lines for different sections of a query.
* Use a new line for each column name.
* Use the SQL Formatter or the MySQL Workbench Beautification tool (Ctrl+B) to clean your code.

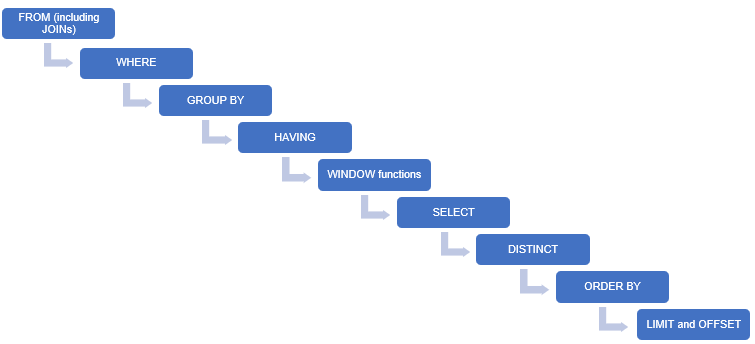
**Indexing:**Indexing is an effective way to optimize query execution, as it selects the required data values instead of processing the entire table. The syntaxes for creating, adding and dropping an index are as follows:

* **CREATE** **INDEX** index\_name
* **ON** **table\_name** (column\_1, column\_2, ...);
* **ALTER** **TABLE** **table\_name**
* **ADD** **INDEX** index\_name(column\_1, column\_2, ...);
* **ALTER** **TABLE** **table\_name**
* **DROP** **INDEX** index\_name;
* --------------------------------------------------------------------------------------------

**Clustered vs non-clustered indexing:**The major differences between clustered and non-clustered indexing are summarised in the table given below.

| **Clustered Index** | **Non-Clustered Index** |
| --- | --- |
| 1. This is mostly the primary key of the table. | 1. It is a combination of one or more columns of the table. |
| 2. It is present within the table. | 2. The unique list of keys is present outside the table. |
| 3. It does not require a separate mapping. | 3. The external table points to different sections of the main table. |
| 4. It is relatively faster. | 4. It is relatively slower. |

**Order of query execution:** The order in which the different SQL statements are executed in a query is depicted in the diagram given below.



Order of Query Execution

**Query optimization techniques:**The points that you should remember while writing a query are as follows:

* Use inner joins wherever possible to avoid having any unnecessary rows in the resultant table.
* Apply all the required filters to get only the required data values from multiple tables.
* Index the columns that are frequently used in the WHERE clause.
* Avoid using DISTINCT while using the GROUP BY clause, as it slows down query processing.
* Avoid using SELECT \* as much as possible. Select only the required columns.
* Use the ORDER BY clause only if it is absolutely necessary, as it is processed late in a query.
* Avoid using LIMIT and OFFSET as much as possible. Instead, apply appropriate filters using the WHERE clause.

**Joins vs nested queries:**Executing a statement with the 'join' clause creates a join index, which is an internal indexing structure. This makes it more efficient than a nested query. However, a nested query would perform better than a join while querying data from a distributed database.