**1.views and stored procedures:**

A view is a stored SELECT statement, and a stored procedure is one or more Transact-SQL statements that execute as a batch.

Views are queried like tables and do not accept parameters. Stored procedures are more complex than views. Stored procedures can have both input and output parameters and can contain statements to control the flow of the code, such as IF and WHILE statements. It is good programming practice to use stored procedures for all repetitive actions in the database.

You will use CREATE VIEW to create a view that selects only two of the columns in the Products table. Then, you will use CREATE PROCEDURE to create a stored procedure that accepts a price parameter and returns only those products that cost less than the specified parameter value.

**2. View and Materialized View:**

Views are a logical virtual table created by “select query” but the result is not stored anywhere in the disk and every time we need to fire the query when we need data, so always we get updated or the latest data from original tables.

Materialized views are also the logical view of our data-driven by the select query but the result of the query will get stored in the table or disk, also the definition of the query will also store in the database.

The performance of Materialized view it is better than normal View because the data of materialized view will be stored in table and table may be indexed so faster for joining also joining is done at the time of materialized views refresh time so no need to every time fire joins statement as in case of view.

**3. b vs b+ tree:**

|  |  |
| --- | --- |
| B tree | B+ Tree |
| All internal and leaf nodes have data pointers | Only leaf nodes have data pointers |
| Since all keys are not available at leaf, search often takes more time. | All keys are at leaf nodes, hence search is faster and accurate. |
| No duplicate of keys is maintained in the tree. | Duplicate of keys are maintained and all nodes are present at leaf. |
| Insertion takes more time and it is not predictable sometimes.  Deletion of internal node is very complex and tree has to undergo lot of transformations. | Insertion is easier and the results are always the same.  Deletion of any node is easy because all node are found at leaf. |
| Leaf nodes are not stored as structural linked list. | Leaf nodes are stored as structural linked list. |
| No redundant search keys are present | Redundant search keys may be present |