INDEXES: (22-11-2024)
========

- it is a database object which is used to retrieve a specific row/rows from a table fastly.
- by using index we can save searching time and improve the performance of database.
- database indexes are similar to book index page in a text book.by using book index page how we are retrieving the required topic from a text book fastly same as by using database index we will retrieve the required row/rows from a table fastly.
- index can be created on a specific column/(s) in the table and this column is known as "indexed key column".
- whenever we are retrieving the required row /rows from a table then we must use this indexed column under "WHERE" clause condition in the query otherwise indexes are not activated.
- all relational databases are supporting the following two types of searching mechanisms those are,
    1. Table scan (Default)
    2. Index scan

1. Table scan:
============

- it is a default scanning mechanism of database.
- in this scan sqlserver is scanning the entire table for required data.so that it will take much time and reduce the performance database.

Ex:

```
SELECT * FROM EMP WHERE SAL=3000;
```

```
      SAL
========
800.0000
1600.0000
1250.0000
2975.0000
1250.0000
2850.0000
2450.0000
3000.0000
5000.0000
1500.0000
1100.0000
950.0000
3000.0000
1300.0000
```

```
WHERE SAL=3000;
```

2. Index scan:
==========

- there are two types of indexes those are,
    i) Non-cluster indexes
    ii) Cluster indexes

i) Non-cluster indexes:
==========

- IN THIS CASE DATA IS STORED IN ONE PLACE AND INDEX PAGE IS STORED IN ANOTHER PLACE.
- SINCE, NON-CLUSTERED INDEX PAGE IS STORED SEPARATELY FROM THE ACTUAL TABLE DATA.
- A NON-CLUSTERED INDEX IS SAME AS BOOK INDEX PAGE IN TEXT BOOK.
- A TABLE CAN HAVE MORE THAN ONE NON-CLUSTERED INDEX.

syntax:
======

CREATE NONCLUSTERED INDEX <INDEX NAME> ON <TABLE NAME>(<COLUMN NAME> <ASC/DESC>);

EX:

```
CREATE TABLE EMP5(EID INT,ENAME VARCHAR(10),SALARY MONEY);

CREATE NONCLUSTERED INDEX NCI ON EMP5(ENAME ASC);
```

TESTING:

```
INSERT INTO EMP5 VALUES(1,'WARD',23000);
INSERT INTO EMP5 VALUES(2,'ALLEN',32000);
INSERT INTO EMP5 VALUES(3,'MILLER',45000);
INSERT INTO EMP5 VALUES(4,'BLAKE',52000);
```

EX:

```
SELECT * FROM EMP5 WHERE ENAME='BLAKE';
```

ii) Cluster indexes:
==========

- A CLUSTERED INDEX DETERMINES THE PHYSICAL ORDER OF DATA IN A TABLE FOR THIS REASON
  A TABLE CAN HAVE ONLY ONE CLUSTERED INDEX.

- A CLUSTERED INDEX CAUSES RECORDS TO BE PHYSICALLY STORED IN A SORTED OR
  SEQUENTIAL ORDER.

- WHEN WE CREATE A TABLE ALONG WITH “PRIMARY KEY” CONSTRAINT THEN INTERNALLY SYSTEM
  IS CREATING A “CLUSTERED INDEX” AUTOMATICALLY.

- CLUSTERED INDEX IS SAME AS DICTIONARY WHERE THE DATA IS ARRANGED BY ALPHABETICAL ORDER.

syntax:
======
CREATE CLUSTERED INDEX <INDEX NAME> ON <TABLE NAME>(<COLUMN NAME> <ASC/DESC>);

EX:
CREATE TABLE EMP6(EID INT,ENAME VARCHAR(10),SALARY MONEY);

CREATE CLUSTERED INDEX CI ON EMP6(ENAME ASC);

TESTING:
INSERT INTO EMP6 VALUES(1,'WARD',23000);
INSERT INTO EMP6 VALUES(2,'ALLEN',32000);
INSERT INTO EMP6 VALUES(3,'MILLER',45000);
INSERT INTO EMP6 VALUES(4,'BLAKE',52000);

EX:
SELECT * FROM EMP6 WHERE ENAME='BLAKE';

25-11-2024:
==========
UNIQUE CLUSTER INDEX AND UNIQUE NON-CLUSTERED INDEX:
==========
- UNIQUE INDEX CAN BE CREATED ON A COLUMN THAT DOES NOT HAVE ANY DUPLICATE VALUES.

- ONCE A UNIQUE INDEX IS CREATED, DUPLICATE VALUES
WILL NOT BE ACCEPTED IN TO A COLUMN.

- IF A TABLE DEFINITION HAS A “PRIMARY KEY OR UNIQUE”
CONSTRAINT THEN SQL SERVER AUTOMATICALLY CREATES A UNIQUE INDEX WHEN WE CREATE THE CREATE TABLE QUERY.

- UNIQUENESS IS A PROPERTY OF AN INDEX SO WE CAN
APPLY ON BOTH CLUSTERED AND NON-CLUSTERED INDEXES.

Apply "UNIQUE" constraint on NON-CLUSTERED INDEX:
==========
syntax:
=======
CREATE UNIQUE NONCLUSTERED INDEX <INDEX NAME> ON <TABLE NAME>(<COLUMN NAME> <ASC/DESC>);

EX:
CREATE TABLE EMP7(SNO INT,NAMES VARCHAR(10));
CREATE UNIQUE NONCLUSTERED INDEX UQNCI ON EMP7(SNO ASC);

TESTING:
INSERT INTO EMP7 VALUES(1,'SMITH');-----ALLOWED
INSERT INTO EMP7 VALUES(1,'ALLEN');-----NOT ALLOWED
INSERT INTO EMP7 VALUES(2,'ALLEN');-----ALLOWED
SELECT * FROM EMP7;

Apply "UNIQUE" constraint on CLUSTERED INDEX:
==========
syntax:
=======
CREATE UNIQUE CLUSTERED INDEX <INDEX NAME> ON <TABLE NAME>(<COLUMN NAME> <ASC/DESC>);

EX:
CREATE TABLE EMP8(SNO INT,NAMES VARCHAR(10));
CREATE UNIQUE CLUSTERED INDEX UQCI ON EMP8(SNO ASC);

INSERT INTO EMP8 VALUES(1,'SMITH');--------ALLOWED
INSERT INTO EMP8 VALUES(1,'ALLEN');------NOT ALLOWED
INSERT INTO EMP8 VALUES(2,'ALLEN');-----ALLOWED
SELECT * FROM EMP8;

Creating CLUSTER INDEX with Primary key constraint:
==========
- whenever we are creating a table along with "PRIMARY KEY" constraint internally system
is creating a CLUSTERED INDEX on a column automatically.

EX:
CREATE TABLE EMP9(EID INT PRIMARY KEY,ENAME VARCHAR(10))'

TESTING:
INSERT INTO EMP9 VALUES(3,'ALLEN');
INSERT INTO EMP9 VALUES(1,'SMITH');
INSERT INTO EMP9 VALUES(2,'JAMES');
SELECT * FROM EMP9;

DIFFERENCE BETWEEN CLUSTERED AND NON-CLUSTERED INDEX:
==========

CLUSTERED INDEX	NON-CLUSTERED INDEX
CLUSTERED INDEX IS FASTER.	NON-CLUSTERED INDEX IS SLOWER.
CLUSTERED INDEX REQUIRES LESS MEMORY FOR OPERATIONS.	NON-CLUSTERED INDEX REQUIRES MORE MEMORY FOR OPERATIONS.
IN CLUSTERED INDEX, INDEX IS THE MAIN DATA.	IN NON-CLUSTERED INDEX, INDEX IS THE COPY OF DATA.
A TABLE CAN HAVE ONLY ONE CLUSTERED INDEX	A TABLE CAN HAVE MULTIPLE NON CLUSTERED INDEX.
IN CLUSTERED INDEX, CLUSTERED KEY DEFINES ORDER OF DATA WITHIN TABLE.	IN NON-CLUSTERED INDEX, INDEX KEY DEFINES ORDER OF DATA WITHIN INDEX.

How to drop an index:
==========
syntax:
=======
DROP INDEX <TN>.<INDEX NAME>;

EX:
DROP INDEX EMP5.NCI;
DROP INDEX EMP6.CI;
==========
==========