Database Design & Applications

The Database Language - SQL Index

Objective

- Creating Index
- Need of Index
- Advantages of Index
- Disadvantages of Index
- When to Create Index
- When not to Create Index

Need of Index

- Indexes are used by queries to find data from tables or views quickly.
- An Index on a table to an index in a book.
- Existence of right indexes can drastically improve the performance of the query.

What is an Index?

An index:

- Is a schema object
- Is used by the MSSQL Server to speed up the retrieval of rows by using a pointer
- Can reduce disk I/O by using a rapid path access method to locate data quickly
- Is independent of the table it indexes
- Is used and maintained automatically by the MSSQL Server

Index Example

- At this moment the Employee table does not have a index on salary column

ID	NAME	SALARY
1	SAM	2500
2	SMITH	6500
3	JOHN	4500
4	SARA	5500
5	KEVIN	3100

```
SELECT * FROM Employee
Where Salary >5000 and Salary < 6000
```

- Since there is no index on Salary column, the query engine performs an entire table scan.

Creating an Index

- Create an index on one or more columns.

```
CREATE INDEX index
ON table (column[ASC|DESC][, column...);
```

- Improve the speed of query access to the LAST_NAME column in the EMPLOYEES table.

```
CREATE INDEX emp_last_name_idx
ON    employees(last_name);
```

Creating an Index

```
CREATE INDEX idx_Employee_Salary
ON Employee (Salary ASC);
```

ID	NAME	SALARY
1	SAM	2500
2	SMITH	6500
3	JOHN	4500
4	SARA	5500
5	KEVIN	3100

Salary	RowAddress
2500	Row Address
3100	Row Address
4500	Row Address
5500	Row Address
6500	Row Address

```
SELECT * FROM Employee
Where Salary >5000 and Salary < 6000
```

- Now with the help of index SQL Engine picks up the row addresses from the index and directly fetches the records from the table.

How Are Indexes Created?

- **Automatically:** A unique index is created automatically when you define a PRIMARY KEY or UNIQUE constraint in a table definition.
- **Manually:** Users can create indexes on columns to speed up access to the rows

Advantages of Index

- CREATE INDEX idx_Employee_Salary
- ON Employee (Salary ASC);

--SELECT statements with where clause

SELECT * FROM Employee

Where Salary >5000 and Salary < 6000

--Delete Statement

- DELETE FROM Employee WHERE Salary >50000;

--UPDATE Statement

- UPDATE Employee SET Salary = 5000 WHERE Salary <5000;

--ORDER BY

- SELECT * FROM Employee ORDER BY Salary DESC

-- GROUP BY

- SELECT Salary, COUNT(Salary) as S_count
- FROM Employee
- GROUP BY Salary

ID	NAME	SALARY
1	SAM	2500
2	SMITH	6500
3	JOHN	4500
4	SARA	5500
5	KEVIN	3100

Salary	RowAddress
2500	Row Address
3100	Row Address
4500	Row Address
5500	Row Address
6500	Row Address

Disadvantages of Index

- **Additional Disk Space:**
  - Indexes are stored separately from table, therefore needs additional space.
- **INSERT , UPDATE and DELETE statements become slow:**
  - When DML statements modifies a table, the data in all the indexes also needs to be updated.

When to Create an Index

You should create an index if:

- A column contains a wide range of values
- A column contains a large number of null values
- One or more columns are frequently used together in a WHERE clause or a join condition
- The table is large and most queries are expected to retrieve less than 2 to 4 percent of the rows

When Not to Create an Index

It is usually not worth creating an index if:

- The table is small
- The columns are not often used as a condition in the query
- Most queries are expected to retrieve more than 2 to 4 percent of the rows in the table
- The table is updated frequently
- The indexed columns are referenced as part of an expression

Dropping Index

- To drop the indexes of a table:
  - DROP INDEX table_name.index_name;
- Example:
  - DROP INDEX employee.idx_f_name;

THANK YOU!