Q

What is the difference between DBMS and RDBMS?

Microsoft SQL Server

DBMS	RDBMS
1. DBMS stores data as file.	RDBMS stores data in **TABULAR** form.
2. No relationship between data.	Data is stored in the form of tables which are **RELATED** to each other. Eg: Foreign key relationship.
3. Normalization is not present.	**NORMALIZATION** is present.
4. It deals with small quantity of data.	It deals with **LARGE** amount of data.
5. Examples: XML	Examples: MySQL, PostgreSQL, SQL Server, Oracle, Microsoft Access etc.

SUBSCRIBE

Q

What is a Constraint is SQL? What are its types. V. IMP.

Microsoft SQL Server

SQL constraints are **used to specify rules for the data in a table**.
Constraints are used to limit the type of data that can go into a table.

1. A **PRIMARY KEY** is a field which can uniquely identify each row in a table.

3. A **FOREIGN KEY** is a field which can uniquely identify each row in an another table.

4. **CHECK** constraint helps to validate the values of a column to meet a particular condition.

```
CREATE TABLE Students (
    ID int NOT NULL PRIMARY KEY,
    Name varchar(255) NOT NULL,
    CourseID int FOREIGN KEY REFERENCES Courses(CourseID),
    Age int NOT NULL CHECK (AGE >= 18),
    AdmissionDate date DEFAULT GETDATE(),
    CONSTRAINT UC_Student UNIQUE (ID,Name)
);
```

2. **NOT NULL** constraint tells that we cannot store a null value in a column.

6. **UNIQUE** constraint tells that all the values in the column must be unique.

5. **DEFAULT** constraint specifies a default value for the column when no value is specified by the user.

SUBSCRIBE

Q

What is the difference between Primary key and Unique key?

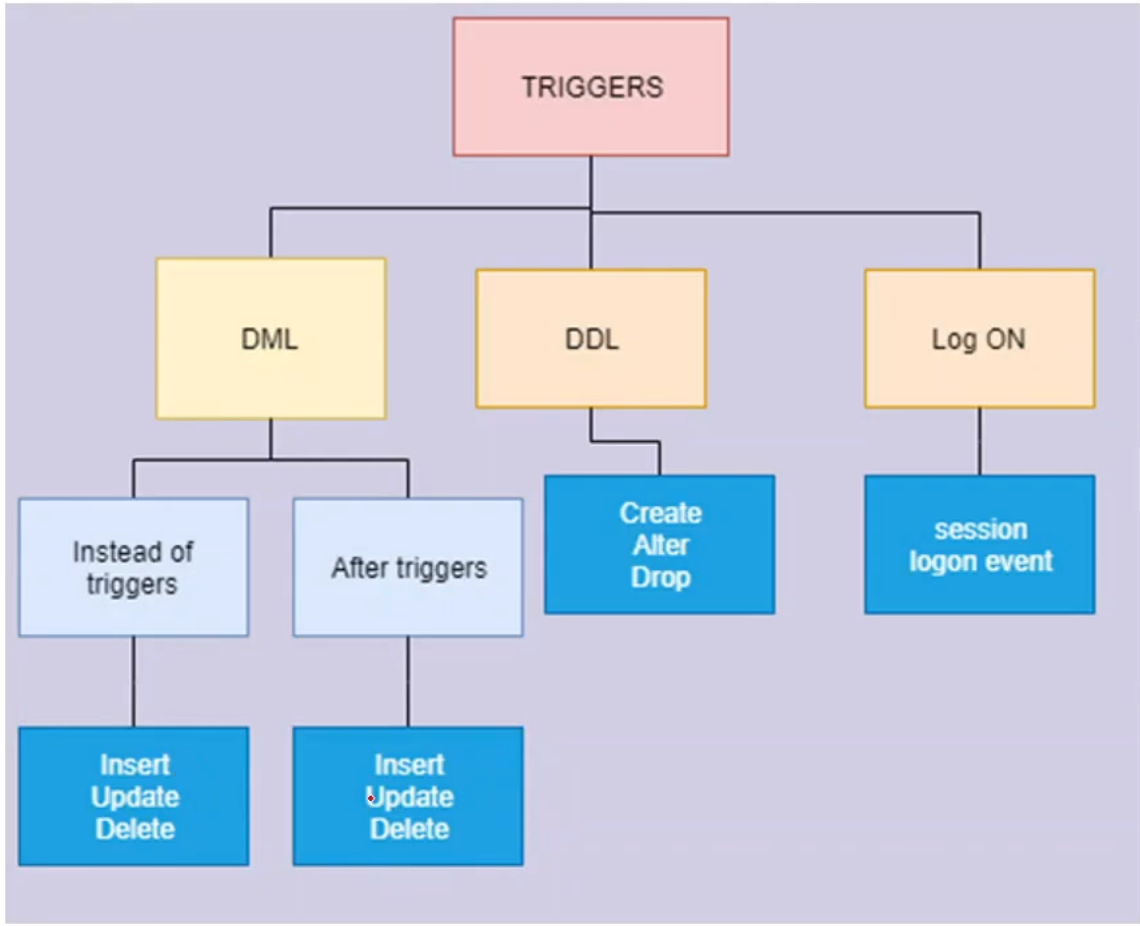
Microsoft SQL Server

	Primary Key	Unique Key
1	Primary Key Can't Accept Null Values.	Unique Key Can Accept Only One Null Value
2	Creates Clustered Index	Creates Non-Clustered Index
3	Only One Primary key in a Table	More than One Unique Key in a Table.

SUBSCRIBE

What Triggers and types of triggers?

Triggers are stored programs, which are **AUTOMATICALLY** executed or fired when some events (insert, delete and update) occur.



Example of Instead of(DML) Trigger

```
1
2  CREATE TRIGGER [dbo].[TRG_VM_EMPDETAILS]
3  ON [dbo].[VW_EMPDETAILS]
4  INSTEAD OF INSERT
5  AS
6  BEGIN
7   -- LOGIC HERE
8
9  END
10
11
12
13
```

Trigger Name

View Name

INSTEAD OF Trigger

This trigger is only for INSERT

An INSTEAD OF trigger is a trigger that allows you to skip an INSERT , DELETE , or UPDATE statement to a table or a view and execute other statements defined in the trigger instead.

What are Triggers and types of triggers?

Triggers are stored programs, which are **AUTOMATICALLY** executed or fired when some events (insert, delete and update) occur.

Locations -> LocationHist

Example of After(DML) Trigger

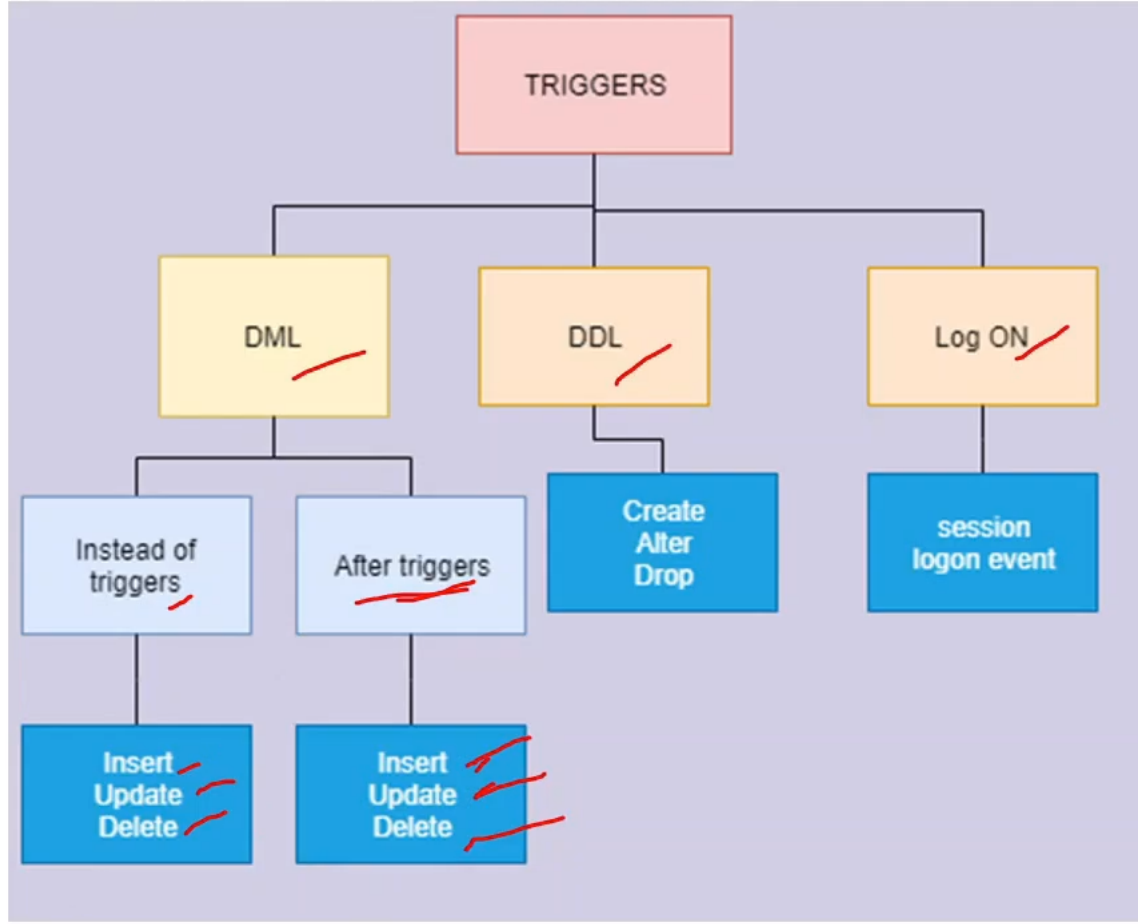
```
CREATE TRIGGER TR_UPD_Locations ON Locations
FOR UPDATE
NOT FOR REPLICATION
AS

BEGIN
    INSERT INTO LocationHist
    SELECT LocationID
        ,getdate()
    FROM inserted
END
```

Table Name

Trigger Name

DML Event

T-SQL block that runs against specified DML Event



In after trigger, Update on the table executed first and then trigger will run. The above example is of after trigger.

Q What is the difference between Having clause and Where clause? V. IMP.

Microsoft SQL Server

WHERE Clause is used before GROUP BY Clause.

HAVING Clause is used after GROUP BY Clause.

```
SELECT COUNT(CustomerID), Country
FROM Customers
WHERE Country = "India"
GROUP BY Country
HAVING COUNT(CustomerID) > 5;
```

WHERE Clause cannot contain AGGREGATE function.

HAVING Clause can contain aggregate function.

```
select EmpName from Employee
GROUP BY EmpName
HAVING SUM(EmpSalary) <30000
```

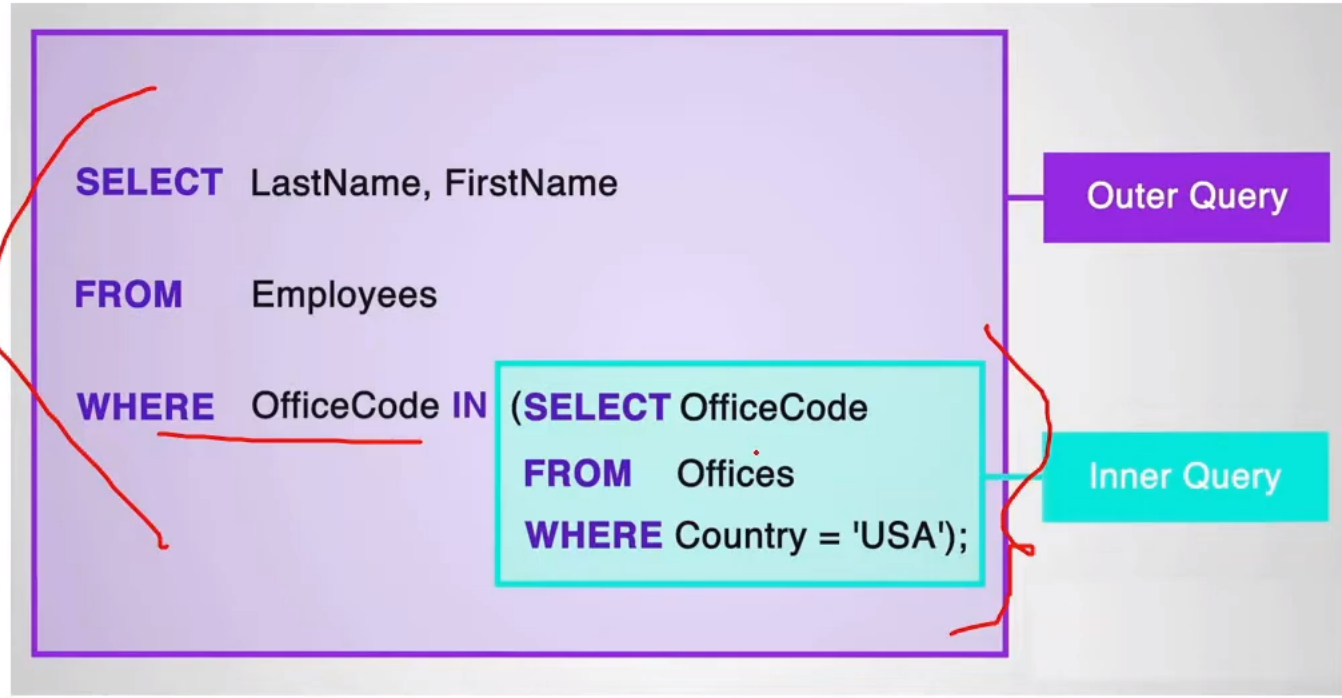
SUBSCRIBE

Q

What is Sub query or Nested query or Inner query in SQL?

Microsoft SQL Server

A Subquery or Inner query or a Nested query is a query within another SQL query and **embedded** within the **WHERE** clause.



SUBSCRIBE

Q

What is Auto Increment/ Identity column in SQL Server?

Microsoft SQL Server

Auto-increment allows a unique number to be **generated automatically** when a new record is inserted into a table.

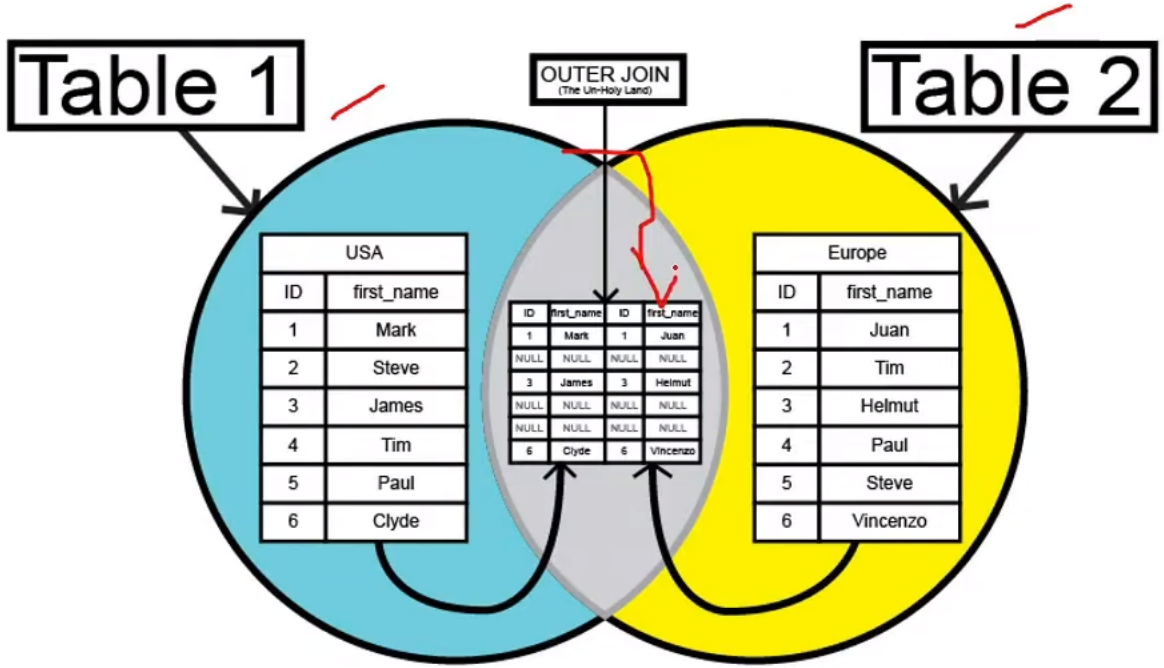
Mostly it is the primary key only.

```
CREATE TABLE Persons (
    Personid int IDENTITY(1,1) PRIMARY KEY,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int
);
```

SUBSCRIBE

What are Joins in SQL?

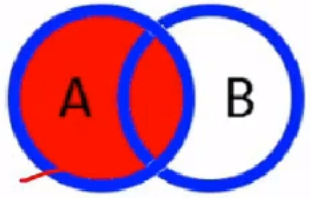
A join clause is used to COMBINE rows from two or more tables, based on a related column between them.



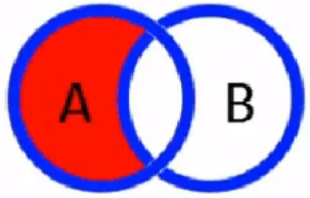
Q What are the types of Joins in SQL Server? V. IMP.

Microsoft SQL Server

LEFT OUTER JOIN

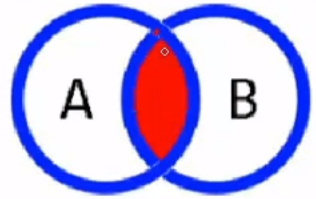



```
SELECT *
FROM TableA a
LEFT JOIN TableB b
    ON a.KEY = b.KEY
```



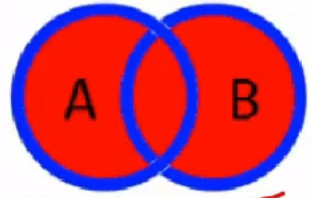

```
SELECT *
FROM TableA a
LEFT JOIN TableB b
    ON a.KEY = b.KEY
WHERE b.KEY IS NULL
```

INNER JOIN

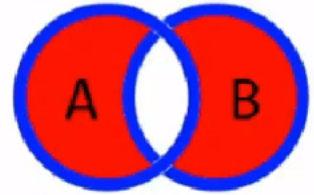



```
SELECT *
FROM TableA a
INNER JOIN TableB b
    ON a.KEY = b.KEY
```

FULL OUTER JOIN

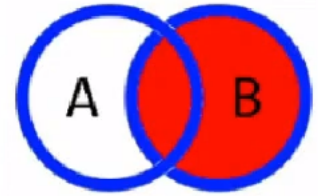



```
SELECT *
FROM TableA a
FULL OUTER JOIN TableB b
    ON a.KEY = b.KEY
```

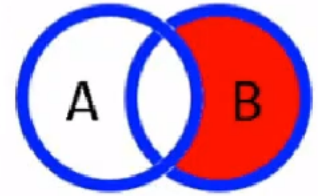



```
SELECT *
FROM TableA a
FULL OUTER JOIN TableB b
    ON a.KEY = b.KEY
WHERE a.KEY IS NULL
    OR b.KEY IS NULL
```

RIGHT OUTER JOIN




```
SELECT *
FROM TableA a
RIGHT JOIN TableB b
    ON a.KEY = b.KEY
```



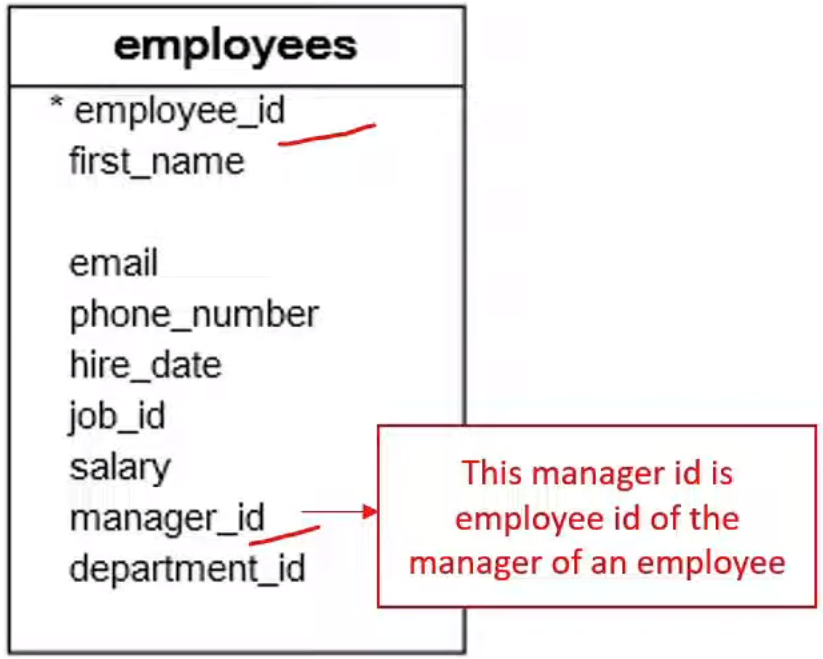

```
SELECT *
FROM TableA a
RIGHT JOIN TableB b
    ON a.KEY = b.KEY
WHERE a.KEY IS NULL
```

SUBSCRIBE

What is Self-Join?

A self join is **a join of a table to itself**.

When to use Self Join



Now your task is to get the employee name with his/her manager name.

```
SELECT

    e.first_name AS employee,
    m.first_name AS manager

FROM

   employees e LEFT JOIN
   employees m

    ON m.employee_id = e.manager_id

ORDER BY manager;
```

employee	manager
Steven King	NULL
Bruce Ernst	Alexander Hunold
David Austin	Alexander Hunold
Valli Pataballa	Alexander Hunold
Diana Lorentz	Alexander Hunold
Alexander Khoo	Den Raphaely

Q

Write a SQL query to fetch all the Employees who are also Managers.

Microsoft SQL Server

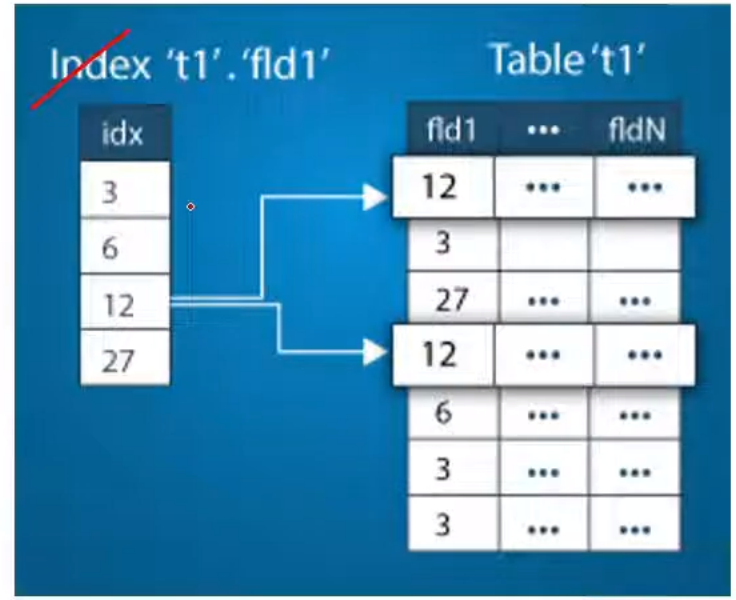
employees
* employee_id
first_name
email
phone_number
hire_date
job_id
salary
manager_id
department_id

```sql
SELECT

    e.first_name AS employee,
    m.first_name AS manager

FROM

    employees e
        INNER JOIN
    employees m ON m.employee_id = e.manager_id
```

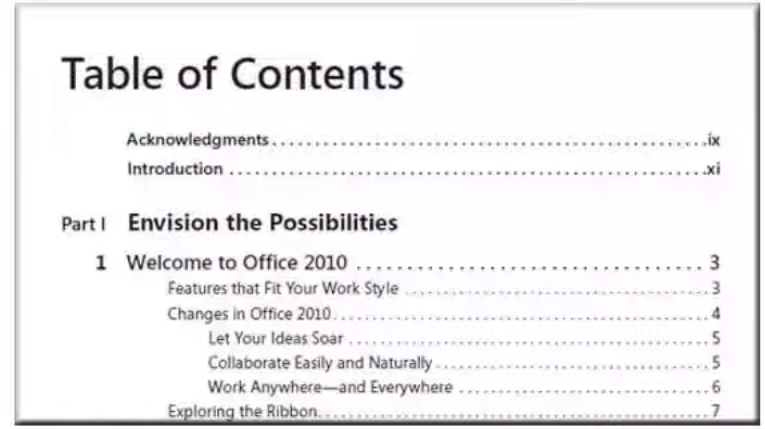
SUBSCRIBE

What are Indexes in SQL Server?

SQL Indexes are used in relational databases to retrieve data **VERY FAST**.



They are similar to indexes at the start of the BOOKS, which purpose is to find a topic quickly.



What is Clustered index?

Microsoft SQL Server

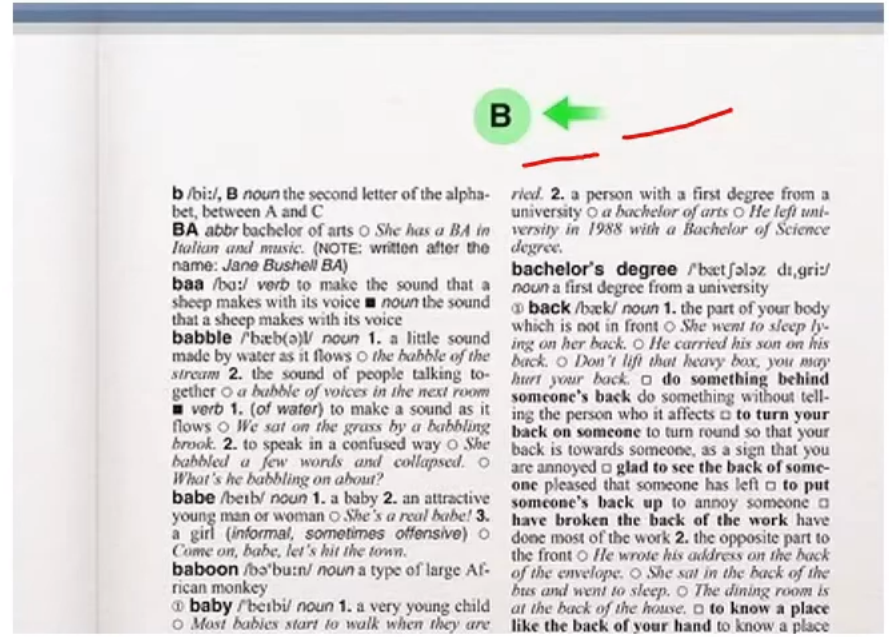
CLUSTERED INDEX

A clustered index defines the order in which data is **physically** stored in a table.

Table data can be sorted in only way, therefore, there can be only one clustered index per table.

In SQL Server, if you set a primary key on a column then it will automatically creates a clustered index on that particular column.

Dictionary



B

b /biː/, **B** *noun* the second letter of the alphabet, between A and C

BA *abbr* bachelor of arts ○ *She has a BA in Italian and music.* (NOTE: written after the name: *Jane Bushell BA*)

baa /bɑː/ *verb* to make the sound that a sheep makes with its voice ■ *noun* the sound that a sheep makes with its voice

babble /ˈbæb(ə)l/ *noun* **1.** a little sound made by water as it flows ○ *the babble of the stream* **2.** the sound of people talking together ○ *a babble of voices in the next room* ■ *verb* **1.** (*of water*) to make a sound as it flows ○ *We sat on the grass by a babbling brook.* **2.** to speak in a confused way ○ *She babbled a few words and collapsed.* ○ *What's he babbling on about?*

babe /beɪb/ *noun* **1.** a baby **2.** an attractive young man or woman ○ *She's a real babe!* **3.** a girl (*informal, sometimes offensive*) ○ *Come on, babe, let's hit the town.*

baboon /bəˈbuːn/ *noun* a type of large African monkey

① **baby** /ˈbeɪbi/ *noun* **1.** a very young child ○ *Most babies start to walk when they are ... ried.* **2.** a person with a first degree from a university ○ *a bachelor of arts* ○ *He left university in 1988 with a Bachelor of Science degree.*

bachelor's degree /ˈbætʃələz dɪˌgriː/ *noun* a first degree from a university

① **back** /bæk/ *noun* **1.** the part of your body which is not in front ○ *She went to sleep lying on her back.* ○ *He carried his son on his back.* ○ *Don't lift that heavy box, you may hurt your back.* □ **do something behind someone's back** do something without telling the person who it affects □ **to turn your back on someone** to turn round so that your back is towards someone, as a sign that you are annoyed □ **glad to see the back of someone** pleased that someone has left □ **to put someone's back up** to annoy someone □ **have broken the back of the work** have done most of the work **2.** the opposite part to the front ○ *He wrote his address on the back of the envelope.* ○ *She sat in the back of the bus and went to sleep.* ○ *The dining room is at the back of the house.* □ **to know a place like the back of your hand** to know a place

SUBSCRIBE

What is Non-Clustered index?

NON-CLUSTERED INDEX

A non-clustered index is stored at one place and table data is stored in another place. So this index is not physically stored.

A table can have multiple non-clustered index in a table.

Book Index

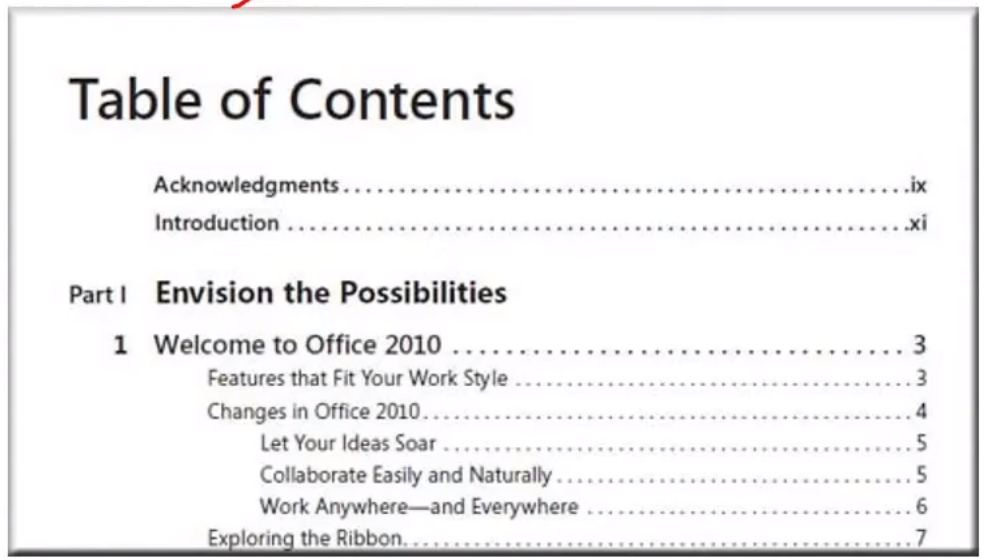
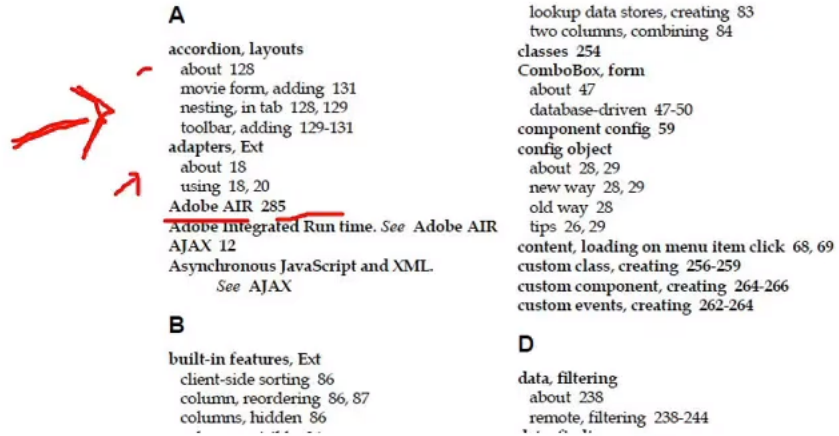


Table of Contents

Index



Q What is the difference between Clustered and Non-Clustered index? V. IMP.

Microsoft SQL Server

1. A clustered index defines the order in which data is **physically** stored in a table. For example Dictionary.

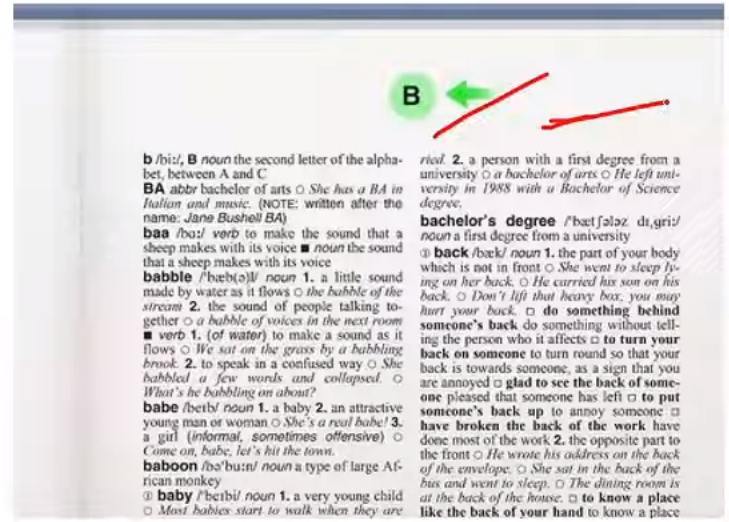
A non-clustered index is stored at one place and table data is stored in another place. For example Book Index.

2. A table can have only one clustered index.

A table can have multiple non-clustered index.

3. Clustered index is faster.

Non-clustered index is slower.



B

b /biː/, **B** *noun* the second letter of the alphabet, between A and C

BA *abbr* bachelor of arts ○ *She has a BA in Italian and music.* (NOTE: written after the name: *Jane Bushell BA*)

baa /bɑː/ *verb* to make the sound that a sheep makes with its voice ■ *noun* the sound that a sheep makes with its voice

babble /ˈbæb(ə)l/ *noun* **1.** a little sound made by water as it flows ○ *the babble of the stream* **2.** the sound of people talking together ○ *a babble of voices in the next room* ■ *verb* **1.** (*of water*) to make a sound as it flows ○ *We sat on the grass by a babbling brook.* **2.** to speak in a confused way ○ *She babbled a few words and collapsed.* ○ *What's he babbling on about?*

babe /beɪb/ *noun* **1.** a baby **2.** an attractive young man or woman ○ *She's a real babe!* **3.** a girl (*informal, sometimes offensive*) ○ *Come on, babe, let's hit the town.*

baboon /bəˈbuːn/ *noun* a type of large African monkey

① **baby** /ˈbeɪbi/ *noun* **1.** a very young child ○ *Most babies start to walk when they are* *ried.* **2.** a person with a first degree from a university ○ *a bachelor of arts* ○ *He left university in 1988 with a Bachelor of Science degree.*

bachelor's degree /ˈbætʃələz dɪˌgriː/ *noun* a first degree from a university

① **back** /bæk/ *noun* **1.** the part of your body which is not in front ○ *She went to sleep lying on her back.* ○ *He carried his son on his back.* ○ *Don't lift that heavy box, you may hurt your back.* □ **do something behind someone's back** do something without telling the person who it affects □ **to turn your back on someone** to turn round so that your back is towards someone, as a sign that you are annoyed □ **glad to see the back of someone** pleased that someone has left □ **to put someone's back up** to annoy someone □ **have broken the back of the work** have done most of the work **2.** the opposite part to the front ○ *He wrote his address on the back of the envelope.* ○ *She sat in the back of the bus and went to sleep.* ○ *The dining room is at the back of the house.* □ **to know a place like the back of your hand** to know a place

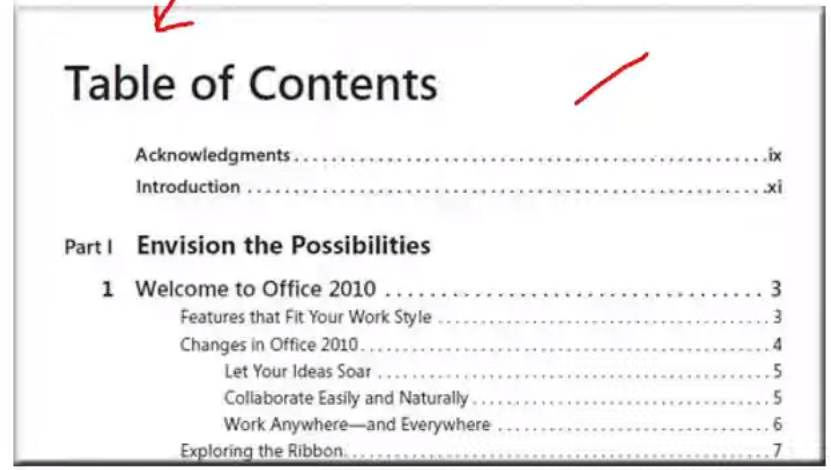


Table of Contents

Q

How to create Clustered and Non-Clustered index in a table?

Microsoft SQL Server

CLUSTERED INDEX

When you create a PRIMARY KEY constraint, a clustered index on the column or columns is automatically created.

```
CREATE CLUSTERED INDEX <index_name>
ON <table_name>(<column_name> ASC/DESC)
```

NON-CLUSTERED INDEX

```
CREATE NONCLUSTERED INDEX <index_name>
ON <table_name>(<column_name> ASC/DESC)
```

SUBSCRIBE

Q

In which column you will apply the indexing to optimize this query. “select id, class from student where name=”happy””?

Microsoft SQL Server

select id, class from student where name=”happy”

The column **after WHERE** condition, which is “NAME” here.

Q What is the difference between Stored Procedure and Functions (at least 3)? V. IMP.

Microsoft SQL Server

1. SP may or may not **return** a value
but function must return a value.

2. SP can have input/**output** parameters
but function only has input parameters.

3. We can **call** function inside SP but cannot call SP from a function.

4. We cannot use SP in **SQL statements** like SELECT, INSERT, UPDATE, DELETE, MERGE, etc. but we can use them with function.

*SELECT *, dbo.fnCountry(city.long) FROM city;*

5. We can use try-catch exception handling in SP but we cannot do that in function.

6. We can use transactions inside SP but it is not possible in function.

```
--Stored Procedure

CREATE PROCEDURE proc_name
(@Ename varchar(50),
@EId int output)
AS
BEGIN

    INSERT INTO Employee (EmpName) VALUES (@Ename)
    SELECT @EId= SCOPE_IDENTITY()

END


--UDF – User Defined Functions

CREATE FUNCTION function_name (parameters)

--only input parameter
RETURNS data_type AS
BEGIN
    SQL statements
    RETURN value

END;
```

SUBSCRIBE

How to optimize a Stored Procedure or SQL Query?

1. Use SET NOCOUNT ON

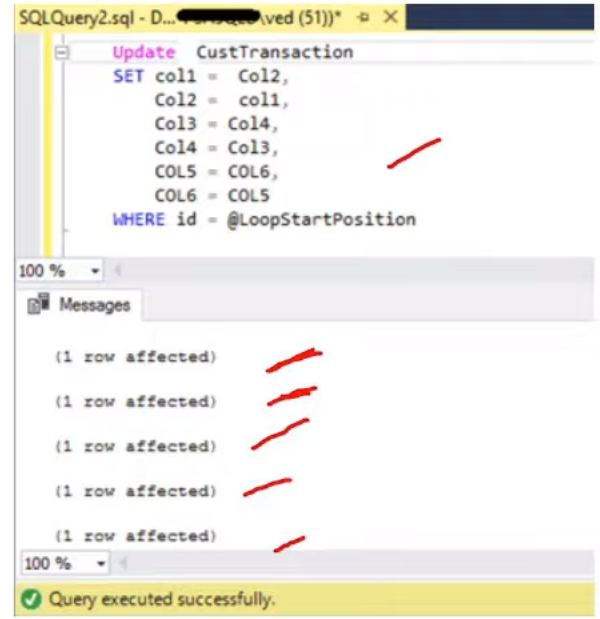
2. Specify column names instead of using * in SELECT statement.

3. Use schema name before objects or tablenames.
   Example: SELECT EmpID, Name FROM dbo.Employee

4. Do not use DYNAMIC QUERIES. They are vulnerable to SQL Injections.

5. Use EXISTS () instead of COUNT ().
Example :SELECT Count(1) FROM dbo.Employee
Example: IF(EXISTS (SELECT 1 FROM db.Employees))

6. Use TRANSACTION when required only



What is a Cursor? Why to avoid them?

A database Cursor is a control which enables traversal/ iteration over the rows or records in the table.

5 step process:

1. Declare
2. Open
3. Fetch using while loop
4. Close
5. Deallocate

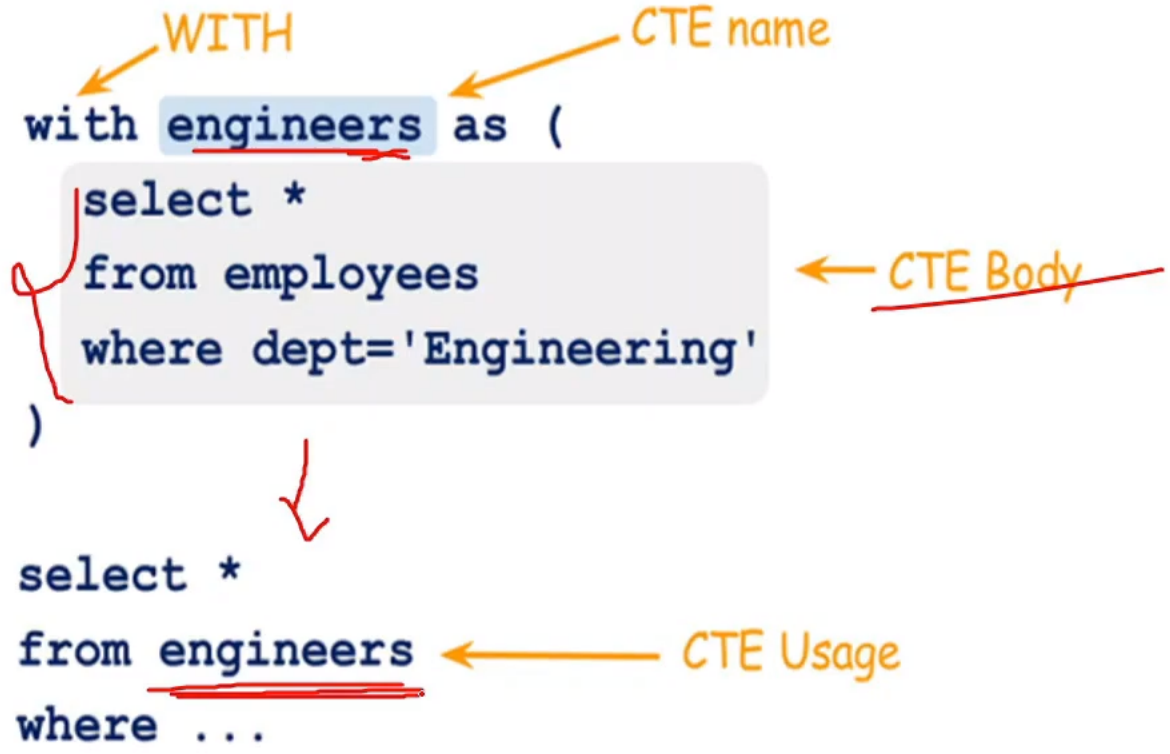
LIMITATION

A cursor is a MEMORY resident set of pointers -- meaning it occupies lots of memory from your system which is not good for performance.

```
DECLARE
    @product_name VARCHAR(MAX),
    @list_price   DECIMAL;

DECLARE cursor_product CURSOR
FOR SELECT
        product_name,
        list_price
    FROM
        production.products;

OPEN cursor_product;

FETCH NEXT FROM cursor_product INTO
    @product_name,
    @list_price;

WHILE @@FETCH_STATUS = 0
    BEGIN
        PRINT @product_name + CAST(@list_price AS varchar);
        FETCH NEXT FROM cursor_product INTO
            @product_name,
            @list_price;
    END;

CLOSE cursor_product;

DEALLOCATE cursor_product;
```

What is CTE in SQL Server?

A Common Table Expression, is a TEMPORARY named result set, that you can reference within a SELECT, INSERT, UPDATE, or DELETE statement.



Q

What is the difference between Delete, Truncate and Drop commands? V. IMP.

Microsoft SQL Server

DELETE

1. It is a DML.

2. It is used to delete the one or more rows(data) of a table.

3. It can be rollback.

```
DELETE FROM Employees
WHERE Emp_Id = 7;
```

TRUNCATE

1. It is a DDL.

2. It is used to delete all rows from the table.

3. It can be rollback.

4. Trincate will remove all the records from the table Employees but **not** the structure/ schema.

```
TRUNCATE TABLE Employees;
```

DROP

1. It is a DDL.

2. It is used to drop the whole table **with STRUCTURE/SCHEMA**.

3. It **can not** be rollback.

4. It will remove the structure/ schema also.

```
DROP TABLE Employees;
```

SUBSCRIBE

Q How to get the Nth highest salary of an employee? V. IMP.

1. The logic is first select TOP 3 salaries in descending order.

```
SELECT DISTINCT TOP N SALARY
    FROM tbl_Employees
    ORDER BY SALARY DESC
```

Salary
5000
10000
6000
4000
2000
7000

10000
7000
6000

2. Put the result in "Result" and then do order by asc

```
SELECT  SALARY
FROM (

    SELECT DISTINCT TOP N SALARY
    FROM tbl_Employees
    ORDER BY SALARY DESC

) RESULT
ORDER BY SALARY
```

Result
6000
7000
10000

3. Select top 1 salary from result set

```
SELECT  TOP 1 SALARY
FROM (

    SELECT DISTINCT TOP N SALARY
    FROM tbl_Employees
    ORDER BY SALARY DESC

) RESULT
ORDER BY SALARY
```

Result
6000

```sql
-- Create a sample table
CREATE TABLE SampleTable (
    ID INT PRIMARY KEY,
    Name VARCHAR(50),
    Age INT
);

-- Inserting data into the table
INSERT INTO SampleTable (ID, Name, Age) VALUES (1, 'John', 25);
INSERT INTO SampleTable (ID, Name, Age) VALUES (2, 'Alice', 30);

-- Selecting data from the table
SELECT * FROM SampleTable;

-- Updating data in the table
UPDATE SampleTable
SET Age = 28
WHERE Name = 'John';

-- Deleting data from the table
DELETE FROM SampleTable
WHERE Name = 'Alice';
```