(26-09-2024) SQLSERVER
===========

- In IT field users are interacting with two types of applications.
  1. Front end applications
  2. Back end applications

1. Front end applications:
=====================

- it is an application where the end-users are interacting directly.

  Ex: Register form,Login form,View form,Home page,.....etc.

Design & Develop:
===============

- UI technologies(html,css,javascript,angular JS,react JS,Json,..etc)

2. Back end applications:
=====================

- it is an application where we store the end-users data / information.

  Ex: Databases.

Design & Develop:
===============

- DB technologies(SQLServer,Oracle,Mysql,Postgresql,Db2,......etc)

Server Side Technologies:
======================

- these technologies are used to establish connection in between front end application and back end application.

Ex: .Net / .Net core , Java , Python ,......etc

27-09-2024:
==========
What is SQLSERVER:
===================

- it is a DB software / Back end tool / RDBMS product(ORDBMS model).

Where we want to use SQLSERVER :
===============================

- SQLSERVER s/w can be used in different types of applications.

  > Banking applications
  > HR management system applications
  > Educational system applications
  > Hospital management system applications

> Transport applications
> Sales & Production management system,............etc

Who want to learn SQLSERVER course:
=================================

- Any person (.Net / .Net core , MSBI ,ADF,SQLSERVER DBA,[Testing,AWS,Devops,DS,PowerBI])

What are the pre-requisites for SQLSERVER course:
==

- No

What about JOB's:
================

- SQL developer / DB developer.
- T/SQL programmer.

Course Details:
=============

Course name : SQLSERVER (Microsoft)
Duration : 45-50 sessions
Time : 6.15 pm - 7.25 pm
Mode Type : Offline & Online

IN SQLSEVER Course :
====================

Topic - 1 : DBMS
Topic - 2 : SQLSERVER
Topic - 3 : SQL
Topic - 4 : NORMALIZATION
Topic - 5 : T / SQL

===
======

30-09-2024: Topic - 1 : DBMS
========== ==============

What is Data ?
=============

- it is a rawfact.(i.e characters,numbers,special char's and symbols)
- data never give meaningfull statements.

Ex:
1001 is data SMITH is data

1002 is data ALLEN is data
1003 is data MILLER is data

What is Information?
================

- processing data is called as "Information".
- information is always provide meaningfull statements.

Ex:

Customer_ID	Customer_Name
===========	=============
1001	SMITH
1002	ALLEN
1003	MILLER

What is Database?
================

- it is a memory which is used to store the collection of inter-related data / information of a particular business organization.

Ex:

```
> SBI_Bank_Database
        > group of branches -----------------------> group of customers
                > group of departments
                        > group of employees
```

Ex:

Inter-related information:
=====================

no employees = no department
no department = no employees

no customers = no products
no products = no customers

Types of Databases?
=================

- there are two types of databases in real world.
    1. OLTP(online transaction processing)
    2. OLAP(online analytical processing)

1. OLTP:
=======

- these databases are used for storing "day-to-day" transactional information.
Ex: SQLServer,Oracle,Mysql,Postgresql,...........etc

2. OLAP:
========

- these databases are used for storing "historical" data/information.(i.e Bigdata)
Ex: Datawarehouse

01-10-2024:
==========

What is DBMS?
============

- it is a s/w which is used to manage and maintain data/information with in the database.
- by using DBMS s/w we will perform the following operations are,

> Create Database
> Create Tables
> Inserting data
> Updating data
> Selecting data
> Deleting data

- Here DBMS will act as an interface between User and Database.

Models of DBMS?
==============

- there are three models in DBMS.

1) Hierarchical Database Management System(HDBMS)
s/w : IMS (information management system)

2) Network Database Management System(NDBMS)
s/w : IDBMS (integrated database management system)

NOTE:
=====

- HDBMS,NDBMS models are outdated in real time.

3) Relational Database Management System(RDBMS):
==

- it again two modules.

i) Object Relational DBMS(ORDBMS):
==============================

- these databases are storing data in the form of "Table".
Table : collection of rows & columns.

Row : Group of columns.
Database : collection of tables.

- a row can be called as "Record / Tuple".
- a column can be called as "Attribute / Field".
- these databases are completely depends on "SQL".so that these are called as "SQL Databases".

Ex: SQLServer,Oracle,Mysql,PostgreSQL,DB2,SYBASE,MAXDB,..........etc

ii) Object Oriented DBMS(OODBMS):
==============================

- these databases are storing data in the form "Object".
- these databases are completely depends on "OOPS" concept but not on "SQL" .so that these are called as "NoSQL Databases".

Ex: MongoDB,Cassandra,...etc

==
=================

02-10-2024 Topic - 2 : SQLSERVER
========== =====================

Introduction to SQLSERVER:
========================

- SQLSERVER is an RDBMS(ORDBMS) product which was introduced by "microsoft" in 1989.
- SQLSERVER is used to store data / information permanently(i.e Hard disk) along with security.
- When we want to deploy(install) sqlserver s/w then we need a platform.

What is Platform:
===============

- it is combination of operating system and micro-processor.
- there are two types of platforms in real world.
  i) Platform Dependent
  ii) Platform Independent

i) Platform Dependent:
===================

- it supports only one operating system with the combination any micro-processor.

Ex: Cobal,Pascal,C,C++.

ii) Platform Independent:
=====================

- it supports any operating system with the combination of any micro-processor.

Ex: .Net core,Java,Python,SQLSERVER,Oracle,Mysql,..........etc

- it means that SQLSERVER s/w can be installed in WINDOWS,LINUX and MAC operating system.

03-10-2024:
===========
Working with SQLSERVER:
======================

- Once we install SQLSERVER s/w internally there are two components are installed in the system automatically.

i) Client component
ii) Server component

i) Client:
=======

- By using a client tool we will perform the following three operations.

Step1: User can connect to SQLSERVER
Username : sa (system admin - Default username)
Password : 123 (created by user at the time of sqlserver s/w installation)
connected.

Step2: User can send a request to SQLSERVER.
Request : SQL query

Step3: User will get response from SQLSERVER.
Response : Result / Output

Ex: SSMS client tool.(sqlserver management studio)

ii) Server:
========

- there are two sub-components in server.

i) Instance
ii) Database

i) Instance:
=========

- it is a temporary memory which was allocated from RAM.
- data can be stored temporarly.

ii) Database:
==========

- it is a permanent memory which was allocated from harddisk.

- data can be stored permanently.

NOTE:
=====

- when we want to work on sqlserver database then we need to follow the following two steps procedure.

Step1: Connect
Step2: Communicate

Connect:
=======

- whenever user want to connect to sqlserver then we required a DB tool is known as "SSMS".

Communicate:
============

- when we want to communicate with database then we required a DB language is called as "SQL".

Steps to connect SQLSERVER:
=========================

```
> go to all programs
        > to microsoft sqlserver folder.
                > click on sqlserver management studio.
                        > open connet to server window.
                                Servertype   : Database Engine
                                ServerName : system name / local host / .
                                Authentication : Windows / SQLServer
                                    Username  : sa  ( for sqlserver authentication)
                                    Password   : 123
                        > click on connect button.
```

How to create a new database in SQLserver:
=====================================

syntax:
=======

```
CREATE DATABASE <db name>;
```

EX:

```
> go to open "new query" editor and write the following statement.

SQL> CREATE DATABASE MYDB6PM;
```

Commands completed successfully.

==
======

(04-10-2024)   Topic - 3 : SQL
==========   =============

Introduction to SQL:
=================

- SQL stands for "structure query language" which is used to communicate with any relational databases like sqlserver,oracle,mysql,db2,postgresql,..................etc.
- SQL was introduced by IBM.the initial name is "SEQUEL" and later renamed as "SQL".
- SQL is not a case-sensitive language i.e user can write SQL queries in either lower case / upper case / combination of lower and upper case characters.

Ex:

```
SELECT * FROM EMP;------>executed
select * from emp;------------> executed
SeleCT * From Emp;---------> executed
```

- Every sql query should ends with " ; " but it is a optional in SQLServer.

Sub-Languages of SQL:
===================

1. Data Definition Language(DDL):
============================

> CREATE
> ALTER
> SP_RENAME
> TRUNCATE
> DROP

2) Data Manipulation Language(DML):
=================================

> INSERT
> UPDATE
> DELETE

3) Data Query / Retrieval Language(DQL / DRL):
==

> SELECT (read only)

4) Transaction Control Language(TCL):
==================================

> COMMIT
> ROLLBACK

> SAVEPOINT

5) Data Control Language(DCL):
========================

> GRANT
> REVOKE

==
=====

1. Data Definition Language(DDL):
==========================

CREATE command:
===============

- to create a new database object such as Tables,Synonyms,Views,Indexes,......etc.

How to create a new table in sqlserver:
===============================

syntax:
======

create table <table name>(<column name1> <datatype>[size],<column name2> <datatype>[size],................);

Datatypes in SQLSERVER:
===================

- it is an attribute to specify what type of data is storing into a column in the table.
- sqlserver supports the following datatypes are,
    > Integer datatypes
    > Decimal datatype
    > Character / String datatypes
    > Date & Time datatypes
    > Money datatypes
    > Binary datatypes
    > Special datatypes

Integer datatypes:
===============

- storing integer values only.
- as per the size and range integer datatypes are again classified into 4 types.

- Tiny int	- 1 byte
- Small int	- 2 bytes
- Int	- 4 bytes
- Bigint	- 8 bytes

Decimal datatype:

===============

- storing decimal / float values only.
- this datatype is having two arguments those are "Precision & Scale".

i.e Decimal(P,S)

What is Precision:
===============

- counting all digits including left and right sides of a decimal point in the expression.
- the default value of precision is 18 and maximum size is 38 digits.

Ex:

i) 56.24
precision = 4

ii) 56737.352
precision = 8

What is Scale:
============

- counting the right side digits of a decimal point from the given expression.

Ex:

i) 56.24
precision = 4
scale = 2

ii) 56737.352
precision = 8
scale = 3

Ex:

```
            PRICE decimal(8,2)
            ================
        0.0 ---------->   0.0
        25.23------->    25.23
        999999.99------->       999999.99
        1000000(1000000.00)--> error
```

05-10-2024:
===========

Character / String datatypes:
=======================

- storing string format data only.
- in database string can be represent with '<string>'.

string format data
||

characters only string	alphanumeric string
[A-Z (or) a - z]	[A-Z / a-z , 0-9 , @,#,$,%,&,_,.......]
Ex: 'smith','SMITH',........etc	Ex: 'smith123@gmail.com',PANCARD,PASSWORD,HTNO,...etc

Types of chracter / string datatypes:
==============================

- there are two types of character / string datatypes.

1. Non-unicode datatypes :
=======================

- to store "localized data"(i.e English Language only)

i) char(size)
ii) varchar(size / max)
iii) text

2. Unicode datatypes:
==================

- to store "globalized data"(i.e All National Languages)

i) Nchar(size)
ii) Nvarchar(size/max)
iii) Ntext

- "N" stands for National language.

i) char(size):
===========

- it is fixed length datatype(i.e static datatype).
- it will store non-unicode characters in the form of 1 char = 1 byte.
- the maximum length of char datatypes is 8000 character(8000 bytes).

Disadvantage:
============

- Memory wasted.

ii) varchar(size/max):
==================

- it is variable length datatype(i.e dynamic datatype).
- it will store non-unicode characters in the form of 1 char = 1 byte.
- the maximum length of varchar(size) is 8000 chracters(8000 bytes)
- the maximum size of varchar(max) is 2gb.

Advantage:
============

- Memory saved.

iii) text:
=======

- it is similar to varchar(max) datatype.
- maximum size is 2gb.

i) Nchar(size):
===========

- it is fixed length datatype(i.e static datatype).
- it will store unicode characters in the form of 1 char = 2 bytes.
- the maximum length of char datatypes is 4000 character(8000 bytes).

Disadvantage:
============

- Memory wasted.

ii) Nvarchar(size/max):
==================

- it is variable length datatype(i.e dynamic datatype).
- it will store unicode characters in the form of 1 char = 2 bytes.
- the maximum length of Nvarchar(size) is 4000 chracters(8000 bytes)
- the maximum size of Nvarchar(max) is 2gb.

Advantage:
============

- Memory saved.

iii) Ntext:
=======

- it is similar to Nvarchar(max) datatype.
- maximum size is 2gb.

07-10-2024:
===========

Date & Time datatypes:
===================

- these datatypes are used for storing date and time information of a particular day.

i) Date:

=======
- to store date information only.
- default date format of sqlserver is 'YYYY/MM/DD'.

ii) Time :
=======
- storing time information only.
- default time format of sqlserver is 'HH:MI:SS.MS'.

iii) Datetime:
===========
- storing date & time information.
- default format of datetime datatype is:
'YYYY/MM/DD HH:MI:SS.MS'
'2024/10/07   18:21:33.998'

Money datatypes:
===============
- storing currency format data.
- it again two types:
> small money ---------- 4 bytes
> money -------------------- 8 bytes

Binary datatypes:
==============
- storing image / audio / video file in the form of "0100101010001" binary format.
- these datatypes are again three types:
i) Binary(size)   : static datatype   : maximum size is 8000 bytes
ii) Varbinary(size/max)   : dynamic datatype : maximum size of
varbinary(size) is 8000 bytes.
maximum size of varbinary(max) is 2 gb.
iii) image   : it is similar to varbinary(max) datatype.
the maximum size is 2gb.

NOTE:
=====
- Instead of text,ntext,image datatypes we will use varchar(max),Nvarchar(max) and varbinary(max) datatypes in the latest versions of sqlserver.

Special datatypes:
================
i) SQL_VARIENT:
==============
- it will store all datatypes data.

- maximum size 8016 bytes.
- it is a dynamic datatype.

Ex:

```
                X sql_varient
                ===========
1021------->    1021
'smith'----->   smith
'DNL234IQ'--->          DBL234IQ
24000.00 ------>        24000.00
'2024/10/07'---->       2024/10/07
'18:39:44.324'--->      18:39:44.324
0101001010------>       0101001010
```

ii) XML :
=======

- it will store .xml file.
- maximum size is 2gb.
- it is a dynamic datatype.

iii) CURSOR:
==========

- it is a referencial datatype for allocating temporary memory for storing a specific table data.

iv) TABLE :
=========

- it will store collection rows & columns of a table.

==
=====

How to create a new table in sqlserver:
==================================

syntax:
======

create table <table name>(<column name1> <datatype>[size],<column name2> <datatype>[size],................);

EX:

Step1: Select the required database from sqlserver:
==

syntax:
=======

USE <DB name>;

EX:

```
USE MYDB6PM;
```

Step2: Create a table:
===================

```
CREATE TABLE STUDENT(STID INT,SNAME CHAR(10),SFEE DECIMAL(6,2));
```

Step3: To view the structure of a table:
================================

syntax:
======

```
SP_HELP <table name>; [ SP_HELP is a pre-defined stored procedure ];
```

Ex:

```
SP_HELP STUDENT;
```

2) ALTER command:
=================

- to change / modify the structure of a table.
- this command is having 4 more sub-commands are:
    i) ALTER - ALTER COLUMN
    ii) ALTER - ADD
    iii) ALTER - DROP
    iv) SP_RENAME

i) ALTER - ALTER COLUMN:
======================

- to change datatype and also the size of datatype of a specific column in the table.

syntax:
======

```
ALTER TABLE <TABLE NAME> ALTER COLUMN <COLUMN NAME> <NEW
DATATYPE>[NEW SIZE];
```

EX:

```
ALTER TABLE STUDENT ALTER COLUMN SNAME VARCHAR(20);
```

ii) ALTER - ADD:
=============

- to add a new column to an existing table.

syntax:
======

```
ALTER TABLE <TABLE NAME> ADD <NEW COLUMN NAME> <DATATYPE>[SIZE];
```

EX:

```
ALTER TABLE STUDENT ADD SADDRESS VARCHAR(50);
```

iii) ALTER - DROP:
===============

- to drop / delete a column from a table.

syntax:
======

```
ALTER TABLE <TABLE NAME> DROP <COLUMN> <COLUMN NAME>;
```

EX:

```
ALTER TABLE STUDENT DROP COLUMN SFEE;
```

08-10-2024:
===========

iv) SP_RENAME:
=============

- it is a pre-defined stored procedure in sqlserver.
- to change a column name / a table name.

How to change a column name:
==========================

syntax:
=======

```
SP_RENAME '<TABLE NAME>.<OLD COLUMN NAME>','<NEW COLUMN NAME>';
```

EX:

```
SP_RENAME 'STUDENT.SNAME','STUDENT_NAME';
```

How to change a table name:
========================

syntax:
=======

```
SP_RENAME '<OLD TABLE NAME>','<NEW TABLE NAME>';
```

EX:

```
SP_RENAME 'STUDENT','STUDENT_DETAILS';
```

TRUNCATE:
==========

- to delete all rows but not columns of a table.
- by using truncate command we cannot delete a specific row from a table because truncate command is not allowed "WHERE" clause condition.

syntax:
======

```
TRUNCATE TABLE <TABLE NAME>;
```

EX:

```
TRUNCATE TABLE STUDENT_DETAILS WHERE STID=1022;------NOT ALLOWED
TRUNCATE TABLE STUDENT_DETAILS;-----ALLOWED
```

DROP:
=====

- to delete / drop a table (i.e collection of rows & columns) from database permanently.

syntax:
======

```
DROP TABLE <TABLE NAME>;
```

EX:

```
DROP TABLE STUDENT_DETAILS
```

2) Data Manipulation Language(DML):
================================

INSERT:
=======

- to insert a new row(i.e data) into a table.

Method-1:
=========

syntax:
======

```
INSERT INTO <TABLE NAME> VALUES(value1,value2,................);
```

Ex:

```
INSERT INTO STUDENT VALUES(1021,'SMITH',5000);
```

- In this method no.of columns in a table and passing value in the query must match.

Method-2:
=========

syntax:
======

```
INSERT INTO <TABLE NAME>(<column name1>,<column name2>,....)
```

```
VALUES(value1,value2,................);
```

Ex:

```
INSERT INTO STUDENT(SNAME)VALUES('ALLEN');
INSERT INTO STUDENT(STID,SFEE)VALUES(1023,2500);
INSERT INTO STUDENT(STID,SNAME,SFEE)VALUES(1024,'MILLER',4800);
INSERT INTO STUDENT(SNAME,SFEE,STID)VALUES('JONES',8200,1025);
```

- In this method user can insert values for required columns only.

How to insert multiple rows into a table:
===================================

syntax-1:
========

```
INSERT INTO <TABLE NAME> VALUES(row1 values),(row2 values),................;
```

Ex:

```
INSERT INTO STUDENT
VALUES(1026,'SCOTT',4500),(1027,'ADAMS',6000),(1028,'WARD',2500);
```

syntax-2:
========

```
INSERT INTO <TABLE NAME>(<column name1>,<column name2>,.......)VALUES(row1
values),(row2 values),................;
```

Ex:

```
INSERT INTO STUDENT(STID)VALUES(1029),(1030),(1031);
```

17-10-2024:
==========

UPDATE:
========

- to update all rows data in a table at a time.

(or)

- to update a specific row data in a table by using "WHERE" clause condition.

syntax:
======

```
UPDATE <TABLE NAME> SET <COLUMN NAME1>=<VALUE1>,<COLUMN
NAME2>=<VALUE2>,
..................[WHERE <condition>];
```

EX:

```
UPDATE STUDENT SET SFEE=5000 WHERE SNAME='SCOTT';
UPDATE STUDENT SET SNAME='WARNER',SFEE=1500 WHERE STID=1028;
UPDATE STUDENT SET SFEE=NULL WHERE SFEE=8200;
UPDATE STUDENT SET STID=NULL,SNAME=NULL,SFEE=NULL WHERE STID=1027;
UPDATE STUDENT SET STID=1027,SNAME='WARD',SFEE=3000 WHERE STID IS NULL;
```

EX:

```
UPDATE STUDENT SET SFEE=NULL;
UPDATE STUDENT SET SFEE= 5000;
```

DELETE:
=======

- to delete all rows from a table at a time.

(or)

- to delete a specific row from a table by using "WHERE" clause condition.

syntax:
=======

```
DELETE FROM <TABLE NAME> [ WHERE <condition> ];
```

EX:
EX:

```
DELETE FROM STUDENT WHERE STID=1025;
DELETE FROM STUDENT WHERE SNAME IS NULL;
DELETE FROM STUDENT;
```

DELETE vs TRUNCATE:
===================

DELETE	TRUNCATE
=======	==========
1. it is a DML operation.	1. it is a DDL operation.
2. supporting to delete a specific row from a table.	2. not supporting to delete a specific row from a table.
3. it supports "WHERE" clause.	3. id does not support "WHERE" clause.
4. deleting rows from a table temporarly.	4. deleting rows from a table permanently.
5. we can restore deleted data into a table by using "ROLLBACK".	5. we cannot restore deleted data into a table by using "ROLLBACK".

6. it will not reset an identity values.	6. it will reset an identity values.
7. the execution speed is slow. (deleting rows one-by-one)	7. the execution speed is fast. (deleting rows as a page)

Data Retrieval / Query Language(DRL/DQL):
=======================================

SELECT command:
================

- to retrieve all rows data from a table at a time.

(or)

- to retrieve a specific row data from a table by using "WHERE" clause condition.

syntax:
======

```
SELECT * FROM <TABLE NAME> [ WHERE <condition> ];
	Here,
		" * "  ------> all columns in a table.
```

EX:

```
SELECT * FROM EMP;
SELECT * FROM EMP WHERE EMPNO=7566;
SELECT * FROM EMP WHERE JOB='MANAGER';
SELECT EMPNO,ENAME,SAL FROM EMP WHERE ENAME='SMITH';
SELECT * FROM EMP WHERE COMM IS NULL;
```

Alias name:
==========

- it is a temporary name / alternate name for columns / table.
- we can create alias names at two levels.

i) Column level alias:
=================

- creating alias name for columns.

ii) Table level alias:
================

- creating alias name for table.

syntax:
======

```
SELECT <column name1> [as] <column alias name1>,..........FROM <table name> [as] <table alias name>;
```

EX:

```
SELECT DEPTNO AS X,DNAME AS Y,LOC AS Z FROM DEPT AS D;
                    (OR)
SELECT DEPTNO  X,DNAME  Y,LOC  Z FROM DEPT  D;
```

DISTINCT keyword:
================

- to eliminate duplicate values from a specific column temporarily.

syntax:
=======

distinct <column name>

Ex:

```
SELECT DISTINCT JOB FROM EMP;
SELECT DISTINCT DEPTNO FROM EMP;
```

==
==