

# **SQL Server**

**By**

## **Mr.Sudhkar L**

### **Naresh Technology**

**SRI RAGHAVENDRA XEROX**

*Software Languages Material Available*

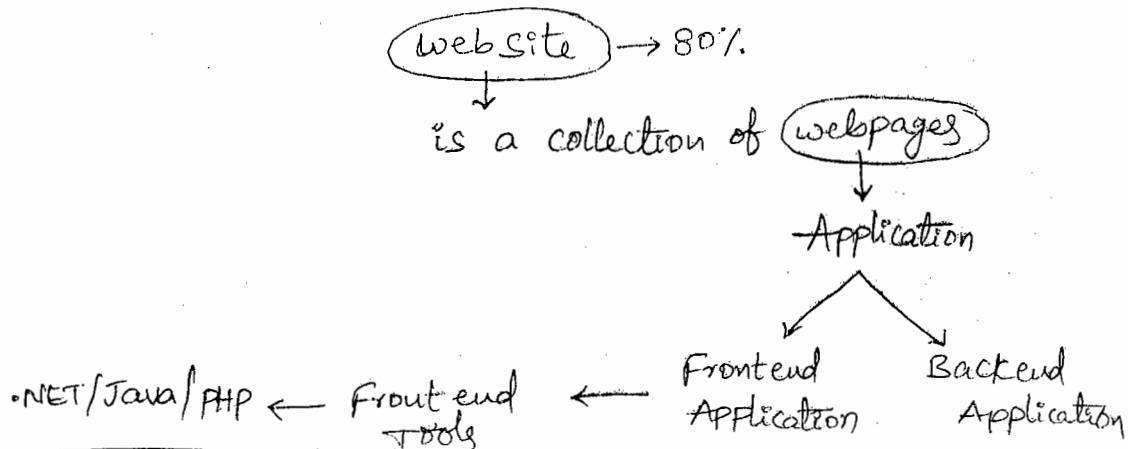
Beside Bangalore Ayyangar Bakery, Opp. C DAC, Ameerpet, Hyderabad.

**Cell: 9951596199**



# SQL

Date: 30-09-15  
Wednesday



## Introduction to DBMS:

Why DBMS :- Generally a person has to interact with huge amount of information every day. In order to interact with huge amount of information we need a system where we can store manipulate providing security and sharing that information all over the world. For this reason the organizations will maintain Database Management system.

Data: Data is nothing but the Rawfact whereas Rawfact is a collection of numbers, characters, special characters, symbols, figures. (or)

whatever we input from the keyboard is known as data.

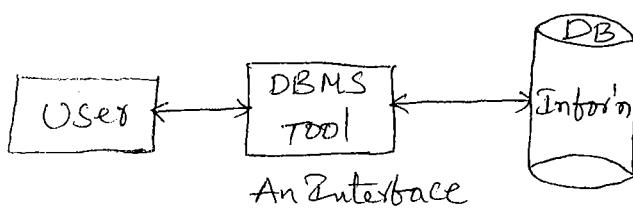
→ Data never provide the meaningful statements to the user.

Information:- processing the data is called as information.

→ Information always provide the meaningful statements to the user.

**DataBase** :- It is a collection of information that has been written in pre-determined manner and stored into a particular location is called as Database.

**DBMS** :- DBMS is a software or tool which can be used to maintain and manage the data within the database in the server.



Here, DBMS Tool will act as an interface between the user and database.

**Models of DBMS** :- DBMS contains the following 6 models

1. File Management System (FMS)
2. Hierarchical Management System (HMS)
3. Networks Database Management System (NDBMS)
4. Relation Database Management System (RDBMS)
5. Object Relation DBMS (ORDBMS)
6. Object oriented Relation DBMS (OORDBMS)

02-10-2015, Friday

1. File Management System (FMS) :- FMS is a first model of DBMS which was introduced in 1950's.  
→ In this model the data will be stored in the form of sequential or continuous stream of characters manner.

KOREK ARQHVAHAR IRB  
The first model emerged in 1950's  
and used sequential stream  
of characters to store data

101SA125000 23/08/15 Madapur 102Kamal 13000  
20/09/15 Ameerpeta 103 Sriram 24000 28/09/15 S.R.Nagar 104  
Ramu 14000 20/10/15 Balangir - - - - -

- - - - - 110Hari 38000 11/11/2015 Barakampeta

### Drawbacks:

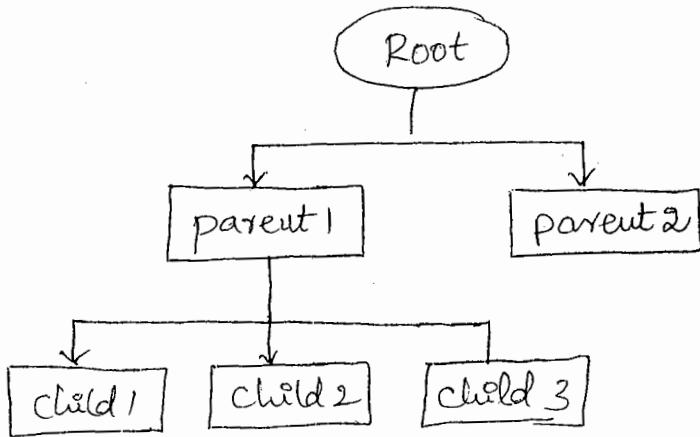
- There is ~~of~~ no security to the information within the files.
- Retrieving the data is time consuming because every required information purpose the user has to perform search operation on a file.
- Inconsistency of the data (There is no proper order).
- This model doesn't support Datatypes mechanism.
- We cannot share the required information to the multiple users.

2. Hierarchy Management System (HMS) :- It is the second model of DBMS which was designed and developed by the IBM Company when the web developing a project is known as Information Management System (IMS) in 1960's.

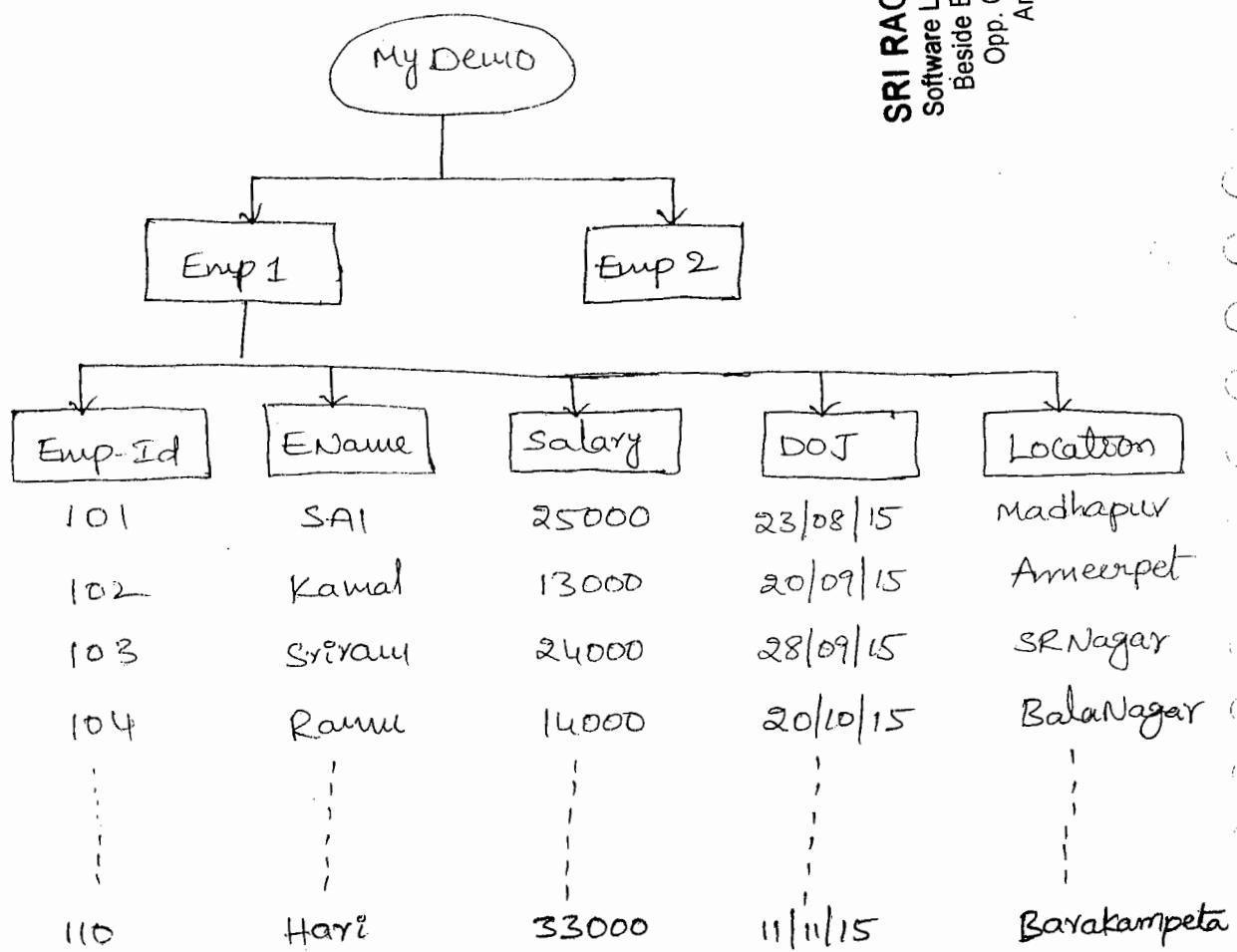
→ In this model the data will be stored in the form of Tree structure and it contains the following levels Those are

1. Root level will Represent Database Name.
2. parent Level will Represent Tables Name.
3. child Level will Represent columns Name.

Ex:-



**SRI RAGHAVENDRA XEROX**  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.



### Advantages:

- The main Advantage of HTMS model means it retrieving the data is faster than FMS model.
- The Data will be maintained in consistency manner (i.e. proper order).

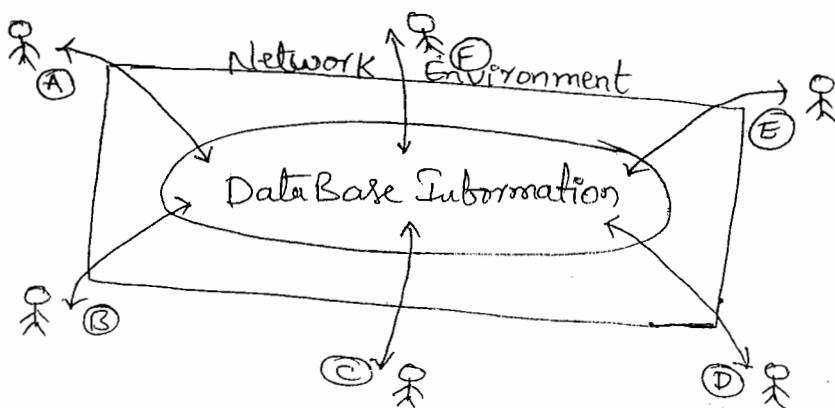
→ This Model is Supported Datatypes Mechanism (Integer, Char, Decimal, Date and Time Datatypes etc).

### Drawbacks:

- In this Model only one person can interact with the Database at a time.
- whenever we add a new value to add existing structure Then the user has to reconstruct the entire structure. so it leads the time consuming and also burden of the programmer.
- This model is also not supported sharing that information to the multiple users.

3. Networks DataBase Management system (NDBMS): It is the Third Model of DBMS which was designed and developed by the IBM Company in 1969.

- In this model the data will be stored in the form of Tree structure and integrated with Networks.



- The Main Advantage of NDBMS is to share the required information to the multiple users and also multiple programmers can interact with the same Database at a time.

05-10-15, Monday

### DISAdvantages:-

1. There is no proper security to the centralized database info's under the N/W's.
2. Database Redundancy will be increased. (Increasing duplicate values).
3. It occupies more Memory.
4. Once The memory will be occupied then the Server burden will be increased.
5. Application performance is Reduced.
6. User will get delay Responses from the Database Server.

### Note:-

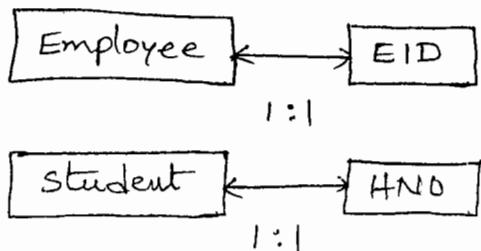
→ The Above 3 models (FMS, HIMS, NDBMS) are outdated.

### 4. Relational DataBase Management System (RDBMS):

- In RDBMS 'are' stands for Relation which can be defined as the commonness between the objects.
- This Relation again classified into 3 types.

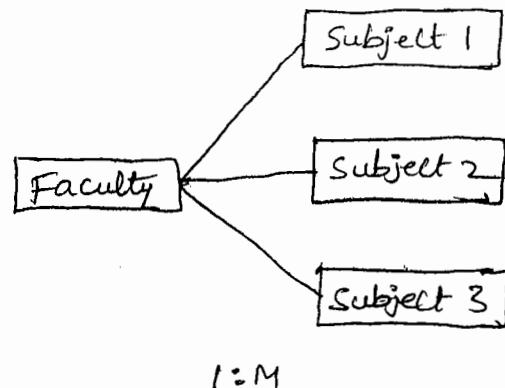
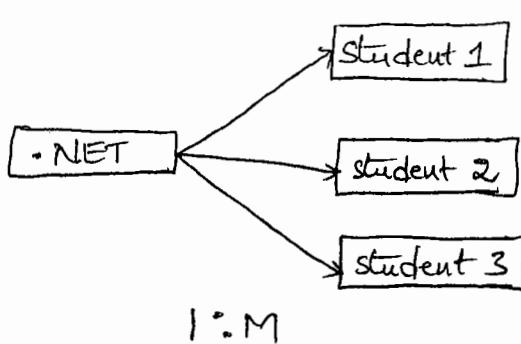
1. One-one Relation:- A Row in a Table is associated with a Row in an other table is known as one-one Relation.

Ex:-



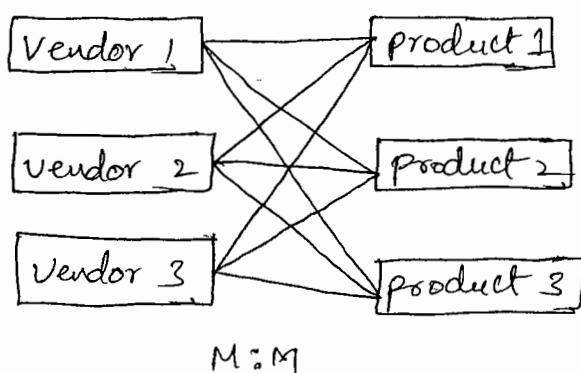
2. One-to-Many :- A Row in a Table is associated with The Many Rows in an other Table is known as one-to-many or Many-to-one Relation.

Ex:-



3. Many-to-Many :- Many Rows in a Table is associated with Many Rows in an other Tables is known as Many-to-Many Relationship.

Ex:-



SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

→ The Above 3 Relationships are called as Degree of Relationship which was designed by the German Scientist Mr. E.F. Codd based on a Mathematical concept is called as Relational-Algebra in 1970.

## Features of RDBMS :-

- ① → In this model the data will be stored in the form of Table format.
- A Table is a collection of Rows and columns. The Horizontal lines are called as Rows (or) records (or) Tuples and the vertical lines are called as columns (or) Attributes (or) Fields.
- An intersection with the between a Row and a column is known as Cell. where as cell is nothing but where we can store the Original Data.
- ② → RDBMS products will provide the High level Security to the DB Information with the Help of Authentication and Authorization modes.
- ③ → To Maintained Accuracy and Consistency of the data in the Database Tables by using Data Integrity mechanism. (Using ~~not~~ Constraints and Triggers).
- ④ → easily Accessing the data from the Database Tables with the Help of Indexes mechanism.
- ⑤ → Share in the Database Tables to the multiple programmers by using Synonyms and views objects.
- ⑥ → providing Re-usability Facilities with the help of Stored procedures and stored functions.
- ⑦ → sharing the Required Database Information from one location to other location or else one system to another system with the help of data files (.Mdf and .Ldf files) and also Backup files (.bak file)

⑦ The father of the RDBMS is Mr. EF. Codd.

\* Examples of RDBMS:-

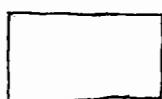
- |              |               |
|--------------|---------------|
| ① SQL Server | ⑤ db2         |
| ② oracle     | ⑥ Informatica |
| ③ mySQL      | ⑦ Ingress     |
| ④ Sybase     | ⑧ Informix    |
|              | ⑨ Teradata.   |

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

06-10-2015. Tuesday

E-R Model :- Before developing the original DB App'n first a DB programmer or Architec will Analyze Manual DB Information (Client requirements) later they will converted into the pictorial format.

→ This pictorial representation is called as E-R Diagram.  
→ Before we want to design the E-R Diagrams we should follow the following symbols. Those are



→ It will represent a Table name or Entity name.



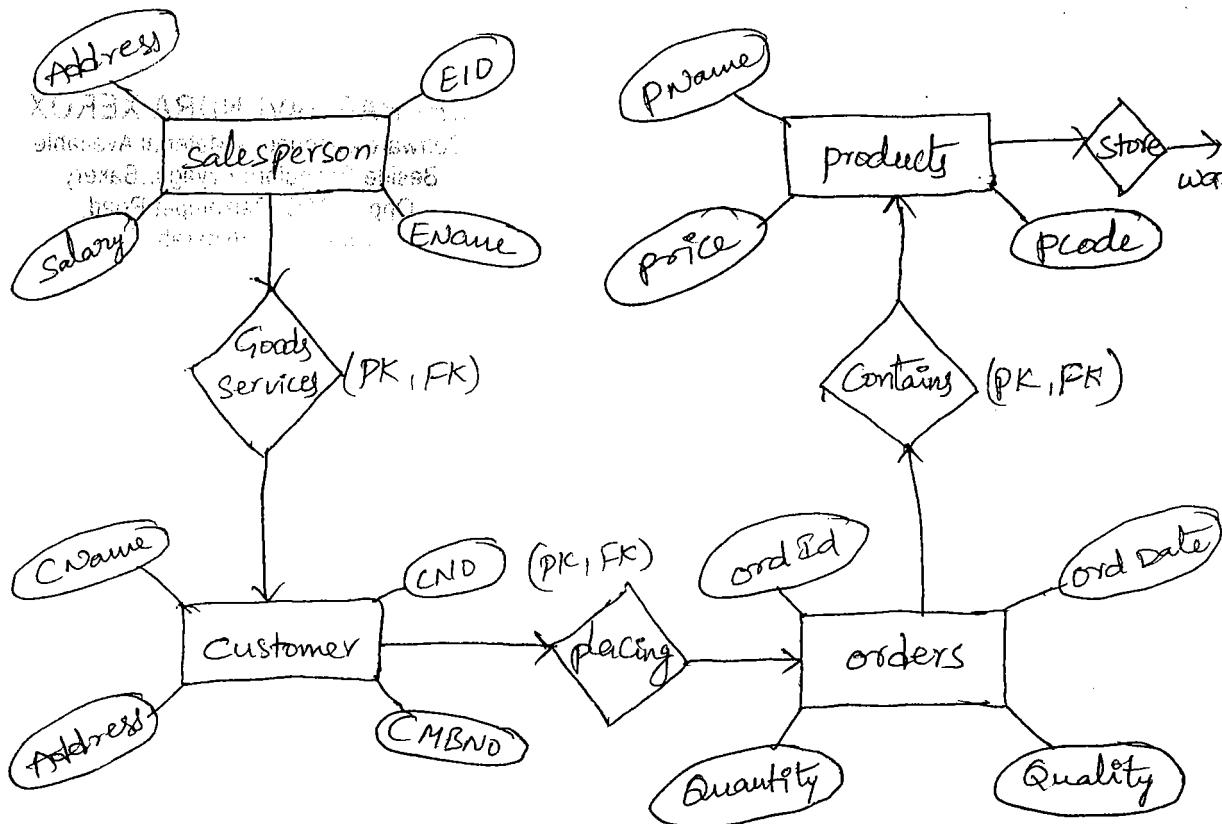
→ It will represent the column names



→ It will represent the relationship between the Entities and

→ → It will represent connecting to the Tables or Entities.

E-R Design a E-R Diagram to represent the Relationship between salesperson and customer.



07-10-2015,

07-10-2015,  
Wednesday

## SQL Server

### Introduction to SQL Server:-

- SQL Server is an RDBMS product which was designed and developed by the Microsoft Company in 1989.
- SQL Server is a Backend tool which is used to design and develop the Backend applications for storing some business related information of a particular organization.
- SQL Server will run or installed on windows operating system only.
- It is a complete platform dependent (only one operating system is supported → Windows)
- SQL Server will provide more Graphical User Interface (GUI) facilities, it means that the user can interact with database without remember in Queries or Commands.
- The first version of SQL Server is 1.0 which is released in 1989. The complete versions of SQL Server

Versions	Year	Release Name	Code Name
1.0	1989	SQL Server 1.0	-
1.1	1991	SQL Server 1.1	-
4.2	1993	SQL Server 4.2	SQ LNT
6.0	1995	SQL Server 6.0	SQ L95
6.5	1996	SQL Server 6.5	Hydra
7.0	1998	SQL Server 7.0	Sphinx
8.0	2000	SQL Server 2000	Shiloh
9.0	2005	SQL Server 2005	YUKON
10.0	2008	SQL Server 2008	Katmai

10.5	2010	SQL Server 2008 R2	Kilimanjaro
11.0	2012	SQL Server 2012	Denali
12.0	2014	SQL Server 2014	SQL 14.

### \* Working with SQL Server:

→ SQL Server is a collection

of databases whereas  
Database is a collection  
of objects like Tables,  
synonyms, Views, procedures,  
Triggers, Functions  
Indexes etc.

Store information

Collection of objects  
↓  
Tables  
Synonyms  
Views  
procedure  
functions  
Triggers  
Indexes



→ SQL Server is also called as  
database environment

where we can create the Backend Applications.

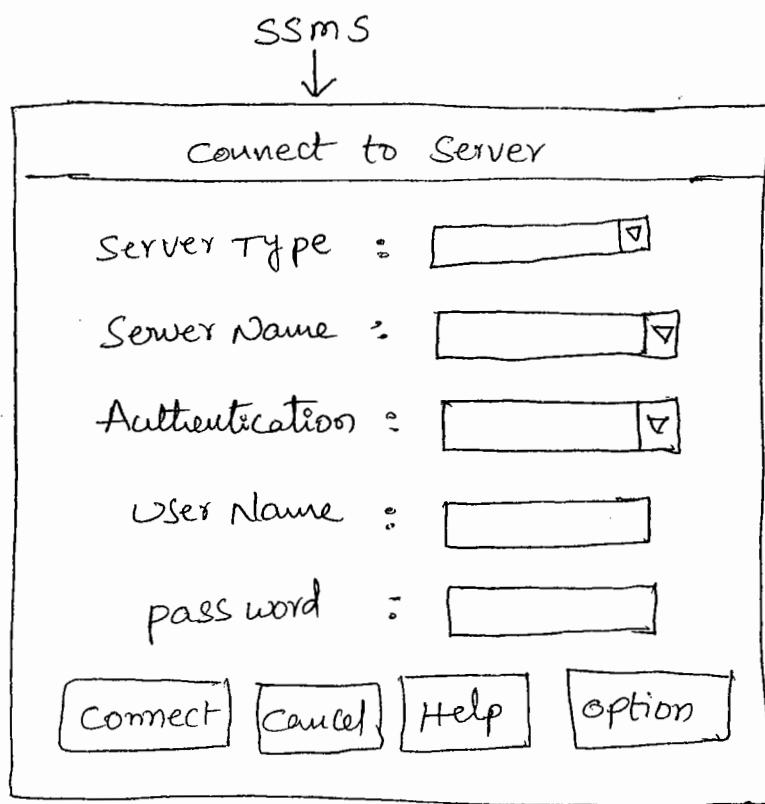
→ we can create maximum 32767 databases under a single instance of SQLserver.

→ when we perform work on SQL Server we need a tool is called as "SSMS" tool. which will be get when we installed SQL Server software in the system.

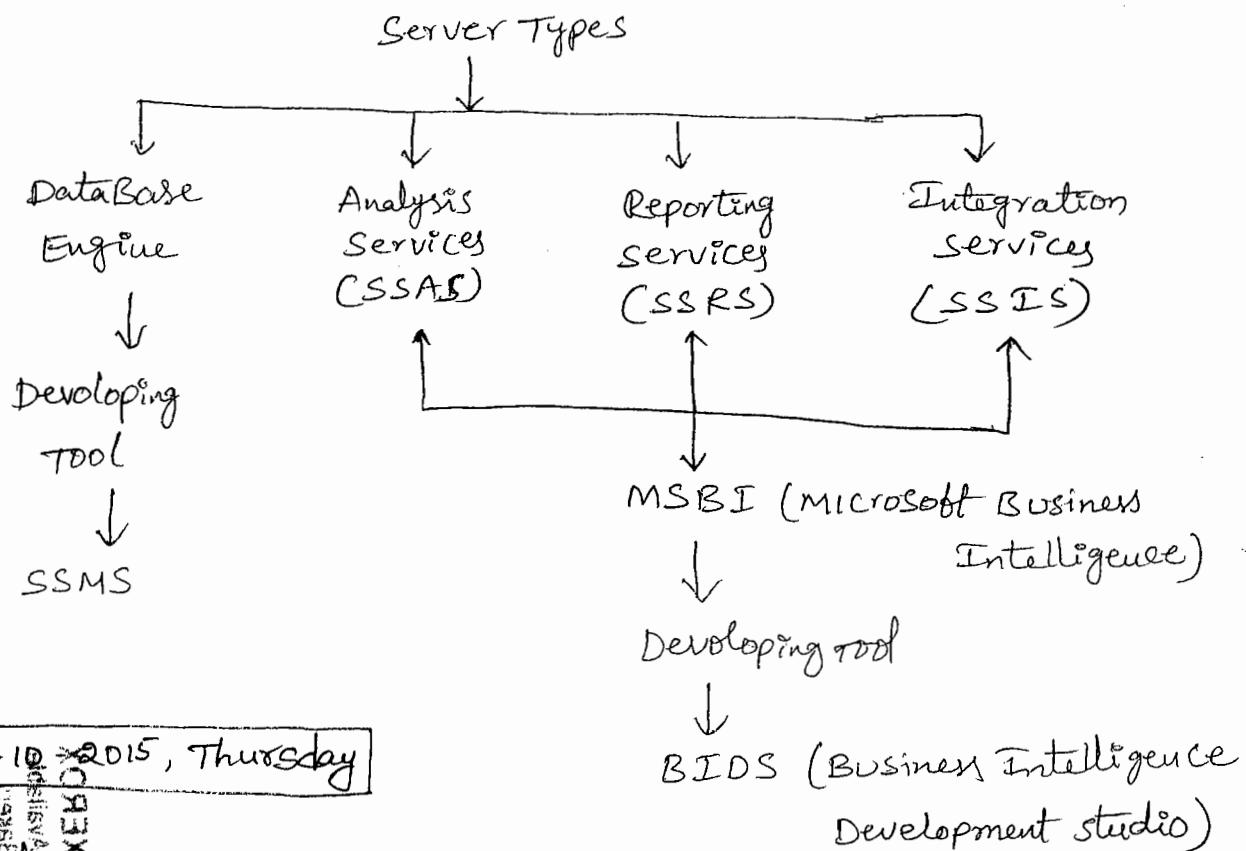
SSMS: It stands for SQL Server Management Studio which will provide two types of environments those are

1. Character Based Environment (CUI)
2. Graphical Based Environment (GUI).

- The above 2 environments are using to perform some database operations like creating database objects, Altering the DB objects, droping the DB objects, inserting data, update data, deleting data, selecting data from the DB Tables in server environment.
- \* → Once we click on ssms tool (icon) we get a window is called as Connect to Server. And it contains the following options Those are



1. Server type: SQL Server contains the following services those are



### \* Database Engine:-

- It is a core component of a SQL Server which can be used for storing the large amount of data, processing the data and providing the security to the information within the server.
- In DataBase Engine the Data will be stored in the form of 2D Format. (That means collection of Rows and columns).

### \* Analysis Service:- (SSAS)

- It is one of the MSBI tool which is used in Datawarehousing environment for storing the information in 3D Format (Graphical Representation of the Data).

### \* Reporting Service (SSRS):-

- SSRS is a MSBI tool which will use to generate or create the various kinds of documents such as MS-word, MS-Excel, .pdf files, .xml files, .~~diff~~ files.
- Report is nothing but a document which is used to store some business related information of a particular organization.

### \* Integration Services (SSIS):-

- SSIS will use to convert from one database table into another database understandable format.
- For example, SQL Server DB table are converting into Oracle DB understandable format.

### 2. Server Name :-

- In which system or machine we installed SQL Server software then the system name will take as a server name for connecting to ~~DB~~ DataBase Server.
- If client and server applications are having within the same machine then we can also use • (or) localhost as a server name.

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

### 3. Authentication :-

- Authentication is nothing but to verify the user credintials (User id, password) before enter into server environment.
- SQL Server contains 2 types of Authentication modes Those are
  1. windows Authentication .
  2. SQL Server Authentication.

09-10-2015, Friday

**Windows Authentication:-** In windows Authentication we will perform on a User Administrator. When we work with windows Authentication there is no required the user-id and password. Because these Id and passwords are created by the operating system by default.

**SQL Server Authentication:-** In SQL Server Authentication we work on the current user. When we work with SQL Server Authentication we should enter user-id and password. These Id and password are created by the user at the time of SQL Server software installation. (At mixed mode option while installing SQL Server software).

### 4. Username:

→ The Default Username of SQL Server is "sa" (System Administrator)

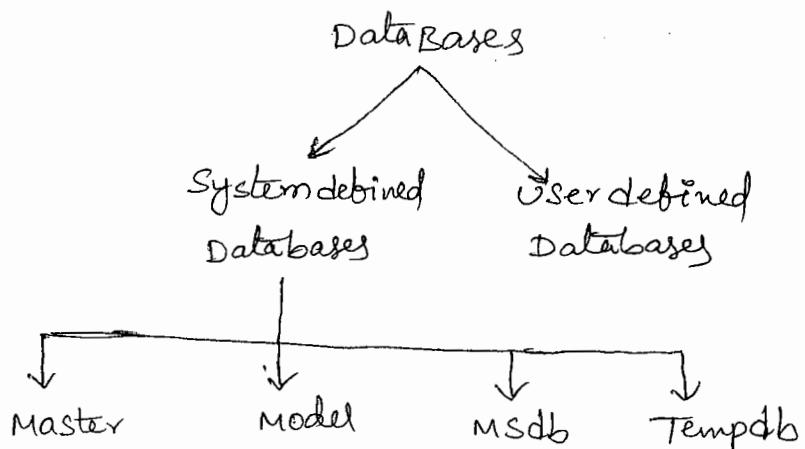
### 5. password:

→ It will create by the user at the time of SQL Server installation.

## Steps to connect to SQL Server:

- Goto All programs
  - Goto Microsoft SQL Server 2014
  - Click on SQL Server 2014 Management Studio option.
  - Select the server type as Database Engine.
  - Enter the server Name as Either systemname/@@1./localhost
  - Select the Authentication either windows or SQL Server Authentication.
  - If we select windows Authentication then there is no required user-id and password.
  - If we select SQL Server Authentication then we should enter user-id and password.
  - Click on Connect Button.
- Once we connect to SQL Server we will get a window is named as Object Explorer window.
- Object Explorer:-** It will provide Graphical User Interface facilities to the user to perform some database operations like creating a new database, a new Table, inserting the data, updating the date, deleting the data, selecting the data from the Database in Server Environment.
- Object Explorer contains the following options those are Databases, security, Server objects, Replication, management etc.

**Databases:** SQL Server contains two types of databases  
those are



10-10-2015, Saturday

### System Databases:-

- These databases are managed and maintain the system by default.
- These are 4 types

**1. Master** → It will store all system level information of SQL Server. Here system level information is nothing but system configuration setting details, all databases details, all users details, login details and object details.

a. To view configuration details :

~~Select \* from SysConfigures~~

SELECT \* FROM SYSCONFIGURES

b. To view All databases Details

SELECT \* FROM SYSDATABASES

c. To view All users Details

SELECT \* FROM SYSUSERS

master  
model  
msdb  
tempdb  
temp  
msdbtemp  
modeltemp  
mastertemp

d. To view All Login Details

```
SELECT * FROM SYSLOGINS
```

e. To view All system Define objects.

```
SELECT * FROM SYSOBJECTS
```

I.W 2. Model ➔ It will work as a Template for Creating The New DataBases in Server Environment.

IV 3. MSdb ➔ It stands for Microsoft DataBase which will store Alerts and Back-up file information.

V 4. Tempdb ➔ It is a Temporary Database memory which will allocate by the System when the user Connect to SQL Server Database.

→ In This Temporary memory the System will store Temporary Tables, views, procedures, Functions, Triggers, objects etc.

→ Once the user Disconnected to SQL Server Database Then The Temporary database Memory will be destroyed automatically.

Ex:- CURSORS.

User-Defined DataBases :-

→ These are created by the User for storing Some business related information.

Ex:- Employee Details,

Department Details,

product Details .

→ The user can create ~~the~~ a Database by using 2 environments Those are ① character Based Environment (CUI)

② Graphical Based Environment (GUI)

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Creating a user defined database by using character Based Environment :→

→ Before creating a user defined database with GUI Environment we need an Editor which will get when we click on New Query option in SQL Server Environment.

Query Editor:

→ It is an Editor which will use to write Queries or commands and also execute the queries or commands in Server Environment.

Syntax: CREATE DATABASE <DBNAME>

Ex:- ~~CREATE~~ CREATE DATABASE mysample

→ To Select the above query and click on either execute option or press F5 keyword.

Creating a user defined database by using Graphical Based Environment:

→ Open SQL Server Management Studio

→ Go to Object Explorer window

→ Go to Databases folder

→ Click on right mouse button

→ Click on New Database option

→ Enter the Database Name as mysample2 in Database name Text Box control.

→ Click on OK.

→ Now go to Databases folder and check it.

Date: 12-10-2015, Monday

Once we create a Database in Server Environment

Then the system will generate or create the two types of files

Those are called as ① primary Datafile  
② Log Datafile.

1. primary Datafile: This file will store All Tables Data and will save with an extension of .mdf (master Data file).

2. Log Datafile: It will store the Query or Transaction information which was executed by the user on a particular Database in the Server.

→ This file will save with an extension of .ldf (log Data file)

#### Note:

→ The above two datafiles are used for Transferring the required Database information From one System to an other system or ~~or~~ from one location to an other location in Real time.

\* The Root Location of .mdf and .ldf files:

C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLServer\MSSQL\DATA

#### Datatypes in SQL Server:-

→ Datatype is Nothing but an attribute which will specify what type of data inserting in a column by the user.

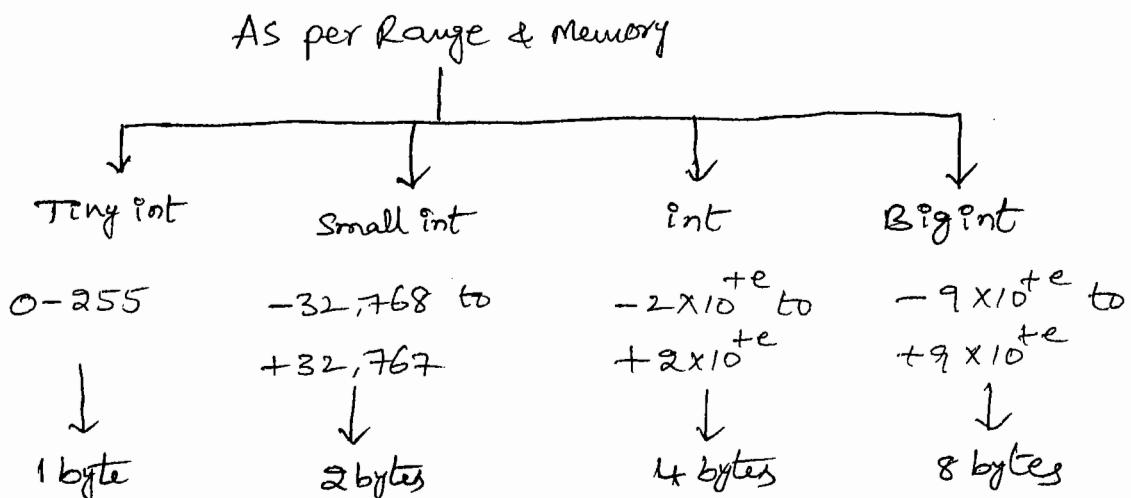
→ SQL Server Supports the following datatypes those are

1. Integer Datatypes.

2. Decimal / numeric Datatypes
3. Money / currency Datatypes
4. Date and Time Datatypes
5. character Datatypes
6. Binary Datatypes.
7. other / special Datatypes.

### 1. Integer datatypes:

- These Datatypes are stored in integer format values only and These can apply applied from EmpId, CACNO, Customer ID, Branch code , column , The Tables etc.
- As for the Range and occupied memory that integer datatypes again classified into 4 types those are



## 2. Decimal/Numeric datatypes:-

→ These are stored the decimal point values only. And will apply ~~on~~ on, price, cost of product, customer balance amount columns in the Table.

→ Decimal datatypes contains 2 types ① Deci

① Decimal (P,S) → New version datatype

② Numeric (P,S) → old version datatype.

But Both are providing will same mechanism.

→ Here P,S are represented precision and scale.

### Precision:

→ It counts left side and right side digits of a decimal point.

→ The default value of precision is 18 and maximum value is 38.

### Scale:

→ It counts the right side digits of a decimal point only.

→ The default value of scale is "0" and it doesn't contain maximum value. That means  $\text{decimal } \cancel{(P,S)} = \text{Decimal}(18,0)$

This is the default value of decimal. →

### Ex. of precision

i) 32.1

precision = 3

### Ex. of scale

i) 32.1

scale = 1

ii) 272.34

precision = 5

ii) 272.34

scale = 2.

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Ex:-

price Decimal (6,2)	
1.1	1.1
23.53 →	23.53
471.817 X	—
4218.72 →	4218.72
32181.81 X	—
2500	2500.00
25000 X	—

precision	stored memory
1 - 9	5 bytes
10 - 19	9 bytes
20 - 28	13 bytes
29 - 38	17 bytes

### 3. Money | currency Datatypes:

- This Datatype is also used to store Decimal format values.
- This Datatype doesn't contain any precision or scale values.
- It is the special datatype which is available in SQL Server RDBMS only product only.
- Money Datatype again classified into 2 types. Those are:

<u>Data type</u>	<u>Range</u>	<u>stored memory</u>
Small Money	-2,147,483,648 to 2,147,483,647	4 bytes
Money	-9,223,372,036,854, 775,808 to +9,223,372,036,854, 775,8087	8 bytes

#### 4 Date and Time DataTypes:

- These are used to store Date and Time information of a particular
- These DataTypes can apply on Employee date of Join, customer account opening date, closing date, product manufacturing date column etc
- These are again classified into the following:

1. Date: It allows Date format information only. The Default format of Date DataType is 'YYYY/MM/DD' or 'YYYY-MM-DD'.  
Ex: '2015/10/13' or '2015-10-13'.

2. Time: It Allows the Time format information only. The Default format of Time DataType is: 'HH:MM:~~MM~~'

Time: 'HH:MM:SS.msec'

Ex: '10:30:47.285'

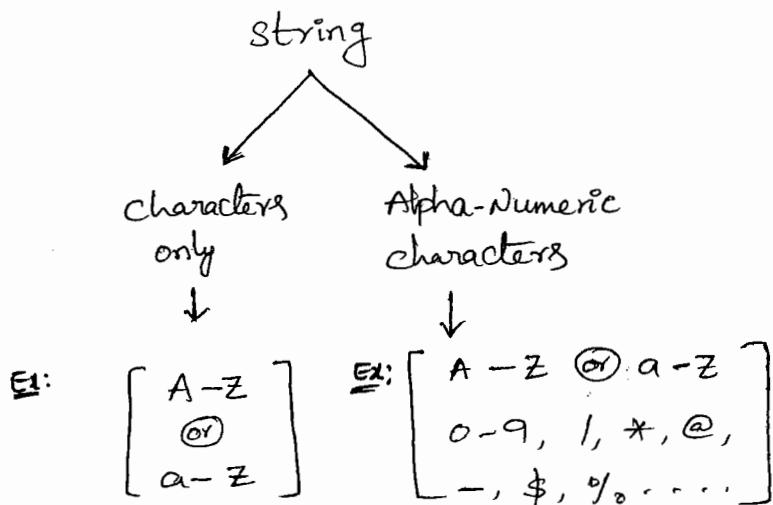
3. DateTime: It allows both time and date information of a particular Table. The default format of DateTime is

'yyyy-MM-DD HH:MM:ss.msec'

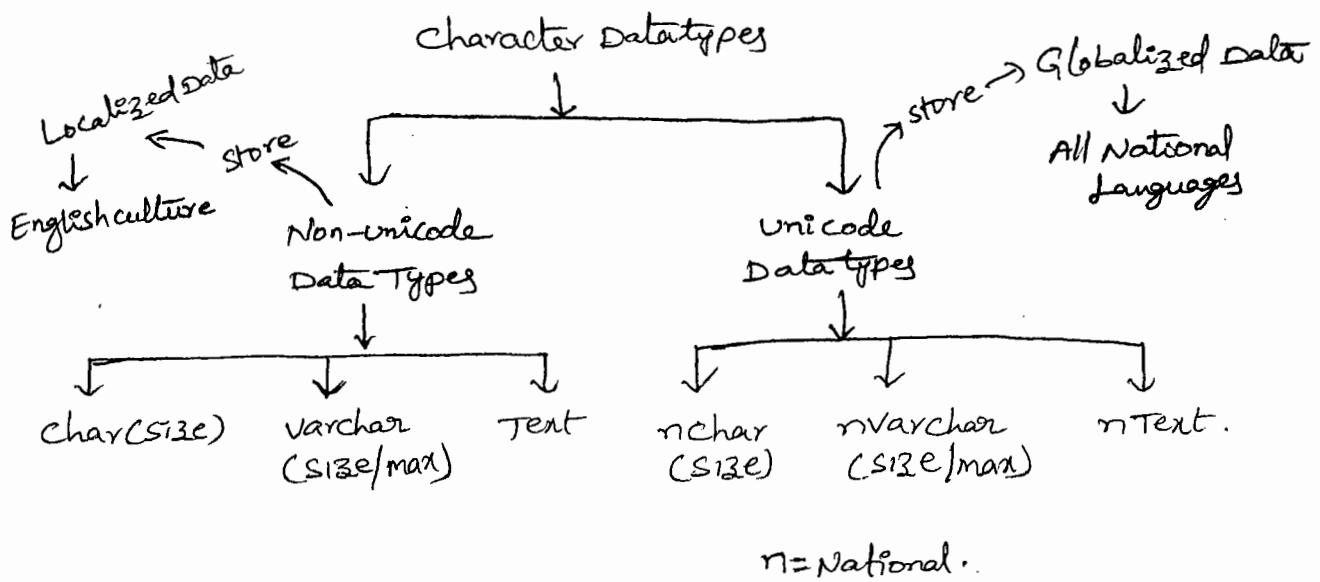
Ex: '2015-10-13 10:30:47.285'.

4. Character Datatypes: It will store the string format values only.

→ Character Datatypes are accepted characters and also Alphanumeric characters.



→ These are again classified into two categories those are:



\* Non-unicode Datatypes: These datatypes are supported to store the localized data.

→ Non-unicode Datatypes are 3 types.

1. `char(size)`: It is the fixed length datatype (static) and will store the characters in Non-code mechanism. That means it will occupy 1 byte for 1 character. The maximum length of `char` datatype is 8000 Bytes.

2. `VARCHAR(size/max)`: It is a variable length datatype (Dynamic) and will store the characters in non-code mechanism. i.e it takes 1 byte for 1 character.

→ The maximum length of `VARCHAR(size)` is 8000 bytes and `VARCHAR(max)` is 2 GB

Date: 15-10-2015, Thursday

\* Differences b/w fixed length Datatypes and Variable Length Data types:

Fixed length DT's

variable length DT's

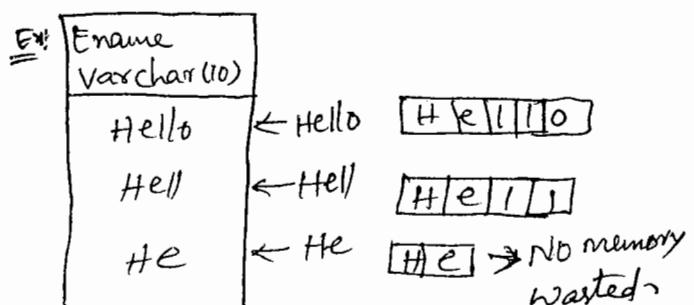
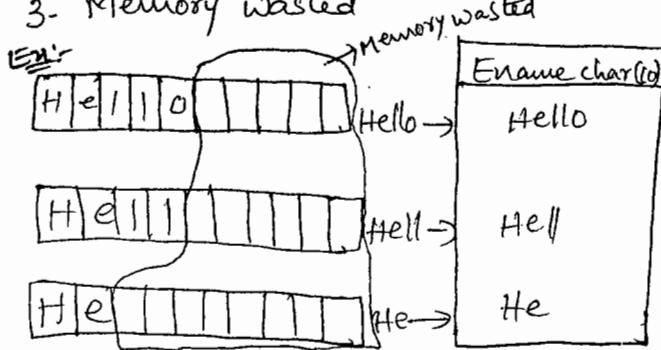
1. It is a static datatype.      ②. It is a Dynamic Datatype.

2. we cannot change the size of data type at runtime level

2. we can change the size of datatype at runtime level.

3. Memory wasted

3. There is No memory wasted.



3. Text :- It will store .txt documents (up to 2 GB file information). It is similar to varchar(max) datatype.

#### \* Unicode Datatypes:

- These DT's are supported to store Globalized data. That means all National languages data format.
- Unicode datatypes are 3 types those are:

1. nchar(size) :- It is the fixed length datatype and will store the characters in unicode mechanism that means it will occupy 2 bytes for 1 character.

→ The Maximum length of nchar datatype is 4000 bytes only.

2. nvarchar(size/max) :- It is a variable length datatype and will store the characters in unicode mechanism that means 1 char = 2 bytes.

→ The maximum length of nvarchar(size) = 4000 bytes and nvarchar(max) = 2 GB.

3. nText :- It will store universal supported Text file information (2GB information). It is similar to nvarchar(max) datatype.

→ Here 'n' will Represent "National".

## 6. Binary Datatypes:

→ These are used to store image files, audio files and video files. These are 3 types.

1. **Binary (size)** → It is the fixed length datatype and will store the file information in Binary format (010101001).  
The Maximum length of Binary (size) = 8000 bytes.

2. **VarBinary (size/max)** → It is a variable length DT and we will store the file information in 0101001 format.  
The Maximum length of VarBinary (size) = 8000 bytes and Varbinary (max) = 2GB.

3. **Image** → It will store all extensions image files like .Jpeg files, .bitmap files, .Tiff file etc. (2GB image files only). It is similar to varBinary (max) datatype.

### NOTE:-

→ Instead of Text, nText, Image datatype we use Varchar(max), nVarchar(max), ~~var~~Binary (max) datatypes in the latest versions of SQL Server software.

→ Old datatype (Available from SQL-1.0 onwards):

Text  
nText  
Image.

→ New Datatypes (Available from SQL-2005 onwards):

varchar(max)

nvarchar(max)

varbinary(max)

### 7) Other or Special Datatypes:

→ SQL Server supports the following Special Datatype.

1. sql\_variant → It will store all datatypes data except Text, ntext and image datatypes.

2. XML → It will store .xml file information (2GB xml files only)

Ex:-

X sql_Variant
101
SAL
39180J
23.52
2500000.00
23/11/2015
10:30:40.248
0101001

3. cursor → It will allocate the reference memory to the cursor object for storing Tables data.

4. Table → It will store the collection of rows and columns

Date: 16-10-2015, Friday

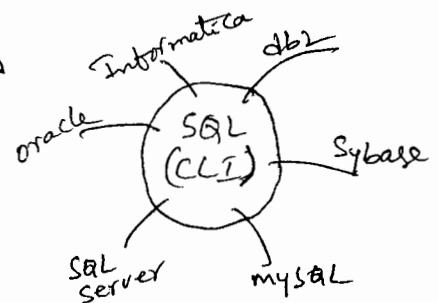
\* Structure Query Language: SQL is ~~not~~ a procedural language which was introduced by the German scientist Mr. E.F. Codd in 1968 and released in ~~to~~ to the market in 1972 with ANSI Approved.

→ SQL is used for communicating with the database in Realtime.

→ SQL is a non-procedural language that means the SQL language doesn't support conditional and control statements (if, else, itelse, nestedif, while, do-while etc)

→ SQL will provide Common Language Interface (CLI) facility.

It means that this is the language which can communicate with any database like SQL Server, Oracle, MySQL, Informatica etc.

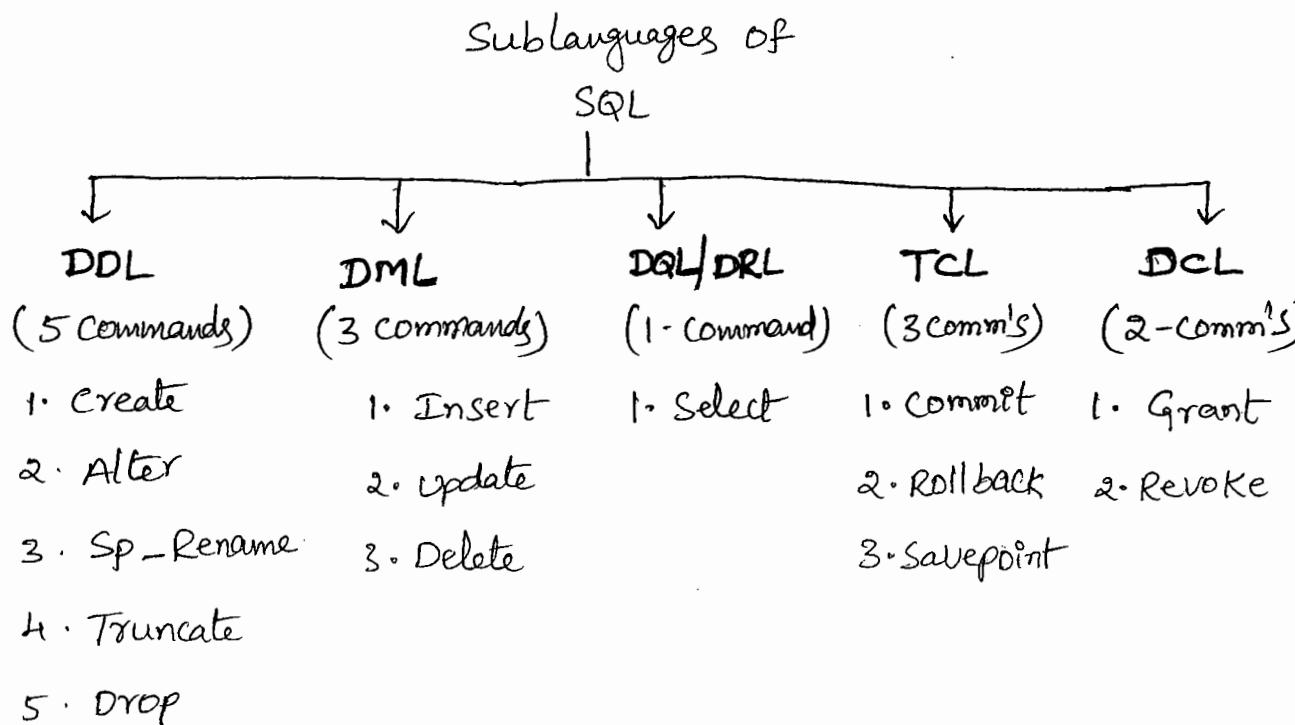


→ SQL pronouns as a sequel as stands for structure English query language.

→ SQL is not a Case Sensitive language (It supports any case characters like Lower case ~~or~~ Upper case)

→ Every SQL statement should ends with a ~~se~~ Semicolon except SQL Server.

→ SQL Contains the following 5 sub-languages those are



1. Data Definition Language (DDL): This language is used for creating a new objects, Altering or modify the structure of the object and droping the objects from database Server.

→ This Language Contains 5 Commands those are:

1. **create** → This Command will use to create a New Tables and also a New Database in the Server.

Syntax: `Create Table <TN> (<column name 1> <datatype> [size], . . . . . <column name n> <datatype> [size])`

Note: In SQL Server every object will be saved with an extension of dbo.<ObjectName>

Ex: dbo.Employee  
dbo.students

\* Create Database MYDB 2 PM

USE MYDB 2 PM

create Table students (STID int, SName char(10),  
Sfee decimal(6,2), Age tinyint)

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

#### \* Rules for creating a New Table:

1. Table names should be unique within the same database
2. column names should be unique within the Table definition.
3. A Table Name should not start with numbers and special characters Except underscore (-) symbol.
4. Dont provide a space in the Table Name. If you want to provide a space in the Table name then we use underscore (-) symbol.
5. A Table should contain minimum one column and maximum 1024 columns. (in oracle 1000 columns only)
6. A Table Name should contain minimum one character and maximum 128 characters. (in oracle min→1 max→30 chars only)
7. Dont use the pre-reserved keywords like Insert, update, delete, select . . . . as a Table Name.  
∴ columns → 1024.  
∴ Rows → Unlimited (oracle also same).

Date: 17-10-2015, Saturday

2. Alter: This Command will use to altering or modify the structure of the Table and also structure of

The Database

→ By using Alter Command we can perform the following 4 operations Those are:

1. Changing datatype and the size of datatype of a particular column in the existing Table.
2. Adding the New columns to add existing Table.
3. Changing the Name of the columns in the Table
4. Dropping unwanted columns from the existing Table.

→ To perform the above 4 operations we need the subCommands of Alter Command.

1. Alter - Alter column:- It will use to change the Datatype from the old datatype to a new datatype and also changing the size of the datatype of a column.

Syntax: Alter Table <TN> Alter Column <columnName> <New Datatypes> [New size];

Ex: Alter Table students Alter Column SName Varchar(50);  
(.. at a time 2 Datatypes are not added)

2. Alter - ADD :- Adding the new columns to an existing Table.

Syntax: Alter Table <TN> ADD <New columnName> <Datatype>  
[size];

Ex:- Alter Table Students ADD SMAILID varchar(50) // Adding Single column.

(∴ Adding column should be place  $\Theta$  Join End of the Table column only)

Ex:2: Alter Table Students ADD SMBNO char(10), SADDRESS  
varchar(40); // Adding multiple columns

3. SP\_Rename :- SP stands for stored procedure. It is a pre-reserved database object which is used to change the Name of the columns and also to change the Name of the Table.

Case 1 :- Syntax for changing column name.

Syntax: SP\_Rename '<TN>. <old columnName>', '<New columnName>';

Ex:- SP\_Rename 'Students . SAddress', 'SADD'

Case 2 :- Syntax for changing in Table Name.

Syntax: SP\_Rename '<old TN>', '<New TN>'

Ex:- SP\_Rename 'Students', 'studentDetails'

4. Alter-Drop :- It will drop unwanted ~~problems~~ <sup>columns</sup> from the Table.

Syntax:- Alter Table <TN> Drop column <column name>;

Ex1:- Alter Table students drop column SMAILID // Dropping a single column.

Ex2:- Alter Table students drop column SMBNO, SADD // Dropping multiple columns.

4. Truncate : It will delete all Rows or Records from the Table at a Time.

- By using Truncate command we cannot delete a specific Record from the Table because Truncate Command is not supported 'where'-Keyword.
- It is a permanent data deletion from the Table (That means we can't restore the Deleted data in to the Table).

Syntax: Truncate Table <TN>

Ex1: Truncate Table students where STID = 102 // Not Allowed

Ex2: Truncate Table students → Allowed. ("Total Table <sup>Rows</sup> Data will be Deleted").

5. **DROP**: It will **DROP** the entire structure of the Table and also the entire structure of the Database Application permanently.

Syntax: `Drop Table <TN>`

Ex : `Drop Table Students`

19-10-2015, MONDAY

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

### \* Creating, Altering and Droping a DataBase:

(i) Creating a New DataBase:- Before creating a New DataBase first we should create .mdf & .ldf files. If we create .mdf & .ldf files we need to follow the following properties: Those are:

1. Name → It will Represent the logical Name of a file (Assign different logical name for every file at the time of creation)

2. File Name → It will Represent the path location of the files (where we store .mdf & .ldf files in the system)

3. Size → It Represent the Initial size of the file.

4. Max Size → It Represent the maximum size of the file.

5. file Growth → It Represent the growth of the file. (1MB, 2MB, 5MB, ...)

Ex: Name = 'xyz',  
File Name = 'E:\....',  
Size = 6 MB,  
Max Size = 10 MB,  
File Growth = 1 MB

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Syntax: Create Database <DB Name> on (<set properties of Data file>) Logon (<set properties of log file>)

Ex: Create Database myDemoDB  
on (Name = 'xyz', filename = 'E:\Datafiles\myDemoDB.mdf', size = 6 MB, maxsize = 10 MB,  
filegrowth = 1 MB)  
Log on (Name = 'ABC', filename = 'E:\Datafiles\myDemoDB\_Log.Ldf', size = 2 MB, Maxsize = 8 MB,  
FileGrowth = 1 MB)

(ii) Altering a Database :- Altering Database is nothing but to modify or change the structure of the Database Application.

A) Adding the New .mdf & .Ldf files to an existing Database (Secondary files) :

Syntax1:- Alter Database <DB Name> ADD File (<set the properties of Data file>)

Syntax2:- Alter Database <DB Name> ADD Log File (<set properties of log file>)

Ex: Alter Database MYDEMO DB

ADD File (Name = 'PQR', filename = 'E:\Datafiles\myDEMO DB1.mdf', size = 10 MB, maxsize = 25 MB  
FileGrowth = 2 MB)

Alter DataBase MY DEMO DB

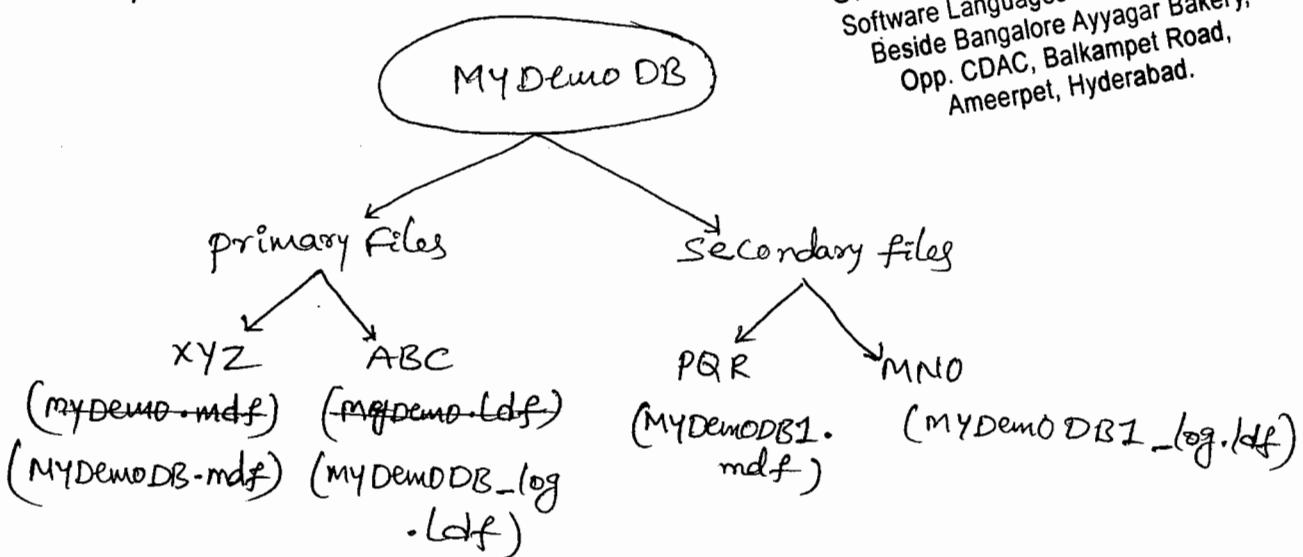
ADD Log file (Name = 'MNO', filename = 'E:\Datafiles\myDEMO DB1\_log.Ldf', size = 8 MB, maxsize = 15 MB  
FileGrowth = 1 MB)

Note: the Above 2 files are called as secondary files  
which are created when the primary ~~memory~~ files  
memory was completed.

→ But the creation of secondary file is just optional.

Q: Can we create morethan 1 .mdf & .ldf file  
on the same DataBase ?

Ans: Yes, we can.



SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

B) Modify the size of .mdf and .ldf files :-

Syntax: Alter Database <DB Name>

    modify file (<set new specifications of data or log files>)

Ex: Alter Database myDemoDB

    Modify File (Name = 'xyz', MaxSize = 50MB)

Alter Database myDemoDB

    Modify File (Name = 'ABC', Size = 10MB, MaxSize = 30MB)

C) Remove .mdf and .ldf files from an existing DataBase:-

Syntax: Alter Database <DB Name>

    Remove File (<Logical Name of the file>)

Ex: Alter Database myDemoDB

    Remove File XYZ or ABC → we cannot Remove  
    the primary data & log files from the DataBase.

~~Alter~~

Ex: Alter Database myDemoDB

    Remove File PQR or MNO → we can Remove  
    the secondary data & log files from the DataBase.

d) modify the Database Name :-

Syntax: Alter Database <Old DB Name>

modify Name = <New DB Name>

Ex: Alter Database myDEMOdb

modify Name = MY ONE

(iii) Drooping a Database: when we drop a database from the Server location then All associated files (.mdf and .ldf files) are dropped permanently.

Syntax: Drop Database <DB Name>

Ex: Drop Database myONE.

20-10-2015, Tuesday

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

\* Data Manipulation Language (DML): This language will use to modify or change the Data in the Table.

→ This Language contains 3 Commands:

1. Insert :- This Command is used to input the New values into an existing Table.

→ By using Insert Command we can insert the values in 2 ways.

(i) Implicit Method → In this method the user has to enter the values for all columns in the Table is called as Implicit method.

Syntax: Insert [Into] <TN> Values (value1, value2, value3, -----)

Ex: Create Table ~~Insert into~~ Students (STID Int, SName Varchar(40), SFee Decimal(6,2), Age TinyInt)

Insert into students (101, 'AA', 2500, 24)

(ii) Explicit method → In this method the user will enter the value for required columns the Table only that means the user can left any column that Table and those columns will take NULL value by default.

Syntax: Insert [Into] <TN> (Required columnNames) values (value1, value2, -----)

Ex: Insert into students (STID, SName) values (102, 'BB')

→ Here 'Into' keyword is just optional but it is mandatory in Oracle.

Ex: Insert students values (103, 'ec', 1300, 22)

Insert students (STID, SFee, Age) values (104, 1400, 24)

\* Syntax for Inserting the multiple Records in to the Table at a Time :

→ Syntax: Insert [Into] <TN> values (Row1 values),  
(Row2 values), ..... ⇒ for Implicit

Insert [Into] <TN> (Required column names)  
Values (Row1 values), (Row2 values), .....  
↳ for Explicit.

Ex: Insert student values (105, 'EE', 7800, 26),  
(106, 'FF', 6900, 25) (107, 'GG', 1100, 28) ⇒ Implicit

Insert students (STID, sname) values (108, 'HH'),  
(109, 'II') (110, 'JJ') ⇒ Explicit.

2. UPDATE: It will update All Records every Time  
and also a specific ~~Record~~ by using 'where' keyword

Syntax: Update <TN> set <column name 1> = value,  
<column name 2> = value, .....  
[where (condition)]

∴ [ ] → is always optional.

1\* WAQ to update student age as 30 whose student Id is equals to 106 ? ( $STID=106$ )

Ans) update students set Age=30 where  $STID=106$

2. WAQ to update student Name as sai , sfee as 5000 whose student age is equals to 24 ? ( $Age=24$ )

Ans. update students set SName='sai', SFee=5000 where Age=24

3. WAQ to update student Age as 25 whose student Age is unknown (Null) [Null=unknown value or undefined value] ?

Ans. update students set Age=25 where Age is Null

[ $\because$  Here we dont use 'Age=null' only using 'Age is Null']

4. WAQ to update all students  $\uparrow$  Age is equals to 21 ? ( $Age=21$ )

Ans. update students set Age=21

3. Delete: It will Delete All Records at a Time  
and ~~also~~ also a specific Record by using 'where' keyword

Syntax: Delete from <TN> [where (Condition)]

1. WAQ to delete student details whose  $STID=104$  ?

Ans. delete from students where  $STID=104$ .

2. WAP to delete students details whose student fee is unknown?

Ans: Delete ~~student~~ from students where sfee is NULL

3. WAP to delete all students details from the Table?

Ans: Delete from students.

\* Differences between Delete and Truncate Commands:

### Delete

1. It is a DML Command
2. It can Delete all Records from the Table at a Time
3. By using Delete command we can delete a specific Record from the Table.
4. It supports 'Where' keyword
5. It is a Temporary data Deletion.
6. we can Rollback (Restore) the deleted data into the Table.
7. It will not Reset and Identity values of a column

### Truncate

1. It is a DDL Command
2. It is also Delete all Records from the Table at a Time.
3. It is not possible to delete a specific Record from the Table.
4. It doesn't Support where keyword.
5. It is a permanent Data Deletion.
6. we cannot Rollback the deleted data into the Table.
7. It will Reset and Identity values of a column.

Date: 21-10-2015, Wednesday

\* Data Query Language (DQL): This Language will use to Retrieve the Data, from an existing Table.

This Language Contains only one command i.e. Select Command.

Select:- It will use to Retrieve or information from the Table. By using Select Command we can retrieve the data from the Tables in 3 ways.

1. projection
2. Selection
3. Joins

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

1. projection:- when we retrieve the data from the Table without any condition is known as projection.

Syntax: Select \* / <List of column Names> from <TN>

1. WAP to display all students details?

Ex: Select \* from students

(Or)

Select STID, SName, SFee, Age from students

2. WAP to display students names and their fee's details?

Ans: Select SName, SFee from students

2. Selection :- When we retrieve the data from the Table based on a condition is known as selection.

Syntax : select \* / < List of column Names > From < TN >  
where (condition)

1. WAG to display student details whose student ID 101 ?

A. select \* from students where STID = 101

2. WAG to display student Name , student fee details whose student age = 25 ?

A. select SName, Sfee from students where Age = 25 .

3. WAG to display students details whose student fee is unknown ?

A. select \* from students where fee is Null .

⇒ Here \* will represent all columns of a Table .

Note : whenever we update, delete and select the data

based on NULL value condition then we use 'is' keyword .

\* Identity (Seed, Increment) : Identity is a function which can be used to generate or create by unique ID values of a particular column automatically .

→ It can be applied on Integer datatype column only .

→ A Table should contain only one Identity column .

→ The Default value of Identity (Seed, Increment) is Identity (1,1)

→ Identity (seed, increment) = Identity (1, 1)

→ Here seed will represent starting or initial value of Id and the default value of seed is 1.

Increment → it will represent the incremental value b/w the Id. The default value of Increment is 1.

\* Example on system defined default seed and Incremental values:

Ex: create Table Employee (EID INT Identity , EName Varchar(30),  
Sal Money)

Insert Employee values (1, 'AA', 25000) → Not Allowed

Insert Employee values ('AA', 25000) → Allowed

Insert Employee (EID, EName, Sal) values (3, BB, 36,000)  
↳ Not Allowed

Insert Employee (EName, Sal) values ('BB', 36,000).  
↳ Allowed.

Output: Identity (1,1) Employee

EID	EName	Sal
1	x	x
2	x	x
3	x	x
4	x	x
5	x	x

Identity (1,1) = Identity (Seed, Increment)

↓  
start(1)

↓  
Increment ①

\* Example on the user defined seed and incremental values:

Ex: Create Table Employee1 (EID Int Identity (100,5),  
EName varchar(30), sal money)

Testing:

Insert Employee1 values ('AA', 25000)  $\Rightarrow$  Allowed

Insert Employee1 (Ename, sal) values ('BB', 36000)  
 $\Rightarrow$  Allowed.

Output: Identity (100,5) Employee1

EID	EName	Sal
100	x	x
105	x	x
110	x	x
115	x	x
120	x	x

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Note: If we want to insert the values into an Identity column in the Table in explicitly then we should follow the following Syntax.

~~Syntax:~~ Set Identity - Insert <TN> off/on

$\Rightarrow$  Here Set will Represent to set the properties of an Identity connection.

OFF → It is a default connection. we cannot Insert a value into an Identity column by explicitly.

ON → we can Insert the values into an Identity column by explicitly.

Ex: 1: set Identity\_Insert Employee ON  
Insert Employee (EID, Ename, sal) values(4, 'DD',  
68000) ⇒ Allowed.

Ex: 2: set Identity\_Insert Employee OFF  
Insert Employee (EID, Ename, sal) values(5, 'EE',  
18000) ⇒ Not Allowed.

Date: 26-10-2015, MONDAY

### \* Functions in SQL Server:

Function is Nothing but to perform some operations and Returns a value to the user.

→ In SQL Server the user will interact with two types of functions.

- ① system or Built-In or pre-defined functions.
- ② User defined functions.

1. System or Built-in functions :→ SQL Server supports some built-in functions those are Numeric (or) Mathematical functions , character (or) string functions Data/Time functions , Aggregative / Group functions .

Syntax: select <Function Name> (value / Expression)

### Numeric (or) Mathematical Functions :-

→ These functions are performing some mathematical operations on the given values.

ABS () :- It returns the positive ~~side to~~ sign values only.

Ex: Select ABS (12) → 12

Select ABS (-12) → 12

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Ceiling () :- It Returns a value is Greater Than or equals to ( $\geq$ ) the given value .

Ex: Select Ceiling (5.0) → 5

Select Ceiling (5.00001) → 6

Select Ceiling (-5.0) → -5

Select Ceiling (-5.00001) → -5

\* **floor()**:- It Returns a value is  $\leq$  the given value.

Ex:- Select floor (5.0)  $\rightarrow$  5

Select floor (5.00001)  $\rightarrow$  5

Select floor (-5.0)  $\rightarrow$  -5

Select floor (-5.00001)  $\rightarrow$  -6

\* **Square()**:- It Returns the "squa"re of given value.

Ex:- Select square (5)  $\rightarrow$  25

\* **SQRT()**:- It Returns the square root value of a given Expression

Ex:- Select SQRT (25)  $\rightarrow$  5

\* **power()**:- It Returns the power of a given value.

Ex:- Select power (2,3)  $\rightarrow$  8

\* **PI()**:- It Returns the constant pi value.

Ex:- Select PI ()  $\rightarrow$  3.14

\* **sign()**:- It Returns +1 for positive values, 0 for Nutral values, -1 for Negative values.

Ex:- Select sign (11)  $\rightarrow$  +1

Select sign (0)  $\rightarrow$  0

Select sign (-11)  $\rightarrow$  -1

\*  $\sin()$  :- It Returns the Trigonometric value of a given degree in Radians format.

Ex: Select  $\sin(0)$   $\rightarrow 0$

Select  $\sin(90)$   $\rightarrow 0.89$ .

Select  $\cos(0)$   $\rightarrow 1$

Select  $\cos(90)$   $\rightarrow -0.44$

Select  $\tan(0)$   $\rightarrow 0$

Select  $\tan(90)$   $\rightarrow -1.99$

\*  $\log()$  :- It Returns the basic Logarithm value of the given Number.

Ex: Select  $\log(3)$   $\rightarrow 1.09$

Select  $\log$

\*  $\log_{10}()$  :- It Returns Base10 Logarithm value.

Select  $\log_{10}(10)$   $\rightarrow 1$

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

\* string | character functions: These functions are performing some operations on the given characters or Expressions.

\* LEN() :- It Returns the Length of a given Expression.

Ex: select LEN ('Hello') → 5

\* ASCII() :- It Returns a ASCII Number of the given character.

Ex: select ASCII ('z') → 90

\* char() :- It Returns a character of the given ASCII number.

Ex: select char (\*90) → z

\* Lower() :- It will Convert from Uppercase characters into Lowercase characters.

Ex: select Lower ('HELLO') → hello

select Lower ('hello') → hello

\* Upper() :- It will Convert from Lower case characters into uppercase characters.

Ex: select Upper ('HELLO') → HELLO

select Upper ('hello') → HELLO

\* LTRIM():- It is used to avoid the Left Side Space of the given Expression.

Ex: select LTRIM('Hello') → Hello

\* RTRIM():- It is used to avoid Right Side Space of the given Expression.

Ex: select RTRIM('Hello ') → Hello.

\* CharIndex():- It Returns the Index value of a Specific character from the given Expression.

Ex: select CharIndex('L', 'Hello') → 3

\* Replace():- This function will use to Replace The old characters with a New characters.

Ex: select Replace('Jack and Tue', 'J', 'BL')  
→ Black and Blue.

\* Reverse():- It Reverse The characters the given Expression.

Ex: select Reverse('SAI') → IAS

\* Replicate():- It Repeats ~~the~~ characters as per The specified Number of Times.

Ex: select Replicate('SAI', 5) → SAI SAI SAI SAI SAI

\* **Left()** :- It Returns the characters as per a Specified Number from the given Expression on the Left side.

Ex: Select Left ('welcome', 3) → wel

\* Right():- It Returns The Characters a per specified Number from the given Expression on Right Side.

Ex: `Select Right('welcome', 4) → Come.`

\* Space():- It will provide some space in between the given two Expressions.

Ex: select ('HAI' + space(50) + 'Hello')  
→ HAI Hello.

\* Date and Time functions: These functions are working on the given Date and Time Expressions.

→ `GetDate()` :- It Returns the current Date and time information of a system.

**Ex:** Select GetDate() → 2015-10-26 15:39:31.073.

⇒ `GetUTCDate()`:- It Returns the current universal Date and Time Information.

Ex: select GetUTCDate() → 2015-10-26 10:12:05.200.

→ Day() :- It Returns the Day value from the given Date Expression.

Ex: Select Day ('2015/10/26') → 26.

→ Month() :- It Returns Month value from the given Date Expression.

Ex: Select Month ('2015/10/26') → 10

→ Year() :- It Returns the year value from the given Date Expression.

Ex: Select year('2015/10/26') → 2015

Date: 27-10-2015, TUESDAY

→ DateName() :- It Returns the Name of the Date and Month of the given Date Expression.

Ex: select DateName(DW, GetDate()) → Tuesday

select DateName(MM, GetDate()) → october

→ DateAdd() :- This function is used to Add No. of Days, Months, Years to the given date Expression.

Ex: select DateAdd(DD, 5, GetDate()) → 2015-11-01

select DateAdd(MM, 5, GetDate()) → 2016-03-27

select DateAdd(YYYY, 5, GetDate()) → 2010-10-27

⇒ DateDiff () :- This function will return the difference between Starting and Ending Date Expressions.

Ex: Select DateDiff (DD, '2014/10/27', GetDate ()) → 365 days

Select DateDiff (MM, '2014/10/27', GetDate ()) → 12 months

Select DateDiff (YYYY, '2014/10/27', GetDate ()) → 1 year.

⇒ Datepart () :- It Returns Date, Month and Year part values from the given Date Expression.

Ex: Select Datepart (DD, GetDate ()) → 27

Select Datepart (MM, GetDate ()) → 10

Select Datepart (YYYY, GetDate ()) → 2015

\* Aggregative / Group Functions : These functions are working on the group of values and returns a single value to the user.

→ SQL Server supports the following aggregative / group functions those are SUM(), AVG(), MAX(), MIN(), COUNT().

Ex: Create Table Employee (EID Int, Ename varchar(40), salary Money)

## Table:

EID	EName	Salary
101	AA	85000.00
102	BB	32000.00
103	CC	47000.00
104	AA	95000.00
105	CC	72000.00
106	NULL	55000.00

→ SUM():- It Returns the Sum of values.

Ex: WAP to find the total salary of an Employee?

Ans: Select sum(Salary) From Employee → 3,86,000.00

→ AVG():- It Returns the Average of Total Salary.

Ex: WAP to display average salary of an employee?

Ans: Select AVG(Salary) From Employee → 64333.3333

→ MAX():- It Returns the Maximum value from the Given Group of values.

Ex: WAP to find out the maximum Salary of Employee Table?

Select Max(Salary) From Employee → 95000.00

→ MIN():- It Returns the minimum value from the Given Group of value.

Ex: WAP to Retrieve MIN Salary from Employee Table?

Select min(Salary) From Employee → 32000.00

⇒ Count () :- This function Returns the No. of Records in a Table. count() again classified into 3 types.

1. Count (\*): It counts all values including duplicate and NULL values in the Table.

Ex: Select count(\*) From Employee → 6

2. count (<column Name>): It counts all values including duplicate but not NULL values in a column.

Ex: select count(Ename) From Employee → 5.

3. count (Distinct <column Name>): It counts unique values only.

Ex: Select count(Distinct Ename) From Employee → 3

\* Distinct Keyword : This keyword is used for avoiding the duplicate values from a particular column in the Table.

Syntax: Count(Distinct <column Name>)

## Operators in SQL Server :

- Operator is a symbol which can be used to perform some operations on the given operand values.
- SQL Server supports the following operators those are
  1. Assignment operators
  2. Arithmetic operators
  3. Comparison operators
  4. Logical operators
  5. Set operators
  6. Special operators

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

1. Assignment operator : It will use to assign the values to the operands . It supports only one operator i.e " = " operator .

\* WAP to display employee details whose EID = 101 ?

Select \* From Employee where EID = 101

\* WAP to update employee salary as 25000 whose Ename = SA ?

update ~~set~~ Employee set salary = 25000 where Ename = SA

\* WAP to delete Employee details whose employee salary = 30,000 ?

Delete from Employee where salary = 30,000 .

2. Arithmetic operators: These operators are used to perform some mathematical calculations on a given operand values. These operators are +, -, \*, / and %.

Ex: select  $100+200 \rightarrow 300$

select  $23-12 \rightarrow 11$

select  $20*2 \rightarrow 40$

select  $30/5 \rightarrow 6$

select  $17\%4 \rightarrow 1$

\* WAP to input student Id, student Name and their subject marks. Find the total marks, Average marks and class of a student?

Step 1: Create Table studentDetails (SID int, sname varchar(30), maths int, phy int, che int, Total int, Average int, class varchar(40))

Step 2: Insert studentDetails (STID, sname, maths, phy, che) values (101, 'AA', 75, 68, 59), (102, 'BB', 64, 59, 48), (103, 'CC', 55, 48, 38), (104, 'DD', 11, 8, 6)

Step 3: Find the total marks of each student.

update studentdetails set Total = maths + phy + che

Step 4: To Find the Average of each student.

update student details set Average = Total / 3

Step 5: To find the class of each student.

update student details set class = case

when Average >= 60 Then 'first'

when Average >= 50 Then 'Second'

when Average >= 40 Then 'Third'

Else

'Fail'

END

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

STID	SName	maths	phy	che	Total	Average	class
101	AA	75	68	59	202	67	first
102	BB	64	59	48	171	57	second
103	CC	55	48	38	141	47	Third
104	DD	11	8	6	25	8	fail.

\* Case() → This function will use to execute multiple conditions at a time and Returns a single value to the user.

Syntax: Case

when <Condition 1> Then statement 1

|

|

:

:

:

when <Condition n> Then 'statement n'

else

'statement'

End.

Date : 29-10-2015, Thursday  
Wednesday

3. Comparison operators: These operators are used to compare the values with the given condition in the query.

→ These operators are  $<$ ,  $>$ ,  $\leq$ ,  $\geq$ ,  $\neq$ ,  $\neq$ ,  $\neq$ ,  $\neq$

\* WAG to display Employee details whose employee salary is greater than 51000 ?

Select \* From Employee where salary > 51000.

\* WAG to update Employee address as SRNAGAR whose employee salary is less than 18000 ?

update Employee set EAddress = 'SRNAGAR' where salary < 18000.

\* WAG to delete Employee details whose EID  $\geq 112$  ?

Delete From Employee where EID  $\geq 112$

4. Logical operators: These operators are used to check more than one condition in the query. These operators are AND, OR, NOT

AND → It will perform the operations when the given all conditions should be True.

Truth Table 'AND'

		Condition1	Condition2	
		T	T	→ T
		T	F	→ F
		F	T	→ F
		F	F	→ F

→ WAQ to display employee details who is working in .Net department and whose salary = 75,000 ?

Select \* From Employee where Dname = ' .Net ' AND Salary = 75000.

→ WAQ to update employee salary as 10000 whose employee address is equals to Hyderabad and their EID is 114 ?

update Employee set salary = 10000 where EAddress = 'Hyd' AND EID = 114.

→ WAQ to delete employee details whose employee salary 93000 and working in HR Department ?

Delete From Employee where Salary = 93000 AND Dname = 'HR'

OR → It will perform some operations if any one condition is True from the given conditions ~~is~~ in the query.

\* WAQ to display Employee details who are working under the EID's are 101, 103, 105, 107, 109 ?

Select \* From Employee where EID = 101 OR EID = 103  
OR EID = 105 OR EID = 107 OR EID = 109.

\* WAQ to update employee Address as Madhapur who are working under the departments are Java, HR, and Finance ?

Truth Table 'OR'		
T	T	→ T
T	F	→ T
F	T	→ T
F	F	→ F

update Employee set EADDRESS = 'MADHAPUR' where  
DName = 'Java' OR DNAME = 'HR' ~~DNAME~~ OR  
DNAME = 'FIN'

\* WAQ to delete Employee details whose employee salaries are 28000, 41000, 44000 and 18000

Delete from Employee where Salary = 28000 OR  
Salary = 41000 OR Salary = ~~44000~~ 44000 OR Salary = 18000.

NOT →

\* WAQ to get Employee details who are not working under .Net and Finance department?

Select \* from Employee where NOT DNAME = '.NET'  
AND NOT DNAME = 'FIN'

5. Special operators :- SQL Server supports the following Special operators Those are Between, Not Between, In, NotIn, IS NULL, IS NOT NULL, Like, Not Like.

Between → It will work on the given Range values only.

→ Between operator is always Returns the values including Source and Destination values.

→ It can be applied on small to Big Range values only. But it doesn't support Big to small Range values.

→ When we implement BETWEEN operator we should used AND operator only.

\* WAQ to get employee details whose employee salaries between 48000 to 66000 ?

Select \* from Employee where Salary Between 48000 AND 66000. OR

Select \* From Employee where (Salary >= 48000) AND (Salary <= 66000).

\* WAQ to ~~update~~ update employee salary as 8000 whose Employee ID Between 105 to 110 ?

~~Select~~ update Employee set Salary = 8000 where EID Between 105 AND 110.

\* WAQ to display employee details whose Salary is Not between 66000 to 93000 ?

Select \* from Employee where Salary NOT Between 66000 AND 93000.

IN :→ It will perform the operations on the given list values in the condition.

→ In operator is an extension of OR operator.

→ The Query execution speed of In operator is faster than 'OR' operator.

- When we use OR operator we will Repeat the column Names again and again in the query whereas as we use In operator there is no need to repeat the column Name in the query.
- When we use OR operator the query length will be increased whereas by using In operator we can Compress the Query Length.

Date : 29-10-2015, Thursday

- \* WAG to display Employee details whose names are SAI, pooja, Suman, Gopi and Ramu?  
Select \* from Employee where EName In ('SAI', 'pooja', 'suman', 'Gopi', 'Ramu')
- \* WAG to update Employee Salary as 35,000 whose Employee salaries are 18000, 41000, 66000?  
update Employee set salary=35000 where salary In (18000, 41000, 66000).
- \* WAG to delete employee details who are working in .Net and java department?  
Delete from Employee where Dname In ('.Net', 'java')

\* WAG to display Employee details who are not working  
Under the Employee ID's are 102, 104, 106 and 108?

Select \* From Employee where EID Not In (102, 104, 106, 108)

is NULL :- This operator will compare the values with  
NULL value in the Table.

→ NULL is Nothing but unknown or un-defined value.

→ It occupies 0 byte memory (It doesn't occupy any space)

\* WAG to display Employee Details whose Employee  
Department Name is unknown?

Select \* From Employee where DName is NULL

\* WAG to get Employee Details whose Employee

Department ~~Name~~ is not Unknown?

Select \* From Employee where DName is Not NULL

<sup>and</sup> <sup>IV.</sup> Like :- It will use to filter the characters in the  
given Expression.

→ When we use like operator we should follow the following  
Symbols those are

(i) % → will represent Any character in the Expression

(ii) \_ → will Represent a single character in the Expression

(iii) [ ] → will Represent set of characters.

\* WAG to get Employee details whose Name starts with 'S' character ?

Select \* from Employee where Ename Like 'S%'

\* WAG to Retrieve Employee details whose Name Ends with 'i' character ?

Select \* from Employee where Ename Like ~~Ename~~ '%i'

\* WAG to display Employee details whose Name Contains Third character is 'O' ?

Select \* from Employee where Ename Like ~~Ename~~ - 0%

\* WAG to Read Employee details whose employee name Contains 3 characters ?

Select \* from Employee where Ename like '---'.

\* WAG to get Employee details whose Name starts with 'A' and Ends with 'L' ?

\* Select \* from Employee where Ename Like 'K%L'

\* WAG TO Retrieve Employee Details whose Name Contains 'H' character ?

Select \* from Employee where Ename Like '%H%

\* WAG to display Employee details whose EID starts with 'I' and ends with 'I' ?

Select \* from Employee

\* WAO to Read Employee details whose name starts with 'H, N, G and P' characters ?

Select \* from Employee where Ename Like '[H,N,G,P]%'

\* WAO to get Employee details whose name starts with A to Z characters ?

Select \* from Employee where Ename Like '[A-Z]%'

\* WAO to display Employee details whose Employee Name not starts with A-Z characters?

Select \* from Employee where Ename NOT Like '[A-Z]%'.

6. Set Operators: Set operators are used to combine the result of 2 or more tables as a single set of values. SQL Server supports the following

- Set operators.
- ① union
  - ② union ALL
  - ③ Intersect
  - ④ Except

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

→ Before Implement Set operators we should follow the Basic Rules are

- 1. No. of columns and order of the columns must be same in all queries.
- 2. Those columns Datatypes should be compatible (well Matched).

Ex: Create Table Emp-HYD (EID INT, Ename Varchar(30),  
Sal Money)

Create Table Emp-Chennai (EID INT, Ename Varchar(30),  
Sal Money)

Emp-Hyd

EID	Ename	Sal
101	SAI	35000.00
102	Gopi	25000.00
103	HARI	63000.00
104	NARESH	78000.00

Emp-Chennai

EID	Ename	SAL
101	SAI	35000.00
105	Pooja	55000.00
106	Neethu	22000.00

1. UNION → It Returns <sup>all</sup> values from the Table without the duplicates

\* WAP to display All Employee details from both Branches?

```
Select * From Emp-Hyd  
Union  
Select * From Emp-Chennai
```

2. UNION ALL → It is same as union operator but returns the values including duplicates.

```
Select * From Emp-Hyd  
Union ALL  
Select * From Emp-Chennai.
```

3. Intersect → It Returns the Common values from the Tables.

\* WAG to display EID, EName who are working in both Branches ?

Select EID, EName From Emp\_Hyd

Intersect

Select EID, EName From Emp\_Chennai

4. Except → It Returns all values from Left hand side Table which are not found in Right hand Side Table.

\* WAG to display Employee Names who are working in Hyderabad Branch but not in Chennai Branch ?

Select EName From Emp\_Hyd

Except

Select EName From Emp\_Chennai.

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

\* WAG to display Employee Names who are working in Chennai but not in Hyderabad ?

Select EName From Emp\_Chennai

Except

Select EName From Emp\_Hyd.

Date, 16-11-15, MONDAY

\* Clauses in SQL Server:- we can add these clauses to the Query for providing the additional facilities like filtering the records, sorting the records, fetching the records and grouping the records in the Table.

→ SQL Server supports the following clauses. Those are

1. where
2. order by
3. Top n
4. Group by
5. Having

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

1. Where: This keyword will use to filter the records in the Table.

Q: WAP to display Employee details whose Employee Name is equal to sai ?

Ans: select \* from Employee where Employee Name = 'sai'  
      ↳ Query part                    ↳ Additional part.

Q: WAP to update student fee as 3500 whose student course is equal to .Net ?

Ans: update students set SFEE = 3500 where course = 'Net'

Q: WAP to delete customer details whose customer A/C NO = 1024 ?

Ans: Delete from customer where CACNO = 1024

Note: where keyword can be implemented on select, update and delete Commands only.

2. Order by: It will used to arrange @ sorting the values either Ascending or Descending manner in the Table.

→ By default orderby clause will arrange the values in Ascending order only.

→ If we want to arrange the values in descending order Then we use "DESC" keyword.

→ Order by clause can be used <sup>on</sup> integer and also character datatypes.

→ It will arrange the values in temporary manner.  
(we can't store permanently).

Q: WAP to display Employee Salary in Ascending manner?

Ans: Select Salary from Employee order by salary.

Q: WAP to display Employee Names in Descending order?

Ans: Select Ename from Employee order by Ename DESC

Note: It can applied on select Command only.

3. Top 'n':- It is used for fetching top most records from the Table. here 'n' will represent no. of records in the Table.

Q: WAP to retrieve Top 5 records from the Employee Table?

Ans: select TOP(5) \* from Employee.

Q: WAP to update Top 3 employee salaries as 10000 ?

Ans: update TOP(3) Employee set salary = 10000 .

Q: WAP to delete Top 2 records from Employee Table ?

Ans: Delete Top(2) from Employee

Q: WAP to get Top 3 highest salaries Employee details from The Table ?

Ans: Select Top(3) \* from Employee details order by salary DESC

Q: WAP to display Top 2 Least Salaries employee details from the Table ?

Ans: Select Top(2) \* From Employee details order by Salary

Date: 17-11-15, Tuesday

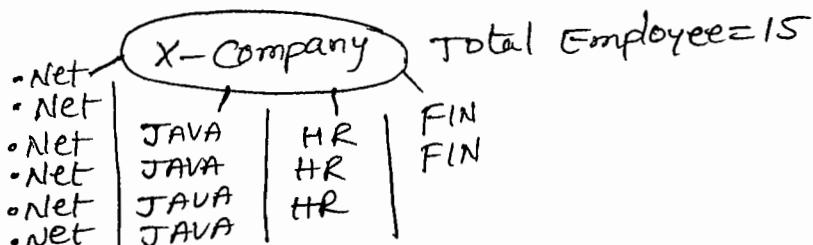
4. Group by :- This clause is used for dividing the similar data into groups. When we use groupby clause we must and should use an aggregative functions like count, sum, max, avg, min().

→ When we implement group by clause in the query first the data in the Table will be divided into a separate groups based on the columns and later an aggregative function will execute on each group to get the results.

Ex: Create Table Employee (EID int, Ename varchar(30), salary money, DName varchar(50), Location varchar(50))

	EID	Ename	salary	Dname	Location
1	101	x	75000	.NET	HYD
2	102	x	66000	JAVA	PUNE
3	103	x	87000	.NET	MUMBAI
4	104	x	41000	JAVA	UP
5	105	x	51000	FIN	MADHAPUR
6	106	x	28000	JAVA	DELHI
7	107	x	48000	.NET	MP
8	108	x	79000	JAVA	HYD
9	109	x	38000	.NET	CHENNAI
10	110	x	44000	HR	HYD
11	111	x	15000	.NET	KERALA

12	112	X	93000	HR	UP
13	113	X	9000	.NET	MP
14	114	X	58000	FIN	HYD
15	115	X	18000	HR	UP



Q: WAQ to display The total no. of Employees working in the org'n?

Ans: Select count(\*) From Employee. O/P: 15

Q: WAQ to findout no.of employee working in each dept

in The org'n ?

Ans: Select DName, Total Employee = count(\*) From Employee  
Group by DName.

Output: DName    Total employee

1. .NET	6
2. JAVA	4
3. FIN	2
4. HR	3

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Q: WAQ to display no.of employee working in each Dept with Location size?

Ans: Select DName, NumofEmployee = Count(\*), Location From Employee Group by DName, Location.

Q: WAQ to display The Total salary of Each Dept in the org'n ?

Ans: Select DName, Total Salary = Sum(Salary) From Employee  
Group by DName.

Output: DName    Total Salary

1. .NET	272000.00
2. JAVA	207000.00
3. HR	155000.00
4. FIN	109000.00

Q: WAP to find max, min and average salaries of each Dept?

Ans: Select DName MAX SAL = max(Salary), MIN SAL = min(Salary),  
AVG SAL = Avg(Salary) From Employee group by DName.

Note: 1. It can applied on select Command only. 2. Group by clause can be applied on more than one column at a time.

5. Having: It is also used for filtering the records in the table just like where clause keyword.

Q: WAP to display DNames if the Dept total Employee is greater than 3?

Ans: Select DName, Count(\*) From Employee Groupby DName having

O/P:

DName	No. of employees
.NET	6
JAVA	4

 count > 3

Q: WAP to get DNames if the Dept total salary is less than 2 Lacs?

Ans: Select DName, sum(Salary) From Employee Groupby DName

O/P: HR, FIN having sum(Salary) < 200000

\* Syntax for creating a new Table from an existing Table:

Syntax 1: Select \* into <NewTN> From <Old TN>

Ex: Select \* into NewEmp From Employee.

~~In This case~~ In This case it will create a new Table by copy all rows and columns data from the old Table.

Syntax 2: Select <Req. column names> Into <NewTN> From <Old TN>

Ex: Select EID, salary, Location Into specEmp From Employee

→ In this case it will create a new Table with specific columns data from the old Table.

Syntax 3: Select \* Into <New TN> From <Old TN> where (Condition)

Ex: Select \* Into dummyEmp From Employee

where EID=0

DName	MAX SAL	MIN SAL	AVG SAL
1. .NET	87000.00	9000.00	45333.33
2. FIN	58000.00	51000.00	54500.00
3. HR	93000.00	18000.00	51666.66
4. JAVA	72000.00	28000.00	51750.00

## \* Syntax for Copy data From one Table to another Table :

→ Before Copy the Data from one to another table we should follow the Basic Rules.

(1) No.of columns and order of the columns must and should be same with in the both Tables.

(2) The columns datatypes should be compactable.

(3) we can copy the Data from one to another Table with the combination of Insert and Select Commands.

## \* Differences Between where and having clauses.

### where clause

1. It will filter the Records before grouping the data in the Table.
2. If filtering column is associated with an aggregative function at this situation we cannot use where clause Keyword.
3. Where clause can use without group by clause Keyword.
4. It will execute on individual rows in the Table.

### Having clause

1. It is also used for filtering the records after grouping the data in the Table.
2. But we can use having clause at this situation.
3. It is not possible to use without group by clause Keyword.
4. It will execute on group of records along with group by clause.

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

[date: 18-11-2015, wednesday]

Syntax: Insert [into] <New TN> select \* from <OLD TN>

Ex: Insert DummyEmp select \* from Employee.

Note: whenever we perform the DML operations on a new Table those operations are not effected to the old Table and viceversa. Becoz Tables ~~are~~ are Independent objects ~~that~~ that means it doesn't depend on another Table.

## SYNONYMS

Synonym: Synonym is database object which can be created as an "alias name/duplicate name" for any object like table, view, procedure etc.

- If we apply any DML operations on Synonym the same operations automatically ~~are~~ effected to Corresponding base Table and viceversa.
- If we create a Synonym, The Synonym will be created on entire Table. It is not possible to create the synonym on partial Table.
- Synonym will become invalid into two cases.
  1. when we drop the Base Table.
  2. when we change the Base Table Name.

→ On Invalid Synonym we cannot apply any DML operations.

→ we cannot create Synonym based on more than one Table at a time.

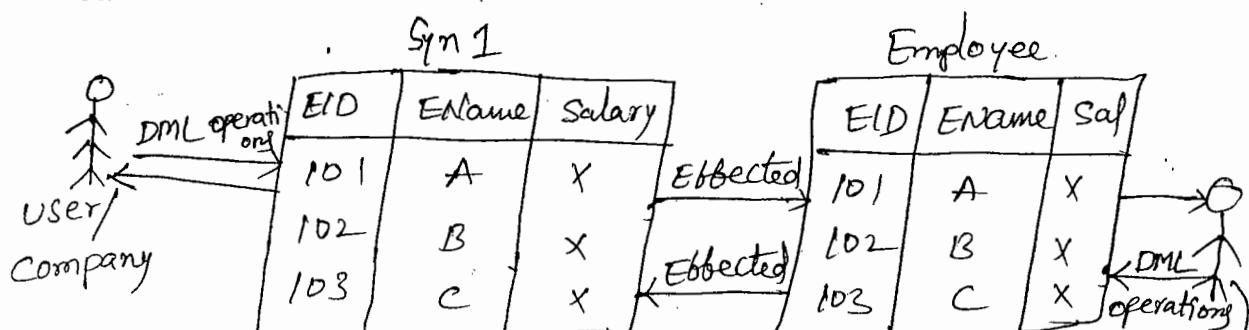
→ When we change the structure of the base Table the Corresponding synonym automatically reflected with same changes.

Syntax: Create Synonym <Synonym name> for <TN/Object Name>

Ex: write a query to create a synonym for accessing the Data from Employee Table.

Aus: Create Synonym syn1 for Employee.

→ To Execute the above Query it will create a Synonym syn1 on Employee Table for accessing the data using DML operations like below.



Testing:

Insert Syn1 values (104, 'Gopi', 89000, 'FIN', 'Kerala')

Update Syn1 set salary = 5000 where EID = 102

Delete from Syn1 where EID = 104.

[Date: 19-11-2015, Thursday]

- Note:
1. Generally the Synonyms are used in Remoting Environment for Accessing the data ~~to~~ from multiple locations.
  2. Synonyms will provide Security for the Original Table Names in the Database.
  3. But the Synonyms do not provide Security to the Table Information. So To overcome this drawback we use another database object i.e "View".

\* Syntax for Dropping a Synonym:

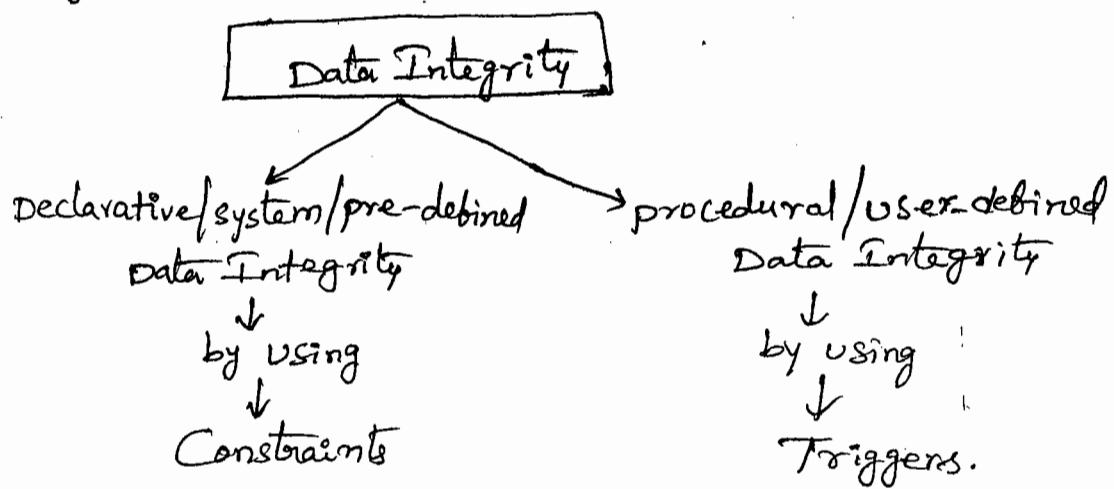
Drop Synonym <Synonym Name>

Ex: Drop Synonym syn4

v.v. Inf.

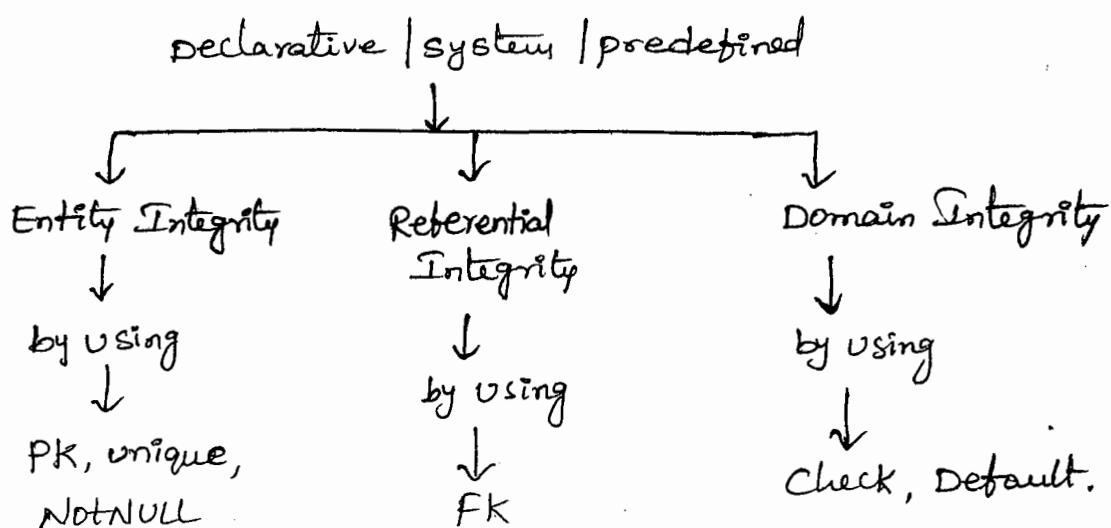
Data Integrity: Data Integrity is Nothing but to maintain Accuracy & Consistency of the Data in the Database Table.

→ This Data Integrity mechanism is again classified into Two ways. Those are



SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

\* Declarative | system | pre-defined Integrity: It can be Implemented with the help of Constraints. This Declarative Integrity again classified into 3 types Those are.



Entity Integrity: Entity Integrity Ensures each row in a Table is a uniquely identifiable entity. you can apply Entity Integrity to a table by specifying a primary key, unique and NotNULL constraint.

Referential Integrity: Referential Integrity Ensures The Relationship between Tables remain preserved as data is inserted, deleted and modified. you can apply Referential Integrity using a FOREIGN KEY constraint.

Domain Integrity: Domain Integrity Ensures the data values inside a database follow defined rules for values, range and format. A database can enforce these rules using CHECK, DEFAULT Constraints.

- \* Constraints:- Constraints are used for Enforcing or Restricting the data
  - For us by using Constraint we can restrict data for but not operations on a Table.
  - Constraints are 6 types in SQL Server those are.
    1. Default
    2. unique
    3. Not NULL
    4. check
    5. primary
    6. foreign constraints.

Date: 20-11-2015, Friday

- 1. Default: This is used to assign the user defined Default value to the particular column in the Table.
- By using Default Constraints we can avoid the system defined Default value (NULL value) from a column while the user inserting the values in an explicit manner.

→ It can be applied on any Datatype column like Integer, character, Decimal... etc.

→ A Table Contains any No.of Default constraints.

Syntax: Create Table <TN> (<columnName> <datatype>  
<constraintName>, .....);

Ex: Create Table customers (CID int, cname varchar(20),  
Bal money Default 500, Age int Default 18)

Testing:

Insert Customers values (1021, 'SAI', 1500, 24)

Insert Customers (CID, cname) values (1022, 'KAMAL').

O/P:

CID	cname	Bal	age
1021	KAMAL	1500	24
1022	SAI	500	18

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

2. Unique: It will avoid Duplicate values of a column  
but accepted NULL values into the column.

→ It can Apply on any Datatype Column and also  
a Table Contains any No.of unique Constraints.

Ex: Create Table products (pcode int unique, pname  
varchar(30) unique, price decimal(6,2) unique)

Testing:

Insert product values (101, 'C', 500) → Allowed

Insert product values (101, 'C', 500) → Not Allowed

Insert product values (NULL, NULL, NULL) → Allowed

Insert product values (NULL, NULL, NULL) → Not Allowed.

Insert product values (102, 'C++', 1000) → Allowed.

Note: A unique constraint column is accepted one null value only.

3. NOT NULL: It avoids null values from the column and accepted duplicate values into the column.

→ It can apply on any datatype column and a table contains any no. of notnull constraints.

Ex: Create Table student (STID int NOTNULL, SName  
varchar(20) NOTNULL)

Testing:

Insert student values (101, 'SAI') → Allowed

Insert student values (101, 'SAI') → Allowed

Insert student values (NULL, NULL) → Not Allowed.

Note: The above 2 constraints have its own drawbacks those are accepting NULL and duplicate values into the Table. So to overcome the above 2 drawbacks then we write the combination of unique and notnull constraints on a column like below.

Ex: Create Table orders (ORNO int unique notnull,  
ORDDate date unique notnull, Quantity int  
unique notnull)

Testing:

Insert orders values (1, '2015/11/20', 11) → allowed

Insert orders values (1, '2015/11/20', 11) → not allowed

Insert orders values (NULL, NULL, NULL) → not allowed

Insert Orders values (2, '2015/11/21', 12) → allowed.

Note: Can we apply the multiply constraints on one column  
in the Table ?

Ans: Yes, we can.

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

4. check: It is used to verify or check the values  
with the user defined condition ~~on~~ a column.

→ It can apply on any datatype column and table  
Contains any No.of check constraint.

Ex: Create Table Emp (EID int unique notnull,  
EName varchar(30) ~~unique~~, notnull,  
~~salary~~ sal money check (sal >= 8000),  
Age int check (Age Between 18 And 30),  
Testing Location varchar(40) check (Location in  
('Hyd', 'up', 'mp'))

### Testing:

Insert Emp values (101, 'SAI', 8000, 18, 'HYD') → Allowed

Insert Emp values (101, 'SAI', 7000, 17, 'HYDERABAD') → Not Allowed

Insert Emp values (102, 'Gopi', 9999, 30, 'mp') → Allowed.

Date: 21-11-2015, Saturday

5. Primary Key Constraints: It is a Combination of Unique and ~~Normal~~ Constraints NotNULL Constraints.

By using primary key Constraint we can avoid duplicate and NULL values from the column at a time.

→ It Can Apply on any Datatype column like Integer, character, Decimal Datatypes etc.

→ A Table should contain one primary key only.

Ex: Create Table Emp (EID int primary key , EName  
Varchar(30) , Sal Money )

Testing: Insert Emp values (101, 'SAI', 25000) → Allowed

Insert Emp values (101, 'SAI', 25000) → Not Allowed

Insert Emp values (NULL, 'SAI', 25000) → Not Allowed

Insert Emp values (102, 'KAMAL', 22000) → Allowed.

\* Composite primary key: If primary key constraint construct with multiple columns in the same table can be called as Composite primary key.

- A Composite primary key mechanism individual columns are accepted duplicate values, but the duplicate combination should not be repeated in the table.
- Composite primary key can apply maximum 16 columns in a table.
- It can be define at the end of the table definition.  
(That means after all columns definition)

Ex: Composite PK { Branch details  
(PK + PK)}

city	BCode	BLocation
HYD	1021	SRNAGAR
HYD	1022	MADHAPUR
HYD	1023	AMEERPET
MP	1021	MP1
MP	1022	MP2
UP	1022	UP1
UP	1023	UP2

Ex: Create Table BranchDetails (city varchar(20),  
BCode int, BLOC varchar(40), primary key (city,  
BCode))

Testing:

Insert BranchDetails values ('HYD', 1021, 'SRNAGAR') → Allowed

Insert BranchDetails values ('HYD', 1021, 'MADHAPUR') → Not Allowed

Insert BranchDetails values ('HYD', 1022, 'MADHAPUR') → Allowed

Insert BranchDetails values ('MP', 1022, 'MP1') → Allowed

Insert BranchDetails values ('MP', 1022, 'MP2') → Not Allowed

Note: 1. Can we apply more than one primary key on a single Table?

Ans: No, we can't

2. Can we apply primary key constraint on multiple columns within the same Table?

Ans: Yes, we can by using Composite primary key.

→ primary key is also called as Composite primary key and Candidate key.

Date: 23-11-2015, Monday

\* FOREIGN KEY: One of the most important Concepts in database is Creating Relationships between Database Tables.

→ These Relationships provide a Mechanism for linking data stored in multiple Tables so that user is retrieving data in an efficient Manner and then verify the existence of one Table data in the other.

To impose a Foreign Key Constraint we require the following things (Rules):

1. In Order to Create a link between two tables we must specify a Foreignkey in one table that references a column in another Table is Primary Key column.

2. We Require two Tables for binding with each other and those two tables have a Common Column name (Optional) and those columns should be same datatypes.

→ If a Table Contains primary key Constraint then can be called as parent Table.

→ If a Table Contains Foreign key Reference then can be called as child Table.

- It Can Apply the Reference on Any Datatype Column like integer, character, date and Time ---etc.

I.W → we can apply maximum 253 Foreign Keys on a Single Table.

→ By Default the foreign key Constraint is Accepted Duplicate and NULL values.

→ By using Foreignkey we can create the 3 types of Relationships. Those are

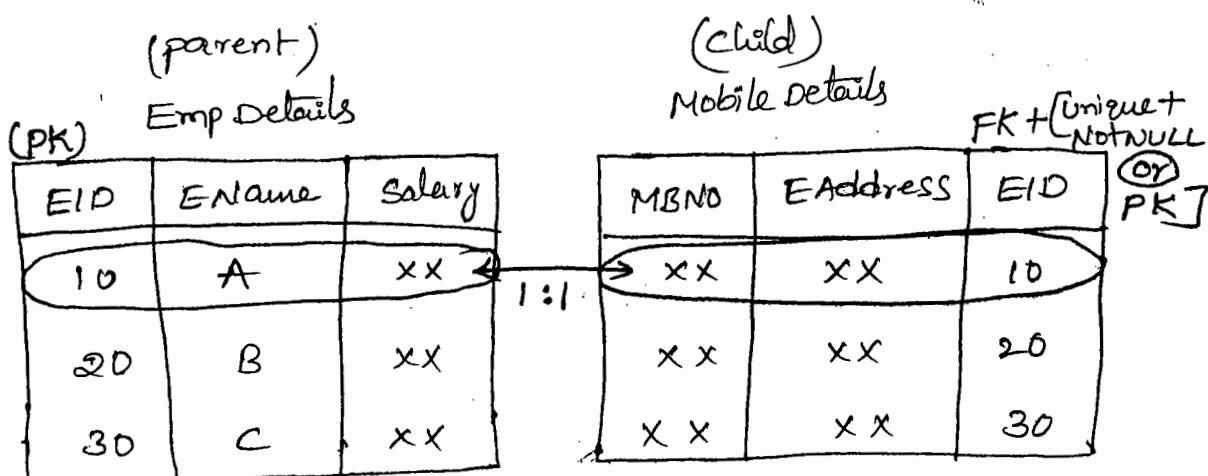
1. one to one Relation
2. one to many / many to one "
3. many to many Relation

## 1. Working with One to One Relation:

### Step 1: Definition or Analysis:

A Row in a Table is Associated with a Row in an other Table is known as One to One Relation.

## Step 2: Designing



Note: To Maintain One to one Relation between the Tables first we should avoid Duplicate and NULL values of a foreignkey column by using primary key or The Combination of unique and NotNull Constraints.

### Step 3: Implementation

Date: 26-11-2015, Tuesday

Step1: Create Table EmpDetails (EID int primary key, Ename varchar(30), SAL Money)

Step2: Insert EmpDetails values (10, 'SA', 25000),  
(20, 'Gopi', 12000), (30, 'RAMU', 42000)

Step3: Create Table mobileDetails (MBNO char(10),  
EAddress varchar(40), EID int unique NOTNULL  
Foreignkey References EmpDetails (EID))

Step4: Insert mobileDetails values ('9874563214',  
'SRNAGAR', 10) → Allowed

Insert mobileDetails values ('9874563214',  
'SRNAGAR', 10) → Not Allowed

Insert mobileDetails values ('8884563214',  
'MADHAPUR', 20) → Allowed

Insert mobileDetails values ('9004563214',  
'AMEERPETA', 30) → Not Allowed.

Step 4

Testing:

→ when we impose the Foreign Key Constraint and establish relation b/w the Tables. The following 3 Rules come into picture.

Rule 1: Cannot insert a value into the foreign key column provided that value is not existing under the Reference key column of the parent Table.

Ex:- Insert Mobile Details Values ('7412589632', 'UP', 40)  
↓  
Not Allowed      Unreferenced Value.

Rule 2: Cannot update the Reference key value of a parent Table provided that value has Corresponding child Record in the Child Table without Addressing what to do with the child Record.

Ex:- Update EmpDetails Set EID=1 where EID=10 → Not Allowed

Rule 3: Cannot delete a record from the parent Table provided that Records Reference key value has child Record in the Child Table without addressing what to do with the child Record.

Ex:- Delete From EmpDetails where EID=30 → Not Allowed.

→ If we want to delete or update a record when they have Corresponding child Records in the Child Table then we provide ~~the~~ set of Rules to perform Delete and update operations known,

as Cascade Rules.

1. On delete cascade: It is used to delete a key value in the parent Table which is Referenced by foreign key in other Table all rows that Contains those foreignkeys in child Table are also deleted.

2. On update cascade: It is used to update a key value in the parent Table which is Referenced by foreign key in other Table all rows that Contains those foreignkeys in child Table are also updated.

\* Imposing on update and on delete cascade Rules while creating a child Table:

→ we can Apply onupdate and onDelete cascade Rules on child Table that to foreignkey column only like below.

Step 1: Create Table EmpDetails1 (EID int primary key, EName varchar(30), SAL Money) → parent Table.

Step 2: Create Table mobileDetails1 (MBNO char(10), EAddress varchar(40), EID int unique notnull foreignkey References EmpDetails1 (EID) on update cascade on delete cascade)

Step 3: Now check the DML operations on parent and the corresponding child Table.

Rule 1: Insert mobileDetails 1 values ('7412589632', 'UP', 40) — Not Allowed

\* Rule 2: update EmpDetails 1 set EID=1 where EID=10 → Allowed

Rule 3: Delete from EmpDetails 1 where EID=30 → Allowed

2. Working with One to Many or Many to one Relation:

1. Definition: A Row in a Table is associated with any Rows in an Other Table is known as One to Many or Many to One Relation.

2. Designing:

(parent)			(child)		
(PK)	Department		(PK)	Employee	(FK)
DNO	DName	DLoc	EID	Ename	Sal
10	.Net	X	101	A	25000
20	Java	X	102	B	12000
30	Testing	X	103	C	XX
			104	D	XX
			105	E	XX
			106	F	XX
			107	G	XX

1:M

### 3. Implementation:

Step 1: Create Table Department (DNO int primary key, DName varchar(30), DLOC Varchar(50)) .

Step 2: Insert Department values (10, ' .Net ', ' hyd ' ),  
(20, ' Java ', ' pune ' ), (30, ' testing ', ' mumbai ' )

Step 3: Create Table Employee (EID int primarykey ,  
EName Varchar(30) , SAL money , DNO int  
Foreignkey References Department (DNO)  
On Delete cascade On Update cascade)

Step 4: Insert Employee values (101, 'AA', 230000, 10),  
(102, 'BB', 52000, 10), (103, 'CC', 36000, 10),  
(104, 'DD', 23000, 20), (105, 'EE', 63000, 20),  
(106, 'FF', 63000, 30)

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

### 4. Testing:

Rule 1: Insert Employee values (107, 'GG', 82000, 40)

Rule 2: update Department set DNO=1 where DNO=10

Rule 3: Delete From Department where DNO=30 .

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

### 3. Working with Many to Many Relation :

1. Definition: Many Rows in a Table is associated with Many Rows in an other Tables This known as Many to Many Relation.

Date: 25-11-2015, Wednesday

#### 2. Designing:

(PK) Customers(P1)			orders (c)			products (P2)		
(PK) CID	CName	CMBNO	(PK) ordNo	ordDate	Quantity	(PK) CID	PCODE	(PK) PCODE
1	A	xx	101	2015/11/20	2	1	10	10
2	B	xx	102	2015/11/20	1	2	10	C
3	C	xx	103	2015/11/21	2	2	20	C++
			104	2015/11/22	11	1	40	.NET
			105	2015/11/22	6	3	30	SQL
			106	2015/11/23	4	1	30	
			107	2015/11/24	8	3	20	
			108	2015/11/25	1	3	10	
			109	2015/11/25	3	2	40	
			110	2015/11/26	5	3	10	

#### 3. Implementation:

Step 1: Create Table customers(CID int primary key, CName varchar(30), CMBNO char(10)).

Step 2: Insert customers values(1, 'AA', '9874563214'),  
(2, 'BB', '8523697442'), (3, 'CC', '7412589632')

step3: Create Table products ( pcode int primary key ,  
pname varchar(10) , price decimal (6,2) )

step4: Insert products values ( 10, 'C', 500),  
(20, 'C++', 1000), (30, '.Net', 3500)  
(40, 'SQL', 1800)

step5: Create Table Orders (ordno int primary key ,  
orddate date , quantity int , CID int #  
Foreign key References customers (CID) on update  
Cascade on Delete Cascade , pcode int Foreign Key  
References products (pcode) on update Cascade  
on Delete Cascade )

Step 6: Insert orders values ( 101, '2015/11/20', 2, 1, 10 ),  
( 102, '2015/11/20', 1, 2, 10 ), ( 103, '2015/11/21', 2, 2, 20 ),  
( 104, '2015/11/22', 1, 1, 40 ), ( 105, '2015/11/22', 6, 3, 30 ),  
( 106, '2015/11/23', 4, 1, 30 ), ( 107, '2015/11/24', 8, 3, 20 ),  
( 108, '2015/11/25', 1, 3, 10 ), ( 109, '2015/11/25', 3, 2, 40 )

#### 4. Testing:

Rule 1: Insert orders values ( 110, '2015/11/26', 5, 4,50 )  
↓  
Not Allowed Unreferenced  
value

Rule 2: update customers set CID=11 where CID=1  
update products set pcode=3 where pcode=30

Rule 3: ~~if~~ Delete from customers where CID=11  
Delete from products where pcode=20

\* Adding the constraints on an Existing Table:

Syntax: Alter Table <TN> ADD Constraint <key Name>  
<Constraint name>(<column Name>)

1. Adding primary key Constraint: Before Adding a primary key constraint to a column first the column will make as and later apply primary key constraint to that column.

Ex: Create Table Emp (EID int , Ename Varchar(20),  
Salary money , Age int)

Alter Table Emp Alter Column EID int NOTNULL

Alter Table Emp ADD Constraint PK Primary Key (EID)

2. Adding unique Constraint:

Alter Table Emp ADD Constraint UQ Unique (Ename)

3. Adding Check Constraint to the column:

Alter Table Emp ADD Constraint CHK Check (Salary >= 15000)

4. Adding Default Constraint on a column:

Alter Table Emp ADD Constraint DF Default 30 For Age

Date: 26-11-2015, Thursday

### Adding a Foreignkey Constraint:

Ex: Create Table Dept (DNO int , Dname varchar(20), EID int)

Alter Table Dept ADD Constraint FK\_Foreignkey (EID)  
References Emp(EID) on update cascade on delete  
cascade.

Note: 1. If we perform update and delete operations on parent Table Then the corresponding child Table Reference values also updated and deleted.

But if you perform update and delete operations on child Table <sup>Reference</sup> column then that will not effect to parent Table Reference column data.

→ we can update child Table Reference column data with the Reference values of parent Table only.

### Syntax for Dropping the Constraints from an Existing Table :

Alter Table <TN> Drop Constraint <key value>

### Dropping Unique, check and default Constraint from the Table :

Alter Table Emp Drop Constraint UQ,CHK,DF

Dropping primary key: Before Drooping a primary key constraint from parent Table first we should drop Foreign key constraint from child Table in the Relation.

Alter Table Dept Drop Constraint FK

Alter Table Emp Drop Constraint PK

\* Database Diagrams: Database Diagrams is nothing but to show the structure of the Table and the relationship among the Tables in graphical representation or Diagrammatical representation

→ steps to create a Sample Database Diagrams:

1. goto open SQLServer Management studio.
2. goto object Explorer window
3. goto Databases folder.
4. goto the required database it which we want to design the Database Diagrams (MYDB 2pm)
5. select Database Diagram folder and click on Right mouse button.
6. click on New Database Diagram option.
7. Select the Required Tables from Add Table window  
(Customers, orders and products)
8. click on Add
9. click on Close.
10. Now go to Save that Diagram with a particular Name.
11. Now check Database Diagrams folder.

Date: 27-11-2015, Friday

## \* Transaction

Transaction: A Transaction is a unit of work that is performed against a database or set of statement (Insert, update, and delete) which should be executed as one unit.

- A Transaction is the propagation of one or more changes to the database.
- For Example, if you are inserting or updating or deleting a record from the Table, Then you are performing Transaction on the Table.

~~Given~~ → The Rule of Transaction Tells that either all the stmts in the Transaction should be execute successfully or none of those statement ~~to~~ be executed.

- To Manage Transaction we provide Transaction Control Language that provides the commands like

- ⇒ BEGIN TRANSACTION
- ⇒ COMMIT
- ⇒ ROLLBACK
- ⇒ SAVEPOINT

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

- 1. BEGIN TRANSACTION: Begin Transaction Command is used to start the Transaction and also used to add nested Transactions.

Syntax: Begin Transaction  
<write statements>

2. COMMIT: Commit is used to make the transaction permanent so we cannot undo or recall the records.
- Commit is used for saving the data that has been changed permanently because whenever you perform any DML operations like UPDATE, INSERT or DELETE then you are required to write Commit at the end of all or every DML operation in order to save it permanently.
- If you do not write commit then your data will be restored into its previous condition.

Syntax: Begin Transaction  
    <Write statements>  
    Commit.

\* Examples on Commit Transaction with DML statements:

Ex 1: Begin Transaction

    Insert Branch values (1021, 'HYD', 'SBI', 'HDFC')  
    Commit

Ex 2: Begin Transaction

    Update Branch set BLOC = 'PUNE' where BCode = 1021  
    Commit.

Ex 3: Begin Transaction

    Delete from Branch where Bcode = 1021  
    Commit.

Ex4: Begin Transaction

Insert Branch values (1023, 'ICICI', 'DELHI')

update Branch set BLOC='KERALA' where BCode=1022

Delete from Branch where BCode=1021

→ In the above Transaction the DML statements are not possible through to Rollback because those statements are committed with the Transaction.

3. RollBack: Rollback command is used to cancel the Transactions and gets back to the initial state where Transaction started (i.e Undo).

→ whereas if you want to restore your data into its previous condition then you can write Rollback at any time after the DML queries has been written.

→ But remember once Commit has been written then you cannot Rollback the data.

→ Moreover you can only Rollback the DML queries that have been written after the last commit.

Syntax: Begin Transaction

Rollback Transaction.

## \* Example on Rollback Transaction with DML statements:

Ex1: Begin Transaction

Insert Branch values (1023, 'ICICI', 'DELHI')

Update Branch set BLOC = 'KERALA' where BCODE = 1022

Delete from Branch where BCODE = 1021

→ In the above Transaction the DML stmts are possible to Rollback because those stmts are not committed with the Transaction.

→ If we want to rollback the above DML stmts can we write the following syntax.

Begin Transaction

Rollback Transaction.

Date: 28-11-2015, Saturday

Save point: Save point is used for dividing (or) breaking a Transaction into multiple units. So that user will have a chance of rollbacking a Transaction up to a location.

→ When a user sets a save point within a Transaction the Save point defines a location to which a Transaction can return part of the Transaction conditionally cancelled.

→ If Condition is rollback to a savepoint, it must be proceed to Completion of the Transaction with Commit statement or it must be cancelled altogether by rolling the Transaction back to ~~to~~ its begining.

Syntax: Begin Transaction

Save Transaction <Transaction Name>  
<write statements>

Ex: Example on Savepoint with delete statement:

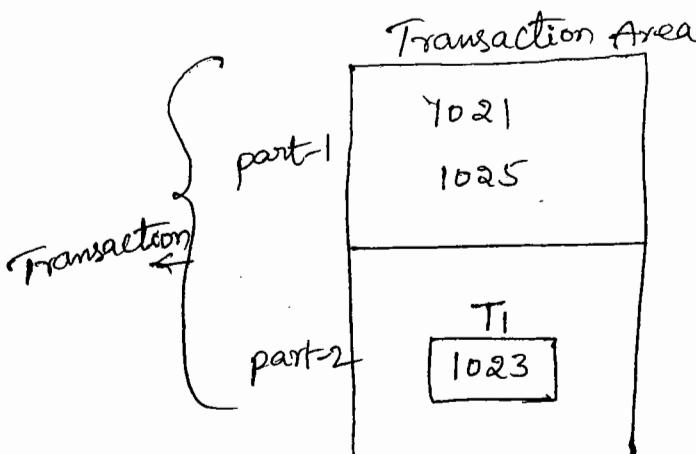
Ex: Begin Transaction

Delete from Branch where BCode = 1021

Delete from Branch where BCode = 1025

Save Transaction T1

→ Delete from Branch where BCode = 1023



SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Case 1: Begin Transaction

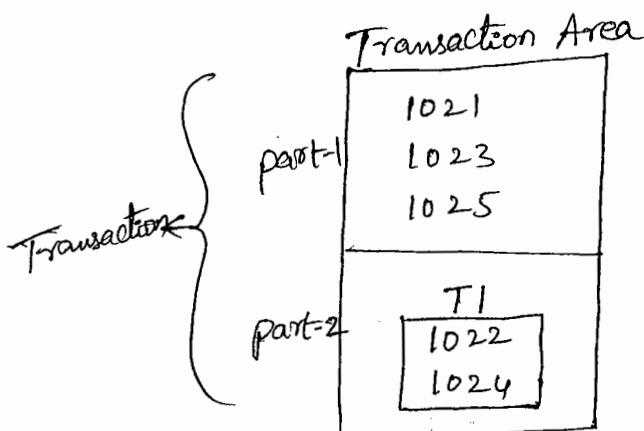
Rollback Transaction. → 1021, 1023, 1025 (All)

→ To Execute that above Transaction it will rollback all ~~regt~~ records i.e 1021, 1023, 1025.

Case 2: Begin Transaction  
Rollback Transaction T1 → 1023 (specific) only.

→ To Execute the above Transaction it will rollback only one record i.e 1023.

Ex 2: Begin Transaction  
Delete from Branch where BCode in (1021, 1023, 1025)  
Roll back Transaction T1  
Delete from Branch where BCode in (1022, 1024)



Case 1: Begin Transaction  
Rollback Transaction

→ whenever we execute the above Transaction it will rollback all records those are 1021, 1022, 1023, 1024, 1025

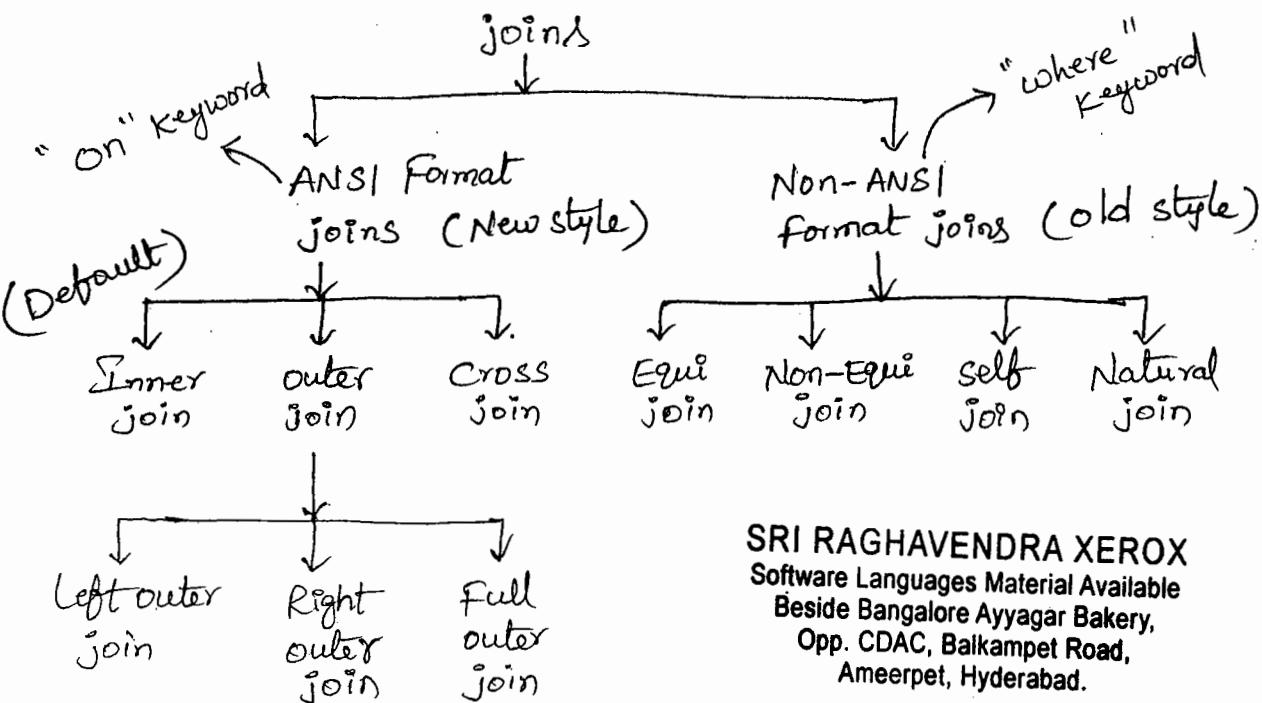
Case 2: Begin Transaction  
Rollback Transaction T1

→ whenever we execute the above Transaction it will rollback the two records only i.e 1022 and 1024.

## \* JOINS in SQL Server:

→ Joins are used for retrieving the data from multiple Tables at a time.

→ Joins are again classified into 2 Categories. Those are.



SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

→ When we retrieve the data for the multiple Tables with "on" keyword condition then that be called as "ANSI format joins".

→ When we retrieve the data from the multiple Tables based on "where" keyword condition then it can be called as "Non-ANSI format joins".

→ The default join of joins is Inner Join.

Syntax: select \* / <list of column names> from <TN1>  
 ↓                    ↓  
 <join key> /, <TN2> ON / WHERE (Joining condition);  
 ANSI                Non-ANSI

[date: 30-11-2015, Monday]

Sample Tables Creating:

Ex: create Table students (STID int, sName varchar(30),  
 MBNO char(10), CID int)

create Table Courses (CID int, cName varchar(30)  
 CFee (Decimal(6,2))

<u>STID</u>	<u>SNAME</u>	<u>MBNO</u>	<del>STID</del>	<u>CID</u>
101	AA	9874563124		10
102	BB	7894561236		20
103	CC	8523697412		30

<u>CID</u>	<u>CNAME</u>	<u>CFee</u>
10	C	500
20	C++	1000
40	.Net	3500
50	SQL	1800

Equi join: when we retrieve the data from multiple  
 tables based on an Equality condition is known as

Equi join.

→ This join mechanism will support only one operator  
 i.e equal operator (=).

→ In this mechanism we can't use  $<$ ,  $>$ ,  $=$ ,  $\geq$ ,  $\leq$ ,  $\neq$ ,  $\exists$ ,  $\forall$  etc.

→ when we use equijoin we should maintain a common column name and that column should contain same data type.

Ex: WAP to retrieve student details and also the corresponding courses details using an equality operator?

Ans: Select \* From students, courses where students.CID = courses.CID → Non-ANSI

(or)

Select S.STID, S.Sname, S.SMBNO, S.CID, C.CID, C.CName, C.CFee From Students S, Courses C where ~~S.CID~~  
 $S.CID = C.CID$  → joining Condition.

Here,

S, C are Alias Names for the Tables Students, Courses.

Alias Names: Alias name is nothing but duplicate or temporary or alternative names.

→ This alias names again we define in two levels

1. Column level Alias Names

2. Table level .. ..

→ when we create a duplicate name for the specific columns in the Table then we called Column level Alias Names.

Syntax: <column name> [AS] <column Alias Name>

→ when we create the duplicate names for the required Tables in the database is called as Table level Alias names.

Syntax: <Table Name> [AS] <Table Alias Name>

Ex: Select CID AS X, CNAME AS Y, CFEE AS Z From COURSES AS C.

Here, AS keyword is a optional.

Ex: Select CID X, CNAME Y, CFEE Z From COURSES C

Note: These Aliases we can mostly implemented in joining Mechanism.

Ex 2: WAG to display students, courses details if course id is equals to 10 ?

Ans: Select \* From STUDENTS S, COURSES C where  
 $\underline{S \cdot CID = C \cdot CID}$  AND  $\underline{S \cdot CID = 10}$   
↓ joining condition                          ↓ Additional Condition.

Ex 3: WAG to Retrive student name, coursename, mobileno From STUDENTS, COURSES Tables ?

Ans: Select \* SNAME, CNAME, MBNO From STUDENTS S, COURSES C where  $S \cdot CID = C \cdot CID$ .

Inner join: It is also used for retrieving the data from multiple tables just like Equi-join.

→ when we use Inner join mechanism we should use a common column name and the datatype is also same in the table.

Ex1: wAQ to display students and the corresponding courses details by using ANSI format?

Ans: Select \* from students Inner Join courses ON  
students.CID = courses.CID → ANSI

Ex2: wAQ to retrieve students, courses details if the course Name = C++?

Ans: select \* from students S Inner join courses ON  
S.CID = C.CID AND cname = C++.

(Or)  
select \* from students S Inner join courses ON  
 $\frac{S.CID = C.CID}{\text{Join Condition}}$      $\frac{\text{where cname} = \text{C++}}{\text{Additional Condition}}$ .

Note: whenever we add the additional condition to ANSI format query then we use "where" keyword.

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Date : 01-12-2015, Tuesday

\* Outer Join: Outer join is an extension of Inner join.

→ In Inner join Mechanism, the user will get matching data from the Tables only. That means user may loose unmatched data from the Tables.

→ To Overcome the above drawback we use outer join mechanism. By using outer join we can retrieve matching data and also unmatched data from the Tables at a time.

→ This outer join is again classified into 3 types.

1. Left outer join: It retrieves matching data from multiple Tables and also unmatched data from Left Hand Side Table only.

Ex: WAP to Display Matching data from students, courses and also unmatched data from students Table only ?

Ans: Select \* from students S Left outer join courses C  
ON S.CID = C.CID .

2. Right outer join: It retrieves the matching data from multiple Tables and also unmatched data from Right Hand Side Table only.

Ex: WAP to Display matching data from students, courses and also unmatched data from Courses Table ?

Ans: Select \* from students s Right outer join Courses c  
ON s.CID = c.CID.

3. Full outer join: It retrieves matching and unmatched data from multiple tables at a time.

Ex: WAP to display matching and unmatched records from Students, Courses Tables?

Ans: Select \* From students s Full outer join Courses c  
ON s.CID = c.CID

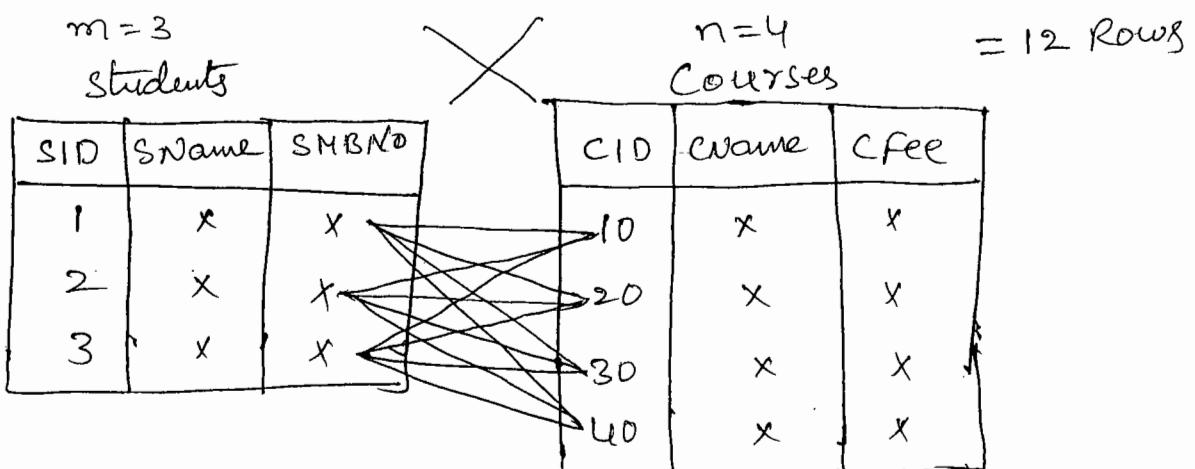
Cross join: When we joins the two table information without any condition is known as cross join.

In cross join mechanism each record of a first table is joined with each record of second table.

For example, The first table contains 'm' no. of records and the second table contains 'n' no. of records. Then we will get the cross product  $m \times n$  records.

product

$$3 \times 4 = 12$$



Ex: Select \* from Students, Courses → NonANSI

Select \* from Students Cross join Courses → ANSI

Non-Equi join: when we retrieve the data from multiple tables based on any condition except an Equality condition is known as Non-Equi join.

→ when we implement Non-equi join there is no required to maintain a common column in the tables.

→ It supports all operators like  $<$ ,  $>$ ,  $=$ ,  $\neq$ ,  $>=$ ,  $<=$ , etc.

Emp Details				Salary Grade		
EID	ENAME	SALARY	MID	GNO	LOWSAL	HIGHSAL
101	SAI	25000.00	101	1	15000.00	28000.00
102	KAMAL	32000.00	108	2	35000.00	52000.00
103	NARESH	48000.00	103	3	65000.00	85000.00
104	NEETESH	82000.00	109			
105	POOJA	11000.00	110			
106	SIDHU	64000.00	106			
107	RAVI	72000.00	111			

Ex: WAP to retrieve Employee details whose Employee salary is greater than Low salary and less than to High salary?

Ans: Select \* from EmpDetails, SalaryGrade where (Salary > Lowsal) AND (Salary < Highsal)

Ques.

\* Self join: A Table joining by itself is known as Self join.

→ Self join can be implemented when any 2 columns have some Relation within the same Table Then we use ~~set~~ Self join mechanism.

→ It can be worked on a single Table only.

→ when we use self join mechanism on a Table then we should create Alias Names to the Table.

→ without Alias names we cannot implement self join mechanism.

→ A Table Contains any no. of Alias Names.

Ques. What is the query to retrieve Manager's details from EmpDetails Table?

Ans: Select E.EID, E.ENAME, E.Salary, M.MID, M.~~ENAME~~,  
M.ENAME ManagersName, M.Salary M.Salary From  
EmpDetails E, EmpDetails M where E.EID = M.MID

→ Here, E, M are Alias Names for EmpDetails Table and ManagersName, M.Salary are Alias Names for ENAME and salary columns in the Table.

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Date: 02-12-15, wednesday

\* Natural join: Natural join is used for avoiding the duplicate columns from a result set.

Ex- WAP to display SID, Sname, MBNO, CID, Cname, CFee details from students, Courses Tables?

Ans: Select SID, Sname, MBNO, CID, Cname, CFee from students S, Courses C where S.CID = C.CID

→ When we execute the above query we will get a compilation error i.e Ambiguous column name CID.

To overcome the above drawback we use a Tablename as an Identity to the column like below.

Ans: Select SID, Sname, MBNO, C.CID, Cname, CFee from students S, Courses C Table Identity.  
where S.CID = C.CID

\* Syntax for joining more than two Tables:

select \* / <list column names> From <TN1> <join key>  
<TN2> ON (Condition) <join key> <TN3> ON (Condition)  
<join key> <TN 4> ON (Condition)  
.....  
<join key> <TN n> ON (Condition)

\* Example for Equi and Inner join:

Register details

RegNo	RegDate	CID
1	2015-12-01	10
2	2015-12-02	20
3	2015-12-03	80

Ex:1: Select \* from students S, Courses C, Register R,  
where  $S.CID = C.CID$  AND  $C.CID = R.CID$ .  $\rightarrow$  Non ANSI

Ex:2: Select \* from students S, Inner join Courses C ON  
 $S.CID = C.CID$  Inner join Register R ON  
 $C.CID = R.CID$   $\rightarrow$  ANSI

\* Examples for outer joins:

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Ex:1: Select \* from students S Left outer join Courses C  
ON  $S.CID = C.CID$  Left outer join Register R ON  
 $C.CID = R.CID$

Ex:2: Select \* from students S Right outer join Courses C  
ON  $S.CID = C.CID$  Right outer join Register R ON  
 $C.CID = R.CID$ .

Ex:3: Select \* from students S Full outer join Courses C  
ON  $S.CID = C.CID$  Full outer join Register R ON  
 $C.CID = R.CID$ .

\* Example for Cross join:

Ex: Select \* From students, courses, Register

⑦

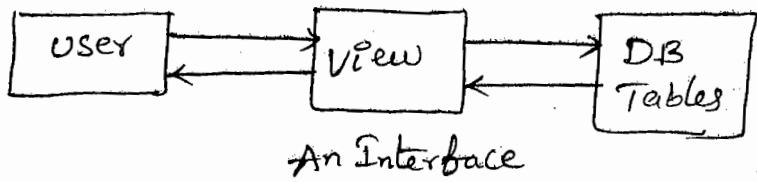
Select \* From students cross join courses  
cross join Register.

Date: 03-12-15, Thursday

\* View's in SQL: view is database object which is like a Table but Logical.

- we can call it as a logical or virtual Table because it does not have a physical existence.
- It will use to get the required information from the Table.
- View will be created by using select statement
- View will not store records in it and will not occupy memory space with help of structure existing in it and records will be displayed from Table.
- It is a dependent because view is extracted from the Table. If we want to access the data from the Table it is not necessary to change the data direct to the Table but we can access by having a view.
- So that whenever a user performs any operation on the view like select, insert, update or delete internally the view performs those operations on a Table.

→ Simply we can say that view will act as an interface between the user and tables.



### Advantages of view:

→ Views are used for security purpose in databases, views restricts the user from viewing certain columns and rows means by using view we can apply the restriction on accessing the particular rows and columns for specific user.

→ views display only those data which are mentioned in the query, So it shows only data which is returned by the query that is defined at the time of creation of the view.

→ It will provide high level security for the original Table Names and also data into the Table.

→ Views are used for sharing the Table information to the multiple programmers in the organization.

### Types of views:

→ The user can create a view in two ways Those are

1. Simple views
2. Complex views

1. Simple View: When we create a view on a single table is known as Simple view.  
→ Through simple view the user can perform all DML operations (insert, update, delete) on a Table.
2. Complex View: When we create a view on more than one table at a time can be called as Complex view.  
→ Through complex view the user cannot perform DML operations on a tables.

Syntax to create a view:

create view <view name> As Select \* / <list column names> From <Table name>

Ex: Create a simple view for accessing the data from Branches Table?

Create view SV1 as Select \* From Branch

→ To execute the above query it will create a view SV1 on Branch Table.

Testing:

Insert SV1 values (1023, '1c1c1', mumbai)

Update SV1 set BLoc='up' where BCode=1022

Delete from SV1 where BCode=1023

→ The above DML operations are performed on Branch Table Through a view SV1.

Ex2: What to create a view for restricting or hiding the specific columns in customer details Table?

Create Table customerDetails (ACNO int, CNAME varchar(30),  
BAL money, PINNO int, MBNO char(10),  
CAddress varchar(50))

Date: 04-12-15, Friday

ACNO	CNAME	BAL	PINNO	MBNO	CAddress
1021	SAI	25000.00	8596	9874563214	AMEERPET
1022	KAMAL	32000	1122	8523697412	SRINAGAR
1023	GUPI	15000.00	NULL	7744112233	MADHAPUR

\* Create view SV2 AS Select ACNO, CNAME, BAL, MBNO,  
CADDRESS from customerDetails  
This Restricted column data is updated by  
the database administrator only.

Now it will create a view on those columns which  
are mention in the query at the time of view creation  
like below.

SV2				
ACNO	CNAME	BAL	MBNO	CAddress
1021	x	x	x	x
1022	x	x	x	x
1023	x	x	x	x

### Testing:

Insert SV2 values (1023, 'Gopi', 15000, '7744112233', 'MADHAPUR')

- In the above query we are inserting the values for ACNO, CNAME, BAL, MBNO and ADDRESS columns only.
- But we can't insert the values to the Restricted column (PINNO) in customerdetails Table through a SV2.

Ex 3: What is the use of creating a view for hiding or restricting a specific rows in Emp Table?

With check option: This keyword is used for verifying or check the values with the user defined condition at the time of view creation.

Q: Create view SV3 As select \* from Emp where salary=15000 with check option.

### Testing:

Insert SV3 values (105, 'Naresh', 14000, 21) → Not Allowed

Insert SV3 values (105, 'Naresh', 16000, 21) → Not Allowed

Insert SV3 values (105, 'Naresh', 15000, 21) → Allowed.

Note: When we create a view on a Table without with check option keyword then the view is Invalid. So that when we create a view for restricting the rows data of the original Table then we should views with check option (at this time view is valid)

Ex: WAG to create a view on multiple tables?

Create view CV1 AS Select S.STID, S.Sname, S.MBNO,  
C.CID, C.CName, C.CFee From Students S Inner Join  
Courses C On S.CID = C.CID.

Testing:

Insert CV1 values (110, 'Hello', '8522336529', 70, 'SAP', 3200)

update CV1 set STID=110, CName='HADOOP' where CID=10

Delete CV1 where STID=20

→ We can not perform the above DML statements on a complex view CV1. Because complex views are not supported DML statements.

\* Differences Between View and Synonym.

### View

1. It is a virtual ~~and~~ logical Table of the Base Table.
2. It doesn't contain physical existence and it doesn't occupy the memory.
3. It can create one multiple Tables at a time.
4. It will provide the security to the Original Table names and also data.
5. By using view we can hide or Restrict the specific rows and columns in the Table.

### Synonym

1. It is a Alias name of the Base Table.
2. It contains physical existence and occupy the memory.
3. It will create on a single Table only.  
A-It will provide security for the original Table names only but-not data.
5. But it is not possible by Using Synonym.

\* Syntax to Drop a view:

Drop view {View Name}

Eg: Drop view SV1

Date: 05-12-15, Saturday

\* Sub-Queries in SQL Server: A query contains another query within it is known as Sub-Query.

(6)

A select statement contains another other select stmt is also called as SubQuery.

- In Sub-Query we have two more queries those are
  - 1. Outer Query
  - 2. Inner Query.
- In Sub-Query Mechanism first The Inner Query is executed and later Outer Query will execute. It means that in SubQuery Outer query is always depends on the result of Inner Query.
- If a select statement contains two more select statements Then it is called as "multiple SubQuery".
- If a select statement contains more than two select statements is called as "Nested-SubQuery".
- we can write maximum the 255 select statements in the singleline SubQuery.
- If we find  $n^{\text{th}}$  record then we write  $n+1$  statements That means Sub-queries will work on  $n+1$  mechanism.

Q10: WAP to display employee details who are getting the first highest salary in the organization ?

Ans: SubQuery = outerQuery + InnerQuery

Inner Query: Select Max(Salary) From EmployeeDetails.

Outer Query : Select \* from Employee Details where  
Salary = (Inner Query)

Subquery : select \* from EmployeeDetails where  
Salary = (Select Max(Salary) From  
EmployeeDetails) → Subquery.

C Ex2: WA Subquery to display Employee SAI colleagues ?

Ans: InnerQuery: Select department from EmployeeDetails  
where Ename = 'SA1'

OuterQuery : Select \* From EmployeeDetails where  
Department = (InnerQuery)

SubQuery: select \* from EmployeeDetails where  
Department = (select Department from  
EmployeeDetails where Ename = 'SA1')

Ex3: Write a Subquery to display second Highest Salary Employee details from the Table?

Ans: Select \* from EmployeeDetails where salary = (select Max(Salary) from EmployeeDetails where salary < (select Max(Salary) from EmployeeDetails)) → multiple subquery.

Ex4: Write a Sub-Query to display employee details whose salary is greater than the maximum salary of Java Department,

Ans: Select \* from EmployeeDetails where salary > (select max(Salary) from EmployeeDetails where department = 'Java')

Ex5: Write a Subquery to retrieve employee details who are getting Third highest salary in the organization?

Ans: Select \* from EmployeeDetails where salary = (select Max(Salary) from EmployeeDetails where salary < (select Max(Salary) from EmployeeDetails where salary < (select Max(Salary) from EmployeeDetails))) → Nested subquery.

Ex6: write a SubQuery to get Employee details who are getting the first least salary in the organization?

Ans: Select \* from EmployeeDetails where salary = (select min(Salary) from EmployeeDetails)

C Eg: write a SubQuery to find the second least salary  
in the Table?

C Ans: Select \* from EmployeeDetails where salary =  
(Select min(salary) from EmployeeDetails where  
Salary > (select min(salary) from EmployeeDetails))

C \* Co-Related SubQuery:

C → In Co-Related subqueries first outer query is executed  
and later inner query will execute that means in  
Co-Related Inner Query is always depends on the result  
of outer query.

C → Co-Related subquery will work on (n-1) mechanism.

C Syntax:

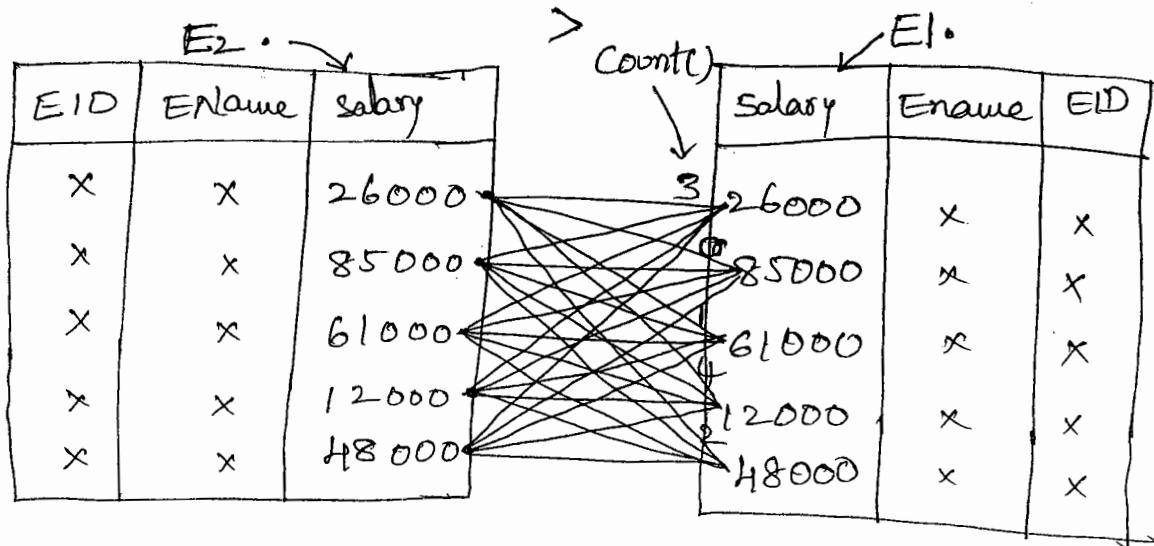
C Select \* from <TN> <Alias Name> where (n-1) =  
(Select count(<column name>) from <TN>  
<alias name2> where <alias name2> . <Column Name>  
(>/<). <alias name1> . <Column Name>)

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Date: 08-12-15, Tuesday

Ex: Write a Co-related Sub-Query to find the first highest salary Employee details?

Ans: Select \* from Emp E1 where 0 = (select count  
(salary) from Emp E2 where E2.Salary > E1.Salary.)



Solutions: if n=1

$$(n-1) \Rightarrow 1-1 \Rightarrow 0$$

$$0 = 3 \Rightarrow X \text{ (false)}$$

$$0 = 0 \Rightarrow 85000 \checkmark \text{ (true)}$$

$$0 = 1 \Rightarrow X \text{ (false)}$$

$$0 = 4 \Rightarrow X \text{ (false)}$$

$$0 = 2 \Rightarrow X \text{ (false)}$$

Ex: write a Co-related subquery to find four highest salary Employee details?

Ans: Select \* from Emp E1 where 3 = (select count (salary) from Emp E2 where E2.Salary > E1.Salary)

Solution from the above Table

If  $n=4$

$$(n-1) = 4-1 = 3$$

$$3 = 3 \Rightarrow 26000 \checkmark$$

$$3 = 0 \Rightarrow X$$

$$3 = 1 \Rightarrow X$$

$$3 = 4 \Rightarrow X$$

$$3 = 2 \Rightarrow X$$

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Ex: write a Co-Related Subquery to find the 5<sup>th</sup> Least salary Employee details in the Table.

Ans: Select \* from Emp E1 where 4 = (select count (salary) from Emp E2 where E2.Salary < E1.Salary)

Solution from the above Table

If  $n=5$

$$(n-1) = 5-1 = 4$$

$$4 = 1 \Rightarrow X$$

$$4 = 4 \Rightarrow 85000 \checkmark$$

$$4 = 3 \Rightarrow X$$

$$4 = 0 \Rightarrow X$$

$$4 = 2 \Rightarrow X$$

## \* Indexes in SQL Server:

→ One of the most important routes to high performance in a SQL Server database is an Index.

→ It is a database object which is used to speed up the querying process by providing quick access to rows in the database Tables.

→ By using Indexes we can save time and can improve the performance of database queries and applications.

→ An Index contains Keys built from one or more columns in the table and map to the storage location of the specified data.

→ When we create an indexes on any column, SQL Server internally maintain a separate table called index Table.

→ So that whenever user trying to retrieve the data from existing Table depends on index Table SQL Server directly go to the table and retrieve required data very quickly.

→ In a Table we can use max 250 indexes. The index type refers to the way the index is stored internally by SQL Server. So a Table can contain the two types of Indexes.

1. Clustered Index
2. Non-clustered Index.

Date: 09-12-15, Wednesday

1. Clustered Indexed: The only time the data rows in a Table are stored in sorted (ascending order only) order structure is when the Table contains a clustered Index.

→ When a Table has a clustered index then is called a clustered Table. If a Table has no clustered index, its data rows are stored in an unordered structure.

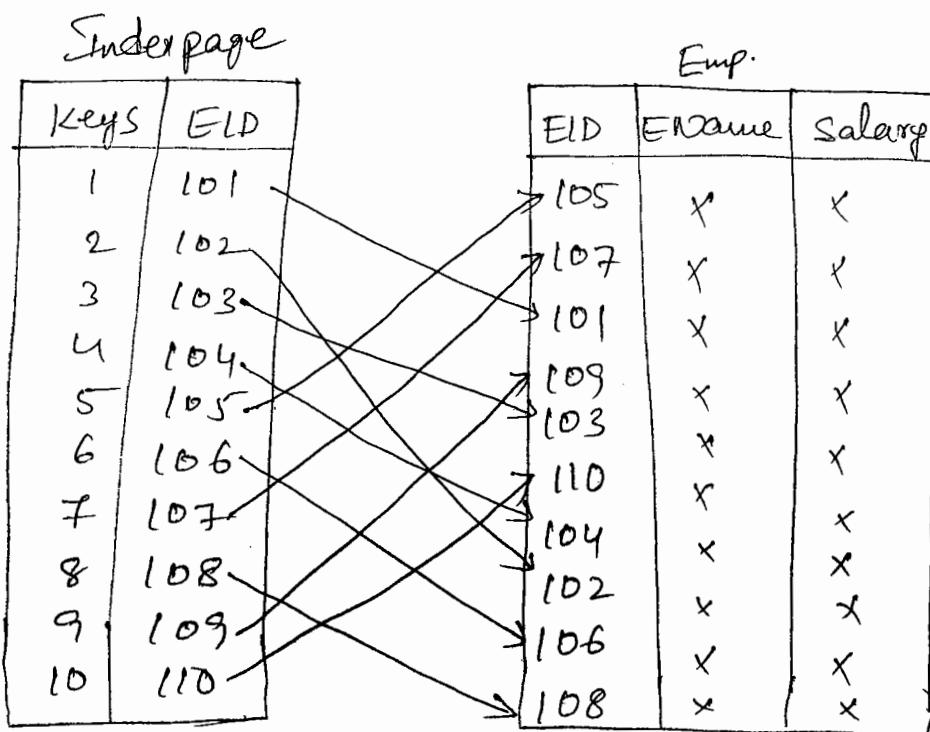
→ A Table can have only 1 clustered index on it, which will be created when primary key constraint is used in a Table.

Index page		(PK)	Emp		
Keys	EID		EID	EName	Salary
1	101	Mapping	101	X	X
2	102		102	X	X
3	103		103	X	X
4	104		104	X	X
5	105		105	X	X
6	106		106	X	X
7	107		107	X	X
8	108		108	X	X
9	109		109	X	X
10	110		110	X	X

2. Non-clustered Indexes: Non-clustered Indexes

- will not have any arrangement order (unordered structure) of the data in the Table. In a Table we can create 249 non-clustered indexes.

→ If we don't mention clustered index in a Table Then default is stored as Non-clustered Indexes.



### Syntax

create Index <IndexName> ON <TN> (columnName)

\* Example on Non-clustered Index Mechanism.

Step1: create Table Emp (EID int, ENAME varchar(20),  
SAL money)

Step2: create Index Ind1 ON Emp(EID)

→ To Execute the above query it will create a Non-clustered index on a Table Emp and will arrange the data in un-ordered manner like below.

Insert Emp values (103, 'CC', 52000)

Insert Emp values (101, 'AA', 45000)

Insert Emp values (105, 'EE', 42000)

Insert Emp values (102, 'BB', 78000)

Insert Emp values (104, 'DD', 10000)

Output:

EID	Ename	Salary
103	x	x
101	x	x
105	x	x
102	x	x
104	x	x

SRI RAJAVENKRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

\* Example on clustered Index Mechanism:

Step 1: Create Table Emp1 (EID int primary key, Ename Varchar(20), SAL Money)

Whenever we execute the above query it will create clustered Index on a Table Emp and will maintain the data in Order like below

Insert EMP1 values (103, 'CC', 52000)

" (101, 'AA', 45000)

" (105, 'EE', 42000)

" (102, 'BB', 78000)

" (104, 'DD', 10000)

Output:

EID	Ename	Salary
101	x	x
102	x	x
103	x	x
104	x	x
105	x	x

→ Advantages of primary key Constraint:

- It will use to avoid duplicate and Null values from the Table.
- It is used for to make the relationship between two Tables with Foreign key Reference.
- It will arrange the values in Sequential order.
- It is also used for retrieving the data with faster rate.
- To creating clustered Index on a Table.

Date: 10-12-15, Thursday

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Date: 10-12-15, Thursday

## T|SQL

(OR)

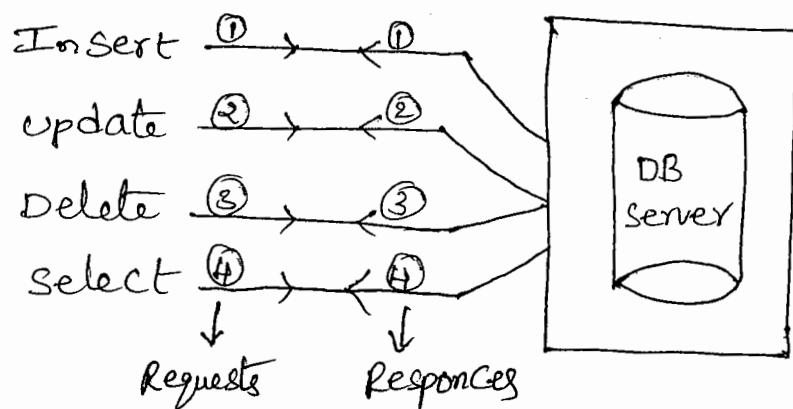
## PL|SQL

- T|SQL is stands for Transact extension of structure Query Language.
- T|SQL is a procedural Language whereas SQL is a Non-procedural Language.
- SQL supports single line stmt execution process only whereas T|SQL supports multiline stmts execution process.
- SQL doesn't support Conditional and control statements like if, if else, multiple if-else, nested if-else and while Loop control statement. But these are possible to implement under T|SQL.
- We can't implement Try and Catch handling methods in SQL whereas we can implement in T|SQL.
- Declaration of variables are not supported in SQL, but supported in T|SQL.
- SQL doesn't provide Re-usability and security facilities to the query information whereas T|SQL provide Reusability and security facilities by using The database objects are stored procedures, stored functions and database Triggers.

→ In T/SQL the user will write the statements under a block.

Block: Block is nothing but collection of statements which are executed as a single unit by the server.

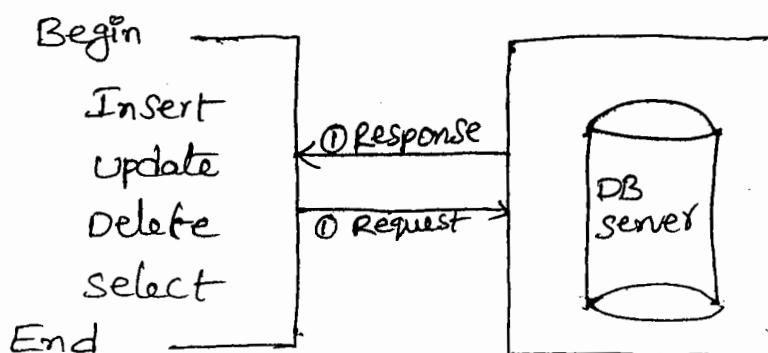
Query Execution process in SQL:



→ In this case every SQL statement is executed individually by the server so that the burden of the server will be increased and application performance will be reduced.

Query Execution process in T/SQL:

Block



Aggregation  
Iteration  
Selection  
Projection  
Join  
Grouping  
Ordering  
Limiting

→ In this case the SQL Stmt are executed as a single unit by the server so that the burden of the Server will be Reduced and Application performance will be increased.

### Types of Blocks :

→ In TSQL the user will interact with 2 types of blocks. 1. Anonymous Blocks  
2. Sub Blocks.

Date, 14-12-15  
Monday

#### Anonymous

1. It is UnNamed Block.
2. These are Temperry Blocks.
3. It will not saved into the DB.
4. Every time Compilation and execution of a program.
5. Anonymous Blocks are Time Consuming blocks.
6. It will not provide Re-usability and Security facilities.

#### SUB BLOCKS

1. It is a Named Block.
2. These are permanent Blocks.
3. These are saved into the DataBase.
4. These are pre compiled programs.  
(First time compilation only)
5. Sub Blocks are Time Saving blocks.
6. It will provide Re-usability and security facilities.

## \* Working with Anonymous Blocks:

→ When we working with Anonymous blocks we should follow the following steps.

Step1: Declare variables: when we declare the variables under PL/SQL blocks we should use declare keyword and prefixed symbol (@).

Syntax: Declare @<variable name><Datatype>[size],

Ex: Declare @X INT, @ENAME varchar(20), . . . . .

Step2: Assigning the values to the variables: when we assign the values to the variables we should use set keyword like below.

Syntax: SET @<variable Name> = value

Ex: SET @X=100      (or) Ex: SET @X=100; SET @ENAME  
SET @ENAME='SAI'      ↓      ='SAI'  
                          [If we want to write same line  
                          use Ex:2]

Step3: To Display the variable values (or) Message on the screen: when we display the variable values and message then we used print statement.

Syntax: PRINT <variable Name>

Ex: print @X  
     print @ENAME  
     print 'Hello'

Ex: write a T/SQL program for interchanging the variable values ?

Ans: declare @A int, @B int, @C int

set @A=100; set @B=200

set @C=@A

set @A=@B

set @B=@C

print 'A value is :-' + cast(@A AS varchar)

print 'B value is :-' + cast(@B AS varchar)

output: A value is :-200

B value is :-100

Ex: write a T/SQL program to input STID, sName and their subjects marks. find the Total marks of a students ?

Ans: Declare @STID INT, @sname varchar(20), @maths int,  
@phy int, @che int, @TM int

set @STID = 101; SET @sname='SAI'

set @mat = 98; set @phy = 87; set @che = 76

print 'student ID is :-' + cast(@STID AS char)

print 'student name is:-' + @sname

print 'Total marks are:-' + cast(@TM AS char)

output: student ID is =101

student name is =SAI

Total marks are = 261

\* **Cast()**: It is a pre-defined function which is used to convert from one datatype to other datatype.

**Syntax:** `Cast (@ <VariableName> As <Target Datatype>)`

**Ex:** `Cast (@x As int)`

`Cast (@x As varchar)`

`Cast (@x As char)`

`Cast (@x As money)`

**Q:** Write a T/SQL program to perform Arithmetic operations on the given two values?

**Ans:** Declare @x int, @y int, @z int

Set @x=12; ~~@y~~ set @y=2

Set @z=@x+@y

Print 'ADD is:-' + cast(@z as char)

Set @z=@x-@y

Print 'SUB is:-' + cast(@z as char)

Set @z=@x\*@y

Print 'MUL is:-' + cast(@z as varchar)

Set @z=@x/@y

Print ~~@z~~ 'DIV is:-' + cast(@z as varchar)

**Output:** Add is = 14

Sub is = 10

Mul is = 24

DIV is = 6

Ex: write a T/SQL program to find the biggest number  
Among the given two Numbers ?

⇒ Syntax for if-else:-

If (Condition)

< statements >

else

< statements >

Ans: Declare @x int, @y int

Set @x=20 ; @y=40

if (@x>@y)

print 'x is bigger Number'

else if (@x=@y)

print 'Both are equal'

else

print 'y is Big Number'.

Ex: write a T/SQL program to find even or odd Number ?

Ans: Declare @x int

Set @x=12

if (@x%2)=0)

print 'x is even'

else

print 'x is odd'.

**SRI RAGHAVENDRA XEROX**  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Ex: Write a T/SQL program to print 1-10 Natural Numbers?

→ Syntax for while-Loop: T/SQL supports only one looping statement i.e while.

while (Condition)

Begin

<Statements>

<Increment / Decrement>

End.

Ans: Declare @X int

Set @X=0

While (@X < 10)

Begin

Set @X = @X + 1

Print @X

End

Date: 15-12-15

Tuesday

CURSOR: Cursor is a memory location for storing database Tables.

→ Cursor is a temporary work area allotted to the client at server when a select stat is executed.

→ A cursor contains information on a select stat and the rows of data accessed by it.

→ This temporary work area is used to store the data retrieved from the database, and manipulate this data.

→ A cursor can hold more than one row but can process only one row at a time. The set of rows the cursor holds is called the "Result set".

### \* Types of cursors in TSQL:

Implicit cursors: These cursors ~~are~~ will be created by SQL Server by default when select stmt will be executed.

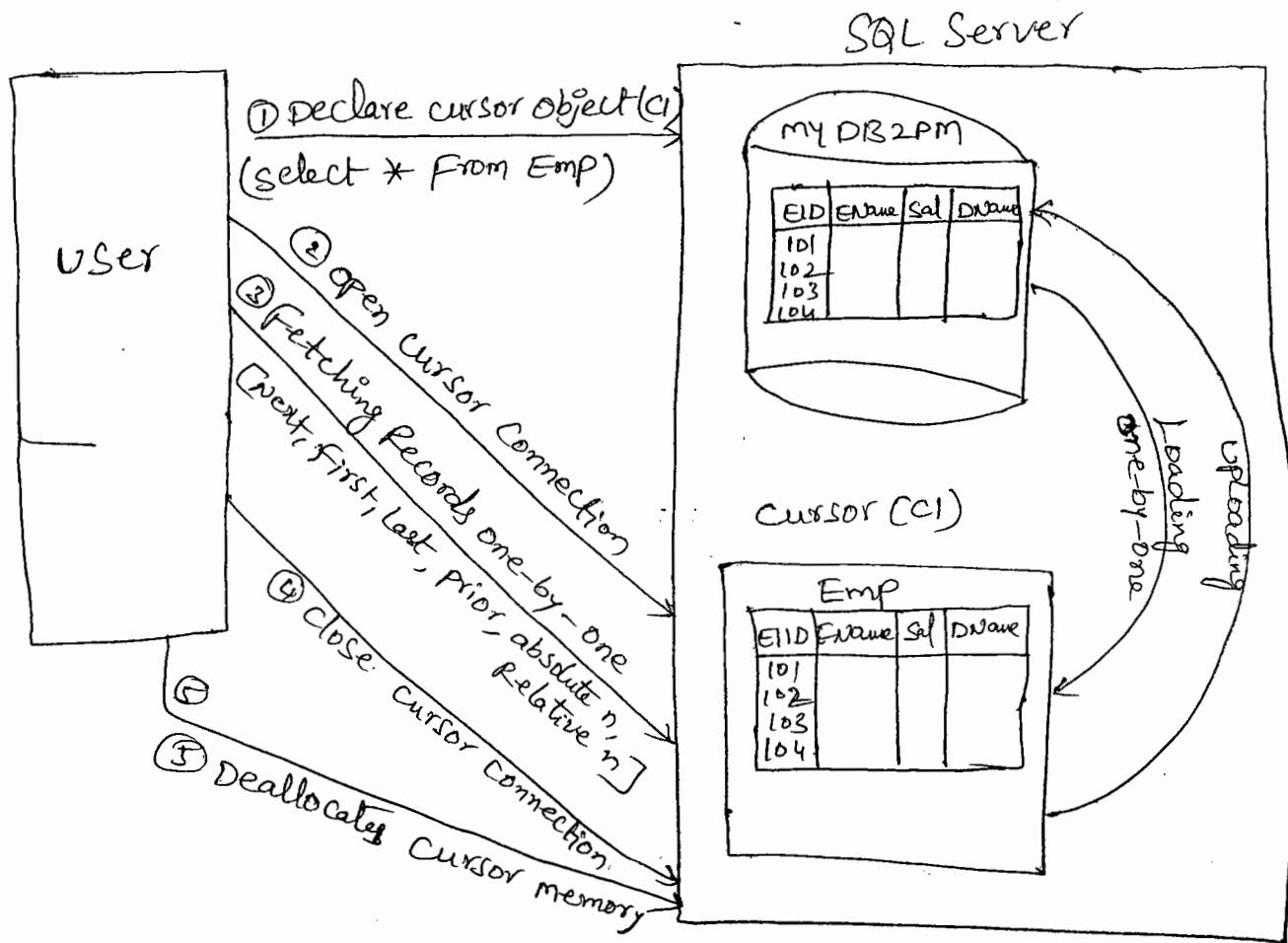
Explicit cursors: When user can create a memory location to store the tables then it is called as Explicit cursors.

→ These cursors will access the records in the Table record by record ~~one~~ or one by one only.

→ "Whenever we want to go for record by record manipulation then explicit cursors will be used".

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

## \* Working with an Explicit cursor:



## \* Steps to create Explicit cursor:

1. **Declaring a cursor:** In this process we define a cursor  
syntax: `Declare <cursor name> cursor for <select stmt>`

2. **open a cursor:** when we open a cursor it will  
internally execute the select stmt that is  
associated with the cursor declaration and  
load the data into cursor

Syntax:- `open <Cursor name>`

3. Fetching Data From the cursor: In this process we access row by row from cursor.

Syntax: Fetch

first / last / next / prior / absolute n / relative n from  
<cursorname> [into <variables>]

4. closing a cursor: In this process, it releases the current result set of the cursor leaving the data structure available for reopening.

Syntax: close <cursor name>

5. DeAllocate a cursor: It removes the cursor reference and deallocate it by destroying the data structure.

Syntax: Deallocate <cursor name>

@@Fetch\_Status: It is global variable use to check whether cursor variable contains records or not. if record is there then the value will be zero otherwise value will be -1.

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

\* Write a cursor program to display Employee details in row-by-row manner by using cursor variables?

Ans: Declare @EID Int, @Ename Varchar(20),  
@Sal Money, @DName Varchar(80),  
@Loc Varchar(30)                          ↳ Declaration of variables

declare c1 cursor for select \* from Employeedetails  
open c1  
fetch next from c1 into @EID, @Ename, @Sal,  
@DName, @Loc → Loop started

while @@fetch\_status = 0 [→ 0 if record/value is there]  
[→ -1 if no record.]

Begin

    print @EID  
    print @Ename  
    print @Sal  
    print @DName  
    print @Loc

Fetch next from c1 into @EID, @Ename, @Sal,  
@DName, @Loc → Loop is continued.

End  
close c1  
Deallocate c1

16-12-15, wednesday}

Note: Declaration of variables and selected columns in select query must and should be same in cursor declaration.

Ex: write a cursor program to fetch the Records in one-by-one Manner without using cursor variables?

Ans: Declare c1 cursor for select \* from EmployeeDetails

open c1

fetch next from c1 → Loop is Start.

while @@fetch\_status = 0

Begin

fetch Next from c1 → Loop is Continue.

END

close c1

Deallocate c1

Ex: write a cursor program to update employee salary as per the following criteria?

Ans: Dept Name Increment Salary %.

SAP 10% (0.1)

HR 8% (0.08)

FIN 5% (0.05)

Declare @EID int, @Dname varchar(20)

Declare c1 cursor for select EID, Dname from Emp

open c1

~~EMP~~

fetch next from c1 into @EID, @Dname

```

while @@FETCH_STATUS = 0
Begin
if @DName = 'SAP'
update Emp set salary += salary * 0.1 where EID = @EID
Else if @DName = 'HR'
update Emp set salary += salary * 0.08 where EID = @EID
Else
update Emp set salary += salary * 0.05 where
EID = @EID
Fetch Next from C1 Into @EID, @DName
End
Close C1
Deallocate C1

```

Ex: Write a cursor program to delete the records in One-by-one Manner From the Table ?

Ans:

```

Declare @EID Int
Declare C1 Cursor for select EID from Emp
Open C1
Fetch Next from C1 Into @EID
while @@Fetch_Status = 0
Begin
Delete from Emp where EID = @EID
Fetch Next from C1 Into @EID
END
Close C1
Deallocate C1

```

\* Fetching Methods: Explicit cursors are supported  
The following 6 fetching methods.

1. Next: It is a default method of cursor which can be used for fetching the records in forward directional only.

2. First: Fetching First record from the Table.

3. Last: Fetching Last record from the Table.

4. prior: Fetching the current record to previous record (Backward directional)

5. Absolute 'n': Fetching exact position record from the Table. (n represent position of the record).

6. Relative n: Fetching the records either incremental or decremental order. (n represent increment or decrement number).

Note: As per the Nature the cursors are again classified into 2 types those are

1. Forward only cursors.

2. Scroll cursors

\* Forward Only and scroll cursors: If a cursor is declared as forward only it allows you to navigate only to the next records in sequential Order and more over it supports only a singleton fashion method that is fetch Next one-by-one).

→ whereas a scroll cursor allows you to navigate/fetch bidirectionally that is top-bottom or bottom-top also. and it supports six different fetch methods are

- Fetch Next
- Fetch First
- Fetch Last
- Fetch Prior
- Fetch Absolute n
- Fetch Relative n.

Ex: write a cursor program for fetching the records from Last to first directional ?

Ans: Declare C1 cursor scroll for select \* from EmployeeDetails  
open C1

Fetch Last from C1 → Loop is start  
while @@fetch\_status = 0

Begin

Fetch prior from C1 → Loop is continued

End

Close C1

Deallocate C1

Date: 17-12-15, Thursday

Q: Write a Cursor program to fetch every third position record from the Table?

Ans: Declare @EID Int

declare c1 cursor scroll for select EID from EmployeeDetails

open c1

fetch first from c1 into @EID

while @@fetch\_status == 0

Begin

print @EID

Fetch Relative 3 from c1 into @EID

End

close c1

Deallocate c1.

Q: Write a cursor program to update the salary of first record and delete the last record from the Table?

Ans: Declare @EID Int

declare c1 cursor scroll for select EID from EmployeeDetails

open c1

fetch first from c1 into @EID

update EmployeeDetails set salary = 89000 where EID = @EID

fetch Last from c1 into @EID

delete from EmployeeDetails where EID = @EID

close c1

Deallocate c1.

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Q: Write a cursor program to delete the 5<sup>th</sup> position record from the Table?

Ans: Declare @EID Int

Declare c1 cursor scroll for select EID from EmployeeDetails

open c1

Fetch Absolute 5 from c1 into @EID

Delete from EmployeeDetails where EID = @EID

close c1

Deallocate c1

Q: Write a Cursor program to implement all Fetching methods?

Ans: Declare @EID Int

Declare c1 cursor scroll for select EID from EmployeeDetails

open c1

fetch last from c1 into @EID

print @EID

fetch prior from c1 into @EID

print @EID

Fetch Absolute 3 from c1 into @EID

print @EID

Fetch Relative -2 from c1 into @EID

print @EID

~~closed~~ fetch first from c1 into @EID

~~deallocated~~ print @EID

Fetch Next from c1 into @EID

print @EID

close c1

Deallocate c1

18-12-15, Thursday

\* SubBlocks or Subprograms: A subprogram is a named block of code that is directly saved on the server and it can be executed when and where it is required. ~~also~~ we have 3 types of subprograms objects in SQL Server.

1. stored procedures / procedure
2. stored functions / Functions
3. Database Triggers

\* stored procedures: A stored procedure is a database object which contains precompiled queries. stored procedures are a block of code designed to perform a task whenever we called.

why we need stored procedures: whenever we want to execute a SQL query from an application the SQL query will be first compiled for execution where the process of compiling is time ~~consuming~~ consuming because Compiling occurs each and every time we execute the query or statement.

→ To Overcome the above problem we write SQL statements or query under stored procedure and execute, because a stored procedure is a pre compiled block of code without compiling the statement gets executed whenever the procedures are called which can increase the performance of an application.

\* Advantages of stored procedures: As there is no unnecessary compilation of queries, this will reduce burden on database.

- Application performance will be improved.
- User will get quick response.
- Code Reusability and security.

Date: 19-12-15, Saturday

### stored procedures with parameters

create procedure <PName>  
declaration  
block { @<variable/parameter Name>  
(optional) <Datatype> & [size],  
-----  
-----

Execution  
block  
(mandatory)  
<statements / queries>  
End

### stored procedures without parameters

create procedure <PName>  
Defined  
Block  
(mandatory)  
AS  
Begin  
<statements / queries>  
End.

- Once the procedure is created it is physically saved on the server as a database object which can be called whenever we required to the user.
- we can call the above procedure from anywhere and from any application that is developed using JAVA or .NET Languages.

\* Syntax to call a stored procedure:

Execute <PName> values

(or)

Exec <PName> values.

Ex1: Create a procedure to print welcome statement on the screen ?

Ans: Create procedure SP1

AS

Begin

print 'welcome to stored procedures'

END.

O/P syntax: Execute SP1      outputs welcome to stored procedures.

(or)

Exec SP1

Ex2: Create a procedure for adding the two numbers ?

Ans: Create procedure SP2

@X Int, @Y Int → Input parameters.

AS

Begin

Declare @Z Int → output parameters.

Set @Z = @X + @Y

Print 'ADD IS :-' + Cast (@Z As char)

End

O/P syntax   Execute SP2 10, 20      O/P :- ADD IS :- 30.

Ex3: Create a procedure to display employee details?

Ans: Create procedure SP3

AS

Begin

Select \* from EmployeeDetails

END

O/PQuery: Exec SP3

Ex: Create a procedure to input employee ID and display the employee ID details from the Table?

Ans: Create procedure SP4

@PEID INT → Input parameter.

AS

Begin

Select \* from EmployeeDetails where EID = @PEID

END

O/PQuery: Exec SP4 101

Ex: Create a procedure to accept customer Account Number and update the customer details in the Table?

Ans: Create procedure SP5

@PACNO int, @PCNAME Varchar(30), @PBAL money,

@PPINNO int, @PMBNO char(10), @PADD Varchar(50)  
↳ Input parameters.

AS

Begin

Update customerdetails set cname = @PCNAME,  
Bal = @PBAL, PINNO = @PPINNO, MBNO = @PMBNO,  
EAddress = @PADD where CACNO = @PACNO  
END.

O/P:

O/P Query or Execution: Exec SP5 1022, 55000, 1122,  
'8899776655', 'UP'.

Date: 21-12-15, monday

Ex 6: Create a procedure to accept employee name & update  
empSal in the database Table?

Ans: Create procedure SP6

@Ename varchar(20), @sal money

AS

Begin

update employee details set sal = @sal where Ename = @Ename

End.

Execution: Exec SP6 'HARI', 19000

Ex 7: Create a procedure to accept Branch code and delete  
the Branch details from the Table?

Ans: Create procedure SP7

@PBCode int

AS

Begin

Delete from Branchs where BCode = @PBCode

End

Execution: Exec SP7 1023

select \* from Branchs.

Ex 8: Create a procedure to ~~accept~~ input the new values  
into the Branch Table?

Ans: Create procedure SP8

@PBCode int, @PBName varchar(20), @PLOC varchar(30)

AS

Begin

Insert Branchs values (@PBCode, @PBName, @PLOC)

End

Execution: Exec SP8 1021, 'ICICI', 'HYD'.

select \* from Branchs.

v.v Imp

Ex: Create a procedure to input the values into multiple tables at a time?

Ans: Create Table Emp (EID int, EName varchar(30),  
                          Sal money, DNO int)

Create Table Dept (DNO int, DName varchar(30), DLoc  
                          varchar(30))

Create procedure SP9

@PEID int, @PENAME varchar(30), @PSAL money, @PDNO  
int, @PDNAME varchar(30), @PLOC varchar(30).

As

Begin

Insert Emp values (@PEID, @PENAME, @PSAL, @PDNO)

Insert Dept values (@PDNO, @PDNAME, @PLOC)

End

Execution: Exec SP9 101, 'AA', 25000, 10, '.Net', 'HYD'

Output:

Emp			
EID	EName	Sal	DNO
101	AA	25000	10

Dept		
DNO	DName	DLoc
10	.Net	HYD

\* Syntax to Drop a stored procedure from the Database :

Syntax: Drop procedure <PName>

Ex: Drop procedure SPI

\* Stored Functions Or User Defined functions :-

→ Function is a block of code similar to procedure which can be used to perform some operations and returning a single value to the user.

→ The user can create two types of functions Those are

(i) scalar valued functions

(ii) Table valued functions.

(i) Scalar valued Functions:- In this case the function will return a single column value to the user.

Syntax: Create Function <FName>(@ <parameter/variable name>  
<datatype>[size])  
Returns (<Attribute / column, datatype>)  
AS  
Begin  
<Body / statements / query>  
Return (<Attribute / column name>)  
End

\* Syntax to call a scalar valued Function (SVF) :-

~~Select~~ select dbo.<FName> <value>

Ex1: Create a SUF to returns the cube of a given Number ?

AnS Create Function SVFI (@x Int)

Returns Int

A8

Begin

Return  $\text{@}x * \text{@}x * \text{@}x$

End

Execution: Select dbo.SVF1(2) → O/P = 8

select dbo.SUF1 (5)  $\Rightarrow$  o/p = 125.

Ex2: Create a SUF to accept the Employee ID and returns that employee salary from the Table?

Ans: create Function SVF2 (@EID Int)

Returns money

As

Begin

Declare @sal Money

**SRI RAGHAVENDRA XEROX**  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

```
Select @sal=sal from employee details where  
Return @sal  
End.
```

EID = @EID

Execution: Select dbo.SVF2(101);  $\Rightarrow$  O/P = 98000

Ex3: Create a SVF to accept Employee Name and return  
That employee Dept Name from The Table?

Ans: Create function SVF3 (@Ename varchar(30))  
Returns varchar(20)

AS

Begin

Declare @DName varchar(30)

Select @DName = DName From Employee details where  
Ename = @Ename

Return @DName

END

Execution: Select dbo.SVF3('SAI')  $\Rightarrow$  O/P = .Net.

Date: 22-12-15, Tuesday

(ii) Table valued Functions:- In this case the function  
will returns more than one column value from the Table.

Syntax: Create Function <Fname>(@<variable/  
parameter name> <datatype> [size])  
Returns Table  
AS  
Return (<select query>)

\* Syntax to call a Table valued Function (TVF):-

Select \* from <Fname>(value)

Ex1: Create a Table valued Function to accept a Employee number and return that employee details from The Table ?

Ans: Create function TVF1 (@PEID Int)

Returns Table

As

Return (Select \* from EmployeeDetails where EID = @PEID)

Execution: select \* from TVF1 (101)

Ex2: Create a TVF to accept Dept.Name and display The list of Employee working in that department ?

Ans: Create function TVF2 (@PDeptName varchar(20))

Returns Table

As

Return (Select \* from EmployeeDetails where DName = @PDeptName)

Execution: select \* from TVF2 ('.net')

⇒ or ('HR')  
TUF2 ('HR')

Ex3: Create a TVF to accept customer A/c NO and Returns That customer balance amount and address from The Table ?

Ans: Create function TVF3 (@ACNO Int)

Returns Table

As

Return (Select Bal, Address from customerdetails  
where CACNO = @ACNO)

Execution:- select \* from TVF3 (1022) ⇒ O/P = 55000 'UP'.

Ex 4: Create a TUF to accept the course ID and returning the course ID students details from students, courses Tables?

Ans: Create function TVF4 (@CID Int)

Returns Table

As

Return (Select S.STID, S.SName, S.MBNO, C.CID,  
C.Cname, C.CFee from Students S

Inner join Courses C on S.CID = C.CID where  
CID = @CID)

Execution: Select \* from TVF4 (10)

\* Syntax to Drop a function :-

Drop function TVF1 / SVF1

\* Differences Between functions and procedures :-

- A Function must return a value whereas procedure Never Returns a value.
- A procedure can have parameters of both input (with parameters) and output (without parameters) whereas a Function can have only Input (with parameter) parameters only.
- In procedure we can perform select, Insert, update and Delete operation whereas function can used only to perform Select cannot be used to perform Insert, update and delete operations.
- we call a procedure using Execute Command whereas as function is called by using "Select Command only".

## Data Base Triggers

\* Database Triggers (parameterless objects) :- A Trigger is a special type of procedure that will be used to provide restriction on the Tables and database when a language "Commands" are Executed. SQL Server includes two types of Triggers. Those are

- ① DML Triggers.
- ② DDL Triggers.

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Date: 23-12-15,  
Wednesday

① DML Triggers: DML Triggers Execute when the user tries to modify or change data through manipulation language Events. Those are Inserting, Update and delete statements on the Table.

→ DML Triggers can be used to enforce business rules and data Integrity. With the help of a DML Trigger, we can enforce integrity which cannot be done with constraints.

Syntax: Create Trigger <TriggerName> on <TN/objectname>  
After / For <Insert / update / delete>  
(New) (Old)  
As  
Begin  
<Trigger Body / statements>  
END.

- Note :- ① Triggers are parameterLess objects.  
② There is no syntax to call in Trigger object because whenever the user execute the DML operations then the related Trigger is invoked automatically.

Ex1: Create a DML Trigger which will restrict insert operation on Branch Table ?

Ans: Create Trigger TR1 on Branchs  
After Insert  
AS  
Begin  
print 'we cannot perform Insert operation'.  
Rollback Transaction  
END.

Test:  
Insert Branchs values(1023, 'HDFC', 'Kerala')  
↳ Not Allowed.

Ex2: Create a Trigger which will restrict the update operation on a Table?

Ans: Create Trigger TR2 on Branchs  
After Update  
AS  
Begin  
print 'we cannot perform update operation'  
Rollback Transaction → Cancel the Transaction  
End

Test:

update Branchs set BLOC='PUNE' where BCODE=1022  
↳ Not Allowed.

Ex3: Create a Trigger which will restrict the all DML operations on a Table?

Ans: Create Trigger TRALL on Branchs

For Insert, update, delete

AS

Begin

Print 'DML operations are Invalid'

Rollback Transaction

END.

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Note: Now we cannot perform the DML operations on a Branch Table. Because those operations are restricted with a Trigger TRALL.

Ex4: Create a Trigger which will Restrict all DML operations on wednesday?

Ans: Create Trigger TRDAY on Emp

After Insert, update, Delete

AS

Begin

If Department(DW, Getdate()) = 4

Begin

Print 'DML Operations are not allowed'

ROLLBACK Transaction

End

End

$\therefore$  SUN = 1  
MON = 2  
TUE = 3  
WED = 4  
THU = 5  
FRI = 6  
SAT = 7

Ex: Create a Trigger which is restricted the DML operations before 4PM ?

Ans: Create Trigger TRTIME on Emp  
For Insert, Update, Delete

∴ 24 Hours format  
4PM = 16 PM.

AS

Begin

If Department (Hh, GetDate ()) < 16

Begin

Print 'Invalid Time'

Rollback Transaction

END

END

Date: 26-12-15, <sup>Saturday</sup> Thursday

\* Magic Tables in SQL Server:- SQL Server allows you to define a Magic Table. Magic Tables are Invisible Tables or virtual Tables. You can see them only with the help of Triggers in SQL Server.

→ Magic Tables are those Tables which allow you to Holding inserted, deleted and updated values during insert, delete and update DML operations on a table in SQL Server.

→ Basically there are two types of magic table in SQL Server. Namely Inserted and Deleted magic Tables. Update can be performed with

help of these two.

## 1. Inserted Magic Table:

→ when we insert the values into the Table then those values we can see in Inserted magic Table like below.

Create Trigger TRI on Branch  
For Insert

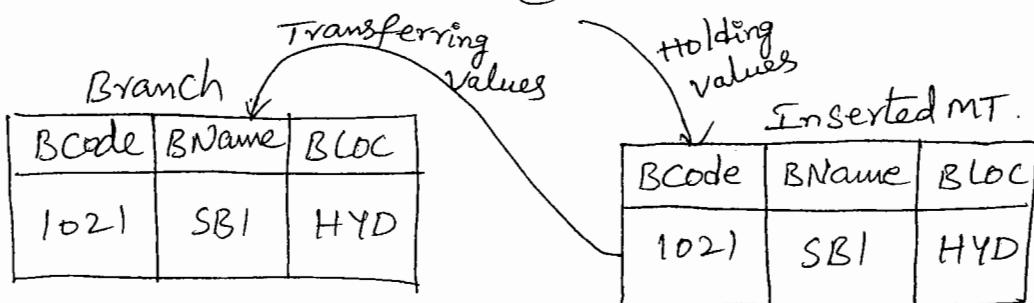
AS

Begin

Select \* from inserted  
END.

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

Ex: Insert Branch values (1021, 'SBI', 'HYD') ↳



## 2. Deleted Magic Table:

→ when we delete the data from the Table the deleted data can see in deleted magic Table like below.

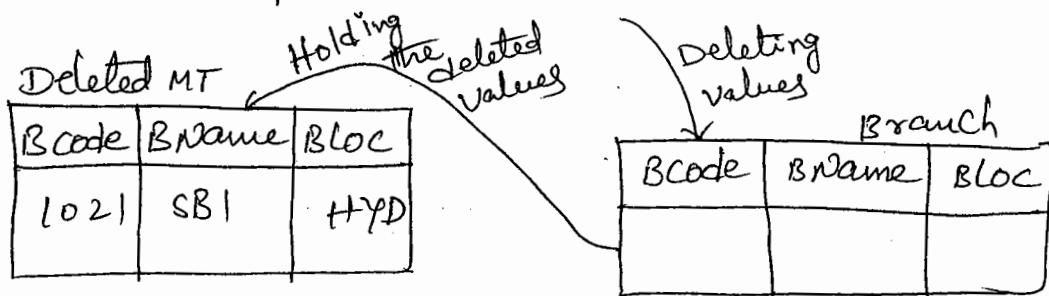
Create Trigger TRD on Branch  
For Delete

AS

Begin

Select \* from deleted  
END.

Ex: Delete from Branch where BCode = 1021 ↳



\* Updating the Values in the Table:

→ when we updated the value in the Table then  
the old value we can see the deleted magic Table  
~~and~~ and a new value can see in Inserted  
magic Table like Below.

create Trigger TRUP on Branch  
for update

AS

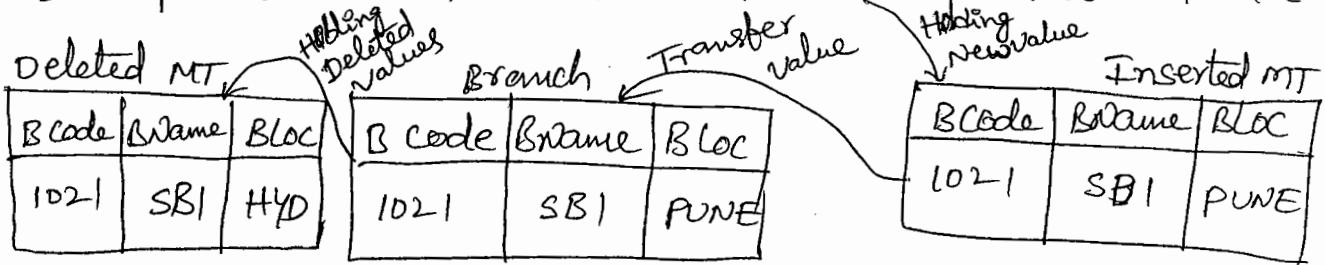
Begin

Select \* from deleted

Select \* from inserted

END.

E: update Branch set BLoc = 'PUNE' where BCode = 1021 ↳



\* Create a trigger which will Convert the Branch Name and Location into LowerCase characters when The user Inserting any case characters ?

Ans: Create Trigger TRLOWER on Branch  
After Insert

As

Begin

Declare @BCode Int, @BName Varchar(10),  
@BLoc . Varchar(20)

Select @BCode = BCode, @BName = BName,  
@BLoc = BLoc from inserted

update Branch set BName = Lower (@BName),  
BLoc = Lower (@BLoc), where BCode = @BCode

END

\* Create a Trigger to restrict updating the salary of an employee if The new salary is Less than to old salary ?

Ans: Create Trigger TRSAL on ~~Branch~~ Emp  
After update

As

Begin

Declare @newsal Money, @oldsal Money

Select @oldsal = sal from deleted

Select @newsal = sal from inserted

If (@newsal < @oldsal)

Begin

print 'New salary should not be less than old salary'

Rollback Transaction

END

END

\* Create a Trigger to restrict Delete operation  
if the EmployeeName = SAI ?

Ans: Create Trigger TRDEL on Emp

After delete

AS

Begin

Declare @Ename varchar(30)

Select @Ename = Ename From deleted

If @Ename = 'SAI'

Begin

Print 'Not Allowed'

RollBack Transaction

END

END.

SRI RAGHAVENDRA XEROX

Software Languages Material Available

Beside Bangalore Ayyagar Bakery,

Opp. CDAC, Balkampet Road,

Ameerpet, Hyderabad.

Execution: Delete from emp where ename = 'SAI' <sup>(Q1)</sup> 'HARI' <sup>(Q2)</sup>.

\* Syntax to draw a DML Trigger?

Syntax: DROP Trigger <TRIGGERNAME>

Ex: DROP Trigger TRDEL

\* DDL Triggers: DDL Triggers are used to restrict the DDL operations like Create, Alter and Drop Commands.

→ These Triggers are working on a particular Database.

Syntax:- Create Trigger <TriggerName> on database  
After/for (<Create\_Table, Alter\_Table,  
Drop\_Table>)

AS

Begin

<Trigger body / statements>;

End

Ex: Create a Trigger which will restrict the DDL operations on myDB5PM Database?

Ans: Create Trigger TRDDL on Database

After Create\_Table, Alter\_Table, Drop\_Table

AS

Begin

print 'DDL are not allowed on myDB5PM'

ROLLBACK Transaction

END.

\* Syntax to drop a DDL Trigger:

Syntax: Drop Trigger <Triggername> on Database

Ex: Drop Trigger TRDDL on Database.

Date: 27-12-15, SUNDAY

## Exception Handling

\* Exception Handling Mechanism in SQL Server:

- We handle errors of a program both in a programming Language as well as Databases also.
- whereas handling an error in a prog'g Language needs stopping the abnormal ~~for~~ termination and allowing the statements which are not related with the error to execute.
- whereas handling an error in SQL Server means stopping the execution of statements which are related with the error.

## Handling errors in SQL Server:

- From SQL Server 2005 we are provided with a structure error handling mechanism with the help of TRY and CATCH blocks which should be used as following.

Begin Try  
<statements>

```
End Try  
Begin Catch  
    <statements>  
End catch.
```

Ex1: Create a procedure for adding the two variable values using Try and Catch Implementation with User defined error statement ?

Ans: Create procedure SPADD  
@X Int , @Y Int  
AS  
Begin  
 Declare @Z Int  
 Begin Try  
 set @Z=@X + @Y  
 print 'ADD is:-' + cast(@Z AS char)  
 END Try  
 Begin Catch  
 print 'Second Number should not be zero'  
 END Catch  
End

Execution: Exec SPADD 10,2 → O/P: ADD is :- 5  
Exec SPADD 10,0 → O/P: Second Number  
should not be zero.

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

## \* Error\_Message:

→ It is a pre-defined method which will be used for displaying the information about the Exception which was raised in Try Block.

Ex: Create a procedure for adding two variable values by using Try catch implementation with System defined Error message?

Ans: Create procedure SPADDI

@X Int, @Y Int

AS

Begin

Declare @Z Int

Begin Try

Set @Z=@X/@Y

Print 'ADD IS:- ' + Cast (@Z as char)

END Try

Begin Catch

Print Error\_Message()

END Catch

END.

Execution: Exec SPADDI 10,2 → Output ADD IS:- 5

Exec SPADDI 10,0 → Divided by zero Error encountered.

\* Authentication :- Authentication is a process of verifying the credentials (UID & password) of a user to login in to the system.

\* Authorization :- Authorization is process of verifying whether the user <sup>has</sup> permissions to perform any operation on the database.

\* Data Control Language :- DCL Commands are used to enforce database security in multiple user's database environment.

These are two types .

- ① GRANT
- ② REVOKE .

1. Grant :- Grant command is used for giving a privilege or permission for a user to perform operations on the database.

Syntax :- GRANT < privilege name > on < object name >  
TO {user} [WITH GRANT OPTION]

privilege name :- used to granted permission to the users for some rights are ALL, EXECUTE and SELECT .

Object Name: It is the name of the database objects like Table, views and stored procedures etc--.

User: used for to whom an access rights is being granted .

with Grant option: Allows a user to grant access rights to other users .

Revoke Command removes user access rights / privileges to the database or taking back the permission that is given to a user .

Revoke <privilege name> on <ObjectName>  
from {user}

Note: The above two operations are performed by the Database Administrator in real time .

\* How to Transfer the Database from one System and other System OR one Location to other Location:

→ when we Transfer the database from one to an other system we have the two ways Those are

1. By using data files (.mdf & .ldf files)
2. By Using Backup file (.bak file)

ii. Using Data files: when we Transfer the DB Information by using Data files then we need to follow the two methods. Those are

1. Detaching method
2. Attaching method.

1. steps to Detaching the Database from Server location:

- 1. Go to open SQL Server Management studio.
- 2. Goto Object Explorer window.
- 3. Goto Databases folder.
- 4. Select the Required Database which we want to Detach from the Server Location (myDB2pm)
- 5. click on Rightmouse button.
- 6. Goto Tasks option
- 7. click on Detach option
- 8. Activate drop and update checkboxes and click on OK.

→ After Detaching the Database from the Server Location Then we have to go to the Root Location (C:drive) where .mdf file and .ldf files are available, select those files cut and paste into a specific folder and stored into a particular Drive in the System.

## 2. Steps to Attaching the Database into Server Location:

1. Goto open ssms
2. Goto object Explorer window
3. Select databases folder and click on Rightmouse button
4. Click on Attach option
5. Click Add button
6. Goto the Location where we stored .mdf & .ldf files
7. Select .mdf Extension file.(MyDB2pm.mdf)
8. Click on OK.
9. Click on OK.
10. Now go to check under databases folder.

## \* By Using Backup file (.bak) :-

→ whenever we Transfer the Database by using backup file then we should follow the two methods those are 1. creating a Backup file  
2. Restoring the Backup file.

### 1. Steps to create a Backup file :-

1. Goto open ssms.
2. Goto object Explorer window
3. ~~Select~~ Goto databases folder
4. Select the Required Database ~~on~~ on which Database we created backup
5. Click on Rightmouse button
6. Goto Tasks option

7. click on Backup option
  8. click on OK.
  9. once we click on OK we get a message is the backup of database myDB2pm completed successfully.
2. Steps to Restore the Backup file:
1. Goto open ssms
  2. Goto object Explorer window
  3. select databases folder and click on Right mouse button
  4. Click on Restore Database option
  5. Select Device Radio button
  6. click on Browse Button.
  7. click on Add button.
  8. Goto the Location where we stored .bak file
  9. select .bak extension file from the Location
  10. click on OK.
  11. click on OK.
  12. Goto options at left side top of the corner
  13. Select a checkbox i.e override ~~the~~ existing DB
  14. click on OK
  15. when we click on ok we get a message i.e Database mydb2pm Restored successfully.
  16. click on OK.
  17. Goto the Databases folder and check it.

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

## Reporting Services

- SSRS is one of the MSBI tool. which can be used to generate or create the various kinds of documents such as MS-word document, MS-Excel document, .pdf document, ~~or~~ .xml document, .tiff documents etc. (tiff) ~~etc~~
- Report is nothing but a document which is used for storing some business related information of ~~any particular organization.~~  
~~particular organization.~~
- When we interact with SQL Server Reporting Service tool then we need a separate developing Environment (BIDS) i.e. BIDS (Business Intelligence Development Studio).
- To get the BIDS Environment when we download SQL Server Data Tools from [www.microsoft.com](http://www.microsoft.com) website.
- After Downloading the above tool then we installed into the system and later we get a icon ie SQL Server Data Tools Icon. this icon will provide BIDS Environment to the user.

## \* Working with The Reports :-



## \* Steps to Creating the Reports :-

1. Goto All programs
2. Goto Microsoft SQL Server 2014
3. Click on SQL Server Data Tools Icon
4. Goto file
5. Goto New
6. Click on project
7. Goto Business Intelligence option at the left side top of the corner.
8. Select Reporting Services
9. Select Report Server project Template .
10. Change the Report Server project Name as Employee Details project and saved into a particular location in the system.

SRI RAGHAVENDRA XEROX  
 Software Languages Material Available  
 Beside Bangalore Ayyagar Bakery,  
 Opp. CDAC, Balkampet Road,  
 Ameerpet, Hyderabad.

11. - click on OK
12. Goto Solution Explorer window
13. Select Reports folder and click Right mouse button.
14. Goto Add option
15. click on New Item option.
16. Select Report Template and change the Name as RDLEmployee details.rdl where rdl stands for Report describe Language.
17. Click on Add button.
18. go to Report Data window.
19. Click on New option
20. Click on Dataset option.
21. select a Radio button i.e use a dataset embedded in my Report.
22. Click on New button.
23. Click on Edit button
24. Enter Server Name as either dot, or systemname or Local host.
25. Select either windows authentication or SQL Server authentication.
26. Select the Required DB from Connect to a database option (myDB 2pm)
27. Click on OK.
28. Click on OK.
29. Enter The select Query in Query editor option like below

Select \* from employeedetails

30. click on OK.
31. Go to ToolBox. Drag and Drop TextBox and Table controls like below.

EID	Ename	Sal	Dname	Loc
[EID]	[Ename]	[Sal]	[Dname]	[Loc]

32. Go to click on preview option in the Environment
33. Once we click on preview option then the Database Tables data is loading into the dataset memory at client side application.
34. Goto Export button.
35. click on word option.
36. Click on Save button.
37. Now Goto the Location where we saved MS-word document and open it.
38. Similarly we will Create ms-Excel, .pdf, .xml files etc.

SRI RAGHAVENDRA XEROX  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

— END —

— Panchalaiah  
2PM Batch.

**SRI RAGHAVENDRA XEROX**

**Software Languages Material Available**

**Beside Bangalore Ayyagar Bakery,**

**Opp. CDAC, Balkampet Road,**

**Ameerpet, Hyderabad.**

SRIRAGHAVENDRA  
Software Languages Material Available  
Beside Bangalore Ayyagar Bakery,  
Opp. CDAC, Balkampet Road,  
Ameerpet, Hyderabad.

**SRI RAGHAVENDRA XEROX**

**Software Languages Material Available**

**Beside Bangalore Ayyagar Bakery,**

**Opp. CDAC, Balkampet Road,**

**Ameerpet, Hyderabad.**