**Stored Procedure:**

Stored Procedure is a prepared sql statements that will write once and execute again and again while required

**Syntax:**

Create procedure Procedure\_Name @Var\_Name1 datatype(Size),@Var\_Name2 datatype(Size)

As

//statements

Go;

Or

Create procedure Procedure\_Name @Var\_Name1 datatype(Size),@Var\_Name2 datatype(Size)

As

//statements

;

**To execute the Stored Procedure**

Exec Procedure\_Name @Var\_Name1 = value1,@Var\_Name2 = value2;

**Ex:**

1.To Fetch the all student list

CREATE PROCEDURE ALLSTUDENT

AS

SELECT \* FROM STUDENT;

EXEC ALLSTUDENT;

2.To fetch the only cse student

CREATE PROCEDURE CSESTUDENT

AS

SELECT \* FROM STUDENT S INNER JOIN DEPT D

ON S.DNO = D.DNO

WHERE S.DNO IN ('CSE01','CSE02');

EXEC CSESTUDENT;

3.To fetch the IT students list

CREATE PROCEDURE ITSTUDENT @DNO CHAR(5)

AS

SELECT S.SID,S.NAME,S.FEE,S.DNO,D.DNAME,D.DLOCTION FROM STUDENT S INNER JOIN DEPT D

ON S.DNO = D.DNO

WHERE S.DNO = @DNO;

EXEC ITSTUDENT @DNO = 'IT001';

SELECT \* FROM STUDENT INNER JOIN DEPT

ON STUDENT.DNO = DEPT.DNO;

4.To fetch the Cse,It and Ece students list

CREATE PROCEDURE CSEANDITANDECE @DNO1 CHAR(5),@DNO2 CHAR(5),@DNO3 CHAR(5)

AS

SELECT \* FROM STUDENT S INNER JOIN DEPT D

ON S.DNO = D.DNO

WHERE S.DNO IN (@DNO1,@DNO2,@DNO3);

EXEC CSEANDITANDECE @DNO1 ='CSE01',@DNO2 = 'IT001',@DNO3 = 'ECE01';

**Procedure for Crud operations:**

**//**

**//**

**//**

**View:**

View is a virtual table created for the existing table.

View is only used for retrieving the data from the database by using view we cannot update the table ,we cannot not insert the new records and also cannot be delete also.

**Advantages:**

View is used for data security propose.

Normally view is used when we want to give the accessing of data to user who don’t know to organize relational database.Since, view only for accessing the data original data not able to modify.

**Syntax:**

Create view View\_Name

As

//statements

;

Ex:

Create view StudentView

As

Select \* from Student;

**IDENTITY SET:**

Identity set is used for automatic increment of values in the specified column.

Syntax:

Col\_name datatype IDENTITY(arg1,arg2);

Arg1 – Arg1 is the value to start the column.

Arg2 - Arg2 is the value to increment for next row or record.

**How to off and on the IDENTITY SET:**

SET IDENTITY\_INSERT TABLE\_NAME ON;//used to On the insert the data explicitly in to the identity column

SET IDENTITY\_INSERT TABLE\_NAME OFF;//used to Off the inserting the data explicitly to the identity column.

**Common Table Expression(CTE):**

A Common Table Expression (CTE) is a named result set in a SQL query

created from a simple SELECT statement that can be used in a subsequent SELECT statement.

In addition, as of SQL Server 2008, you can add a CTE to the new MERGE statement.

**Syntax:**

[ WITH <common\_table\_expression> [ ,...n ] ]

<common\_table\_expression>::=

expression\_name [ ( column\_name [ ,...n ] ) ]

AS

( CTE\_query\_definition )

Advantages:

CTEs,like database views and derived tables //where the *enable* users to more this are easily write and maintain complex queries

Than means increaseses readability and simplification.

**CTE vs View:**

CTEs are temporary result sets and view is a permanent object in the database:

CTEs are temporary result sets when we execute then only creates where as view is a permanent object in the database than are already present in the database that we can access the data.

CTEs are more efficient than temporary tables for small data sets.

BACKUP DATABASE

The BACKUP DATABASE statement is used in SQL Server to create a full back up/copy of an existing SQL database.

Syntax:

BACKUP DATABASE databasename  
TO DISK = 'filepath';

Ex:

BACKUP DATABASE testDB  
TO DISK = 'D:\backups\testDB.bak';

Microsoft SQL Server allows three basic types of SQL Server backup:

* Full backup.
* Differential backup.
* Transaction log backup.

The SQL BACKUP WITH DIFFERENTIAL Statement:

A differential back up only backs up the parts of the database that have changed since the last full database backup.

Syntax:

BACKUP DATABASE databasename  
TO DISK = 'filepath'  
WITH DIFFERENTIAL;

Ex:

BACKUP DATABASE testDB  
TO DISK = 'D:\backups\testDB.bak'  
WITH DIFFERENTIAL;