**Select:** select is a DML command that is used to fetch record that is already saved in table.

Select is used to fetch a part of data from the complex data,

Conditions are applied with select command to select to fetch any specific data from teble.

**Syntax to select all columns of all rows or table:**

Select \*from table\_name;

**Syntax to select specific columns of all rows or table:**

Select column\_name1,column\_name2…. From table\_name;

To select specific rowa we apply conditions with selected command then only those rows will be selected which satisfied the given condition;

**Syntax to select specific columns of specifc rows or table:**

Select column\_name from table\_name where<condition>

Types of Conditions🡪

=

>:

Ex.

Select \*from employee where salary>20000;

<:

Ex;

Select \*from employee where salary<20000;

>=

Ex;

Select \*from employee where salary>=20000;

<=

Ex;

Select \*from employee where salary<=20000;

* 🡪is not equal to

Ex;

Select \*from employee where salary<>20000;

And ---used when bot condition are true

Ex.

Select emp\_name,salary from employee where department=’php developer’ and ‘android developer’

Or ------------used when any one condition satisfied.

ex;

Select emp\_name,salary from employee where department=’php developer’ and ‘android developer’

In

Not in

Between

Is null

Is not null

LIKE🡪THIS OPERATOR IS USED TO COMPARE DATA with part of data or pattern.

When you have a concept of data or have a pattern behalf of that you want to select data from table then like operator is used with condition.

%->ignore character

Ex: where name like ‘%singh’ :select all rows where name ends with singh;

Where college name like **‘feroze%’ :select a**ll rows where college name starts with feroze.

Where college like **‘%gandhi%’ :**  select all rows where Gandhi present somewhere.

* 🡪ignores only a single character

**Ex: where dob like ‘2021-\_\_\_\_\_’**

DBMS-DATABASE MANAGEMENT SYSTEM

**SQL-Structure query language.**

**Columns->column\_name,data\_types,size,integrity constraints(not null,default,check,unique key,primary key,foreign key);**

**DDL🡪CREATE ALTER DROP TRUNCATE,RENAME;**

**DML🡪SELECT INSERT UPDATE DELTE;**

**TCL🡪 COMMIT,ROLLBACK SAVE TRANSACTION**

**DCL🡪 GRANT REVOKE**

**ORDER BY🡪** in table data is saved in randam manner, if you want to arrange data dehalf of a column then **order by**  is used.

Numeric-🡪 assending(small to larger),descending

String🡪 assending(a-z) and descending(z-a)

**Select \*/column\_name from table\_name where <condition> order by column\_name <desc>**

**Ex:**

**Select \*from table\_name where <condition> order by id desc;**

**<FUNCTION>**

**Aggregate Function:🡪**Aggregate function are some function that operate on multiple value that provide on single output.

**Sum(),count(),avg(),min(),max();**

\*It can be only used with select command.

\*To select a value as a column,

\*it can be applied only on numeric value.

Select sum(fee)

Select count(college)

Select max(table\_name) from table\_name

Select count(column\_name) from table\_name

Note:column function ignores null values. That means the column which contain NULL values is not counted by count() function.

**String function:**

**Len()🡪 length of string**

**Upper()🡪convert to uppercase**

**Lower()🡪convert to lowercase;**

**Ltrim()🡪triming from left side.**

**Rtrim()🡪remove right side spaces.**

**Substring()🡪**

**Date &Date Time function 🡪**

**Getdate()🡪 it**  is aa function which return current date and time.

**Ex:** select getdate()

**Current TIMESTAMP:🡪**  It is a property which returns current date and time.

Ex:

select getdate(),CURRENT\_TIMESTAMP

**datename()**

**ex🡪**

select datename (day,getdate())as day,datename(month,getdate())as month,datename(year,getdate()) as year,datename(dayofyear,getdate()) as day\_of\_year,datename(week,getdate()),datename(weekday,getdate())as weekday

**dateadd()**

**datediff()**

**-🡪’**

**Ex:** select datediff(day,'2021-09-20','2021-06-06')as day,datediff(month,'2021-09-20','2021-06-06')as month,datediff(year,'1997-09-20','1998-06-06')as year

select datediff(month,'2021-09-20',getdate())

select datediff(year,'2029-01-01',getdate())

**convert()**

**🡪ex.**

select name, convert(date,getdate()) as 'Current date' from apprentish

select name, convert(date,'2000-09-09 20:0:2') as 'Current date' from apprentish

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **sn** | **name** | **Cont no** | **email** | **age** |
|  |  |  |  |  |
|  |  |  |  |  |

**Select \*,upper(name) as NAME from tbl\_name**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **sn** | **name** | **Cont no** | **email** | **age** | **NAME** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**UPDATE**

**Step1= create new database named “COLLEGE”.**

**Step2= create new table named “student”.**

**Step3=**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **rollno** | **name** | **mobile** | **depid** | **year** | **regdate** | **fee** |
| **Int pk auto-increment** | **Not null**  **varchar** | **varchar** | **int** | **varchar** | **datetime** | **int** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Step4=**

**Create a new table name “department”**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **depid** | **Dep\_name** | **seats** | **hod** |  |  |  |
| **Int pk AI** | **Varchar not null** | **int** | **Varchar null** |  |  |  |
|  |  |  |  |  |  |  |

**UPDATE**

**Update is a DML command it is used to modify pre existing records of table.**

**It is applied on column.**

**If you want to make change any value on table then UPDATE command may used**

**SYNTAX:**

**Update table\_name set column\_name=value where<condition>**

**Ex.-🡪**

update apprentish set dab='1998-06-09' where email='vi383#gmail.com'

select \* from apprentish

**DELETE COMMAND:**

Delete is a DML command use to remove a whole row.

To delete any particular row you can apply any condition with delete command;

Ex🡪

**Syntax🡪**

**Delete from table\_name; //**delete all row of table but not permanently, these deleted rows can be resotred.

delete from apprentish ;

**Delete from table\_name where<condition> //** to delete any specific row from table

delete from apprentish where name='prashant'

convert()

2021-11-08 :

Date function query->

create table studentid

(

rollno int primary key identity(1,2),

name varchar(50)not null,

dob date,

email varchar(100)not null,

regdate datetime

);

insert into studentid values('techpile','10/10/2002','www@gmail.com','09/11/2021 02:51:50'),

('app','10/11/2002','www@gmail.com','09/11/2021 02:51:50');

select name,regdate as 'currentdate'from studentid

select name,convert(date,regdate) as 'currentdate'from studentid

insert into studentid values('ram',convert(date,getdate()),'wwww@gm.com',getdate())

select datename(day,getdate())as day,datename(month,getdate())as month,datename(year,getdate())as year,

datename(dayofyear,getdate()) as day,datename(WEEKDAY,getdate()) as dayname,datename(week,getdate())as week

select \*from studentid //for show table

update commond-:

update is a dml command .it is used to modify pre-existing record of table .

it is applied on columns , if you want to make change value of table then update command is used

**syntax-:**

**update table\_name set column\_name=value where<condition>**

Ex-:

update studentid set email='op@gail'where rollno=2

ex2 -:

update studentid set dob=convert(date,getdate())

delete commond-:

delete command is a dml commond used to remove a whole row from the table.

To delete any particular row you can apply conditions with delete command.

**Syntax-:**

**Delete from table\_name;**

//delete all row of table but not permanently. Those deleted rows can be restored.

Delete from table\_name where<condition>;

//to delete any specific row from table.

Ex-:

delete from studentid where rollno=3

Alter command-:

Alter is a ddl command used to modify schema table. Many type of modification/changes you can perform by alter command

1. add a new column in existing table:

{column\_name data\_type(size) key(primary key,unique key)constraints(check,default,null)identity}

**Syntax-:**

**Alter table table\_name add column\_name datatype size constraint;**

Alter table table\_name add location varchar(20)default=’lucknow’

Ex-:

alter table department add location varchar(20)

2-:

**Drop any existing column from table:**

**Syntax-:**

**Alter table table\_name drop column column\_name;**

Ex-:

alter table department drop column start\_date //for column delete

alter table teachar drop constraint CK\_teacharsalary\_25869641//for constraints delete

both drop(delete) column and constraints-:

alter table teachar drop constraint PK\_teachar\_3F330016FBF16BDE,column empid

imp-all table drop-:

syntax-:

drop table table\_name;

for database delete/drop-:

syntax-:

drop database database\_name;

# Note- all dml command used table name as like

Create table table\_name

alter table table\_name

drop table table\_name

truncate table table\_name

to change data type and size existing columns:

syntax-:

alter table table\_name alter column column\_name data\_type (size);

note-:

by using alter , you can only increase the size of column can’t decrease size

ex-:

alter table department alter column headofdept char(100)

alter table department alter column headofdept varchar(15)

drop command-:

drop is a ddl command used to permanently remove table with schema and record.

That mean drop removes the whole table. Drop removes the existence of table.

Syntax-:

Drop table table\_name;

Truncate command-:

Truncate command is used to remove all the saved record of table permanently.

Record deleted with truncate command can’t be restored.

Truncate saved the schema of table for future use but remove all record

Syntax-:

Truncate table table\_name;

Top clause:-

Top clause restrict the total returning row of select command. Top clause specific the total number of rows you want to select with select command.

select top 3 \*from student

select top 4 name,fee from student

select top 1 \*from student where name like'r%'// from search first latter of name

sub-queries-:

sub-queries are queries written within any other query.

Sub-queries are always written ()

Sub-Queries can be used with select, update, delete, any command

Sub-queries should always return only one column

If you want to use result of another queries within a query then sub –queries are used

Ex -:

select \*from student where fee=(select max(fee)from student)

particular column update -:

update student set fee =(select max(fee)+5000 from student)where rollno=5

distinct(unique)-:

distinct used with select command .

used to select set of unique value only.

select distinct fee from student

select distinct \* from student

select fee from student orderby

**JOINING:**

JOINING IS USE to select two or more table where all table has some common fields.

Types of joining.

1. Inner join
2. Left outer join
3. Right outer join
4. Full outer join
5. Cross join
6. Self join

**INNER JOIN**🡪

Inner join selects only common records of all tables.

**Syntax:**

Select tbl.colum\_name from first\_table name inner join second\_tablename on first\_table\_name.common\_column\_name=second\_table\_name.column\_name;

**Ex;**

select \* from dept

select \* from emp

select emp.emp\_name,emp.salary,dept.dep\_name,dept.hod from emp inner join dept on emp.dept=dept.dep\_name

Left join:🡪

Left join is used to select left table on table when only matched record on second table, the value which is not present in second table then by default NULL value is put inthat place.

**Syntax🡪**

Select tbl.colum\_name from first\_table name left join second\_tablename on first\_table\_name.common\_column\_name=second\_table\_name.column\_name;

Ex:

select emp.emp\_name,emp.salary,dept.dep\_name,dept.hod from emp left join dept on emp.dept=dept.dep\_name

Right join 🡪

right join is used to select right table on table when only matched record on second table, the value which is not present in second table then by default NULL value is put in that place.

**Syntax:**

Select tbl.colum\_name from first\_table name right join second\_tablename on first\_table\_name.common\_column\_name=second\_table\_name.column\_name;

Ex:

**Select employee.emp\_name, department.dept\_name from department left join employee on department.dept\_id=employee.emp\_deptid;**

Full outer join🡪

It returns all records from both table.

**Syntax:**

Select tbl.colum\_name from first\_table name full outer join second\_tablename on first\_table\_name.common\_column\_name=second\_table\_name.column\_name;

**Table aliasing**

Select s.name, d.dep\_name from student s inner join department d on s.dep\_id=d.depid

**Group By: G**roup by clause is used to make group of one or more records that has same value based on a column.

**\*\*\*\*\*\*\*\*When you are using group by clause with select statement you can only select the column which are using in group by or you can use any aggregate function with select.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**++++++++++++++++++++++++++++++**

Syntax: **select column\_name,aggregate\_function(column\_name) from table\_name group by column\_name**

**Ex:**

Select depid,count(\*),sum(fee) from student group by dep\_id.

**Having Clause:🡪** having clause is used when we want to apply condtion on group by.

**Select depid,count(\*),sum(fee) from student group by depid having sum(fee)>=5000**

**------ ---------------------------T-SQL------------------------ -------------**

T-sql stand for **transact sql**.

It provides some advance tools to perform different operation in database.

STANDARD SQL – DDL,DML Queries.

* t-sql🡪 **Triggers**
* **Batch or scripts**
* **User defined variables**
* **Stored procedure**
* **User defined function**

Different tools are—

1.User define variables

2.Control flow statement

a.begin end

b. if-else

c. while

d. try-catch

**USER defined variables:🡪**

User defined variable are just a name to store data from outside world.

In TSQL mainly variable are declared with prefix **DECLARE.** And always variable are decleared with following SYNTAX.

**DACLARE @\_variable\_name data\_type(size),---------- - -**

**By** default used defined variable has null value

**Assign tha value to the variable:🡪 There are 2 ways** to assign values

1.By using SET keword

2.**By using select Keyword--🡪** select keyword is use to set values to the variables. By using select you can set multiple values to different variable at one time.

Syntax🡪

**Select @variable\_name=value,**

**@variable\_name=values------------**

**By using SET keword** 🡪a Set keyword is used to assign one word value

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*How to create SQL query with case.....?\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Case is use to select any column based on the given condition . normally case is used to apply condition within a query.

**SYNTAX:🡪**

**Case when < condiotion> then ‘statement’**

**Example:**

select emp\_name,

case when salary between 12000 and 30000 then 'developer' end from emp

select emp\_name,

case when salary between 31000 and 40000 then salary+1000 end from emp

**if-else:**

**case when <condition> then statement>**

**else**

**end**

**example”🡪**

select emp\_name,salary,

case when salary between 10000 and 30000 then 'developer' else

'senior developer' end from emp

**Ladder else if**

**Syntax -🡪**

**Case when <condition> then statement**

**When <condition> then statement**

**When <condition> then statement**

**.**

**.**

**Else statement**

**BEGIN-END:** begin-end is a keyword of sql also known as control flow statements.

**Begin –end** contains batch(collection of scripts) that is used to defined range of if or else or any conditional statement.

* BEGIN-END can be nested.
* BEGIN\_END defines blocks of conditional or flow control statement like if,else,while etc

If(**condition)**

**Begin**

**……////statement**

**…..**

**…**

**End**

**Else if (condition)**

**Begin**

**….**

**.//statements**

**.**

**end**

**Example**

DECLARE @salary int,

@action int

set @action=1

set @salary=30000

if(@action=1)

begin

select \* from emp

end

else if(@action=2)

begin

select \* from dept

end

**try-catch🡪** try catch is used to handle a error occur due to enter some wrong entry or anything else.

Runtime error of query is handled by sql.

Catch comes always after try blocks

Catch block execute only if there is any error in try block.

If there is any possibility of error in any line of code then those line are put into try block with a error free message of any code within catch block.

**lSyntax🡪**

**Begin try**

**//statement**

**End try**

**Begin catch**

**//statement**

**End catch**

**Example🡪**

declare

@salary int,

@action int

set @salary=40000

set @action=1

if(@action=1)

begin

begin try

update emp set salary=6 where salary=@salary

end try

begin catch

print 'error occured'

end catch

begin

select \* from emp

end

end

**STORED PROCEDURES**

**🡪Stored procedure is collection of some pre-written queries/database commands.**

**Benefits of stored procedure is provide**

* **code reusability**
* **Security to the database operation**

**Syntax:**

**Create procedure procedure\_name**

**As**

**Begin**

**……**

**………//sql statements**

**.**

**.**

**End**

You can create stored procedure in two ways.---

**1.stored procedure with parameters**

**2.stored procedure without parameters**

**TCL(TRANSACTION CONTROL LANGUAGE)**

🡪IT is used to control operations of databse.

1. Commit
2. Rollback
3. Save point

Are some command of transaction tcl

1. **Commit🡪**

Mssql🡪 user interface bases application(it is an auto commit database)🡪ex insert,update,delete

Commit is a single command used to save transaction of database permanently.

**ACID**

* **ATOMICITY**🡪 ALL OR NONE. ALL transactions should be successful or not.
* **CONSISTENCY🡪 after transaction and before transaction databse should be same.**
* **ISOLATION🡪 t**ransaction of databse should not effect to the another transaction of db.
* **DURABILITY🡪 each** database should have a backup system in case of any accident occurs with database.

**ROLLBACK🡪** Rollback sets database to the initial position.

**You can rollback those transaction which are in buffer memory.**

**Those transaction which are written with ‘begin transaction’ and did committed those can be rollbacked.**

**Syntax:🡪**

**Begin transaction**

**rollback**

example🡪

select \* from emp

delete from emp where emp\_id=105

begin transaction

delete from emp where emp\_id=102

begin transaction

rollback

**implicit control🡪automatically saved**

**explicit control🡪 control of saving data is not automatically done.**

**Note🡪 to remove auto-commit system of db transaction are written within begin transaction**

**Begin transaction**

**//sql statements**

**Commits;**

**SAVE TRANSACTION🡪**

**Save transaction is used to set pointer to the particular location of data saved in buffer memory. And create separate memory for transaction executed after pointer.**

**It is used so that you can rollback to the specific number of transaction.**

**Syntax🡪**

**Begin transaction**

**Save transaction <pointer\_name>**

**//sql statements**