***/\*data definition language.............................................................\*/***

create table record1(slno int,firstname varchar(50),age int)

alter table record1 add address varchar(100)

alter table record1 drop column age

drop table record1

***/\*data defenition language.............................................................\*/***

***/\*data manipulation language............................................................\*/***

insert into record1 values (1,'amal',23)

insert into record1 values (2,'abhijith',24)

insert into record1 values (3,'abhilash',26)

insert into record1 values (4,'bimal',21)

insert into record1 (slno,firstname) values (5,'anoop')

update record1 set age=34 where slno=2

delete from record1 where slno=2

delete from record1

***/\*data manipulation language................................................................\*/***

***/\*data query language................................................................\*/***

select \*from record1

select firstname ,age from record1

***/\*data query language.................................................................\*/***

***/\*comparison and logical operators..................................................\*/***

select \*from record1 where age=23

select \*from record1 where age>23

select \*from record1 where age>=23

select \*from record1 where age<23

select \*from record1 where age<=23

select \*from record1 where age<>23

select \*from record1 where age between 22 and 24

select \*from record1 where age in (21,23,26)

select \*from record1 where firstname like 'a%'

select \*from record1 where firstname like '%h'

select \*from record1 where firstname like '%a%'

select \*from record1 where firstname like '\_\_h%'

select \*from record1 where age is null

select \*from record1 where firstname like 'a%' and age>24

select \*from record1 where firstname like 'a%' or age>24

select \*from record1 where firstname not like 'a%'

***/\*comparison and logical operators..................................................\*/***

***/\*constraints while creating table...***

***.............................................\*/***

create table record2(slno int,firstname varchar(50),age int not null)

insert into record2 (slno,firstname) values (1,'anoop')

create table record3(slno int,firstname varchar(50),age int ,unique(slno))

insert into record3 values (2,'adhi',23)

create table record4(slno int,firstname varchar(50),age int,primary key (slno))

insert into record4 values (1,'amal',23)

insert into record4 values (2,'abhijith',24)

insert into record4 values (3,'abhilash',26)

insert into record4 values (4,'bimal',21)

select \*from record4

create table record5 (rollno int,date int,foreign key (rollno) references record4(slno))

insert into record5 values (1,24)

insert into record5 values (6,24)

select \*from record5

create table record6(slno int,firstname varchar(50),age int,check(age>21))

insert into record6 values (2,'amal',4)

select \*from record6

create table record7(slno int,firstname varchar(50),age int default 21)

insert into record7 (slno,firstname) values (7,'amal')

select \*from record7

insert into record7 values (4,'abhilash',26)

***/\*constraints while creating table................................................\*/***

***/\*joins...........................................................................\*/***

select \*from record1

select \*from record8

create table record8 (slno int,date int)

insert into record8 values (1,24)

insert into record8 values (2,25)

insert into record8 values (5,24)

insert into record8 values (9,23)

insert into record8 values (13,25)

SELECT record1.firstname,record8.date FROM record1 INNER JOIN record8 ON record1.slno=record8.slno ORDER BY record1.firstname

SELECT record1.firstname,record8.date FROM record1 left JOIN record8 ON record1.slno=record8.slno ORDER BY record1.firstname

SELECT record1.firstname,record8.date FROM record1 right JOIN record8 ON record1.slno=record8.slno ORDER BY record1.firstname

SELECT record1.firstname,record8.date FROM record1 full JOIN record8 ON record1.slno=record8.slno ORDER BY record1.firstname

***/\*stored procedures.......................................................................\*/***

The benefits of using stored procedures in SQL Server rather than application code stored locally on client computers include: They allow modular programming. They allow faster execution. They can reduce network traffic.

**Need of stored procedure**

***Stored procedures provide improved performance because fewer calls need to be sent to the database. For example, if a stored procedure has four SQL statements in the code, then there only needs to be a single call to the database instead of four calls for each individual SQL statement.***

create table record9a (slno int,firstname varchar(50),city varchar(25),state varchar(50))

insert into record9a values (1,'amal','goa','goa')

insert into record9a values (3,'sreeraj','cochin','kerala')

insert into record9a values (4,'rahul','cochin','kerala')

select \*from record9a

CREATE PROCEDURE priya

@city varchar(25)

AS

BEGIN

SELECT \*FROM record9 WHERE city =@city

END

GO

// drop procedure priya

EXEC priya @city='goa'

CREATE PROCEDURE priya

@city varchar(25)

AS

BEGIN

SELECT \*FROM record9 WHERE city = @city

RETURN @@ROWCOUNT;

END

GO

DECLARE @i INT

EXEC @i=priya @city='cochin'

SELECT @i as countofcity

***/\*stored procedures.......................................................................\*/***

***/\*functions...............................................................................\*/***

**select avg(age) from record1**

**select avg(age) as average from record1**

**select sum(age) as sum from record1**

**select max(age) as maximum from record1**

**select min(age) as minimum from record1**

**select count(age) as countofage from record1**

**select upper(firstname) as uppercase from record1**

**select lower(firstname) as lowercase from record1**

**select getdate()**

**select len(firstname) as length from record1**

**create table record10(name varchar(50),amount float)**

**insert into record10 values('sooraj',43.655645)**

**insert into record10 values('vishak',435.098765)**

**insert into record10 values('dhanush',23.00000676876)**

**select \*from record10**

**select round(amount,3)as placevalues from record10**

***/\*functions...............................................................................\*/***

# SQL | GROUP BY

The GROUP BY Statement in SQL is used to arrange identical data into groups with the help of some functions.

i.e if a particular column has same values in different rows then it will arrange these rows in a group.

Important Points:

* GROUP BY clause is used with the SELECT statement.
* In the query, GROUP BY clause is placed after the WHERE clause.
* In the query, GROUP BY clause is placed before ORDER BY clause if used any.

HAVING Clause

We can use HAVING clause to place conditions to decide which group will be the part of final result-set.

we can not use the aggregate functions like SUM(), COUNT() etc. with WHERE clause. So we have to use HAVING clause if we want to use any of these functions in the conditions.