**SQL QUERY**

use AvneeshDB

select \* from employee Details

select \* from employee Details where salary between 20000 and 40000;

select Top 5 \* from employee\_Details

select distinct Name from employee\_Details;

alter table employee\_Details add City varchar (50);

select \* from employee\_Details

select \* from employee\_Details

alter table employee\_Details

drop column City

select \* from employee\_Details

delete from employee\_Details where emp\_id=111

Update Query

update employee\_Details set Email='avneesh832@gmail.com',

PhoneNumber='7497976557’, City='Ayodhya' where emp\_id=109

select \* from employee\_Details where City in('Delhi','Ayodhya')

select \* from employee\_Details where City not in('Delhi','Ayodhya')

SQL Constraints

* Sql constraints are used to specify rule for the data in a Table.

The following constraints are commonly used in SQL:

* NOT NULL: -ensure a column cannot have a null value.
* UNIQUE: - Ensure that all values in a column are different.
* Primary Key: - A combination of NOT NULL and Unique. Uniquely Identify each row in a Table.
* Foreign Key: - Uniquely identifies a row/record in another Table.
* CHECK: - Ensures that all values in a column satisfies a specific condition
* DEFAULT: - Sets a default values for a column when no value is specified.

HOW TO UPDATE DATABASES

alter database AvneeshDB modify name=new\_AvneeshDB

Update Database using Stored procedure

execute sp\_renamedb ‘new\_AvneeshDB’,’AvneeshDB’;

Update Table Name using Stored Procedure

Execute sp\_rename ‘Voterlist’,’Voterlist2’

Add New Column

alter table Voterlist add Voter\_City varchar(50);

Update Column values

update Voterlist set Voter\_City='Delhi' where Voter\_id=8;

delete column

alter table Voterlist drop column Voter\_City;

change Datatype

alter table Voterlist alter column Voter\_Name nvarchar(50);

IDENTITY AUTO INCREMENT

Auto increment allows a unique number to be generated automatically when a new record is inserted into a table.

create table Teacher(T\_id int primary key identity(100,5),T\_Name varchar(50) not null, Qualification varchar(50) not null,T\_Salary varchar(50) not null);

select \* from Teacher;

insert into Teacher values ('Avneesh’ , B.Tech','30000');

insert into Teacher values('Avneesh','B. Tech','30000');

insert into Teacher values('Anuj','B.sc','12000');

insert into Teacher values('Ankit','B.T.C','10000');

insert into Teacher values('Arpit','B.C.A','10000');

insert into Teacher values('Baby','B.A','8000');

select \* from Teacher;

what is ALIAS in SQL?

SQL Aliases are used to give a table, or a column in a table , a temporary name.

select T\_id as Teacher\_id,T\_Name as Teacher\_Name,T\_Salary as Teacher\_Salary from Teacher;

**Forigen Key**

select \* from customer

insert into customer values(101,'Avneesh','UP','Gonda');

insert into customer values(102,'Ankit','Delhi','Uttam Nagar');

insert into customer values(103,'Baby','UP','Basti');

insert into customer values(104,'Ravi','Haryana','Gurugram');

select \* from customer

create table order\_tbl(Ord\_Id int primary key,

item varchar(50),

Qty int,

Price\_of\_1 int,

c\_id int foreign key references customer(c\_id));

**Stored Procedure**

A stored procedure is a set of Structure Query Language Statement with an assign name, which are stored in a relational database management system as a group ,sot it can be reused and shared by multiple programs.

Type of stored procedure

1. System Stored Procedure
2. User Defined Stored Procedure

Work To Do

1. Stored Procedure Example
2. Stored Procedure with Single Parameter
3. Store Procedure with multiple Parameters, changing the parameter order
4. Alter with Stored Procedure
5. Seeing the text of the Sp. Sp\_helptext
6. Drop with stored procedure
7. Using with Encryption in stored procedures.

use AvneeshDB;

create table Employee(Emp\_Id int,Emp\_Name Varchar(50),Gender Varchar(30),Salary Varchar(30), City Varchar(80));

insert into Employee values(2,'Avneesh Kumar','Male',20000,'Atodhya');

select \* from Employee;

insert into Employee values(3,'Ankit Kumar','Male',40000,'Gonda');

insert into Employee values(4,'Baby','Female',32000,'Basti');

insert into Employee values(5,'Abhinav Kuame','Male',42000,'Lucknow');

select \* from Employee;

create procedure spGetEmployee

as

begin

select Emp\_Name,Gender from Employee;

end

spGetEmployee;

execute spGetEmployee

create procedure spGetEmployeeById

@id int

as

begin

select \* from Employee where Emp\_Id=@id

end

execute spGetEmployeeById 4;

create procedure spGetEmployeeByIdAndByName

@id int,

@Name varchar(50)

as

begin

select \* from Employee where Emp\_Id=@id and Emp\_Name=@Name;

end;

execute spGetEmployeeByIdAndByName 2,'Avneesh Kumar';

alter procedure spGetEmployeeByIdAndByName

@id int,

@Name varchar(50)

with encryption

as

begin

select Emp\_Name,Salary from Employee where

Emp\_Id=@id

and

Emp\_Name=@Name;

end;

execute spGetEmployeeByIdAndByName 2,'Avneesh Kumar';

sp\_helptext spGetEmployeeByIdAndByName;

drop procedure spGetEmployeeByIdAndByName;

Store procedure with Output parameter

create proc spGetEmployeeByGender

@Gender varchar(50),

@EmployeeCout int Output

as

begin

select @EmployeeCout=Count(Emp\_Id) from Employee

where Gender=@Gender

end

Declare @TotalEmployee int

execute spGetEmployeeByGender 'Female',@TotalEmployee output

select @TotalEmployee

select \* from Employee

**Trigger**

**A trigger is a special type of stored procedure that automatically executed when an event occurs the database server.**

**There are three type of trigger.**

1. **DML Triggers(Data Manipulation Language)Insert ,Update,Delete**
2. **DDL Triggers (Data Definition Language) Create,Alter**
3. **LOGON Triggers**

**DML Triggers are fired automatically in response to DML events (Insert,Update,Delete)**

**DML Triggers can be Two Types**

1. **After Triggers (Also Called For Triggers)**
2. **Instead Of Triggers**

-- triggers--

create table StudentTable(Std\_Id int primary key identity,Std\_Name varchar(50),

Gender varchar(30),Std\_class int,Fees Varchar(30))

select \* from StudentTable

insert into StudentTable values('Ajay','Male',7,3000);

insert into StudentTable values('Baby','Female',10,5000);

insert into StudentTable values('Anjani','Female',4,7000);

insert into StudentTable values('Abhi','Male',1,2000);

insert into StudentTable values('Sony','Female',10,4500);

insert into StudentTable values('Sony','Female',4,3000);

insert into StudentTable values('Sunil','Male',11,8000);

insert into StudentTable values('Sanjay','Male',12,12000);

select \* from StudentTable

create trigger tr\_Student\_forinsert on StudentTable

after insert

as

begin

print'Someting Happen';

end