DATA BASE MANAGEMENT SYSTEM

Data Base Objects

Table or entity

View

Store Procedure

Functions & Triggers

What is table?

A real word entity is called table.

Ex: - Student, Staff.

RDBMS

Table = Entity

Column = Attribute

Row = Touple

Data Type

1. Bit (0 or 1)
2. Tinyint (0 to 255) 1 byte of memory
3. Smallint 2bytes of memory
4. Int 4 byte of memory
5. BigInt 8 bytes of memory

For Decimal Values

1. Float 17 precisions 4 to 13 bytes
2. Decimal
3. Numeric (5,3) out of 5 we have 3 decimal digits ( 12345.123)
4. Marks (5,2)
5. Money for salary come with the prefix currency symbol. $1233453435.342

For Character Values

11.Char ‘A’, ‘B’ 1 byte

12. char(20) it will accept up to 20 characters

For Alpha Numeric Character

13. varchar any single character

14. varchar(100) up to 100 characters

15. text it will be more 8000 character in 8KB of each files (clob)

16. varchar(max) it also up to 8000 characters

We can store data in our local language

17. nchar Each character will occupy 2 bytes

18. ntext

19.nvarchar(100)

Data Datatypes

20. Date mm/dd/yyyy

21. DateTime mm/dd/year hh:mm:ss:ms

22. Time hh:mm:ss:ms

23. smalldateTime

24. Timestamp

25. BInaryData(Image)

Commends classified int 4 types

DDL (Data definition Language)

Create, Alter, Drop

DML (Data Manipulation Language)

Insert, update, Delete

DCL (Data Control Language)

Grant, Denay, revoke

DQL (Data Query Language)

Select

How to Create DataBase?

Create DataBase JAVATRAING;

Use JavaTraining;

How to drop DataBase?

Drop DataBase JavaTraining;

create database JavaTraining

use JavaTraining;

drop database JavaTraining;

Create Table Employee(

Id Int,

Name Varchar(30),

Department Varchar(30),

DOD date,

DOJ date,

salary money);

insert into employee values(1,'HART','Computer Science','01/01/1996','04/04/2021',50000);

insert into employee values(2,'Pallavi','BA','01/01/1993','04/04/2023',6000),

(1,'Sathy','Engg','01/01/1990','04/04/2019',80000);

insert into employee values(4,'ART','Information Science',null,'04/04/2021',null);

insert into Employee(id,Name,salary) values(5,'Nazia',10000.12);

select \* from Employee;

DATA BASE MANAGEMENT SYSTEM.docx

Constraints in SQL

1. Not null
2. Unique
3. Check (gender M, F)
4. Default
5. Primary key, composite key primary key (ID, USN);
6. Identity (1,1)

Student Table

|  |  |
| --- | --- |
| ID | USN |
| 1 | 1 |
| 2 | 2 |

1. Foreign key Table name is marks

|  |  |
| --- | --- |
| ID | Marks |
| 1 |  |
| 2 |  |
|  |  |

insert into Employee(id,Name,salary) values(5,'Nazia',10000.12);

select \* from Employee;

-----

Alter table employee add Gender char;

update employee set gender = 'B' where id=1;

update employee set Id= 3 where Name = 'Sathy';

update employee set gender = 'F' where id in(2,5);

update employee set dod = '12/12/2000', salary = 100000, gender = 'F' where id= 4;

select \* from Employee;

delete from Employee where id = 5;

Alter table employee drop column Gender;

drop table Employee;

select \* from employee;

select Name,salary from employee;

select \* from employee where salary>25000;

select \* from employee where salary > 25000 and gender = 'F';

select \* from employee where salary = 80000 ;

select \* from employee where salary != 80000 ;

select \* from employee where salary >= 6000 ;

select \* from employee where salary <= 6000 ;

select \* from employee where Id not in(2,5);

select \* from employee where Id not between 2 and 5;

select \* from employee where doj is null;

select \* from employee where doj is not null and gender is not null;

select \* from employee where doj = null; -- not possible

create table student (id int, name varchar(40) constraint nname not null, mobile bigint);

insert into student(id,name,mobile)values(1,'HART',1234567890);

insert into student(id,name,mobile)values(2,'HART',1234567890);

select \* from student;

drop table student;

create table student (id int, name varchar(40) constraint nname not null, mobile bigint constraint uqmobile unique);

insert into student(id,name,mobile)values(1,'HART',1234567890);

insert into student(id,name,mobile)values(2,'HART',1234567891);

insert into student(id,name,mobile)values(4,'HART',null);

create table student (id int, name varchar(40) constraint nname not null, mobile bigint constraint uqmobile unique,

gender char constraint chgen check(gender in('M','F')));

insert into student(id,name,mobile,gender)values(1,'HART',1234567890,'F');

alter table student drop constraint chgen;

insert into student(id,name,mobile,gender)values(1,'HART',12345674890,'D');

create table student (id int, name varchar(40) constraint nname not null, mobile bigint constraint uqmobile unique,

status int constraint defstatus default(1));

insert into student(id,name,mobile,status)values(1,'HART',1234567895,5);

create table student (id int, name varchar(40) constraint nname not null, mobile bigint constraint uqmobile unique,

status int constraint defstatus default(1), constraint idpk primary key (id));

alter table student drop constraint idpk;

alter table student add constraint idpk primary key (id);

update student set id =1 where mobile = 12345678951;

Create Table Employee(

Id Int identity(1000,1),

Name Varchar(30),

email varchar(30) constraint epk primary key,

Department Varchar(30),

DOD date,

DOJ date,

salary money);

insert into employee values('Harshith','sharshith113@gmail.com','IT','12/12/2020','12/12/2020',234324234);

INSERT INTO Employee (Name, email, Department, DOD, DOJ, salary) VALUES ('Lily Ward', 'lily.ward@example.com', 'Human Resources', NULL, '2022-04-21', 47000.00);

-- Inserting employee 26

INSERT INTO Employee (Name, email, Department, DOD, DOJ, salary) VALUES ('Matthew Turner', 'matthew.turner@example.com', 'Sales', NULL, '2023-01-07', 51000.00);

select \* from employee;

drop table Employee;

create table department (DepID int constraint depID primary key, DepartmentName varchar(30));

-- Inserting department 1

INSERT INTO department (DepID, DepartmentName) VALUES (1, 'Engineering');

-- Inserting department 2

INSERT INTO department (DepID, DepartmentName) VALUES (2, 'Human Resources');

-- Inserting department 3

INSERT INTO department (DepID, DepartmentName) VALUES (3, 'Marketing');

-- Inserting department 4

INSERT INTO department (DepID, DepartmentName) VALUES (4, 'Finance');

-- Inserting department 5

INSERT INTO department (DepID, DepartmentName) VALUES (5, 'Sales');

-- Inserting department 6

INSERT INTO department (DepID, DepartmentName) VALUES (6, 'Customer Service');

-- Inserting department 7

INSERT INTO department (DepID, DepartmentName) VALUES (7, 'Research and Development');

-- Inserting department 8

INSERT INTO department (DepID, DepartmentName) VALUES (8, 'Information Technology');

-- Inserting department 9

INSERT INTO department (DepID, DepartmentName) VALUES (9, 'Operations');

-- Inserting department 10

INSERT INTO department (DepID, DepartmentName) VALUES (10, 'Administration');

select \* from department;

Create Table Employee(

Id Int identity(1000,1),

Name Varchar(30),

email varchar(30) constraint epk primary key,

DepartmentId int constraint DepidFk foreign key references department(DepID),

DOD date,

DOJ date,

salary money);

-- Inserting employees into DepartmentId 1

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('John Doe', 'john.doe@example.com', 1, '2024-03-22', '2020-01-15', 50000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('David Wilson', 'david.wilson@example.com', 1, '2023-11-30', '2020-03-08', 52000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('James Anderson', 'james.anderson@example.com', 1, NULL, '2022-01-22', 51000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Ava Harris', 'ava.harris@example.com', 1, NULL, '2020-05-19', 48000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Daniel Scott', 'daniel.scott@example.com', 1, NULL, '2023-09-25', 52000.00);

-- Inserting employees into DepartmentId 2

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Jane Smith', 'jane.smith@example.com', 2, NULL, '2021-05-10', 45000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Sophia Jackson', 'sophia.jackson@example.com', 2, '2023-04-05', '2021-03-14', 47000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Ethan Lewis', 'ethan.lewis@example.com', 2, NULL, '2021-07-09', 46000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Grace Baker', 'grace.baker@example.com', 2, '2023-03-18', '2021-02-20', 48000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Lily Ward', 'lily.ward@example.com', 2, NULL, '2022-04-21', 47000.00);

-- Inserting employees into DepartmentId 3

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Michael Johnson', 'michael.johnson@example.com', 3, NULL, '2023-02-28', 55000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Mia Clark', 'mia.clark@example.com', 3, '2022-10-15', '2020-02-10', 54000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('William Thomas', 'william.thomas@example.com', 3, NULL, '2020-11-11', 53000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Ryan Adams', 'ryan.adams@example.com', 3, NULL, '2020-06-30', 55000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Avery Morris', 'avery.morris@example.com', 3, NULL, '2020-08-03', 56000.00);

-- Inserting employees into DepartmentId 4

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Emily Davis', 'emily.davis@example.com', 4, NULL, '2022-07-19', 60000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Alexander Brown', 'alexander.brown@example.com', 4, NULL, '2023-01-03', 61000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Olivia Taylor', 'olivia.taylor@example.com', 4, NULL, '2023-06-17', 62000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Charlotte Green', 'charlotte.green@example.com', 4, NULL, '2021-12-28', 63000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Samuel Carter', 'samuel.carter@example.com', 4, NULL, '2021-01-29', 64000.00);

-- Inserting employees into DepartmentId 5

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Emma Martinez', 'emma.martinez@example.com', 5, NULL, '2021-09-05', 48000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Benjamin White', 'benjamin.white@example.com', 5, NULL, '2022-08-27', 49000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Isabella Hill', 'isabella.hill@example.com', 5, NULL, '2022-03-12', 50000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Jackson King', 'jackson.king@example.com', 5, NULL, '2022-09-14', 49000.00);

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Matthew Turner', 'matthew.turner@example.com', 5, NULL, '2023-01-07', 51000.00);

select \* from employee;

select \* from department;

INSERT INTO Employee (Name, email, DepartmentId, DOD, DOJ, salary) VALUES ('Matthew Turner', 'matthe.turner@example.com', 10,'2023-01-07', '2023-01-07', 53000.00);

delete from department where DepID = 10;

alter table employee drop column DepartmentId;

alter table employee add DepartmentId int constraint DepidFk foreign key references department(DepID) on delete cascade on update cascade;

truncate table employee;

update department set DepID = 11 where DepID =10;

delete from department where DepID = 11;

-

--Date function

select getdate();

select DATEPART(dd,getdate());

select DATEPART(dd, '02/29/1996');

select id,name,doj,DATEPART(dd,doj) from Employee;

select \* from employee where DATEPART(dd,doj)=28;

select \* from employee where DATEPART(dd,doj) between 1 and 15;

select \* from employee where DATEPART(dd,doj) < = 5;

-----------------------------------------------------

select DATEPART(mm,getdate());

select \* from employee where DATEPART(mm,dod) = DATEPART(mm,getdate());

select \* from employee where DATEPART(mm,dod) = 4;

select \* from employee where DATEPART(mm,doj) = 5;

select \* from employee where DATEPART(mm,doj)>6;

------------------------------------------------------

select id, name, doj,DATEPART(yyyy,doj) as [Joined] from employee;

select \* from employee where DATEPART(qq,doj)=1;

select datepart (dw,getdate());

select \* from employee where datepart (dw,dod)=4;

-----------------------

select datename(dw,'2023-04-05');

select dateadd(dd,15,getdate());

select dateadd(hh,10,getdate());

select dateadd(yyyy,18,getdate());

----------------------------------------------

select datediff(yyyy,'03/02/1990',getdate());

select datediff(mm,'03/02/1990',getdate());

select datediff(dd,'03/02/1990',getdate());

select datediff(hh,'03/02/1990',getdate());

---------------------------------------------

select doj,DATEDIFF(mm,doj,getdate()) as [experience] from employee;

select id , name ,doj ,datediff(mm,doj,getdate())/12 as [year],datediff(mm,doj,getdate())%12 as [Month] from Employee;

select doj,DATEDIFF(yyyy,doj,getdate()) as [experience] from employee;

---------------------------------------------------------------------

------Character Function

select id,name,len(name) as [Len] from employee;

select id,name,len(name) as [len] from Employee where len(name)<10;

select id,name,len(name) as [len] from Employee where len(name)>10;

select 'Harshith' + 'S'

select \* from employee;

select name +':'+ email from Employee;

select ltrim(' Harshith S ');

select ltrim('A Harshith S dsgdfgdgdfgdfgdfgdf f ');

select rtrim(' Harshith S dsgdfgdgdfgdfgdfgdf f ');

select replace('String function','string','time');

select replicate('T',8);

select replicate('T',-1);

select right('HART',3);

select left('HART',3);

select \* from employee where right(name,1) !='A';

select \* from employee where left(name,1) !='A';

select \* from employee where name like 'B%';

select \* from employee where name like '%n';

select \* from employee where name like '%B%';

select max(salary) from employee;

select \* from department;

select max(salary) from employee where DepartmentId=2;

select max(doj) from employee;

select max(name) from employee;

select min(salary) from employee;

select \* from department;

select min(salary) from employee where DepartmentId=2;

select min(doj) from employee;

select min(name) from employee;

select sum (salary) from employee;

select sum(salary) from Employee where DepartmentId = 2;

select avg(salary) from Employee;

select avg(salary) from Employee where DepartmentId = 1;

select max(salary) from Employee where salary != (select max(salary) from employee);

select \* from employee where salary = (select max(salary) from employee);

select max(salary) from employee where salary not in ((select max(salary) from employee),

(select max(salary) from Employee where salary != (select max(salary) from employee)))

select max(TotalSal) as [salary] from(

select DepartmentId,sum(salary) as[TotalSal] from Employee group by DepartmentId) x;

select DepartmentId,count(\*) from Employee group by DepartmentId;

select name from(

select name, id from Employee) x;

select DepartmentId,count(salary) from Employee group by DepartmentId;

select name, DepartmentId from Employee group by DepartmentId;

select DepartmentId,count(\*) from Employee where salary > 55000 group by DepartmentId;

select DepartmentId,count(\*) from Employee group by DepartmentId having count(\*) <5;

select DepartmentId,count(\*) from Employee group by DepartmentId having count(\*) <5;

select DepartmentId,count(\*) from Employee where salary > 40000 group by DepartmentId having count(\*) <5;

select DepartmentId, count(\*) as [NOE] from employee group by DepartmentId order by [NOE]

select DepartmentId, count(\*) as [NOE] from employee group by DepartmentId order by [NOE] desc

select name from employee;

select name from employee order by name desc;

select DATEPART(yyyy,doj) as[Year Of Joining], count(\*) as [NOE] from employee group by datepart(yyyy,doj);

select DATEPART(yyyy,doj) as[Year Of Joining], count(\*) as [NOE] from employee where datepart(yyyy,doj) between 2021 and 2023

group by datepart(yyyy,doj);

select DepartmentId,totalsalary from(

select DepartmentId, sum(salary) as [totalsalary] from employee where sum(salary) != (

select DepartmentId,sum(salary) as [totalsalary] from Employee group by DepartmentId))x;

;

with x as(select DepartmentId,sum(salary) as [totalsalary] from Employee group by DepartmentId)

select \* from x where totalsalary = (select max(totalsalary) from x);

--cross join

--inner join

--outer join

--left outer

--right outer

--full outer

table A atable b

1 A

2 B

3 C

4 D

m \* n

select \* from Customers;

select \* from Accounts;

select c.Accno,c.Name,c.Address,A.Account, c.status from Customers c cross join Accounts A;

select c.name,a.Account,c.status from Customers c inner join Accounts a on c.Account = a.Id;

select c.Account,a.id,a.Account

from

Customers c inner join Accounts a

on c.Account = a.Id;

select c.Account,a.id ,from Customers c left outer join Accounts a on c.Account = a.id;

select c.Account,a.id from Customers c right outer join Accounts a on c.Account = a.id;

select c.Account,a.id, a.Account, c.name from Customers c full outer join Accounts a on c.Account = a.id;

select \* from department;

select \* from employee;

select e.Id,e.name,d.DepartmentName from Employee e inner join department d on e.DepartmentId = d.DepID;

select \* from AccountTransactions;

select \* from Accounts;

select \* from Customers;

select a.Account,c.address,count(c.accno)as [no of Customers] from Accounts a inner join Customers c on c.Account = a.Id group by c.Address,a.Account;

select c.accno,c.name,t.amount,c.Account from

Customers c inner join AccountTransactions t on t.Accno = c.Accno;

select c.accno,c.name,sum(t.amount) from customers c inner join AccountTransactions t on t.Accno = c.Accno group by c.Accno,c.name;

select c.accno, c.name, count(t.amount) as[countTrans] from Customers c inner join AccountTransactions t on t.Accno = c.Accno group by c.Accno ,c.Name;

select c.accno,c.name,sum(t.amount) as [total deposit] from Customers c inner join AccountTransactions t

on t.Accno = c.Accno where t.TTYpe = 1 group by c.Accno,c.Name;

select c.accno,c.name,sum(t.amount) as [total withdraw] from Customers c inner join AccountTransactions t

on t.Accno = c.Accno where t.TTYpe = 2 group by c.Accno,c.Name;

select c.accno,c.name,DATEPART(yyyy,t.Dot) as [year] , count(t.amount) as [No-Withdraw] from

Customers c inner join AccountTransactions t

on t.Accno = c.Accno where t.TTYpe=2 and DATEPART(yyyy,t.Dot)=2021

group by c.accno,c.Name,DATEPART(yyyy,t.Dot);

select c.accno,c.name,DATEPART(yyyy,t.Dot) as [year] , count(t.amount) as [No-Withdraw] from

Customers c inner join AccountTransactions t

on t.Accno = c.Accno where t.TTYpe=1 and DATEPART(yyyy,t.Dot)=2021

group by c.accno,c.Name,DATEPART(yyyy,t.Dot);

select c.accno,c.name,DATEPART(yyyy,t.Dot) as [year] , count(t.amount) as [No-Withdraw] from

Customers c inner join AccountTransactions t

on t.Accno = c.Accno where t.TTYpe=2

group by c.accno,c.Name,DATEPART(yyyy,t.Dot);

select c.accno,c.name,DATEPART(yyyy,t.dot) as[year],count(t.amount) as [NoWith]

from Customers c inner join AccountTransactions t

on t.Accno =c.Accno

where t.TTYpe = 2 and

datepart(yyyy,t.Dot)=2019

group by c.Accno,c.Name,DATEPART(yyyy,t.dot) having count(t.amount)>5;

select c.accno,c.name,DATEPART(yyyy,t.Dot) as[year], datename(mm,t.Dot) as [month] , sum(t.amount)as [totalAmount], count(t.amount) as [noWithDraw]

from Customers c inner join AccountTransactions t

on t.Accno = c.Accno

where t.TTYpe = 2 and DATEPART(yyyy,t.Dot)=2019

group by c.Accno,c.Name,DATEPART(yyyy,t.Dot),DATENAME(mm,t.Dot) order by noWithDraw;

select c.accno,c.name,DATEPART(yyyy,t.Dot) as[year], datename(mm,t.Dot) as [month] , sum(t.amount)as [totalAmount], count(t.amount) as [noWithDraw]

from Customers c inner join AccountTransactions t

on t.Accno = c.Accno

where t.TTYpe = 2 and DATEPART(yyyy,t.Dot)=2019

group by c.Accno,c.Name,DATEPART(yyyy,t.Dot),DATENAME(mm,t.Dot) having count(t.amount)>4 order by noWithDraw;

select \* from AccountTransactions

select c.accno,c.name,DATEPART(yyyy,t.dot) as [year], sum(t.amount) as [AmountDeposit]

from Customers c inner join AccountTransactions t on t.Accno = c.Accno where t.TTYpe =1 and DATEPART(yyyy,t.Dot) in (2017,2018,2019)

group by c.Accno,c.name,DATEPART(yyyy,t.Dot) having sum(t.amount) > 100000;

select distinct(Accno) from AccountTransactions;

select top 1 \* from Employee order by salary desc;

select top 2 \* from Employee order by salary desc;

select top 3 \* from Employee order by salary desc;

select distinct(Account) from Customers order by Account;

select accno,name,

case(status)

when 1 then 'Active'

else 'DeActive'

end as [Staaaatus]

from Customers where status =1;

select accno,name,address,

case(Account)

when 1 then 'Savings'

when 2 then 'Current Account'

when 3 then 'FD'

when 4 then 'Business'

else 'Other Account'

end as [Account]

from Customers;

select \* from employee;

--user defined store procedure

create procedure sp\_RetriveEmployeeInfo

as

select \* from employee;

exec sp\_RetriveEmployeeInfo;

exec sp\_changeowner

Sp\_help

Sp\_helpdb

create procedure sp\_RetriveEmployeeInfo

as

select \* from employee;

select \* from student;

create procedure sp\_InsertStudent

@id int,

@name varchar(30),

@mobile bigint,

@status int

as

begin

insert into student(id,name,mobile,status) values (@id,@name,@mobile,@status)

end

exec sp\_InsertStudent 4,'sathya','9874563214',2

create procedure sp\_UpdateStudent

@sid int,

@mobilenum bigint

as begin

update student set mobile = @mobilenum where id = @sid

end

exec sp\_UpdateStudent 4,'9743519113'

create procedure sp\_deleteStudent

@id int

as

begin

delete from student where id = @id

end

select \* from student;

exec sp\_deleteStudent 2

-----------------------------------

create trigger trgStCount on student

after insert

as begin

declare @count int

select @count=count(\*) from student

print 'Totl number of student are count = '+cast(@count as varchar(5))

end

insert into student values(5,'Artidsf','2148234234',1);

drop trigger trgStutCount