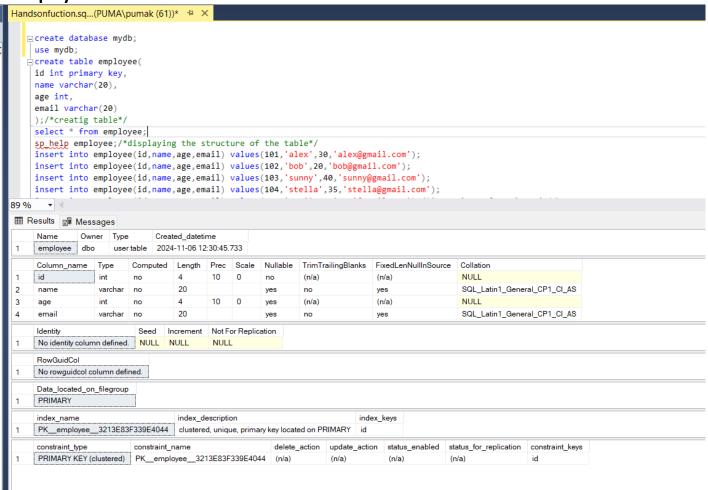
Day-3 Hands on and Functions in Sql

Name: Aathirainathan P

Date: 06-11-2024

1. Display table structure:



2.Alter Table:

```
select distinct name from employee;/*using distinct constraint*/
alter table employee add salary int;/*add one more column into table*/
select *from employee;
update employee set salary=20000 where name='alex';
update employee set salary=30000 where name='bob';
update employee set salary=35000 where name='sunny';

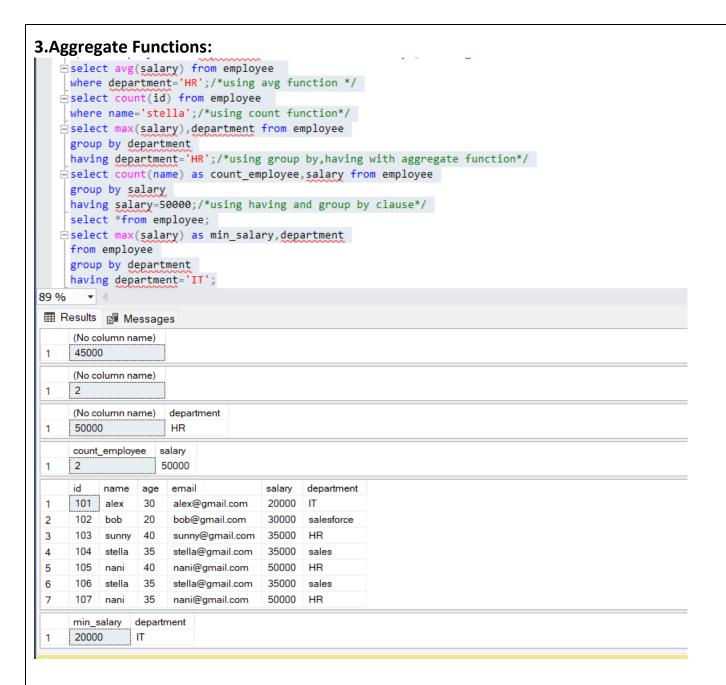
) %

**

Messages

Commands completed successfully.

Completion time: 2024-11-06T12:36:30.8030127+05:30
```



4. Rollback a transaction:

```
begin tran;
     delete from employee where age=40;
     rollback;/*rollback the transaction*/
     insert into employee(id,name,age,email,salary,department) values(101,'aparna',20,'aparna@gmail.com',30000,'sales');
     insert into employee(id,name,age,email,salary,department) values(102,'stella',20,'stella@gmail.com',40000,'HR');
     select *from employee;
     begin tran;
     delete from employee where age=20;
89 %

    Messages

   Commands completed successfully.
   Completion time: 2024-11-06T12:40:18.2809430+05:30
```

5. Savepoint and rollback: begin tran; delete from employee where id=102; save transaction s1;/*create a savepoint s1*/ rollback transaction s1; select *from employee; begin tran; insert into employee(id,name,age,email,salary,department) values(103,'bob',45,'bob@gmail.com',30000,'IT'); save transaction s2;/*creating the savepoint s2*/ Messages Commands completed successfully. Completion time: 2024-11-06T12:41:36.5246230+05:30 6.Union, Union all, Intersection, Except: | INSERT THEO EMPTING HAME, age , email, salary, wepartment/ values(200, millio, 40, milliomg mail.com, joodoo, nr /, select *from emp;/*performing union with two tables*/ select *from employee select *from emp;/*performing union all with two tables*/ select *from employee select *from emp;/*performing intersection between the tables*/ select *from emp except select* from employee;/*performing except operation between two tables*/ 89 % name age email salary department 101 alex 30 alex@gmail.com 20000 IT 2 104 stella 35 stella@gmail.com 35000 sales 105 stella 3 20 stella@gmail.com 40000 HR 106 stella 35 stella@gmail.com 35000 sales 4 5 107 28 60000 salesforce lucky lucky@gmail.com 6 107 nani 35 nani@gmail.com 50000 HR 108 nimmi 30 nimmi@gmail.com 30000 IT 7 109 krish 30 krish@gmail.com 45000 sales age email salary department name 20000 IT 101 alex 30 alex@gmail.com 35 stella@gmail.com 35000 sales 2 104 stella 105 stella 20 stella@gmail.com 40000 HR 3 4 106 stella 35 stella@gmail.com 35000 sales 50000 HR 5 107 nani 35 nani@gmail.com 108 nimmi 30 nimmi@gmail.com 30000 IT 6 7 109 krish 30 krish@gmail.com 45000 sales 45 bob@gmail.com 30000 IT 201 bob 8 id name age email salarv department nimmi@gmail.com 30000 IT 108 nimmi 30 name age email salary department Query executed successfully. PUMA\SQLEXPRE

7.Joins: select name, cid, department from employee course on employee.id=course.id;/*performing the natural join*/ $\dot{\sqsubseteq}$ select * from employee cross join dicourse ;/*performing the cross join*/ /*Performing equi join*/ where employee.id=course.id; /*Perfoming non-equi join*/ where employee.id>=course.id; /*Perfoming self join*/ select e1.name,e1.id,e1.salary from employee e1,employee e2 where e1.salary=e2.salary and e2.name='krish'; select *from employee; % ▼ ∢ Results Messages krish 109 45000

8.String Functions: /* String functions*/ select ascii('CB'); /*return leftmost ascii value*/ select char(66); /*return ascii value to character*/ select len('Microsift sql');/*return length*/ select lower('JHON');/*convert to lowercase*/ select replace('Microsoft sql','sql','server');/*replace*/ select reverse('python');/*reverse the string */ select upper('aparna');/*converts to upper*/ =select str(136.564,8,4);/*STR(number, length, decimals)*/ /* Data Functions*/ 89 % - ▼ - ((No column name) 67 (No column name) 1 В (No column name) (No column name) jhon 1 (No column name) Microsoft server 1 (No column name) nohtyp (No column name) APARNA (No column name) 136.5640

9.Date Functions:

```
/* Date Functions*/
    select getdate ();/*get current date and time*/
    select dateadd (mm, 2, '2023-12-07');/*add months to existed date */
select datediff ( year, convert (datetime, '2006-05-06'), convert ( datetime, '2009-01-01'));/*it will return the difference of date,months,years also*/
select datepart (mm, '2008-5-22');/*return months value*/
    select day ( '2023-05-30');/*return value of date of that particular day*/
     select month ('2023-05-31');/*return month value*/
  select year ( '2023-05-3');/*return year value*/
    /*Mathematical Functions*/
    select abs(-101);/*returns absolute value*/
    select sin(1.5);/*returns angle in radians*/
solart callent/1.01\-/*naturns the smallest on greater to the spacified value*/
9% *
(No column name)
   2024-11-06 12:53:31.587
  2024-02-07 00:00:00.000
     (No column name)
   3
    5
     (No column name)
  30
     (No column name)
   5
     (No column name)
    2023
```

10.Mathematical Functions: /*Mathematical Functions*/ select abs(-101);/*returns absolute value*/ select sin(1.5);/*returns angle in radians*/ select ceiling(14.01);/*returns the smallest or greater to the specified value*/ select exp(4.5);/*returns the exponencial value*/ select floor(14.75); /*Rankig Functions*/ /* row()_number-giving consecutive numbers to rank*/ select * from employee; **▼** 4 ■ 3 % ■ Results Messages (No column name) 101 (No column name) 0.997494986604054 (No column name) (No column name) 90.0171313005218 (No column name) (No column name) 1.68639895357023

11. Ranking Functions:

```
/* row()_number-giving consecutive numbers to rank*/
     select * from employee;
   = select id, name, salary, ROW_NUMBER() over(order by salary desc) as rownumber
     from employee;
     select * from employee;
     /*rank()-used to give rank if duplicates allowed ranking will be changed based on duplicates
   = select id, name, salary, rank() over(order by salary) as rank
     from employee;
     /*dense rank()-used to give ranks consecutively even if duplicates are allowed*/
   select id,name,salary,dense_rank() over(order by salary desc) as rank
     from employee;
     /*ntile() function- it will divide give the rank in groups*/
   select id, name, salary, ntile(2) over(order by salary) as rank
     from employee; /*without condition*/
   ∃select name, salary, ntile(4) over(order by salary) as rank
   from employee where salary>10000;/*with condition*/
98 %
name
               age
                                   salary
                                         department
     101
                                    20000 IT
          alex
                    alex@gmail.com
2
     104
          stella
                35
                     stella@gmail.com
                                   35000 sales
3
     105
          stella
                20
                    stella@gmail.com
                                   40000 HR
                35
                                   35000
4
     106
          stella
                    stella@gmail.com
                                          sales
     107 nani
                35
                    nani@gmail.com
                                   50000
                                         HR
5
6
     108
          nimmi
               30
                     nimmi@gmail.com
                                   30000
                                          IT
                30
                    krish@gmail.com
                                   45000
7
     109
          krish
                                          sales
     id
          name
                salary rownumber
    107 nani
                50000 1
1
                45000 2
     109
2
          krish
                40000 3
3
     105
          stella
                35000 4
4
     106
          stella
          stella
                35000 5
5
     104
     108
          nimmi
                30000 6
     101
                20000 7
7
          alex

    Query executed successfully.
```

12. System Functions: /*System functions*/ select host_name() as Host_name;/*return system hostname*/ select Host_ID();/*return host id*/ select SUSER_ID(); select USER_ID(); select DB_NAME();/*return database name*/ /*Aggregate funtions*/ select count(*) as total_records from employee; select count(*) from employee where salary>30000; 98 % Host_name PUMA (No column name) 20396 (No column name) (No column name)