

Stored Procedure

Assignment 4.1

Question 1: Create a procedure for adding a row in emp table. Display number of rows added. Handle corresponding exception

```
delimiter //

drop procedure if exists addTableRow//

create procedure addTableRow(

in e_no int,

in e_name varchar(20),

in e_job varchar(20),

in e_mgr int,

in e_hiredate date,

in e_sal int,

in e_comm int,

in e_deptno int

)

begin

declare tot int;

declare exit handler for 1062

select "Duplicate Entry" as Error_Message;
```

```
insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno)
values(e_no,e_name,e_job,e_mgr,e_hiredate,e_sal,e_comm,e_deptno);

select count(*) into tot from emp;

select tot as Total_rows;

end//

call addTableRow(7980,'Rehan','DEVELOPER',7839,'2020-08-23',5000,300,50)//
```

Question 2: Create a procedure which takes employee number as parameter and delete the employee record. Procedure must return the empname whose record has be deleted.

```
delimiter //
drop procedure if exists delEmp//
create procedure delEmp(in e_no int ,out e_name varchar(20))
begin
select ename into e_name from emp where empno=e_no;
delete from emp where empno=e_no;
end//
call delEmp(7960,@ename)//
select @ename as Employee//
```

Question 3: Create a procedure which takes department name as parameter and displays all the employee records, no of employee in that department.

```
delimiter //
drop procedure if exists getDeptName//
create procedure getDeptName(in d_name varchar(20))
begin
declare employee int;
select count(*) into employee from emp where deptno=(select deptno from dept
where dname=d_name);
```

```

select * from emp where deptno=(select deptno from dept where
dname=d_name);
select employee as number;
end//
call getDeptName('sales')//
or
delimiter //
drop procedure if exists getDeptName//
create procedure getDeptName(in d_name varchar(20))
begin
select count(*)as employee from emp where deptno=(select deptno from dept
where dname=d_name);
select * from emp where deptno=(select deptno from dept where
dname=d_name);
end//
set @dname='sales'//
call getDeptName(@dname)//

```

Question 4: Create a procedure which takes empno as IN parameter and gives OUT entire record of the employee

```

delimiter //
drop procedure if exists getEmployee//
create procedure getEmployee(
inout e_no int,
out e_name varchar(20),
out e_job varchar(20),
out e_mgr int,
out e_hiredate date,
out e_sal double(7,2),
out e_comm double(7,2),
out e_deptno int
)
begin

```

```

select ename,job,mgr,hiredate,sal,comm,deptno into
e_name,e_job,e_mgr,e_hiredate,e_sal,e_comm,e_deptno from emp where
empno=e_no;
end//
set @e_no=7900//
call
getEmployee(@e_no,@e_name,@e_job,@e_mgr,@e_hiredate,@e_sal,@e_com
m,@e_deptno)//
select
@e_no,@e_name,@e_job,@e_mgr,@e_hiredate,@e_sal,@e_comm,@e_deptno
//

```

Question 5: create a prcedure which would accept 2 dates as arguments and returns

difference between them in years and months Eg: "3 years 6 months"

```

delimiter //
drop procedure if exists testdate//
create procedure testdate(in fdt date,in sdt date,out y int,out m int)
begin
declare y2 int;
declare y1 int;
declare m2 int;
declare m1 int;
-- extract year
select extract(year from fdt) into y1;
select extract(year from sdt) into y2;
set y=y2-y1;
-- extract month
select extract(year from fdt) into m1;
select extract(year from sdt) into m2;
set m=m2-m1;
end//
set @fdt='2010-05-15'//
set @sdt='2020-05-15'//

```

```
call testdate(@fdt,@sdt,@y,@m)//  
select concat(@y,' years ',@m,' month') as datediff//
```

Question 6: Create table with following structures

- **emp_bonus(empno,ename,annual_salary,bonus)**
- **using cursor program fetch empno,ename,sal,comm from emp table.**
- **for each record fetched find annual salary which includes comm**
- **compute various percentage of bonus based on annual salary.**
- **insert empno,ename,annual_salary,bonus into emp_bonus table**
- **display number of rows inserted.**

```
create table emp_bonus(  
empno int primary key,  
ename varchar(20),  
annual_salary double(7,2),  
bonus double(7,2)  
)  
delimiter //  
drop procedure if exists insinbonus//  
create procedure insinbonus()  
begin  
declare e_no int;  
declare e_name varchar(20);  
declare e_sal double(7,2);  
declare e_comm double(7,2);  
declare a_sal double(7,2);  
declare e_bonus double(7,2);  
declare eoc int default 0;  
declare count_row int;  
declare c1 cursor for  
select empno,ename,sal,comm from emp;  
declare continue handler for not found set eoc=1;  
open c1;  
l:loop  
fetch c1 into e_no,e_name,e_sal,e_comm;
```

```

if(eoc=1)then
leave l;
end if;
set a_sal=e_sal*12+ifnull(e_comm,0);
if(a_sal>25000) then
set e_bonus=4000;
else if (a_sal>15000 and a_sal<25000) then
set e_bonus=3000;
else if(a_sal>10000 and a_sal<15000) then
set e_bonus=1000;
else
set e_bonus=600;
end if;
end if;
end if;
insert into emp_bonus values(e_no,e_name,a_sal,e_bonus);
end loop;
close c1;
select count(*) into count_row from emp_bonus;
select count_row as Total;
end//
call insinbonus();//

```

Question 7: Design a cursor program in which you select empno,ename,job,sal into a collection and display the same for any department number passed as argument to cursor.

```

delimiter //
drop procedure if exists curPro//
create procedure curPro(in dept_no int)
begin
declare eoc int default 0;
declare e_no int;
declare e_name varchar(20);
declare e_job varchar(20);
declare e_sal double(7,2);
declare c1 cursor for

```

```

select empno,ename,job,sal from emp where deptno=dept_no;
declare continue handler for not found set eoc=1;
open c1;
myloop:loop
fetch c1 into e_no,e_name,e_job,e_sal;
if(eoc=1) then
leave myloop;
end if;
select e_no,e_name,e_job,e_sal;
end loop;
close c1;
end//
call curPro(10)//

```

Question 8: Write a procedure to create a cursor displays the name and salary of each employee in the EMPLOYEES table whose salary is less than that specified by a passed-in parameter value.

```

delimiter //
drop procedure if exists disemp//
create procedure disemp(in pass_sal int)
begin
declare e_name varchar(20);
declare e_sal int;
declare eoc int default 0;
declare str varchar(500);
declare c1 cursor for
select ename,sal from emp where sal>pass_sal;
declare continue handler for not found set eoc=1;
open c1;
myloop:loop
fetch c1 into e_name,e_sal;
if(eoc=1) then
leave myloop;
end if;
set str=concat_ws('_',str,e_name,e_sal,'\n');

```

```

end loop;
select str as EmployeeDetails;
close c1;
end//
call disemp(2000)//
the below statment can also be used to display the result/output
select concat_ws('_',e_name,e_sal) as Employee;
set str=concat_ws('_',str,e_name,e_sal,'\n');
select e_name,e_sal ;

```

Question 9: Write a procedure to display employee id, name, hire date, and the incentive amount they achieved according to their working experiences,

```

delimiter //
drop procedure if exists disemp//
create procedure disemp()
begin
declare e_id int ;
declare e_name varchar(20);
declare e_hiredate date;
declare e_job varchar(20);
declare incentive_amount double(7,2);
declare exp int;
declare str varchar(4000);
declare eoc int default 0;
-- declare cursor
declare c1 cursor for
select empno,ename,hiredate,job from emp;
-- declare handler to check row is avaiable or not
declare continue handler for not found set eoc=1;
open c1;
-- start loop
myloop:loop
-- fetch recode
fetch c1 into e_id,e_name,e_hiredate,e_job;

```



```

-- close loop if now row found
if(eoc=1) then
leave myloop;
end if;
-- calculate experience
set exp=datediff(now(),e_hiredate)/365;
if(exp>50) then
set incentive_amount=5000;
else if(exp>45 and exp<50) then
set incentive_amount=4000;
else if(exp>40 and exp<45) then
set incentive_amount=3000;
else if(exp>35 and exp<40) then
set incentive_amount=2000;
else if(exp>30 and exp<35) then
set incentive_amount=1000;
else
set incentive_amount=500;
end if;
end if;
end if;
end if;
end if;
-- store recode as a string
set str=concat_ws('_',str,e_id,e_name,e_hiredate,incentive_amount,'\n');
end loop;
-- display recode
select str as empdetails;
close c1;
end//
call disemp()//
below statmetn you can also used to display reoced
select e_id,e_name,e_hiredate,incentive_amount;

```

Question 10 : Write procedure to display the department with more than five employees.

a) simple way:

```
delimiter //
drop procedure disDep//
create procedure disDep(in num int)
begin
select deptno,count(deptno)as Dept from emp group by deptno having
count(deptno)>num;
end//
call disDep(5)//
```

b) using cursor:

```
delimiter //
drop procedure disDep//
create procedure disDep(in num int)
begin
declare dept_no int;
declare employee int;
declare str varchar(500);
declare eoc int default 0;
declare c1 cursor for
select deptno,count(deptno) from emp group by deptno having
count(deptno)>num;
declare continue handler for not found set eoc=1;
open c1;
myloop:loop
fetch c1 into dept_no,employee;
if(eoc=1) then
leave myloop;
end if;
select dept_no,employee;
-- set str=concat(str,d_no,e_count,'\n');
end loop;
```

```
-- select str as dept_emp;  
close c1;  
end//  
call disDep(5)//
```