Course: Database Management Systems (DBMS)

Assesment 4: LOOPS

Done by: Prithak Gajurel Registration Number: 20BCE2921

Write down (handwritten) the respective queries for each Qs. Answer should contain the screenshot of the execution

In a single output window, print odd numbers upto 20 using LOOP [2]

 Using DO-WHILE LOOP insert the following data into table "tourism" and show the table content.

ID	Place
2	Place3
4	Place5
6	Place7
8	Place9
10	Place11
12	Place13

- 3. Using paging and DO-WHILE display the "tourism" contents in three pages (o/ps). [2]
- 4. Using procedure display the tuples of "tourism" where ID is either 4 or 8 or 12. [2]
- Create a new table "place_name" having only one field "name". Using nested stored procedure and DO-WHILE/LOOP copy the place name from the "tourism" table having ID = 2, 6, 10 and insert it to "place_name" table.

Initial creation of table:

Code:

```
• create table tourism(

Id int primary key,

Place varchar(50));
```

desc tourism;

Output:

	Field	Туре	Null	Key	Default	Extra
•	Id	int	NO	PRI	NULL	
	Place	varchar(50)	YES		NULL	

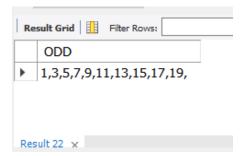
1. In a single output window, print odd numbers upto 20 using LOOP

```
#1
DELIMITER $$
#DROP PROCEDURE IF EXISTS print_odd;
CREATE PROCEDURE print_odd()
BEGIN
   DECLARE x INT;
   DECLARE ODD VARCHAR(100);
    SET x = 1;
    SET ODD='';
   loop_odd: LOOP
       SET ODD = CONCAT(ODD,x,',');
       set x = x + 2;
       IF x>20 THEN
            LEAVE loop_odd;
             _____
         END IF;
     END LOOP;
     select ODD;
 END$$
 call print_odd();
```

HANDWRITTEN:

```
#1
DELIMITER $$
#DROP PROCEDURE IF EXISTS print_odd;
CREATE PROCEDURE print_odd()
BEGIN
DECLARE × INT:
DECLARE ODD VARCHAR(100):
SET x = 1:
SET ODD=":
loop_odd: LOOP
SET ODD = CONCAT(ODD,x,',');
set x = x + 2:
IF x>20 THEN
LEAVE loop_odd;
END IF:
END LOOP:
select ODD:
END$$
call print_odd();
```

OUTPUT:



 Using DO-WHILE LOOP insert the following data into table "tourism" and show the table content.

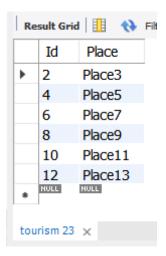
ID	Place
2	Place3
4	Place5
6	Place7
8	Place9
10	Place11
12	Place13

```
#2
Delimiter $$
Drop procedure if exists ins_tourism;
create procedure ins_tourism()
begin
declare a int;
declare xyz varchar(50);
set a = 2;
set xyz = '';
while a <=12 do
    set xyz = concat('Place',a+1);
    insert into tourism
    values(a,xyz);
    set a = a + 2;
      end while;
  end$$
  call ins_tourism();
  select * from tourism;
  truncate table tourism;
```

HANDWRITTEN CODE:

```
#2
Delimiter $$
Drop procedure if exists ins_tourism;
create procedure ins_tourism()
begin
declare a int;
declare xyz varchar(50);
set a = 2;
set xyz = ";
while a <= 12 do
set xyz = concat('Place', a+1);
insert into tourism
values(a,xyz);
set a = a + 2;
end while;
end$$
call ins_tourism();
select * from tourism;
truncate table tourism:
```

OUTPUT:



3. Using paging and DO-WHILE display the "tourism" contents in three pages (o/ps).

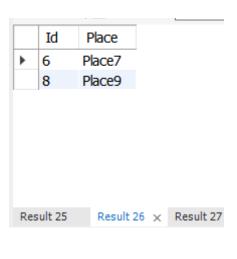
[2]

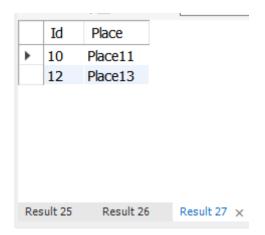
HANDWRITTEN CODE:

```
#3
Delimiter $$
DROP PROCEDURE IF EXISTS page_tourism;
CREATE PROCEDURE page_tourism()
begin
DECLARE count INT;
SET count = 0;
WHILE count (Select count (Id) from tourism) do
SELECT * FROM tourism limit 2 offset count;
SET count = count + 2;
END while;
end$$
call page_tourism();
```

OUTPUTS:

	Id	Place
١	2	Place3
	4	Place5





4. Using procedure display the tuples of "tourism" where ID is either 4 or 8 or 12.

```
#4
Delimiter $$
DROP PROCEDURE IF EXISTS loopdemo;
CREATE PROCEDURE loopdemo()

BEGIN
declare a int;
set a = 4;

while a<= (Select max(Id) from tourism) do
    select Id,Place from tourism
    where Id = a;
    set a = a + 4;

end while;
end$$
call loopdemo();</pre>
```

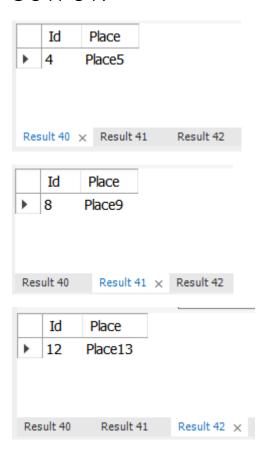
```
Delimiter $$
DROP PROCEDURE IF EXISTS disp_tup;
CREATE PROCEDURE disp_tup()

BEGIN
call loopdemo();
end$$
call disp_tup();
```

HANDWRITTEN CODE:

```
#4
Delimiter $$
DROP PROCEDURE IF EXISTS loopdemo;
CREATE PROCEDURE loopdemo()
BEGIN
declare a int;
set a = 4:
while a = (Select max(Id) from tourism) do
select Id. Place from tourism
where Id = a;
set a = a + 4;
end while;
end$$
call loopdemo();
Delimiter $$
DROP PROCEDURE IF EXISTS disp_tup;
CREATE PROCEDURE disp_tup()
BEGIN
call loopdemo();
end$$
call disp_tup();
```

OUTPUT:



Create a new table "place_name" having only one field "name". Using nested stored procedure and DO-WHILE/LOOP copy the place name from the "tourism" table having ID = 2, 6, 10 and insert it to "place_name" table.

```
CODE:
#5
create table place_name (Name varchar (100));
Delimiter $$
DROP PROCEDURE IF EXISTS np;
CREATE PROCEDURE np()
BEGIN
insert into place_name values(@contname);
end$$
Delimiter $$
DROP PROCEDURE IF EXISTS While_Loop;
CREATE PROCEDURE While_Loop()
BEGIN
DECLARE Counter, MaxId INT;
DECLARE ContName VARCHAR(100);
set counter = (SElect min(Id) from tourism);
set maxID = (SElect max(Id) from tourism);
set @contname='';
while (Counter-1 <= MaxId) do
SET @ContName = (select Place FROM tourism WHERE Id = counter);
call np();
SET Counter = Counter + 4;
END while:
end$$
call While_Loop();
select * from place_name;
drop table place_name;
```

Handwritten code:

```
#5
create table place_name (Name varchar (100));
Delimiter $$
DROP PROCEDURE IF EXISTS np:
CREATE PROCEDURE np()
BEGIN
insert into place_name values (@contname):
end$$
Delimiter $$
DROP PROCEDURE IF EXISTS While_Loop;
CREATE PROCEDURE While_Loop()
BEGIN
DECLARE Counter, MaxId INT:
DECLARE Cont Name VARCHAR (100):
set counter = (SElect min(Id) from tourism):
set maxID = (SElect max(Id) from tourism):
set @contname=":
while (Counter-1 <= MaxId) do
SET @ContName = (select Place FROM tourism WHERE Id = counter );
call np();
SET Counter = Counter + 4:
END while:
end$$
call
```

```
While_Loop();
select * from place_name;
drop table place_name;
```

OUTPUT:

