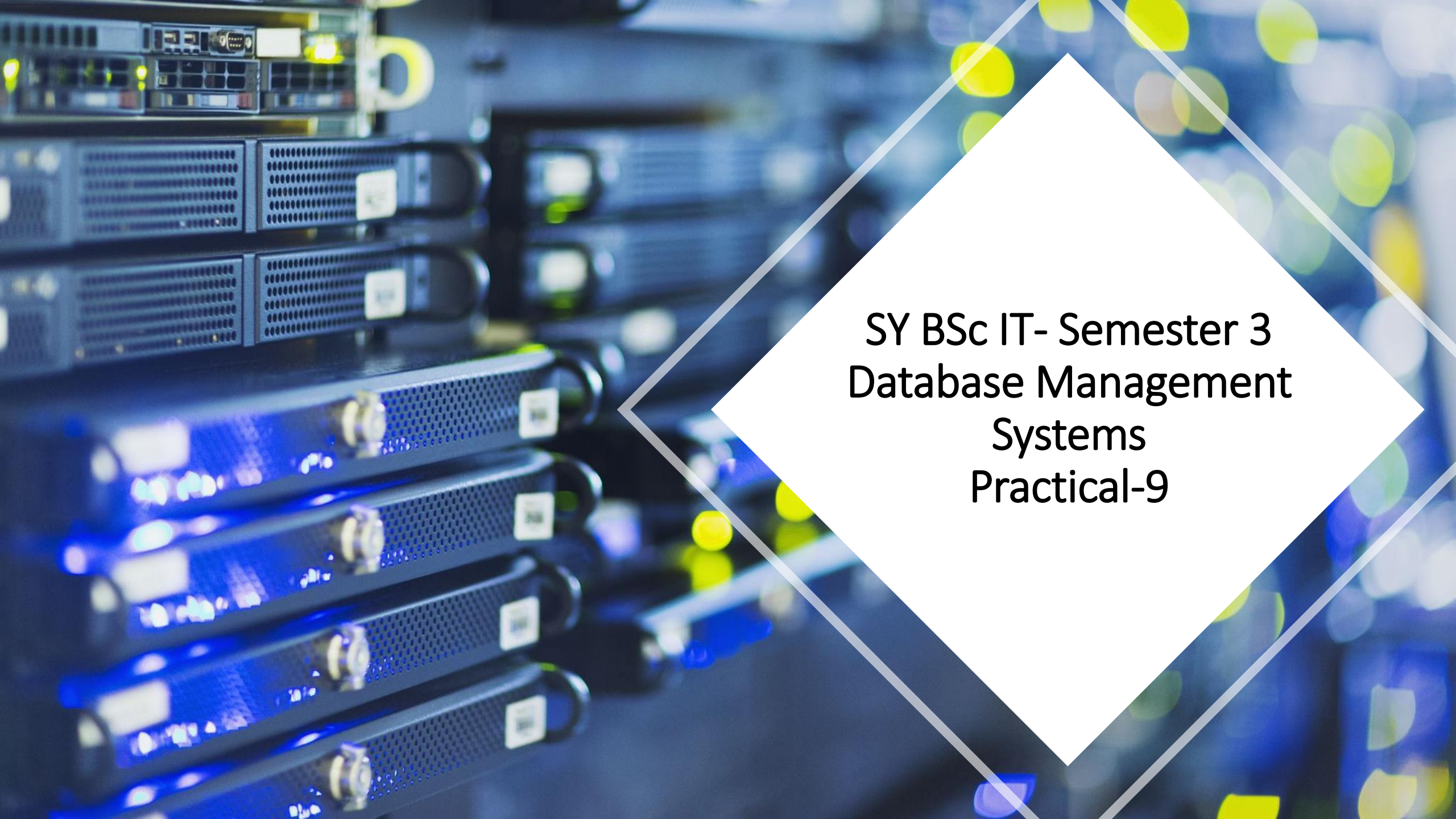

SY BSc IT- Semester 3
Database Management Systems
Practical-9

Revision of Practial-7

- Introduction to PL/SQL

Contents of Practical 9

- Procedure
- Function

Procedures / Functions

A Procedure or Function is a logically grouped set of SQL and PL/SQL statements that perform a specific task.

A stored procedure or function is named PL/SQL code that has been compiled and stored in Oracle

Procedures or Functions can be made dynamic by-passing parameters to it.

A Procedure or Function can then change the way it works depending upon the parameters passed

Procedures

- The syntax of procedure is

CREATE OR REPLACE PROCEDURE <Procedure Name>(<Argument> {IN, OUT, IN OUT} <Data Type>,.....) {IS, AS}

<Variable Declarations>;

<Constants Declaration>;

BEGIN

<PL/SQL sub program body>;

EXCEPTION

<Exception PL/SQL block>;

END; where

- Procedure Name is the name of the procedure to be created
- Argument is the name of the argument to be passed to the procedure
- IN (Default) indicates the parameter will only accept a value from the user
- OUT indicates the parameter will only return a value to the user
- IN OUT indicates the parameter will accept as well as return a value to and from the user
- Data Type indicates the data type of the associated parameter

Functions

- The syntax of function is

CREATE OR REPLACE FUNCTION <Function Name>(<Argument> IN <Data Type>,....) RETURN <Data Type> {IS, AS}

<Variable Declarations>;

<Constants Declaration>;

BEGIN

<PL/SQL sub program body>;

EXCEPTION

<Exception PL/SQL block>;

END; where

- Function Name is the name of the function to be created
- Argument is the name of the argument to be passed to the procedure
- IN indicates the parameter will only accept a value from the user
- Data Type indicates the data type of the associated parameter
- RETURN Data Type (Compulsory) indicates the data type of the functions return value

Procedures / Functions

- Invoking a Procedure can be done as follows

  Procedure Name(Argument List);

  OR

  CALL Procedure Name(Argument List);

- Invoking a Function can be done as follows

  Return Variable := Function Name(Argument List);

- Deleting a stored procedure

  DROP PROCEDURE <Procedure Name>;

- Deleting a stored function

  DROP FUNCTION <Function Name>;

Example For Procedure

- Write a procedure to add two numbers and call that procedure through the PL/SQL block

---

```
CREATE OR REPLACE PROCEDURE ADD_TWO_NOS
   IS
   A NUMBER(3):=10; B NUMBER(3):=20; C NUMBER(5);
BEGIN
   C:=A+B;
   DBMS_OUTPUT.PUT_LINE('SUM OF ' || A || ' AND ' || B || ' IS ' || C);
END;
/
```

Calling procedure:

```
BEGIN
   ADD_TWO_NOS;
END;
/
```

- sql>ed d:\f1.sql
- CREATE OR REPLACE PROCEDURE ADD_TWO_NOS IS
- A NUMBER(3):=10;
- B NUMBER(3):=20;
- C NUMBER(5);
- BEGIN
- C:=A+B;
- DBMS_OUTPUT.PUT_LINE('SUM OF ' || A || ' AND ' || B || ' IS ' || C);
- END;
- /
- To run type:
- sql>@ d:\f1.sql
- o/p: procedure created
- sql>ed d:\f2.sql
- BEGIN
- ADD_TWO_NOS;
- END;
- /
- sql>@ d:\f2.sql
- o/p: SUM OF 10 AND 20 IS 30
- PL/SQL procedure successfully completed.

Example For Procedure

- Write a procedure to add two numbers passed from the calling PL/SQL block.

```
CREATE OR REPLACE PROCEDURE ADD_TWO_PARAMETERNOS(A IN NUMBER,B IN NUMBER)
   IS
   C NUMBER(5);
BEGIN
   C:=A+B;
   DBMS_OUTPUT.PUT_LINE('SUM OF ' || A || ' AND ' || B || ' IS ' || C);
END;
/
DECLARE
   NUM_1 NUMBER(3); NUM_2 NUMBER(3);
BEGIN
   NUM_1:=&NUM_1; NUM_2:=&NUM_2;
   ADD_TWO_PARAMETERNOS(NUM_1,NUM_2);
END;
/
```

Example For Procedure

- Write a procedure to add two numbers passed from the calling PL/SQL block. The addition is done in the procedure but display of the sum is done in the calling program.

```
CREATE OR REPLACE PROCEDURE ADD_TWO_PARAMETERNOS(A IN NUMBER,B IN NUMBER,C OUT
   NUMBER) IS
BEGIN
   C:=A+B;
END;
/
DECLARE
   NUM_1 NUMBER(3); NUM_2 NUMBER(3); NUM_3 NUMBER(5);
BEGIN
   NUM_1:=&NUM_1; NUM_2:=&NUM_2;
   ADD_TWO_PARAMETERNOS(NUM_1,NUM_2,NUM_3);
   DBMS_OUTPUT.PUT_LINE('SUM OF ' || NUM_1 || ' AND ' || NUM_2 || ' IS ' || NUM_3);
END;
/
```

Example For Functions

- Write a function to add two numbers passed to it from main. The two numbers are accepted by them from the users

```
CREATE OR REPLACE FUNCTION ADD_TWO_NOS3(A IN NUMBER,B IN NUMBER) RETURN NUMBER IS
        C NUMBER(3);
BEGIN
        C:=A+B;  RETURN C;
END;
/
DECLARE
        X NUMBER(3); Y NUMBER(3); Z NUMBER(3);
    BEGIN
        X:=&X;    Y:=&Y;
        Z:=ADD_TWO_NOS3(X,Y);
        DBMS_OUTPUT.PUT_LINE('VALUE OF Z IS  '||Z);
END;
/
```

Write a PL/SQL function which will compute and return the maximum of two values:-

```
Create FUNCTION findMax(x IN number, y IN number)  RETURN number  IS     z number;
  BEGIN
  IF x > y THEN    z:= x;  ELSE    z:= y;
  END IF;
  RETURN z;
  END;
--Calling function
DECLARE
  a number;   b number;    c number;
BEGIN
  a:=&a;     b:= &b;
  c := findMax(a, b);
  dbms_output.put_line(' Maximum of (a, b): ' || c);
END;
```

Thank You