

DBMS Lab 11

Author: Dipankar Das

Date: 23-4-2022

Roll: 20051554

[Github Link](#)

Question 1

Enter an employee id from the user. If it exists, display the detail, otherwise, show a user defined error.

Solution

```
create table employee(  
    id number,  
    first varchar(20),  
    salary number(10),  
    hireDate date  
);  
set serveroutput ON;  
  
insert into employee values(01, 'emp', 2012344, '12-Oct-2001');  
insert into employee values(11, 'emp', 50000, '11-Mar-2021');  
insert into employee values(02, 'emp', 560000, '17-May-1990');  
insert into employee values(012, 'emp', 5600002, '17-Jun-1990');  
insert into employee values(123121, 'emp', 23434, '1-Nov-2019');  
insert into employee values(1123, 'emp', 3444, '12-Jan-2020');  
insert into employee values(1231, 'emp', 33344, '21-Oct-2000');  
insert into employee values(31, 'emp', 56744, '12-Oct-2001');  
insert into employee values(100, 'emp', 345, '28-Feb-2022');  
  
DECLARE  
    IDID employee.id%type;  
    ft employee.first%type;  
    sall employee.salary%type;  
    dat employee.hireDate%type;  
    usree employee.id%type;  
    errr EXCEPTION;  
    cnt NUMBER;  
BEGIN  
    usree := &usree;  
    select COUNT(*) into cnt from employee where usree=id;  
    if cnt = 0 then  
        RAISE errr;  
    else  
        select id, first, salary, hireDate into IDID, ft, sall, dat from employee  
        where usree=id;
```

```

        DBMS_OUTPUT.PUT_LINE(IDID||' '||ft||' '||sal||' '||dat);
    end if;

EXCEPTION
    when errr then
        DBMS_OUTPUT.PUT_LINE('NO -ve id');
    when OTHERS then
        DBMS_OUTPUT.PUT_LINE('No employee with given id');

END;

```

Output

```

21      DBMS_OUTPUT.PUT_LINE('NO -ve id');
22      when OTHERS then
23          DBMS_OUTPUT.PUT_LINE('No employee with given id');
24      END;
25      /
Enter value for usree: 1
old 10:      usree := &usree;
new 10:      usree := 1;
1 emp 2012344 12-OCT-01

PL/SQL procedure successfully completed.

SQL>

```

Question 2

Let empid 100 has been suspended from the company. Update the salary of the inputted empid with a increment of 15% except empid 100.

1. Show an error message for this type of exception.
2. Show an error code and error message for this.
3. Link the error code to the exception name.

Solution

```

create table employee(
    id NUMBER,
    name VARCHAR(20),
    salary NUMBER,
    qualif VARCHAR(20),
    qualif_id NUMBER
);

insert into employee values(1, 'emp', 2012344, 'B.Tech.', 23);

```

```

insert into employee values(11, 'emp', 50000, 'M.Tech.', 64);
insert into employee values(2, 'emp', 560000, 'Phd.', 13);
insert into employee values(12, 'emp', 5600002, 'B.S.', 1);
insert into employee values(123121, 'emp', 23434, 'Dr.', 43);
insert into employee values(1123, 'emp', 3444, 'Dr.', 43);
insert into employee values(1231, 'emp', 33344, 'Phd.', 13);
insert into employee values(31, 'emp', 56744, 'B.Tech.', 23);
insert into employee values(3, 'emp', 345, 'Gs', 3);

set SERVEROUTPUT ON;
declare
    usree employee.id%type;
    baseEEE EMPLOYEE%ROWTYPE;
    CURSOR emp(usree employee.id%type) is select * from EMPLOYEE where usree =
EMPLOYEE.id
        for update of SALARY nowait;
    errr EXCEPTION;
    invID EXCEPTION;
    PRAGMA EXCEPTION_INIT(invID, -20000);
begin
    usree := &usree;
    if usree = 100 then
        RAISE invID;
    end if;

    OPEN emp(usree);

    if emp%ISOPEN then
        fetch emp into baseEEE;
        if emp%NOTFOUND then
            RAISE errr;
        end if;
        UPDATE employee SET SALARY=1.15*baseEEE.SALARY WHERE CURRENT OF emp;

    else
        DBMS_OUTPUT.PUT_LINE('unable to open cursor');
    end if;

    close emp;
EXCEPTION
    when invID then
        DBMS_OUTPUT.PUT_LINE(SQLERRM||' This employee is suspended');
    when errr then
        DBMS_OUTPUT.PUT_LINE('No employee with given id');
END;

select * from EMPLOYEE;

```

Output

```

34      DBMS_OUTPUT.PUT_LINE('No employee with given id');
35  END;
36  /
Enter value for usree: 11
old 10:  usree := &usree;
new 10:  usree := 11;

PL/SQL procedure successfully completed.

SQL> select * from employee;

   ID NAME                SALARY QUALIF                QUALIF_ID
-----
    1 emp                2012344 B.Tech.                23
   11 emp                 575000 M.Tech.                64
    2 emp                 560000 Phd.                13
   12 emp                 5600002 B.S.                1
  123121 emp                23434 Dr.                43
  1123 emp                 3444 Dr.                43
  1231 emp                 33344 Phd.                13
   31 emp                 56744 B.Tech.                23
    3 emp                  345 Gs                3

9 rows selected.

SQL>

```

```

SQL>
SQL>
SQL> select * from employee;

no rows selected

SQL> select * from employee;

   ID NAME                SALARY QUALIF                QUALIF_ID
-----
    1 emp                2012344 B.Tech.                23
   11 emp                 500000 M.Tech.                64
    2 emp                 560000 Phd.                13
   12 emp                 5600002 B.S.                1
  123121 emp                23434 Dr.                43
  1123 emp                 3444 Dr.                43
  1231 emp                 33344 Phd.                13
   31 emp                 56744 B.Tech.                23
    3 emp                  345 Gs                3

9 rows selected.

SQL>

```

Question 3

Write a PL/SQL block to retrieve employees from the EMPLOYEE table based on a qualification Id. If the qualification Id returns more than one row, handle the exception with the appropriate handler and print the message 'More than one employee with such qualification'. If the qualification Id returns no employee, handle the exception with the appropriate handler and display the message 'No employees with such qualification'. If the qualification Id returns one employee, then print that employee's name, qualification and salary (predefined server exception problem).

Solution

```

create table employee(
  id NUMBER,
  name VARCHAR(20),
  salary NUMBER,
  qualif VARCHAR(20),
  qualif_id NUMBER
);

insert into employee values(1, 'emp', 2012344, 'B.Tech.', 23);
insert into employee values(11, 'emp', 500000, 'M.Tech.', 64);
insert into employee values(2, 'emp', 560000, 'Phd.', 13);
insert into employee values(12, 'emp', 5600002, 'B.S.', 1);
insert into employee values(123121, 'emp', 23434, 'Dr.', 43);
insert into employee values(1123, 'emp', 3444, 'Dr.', 43);
insert into employee values(1231, 'emp', 33344, 'Phd.', 13);
insert into employee values(31, 'emp', 56744, 'B.Tech.', 23);
insert into employee values(3, 'emp', 345, 'Gs', 3);

SET SERVEROUTPUT ON;
DECLARE
  nam EMPLOYEE.NAME%type;
  qId EMPLOYEE.QUALIF_ID%type;
  sal1 EMPLOYEE.SALARY%type;
  qualif EMPLOYEE.QUALIF%type;

  nam1 EMPLOYEE.NAME%type;
  qId1 EMPLOYEE.QUALIF_ID%type;
  sal11 EMPLOYEE.SALARY%type;

```

```
qualif1 EMPLOYEE.QUALIF%type;

CURSOR curEmp(qId EMPLOYEE.QUALIF_ID%TYPE) is
    select NAME, SALARY, QUALIF, QUALIF_ID
    from EMPLOYEE WHERE qId = QUALIF_ID;
err01 EXCEPTION;
err02 EXCEPTION;

BEGIN
    qId := &qId;
    OPEN curEmp(qId);
    IF curEmp%ISOPEN THEN
        fetch curEmp into nam, sall, qualif, qId;
        IF curEmp%NOTFOUND THEN
            RAISE err02;
        END IF;

        fetch curEmp into nam1, sall1, qualif1, qId1;
        IF curEmp%FOUND THEN
            RAISE err01;
        ELSE
            DBMS_OUTPUT.PUT_LINE(nam||' '||sall||' '||qualif||' '||qId);
        END IF;

    ELSE
        DBMS_OUTPUT.PUT_LINE('unable to open cursor');
    END IF;
    close curEmp;

    EXCEPTION
    WHEN err02 then
        DBMS_OUTPUT.PUT_LINE('No employees with such qualification');

    when err01 then
        DBMS_OUTPUT.PUT_LINE('More than one employee with such qualification');

END;
```

Output

```

43  when err01 then
44      DBMS_OUTPUT.PUT_LINE('More than one employee with such qualification');
45
46  END;
47  /
Enter value for qid: 1
old 19:  qId := &qId;
new 19:  qId := 1;
emp 5600002 B.S. 1

PL/SQL procedure successfully completed.
SQL>

```



```

Version 21.3.0.0.0
SQL> select * from employee;

-----
ID NAME                SALARY QUALIF                QUALIF_ID
-----
1 emp                  2012344 B.Tech.                23
11 emp                  50000 M.Tech.                 64
2 emp                  560000 Phd.                   13
12 emp                  5600002 B.S.                   1
123121 emp             23434 Dr.                     43
1123 emp                3444 Dr.                      43
1231 emp                33344 Phd.                    13
31 emp                  56744 B.Tech.                 23
3 emp                   345 Gs                       3

9 rows selected.
SQL>

```



```

43  when err01 then
44      DBMS_OUTPUT.PUT_LINE('More than one employee with such qu
45
46  END;
47  /
Enter value for qid: 43
old 19:  qId := &qId;
new 19:  qId := 43;
More than one employee with such qualification

PL/SQL procedure successfully completed.
SQL>

```



```

37  close curEmp;
38
39  EXCEPTION
40  WHEN err02 then
41      DBMS_OUTPUT.PUT_LINE('No employees with such qualificati
42
43  when err01 then
44      DBMS_OUTPUT.PUT_LINE('More than one employee with such qu
45
46  END;
47  /
Enter value for qid: 2
old 19:  qId := &qId;
new 19:  qId := 2;
No employees with such qualification

PL/SQL procedure successfully completed.
SQL>

```

Question 4

Write a procedure that is passed a students identification number and returns back the students full name and phone number from the STUDENT table to the calling program. Also write an anonymous block with the procedure call.

Solution

```

SET SERVEROUTPUT ON;
create or replace function ss(A CHAR)
RETURN VARCHAR is
    NAM VARCHAR(40);
    PH CHAR(10);
    re VARCHAR(80);
BEGIN
    SELECT FIRST||' '||LAST, PHONE INTO NAM, PH FROM STUDENT
        WHERE STUDENT.STUDENTID=A;
    re := NAM||' : '||PH;
    return re;
END ss;

DECLARE
    cc VARCHAR(80);
    xy student.STUDENTID%TYPE;
BEGIN
    xy := &xy;
    cc := ss(xy);
    dbms_output.put_line('Ans '||cc);
END;

```

Output

```

6    cc := ss(xy);
7    dbms_output.put_line('Ans ' || cc);
8  END;
9  /
Enter value for xy: '00100'
old   5:   xy := &xy;
new   5:   xy := '00100';
Ans Jose Diaz : 9735551111

PL/SQL procedure successfully completed.

SQL> set lines 150;
SQL> select * from student;

```

STUDE ATE	LAST FACULTYID	FIRST MAJORID	PHONE	STREET	CIT
00100	Diaz	Jose	123 100 9735551111	1 Ford Avenue #7	Hil
-83					
00101	Tyler	Mickey	555 500 7185552222	12 Morris Avenue	Bro
-84					

Question 5

Write a function and pass a department number to it. If the DEPT table does not contain that department number, return a FALSE value, otherwise return a TRUE value. Print the appropriate message in the calling program based on the result.

Solution

```

SET SERVEROUTPUT ON;
create or replace function ss1(A number)
RETURN number is
  dd NUMBER;
  xyz BOOLEAN;
BEGIN
  select count(*) into dd from department where A=DEPTID;
  if dd = 0 then
    xyz := false;
  else
    xyz := true;
  end if;
  return dd;
END ss1;

DECLARE
  c number;
  xx number;

```



```

BEGIN
  xx := &xx;
  c := ss1(xx);
  if c = 1 then
    dbms_output.put_line('TRUE');
  ELSE
    dbms_output.put_line('False');
  end if;
END;

```

Output

```

12 END;
13 /
Enter value for xx: 2
old 5:  xx := &xx;
new 5:  xx := 2;
TRUE
PL/SQL procedure successfully completed.

SQL> DECLARE
2  c number;
3  xx number;
4  BEGIN
5  xx := &xx;
6  c := ss1(xx);
7  if c = 1 then
8  dbms_output.put_line('TRUE');
9  ELSE
10 dbms_output.put_line('False');
11 end if;
12 END;
13 /
Enter value for xx: 7
old 5:  xx := &xx;
new 5:  xx := 7;
False
PL/SQL procedure successfully completed.

SQL>

```

Connected to:
Oracle Database 21c Express Edition Release 21.0
Version 21.3.0.0.0

```

SQL>
SQL> select * from department
2 ;

```

DEPTID	DEPTNAME	FACULTYID
1	Computer Science	111
2	Telecommunications	222
3	Accounting	333
4	Math and Science	444
5	Liberal Arts	555

```

SQL>

```

Question 6

Write a package that contains a procedure and a function. The procedure is passed a room number. If the room number exists, the procedure gets the capacity of the room and the building name from the LOCATION table. If the room number does not exist, the procedure performs the appropriate exception-handling routine. The function is passed a csid and returns maximum number of seats available in the course section.

Solution

```

set SERVEROUTPUT on;

CREATE OR REPLACE PACKAGE q6 AS
  procedure rooom(A location.ROOMNO%TYPE);
  FUNCTION crssss(A CRSSECTION.CSID%TYPE) return NUMBER;
end q6;

CREATE OR REPLACE PACKAGE BODY q6 AS

  procedure rooom(A location.ROOMNO%TYPE)
  is

```



```

dd NUMBER;
errr EXCEPTION;
cappp LOCATION.CAPACITY%type;
bbb LOCATION.BUILDING%type;

begin
  select count(*) into dd from location where LOCATION.ROOMNO=A;
  if dd = 0 THEN
    raise errr;
  end if;
  select CAPACITY, BUILDING into cappp, bbb from location where
LOCATION.ROOMNO=A;
  dbms_output.PUT_LINE(cappp||' '||bbb);
EXCEPTION
  when errr then
    DBMS_OUTPUT.PUT_LINE('no entries wrt given room number');
end room;

FUNCTION crssss(A CRSSECTION.CSID%TYPE)
RETURN NUMBER is
maxC number;
BEGIN
  select MAXCOUNT into maxC from CRSSECTION where CSID=A;
  return maxC;
end crssss;
END q6;

```

Output

The screenshot shows a Windows PowerShell window with a PL/SQL script being executed. The script defines a function `crssss` to return the maximum count for a given course ID (CSID) from the `CRSSECTION` table. It also includes a `room` procedure that checks if a room number exists in the `location` table and outputs the capacity and building name.

The execution shows the following steps:

- Declaration of variables `c`, `xx1`, and `xx2`.
- Assignment of `xx1` to 101 and `xx2` to 1102.
- Execution of the `room` procedure for room 101, which outputs "max count 40".
- Execution of the `crssss` function for CSID 1102, which returns 40.
- Execution of the SQL query `select * from location;`, which returns 11 rows of data.
- Execution of the SQL query `select * from crssection;`, which returns 10 rows of data.

The SQL query results are displayed as follows:

ROOMID	BUILDIN	ROO	CAPACITY	R
11	Gandhi	101	2	O
12	Gandhi	103	2	O
13	Kennedy	202	35	L
14	Kennedy	204	50	L
15	Nehru	301	50	C
16	Nehru	309	45	C
17	Gandhi	105	2	O
18	Kennedy	206	40	L
19	Kennedy	210	30	L
20	Gandhi	107	2	O
21	Gandhi	109	2	O

CSID	COURSE	SE	TERM	FACULTYID	DA	START	ENDTI	ROOMID	MAXCOUNT
1101	CIS265	01	WN03	111	MW	09:00	10:30	13	30
1102	CIS253	01	WN03	123	TR	09:00	10:30	18	40
1103	MA150	02	WN03	444	F	09:00	12:00	15	25
1104	AC101	10	WN03	345	MW	10:30	12:00	16	35
1205	CIS265	01	SP03		MW	09:00	10:30	14	35
1206	CIS265	02	SP03	111	TR	09:00	10:30	18	30
1207	LA123	05	SP03		MW	09:00	10:30	15	30
1208	CIS253	21	SP03	123	TR	09:00	10:30	14	40
1209	CIS253	11	SP03	111	MW	09:00	10:30	18	40
1210	CIS253	31	SP03	123	F	TBA	TBA	19	2