

Assignment 4

DBMS LAB

IT552

Aditya Badayalya

510819056

Fifth Semester

Information Technology (HY)

**Q1. Write a cursor program to delete first two rows of the employee table getting the highest salary**

```
create table emp(  
    eno varchar2(4),  
    name varchar2(10),  
    dno varchar2(4),  
    sal number(8,2),  
    loc varchar2(10)  
);
```

```
insert into emp values('E01', 'STEVE', 'D01', 8000,  
'Kolkata');  
insert into emp values('E02', 'TONY', 'D03', 5000,  
'Kolkata');  
insert into emp values('E03', 'CLINT', 'D02', 6000,  
'Mumbai');  
insert into emp values('E04', 'CLARK', 'D02', 3000,  
'Kolkata');  
insert into emp values('E05', 'PETER', 'D03', 7500,  
'Delhi');  
insert into emp values('E06', 'STEPHEN', 'D01', 4000,  
'Chennai');
```

ENO	NAME	DNO	SAL	LOC
E01	STEVE	D01	8000	Kolkata
E02	TONY	D03	5000	Kolkata
E03	CLINT	D02	6000	Mumbai
E04	CLARK	D02	3000	Kolkata
E05	PETER	D03	7500	Delhi
E06	STEPHEN	D01	4000	Chennai

```
declare  
    cursor c is  
        select eno from emp  
        order by sal desc;  
    enum emp.eno % type;  
    i integer;  
begin  
    open c;  
    for i in 1 .. 2 loop
```

```

        fetch c into enum;
        delete from emp
        where eno = enum;
    end loop;
    if c %ISOPEN then
        close c;
    end if;
end;
```

ENO	NAME	DNO	SAL	LOC
E02	TONY	D03	5000	Kolkata
E03	CLINT	D02	6000	Mumbai
E04	CLARK	D02	3000	Kolkata
E06	STEPHEN	D01	4000	Chennai

[Download CSV](#)

**Q2. Consider a employee table. Write a cursor program to raise the salary of employees by 15% who are in 'Kolkata' and display how many rows are affected in the employee table**

ENO	NAME	DNO	SAL	LOC
E02	TONY	D03	5000	Kolkata
E03	CLINT	D02	6000	Mumbai
E04	CLARK	D02	3000	Kolkata
E06	STEPHEN	D01	4000	Chennai

[Download CSV](#)

```

declare
    row_temp emp % rowtype;
    cursor c is
        select eno, name, dno, sal, loc
        from emp
        where loc = 'Kolkata';
```

```

    cnt number :=0;
    val number;
begin
    open c;
    loop
        fetch c into row_temp;
        exit when c % NOTFOUND;
        val := row_temp.sal;
        val := val*1.15;
        update emp
        set sal = val
        where eno = row_temp.eno;
        cnt := cnt+1;
    end loop;
    if cnt > 0 then
        dbms_output.put_line('The number of affected rows
is '|| cnt);
    else
        dbms_output.put_line('No employee from Kolkata
found');
    end if;
    if c % ISOPEN then
        close c;
    end if;
end;

```

Statement processed.  
The number of affected rows is 2

ENO	NAME	DNO	SAL	LOC
E02	TONY	D03	5750	Kolkata
E03	CLINT	D02	6000	Mumbai
E04	CLARK	D02	3450	Kolkata
E06	STEPHEN	D01	4000	Chennai

Download CSV

Q3. Accept two account numbers from the terminal. Write a PL/SQL procedure to do the following

- a) Transfer RS 2000 from one account to other account.  
If there is no sufficient balance then show appropriate message.

```
create table account (  
    act_no varchar2(8),  
    age      numeric(3),  
    amount numeric(8, 2)  
);
```

```
insert into account values('ACT01', 57, 5000);  
insert into account values('ACT02', 62, 2000);  
insert into account values('ACT03', 65, 8000);  
insert into account values('ACT04', 49, 4500);  
insert into account values('ACT05', 61, 3700);  
insert into account values('ACT06', 68, 9100);
```

ACT_NO	AGE	AMOUNT
ACT01	57	5000
ACT02	62	2000
ACT03	65	8000
ACT04	49	4500
ACT05	61	3700
ACT06	68	9100

act = ACT01, act2 = ACT04

```
declare  
    act1 account.act_no % type := '&act1';  
    act2 account.act_no % type := '&act2';  
    cursor c is  
        select act_no  
        from account;  
    ano account.act_no % type;  
    amt account.amount % type;  
begin  
    open c;
```

```

loop
    fetch c into ano;
    exit when c % NOTFOUND;
    if ano = act1 then
        select amount into amt from account
        where act_no = act1;
        if amt > 1999 then
            update account
            set amount = amount - 2000
            where act_no = act1;
            update account
            set amount = amount + 2000
            where act_no = act2;
            dbms_output.put_line('Transfer
Complete');
        else
            dbms_output.put_line('Insufficient
balance');
        end if;
    end if;
end loop;
if c % ISOPEN then
    close c;
end if;
end;

```

Statement processed.  
Transfer Complete

ACT_NO	AGE	AMOUNT
ACT01	57	3000
ACT02	62	2000
ACT03	65	8000
ACT04	49	6500
ACT05	61	3700
ACT06	68	9100

b) Write a cursor program to raise 5% of the account balance of the first three customers who are older than 60

ACT_NO	AGE	AMOUNT
ACT02	62	2000
ACT03	65	8000
ACT05	61	3700
ACT06	68	9100

[Download CSV](#)

```
declare
  cursor c is
    select act_no from account
    where age > 60;
  ano account.act_no % type;
```

```
begin
  open c;
  for i in 1 .. 3 loop
    fetch c into ano;
    update account
    set amount = amount * 1.05
    where act_no = ano;
  end loop;
  if c%ISOPEN then
    close c;
  end if;
end;
```

ACT_NO	AGE	AMOUNT
ACT02	62	2100
ACT03	65	8400
ACT05	61	3885
ACT06	68	9100

[Download CSV](#)

**Q4. Write a PL/SQL block of code that first inserts a record in an 'emp' table. Update the salary by RS 2000. Then see if the total salary of the employees does not exceed RS 20000. If so then undo the updates made to the salary**

ENO	NAME	DNO	SAL	LOC
E02	TONY	D03	5750	Kolkata
E03	CLINT	D02	6000	Mumbai
E04	CLARK	D02	3450	Kolkata
E06	STEPHEN	D01	4000	Chennai

```

declare
    sal_temp emp.sal % type;
    cnt number;
begin
    insert into emp
    values('E07','SAM','D01',4000,'Mumbai');
    update emp
    set sal = sal + 2000
    where eno = 'E07';
    select sum(sal) into sal_temp from emp;
    if sal_temp > 20000 then
        update emp
        set sal = sal - 2000
        where eno = 'E07';
    end if;
end;
```

ENO	NAME	DNO	SAL	LOC
E02	TONY	D03	5750	Kolkata
E03	CLINT	D02	6000	Mumbai
E04	CLARK	D02	3450	Kolkata
E06	STEPHEN	D01	4000	Chennai
E07	SAM	D01	4000	Mumbai



Q5. Consider the emp table (eno,sal,dno,add). Find salary of the employee for a given department number. If the department number is not included in the table or a number of rows for given dno then handle through exception handler.

ENO	NAME	DNO	SAL	LOC
E02	TONY	D03	5750	Kolkata
E03	CLINT	D02	6000	Mumbai
E04	CLARK	D02	3450	Kolkata
E06	STEPHEN	D01	4000	Chennai
E07	SAM	D01	4000	Mumbai

[Download CSV](#)

```
declare
    cursor c is
        select * from emp;
    emp_temp emp%rowtype;
    flag number :=0 ;
    dno_temp emp.dno % type := '&dno_temp';
    invalid_dept_number exception;
begin
    open c;
    loop
        fetch c into emp_temp;
        exit when c%NOTFOUND;
        if dno_temp = emp_temp.dno then
            flag := 1;
            dbms_output.put_line('The salary of employee '
|| emp_temp.eno || ' is ' || emp_temp.sal);
        end if;
    end loop;
    if flag = 0 then
        raise invalid_dept_number;
    end if;
    exception
        when invalid_dept_number then
            dbms_output.put_line('Invalid department no');
end;
```

```
dno_temp = 'D10'
```

```
Statement processed.  
Invalid department no
```

```
dno_temp = 'D01'
```

```
Statement processed.  
The salary of employee E06 is 4000  
The salary of employee E07 is 4000
```

**Q6. Consider the table product (pno, pname, actual\_price, sale\_price). For a give pno, profit can be calculated from sale\_price and actual\_price. Find out the profit for given pno. If the pno is not included in the table or the profit is zero (handle by user defined exception) then handle the error**

```
create table product (  
    pno                varchar2(4),  
    pname              varchar2(16),  
    actual_price        number(8, 2),  
    sale_price          number(8, 2)  
);
```

```
insert into product values('P01', 'ABC', 100, 120);  
insert into product values('P02', 'BCD', 120, 130);  
insert into product values('P03', 'CDE', 100, 90);  
insert into product values('P04', 'DEF', 100, 100);  
insert into product values('P05', 'EFG', 130, 220);
```

PNO	PNAME	ACTUAL_PRICE	SALE_PRICE
P01	ABC	100	120
P02	BCD	120	130
P03	CDE	100	90
P04	DEF	100	100
P05	EFG	130	220

```

declare
    cursor c is select * from product;
    row_temp product % rowtype;
    pnum product.pno % type := '&pnum';
    flag integer := 0;
    profit number(4, 2);
    invalid_product_no exception;
    profit_zero exception;
begin
    open c;

    loop
        fetch c into row_temp;
        exit when c % NOTFOUND;

        if row_temp.pno = pnum then
            flag := 1;
            profit := row_temp.sale_price -
row_temp.actual_price;

            if profit > 0 then
                dbms_output.put_line('Profit on product
' || pnum || ' is ' || profit);
            end if;

            if profit <= 0 then
                raise profit_zero;
            end if;
        end if;
    end loop;

    if flag = 0 then
        raise invalid_product_no;
    end if;
end;

```

```

        end if;
        if c % ISOPEN then
            close c;
        end if;
    exception
        when invalid_product_no then
            dbms_output.put_line('Invalid product number');
        when profit_zero then
            dbms_output.put_line('Zero profit');
    end;

```

pnum = 'P01'

```

Statement processed.
Profit on product P01 is 20

```

pnum = 'P04'

```

Statement processed.
Zero profit

```

pnum = 'P83'

```

Statement processed.
Invalid product number

```