*Exercise:*

*Date: 13.10.2024*

*Aim:*

CREATE TABLE employee (

id NUMBER PRIMARY KEY,

name VARCHAR2(100) NOT NULL,

salary NUMBER(10, 2),

department VARCHAR2(50)

)

INSERT INTO employee (id, name, salary, department)

VALUES (1, 'John Doe', 50000, 'HR')

INSERT INTO employee (id, name, salary, department)

VALUES (2, 'Jane Smith', 60000, 'Finance')

INSERT INTO employee (id, name, salary, department)

VALUES (3, 'Alice Johnson', 55000, 'Engineering')

INSERT INTO employee (id, name, salary, department)

VALUES (4, 'Bob Brown', 70000, 'Marketing')

INSERT INTO employee (id, name, salary, department)

VALUES (5, 'Charlie White', 48000, 'Sales')

*-------------------------------------------Output:--------------------------------------------*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** |  |  |  |  | **NAME** |  |  |  | **SALARY** | **DEPARTMENT** |
| 1 |  |  |  |  | John Doe |  |  |  | 50000 | HR |
| 2 |  |  |  |  | Jane Smith |  |  |  | 60000 | Finance |
| 3 |  |  |  |  | Alice Johnson |  |  |  | 55000 | Engineering |
| 4 |  |  |  |  | Bob Brown |  |  |  | 70000 | Marketing |
| 5 |  |  |  |  | Charlie White |  |  |  | 48000 | Sales |
|  |  |  |  |  |  |  |  |  |  |  |

create or replace trigger trg\_before\_insert

before insert on employee

for each row

begin

if: NEW.salary is null then

: NEW.salary:=3000;

end if;

end;

*-------------------------------------------Output:--------------------------------------------*

*Trigger created*

*Exercise:*

*Date: 13.10.2024*

*Aim:*

*-----------------------------------------STORED PROCEDURE----------------------------------------*

create or replace function add\_numbers(num1 in number, num2 in number)

return number is

begin

return num1 + num2;

end add\_numbers;

create or replace function subtract\_numbers(num1 in number, num2 in number)

return number is

begin

return num1 - num2;

end subtract\_numbers;

create or replace function multiply\_numbers(num1 in number, num2 in number)

return number is

begin

return num1\*num2;

end multiply\_numbers;

create or replace function divide\_numbers(num1 in number, num2 in number)

return number is

begin

if num2=0 then

return null;

else

return num1/num2;

end if;

end divide\_numbers;

*---------------------------------------------LOCAL PROCEDURE--------------------------------------*

CREATE OR replace procedure perform\_operations(

num1 in number,

num2 in number

) is

function local\_add(n1 in number,n2 in number)

return number is

begin

return n1 + n2;

end local\_add;

function local\_subtract(n1 in number,n2 in number)

return number is

begin

return n1 - n2;

end local\_subtract;

function local\_multiply(n1 in number,n2 in number)

return number is

begin

return n1 \* n2;

end local\_multiply;

function local\_divide(n1 in number,n2 in number)

return number is

begin

if n2 = 0 then

return NULL;

else

return n1 / n2;

end if;

end local\_divide;

-----------------------------------*CALLING FUNCTIONS------------------------------------------*

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Addition(local):' || local\_add(num1,num2));

DBMS\_OUTPUT.PUT\_LINE('Subtraction(local):' || local\_subtract(num1,num2));

DBMS\_OUTPUT.PUT\_LINE('Multiplication(local):' || local\_multiply(num1,num2));

DBMS\_OUTPUT.PUT\_LINE('Division(local):' || local\_divide(num1,num2));

DBMS\_OUTPUT.PUT\_LINE('Addition(stored):' || add\_numbers(num1,num2));

DBMS\_OUTPUT.PUT\_LINE('Subtraction(stored):' || subtract\_numbers(num1,num2));

DBMS\_OUTPUT.PUT\_LINE('Multiplication(stored):' || multiply\_numbers(num1,num2));

DBMS\_OUTPUT.PUT\_LINE('Division(stored):' || divide\_numbers(num1,num2));

END perform\_operations;

*-------------------------------------------Output:-----------------------------------------*

*PROCEDURE CREATED*

exec perform\_operations(10,5)

*-------------------------------------------Output:-----------------------------------------*

Statement processed.  
Addition(local):15  
Subtraction(local):5  
Multiplication(local):50  
Division(local):2  
Addition(stored):15  
Subtraction(stored):5  
Multiplication(stored):50  
Division(stored):2