**Sheet-05**

**Name : Raksha Pahariya**

**Roll No : 2021CSB029**

**Subject : DBMS Lab**

**1.Insert data into a table containing two attributes namely radius and circumference of circles. You may get different values of radius either from keyboard or you may generate different values.**

**Query:**

Create table circle(

Radius number,

Circumference number

);

--insert data into the circle table

Declare

C\_radius number;

C\_circumference number;

Begin

--insert data into the circle table with a loop or generate values as –needed

For i IN 1..10 LOOP

--You can get the radius from the keyboard input or generate different –values as needed

--For simplicity, we generate random values here

C\_radius := ROUND(DBMS\_RANDOM.VALUE(1,20),2);

--Calculate the circumference

C\_circumference :=2\*3.142\*c\_radius;

Insert into circle(Radius,circumference)

Values(c\_radius,c\_circumference);

End LOOP;

Commit;

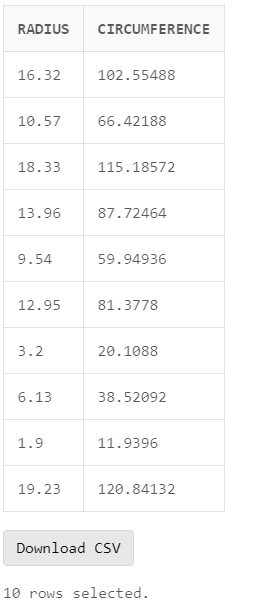
End;

/

Select \* from circle

/

**Output:**

****

**2.Update the balance of each customer from a cust\_acct table showing withdrawal of Rs.1000/-as service charge provided that the customer balance shows at least Rs.1000/-.**

**Query:**

Create table customer(

Customer\_id number primary key,

Customer\_name varchar(100),

Balance number

);

-- Insert sample data into the customer table

Insert into customer(Customer\_id, customer\_name, balance)

Values(1, 'Rohit', 1900);

Insert into customer(customer\_id, customer\_name, balance)

Values(2, 'Gill', 1400);

Insert into customer(Customer\_id, customer\_name, balance)

Values(3, 'Virat', 600);

Insert into customer(customer\_id, customer\_name, balance)

Values(4, 'KL Rahul', 3000);

Insert into customer(Customer\_id, customer\_name, balance)

Values(5, 'Surya', 1000);

Insert into customer(customer\_id, customer\_name, balance)

Values(6, 'Hardik', 999);

-- Update the balance of each customer from the customer table with a charge

Declare

Service\_charge number := 1000;

Begin

Update customer

Set balance = balance - service\_charge

Where balance >= 1000;

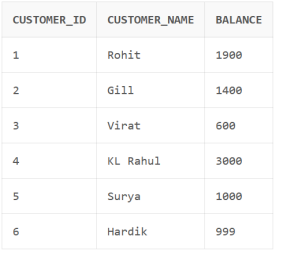
Commit;

End;

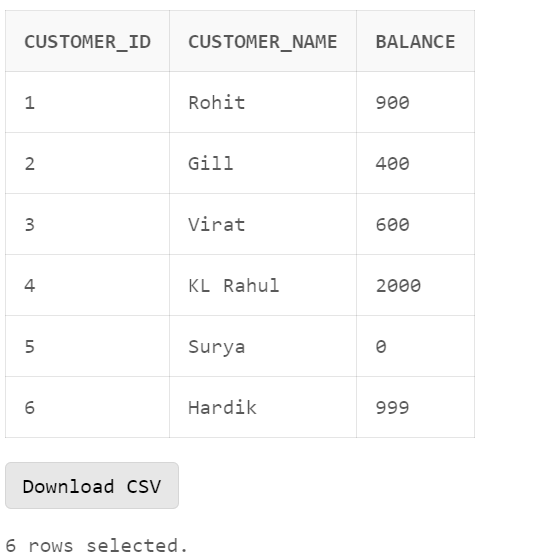
/

Select \* from customer;

**Input:**

****

**Output:**

****

**3. Update the salary of each employee from EMP table by 15% using cursor.**

**Query:**

Create table emp(

Emp\_id number primary key,

Emp\_name varchar(100),

Salary number

);

-- Insert sample employee data into the emp table

Insert into emp(emp\_id, emp\_name, salary)

Values(901, 'Rohit', 900000);

Insert into emp(emp\_id, emp\_name, salary)

Values(902, 'Gill', 500000);

Insert into emp(emp\_id, emp\_name, salary)

Values(903, 'Virat', 1000000);

Insert into emp(emp\_id, emp\_name, salary)

Values(904, 'Rahul', 800000);

Select \* from emp;

-- Update the salary of each employee from the emp table by 15% using a cursor

Declare

cursor emp\_cursor is

Select emp\_id, salary

From emp;

E\_emp\_id emp.emp\_id%TYPE;

E\_salary emp.salary%type;

Begin

For rec IN emp\_cursor LOOP

E\_emp\_id := rec.emp\_id;

E\_salary := rec.salary;

-- Update salary by 15%

E\_salary := E\_salary \* 1.15;

Update emp

Set salary = E\_salary

Where emp\_id = E\_emp\_id;

End LOOP;

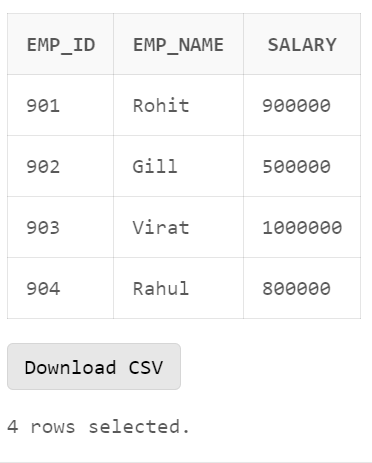
Commit;

End;

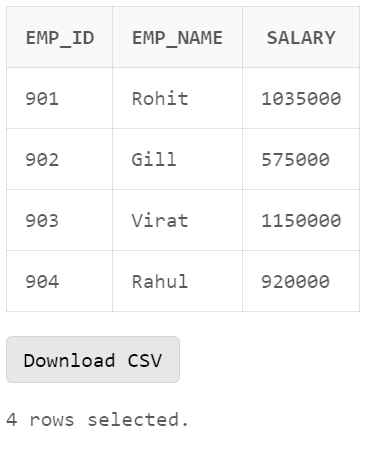
/

Select \* from Emp;

**Input:**



**Output:**

****

**4. Update the balance in the ITEM\_MSTR table each time a transaction takes place in the ITEM\_TR table. If the item\_id is already present in the ITEM\_MSTR table an update is performed to decrease the balance by the quantity specified in the ITEM\_TR table.**

**IF the item\_id is not present in the ITEM\_MSTR table, the tuple is to be inserted.**

**Query:**

Create table item\_mstr(

Item\_id number primary key,

Balance number

);

--create the item\_tr table to track transactions

Create table item\_tr(

Transaction\_id number primary key,

Item\_id number,

Quantity number

);

--insert sample data into the item\_mstr table

Insert into item\_mstr(item\_id,balance)

Values(1,500);

Insert into item\_mstr(item\_id,balance)

Values(3,400);

--insert sample transaction data into the item\_tr table

Insert into item\_tr(transaction\_id,item\_id,quantity)

Values(1,1,10);

Insert into item\_tr(transaction\_id,item\_id,quantity)

Values(2,2,15);

Insert into item\_tr(transaction\_id,item\_id ,quantity)

Values(3,3,25);

Select \* from item\_mstr;

Select\* from item\_tr;

--update the balance in the item\_mstr table based on transactions in item\_tr

Declare

T\_item\_id item\_tr.item\_id%type;

T\_quantity item\_tr.quantity%type;

Begin

For rec in(select item\_id,quantity from item\_tr)LOOP

T\_item\_id := rec.item\_id;

T\_quantity := rec.quantity;

--check if t\_item\_id already exists in item\_mstr

BEGIN

Update item\_mstr

Set balance = balance- t\_quantity

Where item\_id = t\_item\_id;

If sql%notfound then

Raise no\_data\_found;

End if;

Exception

When no\_data\_found then

--insert a new tuple in item\_mstr if item\_id is not found

Insert into item\_mstr(item\_id,balance)

Values(t\_item\_id,t\_quantity);

End;

End LOOP;

Commit;

End;

/

Drop table item\_tr;

/

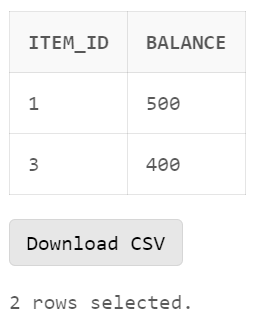
Select\* from item\_mstr;

Drop table item\_mstr;

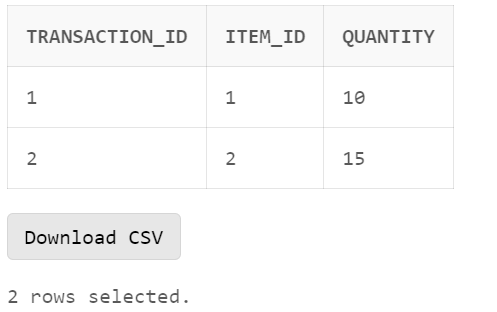
/

**INPUT:**

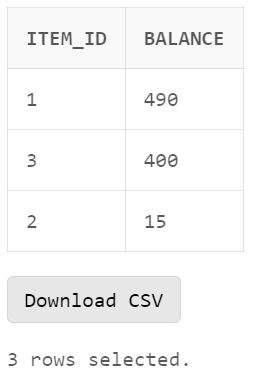
Item\_mstr:



Item\_Tr:



**Output:**

****

5. **Write a procedure for raising salary of some employee by some amount . The procedure to be written may carry two parameters emp\_id and amt to be raised.Include two exceptions which will be raised when either emp\_id is not present or salary is NULL.**

**Query:**

--Create the emp table for employee information

Create table emp(

Emp\_id number primary key,

Emp\_name varchar(100),

Salary number

);

--insert sample employee data into the emp table

Insert into emp(emp\_id,emp\_name,salary)

Values(901, ‘Rohit’,900000);

Insert into emp(emp\_id,emp\_name,salary)

Values(902, ‘Gill’,500000);

Insert into emp(emp\_id,emp\_name,salary)

Values(903, ‘Virat’ ,1000000);

Insert into emp(emp\_id,emp\_name,salary)

Values(904, ‘Rahul’,800000);

Insert into emp(emp\_id,emp\_name)

Values(905, ‘Shreyas’);

Select\* from emp;

--create a procedure to raise the salary of an employee by a specified amount

Create or replace procedure raise\_employee\_salary(

E\_id emp.emp\_id%TYPE,

Amt number

)is

E\_current\_salary Emp.salary%TYPE;

BEGIN

--check if the employee exists and get their current salary

Select salary into e\_current\_salary

From emp

Where emp\_id = e\_id;

--raise the salary by the specified amount

If e\_current\_salary is NULL then

Raise\_Application\_error(-20001, ‘Employee salary is NULL’);

Else

Update emp

Set salary = e\_current\_salary +amt

Where emp\_id =e\_id;

Commit;

End if;

Exception

When no\_data\_found then

Raise\_application\_error(-20002, ‘Employee ID is not Valid.’);

End raise\_employee\_salary;

/

BEGIN

Raise\_employee\_salary(901,100000);

End;

/

Select\*from emp;

/

--call where emp salary is currently NULL

BEGIN

Raise\_employee\_salary(905,50000);

End;

/

--call where emp\_id is not in table

Begin

Raise\_employee\_salary(907,20000);

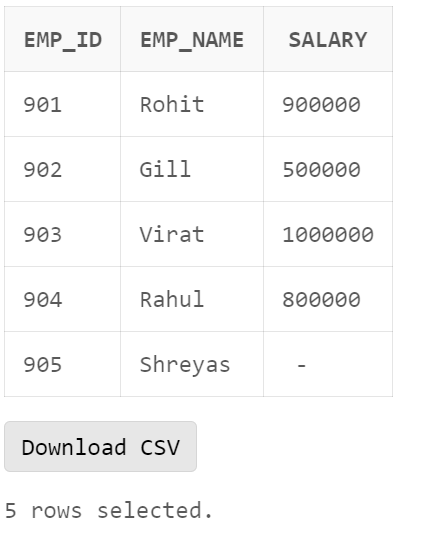
End;

/

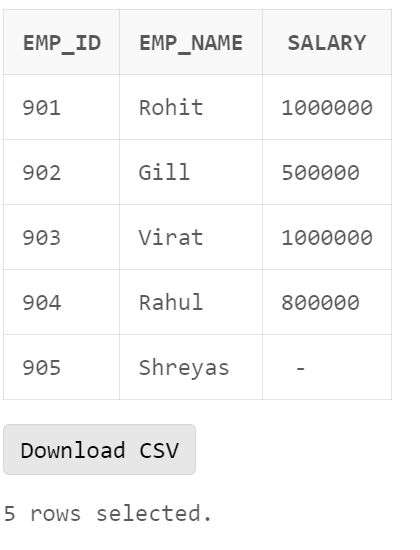
Drop table emp;

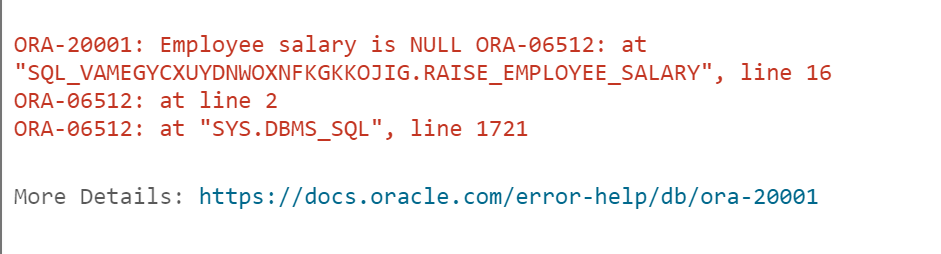
/

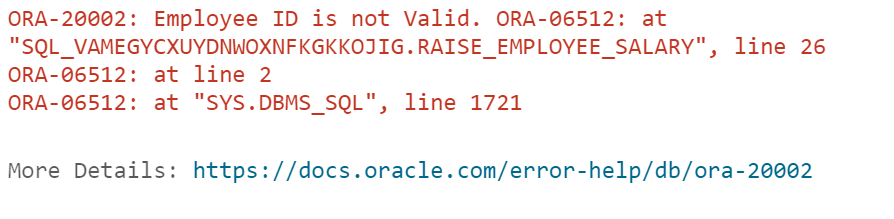
**INPUT:**

****

**Output:**

****

****

****