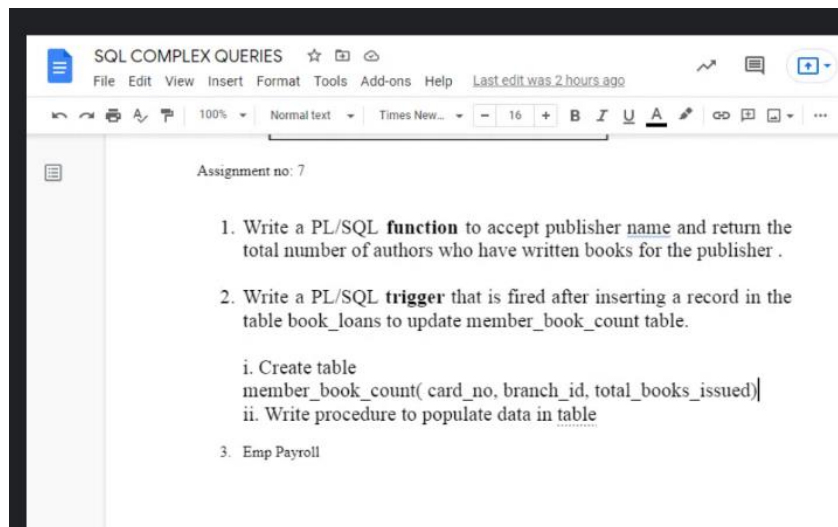


Name – SOUMYA BAIRAGYA

PL/SQL



1 - Function

```
SELECT COUNT(DISTINCT(AUTHOR_NAME))
FROM (BOOK NATURAL JOIN BOOK_AUTHORS)
WHERE PUBLISHER_NAME = 'HARRY';
```

```
SELECT * FROM BOOK NATURAL JOIN BOOK_AUTHORS;
```

```
SELECT * FROM BOOK;
SELECT * FROM BOOK_AUTHORS;
```

```
CREATE OR REPLACE FUNCTION TOTAL_AUTHORS(PNAME IN VARCHAR2)
RETURN NUMBER IS
CNT NUMBER := 0;
BEGIN
SELECT COUNT(DISTINCT(AUTHOR_NAME))
INTO CNT
FROM (BOOK NATURAL JOIN BOOK_AUTHORS)
WHERE PUBLISHER_NAME = PNAME;
RETURN CNT;
```

END;

SELECT TOTAL_AUTHORS('HARRY') FROM DUAL;

2 - Trigger

```
create table member_book_count( card_no number,
branch_id number,
total_book_issued number,
foreign key(branch_id)
references library_branch(branch_id),
foreign key(cardno)
references borrower(card_no)
);
```

```
create or replace procedure populate_data as
cursor c is select count(book_id) as total_books_issued,branch_id, card_no from book_loans where
due_date is null group by branch_id, card_no;
rec c%rowtype;
begin for rec in c loop dbms_output.put_line('books issued ' || rec.total_books_issued || ' cardno
' || rec.card_no || ' branch_id ' || rec.branch_id);
insert into member_book_count values (rec.card_no, rec.branch_id, rec.total_books_issued); end loop;
end;
```

```
begin
populate_data;
end;
create or replace trigger load_trigger
after insert on book_loans
begin
delete from member_book_count;
populate_data;
end;
```

```
books issued 1 cardno 132 branch_id 334
books issued 1 cardno 119 branch_id 673
books issued 1 cardno 119 branch_id 774
```

Statement processed.

0.01 seconds

3 - Procedure

Emp (ID, NAME , BASIC_SAL, DEPTNO, JOB)

EMP_LEAVES(ID, CL, ML, LWP, LIC)

CL = NO DEDUCTION (8)

1 ML = 30% from daily sal

EPF = 1800

DEDUCTION = ML +LWP +LIC+EPF

GROSS SAL= BASIC_SAL + DA +HRA+TA;

NET_SAL = GROSS SAL - DEDUCTION

DA = 121.7 %

HRA = 24.6 %

TA = 11.45 %

PAYROLL(ID, GROSS_SAL, NET_SAL, DEDUCTION)

CREATE TABLE EMPLOYEE(
ID NUMBER PRIMARY KEY,

```
NAME VARCHAR2(30) NOT NULL,  
BASIC_SALARY NUMBER NOT NULL,  
DEPARTMENT_NO NUMBER,  
JOB VARCHAR2(30) NOT NULL  
)
```

```
CREATE TABLE EMPLOYEE_LEAVES(  
ID NUMBER NOT NULL,  
CASUAL_LEAVES NUMBER,  
MEDICAL_LEAVES NUMBER,  
LEAVE_WITHOUT_PAY NUMBER,  
LIC NUMBER,  
  
FOREIGN KEY(ID) REFERENCES EMPLOYEE(ID)  
)
```

```
CREATE TABLE PAYROLL(  
ID NUMBER NOT NULL,  
GROSS_SALARY NUMBER,  
NET_SALARY NUMBER,  
DEDUCTION NUMBER,  
FOREIGN KEY(ID) REFERENCES EMPLOYEE(ID)  
)
```

```
CREATE OR REPLACE PROCEDURE CALCULATE_PAYROLL  
IS  
CURSOR c_emp IS  
SELECT * FROM (EMPLOYEE NATURAL JOIN EMPLOYEE_LEAVES);  
rec c_emp%ROWTYPE;  
DA INT;  
HRA INT;  
TA INT;  
Gross_sal INT;  
Deduction INT;  
casual_leave_ded INT;  
med_leave_ded INT;  
lwp_ded INT;  
daily_sal INT;  
net_sal INT;  
BEGIN  
FOR rec IN c_emp  
LOOP
```

```
daily_sal := (rec.BASIC_SALARY / 30);
if rec.CASUAL_LEAVES > 8 then
    casual_leave_ded := (rec.CASUAL_LEAVES - 8) * (daily_sal);
else
    casual_leave_ded := 0;
end if;
DA := rec.BASIC_SALARY * 1.217;
HRA := rec.BASIC_SALARY * 0.246;
TA := rec.BASIC_SALARY * 0.1145;

Gross_sal := DA + HRA + TA;

med_leave_ded := 0.3 * (daily_sal) * rec.MEDICAL_LEAVES;
lwp_ded := rec.LEAVE_WITHOUT_PAY * daily_sal;

Deduction := casual_leave_ded + med_leave_ded + lwp_ded + rec.LIC;

net_sal := Gross_sal - Deduction;

DBMS_OUTPUT.PUT_LINE('DA = ' || DA || ' HRA = ' || HRA || ' TA = ' || TA || ' GROSS SAL = ' ||
Gross_sal || ' Deductions = ' || Deduction || ' Net Salary = ' || net_sal);
INSERT INTO PAYROLL VALUES(rec.ID, rec.BASIC_SALARY, Gross_sal, net_sal, Deduction);
COMMIT;

END LOOP;
END;
```
