

## Assignment-9

### PL/sql-2

Name-Ashish Goyal

Id-2016ucp1100

Batch-A (1, 2)

---






#### EXPLICIT CURSOR

Employee(eno,ename,dno, salary,job)

Create Tables and Insert values into table:

```
Worksheet  Query Builder
CREATE TABLE EMPLOYEE
(
    eno int PRIMARY KEY,
    ename varchar(20),
    dno int,
    salary int,
    job varchar(20)
);
INSERT INTO employee
VALUES(1,'ASHISH',11,80000,'SOFTWARE DEVELOPER');
INSERT INTO employee
VALUES(2,'UTSAV',11,99000,'SOFTWARE TESTER');
INSERT INTO employee
VALUES(3,'ABHISHEK',11,80000,'SOFTWARE DEVELOPER');
INSERT INTO employee
VALUES(4,'VIJAY',11,90000,'ANALYST');
INSERT INTO employee
VALUES(5,'CHINMAY',11,95000,'SOFTWARE DESIGNER');
INSERT INTO employee
VALUES(6,'SWARAJ',11,75000,'SOFTWARE TESTER');
INSERT INTO employee
VALUES(7,'ANKIT',11,79000,'ANALYST');
```

## Show tables:

Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashback	...
    	Sort..	Filter:						
	ENO	ENAME	DNO	SALARY	JOB			
1	1	ASHISH	11	80000	SOFTWARE DEVELOPER			
2	2	UTSAV	11	99000	SOFTWARE TESTER			
3	3	ABHISHEK	11	80000	SOFTWARE DEVELOPER			
4	4	VIJAY	11	90000	ANALYST			
5	5	CHINMAY	11	95000	SOFTWARE DESIGNER			
6	6	SWARAJ	11	75000	SOFTWARE TESTER			
7	7	ANKIT	11	79000	ANALYST			

1-Display details of top five highest paid employees.

The screenshot shows an SQL IDE with the following tabs: Welcome Page, factorial.sql, 1.sql, QUE1, and A\_TABLE. The 'Worksheet' tab is active, displaying the following SQL code:

```

CREATE TABLE A_table
(
  eno int PRIMARY KEY,
  ename varchar(20),
  dno int,
  salary int,
  job varchar(20)
);

create or replace procedure quel
as var_rows number;
BEGIN
  declare
    CURSOR C1 IS SELECT eno,ename,dno,salary,job FROM Employee order by salary desc;
    eno employee.eno%type;
    ename employee.ename%type;
    dno employee.dno%type;
    salary employee.salary%type;
    job employee.job%type;
  BEGIN
    OPEN C1;
    LOOP
      FETCH C1 INTO
        eno,ename,dno,salary,job ;
      EXIT WHEN (C1%ROWCOUNT >5) or (C1%NOTFOUND);
      INSERT INTO A_table VALUES (eno,ename,dno,salary,job);
      COMMIT;
    END LOOP;
    CLOSE C1;
  END;
END;

```

## Solution Table-

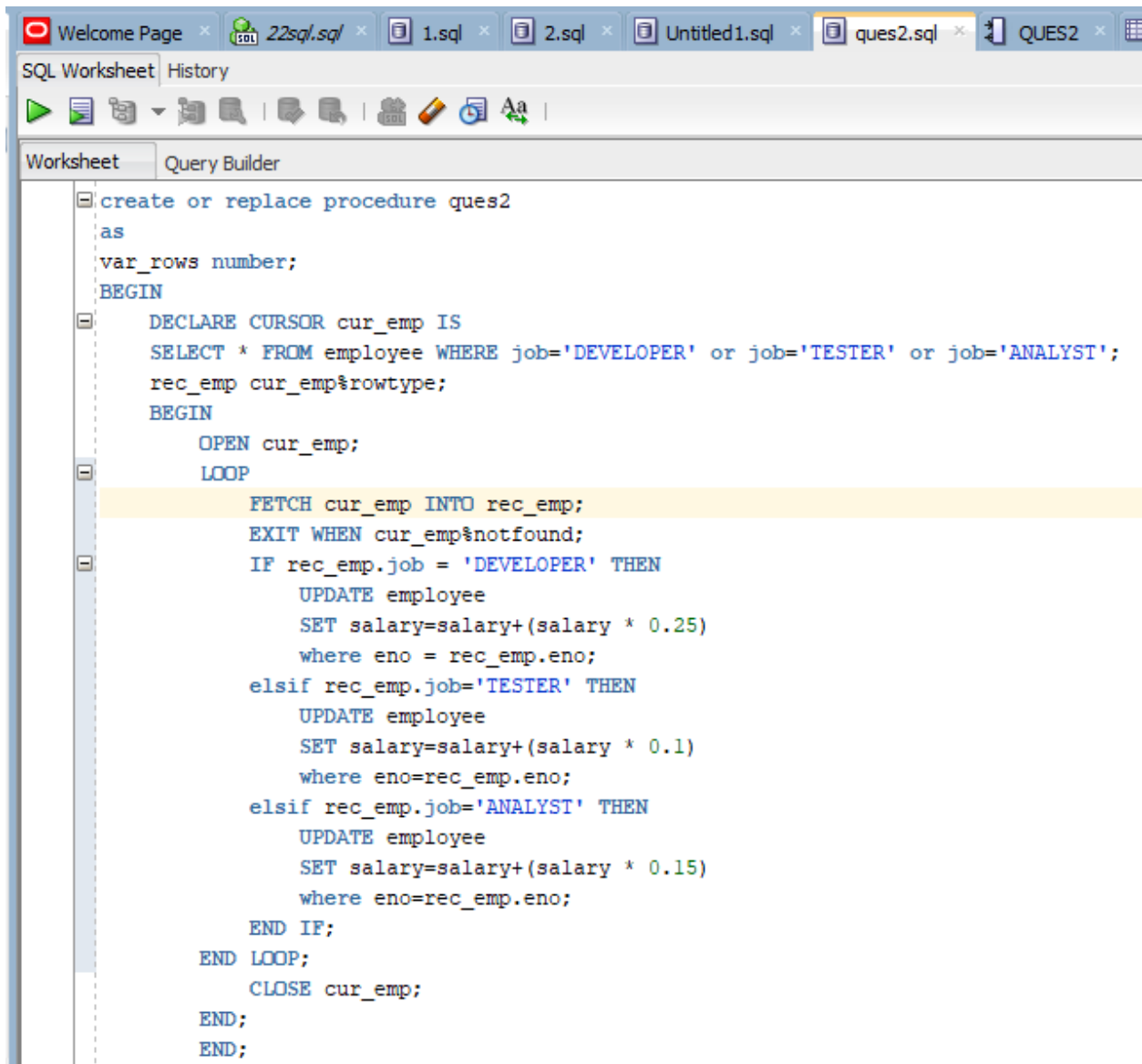
The screenshot shows a data grid with the following columns: ENO, ENAME, DNO, SALARY, and JOB. The data is as follows:

	ENO	ENAME	DNO	SALARY	JOB
1	2	UTSAV	11	99000	SOFTWARE TESTER
2	5	CHINMAY	11	95000	SOFTWARE DESIGNER
3	4	VIJAY	11	90000	ANALYST
4	1	ASHISH	11	80000	SOFTWARE DEVELOPER
5	3	ABHISHEK	11	80000	SOFTWARE DEVELOPER

**2-Increment salaries of the employees according to their job.**

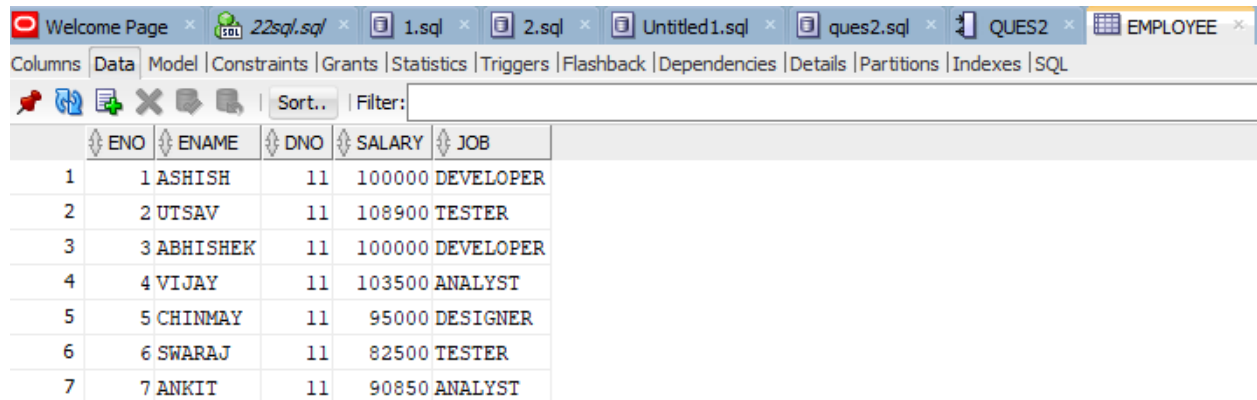
## Initial Values-

Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashback	...
Sort.. Filter:								
	ENO	ENAME	DNO	SALARY	JOB			
1	1	ASHISH	11	80000	SOFTWARE DEVELOPER			
2	2	UTSAV	11	99000	SOFTWARE TESTER			
3	3	ABHISHEK	11	80000	SOFTWARE DEVELOPER			
4	4	VIJAY	11	90000	ANALYST			
5	5	CHINMAY	11	95000	SOFTWARE DESIGNER			
6	6	SWARAJ	11	75000	SOFTWARE TESTER			
7	7	ANKIT	11	79000	ANALYST			



```
create or replace procedure ques2
as
var_rows number;
BEGIN
    DECLARE CURSOR cur_emp IS
    SELECT * FROM employee WHERE job='DEVELOPER' or job='TESTER' or job='ANALYST';
    rec_emp cur_emp%rowtype;
    BEGIN
        OPEN cur_emp;
        LOOP
            FETCH cur_emp INTO rec_emp;
            EXIT WHEN cur_emp%notfound;
            IF rec_emp.job = 'DEVELOPER' THEN
                UPDATE employee
                SET salary=salary+(salary * 0.25)
                where eno = rec_emp.eno;
            elsif rec_emp.job='TESTER' THEN
                UPDATE employee
                SET salary=salary+(salary * 0.1)
                where eno=rec_emp.eno;
            elsif rec_emp.job='ANALYST' THEN
                UPDATE employee
                SET salary=salary+(salary * 0.15)
                where eno=rec_emp.eno;
            END IF;
        END LOOP;
        CLOSE cur_emp;
    END;
END;
```

## Solution Table-



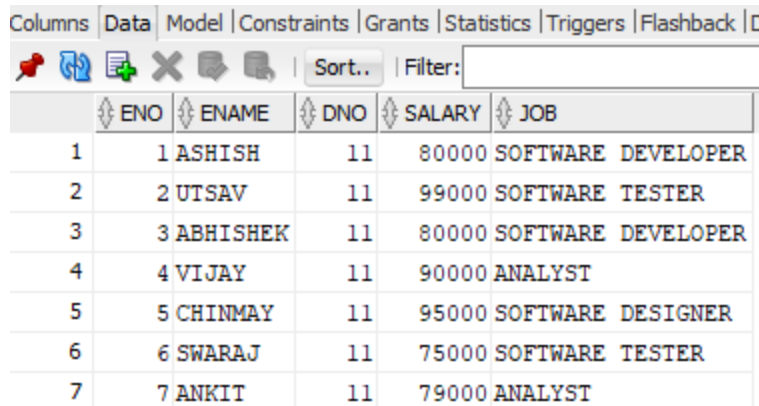
The screenshot shows the SQL Developer interface with the EMPLOYEE table selected. The table has columns: ENO, ENAME, DNO, SALARY, and JOB. The data is as follows:

	ENO	ENAME	DNO	SALARY	JOB
1	1	ASHISH	11	100000	DEVELOPER
2	2	UTSAV	11	108900	TESTER
3	3	ABHISHEK	11	100000	DEVELOPER
4	4	VIJAY	11	103500	ANALYST
5	5	CHINMAY	11	95000	DESIGNER
6	6	SWARAJ	11	82500	TESTER
7	7	ANKIT	11	90850	ANALYST

## TRIGGERS

3-create trigger which will get triggered on insert , update and delete in employee table.

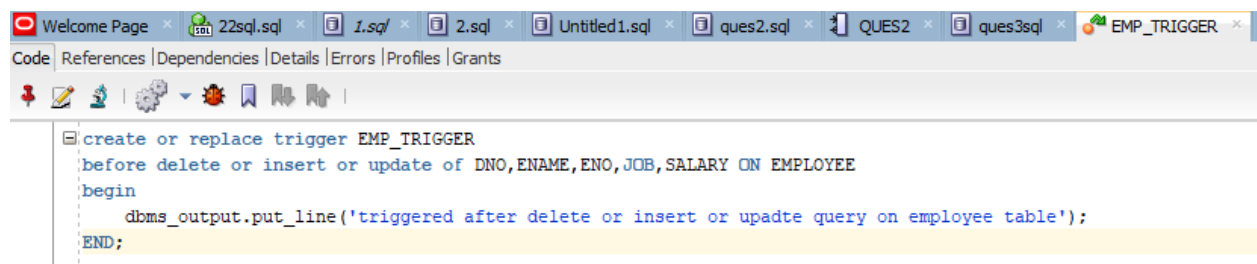
### Initial Values-



The screenshot shows the SQL Developer interface with the EMPLOYEE table selected. The table has columns: ENO, ENAME, DNO, SALARY, and JOB. The data is as follows:

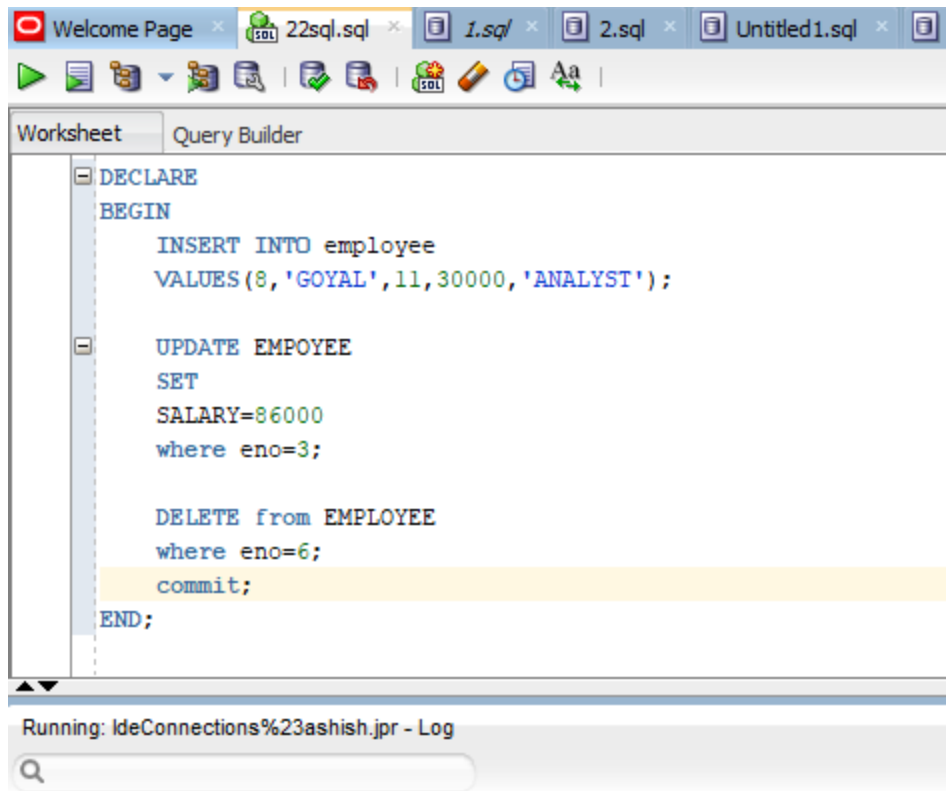
	ENO	ENAME	DNO	SALARY	JOB
1	1	ASHISH	11	80000	SOFTWARE DEVELOPER
2	2	UTSAV	11	99000	SOFTWARE TESTER
3	3	ABHISHEK	11	80000	SOFTWARE DEVELOPER
4	4	VIJAY	11	90000	ANALYST
5	5	CHINMAY	11	95000	SOFTWARE DESIGNER
6	6	SWARAJ	11	75000	SOFTWARE TESTER
7	7	ANKIT	11	79000	ANALYST

## Trigger-



The screenshot shows the SQL Developer interface with the EMP\_TRIGGER trigger created. The trigger is defined as follows:

```
create or replace trigger EMP_TRIGGER
before delete or insert or update of DNO,ENAME,ENO, JOB,SALARY ON EMPLOYEE
begin
    dbms_output.put_line('triggered after delete or insert or upadte query on employee table');
END;
```



Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashba
ENO	ENAME	DNO	SALARY	JOB			
1	ASHISH	11	80000	DEVELOPER			
2	UTSAV	11	99000	TESTER			
3	ABHISHEK	11	86000	DEVELOPER			
4	VIJAY	11	90000	ANALYST			
5	CHINMAY	11	95000	DESIGNER			
6	ANKIT	11	79000	ANALYST			
7	GOYAL	11	30000	ANALYST			

4-

Student(Sid,sname,deptname)

## Department(deptname)

### Create Tables and Insert values into table:

```
Worksheet  Query Builder
CREATE TABLE Department
(
    deptname varchar(20) PRIMARY KEY
);

INSERT INTO department
VALUES('CSE');
INSERT INTO department
VALUES('ECE');
INSERT INTO department
VALUES('EE');
```

```
Worksheet  Query Builder

CREATE TABLE Student
(
    Sid int PRIMARY KEY,
    sname varchar(20),
    deptname varchar(20),
    FOREIGN KEY(deptname) REFERENCES Department(deptname)
);

INSERT INTO student
VALUES(1092,'UTSAV','ECE');
INSERT INTO student
VALUES(1100,'ASHISH','CSE');
INSERT INTO student
VALUES(1103,'ABHISHEK','CSE');
INSERT INTO student
VALUES(1663,'SWARAJ','CSE');
INSERT INTO student
VALUES(1170,'ANKIT','EE');
INSERT INTO student
VALUES(1194,'CHINMAY','ECE');
```

### Show tables:

Welcome Page

Columns Data Model

DEPTNAME

1	CSE
2	ECE
3	EE

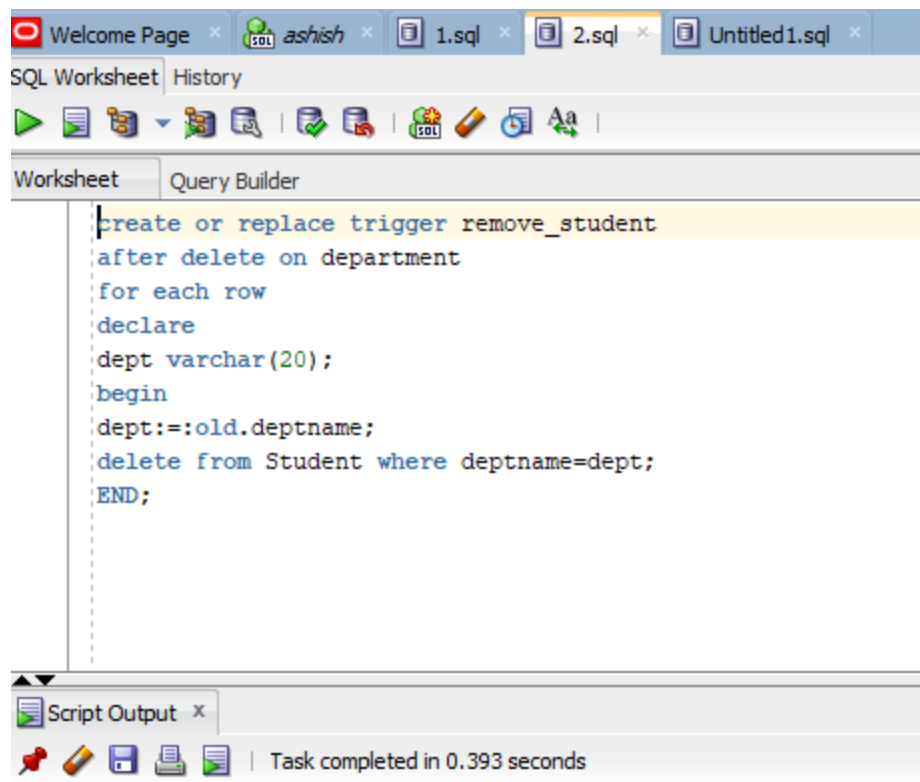
Columns Data Model Constraints Grants

Sort.. Filter

	SID	SNAME	DEPTNAME
1	1092	UTSAV	ECE
2	1100	ASHISH	CSE
3	1103	ABHISHEK	CSE
4	1663	SWARAJ	CSE
5	1170	ANKIT	EE
6	1194	CHINMAY	ECE

**Create a trigger to remove all students of department once department is deleted.**





Trigger REMOVE\_STUDENT compiled

