INDEX

NAME : **Ankit Pardeshi** STD. : **MCA** SEC. : **A** ROLL NO. **15**

7.1

Ans : Create or replace procedure salary(deptid number) as

```
    begin
        update emp set sal=sal+1000 where sal>5000 AND deptno=deptid;
    end;
```

7.2

Ans: create or replace procedure salary1(empid number) as

```
    begin
        update emp set sal=sal+sal*(0.1) where empno=empid;
    end;
```

7.3

Ans: create or replace procedure get_sal(dept number) as

```
    begin
        for s in (select * from emp where deptno = dept)
        loop
          dbms_output.put_line(s.sal);
        end loop;
     end;
```

7.4

Ans: create or replace procedure get_nature(dept number) as

```
    begin
      for s in (select * from emp where deptno = dept)
      loop
        dbms_output.put_line(s.job);
      end loop;
     end;
```

7.5

Ans: create or replace procedure dep_name(deptid number) as

```
    begin
      select dept.dname from dept,emp where emp.deptno=dept.deptno;
    end;
```

8.1

Ans: CREATE OR RELPLACE TRIGGER trig1 before insert on DEPT for each row DECLARE a number;

BEGIN

    if(:new.DEPTNO is Null) then

        raise_application_error(-20001,'error:: DEPTNO cannot be null');

    else

        select count(*) into a from DEPT where DEPTNO =:new.DEPTNO;

        if(a=1) then

            raise_application_error(-20002,'error:: cannot have duplicate DEPTNo ');

        end if;

    end if;

END;

8.2

Ans: CREATE [OR REPLACE] TRIGGER trig2 After delete on DEPT FOR EACH ROW

BEGIN

    DELETE FROM emp WHERE emp.deptno=:new.deptno;

END;

8.3

Ans: CREATE TRIGGER trig3 AFTER DELETE ON emp FOR EACH ROW
`BEGIN

       INSERT INTO log(val1, val2, ...) VALUES (old.val1, old.val2, ...);

  END;