1. Create a table with eid, ename, salary, hire_date and address columns.

```
create table emp
(eid number, ename varchar2(30), salary number(8,2), hire_date date, address  varchar2(50));
```

2. Set primary key on eid.

```
alter table emp add constraint emp_eid_pk primary key(eid)
```

3. Insert five record using substitution variables.

```
insert into emp (eid, ename, salary, hire_date, address)
values(&eid, '&ename', &salary, '&hire_date', '&address');
```

4. Add two columns named as phone, job_title.

```
alter table emp add(phone varchar2(11), job_title varchar2(30));
```

5. Make a query with ename, job_title, salary and salary with 30% bonus.

```
select ename, job_title, salary, salary+(salary*.30) as review_salary
From emp;
```

6. Make a query with ename, salary where salary is equal to minimum salary of employees.

```
select ename, salary
From emp
Where salary =(select MIN(salary) from emp);
```

7. Update all job_title to manager.

```
update emp set job_title='Manager'
```

8. Create a view.

```
create view v_emp as select ename, job_title, salary, salary+(salary*.30) as review_salary
from emp;
```

9. Create a sequence.

```
create sequence eid_seq
increment by 1
start with 100 maxvalue 1000000
nocache nocycle;
```

10. Create an index.

```
create index emp_id_idx On emp(eid)
```

11. Create a trigger

```
create table noman
(id number,
name varchar2(30),
salary number (8,2));
----------------------------------------
create table audit_noman
(id number,
name varchar2(30),
old_salary number (8,2),
new_salary number (8,2), change_date date);
----------------------------------------
desc noman
desc audit_noman
---------------------------------------
insert into noman
values(4, 'C', 5000.00);
--------------------------------------
create or replace trigger update_audit_noman
before update on noman
for each row
```

```
begin
insert into audit_noman values(:old.id, :old.name, :old.salary, :new.salary, sysdate);
end;
----------------------------------------
update noman set salary=6000 where id=12
----------------------------------------
select *from  audit_noman
```

Procedure:

```
create or replace procedure sqNum(x in out number) as
begin
x :=x*x;
end;
-------------------
set serveroutput on;
declare
a number :=6;
begin
seqNum(a);
dbms_output.put_line(a);
end;
```

function with Paramiter

```
create or replace function findMax(x in number, y in number)
Return number is z number;
begin
if x>y then
z:=x;
else
z:=y;
end if;
return z;
end;
----------------------------------------
set serveroutput on
begin
dbms_output.put_line(findMax(45,44));
end;
----------------------------------------
```

function Total Employees

```
create or replace function totalEmployees
return number is total number (3):=0;
begin
select count  (*) into total
from employees;
return total;
end;
--------------------------------
declare
c number(3);
begin
c:= totalEmployees();
dbms_output.put_line('Total Employee is:'||c);
end;
```

created by: @ Abdullah Al Noman