==============Answer=============

```
connnect sys as ssysdba
 enter password:
```

======create database=====

```
 create database oracle;
 create database medical;
```

=======create user/grant======

```
create user motin identified by motin;
grant dba to motin;
grant all privileges to motin;
grant unlimited tablespace to motin;
grant super_user to motin;
alter user motin quota 500m on system;
grant create session, create table, create sequence, create view, create
index to super_user;


connect motin
enter password:
```

=======create table========

```
create table contact (id, number(6) primary key not null, cname
varchar2(50) not null unique, city varchar2(50), phone varchar2(50));

create department(id number(6), name varchar2(50), reg_date date sysdate,
addresss varchar2(50) default 'Bangladesh');

create employee(id number(6) primary key, fname varchar2(50), lname
varchar2(50) not null unique, age number(6),salary
number(8,2)check(salary>=30000) "review salary", joining_date date
check(joining_date<='01_jan_2000'), city varchar2(50),
con_id number(6), dept_name varchar2(50), country varchar2(50) default
'bd' foreign key (con_id) references
contact (id), foreing key (dept_id) references department(name));

create  table emp_backup as select * from employee;
```

=======primary key/unique/add column=======

```
alter table department add constraint dept_id_pk primary key(id);
alter table department add constraint dept_name_uni unique(name);

alter table contact add (job_title varchar2(50), mobile varchar2(15));
alter table contact modify (manager varchar2(50));
alter table contact drop(mobile);
```

======insert/update data=======

```
insrt into department (id, name, salary, age, country) values (&id,
'&name', &salary,
    &age, '&country');
insert into employee (id, fname, lname, job_title,citey) values (101,
abdul, motin, it_programmer, savar);


update deprtment set job_title = 'manager';
update employee set id 01 wherw namw ='motin';
update employee ste name ='motin' where id = 01;
```

=====sequence/index/view/synonym=======

```
create sequence emp_seq increment by 5 start with 5 maxvalue 10000
nocache nocycle;
create sequence dept_seq increment by 1 start with 2 maxvalue 500 nocache
nocycle;

create index emp_id_idx on employee (id, name, salary, country);

create or replace synonym enter for emp_dept_con_view;



create or replace view emp_dept_view as select d.id, d.name, d.city,
e.id, e.fname, e.lname, e.job_title, e.age from department d employee e
   where e.dept_id=d.id;

create or replace view emp_dept_view (id, name, salary, mobile, address)
as select d.id, d.name, e.lname. Max(e.salary), Min(e.salary),
Avg(e.salary) e.country from department d join
   employee e on(e.department_id = d.department_id) group by
d.department_name;
```

=========create trigger========

```
create or replace trigger emp_trigger_after after insert on employee
for each row
declare
begin
insert into emp_backup values(:new.id, :new.name, :new.salary,
:new.mobile :new.country);
dbms_output.put_line('Data inserted successfully');
end;
/
```

```
insert into employee values(101, 'sahin', 32000, o1740197705, 'bd')
```

==========update trigger=========

```
create or replace trigger emp_trigger_update  after update on employee
for each row
declare
begin
update emp_backup
set id=:new.id, name=:new.name, salary=:new.salary, mobile=:new.mobile,
country=:new.country
where id=:old.id or name=:old.name or salary=:old.salary or
country=:old.country;
dbms_output.put_line('Data updated successfully');
end;
/

update employee set id= 101 where name= 'robi';
update employee set name= 'robi' where id= 101;
```

==========delete trigger=========

```
create or replace trigger emp_trigger_delete  after delete on employee
for each row
declare
begin
delete from emp_backup
where id=:old.id or name=:old.name or salary=:old.salary or
country=:old.country;
dbms_output.put_line('Data deleted successfully');
end;
/


delete from employee where name='AAA';
delete from employee where id =101;
```

==========create procedure=========

```
create or replace procedure insertCustomer(
p_id in customer.id%type,
p_name in customer.name%type,
p_mobile in customer.mobile%type,
p_country in customer.country%type);
is
begin
insert into customer(id, naem, mobile, country) values(p_id, p_name,
p_mobile, p_country);
```

```
commit;
end;
/

begin
insertCustomer(101, 'BBB', 01740197705, 'bd');
end;
/
```

==========update procedure=========

```
create or replace procedure updateCustomer(
p_id in customer.id%type,
p_name in customer.name%type,
p_mobile in customer.mobile%type,
p_country in customer.country%type);
is
begin
update customer set id=p_id, name=p_name, mobile=p_mobile,
country=p_country
where id= p_id or name = p_name;
commit;
end;
/

begin
updateCustomer(101, 'CCC', 01740197705, 'bd');
end;
/
```

==========delete procedure=========

```
create or replace procedure deleteCustomer(
p_id in customer.id%type,
p_name in customer.name%type,
p_mobile in customer.mobile%type,
p_country in customer.country%type);
is
begin
delete from customer where id=p_id or name=p_name or mobile= p_mobile or
country= p_country;
commit;
end;
/

begin
deleteCustomer(101, 'SSS', 01740197705, 'bd');
end;
/
```

==========create function=========

```
create or replace function totalCustomer
return number is
total number(2) :=0;
begin
select count (*) into total from customer;
return total;
end;
/

declare
c number(2);
begin
c total cuatomer ();
dbms_output.put_line('Total no of customer: ' || c);
end;
/

insert into customer valuse(101, 'PPP', 26, 'bd');
```