1. create a table Employee with the following schema (emp-no, E-name, E-address, E-Ph-no, Dept-no, Dept-name, job-id, Salary)

```
create table Employe (Emp_no Int(3), E-name char(20),
E_address varchar2(10), E-Ph_no varchar(10), Dept_no
Int(3), Dept_name char(10), Job-Id varchar2(10),
Salary Number(10,2));
```

a. Add a new column: Hiredate to the existing relation

```
alter table employee add hiredate date;
```

b) Change the data type of JOB-ID from char to varchar2.

```
alter table employee modify job-id varchar2(10);
```

c) Change the name of the Emp-no column to E-no

```
alter table employee rename column Emp_no to
E_no.
```

d) Modify the colum width of the job field of Emp table

```
alter table employee modify JOb-id varchar2(15)
```

2. Create a table Employee with following schema: (Emp_no, E-name, E-address, E-Ph-no, Dept_no, Dept-name, Job-id, Salary.

→

a) Insert at least 5 rows in the table

```
Insert into employee(emp-no, E-name, E-address, E-Ph-no,
    Dept-no, Dept-name, Job-id, Salary)
Values (1, abc, warangal, 950280, 22, AIML, ax2, 50000.00
```

b. Select all the information of emptable

select * from employee;

c) update the city of Emp-no-12 with current city as Nagpur

Update employee set E-address='Nagpur' where Emp-no=12;

3

a) Display the details of Employee who works in department MECH.

select * from employe where Dept_name='MECH'.

b) Dellete the Email-id of employee James

update employee set Email=null where Ename ='James'.

c) Display the complete record of employees working in the Sales Department

select * from employee wherer Dept_name='sales';

4) Cre
enar

insert
(101
"
"
"
"
"

a) cou
Sele
fro

b) fe
sel
em

c) f
se
en

5) a)
e

b)
s
Se

c)

4) Create table ~~[illegible]~~ Employee (E_id int(3), Ename char(10), Age int(2) Salary int(5));

insert into employee(EID, E_name, age, salary) values.
(101,'Anu', 22, 9000),
"
"
"
"
"

a) count the number of employee names
Select count(E_name) as number_of_employee from employee;

b) find the max age ?
select max(age) as maximum_ge from employee ;

c) find the minimum age
select min(age) as minimum_age from employee;

5) a) Select salary, count(*) as count from employee group by salary;

b) find salaries in ascend
select salary from employee order by Salary Asc;

c) Select Salary from employee order by salary Desc;

6

~~Create table Employee (EmpNO,~~ EmpName, JOB.
~~Manager_NO, SAL, comission~~

create table Employee (Emp_NO int (3),
Emp_name char (10), JOB char (10), manager_no
int (5), sal int (5), commission int (2))

a

insert into Employee (Emp_no, ename, JOB,
Manager_NO, SAL commission) values (101,
'abhi', 'manager', 1234, 1100, 70)

SQL rollback;

b) ~~Alter table Employee add constraint.~~
Add primary key constraint
Alter table Employee ADD primary key
(EmpNO).

Add not null constraint.
Alter Table Employee modify Ename
varchar (20) NOT NULL;

Alter Table Employee modify JOB varchar(20)
NOT NULL

c

Insert into employee values (104, 'Rahul',
'clerk', NULL, NULL, NULL);

7

a) selec
where
b. colo
sailo
r2.s
igr

b) sel
boats
b. bi d
name
where
and

c) Sel
exists (
103 an

7

a) select s.name from sailors s, res erves r, boats b
where s.sid = r.sid and r.bid = b.bid and
b. color='red' intersect select s2.name from
sailors s2, reserves r2, boats b2, where s2.sid =
r2.sid and r2.bid = b2.bid and b2.color=
'green';

b) select s.name from sailors s, reserves r,
boats b where s.sid = r.sid and r.bid =
b.bid and b.color='red' union all select s2.
name froms sailors s2, reserves r2, boats b2,
where s2.sid = r2. rsid and r2.bid = b2.bid
and b2.color = 'green';

c) Select s.sname from sailors s where
exists (select * from reserves r where r.bid =
103 and r.sid = s.sid);

8

a) select avg(s.age) from Sailors s where s.rating = 10;

b) select s.sname, ~~s~~ s.age from sailors s where s.age = (select max(age) from sailors);

c) Select s.rating, min(s.age) from sailors s group by s.rating;

d) select s.rating, avg(s.age) as average from sailors s group by s.rating having count(*) > 1;

9)

customer table

create table customer (ID, int(1), Name char(10), Age int(2), Address char(15) salary float);

~~insert values (ID, N~~

insert into customer(ID, name, Age, Address, salary) values (1, 'Ramesh', 32, 'Ahmedabad', 2000.00)

4

order table

create table order (OID int(2), Day Date, customer_ID int(2), Amount int(4));

~~select~~

~~9~~

a Selec
from
Inne

outp

3
3

2
4

b Left

Selec
Fro
left

c Right
select
From
right

~~select id, name, amount, day from~~ customer.

a
```
select id, name, amount, day
from customer
Inner Join order ON customer.id=
                    order.customer id
```

output

3	Kaushik	3000	08-OCT-09
3	"	1500	—
	Khilan	1560	—
2 4	chaitali	2060	—

b Left outer Join

```
select id, name, amount, day
From customer
left join order on customer.id=
                  order.customerid;
```

c Right outer join

```
select id, name, amount, day
From customer
right join order on customer.id=
                    order.customerid;
```

10

~~a select s.sname from sailors [illegible] [illegible]~~
~~reserved red boat~~

10

a) select s.sname From sailors where s.sid
In(select r.sid from reserves r where
r.bid IN (select b.bid from from boats
b where b.color='red'));

b) Select s.sname from sailors S where
exists (select * from reserves r where
r.bid=103 and r.sid=s.sid);

11

a) select *
From sailors
where rating > Any (select rating from sailors
where name='Horatio');

b) select *
from Sailors
where rating >=ALL(select rating from
sailors);

```
6  select s.sname from sailors s, reserves r,
   boats b where s.sid = r.sid and
   r.bid = b.bid and b.color='red' union
   select s2.sname from sailors s2, reserves
   r2, boats b2 where s2.sid = r2.sid and
   r2.bid = b2.bid and b2.color='green';
```

---

(2) To Insert

```
create or replace trigger t1
 before insert on sailors
 for each row
  begin
  :new.sname = upper(:new.sname);
  end;
  /
```

to update

```
create or replace trigger t2
 after update of sid on sailors
  for each row
   begin
   if (:new.sid < 80) then
   raise_application_error(-20017,'can't update');
   end if;
   end;
   /
```

13)

b

```
create or replace triggers t16
after
delete on sailors
for each row
begin
 if c:oId.sid = 22 then
  raise-application-error
  (-20019,'you cannot delete this row);
  endif;
  end;
  /
```

14b or 15b

```
declare
 i = number:=1;
 n number;
 begin
 n:= &n;
 while(i <=10)
 loop
  dbms-output.put-leri (n||'x'||i||' = '||n*i);
   i:=i+1;
  end loop;
  end;
  /
```

16b

```
declare
 fact number :=1;
 n number :=11;
 begin
  while n>0 loop
   fact:=n*fact;
   n:n-1;
   end loop;
   dbms-output.put_line(fact);
  End;
```