1.a) Create a table Student with the following fields:

Regno - primary key, Name, Age-not null, Sex-check constraint, Mark1, Mark2, Mark3, total, avg, result.

Query:

Create table student(regno varchar2(15) primary key, name char(20), age number(3) not null, sex char(2)check(sex in ('m','f')), mark1 number(3), mak2 number(3), mark3 number(3), total number(3), avg number(6,2), result char(5));

1.b) Add 5 Records

Query:

insert into student values('&regno', '&name', &age, '&sex', &mark1, &mark2,&mark3, null, null, null);

For calculating total:  update student set total=(mark1+mark2+mark3);

For calculating average: update student set avg=total/3;

For calculating Result: update student set result= case

when(mark1>=40 and mark2>=40 and mark3>=40) then 'pass'

else 'fail' end;

1.c) Select the students whose name starting with 'L'.

Query:

select\* from student where name like 'L%';

1.d) Select the students whose total field is Null.

Query:

select \*from student where total is null;

1.e) Add a Field Rank number(3) at the end of all the columns.

Query:

Alter table student add rank number(3);

1.f) Select the male student whose result is pass.

Query:

select\* from student where ((sex='m')and (result='pass'));

1.g) Select the students whose name ending with 'H'.

Query:

select\*from student where name like '%h';

1.h) Select the students who passed in only one subject.

Query:

select\*from student where

(mark1>=40 and mark2<40 and mark3<40) or

(mark1<40 and mark2>=40 and mark3<40) or

(mark1<40 and mark2<40 and mark3>=40);

1.i) Select the students who got 100 in mark1.

Query: select \* from student where mark1=100;

2.a) Create a table ebbill with the following fields

customer\_id – primary key, customer\_name, address, previous\_reading, current\_reading, unit\_consumed - not null, amount, bill\_status.

Query:

create table ebbill(cust\_id varchar2(10)primary key,cust\_name char(20),address varchar2(30),prev\_reading number(4),curr\_reading number(4),unit\_consumed number(4)not null,amount number(7,2),bill\_status char(10));

2.b) Add 5 records

Query:

insert into ebbill values ('&cust\_id', '&cust\_name', '&address', &prev\_reading, &curr\_reading, &unit\_consumed, null ,'&bill\_status');

For calculating amount:

update ebbill set amount=case

when unit\_consumed<=100 then unit\_consumed\*2

when unit\_consumed>100  and unit\_consumed<=200 then unit\_consumed\*3

when unit\_consumed>200 and unit\_consumed<=300 then unit\_consumed\*4

else unit\_consumed\*5 end;

2. c) Select the details of the customers who have not paid the bill amount.

Query: select\*from ebbill where bill\_status='not paid';

2.d) Arrange the records according to the unit\_consumed.

Query: select \* from ebbill order by unit\_consumed;

2.e) Select the customers whose unit\_consumed is less than 100 units.

Query: select \* from ebbill where unit\_consumed <100;

2.f) Add a new field bill\_date date.

Query: alter table ebbill add bill\_date date;

3.a) Create a table Inventory with the following fields:

product\_id – Primary key, product\_name, supplier\_name, unit\_price, quantity, total\_amt.

Query:

Create table inventory( product\_id varchar2(10) primary key, product\_name varchar2(20), supplier\_name char(20), unit\_price number(4), quantity number(4) not null, total\_amt number(6,2));

3.b) Add 5 records

Query:

Insert into inventory values(‘&product\_id’,’&product\_name’,’&supplier\_name’,&unit\_price,&quantity,null);

3.c) Calculate the total amount.

Query: Update inventory set total\_amt= (unit\_price\* quantity);

3.d) Select the product\_name, unit\_price from the table.

Query: Select product\_name , unit\_price from inventory;

3.e) Arrange the records based on quantity.

Query: Select\* from inventory order by quantity;

3.f) Add a new field stock number(6).

Query: Alter table inventory add stock number(6);

4.a) Create a table employee with the following fields:

emp\_id – Primary key, ename, sex- check constraint, department, basic\_pay, DA, HRA, PF,Gross\_pay, Net\_pay.

Query:

Create table employee(emp-id varchar2(10) primary key, ename char(20), sex char(4) check( sex in(‘m’,’f’)), department char(15), basic\_pay number(8,2), DA number(7,2), HRA number(8,2), PF number(7,2), gross\_pay number(8,2),net\_pay number(9,2));

4.b) Add 5 records.

 Query : insert into emp values('&eid','&ename','&sex','&dep',&basicpay,&DA,&HRA,&PF,null,null);

4.c) Select the employee whose name ending with ‘a’.

QuerySelect \* from employee where name like ‘%a’;

4.d) Calculate GrossPay and NetPay.

For calculating Gross pay:

Query : update emp set grosspay=(basicpay+DA+HRA);

For calculating Net pay:

Query : update emp set netpay=(grosspay-PF);

4.e) Select the employee from IT department.

Query :Select\* from employee where department=’IT’;

4.f) Calculate the total amount paid by the organisation to the employee.

Query :select sum(grosspay)from emp;

4.g) Select the empolyee whose Gross\_pay is above 50000.

Query :Select\* from employee where gross\_pay> 50000;

4.h) Select the employee who got the lowest salary.

Query :Select min(salary) from employee;

5.a) Create a table Telephone with the following fields:

customer\_id- Primary key, customer\_name, street, city, phoneno, bill\_date, due\_date, payment\_date, bill\_amount, bill\_status.

Query :

create table telephone(cust\_id varchar2(5)primary key, cust\_name char(15), street char(15), city char(15), phoneno number(10), bill\_date date, due\_date date, payment\_date date, bill\_amount number(7,2),bill\_status char(10));

5.b) Add 10 records

Query**:** insert into telephone values ('&cid','&cname','&street','&city', &phoneno,'&bill\_date','&due\_date',’&payment\_date’, &bill\_amount,'&bill\_status');

5.c) Select the customer who paid the amount after the due date.

 Query: select\*from telephone where payment\_date>due\_date;

5.d) Select the customers from the respective city (e.g Thambaram).

Query: Select\* from telephone where city=’Thambaram’;

e) Select the details of the customer with the phoneno 234567.

Query: Select \* from telephone where phoneno= 234567;

f) Select the customer whose bill\_amount is less than Rs 500.

Query: Select \* from telephone where bill\_amount< 500;

6.a) Create a table Library with the following fields:

book\_Id- Primary key, Title- not null, Author\_name, Publisher\_name, Year\_of\_ publication.

Query:

Create table library (book\_id varchar2(10) primary key, title char(30) not null, author\_name char(20), publisher\_name char(30), year\_of \_publication varchar2(20));

6.b) Add 5 records.

Query: Insert into library values(‘&book\_id’, ‘&title’, ‘&author\_name’, ‘&publisher\_name’, ‘&year\_of\_publication’);

6.c) Select the books which are published in the year 2000.

Query: Select\* from library where year\_of\_publication=’2000’;

6.d) Delete the records with book\_id 56749.

Query: Delete from library where book\_id=’56749’;

6.e) Modify the field length to 20 for the fieldname Title.

Query: Alter table library modify title char(20);

6.f) Add a new field no\_of\_pages number(6) at the end of all fields.

Query: Alter table library add no\_of\_pages number(6);