Prog. No.....

Date.....

Page No.....

AIM :

create a table customer (cust_no varchar(5), cust_name varchar(15), age number, phone varchar(10))

a) insert 5 records and display it.
b) add new field d_birth with date datatype.
c) create another table Cust_Phone with fields cust_name and phone from customer table.
d) Remove the field age.
e) change the size of the cust_name to 25
f) delete all the records from the table.
g) Rename the table Customer to cust.
h) drop the table.

PROGRAM :

```
create table customer (cust_no varchar(5),
cust_name varchar(15),
age integer,
phone varchar(10));

a) insert into customer values(1,'A',23,94567);
   insert into customer values(2,'B',23,94567);
   insert into customer values(3,'C',23,94567);
   insert into customer values(4,'D',23,94567);
```

(a)

| cust_no | cust_name | age | Phone |
|---|---|---|---|
| 1 | A | 23 | 94567 |
| 2 | B | 23 | 94567 |
| 3 | C | 23 | 94567 |
| 4 | D | 23 | 94567 |
| 5 | E | 23 | 94567 |

(b)

| cust_no | cust_name | age | phone | d_birth |
|---|---|---|---|---|
| 1 | A | 23 | 94567 | NULL |
| 2 | B | 23 | 94567 | NULL |
| 3 | C | 23 | 94567 | NULL |
| 4 | D | 23 | 94567 | NULL |
| 5 | E | 23 | 94567 | NULL |

(c)

| cust_name | Phone |
|---|---|
| A | 94567 |
| B | 94567 |
| C | 94567 |
| D | 94567 |
| E | 94567 |

Prog. No.....

Date.....

Page No.....

```
insert into customer values (5,'E',23,94567);
select * from customer;
```

b) alter table customer add d_birth date;

```
select * from customer;
```

c) Create table cust_phone as select cust_name, phone from customer;

```
select * from cust_phone;
```

(d)

| cust_no | cust_name | phone | d_birth |
|---|---|---|---|
| 1 | A | 94567 | NULL |
| 2 | B | 94567 | NULL |
| 3 | C | 94567 | NULL |
| 4 | D | 94567 | NULL |
| 5 | E | 94567 | NULL |

(e)

| cust_no | cust_name | phone | d_birth |
|---|---|---|---|
| 1 | A | 94567 | NULL |
| 2 | B | 94567 | NULL |
| 3 | C | 94567 | NULL |
| 4 | D | 94567 | NULL |
| 5 | E | 94567 | NULL |

(f)

| cust_no | cust_name | phone | d_birth |
|---|---|---|---|

(g)

| cust_no | cust_name | phone | d_birth |
|---|---|---|---|

Prog. No.....

Date.....

Page No.....

d) alter table customer drop age;

select * from customer;

e) ALTER TABLE customer MODIFY COLUMN cust_name varchar(25);

Select * from customer;

f) TRUNCATE customer;
select * from customer;

g) ALTER TABLE customer RENAME To cust;

select * from cust;

h) drop table cust;

Prog. No.....

Date.....

Page No.....

AIM:

create a hospital table with the fields (doctorid, doctorname, department, qualification, experience)
write the queries to perform the following:

a) insert 5 records
b) Display the details of doctors.
c) Display the details of doctors who have the qualification 'MD'.
d) Display all doctors who have more than 5 years experience but do not have the qualification 'MD'.
e) Display the doctors in 'Skin' department
f) update the experience of doctor with doctorid = 'D003' to 5
g) delete the doctor with doctorid = 'D005'.

PROGRAM:

```
create table hospital (doctor_id char(4),
doctor_name varchar(10), department
varchar(25), qualification varchar (25),
experience int);

i) Insert into hospital values ('D001', 'miya',
                                'cardiologist', 'mbbs', 5);
```

(b)

| doctor_id | doctor_name | department | qualification | experience |
|---|---|---|---|---|
| D001 | miya | cardiologist | mbbs | 5 |
| D002 | john | orthologist | md | 4 |
| D003 | ramesh | skin | mbbs | 3 |
| D004 | madona | dentist | bds | 6 |
| D005 | manoj | optometry | md | 1 |

(c)

| doctor_name |
|---|
| john |
| manoj |

(d)

| doctor_name |
|---|
| madona |

(e)

| doctor_name |
|---|
| ramesh |

Prog. No.....

Date.....

Page No.....

```
insert into hospital values('D002', 'johu',
'orthologist', 'md', 4);
Insert into hospital values('D003', 'ramesh',
'skin', 'mbbs', 3);
insert into hospital values('D004', 'madona',
'dentist', 'bds', 5);
insert into hospital values('D005', 'manoj',
'optometry', 'md', 1);
```

b) Select * from hospital;

c) select doctor_name from hospital where qualification = 'md';

d) select doctor_name from hospital where experience > 5 and qualification != 'md';

e) select doctor_name from hospital where department = 'skin';

(f)

| doctor_id | doctor_name | department | qualification | experience |
|---|---|---|---|---|
| D001 | miya | cardiologist | mbbs | 5 |
| D002 | john | orthologist | md. | 4 |
| D003 | ramesh | skin | mbbs | 5 |
| D004 | madona | dentist | bds | 6 |
| D005 | manoj | optometry | md | 1 |

(g)

| doctor_id | doctor_name | department | qualification | experience |
|---|---|---|---|---|
| D001 | miya | Cardiologist | mbbs | 5 |
| D002 | john | orthologist | md | 4 |
| D003 | ramesh | skin | mbbs | 5 |
| D004 | madona | dentist | bds | 6 |

Prog. No.....

Date.....

Page No.....

f) update hospital set experience = 5
where doctor_id = 'D003';

Select * from hospital;

g) delete from hospital where doctor_id = 'D005';

Select * from hospital;

Prog. No.....

Page No.....

Date.....

AIM:

Create a table employees with (empid, Ename, salary, Department, age). Insert some records. write SQL queries using aggregate functions and group by clause

A. Display the total number of employees.
B. Display the name and age of the oldest employee of each department.
C. Display the average age of employees of each department.
D. Display departments and the average salaries
E. Display the lowest salary in employee table
F. Display the number of employees working in purchase department.
G. Display the highest salary in Sales department.
H. Display the difference between highest and lowest salary.

PROGRAM:

```
create table employee(empid int PRIMARY KEY,
Ename varchar(10), salary int, department
varchar(20), age int);

insert into employee values(001,'vishnu',15000,
```

| Empid | ename | salary | department | age |
|---|---|---|---|---|
| 1 | vishnu | 15000 | marketting | 49 |
| 2 | basith | 150000 | sales | 25 |
| 3 | sreeraj | 30000 | purchase | 35 |
| 4 | infan | 2000 | sales | 35 |
| 5 | safwan | 2000 | sales | 25 |

A)

| count(empid) |
|---|
| 5 |

B)

| ename | department |
|---|---|
| vishnu | marketting |
| basith | sales |
| sreeraj | purchase |

C)

| department | avg(age) |
|---|---|
| marketting | 49.0000 |
| sales | 28.3333 |
| purchase | 35.0000 |

Prog. No.....

Date.....

Page No.....

'marketting', 49);

insert into employee values(002, 'basith', 150000, 'sales', 25);

insert into employee values(003, 'sreeraj', 30000, 'purchase', 35);

insert into employee values(004, 'infan', 2000, 'sales', 35);

insert into employee(005, 'safwan', 2000, 'sales', 25);

select * from employee;

A. select count(empid) from employee;

B. select ename, department from employee where age in (select max(age) from employee group by department;

C. select department, avg(age) from employee group by department;

(D)

| department | avg (salary) |
|---|---|
| marketting | 1500.0000 |
| sales | 51333.3333 |
| purchase | 30000.0000 |

(E)

| min_salary |
|---|
| 2000 |

(F)

| count (Ename) |
|---|
| 1 |

(G)

| max(salary) |
|---|
| 150000 |

| Sal_difference |
|---|
| 148000 |

Prog. No.....

Page No.....

Date.....

D. Select department, avg(salary) from employee group by department;

E. select min(salary) as min_salary from employee;

F. select count(Ename) from employee where department = 'purchase';

G. select max(salary) from employee where department = 'sales';

H select max(salary) - min(salary) as sal difference from employee;

Prog. No.....

Date.....

Page No.....

AIM :

Create a table product with the fields (product_code primary key, product_name, category, quality, price). Insert some records write the queries to perform the following:

a. Display the records in descending order of product_name.

b. Display product_code, product_name with price between 20 and 50.

c. Display the details of products which belongs to the categories of 'bath soap', 'paste' or 'washing powder'.

d. Display the products whose quantity less than 100 or greater than 500.

e. Display the products whose names starts with 's'.

f. Display the products which not belongs to the category 'paste'.

g. Display the products whose second letter is 'a' and belongs to the category 'washing powder'.

| pr_code | pr_name | category | quantity | price |
|---|---|---|---|---|
| 1 | colgate | paste | 10 | 100 |
| 2 | closeup | paste | 9 | 90 |
| 3 | nirma | bath soap | 10 | 600 |
| 4 | nirma | washing powder | 10 | 700 |
| 5 | toy | car | 1 | 200 |
| 6 | toy | bike | 3 | 300 |
| 7 | lux | bath soap | 1 | 20 |
| 8 | lux | bath liquid | 600 | 2000 |
| 9 | Nirma | bath liquid | 300 | 1000 |

(a)

| pr_code | pr_name | category | quantity | price |
|---|---|---|---|---|
| 5 | toy | car | 1 | 200 |
| 6 | toy | bike | 3 | 300 |
| 3 | nirma | bath soap | 10 | 600 |
| 4 | nirma | washing powder | 10 | 700 |
| 9 | nirma | bath liquid | 300 | 1000 |
| 7 | lux | bath soap | 1 | 20 |
| 8 | lux | bath liquid | 600 | 2000 |
| 1 | colgate | paste | 10 | 100 |
| 2 | closeup | paste | 9 | 90 |

Prog. No.....

Date.....

Page No.....

PROGRAM :

```
create table product (pr_code primary key,
pr_name varchar(20), category varchar(20),
quantity int, price int);

insert into product values (1,'colgate','paste',10,100);
insert into product values (2,'closeup','paste',9,90);
insert into product values (3,'nirma',bathsoap',10,600);
insert into product values (4,'nirma','washing powder',
                            10,700);
insert into product values (5,'toy','car',1,200);
insert into product values (6,'toy','bike',3,300);
insert into product values (7,'lux',bath soap',1,20);
insert into product values (8,'lux',bath liquid',600,2000);
insert into product values (9,'nirma','bath liquid',
                            300,1000);

Select * from product;
```

a. select * from product
order by pr_name desc;

(b)

| pr_code | pr_name |
|---|---|
| 7 | lux |

(c)

| pr_name | price |
|---|---|
| colgate | 100 |
| close up | 90 |
| nirma | 600 |
| nirma | 700 |
| lux | 20 |

(d)

| pr_code | pr_name | category | quantity | price |
|---|---|---|---|---|
| 1 | colgate | paste | 10 | 100 |
| 2 | close up | peste | 9 | 90 |
| 3 | nirma | bathsoap | 10 | 600 |
| 4 | nirma | washing powder | 10 | 700 |
| 5 | toy | car | 1 | 200 |
| 6 | toy | bike | 3 | 300 |
| 7 | lux | bath soap | 1 | 20 |
| 8 | lux | bath liquid | 600 | 2000 |

(e) Empty set

Prog. No.....

Date.....

Page No.....

b. select pr_code, prname from product where price between 20 and 50;

c. select prname, price from product where category in ('bath soap', 'paste', 'washing powder');

d. select * from product where quantity < 100 or quantity > 500;

e. select pr_name from product where pr_name like 's %';

(e)

| pr_name |
|---|
| nirma |
| nirma |
| nirma |
| toy |
| toy |
| lux |
| lux |

(g)

| PR_name |
|---|
| Super wash |

Prog. No.....

Date.....

Page No.....

f) select pr_name from product where category != 'paste';

g) insert into product values(10,'super wash', 'washing powder', 600, 2000);

select pr_name from product where pr_name like '_u%' and category = 'washing powder';

Prog. No.....

Date.....

Page No.....

AIM :

Consider the employee database given below. Give an expression in SQL for each of the following queries :

Employee (employee-name, city)
works (employee-name, company-name, salary)
Company (company-name, city)
manages (employee-name, manager-name)

a. find the names of all employees who work in infosys.
b. find the names and cities of residence of all employees who works in wipro.
c. find the names, & cities of all employees who work in infosys & earn more than RS. 10,000
d. find the employees who live in the same cities as the companies for which they work.
e. find all employees who do not work in wipro corporation.
f. find the company that has the most employees.

Date.....

PROGRAM:

```
create talele employee (empname varchar(10)
    primary key , city varchar (10));
create talele company (companyname
  varchar(10) primary key , city varchar (10));
create talele works (empname varchar (10)
 primary key references employee (empname),
 cname varchar (10) references company
 (companyname) , salary int);
create talele manages (empname varchar(10).
 references employee (empname), managesname
 varchar(10) references employee (empname),
 primary key (empname, managername);

insert into employee values ('swathi' , 'kzkd');
insert into employee values ('vishnu', 'tvm');
insert into employee values ('shreya', 'usa');
insert into employee values ('choily', 'dubai');
insert into employee values ('sajid' , 'malappuram');

insert into company values ('infosys' , 'tvm');
insert into company values ('chandrika', 'kolkata');
insert into company values ('wipro' , 'kochi');
insert into company values ('tata', 'mumbai');
insert into company values ('vibro', 'delhi');
```

(a)

| empname |
|---|
| Chaily |
| Swathi |
| Vishnu |

(b)

| empname | city |
|---|---|
| Shreya | usa. |

(c)

| empname | city |
|---|---|
| Shreya | usa |

Prog. No.....

Date.....

Page No.....

```
insert into works values ('Swathi','infosys',10000);
insert into works values ('vishnu','infosys',15000);
insert into works values ('Shreya','wipro',71502);
insert into works values ('chaily','infosys',8000);
insert into works values ('Sajid','vibro',12000);

insert into manages values('Swathi','Shreya');
insert into manages values('vishnu','vishnu');
insert into manages values('Sajid','chaily');
insert into manages values('Sajid','Swathi');
insert into manages values('Shreya','vishnu');
```

a. Select empname from works where cname = 'infosys');

b. select employee.empname, employee.city from employee, works where employee.empname= works.empname and cname = 'wipro';

c. select employee.empname, city from employee, works where employee.empname = works.empname and cname = 'wipro' and salary > 10000;

(d)

| empname |
| --- |
| vishnu |

(e)

| empname |
| --- |
| choily |
| sajid |
| swathi |
| vishnu |

| cname |
| --- |
| infosys |

Prog. No.....

Date.....

Page No.....

d. Select employee.empname from employee, works, Company where employee.empname = works.empname and employee.city = company.city and works.cname = company.companyname ;

e. select empname from works where cname != 'wipro';

f. select cname from works group by cname order by count(*) desc limit 1;

Prog. No.....

Page No.....

Date.....

AIM:

Create table Supplier (Supcode, sname, city)
create table product (pcode, pname)
create table Supl_product (Supcode, pcode, qty)

a. Get all pairs of Supplier numbers such that the 2 Suppliers are located in same city
b. Get Supplier names for Suppliers who Supply product P2.
c. Get product numbers Supplied by more than 1 Supplier.
d. Get Supplier numbers for Suppliers who are located in the same city as Supplier S1
e. Get Supplier names for Suppliers who Supply part P1.
f. Get the number of Suppliers, who are Supplying at least 1 product.
g. For each product Supplied, get the pcode and the total quantity Supplied for that part.

PROGRAM:

create table Supplier (Supcode char(4) primary key, Sname varchar(20), city varchar(15));

Prog. No.....

Date.....

Page No.....

```
create table product (pcode char(4) primary
key, pname varchar (20));
create table splproduct (supcode varchar(4)
references supplier (supcode), pcode
varchar(4) references product (pcode),
qty int, primary key (supcode, pcode));

insert into supplies values ('s1', 'mustafa', 'pdi');
insert into supplies values ('s2', 'kavya', 'ktm');
insert into supplies values ('s3', 'pranav', 'kalndi');
insert into supplies values ('s4', 'manuppa', 'pdi');
insert into supplies values ('s5', 'mdman', 'chtdi');
insert into supplies values ('s6', 'sreya', 'kalndi');

insert into product values ('p1', 'pr1');
insert into product values ('p2', 'pr2');
insert into product values ('p3', 'pr3');
insert into product values ('p4', 'pr4');
insert into product values ('p5', 'pr5');

insert into splproduct values ('s1', 'p1', 150);
insert into splproduct values ('s2', 'p2', 270);
insert into splproduct values ('s3', 'p3', 120);
insert into splproduct values ('s4', 'p4', 320);
insert into splproduct values ('s5', 'p5', 320);
```

a)

| Supcode | Supcode |
|---|---|
| S1 | S4 |
| S3 | S6 |

b)

| Sname |
|---|
| Kavya |

c)

| Pcode |
|---|
| P4 |

d)

| Supcode |
|---|
| S1 |
| S4 |

e)

| Sname |
|---|
| mustafa |

f)

| count(distinct supcode) |
|---|
| 5 |

g)

| Pname | pcode | qty |
|---|---|---|
| pr1 | P1 | 150 |
| pr2 | P2 | 270 |
| pr3 | P3 | 120 |
| pr4 | P4 | 320 |
| pr4 | P4 | 320 |

Prog. No.....

Date.....

Page No.....

a) select first.supcode, second.supcode from supplies first, supplies second where first.city = second.city and first.supcode < second.supcode;

b) select distinct sname from supplies where supcode in (select supcode from splproduct where pcode = 'p2');

c) select pcode from splproduct group by pcode having count(pcode) > 1;

d) select supcode from supplies where city = (select city from supplies where supcode = 'S1');

e) select sname from supplier where supcode in (select supcode from splproduct where pcode = 'p1');

f) select count(distinct supcode) from splproduct;

g) select pname, pcode, qty from product p join splproduct sp on p.pcode = sp.pcode;

Prog. No.....

Date.....

Page No.....

AIM:

create the following tables,

Bank_customer (accno primary key, cust_name, place)

Deposit (accno foreign key, deposit_no, damount)

Loan (accno foreign key, loan_no, Lamount)

write the following queries.

a) Display the details of the customers

b) Display the customers along with deposit amount who have only deposit with the bank.

c) Display the customers along with loan amount who have only loan with the bank.

d) Display the customers they have both loan and deposit with the bank.

e) Display the customer who have neither a loan nor a deposit with the bank.

PROGRAM:

```
create table bank_cust (accno int primary key,
custname varchar(10), place varchar(25));
create talole deposit (accno int references
bank cust (accno), depos it_no int, damount
numeric);
```

(a)

| accno | custname | place |
|---|---|---|
| 101 | ravi | clt |
| 102 | dhasan | tvm |
| 103 | luthufi | mlprm |
| 104 | biju | knr |
| 105 | Jose | klm |
| 106 | Shibu | kch |
| 107 | Shyam | tvm |
| 108 | mohan | knr |

Prog. No.....

Page No.....

Date.....

```
create table loan (accno int references
bank_cust (accno), loan_no int, lamount
numeric);

insert into bank_cust values (101,'ravi','clt');
insert into bank_cust values (102,'dasan','tvm');
insert into bank_cust values (103,'luthufi','mlprm');
insert into bank_cust values (104,'biju','knr');
insert into bank_cust values (105,'Jose','klm');
insert into bank_cust values (106,'shibu','kch');
insert into bank_cust values (107,'shyam','tvm');
insert into bank_cust values (108,'mohan','knr');

insert into deposit values (101,15,400000);
insert into deposit values (102,13,750000);
insert into deposit values (105,12,55000);
insert into deposit values (108,16,750000);

insert into loan values (103, 4,500000);
insert into loan values (104, 2,200000);
insert into loan values (106, 6,300000);
insert into loan values (108, 8,600000);

a) Select * from bank_cust;
```

(b)

| accno | cust_name | damount |
|---|---|---|
| 101 | ravi | 400000 |
| 102 | dhasan | 75000 |
| 103 | jose | 55000 |

(c)

| accno | cust_name | lamount |
|---|---|---|
| 103 | luthufi | 500000 |
| 104 | biju | 200000 |
| 105 | shibu | 300000 |

(d)

| cust_name |
|---|
| mohan |

(e)

| cust_name |
|---|
| shyam |

Prog. No.....

Date.....

Page No.....

b) select b.accno, cust_name, damount from bank_cust b join deposit d on b.accno = d.accno where b.accno not in (select accno from loan);

c) select b.accno, cust_name, lamount from bank_cust b join loan l on b.accno = l.accno where b.accno not in (select accno from deposit);

d) select cust_name from bank_cust where accno in (select accno from loan where accno in (select accno from deposit));

e) select cust_name from bank_cust where accno not in (select accno from loan) union (select accno from deposit));

Prog. No.....

Date.....  Page No.....

AIM :

Prepare a salary report of the employees showing the details such as:
EmpNo, Name, Basic pay, DA, Gross Salary, PF, Net salary, Annual salary, and tax.
for this purpose, create a table named salaries having the following structure.

| Feild name | Type | width |
|---|---|---|
| empno | character | 10 |
| Name | character | 20 |
| Basic | numeric | 6 |

Enter the records of atleast 10 employees use the following information for calculating the details for the report:

DA is fixed as the 40% of the basic pay
PF is fixed as 10% of basic pay
Gross Salary is (Basic pay + DA)
Net Salary is (Gross Salary − PF)
Annual Salary is (12 * net salary).

Prog. No.....

Date.....

Page No.....

Tax is calculated as using the rules;

. If annual Salary is lessthan 100000, No tax
If annual salary is greaterthan 100000 but lessthan or equal to 150000, then the tax is 10% of excess over 100000.
If annual Salary is greater than 150000 but lessthan or equal to 250000, then the tax tax is 20% of excess ovr 150000.
If annual salary is greater than 250000, then the tax is 30% of the excess over 250000.

PROGRAM:

create table Salary (empno int primary key auto increment, name varchar(20), basicpay int, da int, grosssalary int, pf int, netsalary int, annual salary int, tax int);

Insert into salary (empno, name, basicpay) values (1001, 'ritu', 11200);
insert into Salary (name, basic pay) values ('atul', 32500);

| Empno | name | Basicpay | DA | Gross salary | pf | net salary | Annual salary | tax |
|---|---|---|---|---|---|---|---|---|
| 1001 | ritu | 11200 | NULL | NULL | NULL | NULL | NULL | NULL |
| 1002 | atul | 32500 | NULL | NULL | NULL | NULL | NULL | NULL |
| 1003 | vidhya | 35200 | NULL | NULL | NULL | NULL | NULL | NULL |
| 1004 | yathra | 37500 | NULL | NULL | NULL | NULL | NULL | NULL |
| 1005 | sathya | 25000 | NULL | NULL | NULL | NULL | NULL | NULL |

| empno | name | Basicpay | DA | Gross salary | pf | net salary | Annual salary | tax |
|---|---|---|---|---|---|---|---|---|
| 1001 | ritu | 11200 | 4480 | 15680 | 1120 | 14560 | 174720 | 4944 |
| 1002 | atul | 32500 | 13000 | 45500 | 3250 | 42250 | 507000 | 77100 |
| 1003 | vidhya | 35200 | 14080 | 49280 | 3520 | 45760 | 549120 | 89736 |
| 1004 | yathra | 37500 | 15000 | 52500 | 3750 | 48750 | 585000 | 100500 |
| 1005 | sathya | 25000 | 10000 | 35000 | 2500 | 32500 | 390000 | 42000 |

Prog. No.....

Date.....

Page No.....

```
insert into salary (name, basicpay)
Values ('Vidhya', 35200);
Insert into salary (name, basicpay)
Values ('yathra', 37500);
Insert into salary (name, basicpay)
values ('sathya', 25000);

select * from salary;

update salary set da = basic pay * 40/100;
update salary set pf = basicpay * 10/100;
update salary set grosssalary = basicpay + da;
update salary set netsalary = grosssalary - pf;
update salary set annualsalary = 12 * netsalary
update salary set tax = 0 where
    annualsalary < 100000;
update salary set tax = (annualsalary -
100000) * 10/100 where annualsalary <=
150000;
update salary set tax = (annualsalary -
150000) * 20/100 where annualsalary <=
250000;
update salary set tax = (annualsalary -
25000) * 30/100 where annualsalary >
250000;

Select * from salary;
```

Prog. No.....

Page No.....

Date.....

AIM:

create table exam_result (rollno, avg_score, grade) insert 10 records. Assign null values to the field grade. write program block to update the grade field by using following conditions.

avg_score between 90 and 100 - A
avg_score between 75 - 89 - B
avg_score between 60 - 74 - C
avg_score between 50 - 59 - D
avg_score between below 50 - E

PROGRAM:

```
Create table exam_result (rollno int primary
key auto_increment, avgscore int, grade
char);

Insert into exam_result (rollno, avgscore)
values (1, 77);
insert into exam_result (avgscore)
values (84);
Insert into exam_result (avgscore)
values (45);
```

| Rollno | avgscore | Grade |
|---|---|---|
| 1 | 77 | NULL |
| 2 | 84 | NULL |
| 3 | 45 | NULL |
| 4 | 98 | NULL |
| 5 | 64 | NULL |
| 6 | 99 | NULL |
| 7 | 50 | NULL |

| Rollno | avgscore | Grade |
|---|---|---|
| 1 | 77 | B |
| 2 | 84 | B |
| 3 | 45 | E |
| 4 | 98 | A |
| 5 | 64 | C |
| 6 | 99 | A |
| 7 | 50 | D |

Prog. No.....

Page No.....

Date.....

```
insert into exam_result (avgscore) values (98);
insert into exam_result (avgscore) values (64);
insert into exam_result (avgscore) values (99);
insert into exam_result (avgscore) values (50);

Select * from exam_result;

update exam_result set grade = 'a'
where avgscore between 90 and 100;
update exam_result set grade = 'b'
where avgscore between 75 and 89;
update exam_result set grade = 'c'
where avgscore between 60 and 74;
update exam_result set grade = 'D'
where avgscore between 50 and 59;
update exam_result set grade = 'E'
where avgscore < 50;

Select * from exam_result;
```

Output :

Fibonacci series is :

0
1
1
2
3
5
8
13
21
24

Prog. No.....

Page No.....

Date.....

AIM :

create a procedure to print fibonacci number up to a limit, limit is passed as an argument.

PROGRAM :

```
Create procedure generate_fibo (IN n int)
BEGIN
    DECLARE first INT default 0;
    DECLARE second INT default 1;
    DECLARE Third INT;
    DECLARE i INT Default 2;

    Select 'Fibonacci series is :' AS output;
    Select first AS output;
    Select second AS output;

    while i<= n Do
        set third = first + second;
        set first = second;
        set second = third;
        select third AS output
        Set i = i+1
    End while;
End
    call generate_fibo(10);
```

Output:

| is_prime (7) |
| --- |
| 7 is prime |

Prog. No.
Page No.
Date

AIM:
create a function to check whether a given number is prime or not.

PROGRAM:

```
create function is_prime (num int)
Returns varchar (20) deterministic
BEGIN
    DECLARE i int default 2;
    if num <=1 then
    return concat (num, 'is not a prime');
    End if

    while i <= SQRT (num) DO
        if num % i = 0 then
            return Concat (num, ' is
                not a prime');
        End if
        set i = i+1;
    End while;

    return concat (num, ' is prime');
End.

Select is_prime (7);
```

output:

| radius | area |
|---|---|
| 3 | 28.27431 |
| 4 | 50.2655 |
| 5 | 78.5398 |
| 6 | 113.097 |
| 7 | 153.938 |

Prog. No.....

Page No.....

Date.....

AIM:

write a program code to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding value of calculated area in an empty table named areas with field's radius and area.

PROGRAM:

```
create table areas (radius float,
    area float);

create procedure calculate_areas()
BEGIN
    DECLARE current_radius int default 3;
    DECLARE currentradius float;

    while currentradius <=7 DO
        set current_area = PI() *
            current_radius * current_radius;
        insert into areas (radius, area)
        values(current_radius, current_area);
        set current_radius = current_radius
                +1;
```

Prog. No.....

Page No.....

Date.....

```
    End while;
End.
call calculate_area();
select * from areas;
```