

## Program No: 1 Customer Table

### AIM

Create a table customer (cust\_no varchar(5), cust\_name varchar(15), age number, phone varchar(10) )

- insert 5 records and display it
- add new field d\_birth with date datatype
- create another table cust\_phone with fields cust\_name and phone from customer table
- remove the field age
- change the size of the cust\_name to 25
- delete all the records from the table
- rename the table customer to cust
- drop the table

### Program

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

- 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);  
insert into customer values(5,'E',23,94567);

```
select *from customer;
```

### Output

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

(5 rows)

- alter table customer add d\_birth date;

### Output

Column	Type	Modifiers
cust_no	character varying(5)	
cust_name	character varying(15)	
age	numeric	
phone	character varying(10)	
d_birth	date	

- create table cust\_phone as select cust\_name,phone from customer;

```
select *from cust_phone;
```

**Output**

cust_name	phone
A	94567
B	94567
C	94567
D	94567
E	94567

d) alter table customer drop age;

```
select *from customer;
```

**Output**

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

e) ALTER TABLE customer ALTER COLUMN cust\_name TYPE varchar(25);

**Output**

Column	Type	Modifiers
cust_no	character varying(5)	
cust_name	character varying(25)	
phone	character varying(10)	
d_birth	date	

f) TRUNCATE customer;

```
select *from customer;
```

**Output**

cust_no	cust_name	phone	d_o_b
---------	-----------	-------	-------

g) ALTER TABLE customer RENAME TO cust;

```
select *from cust;
```

**Output**

cust_no	cust_name	phone	d_o_b
---------	-----------	-------	-------

h) drop table cust;

## Program No:2 Hospital table

### AIM

Create a table Hospital with the fields (doctorid,doctorname,department,qualification,experience). Write the queries to perform the following.

- Insert 5 records
- Display the details of Doctors
- Display the details of doctors who have the qualification 'MD'
- Display all doctors who have more than 5 years experience but do not have the qualification 'MD'
- Display the doctors in 'Skin' department
- update the experience of doctor with doctored='D003' to 5
- 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);
```

```
A) insert into hospital values('D001','miya','cardiologist','mbbs',5);
insert into hospital values('D002','john','orthologist','md',4);
insert into hospital values('D003','ramesh','skin','mbbs',3);
insert into hospital values('D004','madona','dentist','bds',6);
insert into hospital values('D005','manoj','optometry','md',1);
```

```
B) select * from hospital;
```

### OUTPUT

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

(5 rows)

```
C) select doctor_name from hospital where qualification='md';
```

### OUTPUT

doctor_name
john
manoj

(2 rows)

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

### OUTPUT

doctor_name
-------------

madona  
(1 row)

E) select doctor\_name from hospital where department='skin';

### OUTPUT

doctor\_name  
-----  
ramesh  
(1 row)

F) update hospital set experience=5 where doctor\_id='D003';

select \* from hospital;

### OUTPUT

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

(5 rows)

G) delete from hospital where doctor\_id='D005';

select \* from hospital;

### OUTPUT

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

(4 rows)

## Program No: 3 Aggregate functions

### AIM

Create a table employee with fields (EmpID, EName, Salary, Department, and 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 numeric,department
varchar(20),age int);
```

```
insert into employee values(001,'vishnus',15000,'marketting',49);
insert into employee values(002,'basith ',150000,'sales',25);
insert into employee values(003,'sreerag',30000,'purchase',35);
insert into employee values(004,'infan ',2000,'sales',35);
insert into employee values(005,'safwan ',2000,'sales',25);
```

```
select * from employee;
```

### OUTPUT

empid	ename	salary	department	age
1	vishnus	15000	marketting	49
2	basith	150000	sales	25
3	sreerag	30000	purchase	35
4	infan	200000	sales	35

(4 rows)

```
a) select count(empid)from employee;
```

### OUTPUT

count
4

(1 row)

```
b) select ename,department from employee where age in(select max(age) from employee group by
department);
```

### OUTPUT

ename	department
-------	------------

```

-----+-----
vishnus | marketing
basith  | sales
sreerag | purchase
infan   | sales

```

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

### OUTPUT

```

department |      avg
-----+-----
sales      | 28.333333333333333
marketing  | 49.000000000000000
purchase   | 35.000000000000000
(3 rows)

```

d) select department,avg (salary)from employee group by department;

### OUTPUT

```

department |      avg
-----+-----
sales      | 51333.33333333333
marketing  | 15000.000000000000000
purchase   | 30000.0000000000000
(3 rows)

```

e) select min(salary) as min\_salary from employee;  
select min(salary) as min\_salary from employee;

### OUTPUT

```

min_salary
-----
2000
(1 row)

```

f) select count(ename)from employee where department='purchase';

### OUTPUT

```

count
-----
1
(1 row)

```

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

### OUTPUT

```

max
-----
150000
(1 row)

```

h) select max(salary) - min(salary) as sal\_difference from employee;

**OUTPUT**

sal\_difference

-----
148000
(1 row)

## Program No: 4 Product Table & Select Queries

### AIM

Create a table product with the fields (Product\_code primary key, Product\_Name, Category, Quantity, Price). Insert some records Write the queries to perform the following.

- Display the records in the descending order of Product\_Name
- Display Product\_Code, Product\_Name with price between 20 and 50
- Display the details of products which belongs to the categories of 'bath soap', 'paste', or 'washing powder'
- Display the products whose Quantity less than 100 or greater than 500
- Display the products whose names starts with 's'
- Display the products which not belongs to the category 'paste'
- Display the products whose second letter is 'u' and belongs to the Category 'washing powder'

### PROGRAM

```
create table product(Product_code int primary key, Product_Name varchar(20),Category
varchar(20), quantity int,price numeric);
```

```
insert into product values(1,'colgate','paste',10,100);
insert into product values(2,'close up','paste',9,90);
insert into product values(3,'nirma','bath soap',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;
```

### OUTPUT

product_code	product_name	category	quantity	price
1	colgate	paste	10	100
2	close up	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

(9 rows)

a) select \* from product order by product\_name desc;

### OUTPUT

product_code	product_name	category	quantity	price
--------------	--------------	----------	----------	-------



5	toy	car		1	200
6	toy	bike		3	300
4	nirma	washing powder		10	700
9	nirma	bath liquid		300	1000
3	nirma	bath soap		10	600
8	lux	bath liquid		600	2000
7	lux	bath soap		1	20
1	colgate	paste		10	100
2	close up	paste		9	90

(9 rows)

b) select product\_code,product\_name from product where price between 20 and 50;

### OUTPUT

product\_code | product\_name

-----+-----

7 | lux

(1 row)

C) select product\_name,price from product where category in ('bath soap','paste','washing powder');

### OUTPUT

product\_name | price

-----+-----

colgate | 100

close up | 90

nirma | 600

nirma | 700

lux | 20

(5 rows)

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

### OUTPUT

product\_code | product\_name | category | quantity | price

-----+-----+-----+-----+-----

2 | close up | 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

1 | colgate | paste | 10 | 100

(8 rows)

E) select product\_name from product where product\_name like 's%';

### OUTPUT

product\_name

-----

(0 rows)

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

**OUTPUT**

product\_name

-----

colgate  
close up  
nirma  
nirma  
toy  
toy  
lux  
lux  
nirma  
radhas

(10 rows)

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

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

**OUTPUT**

product\_name

-----

super wash

(1 row)

## Program No:5 Employee Database

### 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, and cities of all employees who work in Infosys and 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.

### PROGRAM

```
create table employee(empname varchar(10) primary key,city varchar(10));
```

```
create table company(company_name varchar(10) primary key ,city varchar(10));
```

```
create table works(empname varchar(10) primary key references employee(empname),cname  
varchar(10) references company(company_name),salary int);
```

```
create table manages(empname varchar(10) references employee(empname),manager_name  
varchar(10) references employee(empname),primary key(empname,manager_name));
```

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

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

```
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('choilly','infosys',8000);  
insert into works values('sajid','vibro',18000);
```

```
insert into manages values('swathi','shreya');  
insert into manages values('vishnu','vishnu');  
insert into manages values('sajid','choilly');  
insert into manages values('sajid','swathi');  
insert into manages values('shreya','vishnu');
```

a) select empname from works where cname='infosys';

**OUTPUT**

empname

-----

swathi

vishnu

choilly

(3 rows)

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

**OUTPUT**

empname | city

-----+-----

shreya | usa

(1 row)

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

**OUTPUT**

empname | city

-----+-----

shreya | usa

(1 row)

d) select employee.empname from employee,works,company where employee.empname=works.empname and employee.city=company.city and works.cname=company.company\_name;

**OUTPUT**

empname

-----

vishnu

(1 row)

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

**OUTPUT**

empname

-----

sajid

swathi

vishnu

choilly

(4 rows)

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

**OUTPUT**

cname

-----

infosys

(1 row)

## Program No: 6 Supplier-Product Database

### AIM

Create table supplier(supcode,sname,city)  
Create table product (pcode,pname)  
Create table supl\_product(supcode,pcode,qty)

- Get all pairs of supplier numbers such that the two suppliers are located in the same city.
- Get supplier names for suppliers who supply product P2.
- Get product numbers supplied by more than one supplier.
- Get supplier numbers for suppliers who are located in the same city as supplier S1.
- Get supplier names for suppliers who supply part P1.
- Get the number of Suppliers, who are supplying at least one product.
- 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));

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 supplier values('s1','musthafa','pgdi');
insert into supplier values('s2','kavya','ktmchi');
insert into supplier values('s3','pranav','kdlni');
insert into supplier values('s4','manuppa','pgdi');
insert into supplier values('s5','mdmah','chtpdi');
insert into supplier values('s6','sreya','kdlni');
```

```
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','p4',320);
```

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

### OUTPUT

```
supcode supcode
s1      s4
s3      s6
```

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

**OUTPUT**

sname

-----

kavya

(1 row)

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

**OUTPUT**

pcode

-----

p4

(1 row)

d) select supcode from supplier where city =(select city from supplier where supcode='s1');

**OUTPUT**

supcode

-----

s1

s4

(2 rows)

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

**OUTPUT**

sname

-----

musthafa

(1 row)

f) select count (distinct supcode )from splproduct;

**OUTPUT**

count

-----

5

(1 row)

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

**OUTPUT**

pname | pcode | qty

-----+-----+-----

pr1 | p1 | 150

pr2 | p2 | 270

pr3 | p3 | 120

pr4 | p4 | 320

pr4 | p4 | 320

(5 rows)

## Program No: 7 Banking Database

### 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

- Display the details of the customers
- Display the customers along with deposit amount who have only deposit with the bank
- Display the customers along with loan amount who have only loan with the bank
- Display the customers they have both loan and deposit with the bank
- Display the customer who have neither a loan nor a deposit with the bank

### PROGRAM

```
create table bank_customer(accno int primary key,customer_name varchar(10),place varchar(25));
```

```
create table deposit(accno int references bank_customer(accno),deposit_no int, damount numeric);
```

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

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

```
insert into deposit values(101,15,400000);
insert into deposit values(102,13,75000);
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_customer;
```

### OUTPUT

```
accno | customer_name | place
```

```
-----+-----+-----
 101 | ravi      | clt
 102 | dasan     | tvn
 103 | luthufi   | mlprm
 104 | biju      | knr
```

105	jose	klm
106	shibu	kch
107	shyam	tvn
108	mohan	knr

(8 rows)

b) select b.accno,customer\_name,damount from bank\_customer b join deposit d on b.accno=d.accno where b.accno not in(select accno from loan);

### OUTPUT

accno	customer_name	damount
-------	---------------	---------

101	ravi	400000
102	dasan	75000
105	jose	55000

(3 rows)

c) select b.accno,customer\_name,lamount from bank\_customer b join loan l on b.accno=l.accno where b.accno not in(select accno from deposit);

### OUTPUT

accno	customer_name	lamount
-------	---------------	---------

103	luthufi	500000
104	biju	200000
106	shibu	300000

(3 rows)

d) select customer\_name from bank\_customer where accno in((select accno from loan)intersect(select accno from deposit));

### OUTPUT

customer_name
---------------

mohan
-------

(1 row)

e) select customer\_name from bank\_customer where accno not in((select accno from loan)union(select accno from deposit));

### OUTPUT

customer_name
---------------

shyam
-------

(1 row)



## Program No: 8 SALARY REPORT

### 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.

Field Name	Type	Width
EmpNo	Character	10
Name	Character	20
Basic	Numeric	6

Enter the records of at least 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 the basic pay.

Gross Salary is (Basic Pay + DA).

Net Salary is (Gross Salary – PF)

Annual Salary is (12 \* Net Salary)

Tax is calculated using the following rules:

If annual salary is less than 100000, No Tax

If annual salary is greater than 100000 but less than or equal to 150000, then the tax is 10% of the excess over 100000.

If annual salary is greater than 150000 but less than or equal to 250000, then the tax is 20% of the excess over 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,annualsalary int,tax int);

Query ok

Insert into salary (empno,name,basicpay) values(1001,'ritu',11200);

Query ok

Insert into salary (name,basicpay) values('atul',32500);

Query ok

Insert into salary (name,basicpay) values('vidhya',35200);

Query ok

Insert into salary (name,basicpay) values('yathra',37500);

Query ok

Insert into salary (name,basicpay) values('sathya',25000);

Query OK

Select \*from salary;

## OUTPUT

EMPNO	NAME	BASICPAY	DA	GROSSSALARY	PF	NETSALARY	ANNUALSALARY
1001	RITU	11200	NULL	NULL	NULL	NULL	NULL
1002	ATUL	32500	NULL	NULL	NULL	NULL	NULL
1003	VIDHYA	35200	NULL	NULL	NULL	NULL	NULL
1004	YATHRA	37500	NULL	NULL	NULL	NULL	NULL
1005	SATHYA	25000	NULL	NULL	NULL	NULL	NULL

Update salary set da=basicpay\*40/100;

Query ok

Update salary set pf=basicpay\*10/100;

Query ok

Update salary set grosssalary=basicpay+da;

Query ok

Update salary set netsalary= grosssalary-pf;

Query ok

Update salary set annuallsalary=12\*netsalary;

Query ok

Update salary set tax=0 where annuallsalary<100000;

Query ok

Update salary set tax= (annuallsalary-100000)\*10/100 where annuallsalary<=150000;

Query ok

Update salary set tax= (annuallsalary-150000)\*20/100 where annuallsalary<=250000;

Query ok

Update salary set tax= (annuallsalary-250000)\*30/100 where annuallsalary >250000;

Query ok

Select \*from salary;

## OUTPUT

EMPNO	NAME	BASICPAY	DA	GROSSSALARY	PF	NETSALARY	ANNUALSALARY
1001	RITU	11200	4480	15680	1120	14560	174720
1002	ATUL	32500	13000	45500	3250	42250	507000
1003	VIDHYA	35200	14080	49280	3520	45760	549120
1004	YATHRA	37500	15000	52500	3750	48750	585000
1005	SATHYA	25000	10000	35000	2500	32500	390000

## Program No: 9 EXAM RESULT

### 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 the following condition.

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

## PROGRAM

Create table exam\_result(roll no int primary key auto\_increment, avgscore int, grade char);

Query ok.

Insert into exam\_result(roll no, avgscore) values(1,77);

Query ok.

Insert into exam\_result(avgscore) values(84);

Query ok

Insert into exam\_result(avgscore) values(45);

Query ok

Insert into exam\_result(avgscore) values(98);

Query ok

Insert into exam\_result(avgscore) values(64);

Query ok

Insert into exam\_result(avgscore) values(99);

Query ok

Insert into exam\_result(avgscore) values(50);

Query ok

Select \*from exam\_result;

## OUTPUT

RNO	AUGSCORE	GRADE
1	77	NULL
2	84	NULL
3	45	NULL
4	98	NULL
5	64	NULL
6	99	NULL
7	50	NULL

7 rows in set (0.00 sec)

Update exam\_result set grade='a' where avgscore between 90 and100;

Query ok

Update exam\_result set grade='a' where avgscore between 90 and100;

Query ok

Update exam\_result set grade='b' where avgscore between 75 and 89;

Query ok

Update exam\_result set grade='c' where avgscore between 60 and 74;

Query ok

Update exam\_result set grade='d' where avgscore between 50 and 59;

Query ok

Update exam\_result set grade='e' where avgscore <50;  
Query ok

Select \*from exam\_result;

## OUTPUT

RNO	AUGSCORE	GRADE
1	77	B
2	84	B
3	45	E
4	98	A
5	64	C
6	99	A
7	50	D

7 rows in set (0.00 sec)

## Program No: 10 Fibonacci Series

### AIM

Create a procedure to print Fibonacci number up to a limit, limit is passed as an argument

## PROGRAM

```
declare
    first number:=0;
    second number:=1;
    third number;
    n number:=&n;
    i number;

begin
    dbms_output.put_line('Fibonacci series is:');
    dbms_output.put_line(first);
    dbms_output.put_line(second);

    for i in 2..n
    loop
        third:=first+second;
        first:=second;
        second:=third;
        dbms_output.put_line(third);
    end loop;
end;
```

## OUTPUT:

```
Enter value for n: 6
old 5: n number:=&n;
new 5: n number:=6;
Fibonacci series is:
0
1
1
2
3
5
8
```

## Program No: 11 Prime or not

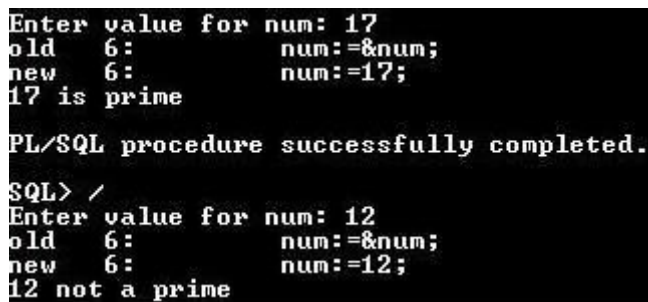
### AIM

Create a function to check whether a given number is prime or not

## PROGRAM

```
declare
    num number;
    i number:=1;
    c number:=0;
begin
    num:=#
    for i in 1..num
    loop
        if((mod(num,i))=0)
        then
            c:=c+1;
        end if;
    end loop;
    if(c>2)
    then
        dbms_output.put_line(num||' not a prime');
    else
        dbms_output.put_line(num||' is prime');
    end if;
end;
/
```

## OUTPUT



```
Enter value for num: 17
old 6:          num:=&num;
new 6:          num:=17;
17 is prime

PL/SQL procedure successfully completed.

SQL> /
Enter value for num: 12
old 6:          num:=&num;
new 6:          num:=12;
12 not a prime
```

## Program No: 12 Area of Circle

### 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

```
SQL>create table areas
      2 (area number(10),
      3 radius number(2));
Table Created.
```

```
SQL>set serveroutput on;
```

```
Declare
Area number(10);
Radius number(2);
Pi constant number(3,2):=3.14;
Begin
Radius:=&radius;
While radius<8
Loop
Area:=pi*power(radius,2);
Insert into areas values(area, radius);
Radius:=radius+1;
end loop;
end;
```

## OUTPUT:

```
Enter value for radius:5
Old 6:radius:=&radius;
New6:radius:=5;
PL/SQL procedure successfully completed.
```

```
SQL>select * from areas;
```

## OUTPUT

AREA	RADIUS
79	5
113	6
154	7