

Date.....

Expt. No.....

Page No.....

Aim :

(create a table customers (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.

SQL :

Create table customers (cust-no varchar(5), cust-name varchar(15), age int, phone varchar(10));

a) insert into customers values (100, 'Amal', 18, '1234567890');

insert into customers values (101, 'Kavya', 19, '2345678901');

insert into customers values (102, 'Twulal', 20, '3456789012');

insert into customers values (103, 'Abay', 20, '4567890123');

insert into customers values (104, 'Abhi', 19, '5678901234');

Select \* from customers;

b) alter table customers add d-birth date;

Select \* from customers;

c) Create table cust-phone as (select cust-name, phone from customers);

Select \* from cust-phone;

Teacher's Signature : .....

Date.....

Expt. No.....

Page No.....

d) alter table customers drop column age;

Select \* from customers;

e) alter table customers modify cust-name varchar(25);

f) delete from customers;

g) alter table customers rename to cust;

h) drop table cust;

Teacher's Signature : .....

Expt. No.....

Output:

Table created

a) 1 row created

cust-no	cust-name	age	Phone
100	Amal	18	1234567890
101	Kavya	19	2345678901
102	Jwala	20	3456789012
103	Abay	20	4567890123
104	Abhi	19	5678901234

b)

cust-no	cust-name	age	Phone	cl-birth
100	Amal	18	1234567890	
101	Kavya	19	2345678901	
102	Jwala	20	3456789012	
103	Abay	20	4567890123	
104	Abhi	19	5678901234	

Date.....

Expt. No.....

Page No.....

c)

cust-name	Phone
Amal	1234567890
Kavya	2345678901
Jwala	3456789012
Abay	4567890123
Abhi	5678901234

d)

cust-no	cust-name	phone	d-birth
100	Amal	1234567890	
101	Kavya	2345678901	
102	Jwala	3456789012	
103	Abay	4567890123	
104	Abhi	5678901234	

e) Table altered.

f) 5 rows deleted.

g) Table altered.

h) Table dropped.

Date.....

Expt. No.....2

Page No.....

Aim:

Create a table hospital with 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'.

SQL:

```
create table hospital (doctorid varchar(5) primary key, doctorname varchar(20),  
department varchar(10), qualification varchar(15), experience int);
```

- a) insert into hospital values ('D003', 'riya', 'ENT', 'MS', 2);  
insert into hospital values ('105', 'Sachind', 'cardiac', 'MD', 3);  
insert into hospital values ('107', 'maxwell', 'Ortho', 'Skin', 1);  
insert into hospital values ('102', 'sharun', 'Onco', 'MS', 6);  
insert into hospital values ('D005', 'harshal', 'dental', 'BDS', 2);  
Select \* from hospital;
- b) Select doctorname, department from hospital;
- c) Select doctorname from hospital where qualification = 'MD';
- d) Select doctorname from hospital where qualification != 'MD' and experience > 5;
- e) Select doctorname from hospital where department = 'Skin';

Teacher's Signature : .....

Expt. No.....

Page No.....

f) update hospital set experience = 5 where doctorid='D003';  
Select \* from hospital;

g) delete from hospital where doctorid = 'D005';  
Select \* from hospital;

Teacher's Signature : .....

Expt. No.....

Output:

Table created.

a) 1 row created.

doctorid	doctorname	department	qualification	Experience
1003	Riya	ENT	MS	2
105	Sachind	cardiac	MD	3
107	maxwell	Skin	MD	1
102	Sharun	Onco	MS	6
0005	Harshal	Dental	BDS	2

b)

doctorname	department
Riya	ENT
Sachind	cardiac
maxwell	Skin
Sharun	Onco
Harshal	Dental

c) Doctor name

Sachind

maxwell

d) Doctorname

Sharun

Date.....

Page No.....

Expt. No.....

e) Doctorname

Maxwell

f)

doctorid	doctorname	department	Qualification	Experience
0003	Riya	ENT	MS	5
105	Sachind	cardiac	MD	3
107	maxwell	skin	MD	1
102	Sharun	onco	MS	6
0005	Harshal	dental	BDS	2

g)

doctorid	doctorname	department	Qualification	Experience
0003	Riya	ENT	MS	5
105	Sachind	cardiac	MD	3
107	maxwell	skin	MD	1
102	Sharun	onco	MS	6

Date.....

Expt. No. 3.....

Page No.....

Aim:

Create the following tables.

Bank\_customer (accno primary key, cust-name, place)

Deposite (accno foreign key, deposite-no, d-amount)

Loan (accno foreign key, loan-no, L-amount)

Write the following queries.

a) Display the details of customers.

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

e) Display the customers who have neither a loan nor a deposite with the bank.

SQL:

```
create table bank_customer (accno varchar(3) primary key, cname varchar(20),  
place varchar(20));
```

```
insert into bank_customer values (103, 'shithin', 'thmsy');
```

```
insert into bank_customer values (102, 'vyshakh', 'koyilandy');
```

```
insert into bank_customer values (101, 'vismaya', 'mukkam');
```

```
insert into bank_customer values (104, 'aswin', 'areekode');
```

```
insert into bank_customer values (105, 'anand', 'manjeri');
```

```
insert into bank_customer values (106, 'anirra', 'malappuram');
```

```
create table deposit (accno varchar(3), deposite-no varchar(3), d-amount  
int, foreign key (accno) references bank_customer (accno));
```

```
insert into deposit values (103, 524, 25000);
```

```
insert into deposit values (106, 525, 30000);
```

Teacher's Signature : .....

Date.....

Page No.....

Expt. No.....

insert into deposite values (101, 527, 38000);  
insert into deposite values (104, 526, 45000);

create table loan(accno varchar(3), loan-no varchar(3), l-amount int,  
foreign key (accno) references bank-customers(accno));  
insert into loan values (101, 641, 37000);  
insert into loan values (102, 642, 28000);  
insert into loan values (103, 643, 32000);  
insert into loan values (101, 644, 29000);  
insert into loan values (105, 645, 43000);

a) select \* from bank-customers;

b) Select cname, d-amount from bank-customers, deposite where bank-  
customer.accno = deposite.accno;

c) Select cname, l-amount from bank-customers, loan where bank-customers.  
accno = loan.accno;

d) Select cname from bank-customers, deposite, loan where bank-customers.  
accno = loan.accno and bank-customers.accno = deposit.accno;

e) Select \* from bank-customers where accno not in (select accno from  
deposit union select accno from loan);

Teacher's Signature : .....

Output:

Table created.

(a)

accno	cname	place
103	Shithin	Thmsy
102	Vyshakh	Koyilandi
101	Vismaya	Mukkam
104	Aswin	Areekode
105	Anand	Manjeri
106	Athira	Malappuram.

b)

cname	d-amount
Shithin	25000
Vismaya	18000
Aswin	45000
Athira	3000

(c)

cname	L-amount
Vyshakh	28000
Vismaya	37000
Vismaya	29000
Anand	32000
Anand	43000

Date.....

Expt. No.....

Page No.....

d) cname

vismaya

e)

accno	cname	place
102	vyshakh	koyilandi
105	anand	manjeri

Date.....

Expt. No...4.....

Page No.....

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 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.

SQL:

```
Create table employee(Empid varchar(3) primary key, ename varchar(15),  
salary int,dept varchar(55),age int);
```

```
insert into employee values (103,'arun','8000','purchase',18);
```

```
insert into employee values (107,'jithin','7800','sales',18);
```

```
insert into employee values (105,'anupama','8300','purchase',19);
```

```
insert into employee values (213,'abay','11000','stock',26);
```

```
insert into employee values (237,'akhil','10800','stock',24);
```

```
Select * from employee;
```

a) Select count(ename) from employee;

b) Select dept,max(age) as maximum-age from employee group by dept;

c) Select dept,avg(age) as average-age from employee group by dept;

Teacher's Signature : .....

Date.....

Expt. No.....

Page No.....

d) select dept, avg (salary) as average\_salary from employee group by dept;

e) Select min (salary) as minimum\_salary from employee;

f) Select count(ename) as number\_of\_employees from employee where dept = 'purchase';

g) Select max(salary) as highest\_salary from employee where dept = 'sales';

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

Teacher's Signature : .....

Output:

Table created

1 row created.

empid	ename	Salary	department	age
103	Arjun	8000	purchase	18
107	Jithin	7800	Sales	18
105	Anupama	8300	purchase	19
213	Abay	11000	Stock	26
237	Akhil	10800	Stock	24

a) count(NAME)

5

b)

Dept	maximum-age
purchase	19
Stock	26
Sales	18

c)

Dept	Average-age
purchase	18.5
Stock	25
Sales	18

Expt. No.....

d) Dept	Average-Salary
purchase	8150
stock	10900
sales	7800

e) Minimum-Salary

7800

f) Number-of-employees

2

g) Highest-Salary

7600

h) Salary-difference

3200

Date.....

Expt. No...5.....

Page No.....

Aim:

Create a table product with fields (product-code primary key, product-name, category, quantity, price). Insert some records write the queries to perform the following.

- a) Display the records in the 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 name starts with 'S'.
- f) Display the products which not belongs to the category 'paste'
- g) Display the products whose second letter is 'U' and belongs to the category 'washing powder'.

SQL:

```
Create table product (product_code Varchar(3) primary key, product_name  
Varchar(15), category Varchar(20), quantity int, price int);
```

```
insert into product values ('101', 'lexi', 'pen', 27, 5);
```

```
insert into product values ('102', 'colgate', 'paste', 8, 34);
```

```
insert into product values ('103', 'surfexel', 'washing powder', 40, 23);
```

```
insert into product values ('104', 'cello', 'pen', 20, 9);
```

```
insert into product values ('105', 'success', 'book', 50, 42);
```

```
Select * from product;
```

a) Select \* from product order by product-name desc;

b) Select product-code, product-name from product where price  
between 20 and 50;

Teacher's Signature : .....

Date.....

Expt. No.....

Page No.....

c) select product-name from product where category in ('soap', 'paste', 'pen');

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

e) select \* from product where product-name like 's%';

f) select \* from product where category not in ('paste');

g) select product-name from product where product-name like '%-ut%' and category in ('washing powder');

Teacher's Signature : .....

Expt. No.....

Output:

Table created.

1 row created.

product-code	product-name	category	quantity	price
101	Iexi	pen	27	5
102	colgate	paste	8	24
103	Surfexel	washing powder	40	23
104	Cello	pen	20	9
105	Success	book	50	42

(a)

product_code	product-name	category	quantity	price
103	Surfexel	washing powder	40	23
105	Success	book	50	45
101	Iexi	pen	27	5
102	colgate	paste	8	34
104	Cello	pen	20	9

(b)

product-code	product-name
102	colgate
103	Surfexel
105	Success

Date.....

Page No.....

Expt. No.....

(c) product-name

Lexi

colgate

cello

(d)

product-code	product-name	category	quantity	price
101	lexi	pen	27	5
102	colgate	paste	8	34
103	surfexel	washing powder	40	23
104	cello	pen	20	9
105	success	book	50	42

(e)

product-code	product-name	category	quantity	price
103	Surfexel	washing powder	40	23
105	success	book	50	42

(f)

product-code	product-name	category	quantity	price
101	lexi	pen	27	5
103	Surfexel	washing powder	48	23
104	cello	pen	20	9
105	success	book	50	42

(g) product-name

surfexel

Date.....

Expt. No.....

Page No.....

Ques:

create table supplies (supcode, sname, city)

create table product (pcode, pname)

create table sup\_product (supcode, pcode, qty)

a) Get all pairs of supplier numbers such that the two suppliers are located in the same city.

b) Get supplier names for suppliers who supply product p2.

c) Get product numbers supplied by more than one supplier.

d) Get supplier numbers for suppliers, who are located in the same city as supplier 1.

e) Get supplier names for supplier who supply part p1.

f) Get the number of suppliers, who are supplying at least one product.

g) For each product suppleid, get the pcode, and the total quantity suppleid for that part.

SQL:

```
Create table Supplier (supcode varchar(3) primary key, Sname varchar(30),  
city varchar(30));
```

```
create table product (pcode varchar(3) primary key, pname varchar(30));
```

```
create table suplproduct (supCode varchar(3), pcode varchar(3), qty int);
```

```
insert into Supplier values ('s1', 'Jumna', 'calicut');
```

```
insert into Supplier values ('s2', 'Niyas', 'malappuram');
```

```
insert into Supplier values ('s3', 'Nithya', 'calicut');
```

```
insert into Supplier values ('s4', 'Amira', 'malappuram');
```

```
insert into Supplier values ('s5', 'Basith', 'thrissur');
```

Teacher's Signature : .....

Date.....

Expt. No.....

Page No.....

insert into product values ('p1', 'pen');  
insert into product values ('p2', 'pencil');  
insert into product values ('p3', 'book');  
insert into product values ('p4', 'bag');

insert into suppl-product values ('s1', 'p1', 20);  
insert into suppl-product values ('s1', 'p2', 15);  
insert into suppl-product values ('s2', 'p1', 30);  
insert into suppl-product values ('s2', 'p2', 40);  
insert into suppl-product values ('s3', 'p4', 80);

- a) Select F.sname as supplier1, S.sname as supplier2 from supplier F, supplier S  
where F.city = S.city and F.supcode <> S.supcode;
- b) Select Sname from supplier, suppl-product where suppl-product.pcode = 'p2'  
and suppl-product.supcode = supplier.supcode;
- c) Select suppl-product.pcode, count(suppl-product.pcode) from suppl-product group by  
suppl-product.pcode having count(suppl-product.supcode) > 1;
- d) Select supcode from supplier where city = (select city from supplier where supcode = 's1');
- e) Select supplier.sname from supplier, suppl-product where suppl-product.supcode =  
supplier.supcode and suppl-product.pcode = 'p1';
- f) Select supplier.supcode from supplier where supplier.supcode in (select distinct  
supcode from suppl-product);
- g) Select pcode, qty from suppl-product;

Teacher's Signature : .....

Date.....

Expt. No.....

Page No.....

Output:

Table created.

Table created.

Table created

Sup-code	Sname	city
S1	Jamnas	calicut
S2	Niyas	malappuram
S3	Nithya	calicut
S4	Athira	malappuram
S5	basith	thrissur.

Pcode	Pname
P1	Pen
P2	pencil
P3	book
P4	bag

Supcode	Pcode	Qty
S1	P1	20
S1	P2	15
S2	P1	30
S2	P2	40
S3	P4	80

(C)

Supplier 1	Supplier 2
Nithya	Jamnas
Athira	Niyas
Jamnas	Nithya
Niyas	Athira

(b) Sname

Jamnas

Niyas

(c) 

Pcode	Count(Supl-product.pcode)
-------	---------------------------

P1	2
P2	2

(d) Supcode

S1

S3

(e) Sname

Jamnas

Niyas

(f) Supcode

S1

S2

S3

(g)

Pcode	Qty
P1	20
P2	15
P1	30
P2	40
P4	80

Date.....

Expt. No. .... 7 .....

Page No.....

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.

SQL:

create table carea (radius number(10), area number(10,4));

declare

rad int := 3;

area number(10,4);

begin

loop

area := 3.14 \* rad \* rad;

insert into carea values (rad, area);

rad := rad + 1;

exit when rad > 7;

end loop;

end;

/

Select \* from area;

Teacher's Signature : .....

Date.....

Expt. No.....

Page No.....

Output:

Table created.

Procedure successfully completed.

Radius	Area
3	28.26
4	50.24
5	78.5
6	113.04
7	153.86

Date.....

Expt. No....8.....

Page No.....

Aim:

Create table phonebook (pname, mobno)

Create a trigger to insert the old records from the table phonebook to del-phonebook (pname, mobno, modify-date) whenever a record is deleted or updated in the phonebook table.

SQL:

Create table phonebook (pname varchar(25), mobno varchar(10));

Insert into phonebook values ('Aithira', '1234567890');

Insert into phonebook values ('Nithya', '2345678901');

Insert into phonebook values ('Basith', '3456789012');

Select \* from phonebook;

Create table del-phonebook (pname varchar(25), mobno varchar(10),  
modify-date date);

Create or replace function old-records() returns trigger

language plpgsql

as \$\$

declare

begin

Insert into del-phonebook values (old.pname, old.mobno, current\_timestamp);

return new;

end;

\$\$;

1

Create trigger clou after delete or update on phone book for each row execute  
procedure old-records();

Delete from phonebook where pname = 'Aithira';

Update phonebook set mobno = '8765432190' where pname = 'Nithya';

Select \* from phonebook;

Teacher's Signature : .....

Date.....

Expt. No.....

Page No.....

output:

Table created.

1 row created.

1 row created.

1 row created.

Pname	mobno
Athira	1234567890
Nithya	2345678901
Basim	3456789012

1 row deleted.

Pname	mobno
Nithya	2345678901
Basim	3456789012

1 row updated.

Pname	mobno
Nithya	8765432100
Basim	3456789012

Date.....

Expt. No. 9.....

Page No.....

Aim:

Consider the employee database given below. Given 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 name 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. 10000.

d) find the employees who live in the same cities as the companies for which they work.

e) Find the employees who do not work in wipro corporation.

f) find the company that has the most employees.

SQL:

```
Create table employee (employee-name varchar(20), city varchar(20));
```

```
insert into employee values ('anupama', 'cochin');
```

```
insert into employee values ('gayathri', 'pune');
```

```
insert into employee values ('nubla', 'bengaluru');
```

```
Select * from employee;
```

```
Create table works (employee-name varchar(20), company-name varchar(30),  
Salary int);
```

```
insert into works values ('anupama', 'wipro', 15000);
```

```
insert into works values ('gayathri', 'infosys', 25000 25000);
```

```
insert into works values ('nubla', 'wipro', 22000);
```

```
Select * from works;
```

Teacher's Signature : .....

Date.....

Expt. No.....

Page No.....

```
create table company (company-name varchar(30), city varchar(20));  
insert into company values ('wipro', 'bengaluru');  
insert into company values ('infosys', 'bengaluru');  
select * from company;
```

```
create table manages (employee-name varchar(20), manager-name varchar(20));  
insert into manages values ('anupama', 'diya');  
insert into manages values ('gayathri', 'yadu');  
insert into manages values ('nubla', 'nourbina');  
select * from manages;
```

- a) Select employee-name from works where company-name = 'infosys';
- b) Select employee.employee-name, employee.city from employee, works where employee.employee-name=works.employee-name and works.company-name = 'wipro';
- c) Select employee.employee-name, employee.city from employee, works where employee.employee-name = works.employee-name and works.company-name = 'Infosys' and salary > 10000;
- d) Select employee.employee-name from employee, works, company where company.company-name = works.company-name and works.employee-name = employee.employee-name and company-city = employee.city;
- e) Select \* from works where company-name not in ('wipro');
- f) Select company-name from works group by company-name having count(distinct employee-name) >= all (select count(distinct employee-name) from works group by company-name);

Teacher's Signature : .....

Expt. No.....

Output

• Table created.

1 row created

1 row created

1 row created

employee-name	city
anupama	cochin
gayathri	pune
nubla	bengaluru

• Table created.

1 row created

1 row created

1 row created

employee-name	company-name	salary
anupama	wipro	15000
gayathri	infosys	25000
nubla	wipro	22000

• Table created.

1 row created

1 row created.

company-name	city
wipro	bengaluru
infosys	bengaluru

• Table created.

1 row created

1 row created.

1 row created.

employee-name	manager-name
anupama	cliya
gayathri	yadu
nubla	noorbin

Expt. No.....

a) employee-name

gayathri

b)

employee-name	city
anupama	cochin
nubla	bengaluru

c)

employee-name	city
gayathri	pune.

d) employee-name

nubla

e)

employee-name	company-name	Salary
gayathri	infosys	25000

f) company-name

wipro

Date.....

Expt. No.....<sup>10</sup>

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.

FieldName	Type	width
EmpNo	character	10
Name	characters	20
Basic-pay	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 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.

Teacher's Signature : .....

Expt. No.....

Page No.....

SQL:

create table salaries (EmpNo varchar(3) primary key, Name varchar(10) not null, Basic-pay int, DA int, Gross-Salary int, PF int, Net-Salary int, Annual-Salary int, Tax int);

insert into salaries (EmpNo, Name, Basic-pay) values ('101', 'Amira', 15);

insert into salaries (EmpNo, Name, Basic-pay) values ('102', 'Nithya', 190);

insert into salaries (EmpNo, Name, Basic-pay) values ('103', 'Ganashyam', 2900);

insert into salaries (EmpNo, Name, Basic-pay) values ('104', 'Abhinand', 23000);

insert into salaries (EmpNo, Name, Basic-pay) values ('105', 'Abhiram', 260000);

insert into salaries (EmpNo, Name, Basic-pay) values ('106', 'Gumnaas', 13000);

insert into salaries (EmpNo, Name, Basic-pay) values ('107', 'Niyas', 1200);

insert into salaries (EmpNo, Name, Basic-pay) values ('108', 'Shuhail', 28);

insert into salaries (EmpNo, Name, Basic-pay) values ('109', 'Basith', 450);

insert into salaries (EmpNo, Name, Basic-pay) values ('110', 'Kiran', 250000);

Update Salaries set DA = (40 \* Basic-pay)/100;

Update Salaries set PF = (10 \* Basic-pay)/100;

Update Salaries set Gross-Salary = Basic-pay + DA;

update salaries set Net-Salary = Gross-Salary - PF;

Update Salaries set Annual-Salary = 12 \* NetSalary;

Update Salaries set Tax = 0 where Annual-Salary < 100000;

Teacher's Signature : .....

Date.....

Expt. No.....

Page No.....

Update salaries set Tax = ((annual-salary - 100000) \* 10) / 100 where annual-salary > 100000 and annual-salary <= 150000;

Update salaries set Tax = ((annual-salary - 150000) \* 20) / 100 where annual-salary > 150000 and annual-salary <= 250000;

Update salaries set Tax = ((annual-salary - 250000) \* 30) / 100 where annual-salary > 250000;

Select \* from salaries;

Teacher's Signature : .....

Date.....

Page No.....

Expt. No.....

Output:

Table created.

10 rows added.

empno	Name	Basic-pay	DA	Gross-Salary	PF	Net-Salary	Annual-Salary	Tax
101	Athira	15						
102	Nithya	190						
103	ganashyam	2900						
104	abhinand	23000						
105	abhiram	260000						
106	Jamnas	13000						
107	niyas	1200						
108	Basith	430						
109	shuhailb	28						
110	Kiran	250000						

10 rows updated.

empno	name	Basic-pay	DA	Gross-Salary	PF	Netsalary	Annual-Salary	Tax
101	Athira	15	6	104000	104000	104000	1040000	104000
102	Nithya	190	76	104000	104000	104000	1040000	104000
103	ganashyam	2900	1160	104000	104000	104000	1040000	104000
104	abhinand	23000	1160	104000	104000	104000	1040000	104000
105	abhiram	260000	104000	104000	104000	104000	1040000	104000
106	Jamnas	1300	5200	104000	104000	104000	1040000	104000
107	niyas	1200	480	104000	104000	104000	1040000	104000
108	Basith	480	112	104000	104000	104000	1040000	104000
109	shuhailb	28	11	104000	104000	104000	1040000	104000
110	Kiran	250000	100000	104000	104000	104000	1040000	104000

10 rows updated.

Date.....

Page No.....

Expt. No.....

Expt. No.....

empno	Name	Basic-pay	DA	Gross-Salary	PF	Net-salary	Annual-Salary	Tax
101	Athira	15	6		2	19		
102	Nithya	190	76		290			
103	ganashyam	2900	1160		2300			
104	abhinand	23000	9200		26000			
105	abhiram	260000	104000		1300			
106	Jumnaas	13000	5200		120			
107	Niyas	1200	480		43			
108	Basilh	430	172		3			
109	shuhaiib	28	11		26000			
110	Kiran	250000	100000					

10 rows updated.

empno	Name	Basic-pay	DA	Gross-Salary	PF	Net-salary	Annual-Salary	Tax
101	Athira	15	6	21	2			
102	Nithya	190	76	266	19			
103	ganashyam	2900	1160	4060	290			
104	abhinand	23000	9200	32200	2300			
105	abhiram	260000	104000	364000	26000			
106	Jumnaas	13000	5200	18200	1300			
107	Niyas	1200	480	1680	120			
108	Basilh	430	172	430				
109	shuhaiib	28	11	682	3			
110	Kiran	250000	100000	350000	25000			

10 rows updated.

empno	Name	Basicpay	DA	Gross salary	PF	Net salary	Annual salary	Tax
101	Athira	15	6	21	2	19		
102	Nithya	190	76	266	19	241		
103	ganashyam	2900	1160	4060	290	3770		
104	abhinand	23000	9200	32200	2300	29900		
105	abhiram	260000	104000	364000	260000	338000		

SQL:

Create

Basic-pay

Tax int

insert in

insert

insert

insert

insert

insert

ins

inse

ins

ins

UP

C

Date.....

Expt. No.....

Page No.....

106	Jamnas	13000	5200	18200	1300	16900			
107	Niyas	1200	480	1680	120	1560			
108	Basidh	430	172	602	43	559			
109	Shuhaiib	28	11	39	3	36			
110	Kiran	250000	100000	350000	25000	325000			

10 rows updated.

empno	Name	Basic-pay	DA	Gross-Salary	PF	Net-Salary	Annual-Salary	Tax
101	Athira	15	6	21	2	19	228	
102	Nijhya	190	76	266	19	247	2964	
103	ganashyam	2900	1160	1160	4060	290	42540	
104	abhinand	23000	9200	32200	2300	29900	358800	
105	abhiram	260000	104000	364000	26000	3380000	4056000	
106	Jamnas	13000	5200	18200	1300	16900	202800	
107	Niyas	1200	480	1680	120	1560	18720	
108	Basidh	430	172	602	43	559	6708	
109	Shuhaiib	28	11	39	3	36	432	
110	Kiran	250000	100000	350000	25000	325000	390000	

8 rows updated.

empno	Name	Basic-pay	DA	Gross-Salary	PF	Net-Salary	Annual-Salary	Tax.
101	Athira	15	6	21	2	19	228	0
102	Nijhya	190	76	266	19	247	2964	0
103	ganashyam	2900	1160	4060	290	3770	42540	0
104	abhinand	23000	9200	32200	2300	29900	358800	0
105	abhiram	260000	104000	364000	26000	3380000	4056000	0
106	Jamnas	13000	5200	18200	1300	16900	202800	0
107	Niyas	1200	480	1680	120	1560	18720	0
108	Basidh	430	172	602	43	559	6708	0
109	Shuhaiib	28	11	39	3	36	432	0
110	Kiran	250000	100000	350000	25000	325000	390000	0

Page No.....

Expt. No.....

0 rows updated

1 row updated

empno	Name	Basic-pay	DA	Gross-Salary	PF	Net-salary	Annual-Salary	Tax
101	anira	15	8	21	2	19	228	0
102	Nithya	190	76	266	19	247	2964	0
103	ganashyam	2900	1160	4060	290	3770	45240	0
104	abhinand	23000	9200	32200	2300	29900	358800	0
105	abhiram	260000	104000	364000	26000	338000	4056000	
106	Jannas	13000	5200	18200	1300	16900	202800	10560
107	Niyas	1200	480	1680	120	1560	18720	0
108	Basil	430	172	602	43	559	6708	0
109	Shuhail	28	11	39	3	36	432	0
110	Kiran	250000	100000	350000	2500	325000	390000	

3 rows updated.

empno	Name	Basic-pay	DA	Gross-Salary	PF	Net-salary	Annual-Salary	Tax
101	anira	5	6	21	2	19	228	0
102	Nithya	190	76	266	19	247	2964	0
103	ganashyam	2900	1160	4060	290	3770	45240	0
104	abhinand	23000	9200	32200	2300	29900	358800	32640
105	abhiram	260000	10400	364000	26000	338000	4056000	1141800
106	Jannas	13000	5200	18200	13000	16900	202800	10560
107	Niyas	1200	480	1680	120	1590	18720	0
108	Basil	430	172	602	43	559	6708	0
109	Shuhail	28	11	39	3	36	432	0
110	Kiran	250000	100000	350000	25000	325000	390000	1095000