**EXPERIMENT-1**

**SQL TO LEARN DDL COMMANDS**

**Aim:-** Create a table emp as shown below.

**FIELD NAME DATA TYPE SIZE**

emp\_no number 3

ename varchar 20

job varchar 20

age number 2

salary number 4

dept\_no number 2

1. Create a constraint specifying emp\_no as the primary key which specifies that empno. should be a unique value with no duplicates.
2. Create another constraint which specifies that Ename does not contain null value.
3. Create a constraint specifying age being between 18 to 60.

**Use the above table to perform the following queries.**

1. Create a table empp\_dept having field emp\_no, ename, dept\_no from emp.
2. Add the field join\_date which is a field that can hold a dete and it specify the joining date of employee.
3. Modify the field salary to hold a maximum of 10 numbers .
4. Rename the table emp to empp33.
5. Delete the column age.
6. Delete all records from empp33.
7. Drop the table empp33.

**Solution**

1.Create a table emp as shown below.

**FIELD NAME DATA TYPE SIZE**

emp\_no number 3

ename varchar 20

job varchar 20

age number 2

salary number 4

dept\_no number 2

SQL> create table emp(emp\_no number(3),ename varchar(20) ,job varchar(20),age number(2) ,salary number(4),dept\_no number(2));

Table created.

a)Create a constraint specifying emp\_no as the primary key which specifies that empno. should be a unique value with no duplicates.

SQL> alter table emp add constraint pri primary key(emp\_no);

Table altered.

b)Create another constraint which specifies that Ename does not contain null value.

SQL> alter table emp modify ename not null;

Table altered.

c)Create a constraint specifying age being between 18 to 60.

SQL> alter table emp add constraint test check(18<age<60);

Table altered.

SQL>desc emp;

Name Null? Type

----------------------------------------------------- -------- ------------------------------------

EMP\_NO NOT NULL NUMBER(3)

ENAME NOT NULL VARCHAR2(20)

JOB VARCHAR2(20)

AGE NUMBER(2)

SALARY NUMBER(4)

DEPT\_NO NUMBER(2)

**2.Use the above table to perform the following queries**:-

a)create a table empp\_dept having field emp\_no, ename, dept\_no from emp.

SQL> create table empp\_dept as select emp\_no,ename,dept\_no from emp;

Table created.

SQL>desc empp\_dept;

Name Null? Type

----------------------------------------------------- -------- ------------------------------------

EMP\_NO NUMBER(3)

ENAME NOT NULL VARCHAR2(20)

DEPT\_NO NUMBER(2)

b)Add the field join\_date which is a field that can hold a date and it specify the joining date of employee.

SQL> alter table empp3 add join\_date date;

Table altered.

SQL>desc empp3;

Name Null? Type

----------------------------------------------------- -------- ------------------------------------

EMP\_NO NOT NULL NUMBER(3)

ENAME NOT NULL VARCHAR2(20)

JOB VARCHAR2(20)

AGE NUMBER(2)

SALARY NUMBER(4)

DEPT\_NO NUMBER(2)

JOIN\_DATE DATE

c)Modify the field salary to hold a maximum of 10 numbers .

SQL> alter table empp3 modify salary number(10);

Table altered.

SQL>desc empp3;

Name Null? Type

----------------------------------------------------- -------- --------------------------------

EMP\_NO NOT NULL NUMBER(3)

ENAME NOT NULL VARCHAR2(20)

JOB VARCHAR2(20)

AGE NUMBER(2)

SALARY NUMBER(10)

DEPT\_NO NUMBER(2)

JOIN\_DATE DATE

d)rename the table empp3 to empp33.

SQL> rename empp3 to empp33;

Table renamed.

e)delete the column age.

SQL> alter table empp33 drop column age;

Table altered.

SQL>desc empp33;

Name Null? Type

----------------------------------------------------- -------- ------------------------

EMP\_NO NOT NULL NUMBER(3)

ENAME NOT NULL VARCHAR2(20)

JOB VARCHAR2(20)

SALARY NUMBER(10)

DEPT\_NO NUMBER(2)

JOIN\_DATE DATE

f)Delete all records from empp33.

SQL> truncate table empp33;

Table truncated.

g)Drop the table empp33.

SQL> drop table empp33;

Table dropped.

**Result:-**The above DDL commands in SQL has been successfully executed.

**EXPERIMENT-2**

**SQL TO LEARN DML COMMANDS**

**Aim:-**Create the table emp as shown below:

**FIELD NAME DATA TYPE SIZE**

emp\_no number 3

ename varchar 20

job varchar 20

salary number 2

dept\_no number 2

age number 2

Create a constraint specifying emp\_no as primary key.

Create another constraint on ename such that it should not contain a null value.

Insert some records in the above table by direct insertion and parameter insertion.

**Use the above table to peform the following queries:**

a)Display the details emp\_no,ename,salary of employees drawing salary between 1500 and 2500.

b)Write the query to find out how many different job titles are stored in the emp relation.

c)How many employees are titled with a given job name.

d)Calculate the total salary of the employees.

e)Calculate the average salary of employees.

f)List the minimim and maximum salary of employees.

g)Determine how many records are there in the table.

h)The employee name ‘john’ is transferred to department no 20 and his salary is increased by Rs. 1000.

i)All employees who are working in dept no 10 and 30 will get 15% increase in salary.

j)Increase the salary of all employees by 25%.

k)Delete employees having emp\_no 5.

l)Delete employees having emp\_no 4 and dept\_no 7.

m)Find the second largest employee in the above table table.

**Solution**

1)Create the table emp. Create a constraint specifying emp\_no as primary key.

Create another constraint on ename such that it should not contain a null value.

SQL> create table emp (emp\_no number(3), enamevarchar(20) not null, jobvarchar(20), salary number(4), dept\_no number(2), age number(2), primary key(emp\_no) 9 );

Table created.

SQL>desc emp;

Name Null? Type

----------------------------------------------------- -------- ------------------------------------

EMP\_NO NOT NULL NUMBER(3)

ENAME NOT NULL VARCHAR2(20)

JOB VARCHAR2(20)

SALARY NUMBER(4)

DEPT\_NO NUMBER(2)

AGE NUMBER(2)

2)Insert some records in the above table by direct insertion and parameter insertion.

SQL> insert into emp values(1,'shefali','manager',2000,7,20);

1 row created.

SQL> insert into emp values(&emp\_no,'&ename','&job',&salary,&dept\_no,&age);

Enter value for emp\_no: 5

Enter value for ename: kanika

Enter value for job: ceo

Enter value for salary: 1300

Enter value for dept\_no: 10

Enter value for age: 21

old 2: values(&emp\_no,'&ename','&job',&salary,&dept\_no,&age)

new 2: values(5,'kanika','ceo',1300,10,21)

1 row created.

SQL> /

Enter value for emp\_no: 4

Enter value for ename: anamika

Enter value for job: engg

Enter value for salary: 1000

Enter value for dept\_no: 7

Enter value for age: 21

old 2: values(&emp\_no,'&ename','&job',&salary,&dept\_no,&age)

new 2: values(4,'anamika','engg',1000,7,21)

1 row created.

SQL> /

Enter value for emp\_no: 6

Enter value for ename: shivam

Enter value for job: engg

Enter value for salary: 1000

Enter value for dept\_no: 30

Enter value for age: 21

old 2: values(&emp\_no,'&ename','&job',&salary,&dept\_no,&age)

new 2: values(6,'shivam','engg',1000,30,21)

1 row created.

SQL> /

Enter value for emp\_no: 9

Enter value for ename: john

Enter value for job: secretary

Enter value for salary: 1700

Enter value for dept\_no: 8

Enter value for age: 21

old 2: values(&emp\_no,'&ename','&job',&salary,&dept\_no,&age)

new 2: values(9,'john','secretary',1700,8,21)

1 row created.

SQL> select \* from emp;

EMP\_NO ENAME JOB SALARY DEPT\_NO AGE

--------- -------------------- -------------------- --------- --------- ---------

1 shefali manager 2000 7 20

5 kanika ceo 1300 10 21

4 anamika engg 1000 7 21

6 shivam engg 1000 30 21

9 john secretary 1700 8 21

3) Display the details emp\_no,ename,salary of employees drawing salary between 1500 and 2500.

SQL> select emp\_no,ename,salary from emp where salary between 1500 and 2500;

EMP\_NO ENAME SALARY

--------- -------------------- ---------

1 shefali 2000

9 john 1700

4)Write the query to find out how many different job titles are stored in the emp relation.

SQL> select count(distinct job) from emp;

COUNT(DISTINCTJOB)

------------------

4

5)How many employees are titled with a given job name.

SQL> select job,count(emp\_no) from emp group by job;

JOB COUNT(EMP\_NO)

-------------------- -------------

ceo 1

engg 2

manager 1

secretary 1

6)Calculate the total salary of the employees.

SQL> select sum(salary) "total salary" from emp;

total salary

------------

7000

7)Calculate the average salary of the employees.

SQL> select avg(salary) "average salary" from emp;

average salary

--------------

1400

8)calculate the max and min salary of the employees.

SQL> select max(salary) "Max salary",min(salary) "min salary" from emp;

max salary min salary

---------- ----------

2000 1000

9)Determine how many records are there in the table.

SQL> select count(emp\_no) from emp;

COUNT(EMP\_NO)

-------------

5

10)The employee name ‘john’ is transferred to department no 20 and his salary is increased by Rs. 1000.

SQL> update emp set dept\_no=20,salary=salary+1000 where ename='john';

1 row updated.

SQL> select \* from emp;

EMP\_NO ENAME JOB SALARY DEPT\_NO AGE

--------- -------------------- -------------------- --------- --------- ---------

1 shefali manager 2000 7 20

5 kanika ceo 1300 10 21

4 anamika engg 1000 7 21

6 shivam engg 1000 30 21

9 john secretary 2700 20 21

11)All employees who are working in dept no 10 and 30 will get 15% increase in salary.

SQL> update emp set salary=salary+0.15\*salary where dept\_no=10 or dept\_no=30;

2 rows updated.

SQL> select \* from emp;

EMP\_NO ENAME JOB SALARY DEPT\_NO AGE

--------- -------------------- ------------ --------- --------- ---------

1 shefali manager 2000 7 20

5 kanika ceo 1495 10 21

4 anamika engg 1000 7 21

6 shivam engg 1150 30 21

9 john secretary 2700 20 21

12)Increase the salary of all employees by 25%.

SQL> update emp set salary=salary+salary\*0.25;

5 rows updated.

SQL> select \* from emp;

EMP\_NO ENAME JOB SALARY DEPT\_NO AGE

--------- - ------------------- --------------- --------- --------- ---------

1 shefali manager 2500 7 20

5 kanika ceo 1869 10 21

4 anamika engg 1250 7 21

6 shivam engg 1438 30 21

9 john secretary 3375 20 21

13)Delete employees having emp\_no 4 and dept\_no 7.

SQL> delete from emp where emp\_no=4 and dept\_no=7;

1 row deleted.

SQL> select \* from emp;

EMP\_NO ENAME JOB SALARY DEPT\_NO AGE

--------- -------------------- ------------- --------- --------- ---------

1 shefali manager 2500 7 20

5 kanika ceo 1869 10 21

6 shivam engg 1438 30 21

9 john secretary 3375 20 21

14)Delete employees having emp\_no 5.

SQL> delete from emp where emp\_no=5;

1 row deleted.

SQL> select \* from emp;

EMP\_NO ENAME JOB SALARY DEPT\_NO AGE

--------- -------------------- ----------- --------- --------- ---------

1 shefali manager 2500 7 20

6 shivam engg 1438 30 21

9 john secretary 3375 20 21

15)Select second max salary from the relation.

SQL> select max(salary) "second max" from emp where salary<(select max(salary) from emp\_48);

second max

----------

2500

**Result**-The above DML commands has been succefully executed in SQL.

**EXPERIMENT-3**

**SQL TO LEARN BASIC SELECT COMMANDS**

**Aim:-** Create a table emp as shown below.

**FIELD NAME DATA TYPE SIZE**

Emp\_Id number 3

First\_name varchar 20

Middle\_name varchar 20

Last\_name varchar 20

Salary number 4

City varchar 20

Dno number 2

Create a constraint specifying emp\_Id as the primary key.

Create another constraint which specifies that First\_name and Last\_name does not contain null value.

**Use the above table to perform the following queries.**

1. Selecting all records from EMP and displaying all columns:
2. Selecting all records from EMP and displaying a specified column Emp\_Id:
3. Selecting all records from EMP and displaying multiple columns separated by commas:
4. Displaying data for a given condition: Selecting all records having Emp\_Id, First\_name and Last\_name from EMP table where Id is 03.
5. Displaying data for a given condition and sorting the output: Select Emp Id and Last name of Employee from EMP from Meerut city order by Emp Id;
6. Displaying data for a given condition and sorting the output on multiple columns, one column sorted in reverse order: Select Emp Id and First name of Employee from EMP from Meerut city order by Emp Id in ascending order and order by Last name in descending order.
7. Displaying data for a given condition and sorting the output using an integer in the place of the spelled-out column name:
8. Displaying data for a given condition and sorting the output by multiple columns using integers, the order of the columns in the sort is different than their corresponding order after the SELECT keyword:
9. Counting the number of records in the Emp table:
10. Counting the number of values for Emp\_ID in the Emp table:
11. Select the Last name of the Employee and rename the column name as Lname using column aliases.
12. Display the Last name of the employee twice, giving the second column an alias named Lname.

**Solution**

1.Create a table emp as shown below.

**FIELD NAME DATA TYPE SIZE**

Emp\_Id number 3

First\_name varchar 20

Last\_name varchar 20

City varchar 15

SQL> create table emp(emp\_Id number(3),First\_name varchar(20) ,Last\_name varchar(20),city varchar(15);

Table created.

1. Create a constraint specifying emp\_Id as the primary key.
2. Create another constraint which specifies that First\_name does not contain null value.

SQL> alter table emp add constraint pri primary key(emp\_Id);

Table altered.

b)Create another constraint which specifies that First\_name does not contain null value.

SQL> alter table emp modify First\_name not null;

Table altered.

1. Selecting all records from EMP and displaying all columns:

Select \* from EMP;

1. Selecting all records from EMP and displaying a specified column Emp\_Id:

Select Emp\_Id from EMP;

1. Selecting all records from EMP and displaying multiple columns separated by commas:

Select Emp\_Id, Last\_name from EMP;

1. Displaying data for a given condition: Selecting all records having Emp\_Id, First\_name and Last\_name from EMP table where Id is 03.

Select Emp\_Id, First\_name, Last\_name from EMP where Emp\_Id=’03’;

1. Displaying data for a given condition and sorting the output: Select Emp Id and Last name of Employee from EMP from Meerut city order by Emp Id;

Select Emp\_Id, Last\_name from EMP where city=’Meerut’ order by Emp\_Id;

1. Displaying data for a given condition and sorting the output on multiple columns, one column sorted in reverse order: Select Emp Id and First name of Employee from EMP from Meerut city order by Emp Id in ascending order and order by Last name in descending order.

Select Emp\_Id, Last\_name from EMP where city=’Meerut’ order by Emp\_Id, Last\_name Desc;

1. Displaying data for a given condition and sorting the output using an integer in the place of the spelled-out column name:

Select Emp\_Id, Last\_name from EMP where city=’Meerut’ order by 1;

1. Displaying data for a given condition and sorting the output by multiple columns using integers, the order of the columns in the sort is different than their corresponding order after the SELECT keyword:

Select Emp\_Id, Last\_name from EMP where city=’Meerut’ order by 2, 1;

1. Counting the number of records in the Emp table:

Select count(\*) from EMP;

1. Counting the number of values for Emp\_ID in the Emp table:

Select count(Emp\_Id) from EMP;

1. Select the Last name of the Employee and rename the column name as Lname using column aliases.

Select Last\_Name Lname from Emp;

1. Display the Last name of the employee twice, giving the second column an alias named Lname.

Select Last\_Name, Last\_Name Lname from Emp;

**Result:-**The above basic SELECT commands have been successfully executed.

**EXPERIMENT-4**

**ADVANCED SELECT COMMANDS**

**Aim:-** Create a table items as shown below.

FIELD NAME DATA TYPE SIZE

Order\_num number 3

Number number 3

Total\_price number 5

Use the above table to perform the following queries.

1. Write a query that select count of number and sum of total price of items in each order from items relation.
2. Write a query that collects the rows of the items table into groups that have identical order numbers and computes the count of rows in each group and the sum of the prices.
3. Write a query that collects the rows of the items table into groups that have identical order numbers and computes the count of rows in each group and the sum of the prices sorted in ascending order of price.
4. Write a query that returns the average total price per item on all orders that have more than two items.
5. Write a query that returns the average of all total price values in the table items.

**Solution**

**Aim:-** Create a table items as shown below.

**FIELD NAME DATA TYPE SIZE**

Order\_num number 3

Number number 3

Total\_price number 5

1. Write a query that select count of number and sum of total price of items in each order from items relation.

Select order\_num, count(\*) number, sum(total\_price) Tot\_Price from items Group by Order\_num;

1. Write a query that collects the rows of the items table into groups that have identical order numbers and computes the COUNT of rows in each group and the SUM of the prices.

Select order\_num, count(\*) number, sum(total\_price) Price from items Group by Order\_num;

1. Write a query that collects the rows of the items table into groups that have identical order numbers and computes the COUNT of rows in each group and the SUM of the prices sorted in ascending order of price.

Select order\_num, count(\*) number, sum(total\_price) Price from items Group by Order\_num order by price;

1. Write a query that returns the average total price per item on all orders that have more than two items.

Select order\_num, count(\*) number, avg (total\_price) Average from items Group by Order\_num having count(\*)>2;

1. Write a query that returns the average of all total\_price values in the table items.

Select avg(total\_price) Average from items having count(\*)>2;

**Result:-**The above advanced SELECT commands have been successfully executed.

**EXPERIMENT-5**

**SQL ILLUSTRATING CONSTRAINTS**

**Aim:-**Write the SQL commands illustrating various types of constraints including primary key, foreign key, not null, check constraint and default constraint.

**Solution**

1.Create table employee.

SQL> create table employee(e\_no number(20),e\_namevarchar(20),e\_salary number(10),e\_dobdate,e\_dept\_namevarchar(20));

Table created.

SQL> insert into employee (e\_no,e\_name,e\_salary,e\_dob,e\_dept\_name) values (1,'kanika',50000,'24-jan-1993','student');

1 row created.

2.Implementing constraints :-

Primary key:-

SQL> alter table employee add constraint pri primary key(e\_no);

Table altered.

SQL> insert into employee values ('&e\_no','&e\_name','&e\_salary','&e\_dob','&e\_dept\_name');

Enter value for e\_no: 1

Enter value for e\_name: s

Enter value for e\_salary: 34565646

Enter value for e\_dob: 6-oct-9000

Enter value for e\_dept\_name: dgfh

old 1: insert into employee values ('&e\_no','&e\_name','&e\_salary','&e\_dob','&e\_dept\_name')

new 1: insert into employee values ('1','s','34565646','6-oct-9000','dgfh')

insert into employee values ('1','s','34565646','6-oct-9000','dgfh')

\*

ERROR at line 1:

ORA-00001: unique constraint (SCOTT.PRI) violated

Unique constraint:-

SQL> alter table employee add constraint one unique(e\_name);

Table altered.

Enter value for e\_no: 2

Enter value for e\_name: kanika

Enter value for e\_salary: 34566

Enter value for e\_dob: 7-jul-2646

Enter value for e\_dept\_name: thj

old 1: insert into employee values ('&e\_no','&e\_name','&e\_salary','&e\_dob','&e\_dept\_name')

new 1: insert into employee values ('2','kanika','34566','7-jul-2646','thj')

insert into employee values ('2','a','34566','7-jul-2646','thj')

\*

ERROR at line 1:

ORA-00001: unique constraint (SCOTT.ONE) violated

Check constraint:-

SQL> alter table employee add constraint test check(e\_salary>3000);

Table altered.

Enter value for e\_no: 3

Enter value for e\_name: gh

Enter value for e\_salary: 5

Enter value for e\_dob: 5-jan-9000

Enter value for e\_dept\_name: fgfg

old 1: insert into employee values ('&e\_no','&e\_name','&e\_salary','&e\_dob','&e\_dept\_name')

new 1: insert into employee values ('3','gh','5','5-jan-9000','fgfg')

insert into employee values ('3','gh','5','5-jan-9000','fgfg')

\*

ERROR at line 1:

ORA-02290: check constraint (SCOTT.TEST) violated

Not null:-

SQL> alter table employee modify e\_dob not null;

Table altered.

Enter value for e\_no: 3

Enter value for e\_name: hj

Enter value for e\_salary: 6789987

Enter value for e\_dob:

Enter value for e\_dept\_name: thjj

old 1: insert into employee values ('&e\_no','&e\_name','&e\_salary','&e\_dob','&e\_dept\_name')

new 1: insert into employee values ('3','hj','6789987','','thjj')

insert into employee values ('3','hj','6789987','','thjj')

\*

ERROR at line 1:

ORA-01400: cannot insert NULL into ("SCOTT"."EMPLOYEE"."E\_DOB")

Default constraint:-

SQL> alter table employee modify e\_dept\_name default 'none';

Table altered.

SQL> insert into employee (e\_no,e\_name,e\_salary,e\_dob) values (6,'shefali',5676454,'7-may-1992');

1 row created.

SQL> select \* from employee;

E\_NO E\_NAME E\_SALARY E\_DOB E\_DEPT\_NAM

----------- --------- --------- ---------- ----------

1 kanika 500000 24-JAN-93 student

6 shefali 5676454 07-MAY-92 none

Foreign key:-

SQL> create table dept(dept\_no number(10),dept\_namevarchar(20));

Table created.

SQL>descdept;

Name Null? Type

----------------------------------------------------- -------- ------------------------------------

DEPT\_NO NUMBER(10)

DEPT\_NAME VARCHAR2(20)

SQL> alter table dept add constraint depn primary key(dept\_name);

Table altered.

SQL> alter table deptadd foreign key(dept\_name) references employee(e\_dept\_name);

Table altered.

**Result:-**The program to illustrate constraints in SQL has been executed successfully.

**EXPERIMENT-6**

**JOIN OPERATIONS**

**Aim:-**Write a program to perform various SQL queries implementing Cartesian product, inner join and outer join.

**Solution**

SQL> create table employee(e\_no number(20) primary key,enamevarchar(20),dob date,salary number(10),dno number(10));

Table created.

SQL> create table dept(dno number(10) primary key,dnamevarchar(10),dlocvarchar(10));

Table created.

SQL>insert into employee values(1,'kanika','24-jan-1993', 15000,2);

1 row created.

SQL>insert into employee values(2,'shefali','6-may-1992', 12000,3);

1 row created.

SQL>insert into deptvalues(2,'ceo','del');

1 row created.

SQL>insert into deptvalues(3,'ceo','chen');

1 row created.

1. Cross join/Cartesian product;

SQL>select \* from employee,dept;

ENO ENAME DOB SALARY DNO DNO DNAME DLOC

--------- ---------- --------- --------- ----------------- --------- ---------- ----------

1 kanika 24-JAN-93 15000 2 2 ceo del

2 shefali 05-MAY-92 12000 3 2 ceo del

1 kanika 24-JAN-93 15000 2 3 ceo chen

2 shefali 05-MAY-92 12000 3 3 ceo chen

1. Inner join:-
2. Equi join-

SQL>select ename from employee,dept where employee.dno=dept.dno;

ENAME

----------

kanika

shefali

1. Theta join-

SQL> select \* from employee,dept where employee.dno<dept.dno;

ENO ENAME DOB SALARY DNO DNO DNAME DLOC

--------- ------------------- ------------ -------- ------- ---------- ----------

1 kanika 24-JAN-93 15000 2 3 ceo chen

SQL> select \* from employee,dept where employee.dno>dept.dno;

ENO ENAME DOB SALARY DNO DNO DNAME DLOC

--------- ---------- ---------- -------- --------- --------- ---------- ----------

2 shefali 05-MAY-92 12000 3 2 ceo del

SQL> select \* from employee,dept where employee.dno>=dept.dno;

ENO ENAME DOB SALARY DNO DNO DNAME DLOC

--------- ---------- --------- --------- --------- --------- ---------- ----------

1 kanika 24-JAN-93 15000 2 2 ceo del

2 shefali 05-MAY-92 12000 3 2 ceo del

2 shefali 05-MAY-92 12000 3 3 ceo chen

SQL> select \* from employee,dept where employee.dno<=dept.dno;

ENO ENAME DOB SALARY DNO DNO DNAME DLOC

--------- ---------- --------- --------- --------- --------- ---------- ----------

1 kanika 24-JAN-93 15000 2 2 ceo del

1 kanika 24-JAN-93 15000 2 3 ceo chen

2 shefali 05-MAY-92 12000 3 3 ceo chen

SQL> select \* from employee,dept where employee.dno!=dept.dno;

ENO ENAME DOB SALARY DNO DNO DNAME DLOC

--------- ---------- --------- --------- --------- --------- ---------- ----------

2 shefali 05-MAY-92 12000 3 2 ceo del

1 kanika 24-JAN-93 15000 2 3 ceo chen

1. Natural join-

SQL> selectemployee.eno, employee.ename, employee.salary, employee.dno, employee.dob, dept.dname,dept.dloc

fromemployee,dept where employee.dno=dept.dno;

ENO ENAME SALARY DNO DOB DNAME DLOC

--------- --------- --------- --------- --------- ---------- ----------

1 kanika 15000 2 24-JAN-93 ceo del

2 shefali 12000 3 05-MAY-92 ceo chen

1. Outer join:-
2. Left outer join-

SQL> select \* from employee,dept where employee.dno(+)=dept.dno;

ENO ENAME DOB SALARY DNO DNO DNAME DLOC

--------- ---------- --------- --------- --------- --------- ---------- ----------

1 kanika 24-JAN-93 15000 2 2 ceo del

2 shefali 05-MAY-92 12000 3 3 ceo chen

1. Right outer join-

SQL> select \* from employee,dept where employee.dno=dept.dno(+);

ENO ENAME DOB SALARY DNO DNO DNAME DLOC

--------- ---------- --------- --------- --------- --------- ---------- ----------

1 kanika 24-JAN-93 15000 2 2 ceo del

2 shefali 05-MAY-92 12000 3 3 ceo chen

1. Full outer join-

SQL> select \* from employee,dept where employee.dno(+)=dept.dno UNION select \* from employee,dept where employee.dno=dept.dno(+);

ENO ENAME DOB SALARY DNODNO DNAME DLOC

--------- ---------- --------- --------- --------- --------- ---------- ----------

1 kanika 24-JAN-93 15000 2 2 ceo del

2 shefali 05-MAY-00 12000 3 3 ceo chen

**Result:-**The program to perform join operations in SQL has been executed successfully.

**EXPERIMENT-7**

**SQL FUNCTIONS**

**Aim:- Create a table Emp as shown below:**

Field Name Data Type Number

Id number 2

Name varchar 20

Work\_Date date

daily\_typing\_pages number 5

Insert some of the records in the above table so that it represents like below:

Id Name Work\_Date daily\_typing\_pages

1 John 2007-01-24 250

2 Ram 2007-05-27 220

3 Jack 2007-05-06 170

3 Jack 2007-04-06 100

4 Jill 2007-04-06 220

5 Zara 2007-06-06 300

5 Zara 2007-02-06 350

Write the following Queries:

1. Select number of employees.
2. Select number of employees having the name as John.
3. Write a query to fetch maximum value of daily\_typing\_pages.
4. Write a query to fetch maximum and minimum value of daily\_typing\_pages.
5. Write a query to calculate average of all the dialy\_typing\_pages.
6. Write a query to calculate average of all the records related to a single person.
7. Write a query to calculate total of all the dialy\_typing\_pages.
8. Write a query to sum up all the records related to a single person.
9. Write a query to find out square root of any number say 16.
10. Write a query to calculate square root of all the dialy\_typing\_pages.
11. Write a query to generate some random numbers between 0 and 1.
12. Write a query to display the data of table in any order.
13. Write a query to concat two strings ‘FIRST’ and ‘SECOND’
14. Write a query to concatenate all the names, employee ID and work\_date.

**Solution**

1. Select number of employees.

Select count(\*) from emp;

Count(\*)

----------

7

1. Select number of employees having the name as John.

Select count(\*) from emp where name=’john’;

Count(\*)

----------

2

1. Write a query to fetch maximum value of daily\_typing\_pages.

Select max(daily\_typing\_pages) from emp;

max(daily\_typing\_pages)

--------

350

1. Write a query to fetch maximum and minimum value of daily\_typing\_pages.

Select min(daily\_typing\_pages) Least,max(daily\_typing\_pages) Maximum from emp;

min(daily\_typing\_pages) max(daily\_typing\_pages)

-------- --------

100 350

5. Write a query to calculate average of all the dialy\_typing\_pages.

Select avg(daily\_typing\_pages) from emp;

avg(daily\_typing\_pages)

--------



6.Write a query to calculate average of all the records related to a single person.

Select name, avg(daily\_typing\_pages) from emp group by name;

Name avg(daily\_typing\_pages)

-------- --------

Jack 135.0000

Jill 220.0000

John 250.0000

Ram 220.0000

Zara 325.0000

7. Write a query to calculate total of all the dialy\_typing\_pages.

Select sum(daily\_typing\_pages) from emp;

sum(daily\_typing\_pages)

--------

1610

8. Write a query to sum up all the records related to a single person.

Select name, sum(daily\_typing\_pages) from emp group by name;

Name sum(daily\_typing\_pages)

-------- -------

Jack 270

Jill 220

John 250

Ram 220

Zara 650

9. Write a query to find out square root of any number say 16.

Select sqrt(16);

sqrt(16)

-------

4

10.Write a query to calculate square root of all the dialy\_typing\_pages.

Select sqrt(16);

Name sqrt(dialy\_typing\_pages)

------- --------

John 15.811388

Ram 14.832397

Jack 13.038405

Jack 10.000000

Jill 14.832397

Zara 17.320508

Zara 18.708287

11.Write a query to generate some random numbers between 0 and 1.

Select rand(), rand(), rand();

Rand() Rand() Rand()

0.4566351518 0.17271283816 0.26431273471

12.Write a query to display the data of table in any order.

Select \* from emp order by rand();

13. Write a query to concat two strings ‘FIRST’ and ‘SECOND’

Select concat(‘FIRST’,’SECOND’);

14. Write a query to concatenate all the names, employee ID and work\_date.

Select concat(id, name, work\_date) from emp;

concat(id, name, work\_date)

------

1John2007-01-24

2ram2007-05-27

3Jack

**Result:-**The program to illustrate SQL Functions has been executed successfully.

**EXPERIMENT-8**

**SQL SUB QUERIES**

**Aim:- Create a table Customers as shown below:**

Field Name Data Type Number

Id number 2

Name varchar 20

Age date

Address varchar 25

Salary number 6

Insert the records as shown below:

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

| ID | NAME | AGE | ADDRESS | SALARY |

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

| 1 | Ramesh | 35 | Ahmedabad | 2000.00 |

| 2 | Khilan | 25 | Delhi | 1500.00 |

| 3 | kaushik | 23 | Kota | 2000.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 6 | Komal | 22 | MP | 4500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

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

Write the following queries:

1. Write a query to display the Id of customers having salary greater than 4500.
2. Write a query to updates SALARY by 0.25 times in the CUSTOMERS table for all the customers whose AGE is greater than or equal to 27.
3. Write a query to delete the records from the CUSTOMERS table for all the customers whose AGE is greater than or equal to 30.
4. Select the id and name of employees who are not residing in Delhi and Mumbai.
5. Consider that another table customer\_detail has the same structure as of customer, then insert the same recirds in tanle customer\_detail.

**Solution**

1. Write a query to display the Id of customers having salary greater than 4500.

Select \* from customers where id IN (select id from customers where salary>4500);

1. Write a query to updates SALARY by 0.25 times in the CUSTOMERS table for all the customers whose AGE is greater than or equal to 27.

Update customers set salary=salary\*0.25 where age IN (select age from customers where age>=27);

1. Write a query to delete the records from the CUSTOMERS table for all the customers whose AGE is greater than or equal to 30.

Delete from customers where age IN (select age from customers where age>=30);

1. Select the id and name of employees who are not residing in Delhi and Mumbai.

Select Id, name from customers where address NOT IN (‘Delhi’, ‘Mumbai’);

1. Consider that another table customer\_detail has the same structure as of customer, then insert the same recirds in tanle customer\_detail.

Insert into customer\_detail select \* from customers wher id IN (select id from customers);

**Result:-**The program to illustrate SQL Sub Queries has been executed successfully.

**EXPERIMENT-9**

**SET OPERATIONS**

**Aim:-**Write a program to perform various SQL queries implementing set operations.

**Solution**

SQL> create table set1( no number(20),name varchar(20),college varchar(10));

Table created.

SQL> create table set2( no number(20),namevarchar(20),college varchar(10));

Table created.

SQL> insert into set1 values(1,'kanika','srm');

1 row created.

SQL> insert into set1 values(2,'shefali','srm');

1 row created.

SQL> insert into set2 values ( 1,'kanika','srm');

1 row created.

SQL> insert into set2 values (3,'shweta','srm');

1 row created.

1. Union:-

SQL> select \* from set1 UNION select \* from set2 ;

NO NAME COLLEGE

--------- --------------- ----------

1 kanika srm

2 shefali srm

3 shweta srm

1. Intersection:-

SQL> select \* from set1 INTERSECT select \* from set2 ;

NO NAME COLLEGE

--------- -------------- ----------

1 kanika srm

1. Difference:-

SQL> select \* from set1 minus select \* from set2 ;

NO NAME COLLEGE

--------- -------------- ----------

2 shefali srm

SQL> select \* from set2 minus select \* from set1 ;

NO NAME COLLEGE

--------- -------------- ----------

3 shweta srm

SQL> create table set3( no number(20),name varchar(20),college varchar(10));

Table created.

SQL> create table set4( num number(20),namesvarchar(20),college varchar(10));

Table created.

SQL> insert into set3 values(1,'kanika','srm');

1 row created.

SQL> insert into set3 values(2,'shefali','srm');

1 row created.

SQL> insert into set4 values ( 1,'kanika','srm');

1 row created.

SQL> insert into set4 values (3,'shweta','srm');

1 row created.

1. Union:-

SQL> selectno,name from set3 union select num as no ,names as name from set4;

NO NAME

--------- --------------------

1 kanika

2 shefali

3 shweta

1. Intersection:-

SQL> selectno,name from set3 intersect select num as no ,names as name from set4;

NO NAME

--------- --------------------

1. kanika
2. Difference:-

SQL> selectno,name from set3 minus select num as no ,names as name from set4;

NO NAME

----- --------------

2 Shefali

SQL> selectno,name from set4 minus select num as no ,names as name from set3;

NO NAME

----- --------------

3 Shweta

**Result:-**The program to perform set operations in SQL has been executed successfully.

**EXPERIMENT-10**

**VIEWS**

**Aim:-**Create a table named Emp having the following fields

Name Data Type Size

Empno number 3

Ename varchar 20

Job varchar 15

Salary number 6

Deptno number 2

Perform the following queries:

1. The organization wants to display only the details of the employees those who are ASP. (Horizontal Partitioning)
2. The organization wants to display only the details like empno, empname, deptno, deptname of the employees. (Vertical portioning)
3. Display all the views generated.
4. Execute the DML commands on the view created.
5. Drop a view.

**Solution**

1. The organization wants to display only the details of the employees those who are ASP. Create a view on emp table named managers

Use select from clause to do horizontal portioning

SQL> create view empview as select \* from emp where job='ASP';

View created.

SQL> select \* from empview;

EMPNO ENAME JOB DEPTNO SAL

---------- -------------------- ---------- ---------- ----------

2 Arjun ASP 2 12000

3 Gugan ASP 2 20000

2. The organization wants to display only the details like empno, empname, deptno, deptname of the employees. (Vertical portioning)

Create a view on emp table named general

Use select from clause to do vertical partioning

SQL> create view empview1 as select empno, empname, deptno, deptname from emp;

View created.

3.Display all the views generated.

SQL> select \* from tab;

TNAME TABTYPE CLUSTERID

---------- ------- ----------

DEPT TABLE

EMP TABLE

EMPVIEW VIEW

EMPVIEW1 VIEW

4. Execute the DML commands on the view created.

SQL> select \* from empview;

EMPNO ENAME JOB DEPTNO SAL

---------- -------------------- ---------- ---------- ----------

2 Arjun ASP 2 12000

1. Gugan ASP 2 20000

5. Drop a view.

SQL> drop view empview1;

View dropped.

**Result:-**The program to illustrate views has been executed successfully.

**EXPERIMENT-11**

**PL/SQL FOR CALCULATING AREA OF CIRCLE**

**Aim:-**Create a table areas having two fields as radius and area. Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 5-10 and then store the radius and the corresponding value of calculated area in the table.

Create table areas having two fields

**FIELD NAME DATA TYPE SIZE**

Radius number 3

Area number 20,3

**Solution**

SQL> create table areas(radius number(3),area number(20,3));

Table created.

**PL/SQL code block**

SQL> declare

radius number(3);

area number(20,3);

pi constant number(3,2):=3.14;

begin

radius:=5;

while(radius<=10)

loop

area:=pi\*radius\*radius;

insert into areas values(radius,area);

radius:=radius+1;

end loop;

end;

/

PL/SQL procedure successfully completed.

**Input and Output:**

SQL> select \* from areas;

RADIUS AREA

--------- ---------

5 78.5

6 113.04

7 153.86

8 200.96

9 254.34

10 314

6 rows selected.

**Result:-**The program to calculate areas of circle having radius from 5 to 10 has been written in PL/SQL has been executed successfully.

**EXPERIMENT-12**

**PL/SQL FOR CALCULATING FACTORIAL OF A NUMBER**

**Aim:-**Write a program to calculate factorial of number in PL/SQL.

**Solution**

**PL/SQL code block**

SQL> set serveroutput on;

SQL> declare

fact number(10):=1;

n number(10);

begin

n:=&n;

while(n>=1)

loop

fact:=fact\*n;

n:=n-1;

end loop;

dbms\_output.put\_line('Answer ='||fact);

end;

/

**Input and Output:**

Enter value for n: 4

old 5: n:=&n;

new 5: n:=4;

Answer =24

PL/SQL procedure successfully completed.

**Result:-**The given program to calculate factorial of a number was successfully executed in PL/SQL.

**EXPERIMENT-13**

**AFTER ROW TRIGGER**

**Aim:-**Create table stud\_master as shown below:-

**FIELD NAME DATA TYPE SIZE**

rollno number 2

name varchar 20

age number 2

phno number 10

city varchar 20

Create table audit\_student as shown below:-

**FIELD NAME DATA TYPE SIZE**

rollno number 2

name varchar 20

operation varchar 15

dateofupdate date

Create a transparent audit system for the table stud\_master . The system must keep track of the records that are being deleted or updated along with operation and the date on which the operation is performed in the table audit\_student.

**PL/SQL code block**

1. Create table stud\_master as shown below:-

**FIELD NAME DATA TYPE SIZE**

rollno number 2

name varchar 20

age number 2

phno number 10

city varchar 20

SQL> create table stud\_master(rollno number(2),name varchar(20),age number(2),phno number(10),city varchar(20));

Table created.

1. Create table audit\_student as shown below:-

**FIELD NAME DATA TYPE SIZE**

rollno number 2

name varchar 20

operation varchar 15

dateofupdate date

SQL> create table audit\_student(rollno number(2),name varchar(20),operation varchar(15),dateofudp date);

Table created.

1. Create a transparent audit system for the table stud\_master . The system must keep track of the records that are being deleted or updated along with operation and the date on which the operation is performed in the table audit\_student.

SQL>create or replace trigger kk after insert or delete or update on stud\_master

2 for each row

3 declare

4 operationvarchar(15);

5 rollno number(2);

6 namevarchar(20);

7 begin

8 if inserting then operation:='insert';

9 end if;

10 if updating then operation:='update';

11 end if;

11 if deleting then operation:='delete';

12 end if;

13 rollno:=:old.rollno;

14 name:=:old.name;

15 insert into audit\_student300 values(rollno,name,operation,sysdate);

16 end;

17 /

Trigger created.

**Input and output**

Insert:-

SQL>insert into stud\_mastervalues('1','kanika','19','9758966442','indore');

1 row created.

Update:-

SQL>update stud\_master set name=’kanu’ where rollno=1;

1 row updated.

Delete:-

SQL>delete from stud\_master where rollno=1;

1 row deleted.

SQL>select \* from audit\_student;

ROLLNO NAME OPERATION DATEOFUPDATE

------------ --------- ----------------- ------------------------

1 kanika insert 03-mar-14

1 kanika update 03-mar-14

1 kanu delete 03-mar-14

**Result:-**The given program to implement ‘after trigger’ has been executed successfully