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.

Registration No:

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)

Registration No:

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.

Registration No:

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;

Registration No:

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.

Registration No:

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

Registration No:

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

Registration No:

```
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
```


Registration No:

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.

Registration No:

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;

Registration No:

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

Registration No:

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

Registration No:

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.

Registration No:

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.

Registration No:

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.

a) Create a constraint specifying emp_Id as the primary key.
b) 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;

2) Selecting all records from EMP and displaying a specified column Emp_Id:

Select Emp_Id from EMP;

3) Selecting all records from EMP and displaying multiple columns separated by commas:

Select Emp_Id, Last_name from EMP;

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.

Select Emp_Id, First_name, Last_name from EMP where Emp_Id=’03’;

Registration No:

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;
Select Emp_Id, Last_name from EMP where city=’Meerut’ 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.
Select Emp_Id, Last_name from EMP where city=’Meerut’ order by Emp_Id, Last_name Desc;

7) 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;

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:
Select Emp_Id, Last_name from EMP where city=’Meerut’ order by 2, 1;

9) Counting the number of records in the Emp table:
Select count(*) from EMP;

10) Counting the number of values for Emp_ID in the Emp table:
Select count(Emp_Id) from EMP;

11) Select the Last name of the Employee and rename the column name as Lname using column aliases.
Select Last_Name Lname from Emp;

12) Display the Last name of the employee twice, giving the second column an alias named Lname.
Select Last_Name, Last_Name Lname from Emp;

Registration No:

Result:-The above basic SELECT commands has 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.

Registration No:

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;

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.

Select order_num, count(*) number, sum(total_price) Price from items Group by Order_num;

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.

Select order_num, count(*) number, sum(total_price) Price from items Group by Order_num order by price;

4. 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;

5. 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;

Registration No:

Result:-The above advanced SELECT commands has 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.

Registration No:

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

Registration No:

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

Registration No:

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.

Registration No:

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.

Registration No:

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

2) Inner join:-

a) Equi join-

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

ENAME

---

kanika

shefali

b) Theta join-

Registration No:

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;

Registration No:

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

c) 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

3) Outer join:-

a. 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

b. 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

c. 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

Registration No:

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

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.

Registration No:

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.

Registration No:

Solution

1. Select number of employees.

```
Select count(*) from emp;
Count(*)
----------
7
```

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

```
Select count(*) from emp where name='john';
Count(*)
----------
2
```

3. Write a query to fetch maximum value of daily_typing_pages.

```
Select  max(daily_typing_pages) from emp;
max(daily_typing_pages)
--------
350
```

4. 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;
```


Registration No:

avg(daily_typing_pages)

230

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

Registration No:

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();

Registration No:

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

Registration No:

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:

Registration No:

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.

Registration No:

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);

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.

Update customers set salary=salary*0.25 where age IN (select age from customers where age>=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.

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

4. 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');

5. 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.

Registration No:

EXPERIMENT-9
SET OPERATIONS

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

Registration No:

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
```

2) Intersection:-

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

        NO NAME       COLLEGE
--------- -------------- ----------
         1 kanika          srm
```

3) 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.
```


Registration No:

```
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
   ```

2. Intersection:-

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

        NO NAME
   --------- --------------------
        1     kanika
   ```

3. 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.

Registration No:

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.

Registration No:

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
```


Registration No:

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

2	Arjun	ASP	2	12000
3	Gugan	ASP	2	20000

5. Drop a view.

SQL> drop view empview1;

View dropped.

Registration No:

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

Registration No:

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
```


Registration No:

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.

Registration No:

EXPERIMENT-12

PL/SQL FOR CALCULATING FACTORIAL OF A NUMBER

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

Registration No:

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.
```


Registration No:

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

Registration No:

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.

2. Create table audit_student as shown below:-

FIELD NAME	DATA TYPE	SIZE
rollno	number	2

Registration No:

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.

3. 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.
```


Registration No:

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
```


Registration No:

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

Registration No: