

Expt no.: 1

DATABASE CREATION

The SQL "create database" statement is used to create a new SQL database.

Creating a basic table involves naming the table and defining its columns and each column's data type.

The SQL CREATE TABLE statement is used to create a new table.

CREATE TABLE is the keyword that telling the database system what you want to do. In this case, you want to create a new table.

The unique name or identifier follows the CREATE TABLE statement.

A copy of an existing table can be created using a combination of the CREATE TABLE statement and the SELECT statement.

You can check the complete details at create table

1. Table name : Programmer

Creation:

```
SQL> create table programmer  
      (pname varchar2(20) not null,  
       dob date not null,  
       DOJ date not null,  
       gender char(1) notnull,  
       prof1 varchar2(10),  
       prof2 varchar2(10),  
       salary number(7,2) not null);
```

Data insertion:

```
SQL> insert into programmer values  
2 ('&pname', '&dob', '&doj', '&gender', '&profi',  
'&prof2', '&salary');
```

2. Name of the table: studies

Table creation:

```
SQL> create table studies  
(pname varchar2(20) not null,  
splace varchar2(20) not null,  
course varchar2(20) not null,  
cost number(5));
```

Row creation:

```
SAL> insert into studies values  
2 ('&pname', '&splace', '&course', '&cost');
```

3. Name of the table: software

Table creation:

```
SQL> create table software  
(pname varchar2(20) not null,  
title varchar2(20) not null,  
DEV-D varchar2(10) not null,  
scost number (10,2) not null,  
Dcost number (10,2) not null,  
sold number (4));
```

Row creation:

```
SAL> insert into software values  
2 ('&pname', '&title', '&dev-d', &scost, &Dcost,  
&sold);
```

SQL

SQL is a standard language for accessing and manipulating databases.

→ SQL stands for structured query language

→ SQL lets you access and manipulate databases

SQL is required:

- To execute queries against a database
- To retrieve data from a database
- To insert records in a database
- To update records in a database
- To delete records from a database
- To create new databases
- To set permissions on tables, procedures and views.

→ Every table is broken up into smaller entities called fields.

→ A field is a column in a table that is designed to maintain specific information about every record in the table.

→ A record, also called a row, is each individual entry that exists in a table.

→ A record is a horizontal entity in a table

Update Operations:

Insert - inserts new data into a database

Delete - deletes data from a database

Update - updates data in a database.

1. Findout the selling cost for the package(s) developed in pascal.

```
SQL> select  
      scost  
    from software  
   where dev_d = 'Pascal';
```

2. Display the names and ages of programmers.

```
SQL> select  
      pname, (sysdate - dob) / 365  
    from programmer;
```

3. Display the names of programmers who done the DAP course.

```
SQL> select  
      pname  
    from studies  
   where course = 'DAP';
```

4. What is the highest number of copies sold by a package.

```
SQL> select  
      max(sold)  
    from software;
```

5. Display the names and date of birth of all programmers born in January.

```
SQL> select  
      phame, dob  
    from programmer  
   where dob like '%.JAN%';
```

6. Display the lowest course fee

```
SQL> select  
      min(cost)  
    from studies;
```

7. How many programmers have done the PGDCA course.

```
SQL> select  
      count(*)  
    from studies  
   where course = 'PGDCA';
```

8. How much revenue has been earned through the sale of packages developed in c.

```
SQL> Select  
      sum(sold * scost)  
    from software  
   where dev_d = 'c';
```

9. Display the details of software developed by Ramesh.

SQL > select
*
from software
where pname='Ramesh';

10. How many programmers have studied at Sabhari.

SQL > select
count(*)
from studies
where splace='Sabhari';

11. Display the details of packages whose sales have been crossed 2000 mark.

SQL > select
*
from programmer
where salary >= 2000;

12. Findout the number of copies, which should be sold in-order to recover the development cost of each package.

SQL > select
title, (dcost/scost)
from software;

13. Display the details of packages for which development cost have been recovered.

```
SQL> select  
*  
from software  
where (sold * scost) - dcost >= 0;
```

14. What is the price of the costlier software developed in Basic.

```
SQL> select  
max(scost)  
from software  
where dev_d = 'Basic';
```

15. How many packages were developed in dbase

```
SQL> select  
count(*)  
from software  
where dev_d = 'Dbase';
```

16. How many programmers studied in Pragathi

```
SQL> select  
count(*)  
from studies  
where splace = 'Pragathi';
```

17. How many programmers paid 5000-10000 for their course

```
SQL> select  
      count(*)  
      from studies  
     where cost >= 5000  
       and cost <= 10000;
```

18. What is the average course fee.

```
SQL> select  
      avg(cost)  
      from studies;
```

19. Display details of programmers knowing C.

```
SQL> select  
      *  
      from programmer  
     where prof1 = 'C'  
       or prof2 = 'C';
```

20. How many programmers know either cobol or pascal.

```
SQL> select  
      count(*)  
      from programmer  
     where prof1 = 'Cobol'  
       or prof1 = 'Pascal'  
       or prof2 = 'Cobol'  
       or prof2 = 'Pascal';
```

21. How many programmers don't know Pascal &c.

```
SQL> select  
      count(*)  
    from programmer  
   where prof1 != 'C'  
     and prof1 != 'Pascal'  
     and prof2 != 'C'  
     and prof2 != 'Pascal';
```

22. How old is the oldest male programmer.

```
SQL> select  
      pname  
    from programmer  
   where gender = 'M'  
     and sysdate - dob = ( select  
                           max(sysdate - dob)  
                         from programmer  
                        where gender = 'M' );
```

23. What is the average age of female programmers.

```
SQL> select  
      avg(sysdate - dob)  
    from programmer  
   where gender = 'F';
```

24. Calculate the experience in years for which each programmer and display along with names in descending order.

```
SQL> select  
      pname, (sysdate - doj) / 365  
    from programmer  
   Order by (sysdate - doj) / 365 desc;
```

25. Who are the programmers who celebrate their birthdays in the current month.

```
SQL> select
      pname
    from programmer
   where substr(dob,4,3) = (select
                                substr(sysdate,4,3)
                              from dual);
```

26. What are the languages known by male programmers.

```
SQL> select
      prof1
    from programmer
   where gender = 'M'
 Union
 Select
      prof2
    from programmer;
```

27. What is the average salary of programmer.

```
SQL> select
      avg(salary)
    from programmer;
```

28. How many male programmers are there

```
SQL> select
      count(*)
    from programmer
   where gender = 'M';
```

29. How many people draw 2000-4000

SQL> select

```
count(*)  
from programmer  
where salary >= 2000  
and salary <= 4000;
```

30. Display the details of programmers who don't know fortran, cobol or pascal

SQL> select

```
phame  
from programmer  
where prof1 not in ('Fortran', 'Cobol', 'Pascal')  
and prof2 not in ('Fortran', 'Cobol', 'Pascal');
```

31. In which month did most of the programmers join.

SQL> select

```
substr(doj, 4, 3), count(*)  
from programmer  
group by substr(doj, 4, 3)  
having count(*) = (select  
max(count(*))  
from programmer  
group by substr(doj, 4, 3));
```

32. In which language are most of programmers proficient.

SQL > select

```
prof1, count(*)  
from (select  
prof1  
from programmer  
Union all)
```

group by prof1

having count(*) = (select

```
max(count(*))  
from (select  
prof1  
from programmer  
Union all)  
group by prof1);
```

33. Who are the female programmers earning more than the highest paid male programmers.

SQL > select

```
*  
from programmer  
where salary > (select
```

```
max(salary)  
from programmer  
where gender = 'M')
```

and gender = 'F';

34. Which language have been stated as prof1 by most of the programmers.

```
SQL> Select  
      prof1, count(*)  
    from programmer  
   group by prof1  
 having count(*) = (select  
                      max(count(*))  
                    from programmer  
                   group by prof1);
```

Part-II

1. Display the number of packages developed in each language.

```
SQL> select  
dev_d, count(title)  
from software  
group by dev_d;
```

2. Display the number of packages developed by each person.

```
SQL> select  
pname, count(*)  
from software  
group by pname;
```

3. Display the number of male and female programmers.

```
SQL> select  
gender, count(*)  
from programmer  
group by gender;
```

4. Display the costliest and highest selling package developed in each language.

```
SQL> select  
dev_d, max(scost), max(sold)  
from software  
group by dev_d;
```

5. Display the number of people born in each year.

SQL > select

```
substr(dob,7,4), count(*)  
from programmer  
group by substr(dob,7,4);
```

6. Display the number of people joined in each year.

SQL > select

```
substr(doj,7,4), count(*)  
from programmer  
group by substr(doj,7,4);
```

7. Display the number of people born in each month.

SQL > select

```
substr(dob,1,2), count(*)  
from programmer  
group by substr(dob,1,2);
```

8. Display the number of people joined in each month.

SQL > select

```
substr(doj,1,2), count(pname)  
from programmer  
group by substr(doj,1,2);
```

9. Display language wise count of prof1

SQL > select

```
prof1, count(pname)  
from programmer  
group by prof1;
```

10. Display language wise count of prof2.

SQL > select

```
prof2, count(Pname)  
from programmer  
group by prof2;
```

11. Display the number of people in each language.

SQL > select

```
Salary, count(pname)  
from programmer  
group by salary;
```

12. Display the number of people studied in each institute.

SQL > select

```
splace, count(pname)  
from studies  
group by splace;
```

13. Display the number of people studied in each course.

SQL> select
course, count(pname)
from studies
group by course;

14. Display the total development cost of packages developed in each language.

SQL> select
dev-d, sum(dcost)
from software
group by dev-d;

15. Display the selling cost of packages developed in each language.

SQL> select
dev-d, sum(scost)
from software
group by dev-d;

16. Display the costs of packages developed by each programmer language wise.

SQL> select
pname, dev-d, scost
from software
order by dev-d, pname;

17. Display the number of packages sold by each programmer.

SQL > select

```
    Pname, sum(sold)
    from software
    group by Pname;
```

18. Display the sales cost of packages developed by each programmer.

SQL > select

```
    Pname, sum(scost * sold)
    from software
    group by Pname;
```

19. Display the sales cost of packages developed by each programmer language wise.

SQL > select

```
    Pname, dev-d, sum(sold * scost)
    from software
    group by Pname, dev-d;
```

20. Display the language name with average development cost, average selling cost and average price per copy.

SQL > select

```
    dev-d, avg(dcost), avg(scost)
    from software
    group by dev-d;
```

Q1. Display the programmers name, costliest/cheapest package developed by him/her

SQL > select
pname, max(dcost), min(dcost)
from software
group by pname;

Q2. Display each institute name with number of courses average cost per course.

SQL > select
splace, count(course), avg(cost)
from studies
group by splace;

Q3. Display each institute name with number of students

SQL > select
splace, count(pname)
from studies
group by splace;

Q4. Display the number of male and female programmers.

SQL > select
pname, gender
from programmer
order by gender desc;

25. Display the programmers name and their packages.

SQL > select

```
Pname, title  
from software;
```

26. Display the number of packages in each language except c and c++

SQL > select

```
dev_d, count(*)  
from software, group by dev_d  
having dev_d not in ('c', 'c++');
```

27. Display the number of packages in each language for which development cost is less than 1000.

SQL > select

```
dev_d, count(*)  
from software  
where dcost < 1000,  
group by dev_d;
```

28. Display the average difference between scost & dcost for each language.

SQL > select

```
dev_d, avg(scost - dcost)  
from software  
group by dev_d;
```

29. Display the total scost, dcost and amount to be recovered for each programmer for those whose dcost has not yet been recovered.

SQL> select

```
pname, sum(scost), sum(dcost),
sum(dcost - sold * scost)
from software
group by pname
having sum(Dcost - (sold * scost)) > 0;
```

30. Display the highest, lowest and average salaries for those earning more than 2000.

SQL> select

```
max(salary), min(salary), avg(salary)
from programmer
where salary > 2000;
```

31. Display the name of programmer(s) with highest salary.

SQL> select

```
pname
from programmer
where salary > (select
max(salary)
from programmer);
```