create table studSection

(

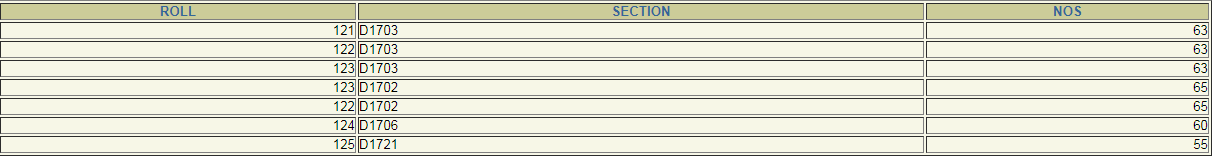
roll number(3),

section Varchar2(20),

nos number(2)

);

select \* from studSection



**GROUP BY CLAUSE**

This allows us to use simultaneous column name and group functions.

select max(nos),section from studSection group by Section

select count(nos),section from studSection group by Section

**HAVING CLAUSE**

This is used to specify conditions on rows retrieved by using group by clause.

select max(nos),section from studSection group by Section having count(\*) > 1

**select count(nos),section from studSection group by Section having count(\*) > 1**

EMPNO ENAME JOB DEPTNO SAL

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

1 Mathi AP 1 10000

2 Arjun ASP 2 15000

3 Gugan ASP 1 15000

4 Karthik Prof 2 30000

5 Akalya AP 1 10000

**Q1**: **Display all the details of the records whose employee name starts with ‘A’.**

Solution:

select \* from emp where ename like 'A%';

EMPNO ENAME JOB DEPTNO SAL

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

2 Arjun ASP 2 15000

5 Akalya AP 1 10000

**Q2: Display all the details of the records whose employee name does not starts with ‘A’.**

**Ans:**

select \* from emp where ename not like 'A%';

EMPNO ENAME JOB DEPTNO SAL

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

1 Mathi AP 1 10000

3 Gugan ASP 1 15000

4 Karthik Prof 2 30000

**Q3: Display the rows whose salary ranges from 15000 to 30000.**

select \* from emp where sal between 15000 and 30000;

EMPNO ENAME JOB DEPTNO SAL

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

2 Arjun ASP 2 15000

3 Gugan ASP 1 15000

4 Karthik Prof 2 30000

**Q4: Calculate the total and average salary amount of the emp table.**

select sum(sal),avg(sal) from emp;

SUM(SAL) AVG(SAL)

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

80000 16000

Q5: **Count the total records in the emp table.**

select count(\*) from emp;

COUNT(\*)

---------

5

**Q6: Determine the max and min salary and rename the column as max\_salary and**

**min\_salary.**

select max(sal) as max\_salary, min(sal) as min\_salary from emp;

MAX\_SALARY MIN\_SALARY

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

30000 10000

**Q9: Find how many job titles are available in employee table.**

select count(job) from emp;

COUNT(JOB)

----------

4

select count(distinct job) from emp;

COUNT(DISTINCTJOB)

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

2

**Q10: What is the difference between maximum and minimum salaries of employees in the**

**organization?**

**:**

SQL> select max(sal), min(sal) from emp;

MAX(SAL) MIN(SAL)

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

20000 10000

**Nested Queries:** Nesting of queries one within another is known as a nested

queries.

**Sub queries** The query within another is known as a sub query. A statement

containing sub query is called parent statement. The rows returned by sub query are

used by the parent statement.

**Types**

**1. Sub queries that return several values**

Sub queries can also return more than one value. Such results should be made use

along with the operators in and any.

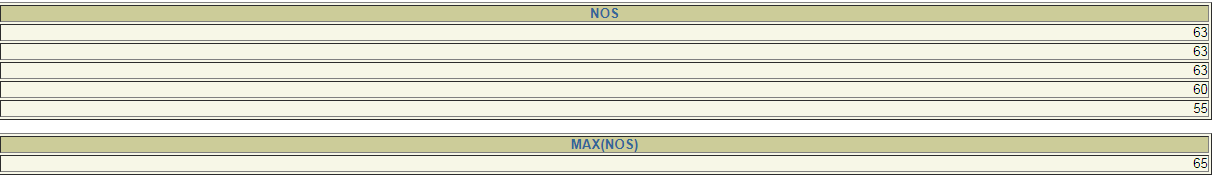
select roll from studSection where section = (Select section from studSection group by section having count(\*) > 2)

**2. Correlated sub query**

Correlated Sub query is evaluated once per row processed by the parent statement.

select nos from studSection where nos < (Select max(nos) from studSection );

Select max(nos) from studSection;



NOTE : A sub query is evaluated once for the entire parent statement.

**Nested Queries:**

**Example:** select ename, eno, address where salary > (select salary from employee where

ename =’jones’);

**1.Subqueries that return several values**

**Example:** select ename, eno, from employee where salary <any (select salary from

employee where deptno =10’);

**2.Correlated subquery**

**Example:** select \* from emp x where x.salary > (select avg(salary) from emp where deptno

=x.deptno);

**Queries:**

**Q1: Display all employee names and salary whose salary is greater than minimum salary of**

**the company and job title starts with ‘M’.**

Solution:

1. Use select from clause.

2. Use like operator to match job and in select clause to get the result.

**Ans:**

SQL> select ename,sal from emp where sal>(select min(sal) from emp where job like 'A%');

ENAME SAL

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

Arjun 12000

Gugan 20000

Karthik 15000

**Q2: Issue a query to find all the employees who work in the same job as Arjun.**

**Ans:**

SQL> select \* from emp;

EMPNO ENAME JOB DEPTNO SAL

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

1 Mathi AP 1 10000

2 Arjun ASP 2 12000

3 Gugan ASP 2 20000

4 Karthik AP 1 15000

SQL> select ename from emp where job=(select job from emp where ename='Arjun');

ENAME

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

Arjun

Gugan

**Q3: Issue a query to display information about employees who earn more than any**

**employee in dept 1.**

**Ans:**

SQL> select \* from emp where sal>(select max(sal) from emp where empno=1);

EMPNO ENAME JOB DEPTNO SAL

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

2 Arjun ASP 2 12000

3 Gugan ASP 2 20000

4 Karthik AP 1 15000

**Set Operators:**

The Set operator combines the result of 2 queries into a single result. The following

are the operators:

· Union · Union all

· Intersect · Minus

**The rules to which the set operators are strictly adhere to :**

· The queries which are related by the set operators should have a same number of

column and column definition.

· Such query should not contain a type of long.

· Labels under which the result is displayed are those from the first select statement.

**SQL commands:**

**Union:** Returns all distinct rows selected by both the queries

**Syntax:**

Query1 Union Query2;

**Union all:** Returns all rows selected by either query including the duplicates.

**Syntax:**

Query1 Union all Query2;

**Intersect:** Returns rows selected that are common to both queries.

**Syntax:**

Query1 Intersect Query2;

**Minus:** Returns all distinct rows selected by the first query and are not by the second

**Syntax:**

Query1 minus Query2;

**Queries:**

**Q1: Display all the dept numbers available with the dept and emp tables avoiding**

**duplicates.**

Solution:

1. Use select from clause. 2. Use union select clause to get the result.

**Ans:**

SQL> select deptno from emp union select deptno from dept;

DEPTNO

----------

1

2

12

30

40

Q2: **Display all the dept numbers available with the dept and emp tables.**

Solution:

1. Use select from clause. 2. Use union all in select clause to get the result.

Ans:

SQL> select deptno from emp union all select deptno from dept;

DEPTNO

----------

1

2

2

1

12

1

2

30

40

9 rows selected.

**Q3**: **Display all the dept numbers available in emp and not in dept tables and vice versa.**

Solution:

1. Use select from clause.

2. Use minus in select clause to get the result.

Ans:

SQL> select deptno from emp minus select deptno from dept;

DEPTNO

----------

12

SQL> select deptno from dept minus select deptno from emp;

DEPTNO

----------

30

40