DBMS LAB
Lab Assignment number 06

Name: Aamir Ansari **Batch:** A **Roll no.** 01

Aim: Experiment to study co-related queries and complex queries.

Theory:

Aggregate functions: -
Which takes set of values as input and returns single value as output.
Aggregate function generates summary value.

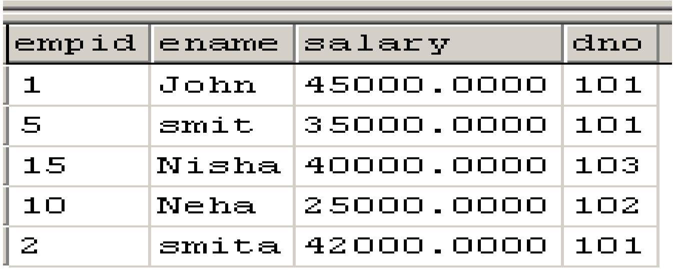
Sum():-The sum function returns the sum of values from given set.
Syntax:-
Select sum(distinct|all colname) from tablename
Default-all
Eg. Select sum(distinct salary) as totalsal from employee

Avg():-The Avg() function returns the average of values from given set.
Syntax:-
Select Avg(distinct|all colname) from tablename
Default-all

Eg. Select avg(salary) as Avgsal from employee

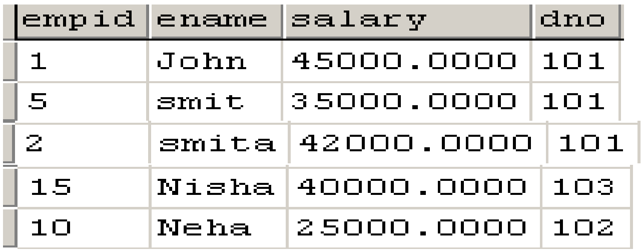
count():-The count() function returns number of values from given set.
Syntax:-
Select count(distinct|all colname) from tablename
Select count(*) from employee
↓
Number of records
Min():-The min() function returns minimum(smallest) value from given set.
Max():-The max() function returns maximum(largest) value from given set.

--All aggregate functions by default ignores null value.
Min(),Max(),count() can be used with numeric, character,date/time columns.
Group by clause:-
Group by clause is used to form group the records based on some criteria.
Group by clause applies aggregate function on subgroup of tuples from a relation where subgroups are formed on some column value.
Eg. group by colname
--Column name is called grouping column.



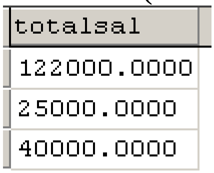
empid	ename	salary	dno
1	John	45000.0000	101
5	smit	35000.0000	101
15	Nisha	40000.0000	103
10	Neha	25000.0000	102
2	smita	42000.0000	101

--after grouping



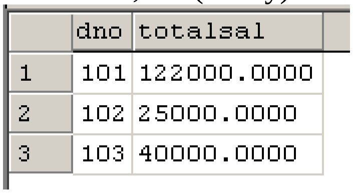
empid	ename	salary	dno
1	John	45000.0000	101
5	smit	35000.0000	101
2	smita	42000.0000	101
15	Nisha	40000.0000	103
10	Neha	25000.0000	102

Select sum(salary) as totalsal from emp group by dno



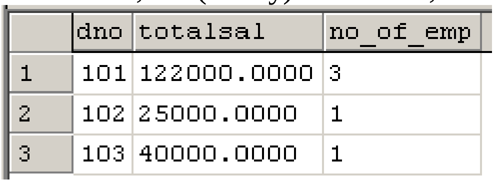
totalsal
122000.0000
25000.0000
40000.0000

select dno,sum(salary) as totalsal from emp group by dno



	dno	totalsal
1	101	122000.0000
2	102	25000.0000
3	103	40000.0000

Select dno,sum(salary) as totalsal, count(*) as no_of_emp from emp group by dno



	dno	totalsal	no_of_emp
1	101	122000.0000	3
2	102	25000.0000	1
3	103	40000.0000	1

--There are restrictions on columns that we can specify in the
select list.The only columns allowed in the select list are
Grouping columns(column that we included after group by clause.
Aggregate function.

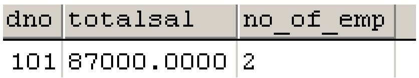
Select dno, salary as totalsal from emp group by dno --error

↓

(not aggregate function | not grouping column)

Eg. Display dno and number of employees for every
Department those are earning salary more than 40000.

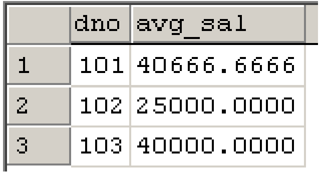
Select dno,count(empid) from emp where salary>40000
group by dno



dno	totalsal	no_of_emp
101	87000.0000	2

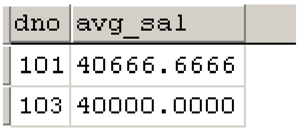
Having clause:-
Having clause is used to apply condition on group rather than tuple.
Where clause can not apply condition on group of value.

Eg. Display dno and average salary for every dept.
select dno,avg(salary) as avg_sal from emp group by dno



	dno	avg_sal
1	101	40666.6666
2	102	25000.0000
3	103	40000.0000

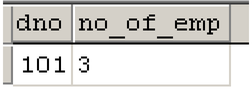
Eg. Display dno and average salary of departments where average salary of department is >=40000.
select dno,avg(salary) as avg_sal from emp group by dno having avg(salary)>=40000



dno	avg_sal
101	40666.6666
103	40000.0000

Eg. Display dno and no_of_emp of those depts where no_of_emp those are earning salary within range 30000 to 40000 are at least 2.

select dno,count(empid) as no_of_emp from emp where salary between 30000 and 45000
group by dno having count(empid)>=2

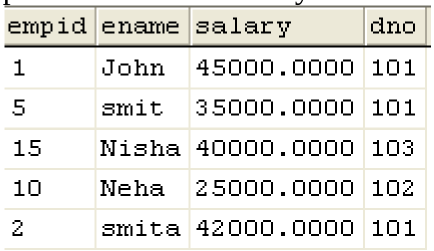


dno	no_of_emp
101	3

Where □ Group By □ Having

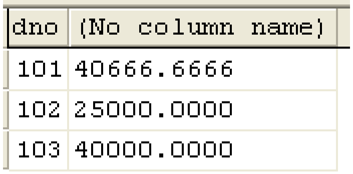
Co-related queries:-
Whenever a condition in the where clause of a nested query references to some attributes of a relation declared in the outer query then two queries are said to be co-related.
Correlated subquery runs once for each row selected by the outer query. It contains a reference to a value from the row selected by the outer query.

Eg. Display employee details of employees who are earning salary more than average salary of department for which they are working.



empid	ename	salary	dno
1	John	45000.0000	101
5	smit	35000.0000	101
15	Nisha	40000.0000	103
10	Neha	25000.0000	102
2	smita	42000.0000	101

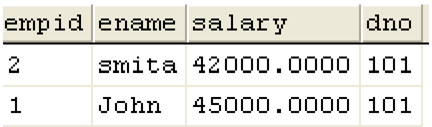
select dno,avg(salary) from emp group by dno



dno	(No column name)
101	40666.6666
102	25000.0000
103	40000.0000

select *from emp as e1 where e1.salary>

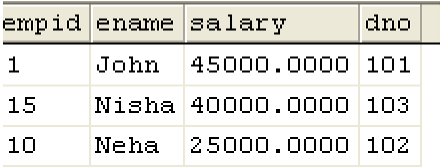
(select avg(salary) from emp as e2 where e1.dno=e2.dno)



empid	ename	salary	dno
2	smita	42000.0000	101
1	John	45000.0000	101

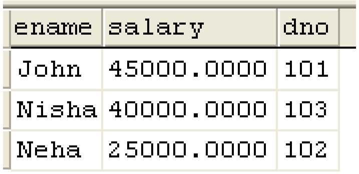
Nested subquery runs only once for the entire nesting (outer) query. It does not contain any reference to the outer query row.
Correlated subquery runs once for each row selected by the outer query

select *from emp as e1 where e1.salary>=all
(select salary from emp as e2 where e1.dno=e2.dno)



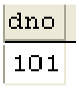
empid	ename	salary	dno
1	John	45000.0000	101
15	Nisha	40000.0000	103
10	Neha	25000.0000	102

select *from emp where salary
in (select max(salary) from emp group by dno)



ename	salary	dno
John	45000.0000	101
Nisha	40000.0000	103
Neha	25000.0000	102

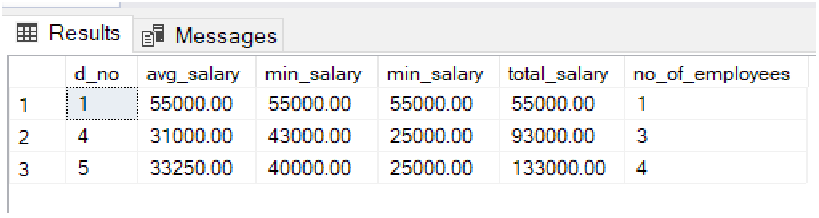
Exists and Not Exists:-
Exists and Not Exists operator is used to check where the result of co-related nested query is empty or not.
Exists:-
Exists operator return true if there is at least one tuple in the output of subquery.
Not Exists: -
Not Exists operator return true if there are no tuple in the output of subquery.(empty)
select *from dept as d where exists(select *from emp as e where e.dno=d.dno)
select *from dept as d where not exists(select *from emp as e where e.dno=d.dno)
Derived Relations
SQL allows a subquery expression to be used in the from clause.
If such an expression is used, the result relation must be given a name, and the attributes can be renamed.
Eg.
select dno from (select dno,count(*) from emp group by dno) as deptinfo(dno,noofemp) where noofemp>1



dno
101

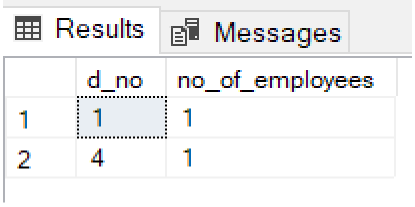
Complex Queries

```
-- Calculate avg,min,max,total salary and no of emp in each dept.
SELECT d_no, AVG(salary) as avg_salary, MAX(salary) as min_salary, MIN(salary) as min_salary,
SUM(salary) as total_salary, COUNT(*) as no_of_employees
FROM Employee
GROUP BY d_no;
```



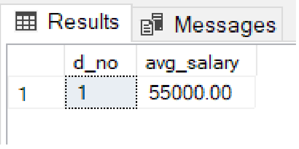
	d_no	avg_salary	min_salary	min_salary	total_salary	no_of_employees
1	1	55000.00	55000.00	55000.00	55000.00	1
2	4	31000.00	43000.00	25000.00	93000.00	3
3	5	33250.00	40000.00	25000.00	133000.00	4

```
-- Display dept no and number of emp for every dept those are earning salary more than 40K.
SELECT d_no, COUNT(*) as no_of_employees
FROM Employee
WHERE salary > 40000
GROUP BY d_no;
```



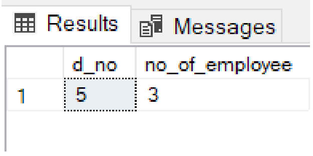
	d_no	no_of_employees
1	1	1
2	4	1

```
-- Display d_no, avg salary of department whose average salary is greater 40000
SELECT d_no, AVG(salary) as avg_salary
FROM Employee
GROUP BY d_no
HAVING (AVG(salary) > 40000);
```



	d_no	avg_salary
1	1	55000.00

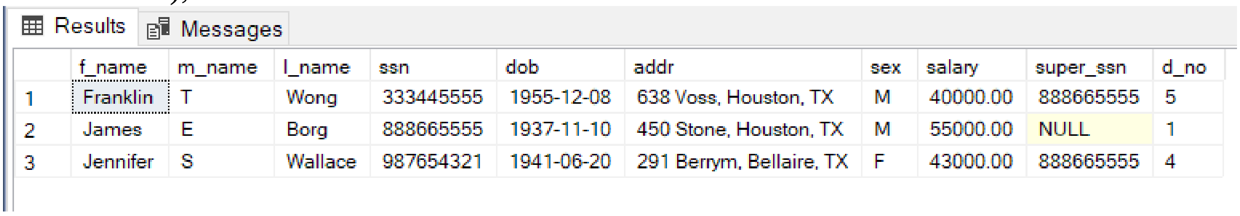
```
-- Display dno and no of emp of those dept where no of emp those are earning salary with-in the
range 30k and 40k are atleast 2.
SELECT d_no, COUNT(*) as no_of_employee
FROM Employee
WHERE salary BETWEEN 30000 AND 40000
GROUP BY d_no
HAVING(COUNT(*) >= 2);
```



	d_no	no_of_employee
1	5	3

Co-related Query and nested

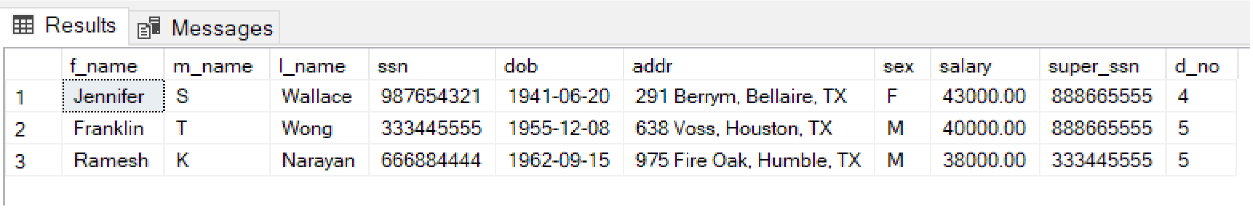
```
-- Display emp details of emp with Maximum salary in each dept
SELECT *
FROM Employee
WHERE salary IN (
                        SELECT MAX(salary)
                        FROM Employee
                        GROUP BY d_no
                );
```



	f_name	m_name	l_name	ssn	dob	addr	sex	salary	super_ssn	d_no
1	Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000.00	888665555	5
2	James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000.00	NULL	1
3	Jennifer	S	Wallace	987654321	1941-06-20	291 Berrym, Bellaire, TX	F	43000.00	888665555	4

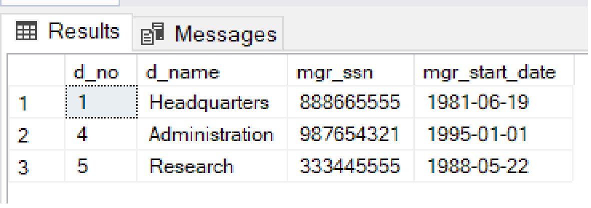
Co-related Queries

```
-- Display emp details of emp earning salary more than avg salary of dept for which they are
working.
SELECT *
FROM Employee AS e1
WHERE e1.salary > (
                                SELECT AVG(salary)
                                FROM Employee AS e2
                                WHERE e1.d_no = e2.d_no
                );
```



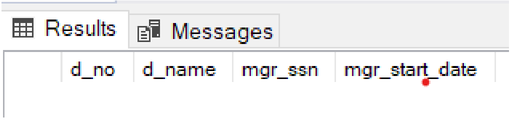
	f_name	m_name	l_name	ssn	dob	addr	sex	salary	super_ssn	d_no
1	Jennifer	S	Wallace	987654321	1941-06-20	291 Berrym, Bellaire, TX	F	43000.00	888665555	4
2	Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000.00	888665555	5
3	Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000.00	333445555	5

```
-- Display the details of those dept which have emp working for them.
SELECT *
FROM Department AS d
WHERE EXISTS (
                        SELECT *
                        FROM Employee AS e
                        WHERE e.d_no = d.d_no
                );
```



	d_no	d_name	mgr_ssn	mgr_start_date
1	1	Headquarters	888665555	1981-06-19
2	4	Administration	987654321	1995-01-01
3	5	Research	333445555	1988-05-22

```
-- Display the details of those dept which have no emp working for them.
SELECT *
FROM Department AS d
WHERE NOT EXISTS (
                                        SELECT *
                                        FROM Employee AS e
                                        WHERE e.d_no = d.d_no
                              );
```



	d_no	d_name	mgr_ssn	mgr_start_date

Conclusion: We have successfully studied and implemented co-related queries and complex queries.