

```
-----|  
Module 4:|  
-----|
```

Using Aggregate Functions

```
-----  
-----  
--(Avg,count,count(*),max,min,sum,stdev) ==make calculations  
  
/*SQL Server summarizes values for an entire table or for groups of columns within the  
table, producing a single value for each set of rows for the specified columns*/  
/*all aggregate functions return a NULL if no rows satisfy the WHERE clause.The COUNT(*)  
function returns a value of zero if no rows satisfy the WHERE clause.*/  
/*aggregate function except count(*) ignore null*/
```

```
select count (*) from employee--count rows=3, if there is null count it
```

```
select count(boss) from employee--count only those whom who have values in fax,
```

```
select max(salary) from EMPLOYEE
```

```
select min(salary) from EMPLOYEE
```

```
select avg(salary) from EMPLOYEE
```

```
select sum(salary) from EMPLOYEE
```

Using the GROUP BY Clause

```
select count(*) as 'Number of employees' from EMPLOYEE group by deptno
```

```
select deptno, count(*) as 'Number of employee' from EMPLOYEE group by deptno  
-- what you select is what you group by minus the aggregation function
```

```
select name,deptno,count(*) as 'Number of employees' from EMPLOYEE group by name,deptno
```

```
select deptno, count(*) as 'Number of employees' from employee where employee.name like  
'%a%'  
group by deptno
```

```
select deptno, count(*) as 'Number of employees' from employee group by deptno  
having count(*)>3
```

```
select deptno, count(*) as 'Number of employees' from employee group by deptno  
having count(*)>3 order by deptno
```

-----Joins-----

Inner join


```
/*the default*/
/*join tables by comparing values in common columns & display the rows that match join
condition*/
select * from student select * from class
select e.name from employee e inner join department d on e.deptno=d.deptno
--move on each class first a & see if any st has a no, then see b find ahmed and ahmed
etc.
--and we can add where c_name='b'
```

outer join

/*return rows that match the join condition between 2 tables & return + the unmatched
rows from either side (left or right). It display null in the unmatched rows*/

```
select e.name from employee e left outer join DEPARTMENT d on e.deptno=d.deptno
--get all student in class & those not in class
```

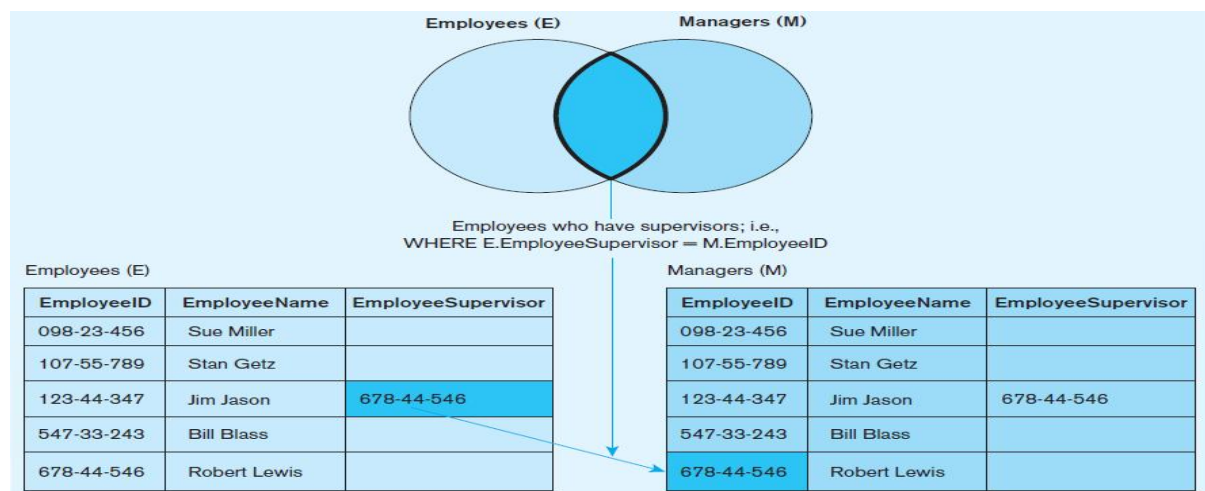
```
select d.deptno from DEPARTMENT d right outer join EMPLOYEE e on e.deptno=d.deptno
--get all class names that have st & get even those that don't have student
```

cross join

/*it returns every combinations of all the rows in the joined tables */
/*it generate test data*/

```
select e.name,d.name from EMPLOYEE e cross join department d
```

```
self join
```

```
select e.employeeName,m.employeeName from EMPLOYEE e, employee m where
(e.employeeSupervisor=m.employeeID)
```

combining multiple result set:

Each SELECT statement within UNION must have the same number of columns
The columns must also have similar data types
The columns in each SELECT statement must also be in the same order

```
select DEPARTMENT.name from DEPARTMENT
```

```
union
```

```
select EMPLOYEE.name from EMPLOYEE
```

```
select d.name,d.deptno from DEPARTMENT d
```

```
union
```

```
select e.name, e.salary from EMPLOYEE e
```

SELECT column list, function(), function(), ...

FROM table1

INNER JOIN table2

...

ON table1.col1 = table2.col2

...

WHERE *criteria for row selection*

[AND criteria for row selection]

[OR criteria for row selection]

GROUP BY column list

HAVING *criteria for function results /*not necessarily with group by*/*

ORDER BY column list

-----nested select-----

Exercise(nested select) : select the names of employees in departments with total average salary >2000

```
select e.name,e.deptno from Employee e where deptno in(select deptno from Employee group  
by deptno having avg(salary) >2000);
```