SQL JOIN

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

```
--SQL query to find employees along with their department details.
select e.emp_id, e.emp_name, e.job_name,  e.dep_id,  d.dep_name
from Employees e , department d where e.dep_id = d.dep_id;
```

68319	KAYLING	PRESIDENT	1001	Finance
66928	BLAZE	MANAGER	3001	Marketing
67832	CLARE	MANAGER	1001	Finance
65646	JONAS	MANAGER	2001	Audit
67858	SCARLET	ANALYST	2001	Audit
69062	FRANK	ANALYST	2001	Audit
63679	SANDRINE	CLERK	2001	Audit
64989	ADELYN	SALESMAN	3001	Marketing
65271	WADE	SALESMAN	3001	Marketing
66564	MADDEN	SALESMAN	3001	Marketing
68454	TUCKER	SALESMAN	3001	Marketing
68736	ADNRES	CLERK	2001	Audit
69000	JULIUS	CLERK	3001	Marketing
69324	MARKER	CLERK	1001	Finance

```
--SQL query to identify those employees who earn 60000 or more per year or do not work as ANALYST. Return employee name, job name, (12*salary) as Annual Salary, department name, and grade.
select e.emp_name,e.job_name,(12*e.salary) as annual_sal ,d.dep_name,s.grade
from Employees e, department d , Salary_grade s where e.dep_id= d.dep_id
and e.salary between s.min_sal and s.max_sal
and ((12*e.salary)>60000 or e.job_name <> 'ANALYST');
```

SANDRINE	CLERK	10800	Audit	1
ADNRES	CLERK	14400	Audit	1
JULIUS	CLERK	12600	Marketing	1
WADE	SALESMAN	16200	Marketing	2
MADDEN	SALESMAN	16200	Marketing	2
MARKER	CLERK	16800	Finance	2
ADELYN	SALESMAN	20400	Marketing	3
TUCKER	SALESMAN	19200	Marketing	3
BLAZE	MANAGER	33000	Marketing	4
CLARE	MANAGER	30600	Finance	4
JONAS	MANAGER	35484	Audit	4
KAYLING	PRESIDENT	72000	Finance	5

```sql
--write a SQL query to identify employees whose salaries are higher than their managers' salaries.
select w.emp_name,w.job_name,w.manager_id,w.salary,m.emp_name 'Manager',
m.emp_id,m.salary "Manager_Salary"
from Employees w,Employees m
where w.manager_id = m.emp_id and w.salary > m.salary;
```

SCARLET	ANALYST	65646	3100	JONAS	65646	2957
FRANK	ANALYST	65646	3100	JONAS	65646	2957

```sql
--SQL query to find those employees whose salary is between 2000 and 5000 (Begin and end values are included.) and location is PERTH. Return employee name, department ID, salary, and commission.
select e.emp_name,e.dep_id,e.salary,e.commission  from Employees e, department d
where e.dep_id = d.dep_id
and d.dep_location= 'PERTH'
and e.salary between 2000 and 5000;
```

BLAZE	3001	2750	NULL

```sql
--write a SQL query to find the employees whose department ID is 1001 or 3001 and whose salary grade is not 4.
--They joined the company before 1992-12-31. Return grade, employee name.
select e.emp_name,s.grade  from Employees e , salary_grade s
where e.dep_id in (1001, 3001)
and (e.salary between s.min_sal and s.max_sal)
and  s.grade<>4 and e.hire_date <'1992-12-31';
```

JULIUS	1
WADE	2
MADDEN	2
MARKER	2
ADELYN	3
TUCKER	3
KAYLING	5

```sql
--write a SQL query to find those employees whose manager name is JONAS.
select e.emp_id,e.emp_name,e.manager_id,e.salary,m.emp_name as 'Manager_name'
from Employees e ,Employees m where e.manager_id =m.emp_id and m.emp_name='JONAS'
```

67858	SCARLET	65646	3100	JONAS
69062	FRANK	65646	3100	JONAS

```sql
--SQL query to find the name,salary grade and salary of the employee FRANK.
--Salary should be equal to the maximum salary within his or her salary group
select e.emp_name , e.salary,s.grade from  Employees e, salary_grade s
```

```sql
where e.emp_name = 'FRANK' and e.salary = s.max_sal
and e.salary between s.min_sal and s.max_Sal ;
```

FRANK	3100	4

```sql
--SQL query to find the employees who joined in 1991 and whose department location is  SYDNEY or MELBOURNE with a salary range of
--2000 to 5000(Begin and end values are included).Return employee ID, employee name, department ID, salary, and department location.
select e.emp_id,e.emp_name,e.dep_id,e.salary,d.dep_location from Employees e,department d
where e.dep_id=d.dep_id
and  e.salary between 2000 and 5000
and d.dep_location in ('SYDNEY','MELBOURNE')
and format(e.hire_date,'yyyy')='1991';
```

67832	CLARE	1001	2550	Sydney
65646	JONAS	2001	2957	Melbourne
69062	FRANK	2001	3100	Melbourne

```sql
--SQL query to find the employees of MARKETING department who come from MELBOURNE or PERTH, are in grades 3 ,4, and 5 and have at least 25 years of experience.Return departmentID, employeeID,employeename, salary,departmentname, department location and grade.

select e.emp_id,e.emp_name,e.dep_id,e.salary,d.dep_name,d.dep_location ,s.grade
from Employees e, department d ,salary_grade s
where e.dep_id=d.dep_id and d.dep_name='MARKETING'
and d.dep_location in ('PERTH','MELBOURNE')
and s.grade in (3,4,5)
and e.salary between s.min_sal and s.max_sal
and datediff(year,e.hire_date,GETDATE())>25;
```

66928	BLAZE	3001	2750	Marketing	Perth	4
64989	ADELYN	3001	1700	Marketing	Perth	3
68454	TUCKER	3001	1600	Marketing	Perth	3

```sql
--write a SQL query to find those employees who are senior to their manager.
select e.emp_name , e.emp_id,m . emp_name , m.emp_id, e.hire_date,m. hire_date from
Employees e, Employees m where e.manager_id=m.emp_id and e.hire_date <m.hire_date;
```

BLAZE	66928	KAYLING	68319	1991-05-01	1991-11-18
CLARE	67832	KAYLING	68319	1991-06-09	1991-11-18
JONAS	65646	KAYLING	68319	1991-04-02	1991-11-18
SANDRINE	63679	FRANK	69062	1990-12-18	1991-12-03
ADELYN	64989	BLAZE	66928	1991-02-20	1991-05-01
WADE	65271	BLAZE	66928	1991-02-22	1991-05-01

```sql
--SQL query to find those employee name who joined after 1991, excluding MARKER or ADELYN in the departments PRODUCTION or AUDIT.
select e.emp_name from Employees e,department d,salary_grade s
where e.dep_id = d.dep_id
and d.dep_name in ('PRODUCTION','AUDIT')
and e.salary between s.min_sal and s.max_sal
and e.emp_name not in ('MARKER','ADELYN')
and format(hire_date,'yyyy') >'1991';
```

SCARLET
ADNRES

```sql
--Query to find the location of all the employees working in the FINANCE or AUDIT department.Sort by department ID in ASC order.
select e.emp_name, d.dep_name, d.dep_id from Employees e inner join  department d on
e.dep_id=d.dep_id where d.dep_name in ('FINANCE','AUDIT') order by e.dep_id ;
```

CLARE	Finance	1001
KAYLING	Finance	1001
MARKER	Finance	1001
ADNRES	Audit	2001
JONAS	Audit	2001
SCARLET	Audit	2001
FRANK	Audit	2001
SANDRINE	Audit	2001

```sql
--SQL query to find the employees along with grades in ascending order.
select e.emp_name, e.emp_id,s.grade from Employees e , salary_grade s
where e.salary between s.min_sal and s.max_sal order by grade;
```

SANDRINE	63679	1
ADNRES	68736	1
JULIUS	69000	1
WADE	65271	2
MADDEN	66564	2
MARKER	69324	2
ADELYN	64989	3
TUCKER	68454	3
BLAZE	66928	4
CLARE	67832	4
JONAS	65646	4
SCARLET	67858	4
FRANK	69062	4
KAYLING	68319	5

```sql
--query to find the employees according to the department in ascending order.Return name,department,salary,and grade
select e.emp_name ,d.dep_name, e.salary, s.grade
from Employees e,department d,salary_grade s
where e.dep_id = d.dep_id
and e.salary between s.min_sal and s.max_sal
order by  e.dep_id ;
```

MARKER	Finance	1400	2
CLARE	Finance	2550	4
KAYLING	Finance	6000	5
JONAS	Audit	2957	4
SCARLET	Audit	3100	4
FRANK	Audit	3100	4
SANDRINE	Audit	900	1
ADNRES	Audit	1200	1
JULIUS	Marketing	1050	1
WADE	Marketing	1350	2
MADDEN	Marketing	1350	2
ADELYN	Marketing	1700	3
TUCKER	Marketing	1600	3
BLAZE	Marketing	2750	4

```sql
--SQL query to list the grade, number of employees, and maximum salary of each grade
SELECT s.grade, count(*) 'count', max(salary) as Max_sal FROM Employees e, salary_grade s
WHERE e.salary between s.min_sal and s.max_sal
GROUP BY s.grade;
```

1	3	1200
2	3	1400
3	2	1700
4	5	3100
5	1	6000

```sql
-- SQL query to identify departments with at least two SALESMEN in each grade.
Return department name, grade and number of employees.
SELECT d.dep_name, s.grade,count(*) EmpCount FROM Employees e,  department d,salary_grade s
WHERE e.dep_id = d.dep_id
and e.job_name = 'SALESMAN'
and e.salary between s.min_sal and s.max_sal
GROUP BY d.dep_name, s.grade HAVING count(*)>= 2;
```

Marketing	2	2
Marketing	3	2