

Sql code Case study

-- Employee Table

```
create table employee (  
    employeeid int primary key,  
    name varchar(50),  
    departmentid int,  
    salary int,  
    hiredate date  
);  
drop table employee
```

-- Department Table

```
create table department (  
    departmentid int primary key,  
    departmentname varchar(50)  
);
```

-- Performance Table

```
create table performance (  
    performanceid int primary key,  
    employeeid int,  
    reviewdate date,  
    score int,  
    foreign key (employeeid) references employee(employeeid)  
);
```

-- Departments

insert into department values

(1, 'HR'),

(2, 'Engineering'),

(3, 'Marketing');

-- Employees

insert into employee values

(1, 'Arjun', 2, 75000, '2023-02-15'),

(2, 'Meena', 1, 60000, '2024-07-01'),

(3, 'Ravi', 2, 85000, '2024-03-10'),

(4, 'Sneha', 3, 72000, '2022-09-05'),

(5, 'Kumar', 1, 90000, '2024-08-20');

-- Performance

insert into performance values

(101, 1, '2025-01-10', 88),

(102, 2, '2025-02-14', 92),

(103, 3, '2025-01-30', 95),

(104, 4, '2024-11-20', 78),

(105, 5, '2025-01-15', 85);

select * from employee

select * from department

select * from performance

SQLQuery1.sql | **SQLQuery5.sql** * | SQLQuery4.sql * | SQLQuery3.sql * | SQLQuery1.sql * | SqlSchemaCompare2* | SQLQuery2.sql *

Practicetask1

```

(4, 'Sneha', 3, 72000, '2022-09-05'),
(5, 'Kumar', 1, 90000, '2024-08-20');

-- Performance
insert into performance values
(101, 1, '2025-01-10', 88),
(102, 2, '2025-02-14', 92),
(103, 3, '2025-01-30', 95),
(104, 4, '2024-11-20', 78),
(105, 5, '2025-01-15', 85);

select * from employee
select * from department
select * from performance

```

100 % No issues found

T-SQL Results Message

	employeeid	name	departmentid	salary	hiredate
1	1	Arjun	2	75000	2023-02-15
2	2	Meena	1	60000	2024-07-01
3	3	Ravi	2	85000	2024-03-10
4	4	Sneha	3	72000	2022-09-05
5	5	Kumar	1	90000	2024-08-20

	departmentid	departmentname
1	1	HR
2	2	Engineering
3	3	Marketing

	performanceid	employeeid	reviewdate	score
1	101	1	2025-01-10	88
2	102	2	2025-02-14	92
3	103	3	2025-01-30	95
4	104	4	2024-11-20	78
5	105	5	2025-01-15	85

--Top 3 score

select top 3

e.name,

d.departmentname,

p.score,

p.reviewdate

from performance p

join employee e on p.employeeid = e.employeeid

join department d on e.departmentid = d.departmentid

order by p.score desc;

```
select top 3
    e.name,
    d.departmentname,
    p.score,
    p.reviewdate
from performance p
join employee e on p.employeeid = e.employeeid
join department d on e.departmentid = d.departmentid
order by p.score desc;

-- Department Average Score
select
    d.departmentname,
```

-- Department Average Score

select

d.departmentname,

avg(p.score) as avg_score

from performance p

join employee e on p.employeeid = e.employeeid

join department d on e.departmentid = d.departmentid

group by d.departmentname;

```
p.reviewdate
from performance p
join employee e on p.employeeid = e.employeeid
join department d on e.departmentid = d.departmentid
order by p.score desc;

-- Department Average Score
select
    d.departmentname,
    avg(p.score) as avg_score
from performance p
join employee e on p.employeeid = e.employeeid
join department d on e.departmentid = d.departmentid
group by d.departmentname;
```

--Salary vs Score (Salary > avg and Score > 80)

select

e.name,

e.salary,

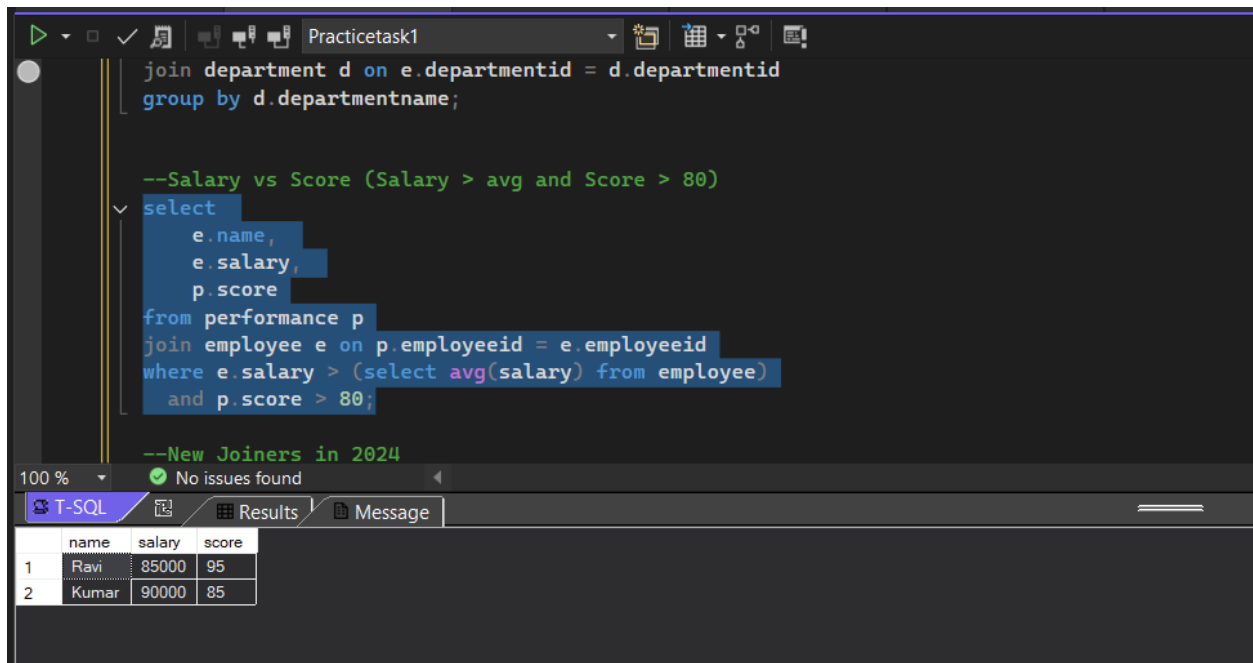
p.score

from performance p

join employee e on p.employeeid = e.employeeid

where e.salary > (select avg(salary) from employee)

and p.score > 80;



The screenshot shows a SQL Server Enterprise Manager window titled 'Practicetask1'. The 'T-SQL' tab is active, displaying a query. The query includes a join statement for department, followed by a comment '--Salary vs Score (Salary > avg and Score > 80)', and a select statement with columns e.name, e.salary, and p.score. The query is joined on employeeid and filtered by salary greater than the average and score greater than 80. Below the query, a comment '--New Joiners in 2024' is visible. The 'Results' tab shows a table with 2 rows and 3 columns: name, salary, and score. The first row is Ravi with salary 85000 and score 95. The second row is Kumar with salary 90000 and score 85. The 'Message' tab shows 'No issues found'.

```
join department d on e.departmentid = d.departmentid
group by d.departmentname;

--Salary vs Score (Salary > avg and Score > 80)
select
    e.name,
    e.salary,
    p.score
from performance p
join employee e on p.employeeid = e.employeeid
where e.salary > (select avg(salary) from employee)
and p.score > 80;

--New Joiners in 2024
```

	name	salary	score
1	Ravi	85000	95
2	Kumar	90000	85

--New Joiners in 2024

select * from employee

where hiredate >= '2024-01-01' and hiredate <= '2024-12-31';

100 % No issues found

	employeeid	name	departmentid	salary	hiredate
1	2	Meena	1	60000	2024-07-01
2	3	Ravi	2	85000	2024-03-10
3	5	Kumar	1	90000	2024-08-20