

# CSE5011 – Database Systems and Design

Assessment - 3  
(Ex-5 and Ex-6)

Rahul Pathak  
21MAI0069

Vellore Institute of Technology

## Exercise: V

### Sub Query and View

#### 1. Find the employee who is getting highest salary in the department research

```
select FIRST_NAME||' '||MID_NAME||' '||LAST_NAME as employee , SALARY from employee
where salary = (select max(salary) from employee where DEPARTMENT_NUMBER in
(select DEPARTMENT_NUMBER from department where DEPARTMENT_NAME = 'Research'))
```

Results Explain Describe Saved SQL History

EMPLOYEE	SALARY
Kit R Harrington	73000

1 rows returned in 0.00 seconds

[CSV Export](#)

#### 2. Find the employees who earn the same salary as the minimum salary for each Department

```
select DEPARTMENT_NAME, FIRST_NAME||' '||MID_NAME||' '||LAST_NAME as EMPLOYEE, SALARY from employee e, department d
where e.DEPARTMENT_NUMBER = d.DEPARTMENT_NUMBER and SALARY in
(select MIN(SALARY) from EMPLOYEE group by DEPARTMENT_NUMBER )
```

Results Explain Describe Saved SQL History

DEPARTMENT_NAME	EMPLOYEE	SALARY
Headquarter	John B Smith	30000
Administration	Frankin T Wong	40000
Production	Robert D Junior	90000
Finance	Ramesh K Narayan	38000
Research	George R Martin	56000

5 rows returned in 0.00 seconds

[CSV Export](#)

### 3. Find the employee whose salary is greater than average salary of department 2

```
select FIRST_NAME||' '||MID_NAME||' '||LAST_NAME as employee, SALARY from employee  
where salary > (select AVG(SALARY) from EMPLOYEE where DEPARTMENT_NUMBER = 2)
```

**Results** Explain Describe Saved SQL History

EMPLOYEE	SALARY
Chris Evans	91000
Chris Hemsworth	92000
Elizabeth Olsen	93000
Scarlett Johansson	94000
Mark A Ruffalo	89000
Tom Hiddleston	88000
Tom Holland	87000
Kit R Harrington	73000
Doug E Gilbert	80000
Joyce PAN	70000
More than 10 rows available. Increase rows selector to view more rows.	

10 rows returned in 0.00 seconds

[CSV Export](#)

### 4. List out all the department names with their individual employees strength

```
select DEPARTMENT_NAME, count(*) as employee_strength from  
employee e, department d where e.DEPARTMENT_NUMBER = d.DEPARTMENT_NUMBER  
group by DEPARTMENT_NAME
```

**Results** Explain Describe Saved SQL History

DEPARTMENT_NAME	EMPLOYEE_STRENGTH
Administration	4
Research	2
Headquarter	5
Finance	2
Production	2

5 rows returned in 0.00 seconds

[CSV Export](#)

## 5. Find out the department name having highest employee strength

```
select department_name, count(e.department_number) as highest_emp_strength
from department d, employee e
where d.department_number = e.department_number
having count(e.department_number) =
(select MAX(count(*)) from employee group by department_number)
group by department_name
```

**Results** Explain Describe Saved SQL History

DEPARTMENT_NAME	HIGHEST_EMP_STRENGTH
Headquarter	5

1 rows returned in 0.00 seconds

[CSV Export](#)

## 6. List out all the departments and average salary drawn by their employees

```
select department_name, AVG(salary) as average_salary from employee e, department d
where e.department_number = d.department_number
group by department_name
```

**Results** Explain Describe Saved SQL History

DEPARTMENT_NAME	AVERAGE_SALARY
Administration	65000
Research	64500
Headquarter	77800
Finance	62500
Production	90500

5 rows returned in 0.00 seconds

[CSV Export](#)

## 7. Find maximum average salary for each department.

```
select department_name, AVG(salary) as MAX_AVG_SALARY from  
employee e, department d  
where e.department_number = d.department_number  
having AVG(SALARY) = (select MAX(AVG(SALARY)) from employee group by department_number) group by department_name
```

Results Explain Describe Saved SQL History

DEPARTMENT_NAME	MAX_AVG_SALARY
Production	90500

1 rows returned in 0.00 seconds [CSV Export](#)

## 8. Create a view to display the employee details who is working in IT department.

```
create view IT_employee as  
select * from employee where department_number =  
(select department_number from department where department_name = 'IT')
```

Results Explain Describe Saved SQL History

View created.

0.78 seconds

```
create view IT_employee as  
select * from employee where department_number =  
(select department_number from department where department_name = 'IT')  
select * from IT_employee
```

Results Explain Describe Saved SQL History

FIRST_NAME	MID_NAME	LAST_NAME	SSN_NUMBER	BIRTHDAY	ADDRESS	SEX	SALARY	SUPERVISOR_SSN	DEPARTMENT_NUMBER
Joyce	-	PAN	124	07-FEB-73	Vellore	F	70000	-	4

1 rows returned in 0.00 seconds [CSV Export](#)

## 9. Create a logical table to store employee details who is getting salary more than 10000.

```
create table logical_emp_table as  
select first_name || ' ' || mid_name || ' ' || last_name as FULL_NAME, salary from employee where salary > 10000
```

**Results** Explain Describe Saved SQL History

Table created.

0.67 seconds

```
create table logical_emp_table as  
select first_name || ' ' || mid_name || ' ' || last_name as FULL_NAME, salary from employee where salary > 10000  
select * from logical_emp_table
```

**Results** Explain Describe Saved SQL History

FULL_NAME	SALARY
Frankin T Wong	40000
Jennifer S Wallace	43000
John B Smith	30000
Ramesh K Narayan	38000
Chris Evans	91000
Chris Hemsworth	92000
Elizabeth Olsen	93000
Scarlett Johansson	94000
Mark A Ruffalo	89000
Tom Hiddleston	88000
Tom Holland	87000
George R Martin	56000
Kit R Harrington	73000
Doug E Gilbert	80000
Joyce PAN	70000
Robert D Junior	90000

16 rows returned in 0.14 seconds

[CSV Export](#)

## 10. Create a table to store the employees details based on the department no

```
create table emp_details_with_dep_no as  
select first_name || ' ' || mid_name || ' ' || last_name as FULL_NAME, department_number from employee
```

**Results** Explain Describe Saved SQL History

Table created.

0.67 seconds

```
create table emp_details_with_dep_no as  
select first_name || ' ' || mid_name || ' ' || last_name as FULL_NAME, department_number from employee  
select * from emp_details_with_dep_no
```

**Results** Explain Describe Saved SQL History

FULL_NAME	DEPARTMENT_NUMBER
Frankin T Wong	2
Jennifer S Wallace	2
John B Smith	1
Ramesh K Narayan	3
Chris Evans	5
Chris Hemsworth	1
Elizabeth Olsen	1
Scarlett Johansson	1
Mark A Ruffalo	2
Tom Hiddleston	2
Tom Holland	3
George R Martin	7
Kit R Harrington	7
Doug E Gilbert	1
Joyce PAN	4
Robert D Junior	5

16 rows returned in 0.23 seconds

[CSV Export](#)

## Exercise: VI

### Joins

Consider the schema given in exercise 2, and execute the following queries

1. Retrieve the names of all employees in department 5 who work more than 10 hours per week on ProductX project.

```
select e.first_name || ' ' || e.mid_name || ' ' || e.last_name as Full_name,  
p.project_name, p.project_number, p.department_number, w.hours  
from employee e, project p, works_on w  
where e.SSN_NUMBER = w.EMPLOYEE_SSN  
and p.project_number = w.project_no  
and p.department_number = 5  
and w.hours > 10
```

**Results** Explain Describe Saved SQL History

FULL_NAME	PROJECT_NAME	PROJECT_NUMBER	DEPARTMENT_NUMBER	HOURS
Chris Hemsworth	ProjectF	77	5	20
Chris Hemsworth	ProjectG	22	5	20
Mark A Ruffalo	ProjectG	22	5	17

3 rows returned in 0.00 seconds

[CSV Export](#)

2. List the names of all employees who have a dependent with the same first name as themselves.

```
select e.first_name || ' ' || e.mid_name || ' ' || e.last_name as Full_name,  
d.depn_name, d.relationship from  
employee e, dependent d  
where e.SSN_NUMBER = d.EMPLOYEE_SSN  
and e.first_name = d.depn_name
```

**Results** Explain Describe Saved SQL History

FULL_NAME	DEPN_NAME	RELATIONSHIP
Elizabeth Olsen	Elizabeth	Daughter

1 rows returned in 0.01 seconds

[CSV Export](#)



### 3. Find the names of all the employees who are directly supervised by 'Franklin Wong'.

```
select e.first_name || ' ' || e.mid_name || ' ' || e.last_name as employee_name,  
s.first_name || ' ' || s.mid_name || ' ' || s.last_name as supervisor_name  
from employee e, employee s  
where e.supervisor_ssn = s.ssn_number  
and s.first_name = 'Frankin'  
and s.last_name = 'Wong'
```

**Results** Explain Describe Saved SQL History

EMPLOYEE_NAME	SUPERVISOR_NAME
Tom Hiddleston	Frankin T Wong
Doug E Gilbert	Frankin T Wong
Joyce PAN	Frankin T Wong

3 rows returned in 0.01 seconds [CSV Export](#)

### 4. Retrieve the names of all who do not work on any project.

```
select first_name || ' ' || mid_name || ' ' || last_name as employee_name, SSN_NUMBER  
from employee where SSN_NUMBER not in (select e.SSN_NUMBER from employee e, works_on w  
where e.ssn_number = w.employee_ssn)
```

**Results** Explain Describe Saved SQL History

EMPLOYEE_NAME	SSN_NUMBER
Tom Holland	987654321
Doug E Gilbert	123
John B Smith	678
Frankin T Wong	125
Joyce PAN	124
Jennifer S Wallace	564
George R Martin	981380392
Kit R Harrington	981380393
Ramesh K Narayan	234
Elizabeth Olsen	333445555

10 rows returned in 0.00 seconds [CSV Export](#)

5. Find the names and addresses of all employees who work on at least one project located in Houston but whose department has no location in Houston.

```
select e.first_name || ' ' || e.mid_name || ' ' || e.last_name as FULL_NAME, e.address,  
p.project_name, p.project_number, p.project_location, d.dep_no  
from project p, works_on w, dept_locations d, employee e  
where p.project_number = w.project_no  
and p.department_number = d.dep_no  
and e.SSN_NUMBER = w.EMPLOYEE_SSN  
and p.project_location = 'Houston'  
and d.dep_no not in (select dep_no from dept_locations where dep_loc = 'Houston')
```

Results Explain Describe Saved SQL History

FULL_NAME	ADDRESS	PROJECT_NAME	PROJECT_NUMBER	PROJECT_LOCATION	DEP_NO
Scarlett Johansson	Manhattan, New York City, USA	Project1	1	Houston	3

1 rows returned in 0.02 seconds

[CSV Export](#)

6. List the names of all managers who have no dependents.

```
select first_name||' '||mid_name||' '||last_name as employee_with_no_dependents  
from employee, department  
where ssn_number = manager_ssn  
and manager_ssn not in (select employee_ssn from dependent)
```

Results Explain Describe Saved SQL History

EMPLOYEE_WITH_NO_DEPENDENTS
Ramesh K Narayan
Doug E Gilbert
Chris Evans
Jennifer S Wallace
John B Smith

5 rows returned in 0.03 seconds

[CSV Export](#)

7. List the employee's names and the department names if they happen to manage a department.

```
select first_name||' '||mid_name||' '||last_name as employee_as_manager, department_name  
from employee, department  
where ssn_number = manager_ssn
```

**Results** Explain Describe Saved SQL History

EMPLOYEE_AS_MANAGER	DEPARTMENT_NAME
Ramesh K Narayan	Finance
Doug E Gilbert	IT
Chris Evans	Mental Health
Chris Hemsworth	Research
Jennifer S Wallace	Administration
John B Smith	Headquarter
Robert D Junior	Production

7 rows returned in 0.01 seconds

[CSV Export](#)

8. For each project retrieve the project number, project name and the number of employees who work on that project.

```
select project_name, project_number, count(employee_ssn) as no_of_employee from  
project left join works_on  
on project_number = project_no  
group by project_name, project_number  
order by project_name
```

**Results** Explain Describe Saved SQL History

PROJECT_NAME	PROJECT_NUMBER	NO_OF_EMPLOYEE
ProjectA	3388	2
ProjectB	1945	2
ProjectC	6688	0
ProjectD	2423	0
ProjectE	7745	0
ProjectF	77	1
ProjectG	22	2
ProjectH	43	0
ProjectI	1	1
ProjectJ	12	1

10 rows returned in 0.04 seconds

[CSV Export](#)

9. For each project, list the project name and the total hours per week (by all employees) spent on that project.

```
select project_name, sum(hours) as total_hours_per_week from  
project left join works_on  
on project_number = project_no  
group by project_name, project_number  
order by project_name
```

**Results** Explain Describe Saved SQL History

PROJECT_NAME	TOTAL_HOURS_PER_WEEK
ProjectA	72.5
ProjectB	29
ProjectC	-
ProjectD	-
ProjectE	-
ProjectF	20
ProjectG	37
ProjectH	-
ProjectI	11.5
ProjectJ	13

10 rows returned in 0.00 seconds

[CSV Export](#)

10. Retrieve the names of the employees who have 2 or more dependents.

```
select first_name||' '||mid_name||' '||last_name as employee_name,  
count(employee_ssn) as dependent_count  
from employee, dependent  
where ssn_number = employee_ssn  
group by first_name||' '||mid_name||' '||last_name  
having count(employee_ssn) >= 2
```

**Results** Explain Describe Saved SQL History

EMPLOYEE_NAME	DEPENDENT_COUNT
Elizabeth Olsen	4

1 rows returned in 0.03 seconds

[CSV Export](#)