

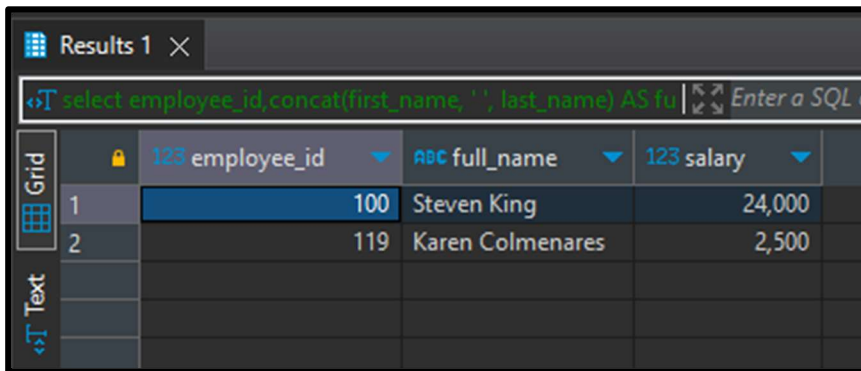
## Jawaban Quiz SQL: Rahmat-Agung-Hadiwardoyo

1. Buat query untuk menampilkan employee yang memiliki gaji tertinggi dan ter-endah. Gunakan union, min, max. (name masih berdasarkan first\_name, belum digabung dengan last\_name)

Query:

```
select employee_id,concat(first_name, ' ', last_name) AS full_name,salary
from employees
where salary = (select MIN(salary) from employees )
union
select employee_id, concat(first_name, ' ', last_name) AS full_name,salary
from employees
where salary = (select MAX(salary) from employees )
order by salary desc
```

Hasil:



The screenshot shows a SQL query result in a grid view. The query is: `select employee_id,concat(first_name, ' ', last_name) AS full_name,salary from employees where salary = (select MIN(salary) from employees ) union select employee_id, concat(first_name, ' ', last_name) AS full_name,salary from employees where salary = (select MAX(salary) from employees ) order by salary desc`. The results table has four columns: an index, employee\_id, full\_name, and salary. The first row shows employee\_id 100, full\_name Steven King, and salary 24,000. The second row shows employee\_id 119, full\_name Karen Colmenares, and salary 2,500.

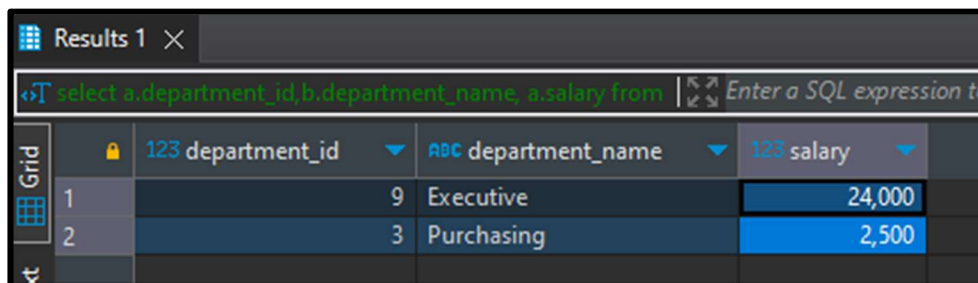
	123 employee_id	ABC full_name	123 salary
1	100	Steven King	24,000
2	119	Karen Colmenares	2,500

2. Buat query untuk menampilkan department yang memiliki gaji tertinggi dan ter-endah. Gunakan union, min, max.

Query:

```
select a.department_id,b.department_name, a.salary
from employees a join departments b
on a.department_id = b.department_id
where salary = (select MIN(salary) from employees )
union
select a.department_id,b.department_name, a.salary
from employees a join departments b
on a.department_id = b.department_id
where salary = (select MAX(salary) from employees )
order by salary desc
```

Hasil:



The screenshot shows a SQL query result in a grid view. The query is: `select a.department_id,b.department_name, a.salary from employees a join departments b on a.department_id = b.department_id where salary = (select MIN(salary) from employees ) union select a.department_id,b.department_name, a.salary from employees a join departments b on a.department_id = b.department_id where salary = (select MAX(salary) from employees ) order by salary desc`. The results table has four columns: an index, department\_id, department\_name, and salary. The first row shows department\_id 9, department\_name Executive, and salary 24,000. The second row shows department\_id 3, department\_name Purchasing, and salary 2,500.

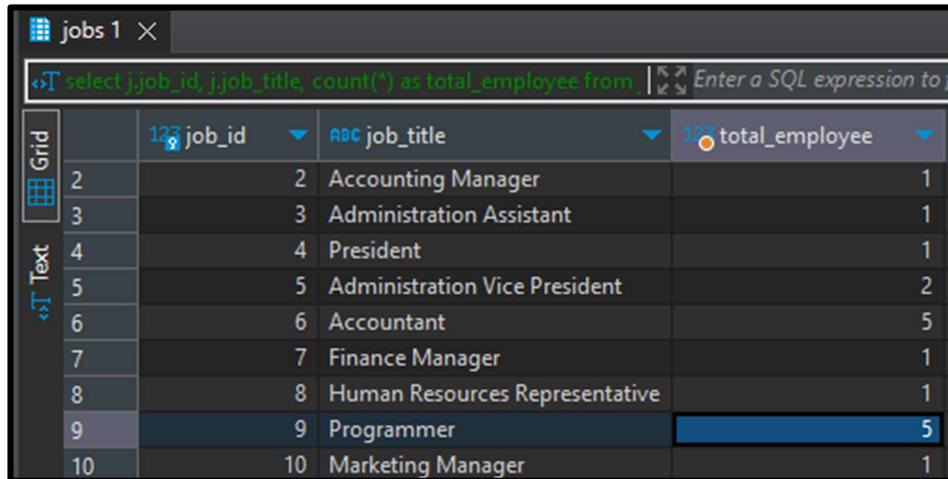
	123 department_id	ABC department_name	123 salary
1	9	Executive	24,000
2	3	Purchasing	2,500

3. Buat query untuk menampilkan jumlah employee berdasarkan job role nya.

Query:

```
select j.job_id, j.job_title, count(*) as total_employee
from jobs j join employees e
on j.job_id = e.job_id
group by j.job_id
order by j.job_id
```

Hasil:



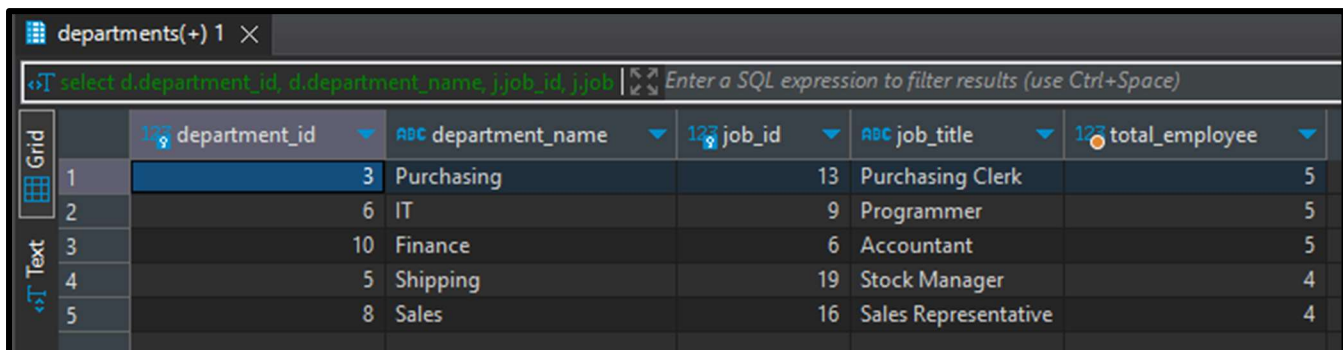
	123 job_id	ABC job_title	123 total_employee
2	2	Accounting Manager	1
3	3	Administration Assistant	1
4	4	President	1
5	5	Administration Vice President	2
6	6	Accountant	5
7	7	Finance Manager	1
8	8	Human Resources Representative	1
9	9	Programmer	5
10	10	Marketing Manager	1

4. Buat query untuk menampilkan Total employee tiap department dan jobs yang totalnya melebihi dari 3 orang.

Query:

```
select d.department_id, d.department_name, j.job_id, j.job_title,
count(e.employee_id) as total_employee
from employees e join departments d
on d.department_id = e.department_id join jobs j
on e.job_id = j.job_id
group by d.department_id, j.job_id
having count(e.employee_id) >3
order by total_employee desc
```

Hasil:



	123 department_id	ABC department_name	123 job_id	ABC job_title	123 total_employee
1	3	Purchasing	13	Purchasing Clerk	5
2	6	IT	9	Programmer	5
3	10	Finance	6	Accountant	5
4	5	Shipping	19	Stock Manager	4
5	8	Sales	16	Sales Representative	4

5. Buat query untuk menampilkan employee yang memiliki masa kerja  $\leq 25$  tahun. Gunakan function age & extract.

Query:

```
select first_name, last_name, concat(first_name, ' ', last_name) as
full_name, hire_date, age(now(), hire_date) as "working period"
from employees
where age(now(), hire_date) <= '25 years'
order by hire_date desc
```

Hasil:

	first_name	last_name	full_name	hire_date	working period
1	Charles	Johnson	Charles Johnson	2000-01-04	23 years 8 mons 14 days 00:46:35.949766
2	Luis	Popp	Luis Popp	1999-12-07	23 years 9 mons 11 days 00:46:35.949766
3	Karen	Colmenares	Karen Colmenares	1999-08-10	24 years 1 mon 8 days 00:46:35.949766
4	Kimberely	Grant	Kimberely Grant	1999-05-24	24 years 3 mons 25 days 00:46:35.949766
5	Diana	Lorentz	Diana Lorentz	1999-02-07	24 years 7 mons 11 days 00:46:35.949766
6	Guy	Himuro	Guy Himuro	1998-11-15	24 years 10 mons 3 days 00:46:35.949766
7	Irene	Mikkilineni	Irene Mikkilineni	1998-09-28	24 years 11 mons 20 days 00:46:35.949766

6. Buat script update untuk menaikkan gaji pegawai yang masa kerjanya  $\leq 25$  tahun, naikan gaji sekarang + 100, dan tampilkan.

Query:

```
update employees set salary = salary + 100 where age(now(), hire_date)
>= '25 years'

select concat(first_name, ' ', last_name) as full_name, hire_date,
salary, age(now(), hire_date) as "working period"
from employees
order by age(now(), hire_date)
```

Hasil:

	full_name	hire_date	123 salary	working period
	Kimberely Grant	1999-05-24	7,000	24 years 3 mons 25 days 00:53:10.661458
	Diana Lorentz	1999-02-07	4,200	24 years 7 mons 11 days 00:53:10.661458
	Guy Himuro	1998-11-15	2,600	24 years 10 mons 3 days 00:53:10.661458
	Irene Mikkilineni	1998-09-28	2,700	24 years 11 mons 20 days 00:53:10.661458
	Jack Livingston	1998-04-23	8,400	25 years 4 mons 25 days 00:53:10.661458
	Jonathon Taylor	1998-03-24	8,600	25 years 5 mons 25 days 00:53:10.661458
	Jose Manuel Urman	1998-03-07	7,800	25 years 6 mons 11 days 00:53:10.661458
	Valli Pataballa	1998-02-05	4,800	25 years 7 mons 13 days 00:55:49.580549
	Shelli Baida	1997-12-24	2,900	25 years 8 mons 25 days 00:55:49.580549
	Shanta Vollman	1997-10-10	6,500	25 years 11 mons 8 days 00:55:49.580549
	Ismael Sciarra	1997-09-30	7,700	25 years 11 mons 20 days 00:55:49.580549

7. Buat script update untuk memindahkan department yang jumlah pegawai nya 1 orang, dipindahkan ke department Sales, kecuali department HRD.

Query:

```
update employees e
set department_id = ( select department_id from departments where
department_name ='Sales')
where e.department_id IN (
SELECT d.department_id FROM departments d
JOIN employees e ON d.department_id = e.department_id
GROUP BY d.department_id
HAVING COUNT(e.department_id) = 1
) and e.department_id !=4
```

Hasil:

Name	Value
Updated Rows	2
Query	update employees e set department_id = ( select department_id from departments where department_name ='Sales') where e.department_id IN ( SELECT d.department_id FROM departments d JOIN employees e ON d.department_id = e.department_id GROUP BY d.department_id HAVING COUNT(e.department_id) = 1 ) and e.department_id !=4
Start time	Mon Sep 18 01:45:33 ICT 2023
Finish time	Mon Sep 18 01:45:33 ICT 2023

		phone_number	hire_date	job_id	salary	manager_id	department_id
1	hartstein@sqltutorial.org	515.123.5555	1996-02-17	10	13,000	100	2
2	sqltutorial.org	603.123.6666	1997-08-17	11	6,000	201	2
3	khoo@sqltutorial.org	515.127.4562	1995-05-18	13	3,100	114	3
4	ely@sqltutorial.org	515.127.4561	1994-12-07	14	11,000	100	3
5	menares@sqltutorial.org	515.127.4566	1999-08-10	13	2,500	114	3
6	ro@sqltutorial.org	515.127.4565	1998-11-15	13	2,600	114	3
7	as@sqltutorial.org	515.127.4564	1997-07-24	13	2,800	114	3
8	ja@sqltutorial.org	515.127.4563	1997-12-24	13	2,900	114	3
9	vris@sqltutorial.org	515.123.7777	1994-06-07	8	6,500	101	4
10	erett@sqltutorial.org	650.501.2876	1997-03-03	17	3,900	123	5
11	weiss@sqltutorial.org	650.123.1234	1996-07-18	19	8,000	100	5
12	p@sqltutorial.org	650.123.2234	1997-04-10	19	8,200	100	5
13	ufing@sqltutorial.org	650.123.3234	1995-05-01	19	7,900	100	5
14	lman@sqltutorial.org	650.123.4234	1997-10-10	19	6,500	100	5
15	kilineni@sqltutorial.org	650.124.1224	1998-09-28	18	2,700	120	5
16	@sqltutorial.org	650.501.1876	1996-02-04	17	4,000	123	5
17	alla@sqltutorial.org	590.423.4560	1998-02-05	9	4,800	103	6
18	ntz@sqltutorial.org	590.423.5567	1999-02-07	9	4,200	103	6
19	.hunold@sqltutorial.org	590.423.4567	1990-01-03	9	9,000	102	6
20	st@sqltutorial.org	590.423.4568	1991-05-21	9	6,000	103	6
21	tin@sqltutorial.org	590.423.4569	1997-06-25	9	4,800	103	6
22	baer@sqltutorial.org	515.123.8888	1994-06-07	12	10,000	101	8
23	tners@sqltutorial.org	[NULL]	1997-01-05	15	13,500	100	8
24	jston@sqltutorial.org	[NULL]	1998-04-23	16	8,400	100	8
25	.grant@sqltutorial.org	[NULL]	1999-05-24	16	7,000	100	8
26	hnson@sqltutorial.org	[NULL]	2000-01-04	16	6,200	100	8
27	ell@sqltutorial.org	[NULL]	1996-10-01	15	14,000	100	8
28	halen@sqltutorial.org	515.123.4444	1987-09-17	3	4,400	101	8
29	taylor@sqltutorial.org	[NULL]	1998-03-24	16	8,600	100	8
30	ig@sqltutorial.org	515.123.4567	1987-06-17	4	24,000	[NULL]	9
31	chhar@sqltutorial.org	515.123.4568	1989-09-21	5	17,000	100	9
32	in@sqltutorial.org	515.123.4569	1993-01-13	5	17,000	100	9
33	enberg@sqltutorial.org	515.124.4569	1994-08-17	7	12,000	101	10
34	i@sqltutorial.org	515.124.4269	1997-09-28	6	8,200	108	10
35	arra@sqltutorial.org	515.124.4369	1997-09-30	6	7,700	108	10
36	uel.urman@sqltutorial.org	515.124.4469	1998-03-07	6	7,800	108	10
37	@sqltutorial.org	515.124.4567	1999-12-07	6	6,900	108	10
38	iet@sqltutorial.org	515.124.4169	1994-08-16	6	9,000	108	10
39	etz@sqltutorial.org	515.123.8181	1994-06-07	1	8,300	205	11
40	ggins@sqltutorial.org	515.123.8080	1994-06-07	2	12,000	101	11