JOBSHEET 3 BASIS DATA LANJUT

QUERY LANJUT (SELECT, JOIN, CTE)

GARGARINA NANDA ISWATI

NIM. 244107060100

SIB 2F – 12

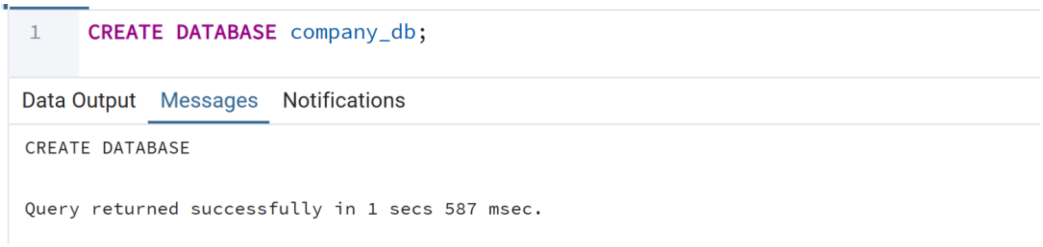
JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2025

SETUP AWAL

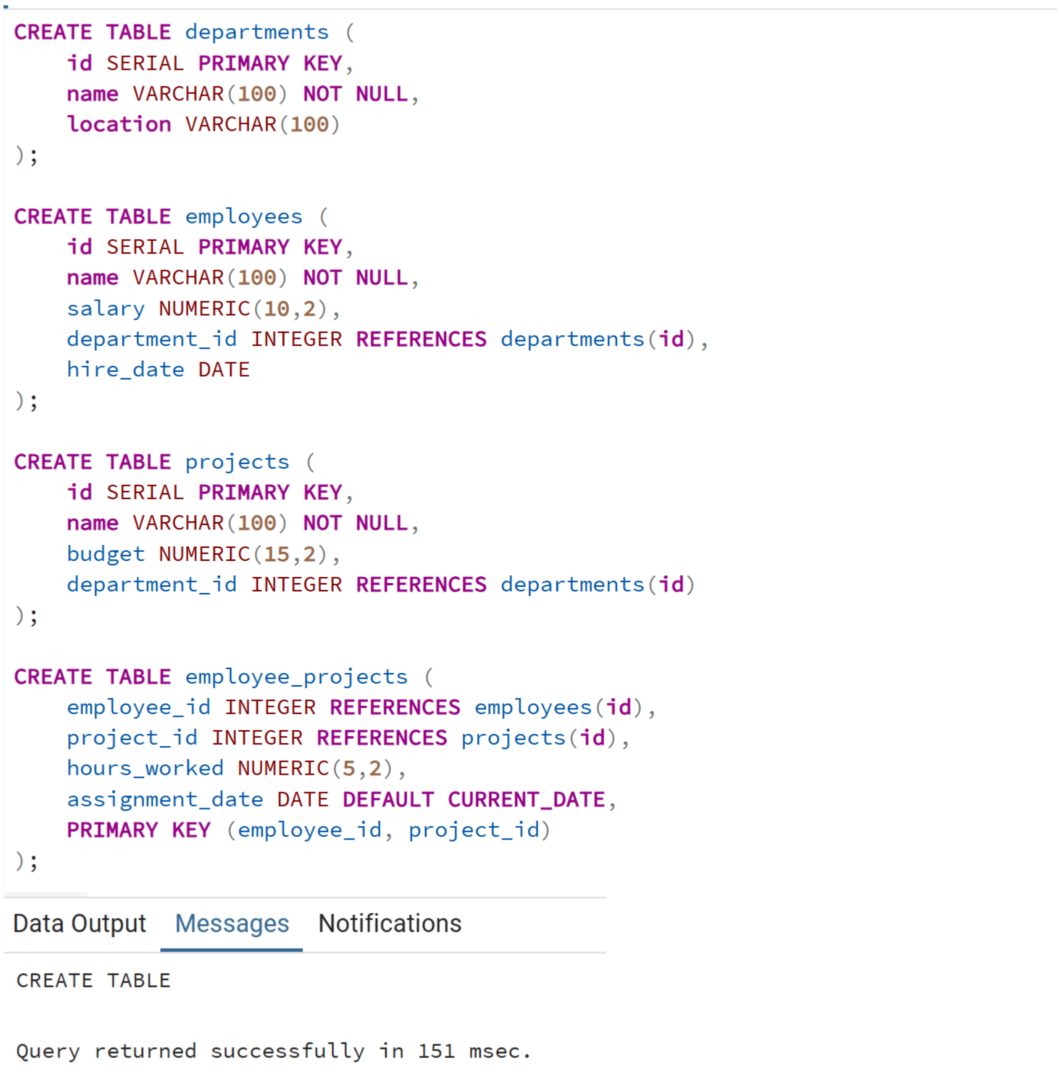
1. Membuat Database baru



```sql
CREATE DATABASE company_db;
```

```
CREATE DATABASE

Query returned successfully in 1 secs 587 msec.
```

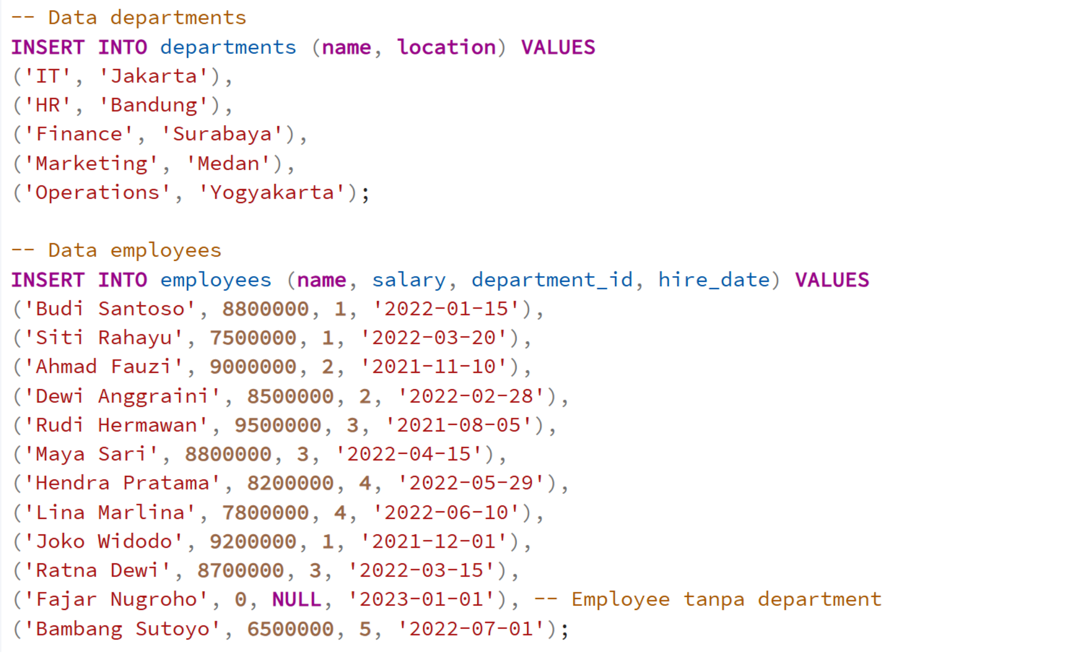
2. Membuat tabel-tabel



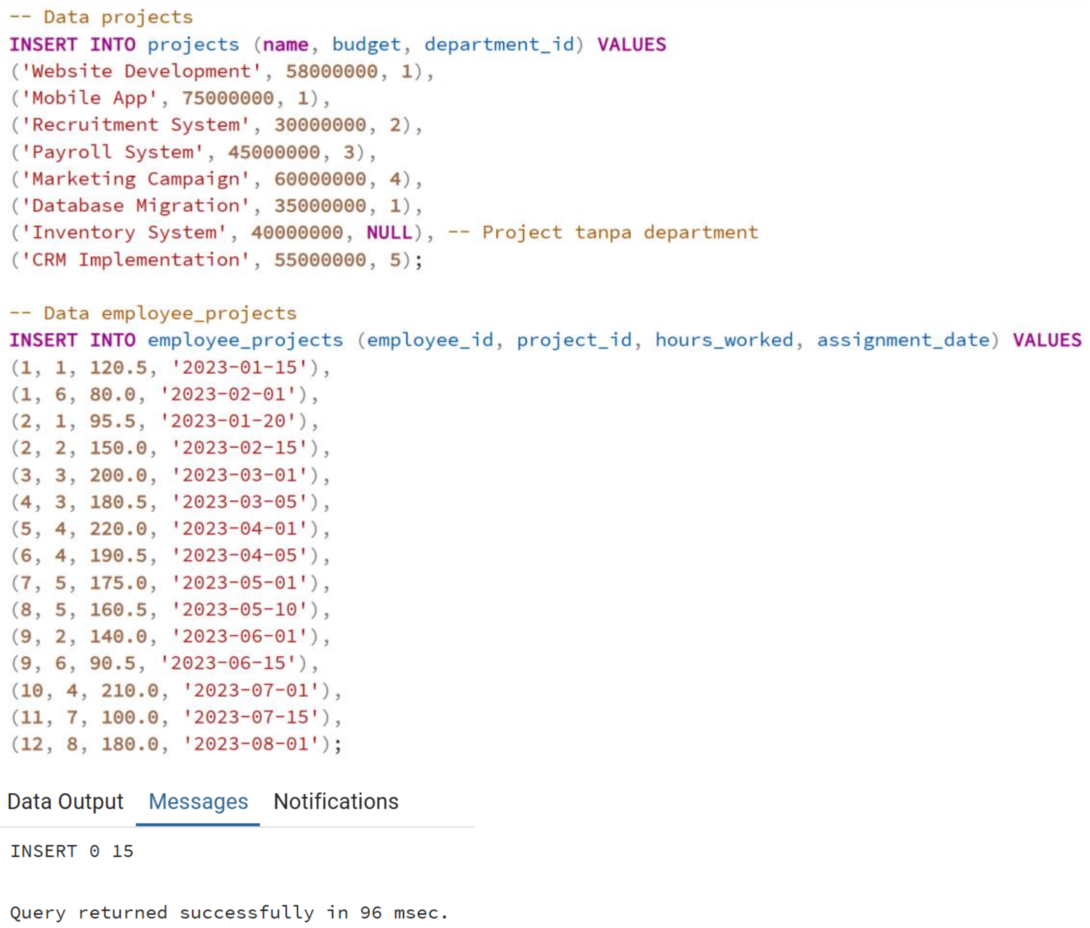
```sql
CREATE TABLE departments (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    location VARCHAR(100)
);

CREATE TABLE employees (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    salary NUMERIC(10,2),
    department_id INTEGER REFERENCES departments(id),
    hire_date DATE
);

CREATE TABLE projects (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    budget NUMERIC(15,2),
    department_id INTEGER REFERENCES departments(id)
);

CREATE TABLE employee_projects (
    employee_id INTEGER REFERENCES employees(id),
    project_id INTEGER REFERENCES projects(id),
    hours_worked NUMERIC(5,2),
    assignment_date DATE DEFAULT CURRENT_DATE,
    PRIMARY KEY (employee_id, project_id)
);
```

```
CREATE TABLE

Query returned successfully in 151 msec.
```

3. Insert Data Sample



```sql
-- Data departments
INSERT INTO departments (name, location) VALUES
('IT', 'Jakarta'),
('HR', 'Bandung'),
('Finance', 'Surabaya'),
('Marketing', 'Medan'),
('Operations', 'Yogyakarta');

-- Data employees
INSERT INTO employees (name, salary, department_id, hire_date) VALUES
('Budi Santoso', 8800000, 1, '2022-01-15'),
('Siti Rahayu', 7500000, 1, '2022-03-20'),
('Ahmad Fauzi', 9000000, 2, '2021-11-10'),
('Dewi Anggraini', 8500000, 2, '2022-02-28'),
('Rudi Hermawan', 9500000, 3, '2021-08-05'),
('Maya Sari', 8800000, 3, '2022-04-15'),
('Hendra Pratama', 8200000, 4, '2022-05-29'),
('Lina Marlina', 7800000, 4, '2022-06-10'),
('Joko Widodo', 9200000, 1, '2021-12-01'),
('Ratna Dewi', 8700000, 3, '2022-03-15'),
('Fajar Nugroho', 0, NULL, '2023-01-01'), -- Employee tanpa department
('Bambang Sutoyo', 6500000, 5, '2022-07-01');
```



```sql
-- Data projects
INSERT INTO projects (name, budget, department_id) VALUES
('Website Development', 58000000, 1),
('Mobile App', 75000000, 1),
('Recruitment System', 30000000, 2),
('Payroll System', 45000000, 3),
('Marketing Campaign', 60000000, 4),
('Database Migration', 35000000, 1),
('Inventory System', 40000000, NULL), -- Project tanpa department
('CRM Implementation', 55000000, 5);

-- Data employee_projects
INSERT INTO employee_projects (employee_id, project_id, hours_worked, assignment_date) VALUES
(1, 1, 120.5, '2023-01-15'),
(1, 6, 80.0, '2023-02-01'),
(2, 1, 95.5, '2023-01-20'),
(2, 2, 150.0, '2023-02-15'),
(3, 3, 200.0, '2023-03-01'),
(4, 3, 180.5, '2023-03-05'),
(5, 4, 220.0, '2023-04-01'),
(6, 4, 190.5, '2023-04-05'),
(7, 5, 175.0, '2023-05-01'),
(8, 5, 160.5, '2023-05-10'),
(9, 2, 140.0, '2023-06-01'),
(9, 6, 90.5, '2023-06-15'),
(10, 4, 210.0, '2023-07-01'),
(11, 7, 100.0, '2023-07-15'),
(12, 8, 180.0, '2023-08-01');
```

```
INSERT 0 15

Query returned successfully in 96 msec.
```

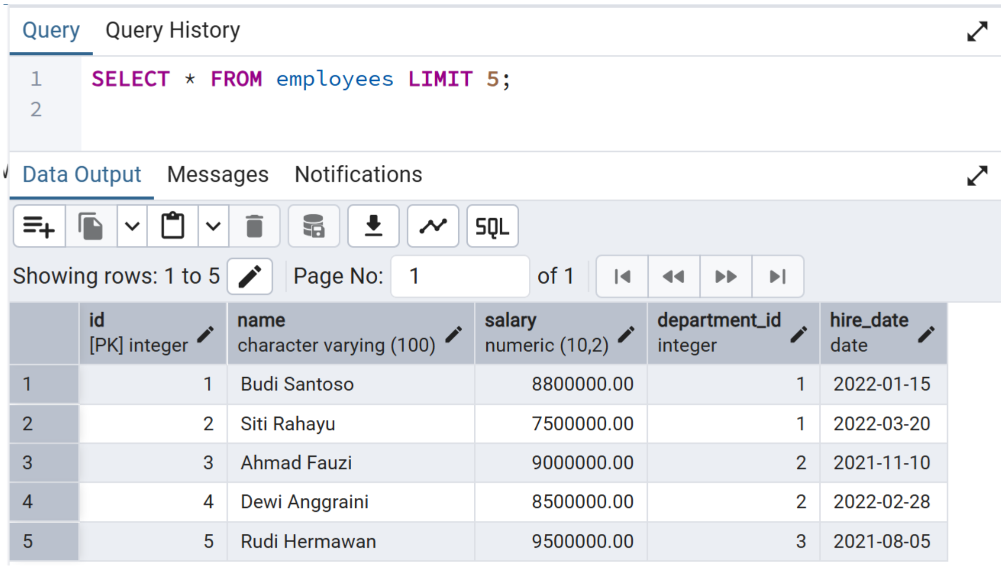
Studi Kasus: Sistem Manajemen Perusahaan XYZ

Database perusahaan dengan berbagai entitas: departments, employees, projects, dan hubungan many-to-many antara employees dan projects.

Praktikum 1: SELECT dengan LIMIT/OFFSET

Latihan Praktis

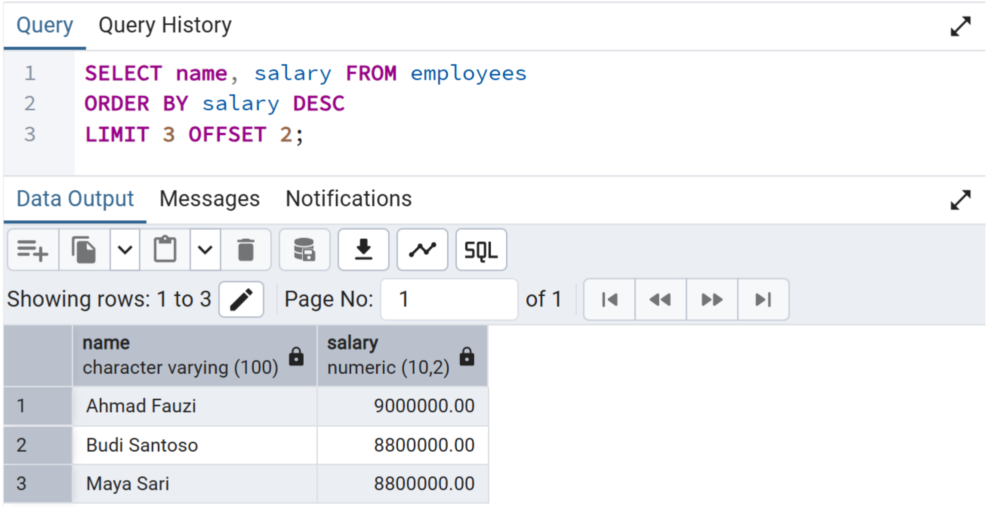
1. Basic SELECT dengan LIMIT



```sql
SELECT * FROM employees LIMIT 5;
```

	id [PK] integer	name character varying (100)	salary numeric (10,2)	department_id integer	hire_date date
1	1	Budi Santoso	8800000.00	1	2022-01-15
2	2	Siti Rahayu	7500000.00	1	2022-03-20
3	3	Ahmad Fauzi	9000000.00	2	2021-11-10
4	4	Dewi Anggraini	8500000.00	2	2022-02-28
5	5	Rudi Hermawan	9500000.00	3	2021-08-05

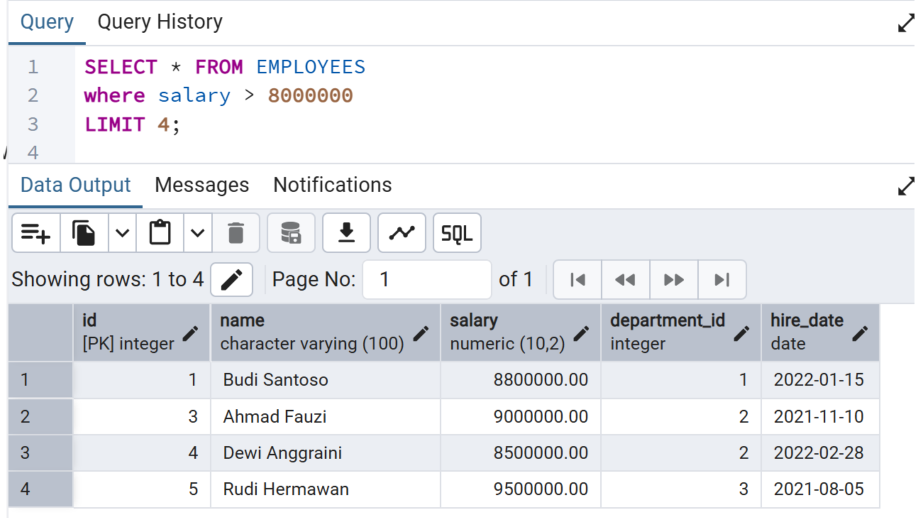
2. SELECT dengan ORDER BY dan LIMIT/OFFSET



```sql
SELECT name, salary FROM employees
ORDER BY salary DESC
LIMIT 3 OFFSET 2;
```

	name character varying (100)	salary numeric (10,2)
1	Ahmad Fauzi	9000000.00
2	Budi Santoso	8800000.00
3	Maya Sari	8800000.00

3. SELECT dengan WHERE dan LIMIT

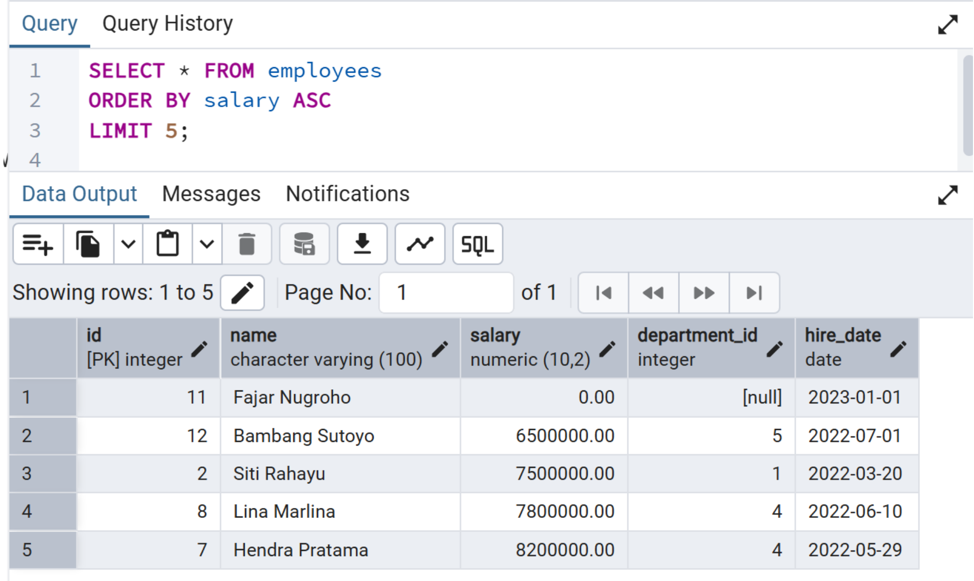


```sql
SELECT * FROM EMPLOYEES
where salary > 8000000
LIMIT 4;
```

	id [PK] integer	name character varying (100)	salary numeric (10,2)	department_id integer	hire_date date
1	1	Budi Santoso	8800000.00	1	2022-01-15
2	3	Ahmad Fauzi	9000000.00	2	2021-11-10
3	4	Dewi Anggraini	8500000.00	2	2022-02-28
4	5	Rudi Hermawan	9500000.00	3	2021-08-05

Soal Latihan Praktikum 1

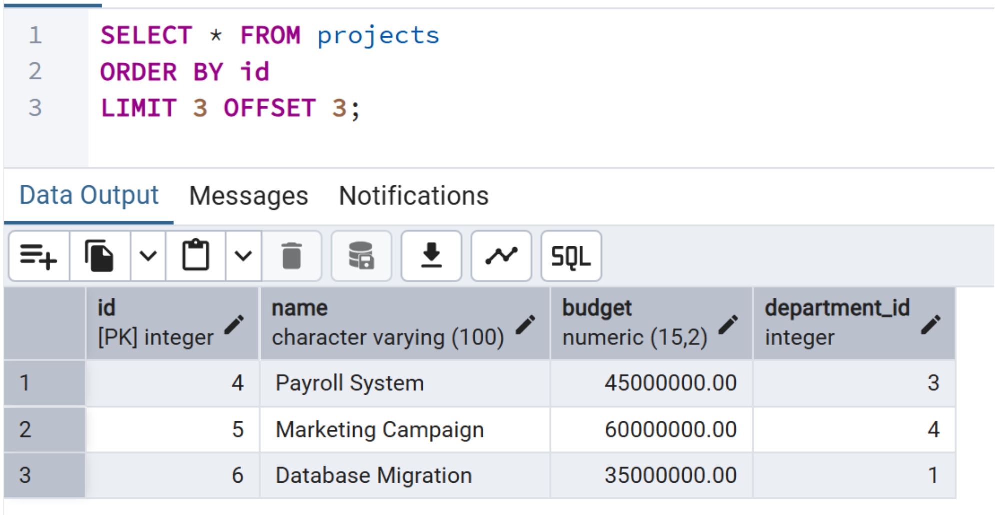
1. Tampilkan 5 employee dengan gaji terendah



```sql
SELECT * FROM employees
ORDER BY salary ASC
LIMIT 5;
```

	id [PK] integer	name character varying (100)	salary numeric (10,2)	department_id integer	hire_date date
1	11	Fajar Nugroho	0.00	[null]	2023-01-01
2	12	Bambang Sutoyo	6500000.00	5	2022-07-01
3	2	Siti Rahayu	7500000.00	1	2022-03-20
4	8	Lina Marlina	7800000.00	4	2022-06-10
5	7	Hendra Pratama	8200000.00	4	2022-05-29

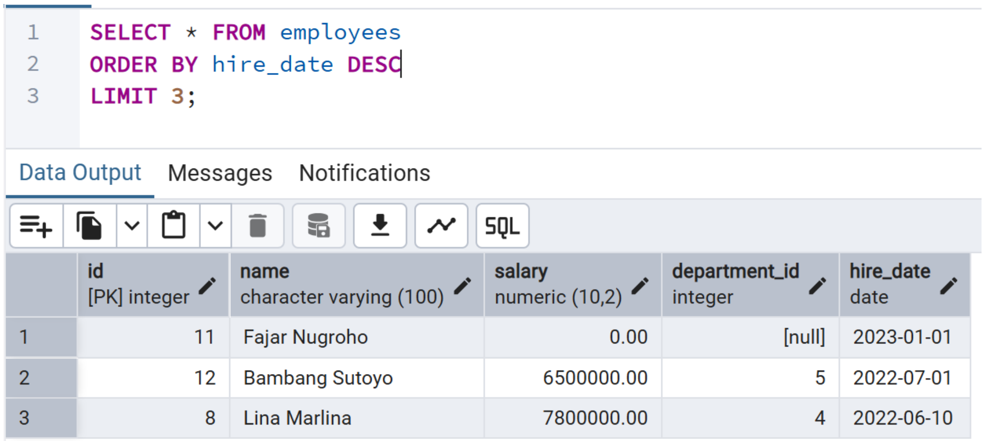
2. Tampilkan halaman 2 data projects (3 data per halaman)



```sql
SELECT * FROM projects
ORDER BY id
LIMIT 3 OFFSET 3;
```

	id [PK] integer	name character varying (100)	budget numeric (15,2)	department_id integer
1	4	Payroll System	45000000.00	3
2	5	Marketing Campaign	60000000.00	4
3	6	Database Migration	35000000.00	1

3. Tampilkan 3 employee yang dihire paling akhir

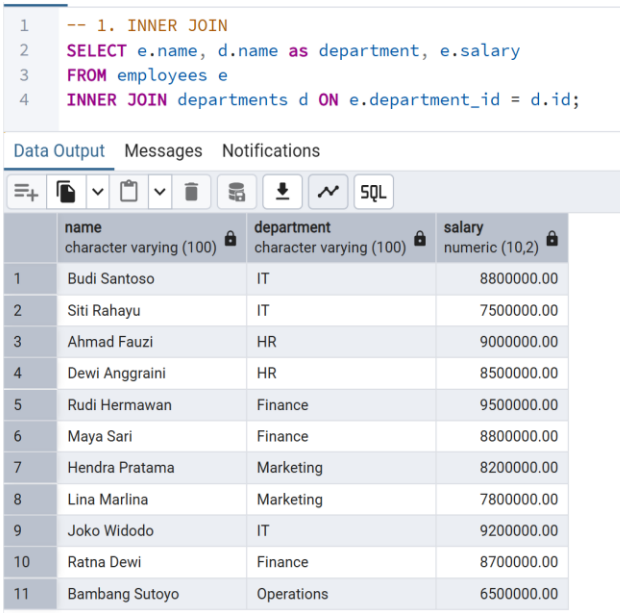


```sql
SELECT * FROM employees
ORDER BY hire_date DESC
LIMIT 3;
```

	id [PK] integer	name character varying (100)	salary numeric (10,2)	department_id integer	hire_date date
1	11	Fajar Nugroho	0.00	[null]	2023-01-01
2	12	Bambang Sutoyo	6500000.00	5	2022-07-01
3	8	Lina Marlina	7800000.00	4	2022-06-10

Praktikum 2: JOIN Operations

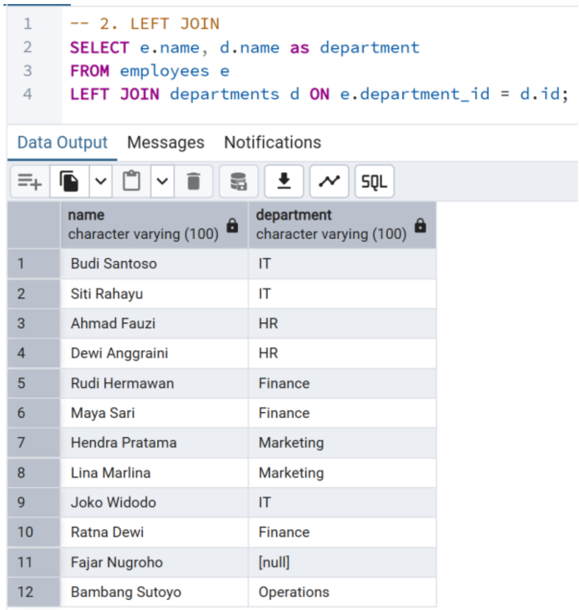
1. INNER JOIN



```sql
-- 1. INNER JOIN
SELECT e.name, d.name as department, e.salary
FROM employees e
INNER JOIN departments d ON e.department_id = d.id;
```

	name character varying (100)	department character varying (100)	salary numeric (10,2)
1	Budi Santoso	IT	8800000.00
2	Siti Rahayu	IT	7500000.00
3	Ahmad Fauzi	HR	9000000.00
4	Dewi Anggraini	HR	8500000.00
5	Rudi Hermawan	Finance	9500000.00
6	Maya Sari	Finance	8800000.00
7	Hendra Pratama	Marketing	8200000.00
8	Lina Marlina	Marketing	7800000.00
9	Joko Widodo	IT	9200000.00
10	Ratna Dewi	Finance	8700000.00
11	Bambang Sutoyo	Operations	6500000.00

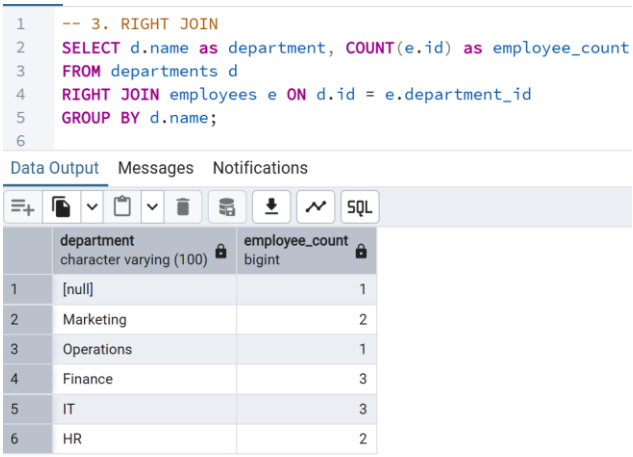
2. LEFT JOIN



```sql
-- 2. LEFT JOIN
SELECT e.name, d.name as department
FROM employees e
LEFT JOIN departments d ON e.department_id = d.id;
```

	name character varying (100)	department character varying (100)
1	Budi Santoso	IT
2	Siti Rahayu	IT
3	Ahmad Fauzi	HR
4	Dewi Anggraini	HR
5	Rudi Hermawan	Finance
6	Maya Sari	Finance
7	Hendra Pratama	Marketing
8	Lina Marlina	Marketing
9	Joko Widodo	IT
10	Ratna Dewi	Finance
11	Fajar Nugroho	[null]
12	Bambang Sutoyo	Operations

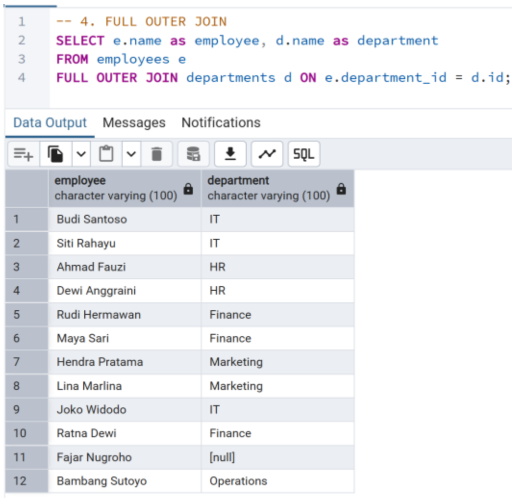
3. RIGHT JOIN



```sql
-- 3. RIGHT JOIN
SELECT d.name as department, COUNT(e.id) as employee_count
FROM departments d
RIGHT JOIN employees e ON d.id = e.department_id
GROUP BY d.name;

```

	department character varying (100)	employee_count bigint
1	[null]	1
2	Marketing	2
3	Operations	1
4	Finance	3
5	IT	3
6	HR	2

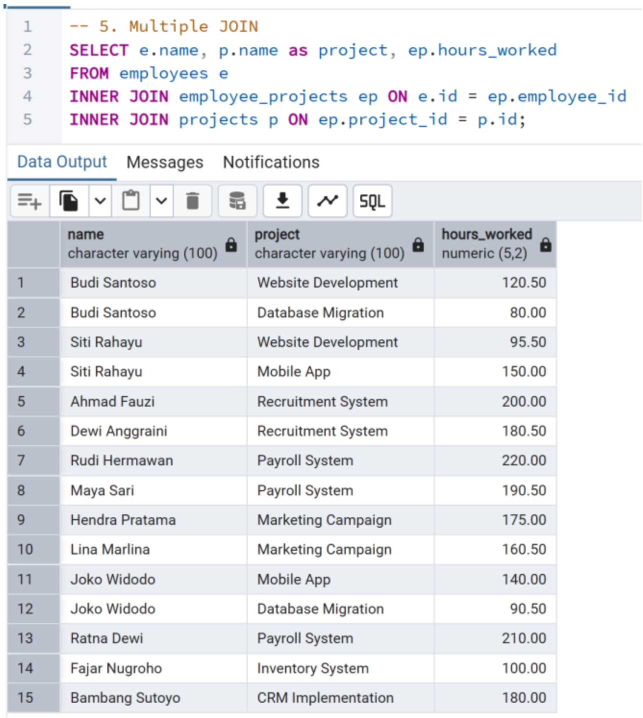
4. FULL OUTER JOIN



```sql
-- 4. FULL OUTER JOIN
SELECT e.name as employee, d.name as department
FROM employees e
FULL OUTER JOIN departments d ON e.department_id = d.id;
```

	employee character varying (100)	department character varying (100)
1	Budi Santoso	IT
2	Siti Rahayu	IT
3	Ahmad Fauzi	HR
4	Dewi Anggraini	HR
5	Rudi Hermawan	Finance
6	Maya Sari	Finance
7	Hendra Pratama	Marketing
8	Lina Marlina	Marketing
9	Joko Widodo	IT
10	Ratna Dewi	Finance
11	Fajar Nugroho	[null]
12	Bambang Sutoyo	Operations

5. Multiple JOIN

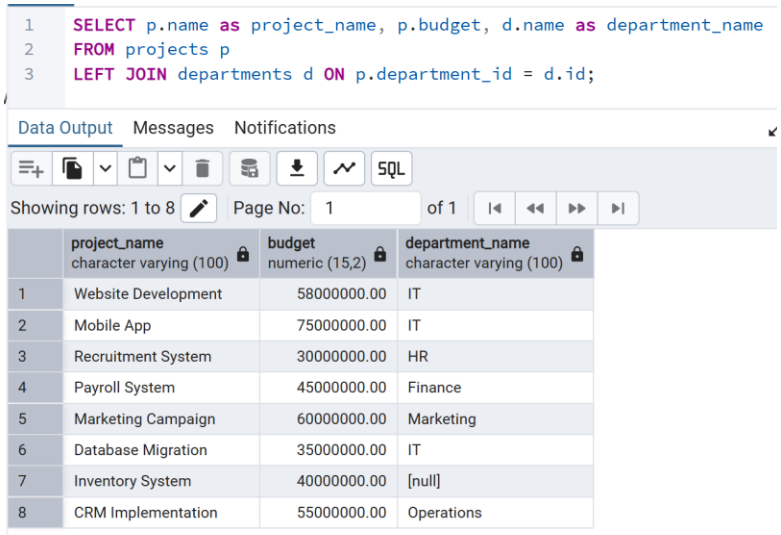


```sql
-- 5. Multiple JOIN
SELECT e.name, p.name as project, ep.hours_worked
FROM employees e
INNER JOIN employee_projects ep ON e.id = ep.employee_id
INNER JOIN projects p ON ep.project_id = p.id;
```

	name character varying (100)	project character varying (100)	hours_worked numeric (5,2)
1	Budi Santoso	Website Development	120.50
2	Budi Santoso	Database Migration	80.00
3	Siti Rahayu	Website Development	95.50
4	Siti Rahayu	Mobile App	150.00
5	Ahmad Fauzi	Recruitment System	200.00
6	Dewi Anggraini	Recruitment System	180.50
7	Rudi Hermawan	Payroll System	220.00
8	Maya Sari	Payroll System	190.50
9	Hendra Pratama	Marketing Campaign	175.00
10	Lina Marlina	Marketing Campaign	160.50
11	Joko Widodo	Mobile App	140.00
12	Joko Widodo	Database Migration	90.50
13	Ratna Dewi	Payroll System	210.00
14	Fajar Nugroho	Inventory System	100.00
15	Bambang Sutoyo	CRM Implementation	180.00

Soal Latihan Praktikum 2

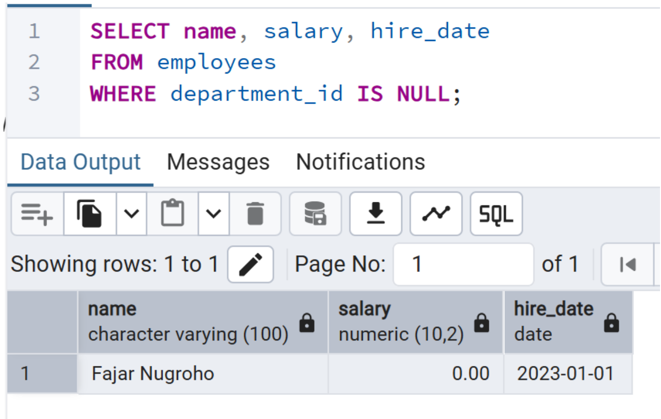
1. Tampilkan semua projects beserta department penanggung jawabnya



```sql
SELECT p.name as project_name, p.budget, d.name as department_name
FROM projects p
LEFT JOIN departments d ON p.department_id = d.id;
```

	project_name character varying (100)	budget numeric (15,2)	department_name character varying (100)
1	Website Development	58000000.00	IT
2	Mobile App	75000000.00	IT
3	Recruitment System	30000000.00	HR
4	Payroll System	45000000.00	Finance
5	Marketing Campaign	60000000.00	Marketing
6	Database Migration	35000000.00	IT
7	Inventory System	40000000.00	[null]
8	CRM Implementation	55000000.00	Operations

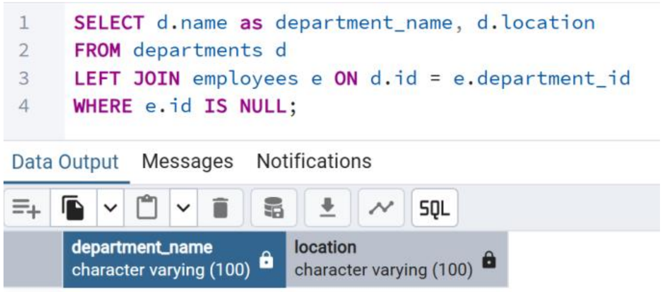
2. Tampilkan employee yang tidak memiliki department



```sql
SELECT name, salary, hire_date
FROM employees
WHERE department_id IS NULL;
```

	name character varying (100)	salary numeric (10,2)	hire_date date
1	Fajar Nugroho	0.00	2023-01-01

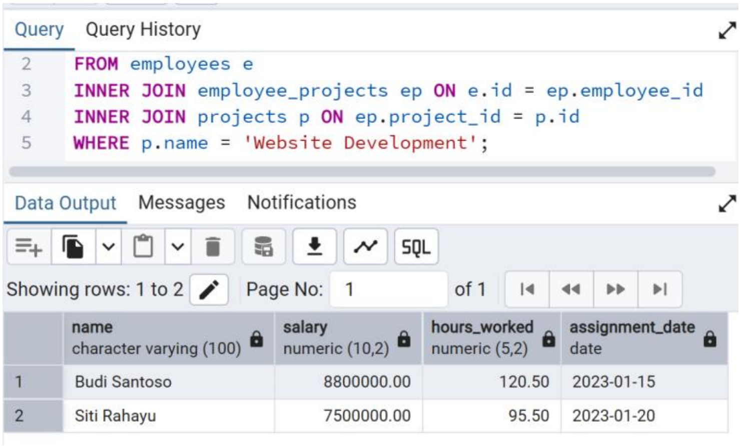
3. Tampilkan department yang tidak memiliki employee



```sql
SELECT d.name as department_name, d.location
FROM departments d
LEFT JOIN employees e ON d.id = e.department_id
WHERE e.id IS NULL;
```

	department_name character varying (100)	location character varying (100)

4. Tampilkan employee yang bekerja di project 'Website Development'

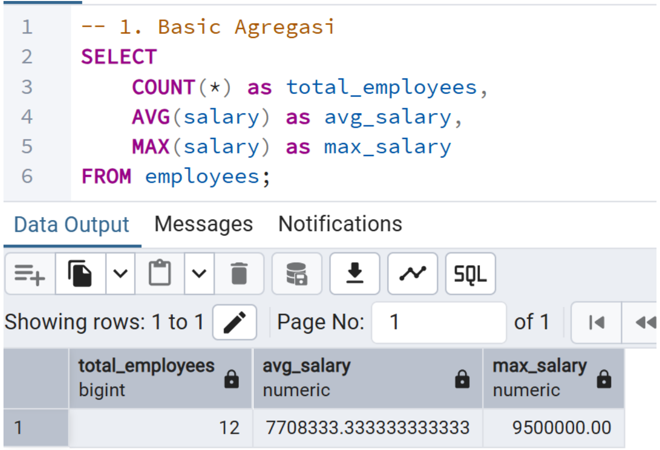


```sql
FROM employees e
INNER JOIN employee_projects ep ON e.id = ep.employee_id
INNER JOIN projects p ON ep.project_id = p.id
WHERE p.name = 'Website Development';
```

	name character varying (100)	salary numeric (10,2)	hours_worked numeric (5,2)	assignment_date date
1	Budi Santoso	8800000.00	120.50	2023-01-15
2	Siti Rahayu	7500000.00	95.50	2023-01-20

Praktikum 3: Fungsi Agregasi

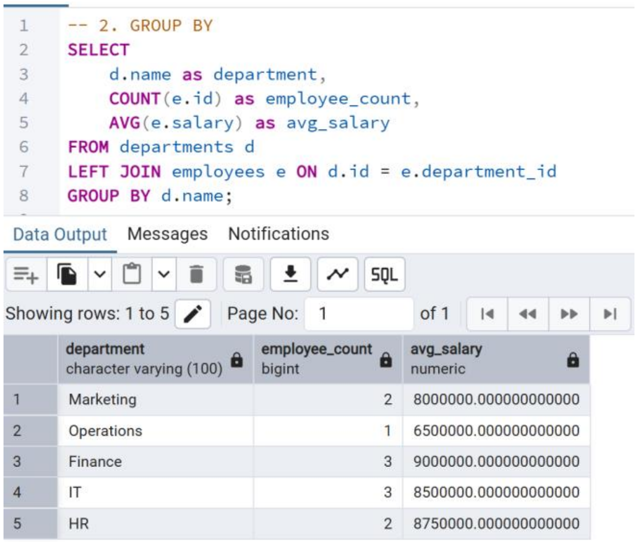
1. Basic Agregasi



```sql
-- 1. Basic Agregasi
SELECT
    COUNT(*) as total_employees,
    AVG(salary) as avg_salary,
    MAX(salary) as max_salary
FROM employees;
```

	total_employees bigint	avg_salary numeric	max_salary numeric
1	12	7708333.333333333333	9500000.00

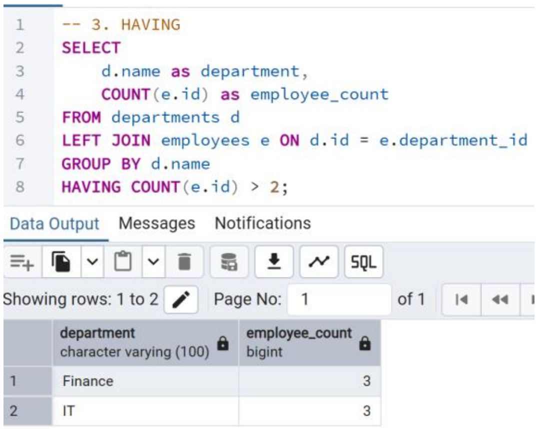
2. GROUP BY



```sql
-- 2. GROUP BY
SELECT
    d.name as department,
    COUNT(e.id) as employee_count,
    AVG(e.salary) as avg_salary
FROM departments d
LEFT JOIN employees e ON d.id = e.department_id
GROUP BY d.name;
```

	department character varying (100)	employee_count bigint	avg_salary numeric
1	Marketing	2	8000000.000000000000
2	Operations	1	6500000.000000000000
3	Finance	3	9000000.000000000000
4	IT	3	8500000.000000000000
5	HR	2	8750000.000000000000

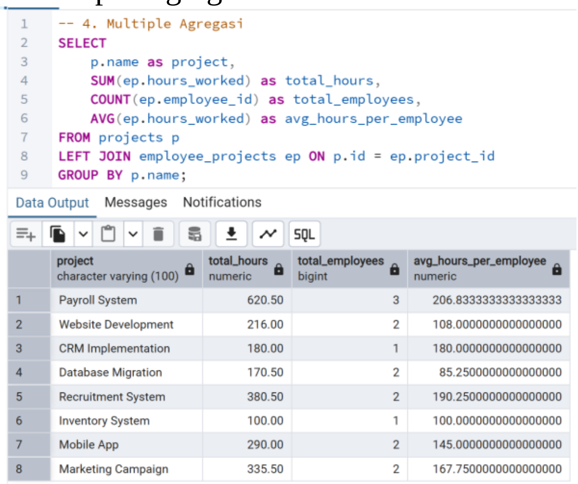
3. HAVING



```sql
-- 3. HAVING
SELECT
    d.name as department,
    COUNT(e.id) as employee_count
FROM departments d
LEFT JOIN employees e ON d.id = e.department_id
GROUP BY d.name
HAVING COUNT(e.id) > 2;
```

	department character varying (100)	employee_count bigint
1	Finance	3
2	IT	3

4. Multiple Agregasi

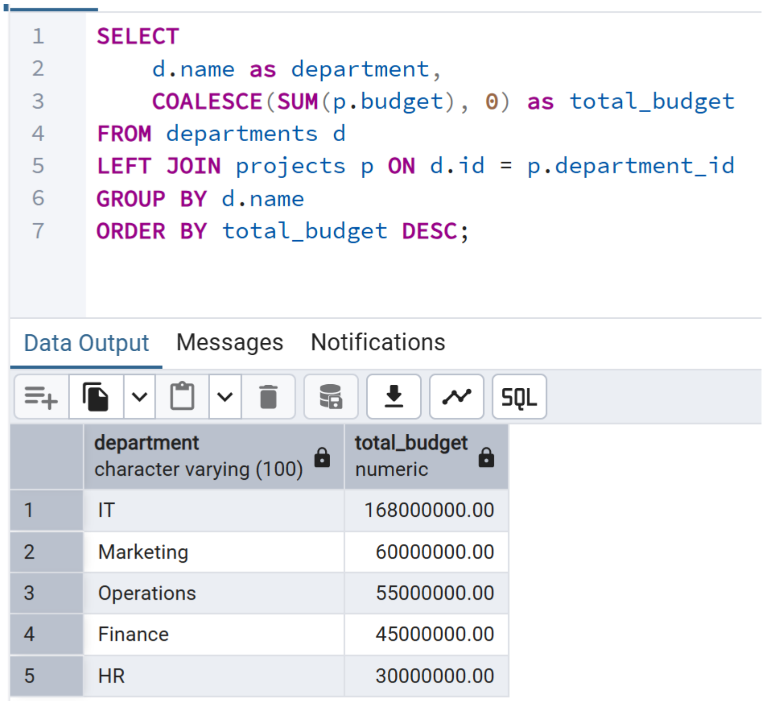


```sql
-- 4. Multiple Agregasi
SELECT
    p.name as project,
    SUM(ep.hours_worked) as total_hours,
    COUNT(ep.employee_id) as total_employees,
    AVG(ep.hours_worked) as avg_hours_per_employee
FROM projects p
LEFT JOIN employee_projects ep ON p.id = ep.project_id
GROUP BY p.name;
```

	project character varying (100)	total_hours numeric	total_employees bigint	avg_hours_per_employee numeric
1	Payroll System	620.50	3	206.8333333333333333
2	Website Development	216.00	2	108.0000000000000000
3	CRM Implementation	180.00	1	180.0000000000000000
4	Database Migration	170.50	2	85.2500000000000000
5	Recruitment System	380.50	2	190.2500000000000000
6	Inventory System	100.00	1	100.0000000000000000
7	Mobile App	290.00	2	145.0000000000000000
8	Marketing Campaign	335.50	2	167.7500000000000000

Soal Latihan Praktikum 3

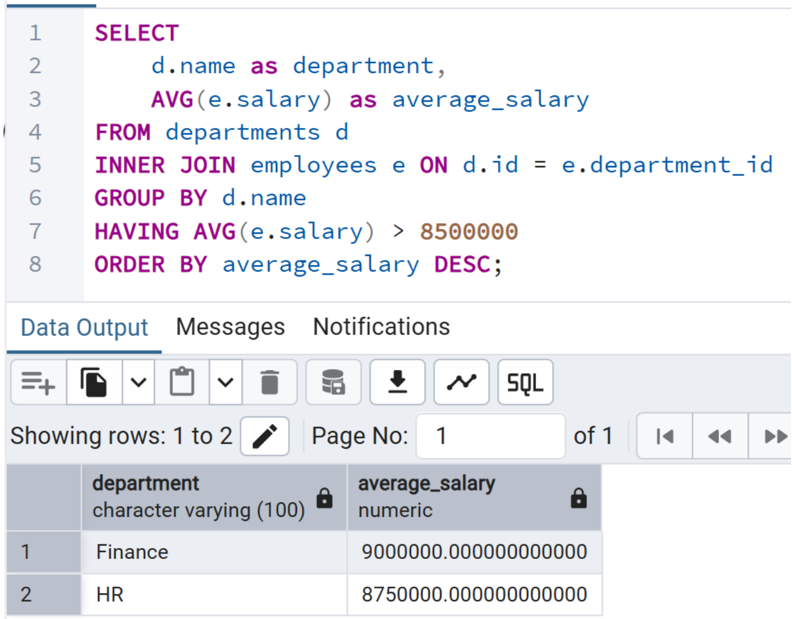
1. Tampilkan total budget projects per department



```
SELECT
    d.name as department,
    COALESCE(SUM(p.budget), 0) as total_budget
FROM departments d
LEFT JOIN projects p ON d.id = p.department_id
GROUP BY d.name
ORDER BY total_budget DESC;
```

	department character varying (100)	total_budget numeric
1	IT	168000000.00
2	Marketing	60000000.00
3	Operations	55000000.00
4	Finance	45000000.00
5	HR	30000000.00

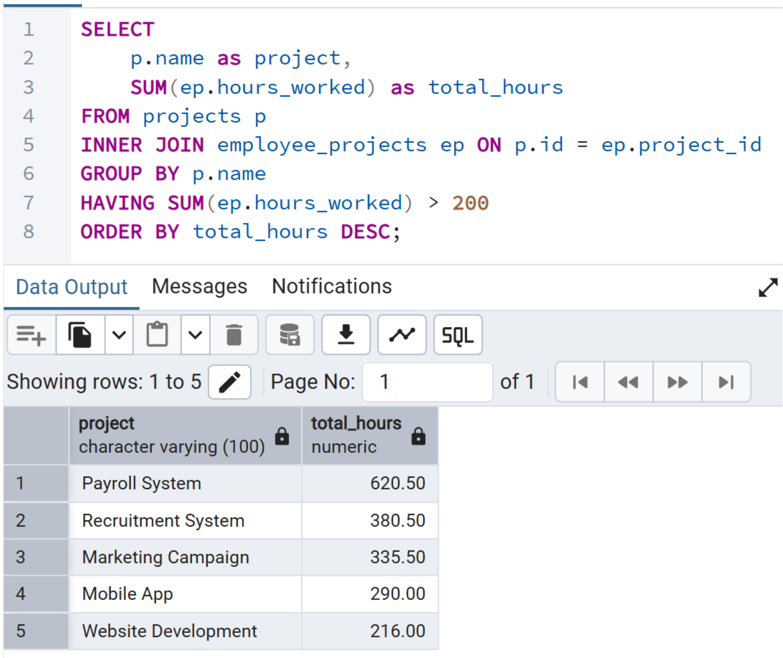
2. Tampilkan department dengan rata-rata gaji di atas 8.5 juta



```
SELECT
    d.name as department,
    AVG(e.salary) as average_salary
FROM departments d
INNER JOIN employees e ON d.id = e.department_id
GROUP BY d.name
HAVING AVG(e.salary) > 8500000
ORDER BY average_salary DESC;
```

	department character varying (100)	average_salary numeric
1	Finance	9000000.000000000000
2	HR	8750000.000000000000

3. Tampilkan project dengan total jam kerja lebih dari 200 jam

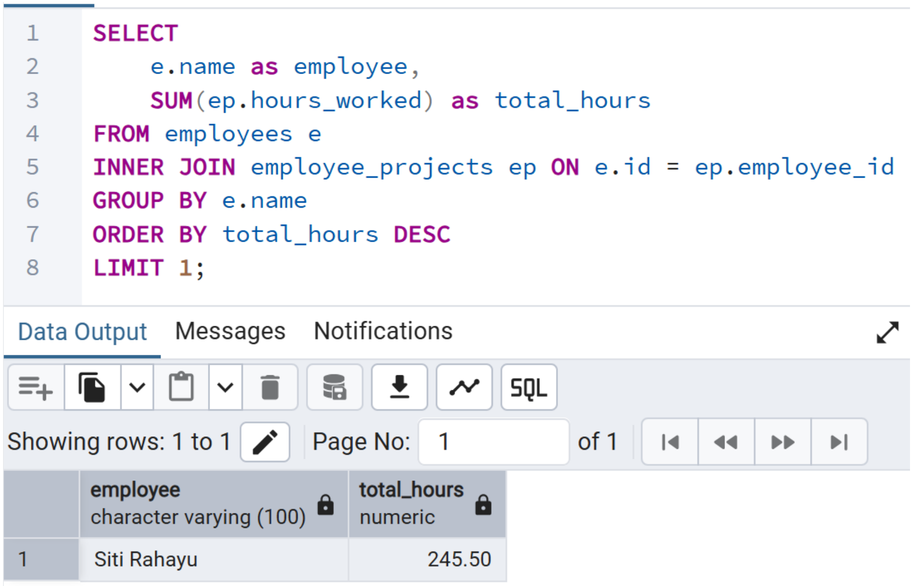


```sql
SELECT
    p.name as project,
    SUM(ep.hours_worked) as total_hours
FROM projects p
INNER JOIN employee_projects ep ON p.id = ep.project_id
GROUP BY p.name
HAVING SUM(ep.hours_worked) > 200
ORDER BY total_hours DESC;
```

Showing rows: 1 to 5 Page No: 1 of 1

	project character varying (100)	total_hours numeric
1	Payroll System	620.50
2	Recruitment System	380.50
3	Marketing Campaign	335.50
4	Mobile App	290.00
5	Website Development	216.00

4. Tampilkan employee dengan total jam kerja terbanyak



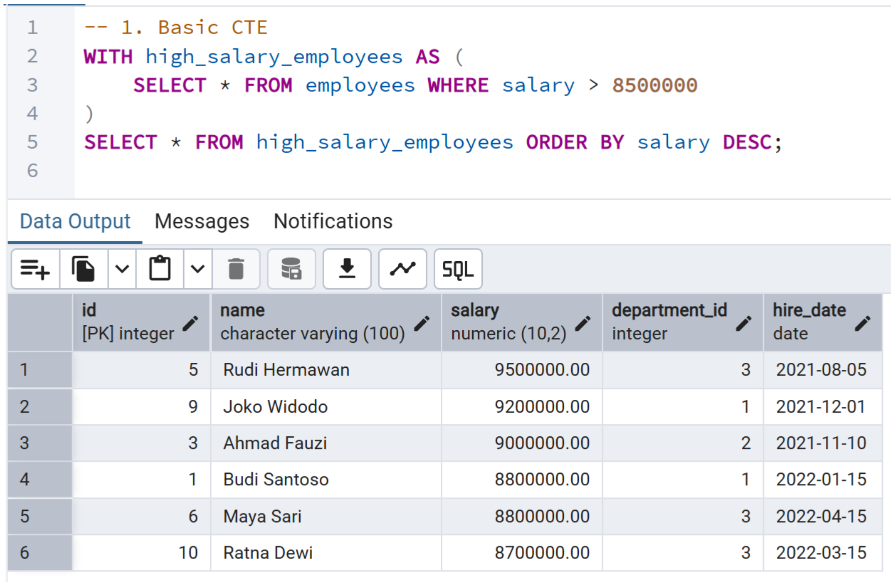
```sql
SELECT
    e.name as employee,
    SUM(ep.hours_worked) as total_hours
FROM employees e
INNER JOIN employee_projects ep ON e.id = ep.employee_id
GROUP BY e.name
ORDER BY total_hours DESC
LIMIT 1;
```

Showing rows: 1 to 1 Page No: 1 of 1

	employee character varying (100)	total_hours numeric
1	Siti Rahayu	245.50

Praktikum 4: Common Table Expression (CTE)

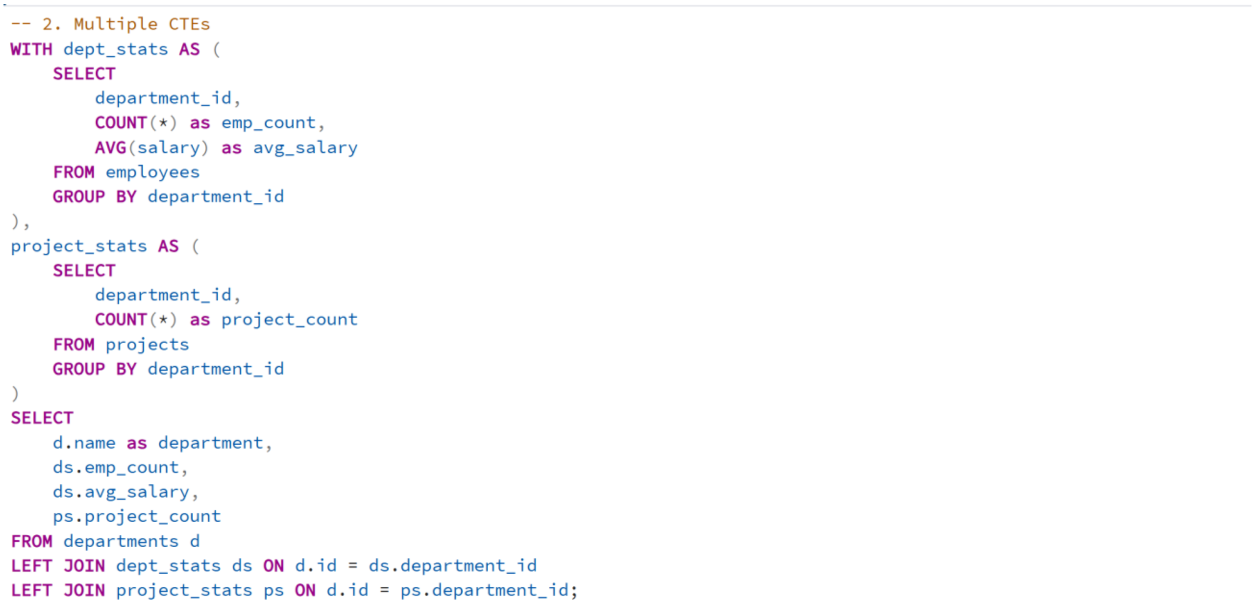
1. Basic CTE



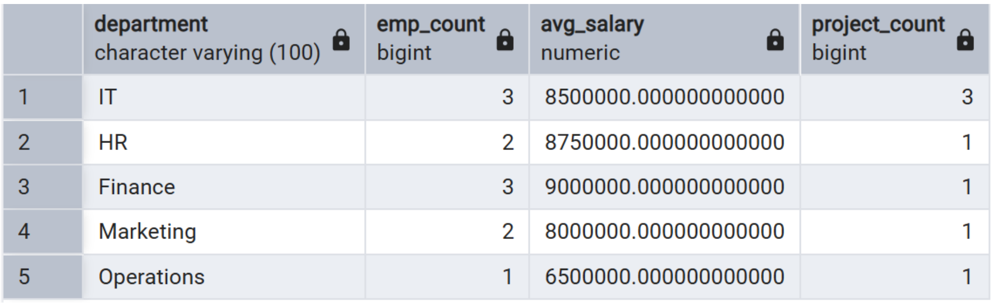
```sql
-- 1. Basic CTE
WITH high_salary_employees AS (
    SELECT * FROM employees WHERE salary > 8500000
)
SELECT * FROM high_salary_employees ORDER BY salary DESC;

```

	id [PK] integer	name character varying (100)	salary numeric (10,2)	department_id integer	hire_date date
1	5	Rudi Hermawan	9500000.00	3	2021-08-05
2	9	Joko Widodo	9200000.00	1	2021-12-01
3	3	Ahmad Fauzi	9000000.00	2	2021-11-10
4	1	Budi Santoso	8800000.00	1	2022-01-15
5	6	Maya Sari	8800000.00	3	2022-04-15
6	10	Ratna Dewi	8700000.00	3	2022-03-15

2. Multiple CTEs



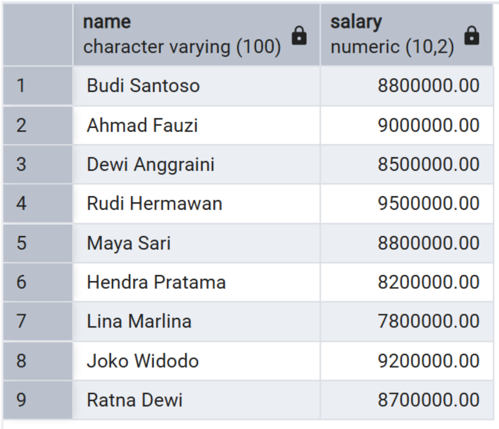
```sql
-- 2. Multiple CTEs
WITH dept_stats AS (
    SELECT
        department_id,
        COUNT(*) as emp_count,
        AVG(salary) as avg_salary
    FROM employees
    GROUP BY department_id
),
project_stats AS (
    SELECT
        department_id,
        COUNT(*) as project_count
    FROM projects
    GROUP BY department_id
)
SELECT
    d.name as department,
    ds.emp_count,
    ds.avg_salary,
    ps.project_count
FROM departments d
LEFT JOIN dept_stats ds ON d.id = ds.department_id
LEFT JOIN project_stats ps ON d.id = ps.department_id;
```



	department character varying (100)	emp_count bigint	avg_salary numeric	project_count bigint
1	IT	3	8500000.000000000000	3
2	HR	2	8750000.000000000000	1
3	Finance	3	9000000.000000000000	1
4	Marketing	2	8000000.000000000000	1
5	Operations	1	6500000.000000000000	1

3. CTE vs Subquery

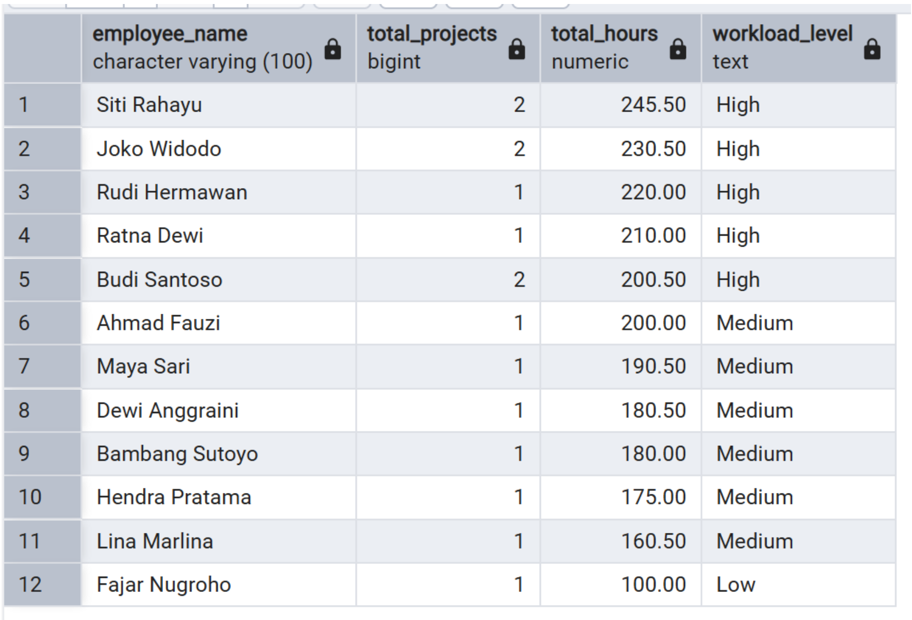
```
-- 3. CTE vs Subquery

-- Subquery (lebih rumit)
SELECT name, salary
FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);

-- CTE (lebih mudah dibaca)
WITH avg_salary AS (
    SELECT AVG(salary) as avg_sal FROM employees
)
SELECT name, salary
FROM employees, avg_salary
WHERE salary > avg_sal;
```



	name character varying (100) 🔒	salary numeric (10,2) 🔒
1	Budi Santoso	8800000.00
2	Ahmad Fauzi	9000000.00
3	Dewi Anggraini	8500000.00
4	Rudi Hermawan	9500000.00
5	Maya Sari	8800000.00
6	Hendra Pratama	8200000.00
7	Lina Marlina	7800000.00
8	Joko Widodo	9200000.00
9	Ratna Dewi	8700000.00

4. CTE Complex Query

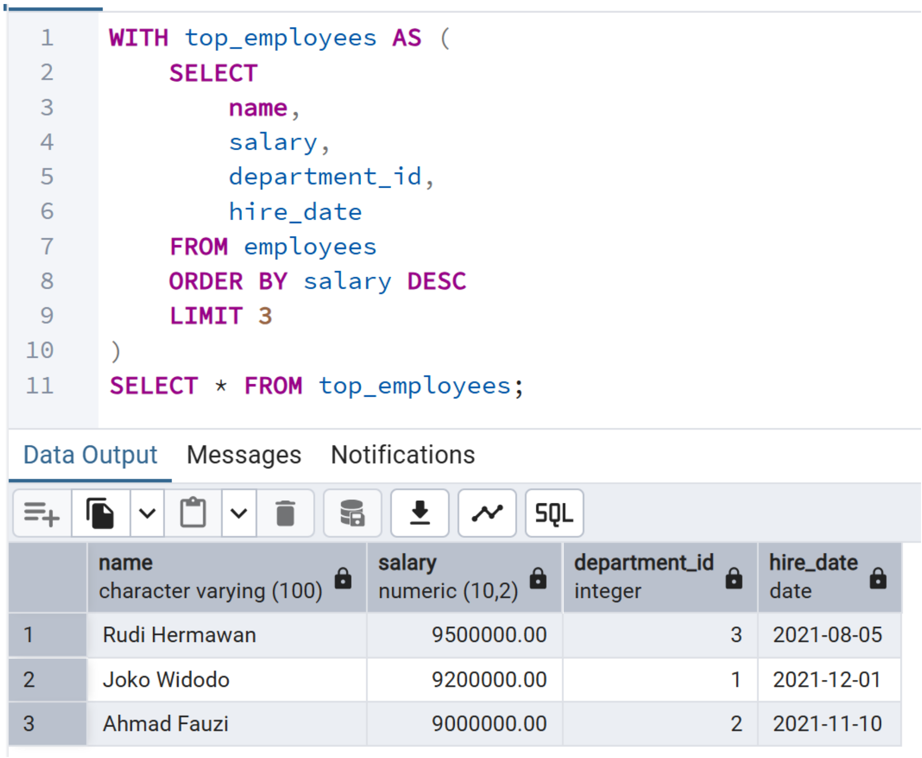
```
-- 4. CTE Complex Query
WITH employee_project_stats AS (
    SELECT
        e.name as employee_name,
        COUNT(ep.project_id) as total_projects,
        SUM(ep.hours_worked) as total_hours
    FROM employees e
    LEFT JOIN employee_projects ep ON e.id = ep.employee_id
    GROUP BY e.name
)
SELECT
    employee_name,
    total_projects,
    total_hours,
    CASE
        WHEN total_hours > 200 THEN 'High'
        WHEN total_hours > 100 THEN 'Medium'
        ELSE 'Low'
    END as workload_level
FROM employee_project_stats
ORDER BY total_hours DESC;
```



	employee_name character varying (100)	total_projects bigint	total_hours numeric	workload_level text
1	Siti Rahayu	2	245.50	High
2	Joko Widodo	2	230.50	High
3	Rudi Hermawan	1	220.00	High
4	Ratna Dewi	1	210.00	High
5	Budi Santoso	2	200.50	High
6	Ahmad Fauzi	1	200.00	Medium
7	Maya Sari	1	190.50	Medium
8	Dewi Anggraini	1	180.50	Medium
9	Bambang Sutoyo	1	180.00	Medium
10	Hendra Pratama	1	175.00	Medium
11	Lina Marlina	1	160.50	Medium
12	Fajar Nugroho	1	100.00	Low

Soal Latihan Praktikum 4

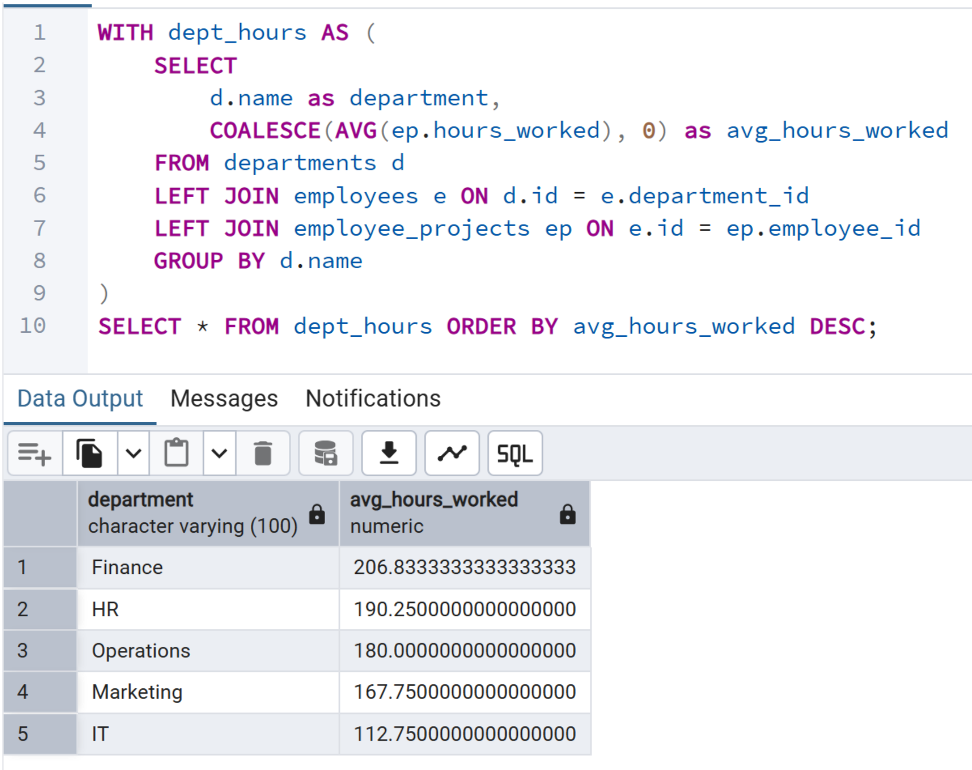
1. Buat CTE untuk menampilkan top 3 employee dengan gaji tertinggi



```
WITH top_employees AS (
    SELECT
        name,
        salary,
        department_id,
        hire_date
    FROM employees
    ORDER BY salary DESC
    LIMIT 3
)
SELECT * FROM top_employees;
```

	name character varying (100)	salary numeric (10,2)	department_id integer	hire_date date
1	Rudi Hermawan	9500000.00	3	2021-08-05
2	Joko Widodo	9200000.00	1	2021-12-01
3	Ahmad Fauzi	9000000.00	2	2021-11-10

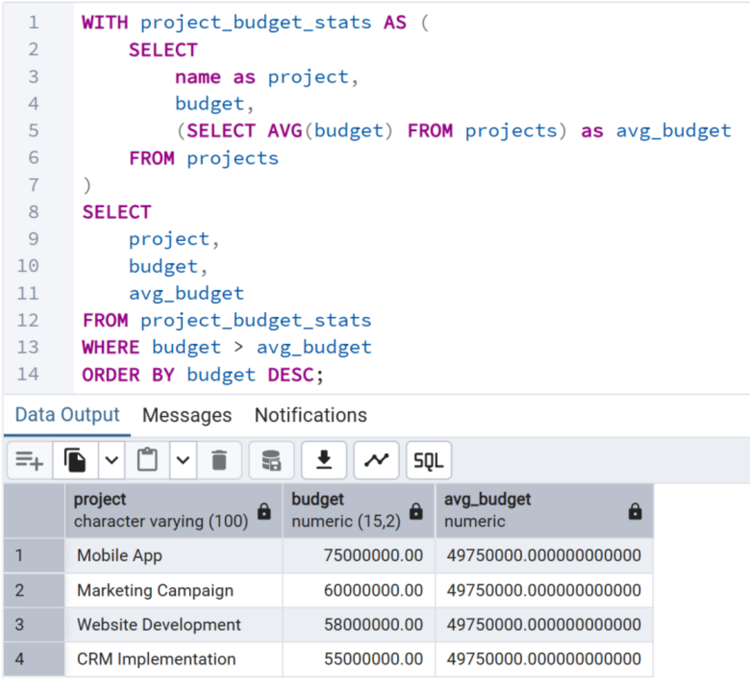
2. Buat CTE untuk menghitung rata-rata jam kerja per department



```sql
WITH dept_hours AS (
    SELECT
        d.name as department,
        COALESCE(AVG(ep.hours_worked), 0) as avg_hours_worked
    FROM departments d
    LEFT JOIN employees e ON d.id = e.department_id
    LEFT JOIN employee_projects ep ON e.id = ep.employee_id
    GROUP BY d.name
)
SELECT * FROM dept_hours ORDER BY avg_hours_worked DESC;
```

	department character varying (100)	avg_hours_worked numeric
1	Finance	206.8333333333333333
2	HR	190.2500000000000000
3	Operations	180.0000000000000000
4	Marketing	167.7500000000000000
5	IT	112.7500000000000000

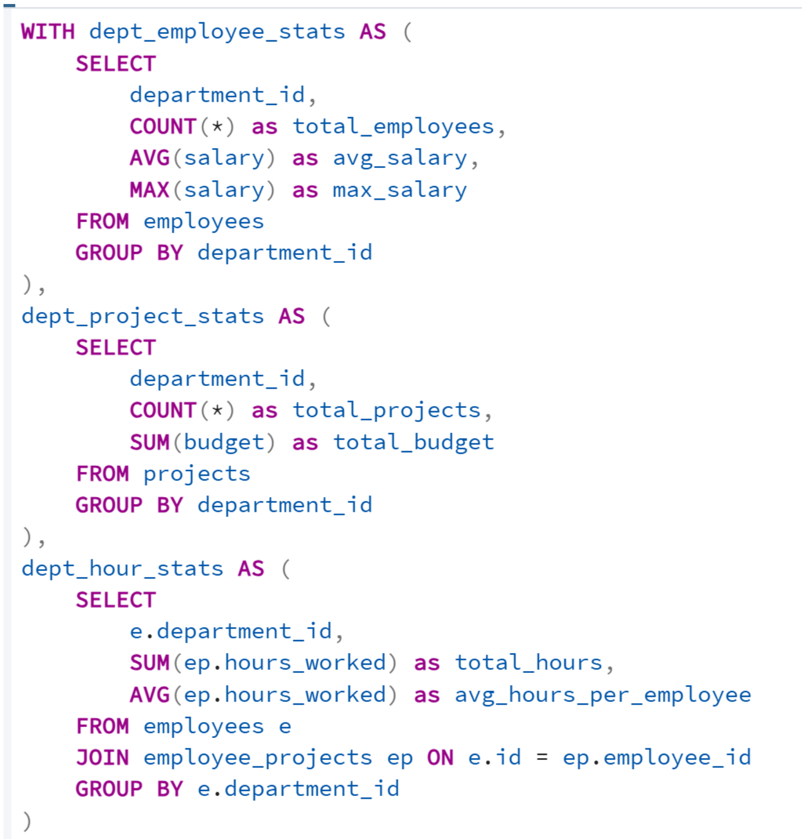
3. Buat CTE untuk menampilkan project dengan budget di atas rata-rata



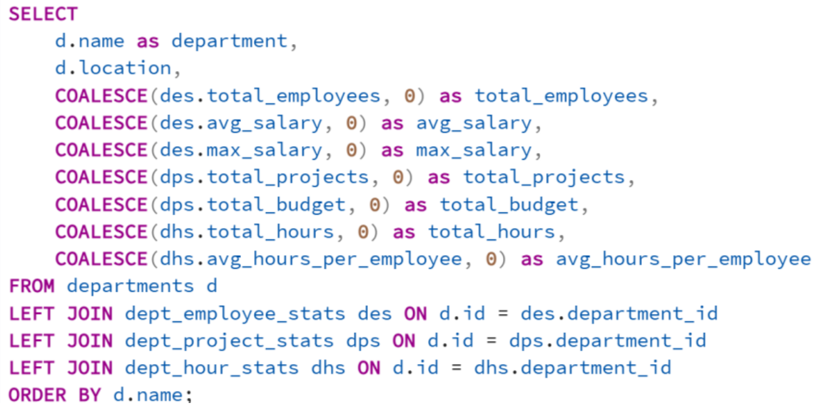
```sql
WITH project_budget_stats AS (
    SELECT
        name as project,
        budget,
        (SELECT AVG(budget) FROM projects) as avg_budget
    FROM projects
)
SELECT
    project,
    budget,
    avg_budget
FROM project_budget_stats
WHERE budget > avg_budget
ORDER BY budget DESC;
```

	project character varying (100)	budget numeric (15,2)	avg_budget numeric
1	Mobile App	75000000.00	49750000.000000000000
2	Marketing Campaign	60000000.00	49750000.000000000000
3	Website Development	58000000.00	49750000.000000000000
4	CRM Implementation	55000000.00	49750000.000000000000

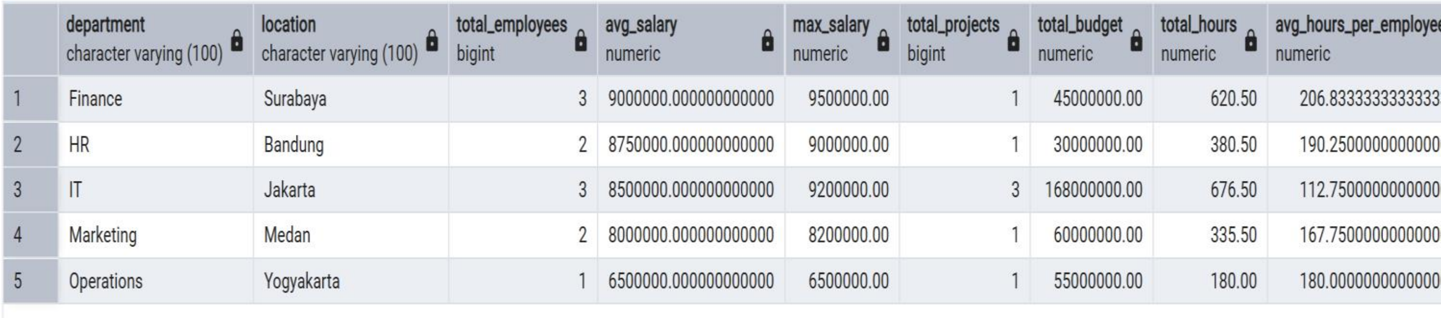
4. Buat CTE berjenjang (multiple CTEs) untuk analisis department lengkap



```
WITH dept_employee_stats AS (
    SELECT
        department_id,
        COUNT(*) as total_employees,
        AVG(salary) as avg_salary,
        MAX(salary) as max_salary
    FROM employees
    GROUP BY department_id
),
dept_project_stats AS (
    SELECT
        department_id,
        COUNT(*) as total_projects,
        SUM(budget) as total_budget
    FROM projects
    GROUP BY department_id
),
dept_hour_stats AS (
    SELECT
        e.department_id,
        SUM(ep.hours_worked) as total_hours,
        AVG(ep.hours_worked) as avg_hours_per_employee
    FROM employees e
    JOIN employee_projects ep ON e.id = ep.employee_id
    GROUP BY e.department_id
)
```



```
SELECT
    d.name as department,
    d.location,
    COALESCE(des.total_employees, 0) as total_employees,
    COALESCE(des.avg_salary, 0) as avg_salary,
    COALESCE(des.max_salary, 0) as max_salary,
    COALESCE(dps.total_projects, 0) as total_projects,
    COALESCE(dps.total_budget, 0) as total_budget,
    COALESCE(dhs.total_hours, 0) as total_hours,
    COALESCE(dhs.avg_hours_per_employee, 0) as avg_hours_per_employee
FROM departments d
LEFT JOIN dept_employee_stats des ON d.id = des.department_id
LEFT JOIN dept_project_stats dps ON d.id = dps.department_id
LEFT JOIN dept_hour_stats dhs ON d.id = dhs.department_id
ORDER BY d.name;
```

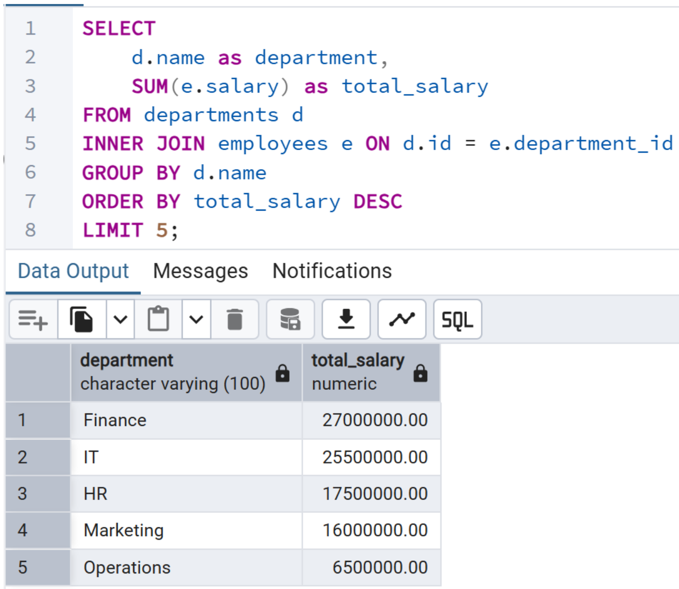


	department character varying (100)	location character varying (100)	total_employees bigint	avg_salary numeric	max_salary numeric	total_projects bigint	total_budget numeric	total_hours numeric	avg_hours_per_employe numeric
1	Finance	Surabaya	3	9000000.000000000000	9500000.00	1	45000000.00	620.50	206.8333333333333
2	HR	Bandung	2	8750000.000000000000	9000000.00	1	30000000.00	380.50	190.2500000000000
3	IT	Jakarta	3	8500000.000000000000	9200000.00	3	168000000.00	676.50	112.7500000000000
4	Marketing	Medan	2	8000000.000000000000	8200000.00	1	60000000.00	335.50	167.7500000000000
5	Operations	Yogyakarta	1	6500000.000000000000	6500000.00	1	55000000.00	180.00	180.0000000000000

Tugas Akhir Integrasi

Buat query lengkap yang menggabungkan semua konsep untuk menjawab pertanyaan berikut:

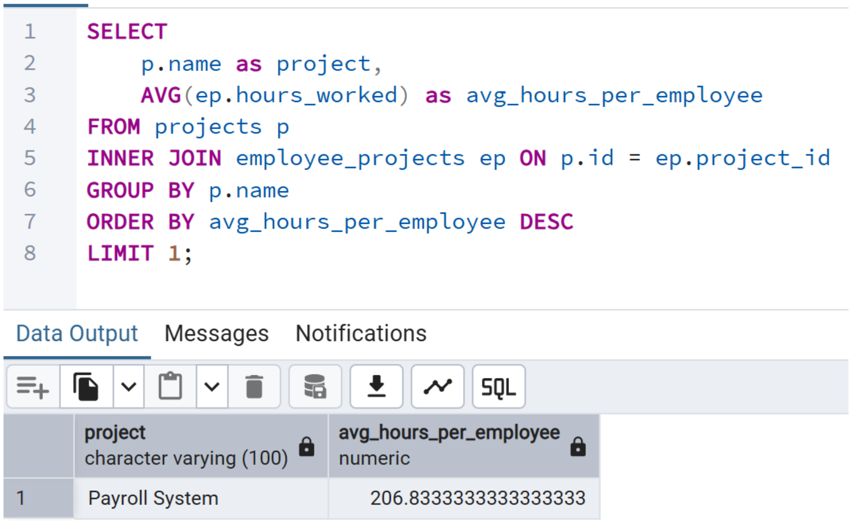
1.Tampilkan 5 department dengan total gaji tertinggi



```
SELECT
    d.name as department,
    SUM(e.salary) as total_salary
FROM departments d
INNER JOIN employees e ON d.id = e.department_id
GROUP BY d.name
ORDER BY total_salary DESC
LIMIT 5;
```

	department character varying (100)	total_salary numeric
1	Finance	27000000.00
2	IT	25500000.00
3	HR	17500000.00
4	Marketing	16000000.00
5	Operations	6500000.00

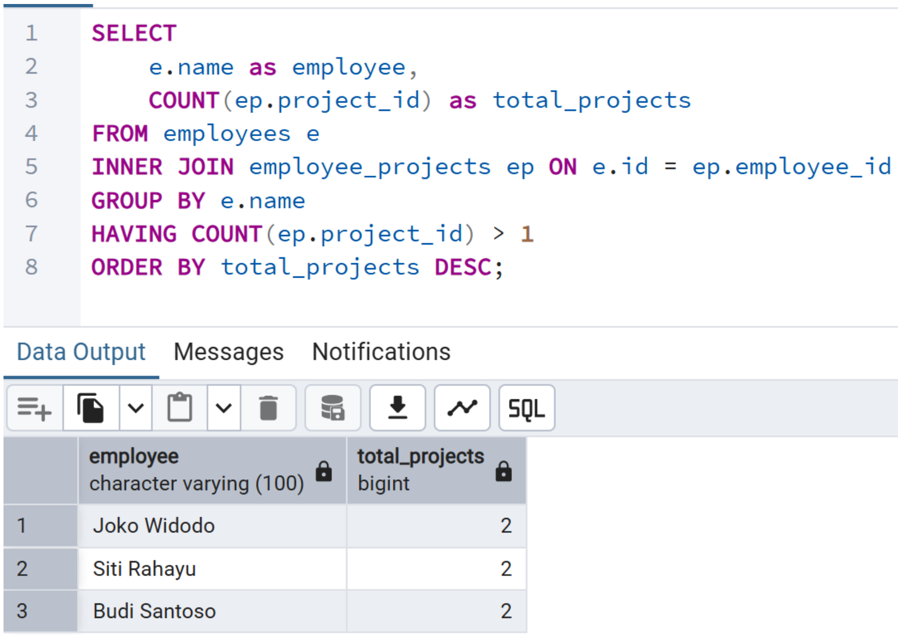
2.Tampilkan project dengan rata-rata jam kerja per employee tertinggi 16



```
SELECT
    p.name as project,
    AVG(ep.hours_worked) as avg_hours_per_employee
FROM projects p
INNER JOIN employee_projects ep ON p.id = ep.project_id
GROUP BY p.name
ORDER BY avg_hours_per_employee DESC
LIMIT 1;
```

	project character varying (100)	avg_hours_per_employee numeric
1	Payroll System	206.8333333333333333

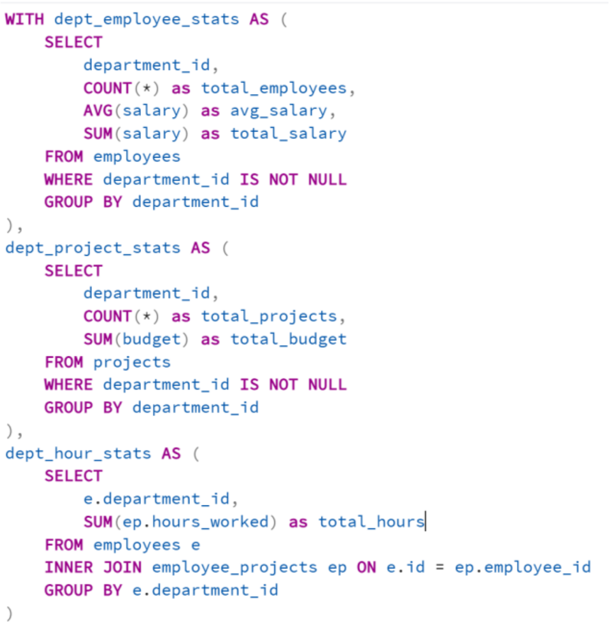
3.Tampilkan employee yang bekerja di lebih dari 1 project



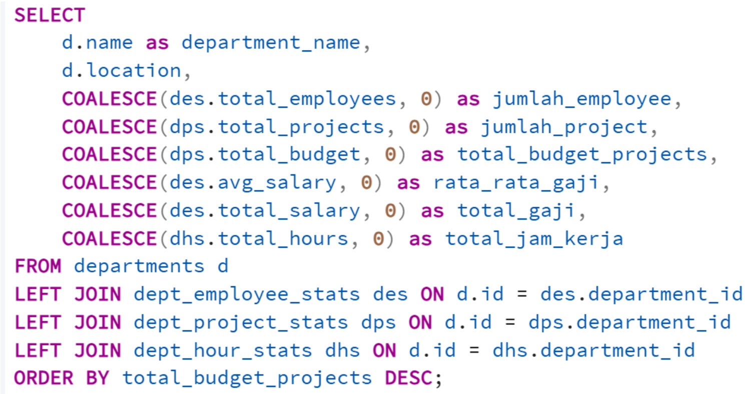
```sql
SELECT
    e.name as employee,
    COUNT(ep.project_id) as total_projects
FROM employees e
INNER JOIN employee_projects ep ON e.id = ep.employee_id
GROUP BY e.name
HAVING COUNT(ep.project_id) > 1
ORDER BY total_projects DESC;
```

	employee character varying (100)	total_projects bigint
1	Joko Widodo	2
2	Siti Rahayu	2
3	Budi Santoso	2

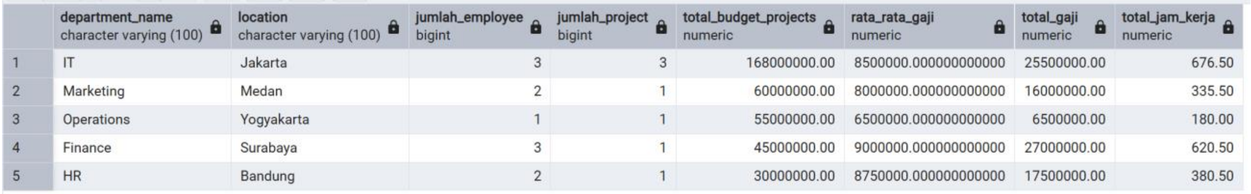
4.Buat laporan performance department: nama department, jumlah employee, jumlah project, total budget projects, rata-rata gaji, total jam kerja



```sql
WITH dept_employee_stats AS (
    SELECT
        department_id,
        COUNT(*) as total_employees,
        AVG(salary) as avg_salary,
        SUM(salary) as total_salary
    FROM employees
    WHERE department_id IS NOT NULL
    GROUP BY department_id
),
dept_project_stats AS (
    SELECT
        department_id,
        COUNT(*) as total_projects,
        SUM(budget) as total_budget
    FROM projects
    WHERE department_id IS NOT NULL
    GROUP BY department_id
),
dept_hour_stats AS (
    SELECT
        e.department_id,
        SUM(ep.hours_worked) as total_hours
    FROM employees e
    INNER JOIN employee_projects ep ON e.id = ep.employee_id
    GROUP BY e.department_id
)
```



```sql
SELECT
    d.name as department_name,
    d.location,
    COALESCE(des.total_employees, 0) as jumlah_employee,
    COALESCE(dps.total_projects, 0) as jumlah_project,
    COALESCE(dps.total_budget, 0) as total_budget_projects,
    COALESCE(des.avg_salary, 0) as rata_rata_gaji,
    COALESCE(des.total_salary, 0) as total_gaji,
    COALESCE(dhs.total_hours, 0) as total_jam_kerja
FROM departments d
LEFT JOIN dept_employee_stats des ON d.id = des.department_id
LEFT JOIN dept_project_stats dps ON d.id = dps.department_id
LEFT JOIN dept_hour_stats dhs ON d.id = dhs.department_id
ORDER BY total_budget_projects DESC;
```



	department_name character varying (100)	location character varying (100)	jumlah_employee bigint	jumlah_project bigint	total_budget_projects numeric	rata_rata_gaji numeric	total_gaji numeric	total_jam_kerja numeric
1	IT	Jakarta	3	3	168000000.00	8500000.000000000000	25500000.00	676.50
2	Marketing	Medan	2	1	60000000.00	8000000.000000000000	16000000.00	335.50
3	Operations	Yogyakarta	1	1	55000000.00	6500000.000000000000	6500000.00	180.00
4	Finance	Surabaya	3	1	45000000.00	9000000.000000000000	27000000.00	620.50
5	HR	Bandung	2	1	30000000.00	8750000.000000000000	17500000.00	380.50