Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота № 2

з дисципліни «Професійне використання SQL та PL/SQL для СУБД Oracle»

Тема: «Робота в тестовій базі даних. Відпрацювання вибірки по одній та по декільком таблицям із застосуванням умов WHERE, сортування ORDER BY, та сполучень JOIN.»

Виконали:
студент групи
ІМ-13 Первєєв Євгеній Олексійович

Перевірила:
Ульяницька К.О.

Київ 2024

Тема: «Робота в тестовій базі даних. Відпрацювання вибірки по одній та по декільком таблицям із застосуванням умов WHERE, сортування ORDER BY, та сполучень JOIN.»

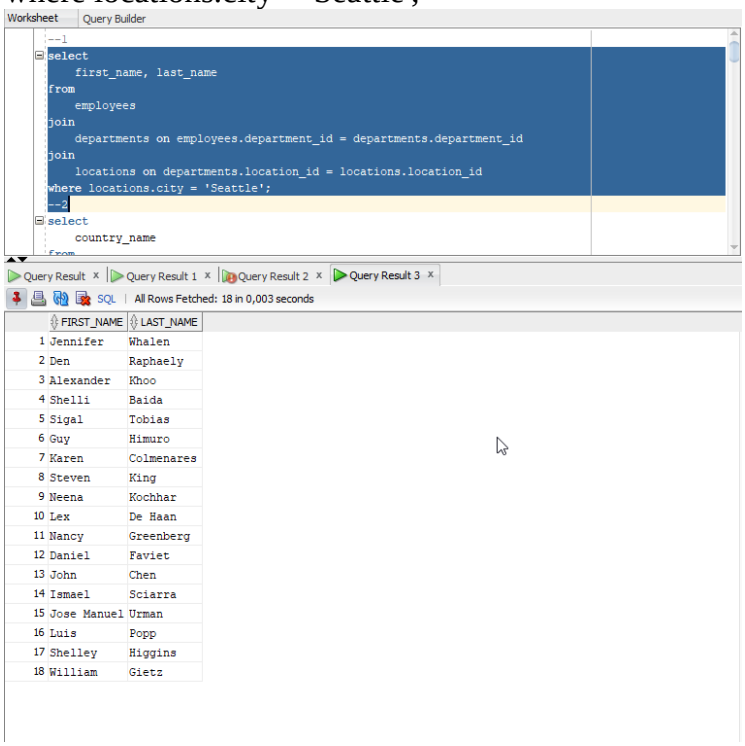
Мета: Навчитись створювати запити по конструкції SELECT із застосуванням вивчених механізмів.

Хід роботи

1. Написати запити для відпрацювання завдань по схемі HR (тестова база даних з лабораторної роботи №1).

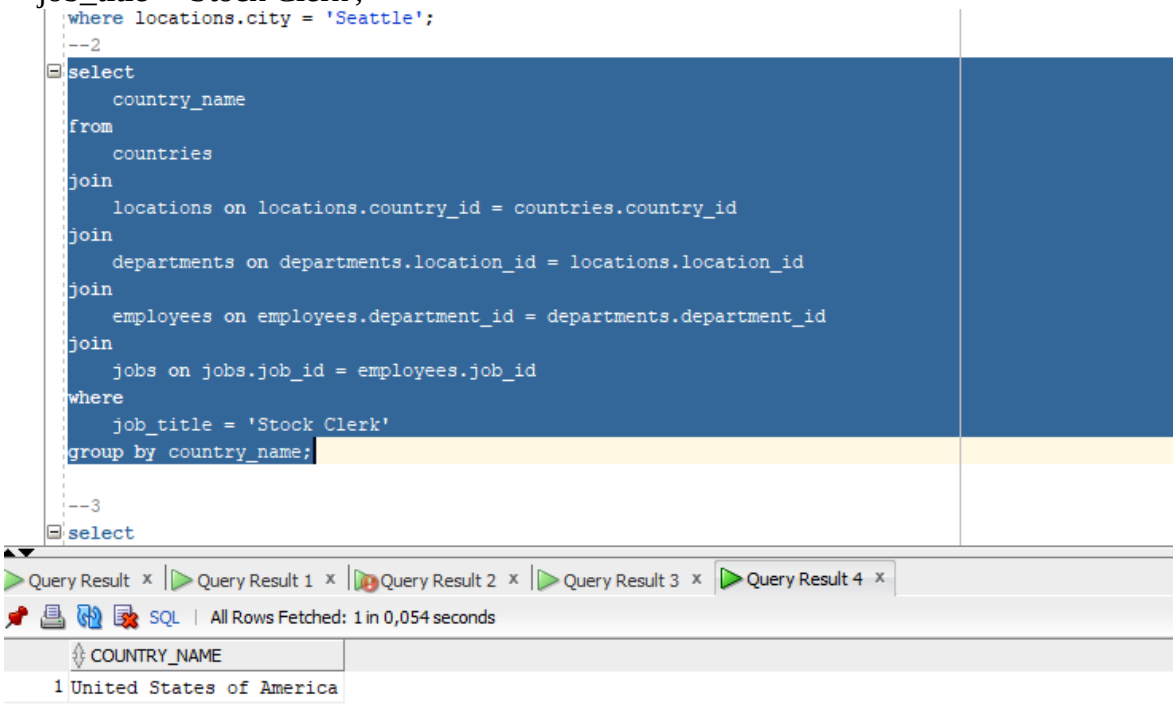
1. Вибрати усіх робітників, локація департаменту яких знаходиться у Seattle.

```
--1
select
    first_name, last_name
from
    employees
join
    departments on employees.department_id = departments.department_id
join
    locations on departments.location_id = locations.location_id
where locations.city = 'Seattle';
```



2. Вибрати всі країни, робітники з яких працюють на посаді «Stock Clerk».

```
--2
select
    distinct country_name
from
    countries
join
    locations on locations.country_id = countries.country_id
join
    departments on departments.location_id = locations.location_id
join
    employees on employees.department_id = departments.department_id
join
    jobs on jobs.job_id = employees.job_id
where
    job_title = 'Stock Clerk';
```



3. Вибрати всіх робітників з регіону Europe, в яких ЗП не більше 9000 і менше або дорівнює 17000.

```
--3
select
    employees.first_name, employees.last_name, employees.salary
from
    employees
join
    departments on employees.department_id = departments.department_id
join
    locations on departments.location_id = locations.location_id
join
    countries on locations.country_id = countries.country_id
join
    regions on countries.region_id = regions.region_id
where
    regions.region_name = 'Europe' and employees.salary between 9000 and 170001;
```

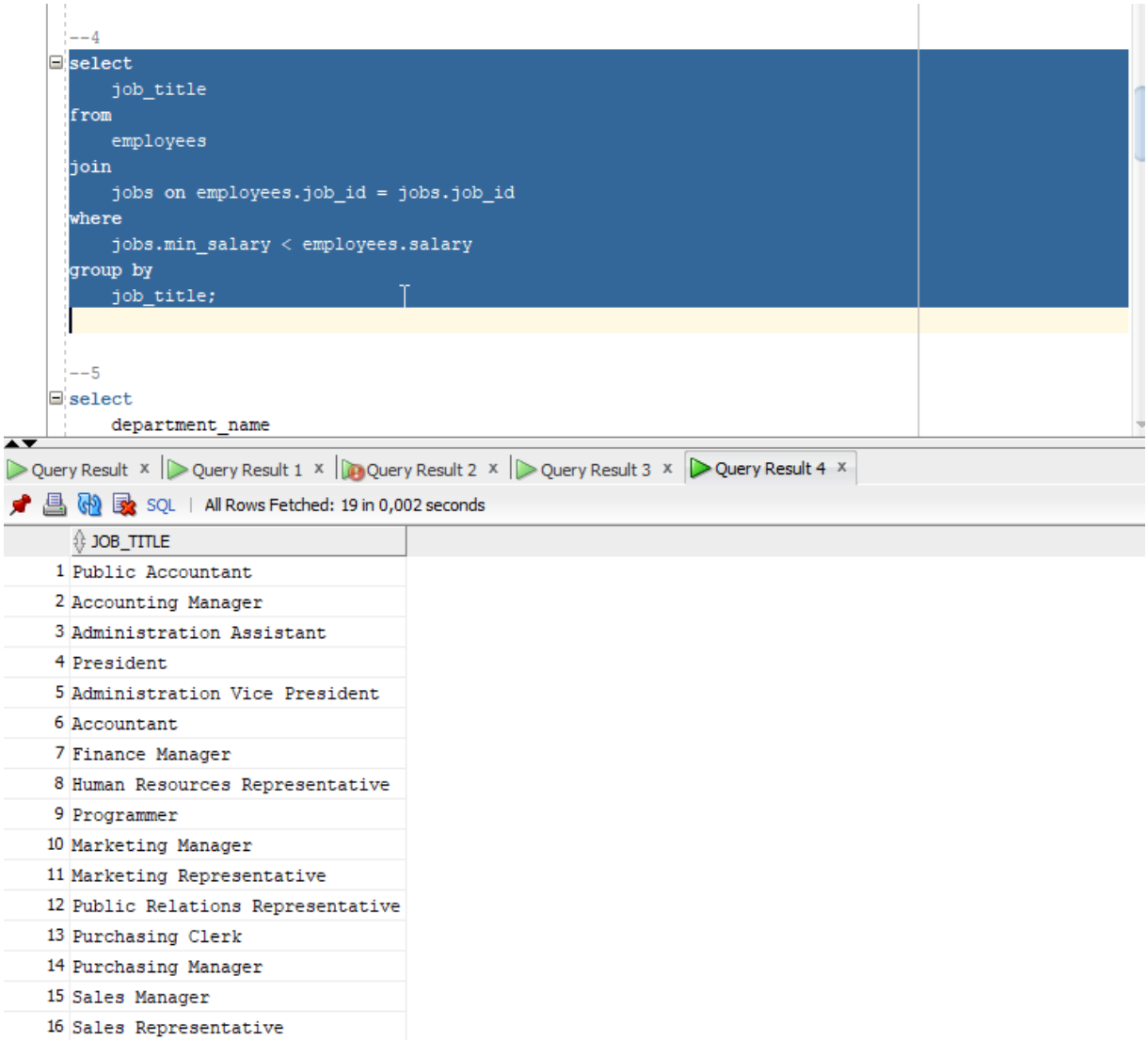
Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4

SQL | All Rows Fetched: 18 in 0,008 seconds

	FIRST_NAME	LAST_NAME	SALARY
1	John	Russell	14000
2	Karen	Partners	13500
3	Alberto	Errazuriz	12000
4	Gerald	Cambrault	11000
5	Eleni	Zlotkey	10500
6	Peter	Tucker	10000
7	David	Bernstein	9500
8	Peter	Hall	9000
9	Janette	King	10000
10	Patrick	Sully	9500
11	Allan	McEwen	9000
12	Clara	Vishney	10500
13	Danielle	Greene	9500
14	Lisa	Ozer	11500
15	Harrison	Bloom	10000
16	Tayler	Fox	9600

4. Вивести всі роботи, робітники на яких отримують заробітну плату вище мінімальної згідно атрибуту MIN_SALARY таблиці JOBS.

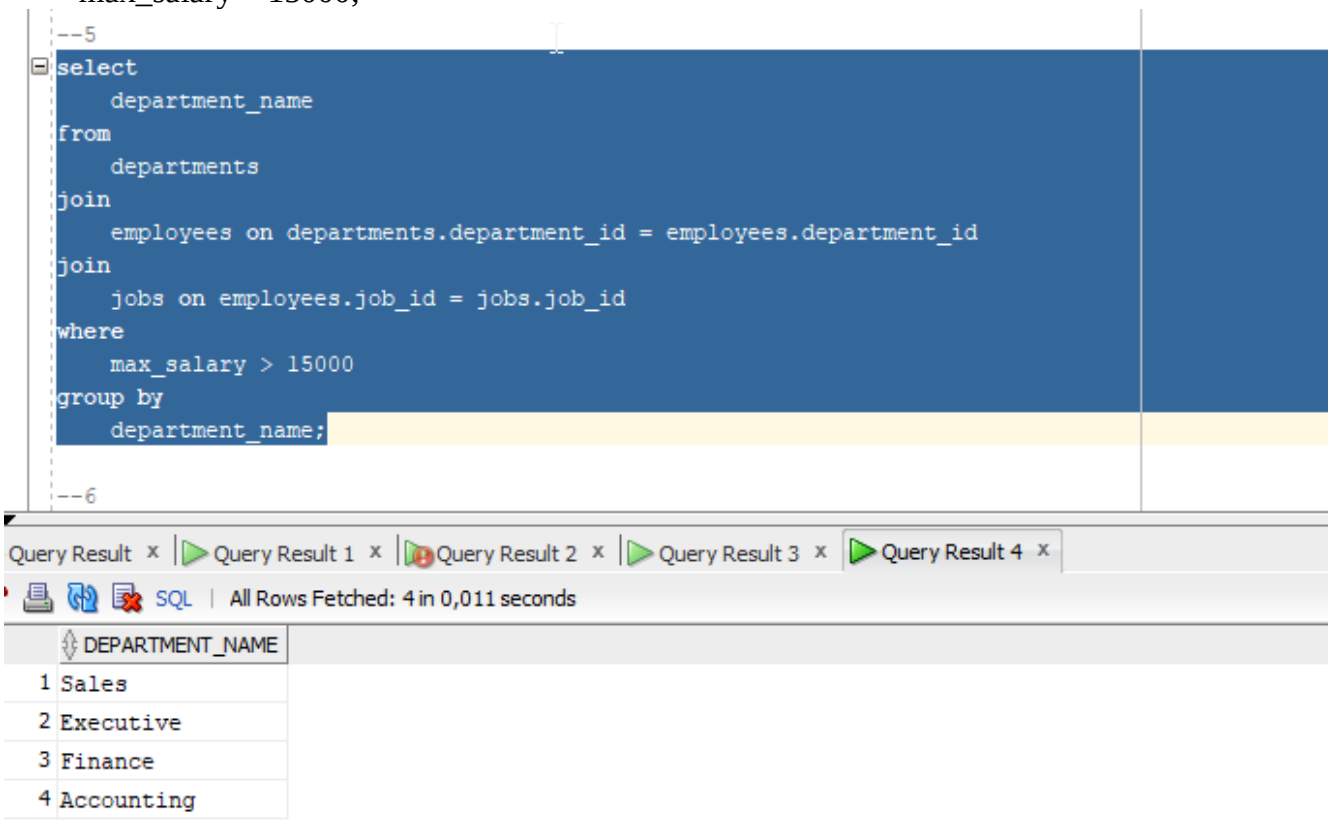
```
--4
select
    distinct job_title
from
    employees
join
    jobs on employees.job_id = jobs.job_id
where
    jobs.min_salary < employees.salary;
```



5. Вивести департаменти, робітники яких працюють на посадах з Максимал--ьною заробітною платою вище 15000.

```
--5
select
```

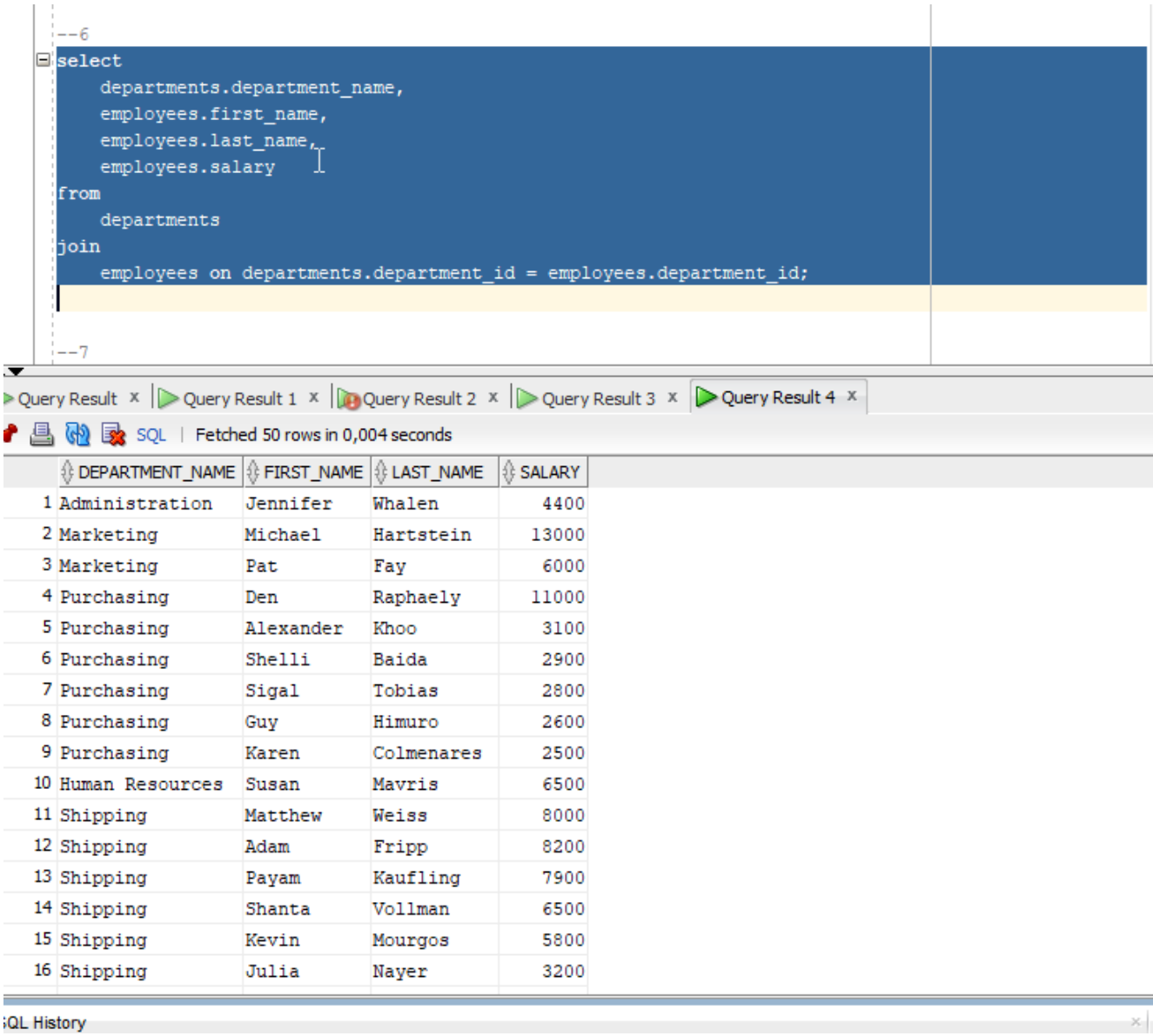
```
    distinct department_name
from
    departments
join
    employees on departments.department_id = employees.department_id
join
    jobs on employees.job_id = jobs.job_id
where
    max_salary > 15000;
```



6. Вивести всі департаменти та працівників, що в них працюють з їхньою заробітною платою.

```
--6
select
    departments.department_name,
    employees.first_name,
```

```
    employees.last_name,
    employees.salary
from
    departments
join
    employees on departments.department_id = employees.department_id;
```



	DEPARTMENT_NAME	FIRST_NAME	LAST_NAME	SALARY
1	Administration	Jennifer	Whalen	4400
2	Marketing	Michael	Hartstein	13000
3	Marketing	Pat	Fay	6000
4	Purchasing	Den	Raphaely	11000
5	Purchasing	Alexander	Khoo	3100
6	Purchasing	Shelli	Baida	2900
7	Purchasing	Sigal	Tobias	2800
8	Purchasing	Guy	Himuro	2600
9	Purchasing	Karen	Colmenares	2500
10	Human Resources	Susan	Mavris	6500
11	Shipping	Matthew	Weiss	8000
12	Shipping	Adam	Fripp	8200
13	Shipping	Payam	Kaufling	7900
14	Shipping	Shanta	Vollman	6500
15	Shipping	Kevin	Mourgos	5800
16	Shipping	Julia	Nayer	3200

7. Вивести всіх працівників, їх посади та країни (COUNTRIES), де вони працюють.

```
--7
select
    employees.first_name,
    employees.last_name,
    jobs.job_title,
```

```
    countries.country_name
from
    departments
join
    employees on departments.department_id = employees.department_id
join
    jobs on employees.job_id = jobs.job_id
join
    locations on  departments.location_id = locations.location_id
join;
```

```
--7
select
    employees.first_name,
    employees.last_name,
    jobs.job_title,
    countries.country_name
from
    departments
join
    employees on departments.department_id = employees.department_id
join
    jobs on employees.job_id = jobs.job_id
join
    locations on  departments.location_id = locations.location_id
join
    countries on locations.country_id = countries.country_id;
```

Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4

SQL | Fetched 50 rows in 0,008 seconds

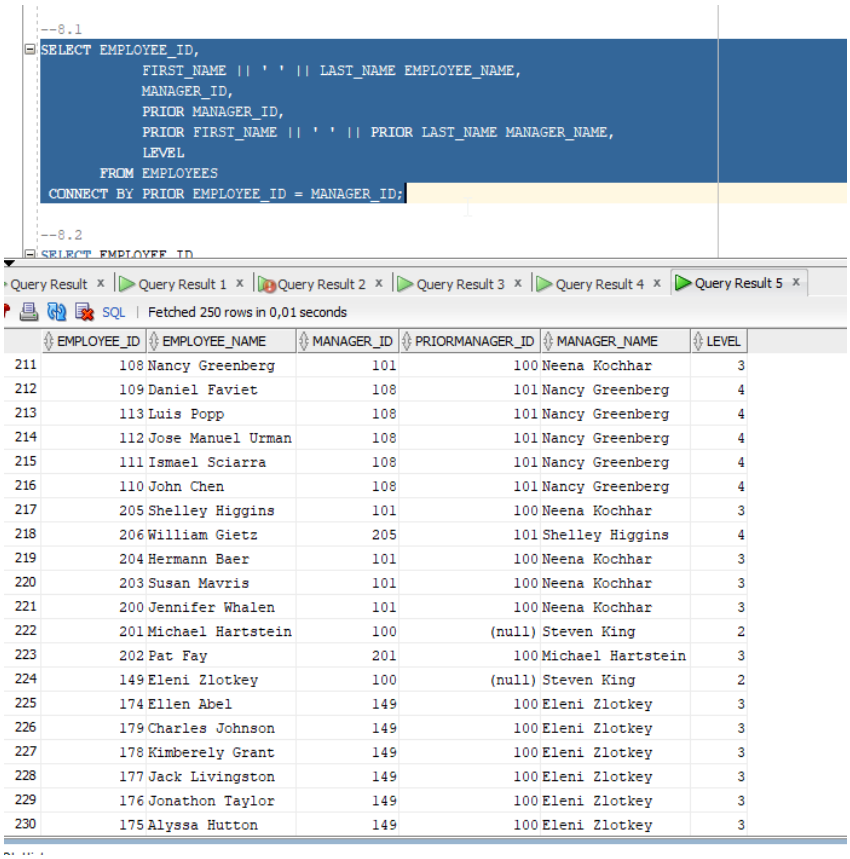
	FIRST_NAME	LAST_NAME	JOB_TITLE	COUNTRY_NAME
1	Alexander	Hunold	Programmer	United States of America
2	Bruce	Ernst	Programmer	United States of America
3	Diana	Lorentz	Programmer	United States of America
4	Valli	Pataballa	Programmer	United States of America
5	David	Austin	Programmer	United States of America
6	Nandita	Sarchand	Shipping Clerk	United States of America
7	Alexis	Bull	Shipping Clerk	United States of America
8	Julia	Dellinger	Shipping Clerk	United States of America
9	Anthony	Cabrio	Shipping Clerk	United States of America
10	Kelly	Chung	Shipping Clerk	United States of America
11	Jennifer	Dilly	Shipping Clerk	United States of America
12	Timothy	Gates	Shipping Clerk	United States of America
13	Randall	Perkins	Shipping Clerk	United States of America
14	Sarah	Bell	Shipping Clerk	United States of America
15	Britney	Everett	Shipping Clerk	United States of America
16	Samuel	McCain	Shipping Clerk	United States of America
17	Vance	Jones	Shipping Clerk	United States of America
18	Alana	Walsh	Shipping Clerk	United States of America

8. Вивести ієрархію робітників

8.1 Ієрархія до 3

```
 --8.1
SELECT EMPLOYEE_ID,
        FIRST_NAME || ' ' || LAST_NAME EMPLOYEE_NAME,
        MANAGER_ID,
```

```
        PRIOR MANAGER_ID,
        PRIOR FIRST_NAME || ' ' || PRIOR LAST_NAME MANAGER_NAME,
        LEVEL
    FROM EMPLOYEES
CONNECT BY PRIOR EMPLOYEE_ID = MANAGER_ID;
```



```
--8.1
SELECT EMPLOYEE_ID,
        FIRST_NAME || ' ' || LAST_NAME EMPLOYEE_NAME,
        MANAGER_ID,
        PRIOR MANAGER_ID,
        PRIOR FIRST_NAME || ' ' || PRIOR LAST_NAME MANAGER_NAME,
        LEVEL
    FROM EMPLOYEES
CONNECT BY PRIOR EMPLOYEE_ID = MANAGER_ID;

--8.2
SELECT EMPLOYEE_ID,
```

Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4 | Query Result 5

SQL | Fetched 250 rows in 0,01 seconds

	EMPLOYEE_ID	EMPLOYEE_NAME	MANAGER_ID	PRIORMANAGER_ID	MANAGER_NAME	LEVEL
211	108	Nancy Greenberg	101	100	Neena Kochhar	3
212	109	Daniel Faviet	108	101	Nancy Greenberg	4
213	113	Luis Popp	108	101	Nancy Greenberg	4
214	112	Jose Manuel Urman	108	101	Nancy Greenberg	4
215	111	Ismael Sciarra	108	101	Nancy Greenberg	4
216	110	John Chen	108	101	Nancy Greenberg	4
217	205	Shelley Higgins	101	100	Neena Kochhar	3
218	206	William Gietz	205	101	Shelley Higgins	4
219	204	Hermann Baer	101	100	Neena Kochhar	3
220	203	Susan Mavris	101	100	Neena Kochhar	3
221	200	Jennifer Whalen	101	100	Neena Kochhar	3
222	201	Michael Hartstein	100	(null)	Steven King	2
223	202	Pat Fay	201	100	Michael Hartstein	3
224	149	Eleni Zlotkey	100	(null)	Steven King	2
225	174	Ellen Abel	149	100	Eleni Zlotkey	3
226	179	Charles Johnson	149	100	Eleni Zlotkey	3
227	178	Kimberely Grant	149	100	Eleni Zlotkey	3
228	177	Jack Livingston	149	100	Eleni Zlotkey	3
229	176	Jonathon Taylor	149	100	Eleni Zlotkey	3
230	175	Alyssa Hutton	149	100	Eleni Zlotkey	3

8.2 Повна ієрархія

```
SELECT EMPLOYEE_ID,
        FIRST_NAME || ' ' || LAST_NAME EMPLOYEE_NAME,
        MANAGER_ID,
        PRIOR MANAGER_ID,
        PRIOR FIRST_NAME || ' ' || PRIOR LAST_NAME MANAGER_NAME,
        LEVEL
```

FROM EMPLOYEES
WHERE LEVEL < 4
CONNECT BY PRIOR EMPLOYEE_ID = MANAGER_ID;

```
--8.2
SELECT EMPLOYEE_ID,
            FIRST_NAME || ' ' || LAST_NAME EMPLOYEE_NAME,
            MANAGER_ID,
            PRIOR MANAGER_ID,
            PRIOR FIRST_NAME || ' ' || PRIOR LAST_NAME MANAGER_NAME,
            LEVEL
       FROM EMPLOYEES
        WHERE LEVEL < 4
 CONNECT BY PRIOR EMPLOYEE_ID = MANAGER_ID;
```

Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4 | Query Result 5 | Query...

SQL | Fetched 50 rows in 0,003 seconds

	EMPLOYEE_ID	EMPLOYEE_NAME	MANAGER_ID	PRIORMANAGER_ID	MANAGER_NAME	LEVEL
1	101	Neena Kochhar	100	(null)		1
2	108	Nancy Greenberg	101	100	Neena Kochhar	2
3	109	Daniel Faviet	108	101	Nancy Greenberg	3
4	113	Luis Popp	108	101	Nancy Greenberg	3
5	112	Jose Manuel Urman	108	101	Nancy Greenberg	3
6	111	Ismael Sciarra	108	101	Nancy Greenberg	3
7	110	John Chen	108	101	Nancy Greenberg	3
8	205	Shelley Higgins	101	100	Neena Kochhar	2
9	206	William Gietz	205	101	Shelley Higgins	3
10	204	Hermann Baer	101	100	Neena Kochhar	2
11	203	Susan Mavris	101	100	Neena Kochhar	2
12	200	Jennifer Whalen	101	100	Neena Kochhar	2
13	201	Michael Hartstein	100	(null)		1
14	202	Pat Fay	201	100	Michael Hartstein	2
15	149	Eleni Zlotkey	100	(null)		1
16	174	Ellen Abel	149	100	Eleni Zlotkey	2
17	179	Charles Johnson	149	100	Eleni Zlotkey	2
18	178	Kimberely Grant	149	100	Eleni Zlotkey	2
19	177	Jack Livingston	149	100	Eleni Zlotkey	2
20	176	Jonathon Taylor	149	100	Eleni Zlotkey	2

2. Написати приклади завдань та відпрацьованих по ним запитах по всім видам JOIN (INNER, OUTER, CROSS) та застосуванню ORDER BY по предметній області з лабораторної роботи №2.

INSERT INTO Unfinished_Medicines (Name, A_Method_ID, P_Method_ID, Recipe_ID, P_Time) VALUES ('Мазь', null, null, null, null);

INSERT INTO Unfinished_Medicines (Name, A_Method_ID, P_Method_ID, Recipe_ID,

P_Time) VALUES ('Мазь2', null, null, null, null);

INSERT INTO Unfinished_Medicines (Name, A_Method_ID, P_Method_ID, Recipe_ID, P_Time) VALUES ('Мазь2', null, null, null, null);

INSERT INTO Finished_Medicines (Name,a_method_id) VALUES (null, 1);
INSERT INTO Finished_Medicines (Name,a_method_id) VALUES ('null', 2);
INSERT INTO Finished_Medicines (Name,a_method_id) VALUES ('null', 3);

INSERT INTO application_methods (Name) VALUES ('am1');
INSERT INTO application_methods (Name) VALUES ('am2');
INSERT INTO application_methods (Name) VALUES ('am3');
INSERT INTO application_methods (Name) VALUES ('am4');

--full join
select * from unfinished_medicines U_M
full join preparation_methods P_M on P_M.id =u_m.p_method_id;

	ID	NAME	A_METHOD_ID	P_METHOD_ID	RECIPE_ID	P_TIME	ID_1	NAME_1
1	21	Мазь	(null)	(null)	(null)	(null)	(null	(null)
2	22	Мазь2	(null)	(null)	(null)	(null)	(null)	(null)
3	1	Мазь за рец1	2	1	1	15	1	Змішування
4	2	Розчин за рец2	3	2	2	10	2	Зміш-відст-фільтр
5	3	Мікстура за рец1	1	2	3	30	2	Зміш-відст-фільтр

--left join
select
    *
from
    Finished_Medicines FM
left join
    application_methods AP
on
    FM.a_method_id = AP.id;

	ID	NAME	A_METHOD_ID	ID_1	NAME_1
1	23	Sustanon	2	2	Внутрішнє заст.
2	26	null	2	2	Внутрішнє заст.
3	2	Мазь Вишневського	2	2	Внутрішнє заст.
4	4	Вазелін	2	2	Внутрішнє заст.
5	27	null	3	3	Для змішування
6	21	Trenbolon	1	1	Зовнішнє заст.
7	25	(null)	1	1	Зовнішнє заст.
8	1	Ношпа	1	1	Зовнішнє заст.
9	3	Аспірин	1	1	Зовнішнє заст.
10	5	Аспаркам	1	1	Зовнішнє заст.
11	6	Детралекс	1	1	Зовнішнє заст.

```
--right join
select
    *
from
    Finished_Medicines FM
right join
    application_methods AP
on
FM.a_method_id = AP.id;
```

	ID	NAME	A_METHOD_ID	ID_1	NAME_1
1	21	Trenbolon	1	1	Зовнішнє заст.
2	23	Sustanon	2	2	Внутрішнє заст.
3	25	(null)	1	1	Зовнішнє заст.
4	26	null	2	2	Внутрішнє заст.
5	27	null	3	3	Для змішування
6	1	Ношпа	1	1	Зовнішнє заст.
7	2	Мазь Вишневського	2	2	Внутрішнє заст.
8	3	Аспірин	1	1	Зовнішнє заст.
9	4	Вазелін	2	2	Внутрішнє заст.
10	5	Аспаркам	1	1	Зовнішнє заст.
11	6	Детралекс	1	1	Зовнішнє заст.
12	(null)	(null)	(null)	23	am3
13	(null)	(null)	(null)	21	am1
14	(null)	(null)	(null)	22	am2
15	(null)	(null)	(null)	24	am4

Приклади з використанням left and right joins в HR

```
--Виводить працівників, їх департамент та адресу роботи
select em.first_name, em.last_name, dep.department_name, loc.street_address from employees em
join departments dep on em.department_id = dep.department_id
join locations loc on dep.location_id = loc.location_id;
```

Worksheet | Query Builder

```
--Виводить працівників, їх департамент та адресу роботи
select em.first_name, em.last_name, dep.department_name, loc.street_address from employees em
join departments dep on em.department_id = dep.department_id
join locations loc on dep.location_id = loc.location_id;


--Виводить працівника без департаменту
select em.first_name, em.last_name, dep.department_name from employees em
left join departments dep on em.department_id = dep.department_id
where dep.department_id is NULL;
```

Query Result | Query Result 1 | Query Result 2

SQL | Fetched 50 rows in 0,006 seconds

	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME	STREET_ADDRESS
1	Ellen	Abel	Sales	Magdalen Centre, The Oxford Science Park
2	Sundar	Ande	Sales	Magdalen Centre, The Oxford Science Park
3	Mozhe	Atkinson	Shipping	2011 Interiors Blvd
4	David	Austin	IT	2014 Jabberwocky Rd
5	Hermann	Baer	Public Relations	Schwanthalerstr. 7031
6	Shelli	Baida	Purchasing	2004 Charade Rd
7	Amit	Banda	Sales	Magdalen Centre, The Oxford Science Park
8	Elizabeth	Bates	Sales	Magdalen Centre, The Oxford Science Park
9	Sarah	Bell	Shipping	2011 Interiors Blvd
10	David	Bernstein	Sales	Magdalen Centre, The Oxford Science Park
11	Laura	Bissot	Shipping	2011 Interiors Blvd
12	Harrison	Bloom	Sales	Magdalen Centre, The Oxford Science Park
13	Alexis	Bull	Shipping	2011 Interiors Blvd

--Виводить працівника без департаменту
select em.first_name, em.last_name, dep.department_name from employees em
left join departments dep on em.department_id = dep.department_id
where dep.department_id is NULL;

```
--Виводить працівника без департаменту
select em.first_name, em.last_name, dep.department_name from employees em
left join departments dep on em.department_id = dep.department_id
where dep.department_id is NULL;


--Виводить ті департаменти в яких немає працівників
select em.first_name, em.last_name, dep.department_name, dep.department_id from employees em
right join departments dep on em.department_id = dep.department_id
```

Query Result | Query Result 1

SQL | All Rows Fetched: 1 in 0,005 seconds

	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
1	Kimberely	Grant	(null)

--Виводить ті департаменти в яких немає працівників
select em.first_name, em.last_name, dep.department_name, dep.department_id from employees em
right join departments dep on em.department_id = dep.department_id
where em.department_id is NULL;

```
--Виводить ті департаменти в яких немає працівників
select em.first_name, em.last_name, dep.department_name, dep.department_id from employees em
right join departments dep on em.department_id = dep.department_id
where em.department_id is NULL;
```

Query Result

All Rows Fetched: 16 in 0,007 seconds

	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME	DEPARTMENT_ID
1	(null)	(null)	Treasury	120
2	(null)	(null)	Corporate Tax	130
3	(null)	(null)	Control And Credit	140
4	(null)	(null)	Shareholder Services	150
5	(null)	(null)	Benefits	160
6	(null)	(null)	Manufacturing	170
7	(null)	(null)	Construction	180
8	(null)	(null)	Contracting	190
9	(null)	(null)	Operations	200
10	(null)	(null)	IT Support	210
11	(null)	(null)	NOC	220
12	(null)	(null)	IT Helpdesk	230
13	(null)	(null)	Government Sales	240
14	(null)	(null)	Retail Sales	250
15	(null)	(null)	Recruiting	260
16	(null)	(null)	Payroll	270

ВИСНОВОК

У цій лабораторній роботі ми вивчали створення складних запитів за допомогою конструкції SELECT в мові SQL(WHERE, ORDER BY,JOIN). Використовуючи ці механізми, ми були в змозі створювати складні запити, які відображали результати з декількох таблиць, фільтрували їх за певними умовами, сортували та об'єднували дані в зручному для аналізу вигляді. Вміння працювати з цими механізмами є важливим для розробки ефективних та потужних запитів у базах даних.