

## Лабораторна робота 1

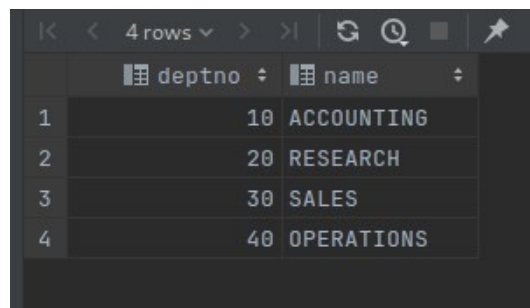
Завдання лабораторної виконую на 89%

### Завдання 1

У сценарії init.sql наведено сценарій розміщення інформації з Emp, Dept, salgrade у метамоделі EAV/CR. Виконайте сценарії і напишіть запити, які дадуть відповіді на наступні запитання на основі створених таблиць.

1. Виведіть список підрозділів фірми (Раніше ці дані зберігалися у таблиці DEPT)

```
select object_id as deptno, name from objects where object_type_id = 3;
```

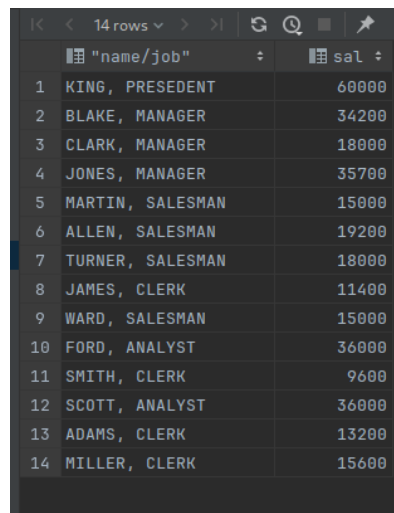


	deptno	name
1	10	ACCOUNTING
2	20	RESEARCH
3	30	SALES
4	40	OPERATIONS

2. Сформувавати запит, що виводить 2 стовпці інформації:

- прізвище працівника та, через кому, його посаду;
- його зарплату за рік.

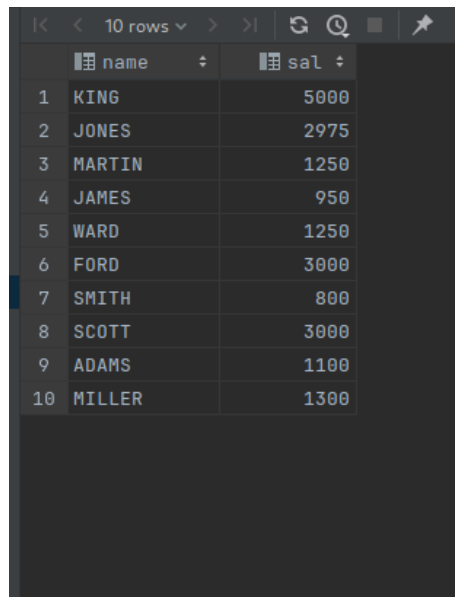
```
select
  (objects.name || ', ' || p1.text_value) as "name/job",
  p2.numeric_value * 12 as sal
from objects
join params p1 on (objects.object_id = p1.object_id and p1.attr_id = 1)
join params p2 on (objects.object_id = p2.object_id and p2.attr_id = 3)
where object_type_id = 1
;
```



	"name/job"	sal
1	KING, PRESEDENT	60000
2	BLAKE, MANAGER	34200
3	CLARK, MANAGER	18000
4	JONES, MANAGER	35700
5	MARTIN, SALESMAN	15000
6	ALLEN, SALESMAN	19200
7	TURNER, SALESMAN	18000
8	JAMES, CLERK	11400
9	WARD, SALESMAN	15000
10	FORD, ANALYST	36000
11	SMITH, CLERK	9600
12	SCOTT, ANALYST	36000
13	ADAMS, CLERK	13200
14	MILLER, CLERK	15600

3. Сформувати запит, що виводить прізвище та зарплату працівників, зарплата яких лежить поза діапазоном від \$1500 до \$2850.

```
select
    objects.name,
    p1.numeric_value as sal
from objects
join params p1 on (objects.object_id = p1.object_id and p1.attr_id = 3)
where
    object_type_id = 1
    and p1.numeric_value not between 1500 and 2850
;
```

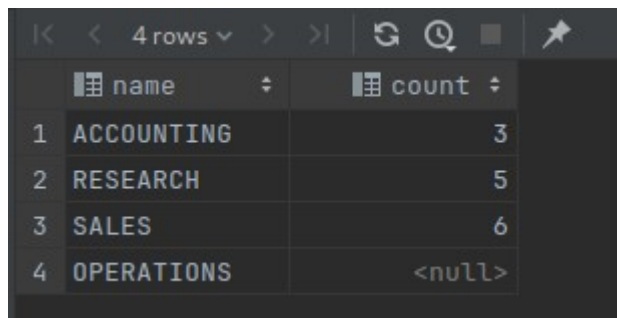


The screenshot shows a database interface with a table of 10 rows. The table has two columns: 'name' and 'sal'. The rows are numbered 1 through 10. The names are KING, JONES, MARTIN, JAMES, WARD, FORD, SMITH, SCOTT, ADAMS, and MILLER. The salaries are 5000, 2975, 1250, 950, 1250, 3000, 800, 3000, 1100, and 1300 respectively. The interface includes a toolbar at the top with icons for navigation and search.

	name	sal
1	KING	5000
2	JONES	2975
3	MARTIN	1250
4	JAMES	950
5	WARD	1250
6	FORD	3000
7	SMITH	800
8	SCOTT	3000
9	ADAMS	1100
10	MILLER	1300

4. Підрахуйте кількість працівників у кожному підрозділі, включаючи підрозділи без співробітників. (2 колонки – назва підрозділу та кількість людей)

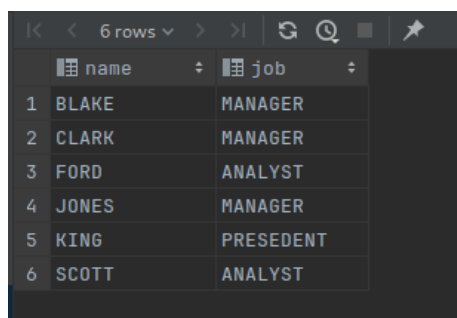
```
select
    objects.name,
    p1.c as count
from objects
left join (
    select
        numeric_value,
        count(*) as c
    from params
    where attr_id = 5
    group by numeric_value
) p1 on (p1.numeric_value = objects.object_id)
where
    object_type_id = 3
;
```



	name	count
1	ACCOUNTING	3
2	RESEARCH	5
3	SALES	6
4	OPERATIONS	<null>

5. Виведіть імена та посади керівників.

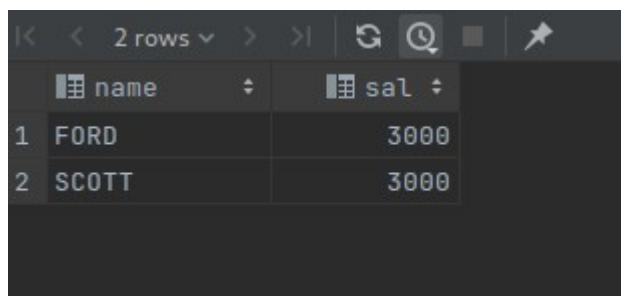
```
select distinct
    o2.name,
    p1.text_value as job
from objects o1
join objects o2 on (o1.parent_id = o2.object_id)
join params p1 on (p1.object_id = o2.object_id and attr_id = 1)
where
    o2.object_type_id = 1
;
```



	name	job
1	BLAKE	MANAGER
2	CLARK	MANAGER
3	FORD	ANALYST
4	JONES	MANAGER
5	KING	PRESEDENT
6	SCOTT	ANALYST

6. Виведіть співробітників, що мають більшу зарплату більше ніж їх керівники.

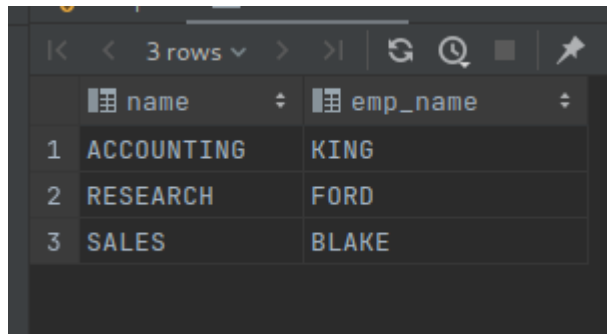
```
select
  o1.name,
  p1.numeric_value as sal
from objects o1
join objects o2 on (o1.parent_id = o2.object_id)
join params p1 on (p1.object_id = o1.object_id and p1.attr_id = 3)
join params p2 on (p2.object_id = o2.object_id and p2.attr_id = 3)
where
  o2.object_type_id = 1
  and p2.numeric_value < p1.numeric_value
;
```



	name	sal
1	FORD	3000
2	SCOTT	3000

7. Виведіть назву відділу та співробітника, що має найбільшу зарплату у своєму відділі.

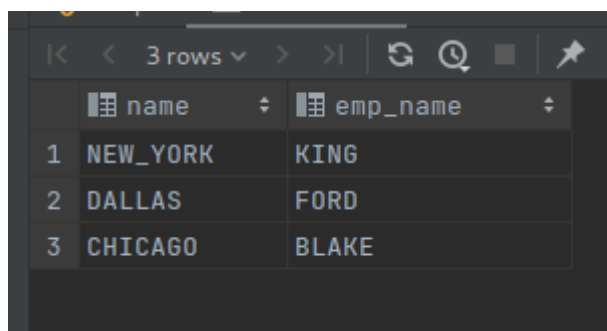
```
select
  o1.name,
  o2.name as emp_name
from objects o1
join (
  select distinct on (dept) empno, dept, sal from (
    select
      pp1.object_id as empno,
      pp2.numeric_value as dept,
      pp1.numeric_value as sal
    from params pp1
    join params pp2 on (
      pp1.object_id = pp2.object_id
      and pp1.attr_id = 3 and pp2.attr_id = 5
    )
  ) pr1
  order by dept desc, sal desc
) pr2 on (o1.object_id = pr2.dept)
join objects o2 on (o2.object_id = pr2.empno)
where
  o1.object_type_id = 3;
```



	name	emp_name
1	ACCOUNTING	KING
2	RESEARCH	FORD
3	SALES	BLAKE

8. Виведіть назву місті та співробітника, що має найбільшу зарплату у своєму місті.

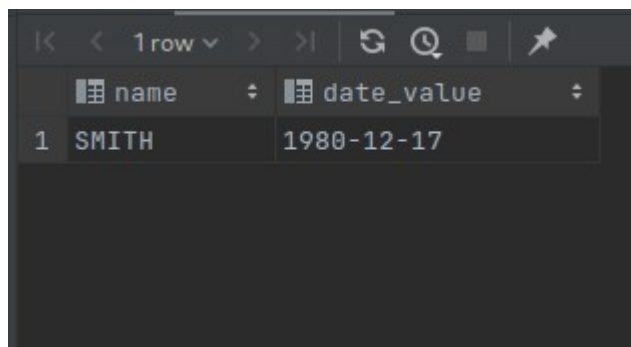
```
select
  p1.text_value as name,
  o2.name as emp_name
from params p1
join (
  select distinct on (loc) empno, dept, loc, sal from (
    select
      pp1.object_id as empno,
      pp2.numeric_value as dept,
      pp1.numeric_value as sal,
      pp3.text_value as loc
    from params pp1
    join params pp2 on (
      pp1.object_id = pp2.object_id
      and pp1.attr_id = 3 and pp2.attr_id = 5
    )
    join params pp3 on (
      pp2.numeric_value = pp3.object_id
      and pp1.attr_id = 3 and pp3.attr_id = 8
    )
  ) pr1
  order by loc desc, sal desc
) pr2 on (p1.text_value = pr2.loc)
join objects o2 on (o2.object_id = pr2.empno)
where
  p1.attr_id = 8;
```



	name	emp_name
1	NEW_YORK	KING
2	DALLAS	FORD
3	CHICAGO	BLAKE

9. Людина, що була першою прийнята на роботу.

```
select
  objects.name,
  params.date_value
from objects
join params on (
  params.object_id = objects.object_id
  and params.attr_id = 2
)
where object_type_id = 1
order by params.date_value asc
limit 1
;
```



The screenshot shows a database query result viewer with a toolbar at the top containing icons for navigation, search, and other functions. Below the toolbar is a table with two columns: 'name' and 'date\_value'. The table contains one row of data.

	name	date_value
1	SMITH	1980-12-17

10. Відділ та середній дохід (зарплата + премія) по відділу.

```
select
  o1.name,
  p1.avg_money
from objects o1
join (
  select pr1.deptno, avg(pr1.money) as avg_money from (
    select
      p1.numeric_value + coalesce(p2.numeric_value, 0) as money,
      p3.numeric_value as deptno
    from params p1
    left join params p2 on (
      p1.object_id = p2.object_id
      and p2.attr_id = 4
    )
    join params p3 on (
      p1.object_id = p3.object_id
      and p3.attr_id = 5
    )
    where p1.attr_id = 3
  ) pr1
  group by deptno
) p1
on (o1.object_id = p1.deptno)
where
  o1.object_type_id = 3;
```

	name	avg_money
1	ACCOUNTING	2600
2	RESEARCH	2175
3	SALES	1933.3333333333333

11. Виведіть посаду та діапазон заробітних плат за цією посадою (мінімальна-максимальна).

```
select
  p1.text_value as job,
  max(p2.numeric_value) as maxsal,
  min(p2.numeric_value) as minsal
from params p1
join params p2 on (
  p1.object_id = p2.object_id
  and p1.attr_id = 1
  and p2.attr_id = 3
)
group by p1.text_value
```

	job	maxsal	minsal
1	MANAGER	2975	1500
2	ANALYST	3000	3000
3	SALESMAN	1600	1250
4	CLERK	1300	800
5	PRESEDENT	5000	5000



## Задання 2

Доповніть (Insert) таблиці із завдання 1 відповідно до даних предметної області. Додайте 1-2 типи об'єктів із предметної області вашої курсової роботи.


Дані таблиці Object\_types описуються деревом (+ 1-2 типи об'єктів з курсової):

- ◆ Контейнер (тип All)
  - Географічні об'єкти
    - Область
    - Місто
    - Вулиця
    - Будівля
  - Частини будівель
    - Поверх
    - Квартира
  - Обладнання
    - Мережеве обладнання
    - Кухонне обладнання

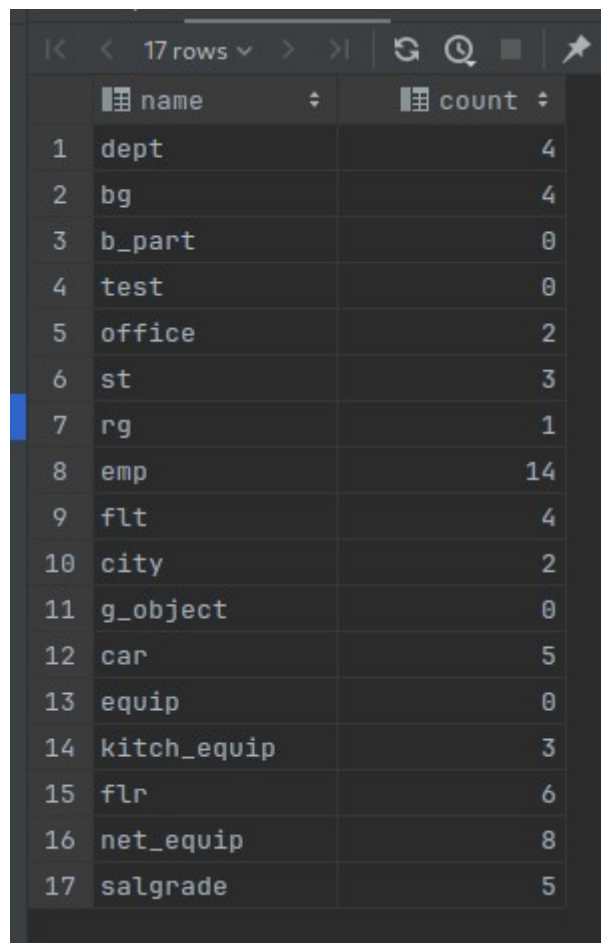
Дані таблиці Objects описуються деревом:

- Сумська область
  - Охтирка
    - Харківська вулиця
      - Будинок 1
        - Поверх 1
  - Суми
    - Харківська вулиця
      - Будинок 1
        - Поверх 1
    - Миколи Сумцова (ex. Римського-Корсакова)
      - Будинок 1
        - Поверх 1
        - Поверх 2
      - Будинок 2
        - Поверх 1
        - Поверх 2

Розмістите поверхх кілька квартир (разом 3-5), у будівлях, на поверхх, в кімнатах – обладнання різного виду (кавоварки, холодильники, роутери, ноутбуки) – разом до 10-15 об'єктів. Додайте 3-5 об'єктів із предметної області Вашої курсової роботи.

 Напишіть запит, що видає назви об'єктних типів, та скільки екземплярів об'єктів цього типу є у Вашій базі (включаючи типи без екземплярів.)

```
select
    object_types.name,
    count(o.object_id)
from object_types
left join objects o on (object_types.object_type_id = o.object_type_id)
group by object_types.name
;
```




	name	count
1	dept	4
2	bg	4
3	b_part	0
4	test	0
5	office	2
6	st	3
7	rg	1
8	emp	14
9	flt	4
10	city	2
11	g_object	0
12	car	5
13	equip	0
14	kitch_equip	3
15	flr	6
16	net_equip	8
17	salgrade	5

## Завдання 3



1. Виведіть назви об'єктів типу «Будівля».

```
select
    object_id,
    name
from objects
where object_type_id = 10;
```

	 object_id	 name
1	1009	House 1
2	1014	House 1
3	1016	House 1
4	1017	House 2

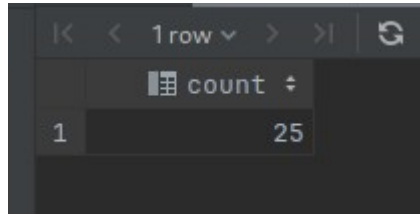
2. Виведіть назви об'єктів, що належать до типу «Обладнання» або його спадкоємцям.

```
select
    objects.object_id,
    objects.name
from objects
join object_types ot on (objects.object_type_id = ot.object_type_id)
where ot.object_type_id = 6 or ot.parent_id = 6
;
```

	 object_id	 name
1	1029	Computer
2	1030	Router
3	1031	Coffe machine
4	1026	Computer
5	1027	Router
6	1028	Coffe machine
7	1032	Laptop
8	1033	Switch
9	1034	Oven
10	1035	Laptop
11	1036	Router

3. Скільки об'єктів знаходяться в об'єктах типу «Будівля» (включаючи всі вкладанні об'єкти).

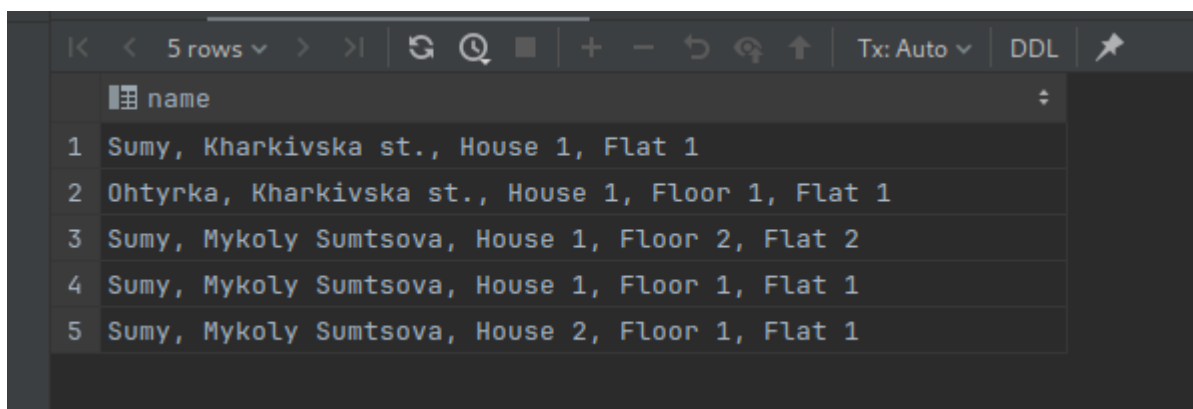
```
with recursive cte_query as (  
    select o.* from objects o where object_type_id = 10  
    union  
    select o.*  
    from cte_query  
    inner join objects o on cte_query.object_id = o.parent_id  
)  
select count(*) from cte_query;
```



	count
1	25

4. Виведіть адреси всіх квартир у форматі м.<Місто>, <Вулиця> <Будинок>, кв <Квартира>.

```
with recursive cte_query as (  
    select o.* from objects o where object_type_id = 12  
    union  
    select o.object_id, o.parent_id, o.object_type_id, o.name || ', ' ||  
cte_query.name  
    from cte_query  
    inner join objects o on coalesce(cte_query.parent_id, 0) =  
o.object_id  
)  
select name from cte_query  
where object_type_id = 8;
```



	name
1	Sumy, Kharkivska st., House 1, Flat 1
2	Ohtyrka, Kharkivska st., House 1, Floor 1, Flat 1
3	Sumy, Mykoly Sumtsova, House 1, Floor 2, Flat 2
4	Sumy, Mykoly Sumtsova, House 1, Floor 1, Flat 1
5	Sumy, Mykoly Sumtsova, House 2, Floor 1, Flat 1

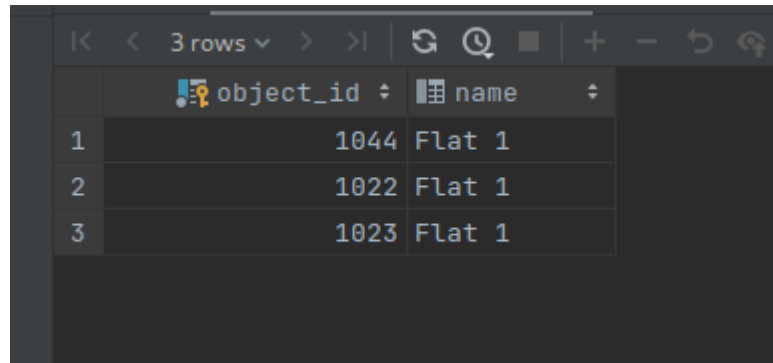
5. Для всього мережевого обладнання у Сумах виведіть його назву та адресу (див 3.4) У випадках, коли обладнання розташоване у будинку чи на поверсі – квартира не зазначається.

```
with recursive cte_query as (  
    select o.* from objects o where object_type_id = 13  
    union  
    select o.object_id, o.parent_id, o.object_type_id, o.name || ', ' ||  
cte_query.name  
    from cte_query  
    inner join objects o on coalesce(cte_query.parent_id, 0) =  
o.object_id  
)  
select name from cte_query  
where object_type_id = 8 and name like 'Sumy%';
```

	name
1	Sumy, Kharkivska st., House 1, Flat 1, Computer
2	Sumy, Kharkivska st., House 1, Flat 1, Router
3	Sumy, Mykoły Sumtsova, House 1, Floor 1, Flat 1, Computer
4	Sumy, Mykoły Sumtsova, House 1, Floor 1, Flat 1, Router
5	Sumy, Mykoły Sumtsova, House 1, Floor 2, Flat 2, Laptop
6	Sumy, Mykoły Sumtsova, House 1, Floor 2, Flat 2, Switch
7	Sumy, Mykoły Sumtsova, House 2, Floor 1, Flat 1, Laptop
8	Sumy, Mykoły Sumtsova, House 2, Floor 1, Flat 1, Router

10. В яких квартирах є кавоварки?

```
select
  o1.object_id,
  o1.name
from objects o1
join objects o2 on o2.parent_id = o1.object_id
where o2.name = 'Coffe machine';
```



The screenshot shows a database interface with a table containing 3 rows. The table has two columns: 'object\_id' and 'name'. The rows are numbered 1, 2, and 3. The data is as follows:


	object_id	name
1	1044	Flat 1
2	1022	Flat 1
3	1023	Flat 1

## Завдання 4. Entity-Attribute-Value

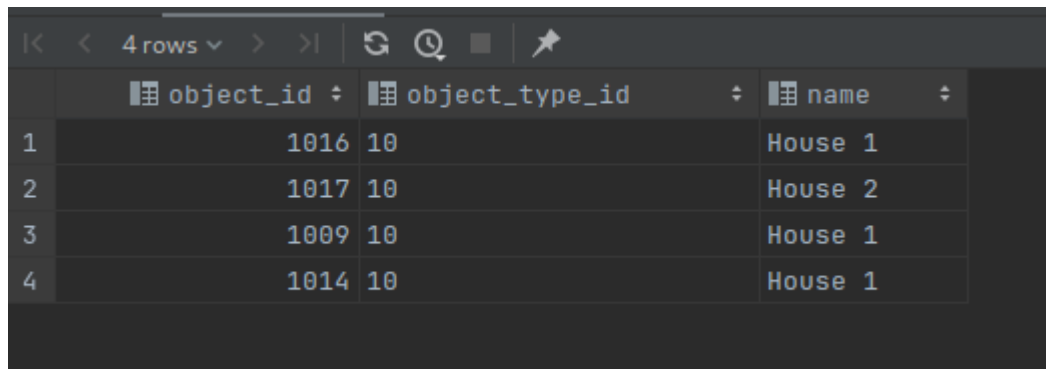
На основі таблиць завдань 1-3 побудуйте представлення, що емулює сховище даних на основі EAV (Створюємо view на основі таблиць завдання 1, який організовує та відображає дані відповідно до підходу EAV. Подання зберігає номер об'єкта, назву атрибута та значення атрибута. Тип об'єкта та його назва об'єкта стають атрибутами об'єкта.)

```
create view eav_view as(
  select object_id, 'parent_id' as attr, cast(parent_id as varchar) as
value from objects
  union
  select object_id, 'object_type_id' as attr, cast(object_type_id as
varchar) as value from objects
  union
  select object_id, 'name' as attr, name as value from objects
  union
  select object_id, 'numeric_value' as attr, cast(numeric_value as varchar)
as value from params
  union
  select object_id, 'text_value' as attr, text_value as value from params
  union
  select object_id, 'date_value' as attr, cast(date_value as varchar) as
value from params
)
```

WHERE		ORDER BY	
	object_id	attr	value
1	1037	name	Sumy office
2	1041	object_type_id	15
3	7369	object_type_id	1
4	4	object_type_id	2
5	1011	parent_id	1006
6	1029	name	Computer
7	7839	object_type_id	1
8	1041	parent_id	<null>
9	7788	name	SCOTT
10	1	parent_id	<null>
11	1029	object_type_id	13
12	20	object_type_id	3
13	1024	name	Flat 2
14	1021	name	Floor 2

 Виконайте завдання 3.1 із використанням новоствореного view

```
select
  o1.object_id,
  o1.value as object_type_id,
  o2.value as name
from eav_view o1
join eav_view o2 on (o1.object_id = o2.object_id and o2.attr = 'name')
where o1.attr = 'object_type_id' and o1.value = '10'
```



	object_id	object_type_id	name
1	1016	10	House 1
2	1017	10	House 2
3	1009	10	House 1
4	1014	10	House 1

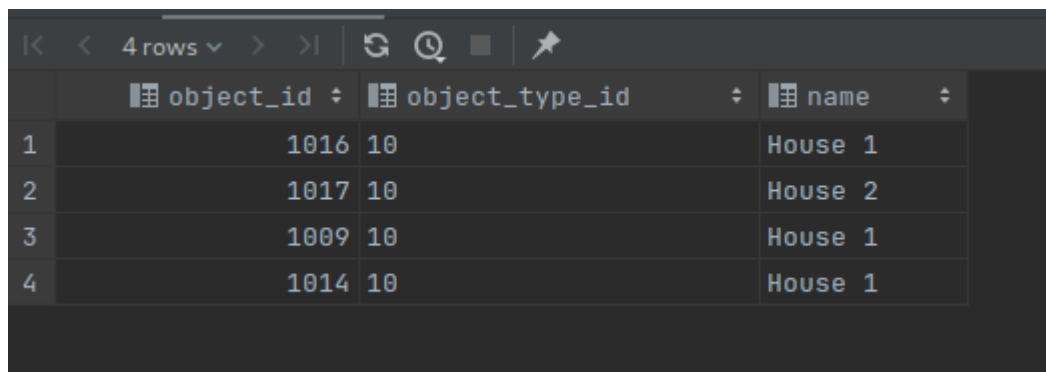


## Завдання 5

Виберіть по 3 питання із завдань 1 та 3 (всього 6). Виконайте їх із використанням подання із завдання 4.

1. Виведіть назви об'єктів типу «Будівля».

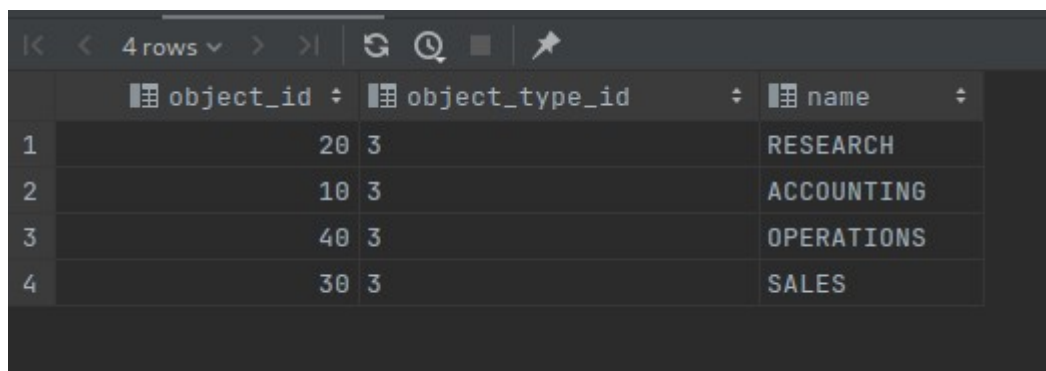
```
select
  o1.object_id,
  o1.value as object_type_id,
  o2.value as name
from eav_view o1
join eav_view o2 on (o1.object_id = o2.object_id and o2.attr = 'name')
where o1.attr = 'object_type_id' and o1.value = '10'
```



	object_id	object_type_id	name
1	1016	10	House 1
2	1017	10	House 2
3	1009	10	House 1
4	1014	10	House 1

2. Виведіть список підрозділів фірми (Раніше ці дані зберігалися у таблиці DEPT)

```
select
  o1.object_id,
  o1.value as object_type_id,
  o2.value as name
from eav_view o1
join eav_view o2 on (o1.object_id = o2.object_id and o2.attr = 'name')
where o1.attr = 'object_type_id' and o1.value = '3'
```



	object_id	object_type_id	name
1	20	3	RESEARCH
2	10	3	ACCOUNTING
3	40	3	OPERATIONS
4	30	3	SALES

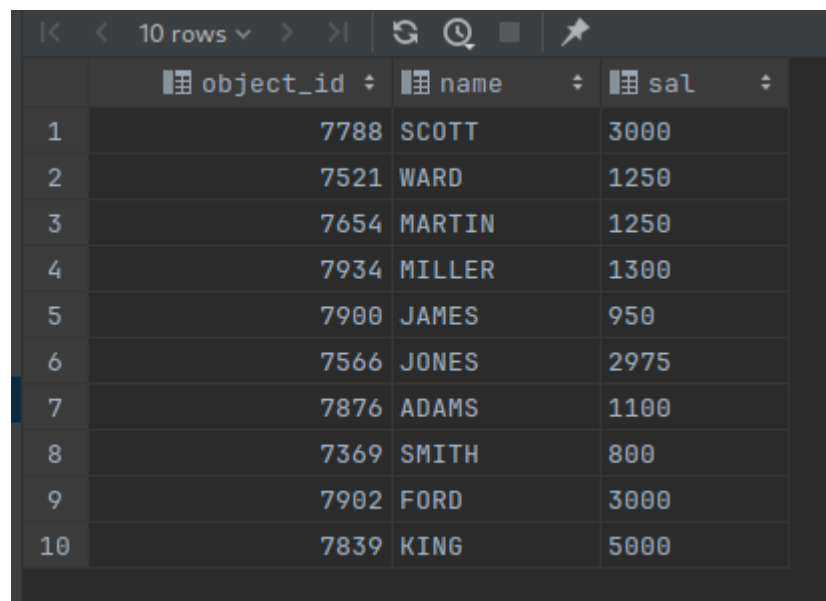
3. Сформувати запит, що виводить 2 стовпці інформації:
- а. прізвище працівника та, через кому, його посаду;
  - б. його зарплату за рік.

```
select
  o1.object_id,
  (o2.value || ', ' || o3.value) as "Name/Job",
  o4.value as sal
from eav_view o1
join eav_view o2 on (o1.object_id = o2.object_id and o2.attr = 'name')
join eav_view o3 on (o1.object_id = o3.object_id and o3.attr = '1' and
o3.value is not NULL )
join eav_view o4 on (o1.object_id = o4.object_id and o4.attr = '3' and
o4.value is not NULL)
where o1.attr = 'object_type_id' and o1.value = '1'
```

	object_id	"Name/Job"	sal
1	7788	SCOTT, ANALYST	3000
2	7521	WARD, SALESMAN	1250
3	7654	MARTIN, SALESMAN	1250
4	7934	MILLER, CLERK	1300
5	7782	CLARK, MANAGER	1500
6	7900	JAMES, CLERK	950
7	7698	BLAKE, MANAGER	2850
8	7566	JONES, MANAGER	2975
9	7876	ADAMS, CLERK	1100
10	7844	TURNER, SALESMAN	1500
11	7499	ALLEN, SALESMAN	1600
12	7369	SMITH, CLERK	800
13	7902	FORD, ANALYST	3000
...	---	-----	----

4. Сформулювати запит, що виводить прізвище та зарплату працівників, зарплата яких лежить поза діапазоном від \$1500 до \$2850.

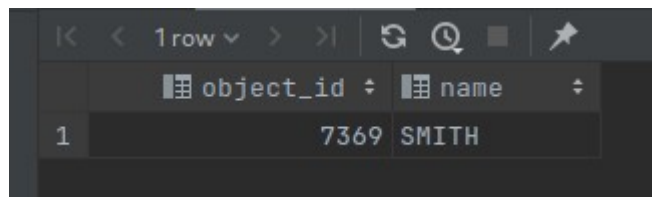
```
select
  o1.object_id,
  o2.value as name,
  o3.value as sal
from eav_view o1
join eav_view o2 on (o1.object_id = o2.object_id and o2.attr = 'name')
join eav_view o3 on (
  o1.object_id = o3.object_id
  and o3.attr = '3'
  and o3.value is not NULL
)
where
  o1.attr = 'object_type_id'
  and o1.value = '1'
  and (case when (o3.value ~ '^[0-9\.]+'$) then o3.value::int not between
1500 and 2850 else true end);
```



	object_id	name	sal
1	7788	SCOTT	3000
2	7521	WARD	1250
3	7654	MARTIN	1250
4	7934	MILLER	1300
5	7900	JAMES	950
6	7566	JONES	2975
7	7876	ADAMS	1100
8	7369	SMITH	800
9	7902	FORD	3000
10	7839	KING	5000

5. Людина, що була першою прийнята на роботу.

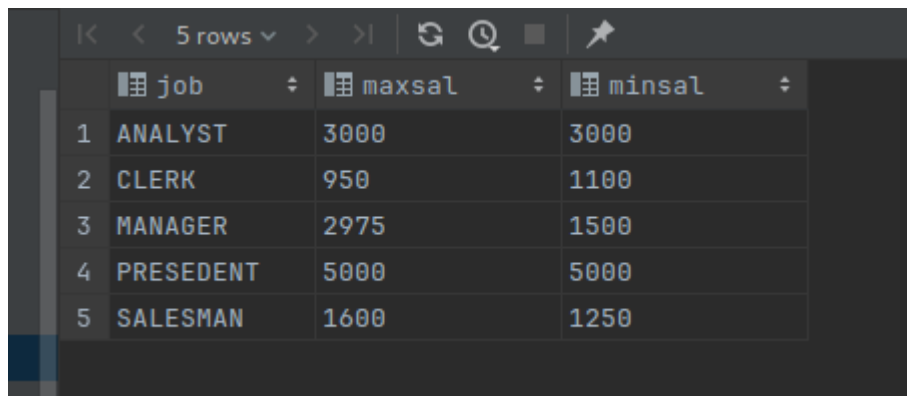
```
select
  o1.object_id,
  o2.value as name,
  o3.value as date
from eav_view o1
join eav_view o2 on (o1.object_id = o2.object_id and o2.attr = 'name')
join eav_view o3 on (
  o1.object_id = o3.object_id
  and o3.attr = '2'
  and o3.value is not NULL
)
where
  o1.attr = 'object_type_id'
  and o1.value = '1'
order by o3.value::date
limit 1
```



	object_id	name
1	7369	SMITH

6. Виведіть посаду та діапазон заробітних плат за цією посадою (мінімальна-максимальна).

```
select
  o2.value as job,
  max(o3.value) as maxsal,
  min(o3.value) as minsal
from eav_view o1
join eav_view o2 on (
  o1.object_id = o2.object_id
  and o2.attr = '1'
  and o2.value is not NULL
)
join eav_view o3 on (
  o2.object_id = o3.object_id
  and o3.attr = '3'
  and o3.value is not NULL
)
where
  o1.attr = 'object_type_id'
  and o1.value = '1'
group by o2.value
```



The screenshot shows a database query result with 5 rows. The columns are job, maxsal, and minsal. The data is as follows:

	job	maxsal	minsal
1	ANALYST	3000	3000
2	CLERK	950	1100
3	MANAGER	2975	1500
4	PRESEDENT	5000	5000
5	SALESMAN	1600	1250