

# Лабораторна робота 6

## «DML»

Для завдань/розділів відмічених « » – скріншот – обов'язково. Для інших – за домовленістю із викладачем.

Завдання лабораторної роботи виконую всі

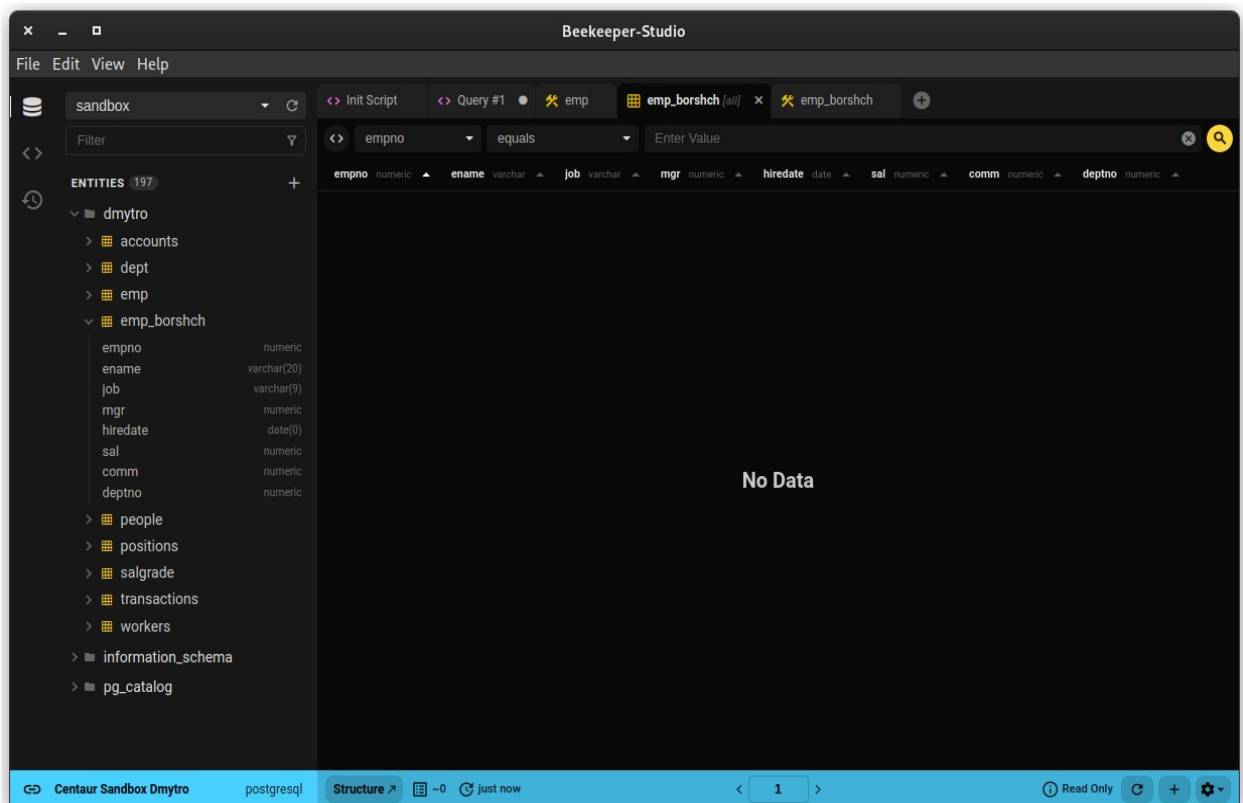
### Основне завдання

#### Insert

1. Створіть таблицю emp\_yourname (де yourname-Ваше прізвище), з такими ж полями, як і в таблиці emp. Для цього можна скористатися запитом:

```
create table emp_yourname as select * from emp where 1=2
```

Перевірте, чи все пройшло правильно.



Що дає умова 1 = 2?

1 = 2 означає завжди False тому не будуть скопійовані рядки, а лише структура таблиці.

**\*\*** Які обмеження із оригінальної таблиці буди перенесені, а які ні?  
Primary key не зберігся, але default значення ідентичні.

2. Вставте у таблицю emp\_ouiname відомості про себе, як нового співробітника, який працює у відділі 10. Заповніть значення полів за вашим бажанням. Дата прийому на роботу - поточна.

```
INSERT INTO emp_borshch (empno, ename, hiredate, deptno) VALUES (1, 'BORSHCH', current_date, 10);
```

3. Очистіть таблицю emp\_ouiname із застосуванням Truncate. Чи вийде те саме із dept?

```
TRUNCATE TABLE emp_borshch;
```

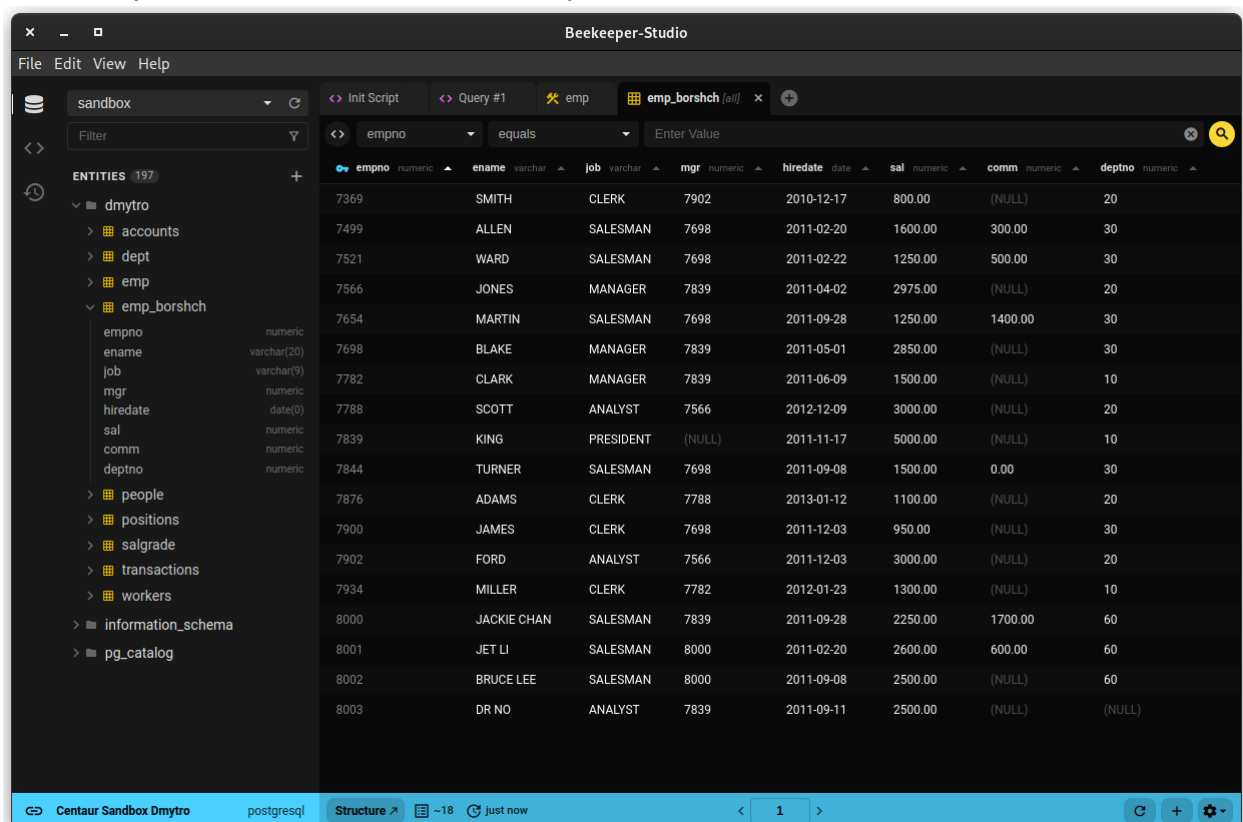
4. Переконайтеся, що таблиця emp\_ouiname порожня.

```
SELECT * FROM emp_borshch
```

5. Вставте всі дані з таблиці Emp у таблицю emp\_ouiname .

```
INSERT INTO emp_borshch SELECT * FROM emp;
```

6. Переконайтеся, що всі співробітники додані.



The screenshot shows the Beekeeper-Studio interface. On the left, the 'ENTITIES' sidebar lists databases and tables, with 'emp\_borshch' selected. The main area displays the table structure and data. The table has columns: empno (numeric), ename (varchar), job (varchar), mgr (numeric), hiredate (date), sal (numeric), comm (numeric), and deptno (numeric). The data is as follows:

empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	2010-12-17	800.00	(NULL)	20
7499	ALLEN	SALESMAN	7698	2011-02-20	1600.00	300.00	30
7521	WARD	SALESMAN	7698	2011-02-22	1250.00	500.00	30
7566	JONES	MANAGER	7839	2011-04-02	2975.00	(NULL)	20
7654	MARTIN	SALESMAN	7698	2011-09-28	1250.00	1400.00	30
7698	BLAKE	MANAGER	7839	2011-05-01	2850.00	(NULL)	30
7782	CLARK	MANAGER	7839	2011-06-09	1500.00	(NULL)	10
7788	SCOTT	ANALYST	7566	2012-12-09	3000.00	(NULL)	20
7839	KING	PRESIDENT	(NULL)	2011-11-17	5000.00	(NULL)	10
7844	TURNER	SALESMAN	7698	2011-09-08	1500.00	0.00	30
7876	ADAMS	CLERK	7788	2013-01-12	1100.00	(NULL)	20
7900	JAMES	CLERK	7698	2011-12-03	950.00	(NULL)	30
7902	FORD	ANALYST	7566	2011-12-03	3000.00	(NULL)	20
7934	MILLER	CLERK	7782	2012-01-23	1300.00	(NULL)	10
8000	JACKIE CHAN	SALESMAN	7839	2011-09-28	2250.00	1700.00	60
8001	JET LI	SALESMAN	8000	2011-02-20	2600.00	600.00	60
8002	BRUCE LEE	SALESMAN	8000	2011-09-08	2500.00	(NULL)	60
8003	DR NO	ANALYST	7839	2011-09-11	2500.00	(NULL)	(NULL)

7. Оновіть сценарій заповнення таблиці emp, так щоб ви були присутні у даних цієї таблиці. Додайте себе у неї. (Мета - зробіть так, щоб Ви і зараз були у таблиці і у всіх майбутніх її перествореннях - теж з'являлись)

empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	2010-12-17	800.00	(NULL)	20
7499	ALLEN	SALESMAN	7698	2011-02-20	1600.00	300.00	30
7521	WARD	SALESMAN	7698	2011-02-22	1250.00	500.00	30
7566	JONES	MANAGER	7839	2011-04-02	2975.00	(NULL)	20
7654	MARTIN	SALESMAN	7698	2011-09-28	1250.00	1400.00	30
7698	BLAKE	MANAGER	7839	2011-05-01	2850.00	(NULL)	30
7782	CLARK	MANAGER	7839	2011-06-09	1500.00	(NULL)	10
7788	SCOTT	ANALYST	7566	2012-12-09	3000.00	(NULL)	20
7839	KING	PRESIDENT	(NULL)	2011-11-17	5000.00	(NULL)	10
7844	TURNER	SALESMAN	7698	2011-09-08	1500.00	0.00	30
7876	ADAMS	CLERK	7788	2013-01-12	1100.00	(NULL)	20
7900	JAMES	CLERK	7698	2011-12-03	950.00	(NULL)	30
7902	FORD	ANALYST	7566	2011-12-03	3000.00	(NULL)	20
7934	MILLER	CLERK	7782	2012-01-23	1300.00	(NULL)	10
8000	JACKIE CHAN	SALESMAN	7839	2011-09-28	2250.00	1700.00	60
8001	JET LI	SALESMAN	8000	2011-02-20	2600.00	600.00	60
8002	BRUCE LEE	SALESMAN	8000	2011-09-08	2500.00	(NULL)	60
8003	DR NO	ANALYST	7839	2011-09-11	2500.00	(NULL)	(NULL)
8004	BORSHCH	TRAINEE	7839	2022-11-19	0.00	(NULL)	10

## Update

1. Додайте всім службовцям із таблиці emp\_yourname \$200 зарплати.

```
UPDATE emp_borshch SET sal = sal + 200;
```

2. \* Одним запитом додайте всіма службовцям ще \$200 зарплати та \$100 премії.

```
UPDATE emp_borshch SET sal = sal + 200, comm = comm + 100;
```

3. \*\* Для людей, з доходом менше 3000, додайте 25% зарплати та Z у.о. до премії, тож дохід був щонайменше 3000. Знижувати премію не можна. (Якщо після підвищення зарплати дохід вже вищий за 3000 - просто не підвищуємо премію)

```
UPDATE emp_borshch SET comm = 0 WHERE comm IS NULL;
```

```
UPDATE emp_borshch SET sal = sal + sal*0.25 WHERE (sal + comm) < 3000;
```

```
UPDATE emp_borshch SET comm = comm + (3000 - (sal + comm)) WHERE (sal + comm) < 3000;
```

4. Заберіть у всіх службовців премію.

```
UPDATE emp_borshch SET comm = 0;
```

## Delete

5. Видаліть службовців із 10 відділу з таблиці emp\_ourname.

Переконайтеся, що службовці видалені.

```
DELETE FROM emp_borshch WHERE deptno = 10;
```

```
SELECT * FROM emp_borshch WHERE deptno = 10;
```

## NULL vs DEFAULT

6. Створимо таблицю:

```
CREATE TABLE TBL1(A integer, B integer DEFAULT 5);
```

Якими будуть результати команд:

Insert into tbl1 values (default, default);

значення: null, 5

Insert int tbl1 values (null, null);

значення: null, null

Insert int tbl1(a) values (default);

значення: null, 5

Insert int tbl1(b) values (default);

значення: null, 5

Insert int tbl1(a) values (null);

значення: null, 5

Insert int tbl1(b) values (null);

значення: null, null

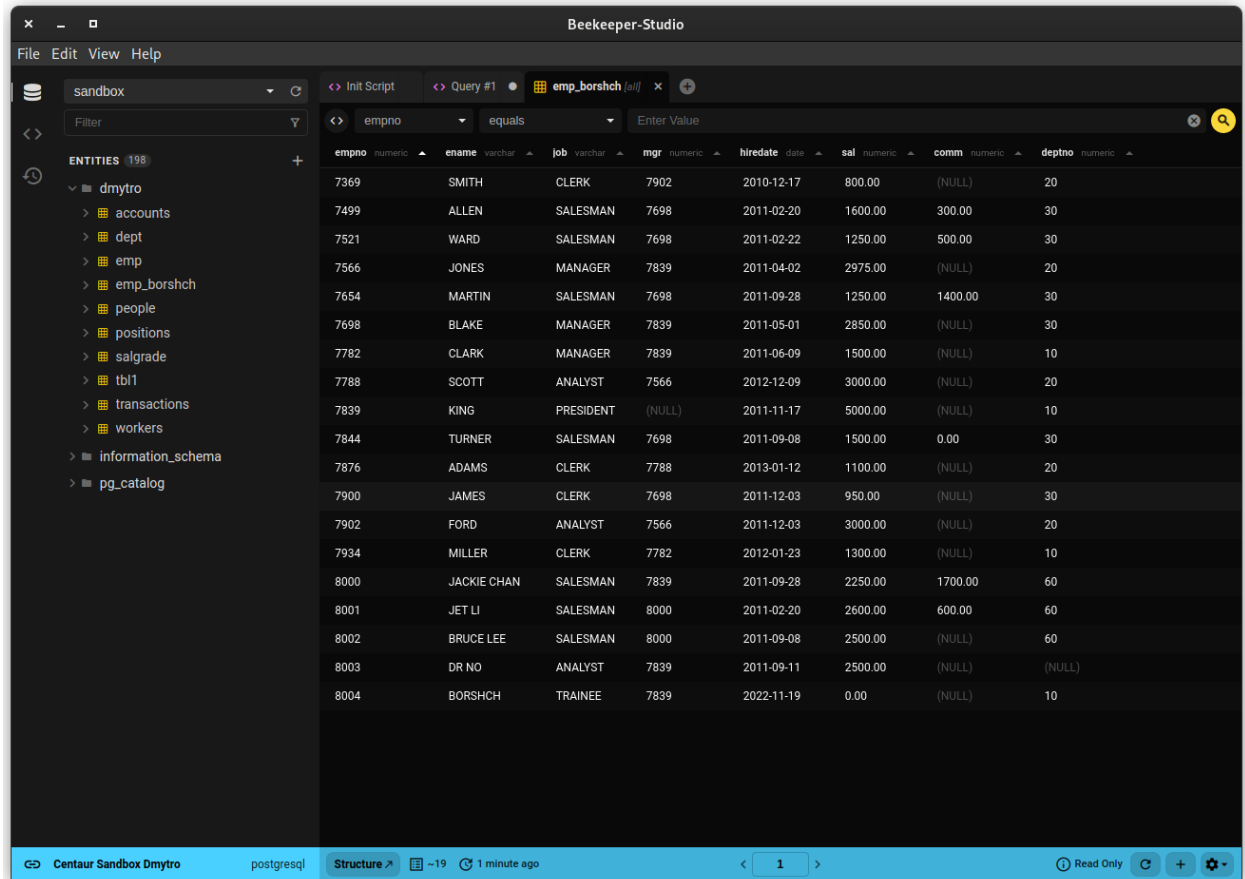
Insert int tbl1 values ();

в postgres отримав помилку синтаксису

## Корельовані підзапити в UPDATE I DELETE

7. Створіть таблицю emp\_username на основі таблиці EMP та скопіюйте у неї всі дані.

```
CREATE TABLE emp_borshch AS SELECT * FROM emp;
```



The screenshot shows the Beekeeper-Studio interface. On the left, the 'ENTITIES' sidebar lists databases and tables, including 'emp\_borshch'. The main panel displays the table structure and data for 'emp\_borshch'. The table has columns: empno, ename, job, mgr, hiredate, sal, comm, and deptno. The data is as follows:

empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	2010-12-17	800.00	(NULL)	20
7499	ALLEN	SALESMAN	7698	2011-02-20	1600.00	300.00	30
7521	WARD	SALESMAN	7698	2011-02-22	1250.00	500.00	30
7566	JONES	MANAGER	7839	2011-04-02	2975.00	(NULL)	20
7654	MARTIN	SALESMAN	7698	2011-09-28	1250.00	1400.00	30
7698	BLAKE	MANAGER	7839	2011-05-01	2850.00	(NULL)	30
7782	CLARK	MANAGER	7839	2011-06-09	1500.00	(NULL)	10
7788	SCOTT	ANALYST	7566	2012-12-09	3000.00	(NULL)	20
7839	KING	PRESIDENT	(NULL)	2011-11-17	5000.00	(NULL)	10
7844	TURNER	SALESMAN	7698	2011-09-08	1500.00	0.00	30
7876	ADAMS	CLERK	7788	2013-01-12	1100.00	(NULL)	20
7900	JAMES	CLERK	7698	2011-12-03	950.00	(NULL)	30
7902	FORD	ANALYST	7566	2011-12-03	3000.00	(NULL)	20
7934	MILLER	CLERK	7782	2012-01-23	1300.00	(NULL)	10
8000	JACKIE CHAN	SALESMAN	7839	2011-09-28	2250.00	1700.00	60
8001	JET LI	SALESMAN	8000	2011-02-20	2600.00	600.00	60
8002	BRUCE LEE	SALESMAN	8000	2011-09-08	2500.00	(NULL)	60
8003	DR NO	ANALYST	7839	2011-09-11	2500.00	(NULL)	(NULL)
8004	BORSHCH	TRAINEE	7839	2022-11-19	0.00	(NULL)	10

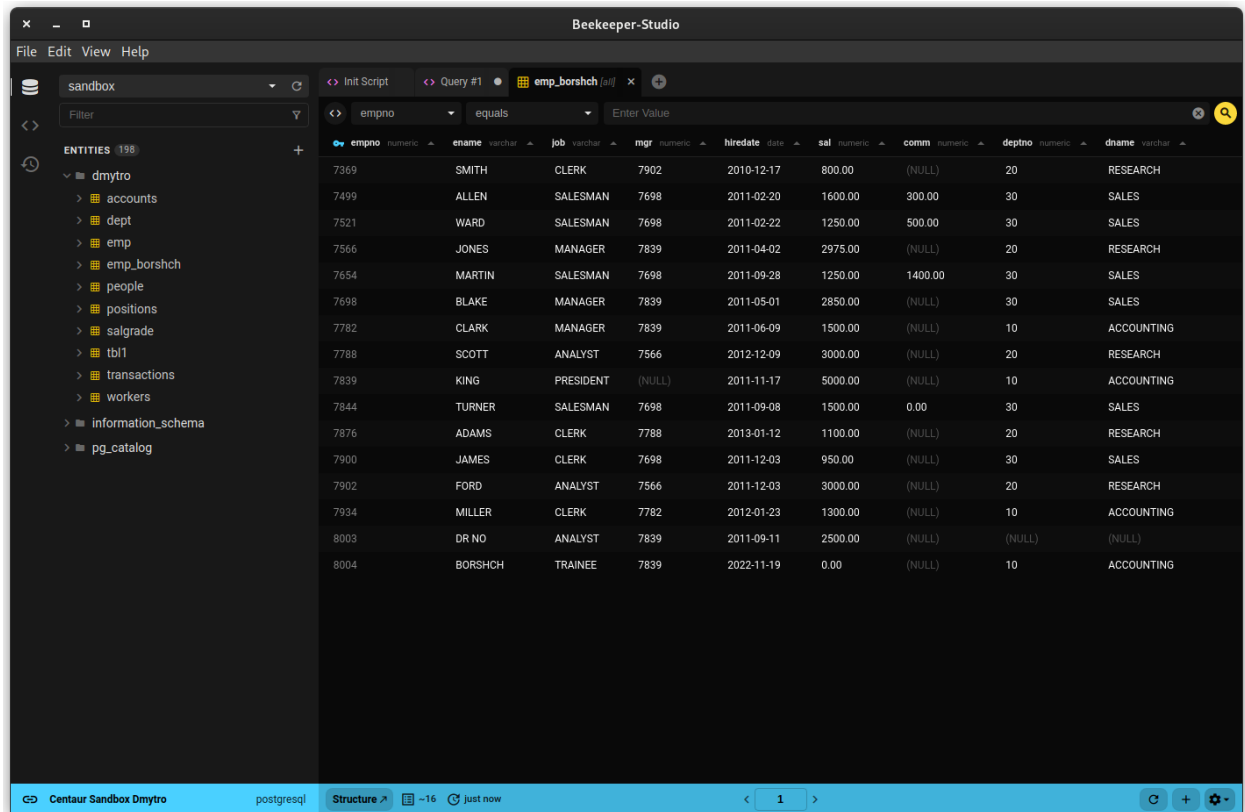
8. Денормалізуйте таблицю emp\_username додавши до неї стовпець dname. Заповніть цей стовпець на основі даних таблиці DEPT.

```
ALTER TABLE emp_borshch ADD DNAME VARCHAR;
```

```
UPDATE emp_borshch SET dname = (SELECT dname FROM dept WHERE dept.deptno = emp_borshch.deptno);
```

9. \* Видалить відомості з таблиці emp\_ouiname про службовців, які працюють у HONKONG

```
DELETE FROM emp_borshch WHERE dname in (SELECT dname FROM dept WHERE loc = 'HONKONG');
```



The screenshot shows the Beekeeper-Studio interface with the 'emp\_borshch' table selected. The table contains 16 rows of employee data. The columns are: empno, ename, job, mgr, hiredate, sal, comm, deptno, and dname. The data is as follows:

empno	ename	job	mgr	hiredate	sal	comm	deptno	dname
7369	SMITH	CLERK	7902	2010-12-17	800.00	(NULL)	20	RESEARCH
7499	ALLEN	SALESMAN	7698	2011-02-20	1600.00	300.00	30	SALES
7521	WARD	SALESMAN	7698	2011-02-22	1250.00	500.00	30	SALES
7566	JONES	MANAGER	7839	2011-04-02	2975.00	(NULL)	20	RESEARCH
7654	MARTIN	SALESMAN	7698	2011-09-28	1250.00	1400.00	30	SALES
7698	BLAKE	MANAGER	7839	2011-05-01	2850.00	(NULL)	30	SALES
7782	CLARK	MANAGER	7839	2011-06-09	1500.00	(NULL)	10	ACCOUNTING
7788	SCOTT	ANALYST	7566	2012-12-09	3000.00	(NULL)	20	RESEARCH
7839	KING	PRESIDENT	(NULL)	2011-11-17	5000.00	(NULL)	10	ACCOUNTING
7844	TURNER	SALESMAN	7698	2011-09-08	1500.00	0.00	30	SALES
7876	ADAMS	CLERK	7788	2013-01-12	1100.00	(NULL)	20	RESEARCH
7900	JAMES	CLERK	7698	2011-12-03	950.00	(NULL)	30	SALES
7902	FORD	ANALYST	7566	2011-12-03	3000.00	(NULL)	20	RESEARCH
7934	MILLER	CLERK	7782	2012-01-23	1300.00	(NULL)	10	ACCOUNTING
8003	DR NO	ANALYST	7839	2011-09-11	2500.00	(NULL)	(NULL)	(NULL)
8004	BORSHCH	TRAINEE	7839	2022-11-19	0.00	(NULL)	10	ACCOUNTING

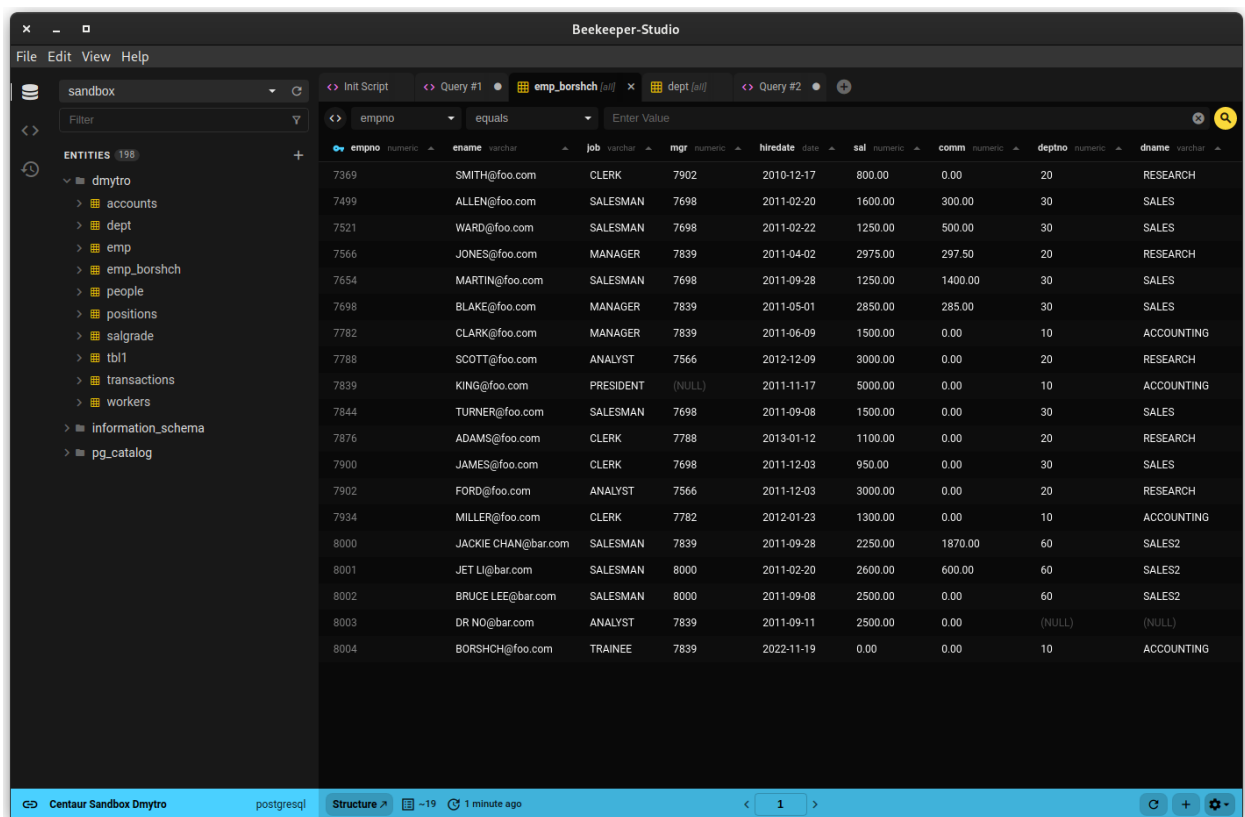
10. \* Ми хочемо додати до email співробітника назву домену. Наприклад, «JohnDoe»→«JohnDoe@foo.com». Оновить данні у колонці email, так щоб співробітники із USA (New York, Boston, Dallas, ... ) мали домен «foo.com», а всі інші «bar.com». Вважаємо що кількість співробітників у EMP велика і наперед не відома. Для «\*\*» - зробіть оновлення одним запитом.

```
UPDATE emp_borshch SET ename = CASE
    WHEN (emp_borshch.deptno NOT IN (SELECT deptno FROM dept WHERE
loc = 'HONKONG')) THEN
        (ename || '@foo.com')
    ELSE
        (ename || '@bar.com')
END
```

empno	ename	job	mgr	hiredate	sal	comm	deptno	dname
7369	SMITH@foo.com	CLERK	7902	2010-12-17	800.00	(NULL)	20	(NULL)
7499	ALLEN@foo.com	SALESMAN	7698	2011-02-20	1600.00	300.00	30	(NULL)
7521	WARD@foo.com	SALESMAN	7698	2011-02-22	1250.00	500.00	30	(NULL)
7566	JONES@foo.com	MANAGER	7839	2011-04-02	2975.00	(NULL)	20	(NULL)
7654	MARTIN@foo.com	SALESMAN	7698	2011-09-28	1250.00	1400.00	30	(NULL)
7698	BLAKE@foo.com	MANAGER	7839	2011-05-01	2850.00	(NULL)	30	(NULL)
7782	CLARK@foo.com	MANAGER	7839	2011-06-09	1500.00	(NULL)	10	(NULL)
7788	SCOTT@foo.com	ANALYST	7566	2012-12-09	3000.00	(NULL)	20	(NULL)
7839	KING@foo.com	PRESIDENT	(NULL)	2011-11-17	5000.00	(NULL)	10	(NULL)
7844	TURNER@foo.com	SALESMAN	7698	2011-09-08	1500.00	0.00	30	(NULL)
7876	ADAMS@foo.com	CLERK	7788	2013-01-12	1100.00	(NULL)	20	(NULL)
7900	JAMES@foo.com	CLERK	7698	2011-12-03	950.00	(NULL)	30	(NULL)
7902	FORD@foo.com	ANALYST	7566	2011-12-03	3000.00	(NULL)	20	(NULL)
7934	MILLER@foo.com	CLERK	7782	2012-01-23	1300.00	(NULL)	10	(NULL)
8000	JACKIE CHAN@bar.com	SALESMAN	7839	2011-09-28	2250.00	1700.00	60	(NULL)
8001	JET LI@bar.com	SALESMAN	8000	2011-02-20	2600.00	600.00	60	(NULL)
8002	BRUCE LEE@bar.com	SALESMAN	8000	2011-09-08	2500.00	(NULL)	60	(NULL)
8003	DR NO@bar.com	ANALYST	7839	2011-09-11	2500.00	(NULL)	(NULL)	(NULL)
8004	BORSHCH@foo.com	TRAINEE	7839	2022-11-19	0.00	(NULL)	10	(NULL)

11. \* Додати 10% премії людям, у яких керівник працює у іншому відділі ( для «\*\*» - іншому місті). Якщо премії раніш не було - встановлюємо у розмірі 10%.

```
UPDATE emp_borshch a SET comm = CASE
    WHEN (comm > 0) THEN
        comm + comm * 0.1
    ELSE
        sal * 0.1
    END
WHERE a.deptno ≠ (SELECT deptno FROM emp_borshch b WHERE b.empno =
a.mgr);
```



The screenshot shows the Beekeeper-Studio interface with the 'emp\_borshch' table selected. The table contains 20 rows of employee data. The columns are: empno (numeric), ename (varchar), job (varchar), mgr (numeric), hiredate (date), sal (numeric), comm (numeric), deptno (numeric), and dname (varchar). The data is as follows:

empno	ename	job	mgr	hiredate	sal	comm	deptno	dname
7369	SMITH@foo.com	CLERK	7902	2010-12-17	800.00	0.00	20	RESEARCH
7499	ALLEN@foo.com	SALESMAN	7698	2011-02-20	1600.00	300.00	30	SALES
7521	WARD@foo.com	SALESMAN	7698	2011-02-22	1250.00	500.00	30	SALES
7566	JONES@foo.com	MANAGER	7839	2011-04-02	2975.00	297.50	20	RESEARCH
7654	MARTIN@foo.com	SALESMAN	7698	2011-09-28	1250.00	1400.00	30	SALES
7698	BLAKE@foo.com	MANAGER	7839	2011-05-01	2850.00	285.00	30	SALES
7782	CLARK@foo.com	MANAGER	7839	2011-06-09	1500.00	0.00	10	ACCOUNTING
7788	SCOTT@foo.com	ANALYST	7566	2012-12-09	3000.00	0.00	20	RESEARCH
7839	KING@foo.com	PRESIDENT	(NULL)	2011-11-17	5000.00	0.00	10	ACCOUNTING
7844	TURNER@foo.com	SALESMAN	7698	2011-09-08	1500.00	0.00	30	SALES
7876	ADAMS@foo.com	CLERK	7788	2013-01-12	1100.00	0.00	20	RESEARCH
7900	JAMES@foo.com	CLERK	7698	2011-12-03	950.00	0.00	30	SALES
7902	FORD@foo.com	ANALYST	7566	2011-12-03	3000.00	0.00	20	RESEARCH
7934	MILLER@foo.com	CLERK	7782	2012-01-23	1300.00	0.00	10	ACCOUNTING
8000	JACKIE CHAN@bar.com	SALESMAN	7839	2011-09-28	2250.00	1870.00	60	SALES2
8001	JET Li@bar.com	SALESMAN	8000	2011-02-20	2600.00	600.00	60	SALES2
8002	BRUCE LEE@bar.com	SALESMAN	8000	2011-09-08	2500.00	0.00	60	SALES2
8003	DR NO@bar.com	ANALYST	7839	2011-09-11	2500.00	0.00	(NULL)	(NULL)
8004	BORSHCH@foo.com	TRAINEE	7839	2022-11-19	0.00	0.00	10	ACCOUNTING



## \* «Бонуси»

\*\* виконайте завдання із використанням merge

### 12. Завдання:

Створіть повторно emp\_username. Щороку співробітникам переглядають премії. Роблять це на основі їхнього статусу:

Silver - премія 25% зарплати

Gold - премія 50% зарплати

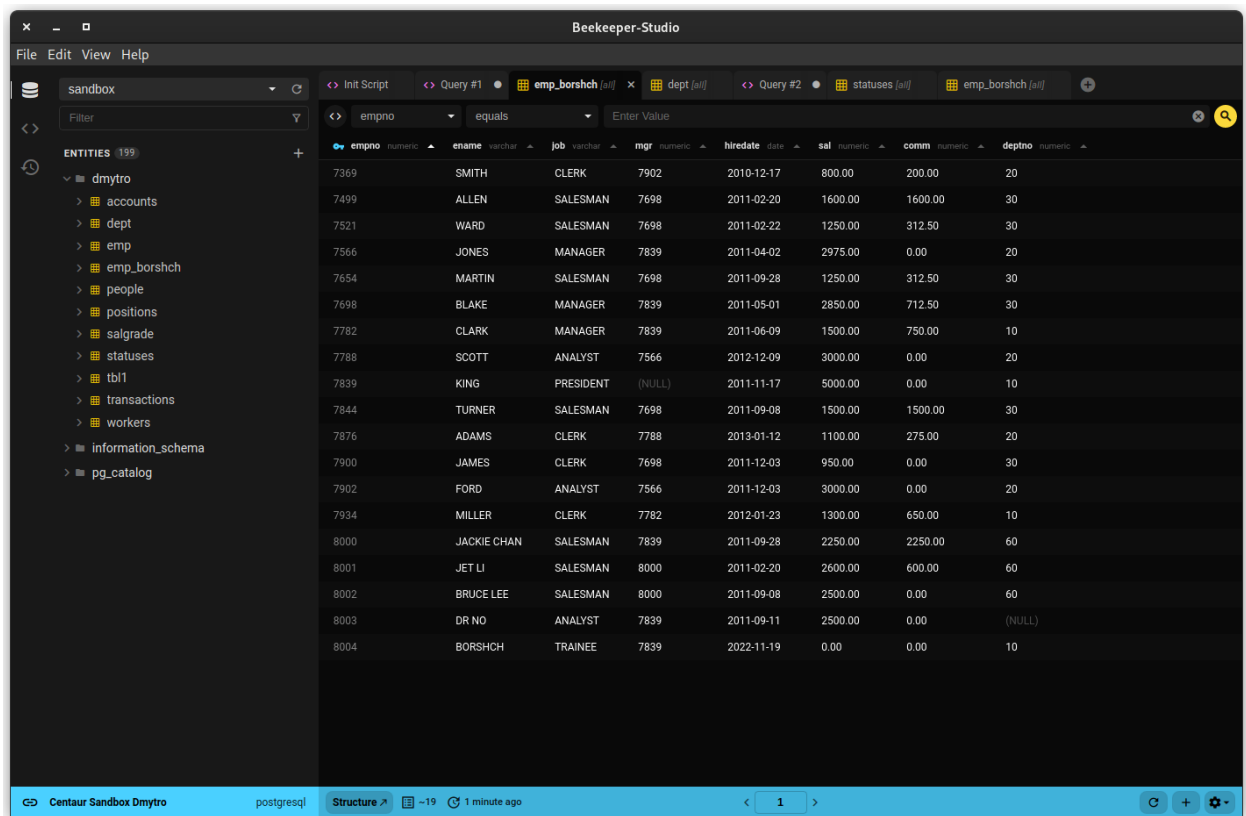
Platinum - премія 100% зарплати

Якщо людини немає у списку - їй премія не підвищується.

Таблиця із надбавками створюється сценарієм наведеним нижче.

Знижувати премію не можна: якщо в результаті премія буде нижчою - залишаємо стару премію. Виконайте оновлення одним запитом.

```
UPDATE emp_borshch SET comm = 0 WHERE comm IS NULL;  
MERGE INTO emp_borshch e  
USING statuses s  
ON e.empno = s.empno  
WHEN MATCHED THEN  
    UPDATE SET comm = CASE  
        WHEN s.status = 'Silver' AND (e.comm < e.sal*0.25) THEN  
            e.sal*0.25  
        WHEN s.status = 'Gold' AND (e.comm < e.sal*0.5) THEN  
            e.sal*0.5  
        WHEN s.status = 'Platinum' AND (e.comm < e.sal) THEN  
            e.sal  
        END  
WHEN NOT MATCHED THEN  
    DO NOTHING  
;
```



empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	2010-12-17	800.00	200.00	20
7499	ALLEN	SALESMAN	7698	2011-02-20	1600.00	1600.00	30
7521	WARD	SALESMAN	7698	2011-02-22	1250.00	312.50	30
7566	JONES	MANAGER	7839	2011-04-02	2975.00	0.00	20
7654	MARTIN	SALESMAN	7698	2011-09-28	1250.00	312.50	30
7698	BLAKE	MANAGER	7839	2011-05-01	2850.00	712.50	30
7782	CLARK	MANAGER	7839	2011-06-09	1500.00	750.00	10
7788	SCOTT	ANALYST	7566	2012-12-09	3000.00	0.00	20
7839	KING	PRESIDENT	(NULL)	2011-11-17	5000.00	0.00	10
7844	TURNER	SALESMAN	7698	2011-09-08	1500.00	1500.00	30
7876	ADAMS	CLERK	7788	2013-01-12	1100.00	275.00	20
7900	JAMES	CLERK	7698	2011-12-03	950.00	0.00	30
7902	FORD	ANALYST	7566	2011-12-03	3000.00	0.00	20
7934	MILLER	CLERK	7782	2012-01-23	1300.00	650.00	10
8000	JACKIE CHAN	SALESMAN	7839	2011-09-28	2250.00	2250.00	60
8001	JET LI	SALESMAN	8000	2011-02-20	2600.00	600.00	60
8002	BRUCE LEE	SALESMAN	8000	2011-09-08	2500.00	0.00	60
8003	DR NO	ANALYST	7839	2011-09-11	2500.00	0.00	(NULL)
8004	BORSHCH	TRAINEE	7839	2022-11-19	0.00	0.00	10

## «Приєднання»

13. \*\* У передній лаб.роботі був файл "union.sql" - з новими співробітниками. Додайте їх у таблицю emp одним запитом. Усі вони працюють у новому відділі (створіть його завчасно). Керівник - KING, посада «BA»

```
INSERT INTO emp (ename, job, mgr, hiredate, sal, comm, deptno) (SELECT
    (first_name || ' ' || second_name) as ename,
    'BA' as job,
    7839 as mgr,
    current_date as hiredate,
    salary as sal,
    0 as comm,
    101 as deptno
FROM villain_inc
);
```

Beekeeper-Studio

File Edit View Help

sandbox

Filter

ENTITIES 200

- dmytro
  - accounts
  - dept
  - emp
    - empno int4
    - ename varchar(40)
    - job varchar(9)
    - mgr int4
    - hiredate date(0)
    - sal int4
    - comm int4
    - deptno int4
  - emp\_borshch
  - people
  - positions
  - salgrade
  - statuses
  - tbl1
  - transactions
  - villain\_inc
  - workers
- information\_schema
- pg\_catalog

empno equals Enter Value

empno	ename	job	mgr	hiredate	sal	comm	deptno
13	Le Chiffre	BA	7839	2022-11-19	10000	0	101
14	Buonaparte Ign...	BA	7839	2022-11-19	10000	0	101
15	Hugo von der D...	BA	7839	2022-11-19	7000	0	101
16	Jack Spang	BA	7839	2022-11-19	8000	0	101
17	Seraffimo Spang	BA	7839	2022-11-19	3000	0	101
18	Rosa Klebb	BA	7839	2022-11-19	4000	0	101
19	Auric Goldfinger	BA	7839	2022-11-19	5000	0	101
20	Colonel von Ha...	BA	7839	2022-11-19	3000	0	101
21	Hector Gonzales	BA	7839	2022-11-19	2000	0	101
22	Aristotle Krista...	BA	7839	2022-11-19	3000	0	101
23	Milton Krest	BA	7839	2022-11-19	5200	0	101
24	Emilio Largo	BA	7839	2022-11-19	0	0	101
7369	SMITH	CLERK	7902	2010-12-17	800	(NULL)	20
7499	ALLEN	SALESMAN	7698	2011-02-20	1600	300	30
7521	WARD	SALESMAN	7698	2011-02-22	1250	500	30
7566	JONES	MANAGER	7839	2011-04-02	2975	(NULL)	20
7654	MARTIN	SALESMAN	7698	2011-09-28	1250	1400	30
7698	BLAKE	MANAGER	7839	2011-05-01	2850	(NULL)	30
7782	CLARK	MANAGER	7839	2011-06-09	1500	(NULL)	10
7788	SCOTT	ANALYST	7566	2012-12-09	3000	(NULL)	20
7839	KING	PRESIDENT	(NULL)	2011-11-17	5000	(NULL)	10
7844	TURNER	SALESMAN	7698	2011-09-08	1500	0	30
7876	ADAMS	CLERK	7788	2013-01-12	1100	(NULL)	20
7900	JAMES	CLERK	7698	2011-12-03	950	(NULL)	30

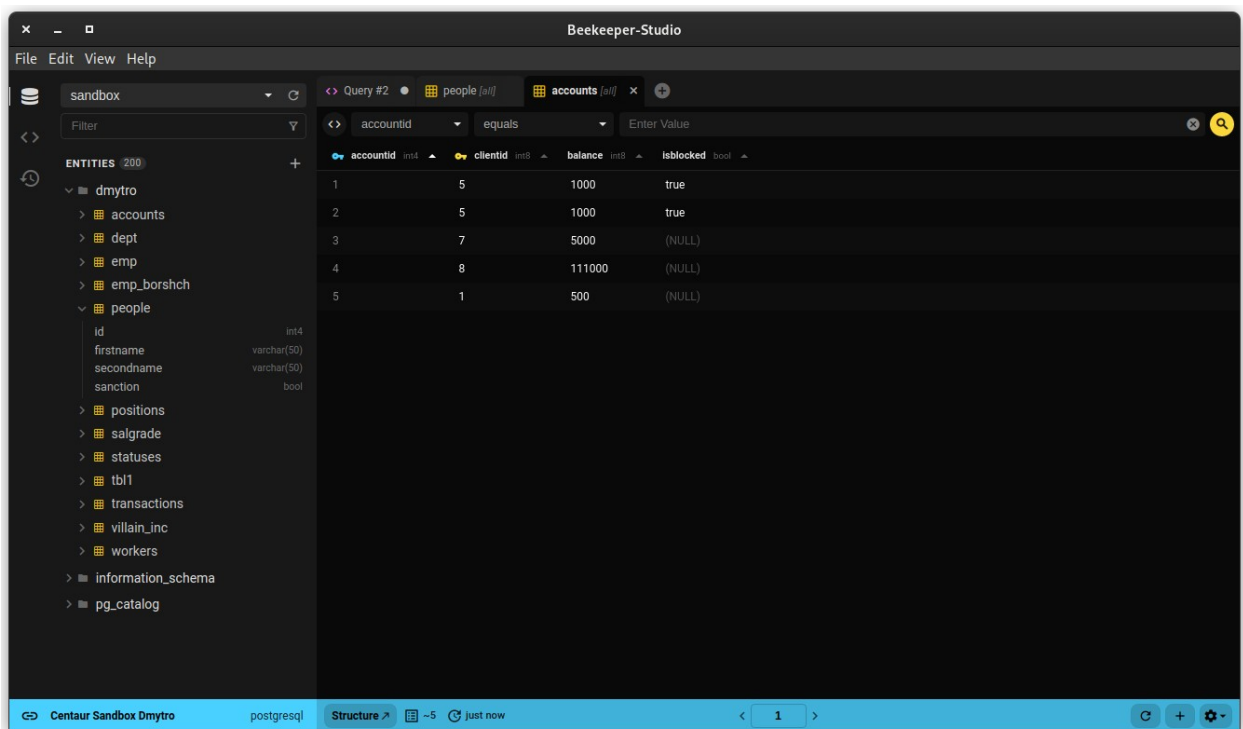
Centaur Sandbox Dmytro postgresql Structure ~31 just now 1

# Завдання за варіантами

## Варіант 1

1. Фізичні можуть потрапляти під санкції. Це треба зберігати у окремому атрибуті таблиці (додайте його). У рахунках додайте атрибут, чи є рахунок заблокованим ( із значенням по замовченню «ні»). Заблокуйте рахунки осіб, що є під санкціями. (Вважаємо що їх перелік невідомий заздалегідь та може бути дуже довгим).

```
UPDATE accounts SET isBlocked = True WHERE clientID IN (SELECT id FROM people WHERE sanction = True);
```

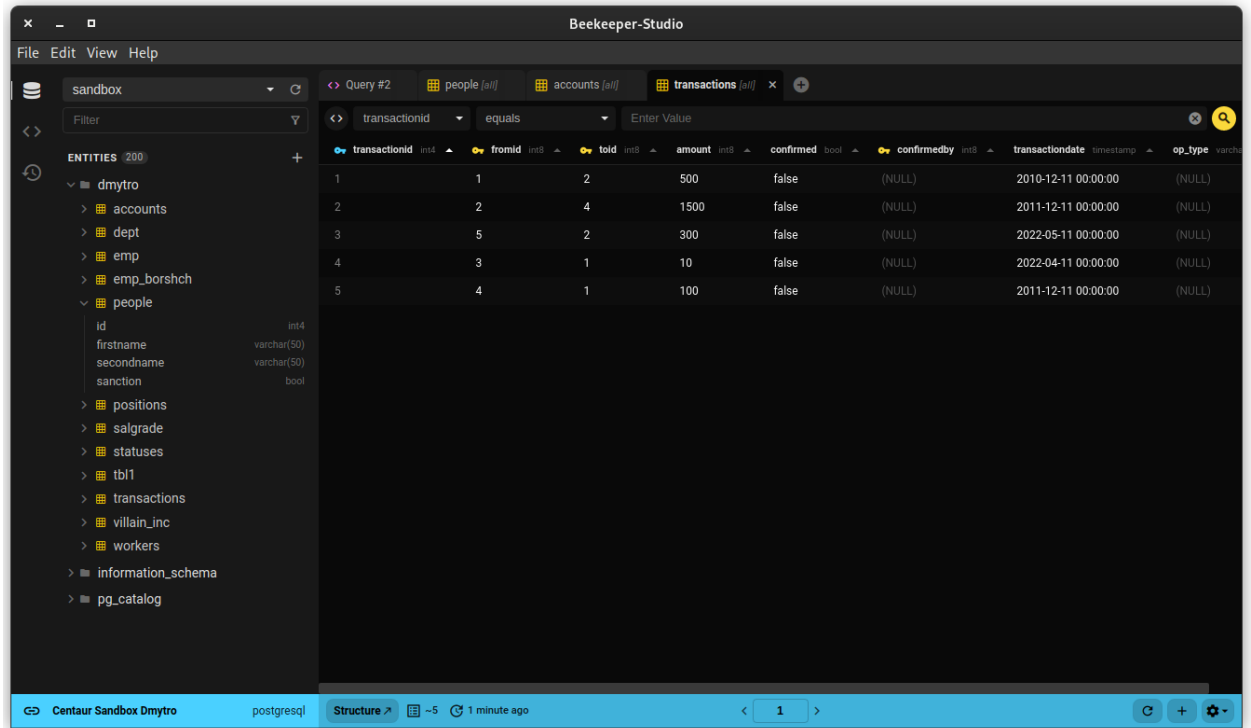


2. \*\* Юридичні особи теж можуть потрапляти під санкції. При цьому під санкції потрапляють і всі афільовані ( пов'язані) з ними фізичні особи.

■

3. Оновіть таблицю із операціями додавши до неї колонку `op_type` (тип операції). Заповніть її значеннями «income» та «outcome» для зарахувань та зняття коштів, відповідно. Для «\*» - одним запитом.

```
ALTER TABLE transactions ADD op_type VARCHAR;
```

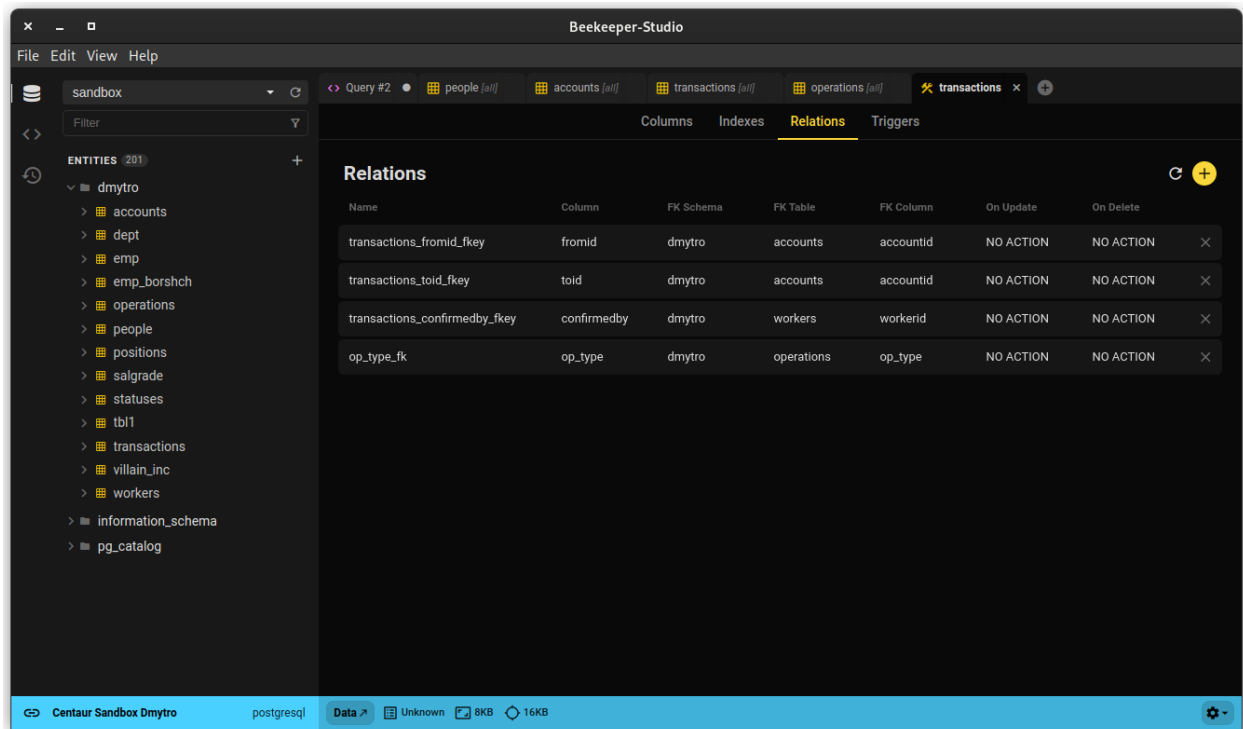


The screenshot shows the Beekeeper-Studio interface. On the left, a sidebar lists database entities under 'sandbox', including 'dmytro' with sub-entities like 'accounts', 'dept', 'emp', 'emp\_borshch', 'people', 'positions', 'salgrade', 'statuses', 'tbl1', 'transactions', 'villain\_inc', 'workers', 'information\_schema', and 'pg\_catalog'. The 'people' entity is expanded, showing fields: 'id' (int4), 'firstname' (varchar(50)), 'secondname' (varchar(50)), and 'sanction' (bool). The main panel displays 'Query #2' with a table view of the 'transactions' table. The table has columns: 'transactionid' (int4), 'fromid' (int8), 'toid' (int8), 'amount' (int8), 'confirmed' (bool), 'confirmedby' (int8), 'transactiondate' (timestamp), and 'op\_type' (varchar). The data shows 5 transactions, all with 'confirmed' status 'false' and 'confirmedby' as '(NULL)'. The 'op\_type' column is currently empty for all rows.

transactionid	fromid	toid	amount	confirmed	confirmedby	transactiondate	op_type
1	1	2	500	false	(NULL)	2010-12-11 00:00:00	(NULL)
2	2	4	1500	false	(NULL)	2011-12-11 00:00:00	(NULL)
3	5	2	300	false	(NULL)	2022-05-11 00:00:00	(NULL)
4	3	1	10	false	(NULL)	2022-04-11 00:00:00	(NULL)
5	4	1	100	false	(NULL)	2011-12-11 00:00:00	(NULL)

4. \* Зробіть так, щоб БД гарантувала, що можливими типами операції є виключно «income», «outcome» та «bonus».

```
CREATE TABLE operations(  
    op_type varchar UNIQUE  
);  
ALTER TABLE transactions ADD CONSTRAINT op_type_fk FOREIGN KEY  
(op_type) REFERENCES operations (op_type);  
INSERT INTO operations VALUES ('income');  
INSERT INTO operations VALUES ('outcome');  
INSERT INTO operations VALUES ('bonus');
```



5. \*\* Зробіть так, щоб гарантувати, що сума грошей вказана в операції для «income»- позитивна, «outcome»- негативна, для «bonus» – відмінна від 0.

-