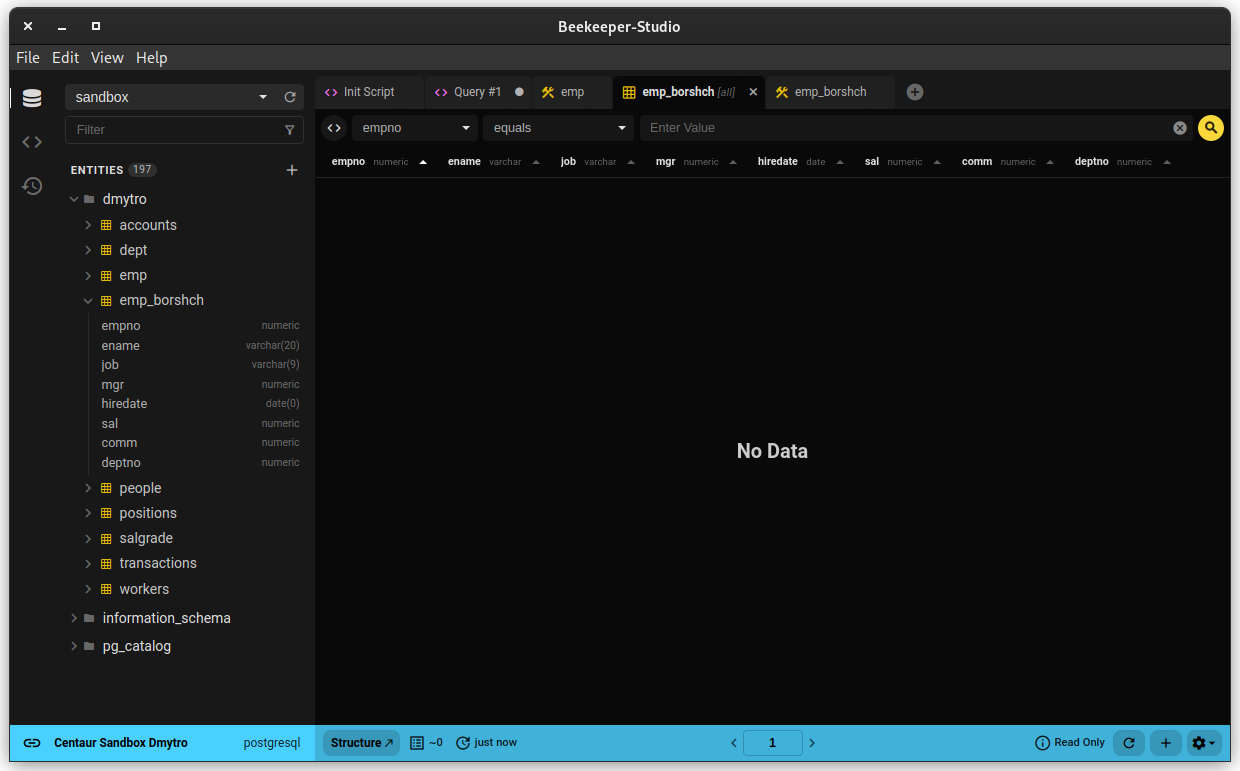
Лабораторна робота 6 «DML»  
⚠️ Для завдань/розділів відмічених «📷» – скріншот – обов’язково. Для інших – за домовленістю із викладачем.  
Завдання лабораторної роботи виконую всі

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\_yourname 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\_yourname із застосуванням Truncate. Чи вийде те саме із dept?

TRUNCATE TABLE emp\_borshch;

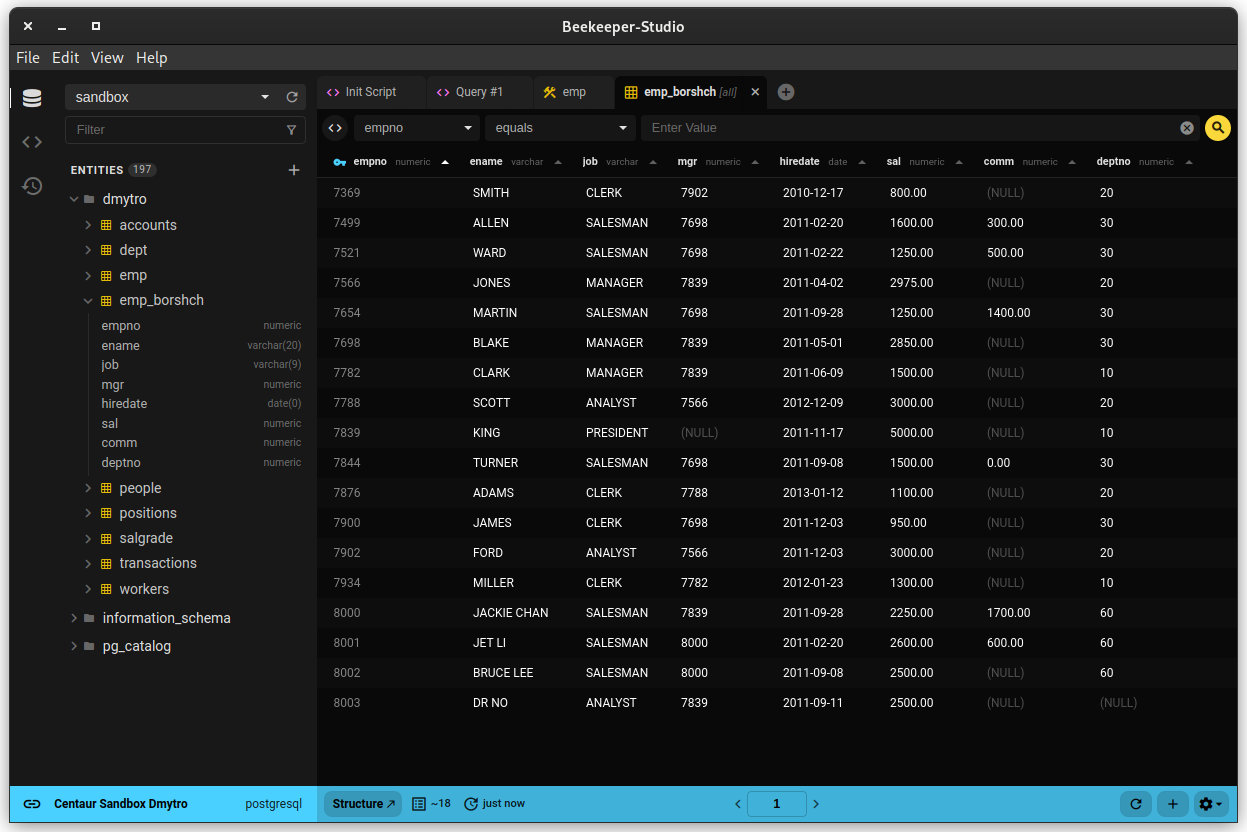
4. Переконайтеся, що таблиця emp\_yourname порожня.

SELECT \* FROM emp\_borshch

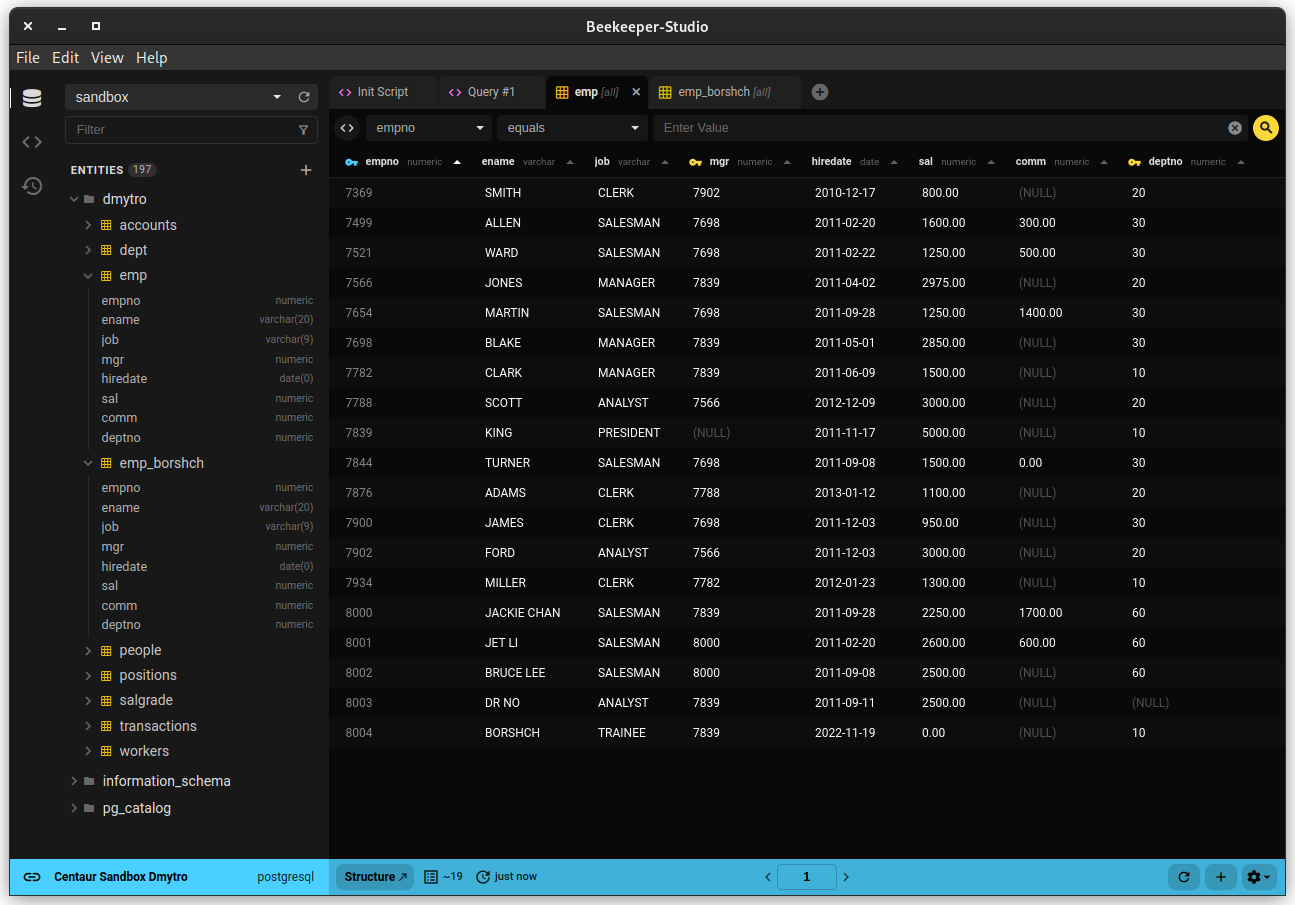
5. Вставте всі дані з таблиці Emp у таблицю emp\_yourname.

INSERT INTO emp\_borshch SELECT \* FROM emp;

6. 📷 Переконайтеся, що всі співробітники додані.



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)



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\_yourname. 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

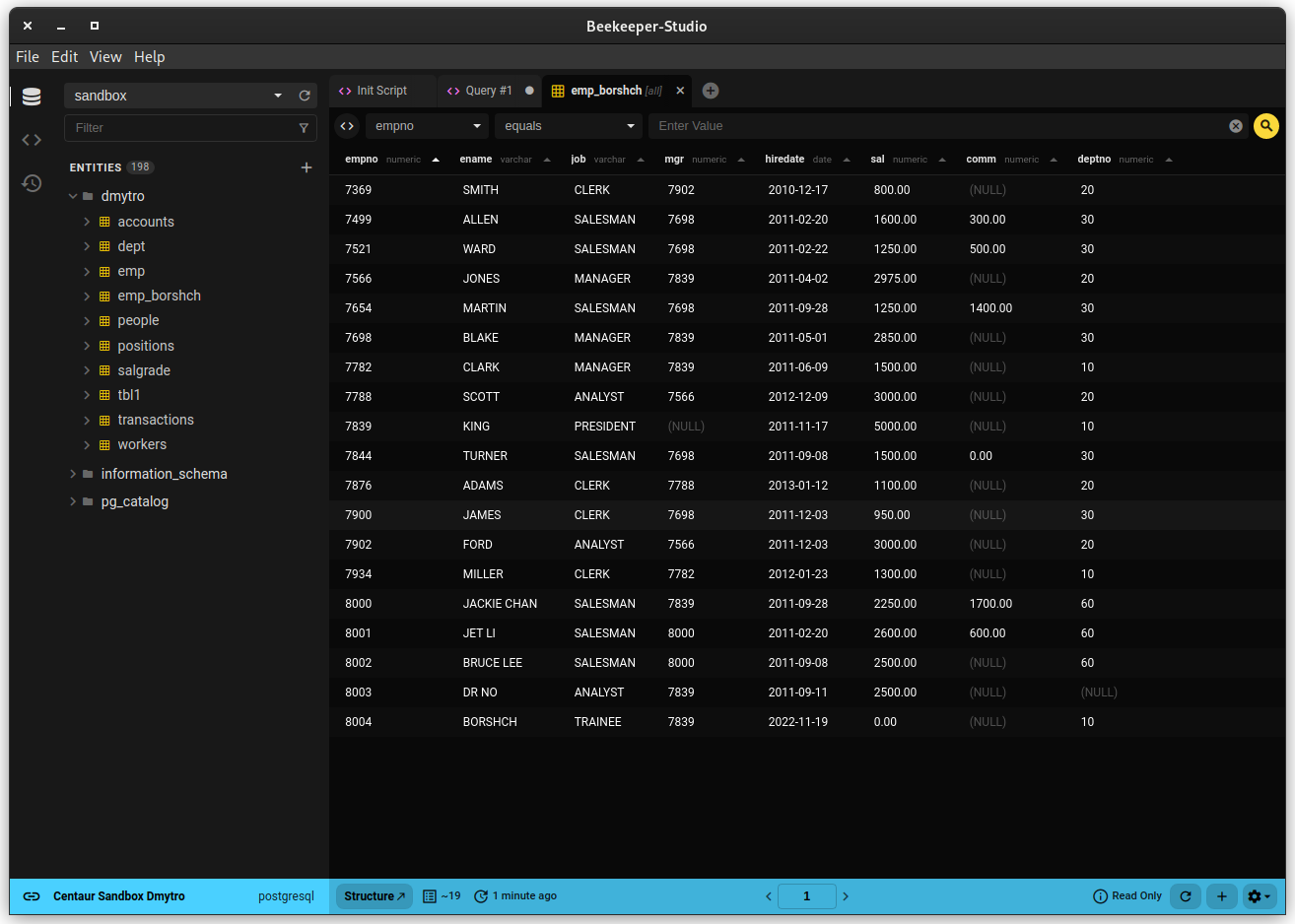
Insert int tbl1(b) values (null);

значення: null, null  
Insert int tbl1 values ();

в postgres отримав помилку синтаксису

Корельовані підзапити в UPDATE І DELETE 📷  
7. Створіть таблицю emp\_yourname на основі таблиці EMP та скопіюйте у неї всі дані.

CREATE TABLE emp\_borshch AS SELECT \* FROM emp;



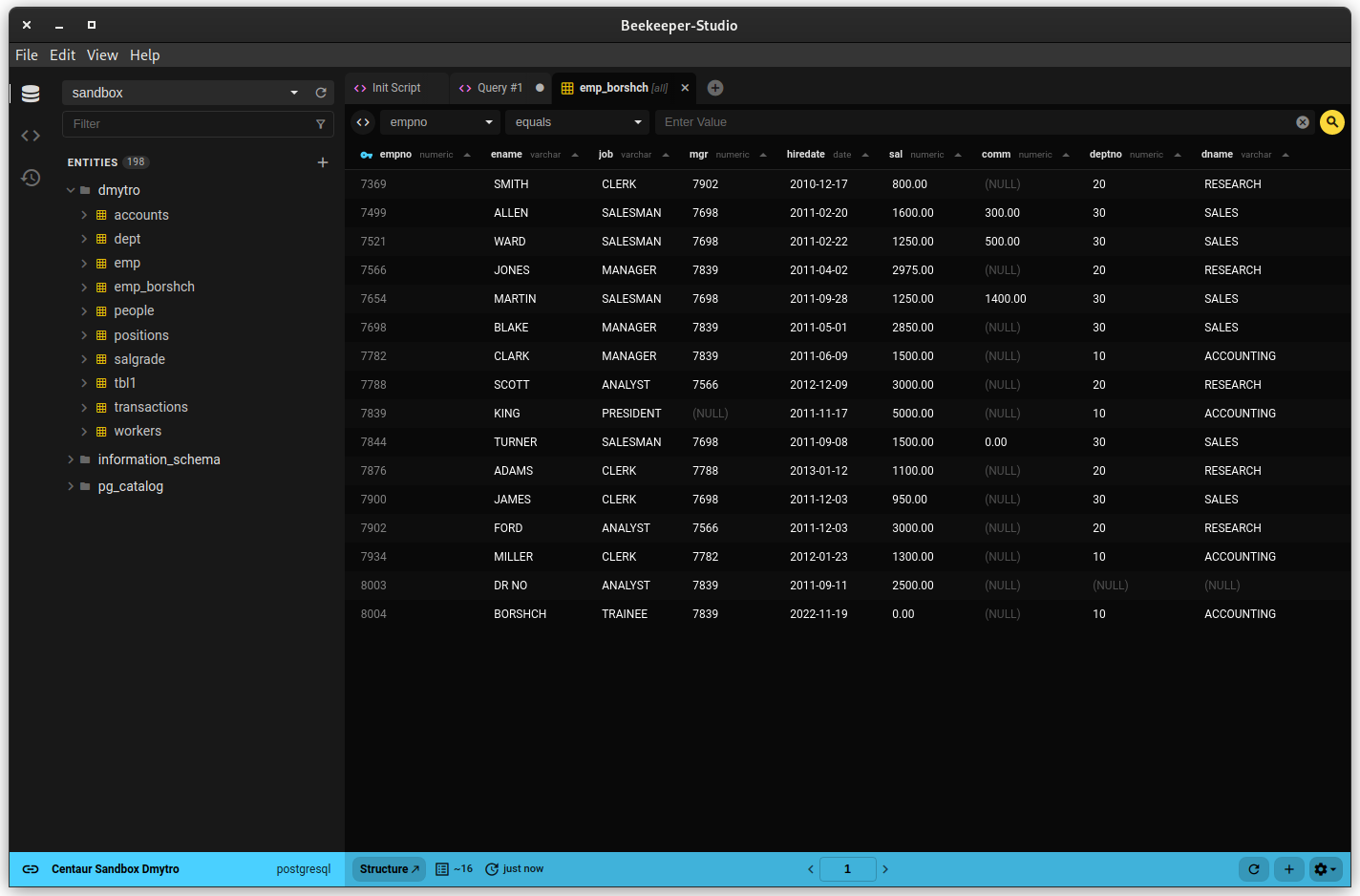
8. Денормалізуйте таблицю emp\_yourname додавши до неї стовпець dname. Заповніть цей стовпець на основі даних таблиці DEPT.

ALTER TABLE emp\_borshch ADD DNAME VARCHAR;

UPDATE emp\_borshch SET dname = (SELECT dname FROM dept WHERE dept.deptno = emp\_borshch.deptno);

9. \* Видаліть відомості з таблиці emp\_yourname про службовців, які працюють у HONKONG

DELETE FROM emp\_borshch WHERE dname in (SELECT dname FROM dept WHERE loc = 'HONKONG');



10. \* Ми хочемо додати до email співробітника назву домену. Наприклад, «JohnDoe»→«JohnDoe@foo.com». Оновіть данні у колонці email, так щоб співробітники із USA (New York, Boston, Dallas, ... ) мали домен «foo.com», а всі інщі «bar.com». Вважаємо що кількість співробітників у EMP велика і на перед 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

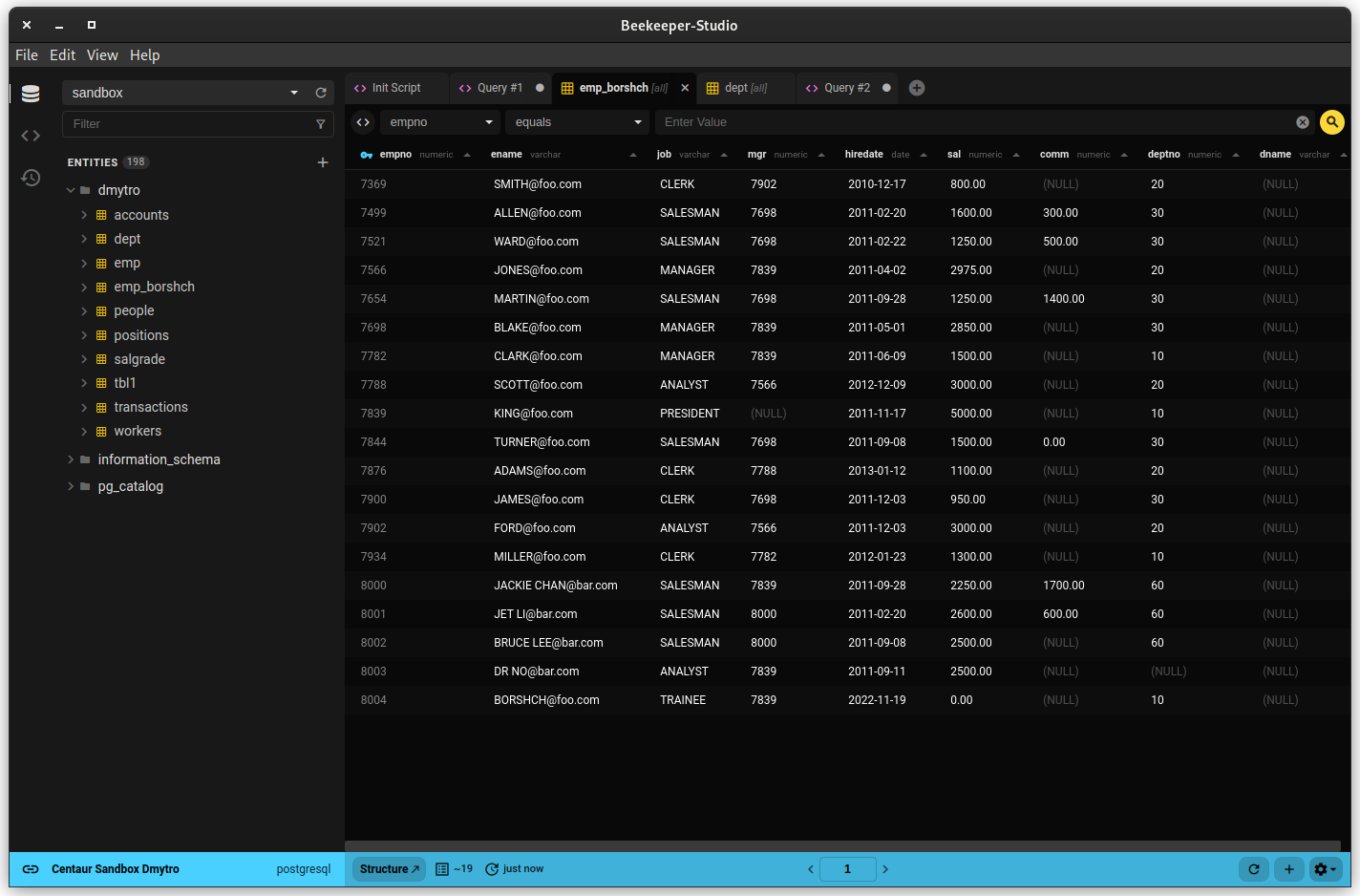
(ename || '@foo.com')

ELSE

(ename || '@bar.com')

END

;



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

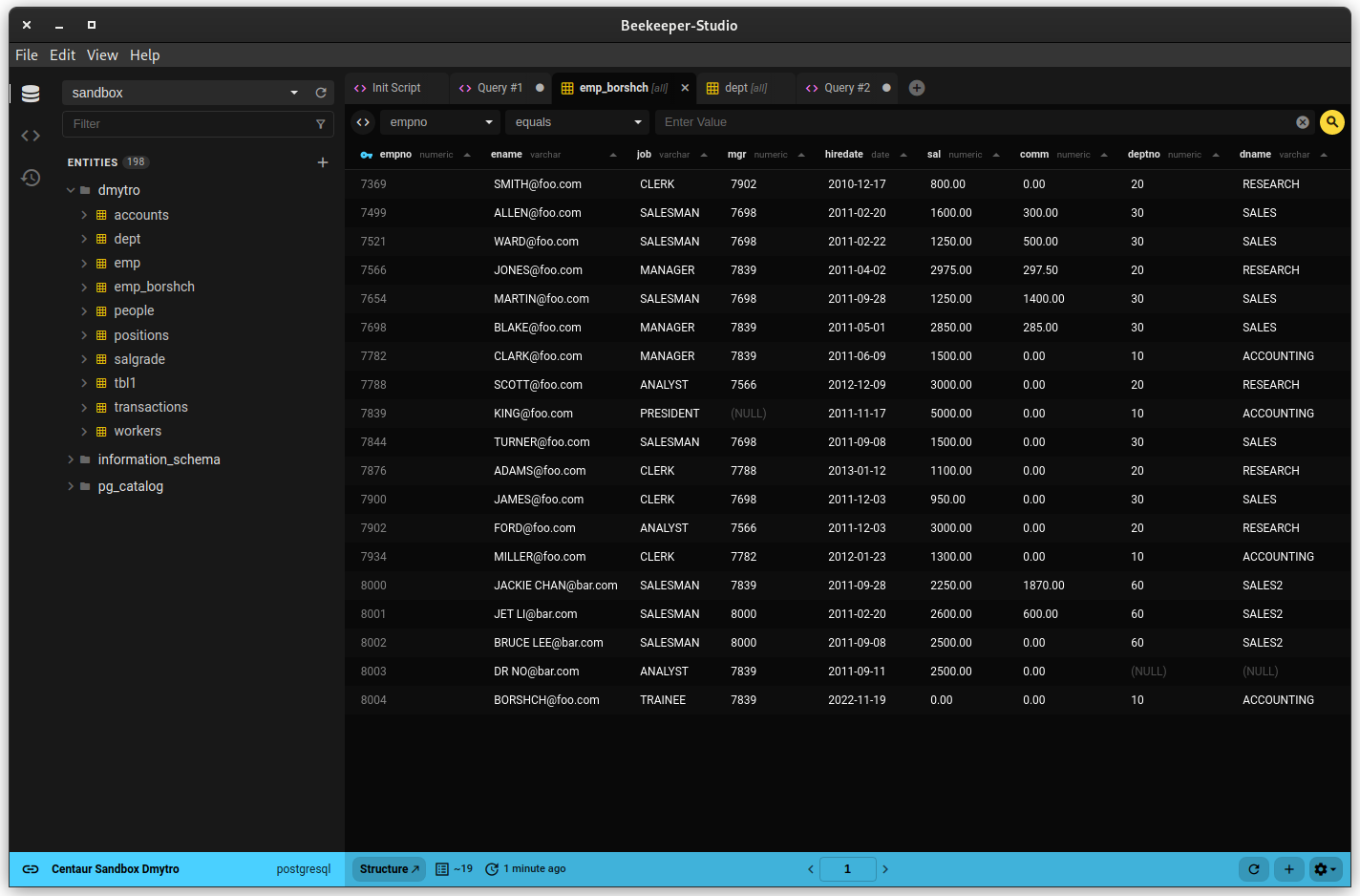
comm + comm \* 0.1

ELSE

sal \* 0.1

END

WHERE a.deptno != (SELECT deptno FROM emp\_borshch b WHERE b.empno = a.mgr);



\* «Бонуси» 📷  
\*\* виконайте завдання із використанням merge  
12. Завдання:  
Створіть повторно emp\_yourname. 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

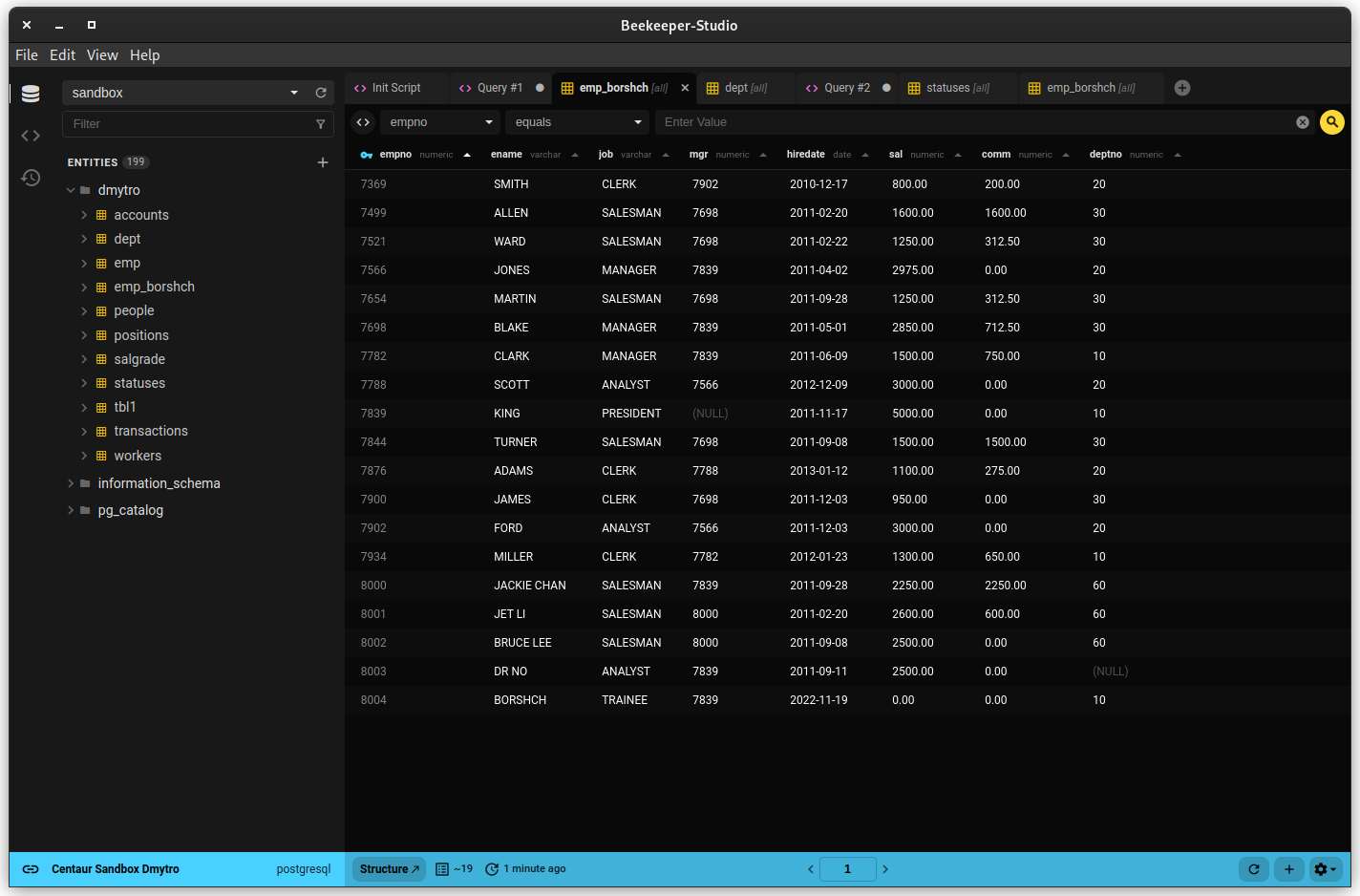
e.sal

END

WHEN NOT MATCHED THEN

DO NOTHING

;

  
«Приєднання»  
13. \*\* У переднії 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,

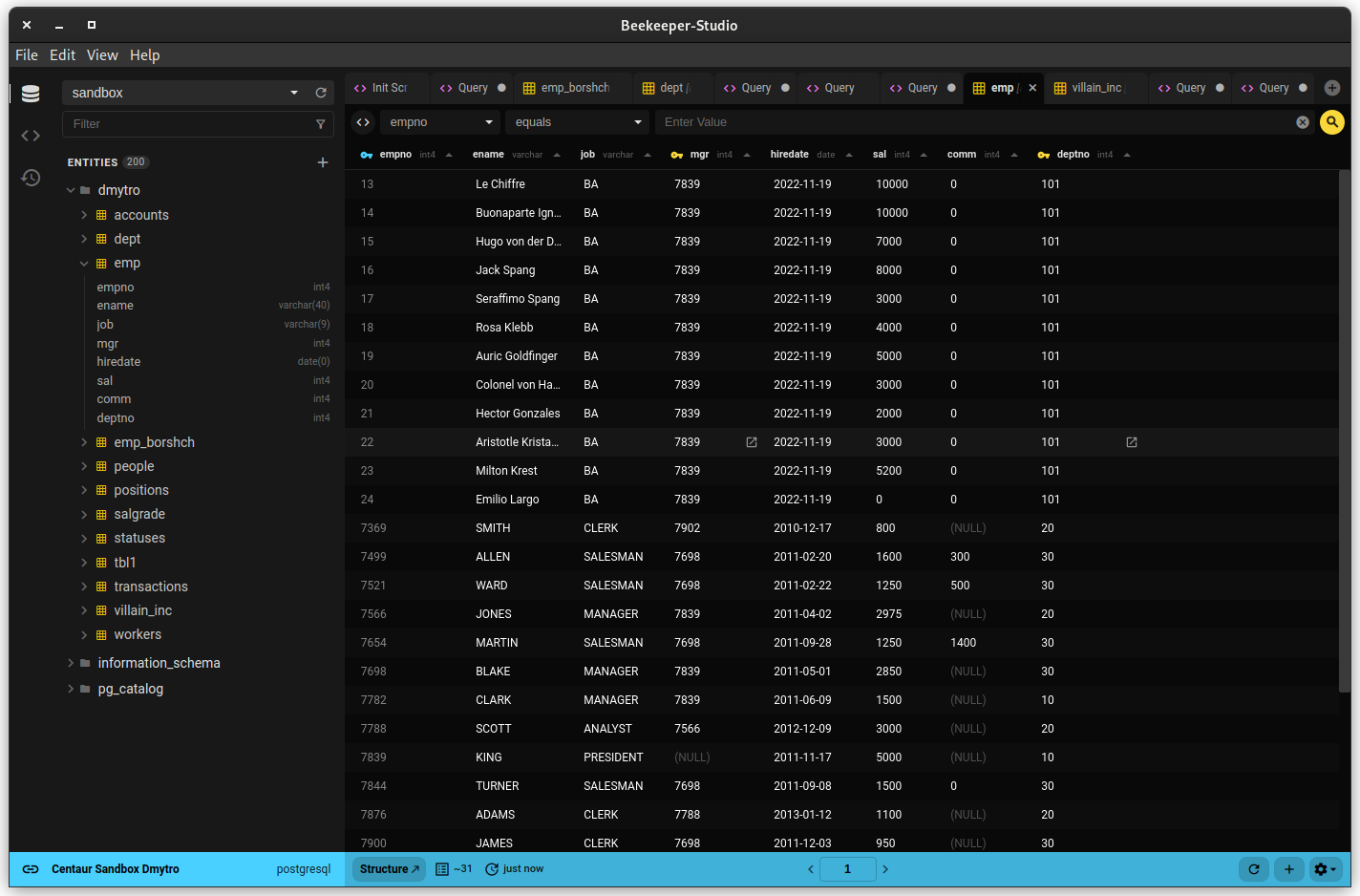
0 as comm,

101 as deptno

FROM villain\_inc

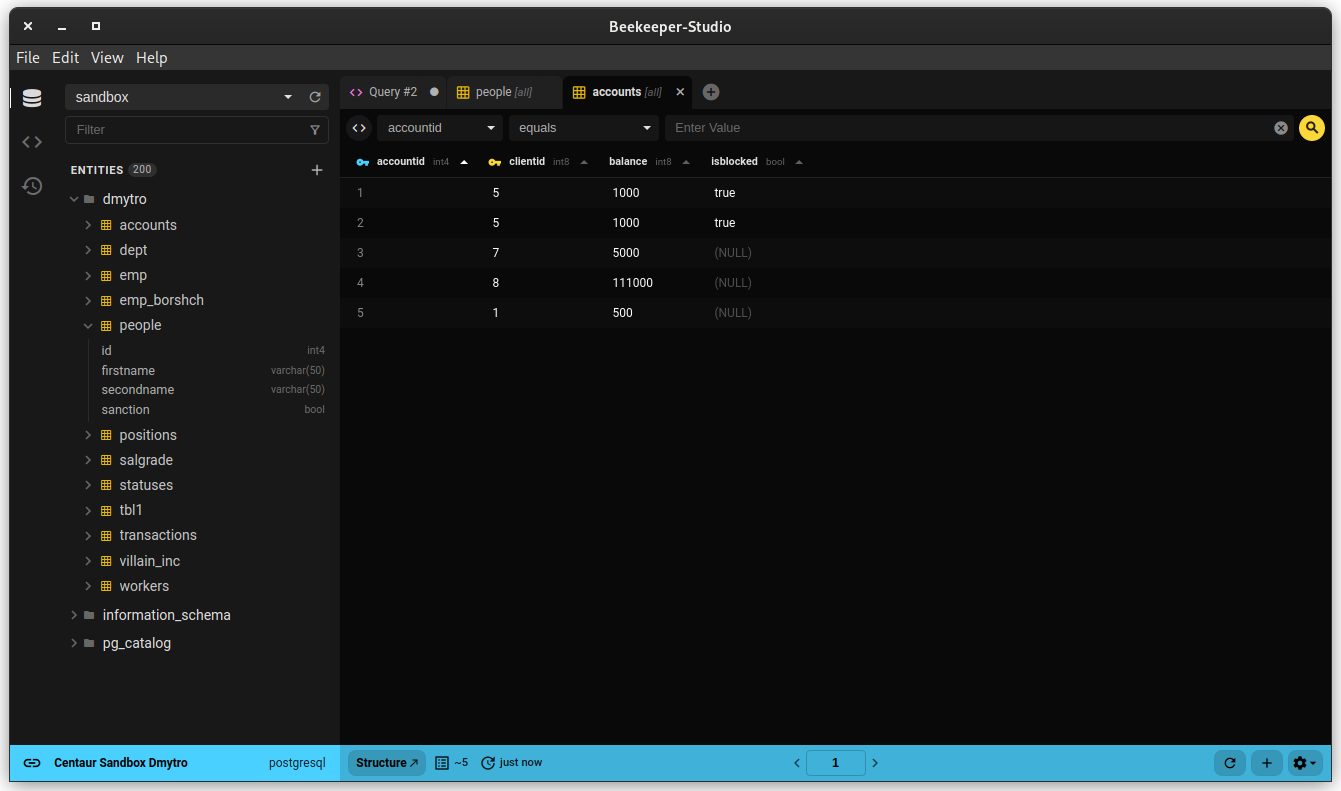
)

;



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);

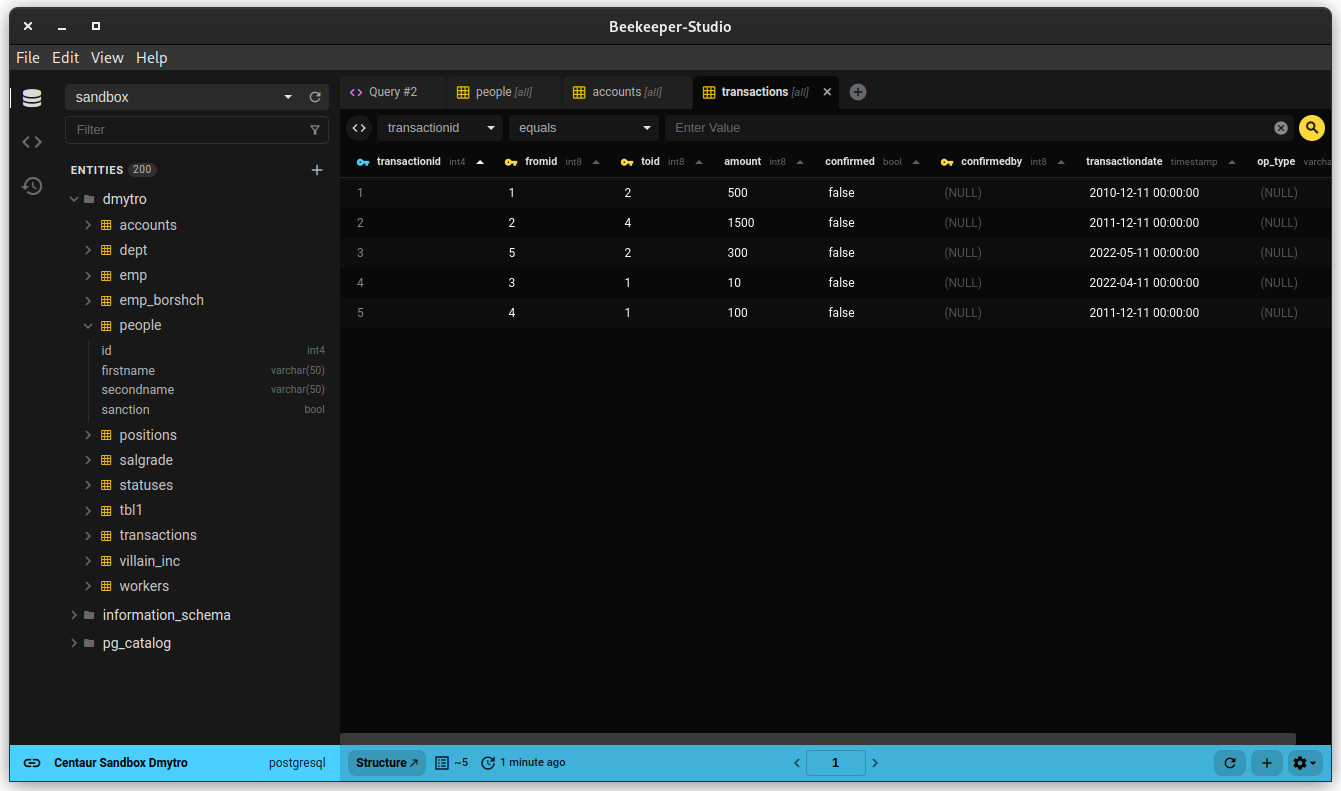


2. \*\* Юридичні особи теж можуть потрапляти під санкції. При цьому під санкції потрапляють і всі афільовані ( пов’язані) з ними фізичні особи.

-

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

ALTER TABLE transactions ADD op\_type VARCHAR;



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

CREATE TABLE operations(

op\_type varchar UNIQUE

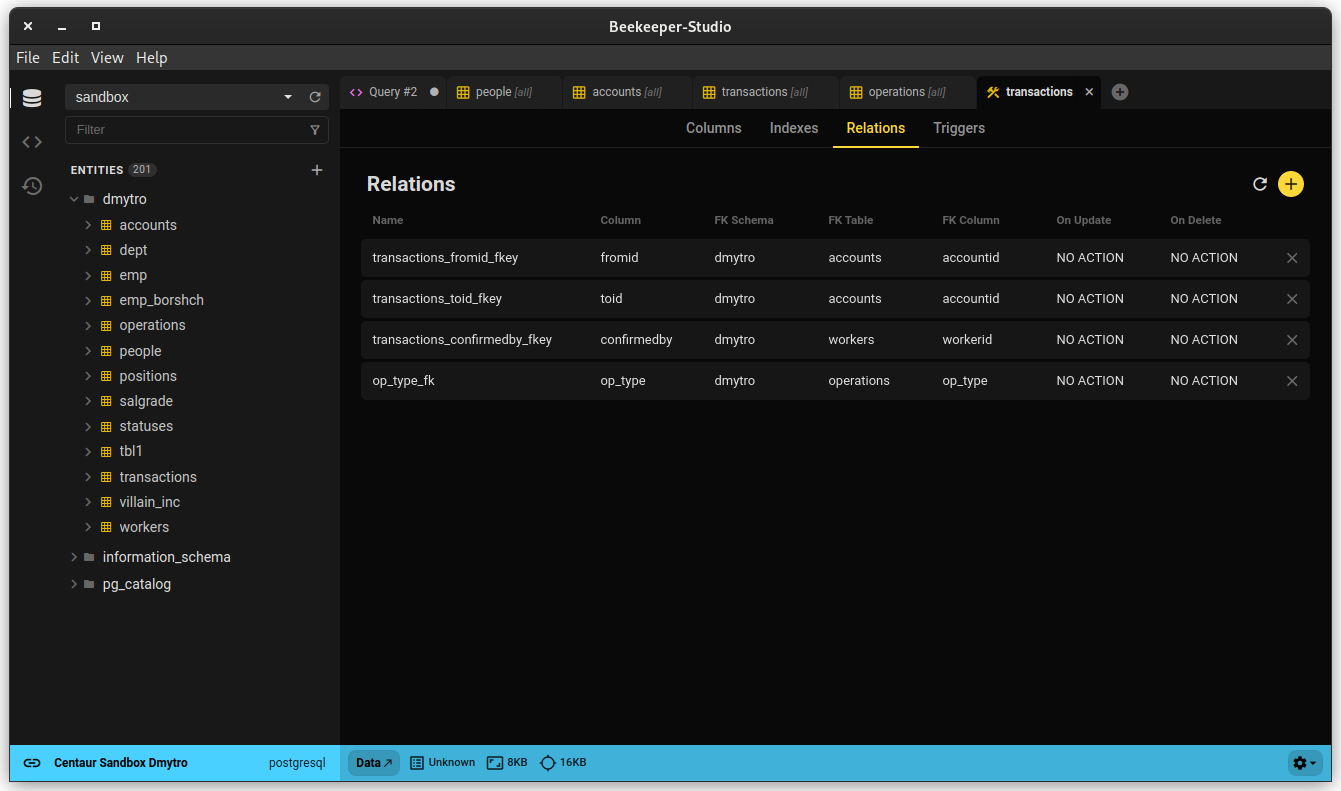
);

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.

-