МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Челябинский государственный университет»**

**(ФГБОУ ВО «ЧелГУ»)**

Институт информационных технологий

**ЛАБОРАТОРНАЯ РАБОТА**

**по дисциплине: Базы и хранилища данных**

**Лабораторная работа SQL №5 «Операторы DDL и DML»**

Выполнил: Зинатов А.А.

Группа: ПИ-201

Руководитель: Барабанщиков И.В.

Челябинск 2023

1. Подключиться к БД tysql (см. Рис. 1).

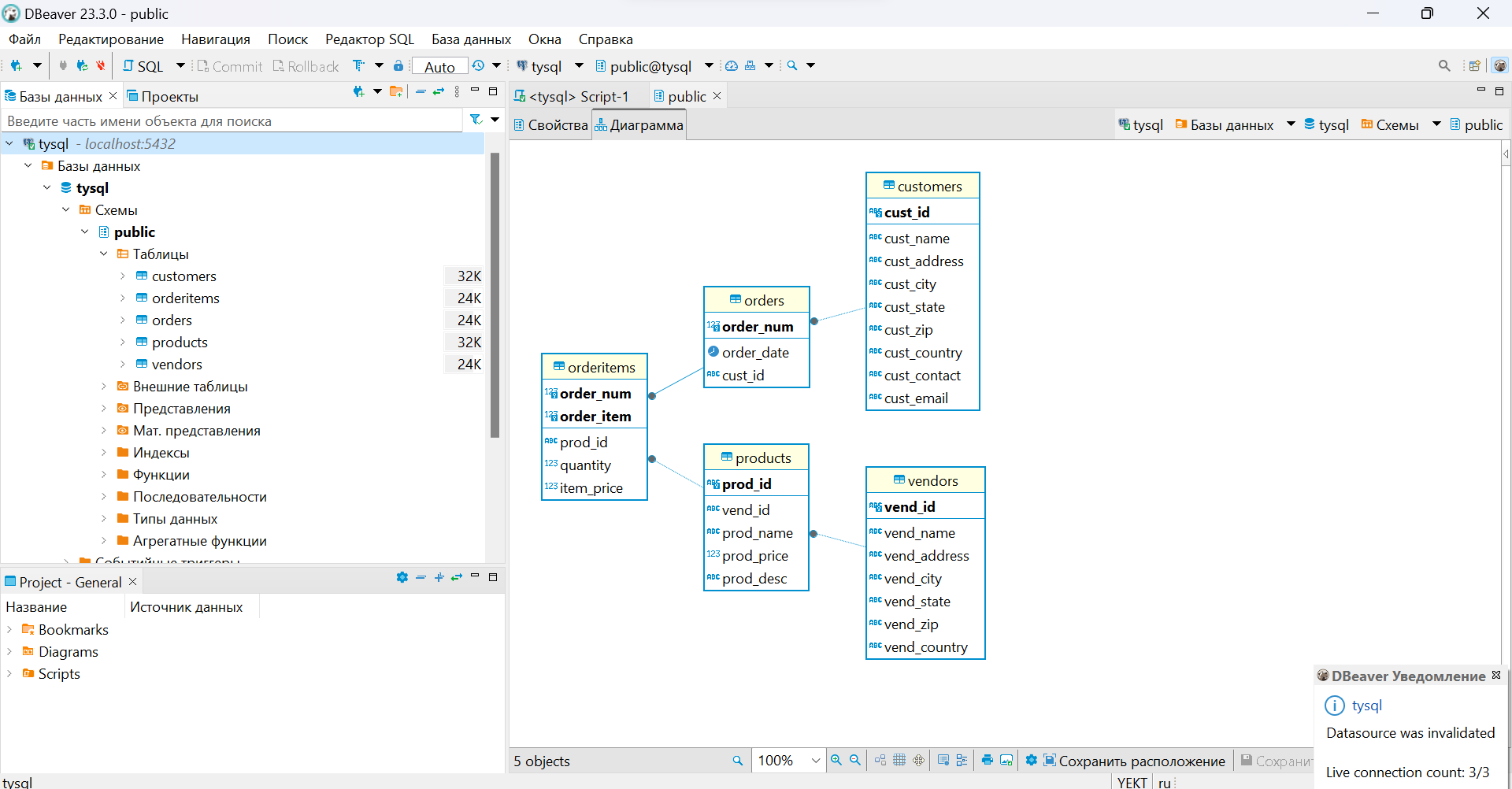


Рис. 1

1. По нижеприведенному описанию создать 2 таблицы с помощью команд CREATE TABLE (имена для таблицы и ее колонок выбрать самостоятельно) (см. Рис. 2.1; Рис. 2.2):

Таблица «Отдел»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя колонки | Тип | Обязатель-ность | Проверка (check) | Знач. по умолч | Уникаль-  ность | Ключ |
| Код отдела | Целое число | Да |  |  | Да | PK |
| Имя отдела | Текст | Да |  |  | Да |  |

Таблица «Отдел»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя колонки | Тип | Обязатель-ность | Проверка (check) | Знач. по умолч | Уникаль-  ность | Ключ |
| Код сотрудника | Целое число | Да |  |  | Да | PK |
| Код отдела | Целое число | Да |  |  | Нет | FK |
| Фамилия | Текст | Да |  |  | Нет |  |
| Имя | Текст | Да |  |  | Нет |  |
| Возраст, лет | Целое число | Да | > 18 |  | Нет |  |
| Телефон | Текст | Нет |  |  | Да |  |
| Оклад | Целое число | Да |  | 1000 | Нет |  |

create table if not exists Departments

(

dep\_id INT NOT NULL UNIQUE,

dep\_name VARCHAR(255) NOT NULL UNIQUE,

PRIMARY KEY (dep\_id)

);

create table if not exists Staff

(

emp\_id INT NOT NULL UNIQUE,

dep\_id INT NOT NULL,

emp\_surname VARCHAR(255) NOT NULL,

emp\_name VARCHAR(255) NOT NULL,

emp\_age INT NOT NULL CHECK (emp\_age > 18),

emp\_phone\_number VARCHAR(30) UNIQUE,

emp\_sal INT NOT NULL DEFAULT 1000,

PRIMARY KEY (emp\_id),

FOREIGN KEY (dep\_id) REFERENCES Departments(dep\_id)

);

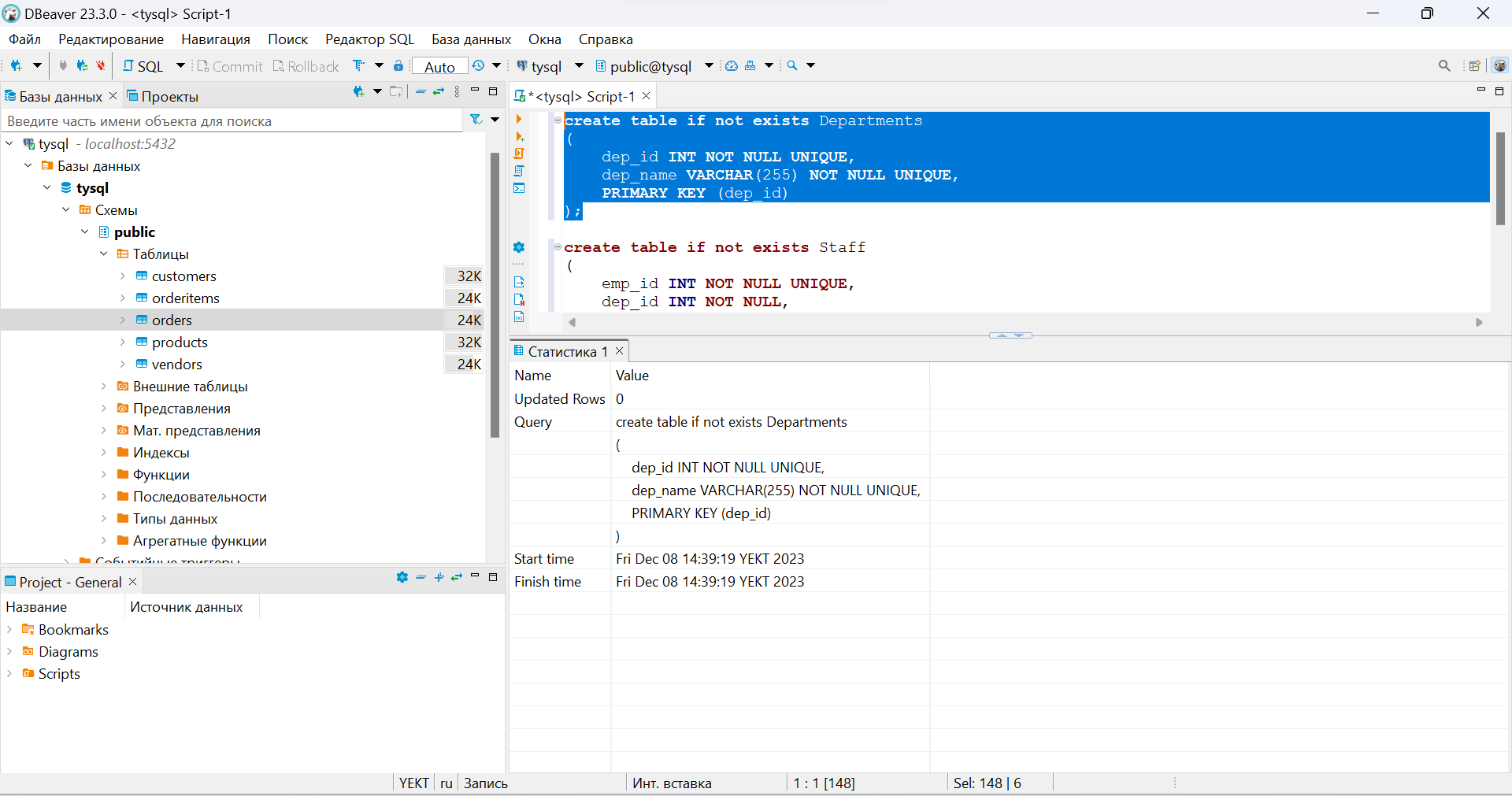


Рис. 2.1

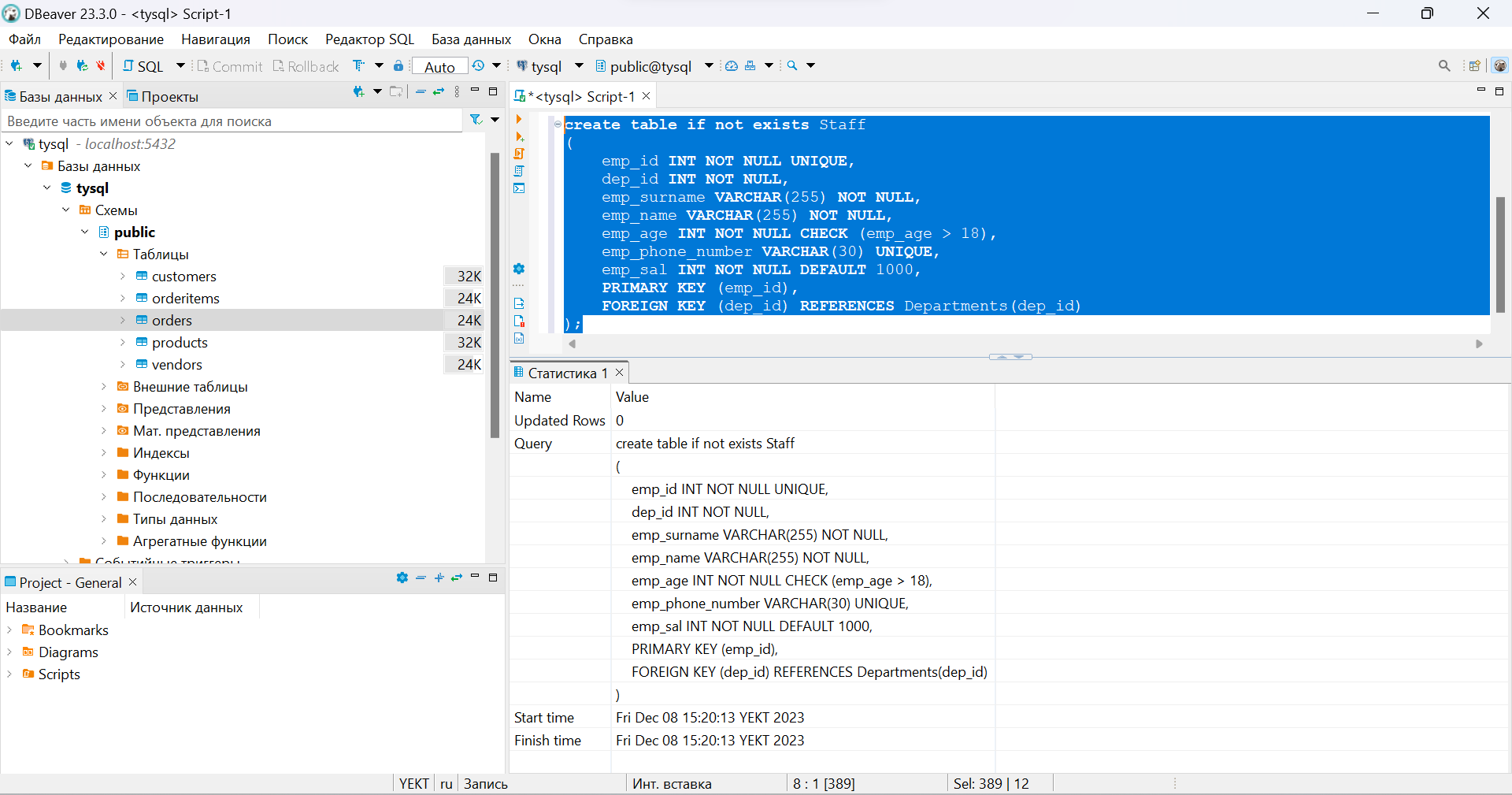


Рис. 2.2

1. В ставить данные в таблицы, выполнив команду INSERT (см. Рис. 3.1; Рис. 3.2):

* Вставить 3 записи в таблицу отделов (имена отделов: «Бухгалтерия», «ИТ», «АХО»)
* Вставить 5 записей в таблицу «Отдел» (из них 2 в Бухгалтерию, 3 в ИТ, 0 в АХО.

INSERT INTO departments (dep\_id, dep\_name)

VALUES (1, 'Бухгалтерия'),

(2, 'ИТ'),

(3, 'АХО');

INSERT INTO staff (emp\_id, dep\_id, emp\_surname, emp\_name, emp\_age, emp\_phone\_number, emp\_sal)

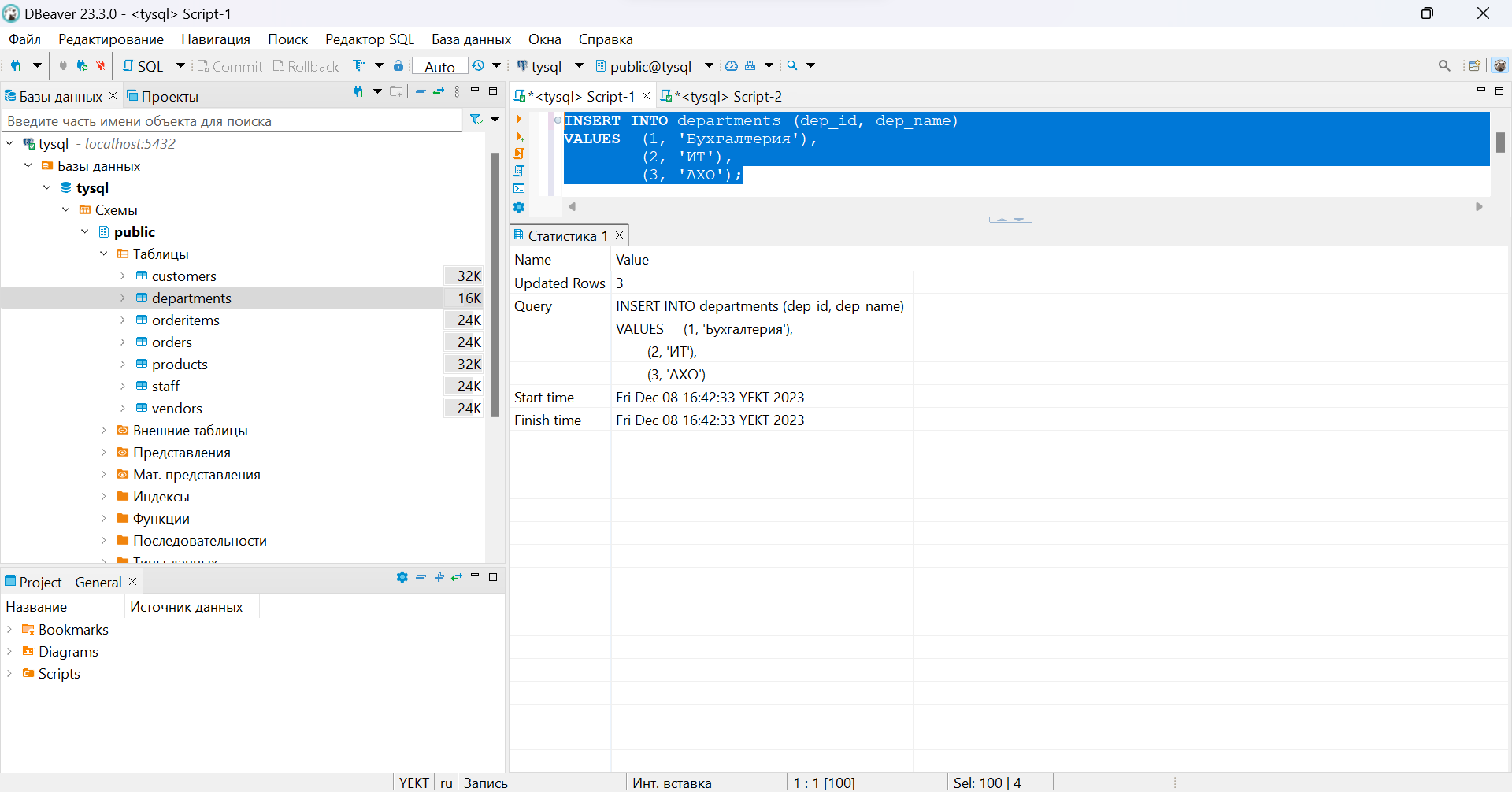
VALUES (1, 1, 'Иванов', 'Иван', 20, +79085001111, 10000),

(2, 1, 'Александрийский', 'Александр', 21, +79085001112, 20000),

(3, 2, 'Петрович', 'Пётр', 22, +79085001114, 30000),

(4, 2, 'Лопатов', 'Шаляпин', 23, +79085001115, 40000),

(5, 2, 'Грозный', 'Иван', 493, +79085001116, 9000000);

Рис. 3.1

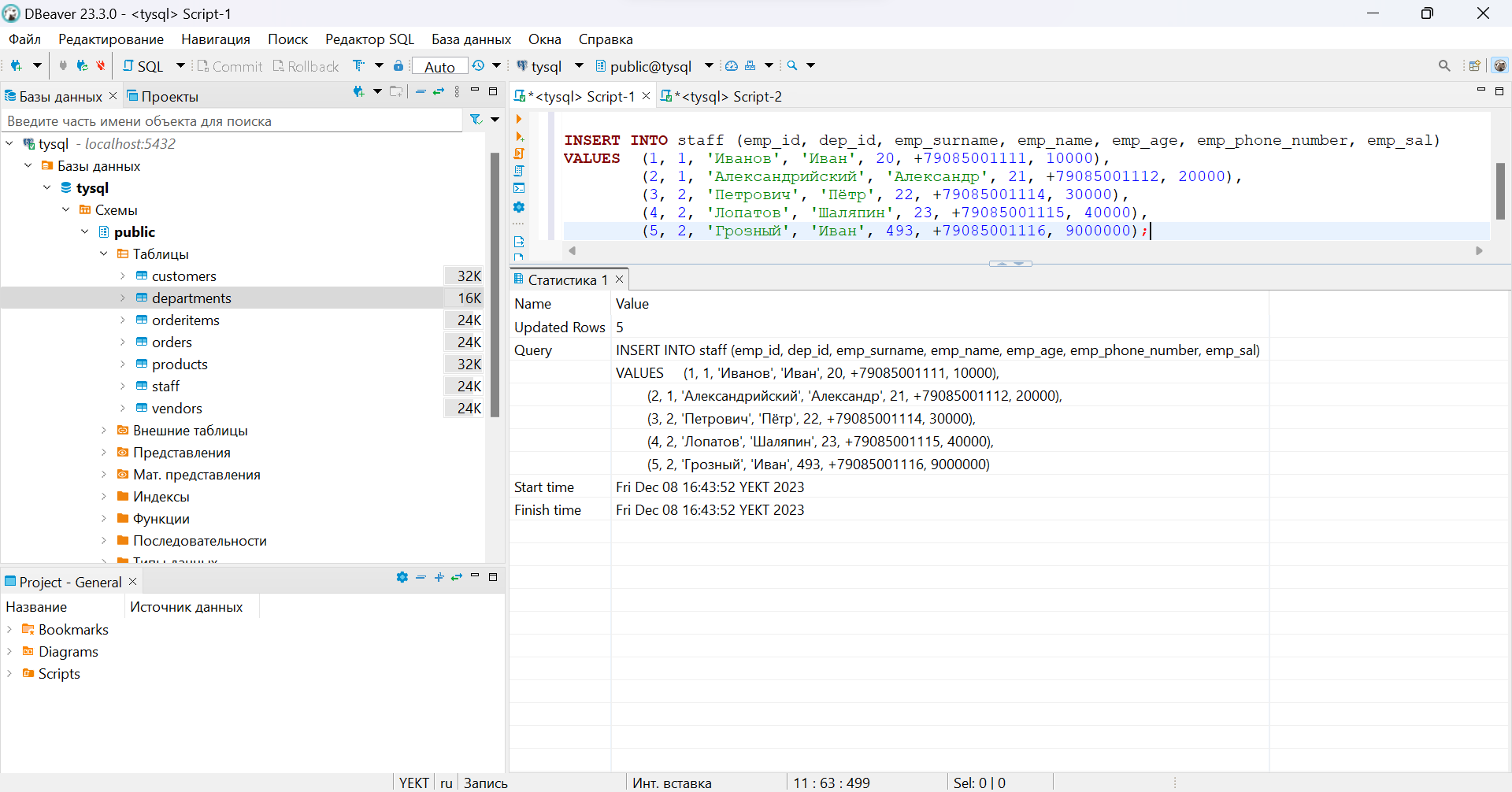


Рис. 3.2

1. Проверить результат вставки, выполнив команду SELECT (см. Рис. 4.1; Рис. 4.2).
2. select \* from departments d (Рис. 4.1)
3. select \* from staff s (Рис. 4.2)

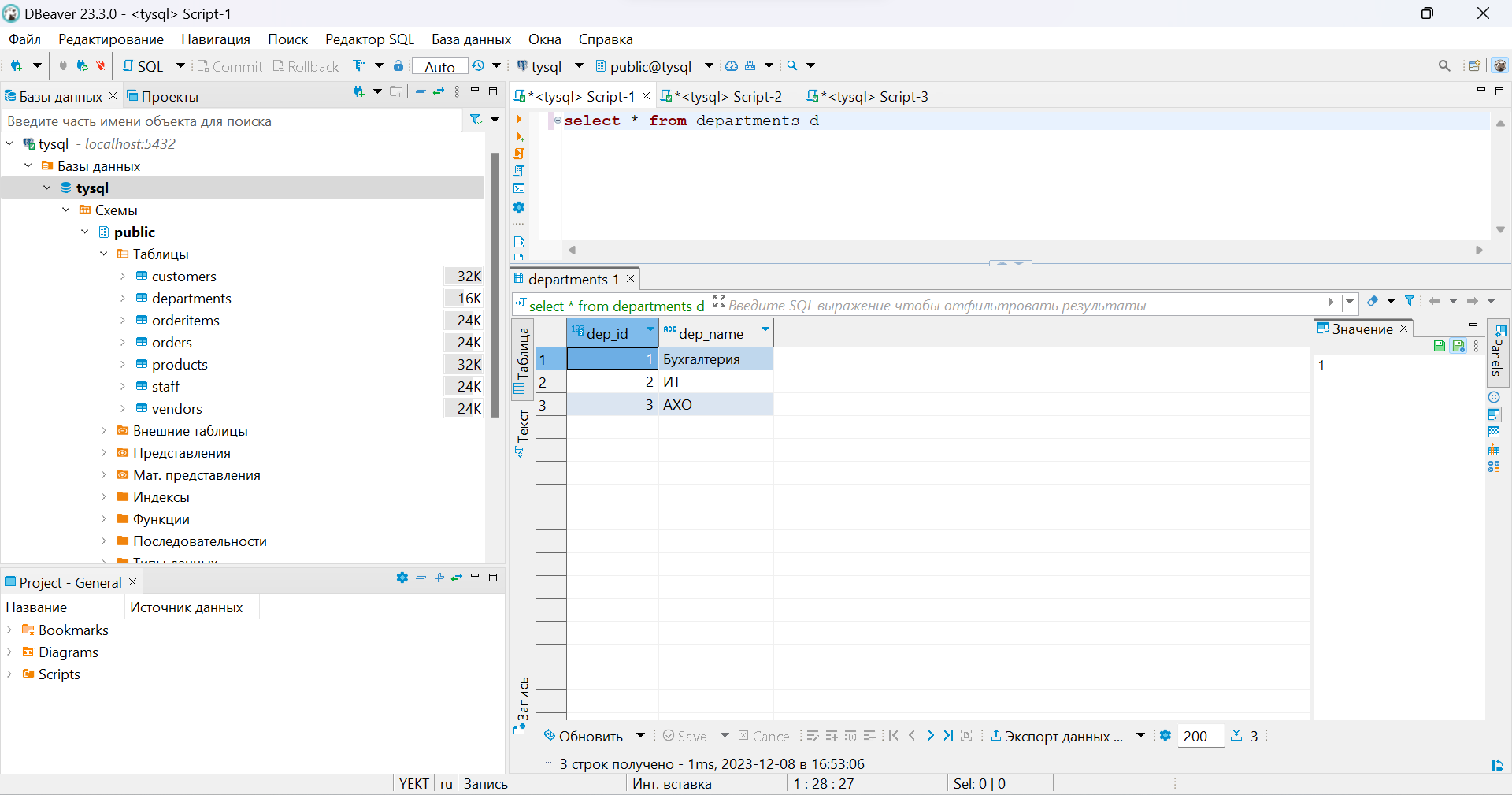


Рис. 4.1

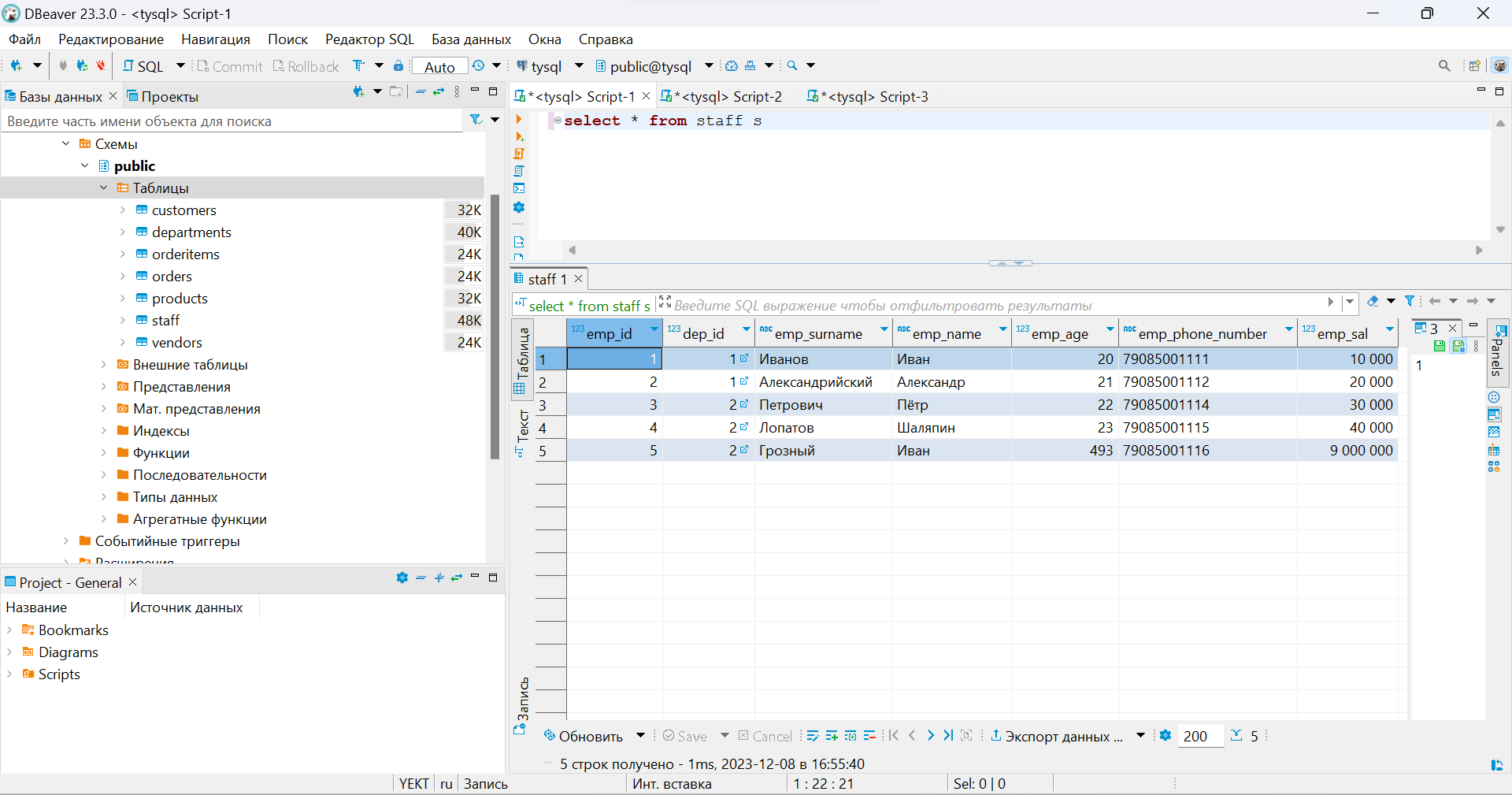


Рис. 4.2

1. Изменить данные в таблице, выполнив команду UPDATE (см. Рис. 5.1; Рис. 5.2; Рис. 5.3):

* Изменить оклады сотрудников ИТ, увеличив их в 2 раза.
* Изменить телефон любого сотрудника, сделав его таким, какой уже есть у другого сотрудника (нарушение уникальности)
* Изменить возраст сотрудников отдела Бухгалтерия, установив его в 16 лет.

UPDATE staff

SET emp\_sal = emp\_sal \* 2

WHERE staff.dep\_id = 2;

UPDATE staff

SET emp\_phone\_number = 79085001116

WHERE staff.emp\_id = 1;

UPDATE staff

SET emp\_age = 16

WHERE staff.dep\_id = 1;

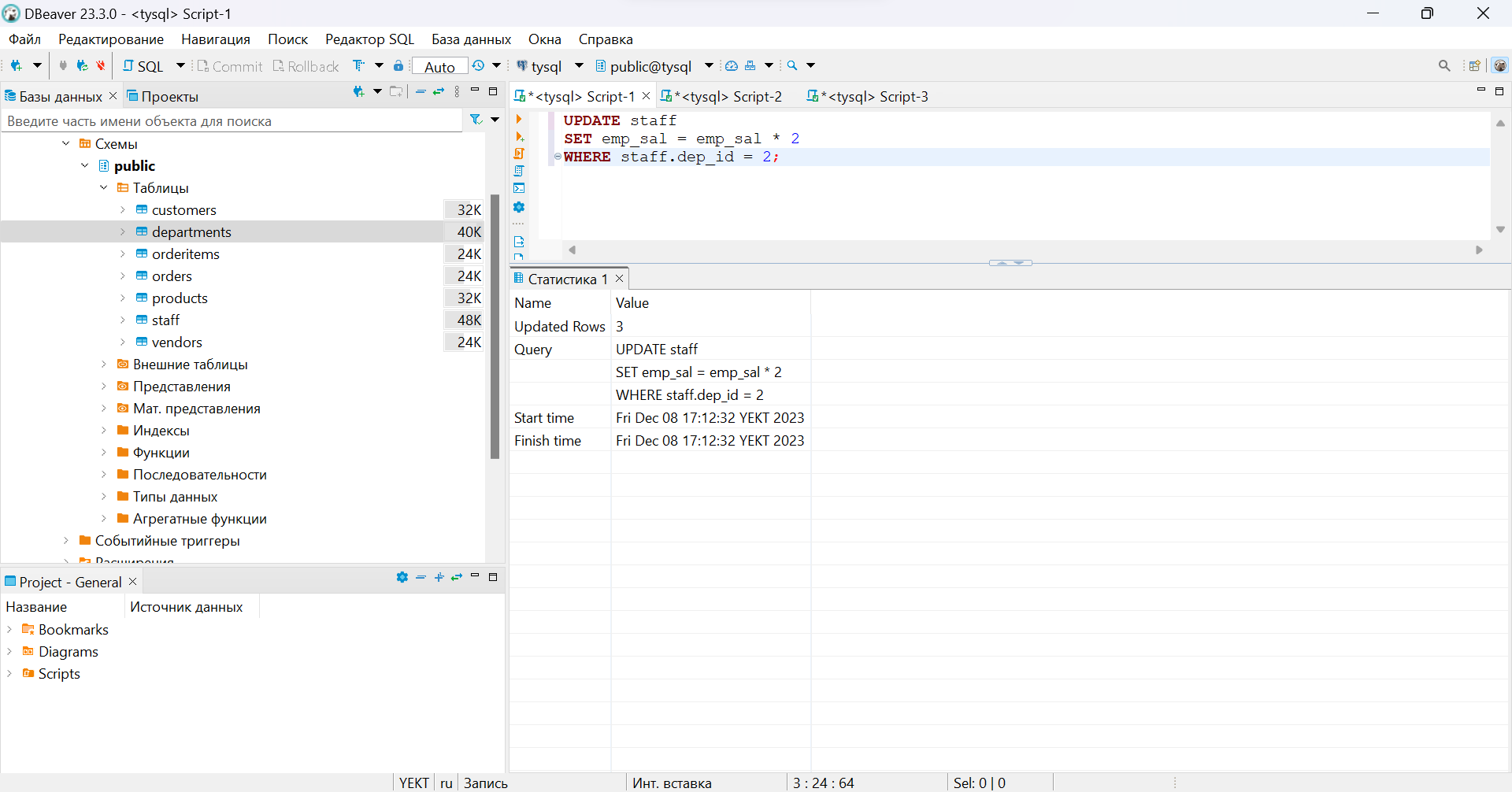


Рис. 5.1

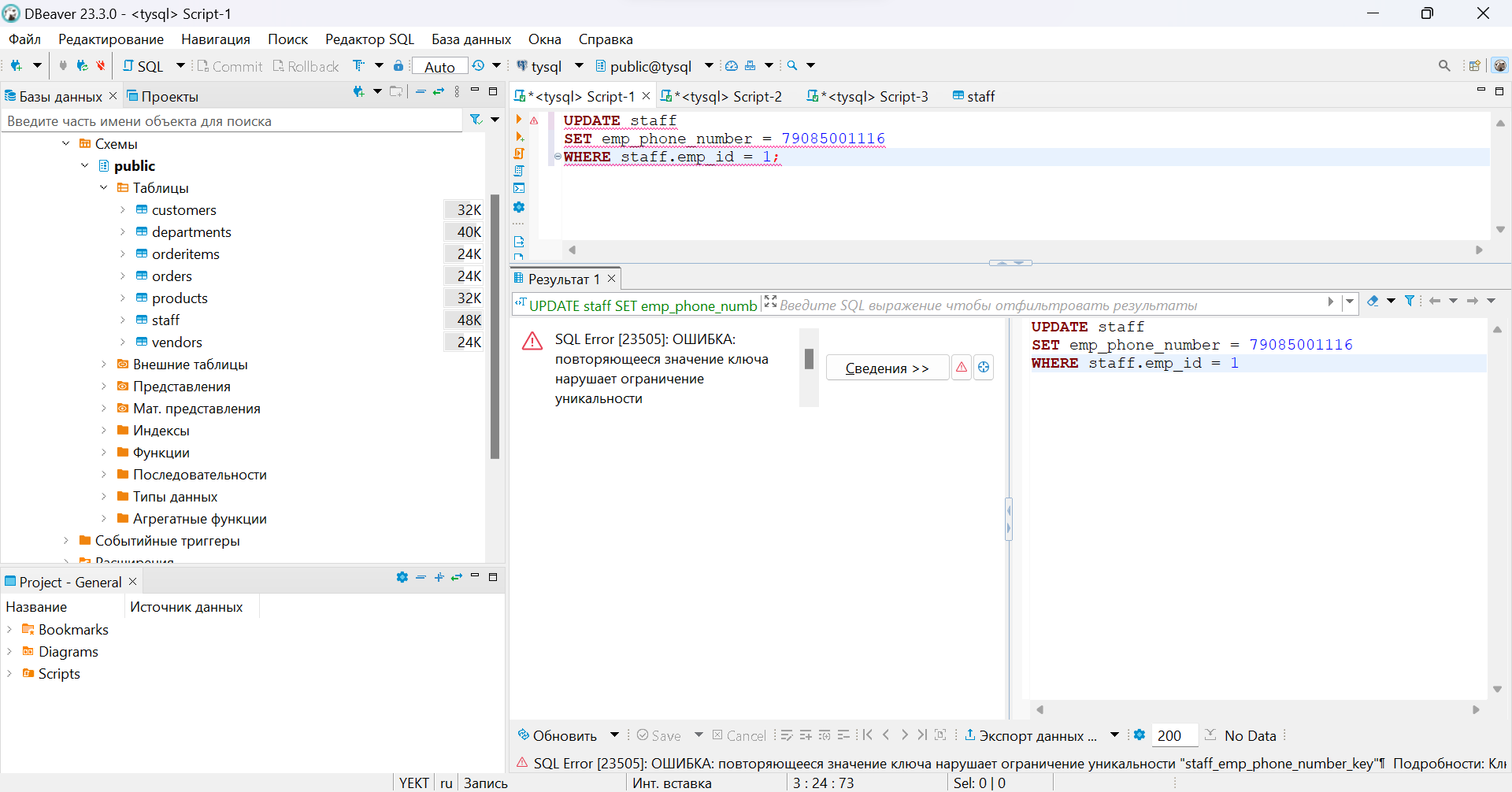


Рис. 5.2

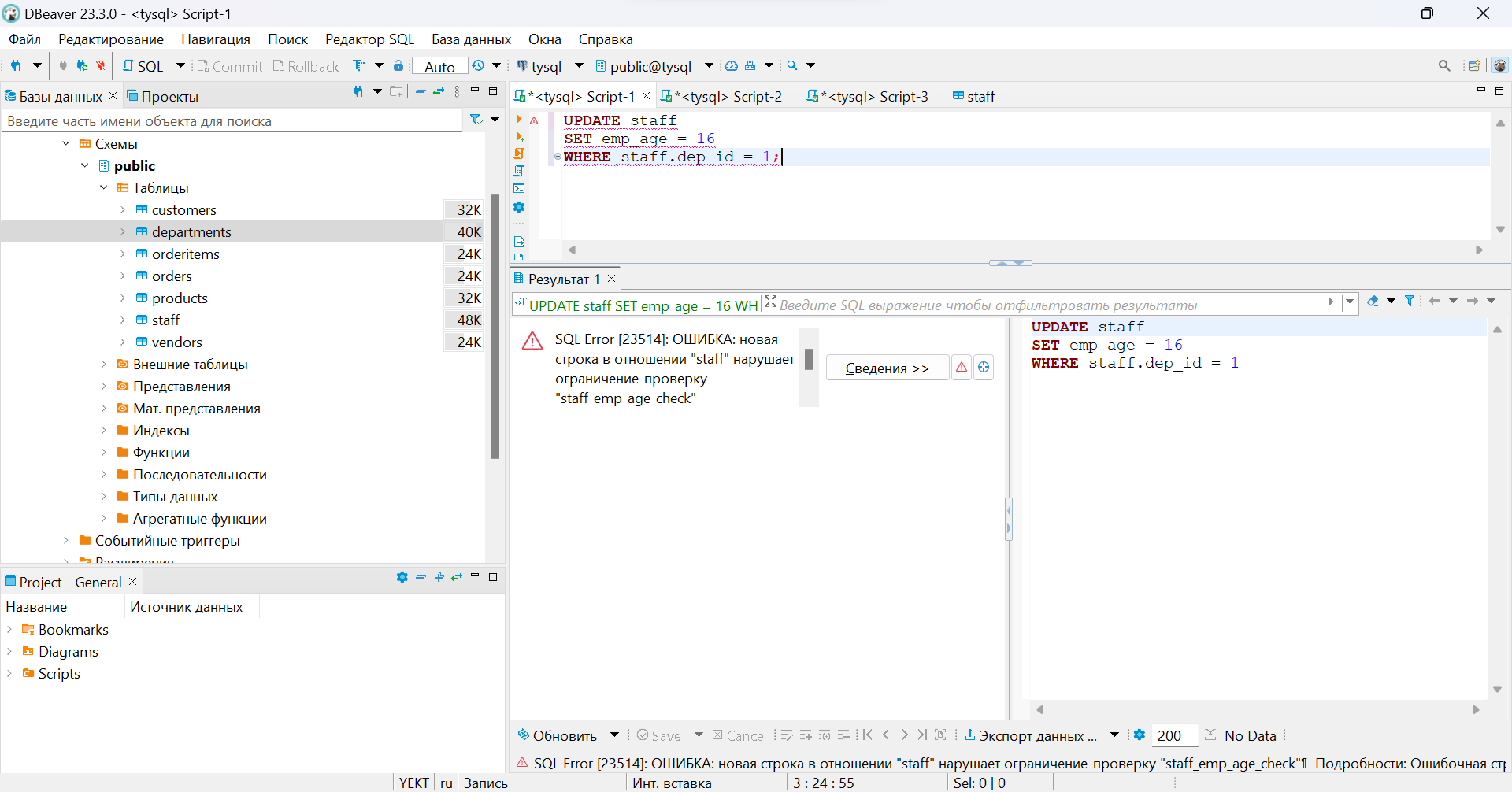


Рис. 5.3

1. Проверить результат обновления данных, выполнив команду SELECT (см. Рис. 6).

select \* from staff s

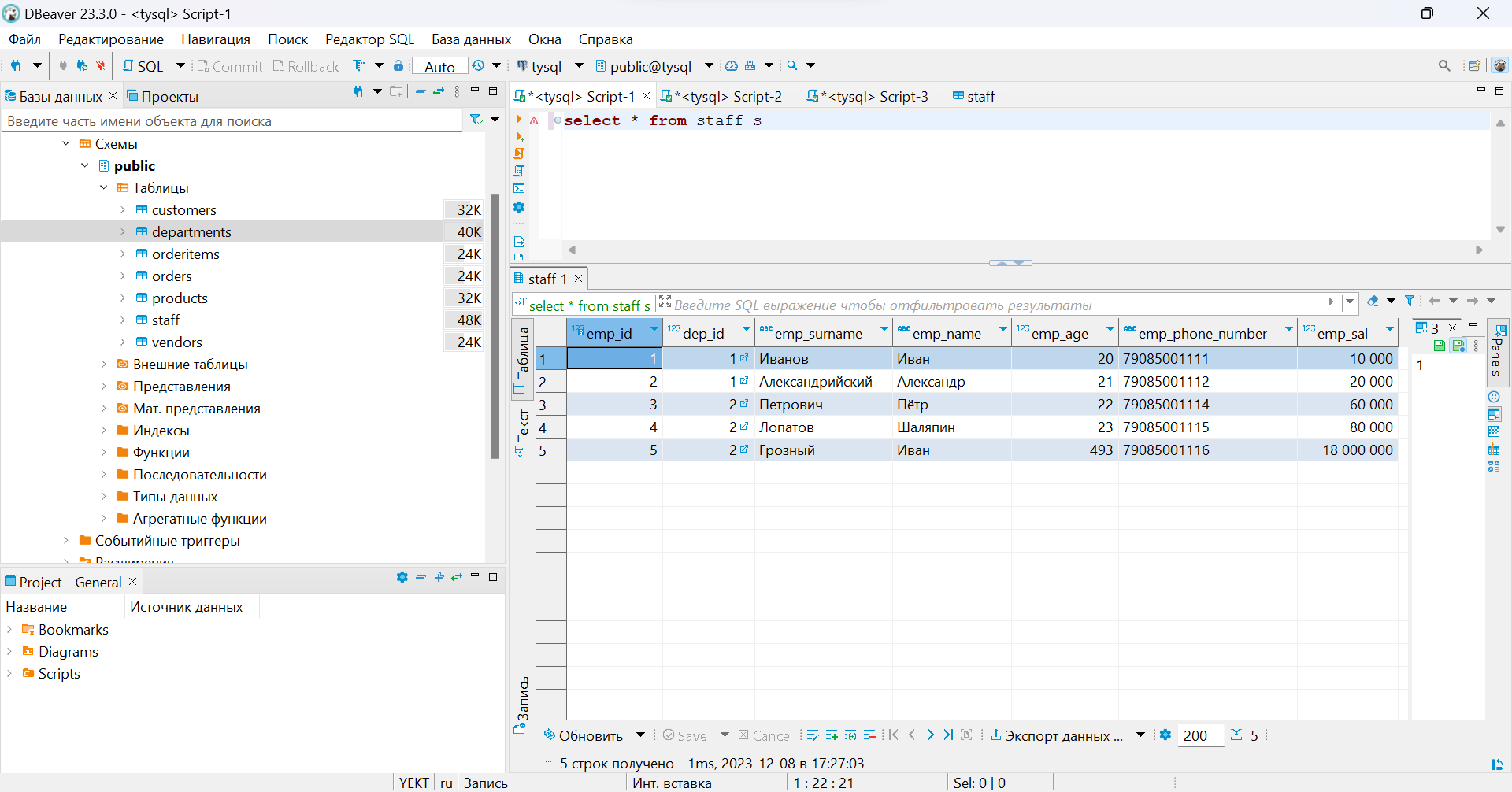


Рис. 6

1. Удалить строки из таблицы (см. Рис. 7.1; Рис. 7.2; Рис. 7.3):

* Удалить запись об отделе АХО из таблицы Отдел
* Удалить запись об отделе Бухгалтерия из таблицы Отдел
* Удалить запись о любом 1-ом сотруднике отдела ИТ

delete from departments

where dep\_name = 'АХО';

delete from departments

where dep\_name = 'Бухгалтерия';

delete from staff

where emp\_id = 5;

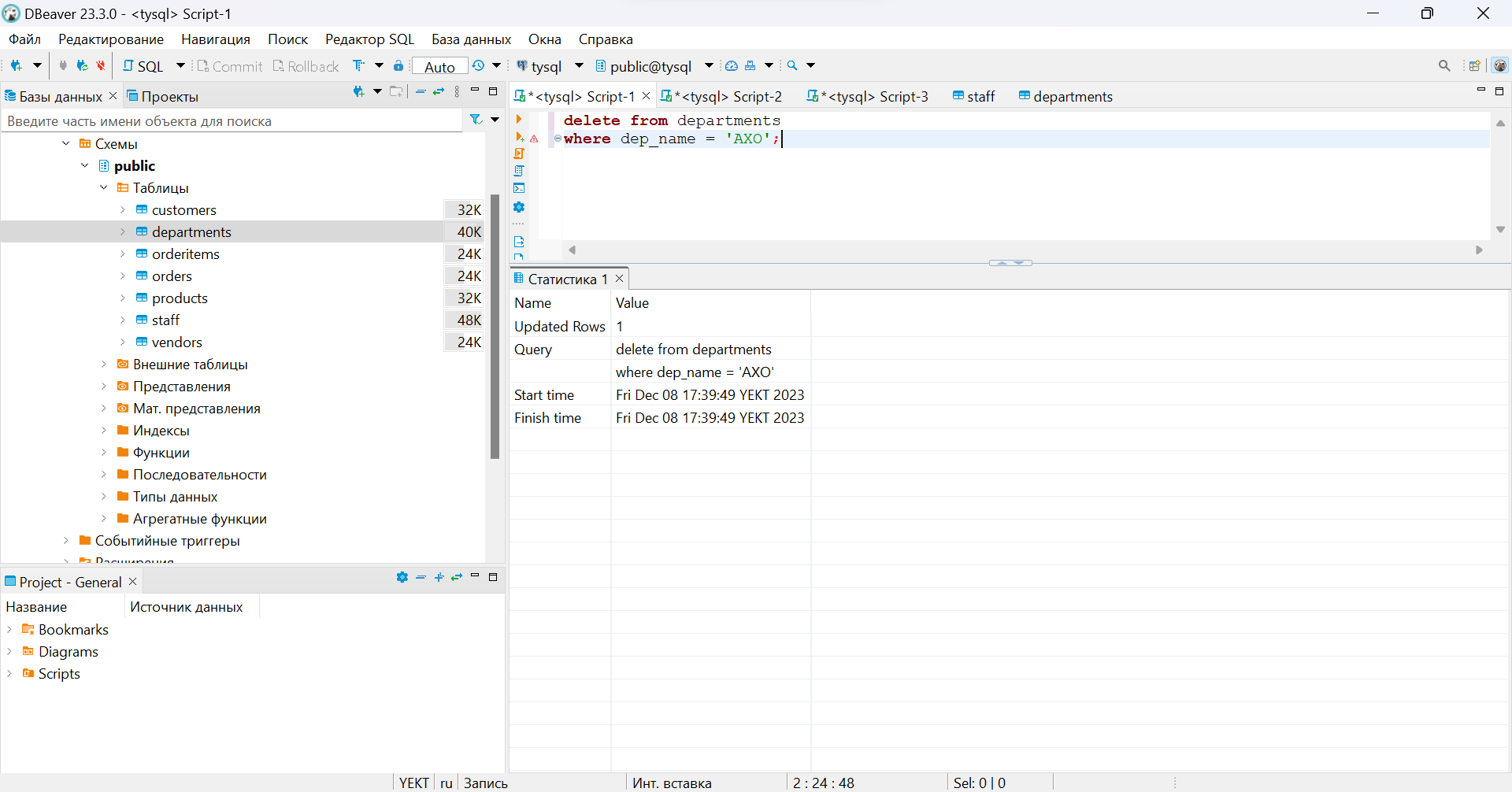


Рис. 7.1

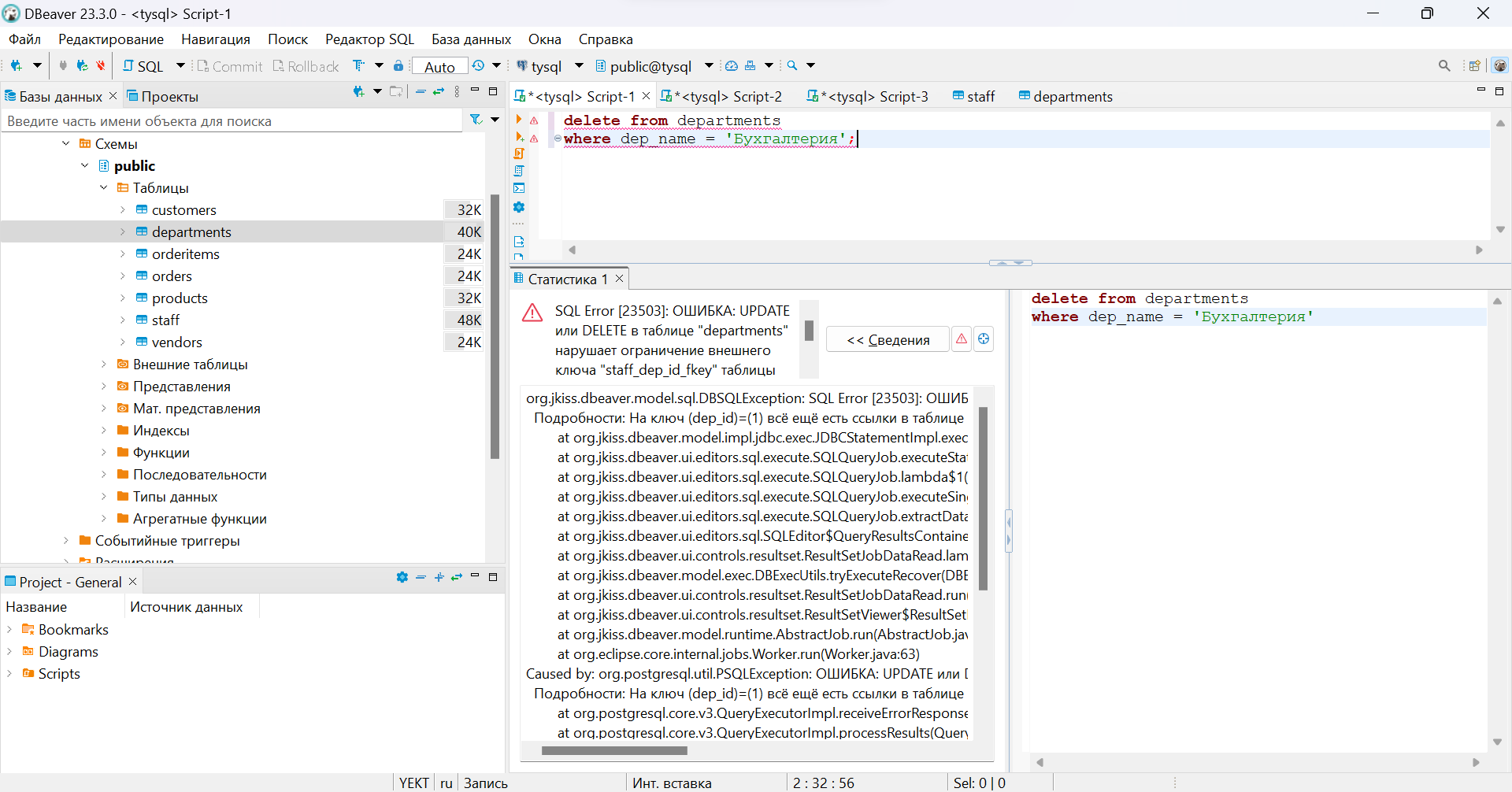


Рис. 7.2

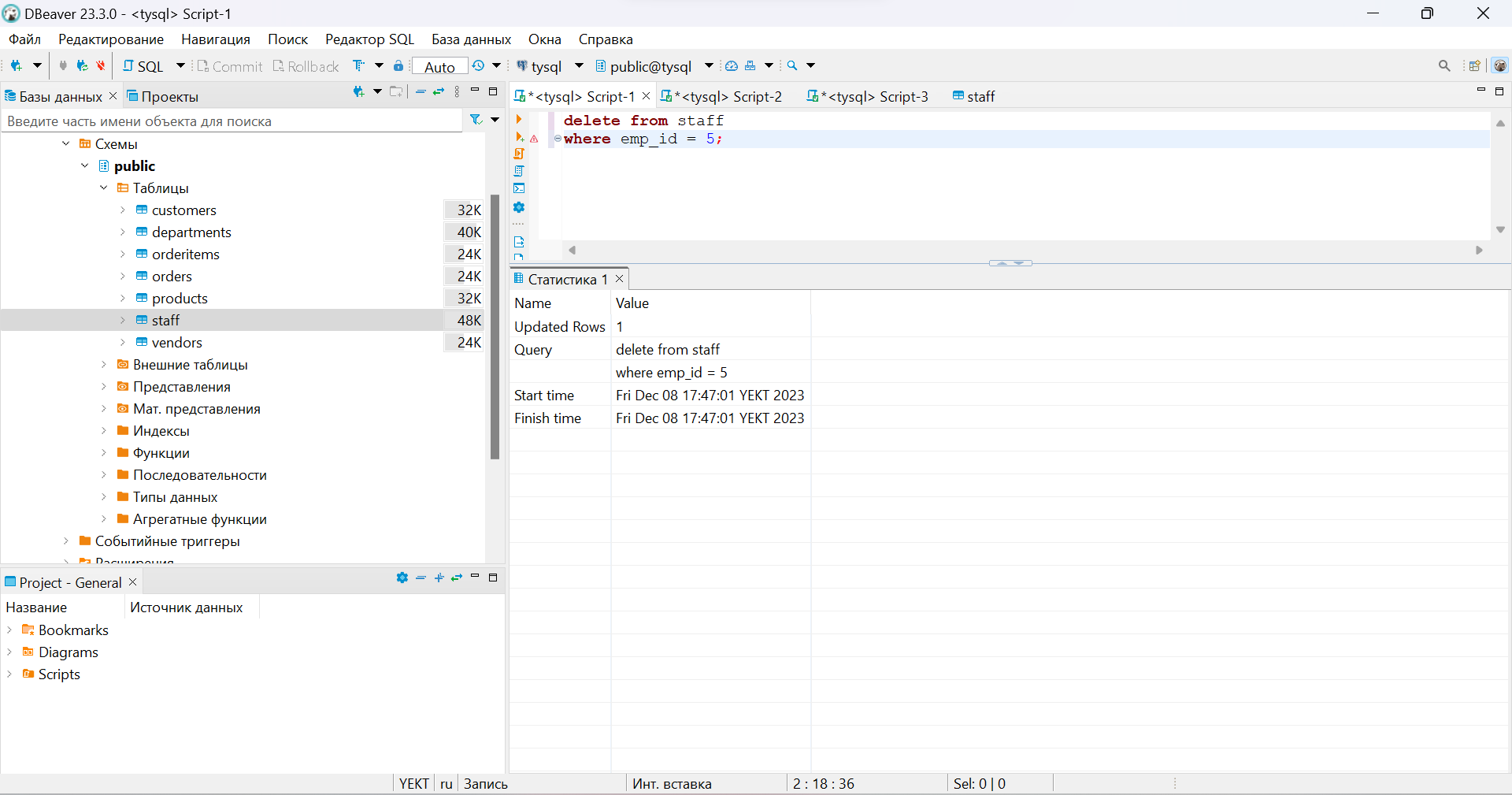


Рис. 7.3

1. Проверить результат удаления данных, выполнив команду SELECT из таблиц Отдел и Сотрудник (см. Рис. 8.1; Рис. 8.2).

select \* from departments d

select \* from staff s

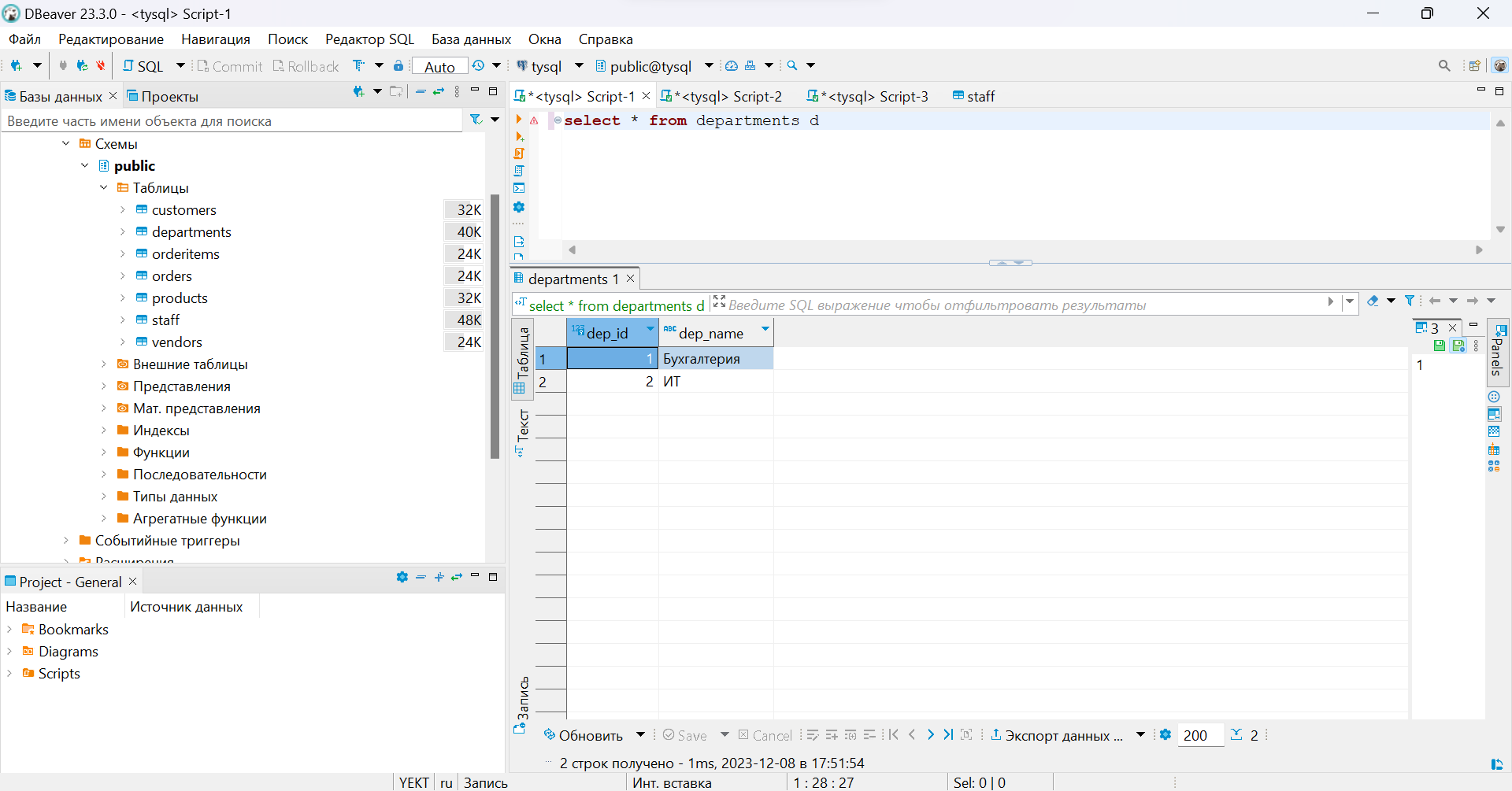


Рис. 8.1

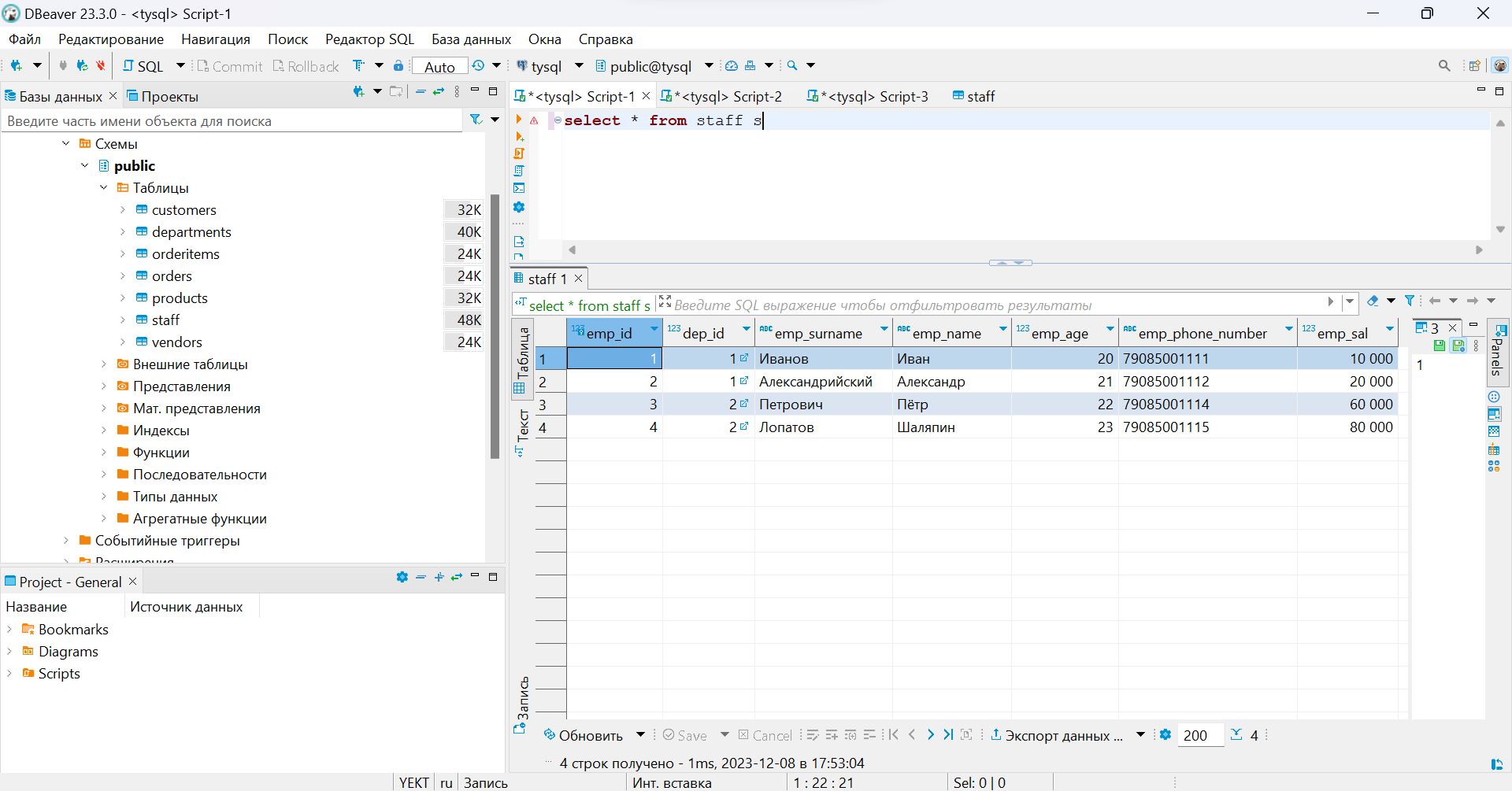


Рис. 8.2

1. Создать с помощью команды CREATE VIEW простой обзор, ограничивающий выборку из базовой таблицы. Обзор должен показывать следующие данные (см. Рис. 9):

* Код сотрудника
* Код отдела
* Фамилию сотрудника
* Имя сотрудника
* Возраст
* Телефон

CREATE VIEW staff\_v (emp\_id, dep\_id, emp\_surname, emp\_name, emp\_age, emp\_phone\_number)

AS SELECT s.emp\_id, s.dep\_id, s.emp\_surname, s.emp\_name, s.emp\_age, s.emp\_phone\_number

FROM staff s;

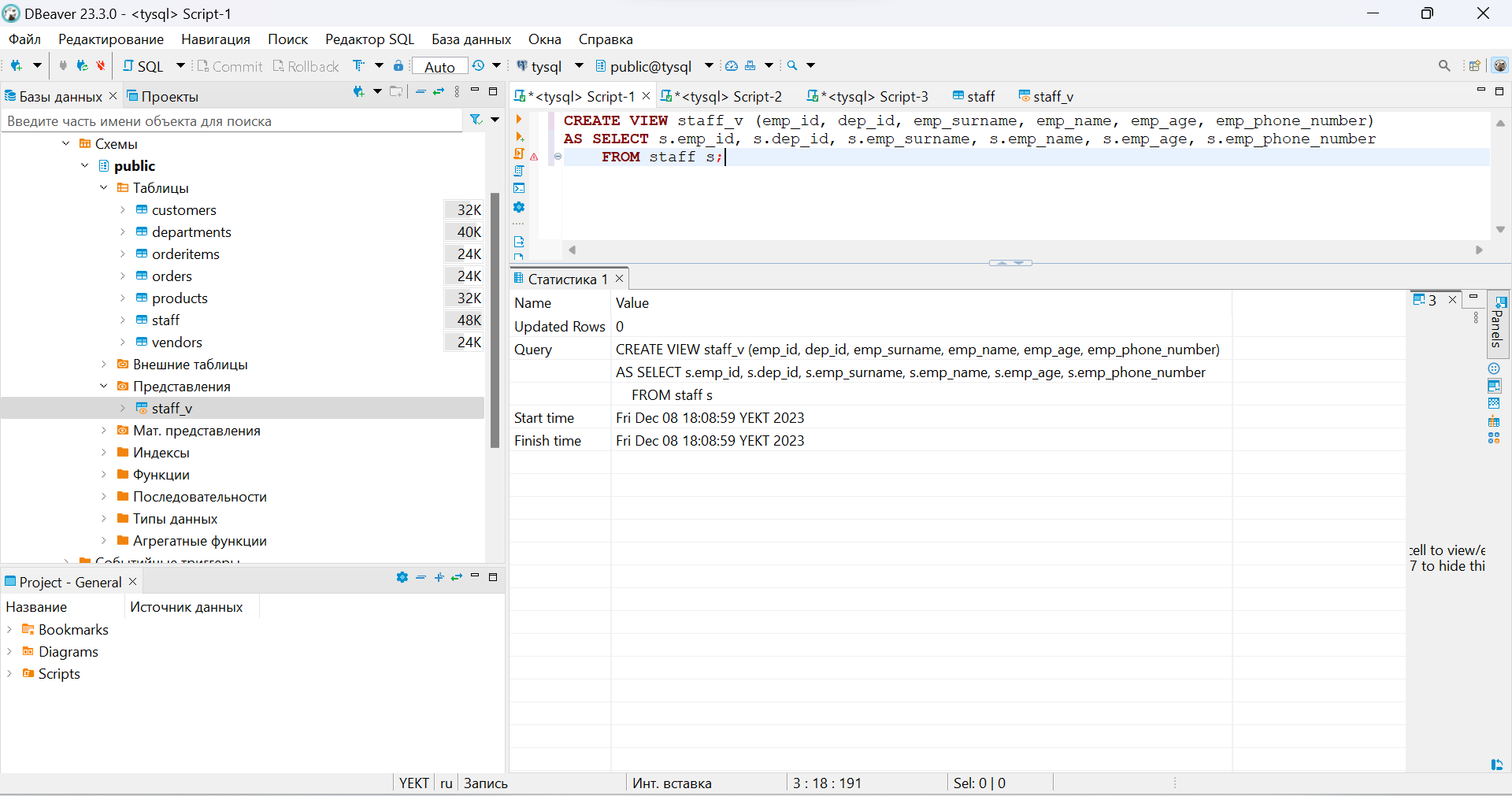


Рис. 9

1. Выполнить чтение данных из обзора из п.9, выполнив команду SELECT (см. Рис. 10):

select \* from staff\_v sv

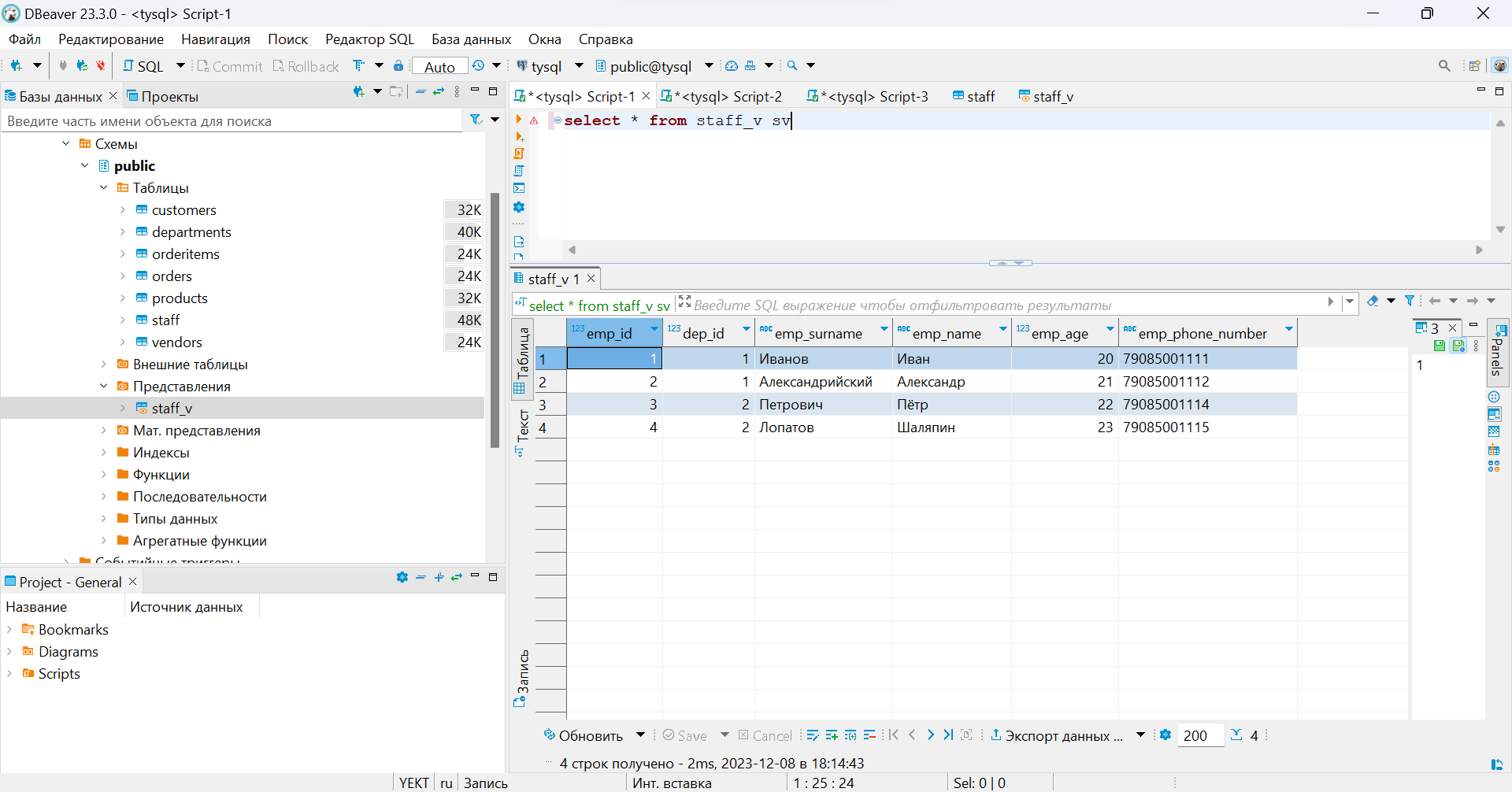


Рис. 10

1. Выполнить вставку данных через обзор из п.9 (см. Рис. 11):

* Добавить 1-го сотрудника в отдел ИТ

INSERT INTO staff\_v (emp\_id, dep\_id, emp\_surname, emp\_name, emp\_age, emp\_phone\_number)

VALUES (5, 2, 'Безручкова', 'Екатерина', 19, +79085001116);

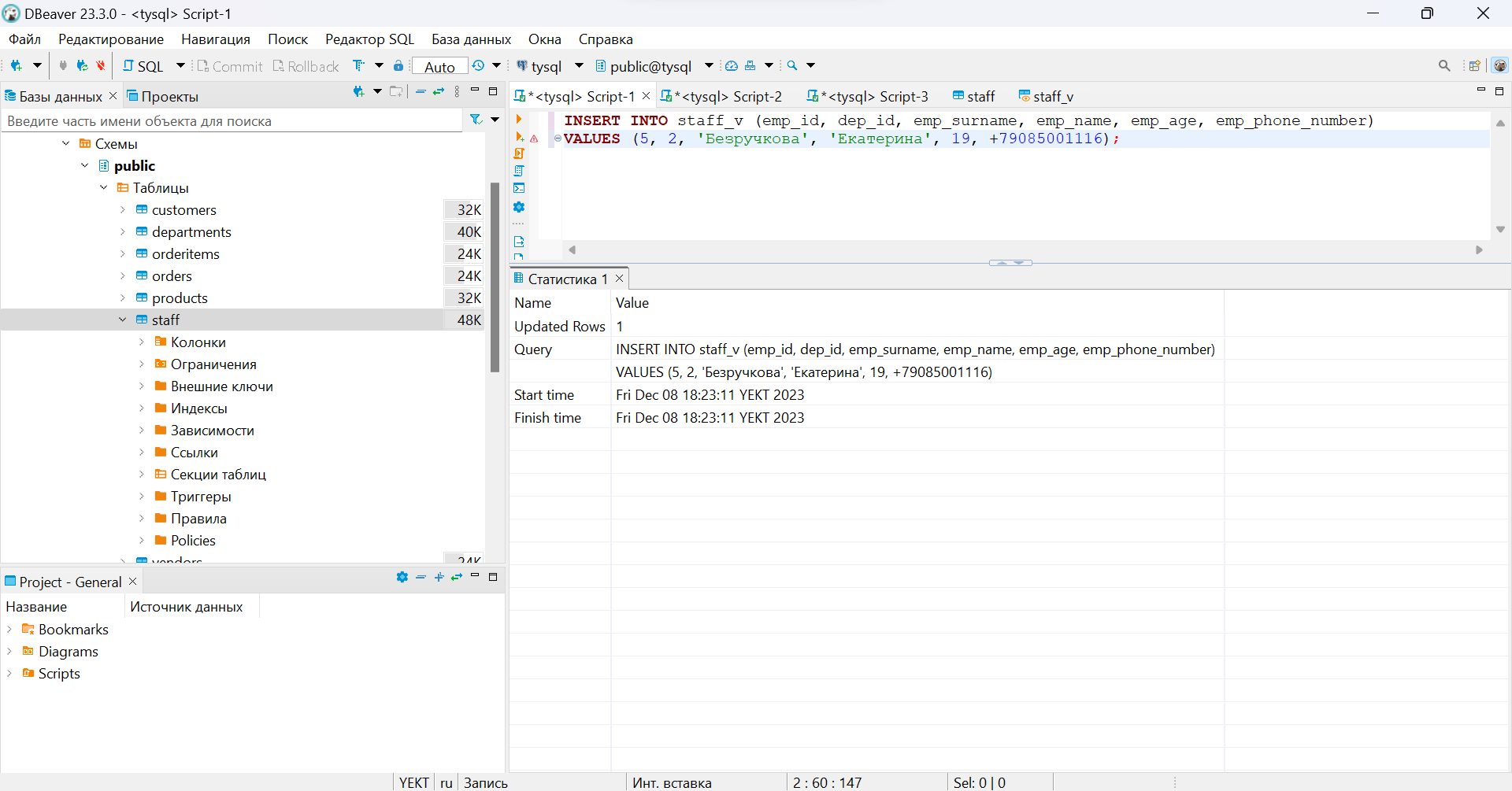


Рис. 11

1. Создать с помощью команды CREATE VIEW сложный обзор, в котором используется соединение таблиц и групповые функции. Обзор должен выводить следующие данные (см. Рис. 12):

* Код отдела
* Имя отдела
* Суммарный оклад всех сотрудников отдела

CREATE VIEW departments\_v (dep\_id, dep\_name, sum\_of\_sal\_all)

AS SELECT d.dep\_id, d.dep\_name, sum(s.emp\_sal)

FROM departments d

join staff s on d.dep\_id = s.dep\_id

group by d.dep\_id, d.dep\_name;

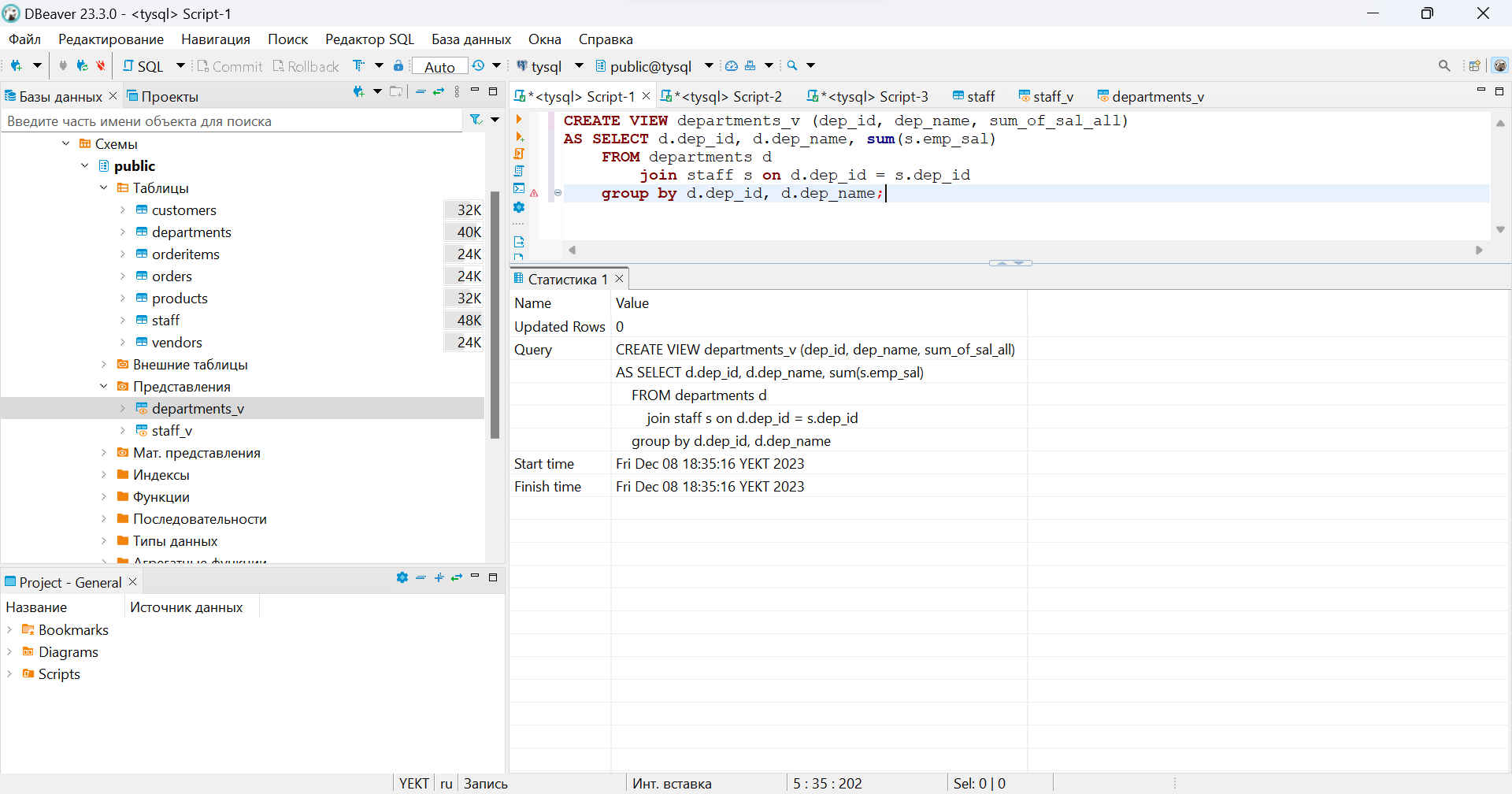


Рис. 12

1. Выполнить чтение данных из обзора из п.9, выполнив команду SELECT (см. Рис. 13):

select \* from departments\_v dv

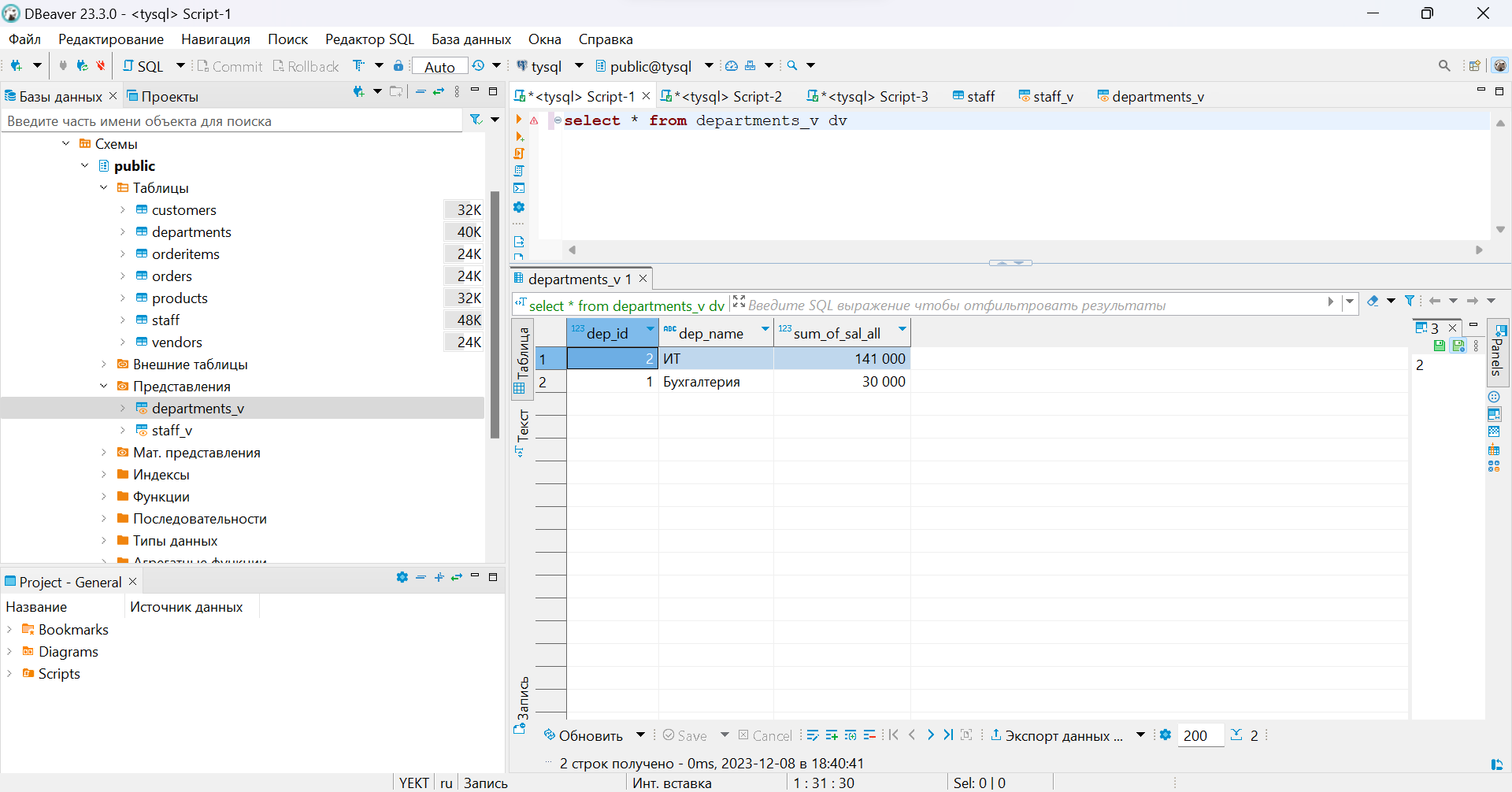


Рис. 13

1. Попробовать выполнить вставку данных через сложный обзор из п.12 (см. Рис. 14):

* Добавить новый отдел - Менеджеры

INSERT INTO departments\_v (dep\_id, dep\_name)

VALUES (3, 'Менеджеры');

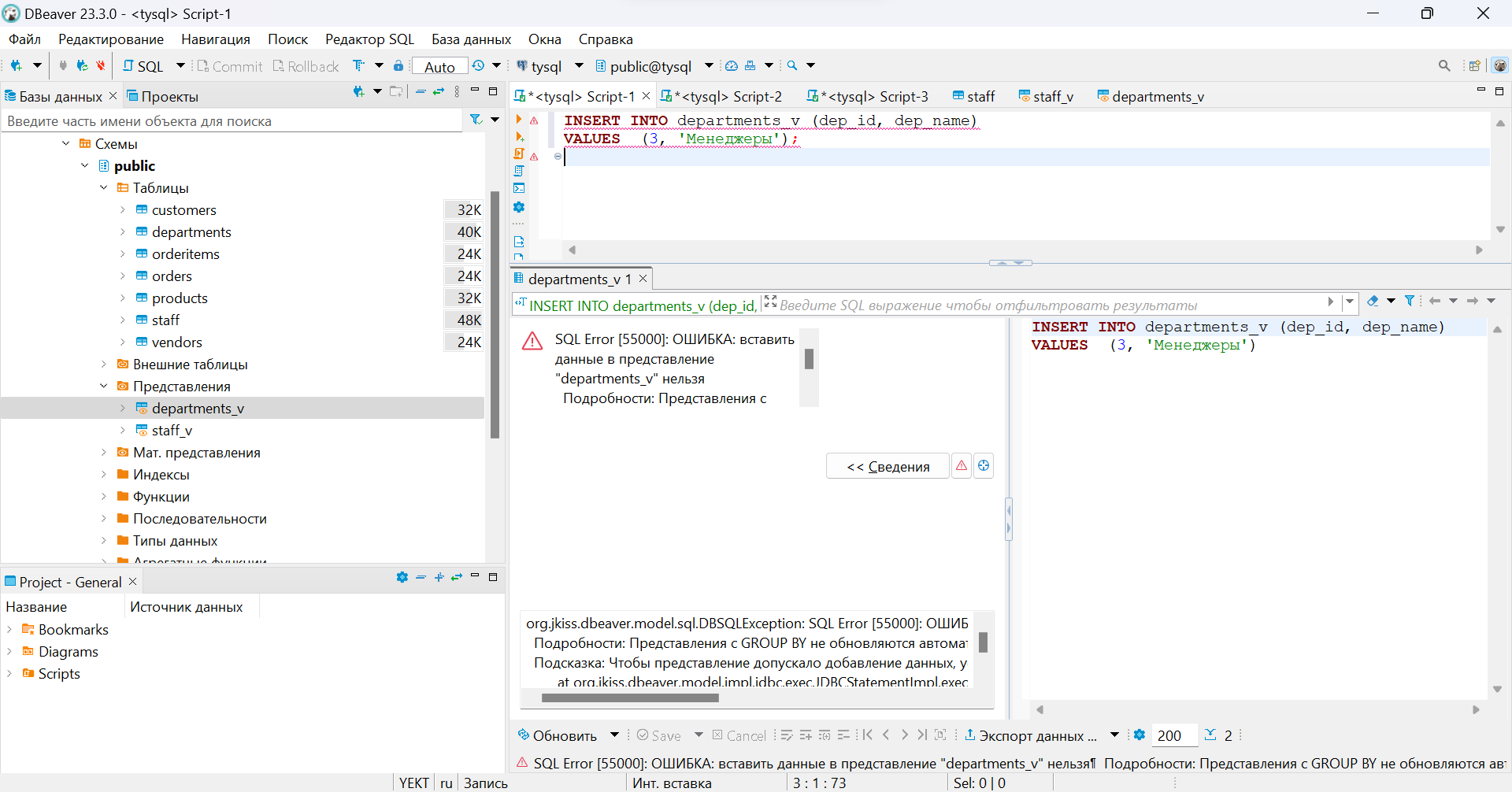


Рис. 14

1. Создать материализованное представление, которое должно содержать следующие данные (см. Рис. 15):

* Код отдела
* Имя отдела
* Количество сотрудников в отделе

CREATE MATERIALIZED VIEW staff\_mv (dep\_id, dep\_name, count\_of\_staff)

as select d.dep\_id, d.dep\_name, count(s.emp\_id)

from departments d

join staff s on d.dep\_id = s.dep\_id

group by d.dep\_id, d.dep\_name;

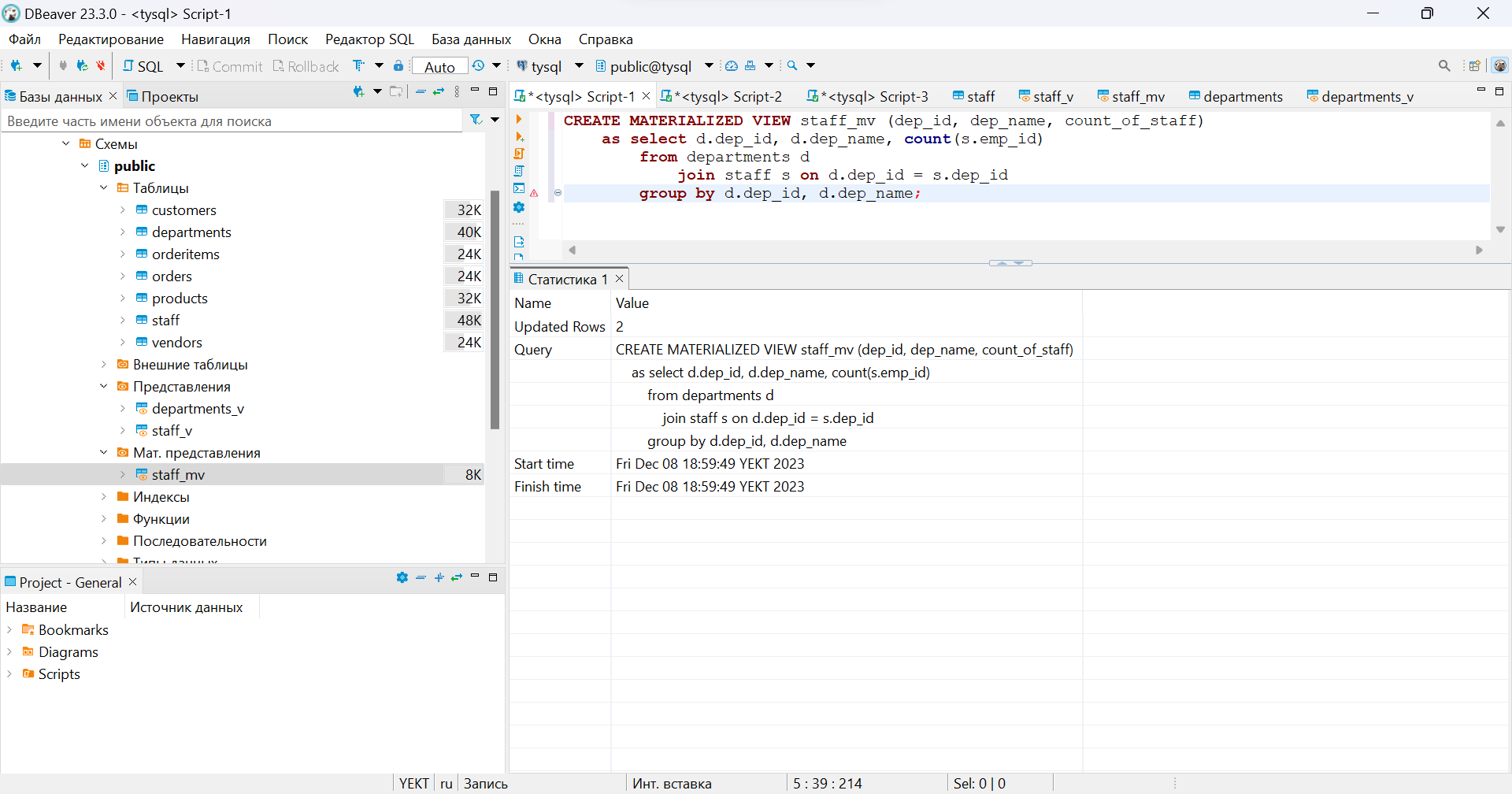


Рис. 15

1. Обновить материализованное представление из п.15 (см. Рис. 16).

REFRESH MATERIALIZED VIEW staff\_mv;

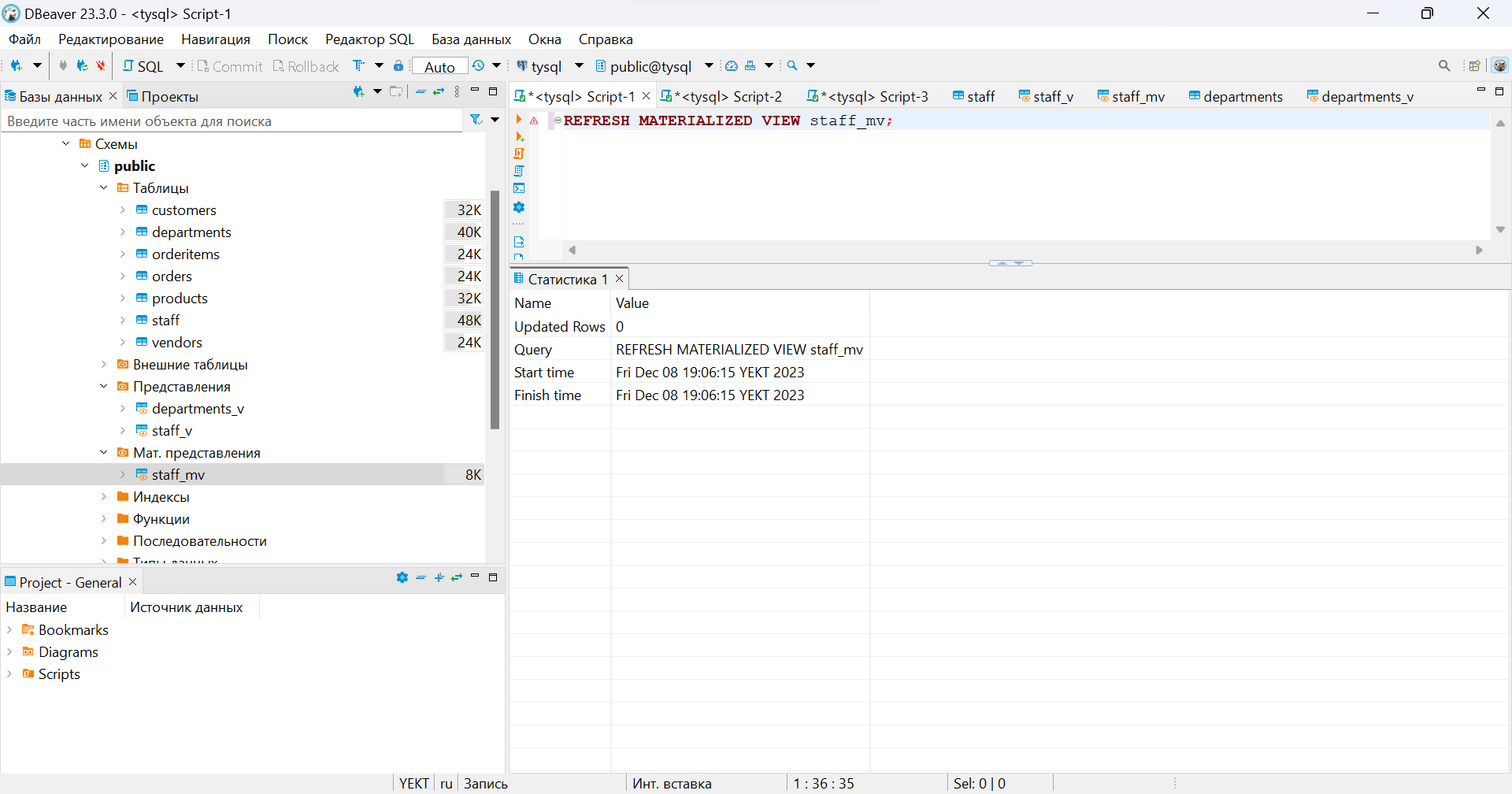


Рис. 16

1. Выполнить чтение данных из материализованного представления из п.15 (см. Рис. 17).

select \* from staff\_mv sm

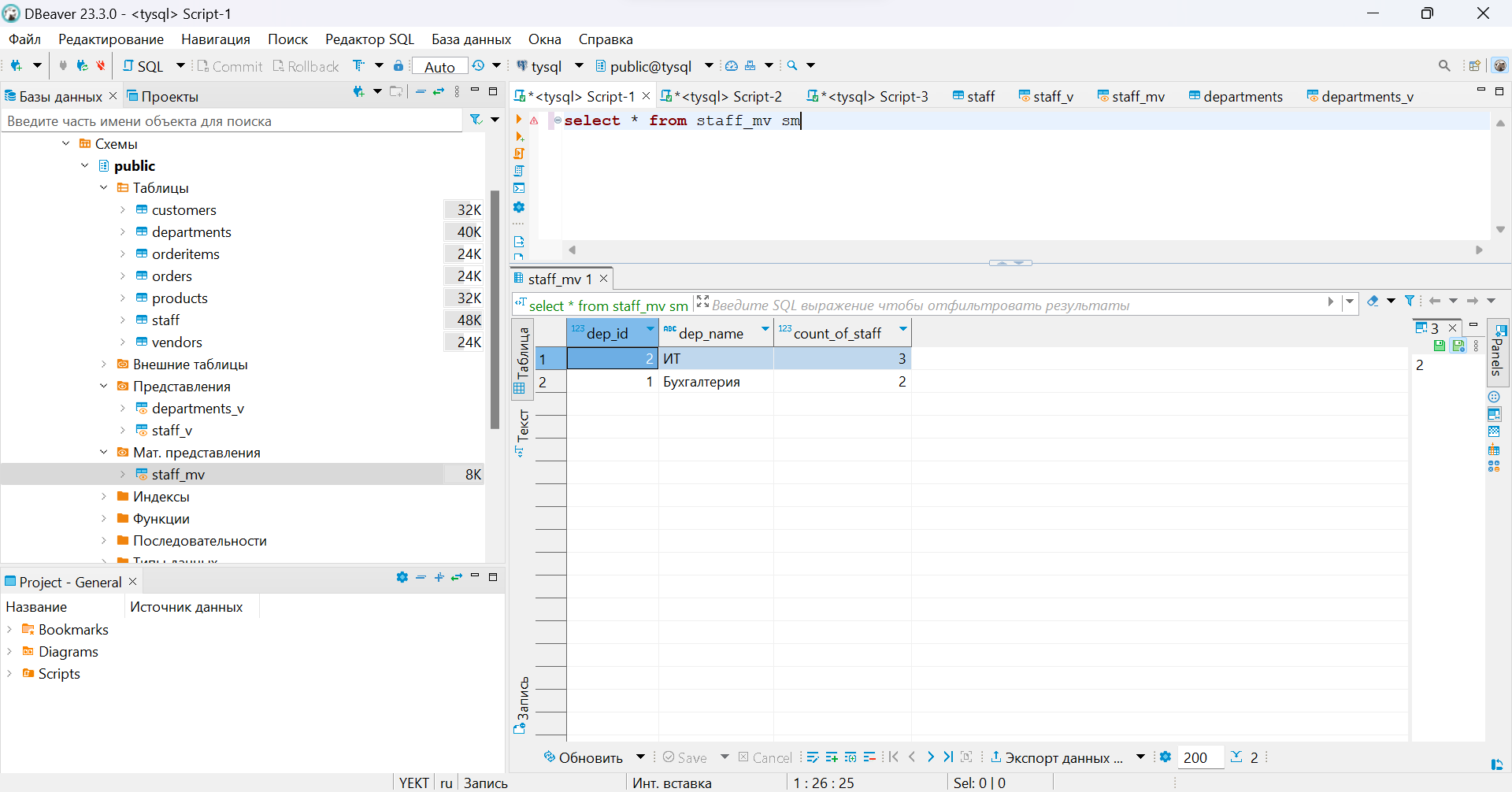


Рис. 17

1. Выполнить вставку новой записи в таблицу сотрудников (см. Рис. 18):

* Добавить сотрудника в отдел ИТ

INSERT INTO staff (emp\_id, dep\_id, emp\_surname, emp\_name, emp\_age, emp\_phone\_number, emp\_sal)

VALUES (6, 2, 'Петренко', 'Юрий', 50, +79085001117, 100000);

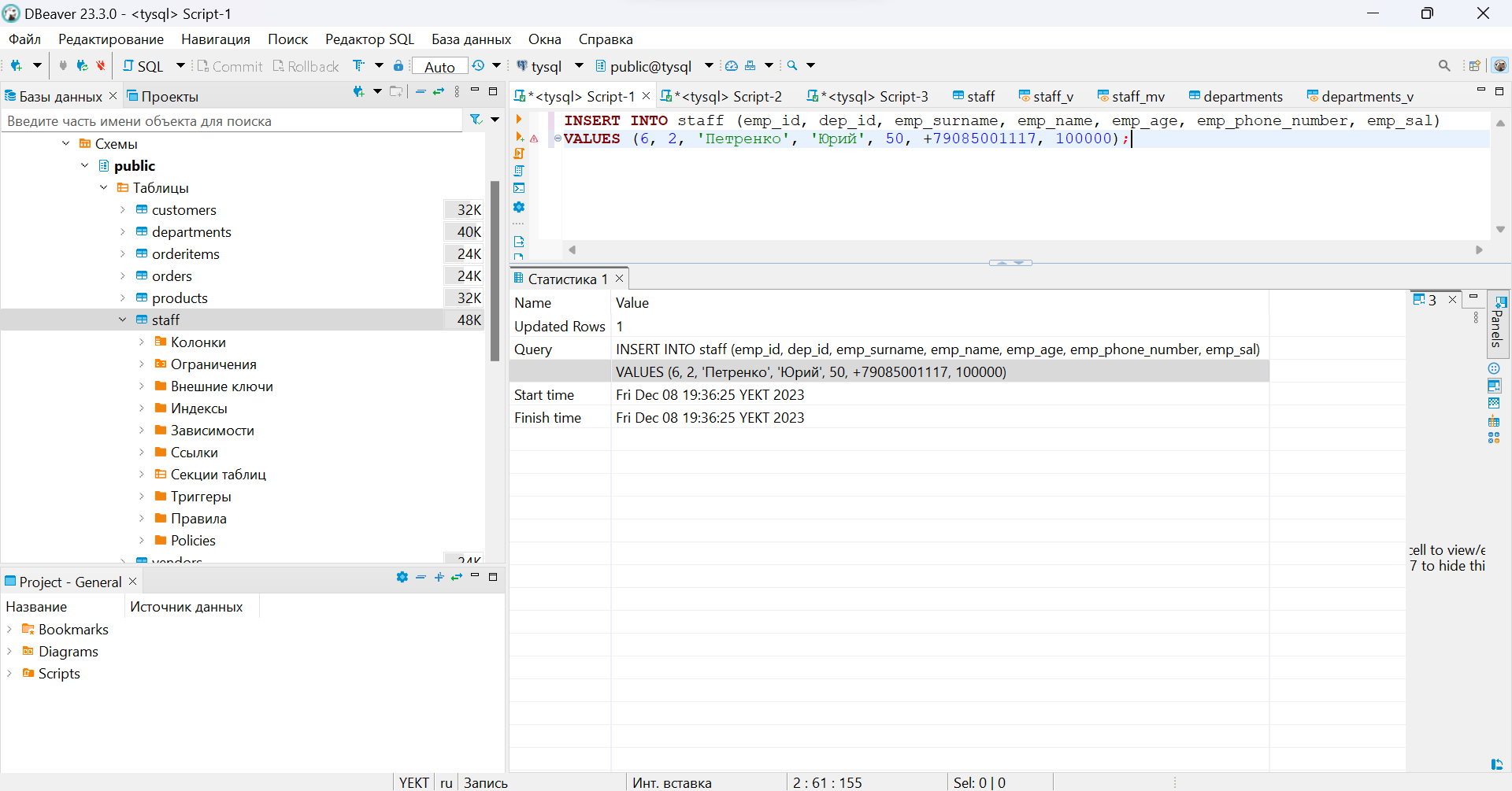


Рис. 18

1. Выполнить чтение данных из материализованного представления из п.15 (см. Рис. 19).

select \* from staff\_mv sm

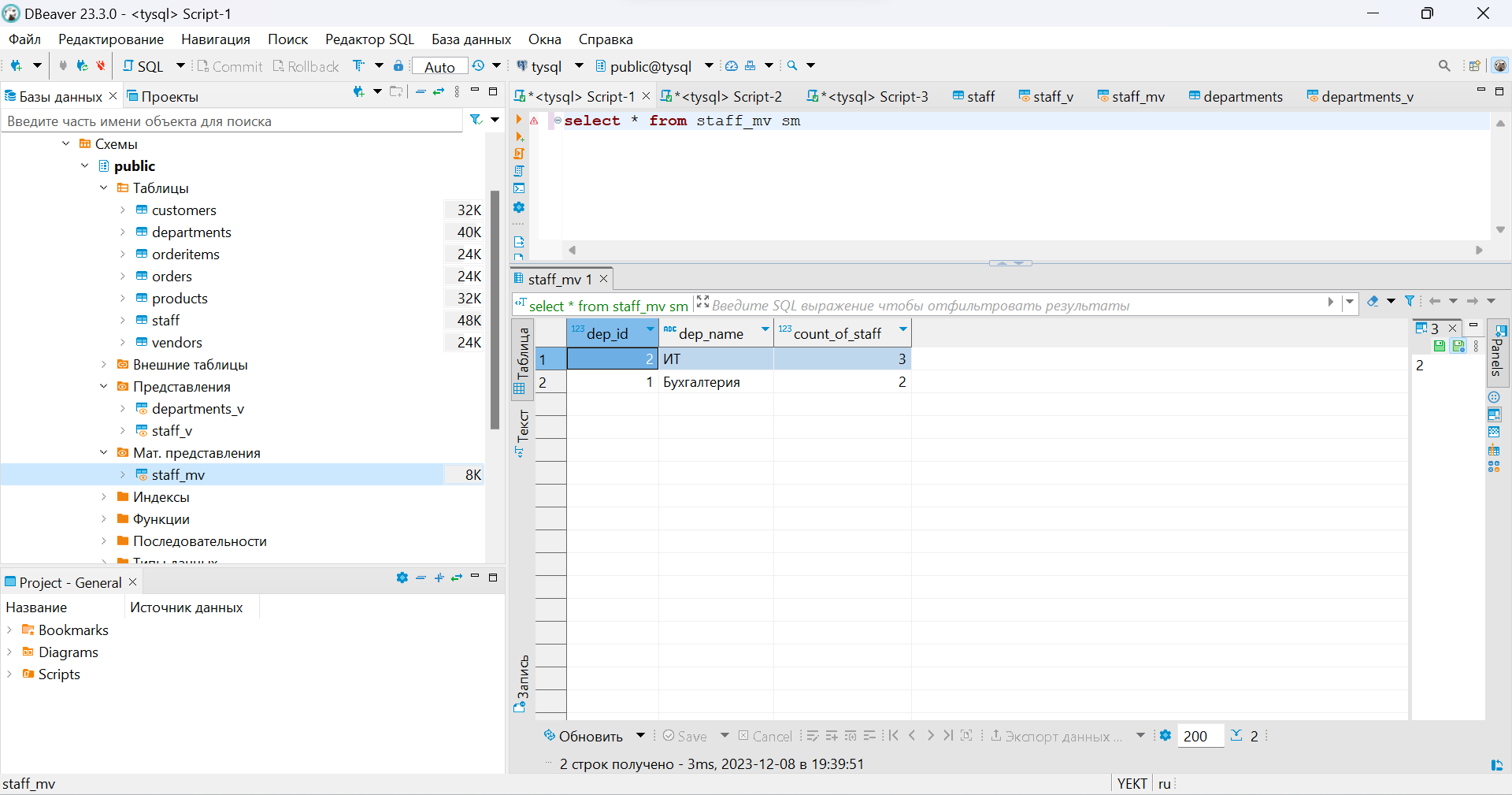


Рис. 19

1. Обновить материализованное представление из п.15 (см. Рис. 20).

REFRESH MATERIALIZED VIEW staff\_mv;

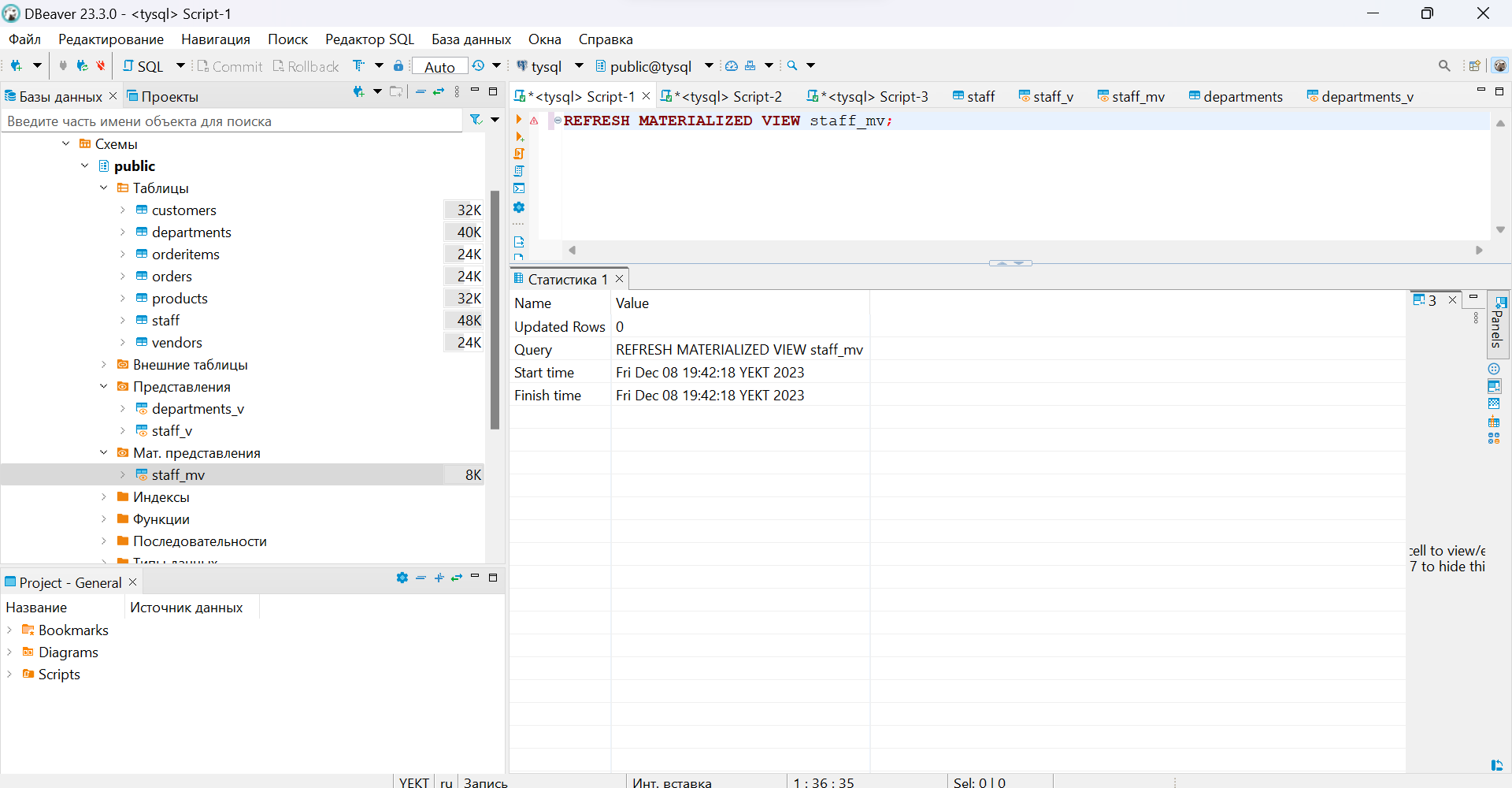


Рис. 20

1. Выполнить чтение данных из материализованного представления из п.15 (см. Рис. 21).

select \* from staff\_mv sm

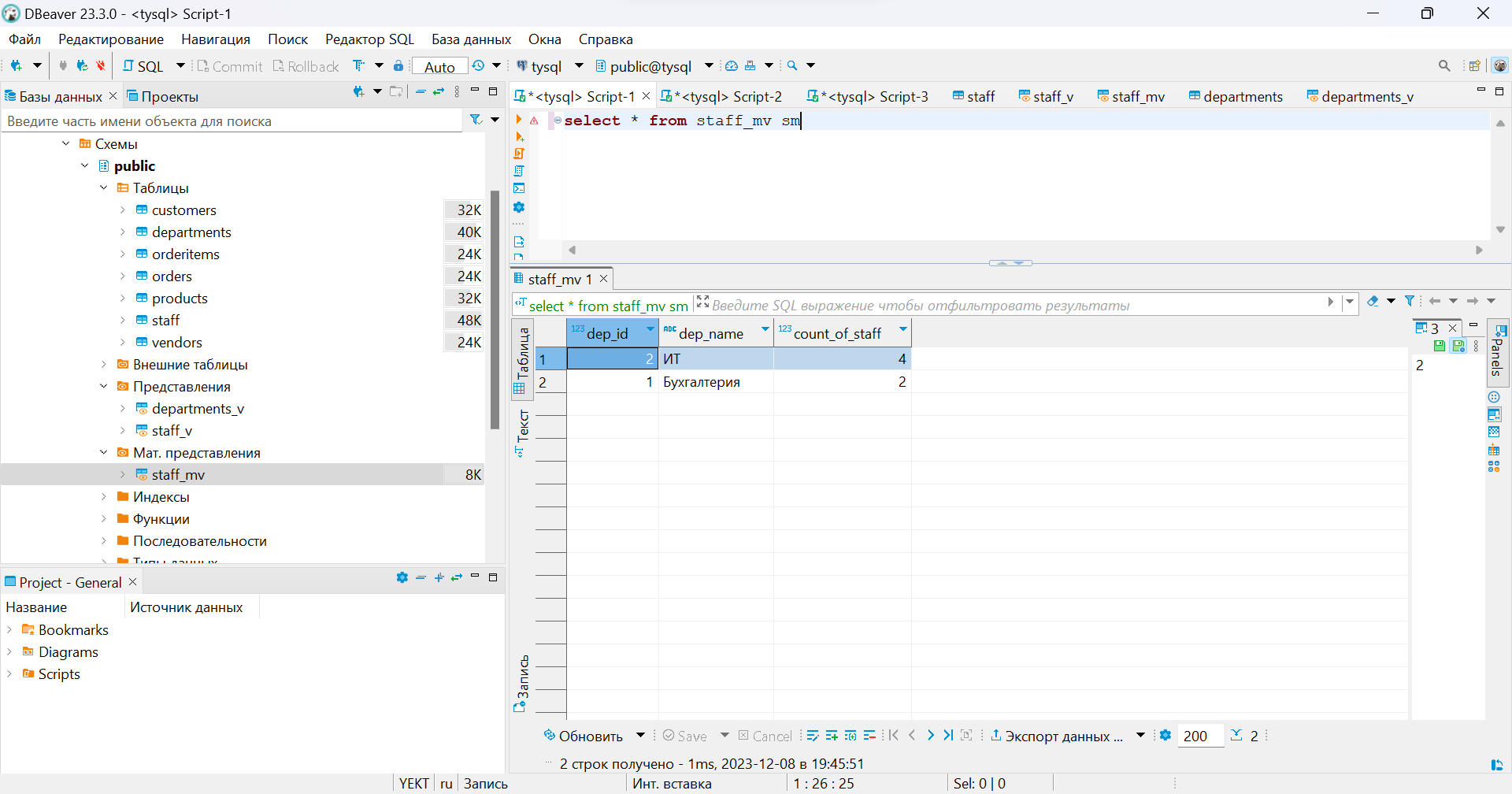


Рис. 21