Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОННИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Лабораторная работа №3

Реализация SQL-запросов для создания базы данных

Студент: К.В. Горбачевский

Преподаватель: Д.В. Куприянова

МИНСК 2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc160389953)

[1 СОЗДАНИЕ ТАБЛИЦ 5](#_Toc160389954)

[2 НАПОЛНЕНИЕ ТАБЛИЦ 7](#_Toc160389955)

[3 ДРУГИЕ SQL-КОМАНДЫ 22](#_Toc160389956)

[4 СОЗДАНИЕ ВРЕМЕННОЙ ТАБЛИЦЫ 23](#_Toc160389957)

[5 ГЕНЕРАЦИЯ ERD-ДИАГРАММЫ 24](#_Toc160389958)

[6 ЭКСПОРТ РЕЗУЛЬТАТОВ 26](#_Toc160389959)

[ЗАКЛЮЧЕНИЕ 27](#_Toc160389960)

# ВВЕДЕНИЕ

В ходе выполнения данной лабораторной работы необходимо cформировать и выполнить SQL-запросы для создания таблиц базы данных, а также заполнить таблицы данными с помощью оператора INSERT и пользовательского интерфейса.

1) Создать в СУБД новую схему данных для хранения пользовательских

объектов.

2) В этой новой схеме данных с помощью скрипта с запросами на языке SQL реализовать таблицы, соответствующие реляционным отношениям схемы, полученной в лабораторной работе №2, с помощью одного (желательно) оператора create table для каждой таблицы в следующем порядке:

• реализовать простую структуру таблиц, включающую только набор столбцов с добавлением описаний первичного ключа;

• дополнить описание таблицы реализацией ограничений для описания внешних ключей;

• дополнить описание таблицы реализацией ограничений для описания бизнес-правил;

3) Заполнить с помощью SQL-скрипта с использованием оператора insert половину таблиц строками данных для проверки правильного выбора первичных ключей и работоспособности ссылок между таблицами. Другую половину таблиц заполнить при помощи пользовательского интерфейса.

4) Рассмотреть простые действия по изменению структуры таблицы и реализовать их с помощью оператора alter table.

5) Создать временную таблицу и удалить ее

6) Экспортировать результаты работы в SQL-скрипт и сравнить со скриптом, выгруженным в этапах 2 и 3.

# СОЗДАНИЕ ТАБЛИЦ

Для написания SQL-запросов требуется открыть панель SQL tools:

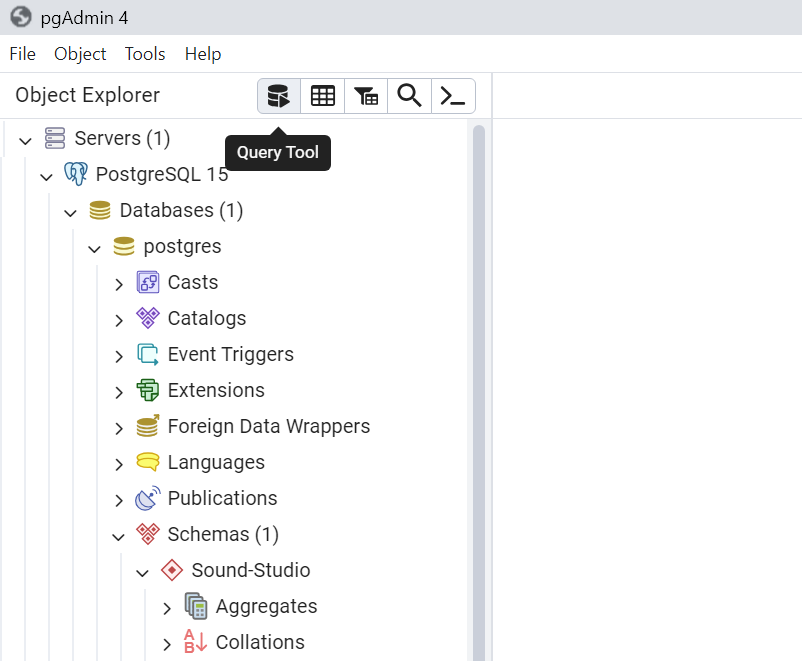


Рисунок 1.1 – Панель Query Tool

Ниже приведены SQL-запросы для создания всех таблиц по варианту:

CREATE TABLE IF NOT EXISTS "Sound-Studio"."Instrument"

(

id integer NOT NULL GENERATED ALWAYS AS IDENTITY,

name "char"[] NOT NULL,

type "char"[] NOT NULL,

rent integer NOT NULL,

PRIMARY KEY (id)

);

CREATE TABLE IF NOT EXISTS "Sound-Studio"."Room"

(

id integer NOT NULL,

square integer NOT NULL,

rent integer NOT NULL,

"personsCount" integer NOT NULL,

PRIMARY KEY (id)

);

CREATE TABLE IF NOT EXISTS "Sound-Studio"."Record Type"

(

id integer NOT NULL,

type "char"[] NOT NULL,

genre "char"[],

"stereo/mono" "char"[] NOT NULL,

PRIMARY KEY (id)

);

CREATE TABLE IF NOT EXISTS "Sound-Studio"."Artist"

(

id integer NOT NULL,

name "char"[] NOT NULL,

genre "char"[] NOT NULL,

"personsCount" integer NOT NULL,

PRIMARY KEY (id)

);

CREATE TABLE IF NOT EXISTS "Sound-Studio"."Employee"

(

id integer NOT NULL,

job "char"[] NOT NULL,

"workShift" integer NOT NULL,

name "char"[] NOT NULL,

PRIMARY KEY (id)

);

CREATE TABLE IF NOT EXISTS "Sound-Studio"."Record"

(

id integer NOT NULL GENERATED ALWAYS AS IDENTITY,

"totalPrice" integer NOT NULL,

"hoursCount" integer NOT NULL,

date date NOT NULL,

"recordTypeId" integer NOT NULL,

"employeeId" integer NOT NULL,

"roomId" integer NOT NULL,

"artistId" integer NOT NULL,

PRIMARY KEY (id)

);

CREATE TABLE IF NOT EXISTS "Sound-Studio"."Instrument\_Record"

(

id integer NOT NULL GENERATED ALWAYS AS IDENTITY,

"Instruments\_id" integer NOT NULL,

"Record\_id" integer NOT NULL,

PRIMARY KEY (id)

);

# НАПОЛНЕНИЕ ТАБЛИЦ

Для заполнения таблиц информацией используется директива *INSERT*. Ниже приведены SQL-запросы и скриншоты заполненных таблиц:

1. Таблица Artist:

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('The Smashing Pumpkins', 'Rock', 6);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Metallica', 'Metal', 4);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('The Beatles', 'Rock', 4);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Pink Floyd', 'Progressive Rock', 4);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Queen', 'Rock', 4);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Led Zeppelin', 'Rock', 4);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Nirvana', 'Grunge', 3);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('The Doors', 'Rock', 4);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Radiohead', 'Alternative Rock', 5);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('AC/DC', 'Rock', 5);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('The Who', 'Rock', 4);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Guns N Roses', 'Hard Rock', 5);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('The Beach Boys', 'Rock', 5);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('R.E.M.', 'Alternative Rock', 4);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('The Eagles', 'Rock', 6);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Black Sabbath', 'Metal', 4);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('U2', 'Rock', 4);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('The Police', 'Rock', 3);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Fleetwood Mac', 'Rock', 5);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Deep Purple', 'Rock', 5);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Aerosmith', 'Rock', 5);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Green Day', 'Punk Rock', 3);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Pearl Jam', 'Grunge', 5);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Red Hot Chili Peppers', 'Alternative Rock', 4);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('The Clash', 'Punk Rock', 4);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Blur', 'Alternative Rock', 4);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('The Cure', 'Alternative Rock', 4);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Ramones', 'Punk Rock', 4);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('Oasis', 'Rock', 5);

INSERT INTO "Sound-Studio"."Artist"(name, genre, "personsCount") VALUES ('The Smiths', 'Indie Rock', 4);

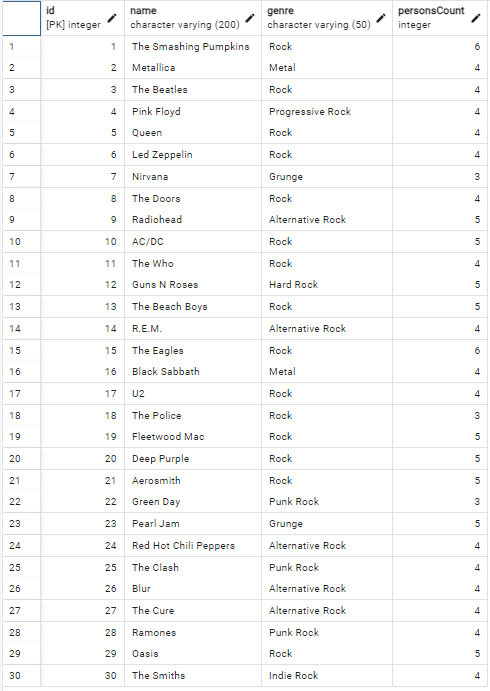


Рисунок 2.1 – Строки таблицы «Artist»

1. Таблица Employee:

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Звукорежиссер', '2', 'Иван Петров');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Продюсер', '1', 'Мария Сидорова');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Менеджер по маркетингу', '3', 'Алексей Иванов');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Техник', '1', 'Елена Козлова');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Менеджер по продажам', '2', 'Дмитрий Федоров');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Звуковой дизайнер', '3', 'Анна Николаева');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Композитор', '1', 'Сергей Смирнов');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Аранжировщик', '2', 'Ольга Павлова');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Текстовый автор', '3', 'Павел Кузнецов');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Сценарист', '1', 'Татьяна Игнатьева');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Режиссер клипа', '2', 'Владимир Степанов');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Оператор', '3', 'Максим Лебедев');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Агент по бронированию студии', '1', 'Евгений Романов');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Менеджер по связям с общественностью', '2', 'Алина Васильева');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Программист', '3', 'Денис Жуков');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Дизайнер', '1', 'Наталья Яковлева');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Тестировщик', '2', 'Артем Борисов');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Контент-менеджер', '3', 'Маргарита Калинина');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Администратор', '1', 'Ирина Степанова');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Фотограф', '2', 'Валентин Попов');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Видеограф', '3', 'Юлия Смирнова');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Менеджер по контрактам', '1', 'Андрей Кузьмин');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Финансовый аналитик', '2', 'Олег Михайлов');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Юрист', '3', 'Екатерина Волкова');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Технический писатель', '1', 'Полина Никитина');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Системный администратор', '2', 'Игорь Попов');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('HR-специалист', '3', 'Марина Козлова');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Логист', '1', 'Виктор Сидоров');

INSERT INTO "Sound-Studio"."Employee"(job, "workShift", name) VALUES ('Рекрутер', '2', 'Анна Иванова');



Рисунок 2.2 – Строки таблицы «Employee»

1. Таблица Instruments:

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Фортепиано', 'Клавишные', 5);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Электрогитара', 'Струнные', 7);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Барабаны', 'Ударные', 10);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Саксофон', 'Духовые', 8);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Бас-гитара', 'Струнные', 7);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Синтезатор', 'Клавишные', 6);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Труба', 'Духовые', 9);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Контрабас', 'Струнные', 10);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Флейта', 'Духовые', 8);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Акустическая гитара', 'Струнные', 6);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Ударная установка', 'Ударные', 12);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Трубчатые колокола', 'Ударные', 4);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Виолончель', 'Струнные', 9);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Кларнет', 'Духовые', 7);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Тамбурин', 'Ударные', 3);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Электро-барабаны', 'Ударные', 15);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Пианино', 'Клавишные', 7);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Альт', 'Струнные', 8);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Банджо', 'Струнные', 6);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Микрофон', 'Звукозапись', 4);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Аккордеон', 'Клавишные', 8);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Губная гармошка', 'Духовые', 5);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Электронная скрипка', 'Струнные', 10);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Микшерный пульт', 'Звукозапись', 10);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('DJ-пульт', 'Звукозапись', 15);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Семплер', 'Звукозапись', 12);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Акустический рояль', 'Клавишные', 10);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Концертный гранд-рояль', 'Клавишные', 15);

INSERT INTO "Sound-Studio"."Instrument"(name, type, rent) VALUES ('Электронная барабанная установка', 'Ударные', 20);

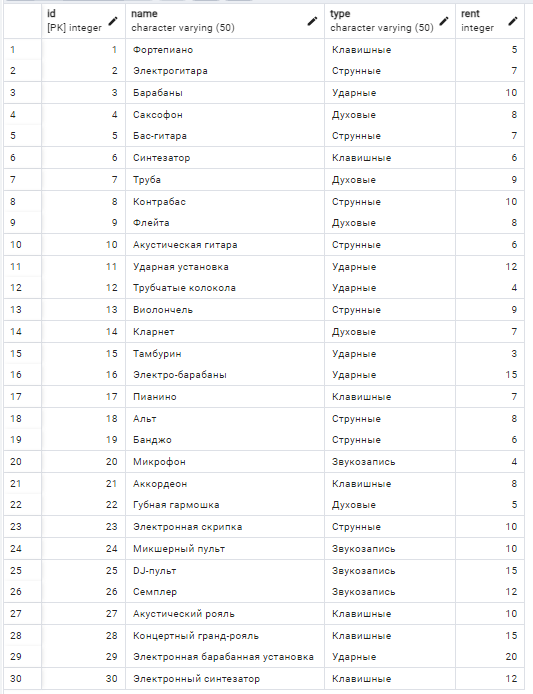


Рисунок 2.3 – Строки таблицы «Instrument»

1. Таблица Room:

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (25, 12, 4);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (30, 15, 6);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (20, 10, 3);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (35, 18, 8);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (28, 14, 5);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (22, 11, 4);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (32, 16, 7);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (27, 13, 5);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (18, 9, 3);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (40, 20, 10);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (26, 14, 6);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (23, 12, 4);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (38, 18, 9);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (21, 11, 3);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (29, 15, 7);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (34, 17, 8);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (31, 16, 6);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (24, 12, 5);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (37, 19, 9);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (19, 10, 4);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (36, 18, 8);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (33, 17, 7);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (25, 13, 5);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (20, 10, 4);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (30, 15, 7);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (28, 14, 6);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (22, 12, 5);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (35, 18, 9);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (27, 13, 6);

INSERT INTO "Sound-Studio"."Room"(square, rent, "personsCount") VALUES (23, 11, 4);

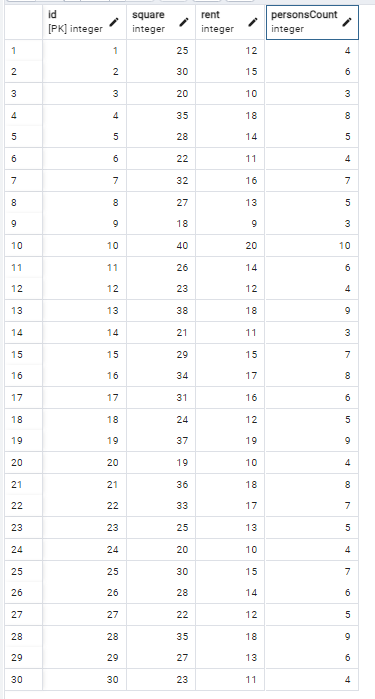


Рисунок 2.4 – Строки таблицы «Room»

1. Таблица RecordType:

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Вокал', 'Рок', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Гитара', 'Рок', 'mono');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Барабаны', 'Рок', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Клавишные', 'Рок', 'mono');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Бас-гитара', 'Рок', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Скрипка', 'Классика', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Фортепиано', 'Классика', 'mono');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Смычковые', 'Классика', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Духовые', 'Джаз', 'mono');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Контрабас', 'Джаз', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Труба', 'Джаз', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Тенор-саксофон', 'Джаз', 'mono');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Флейта', 'Классика', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Акустическая гитара', 'Фолк', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Банджо', 'Фолк', 'mono');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Мандолина', 'Фолк', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Скрипка', 'Фолк', 'mono');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Акустическая гитара', 'Акустика', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Виолончель', 'Классика', 'mono');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Концертный рояль', 'Классика', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Вокал', 'Поп', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Гитара', 'Поп', 'mono');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Клавишные', 'Поп', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Бас-гитара', 'Поп', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Электронные', 'Электроника', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Синтезатор', 'Электроника', 'mono');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Сэмплер', 'Электроника', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Скрэтч', 'Хип-хоп', 'mono');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Вокал', 'Хип-хоп', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Бас-гитара', 'Хип-хоп', 'stereo');

INSERT INTO "Sound-Studio"."RecordType"(type, genre, "stereo/mono") VALUES ('Ударные', 'Хип-хоп', 'mono');

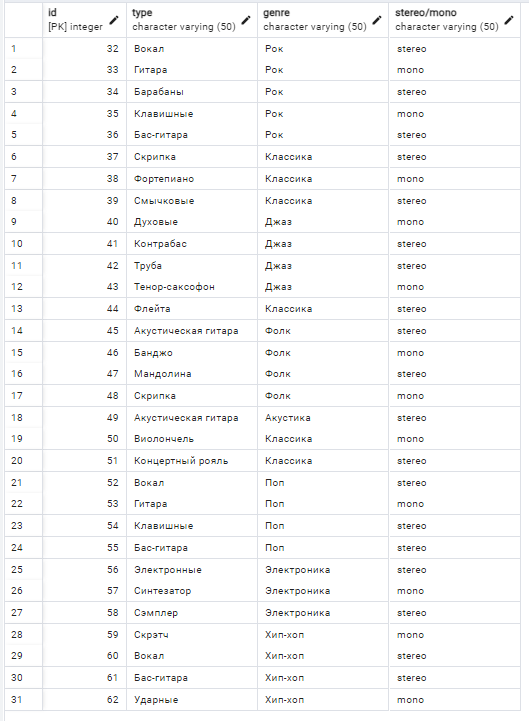


Рисунок 2.5 – Строки таблицы «RecordType»

1. Таблица Record:

INSERT INTO "Sound-Studio"."Record"("totalPrice", "hoursCount", "date", "recordTypeId", "employeeId", "roomId", "artistId")

VALUES

(250, 4, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(300, 3, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(200, 2, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(350, 5, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(270, 3, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(320, 4, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(280, 3, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(400, 6, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(260, 3, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(380, 5, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(290, 3, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(250, 4, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(300, 3, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(200, 2, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(350, 5, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(270, 3, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(320, 4, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(280, 3, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(400, 6, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(260, 3, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(380, 5, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(290, 3, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(250, 4, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(300, 3, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(200, 2, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(350, 5, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(270, 3, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(320, 4, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(280, 3, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(400, 6, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(260, 3, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(380, 5, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1),

(290, 3, CURRENT\_DATE - CAST(RANDOM() \* 365 AS INTEGER) \* INTERVAL '1 day', FLOOR(RANDOM() \* 31) + 32, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1, FLOOR(RANDOM() \* 30) + 1);

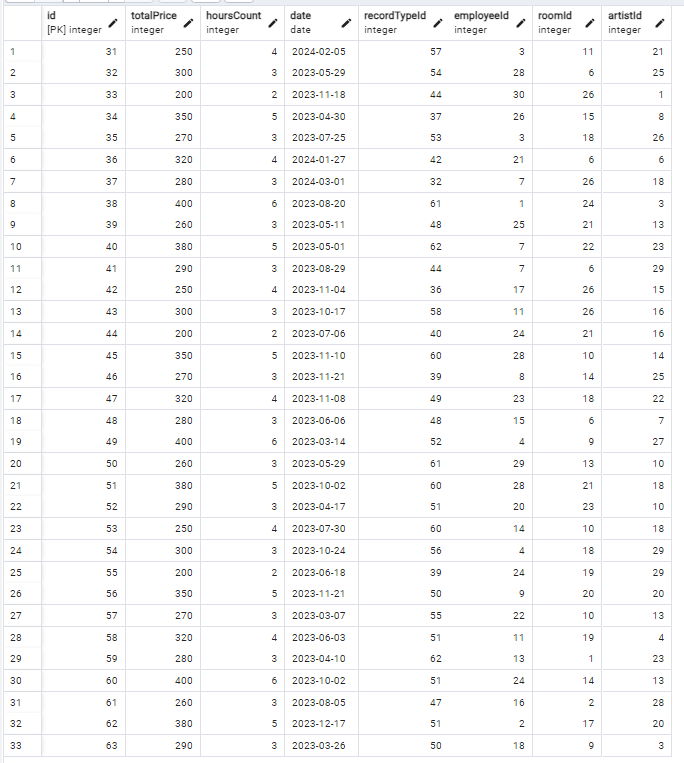


Рисунок 2.6 – Строки таблицы «Record»

1. Таблица Instrument\_Record:

INSERT INTO "Sound-Studio"."Instrument\_Record"("Instruments\_id", "Record\_id")

VALUES

(1, 38),

(5, 56),

(12, 43),

(7, 54),

(2, 50),

(25, 33),

(9, 60),

(17, 45),

(29, 41),

(8, 59),

(4, 55),

(16, 53),

(20, 37),

(22, 46),

(11, 51),

(30, 48),

(6, 44),

(10, 52),

(18, 42),

(26, 47),

(3, 34),

(21, 58),

(15, 36),

(27, 35),

(19, 39),

(13, 61),

(14, 57),

(28, 49),

(24, 62),

(23, 40);

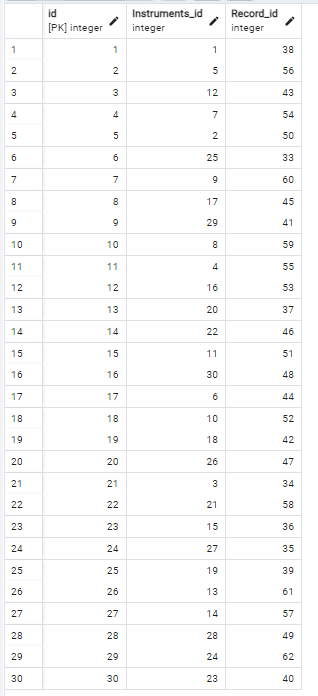


Рисунок 2.7 – Строки таблицы «Instrument\_Record»

# ДРУГИЕ SQL-КОМАНДЫ

Помимо расссмотренных SQL-команд *INSERT* и *CREATE TABLE* существуют другие команды на удаление, изменения таблиц. В моем случае я менял тип данный у столбца таблицы Artist используя SQL-команду *ALTER TABLE:*

ALTER TABLE "Sound-Studio"."Artist"

ALTER COLUMN genre TYPE VARCHAR(50);

# 4 СОЗДАНИЕ ВРЕМЕННОЙ ТАБЛИЦЫ

Создадим временную таблицу BlackList которая будет хранить одно поле Id пользователя

CREATE TEMP TABLE BlackList (id INT);

Вставим во временную таблицу тестовые данные

INSERT INTO BlackList(id) VALUES(1), (2), (3);

Сделаем выборку

SELECT \* FROM BlackList;

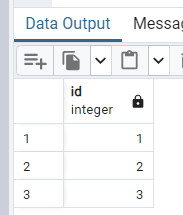


Рисунок 4.1 – Выборка данных из временной таблицы

Удалим временную таблицу.

DROP TABLE BlackList;

# 5 ГЕНЕРАЦИЯ ERD-ДИАГРАММЫ

После создания всех таблиц мы можем выгрузить структуру таблиц в виде UML диаграммы и сравнить ее с проделанной UML диаграммой из второй лабораторной работы. Выгрузим ERD диаграмму с помощью пункта меню ERD For Schema.

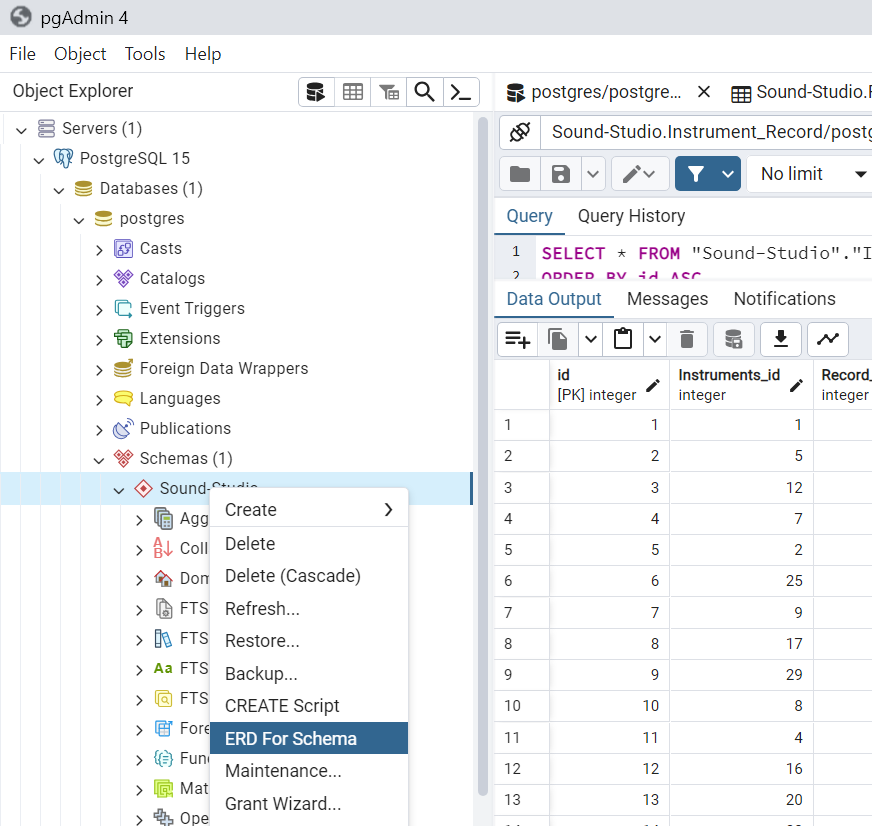


Рисунок 5.1 – Выгрузка диаграммы

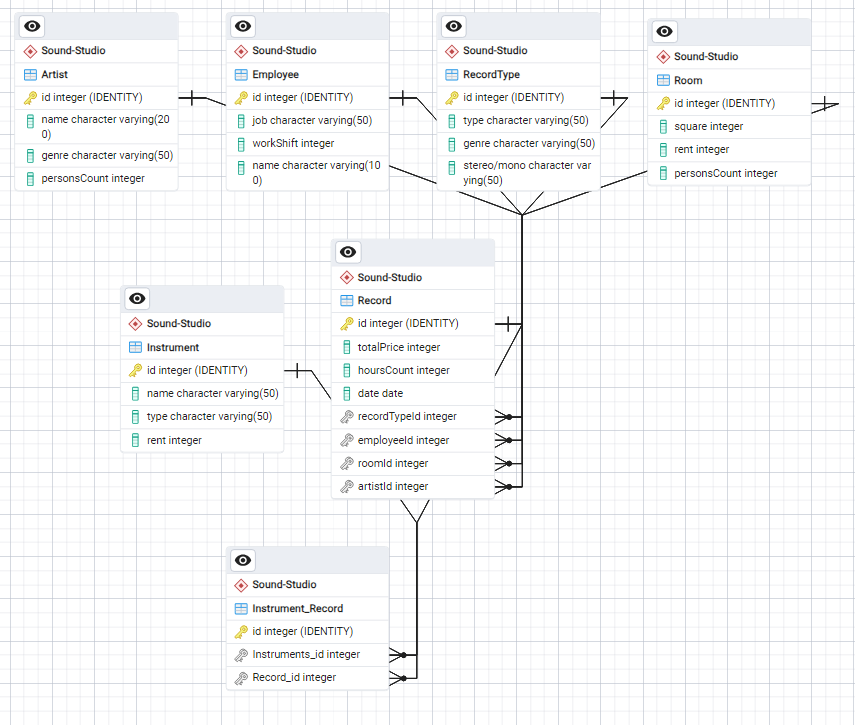


Рисунок 5.2 – Полученная EDR-диаграмма

# 6 ЭКСПОРТ РЕЗУЛЬТАТОВ

На случай восстановления БД или создания второй ее копии нужно скопировать точную структуру таблиц, для этого в PostgreAdmin есть специальное меню Schema -> Backup, там нужно выбрать имя файла и выбрать Format: Plain.

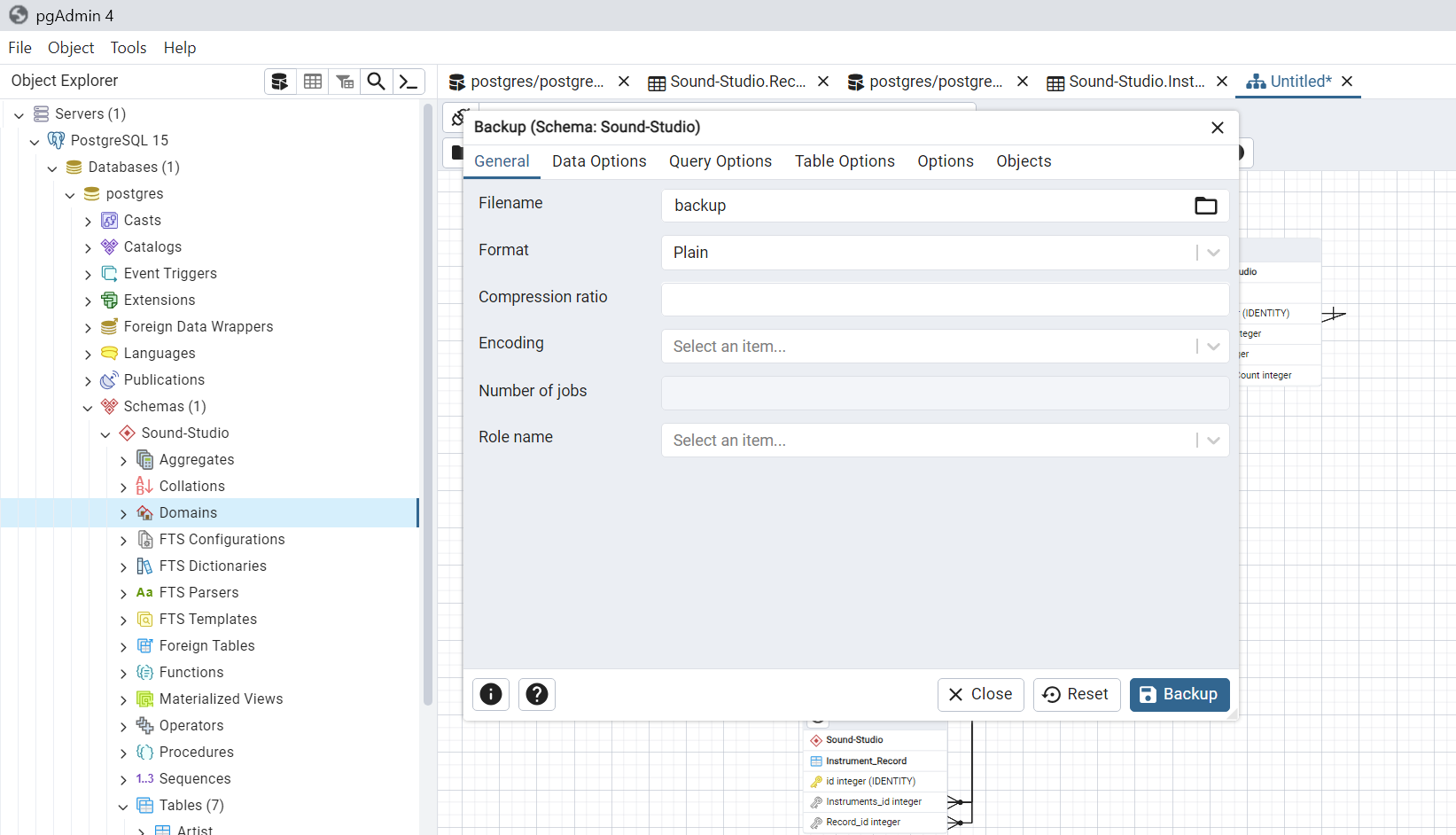


Рисунок 6.1 – Backup

# ЗАКЛЮЧЕНИЕ

В ходе данной лабораторной работы были изучены основы создания SQL-запросов в Postgres на базе «PgAdmin 4». Было выполнено сравнение ожидаемой ERD-диаграммы и сгенерированной путем написания SQL-запросов. Был сделан бэкап схемы, путем выгрузки ее в файл