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](#_Toc159241763)

[1 ER-ДИАГРАММА 5](#_Toc159241764)

[2 ВИД «БУМАЖНОГО» ПРЕОБРАЗОВАНИЯ 6](#_Toc159241765)

[3 ВИД «АВТОМАТИЧЕСКОГО» ПРЕОБРАЗОВАНИЯ 9](#_Toc159241766)

[ЗАКЛЮЧЕНИЕ 13](#_Toc159241767)

# ВВЕДЕНИЕ

В лабораторной работе выполняется реализация схемы базы данных по ранее построенной реляционной схеме данных (см. лабораторную работу

№2). Требуется сформировать SQL-запросы для создания таблиц базы данных и выполнить их в СУБД. Требуется заполнить таблицы данными с помощью оператора INSERT.

# СОЗДАНИЕ ТАБЛИЦ

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)

);

# ВИД «БУМАЖНОГО» ПРЕОБРАЗОВАНИЯ

Порядок перевода ER-модели в реляционную модель выполняется с

помощью алгоритма, состоящего из пяти шагов:

*Шаг 1.* Каждый объект на ER-диаграмме превращается в реляционное

отношение (далее для краткости – таблицу), имя объекта становится именем

таблицы. Можно выделить шесть таблиц со следующими именами:

«Инструмент», «Комната», «Артист», «Сотрудник», «Тип записи», «Запись».

*Шаг 2.* Каждый атрибут объекта становится столбцом с тем же именем.

*Шаг 3.* Уникальные атрибуты объекта превращаются в первичный ключ

таблицы. Таким образом были добавлены следующие первичные ключи: id

инструмента, id комнаты, id артиста, id типа записи, id записи. (рис. 2.1)

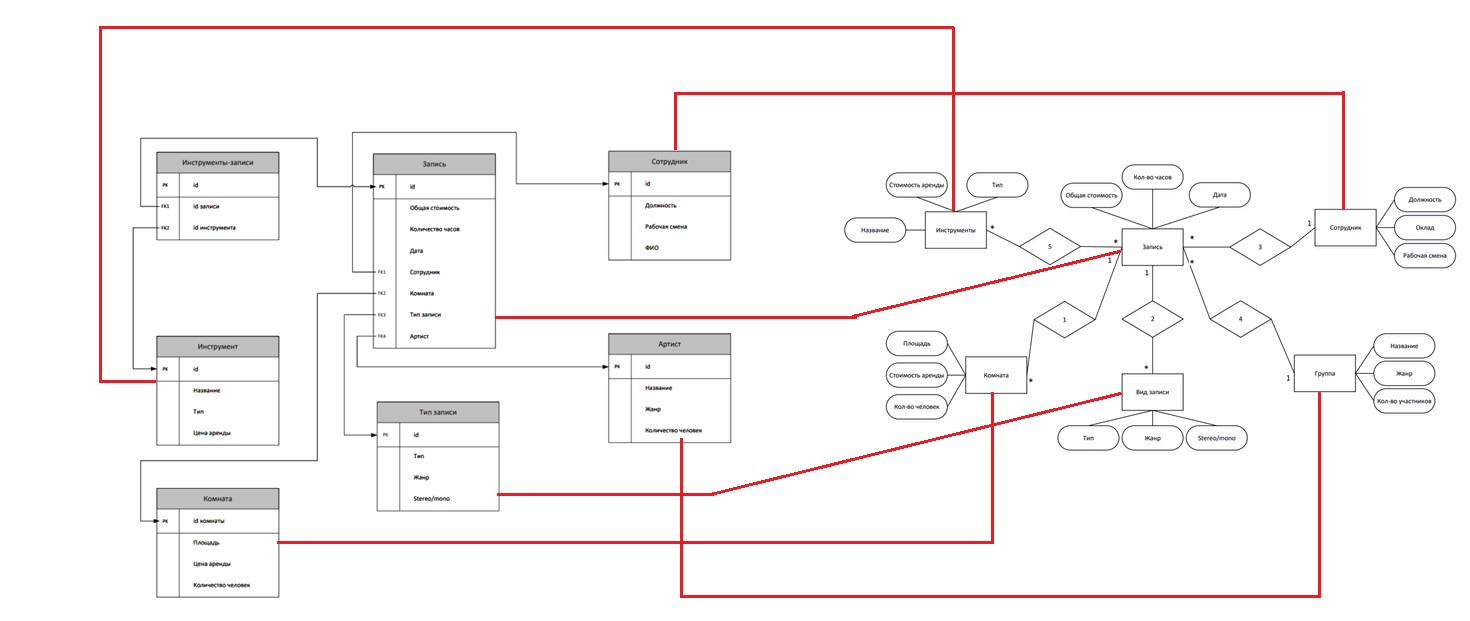


Рисунок 2.1 – сопоставление объектов URD и UML

*Шаг 4.* Связи «один-ко-многим» становятся ссылками в уже

существующих таблицах, при этом внешний ключ добавляется в виде столбца

в таблицу, соответствующую объекту со стороны «многие» связи. Внешние

ключи ссылаются на первичные ключи целевых таблиц. (рис. 2.2)

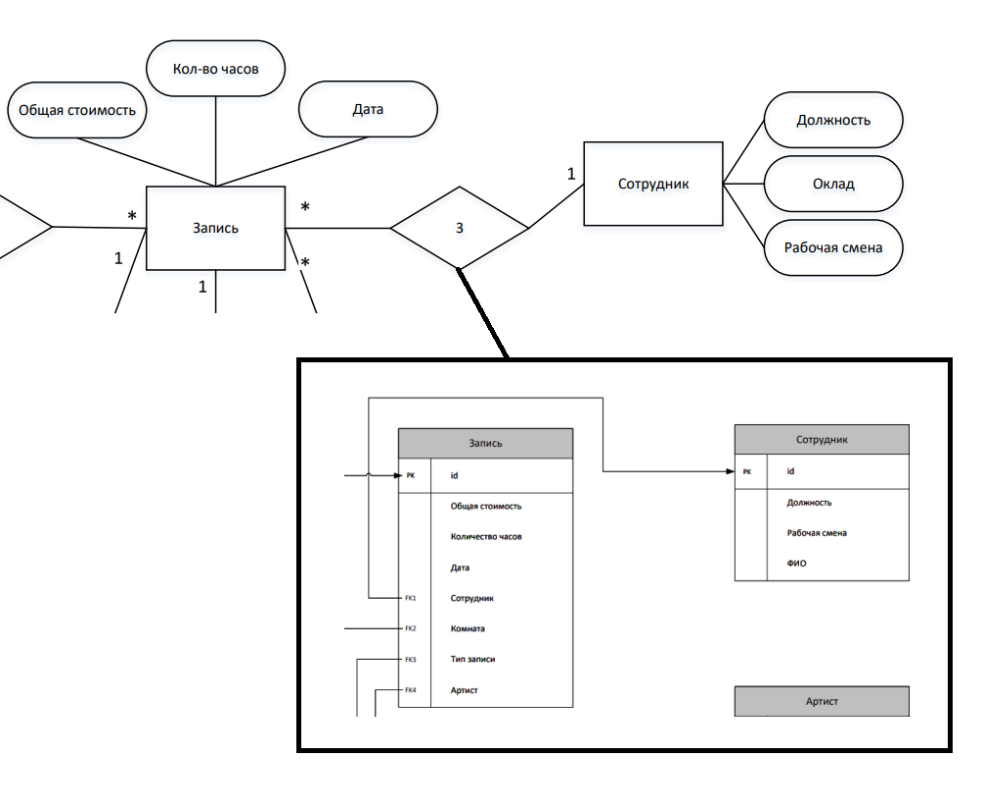


Рисунок 2.2 – Связь «один-ко-многим»

*Шаг 5.* Связи «многие-ко-многим» реализуются через отдельную

промежуточную таблицу. Была создана таблица «Инструменты-записи», в

которой находятся два поля внешних ключей: «id записи» и «id инструмента».

Первичный ключ в отношении «Инструменты-записи» построен как простой на основе суррогатного поля (PK).

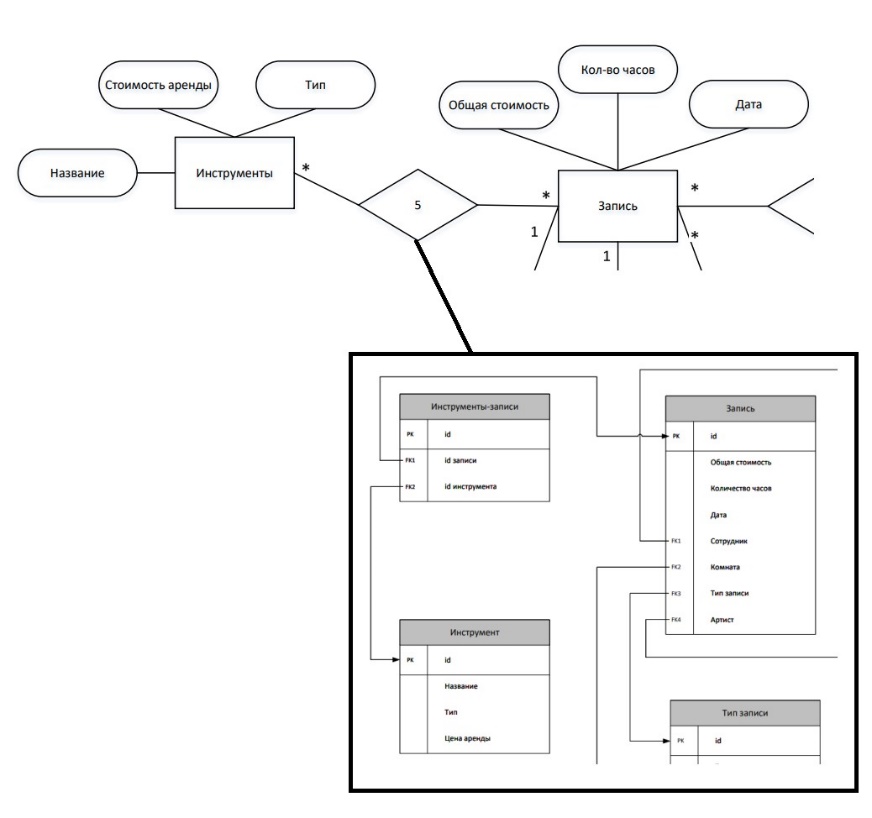


Рисунок 2.3 – Связь «многие-ко-многим»

UML-диаграмма реляционной схемы данных «бумажного» преобразования представлена на рисунке 2.4.

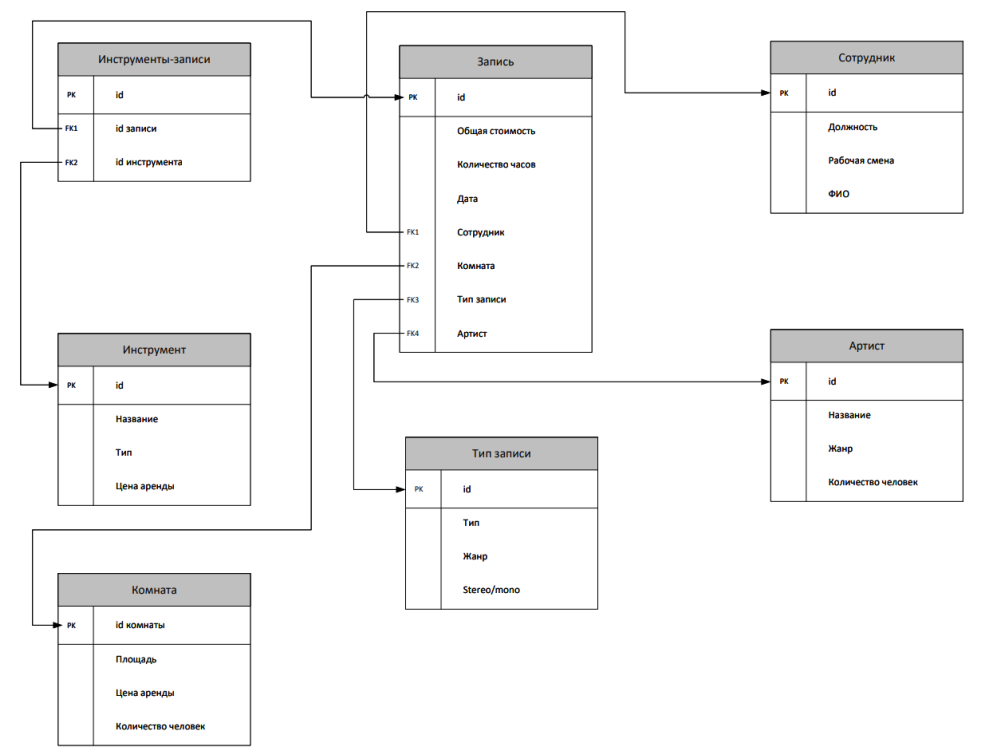


Рисунок 2.4 – UML-диаграмма

# 3 ВИД «АВТОМАТИЧЕСКОГО» ПРЕОБРАЗОВАНИЯ

Для перевода ER-диаграммы в реляционную диаграмму используется графический инструмента администрирования и проектирования баз данных – pgAdmin 4. Для проведения операций были выполнены следующие шаги:

*Шаг 1.* Открыть программу pgAdmin.

*Шаг 2.* Открыть вкладу Tools, а далее ERD Tool (рис. 3.1)

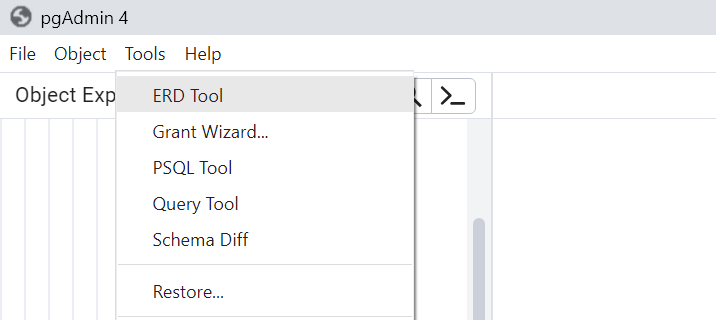


Рисунок 3.1

*Шаг 3.* В открывшейся зоне создать таблицу (рис. 3.2)

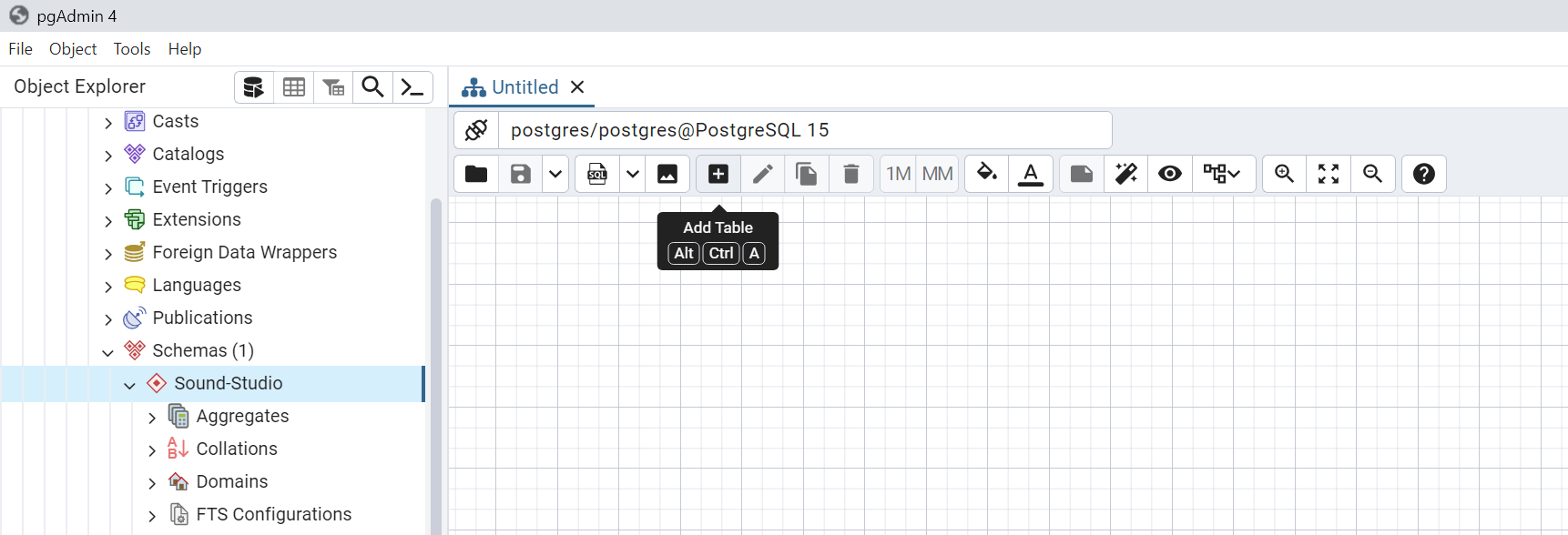


Рисунок 3.2

*Шаг 4.* Ввести имя таблицы и добавить необходимые колонки (рис. 3.3)

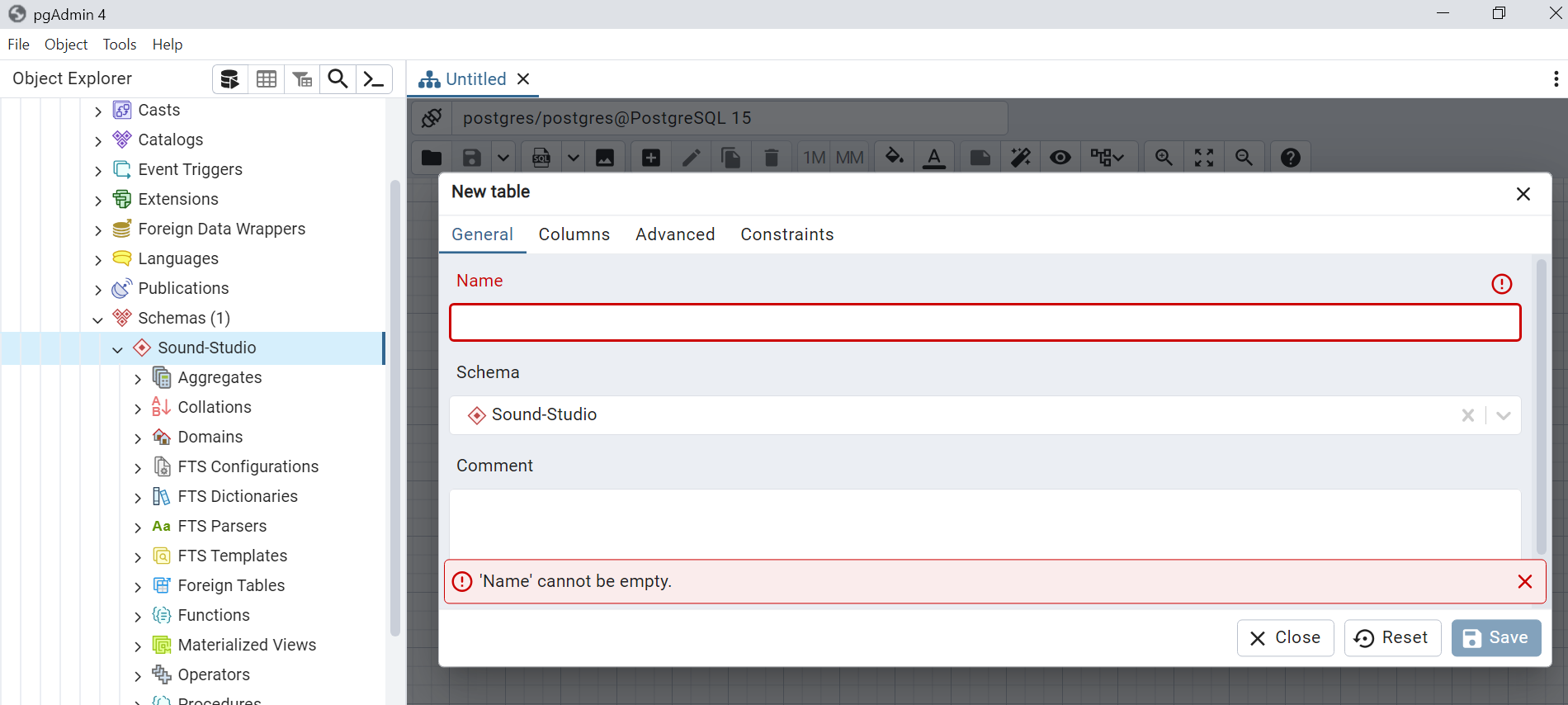


Рисунок 3.3

*Шаг 5.* Создать остальные нужные таблицы, и добавить связи между ними (рис. 3.4, рис 3.5)

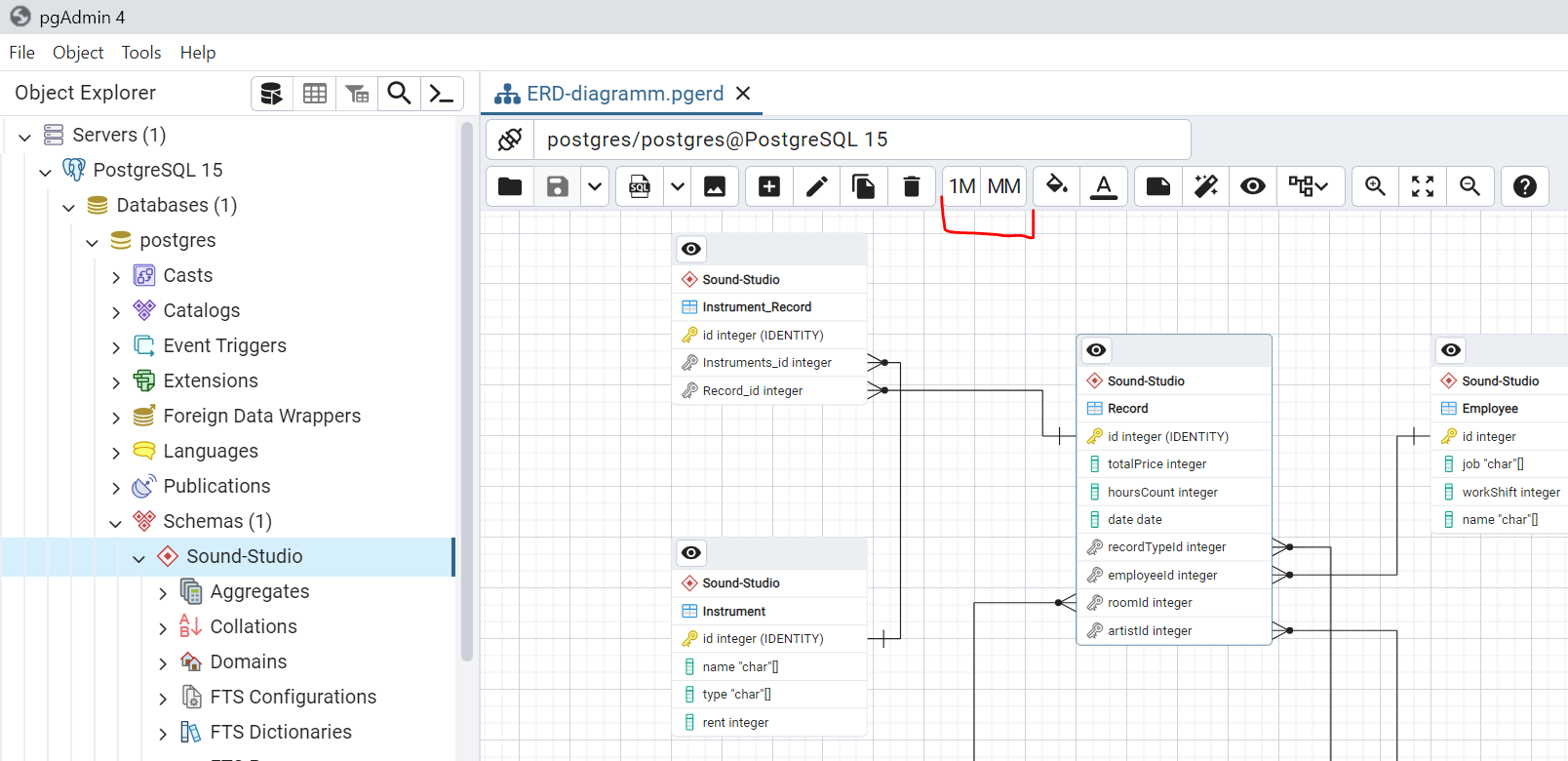


Рисунок 3.4

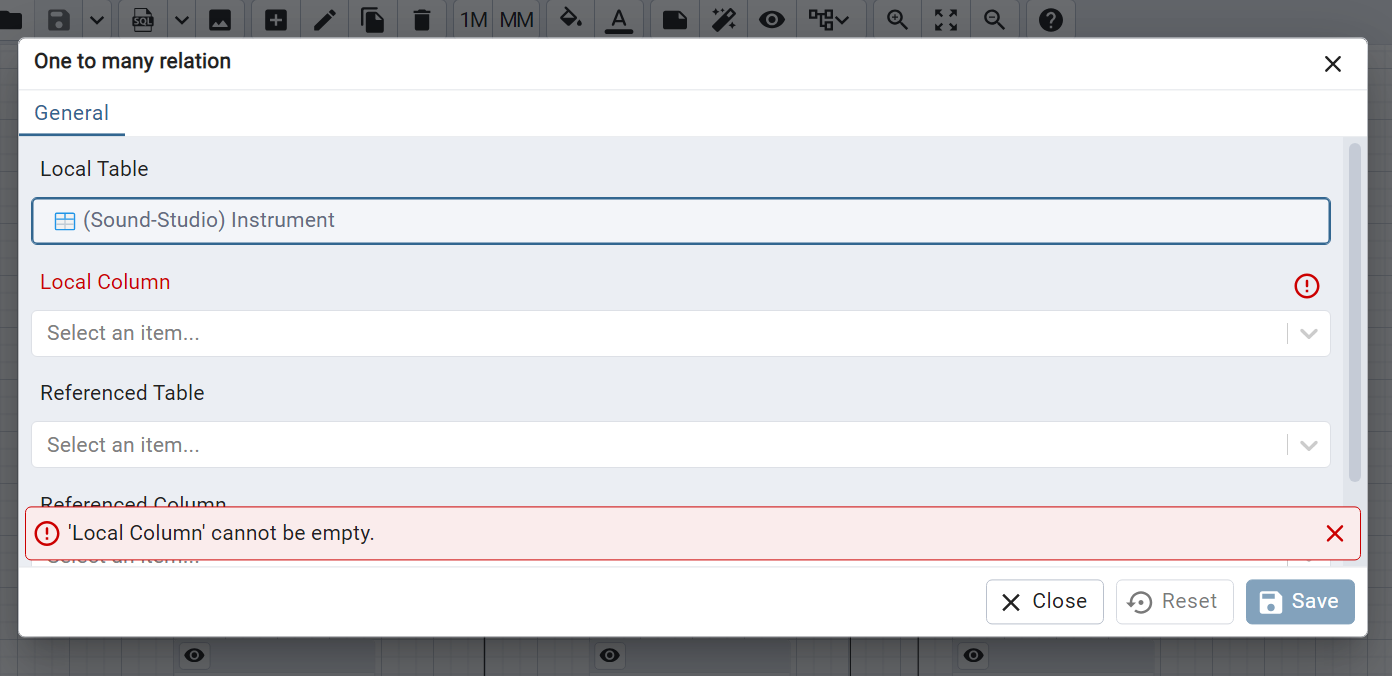


Рисунок 3.5

*Шаг 6.* Нажать на кнопку Generate SQL и выполнить SQL-код (рис. 3.6, рис 3.7)

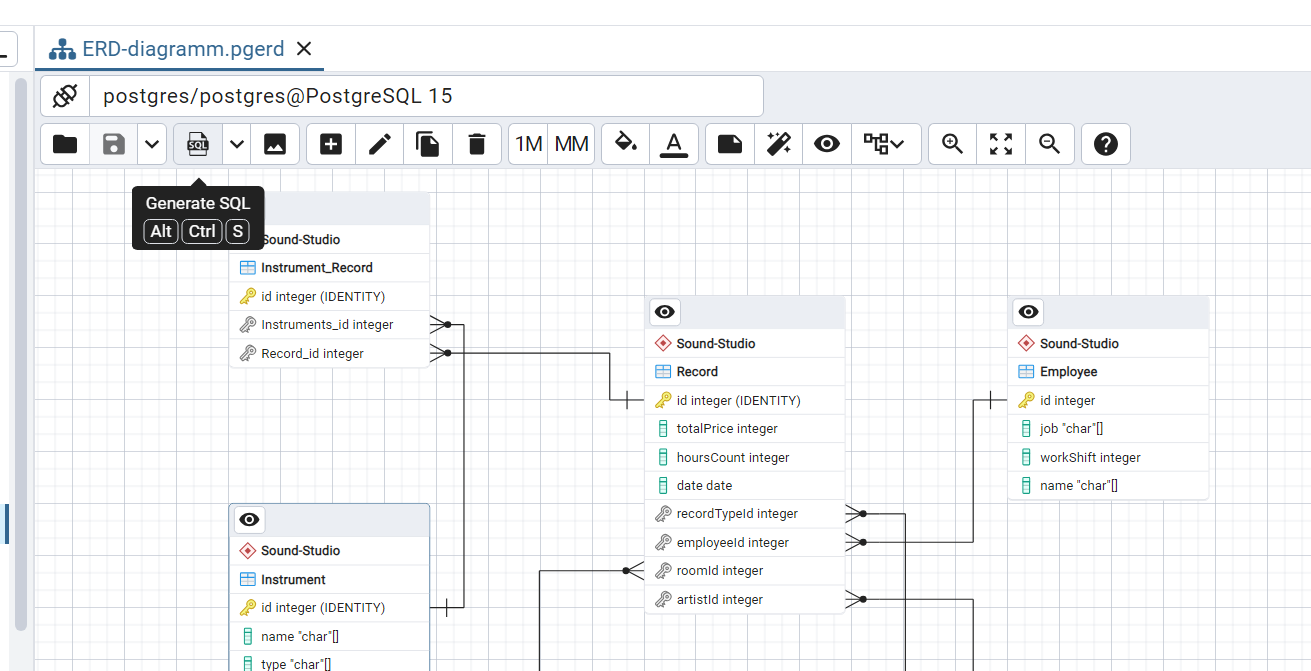


Рисунок 3.6

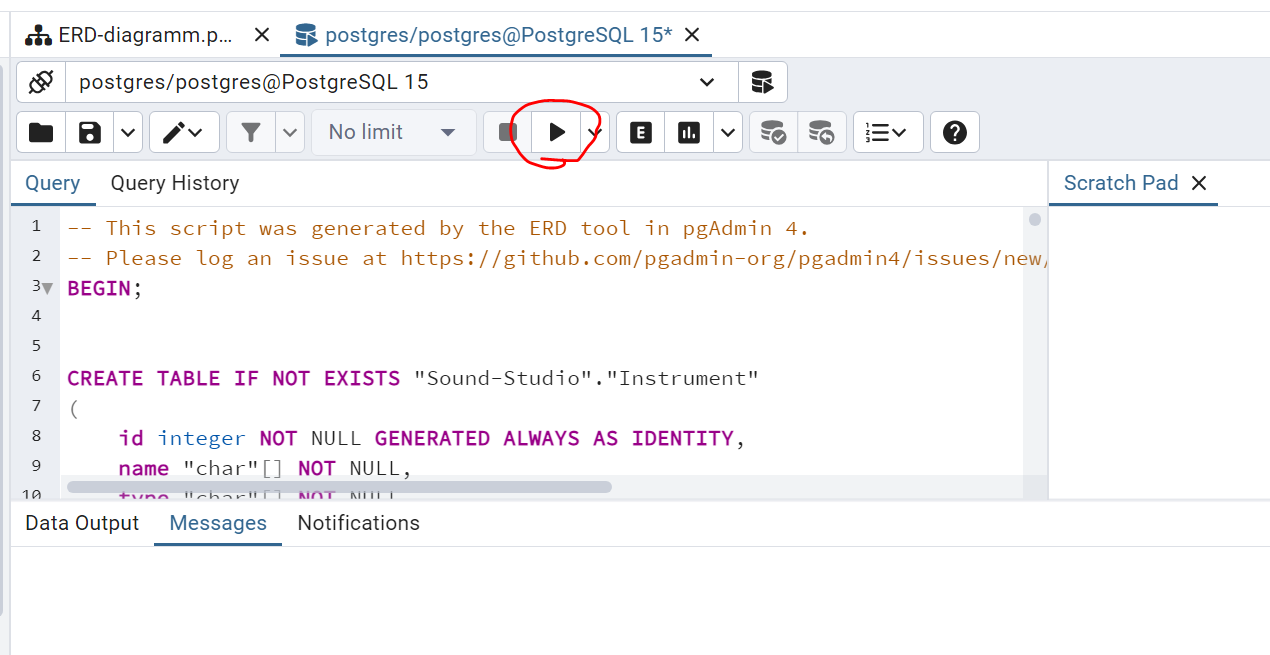


Рисунок 3.7

*Шаг 7.* Сохранить проект и ERD-диаграмму

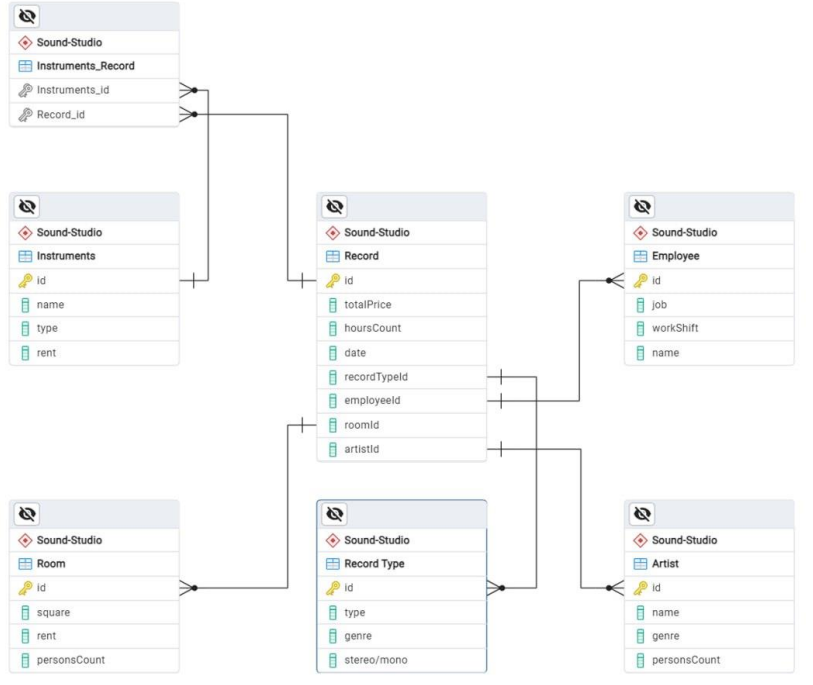


Рисунок 3.8 – ERD-диаграмма

# ЗАКЛЮЧЕНИЕ

Сравнивая результаты «автоматизированного» вида преобразования и «бумажного» ошибочных несоответствий обнаружено не было. Все связи, первичные и внешние ключи расставлены верно.

В ходе выполнения данной лабораторной работы было выполнено логическое проектированиеБД путем построения реляционной схемы данных по ранее спроектированной ER-модели. ER- диаграмму была преобразована в реляционную модель в двух вариантах: вид «бумажного» варианта преобразования и «автоматизированный».