

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ
Факультет компьютерных систем и сетей
Кафедра электронных вычислительных машин
Дисциплина: Базы данных

Тема «Репетиционная база»
Лабораторная работа №2
Создание реляционной схемы данных

Студент:
Преподаватель:

А.С. Бригадир
Д.В. Куприянова

МИНСК 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 ОПИСАНИЕ РЕЛЯЦИОННОЙ МОДЕЛИ.....	4
1.1 Описание таблиц и сущностей	4
1.2 Типы связей	5
2 СХЕМА ER-МОДЕЛЬ	6
3 ВИД «АВТОМАТИЧЕСКОГО» ПРЕОБРАЗОВАНИЯ	9
ЗАКЛЮЧЕНИЕ	13

ВВЕДЕНИЕ

Основная цель работы заключается в логическом проектировании базы данных путем построения реляционной схемы данных по ранее спроектированной в лабораторной работе №1 ER-модели и преобразовании ER-диаграммы в реляционную модель данных. Процесс преобразования модели будет разбит на два этапа: «бумажное» преобразование, где схема будет разработана вручную, и «автоматизированное» преобразование с использованием специализированных инструментов.

Каждый этап требует тщательного анализа, так как важно не только создать корректную модель, но и убедиться в том, что все связи между сущностями и атрибутами правильно реализованы.

Важным аспектом работы является сравнение результатов обоих методов преобразования. Это позволит выявить возможные расхождения и ошибки, а также даст возможность провести анализ и устранить несоответствия, что обеспечит целостность и корректность проектируемой базы данных.

1 ОПИСАНИЕ РЕЛЯЦИОННОЙ МОДЕЛИ

Исходное задание: создать реляционную схему данных, преобразовать ER-диаграмму в реляционную модель данных, нормализовать реляционные отношения.

1.1 Описание таблиц и сущностей

В рамках реляционной модели «Репетиционная база» предусмотрены следующие таблицы и соответствующие им сущности.

Таблица Репточка соответствует сущности репетиционной точки и включает поля: id репточки (уникальный идентификатор репточки), рейтинг, телефон, график, название, адрес.

Таблица Комната отражает сущность комнаты и содержит поля: id комнаты (уникальный идентификатор комнаты), кондиционер (наличие кондиционера), прайс, поддержка записи, площадь, id репточки (ссылка на репточку, к которой относится комната, как внешний ключ).

Таблица Бронь соответствует сущности записи о бронировании и включает поля: id брони (уникальный идентификатор записи о бронировании), время, стоимость, дата создания, статус, количество людей, id комнаты (ссылка на комнату, которая забронирована, как внешний ключ), id пользователя (ссылка на пользователя, который оформил бронь, как внешний ключ).

Таблица Сотрудник отражает сущность работника репетиционной точки и включает поля: id сотрудника (уникальный идентификатор сотрудника), ФИО, должность, стаж, телефон, возраст, id репточки (ссылка на репточку, в которой работает сотрудник, как внешний ключ).

Таблица Услуга отражает сущность оказываемых репточкой услуг содержит поля: id услуги (уникальный идентификатор услуги), название, прайс, тип, требования, id репточки (ссылка на репточку, к которой относится услуга, как внешний ключ).

Таблица Оборудование отражает сущность оборудования, относящегося к репточке, и включает поля: id оборудования (уникальный идентификатор оборудования), название, тип, марка, модель, состояние, id репточки (ссылка на репточку, к которой относится оборудование, как внешний ключ).

Таблица Пользователь соответствует сущности пользователя и содержит поля: id пользователя (уникальный идентификатор пользователя), ФИО, телефон, почта, дата регистрации.

Таблица Услуга-Бронь является промежуточной и используется для реализации связи многие-ко-многим между услугами и бронированиями. Она включает поля: id услуги (ссылка на услугу как внешний ключ), id брони (ссылка на бронь как внешний ключ).

Таблица Оборудование-Бронь является промежуточной и используется для реализации связи многие-ко-многим между оборудованием и

бронированиями. Она включает поля: id оборудования (ссылка на оборудование как внешний ключ), id брони (ссылка на бронь как внешний ключ).

1.2 Типы связей

1. Связь «Репточка – Комната» (один-ко-многим): каждая репетиционная студия может включать несколько комнат, но каждая комната относится только к одной студии.

2. Связь «Комната – Бронь» (один-ко-многим): каждая комната может быть забронирована множество раз, но одно бронирование относится только к одной комнате.

3. Связь «Репточка – Сотрудник» (один-ко-многим): в одной репетиционной студии работают несколько сотрудников, но каждый сотрудник относится только к одной студии.

4. Связь «Репточка – Услуга» (один-ко-многим): одна репточка может предоставлять несколько услуг, но каждая услуга относится к одной репточке.

5. Связь «Бронь – Услуга» (многие-ко-многим): одно бронирование может включать несколько услуг, и одна услуга может быть добавлена к нескольким бронированиям.

6. Связь «Репточка – Оборудование» (один-ко-многим): каждый элемент оборудования относится только к одной репточке, а репточка включает различное оборудование.

7. Связь «Бронь – Оборудование» (многие-ко-многим): каждая бронь может включать аренду различного оборудования, каждый элемент оборудования может относиться к разным бронированиям.

8. Связь «Пользователь – Бронь» (один-ко-многим): один клиент может иметь несколько бронирований, но каждое бронирование связано только с одним клиентом.

2 СХЕМА ER-МОДЕЛЬ

Схема ER-диаграмма представлена на рисунке 1.

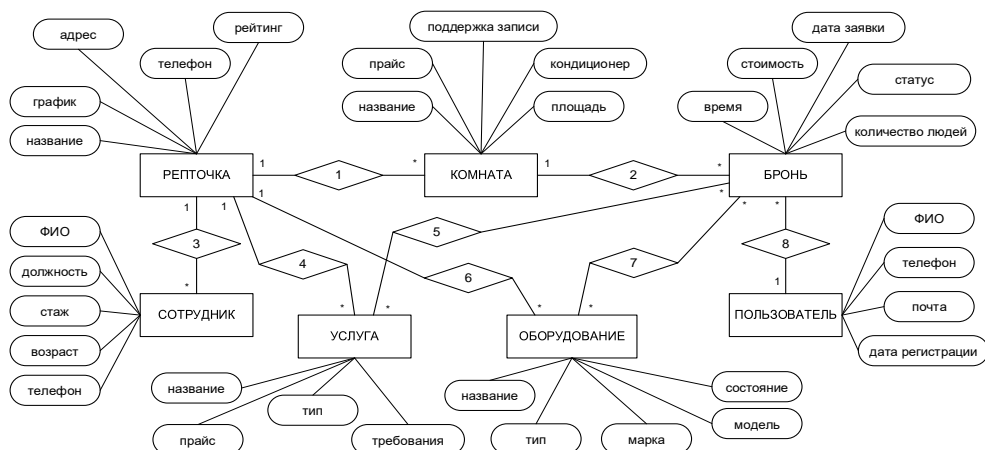


Рисунок 1 – ER-диаграмма

Порядок перевода ER-модели в реляционную модель выполняется с помощью алгоритма, состоящего из пяти шагов:

1. Каждый объект на ER-диаграмме превращается в реляционное отношение (далее для краткости – таблицу), имя объекта становится именем таблицы. Можно выделить семь таблиц со следующими именами: «Репточка», «Сотрудник», «Комната», «Услуга», «Оборудование», «Пользователь», «Бронь».

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

3. Уникальные атрибуты объекта превращаются в первичный ключ таблицы. Таким образом были добавлены следующие первичные ключи: id репточки, id сотрудника, id комнаты, id услуги, id оборудования, id брони, id пользователя. (рис. 2.1)

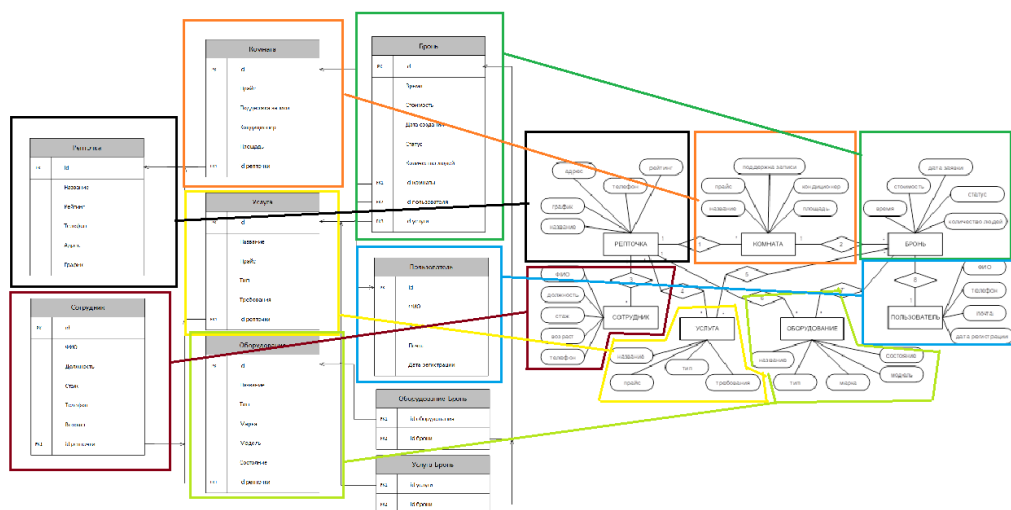


Рисунок 2.1 – сопоставление объектов URD и UML

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 связи «многие». Внешние ключи ссылаются на первичные ключи целевых таблиц. (рис. 2.2)

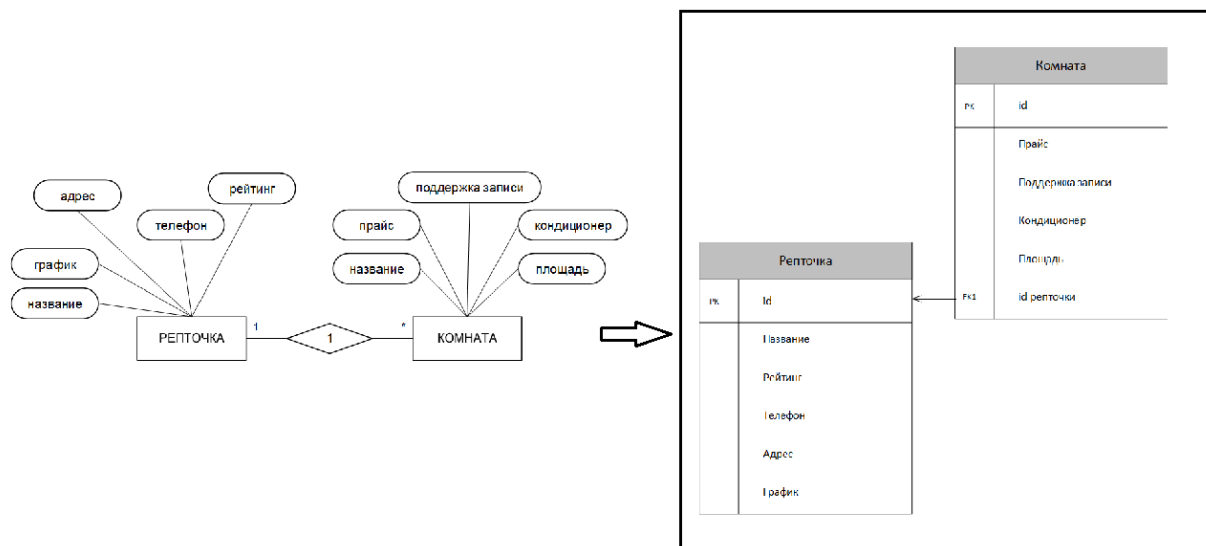


Рисунок 2.2 – Связь «один-ко-многим»

5. Связи «многие-ко-многим» реализуются через отдельную промежуточную таблицу. Была создана таблица «Оборудование-Бронь», в которой находятся два поля внешних ключей: «id оборудования» и «id брони».

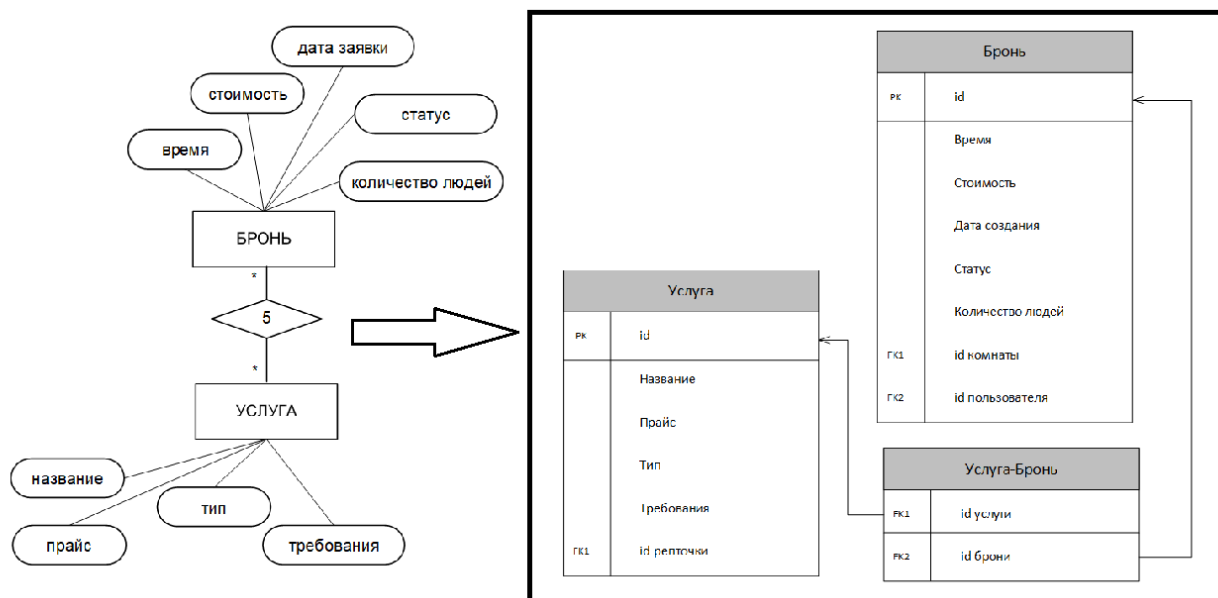


Рисунок 2.3 – Связь «многие-ко-многим»

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

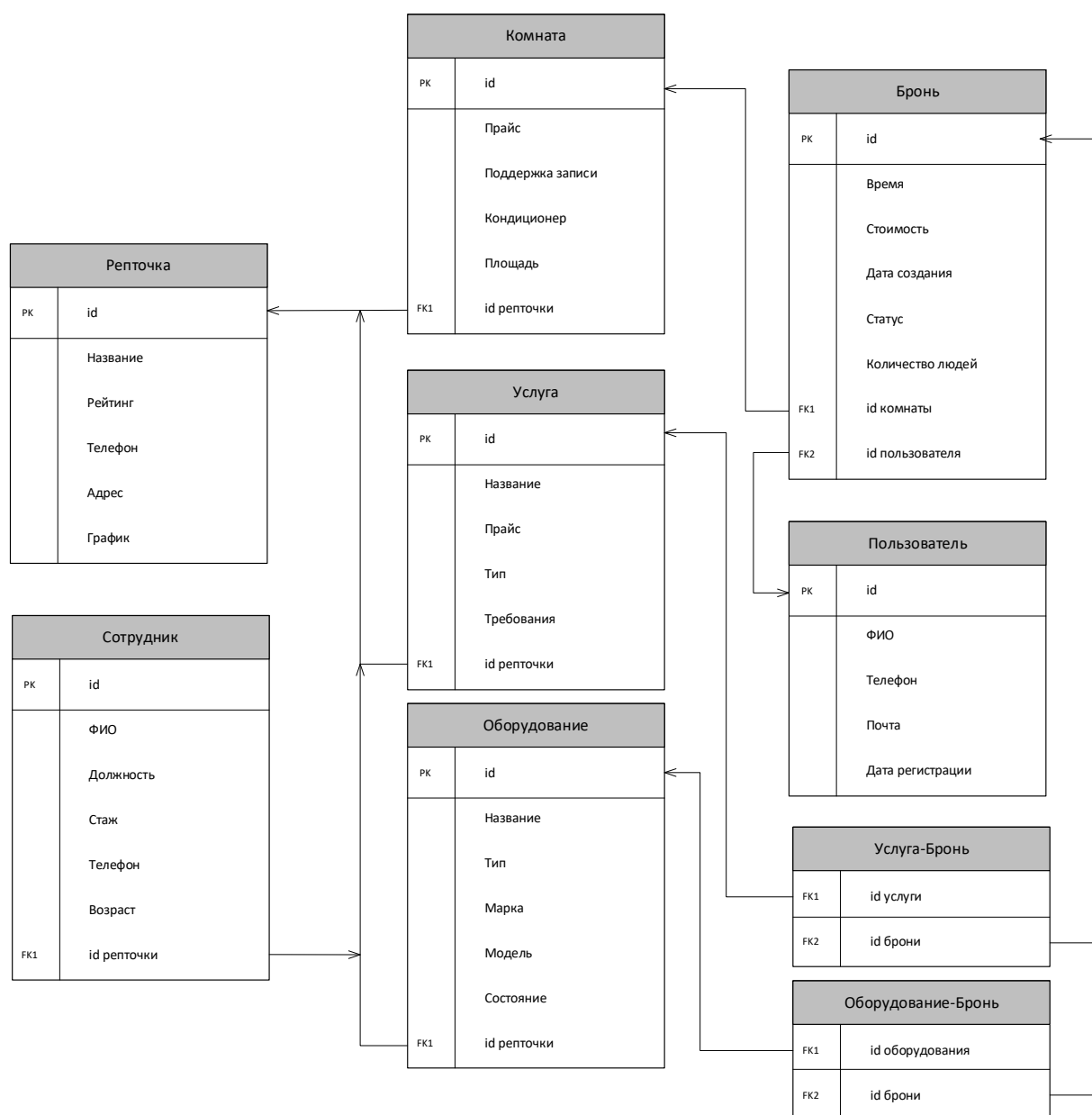


Рисунок 2.4 – UML-диаграмма

3 ВИД «АВТОМАТИЧЕСКОГО» ПРЕОБРАЗОВАНИЯ

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

1. Открыть программу pgAdmin.
2. Открыть вкладку Tools, а далее ERD Tool (рис. 3.1)

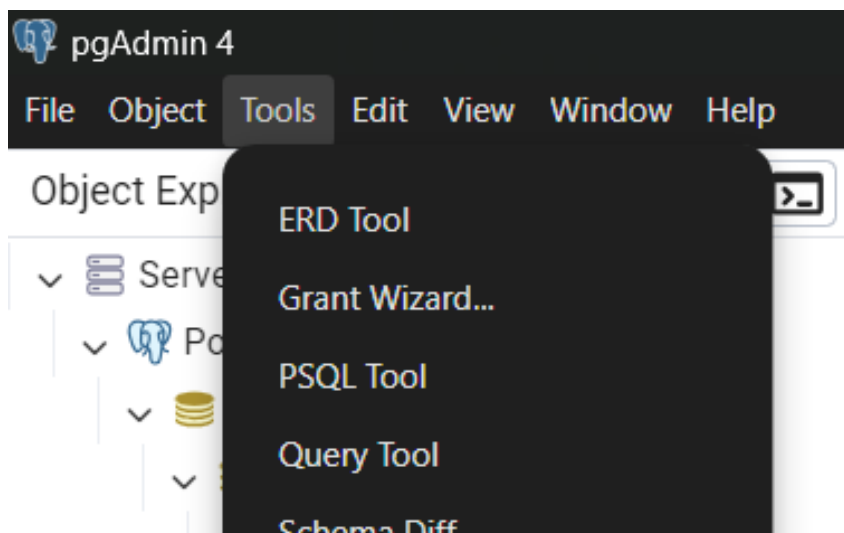


Рисунок 3.1 – Вкладка Tools

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

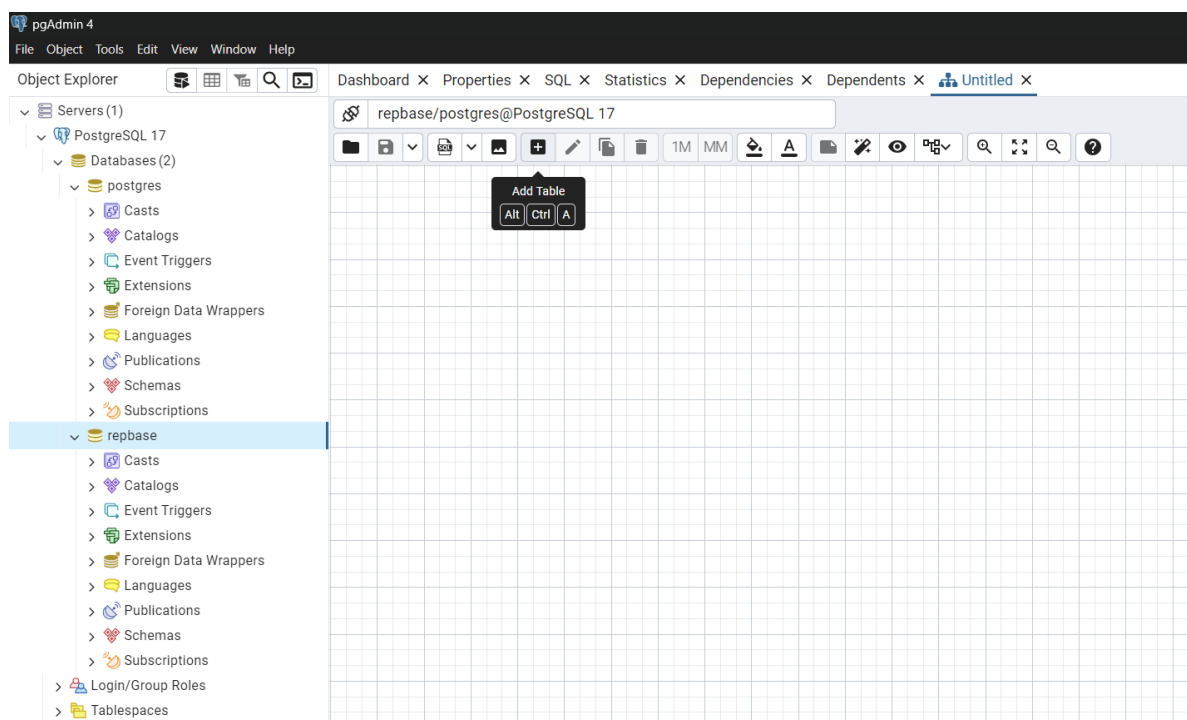


Рисунок 3.2 – Создание таблицы

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

New table

General Columns Advanced Constraints

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	name	character varying	100		<input type="checkbox"/>	<input type="checkbox"/>	
	address	character varying	100		<input type="checkbox"/>	<input type="checkbox"/>	
	phone	character varying	15		<input type="checkbox"/>	<input type="checkbox"/>	
	schedule	character varying	150		<input type="checkbox"/>	<input type="checkbox"/>	
	rating	real			<input type="checkbox"/>	<input type="checkbox"/>	

Close Reset Save

Рисунок 3.3 – Добавление колонок и имени таблицы

5. Создать остальные нужные таблицы и добавить связи между ними (рис 3.4)

One to many relation

General

Local Table (public) equipment

Local Column id_rehearsal_point

Referenced Table (public) rehearsal_points

Referenced Column id

Close Reset Save

Рисунок 3.4 – Добавление связей

6. Нажать на кнопку Generate SQL и выполнить SQL-код (рис. 3.5)

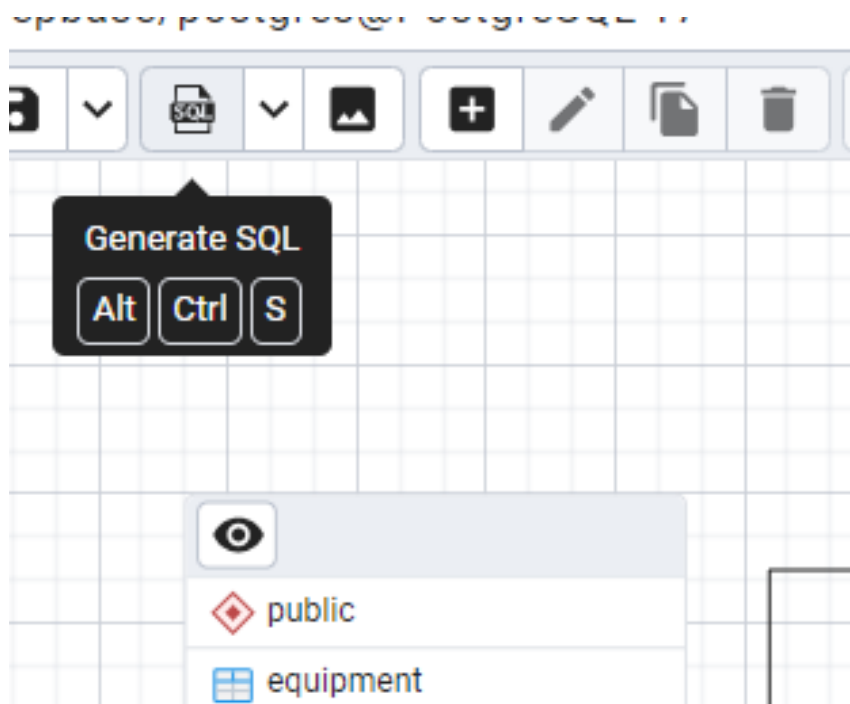


Рисунок 3.5 – Генерация SQL кода

На рисунке 3.6 показан SQL код для создание реляционной модели.

```
Dashboard x Properties x SQL x Statistics x Dependencies x Dependents x Untitled* x rebase/postgres@PostgreSQL 17* x
rebase/postgres@PostgreSQL 17
No limit
Query Query History Scratch Pad x
1 -- This script was generated by the ERD tool in pgAdmin 4.
2 -- Please log an issue at https://github.com/pgadmin-org/pgadmin4/issues/new/choose if you find any bugs, inclu
3 BEGIN;
4
5
6 CREATE TABLE IF NOT EXISTS public.rehearsal_points
7 (
8     id integer,
9     rating real,
10    phone character varying(15),
11    schedule json,
12    name text,
13    address text,
14    PRIMARY KEY (id)
15 );
16
17 CREATE TABLE IF NOT EXISTS public.rooms
18 (
19     id integer,
20     air_conditioner boolean,
21     price integer,
22     recording_support boolean,
23     area integer,
24     id_rehearsal_point integer,
25     PRIMARY KEY (id)
26 );
27
28 CREATE TABLE IF NOT EXISTS public.booking
29 (
30     id integer,
31     time_start time without time zone,
32     time_end time without time zone,
33     cost integer.
```

Рисунок 3.6 – SQL код

7. Сохранить проект и ERD-диаграмму (рис. 3.7)

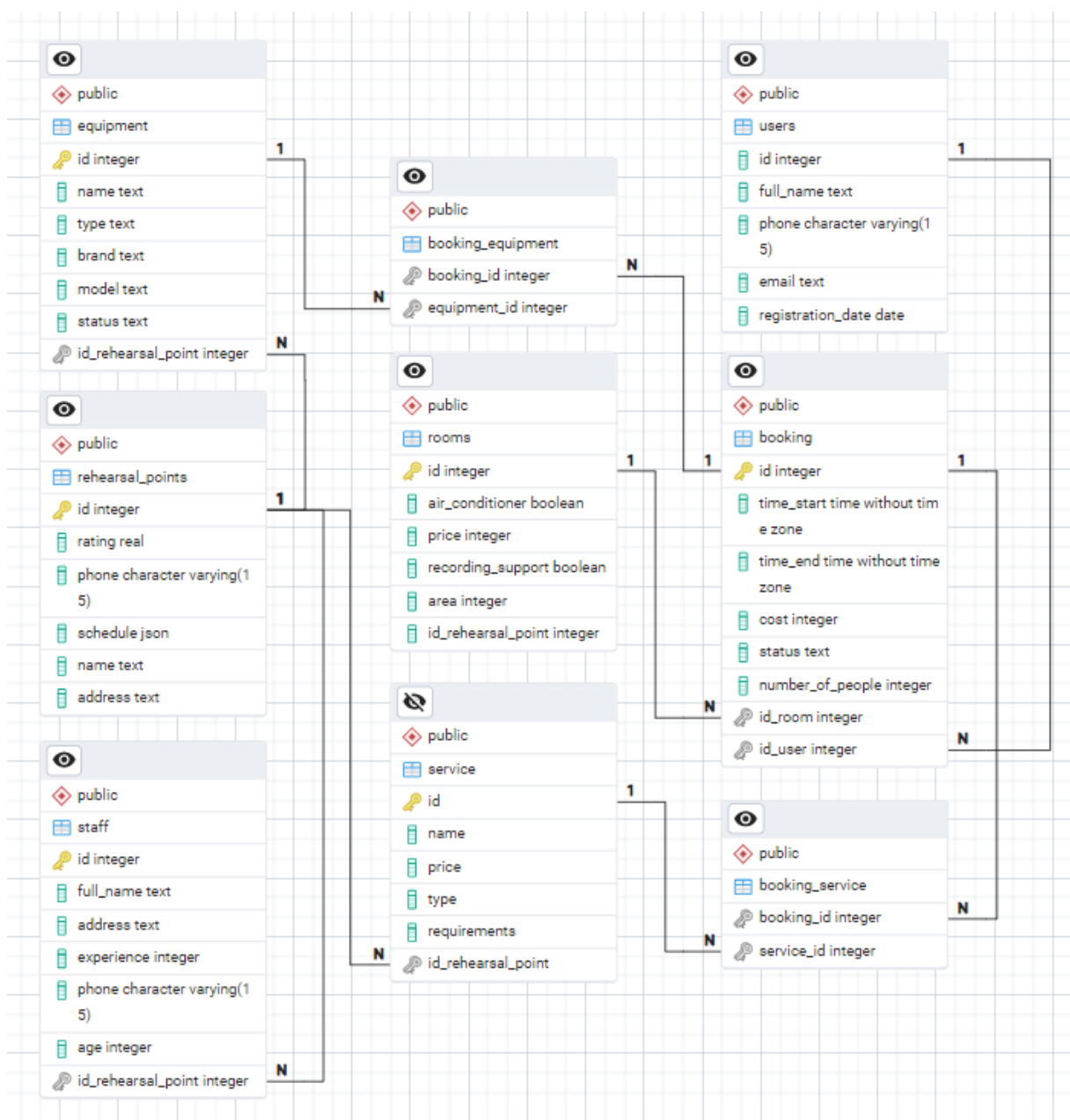


Рисунок 3.7 – ERD-диаграмма

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы была успешно реализована логическая схема проектирования базы данных «Репетиционная база». Оба этапа преобразования – «бумажное» и «автоматизированное» – были выполнены с соблюдением всех необходимых шагов и стандартов, что позволило создать корректную реляционную модель. Сравнение результатов показало, что все связи, первичные и внешние ключи были расставлены верно, и ошибок в данных не обнаружено.