

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей  
Кафедра электронных вычислительных машин  
Дисциплина: Базы данных

ОТЧЁТ  
по лабораторной работе №2  
«Создание реляционной схемы данных»  
на тему  
«Континентальная хоккейная лига»

Студент

М.А. Бекетова

Преподаватель

Д.В. Куприянова

Минск 2025

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 ОПИСАНИЕ РЕЛЯЦИОННОЙ МОДЕЛИ «КОНТИНЕНТАЛЬНАЯ ХОККЕЙНАЯ ЛИГА» .....	4
1.1 Описание таблицы и сущностей .....	4
1.2 Типы связей .....	4
2 СХЕМА ER-МОДЕЛЬ .....	6
3 ВИД «АВТОМАТИЧЕСКОГО» ПРЕОБРАЗОВАНИЯ .....	9
ЗАКЛЮЧЕНИЕ .....	12

## ВВЕДЕНИЕ

Цель данной работы — преобразование ER-диаграммы в реляционную модель данных, что включает разработку реляционной схемы и нормализацию отношений. Это необходимо для обеспечения структурированной и эффективной организации данных, их целостности и удобства работы с базой.

Процесс проектирования разделяется на два этапа. На первом этапе выполняется ручное («бумажное») преобразование, в ходе которого ER-диаграмма переводится в реляционную схему вручную. Этот процесс требует анализа структуры данных, выделения сущностей, их атрибутов и взаимосвязей, а также определения первичных и внешних ключей.

На втором этапе выполняется автоматизированное преобразование с использованием специализированных инструментов, таких как pgAdmin, для работы с базами данных. Этот подход позволяет упростить процесс проектирования, минимизировать вероятность ошибок и автоматически генерировать SQL-код для создания таблиц и связей.

Ключевая часть работы – сравнение результатов обоих методов преобразования. Это позволит выявить возможные расхождения и при необходимости внести коррективы в реляционную схему. Такой анализ не только обеспечивает целостность и корректность проектируемой базы данных, но и даёт возможность оценить преимущества и недостатки каждого из методов преобразования.

Таким образом, данная работа направлена на детальное изучение процесса перехода от ER-моделирования к реляционной схеме, выявление потенциальных проблем на разных этапах проектирования и выработку наилучших решений для создания разработки базы данных.

# **1 ОПИСАНИЕ РЕЛЯЦИОННОЙ МОДЕЛИ «КОНТИНЕНТАЛЬНАЯ ХОККЕЙНАЯ ЛИГА»**

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

## **1.1 Описание таблицы и сущностей**

В рамках реляционной модели «Континентальная хоккейная лига» предусмотрены следующие таблицы и соответствующие им сущности.

Таблица Клуб соответствует сущности клуба и включает поля: id клуба (уникальный идентификатор клуба), название, город, год основания, конференция.

Таблица Игрок отражает сущность игрока и содержит поля: id игрока (уникальный идентификатор игрока), ФИО, возраст, страна, номер, амплуа, хват, очки.

Таблица Матч соответствует сущности матча и включает поля: id матча (уникальный идентификатор матча), дата, тип завершения, счет, статус.

Таблица Арена представлена сущностью типа арены и содержит поля: id арены (уникальный идентификатор арены), название, город, год постройки, вместимость.

Таблица Арбитр соответствует сущности арбитра и включает поля: id арбитра (уникальный идентификатор арбитра), ФИО, номер, возраст, амплуа.

Таблица Тренер отражает сущность тренера и содержит поля: id тренера (уникальный идентификатор тренера), ФИО, возраст, страна, количество игр.

Таблица Статистика-клуба соответствует сущности статистики клуба и включает поля: id статистики клуба (уникальный идентификатор статистики клуба), игры, выигрыши, выигрыши ОТ, выигрыши ПБ, шайбы, очки.

Также предусмотрена таблица Клуб-Матч, которая является промежуточной и используется для реализации связи многие-ко-многим. Она включает поля: id клуба (ссылка на клуб, как внешний ключ), id матча (ссылка на матч, как внешний ключ), роль и результат.

Еще одна таблица – таблица Матч-Арбитр, промежуточная, для реализации связи многие-ко-многим. Она включает поля: id арбитра (ссылка на арбитра, как внешний ключ) и id матча (ссылка на матч, как внешний ключ).

Также таблица Матч-Игрок, промежуточная, для реализации связи многие-ко-многим. Она включает поля: id игрока (ссылка на игрока – внешний ключ), id матча (ссылка на матч – внешний ключ), голы, передачи и штрафы.

## **1.2 Типы связей**

1. Связь «Клуб – Игрок» (один-ко-многим): один клуб может иметь много игроков, но игрок принадлежит только одному клубу.

2. Связь «Клуб – Матч» (многие-ко-многим): в каждом матче участвуют два клуба, в это же время один клуб принимает участие во множестве матчей.

Атрибуты связи: «роль» (роль команды в матче: гости или хозяева) и «результат» (результат команды в матче: выигрыш или поражение).

3. Связь «Арена – Матч» (один-ко-многим): каждый матч проходит на одной арене, а одна арена может принимать множество матчей.

4. Связь «Матч – Арбитр» (многие-ко-многим): на один матч назначаются несколько арбитров, в то же время отдельно взятый арбитр может обслуживать много матчей.

5. Связь «Матч – Игрок» (многие-ко-многим): в каждом матче участвует множество игроков из обоих клубов, с другой стороны, отдельно взятый игрок может участвовать во многих матчах.

Атрибуты связи: «голы» (количество голов, забитых игроком), «передачи» (количество результативных передач, выполненных игроком) и «штрафы» (количество штрафов, заработанных игроком).

6. Связь «Клуб – Тренер» (один-ко-многим): у одного клуба может быть несколько тренеров, в то время как отдельно взятый тренер принадлежит только одному клубу в определенный момент времени.

7. Связь «Клуб – Статистика клуба» (один-к-одному): каждая запись статистики относится только к одному клубу и у одного клуба есть только одна запись статистики.

8. Связь «Клуб – Арена» (один-к-одному): один клуб имеет одну домашнюю арену в определенный момент времени, также как и одна арена может быть домашней лишь для одного клуба в определенный момент времени.

## 2 CXEMA ER-MODEЛЬ

Схема ER-диаграмма представлена на рисунке 2.1.

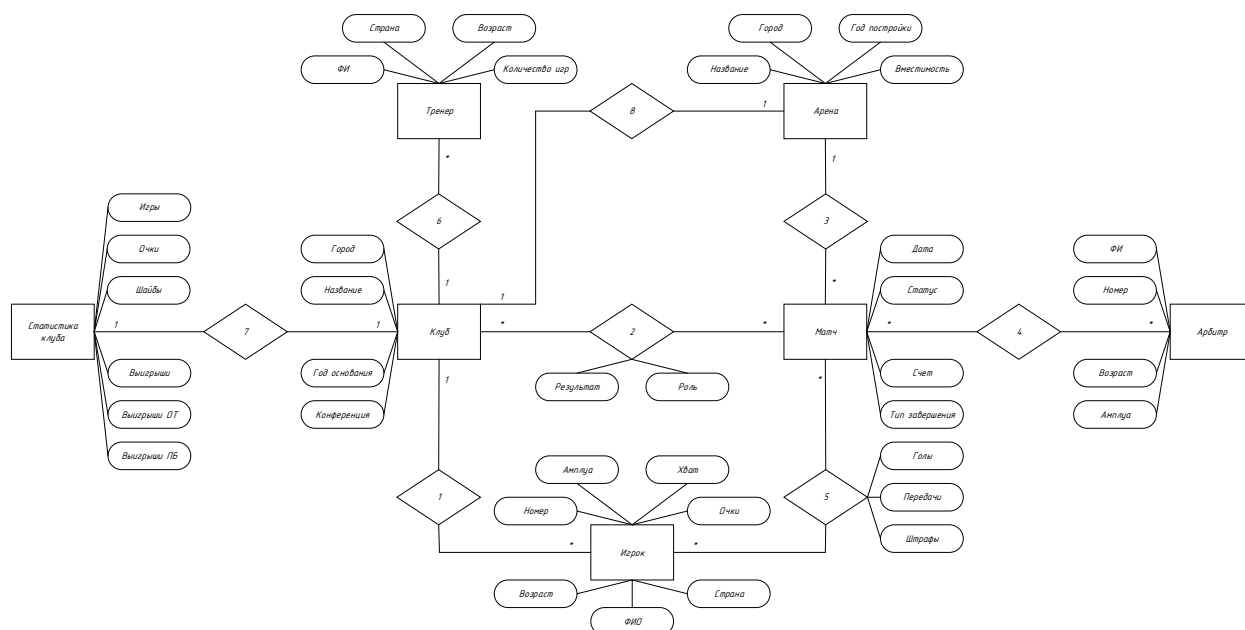


Рисунок 2.1 – ER-диаграмма

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: «Клуб», «Игрок», «Матч», «Арена», «Арбитр», «Тренер», «Статистика клуба».

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 клуба, id игрока, id матча, id арены, id арбитра, id тренера, id статистики клуба (рис. 2.2).

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. 2.3).

5. Связи «многие-ко-многим» реализуются через отдельную промежуточную таблицу. Например, была создана таблица «Матч-Игрок», в которой находятся два поля внешних ключей: «id игрока» и «id матча», а также три атрибута: «голы», «передачи» и «штрафы».

Первичный ключ в отношении «Матч-Игрок» построен как простой на основе суррогатного поля (PK) (рис. 2.4).

UML-диаграмма реляционной схемы данных «бумажного» преобразования представлена на рисунке 2.5.

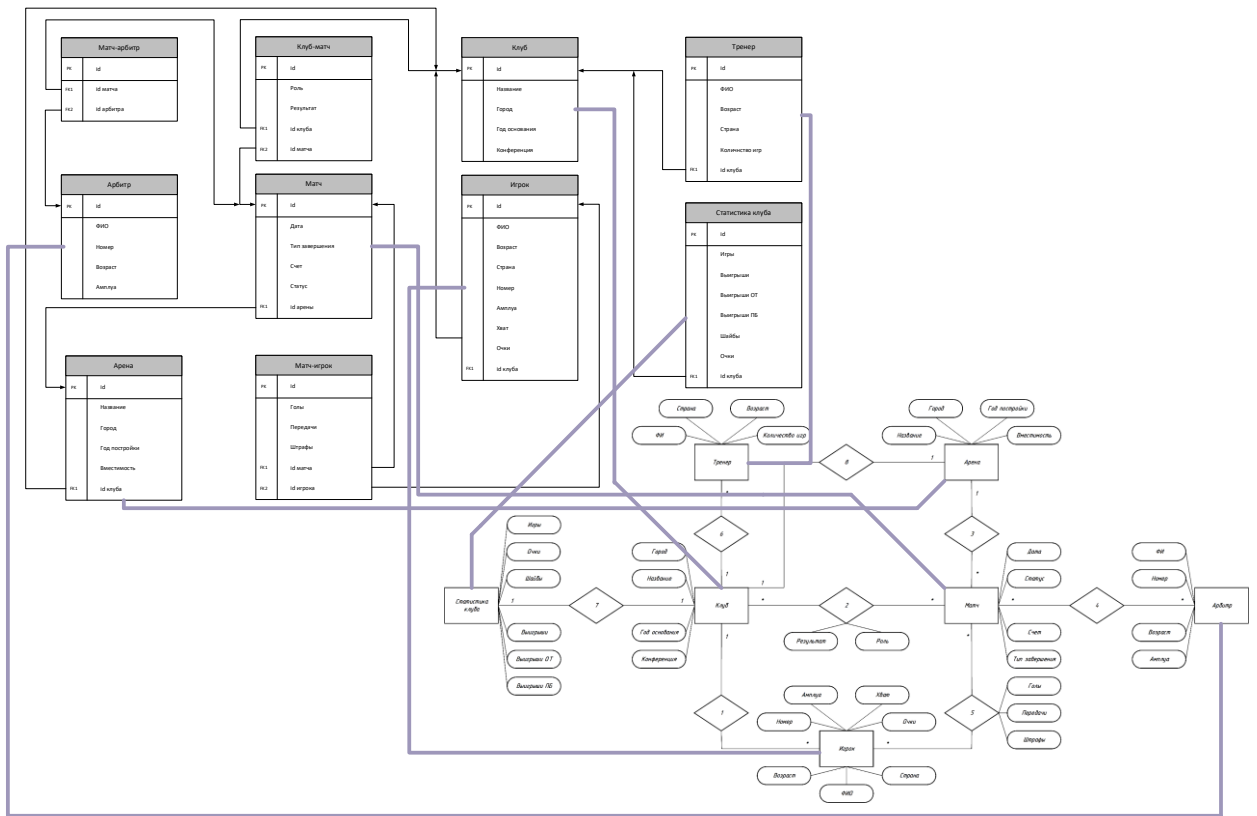


Рисунок 2.2 – сопоставление объектов URD и UML

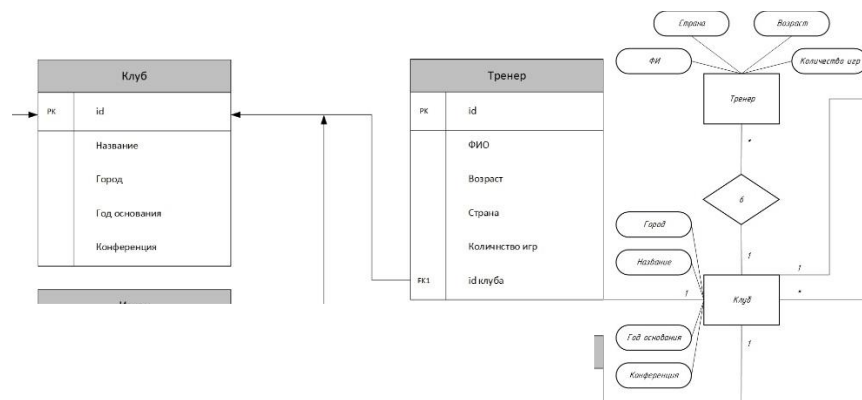


Рисунок 2.3 – Связь «один-ко-многим»

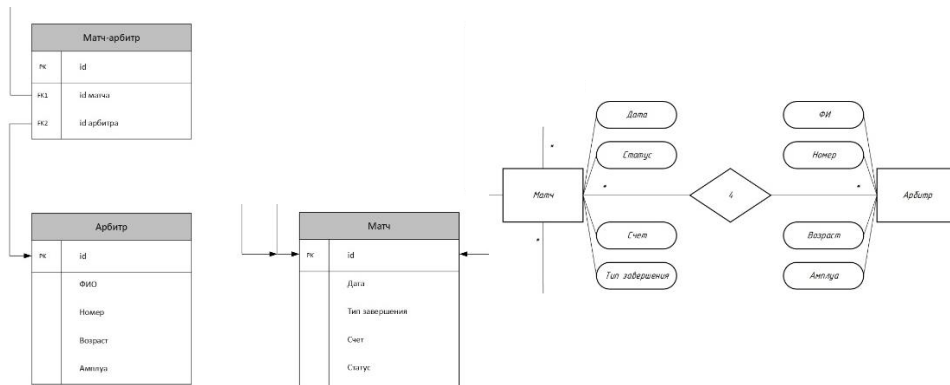


Рисунок 2.4 – Связь «многие-ко-многим»



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).

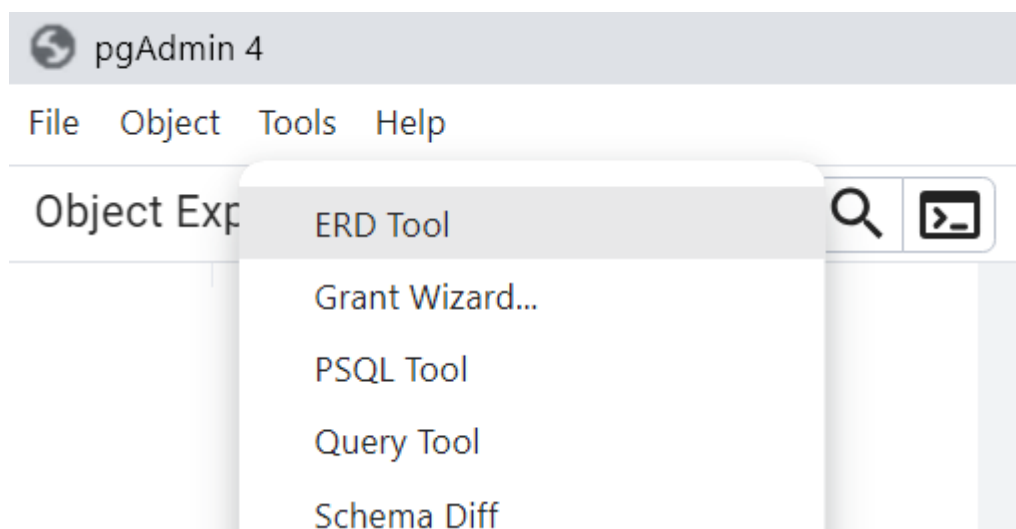


Рисунок 3.1 – Вкладка Tools

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

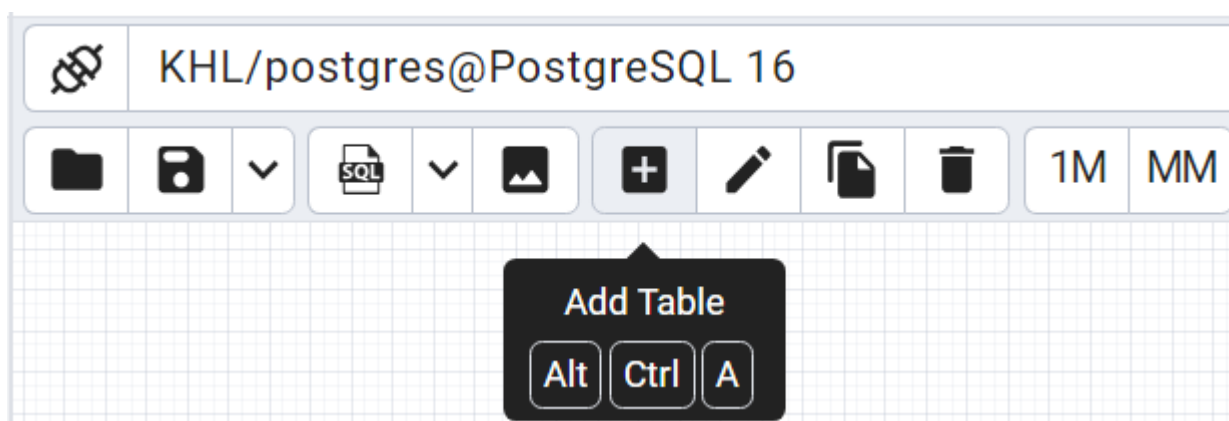


Рисунок 3.2 – Создание таблицы

4. Ввести имя таблицы и добавить необходимые колонки (рис. 3.3).
5. Создать остальные нужные таблицы, и добавить связи между ними (рис 3.4).
6. Нажать на кнопку Generate SQL и выполнить SQL-код (рис. 3.5). На рисунке 3.6 показан SQL код для создание реляционной модели.
7. Сохранить проект и ERD-диаграмму (рис. 3.7).

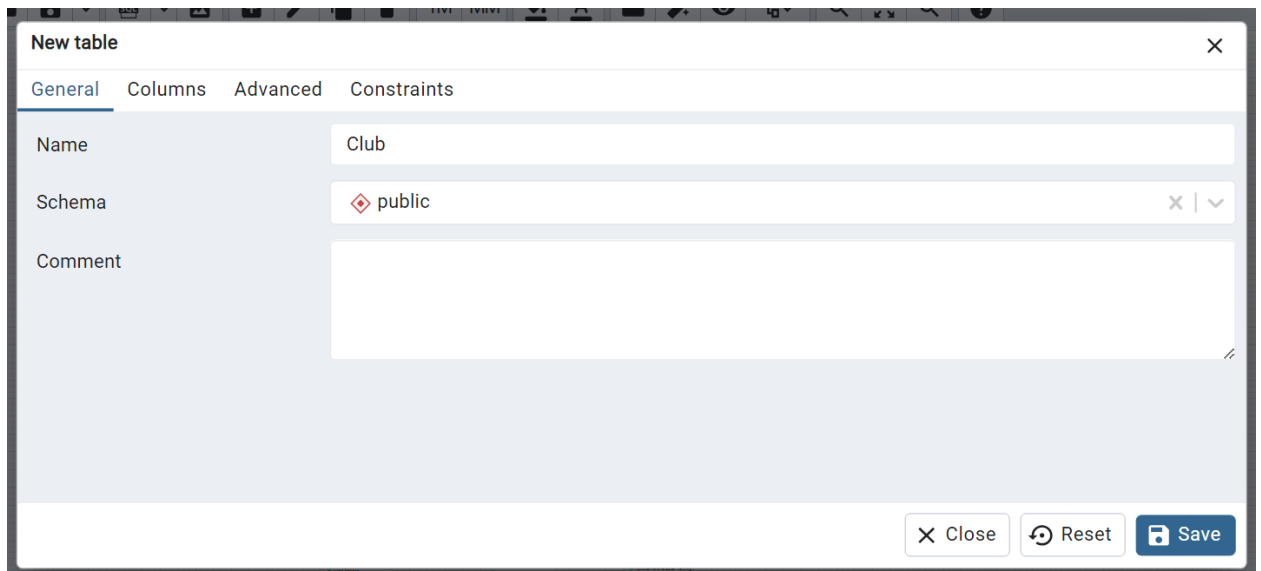


Рисунок 3.3 – Добавление колон и имени таблицы

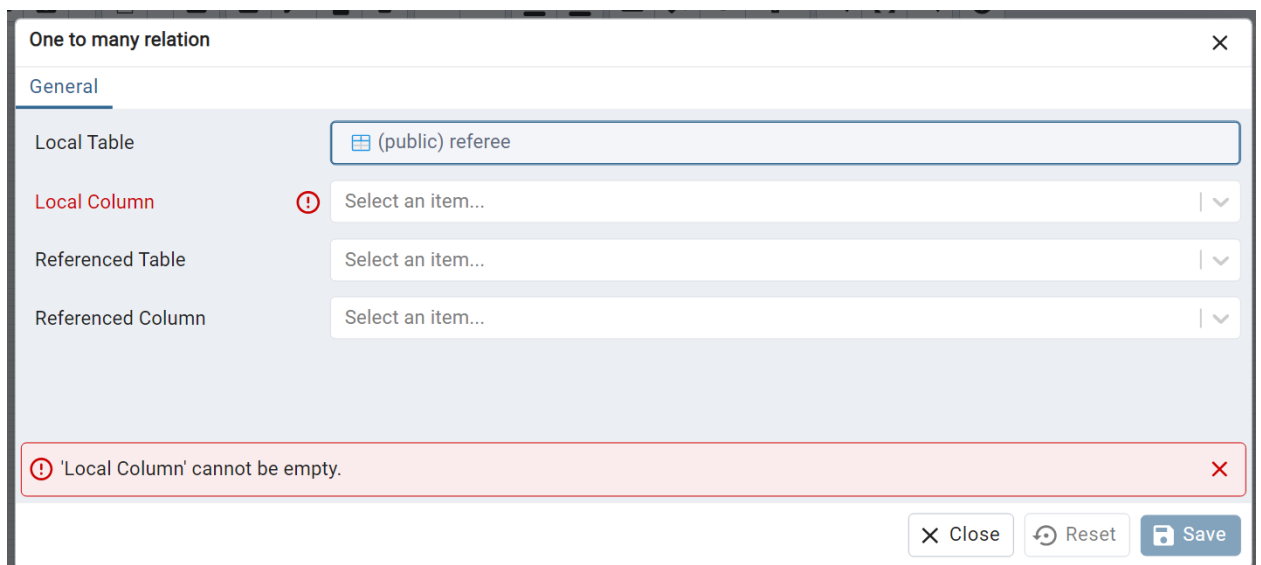


Рисунок 3.4 – Добавление связей

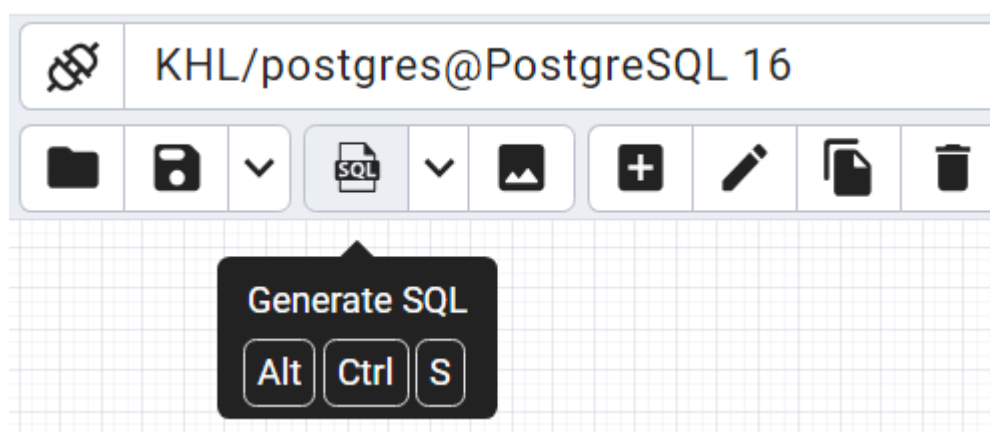


Рисунок 3.5 – Генерация SQL кода

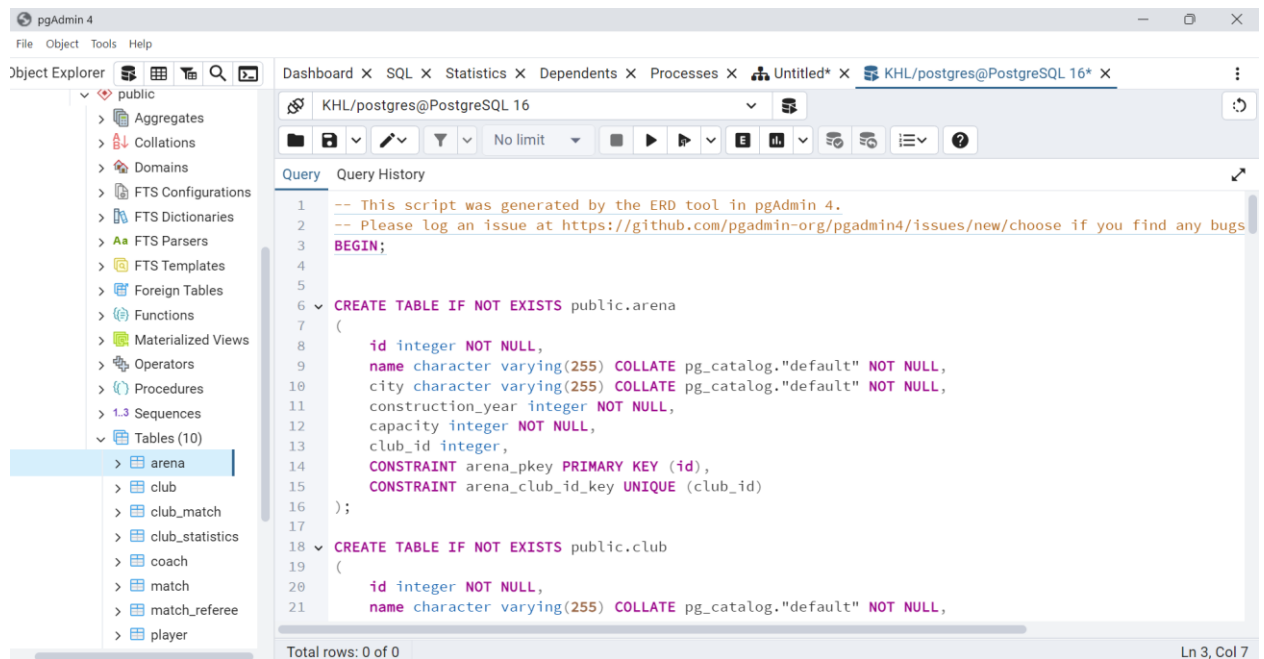


Рисунок 3.6 – SQL код

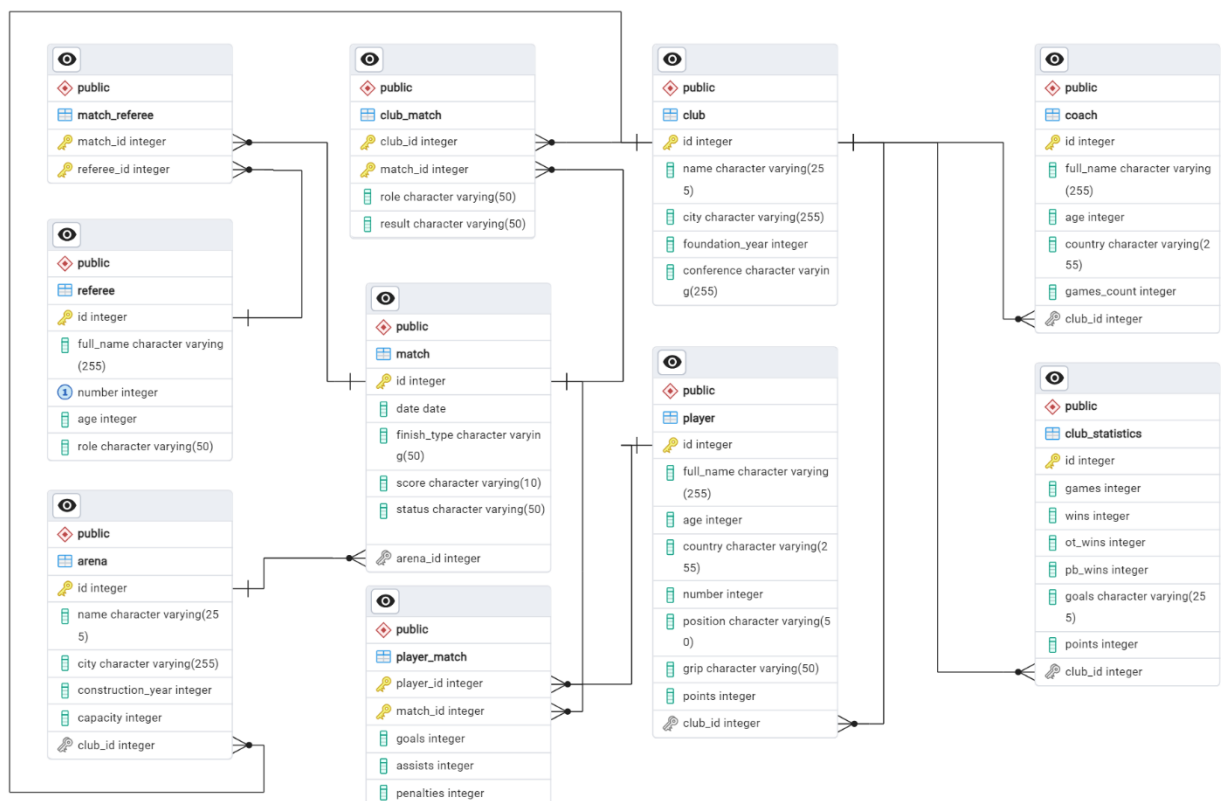


Рисунок 3.7 – ERD-диаграмма

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения лабораторной работы была успешно реализована логическая схема проектирования базы данных «Континентальная хоккейная лига». 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.