**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**КУРСОВОЙ ПРОЕКТ**

**по дисциплине**

**«Проектирование и администрирование баз данных»**

**Тема: «Проектирование базы данных научных конференций»**

**Выполнил: ст. гр. 221-352 Петроченко К.А.**

**Проверил: Тимакин О. А.**

**Москва – 2023**

**Содержание**

[Введение 3](#_Toc148377782)

[1. Исследование предметной области 6](#_Toc148377783)

[2. Инфологическое проектирование 9](#_Toc148377784)

[3. Логическое проектирование базы данных 12](#_Toc148377785)

[4. Физическое проектирование базы данных 18](#_Toc148377786)

[5. Руководство пользователя 19](#_Toc148377787)

[Заключение 24](#_Toc148377788)

[Список использованных источников 25](#_Toc148377789)

[Приложение 1 26](#_Toc148377790)

[Приложение 2 27](#_Toc148377791)

[Приложение 3 29](#_Toc148377792)

[Приложение 4 32](#_Toc148377793)

# Введение

Научные конференции играют важную роль в мире науки и исследований. Они представляют собой площадку для обмена уникальными научными результатами, проведения глубоких дискуссий и установления новых научных связей. Каждый год количество ученых и исследователей, принимающих участие в таких мероприятиях, неуклонно растет. С этим ростом возникает потребность в более эффективном способе управления информацией о конференциях и ее участниках.

Целью данной работы является разработка и создание базы данных для управления информацией о научных конференциях и ученых, принимающих в них участие. Мы будем описывать ключевые характеристики и атрибуты ученых, такие как имя, принадлежность к организациям, странам и научные степени. Кроме того, мы будем также записывать данные о самих конференциях: название, место проведения, дату, тип участия, тему доклада и возможность публикации.

С постоянным ростом объема данных и разнообразия конференций требуется система, которая обеспечит структурированное хранение информации, удобный доступ к данным и возможность анализа и составления отчетов. Эффективная база данных позволит автоматизировать процессы учета научных конференций, упростить поиск и анализ данных, что сделает научные исследования более организованными и результативными. В рамках данной курсовой работы мы планируем рассмотреть не только структуру и реализацию базы данных, но и провести анализ информации о конференциях и ученых, чтобы предоставить полезные сведения для исследователей, администраторов и всех тех, кто заинтересован в улучшении процесса научного обмена. Создание базы данных для управления данными о научных конференциях является важным шагом в развитии научного сообщества и его совершенствовании.

Таким образом, для цели данной курсовой работы предметом исследования выбрана организация и управление информацией о научных конференциях, в которых участвуют исследователи в области [вашей области научных интересов]. Объектом исследования является процесс создания базы данных, предназначенной для эффективного управления информацией о научных конференциях и ученых-участниках этих конференций.

Для развертывания базы данных PostgreSQL требуется специализированное оборудование и программное обеспечение (ПО), обеспечивающие надежное и стабильное функционирование базы данных и соответствующие требованиям по производительности. В следующем перечне представлены основные компоненты, необходимые для успешной реализации базы данных:

Оборудование:

1) Сервер: для обеспечения высокой производительности и надежности рекомендуется использовать мощное серверное оборудование с достаточным объемом оперативной памяти, процессором высокой производительности и хранилищем данных с поддержкой высокой скорости чтения и записи. Это обеспечит эффективное хранение и обработку данных о научных конференциях и ученых.

2) Хранилище данных: для надежного хранения базы данных PostgreSQL рекомендуется использовать накопители с высокой емкостью и быстродействием, такие как жесткие диски (HDD) или твердотельные накопители (SSD). Это обеспечит быстрый доступ к данным и минимизацию времени на обработку запросов.

Программное обеспечение:

1) Операционная система: PostgreSQL совместим с различными операционными системами, включая Linux, Windows и macOS.

2) PostgreSQL: необходимо провести установку и настройку базы данных PostgreSQL на выбранной операционной системе. Вы можете загрузить PostgreSQL с официального веб-сайта и следовать инструкциям по установке и настройке для выбранной операционной системы.

3) Управление базой данных: для более удобного администрирования и мониторинга базы данных PostgreSQL, вы можете использовать графические среды управления, такие как pgAdmin, DBeaver, phpPgAdmin и другие. Эти инструменты предоставляют интуитивно понятные графические интерфейсы для создания, управления и мониторинга базы данных.

Важно отметить, что требования к оборудованию и программному обеспечению могут варьироваться в зависимости от объема информации о научных конференциях и характера вашего исследования.

# 1. Исследование предметной области

Предметной областью данной исследовательской работы являются научные конференции и исследователи, участвующие в этих мероприятиях. Научные конференции представляют собой события, на которых ученые и исследователи со всего мира собираются для обмена научными знаниями, обсуждения своих исследовательских результатов и установления новых научных связей. Конференции могут охватывать широкий спектр научных областей, начиная от естественных и точных наук и заканчивая областями гуманитарных и социальных исследований.

Создаваемая база данных будет полезна следующим пользователям:

1. Исследователи и ученые:

Ученые смогут использовать базу данных для поиска научных конференций, связанных с их областью исследований, и для получения информации о других ученых, участвующих в этих конференциях.

1. Организаторы конференций:

Организаторы конференций смогут использовать базу данных для планирования и управления мероприятиями, включая выбор места проведения, управление участниками и оповещение о важных событиях.

1. Академические учебные заведения:

Университеты и научные институты смогут использовать базу данных для анализа активности своих исследователей на научных конференциях и оценки их вклада в научное сообщество.

1. Библиотеки и научные издательства:

Организации, занимающиеся научной информацией, могут использовать базу данных для классификации и архивации научных публикаций, связанных с конференциями.

С использованием созданной базы данных предполагается решать следующие задачи и выполнять запросы:

1. Поиск конференций:

Пользователи смогут искать научные конференции по различным параметрам, таким как название, дата, место проведения, область исследований и тип участия.

1. Поиск ученых:

Поиск ученых, участвующих в конференциях, с учетом их имени, организации, страны и ученой степени.

1. Анализ активности ученых:

Определение наиболее активных ученых, участвующих в конференциях, и анализ их участия по времени, тематике и публикациям.

1. Управление информацией о конференциях:

Добавление, редактирование и удаление данных о научных конференциях, обновление информации о датах и местах проведения.

1. Управление данными о ученых:

Добавление новых ученых, обновление их профилей, а также управление информацией о публикациях и ученых степенях.

Частота решения задач и используемые бизнес-правила будут зависеть от конкретных потребностей пользователей и актуальности данных. Например:

1. Поиск конференций может выполняться ежедневно, так как информация

о предстоящих конференциях может изменяться быстро.

1. Поиск ученых может осуществляться по мере необходимости, когда

пользователь ищет информацию о конкретном ученом.

1. Анализ активности ученых может проводиться периодически,

например, ежеквартально, чтобы выявить изменения в активности

исследователей.

1. Управление информацией о конференциях и ученых может

осуществляться по мере поступления новых данных или обновления

существующей информации.

Бизнес-правила будут включать в себя правила валидации данных (например, проверка на уникальность названий конференций) и процедуры обновления данных (например, автоматическое обновление информации о конференциях по данным из официальных источников).

# 2. Инфологическое проектирование

В данном разделе представлено инфологическое проектирование базы данных для системы учета научных конференций. В процессе инфологического проектирования были определены сущности, атрибуты сущностей и их связи. Инфологическая модель базы данных является высокоуровневой и не зависит от конкретной реализации СУБД.

Таблица 2.1 – Формирование сущностей

|  |  |  |
| --- | --- | --- |
| Название сущности | Описание сущности | Количество экземпляров |
| Ученый | Информация об ученых и их ученых степенях. | Много |
| Организация | Информация об организациях, к которым принадлежат ученые. | Много |
| Страна | Информация о странах, в которых проживают ученые и расположены организации. | Много |
| Ученая степень | Информация об ученых степенях, которые могут иметь ученые. | Много |
| Конференция | Информация о научных конференциях, включая название, место проведения, дату, тип участия, тему доклада и публикацию (да/нет). | Много |

В таблице 1.1 "Формирование сущностей" представлены сущности, которые будут использоваться в базе данных системы учета научных конференций. В данной таблице также указано количество экземпляров каждой сущности.

Далее в таблице 1.2 будут представлены названия сущностей, названия атрибутов, описание атрибутов, диапазон значений, единицы измерения и пример для атрибутов сущностей, представленных в таблице 1.1.

Таблица 2.2 – Формирование сущностей

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название атрибута | Описание атрибута | Диапазон значений | Единица измерения | Пример |
| Имя ученого | Имя ученого | Текст | Н/Д | Иван |
| Организация | Название организации, к которой принадлежит ученый. | Текст | Н/Д | Институт физики |
| Страна | Название страны, в которой проживает ученый. | Текст | Н/Д | Россия |
| Ученая степень | Уровень ученой степени у ученого. | Текст | Н/Д | Доктор |
| Название конференции | Название научной конференции. | Текст | Н/Д | Конференция А |
| Место проведения | Место проведения научной конференции. | Текст | Н/Д | Город Москва |
| Дата | Дата начала научной конференции. | Дата в формате ДД-ММ-ГГГГ | Дни | 01-01-2023 |
| Тип участия | Тип участия ученого в конференции, например, доклад, сообщение, стендовый доклад и другие. | Текст | Н/Д | Доклад |
| Тема доклада | Тема доклада ученого на конференции. | Текст | Н/Д | Правовое регулирование труда моряков |
| Публикация | Публикация доклада ученого на конференции (да/нет). | Да/Нет | Н/Д | Да |

Таблицf 2.2 описывает атрибуты различных сущностей, связанных с организацией и управлением учебным процессом. Далее используя представленные сущности, будет проведена работа по выявлению, установлению и описанию связей между ними.

Таблица 2.3 – Установление связей между сущностями

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название первой сущности | Название второй сущности | Название связи | Тип связи | Обоснование выбора типа связи |
| Ученый | Организация | Принадлежит | Многие к одному | Один ученый может принадлежать только одной организации, но одна организация может иметь множество ученых. |
| Ученый | Страна | Проживает в | Многие к одному | Один ученый может проживать только в одной стране, но одна страна может иметь множество ученых. |
| Ученый | Ученая степень | Имеет степень | Многие к одному | Один ученый может иметь одну или более ученых степеней. |
| Ученый | Конференция | Участвовал в | Многие ко многим | Один ученый может участвовать в нескольких конференциях, и на одной конференции может быть множество ученых. |

Таблица 2.10 представляет связи между различными сущностями в системе. Она описывает тип связи и обоснование выбора этого типа для каждой пары сущностей. Связи в таблице указывают на то, какие сущности связаны между собой и как эти связи устанавливаются.

# 3. Логическое проектирование базы данных

Целью данного этапа является построение реляционной логической модели. Реляционная логическая модель представляет собой совокупность нормализованных отношений, в которых реализованы связи между объектами предметной области и выполнены все преобразования, необходимые для ее эффективной реализации в среде конкретной СУБД. Существует множество СУБД для реляционных баз данные таких как:

1. MySQL:

* Легковесная и простая в использовании СУБД с открытым исходным кодом;
* Хорошо подходит для небольших и средних проектов, особенно веб-приложений;
* Обладает высокой производительностью при обработке простых запросов;
* Имеет некоторые ограничения в функциональности и возможностях сравнительно с другими СУБД.

1. SQLite:

* Компактная и встраиваемая СУБД, которая не требует отдельного сервера;
* Подходит для небольших проектов или в случаях, когда требуется локальное хранение данных;
* Обладает простым синтаксисом запросов и низкими требованиями к системным ресурсам;
* Не поддерживает масштабирование для больших проектов и не обеспечивает возможности сетевого доступа.

1. Microsoft Access:

* База данных, разработанная для использования на платформе Microsoft Windows;
* Предназначена для небольших и средних проектов, особенно для индивидуальных пользователей или небольших рабочих групп;
* Обладает интуитивным пользовательским интерфейсом и интеграцией с другими продуктами Microsoft;
* Имеет ограниченные возможности в масштабировании, производительности и многопользовательской работе.

1. PostgreSQL:

* Мощная и полнофункциональная СУБД с открытым исходным кодом;
* Подходит для проектов любого размера и сложности, включая крупные предприятия;
* Обладает расширенными возможностями, такими как поддержка сложных запросов, транзакций, уровня изоляции и масштабируемости;
* Поддерживает различные типы данных, географические объекты, хранимые процедуры и многое другое;
* PostgreSQL имеет активное сообщество разработчиков и обновляется регулярно для улучшения производительности и безопасности.

Мною была выбрана такая СУБД как PostgreSQL, так как в целом, она является превосходным выбором для проектов любого масштаба, особенно когда требуется мощная функциональность, надежность и масштабируемость базы данных.

Для управления базой данных мною была выбрана такая программа как HeidiSQL, которая предоставляет графический интерфейс для работы с различными системами управления базами данных (СУБД), включая MySQL, MariaDB, PostgreSQL и Microsoft SQL Server. С помощью HeidiSQL можно создавать, изменять и удалять таблицы, выполнять запросы, управлять пользователями и привилегиями, просматривать и редактировать данные в таблицах, а также импортировать и экспортировать данные. Она предоставляет удобный и интуитивно понятный интерфейс, что делает работу с базами данных более эффективной и удобной.

Чтобы полностью понять структуру базы данных и взаимосвязи между таблицами нужно спроектировать ER-диаграмму.

ER-диаграмма (Entity-Relationship diagram) — это графическое представление сущностей (entities), связей (relationships) между ними и атрибутов (attributes), которые описывают эти сущности и связи в базе данных. ER-диаграммы используются для моделирования и проектирования баз данных, а также для описания и анализа сложных систем и процессов.

На ER-диаграмме сущности представляются в виде прямоугольников, а атрибуты - в виде овалов, связи между сущностями - в виде линий, на которых могут быть указаны ограничения, например, кратность отношения. ER-диаграмма позволяет визуализировать структуру данных, а также определять связи между сущностями и атрибутами, что упрощает понимание и проектирование баз данных.

В таблицах 3.1 – 3.11 представлены структуры таблиц базы данных

Сокращения, используемые в таблицах:

1. PK – Primary Key (Первичный ключ)
2. FK – Foreign Key (Внешний ключ)
3. NN – Not Null (Не нулевой)

Типы, используемые в таблицах:

1. Serial: автоматически увеличивающееся целое число, используется для уникальных идентификаторов.
2. Varchar: строка с переменной длиной, используется для хранения текстовых данных переменной длины.
3. Bigint: Целое число большого диапазона (от -9,2…18 до +9,2…18).
4. Time: хранит только время (часы, минуты, секунды).
5. Date: хранит только дату.
6. Bool: логический тип данных, принимает значения true или false.

Таблица 3.1 – Таблица scientists

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип | Примечания |
| Ученый ID | researcher\_id | INT | PK, NN |
| Имя | name | VARCHAR(255) | NN |
| Организация ID | org\_id | INT | FK, NN |
| Страна ID | country\_id | INT | FK, NN |
| Ученая степень ID | degree\_id | INT | FK, NN |

Таблица 3.1 представляет структуру таблицы «scientists» базы данных.

Таблица 3.2 – Таблица organizations

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип | Примечания |
| Организация ID | org\_id | INT | PK, NN |
| Название | name | VARCHAR(255) | NN |
| Страна ID | country\_id | INT | FK, NN |

Таблица 3.2 представляет структуру таблицы «organizations» базы данных.

Таблица 3.3 – Таблица countries

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип | Примечания |
| Страна ID | country\_id | INT | PK, NN |
| Название | name | VARCHAR(255) | NN |

Таблица 3.3 представляет структуру таблицы «groups» базы данных.

Таблица 3.4 – Таблица degrees

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип | Примечания |
| Ученая степень ID | degree\_id | INT | PK, NN |
| Степень | degree | VARCHAR(255) | NN |

Таблица 3.4 представляет структуру таблицы «degrees» базы данных.

Таблица 3.5 – Таблица conferences

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип | Примечания |
| Конференция ID | conference\_id | INT | PK, NN |
| Название | name | VARCHAR(255) | NN |
| Место проведения | location | VARCHAR(255) | NN |
| Дата | date | DATE | NN |
| Тип участия ID | type\_id | INT | FK, NN |
| Тема доклада | topic | VARCHAR(255) | NN |
| Публикация | publication | BOOLEAN | NN |

Таблица 3.5 представляет структуру таблицы «conferences» базы данных.

Таблица 3.6 – Таблица scientist\_conference

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип | Примечания |
| Конференция ID | conference\_id | INT | PK, NN |
| Ученый ID | researcher\_id | INT | PK, NN |

Таблица 3.6 представляет структуру таблицы «scientist\_conference» базы данных.

Таблица 3.6 – Таблица types

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип | Примечания |
| Тип участия ID | type\_id | INT | PK, NN |
| Тип | type | VARCHAR(255) | NN |

Таблица 3.6 представляет структуру таблицы «types» базы данных.

Таблицы 3.1 – 3.6 представляют структуру базы данных для управления информацией о научных конференциях, включая данные об ученых, организациях, странах, ученых степенях, конференциях, участиях, темах докладов и публикациях. Эти таблицы представляют основные сущности и атрибуты, необходимые для хранения соответствующей информации о научных конференциях. Для более полного понимания структуры базы данных и взаимосвязей между таблицами, рекомендуется обратиться к приложению 1, где представлена ER-диаграмма. Эта диаграмма визуализирует связи между таблицами и помогает лучше представить структуру и организацию данных в базе данных для управления информацией о научных конференциях.

# 4. Физическое проектирование базы данных

Поскольку была выбрана СУБД с открытым исходным кодом PostgreSQL и программа для управления базой данных HeidiSQL, то таблицы спроектированной базы данных будут иметь вид, представленный в приложении 2

В приложении 3 представлен код на языке структурированных запросов (SQL) для создания базы данных, описанных таблиц и заполнения их тестовыми данными. После выполнения представленного кода мы получаем готовую базу данных с заполненными таблицами.

# 5. Руководство пользователя

Для работы с базой данных необходимы следующие программные средства:

* PostgreSQL (В качестве базы данных);
* pgAdmin, HeidiSQ, … (В качестве СУБД);
* Дамп базы данных или сервер с готовой базой данных.

Для проектируемой базы данных будет достаточно следующего оборудования:

1. Операционная система: Ubuntu;
2. Процессор: с частотой не менее 1 ГГц;
3. ОЗУ: не менее 1 ГБ;
4. Дисковое пространство: не менее 1 ГБ. В зависимости от наполнения базы данных может потребоваться большее пространство.

Для начала работы нужно установить всё необходимое и запустить СУБД, в примере будет фигурировать HeidiSQL.

Для подключения к базе данных нужно:

1. Создать подключение;
2. Задать имя подключению;
3. Выбрать PostgreSQL (TCP/IP);
4. Выбрать библиотеку libpq-15.dll или новее;
5. Задать ip адрес базы данных (127.0.01 – локальная, 10.175.15.46 (например) – удалённая);
6. Ввести имя пользователя пользователя (По умолчанию postgres);
7. Ввести пароль пользователя;
8. Ввести порт на котором работает база данных (Для PostgreSQL по умолчанию 5432);
9. Ввести название базы данных.

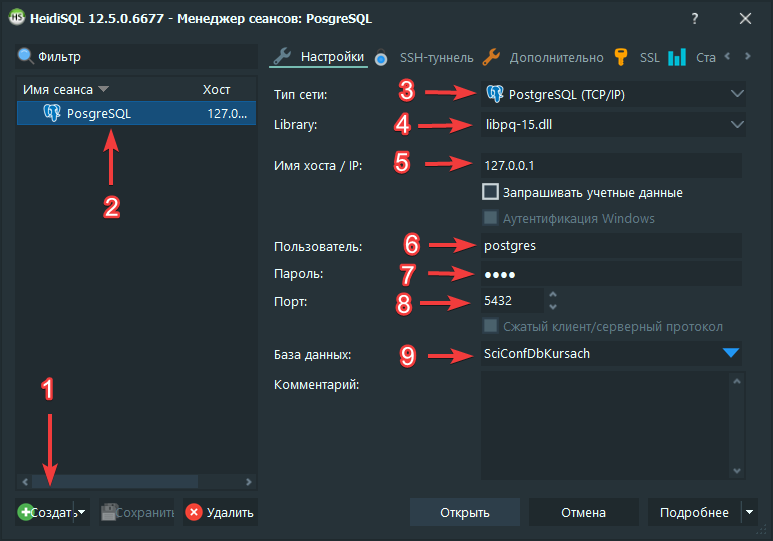


Рисунок 5.1 – Подключение

Далее перед нами предстаёт интерфейс базы данных. Чтобы выполнять запросы, посредством которых мы сможем взаимодействовать с базой данных, нужно перейти во вкладку «Запрос».

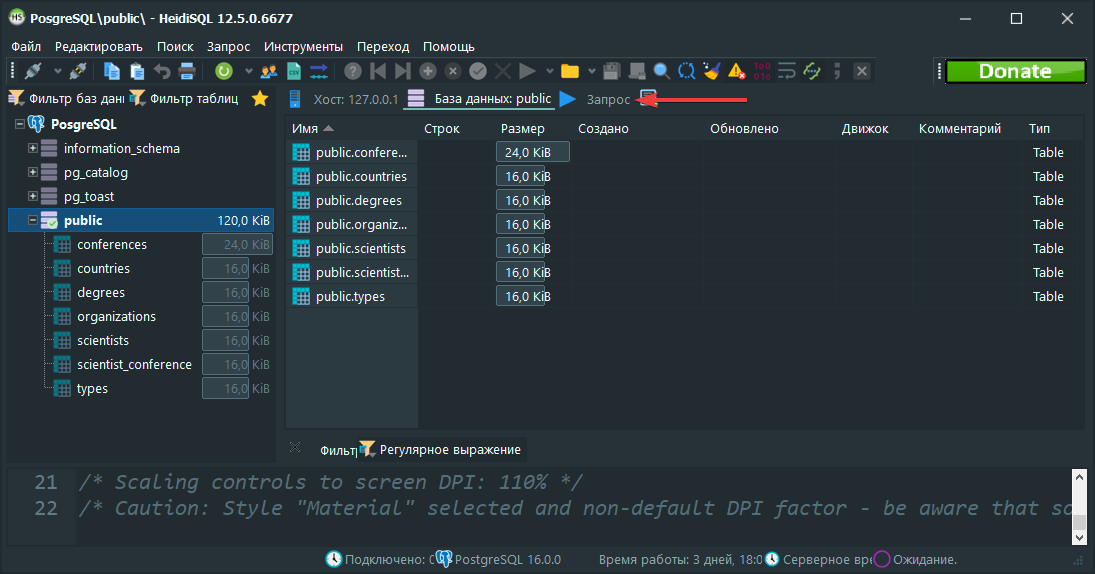


Рисунок 5.2 – Интерфейс

Чтобы получить данные из базы данных, нужно во вкладке запрос написать соответствующий SQL запрос, например для получения адресов зданий необходимо написать и выполнить следующий SQL запрос: SELECT type AS "Тип" FROM types

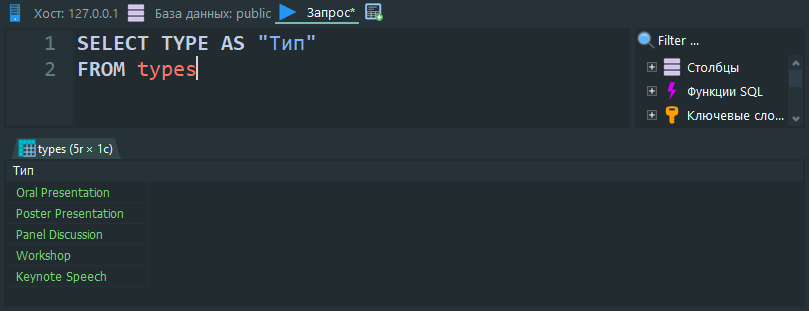


Рисунок 5.3 – Пример получения данных

Для добавление нового адреса здания необходимо написать и выполнить следующий SQL запрос: INSERT INTO types (type) VALUES ('Тип участия');

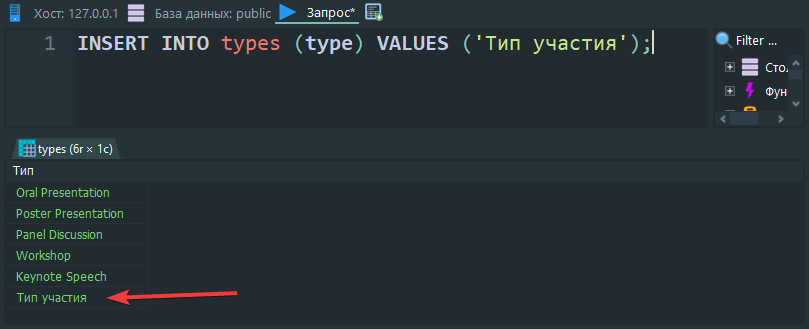


Рисунок 5.4 – Пример добавления данных

Для обновления записи адреса здания необходимо написать и выполнить следующий SQL запрос: UPDATE types SET type = 'Новый тип участия' WHERE type = 'Тип участия'

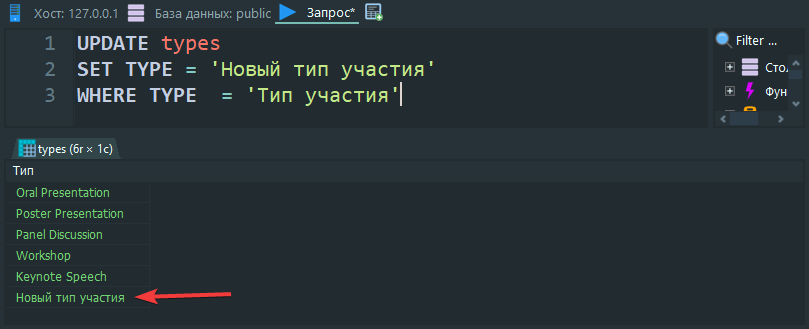


Рисунок 5.5 – Пример обновления данных

Для удаления записи адреса здания необходимо написать и выполнить следующий SQL запрос: DELETE FROM types WHERE type= 'Новый тип участия'

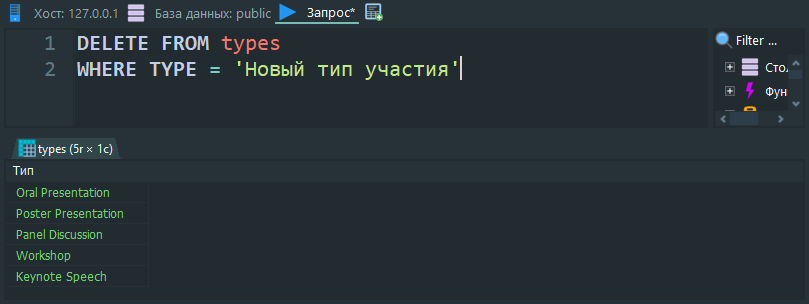


Рисунок 5.6 – Пример удаления данных

Таким образом выполняя подобные SQL запросы можно получать, добавлять, обновлять и удалять данные из таблиц базы данных. Выше были представлены основные запросы, которые будут использоваться чаще всего. Но на этом администрирование базы данных не ограничивается, существует ещё огромное множество запросов, с помощь которых можно управлять базой данных.

# Заключение

В данной курсовой работе была произведена разработка базы данных для научных конференций, предназначенной для эффективного управления информацией о ученых, организациях, конференциях и их участниках. Была представлена диаграмма сущность-связь, которая позволила определить основные сущности системы, их свойства и отношения между ними.

В работе были описаны свойства каждой сущности, а также их отношения между собой. Для реализации связей между сущностями использовались внешние ключи. Кроме того, были представлены SQL-запросы для создания таблиц и связей между ними.

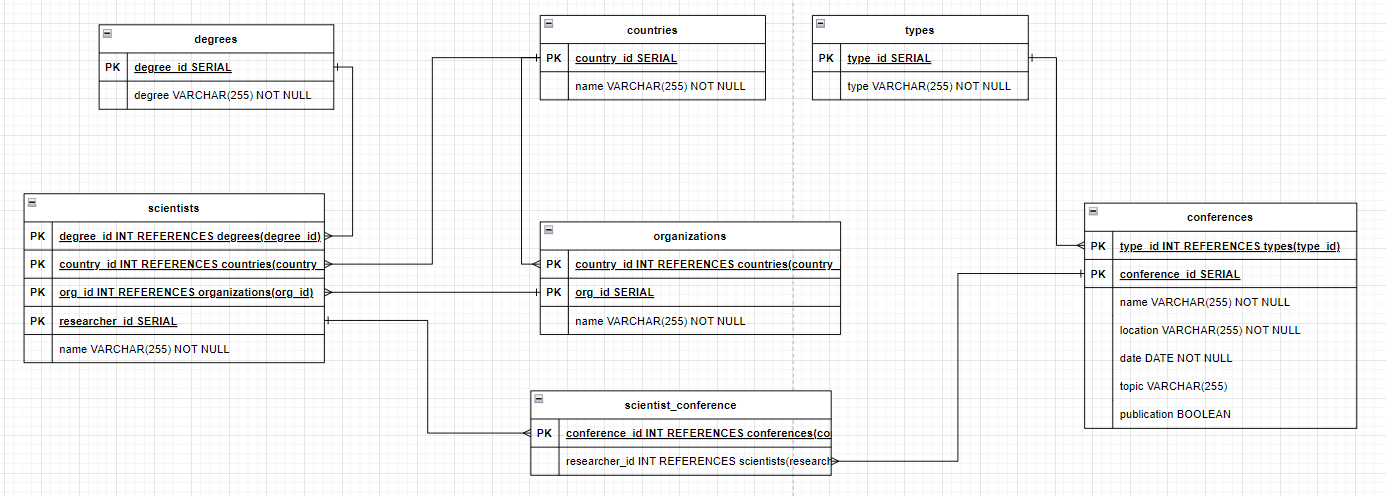
В работе использовался PostgreSQL — это реляционная база данных с открытым кодом, которая поддерживается в течение 30 лет разработки и является одной из наиболее известных среди всех существующих реляционных баз данных.

В результате выполнения курсовой работы была разработана база данных, которая может использоваться для для эффективного управления информацией о научных конференциях. Данная база данных может быть расширена и доработана в соответствии с требованиями конкретных научных мероприятий или приложений.

# Список использованных источников

1. Информация о ER – диаграммах: <https://clck.ru/34Mhwu>. (Дата обращения 03.06.2023).
2. Дейт К. Дж. Введение в системы баз данных. / К. Дж. Дейт. - М.: Вильямс, 2014. - 976 с.
3. Сильверман Р. SQL All-in-One For Dummies. / Р. Сильверман, А. Брассинский, Дж. Хаслинг. - Wiley, 2019. - 816 с.
4. О'Коннор Дж. PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database / Дж. О'Коннор. - O'Reilly Media, 2017. – 338с.
5. Бекман Д. PostgreSQL: программирование и производительность. / Д. Бекман. - М.: ДМК Пресс, 2016. - 416 с.
6. Гарсия-Молина Х. Хранилища данных: полное руководство. / Х. Гарсия-Молина, Д. Ульман, Д. Уидом. - М.: Вильямс, 2014. - 1120 с.
7. Рамакришнан Р. Системы управления базами данных. / Р. Рамакришнан, Й. Герхке. - М.: Вильямс, 2015. - 1456 с.
8. Аллен М. Oracle PL/SQL Programming. / М. Аллен, Л. Хантер. - O'Reilly Media, 2014. - 1392 с.
9. Клейн П. Разработка баз данных в среде Delphi. / П. Клейн. - М.: ДМК Пресс, 2012. - 384 с.
10. Бернстайн П.А. Мультимедийные базы данных: проектирование, разработка и поддержка. / П.А. Бернстайн, Н. Ньюкомб. - М.: Издательский дом "Вильямс", 2013. - 592 с.
11. Стоянов И. Распределенные базы данных. / И. Стоянов. - М.: Лаборатория Базовых Знаний, 2011. - 320 с.

# Приложение 1



# Приложение 2

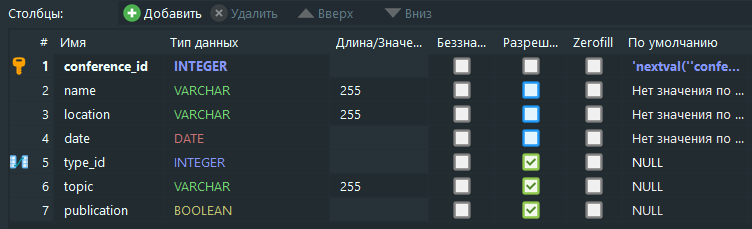


Рисунок приложения 2.1 – Таблица «conferences»

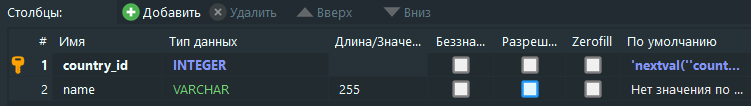


Рисунок приложения 2.2 – Таблица «countries»

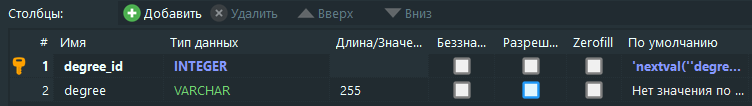


Рисунок приложения 2.3 – Таблица «degrees»

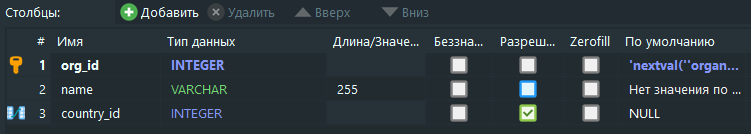


Рисунок приложения 2.4 – Таблица «organizations»

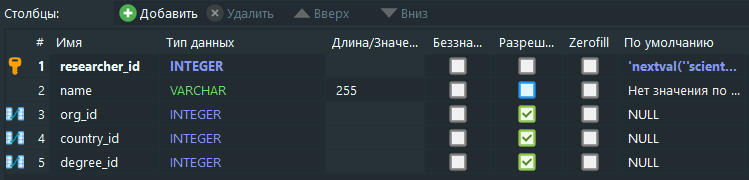


Рисунок приложения 2.5 – Таблица «scientists»

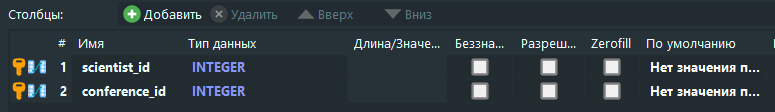


Рисунок приложения 2.6 – Таблица «scientist\_conference»

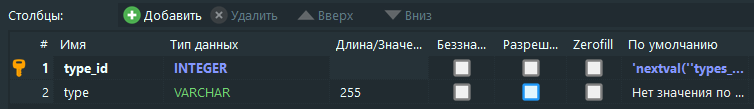


Рисунок приложения 2.7 – Таблица «types»

# Приложение 3

Код для создания и заполнения данными базы данных с таблицами conferences, countries, degrees, organizations, scientists, scientist\_conference, types:

-- Создание таблицы countries (Страны)

CREATE TABLE countries (

country\_id SERIAL PRIMARY KEY,

name VARCHAR(255) NOT NULL

);

-- Создание таблицы degrees (Ученые степени)

CREATE TABLE degrees (

degree\_id SERIAL PRIMARY KEY,

degree VARCHAR(255) NOT NULL

);

-- Создание таблицы types (Типы участия на конференциях)

CREATE TABLE types (

type\_id SERIAL PRIMARY KEY,

type VARCHAR(255) NOT NULL

);

-- Создание таблицы organizations (Организации)

CREATE TABLE organizations (

org\_id SERIAL PRIMARY KEY,

name VARCHAR(255) NOT NULL,

country\_id INT REFERENCES countries(country\_id)

);

-- Создание таблицы scientists (Ученые)

CREATE TABLE scientists (

researcher\_id SERIAL PRIMARY KEY,

name VARCHAR(255) NOT NULL,

org\_id INT REFERENCES organizations(org\_id),

country\_id INT REFERENCES countries(country\_id),

degree\_id INT REFERENCES degrees(degree\_id)

);

-- Создание таблицы conferences (Научные конференции)

CREATE TABLE conferences (

conference\_id SERIAL PRIMARY KEY,

name VARCHAR(255) NOT NULL,

location VARCHAR(255) NOT NULL,

date DATE NOT NULL,

type\_id INT REFERENCES types(type\_id),

topic VARCHAR(255),

publication BOOLEAN

);

-- Создание таблицы scientist\_conference (Связь между учеными и конференциями)

CREATE TABLE scientist\_conference (

scientist\_id INT REFERENCES scientists(researcher\_id),

conference\_id INT REFERENCES conferences(conference\_id),

PRIMARY KEY (scientist\_id, conference\_id)

);

-- Вставка данных в таблицы

-- Вставка данных для таблицы countries

INSERT INTO countries (name) VALUES ('France');

INSERT INTO countries (name) VALUES ('Canada');

INSERT INTO countries (name) VALUES ('Australia');

-- Вставка данных для таблицы degrees

INSERT INTO degrees (degree) VALUES ('Ph.D.');

INSERT INTO degrees (degree) VALUES ('M.Sc.');

INSERT INTO degrees (degree) VALUES ('B.Sc.');

INSERT INTO degrees (degree) VALUES ('D.Sc.');

INSERT INTO degrees (degree) VALUES ('M.A.');

-- Вставка данных для таблицы types

INSERT INTO types (type) VALUES ('Oral Presentation');

INSERT INTO types (type) VALUES ('Poster Presentation');

INSERT INTO types (type) VALUES ('Panel Discussion');

INSERT INTO types (type) VALUES ('Workshop');

INSERT INTO types (type) VALUES ('Keynote Speech');

-- Вставка данных для таблицы organizations

INSERT INTO organizations (name, country\_id) VALUES ('Sorbonne University', 1);

INSERT INTO organizations (name, country\_id) VALUES ('University of Toronto', 2);

INSERT INTO organizations (name, country\_id) VALUES ('University of Sydney', 3);

-- Вставка данных для таблицы scientists

INSERT INTO scientists (name, org\_id, country\_id, degree\_id) VALUES ('Marie Dupont', 1, 1, 1);

INSERT INTO scientists (name, org\_id, country\_id, degree\_id) VALUES ('David Li', 2, 2, 2);

-- Вставка данных для таблицы conferences

INSERT INTO conferences (name, location, date, type\_id, topic, publication) VALUES ('International Conference 2023', 'Paris', '2023-06-25', 1, 'Science', TRUE);

INSERT INTO conferences (name, location, date, type\_id, topic, publication) VALUES ('Global Summit 2023', 'Toronto', '2023-09-05', 2, 'Technology', TRUE);

-- Вставка данных для таблицы scientist\_conference

INSERT INTO scientist\_conference (scientist\_id, conference\_id) VALUES (1, 1);

INSERT INTO scientist\_conference (scientist\_id, conference\_id) VALUES (2, 2);

# Приложение 4

Антиплагиат:

