Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Российский университет транспорта (МИИТ)» (РУТ (МИИТ)

Институт транспортной техники и систем управления  
  
Кафедра «Управление и защита информации»

КУРСОВОЙ ПРОЕКТ

Дисциплина:

«Информационное обеспечение систем управления»

Тема:

«Проектирование реляционной базы данных»

Выполнил: студ. гр. ВУИ-451

Лихачев С. С.

Проверил: доц., к.т.н.

Васильева М. А.

Москва 2024

Оглавление

[1. Инфологическое проектирование 2](#_Toc165927672)

[1.1 Анализ предметной области 2](#_Toc165927673)

[1.2 Выделим основные сущности базы данных: 3](#_Toc165927674)

[1.3 Дополнительные сущности базы данных 3](#_Toc165927675)

[1.4 Анализ информационных задач и круга пользователей системы 4](#_Toc165927676)

[2. Логическое проектирование реляционной БД 5](#_Toc165927677)

[3. Физическое проектирование БД 11](#_Toc165927678)

[3.1 Установка PostGIS 11](#_Toc165927679)

[3.2 Создание БД 13](#_Toc165927680)

[3.3 Создание таблиц 13](#_Toc165927681)

[4. Добавление данных в БД 21](#_Toc165927682)

[4.1 Создание последовательности 21](#_Toc165927683)

[4.2 Написание скриптов на добавление данных. 21](#_Toc165927684)

[5. Написание готовых запросов 28](#_Toc165927685)

### Инфологическое проектирование

### 1.1 Анализ предметной области

База данных создается для приложения для туристов, которые занимаются туризмом (трекинг, хайкинг, альпинизм).

Приложение должно осуществлять следующие функции:

• Пользователи могут просматривать маршруты, читать описание, смотреть фото и отзывы других путешественников;

• Пользователи могут создавать собственные маршруты и делиться, чтобы другие туристы могли ходить по ним;

• Некоторые пользователи могут зарегистрироваться в приложении не только как туристы, но и как арендодатели. Арендодатели могут размещать в приложении снаряжение, которое будут сдавать в аренду обычным пользователям;

• В приложении можно делиться результатами своих походов. Затраченное время, перепад высот и т.д.;

• Туристы могут регистрироваться на какой-либо поход в этом приложении. В приложении будет отмечено, что пользователи находятся на маршруте. Функция отслеживания местоположения пользователя в реальном времени, чтобы другие пользователи могли видеть, где он находится на маршруте. В случае необходимости данные могут быть переданы в МЧС;

• В приложении можно объявить о поиске компании для похода;

• В приложении можно получать достижения. При прохождении определенного количества походов, набора высоты и т.д. путешественникам присваиваются достижения, которые отображаются в его профиле;

• Функция поиска и фильтрации маршрутов по различным параметрам, таким как сложность, продолжительность, тип активности и т.д;

• Функция создания и присоединения к спискам желаний, чтобы пользователи могли сохранять маршруты, снаряжение и другие элементы, которые они хотели бы попробовать в будущем;

• Возможность создания и присоединения к форумам и сообществам, где пользователи могут общаться, делиться опытом, задавать вопросы и получать советы от других путешественников;

• Возможность создания списка друзей, чтобы пользователи могли следить за активностью своих друзей и получать уведомления о их новых маршрутах, достижениях и т.д.

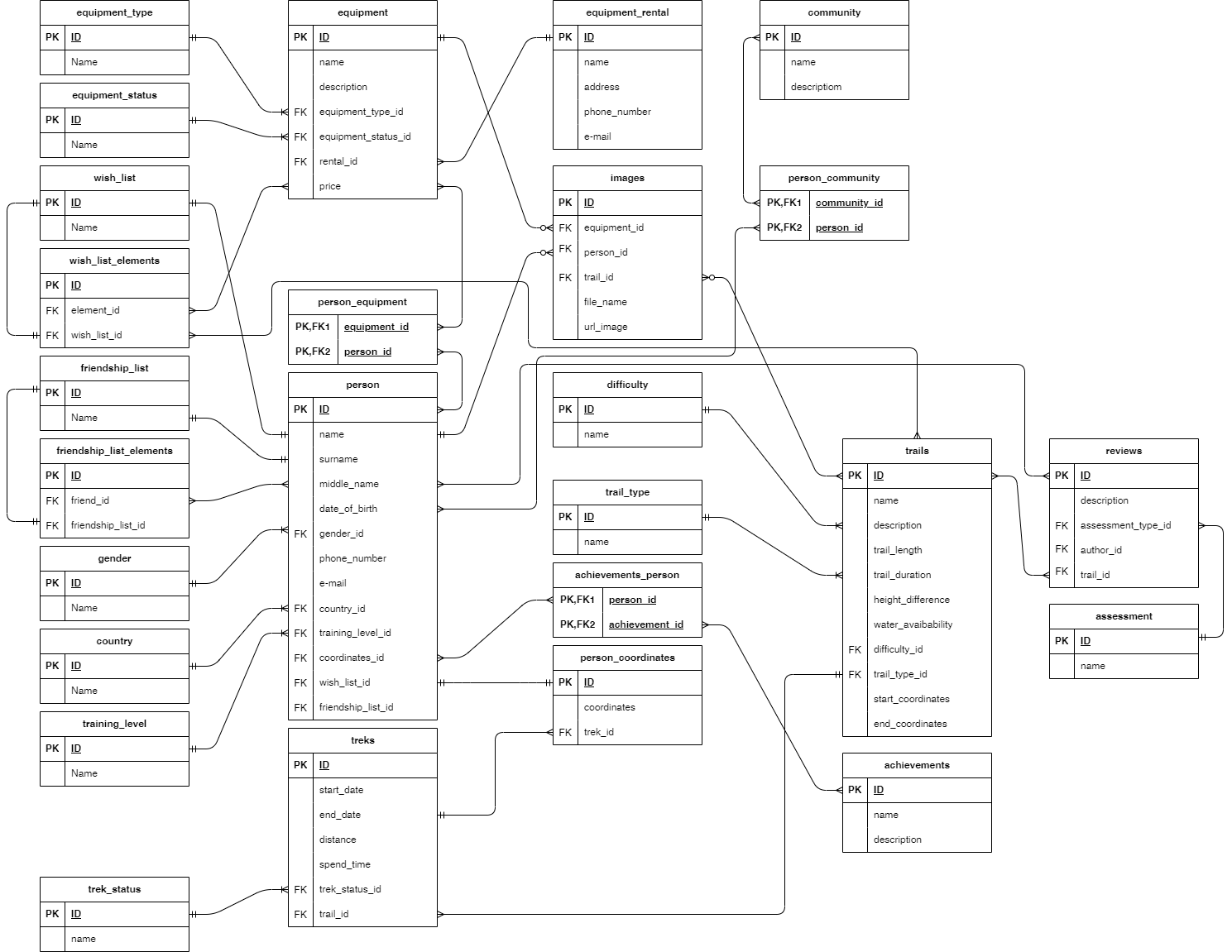
### 1.2 Выделим основные сущности базы данных:

1. Пользователи: информация о пользователях, включая их профили, контактные данные, достижения, списки друзей и конкурентов.
2. Маршруты: информация о маршрутах, включая их описание, сложность, продолжительность, тип активности, фото, отзывы и т.д.
3. Арендодатели: информация о арендодателях, включая их профили, контактные данные, список снаряжения, которое они предлагают в аренду.
4. Снаряжение: информация о снаряжении, которое можно арендовать, включая его описание, стоимость, доступность и т.д.
5. Походы: информация о походах, включая их описание, сложность, продолжительность, тип активности, список участников, результаты и т.д.
6. Сообщества: информация о сообществах, включая их названия, описание, список участников и т.д.
7. Списки желаний: информация о списках желаний, включая их описание, список элементов, которые пользователи хотели бы попробовать в будущем.
8. Списки друзей.
9. Достижения: информация о достижениях, включая их описание, условия получения, список пользователей, которые их достигли и т.д.
10. Отзывы: информация о отзывах, включая их описание, оценку, автора и т.д.
11. Местоположение: информация о местоположении пользователей, включая их текущее местоположение, маршрут, на котором они находятся и т.д.

### 1.3 Дополнительные сущности базы данных

Дополнительные сущности представляют собой справочники, на которые ссылаются основные сущности.

1. Страны – country;
2. Уровень подготовки туриста – training\_level;
3. Статус похода – trek\_status;
4. Пол – gender;
5. Статус доступности экипировки, сдаваемой в аренду – equipment\_status;
6. Тип экипировки – equipment\_type;
7. Тип маршрута – trail\_type;
8. Тип оценки – assessment;
9. Сложность маршрута – difficulty.

Рисунок 1 – ER диаграмма.

### 1.4 Анализ информационных задач и круга пользователей системы

Приложение создается для обслуживания следующих групп пользователей:

* Администрация приложения;
* Туристы;
* Арендодатели снаряжения.

**Функциональные возможности:**

* Ведение БД (запись, чтение, модификация, удаление);
* Обеспечение логической непротиворечивости БД;
* Обеспечение защиты данных от несанкционированного или случайного доступа (определение прав доступа);
* Реализация наиболее часто встречающихся запросов в готовом виде.

**Готовые запросы:**

Получить список людей и их достижений и получить список людей на маршрутах.

### 2. Логическое проектирование реляционной БД

В таблицах ниже представлены сущности БД и их свойства.

Первым делом будут перечислены сущности, не имеющие внешних ключей.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **country** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Наименование | Name | varchar, 25 |  | NOT\_NULL |

Таблица 1 – схема отношения «Страна»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **training\_level** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Наименование | Name | varchar, 25 |  | NOT\_NULL |

Таблица 2 – схема отношения «Уровень подготовки»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **trek\_status** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Наименование | Name | varchar, 25 |  | NOT\_NULL |

Таблица 3 – схема отношения «Статус похода»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **gender** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Наименование | Name | varchar, 25 |  | NOT\_NULL |

Таблица 4 – схема отношения «Пол»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **equipment\_status** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Наименование | Name | varchar, 25 |  | NOT\_NULL |

Таблица 5 – схема отношения «Статус снаряжения»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **equipment\_type** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Наименование | Name | varchar, 25 |  | NOT\_NULL |

Таблица 6 – схема отношения «Тип экипировки»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **trail\_type** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Наименование | Name | varchar, 25 |  | NOT\_NULL |

Таблица 7 – схема отношения «Тип маршрута»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **assessment** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Наименование | Name | varchar, 25 |  | NOT\_NULL |

Таблица 8 – схема отношения «Тип оценки»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **difficulty** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Наименование | Name | varchar, 25 |  | NOT\_NULL |

Таблица 9 – схема отношения «Тип сложности»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **wish\_list** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Наименование | Name | varchar, 25 |  | NOT\_NULL |

Таблица 10 – схема отношения «Лист желаний»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **friendship\_list** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Наименование | Name | varchar, 25 |  | NOT\_NULL |

Таблица 11 – схема отношения «Список друзей»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **community** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Наименование | name | varchar, 100 |  | NOT\_NULL |
| Описание | description | varchar, 100 |  | NULL |

Таблица 12 – схема отношения «Список сообществ»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **equipment\_rental** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Наименование | name | varchar, 25 |  | NOT\_NULL |
| Адрес | adress | varchar, 100 |  | NOT\_NULL |
| Номер телефона | phone\_number | varchar, 11 |  | NOT\_NULL |
| E-mail | email | varchar, 25 |  | NULL |

Таблица 13 – схема отношения «Арендодатели снаряжения»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **achievements** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Наименование | name | varchar, 100 |  | NOT\_NULL |
| Описание | description | varchar, 100 |  | NULL |

Таблица 14 – схема отношения «Достижения»

Далее будут перечислены сущности, которые имеют внешние ключи, ссылающиеся на ранее заявленные сущности.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **achievements\_person** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| Достижение | achievement\_id | int | FOREIGN\_KEY (achievements) | NOT\_NULL |
| Персоналия | person\_id | int | FOREIGN\_KEY (person) | NOT\_NULL |

Таблица 15 – кросс-таблица для связи таблиц achievements и person

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **trails** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Наименование | name | varchar, 100 |  | NOT\_NULL |
| Описание | description | varchar, 100 |  | NULL |
| Длина маршрута | trail\_lenght | DECIMAL(10, 2) |  | NOT\_NULL |
| Длительность маршрута | trail\_duration | DECIMAL(10, 2) |  | NOT\_NULL |
| Перепад высот | height\_difference | DECIMAL(10, 2) |  | NOT\_NULL |
| Вода | water\_avaibability | boolean |  | NOT\_NULL |
| Сложность | difficulty\_id | int | FOREIGN\_KEY (difficulty) | NOT\_NULL |
| Тип активности | trail\_type\_id | int | FOREIGN\_KEY (trail\_type) | NOT\_NULL |
| Координаты старта | start\_coordinates | geography |  | NULL |
| Координаты финиша | end\_coordinates | geography |  | NULL |

Таблица 16 – схема отношения «Маршруты»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **treks** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Время старта | start\_date | timestamp |  | NOT\_NULL |
| Время финиша | end\_date | timestamp |  | NULL |
| Дистанция | distance | DECIMAL(10, 2) |  | NULL |
| Потраченное время | spend\_time | bigint |  | NULL |
| Статус похода | trek\_status\_id | int | FOREIGN\_KEY (trek\_status) | NOT\_NULL |
| Маршрут | trail\_id | int | FOREIGN\_KEY (trail\_id) | NULL |

Таблица 17 – схема отношения «Походы»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **person\_community** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| Сообщество | community\_id | int | FOREIGN\_KEY (community) | NOT\_NULL |
| Персоналия | person\_id | int | FOREIGN\_KEY (person) | NOT\_NULL |

Таблица 17 – кросс-таблица для связи таблиц community и person

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **equipment** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Имя | name | varchar, 45 |  | NOT\_NULL |
| Описание | description | varchar, 200 |  | NULL |
| Тип снаряжения | epuipment\_type\_id | int | FOREIGN\_KEY (equipment\_type) | NOT\_NULL |
| Статус снаряжения | equipment\_status\_id | int | FOREIGN\_KEY (equipment\_status) | NOT\_NULL |
| Арендодатель | rental\_id | int | FOREIGN\_KEY (equipment\_rental) | NOT\_NULL |
| Стоимость | price | DECIMAL(10, 2) |  | NULL |
| Фото | equip\_photo\_id | int |  | NULL |

Таблица 18 – схема отношения «Снаряжение»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **person\_coordinates** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Координаты | coorginates | geography |  | NULL |
| Поход | trek\_id | int | FOREIGN\_KEY (treks) | NULL |

Таблица 19 – схема отношения «Координаты туриста»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **images** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Экипировка | equipment\_id | int | FOREIGN\_KEY (equipment) | NULL |
| Пользователь | person\_id | int | FOREIGN\_KEY (person) | NULL |
| Маршрут | trail\_id | int | FOREIGN\_KEY (trails) | NULL |
| Наименование файла | file\_name | varchar, 100 |  | NULL |
| Ссылка на файл | url\_image | varchar, 100 |  | NULL |

Таблица 20 – схема отношения «Изображения»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **rewiews** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| Описание | description | varchar, 250 |  | NOT\_NULL |
| Оценка | assessment\_type\_id | int | FOREIGN\_KEY (asessment) | NOT\_NULL |
| Автор | author\_id | int | FOREIGN\_KEY (person) | NOT\_NULL |
| Маршрут | trail\_id | int | FOREIGN\_KEY (trails) | NOT\_NULL |

Таблица 21 – схема отношения «Отзывы о маршрутах»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **wish\_list\_elements** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| id элемента списка (маршрут) | element\_id\_trail | int | FOREIGN\_KEY (trails) | NULL |
| id элемента списка (экипировка) | element\_id\_equip | int | FOREIGN\_KEY (equipment) | NULL |
| id списка желаний | wish\_list\_id | int | FOREIGN\_KEY (wish\_list) | NOT\_NULL |

Таблица 22 – схема отношения «Элементы списка желаний»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **friendship\_list\_elements** | | | | |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** | **Обязательно** |
| ID | ID | int | PRIMARY\_KEY | NOT\_NULL |
| id элемента списка | friend\_id | int | FOREIGN\_KEY (person) | NOT\_NULL |
| id списка желаний | friendship\_list\_id | int | FOREIGN\_KEY (friendship\_list) | NOT\_NULL |

Таблица 22 – схема отношения «Элементы списка друзей»

### 3. Физическое проектирование БД

### 3.1 Установка PostGIS

Перед созданием базы данных и таблиц в этой базе данных необходимо учесть, что в таблицах «trails» и «person\_coordinates» есть поля, для которых используется тип данных **geography**. Данный тип данных используется для записи координат.

Для того, чтобы в базе данных появилась возможность записывать координаты пользователя необходимо установить расширение PostGIS. PostGIS: это расширение для PostgreSQL, которое добавляет поддержку географических типов данных, функций и операторов для работы с пространственными данными.

Для этого с помощью «Stack Builder» необходимо загрузить и установить дистрибутив данного расширения (см. рисунки 2, 3, 4, 5).

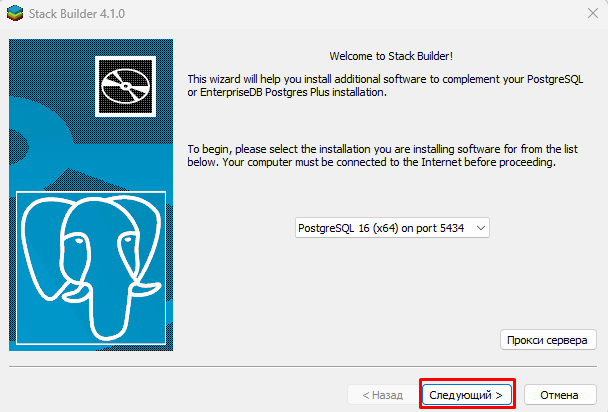


Рисунок 2 – установка PostGIS

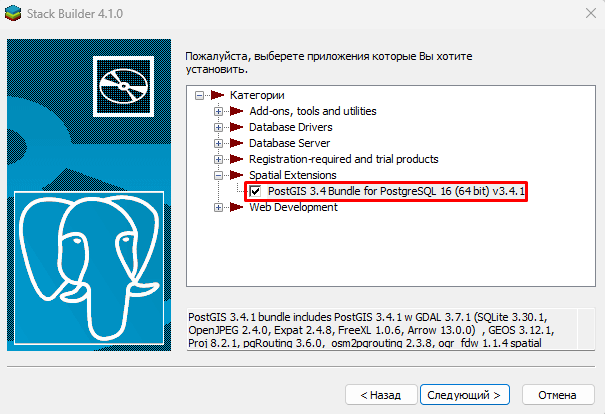


Рисунок 3 – установка PostGIS

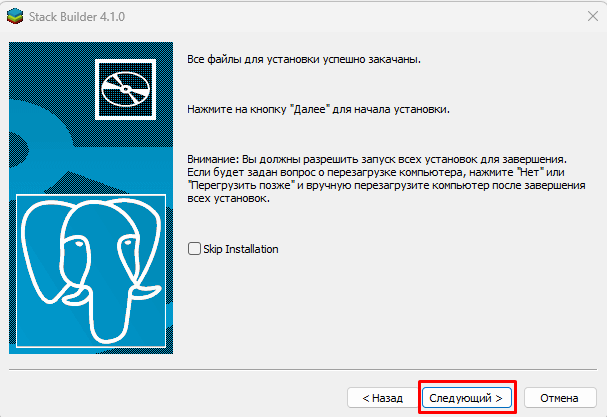


Рисунок 4 – установка PostGIS

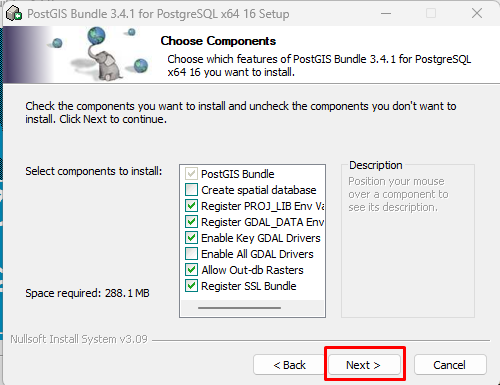


Рисунок 5 – установка PostGIS

После чего необходимо подключиться к базе данных и инициализировать установленное расширение с помощью скрипта:

**CREATE** **DATABASE** gisdb;

\**connect** gisdb;

-- Enable PostGIS (includes raster)

**CREATE** **EXTENSION** postgis;

### 3.2 Создание БД

БД «trekking\_app» проектируется в среде PostgreSQL. Для непосредственной работы используется программа «DBeaver».

Для физического создания БД в программе «DBeaver» необходимо выполнить скрипты. Каждый скрипт выполняет то или иное действие по созданию БД.

Создание базы данных «trekking\_app»:

**CREATE** **DATABASE** trekking\_app;

### 3.3 Создание таблиц

Создание таблиц необходим начать с справочников, не имеющих внешних ключей. Это необходимо для того, чтобы при создании таблиц, которые ссылаются на эти справочники, не возникало ошибок.

Создание таблицы country

**CREATE** **TABLE** country (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**Name** **VARCHAR**(25) **NOT** **NULL**

);

Создание таблицы training\_level

**CREATE** **TABLE** training\_level (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**Name** **VARCHAR**(25) **NOT** **NULL**

);

Создание таблицы trek\_status

**CREATE** **TABLE** trek\_status (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**Name** **VARCHAR**(25) **NOT** **NULL**

);

Создание таблицы gender

**CREATE** **TABLE** gender (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**Name** **VARCHAR**(25) **NOT** **NULL**

);

Создание таблицы equipment\_status

**CREATE** **TABLE** equipment\_status (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**Name** **VARCHAR**(25) **NOT** **NULL**

);

Создание таблицы equipment\_type

**CREATE** **TABLE** equipment\_type (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**Name** **VARCHAR**(25) **NOT** **NULL**

);

Создание таблицы trail\_type

**CREATE** **TABLE** trail\_type (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**Name** **VARCHAR**(25) **NOT** **NULL**

);

Создание таблицы assessment

**CREATE** **TABLE** assessment (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**Name** **VARCHAR**(25) **NOT** **NULL**

);

Создание таблицы difficulty

**CREATE** **TABLE** difficulty (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**Name** **VARCHAR**(25) **NOT** **NULL**

);

Создание таблицы wish\_list

**CREATE** **TABLE** wish\_list (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**Name** **VARCHAR**(25) **NOT** **NULL**

);

Создание таблицы friendship\_list

**CREATE** **TABLE** friendship\_list (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**Name** **VARCHAR**(25) **NOT** **NULL**

);

Создание таблицы equipment\_rental

**CREATE** **TABLE** equipment\_rental (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**name** **VARCHAR**(25) **NOT** **NULL**,

adress **VARCHAR**(100) **NOT** **NULL**,

phone\_number **VARCHAR**(25) **NOT** **NULL**,

email **VARCHAR**(25)

);

Далее создаются более сложные таблицы, которые имеют внешние ключи, ссылающиеся на созданные ранее таблицы.

Создание таблицы trails

**CREATE** **TABLE** trails (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**name** **VARCHAR**(100) **NOT** **NULL**,

description **VARCHAR**(100),

trail\_lenght **DECIMAL**(10, 2) **NOT** **NULL**,

trail\_duration **DECIMAL**(10, 2) **NOT** **NULL**,

height\_difference **DECIMAL**(10, 2) **NOT** **NULL**,

water\_avaibability **BOOLEAN** **NOT** **NULL**,

difficulty\_id **INT** **NOT** **NULL**,

trail\_type\_id **INT** **NOT** **NULL**,

start\_coordinates geography,

end\_coordinates geography,

**FOREIGN** **KEY** (difficulty\_id) **REFERENCES** difficulty (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**FOREIGN** **KEY** (trail\_type\_id) **REFERENCES** trail\_type (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

);

Создание таблицы equipment

**CREATE** **TABLE** equipment (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**name** **VARCHAR**(25) **NOT** **NULL**,

description **VARCHAR**(100),

equipment\_type\_id **INT** **NOT** **NULL**,

equipment\_status\_id **INT** **NOT** **NULL**,

rental\_id **INT**,

price **DECIMAL**(10, 2),

**FOREIGN** **KEY** (equipment\_type\_id) **REFERENCES** equipment\_type (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**FOREIGN** **KEY** (equipment\_status\_id) **REFERENCES** equipment\_status (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**FOREIGN** **KEY** (rental\_id) **REFERENCES** equipment\_rental (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

);

Создание таблицы wish\_list\_elements

**CREATE** **TABLE** wish\_list\_elements (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

element\_id\_equip **INT**,

element\_id\_trail **INT**,

wish\_list\_id **INT** **NOT** **NULL**,

**FOREIGN** **KEY** (element\_id\_equip) **REFERENCES** equipment (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**FOREIGN** **KEY** (element\_id\_trail) **REFERENCES** trails (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**FOREIGN** **KEY** (wish\_list\_id) **REFERENCES** wish\_list (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

);

Создание таблицы treks

**CREATE** **TABLE** treks (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

start\_date **TIMESTAMP** **NOT** **NULL**,

end\_date **TIMESTAMP**,

distance **DECIMAL**(10, 2),

spend\_time **BIGINT**,

trek\_status\_id **INT** **NOT** **NULL**,

trail\_id **INT**,

**FOREIGN** **KEY** (trek\_status\_id) **REFERENCES** trek\_status (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**FOREIGN** **KEY** (trail\_id) **REFERENCES** trails (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

);

Создание таблицы person\_coordinates

**CREATE** **TABLE** person\_coordinates (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

coordinates GEOGRAPHY, -- тип данных GEOGRAPHY используется для работы с географическими координатами

trek\_id **INT**,

**FOREIGN** **KEY** (trek\_id) **REFERENCES** treks(ID) -- ссылка на таблицу treks по полю ID

);

Создание таблицы person

**CREATE** **TABLE** person (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**name** **VARCHAR**(25) **NOT** **NULL**,

surname **VARCHAR**(25) **NOT** **NULL**,

middle\_name **VARCHAR**(25) **NOT** **NULL**,

date\_of\_birth **DATE** **NOT** **NULL**,

gender\_id **INT** **NOT** **NULL**,

phone\_number **VARCHAR**(11) **NOT** **NULL**,

email **VARCHAR**(25),

country\_id **INT** **NOT** **NULL**,

training\_level\_id **INT** **NOT** **NULL**,

coordinates\_id **INT**,

wish\_list\_id **INT**,

friendship\_list\_id **INT**,

**FOREIGN** **KEY** (country\_id) **REFERENCES** country (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**FOREIGN** **KEY** (training\_level\_id) **REFERENCES** training\_level (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**FOREIGN** **KEY** (coordinates\_id) **REFERENCES** person\_coordinates (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**FOREIGN** **KEY** (wish\_list\_id) **REFERENCES** wish\_list (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**FOREIGN** **KEY** (friendship\_list\_id) **REFERENCES** friendship\_list (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

);

Создание таблицы friendship\_list\_elements

**CREATE** **TABLE** friendship\_list\_elements (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

friend\_id **INT** **NOT** **NULL**,

friendship\_list\_id **INT** **NOT** **NULL**,

**FOREIGN** **KEY** (friend\_id) **REFERENCES** person (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**FOREIGN** **KEY** (friendship\_list\_id) **REFERENCES** friendship\_list (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

);

Создание таблицы community

**CREATE** **TABLE** community (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**name** **VARCHAR**(100) **NOT** **NULL**,

description **VARCHAR**(100)

);

Создание таблицы person\_community

**CREATE** **TABLE** person\_community (

community\_id **INT** **NOT** **NULL**,

person\_id **INT** **NOT** **NULL**,

**FOREIGN** **KEY** (community\_id) **REFERENCES** community (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**FOREIGN** **KEY** (person\_id) **REFERENCES** person (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

);

Создание таблицы achievements

**CREATE** **TABLE** achievements (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

**name** **VARCHAR**(100) **NOT** **NULL**,

description **VARCHAR**(100)

);

Создание таблицы achievements\_person

**CREATE** **TABLE** achievements\_person (

achievement\_id **INT** **NOT** **NULL**,

person\_id **INT** **NOT** **NULL**,

**FOREIGN** **KEY** (achievement\_id) **REFERENCES** achievements (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**FOREIGN** **KEY** (person\_id) **REFERENCES** person (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

);

Создание таблицы reviews

**CREATE** **TABLE** reviews (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

description **VARCHAR**(250) **NOT** **NULL**,

assessment\_type\_id **INT** **NOT** **NULL**,

author\_id **INT** **NOT** **NULL**,

trail\_id **INT** **NOT** **NULL**,

**FOREIGN** **KEY** (assessment\_type\_id) **REFERENCES** assessment (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**FOREIGN** **KEY** (author\_id) **REFERENCES** person (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**,

**FOREIGN** **KEY** (trail\_id) **REFERENCES** trails (ID)

**ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

);

Создание таблицы images

**CREATE** **TABLE** images (

ID **INT** **PRIMARY** **KEY** **NOT** **NULL**,

equipment\_id **INT**,

person\_id **INT**,

trail\_id **INT**,

file\_name **VARCHAR**(100),

url\_image **VARCHAR**(100),

**FOREIGN** **KEY** (equipment\_id) **REFERENCES** equipment (ID)

**ON** **DELETE** **SET** **NULL**,

**FOREIGN** **KEY** (person\_id) **REFERENCES** person (ID)

**ON** **DELETE** **SET** **NULL**,

**FOREIGN** **KEY** (trail\_id) **REFERENCES** trails (ID)

**ON** **DELETE** **SET** **NULL**

);

**ON** **DELETE** **SET** **NULL**, **ON** **DELETE** **CASCADE** и **ON** **UPDATE** **CASCADE** - это команды, которые определяют поведение внешнего ключа в базе данных при выполнении операций удаления или обновления.

**ON DELETE SET NULL:**

Когда используется ON DELETE SET NULL, это означает, что если родительская запись, на которую ссылается внешний ключ, будет удалена, то значение внешнего ключа в дочерней записи будет установлено в NULL. Это предотвращает нарушение ссылочной целостности, так как дочерняя запись больше не ссылается на несуществующую родительскую запись.

**ON DELETE CASCADE:**

ON DELETE CASCADE указывает, что при удалении родительской записи все дочерние записи, которые ссылаются на нее, также должны быть удалены. Это позволяет поддерживать целостность данных, так как все связанные записи удаляются вместе.

**ON UPDATE CASCADE:**

ON UPDATE CASCADE работает аналогично ON DELETE CASCADE, но для операций обновления. Если значение в родительской записи изменяется, то все дочерние записи, которые ссылаются на это значение, также должны быть обновлены.

В результате выполнения всех представленных ранее скриптов на создание таблиц, DBeaver автоматически формирует ER-диаграмму по созданной БД, которая представлена на рисунке 6.

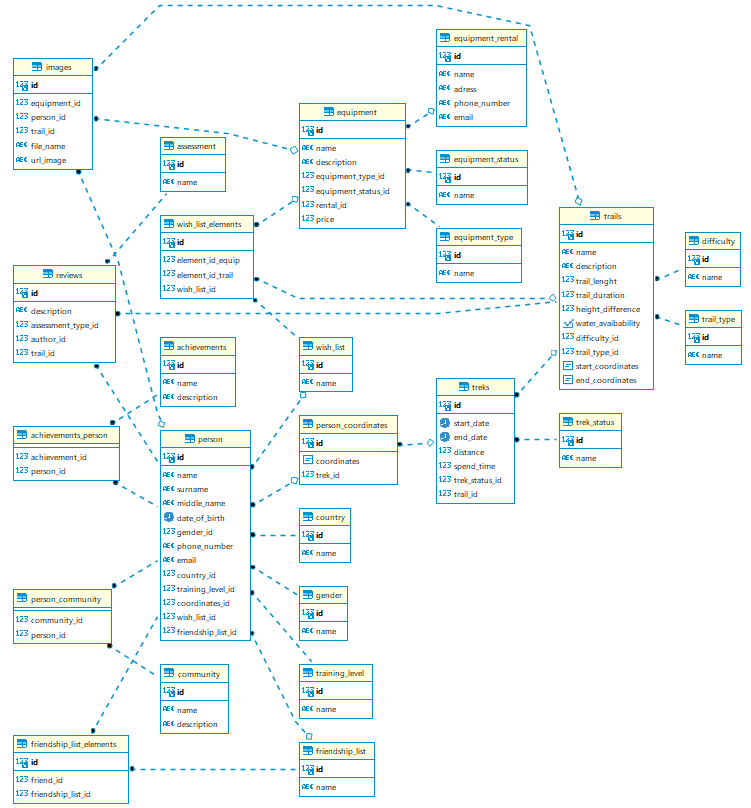


Рисунок 6 – ER-диаграмма.

### 4. Добавление данных в БД

### 4.1 Создание последовательности

Перед тем, как добавлять данные в таблицу, необходимо позаботиться о том, чтобы БД автоматически генерировала идентификаторы (ID) добавляемых в неё объектов. Для выполнения этой задачи я буду использовать последовательность – SEQUENCE.

Я хочу, чтобы ID были шестизначными, поэтому начну последовательность с 1000000. Для того, чтобы создать такую последовательность, я буду использовать скрипт:

**CREATE** **SEQUENCE** sequence\_generator\_id

**START** **WITH** 1000000

**INCREMENT** **BY** 1

**NO** **MINVALUE**

**NO** **MAXVALUE**

**CACHE** 1;

Здесь sequence\_generator\_id – это имя моей последовательности.

Параметры START WITH 1, INCREMENT BY 1, NO MINVALUE, NO MAXVALUE и CACHE 1 определяют начальное значение, шаг увеличения, отсутствие минимального и максимального значений, а также размер кэша соответственно.

### 4.2 Написание скриптов на добавление данных.

Добавление данных в таблицу country

**INSERT** **INTO** country (ID, **Name**) **VALUES**

(1, 'Россия'),

(2, 'Австралия'),

(3, 'Австрия'),

(4, 'Бангладеш'),

(5, 'Барбадос'),

(6, 'Беларусь'),

(7, 'Белиз'),

(8, 'Бельгия'),

(9, 'Бенин'),

(10, 'Болгария'),

(11, 'Ботсвана'),

(12, 'Бразилия'),

(13, 'Бруней'),

(14, 'Буркина-Фасо'),

(15, 'Бурунди'),

(16, 'Вануату'),

(17, 'Великобритания'),

(18, 'Венгрия'),

(19, 'Вьетнам'),

(20, 'Габон'),

(21, 'Гамбия'),

(22, 'Гана'),

(23, 'Гваделупа'),

(24, 'Гватемала'),

(25, 'Гвинея'),

(26, 'Гвинея-Бисау'),

(27, 'Германия'),

(28, 'Гондурас'),

(29, 'Гонконг'),

(30, 'Гренада'),

(31, 'Греция'),

(32, 'Грузия'),

(33, 'Дания'),

(34, 'Джибути'),

(35, 'Доминика'),

(36, 'Доминиканская Республика'),

(37, 'Египет'),

(38, 'Замбия'),

(39, 'Зимбабве'),

(40, 'Исландия'),

(41, 'Индия'),

(42, 'Индонезия'),

(43, 'Иордания'),

(44, 'Ирак'),

(45, 'Иран'),

(46, 'Ирландия'),

(47, 'Исландия'),

(48, 'Италия'),

(49, 'Ямайка'),

(50, 'Япония'),

(51, 'Джибути'),

(52, 'Кения'),

(53, 'Кипр'),

(54, 'Киргизия'),

(55, 'Кирибати'),

(56, 'Китай'),

(57, 'Колумбия'),

(58, 'Коморы'),

(59, 'Конго'),

(60, 'Коста-Рика'),

(61, 'Кот-д Ивуар'),

(62, 'Куба'),

(63, 'Кувейт'),

(64, 'Лаос'),

(65, 'Латвия'),

(66, 'Лесото'),

(67, 'Либерия'),

(68, 'Ливан'),

(69, 'Ливия'),

(70, 'Литва'),

(71, 'Люксембург'),

(72, 'Маврикий'),

(73, 'Мавритания'),

(74, 'Мадагаскар'),

(75, 'Малави'),

(76, 'Малайзия'),

(77, 'Мали'),

(78, 'Мальдивы'),

(79, 'Мальта'),

(80, 'Марокко'),

(81, 'Маршалловы Острова'),

(82, 'Мексика'),

(83, 'Мозамбик'),

(84, 'Монако'),

(85, 'Монголия'),

(86, 'Намибия'),

(87, 'Науру'),

(88, 'Непал'),

(89, 'Нигер'),

(90, 'Нигерия'),

(91, 'Нидерланды'),

(92, 'Новая Зеландия'),

(93, 'Норвегия'),

(94, 'Оман');

Добавление данных в таблицу training\_level

**INSERT** **INTO** training\_level (ID, **Name**) **VALUES**

(1, 'Начинающий'),

(2, 'Средний'),

(3, 'Продвинутый'),

(4, 'Эксперт');

Добавление данных в таблицу trek\_status

**INSERT** **INTO** trek\_status (ID, **Name**) **VALUES**

(1, 'Планируется'),

(2, 'В процессе'),

(3, 'Завершен'),

(4, 'Отменен');

Добавление данных в таблицу gender

**INSERT** **INTO** gender (ID, **Name**) **VALUES**

(1, 'Мужской'),

(2, 'Женский');

Добавление данных в таблицу equipment\_status

**INSERT** **INTO** equipment\_status (ID, **Name**) **VALUES**

(1, 'Доступно'),

(2, 'Забронировано'),

(3, 'Недоступно'),

(4, 'В ремонте');

Добавление данных в таблицу equipment\_type

**INSERT** **INTO** equipment\_type (ID, **Name**) **VALUES**

(1, 'Рюкзак'),

(2, 'Палатка'),

(3, 'Коврик'),

(4, 'Спальный мешок'),

(5, 'Треккинговые палки'),

(6, 'Ботинки'),

(7, 'Куртка'),

(8, 'Брюки'),

(9, 'Шапка'),

(10, 'Перчатки'),

(11, 'Горнолыжные ботинки'),

(12, 'Горнолыжная куртка'),

(13, 'Горнолыжные брюки'),

(14, 'Горнолыжные очки'),

(15, 'Горнолыжные перчатки'),

(16, 'Крепления'),

(17, 'Лыжи'),

(18, 'Сноуборд'),

(19, 'Крепления для сноуборда'),

(20, 'Крепления для лыж');

Добавление данных в таблицу trail\_type

**INSERT** **INTO** trail\_type (ID, **Name**) **VALUES**

(1, 'Треккинг'),

(2, 'Альпинизм'),

(3, 'Скалолазание'),

(4, 'Ски-тур'),

(5, 'Бэккантри'),

(6, 'Фрирайд'),

(7, 'Ски-альпинизм'),

(8, 'Трэил-раннинг');

Добавление данных в таблицу assessment

**INSERT** **INTO** assessment (ID, **Name**) **VALUES**

(1, 'Очень плохо'),

(2, 'Плохо'),

(3, 'Удовлетворительно'),

(4, 'Хорошо'),

(5, 'Отлично');

Добавление данных в таблицу difficulty

**INSERT** **INTO** difficulty (ID, **Name**) **VALUES**

(1, 'Легкий'),

(2, 'Средний'),

(3, 'Сложный'),

(4, 'Очень сложный');

Добавление данных в таблицу equipment\_rental

**INSERT** **INTO** equipment\_rental (ID, **name**, adress, phone\_number, email) **VALUES**

(**nextval**('sequence\_generator\_id'), 'Экстрим-тур', 'ул. Кирова, д. 1, Пятигорск', '78791234567', 'extremetour@example.com'),

(**nextval**('sequence\_generator\_id'), 'Горный мир', 'ул. Ленина, д. 20, Ессентуки', '78799876543', 'gornymir@example.com'),

(**nextval**('sequence\_generator\_id'), 'Вершина', 'ул. Мира, д. 30, Кисловодск', '78796543210', 'vershina@example.com'),

(**nextval**('sequence\_generator\_id'), 'Туризм+', 'пр. Победы, д. 15, Железноводск', '78793456789', 'turizmplus@example.com'),

(**nextval**('sequence\_generator\_id'), 'Горный клуб', 'ул. Советская, д. 25, Нальчик', '78664567890', 'gornyclub@example.com');

Добавление данных в таблицу trails

**INSERT** **INTO** trails (ID, **name**, description, trail\_lenght, trail\_duration, height\_difference, water\_avaibability, difficulty\_id, trail\_type\_id, start\_coordinates, end\_coordinates) **VALUES**

(1, 'Тропа к озеру', 'Тропа ведет к живописному озеру', 10000.00, 4.00, 300.00, **TRUE**, 1, 1, 'POINT(55.7558261 37.6173891)', 'POINT(55.7558261 37.6173891)'),

(2, 'Тропа через лес', 'Тропа проходит через густой лес', 7500.75, 3.00, 250.00, **FALSE**, 2, 2, 'POINT(55.7558261 37.6173891)', 'POINT(55.7558261 37.6173891)'),

(3, 'Тропа к водопаду', 'Тропа ведет к красивому водопаду', 20000.00, 24.50, 1000.00, **TRUE**, 3, 3, 'POINT(55.7558261 37.6173891)', 'POINT(55.7558261 37.6173891)'),

(4, 'Тропа к горной вершине', 'Тропа ведет к горной вершине', 25000.00, 32.00, 1400.00, **FALSE**, 4, 4, 'POINT(55.7558261 37.6173891)', 'POINT(55.7558261 37.6173891)'),

(5, 'Тропа к древнему монастырю', 'Тропа ведет к древнему монастырю', 17700.50, 18.00, 800.00, **TRUE**, 3, 5, 'POINT(55.7558261 37.6173891)', 'POINT(55.7558261 37.6173891)'),

(6, 'Тропа к горному хребту', 'Тропа ведет к горному хребту', 5000.00, 3.00, 500.00, **FALSE**, 2, 6, 'POINT(55.7558261 37.6173891)', 'POINT(55.7558261 37.6173891)'),

(7, 'Тропа к горному озеру', 'Тропа ведет к горному озеру', 4000.00, 2.5, 300.00, **TRUE**, 4, 7, 'POINT(55.7558261 37.6173891)', 'POINT(55.7558261 37.6173891)'),

(8, 'Тропа к леднику', 'Тропа ведет к леднику', 4300.00, 1.5, 150.00, **FALSE**, 1, 8, 'POINT(55.7558261 37.6173891)', 'POINT(55.7558261 37.6173891)');

Добавление данных в таблицу equipment

**INSERT** **INTO** equipment (ID, **name**, description, equipment\_type\_id, equipment\_status\_id, rental\_id, price) **VALUES**

(1, 'Рюкзак', 'Описание рюкзака', 1, 1, 1000015, 2000.00),

(2, 'Палатка', 'Описание палатки', 2, 1, 1000016, 5000.00),

(3, 'Коврик', 'Описание коврика', 3, 1, 1000017, 1000.00),

(4, 'Спальный мешок', 'Описание спального мешка', 4, 1, 1000018, 3000.00),

(5, 'Треккинговые палки', 'Описание треккинговых палок', 5, 1, 1000019, 2500.00),

(6, 'Ботинки', 'Описание ботинок', 6, 1, 1000015, 4000.00),

(7, 'Куртка', 'Описание куртки', 7, 1, 1000016, 3500.00),

(8, 'Брюки', 'Описание брюк', 8, 1, 1000017, 2500.00),

(9, 'Шапка', 'Описание шапки', 9, 1, 1000018, 1000.00),

(10, 'Перчатки', 'Описание перчаток', 10, 1, 1000019, 1500.00),

(11, 'Горнолыжные ботинки', 'Описание горнолыжных ботинок', 11, 1, 1000015, 6000.00),

(12, 'Горнолыжная куртка', 'Описание горнолыжной куртки', 12, 1, 1000016, 5000.00),

(13, 'Горнолыжные брюки', 'Описание горнолыжных брюк', 13, 1, 1000017, 4000.00),

(14, 'Горнолыжные очки', 'Описание горнолыжных очков', 14, 1, 1000018, 3000.00),

(15, 'Горнолыжные перчатки', 'Описание горнолыжных перчаток', 15, 1, 1000019, 2000.00),

(16, 'Крепления', 'Описание креплений', 16, 1, 1000015, 2500.00),

(17, 'Лыжи', 'Описание лыж', 17, 1, 1000016, 8000.00),

(18, 'Сноуборд', 'Описание сноуборда', 18, 1, 1000017, 10000.00),

(19, 'Крепления для сноуборда', 'Описание креплений для сноуборда', 19, 1, 1000018, 4000.00),

(20, 'Крепления для лыж', 'Описание креплений для лыж', 20, 1, 1000019, 3500);

Добавление данных в таблицу person

**INSERT** **INTO** person (ID, **name**, surname, middle\_name, date\_of\_birth, gender\_id, phone\_number, email, country\_id, training\_level\_id, coordinates\_id, wish\_list\_id, friendship\_list\_id) **VALUES**

(**nextval**('sequence\_generator\_id'), 'Иван', 'Иванов', 'Иванович', '1990-01-01', 1, '+9161234567', 'ivanov@example.com', 1, 1, **NULL**, **NULL**, **NULL**),

(**nextval**('sequence\_generator\_id'), 'Алексей', 'Смирнов', 'Алексеевич', '1995-04-15', 1, '79265487965', 'alexey.smirnov@mail.com', 1, 2, **NULL**, **NULL**, **NULL**),

(**nextval**('sequence\_generator\_id'), 'Мария', 'Петрова', 'Мариванна', '1980-02-28', 2, '79151234567', 'mariap@example.com', 1, 3, **NULL**, **NULL**, **NULL**),

(**nextval**('sequence\_generator\_id'), 'Анна', 'Сидорова', 'Анна', '1998-05-05', 2, '79291234567', 'anna@example.com', 2, 1, **NULL**, **NULL**, **NULL**),

(**nextval**('sequence\_generator\_id'), 'Егор', 'Степанов', 'Егорович', '2001-03-10', 1, '79145678901', 'egor@example.com', 2, 2, **NULL**, **NULL**, **NULL**);

Добавление данных в таблицу achievements

**INSERT** **INTO** achievements (ID, **name**, description) **VALUES**

(1, 'Восхождение на Эльбрус', 'Восхождение на самую высокую вершину России и Европы'),

(2, 'Восхождение на Эверест', 'Восхождение на высочайшую вершину мира'),

(3, 'Первопрохождение', 'Первое восхождение на ранее непокоренную вершину'),

(4, 'Спуск на лыжах с вершины', 'Спуск на лыжах с вершины горы'),

(5, 'Спуск на сноуборде с вершины', 'Спуск на сноуборде с вершины горы'),

(6, 'Спуск на велосипеде с вершины', 'Спуск на велосипеде с вершины горы'),

(7, 'Прохождение трассы за отведенное время', 'Прохождение сложной трассы за установленное время'),

(8, 'Участие в горной гонке', 'Участие в горной гонке на время или на выносливость'),

(9, 'Установление рекорда', 'Установление рекорда скорости, высоты или дальности в горном спорте'),

(10, 'Организация горного лагеря', 'Организация горного лагеря для восхождения или отдыха');

Добавление данных в таблицу achievements\_person

**INSERT** **INTO** achievements\_person (achievement\_id, person\_id) **VALUES**

(1, 1000020),

(2, 1000020),

(3, 1000021),

(4, 1000021),

(3, 1000022),

(4, 1000022),

(5, 1000023),

(6, 1000023),

(7, 1000024),

(10, 1000024);

Добавление данных в таблицу community

**INSERT** **INTO** community (ID, **name**, description) **VALUES**

(1, 'Клуб любителей горных походов', 'Клуб для тех, кто любит покорять горные вершины'),

(2, 'Клуб пеших походов', 'Клуб для любителей долгих пеших прогулок'),

(3, 'Клуб водных походов', 'Клуб для любителей сплавов по рекам и озерам'),

(4, 'Клуб зимних походов', 'Клуб для любителей зимних приключений в горах и лесах'),

(5, 'Клуб велосипедных походов', 'Клуб для любителей путешествий на велосипедах');

Добавление данных в таблицу person\_community

**INSERT** **INTO** person\_community (community\_id, person\_id) **VALUES**

(5, 1000020),

(2, 1000020),

(3, 1000021),

(4, 1000021),

(3, 1000022),

(4, 1000022),

(5, 1000023),

(4, 1000023),

(2, 1000024),

(5, 1000024);

Добавление данных в таблицу reviews

**INSERT** **INTO** reviews (ID, description, assessment\_type\_id, author\_id, trail\_id) **VALUES**

(**nextval**('sequence\_generator\_id'), 'Отличная тропа к озеру, живописные виды', 5, 1000020, 1),

(**nextval**('sequence\_generator\_id'), 'Тропа через лес была утомительной', 2, 1000021, 2),

(**nextval**('sequence\_generator\_id'), 'Красивый водопад, тропа к нему великолепна', 5, 1000022, 3),

(**nextval**('sequence\_generator\_id'), 'Тропа к горной вершине была сложной, но впечатляющей', 4, 1000023, 4),

(**nextval**('sequence\_generator\_id'), 'Древний монастырь - прекрасное место, тропа к нему хорошая', 4, 1000024, 5),

(**nextval**('sequence\_generator\_id'), 'Тропа к горному хребту была средней сложности', 3, 1000020, 6),

(**nextval**('sequence\_generator\_id'), 'Тропа к горному озеру была легкой и приятной', 4, 1000021, 7),

(**nextval**('sequence\_generator\_id'), 'Ледник впечатляет, но тропа к нему непроходимая', 1, 1000022, 8);

Добавление данных в таблицу friendship\_list

**INSERT** **INTO** friendship\_list (ID) **VALUES**

(**nextval**('sequence\_generator\_id')),

(**nextval**('sequence\_generator\_id')),

(**nextval**('sequence\_generator\_id')),

(**nextval**('sequence\_generator\_id')),

(**nextval**('sequence\_generator\_id'));

Добавление данных в таблицу friendship\_list\_elements

**INSERT** **INTO** friendship\_list\_elements (id, friend\_id, friendship\_list\_id) **VALUES**

(**nextval**('sequence\_generator\_id'),1000021, 1000046),

(**nextval**('sequence\_generator\_id'),1000022, 1000046),

(**nextval**('sequence\_generator\_id'),1000023, 1000046),

(**nextval**('sequence\_generator\_id'),1000023, 1000047),

(**nextval**('sequence\_generator\_id'),1000024, 1000047),

(**nextval**('sequence\_generator\_id'),1000022, 1000047),

(**nextval**('sequence\_generator\_id'),1000021, 1000048),

(**nextval**('sequence\_generator\_id'),1000023, 1000048),

(**nextval**('sequence\_generator\_id'),1000020, 1000048),

(**nextval**('sequence\_generator\_id'),1000022, 1000049),

(**nextval**('sequence\_generator\_id'),1000024, 1000049),

(**nextval**('sequence\_generator\_id'),1000020, 1000050);

Добавление данных в таблицу images

**INSERT** **INTO** images (ID, equipment\_id, person\_id, trail\_id, file\_name, url\_image) **VALUES**

(**nextval**('sequence\_generator\_id'), 1, **NULL**, **NULL**, 'image1.jpg', 'url\_image1'),

(**nextval**('sequence\_generator\_id'), **NULL**, 1000021, **NULL**, 'image2.jpg', 'url\_image2'),

(**nextval**('sequence\_generator\_id'), **NULL**, **NULL**, 3, 'image3.jpg', 'url\_image3'),

(**nextval**('sequence\_generator\_id'), 4, **NULL**, **NULL**, 'image4.jpg', 'url\_image4'),

(**nextval**('sequence\_generator\_id'), **NULL**, 1000024, **NULL**, 'image5.jpg', 'url\_image5');

Добавление данных в таблицу treks

**INSERT** **INTO** treks (ID, start\_date, end\_date, distance, spend\_time, trek\_status\_id, trail\_id) **VALUES**

(**nextval**('sequence\_generator\_id'), '2023-01-01 00:00:00', **NULL**, 10.00, 10000, 1, 1),

(**nextval**('sequence\_generator\_id'), '2023-01-02 00:00:00', **NULL**, 20.00, 20000, 2, 2),

(**nextval**('sequence\_generator\_id'), '2023-01-03 00:00:00', '2023-01-03 00:00:00', 30.00, 30000, 3, 3), -- Завершен

(**nextval**('sequence\_generator\_id'), '2023-01-04 00:00:00', **NULL**, 40.00, 40000, 4, 4);

Добавление данных в таблицу person\_coordinates

**INSERT** **INTO** person\_coordinates (ID, coordinates , trek\_id) **VALUES**

(**nextval**('sequence\_generator\_id'), 'POINT(55.7558261 37.6173891)', 1000069),

(**nextval**('sequence\_generator\_id'), 'POINT(55.7558261 37.6173891)', 1000069),

(**nextval**('sequence\_generator\_id'), 'POINT(55.7558261 37.6173891)', **NULL**);

### 5. Написание готовых запросов

Вывести список достижений туристов:

**SELECT**

p."name" **AS** "Имя",

p.surname **AS** "Фамилия" ,

a."name" **AS** "Достижение" ,

a.description **AS** "Описание"

**FROM** person p

**JOIN** achievements\_person ap **ON** ap.person\_id = p.id

**JOIN** achievements a **ON** ap.achievement\_id = a.id

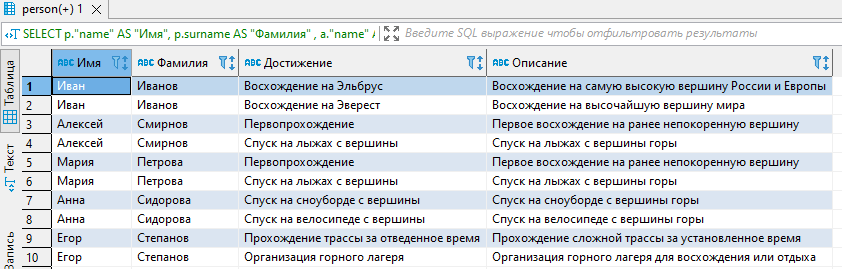


Рисунок 7 – Результат выполнения готового скрипта

Вывести информацию о туристах, находящихся на данный момент в походе:

**SELECT**

p."name" **AS** "Имя",

p.surname **AS** "Фамилия",

g."name" **AS** "Пол",

c."name" **AS** "Страна",

tl."name" **AS** "Уровень подготовки",

t2."name" **AS** "Маршрут",

pc.coordinates **AS** "Координаты"

**FROM** treks t

**JOIN** person\_coordinates pc **ON** pc.trek\_id = t.id

**JOIN** person p **ON** p.coordinates\_id = pc.id

**JOIN** trails t2 **ON** t2.id = t.trail\_id

**JOIN** country c **ON** c.id = p.country\_id

**JOIN** training\_level tl **ON** tl.id = p.training\_level\_id

**JOIN** gender g **ON** g.id = p.gender\_id

**WHERE** t.trek\_status\_id = 2

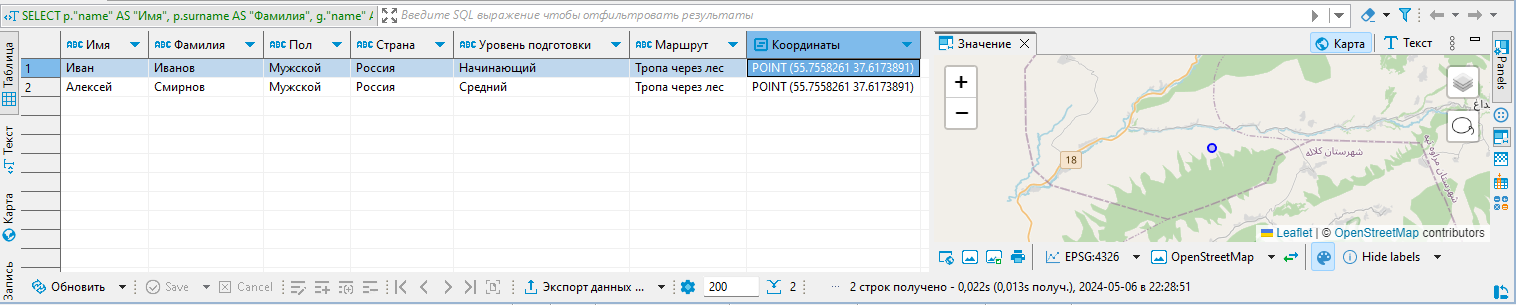


Рисунок 8 – Результат выполнения готового скрипта

### 6. Вывод

### В процессе выполнения курсовой работы изучено проектирование реляционной базы данных и реализация базы данных в среде PostgreSQL.