КОМИТЕТ ПО ОБРАЗОВАНИЮ ПРАВИТЕЛЬСТВА САНКТ-ПЕТЕРБУРГА

Санкт-Петербургское государственное  
бюджетное профессиональное образовательное учреждение  
«КОЛЛЕДЖ ЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

**«МДК.07.01 Управление и автоматизация баз данных»**

**ОТЧЁТ**

**по лабораторной работе №2**

**«Проектирование базы данных»**

Работу выполнил студент 325гр.:

Шлычков И. Д.

Преподаватель: Фомин А.В.

Санкт-Петербург 2025

**Введение**

В данной лабораторной я представил собой базу данных для хранения информации о регионах, странах, городах, среднесуточных измерениях температуры и координатах береговых линий.

В базе данных реализованы следующие связи:

* Один регион включает несколько стран.
* Одна страна включает несколько городов.
* В одном городе проводится множество измерений температуры.
* Береговая линия состоит из множества координатных точек

**Описание таблиц**

Таблица Region

Таблица Region содержит список регионов мира. Она необходима для группировки стран по регионам, что упрощает анализ данных по разным частям света. Таблица Region и также структура таблицы представлена на рисунке 1-2.

* region.identifier используется в таблице country как внешний ключ.

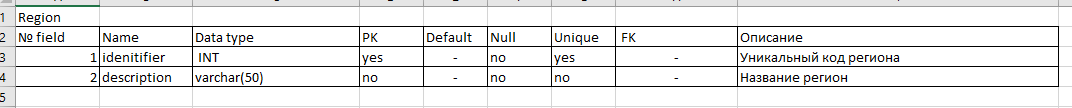


Рис 1 – таблица Region

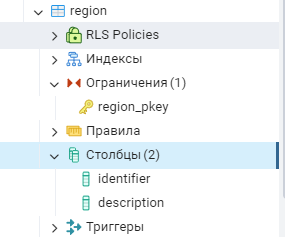


Рис 2 – структура таблицы Region

Таблица Country

Таблица Country содержит список стран и связывает их с регионами. Это позволяет определить, к какому региону принадлежит каждая страна. Таблица Country и также структура таблицы представлена на рисунке 3-4.

* country.region ссылается на region.identifier.
* country.identifier используется в таблице city как внешний ключ.

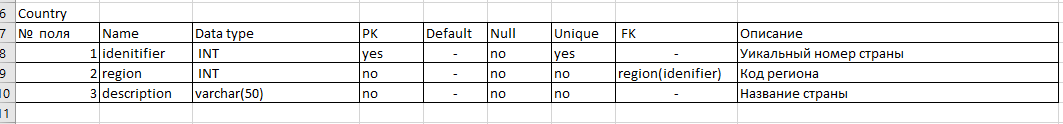


Рис 3 – таблица Country

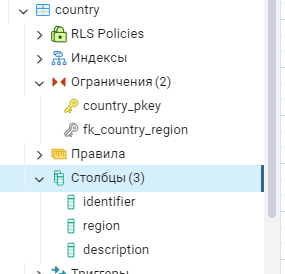


Рис 4 – структура таблицы Country

Таблица City

Таблица City хранит список городов, их местоположение (широта, долгота) и принадлежность к стране. Таблица City и также структура таблицы представлена на рисунке 5-6.

* city.country ссылается на country.identifier.
* city.identifier используется в таблице measurement как внешний ключ.

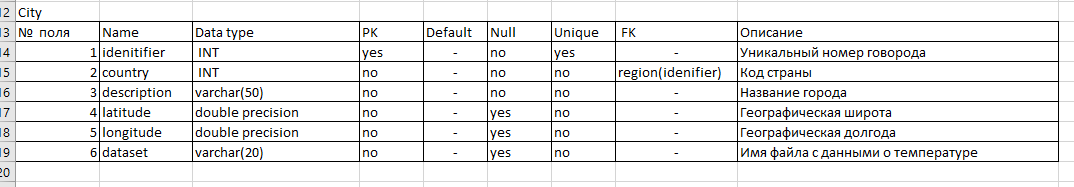


Рис 5 – таблица City

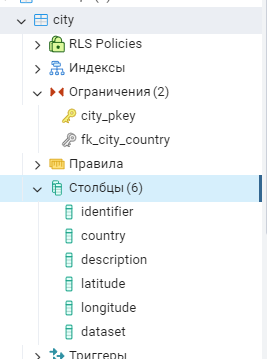


Рис 6 – структура таблицы City

Таблица Measurement

Таблица measurement содержит информацию о среднесуточных измерениях температуры в разных городах. Таблица Measurement и также структура таблицы представлена на рисунке 7-8.

* measurement.city ссылается на city.identifier.

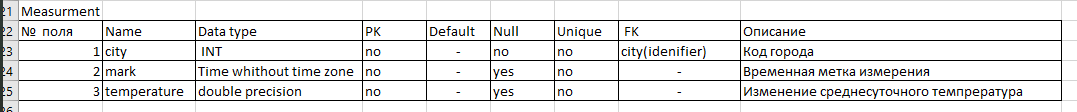


Рис 7 – таблица Measurement

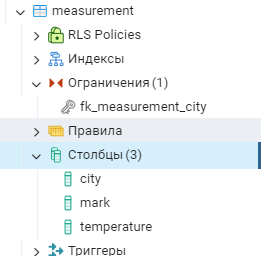


Рис 8 – структура таблицы City

Таблица Coastline

Таблица Coastline хранит информацию о координатах точек, из которых состоит береговая линия. Таблица Coastline и также структура таблицы представлена на рисунке 9-10.

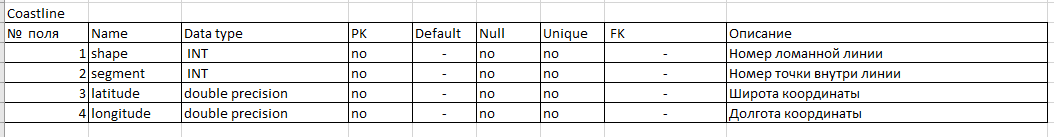


Рис 9 – таблица Coastline

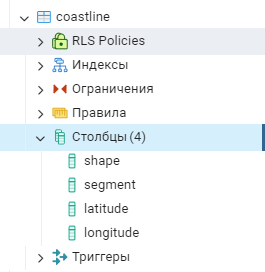


Рис 10 – структура таблицы Coastline

Здесь представлена модель ER диаграммы в приложении Dia, а также в PgAdmin4. Они представлены на рисунках 11-12

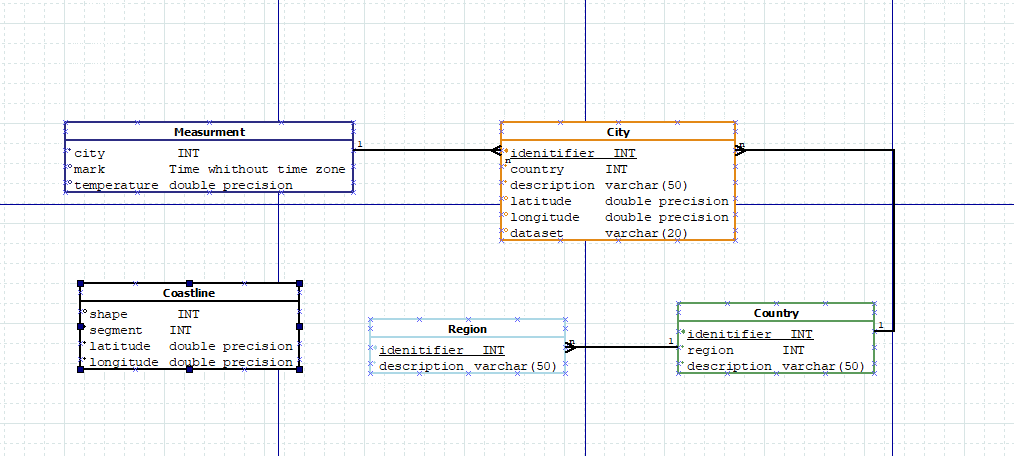


Рис 11 – ER диаграмма в Dia

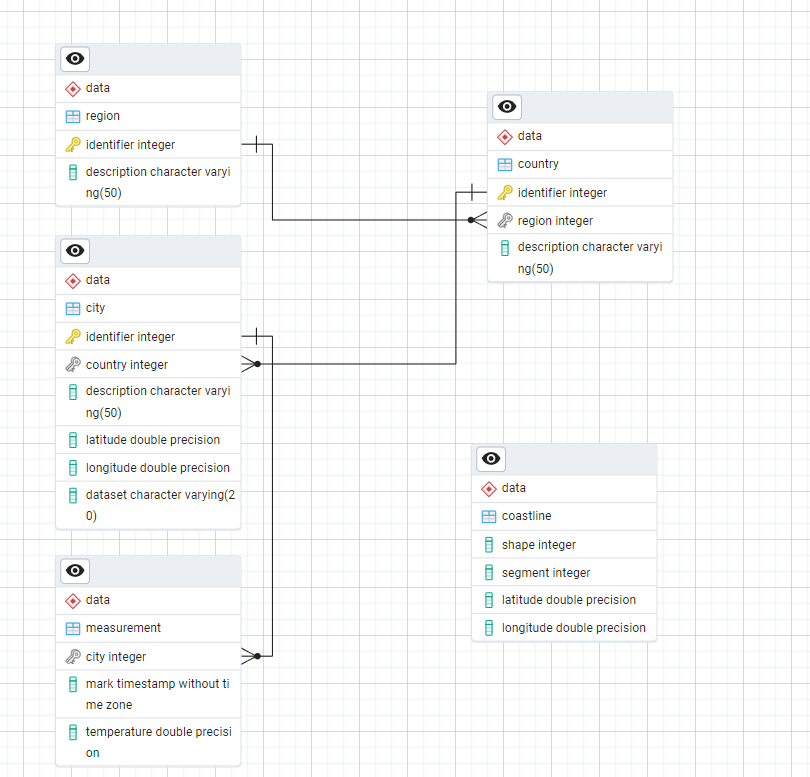


Рис 12 – ER диаграмма в PgAdnin4

Тут представлен код сценария (создание таблиц, ключей, индексы, значения по умолчанию и связи выделите отдельно командами ALTER TABLE).

create schema data;

DROP TABLE IF EXISTS data.region;

DROP TABLE IF EXISTS data.country;

DROP TABLE IF EXISTS data.city;

DROP TABLE IF EXISTS data.measurement;

DROP TABLE IF EXISTS data.coastline;

CREATE TABLE data.region (

identifier INTEGER PRIMARY KEY,

description VARCHAR(50) NOT NULL

);

CREATE TABLE data.country (

identifier INTEGER PRIMARY KEY,

region INTEGER NOT NULL,

description VARCHAR(50) NOT NULL

);

CREATE TABLE data.city (

identifier INTEGER PRIMARY KEY,

country INTEGER NOT NULL,

description VARCHAR(50) NOT NULL,

latitude DOUBLE PRECISION,

longitude DOUBLE PRECISION,

dataset VARCHAR(20)

);

CREATE TABLE data.measurement (

city INTEGER NOT NULL,

mark TIMESTAMP WITHOUT TIME ZONE NOT NULL,

temperature DOUBLE PRECISION NOT NULL

);

CREATE TABLE data.coastline (

shape INTEGER NOT NULL,

segment INTEGER NOT NULL,

latitude DOUBLE PRECISION NOT NULL,

longitude DOUBLE PRECISION NOT NULL

);

ALTER TABLE data.country

ADD CONSTRAINT fk\_country\_region

FOREIGN KEY (region) REFERENCES data.region(identifier);

ALTER TABLE data.city

ADD CONSTRAINT fk\_city\_country

FOREIGN KEY (country) REFERENCES data.country(identifier);

ALTER TABLE data.measurement

ADD CONSTRAINT fk\_measurement\_city

FOREIGN KEY (city) REFERENCES data.city(identifier);