МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**====================================================================**

**=============================================================================================**

Курсовой проект

По теме «Разработка защищенной программной информационной системы “Страны и флаги” с использование технологии PostgreSQL и языка Go»

Выполнил: ------------

Проверил:-------------

------------

Оглавление

[**Введение** 3](#_Toc93629423)

[**I)** **Постановка задачи** 4](#_Toc93629424)

[**II)**  **Описание предметной области** 5](#_Toc93629425)

[**III)**  **Проектирование БД** 6](#_Toc93629426)

[**IV)** **ER – диаграмма предметной области «Страны и флаги»** 9](#_Toc93629427)

[**V)** **Создание БД «Страны и флаги»** 10](#_Toc93629428)

[**VI)** **Заполнение БД «Страны и флаги»** 13](#_Toc93629429)

[**VII)** **Выполнение SQL – запросов на выбор данных** 17](#_Toc93629430)

[**VIII)** **Создание представлений** 23](#_Toc93629431)

[**IX)** **Создание хранимых процедур** 26](#_Toc93629432)

[**X)** **Разработка программного обеспечения** 31](#_Toc93629433)

[**XI)** **Внешний вид клиентского приложения** 33](#_Toc93629434)

[**Заключение** 35](#_Toc93629435)

[**Приложение** 36](#_Toc93629436)

# **Введение**

Информация о стране и её характеристика – это первейшая информация, которой пользуются люди по всему миру. Когда человек слышит название страны, у него начинают появляться вопросы, где она находится, а какой у неё флаг, а какой язык преобладает в данной стране, какая форма правления и религия. А потом уже человек вспоминает про экономику и прочую информацию.

Людям необходимо знать информацию о стране по многим причина: путешествие, проживание, предпринимательство… Несмотря на причину, знать характеристику государства, в которое ты хочешь приехать или с которым ты хочешь взаимодействовать, необходимо.

Из-за большого количества государств и объема информации о них, возникла потребность хранить эту информацию в структурированном виде. С возможностью добавлять, удалять, и модифицировать данные о стране. Хорошим решением является хранение данных в базе данных (БД). Она поможет структурировать и сохранить всю информацию о государствах. А разработка программного обеспечения для взаимодействия с базой данных, поможет автоматизировать и упростить эти процессы.

# **Постановка задачи**

1. Выбор предметной области
2. Краткое описание сущностей базы данных (БД) и их характеристик
3. Построение структурой диаграммы сущность – связь
4. Создание и заполнение БД
5. Создание представлений, хранимых процедур, запросов на выборку данных для БД «Страны и флаги»
6. Создание пользовательского клиентского приложения для работы с созданной БД

# **II) Описание предметной области**

БД «Страны и флаги» предназначено для хранения информации о странах, континентах нашей планеты, существующих организациях, в которых могут состоять государства, языке и форме правления, размере территории, вероисповедании государства. БД состоит из следующих таблиц:

1. Основные таблицы:
   1. Таблица «Cтраны» (Countries) – содержит основную информацию о странах: название страны (CountriesName), флаг (Flag), вероисповедание (ReligionID), языке (LanguagesID), форме правления (GovernmentFormID), характеристику размера территории (TerritorySizeID)
   2. Таблица «Религии» (Religions) – содержит основную информацию о религиях: название религии (Title)
   3. Таблица «Языки» (Languages) – содержит основную информацию о языках: название языка (Language)
   4. Таблица «Форма правления» (GovernmentForms) – содержит основную информацию о формах правления: форма правления (Form)
   5. Таблица «Типе территории» (TerritorySizes) – содержит основную информацию о типе территории: тип территории (Type)
   6. Таблица «Континенты» (Сontinents) – содержит основную информацию о религиях: название континента (Name)
   7. Таблица «Ассоциации» (Associations) – содержит основную информацию о религиях: название ассоциации (Title)
2. Промежуточные таблицы:
   1. Таблица «Континенты и страны» (СontinentsOfCountries) – устанавливает связь «многие ко многим» для таблиц «Континенты» (Сontinents) и «Cтраны» (Countries). Поля таблицы: идентификатор континента (СontinentsID), идентификатор страны (CountriesID)
   2. Таблица «Ассоциации и страны» (AssociationsOfCountries) – устанавливает связь «многие ко многим» для таблиц «Ассоциации» (Associations) и «Cтраны» (Countries). Поля таблицы: идентификатор ассоциации (AssociationsID), идентификатор страны (CountriesID)

# **III) Проектирование БД**

1. Характеристика таблицы «Страны» (Countries):

|  |  |  |
| --- | --- | --- |
| Поля | Тип данных | Описание |
| id | serial | Первичный автоинкрементирующийся ключ. Идентификатор альбома. Не ноль. |
| CountriesName | varchar(128) | Название страны |
| Flag | varchar(256) | Флаг страны |
| ReligionID | smallint | Вероисповедание страны |
| LanguagesID | smallint | Язык в стране |
| GovernmentFormID | smallint | Форма правления |
| TerritorySizeID | smallint | Тип территории |

1. Характеристика таблицы «Религии» (Religions)

|  |  |  |
| --- | --- | --- |
| Поля | Тип данных | Описание |
| id | serial | Первичный автоинкрементирующийся ключ. Идентификатор альбома. Не ноль. |
| Title | varchar(512) | Название религии |

1. Характеристика таблицы «Языки» (Languages)

|  |  |  |
| --- | --- | --- |
| Поля | Тип данных | Описание |
| id | serial | Первичный автоинкрементирующийся ключ. Идентификатор альбома. Не ноль. |
| Language | varchar(512) | Язык |

1. Характеристика таблицы «Форма правления» (GovernmentForms)

|  |  |  |
| --- | --- | --- |
| Поля | Тип данных | Описание |
| id | serial | Первичный автоинкрементирующийся ключ. Идентификатор альбома. Не ноль. |
| Form | varchar(512) | Форма правления |

1. Характеристика таблицы «Типе территории» (TerritorySizes)

|  |  |  |
| --- | --- | --- |
| Поля | Тип данных | Описание |
| id | serial | Первичный автоинкрементирующийся ключ. Идентификатор альбома. Не ноль. |
| Type | varchar(512) | Тип территории |

1. Характеристика таблицы «Континенты» (Сontinents)

|  |  |  |
| --- | --- | --- |
| Поля | Тип данных | Описание |
| id | serial | Первичный автоинкрементирующийся ключ. Идентификатор альбома. Не ноль. |
| Name | varchar(128) | Название континента |

1. Характеристика таблицы «Ассоциации» (Associations)

|  |  |  |
| --- | --- | --- |
| Поля | Тип данных | Описание |
| id | serial | Первичный автоинкрементирующийся ключ. Идентификатор альбома. Не ноль. |
| Title | varchar(1024) | Название ассоциации |

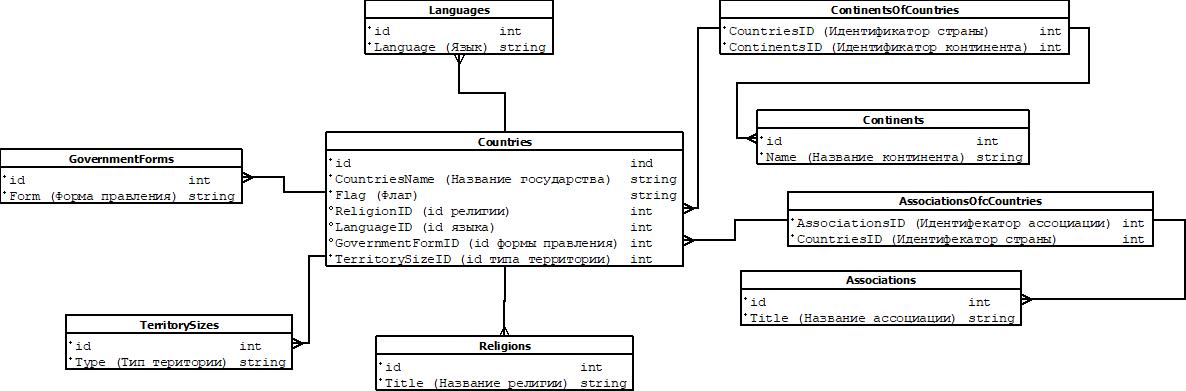
1. Характеристика таблицы «Континенты и страны» (СontinentsOfCountries)

|  |  |  |
| --- | --- | --- |
| Поля | Тип данных | Описание |
| id | serial | Первичный автоинкрементирующийся ключ. Идентификатор альбома. Не ноль. |
| CountriesID | int | Идентификатор страны |
| СontinentsID | int | Идентификатор континента |

1. Характеристика таблицы «Ассоциации и страны» (AssociationsOfCountries)

|  |  |  |
| --- | --- | --- |
| Поля | Тип данных | Описание |
| id | serial | Первичный автоинкрементирующийся ключ. Идентификатор альбома. Не ноль. |
| CountriesID | int | Идентификатор страны |
| AssociationsID | int | Идентификатор ассоциации |

# **IV) ER – диаграмма предметной области «Страны и флаги»**



# **V) Создание БД «Страны и флаги»**

Создание БД:

Изображение выглядит как текст

Автоматически созданное описание

Создадим таблицы в базе данных:

1. Создание таблицы «Cтраны» (Countries):

Изображение выглядит как текст

Автоматически созданное описание

1. Создание таблицы «Религии» (Religions)

Изображение выглядит как текст

Автоматически созданное описание

1. Создание таблицы «Языки» (Languages)

Изображение выглядит как текст

Автоматически созданное описание

1. Создание таблицы «Форма правления» (GovernmentForms)

Изображение выглядит как текст

Автоматически созданное описание

1. Создание таблицы «Типе территории» (TerritorySizes)

Изображение выглядит как текст

Автоматически созданное описание

1. Создание таблицы «Континенты» (Сontinents)

Изображение выглядит как текст

Автоматически созданное описание

1. Создание таблицы «Ассоциации» (Associations)

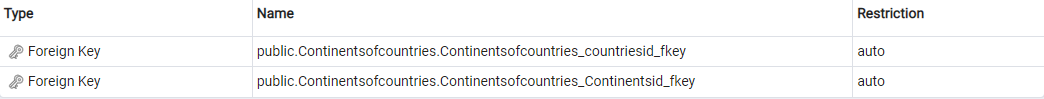
Изображение выглядит как текст

Автоматически созданное описание

1. Создание таблицы «Континенты и страны» (СontinentsOfCountries)

Изображение выглядит как текст

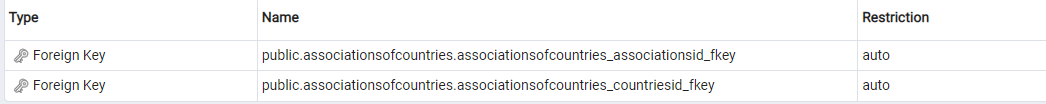
Автоматически созданное описание



1. Создание таблицы «Ассоциации и страны» (AssociationsOfCountries)

Изображение выглядит как текст

Автоматически созданное описание



Для установления связи «одно ко многим» для БД «Страны» использовались следующие команды:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

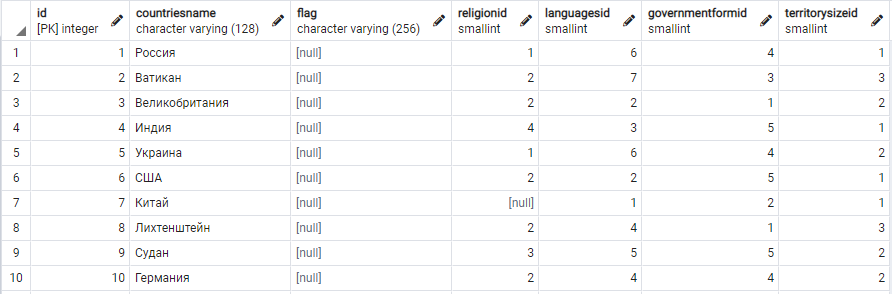
Автоматически созданное описание

# **VI) Заполнение БД «Страны и флаги»**

Заполним таблицы в базе данных:

1. Заполнение таблицы «Cтраны» (Countries):

INSERT INTO Countries (CountriesName, ReligionID, LanguagesID, GovernmentFormID, TerritorySizeID) VALUES ('Россия', 1, 6, 4, 1), ('Ватикан', 2, 7, 3, 3), ('Великобритания', 2, 2, 1, 2), ('Индия', 4, 3, 5, 1), ('Украина', 1, 6, 4, 2), ('США', 2, 2, 5, 1), ('Китай', NULL, 1, 2, 1), ('Лихтенштейн', 2, 4, 1, 3), ('Судан', 3, 5, 5, 2), ('Германия', 2, 4, 4, 2);



1. Заполнение таблицы «Религии» (Religions)

INSERT INTO Religions (title) VALUES ('Православие'), ('Католицизм'), ('Ислам'), ('Индуизм'), ('Буддизм');

Изображение выглядит как стол

Автоматически созданное описание

1. Заполнение таблицы «Языки» (Languages)

INSERT INTO Languages (Language) VALUES ('Китайский'), ('Английский'), ('Хинди'), ('Немецкий'), ('Арабский'), ('Русский'), ('Итальянский');

Изображение выглядит как стол

Автоматически созданное описание

1. Заполнение таблицы «Форма правления» (GovernmentForms)

INSERT INTO GovernmentForms (Form) VALUES ('Монархия'), ('Однопартийная система'), ('Теократия'), ('Полупрезидентская республика'), ('Президентская республика без премьер-министра');

Изображение выглядит как текст

Автоматически созданное описание

1. Заполнение таблицы «Типе территории» (TerritorySizes)

INSERT INTO TerritorySizes (type) VALUES ('Большие (страны-гиганты)'), ('«крупные» страны'), ('Микрогосударства');

Изображение выглядит как текст, стол

Автоматически созданное описание

1. Заполнение таблицы «Континенты» (Сontinents)

INSERT INTO Сontinents (Name) VALUES ('Африка'), ('Европа'), ('Азия'), ('Северная Америка'), ('Южная Америка'), ('Антарктида'), ('Австралия');

Изображение выглядит как стол

Автоматически созданное описание

1. Заполнение таблицы «Ассоциации» (Associations)

INSERT INTO Associations (Title) VALUES ('СНГ'), ('ЕС'), ('НАТО');

Изображение выглядит как стол

Автоматически созданное описание

1. Заполнение таблицы «Континенты и страны» (СontinentsOfCountries)

INSERT INTO СontinentsOfCountries (CountriesID, СontinentsID) VALUES (1, 2), (1, 3), (2, 2), (3, 2), (4, 3), (5, 2), (6, 4), (7, 3), (8, 2), (9, 1), (10, 2);

Изображение выглядит как текст, мебель, шкафчик, легкий

Автоматически созданное описание

1. Заполнение таблицы «Ассоциации и страны» (AssociationsOfCountries)

INSERT INTO AssociationsOfCountries (CountriesID, AssociationsID) VALUES (1, 1), (3, 3), (6, 3), (10, 2), (10, 3);

Изображение выглядит как стол

Автоматически созданное описание

# **VII) Выполнение SQL – запросов на выбор данных**

Выполнение нескольких запросов на выбор данных из таблицы:

1. Выбираем из таблицы все поля тех элементы, у которых параметр ReligionID равен 1 или 2

SELECT \* FROM Countries WHERE (ReligionID = 1) OR (ReligionID = 2)

Изображение выглядит как стол

Автоматически созданное описание

1. Выбираем из таблицы все поля тех элементов, у которых id равен 2 или 3

SELECT \* FROM Сontinents WHERE id IN (2, 3)

Изображение выглядит как стол

Автоматически созданное описание

1. Выбираем из таблицы все поля тех элементов, у которых CountriesName (Название страны) начинается на **В**

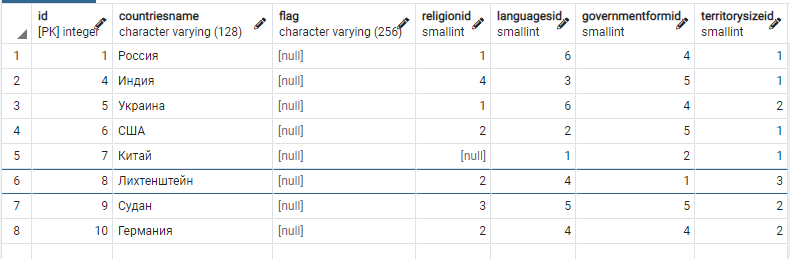
SELECT \* FROM Countries WHERE CountriesName LIKE 'В%'

Изображение выглядит как стол

Автоматически созданное описание

1. Выбираем из таблицы все поля тех элементов, у которых CountriesName (Название страны) не начинается на букву **В**

SELECT \* FROM Countries WHERE CountriesName NOT LIKE 'В%'



1. Выбираем из таблицы все поля тех элементов, у которых CountriesName (Название страны) заканчивается на **ия**

SELECT \* FROM Countries WHERE CountriesName LIKE '%\_ия'

Изображение выглядит как стол

Автоматически созданное описание

1. Выбираем из таблицы все поля тех элементов, у которых CountriesName (Название страны) имеет в своем названии две буквы с

SELECT \* FROM Countries WHERE CountriesName LIKE '%сс%'



1. Возвращает количество элементов в таблице Countries. При этом называя поле со значением CountCountries

SELECT count(\*) AS CountCountries FROM Countries

Изображение выглядит как текст

Автоматически созданное описание

1. Вычисляем количество представителей каждой из религий в таблице Countries . После чего получаем таблицу со столбцами: имя страны – количество представителей

SELECT Religions.title, count(\*) AS CountCountries FROM Countries, Religions WHERE Countries.ReligionID = Religions.id AND ReligionID IS NOT NULL GROUP BY 1Изображение выглядит как стол

Автоматически созданное описание

1. Вычисляем количество представителей каждой из религий в таблице Countries. После чего получаем таблицу со столбцами: имя страны – количество представителей. После чего, находим религию с максимальным количеством представителей

WITH some\_count AS (

SELECT Religions.title AS title, count(\*) AS CountCountries FROM Countries, Religions WHERE Countries.ReligionID = Religions.id AND ReligionID IS NOT NULL GROUP BY 1

)

SELECT \* FROM some\_count WHERE (CountCountries) IN (SELECT max(CountCountries) FROM some\_count)

Изображение выглядит как текст

Автоматически созданное описание

1. Какой тип правительства встречается чаще всего и какую долю от общего числа государств они составляют ?

SELECT governmentforms.form, t.governmentformid, round(100.0 \* cnt/ sum(cnt) OVER (), 2) AS percent

FROM governmentforms, (

SELECT governmentformid, count(\*) cnt FROM countries GROUP BY governmentformid

) t WHERE governmentforms.id = t.governmentformid ORDER BY percent DESC;

Изображение выглядит как текст

Автоматически созданное описание

1. Вывести полупрезидентские республики, являющееся странами гигантами и маленькие монархии.

SELECT countriesname, (SELECT governmentforms.form governmentforms FROM governmentforms WHERE countries.governmentformid = governmentforms.id ) as type\_governmentforms

FROM countries WHERE territorysizeid < 2 and governmentformid = 4

UNION SELECT countriesname, (SELECT governmentforms.form governmentforms FROM governmentforms WHERE countries.governmentformid = governmentforms.id ) as type\_governmentforms FROM countries WHERE territorysizeid > 2 and governmentformid = 1

Изображение выглядит как стол

Автоматически созданное описание

1. Написать запрос, который выводит таблицу с полями: название страны, название конфессии, язык.

SELECT Countries.countriesname, Religions.title Religions, Languages.language Languages FROM Countries

INNER JOIN Religions ON Countries.religionid = Religions.id

INNER JOIN Languages ON Countries.languagesid = Languages.id

Изображение выглядит как стол

Автоматически созданное описание

1. Какое количество государств расположено на каждом из континентов

SELECT Сontinents.name, t.count FROM Сontinents,

(SELECT Сontinentsid, count(\*) FROM Сontinentsofcountries GROUP BY Сontinentsid) t

WHERE t.Сontinentsid = Сontinents.id

Изображение выглядит как стол

Автоматически созданное описание

1. Вывести европейские страны которые состоят в НАТО

SELECT (SELECT countries.countriesname FROM countries WHERE countries.id = Сontinentsofcountries.countriesid) as countries

FROM Сontinentsofcountries, associationsofcountries WHERE Сontinentsofcountries.Сontinentsid = 2 and associationsofcountries.associationsid = 3

and Сontinentsofcountries.countriesid = associationsofcountries.countriesid

Изображение выглядит как текст

Автоматически созданное описание

1. Какой процент англо говорящих стран исповедуют католицизм

SELECT round(100.0 \* re.count/ en.count, 2) AS percent FROM (

SELECT count(\*) FROM countries WHERE countries.languagesid = 2

) en, (SELECT count(\*) FROM countries WHERE countries.religionid = 2 and countries.languagesid = 2) re

Изображение выглядит как текст

Автоматически созданное описание

1. Вывести таблицу с информацией о том, как кая религия самая многочисленная

WITH some\_count AS (

SELECT Religions.title AS title, count(\*) AS CountCountries FROM Countries, Religions WHERE Countries.ReligionID = Religions.id AND ReligionID IS NOT NULL GROUP BY 1

)

SELECT \* FROM some\_count WHERE (CountCountries) IN (SELECT max(CountCountries) FROM some\_count)

Изображение выглядит как текст

Автоматически созданное описание

# **VIII) Создание представлений**

1. Создать представление из всех католических стран

CREATE VIEW catholic AS SELECT \* FROM countries WHERE religionid = 2 GROUP BY countries.id;

SELECT \* FROM catholic

Изображение выглядит как стол

Автоматически созданное описание

1. Создать представление из количества стран представителей каждой из религий

CREATE VIEW some\_counts AS (

SELECT Religions.title AS title, count(\*) AS CountCountries FROM Countries, Religions

WHERE Countries.ReligionID = Religions.id AND ReligionID IS NOT NULL GROUP BY 1);

SELECT \* FROM some\_counts

Изображение выглядит как стол

Автоматически созданное описание

1. Создать представление из количества государств, расположено на каждом континенте

CREATE VIEW countCountries AS (

SELECT Сontinents.name, t.count FROM Сontinents,

(SELECT Сontinentsid, count(\*) FROM Сontinentsofcountries GROUP BY Сontinentsid) t

WHERE t.Сontinentsid = Сontinents.id);

SELECT \* FROM some\_counts

Изображение выглядит как стол

Автоматически созданное описание

1. Создать представление с информацией, какой тип правительства встречается и какую долю от всех государств имеет

CREATE VIEW govermentPercent AS (

SELECT governmentforms.form, t.governmentformid, round(100.0 \* cnt/ sum(cnt) OVER (), 2) AS percent

FROM governmentforms, (

SELECT governmentformid, count(\*) cnt FROM countries GROUP BY governmentformid

) t WHERE governmentforms.id = t.governmentformid ORDER BY percent DESC

);

SELECT \* FROM govermentPercent

Изображение выглядит как текст

Автоматически созданное описание

1. Вернуть представление с информацией о континенте и странах на нем

CREATE VIEW govermentInCountinents AS (

SELECT Сontinents.name, (SELECT STRING\_AGG(countries.countriesname, ', ') FROM countries, Сontinentsofcountries cc WHERE countries.id = cc.countriesid and cc.Сontinentsid = Сontinents.id ) FROM Сontinents

)

SELECT \* FROM govermentInCountinents

Изображение выглядит как текст

Автоматически созданное описание

# **IX) Создание хранимых процедур**

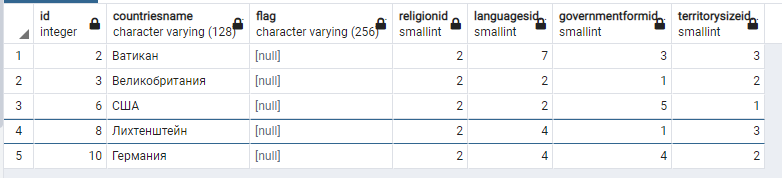
1. Хранимая процедура «Католические страны»
   1. Заголовок: catholic
   2. Назначение: возвращает список всех католических государств
   3. Входные параметры: нет
   4. Выходные параметры: SETOF countries
   5. Текст процедуры:

CREATE OR REPLACE FUNCTION catholic() RETURNS SETOF countries AS $$

SELECT \* FROM countries WHERE religionid = 2 GROUP BY countries.id;

$$ LANGUAGE sql;

* 1. Результат выполнения SELECT \* FROM catholic ();



1. Хранимая процедура «Монархические страны»
   1. Заголовок: govermate
   2. Назначение: возвращает список всех стран с монархией
   3. Входные параметры: нет
   4. Выходные параметры: SETOF countries
   5. Текст процедуры:

CREATE OR REPLACE FUNCTION govermate() RETURNS SETOF countries AS $$

SELECT \* FROM countries WHERE governmentformid = 1 GROUP BY countries.id;

$$ LANGUAGE sql;

2.6 Результат выполнения SELECT \* FROM govermate();

Изображение выглядит как текст

Автоматически созданное описание

1. Хранимая процедура «Язык, используемый в определенных странах»
   1. Заголовок: languages
   2. Назначение: возвращает список всех стран с определенным языком
   3. Входные параметры: int
   4. Выходные параметры: SETOF countries
   5. Текст процедуры:

CREATE OR REPLACE FUNCTION languages(IN id\_lang int) RETURNS SETOF countries AS $$

SELECT \* FROM countries WHERE languagesid = id\_lang GROUP BY countries.id;

$$ LANGUAGE sql;

* 1. Результат выполнения SELECT \* FROM languages(1);



1. Хранимая процедура «Тип территории»

4.1 Заголовок: territory

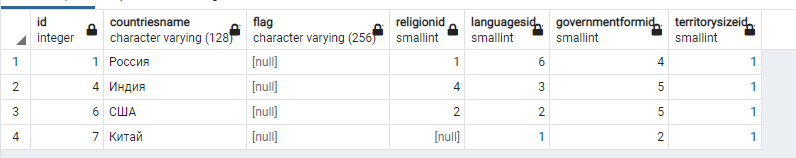
* 1. Назначение: возвращает список всех стран с определенным типом территории
  2. Входные параметры: int
  3. Выходные параметры: SETOF countries
  4. Текст процедуры:

CREATE OR REPLACE FUNCTION territory(IN id\_ter int) RETURNS SETOF countries AS $$

SELECT \* FROM countries WHERE territorysizeid = id\_ter GROUP BY countries.id;

$$ LANGUAGE sql;

4.6 Результат выполнения SELECT \* FROM territory(1);



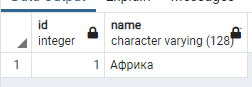
1. Хранимая процедура «Название континента по id»
   1. Заголовок: continent\_name
   2. Назначение: возвращает название континента в зависимости от заданного id
   3. Входные параметры: int
   4. Выходные параметры: SETOF Сontinents
   5. Текст процедуры:

CREATE OR REPLACE FUNCTION continent\_name(IN id\_ter int) RETURNS SETOF Сontinents AS $$

SELECT \* FROM Сontinents WHERE id = id\_ter;

$$ LANGUAGE sql;

5.6 Результат выполнения SELECT \* FROM continent\_name(1);



1. Хранимая процедура «Процент государств в зависимости от заданных параметров»
   1. Заголовок: per
   2. Назначение: возвращает процент государст с определенным языком и религией
   3. Входные параметры: int – id языка, int – id религии
   4. Выходные параметры: int
   5. Текст процедуры:

CREATE OR REPLACE FUNCTION per(IN id\_lang int, IN id\_rel int) RETURNS INT AS $$ SELECT round(100.0 \* re.count/ en.count, 2) AS percent FROM ( SELECT count(\*) FROM countries WHERE countries.languagesid = id\_lang ) en, (SELECT count(\*) FROM countries WHERE countries.religionid = id\_rel and countries.languagesid = id\_lang) re;

$$ LANGUAGE sql;

6.6 Результат выполнения SELECT \* FROM per(2, 2);

Изображение выглядит как стол

Автоматически созданное описание

1. Хранимая процедура «Набор верований и религий в государствах»

7.1 Заголовок: persen

7.2 Назначение: возвращает набор проверки различных наборов верований и государств

7.3 Входные параметры: нет

7.4 Выходные параметры: SETOF INT

7.5 Текст процедуры:

CREATE OR REPLACE FUNCTION per(IN id\_lang int, IN id\_rel int) RETURNS INT AS $$

SELECT round(100.0 \* re.count/ en.count, 2) AS percent FROM (

SELECT count(\*) FROM countries WHERE countries.languagesid = id\_lang

) en, (SELECT count(\*) FROM countries WHERE countries.religionid = id\_rel and countries.languagesid = id\_lang) re;

$$ LANGUAGE sql;

CREATE OR REPLACE FUNCTION persen() RETURNS SETOF INT AS $$

BEGIN

RETURN NEXT per(1, 1);

RETURN NEXT per(2, 2);

RETURN NEXT per(1, 2);

END

$$ LANGUAGE plpgsql;

7.6 Результат выполнения SELECT \* FROM persen();

Изображение выглядит как стол

Автоматически созданное описание

1. Хранимая процедура «Континенты с наличием определённого языка»

8.1 Заголовок: get\_cont

8.2 Назначение: в зависимости от языка, выводит континенты, на которых он используется

8.3 Входные параметры: text

8.4 Выходные параметры: TABLE (name text)

8.5 Текст процедуры:

CREATE OR REPLACE FUNCTION get\_cont(IN lang text) RETURNS TABLE (name text)

AS $$

SELECT Сontinents.name FROM Сontinents, Сontinentsofcountries cc,

(SELECT countries.id FROM countries WHERE countries.languagesid = (SELECT languages.id FROM languages WHERE languages.language = lang)) as langid

WHERE cc.Сontinentsid = Сontinents.id and cc.countriesid = langid.id

$$ LANGUAGE sql;

8.6 Результат выполнения SELECT \* FROM get\_cont('Английский');

Изображение выглядит как текст

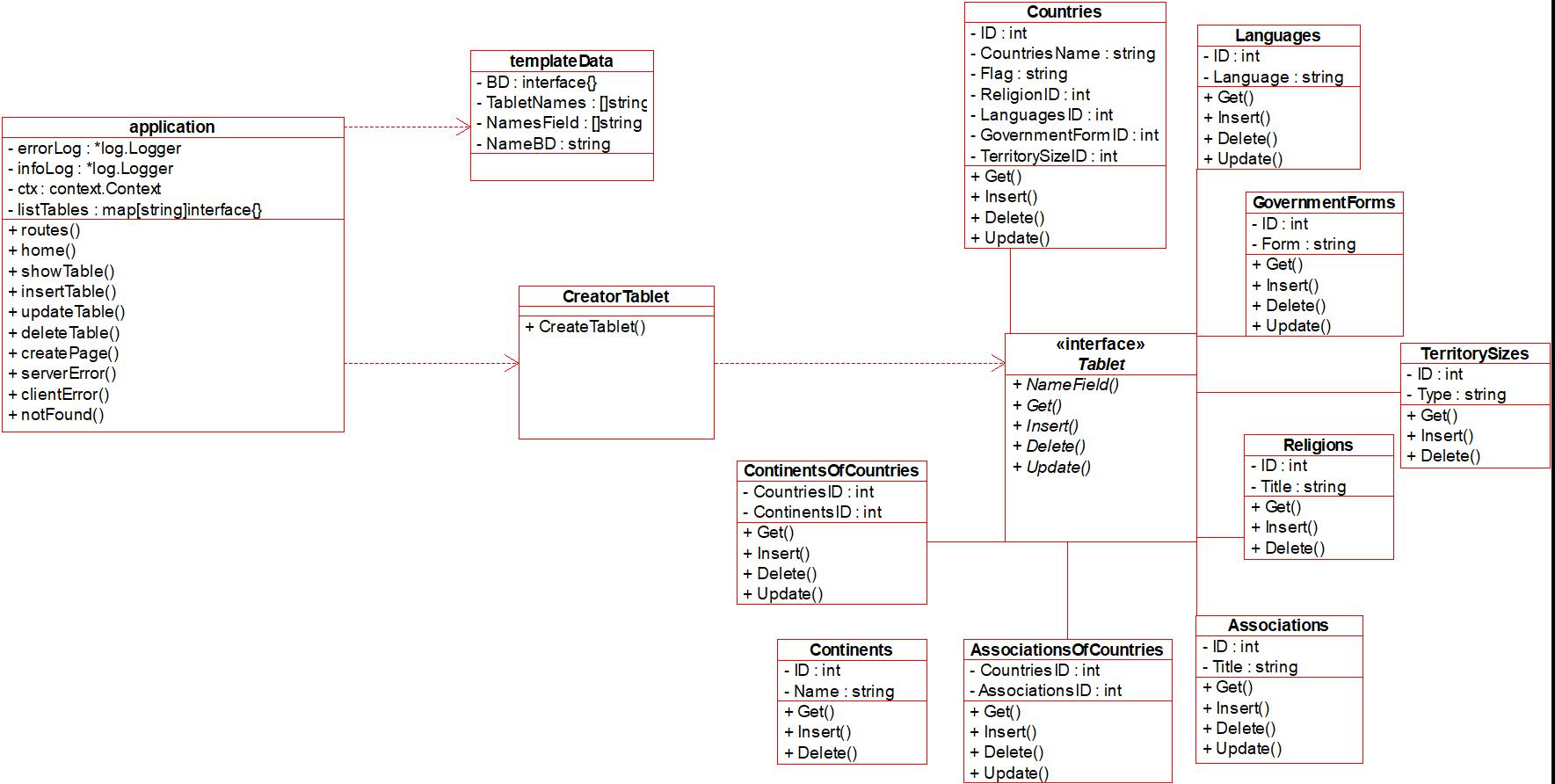
Автоматически созданное описание

# **X) Разработка программного обеспечения**

Для разработки ПО я использовал язык Go для backend и JavaScript для frontend части.

По причине отсутствия классов в языке Go, я решил выделить основные объекты и их методы, используемые в программе.

Схема взаимоотношения объектов



1. Описание моделей

Для начала были созданы модели следующих таблиц: Countries, Languages, GovernmentForms, TerritorySizes, Religions, Associations, AssociationsOfCountries, Сontinents, СontinentsOfCountries. Поля для этих моделей были выбраны в соответствии с их типом в базе данных. Также, для каждой модели были созданы 4 метода (функции):

* 1. Get() – Используется для возврата из БД всех элементов выбранной таблицы
  2. Insert() – Используется для записи в БД нового элемента
  3. Delete() – Используется для удаления из БД определенного элемента
  4. Update() – Используется для обновления определенного поля выбранного элемента, хранящегося в БД.

1. Описание объекта CreatorTablet и интерфейса Tablet

Для создания моделей БД используется объект CreatorTablet, который возвращает общий для всех моделей интерфейс Tablet

1. Описание объекта templateData

Описание полей с состояние объекта:

templateData – это структура, со следующими полями:

DB : interface{} – используется для хранения модели БД

tabletName: string[] – имена всех моделей БД

NamesField: string – названия полей выбранной модели

NameDB: string – Имя выбранной БД

Данный объект, используется для хранения выбранной, на данный момент, модели БД и для вывода её полей пользователю.

1. Описание объекта application

Описание полей с состояние объекта:

errorLog; \*log.Logger – хранит указатель на логгер, который сообщает об ошибке, в случае её возникновения

infoLog; \*log.Logger – хранит указатель на логгер, который записывает текущие действия пользователя

ctx: context:Context – хранит драйвер, используемый для взаимодействием БД postgreSQL

listTables: map[string]interface{} – хранит дескрипторы для всех БД

Описание методов объекта:

routes() – инициализирует маршруты и привязывает к ним методы объекта application

home() – выводит главную страницу сайта

showTable() – выводит определенную таблицу на сайте (Для GET запросов)

insertTable() – ввод элемента в определенную таблицу (Для POST запросов)

updateTable() - обновление элемента определенной таблицы (Для POST запросов)

delete Table() - удаление элемента из определенной таблицы (Для POST запросов)

createPage() – используется для создания страниц

serverError() – сообщает об ошибке на сервере

clientError() – сообщает об ошибке у клиента

notFound() – сообщает, о том что страница не найдена

Логика, связанная с настройкой и запуском сервера содержится в функции main.

# **XI) Внешний вид клиентского приложения**

Описание клиентского приложения:

1. Общая структура веб – приложения

Общая структура


Автоматически созданное описание

1. Боковое меню для переключения между таблицами

Изображение выглядит как текст

Автоматически созданное описание

1. Поле с контентом. Используется для вывода содержимого из БД

Изображение выглядит как стол

Автоматически созданное описание

1. Поле для редактирования БД.

Изображение выглядит как текст

Автоматически созданное описание

# **Заключение**

В результате выполнения курсового проекта были получены навык проектирования БД, построения ER – диаграмм, создания, удаления, модификации таблиц, записей в таблицах, представлений, создание хранимых процедур и выполнение различных видов запроса.

Для решения поставленных задач был изучен основной синтаксис язык PostgreSQL v14, были получены навыки использования приложения pgAdmin 4 v6 – среды для настройки баз данных, использующих PostgreSQL. Также было разработано веб-приложение, для взаимодействия с БД, написанное на Go и использующее для взаимодействия с БД драйвер pgx. В веб-приложение для работы с БД были включены функции добавления, изменения, удаления данных из таблиц.

Сложной для меня задачей было использование нового для меня драйвера для взаимодействия с БД. Так как я до этого использовал драйвер pq, который на данный момент устарел и перестал обновляться.

На протяжении курса полезным оказалось проектирование БД и интересно было изучить выполнение различных запросов для взаимодействия с БД.

# **Приложение**

Репозиторий с веб-приложением: <https://github.com/XFroggyX/InteractionGOandPSQL>

Используемая литература

<https://github.com/jackc/pgx> - документация драйвера для взаимодействия с БД