МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ

ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



Основная профессиональная образовательная программа

Направление подготовки 09.03.01 Информатика и вычислительная техника

Направленность (профиль) «Технологии разработки программного обеспечения»

форма обучения – очная

**ОТЧЁТ**

по реализации проекта для дисциплины «Базы данных»

по направлению “09.03.01 – Информатика и вычислительная техника”

(профиль: “Технологии разработки программного обеспечения ”)

Преподаватель: к.ф-м.н., доцент кафедры ИТиЭО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Жуков Н. Н.)

Преподаватель: ассистент кафедры ИТиЭО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Иванова Е. А.)

Студенты 2 курса:

Литовченко Д. В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нюхалов Д. Г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Царулкова А. В. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2020

**Оглавление**

[Ответственные](#_gjdgxs) **2**

[Предметная область](#_30j0zll) **3**

[Ход выполнения нормализации](#_1fob9te) **4**

[Объяснение выбранной СУБД](#_3znysh7) **6**

[ER – диаграмма](#_2et92p0) **7**

[**Исходный текст запросов**](#_1g5pny10fck0) **8**

[По созданию таблиц (с проверкой)](#_3dy6vkm) 8

# 

# Ответственные

Нюхалов Д. Г. – нормализовал отношения, связанные с резюме, производил реализацию на Sqlite

Царулкова А. В. – нормализовала отношения, связанные с вакансиями и пользователем, отвечала за теоретическую часть

Литовченко Д. В. – нормализовал отношения, связанные с компаниями, создавал ER-диаграмму.

# Предметная область

Организация-заказчик: городская студенческая биржа труда и обучения Санкт-Петербургского государственного университета технологии и дизайна (Региональный центр содействия трудоустройству и адаптации к рынку труда выпускников), адрес сайта и работающей версии системы: [www.embit.ru](http://www.embit.ru/)

Городская студенческая биржа труда и обучения (ГСБТО) Санкт-Петербургского государственного университета технологии и дизайна, является также Региональным центром содействия трудоустройству и адаптации к рынку труда выпускников.

Основными направлениями деятельности ГСБТО являются:

* Оказание содействия в трудоустройстве студентам и выпускникам учебных заведений города на частичную и полную, временную и постоянную занятость.
* Проведение социологических и маркетинговых исследований рынка труда.
* Осуществляется психологическая поддержка соискателей с целью повышения конкурентоспособности и адаптации к рынку труда.

Для успешной деятельности по этим направлениям разработана информационная система (далее «Система»), которая в настоящий момент устарела.

Перед нами стоит задача: спроектировать и разработать базу данных с учетом современных тенденций и требований для этой системы.

Основным назначением Системы является содействие в трудоустройстве студентов, выпускников ВУЗов.

База данных должна позволять решать следующие задачи:

* предоставлять возможность поиска вакансий и резюме соискателей в базе данных системы;
* предоставлять возможность добавлять новые вакансии и резюме в базу данных системы, а также изменять уже существующие данные;
* позволять открывать новые филиалы (пункты, где студентам будут оказывать содействие в трудоустройстве) в других городах (с помощью сотрудников этих филиалов и самостоятельно) с возможностью добавления компаний и вакансий в эти филиалы;
* позволять формировать реестр компаний, трудоустраивающих студентов и выпускников, предоставляющих места для прохождения практик.

# Ход выполнения нормализации

Пользователи:

Отношение хранит все необходимые данные о пользователе в атрибутах: Логин, Пароль, а для того чтобы определить, к какой группе относится пользователь, добавлен атрибут ID группы, который связывает эту таблицу с сущностью Тип пользователя. Введен искусственный первичный ключ ID пользователя. Все элементы обязательны к заполнению. Содержимое атрибута Логин уникально.

Тип пользователя

Содержит все названия группы пользователей на сайте в атрибуте Название группы. Внешний и первичный ключ: ID группы. Все элементы обязательны к заполнению. Название группы уникально.

Информация о вакансии:

Каждая вакансия имеет свое наименование, связь с компанией (ID компании), которая ее разместила, а также основные условия: Тип занятости, Зарплата, Опыт работы, а также ID раздела, к которому она относится. Первичными ключом отношения является ID Вакансии. Все элементы обязательны к заполнению.

Вакансия требования:

Для каждой вакансии существуют свои требования, которые отражены в данном отношении. Эти требования отражены в таких атрибутах, как Личные Качества, Прописка, Гражданство, Форма образования. Прописка и гражданство имеют тип BOOLEAN. Первичным и внешним ключом был использован атрибут ID Вакансии, связывающий это отношение с отношением Информация о вакансии. Все элементы обязательны к заполнению, кроме Личных качеств.

Вакансия языки:

Для работодателей часто является важным знание определенных языков. Для этого создано данное отношение с атрибутами Язык, Уровень знания. ID Вакансии является внешним ключом, связывающим с отношением Информация о вакансии Здесь использован составной первичный ключ: {ID Вакансии, Язык}. Все элементы обязательны к заполнению.

Существующие навыки:

Здесь находится список возможных необходимых навыков работника. Имеет атрибуты ID Наыка и Навыки. ID Навыка - внешний и первичный ключ. Связана с отношением Вакансия навыки. Все элементы обязательны к заполнению. Навыки уникальны.

Вакансия навыки:

Отношение, которое хранит все нужные навыки для конкретной вакансии. Состоит из двух атрибутов, которые и входят в состав первичного составного ключа: {ID Вакансии, ID Навыка}. Все элементы обязательны к заполнению.

Разделы:

Список существующих разделов вакансии. Включает атрибуты ID Раздела и Название раздел. ID Раздела - первичный ключ. Все элементы обязательны к заполнению, Раздел уникален.

Информация о соискателе:

Содержит основные данные о соискателе: Фамилия, Имя, Отчество (не обязательное для заполнения), Почта Телефон, Дата рождения, Личные Качества и атрибут-внешний ключ для связи с сущностью Пользователь - ID Пользователя. Первичный ключ ID Соискателя.

Соискатель пожелания:

Содержит атрибуты для упрощения поиска резюме: ID Раздела, Регион, Город, Гражданство, Прописка. Гражданство и Прописка также имеют тип BOOLEAN. Все атрибуты обязательны к заполнению. Первичный ключ ID Соискателя.

Соискатель образование:

Хранит детали образования соискателя: Специальность, Тип образования, Специальность, ВУЗ (СУЗ), Начало обучения, Конец обучения. Первичный ключ ID Соискателя. НачалоОбуч, КонецОбуч не обязательны для заполнения.

Соискатель навыки:

Хранит все имеющиеся навыки соискателя. Имеет составной первичный ключ {ID Соискателя, ID Навыка}.

Соискатель языки:

Содержит знание языков конкретного соискателя. Здесь использован составной первичный ключ: {ID Соискателя, Язык}. Все элементы обязательны к заполнению.

Информация о компании:

Содержит самую основную информацию о компании: Название компании, её Описание в свободной форме. Первичный ключ: ID Компании. Название компании обязательно к заполнению. ID Пользователя - внешний ключ, который связывает с отношением Пользователь.

Компания адрес:

Содержит адрес компании. Состоит из атрибутов: Страна, Регион, Город, Улица, Дом. Первичный и внешний ключ: ID Компании. Все элементы обязательны к заполнению.

Компания почты:

Содержит все почты компании. Состоит из атрибутов: ID Компании, Почта. Первичный и внешний ключ: ID Компании. Все элементы обязательны к заполнению.

Компания сайты:

Содержит все сайты компании. Состоит из атрибутов: ID Компании, Сайты. Первичный и внешний ключ: ID Компании. Все элементы обязательны к заполнению.

Компания соцсети:

Содержит все соцсети компании. Состоит из атрибутов: ID Компании, Соцсеть. Первичный и внешний ключ: ID Компании. Все элементы обязательны к заполнению.

Компания телефоны:

Содержит все телефоны компании. Состоит из атрибутов: ID Компании, Телефоны. Первичный и внешний ключ: ID Компании. Все элементы обязательны к заполнению.

# Объяснение выбранной СУБД

Была выбрана СУБД Sqlite из-за простоты ее реализации, установки и размещения на сервере, а также по причине наличия огромного количества решений для конфигурации базы данных sqlite, таких как, например, Sqlite Studio или библиотека sqlite3 в Python.

# ER – диаграмма

# 

Текст запросов

## По созданию таблиц (с проверкой)

[**Запросы на Github Gist**](https://gist.github.com/MeiJohnson/1472006d3e9f714d99e7c92d4d8fdd4c)

**Примеры:**

CREATE TABLE [Компания адрес] (

   [ID компании] INTEGER PRIMARY KEY

                         NOT NULL

                         REFERENCES [Информация о компании] ([ID компании]) ON DELETE CASCADE

                                                                            ON UPDATE CASCADE,

   Страна        CHAR    NOT NULL,

   Город         CHAR    NOT NULL,

   Улица         CHAR    NOT NULL,

   Дом           CHAR    NOT NULL,

   Квартира      CHAR    NOT NULL

);

CREATE TABLE [Вакансия требования] (

   [ID Вакансии]       INTEGER PRIMARY KEY

                               NOT NULL

                               REFERENCES [Информация о вакансии] ([ID вакансии]) ON DELETE CASCADE

                                                                                  ON UPDATE CASCADE,

   [Личные качества]   CHAR,

   Прописка            BOOLEAN NOT NULL,

   Гражданство         BOOLEAN NOT NULL,

   [Форма образования] CHAR    NOT NULL

);

**Запрос для поиска по всем вакансиям**

CREATE TABLE Search\_result (

    ID\_вакансии,

    Компания,

    Раздел,

    Зарплата,

    [Опыт работы],

    Тип\_занятости,

    Гражданство,

    Прописка,

    [Личные качества],

    Форма\_образования

);

INSERT INTO Search\_result

SELECT Vacancies.ID\_вакансии,

       (

           SELECT Companies.Название\_компании

             FROM Companies

            WHERE Companies.ID\_компании = Vacancies.ID\_компании

       )

       AS Компания,

       (

           SELECT Fields.[Название раздела]

             FROM Fields

            WHERE Fields.ID\_раздела = Vacancies.ID\_раздела

       )

       AS Раздел,

       Vacancies.Зарплата,

       Vacancies.Опыт\_работы,

       Vacancies.Тип\_занятости,

       Vacancy\_requirements.Гражданство,

       Vacancy\_requirements.Прописка,

       Vacancy\_requirements.Личные\_качества,

       Vacancy\_requirements.Форма\_образования

  FROM Vacancies

       INNER JOIN

       Vacancy\_requirements ON Vacancies.ID\_вакансии = Vacancy\_requirements.ID\_вакансии;

DELETE FROM Search\_result WHERE ID\_вакансии = 4;

**Запрос для выборки вакансии**

CREATE TABLE Selection (

    [ID\_пользователя] INTEGER NOT NULL

                              REFERENCES Users ([ID\_Пользователя]) ON DELETE CASCADE

                                                        ON UPDATE CASCADE,

    [ID\_вакансии]     INTEGER NOT NULL

                              REFERENCES Vacancies ([ID\_вакансии]) ON DELETE CASCADE

                                                        ON UPDATE CASCADE,

    Status            CHAR    NOT NULL,

    PRIMARY KEY (

        [ID\_пользователя],

        [ID\_вакансии]

    )

);

(SELECT [ID\_вакансии]

    FROM Selection

    WHERE

        [ID\_пользователя] = 1 AND Status = 'True')

AS Selection\_vacancy

INNER JOIN Vacancies ON

    Vacancies.ID\_вакансии = Selection\_vacancy.ID\_вакансии;

# ER – диаграмма

# Исходный текст запросов

## По созданию таблиц (с проверкой)

[**Запросы на Github Gist**](https://gist.github.com/MeiJohnson/1472006d3e9f714d99e7c92d4d8fdd4c)

**Примеры:**

CREATE TABLE [Компания адрес] (

[ID компании] INTEGER PRIMARY KEY

NOT NULL

REFERENCES [Информация о компании] ([ID компании]) ON DELETE CASCADE

ON UPDATE CASCADE,

Страна CHAR NOT NULL,

Город CHAR NOT NULL,

Улица CHAR NOT NULL,

Дом CHAR NOT NULL,

Квартира CHAR NOT NULL

);

CREATE TABLE [Вакансия требования] (

[ID Вакансии] INTEGER PRIMARY KEY

NOT NULL

REFERENCES [Информация о вакансии] ([ID вакансии]) ON DELETE CASCADE

ON UPDATE CASCADE,

[Личные качества] CHAR,

Прописка BOOLEAN NOT NULL,

Гражданство BOOLEAN NOT NULL,

[Форма образования] CHAR NOT NULL

);

**Запрос для поиска вакансий и исключения из него просмотренных вакансий**

CREATE TABLE Search\_result (

[ID вакансии],

Компания,

Раздел,

Зарплата,

[Опыт работы],

Тип\_занятости,

Гражданство,

Прописка,

[Личные качества],

Форма\_образования

);

INSERT INTO Search\_result

SELECT [Информация о вакансии].[ID вакансии],

(

SELECT [Информация о компании].Название компании

FROM [Информация о компании]

WHERE [Информация о компании].[ID компании] = [Информация о вакансии].[ID компании]

)

AS Компания,

(

SELECT Разделы.[Название раздела]

FROM Разделы

WHERE Разделы.[ID раздела] = [Информация о вакансии].[ID раздела]

)

AS Раздел,

[Информация о вакансии].Зарплата,

[Информация о вакансии].[Опыт работы],

[Информация о вакансии].[Тип занятости],

[Требования вакансии].Гражданство,

[Требования вакансии].Прописка,

[Требования вакансии].[Личные качества],

[Требования вакансии].[Форма образования]

FROM [Информация о вакансии]

INNER JOIN

[Требования вакансии] ON [Информация о вакансии].[ID вакансии] = [Требования вакансии].[ID вакансии];

DELETE FROM Search\_result WHERE [ID вакансии] = 4;