**Руководство разработчика**

Офисного приложения «Data manager»

для управления базой данных сотрудников компании

**Разработчики**:

Прошак Никита

Родионов Дмитрий

Хуснутдинов Фазиль

Москва 2021

Оглавление

[Описание базы данных 3](#_Toc75471401)

[Структура приложения 4](#_Toc75471402)

[Технические требования 11](#_Toc75471403)

# Описание базы данных

База данных состоит из трех справочников: workers, children, otdeli.

1. workers. Состоит из следующих полей:

* ФИО
* Дата рождения
* ФИО Ребенка
* Прививка от COVID-19
* Номер отдела
* Должность
* З/П в месяц

Содержит в себе основную информацию о сотрудниках, входящих в штат компании.

1. children. Состоит из следующих полей:

* ФИО Ребенка
* Дата рождения ребенка
* Номер садика

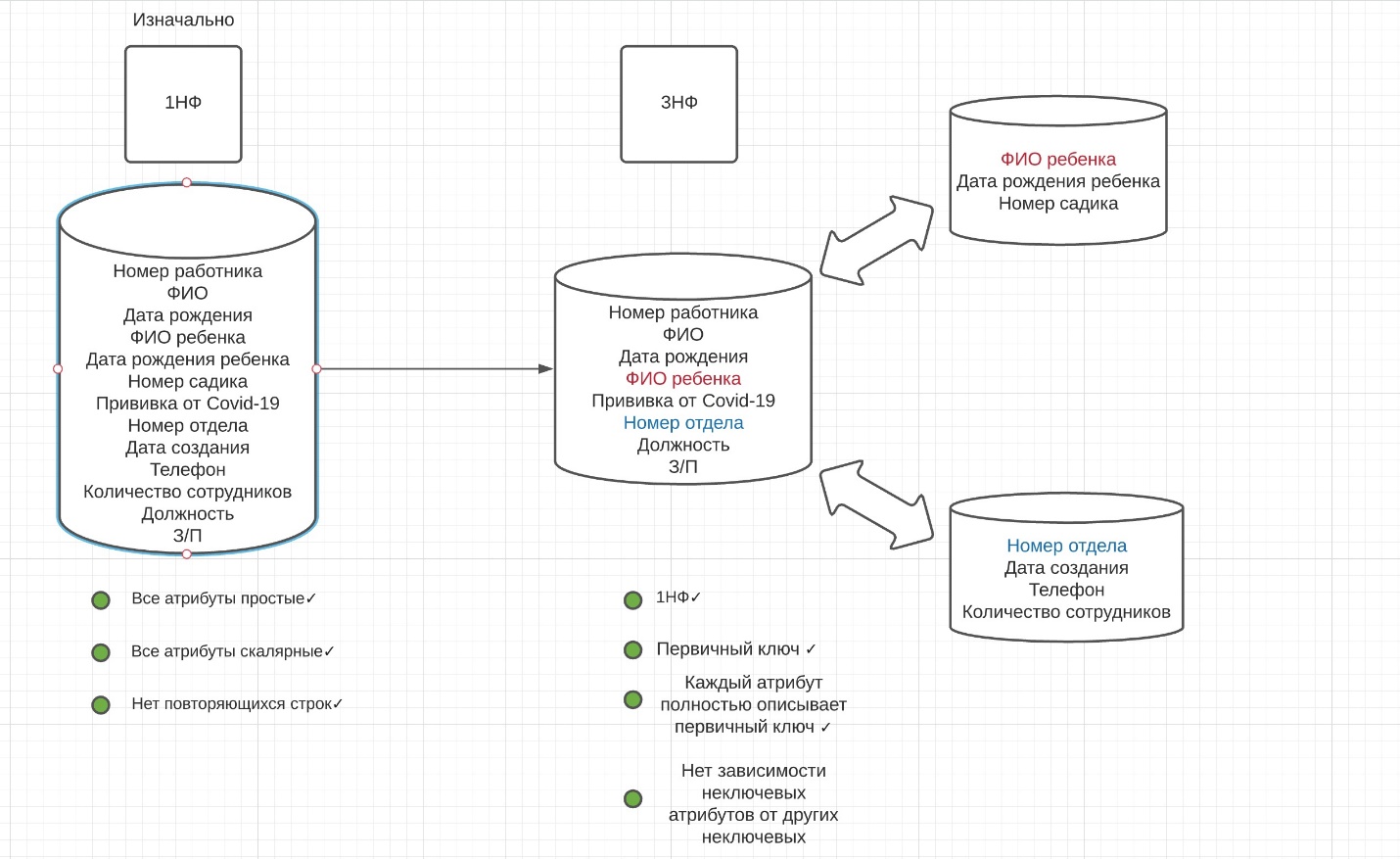
Содержит в себе информацию о детях сотрудников, связан с ними по ключу в виде номера сотрудника.

1. otdeli. Состоит из следующих полей:

* Номер отдела
* Дата создания
* Телефон
* Количество сотрудников

Содержит в себе подробную информацию по отделам. Связан с первым справочником при помощи ключа в виде номера отдела.

Связи между справочниками БД наглядно изображены на схеме 1:



*Схема 1. Иллюстрация к БД*

# Структура приложения

Приложение делится на каталоги:

* data – содержит в себе справочники в формате «.pic» и текстовые файлы для вызова их в приложении
* scripts – хранилище скриптов: там находятся составные модули программы и все файлы «.py»
* notes – в данной директории хранится документация к проекту.

Приложение в хранилище scripts делится на модули, указанные в следующем списке:

* main.py
* windows\_creating.py
* plotting.py
* db\_interaction.py
* checking.py

Остановимся на каждом более подробно:

1. **main.py** – главный скрипт приложения. Он содержит в себе корневой объект графической оболочки приложения, описание его параметров и подключает из остальных модулей функции, необходимые для корректной работы клавиш и алгоритма программы. Отдельных функций не содержит – только объявление объектов библиотек Tkinter, Pandas, os, datetime и функций модуля checking.py
2. **windows\_creating.py –** модуль, содержащий в себе функции создания и изменения окон главного объекта TCL/TK. Включает в себя импорт модулей библиотеки tkinter, а также функции-обработчики данных из модуля db\_interacton.py, функции для проверки типа из модуля checking.py и функцию для создания графика из модуля plotting.py . Модуль содержит в себе следующие функции:

\*(Во всех последующих функциях параметр content является объектом ttk.Frame)

* **add\_workers\_row(content):**

Функция ответственна за создание окна для добавления записи в исходную базу данных workers. Входной параметр – content(\*). Ничего не возвращает.

* **add\_children\_row(content):**

Функция ответственна за создание окна для добавления записи в исходную базу данных children. Входной параметр – content(\*). Ничего не возвращает.

* **add\_otdeli\_row(content):**

Функция ответственна за создание окна для добавления записи в исходную базу данных otdeli. Входной параметр – content(\*). Ничего не возвращает.

* **one\_attr\_search(content):**

Создание окна для поиска по одному атрибуту. Входной параметр – content(\*). Ничего не возвращает.

* **many\_attr\_search(content):**

Создание окна для поиска по нескольким атрибутам. Входной параметр – content(\*). Ничего не возвращает.

* **one\_attr\_search\_filter(content):**

Создание окна для поиска по атрибутам и одному условию. Входной параметр – content(\*). Ничего не возвращает.

* **many\_attr\_search\_filter(content):**

Создание окна для поиска по нескольким атрибутам и условиям. Входной параметр – content(\*). Ничего не возвращает.

* **back\_graph\_rep(content):**

Функция для составления графических отчетов. Входной параметр – content(\*). Ничего не возвращает.

* **on\_click():**

Подфункция для считывания значений из combobox. Входных параметров нет, ничего не возвращает. Является подфункцией в функции back\_graph\_rep(content).

* **show\_dict\_workers(content):**

Функция для работы со словарями справочника workers. Входной параметр – content(\*). Ничего не возвращает.

* **show\_dict\_children(content):**

Функция для работы со словарями справочника children. Входной параметр – content(\*). Ничего не возвращает.

* **show\_dict\_deps(content):**

Функция для работы со словарями справочника otdeli. Входной параметр – content(\*). Ничего не возвращает.

* **create\_otchet\_window(text, df):**

Функция создания окна текстового отчета. Входные параметры: text – изменяемый виджет, df –объект DataFrame. Ничего не возвращает.

* **change\_row(content, db\_name):**

Функция создания окна для поиска записи по индексу. Входные параметры – content(\*), db\_name – объект, из которого запрашиваются данные. Ничего не возвращает.

* **show\_entry(content, index, db\_name):**

Создание окна для изменения или удаления записи. Входные параметры – content(\*), index – индекс строки, db\_name – объект, из которого запрашиваются данные. Ничего не возвращает.

* **show\_workers\_entry(content, index, db):**

Создание окна для изменения или удаления записи в справочнике workers. Входные параметры – content(\*), index – индекс строки, db – объект DataFrame. Ничего не возвращает.

* **show\_children\_entry(content, index, db):**

Создание окна для изменения или удаления записи в справочнике children. Входные параметры – content(\*), index – индекс строки, db – объект DataFrame. Ничего не возвращает.

* **show\_otdeli\_entry(content, index, db):**

Создание окна для изменения или удаления записи в справочнике otdeli. Входные параметры – content(\*), index – индекс строки, db – объект DataFrame. Ничего не возвращает.

3. **plotting.py** – модуль, отвечающий за построение графиков. Включает в себя объекты модуля matplotlib, а также функции модуля db\_interaction.py. Содержит в себе следующие функции:

* **scatter():**

Функция, отвечающая за построение рассеянной диаграммы зарплат работников по возрасту. Функция не имеет входных параметров. Ничего не возвращает.

* **bar\_covid():**

Функция, отвечающая за построение столбчатой диаграммы вакцинированных и не вакцинированных от COVID-19. Функция не имеет входных параметров. Ничего не возвращает.

* **bar\_sadik()**:

Функция, отвечающая за построение столбчатой диаграммы распределения детей по садикам. Функция не имеет входных параметров. Ничего не возвращает.

* **bar\_otdel\_salary():**

Функция, отвечающая за построение столбчатой диаграммы средней зарплаты по отделам. Функция не имеет входных параметров. Ничего не возвращает.

* **bar\_otdeli():**

Функция, отвечающая за построение столбчатой диаграммы распределения сотрудников по отделам. Функция не имеет входных параметров. Ничего не возвращает.

* **hist\_salary():**

Функция, отвечающая за построение гистограммы распределения зарплат рабочих. Функция не имеет входных параметров. Ничего не возвращает.

* **hist\_birth\_workers():**

Функция, отвечающая за построение гистограммы распределения рабочих по годам рождения. Функция не имеет входных параметров. Ничего не возвращает.

* **hist\_birth\_children():**

Функция, отвечающая за построение гистограммы распределения детей по годам рождения. Функция не имеет входных параметров. Ничего не возвращает.

* make\_plot(dictionary, graph\_type, num):

Функция, генерирующая график. Входные параметры: dictionary – наименование справочника, graph\_type – тип графика, требуемого к построению, num – наименование требуемого графика.

1. **db\_interaction.py** – модуль, ответственный за запросы к справочникам и генерации текстовых отчетов, операций сохранения, добавления и удаления. Содержит в себе импорт модулей pandas, tkinter, datetime, а также набор функций из модуля checking.py . Содержит в себе следующие функции:

* **init\_db():**

Функция отвечает за загрузку данных из внутреннего каталога. Не имеет входных параметров, ничего не возвращает.

* **load\_default\_text():**

Функция отвечает за загрузку текста стартового экрана. Не имеет входных параметров, ничего не возвращает.

* **load\_info\_text():**

Функция отвечает за загрузку текста с информацией о приложении. Не имеет входных параметров. Ничего не возвращает.

* **load\_help\_text():**

Функция отвечает за загрузку текста со справкой о приложении. Не имеет входных параметров, ничего не возвращает.

* **end(root):**

Функция, отвечающая за закрытие главного окна. Входной параметр – root – указатель на объект .Tk() . Ничего не возвращает.

* **return\_dict(dict\_name):**

Функция отвечает за возврат объекта в виде справочника по его названию. Входной параметр – dict\_name – наименование требуемого справочника. Возвращает: либо объект DataFrame, либо None, если название не является корректным.

* **data\_transfer(data\_):**

Функция отвечает за приведение даты в формате строки к типу datetime. Входной параметр – data\_ – строка. Возвращает – объект Datetime, соответствующий введенной строке.

* **adding\_to\_workers(fio, birth, child, vac, dep, prof, pay):**

Функция отвечает за добавление строки в справочник workers.

Входные параметры: fio – ФИО, birth – Дата рождения, child – ФИО Ребенка, vac – наличие прививки, dep – номер отдела, prof – должность, pay – заработная плата. Ничего не возвращает.

* **adding\_to\_children(fio, birth\_ch, k\_gard):**

Функция отвечает за добавление строки в справочник children.

Входные параметры: fio – ФИО Ребенка, birth\_ch – Дата рождения ребенка, k\_gard – номер садика. Ничего не возвращает.

* **adding\_to\_otdeli(num, date, tel, num\_workers):**

Функция отвечает за добавление строки в справочник otdeli.

Входные параметры: num – номер отдела, date – дата создания отдела, tel – номер телефона, num\_workers – количество работников. Ничего не возвращает.

* **deleting(dict\_name, index):**

Функция отвечает за удаление строки по индексу из объекта DataFrame. Входные параметры: dict\_name – наименование справочника, index – индекс. Возвращает – копию объекта DataFrame, но без строки с индексом index.

* **save():**

Функция отвечает за сохранение копии объекта DataFrame в формате Pickle. Входных параметров нет, ничего не возвращает.

* **back\_attr(dict\_name, name):**

Функция возвращает поле по имени заданного атрибута. Входные данные: dict\_name – наименование словаря, name – имя запрашиваемого атрибута. Возвращет – объект Dataframe.

* **back\_named\_col(dict\_name, names):**

Функция возвращает множество полей по заданным атрибутам, если таковые имеются. Входные данные: dict\_name – наименование словаря, name – имена запрашиваемых атрибутов. Возвращает – объект DataFrame.

* **back\_names\_cond(dict\_name, names, by\_name, condition):**

Функция возвращает множество полей атрибутов, если параметр by\_name удовлетворяет condition.

Входные данные: dict\_name – наименование словаря, names – имена выводимых атрибутов, by\_name – имя атрибута, по которому происходит сравнение, condition – условие для проверки по by\_name. Возвращает – объект DataFrame.

* **back\_many\_cond(dict\_name, names, by\_names, conditions):**

Функция возвращает поля атрибутов, если параметр by\_name[i] удовлетворяет condition[i] (кол-во: 2).

Входные данные: dict\_name – наименование словаря, names – имена выводимых атрибутов, by\_names – имена атрибутов, по которым происходит сравнение, conditions – условия для i-того имени. Возвращает – объект DataFrame.

* **remove\_workers(index):**

Функция отвечает за удаление записи из справочника workers по индексу. Входной параметр – index – индекс. Ничего не возвращает.

* **remove\_children(index):**

Функция отвечает за удаление записи из справочника children по индексу. Входной параметр – index – индекс. Ничего не возвращает.

* **remove\_otdeli(index):**

Функция отвечает за удаление записи из справочника otdeli по индексу. Входной параметр – index – индекс. Ничего не возвращает.

* **save\_workers(index, fio, child, vac, dep, prof, pay):**

Функция отвечает за добавление новой записи в справочник workers и его сохранение. Входные параметры: index – индекс, fio – ФИО, birth – Дата рождения, child – ФИО Ребенка, vac – наличие прививки, dep – номер отдела, prof – должность, pay – заработная плата. Ничего не возвращает.

* **save\_children(index, fio, birth, garden):**

Функция отвечает за добавление новой записи в справочник children и его сохранение. Входные параметры: index – индекс, fio – ФИО Ребенка, birth– Дата рождения ребенка, garden – номер садика. Ничего не возвращает.

* **save\_otdeli(index, num, date, tel, num\_workers):**

Функция отвечает за добавление новой записи в справочник otdeli и его сохранение. Входные параметры: index – индекс, num – номер отдела, date – дата создания отдела, tel – номер телефона, num\_workers – количество работников. Ничего не возвращает.

1. checking.py – модуль, содержащий набор функций для проверки формата вводимых данных. Включает в себя следующие функции:

* **chars(string):**

Функция проверяет, все ли введенные символы, кроме точек, являются буквами. Входные данные: string – строка, выходные данные – в случае истины возвращает string, в случае обратного – логическую константу False.

* **numerical(string):**

Функция проверяет, все ли введенные в строку символы являются числами. Входные данные – string – строка. В случае истины возвращает string, в случае обратного – False.

* **phone\_number(number):**

Функция проверяет, является ли введенная строка номером телефона. Входные данные – number – строка. В случае истины возвращает True, в случае обратного – False.

* **data(data\_str):**

Функция проверяет, является ли введенная строка датой. Входные данные – data\_str – строка. В случае истины возвращает True, в обратном случае – False.

# Технические требования

Минимальными техническими требованиями для работы программы являются требования, указанные для корректной работы продукта Anaconda c версией Python 3.8 (с подробным описанием установки и документации можно ознакомиться по ссылке: <https://docs.anaconda.com/anaconda/install/>); персональный компьютер с предустановленной версией Windows 7 (и выше), MacOS или Linux, c характеристиками железа, позволяющему запустить анаконду. Также для корректного отображения необходим экран с разрешением 1280 x 720, а также устройства ввода: клавиатура и мышь.