ЛАБОРАТОРНЫЙ ПРАКТИКУМ DATA-MANAGEMENT

[Lab\_1a 2](#_Toc41465836)

[Вариант 0 (тестовый вариант) 2](#_Toc41465837)

[Вариант 1 3](#_Toc41465838)

[Вариант 2 3](#_Toc41465839)

[Вариант 3 4](#_Toc41465840)

[Вариант 4 5](#_Toc41465841)

[Вариант 5 6](#_Toc41465842)

[Вариант 6 7](#_Toc41465843)

[Вариант 7 7](#_Toc41465844)

[Вариант 8 7](#_Toc41465845)

[Вариант 9 7](#_Toc41465846)

[Вариант 10 7](#_Toc41465847)

[Lab\_1b 7](#_Toc41465848)

[Вариант 0 (тестовый вариант) ДЛЯ НАС 7](#_Toc41465849)

[Lab\_1c 7](#_Toc41465850)

[Lab\_1d 7](#_Toc41465851)

# **Lab\_1a**

По предметной области, согласно выданному преподавателем варианту заданий, используя SQL написать запросы на создание следующих таблиц (таблицы).

SQL запрос сохранить и отправить в файле <ваши\_инициалы>\_lab1a\_<номер\_варианта>.txt

*Пример*: gar\_lab1a\_0.txt

Варианты типов данных:

* INTEGER - для целого числа,
* TEXT - для строки символов,
* REAL - для чисел с плавающей точкой (действительных),
* BOOLEAN – логический тип, для статуса истина-ложь,
* DATETIME - для хранения информации о дате и времени.

SQLite <https://www.sqlite.org/index.html>

## **Вариант 0 (тестовый вариант)**

Крупная компания нуждается в БД, которая будет хранить список сотрудников и информацию о них. Необходимо создать следующие таблицы:

Таблица **employees** содержит следующие поля:

* **id** – индентификатор пользователя, целое число, НЕ может содержать NULL,
* **name** - имя сотрудника, НЕ может быть NULL,
* **birthday** - дата рождения сотрудника, может содержать NULL,
* **sex** - пол сотрудника, строка, НЕ может быть NULL,
* **active** - активен ли пользователь или нет, тип BOOL, по умолчанию FALSE,
* **salary** - зарплата сотрудника, целое, НЕ может быть NULL

Пример SQL запроса:

1. create table employees (
2. id INTEGER not null,
3. name TEXT not null,
4. birthday DATETIME,
5. sex TEXT not NULL,
6. active BOOLEAN default FALSE,
7. salary INTEGER not null
8. );

**ВАРИАНТЫ ЗАДАНИЙ**

## **Вариант 1**

Требуется создать БД для диспансера, в которой будут храниться данные о сотрудниках и информация о принимаемых законах. База данных состоит из слудющих таблиц:

Таблица "**employee\_data**" содержит следующие поля:

    \* **id** – идентификатор пользователя, целое число, НЕ может содержать NULL,

    \* **name\_employee** - имя сотрудника, НЕ может быть NULL

    \* **position** - должность сотрудника

    \* **employee\_growth** - рост сотрудника, целое число

    \* **employee\_weight** - вес сотрудника, действительное число

Таблица "**patient\_data**" содержит следующие поля:

    \* **id** – идентификатор больного, целое число, НЕ может содержать NULL,

    \* **name\_patient** - имя пациента, НЕ может быть NULL

    \* **diagnosis** - диагноз, текст

    \* **patient\_growth** - рост пациента, целое число

    \* **patient\_weight** - вес пациента, действительное число

Таблица "**incident\_data**" содержит следующие поля:

    \* **id** – идентификатор инцидента, целое число, НЕ может содержать NULL,

    \* **date\_incident** - дата инцидента, НЕ может содержать NULL

    \* **name\_victim** - имя пострадавшего, текст, НЕ может содержать NULL

    \* **name\_initiator** - имя инициатора,

    \* **degree\_damage** - степень тяжести повреждения, целое число

    \* **damage** - повреждение, текст, НЕ может содержать NULL

    \* **start\_sick** - дата, начало больничного

    \* **period\_sick** - срок больничного(дни), целое число

## **Вариант 2**

 Требуется создать БД для гос.думы, в которой будут храниться данные о сотрудниках и информация о принимаемых законах. База данных состоит из двух таблиц:

Таблица "**employee\_data**" содержит следующие поля:

    \* **id** – идентификатор сотрудника, целое число, НЕ может содержать NULL,

    \* **name** - имя сотрудника, НЕ может быть NULL

    \* **data\_revenue** - данные о доходах, действительное число

    \* **data\_estate** - данные о недвижимости, текст

Таблица "**data\_laws**" содержит следующие поля:

    \* **id** – идентификатор закона, целое число, НЕ может содержать NULL,

    \* **law\_number** - номер закона, целое число, НЕ может содержать NULL,

    \* **rights\_section** - раздел права (тема), текст

    \* **title** - краткое наименование закона,

    \* **who\_nominated** - кто выдвинул,

    \* **who\_corrected** - кто корректировал,

    \* **votes\_for** - голоса “за”, целое число, НЕ может содержать NULL

    \* **votes\_againist** - голоса “против”, целое число, НЕ может содержать NULL

## **Вариант 3**

Требуется создать БД для строительной компании, в которой будут храниться данные о сотрудниках, строительных объектах и информация о проведенных работах. База данных состоит из трех таблиц:

Таблица "**employee\_data**" содержит следующие поля:

    \* **id** – идентификатор пользователя, целое число, НЕ может содержать NULL,

    \* **name\_employee** - имя работника, текст, НЕ может быть NULL,

    \* **citizenship** - гражданство, текст

    \* **end\_permission** - срок окончания разрешения на работу для мигрантов, дата

    \* **position** - должность работника,

    \* **tariff\_rate** - тарифная ставка рабочего, действительное число

Таблица "**construction\_objects**" содержит следующие поля:

    \* **id** – идентификатор строения, целое число, НЕ может содержать NULL,

    \* **name\_object** - название объекта строительства, текст

    \* **city** - имя города объекта,

    \* **type\_financing** - тип финансирования (частное, кредит, государственное, долевое), текст

    \* **start\_date** - дата начала работ

    \* **deadline** - срок сдачи, дата

Таблица "**work\_data**" содержит следующие поля:

    \* **id** – идентификатор работы, целое число, НЕ может содержать NULL,

    \* **name\_work** - название работы, НЕ может содержать NULL

    \* **work\_description** - описание работы,

    \* **name\_employee** - имя работника, выполнившего работу, НЕ может быть NULL, текст

    \* **number\_hours** - количество часов на выполнение, целое число

## **Вариант 4**

Требуется создать БД автомобильной компании, в которой будут храниться данные о заказах. База данных состоит из трех таблиц:

Таблица "**details\_data**" содержит следующие поля:

    \* **id** – идентификатор детали, целое число, НЕ может содержать NULL,

    \* **number\_detail** - номер детали, целое число, НЕ может содержать NULL,

    \* **manufacturer** - производитель, текст

    \* **name\_rus** - название на русском,

    \* **name\_eng** - название на английском,

    \* **weight** - вес детали, действительное число, НЕ может содержать NULL

    \* **type** - тип детали, текст, НЕ может содержать NULL

    \* **price** - цена детали, действительное число, НЕ может содержать NULL

    \* **warehouse** - склад хранения, текст, НЕ может содержать NULL

    \* **quantity** - количество деталей на складе, целое число, НЕ может содержать NULL

Таблица "**invoice\_data**" содержит следующие поля:

    \* **id** – идентификатор накладной, целое число, НЕ может содержать NULL,

    \* **invoice\_number** - номер накладной, целое число, , НЕ может содержать NULL

    \* **date\_sale** - дата продажи, НЕ может содержать NULL

    \* **warehouse\_sale** - склад продажи, текст, НЕ может содержать NULL

    \* **quantity** - количество поданного, целое число, НЕ может содержать NULL

Таблица "**order\_data**" содержит следующие поля:

    \* **id** – идентификатор заказа, целое число, НЕ может содержать NULL,

    \* **order\_date** - дата заказа, НЕ может содержать NULL

    \* **order\_list** - что заказано, текст, НЕ может содержать NULL

    \* **warehouse\_loading** - склад погрузки (откуда), текст

    \* **delivery\_date-** дата поставки (когда получат товар), дата

    \* **warehouse\_delivery-** склад поставки (куда), текст

    \* **quantity** - количество, целое число, НЕ может содержать NULL

## **Вариант 5**

Требуется создать БД страховой компании, в которой будут храниться данные о страховых случаях. База данных состоит из трех таблиц:

Таблица "**agents\_data**" содержит следующие поля:

    \* **id** – идентификатор агента, целое число, НЕ может содержать NULL,

    \* **name\_agent** - имя агента, текст, НЕ может содержать NULL,

    \* **birthday\_agent** - дата рождения агента

    \* **experience** - опыт работы, действительное число

Таблица "**policyholders\_data**" содержит следующие поля:

    \* **id** – идентификатор страхователя, целое число, НЕ может содержать NULL,

    \* **name\_policyholder** - имя страхователя, текст, НЕ может содержать NULL,

    \* **birthday\_policyholder** - дата рождения страхователя

    \* **gender** - пол, текст

    \* **nationality** - национальность, текст

    \* **type\_insurance** - тип страховки (травма, болезнь, смерть, и т.п.), текст, НЕ может содержать NULL

    \* **date\_insurance** - дата страхования, НЕ может содержать NULL

    \* **city\_insurance** - город страхования, текст

Таблица "**cases\_data**" содержит следующие поля:

    \* **id** – идентификатор страхового случая, целое число, НЕ может содержать NULL,

    \* **name\_policyholder** - имя страхователя, текст, НЕ может содержать NULL,

    \* **refund\_amount** - сумма возмещения, действительное число, НЕ может содержать NULL

    \* **number\_policy** - номер полиса, целое число, НЕ может содержать NULL,

    \* **prize\_amount** - сумма страховой премии, действительное число, НЕ может содержать NULL

    \* **policyholders\_amount** - сумма (стоимость) страховки, действительное число, НЕ может содержать NULL

## **Вариант 6**

## **Вариант 7**

## **Вариант 8**

## **Вариант 9**

## **Вариант 10**

# **Lab\_1b**

На основе ORM **peewee** необходимо создать БД **mydatabase.db** по предметной области из Lab\_1a и проинициализировать нужные таблицы.

Скрипт сохранить и отправить в файле <*ваши\_инициалы>\_*lab1b\_<*номер\_варианта*>.py

*Пример*: gar\_lab1b\_0.py

Варианты типов данных:

* IntegerField - для целого числа,
* TextField - для строки символов,
* FloatField - для чисел с плавающей точкой (действительных),
* BooleanField - логический тип, для статуса истина-ложь
* DateTimeField - для хранения информации о дате и времени.

Документация peewee - <http://docs.peewee-orm.com/en/latest/>

Типы полей - <http://docs.peewee-orm.com/en/latest/peewee/api.html#fields-api>

## **Вариант 0 (тестовый вариант) ДЛЯ НАС**

1. # вариант 0 lab1b
2. # это правильная выполненая работа
3. from peewee import \*
5. DATABASE\_NAME = 'mydatabase.db'
7. db = SqliteDatabase(DATABASE\_NAME)

10. class employees(Model): # таблица "employees"
11. id = IntegerField(default=1)
12. name = TextField()
13. birthday = DateTimeField(null=True)
14. sex = TextField()
15. active = BooleanField(default=False)
16. salary = IntegerField()
18. class Meta:
19. database = db
20. table\_name = 'employees'
22. db.connect()
23. db.create\_tables([employees])
24. db.commit()
25. db.close()

# **Lab\_1c**

В скрипте из Lab1\_b необходимо реализовать функцию **add\_row**, которая добавит запись в нужную таблицу.

Функция принимает запись типа кортеж (что добавить) и имя таблицы (куда добавить).

Пример вызова функции:

1. add\_row((1, ”Alexandr”, 2002-03-07,”m”, True, 60000), ”employees”)

Скрипт сохранить и отправить в файле <*ваши\_инициалы>\_*lab1с\_<*номер\_варианта*>.py

*Пример*: gar\_lab1с\_0.py

# **Lab\_1d**

В скрипте необходимо реализовать функцию **show\_table**, которая выводит содержимое таблицы в красивом виде (с разделителями). В качестве аргумента функция принимает имя таблицы.

Пример вызова функции:

1. **show\_table**(”employees”))

 Скрипт сохранить и отправить в файле <*ваши\_инициалы>\_*lab1d\_<*номер\_варианта*>.py

*Пример*: gar\_lab1d\_0.py