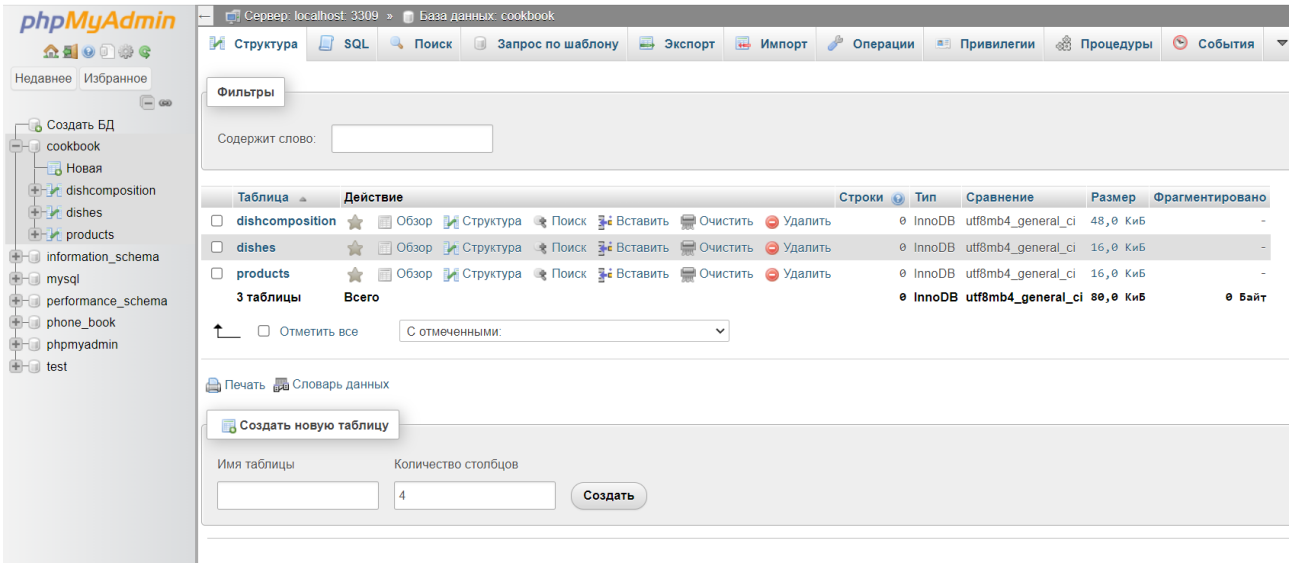
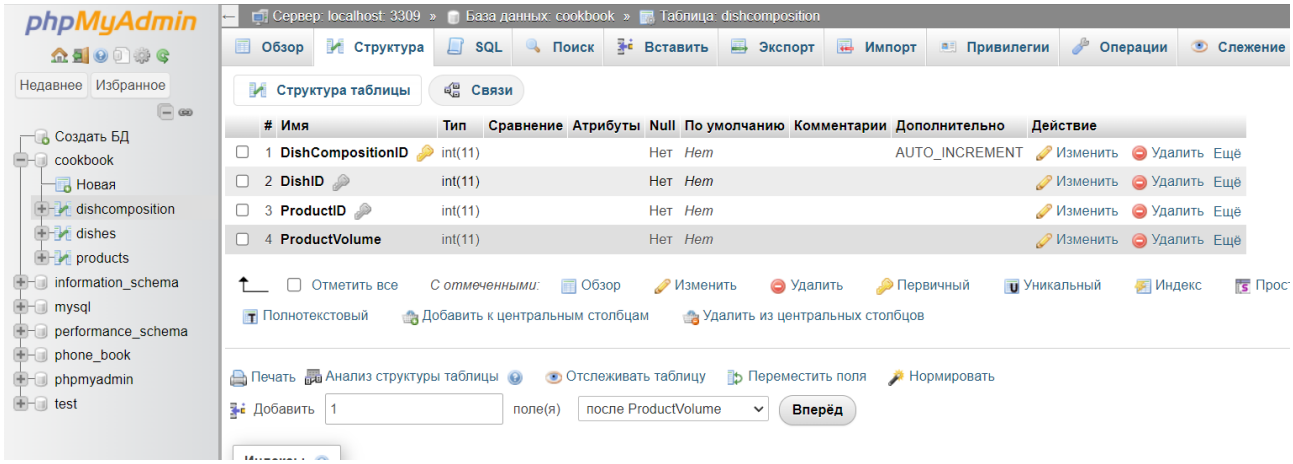
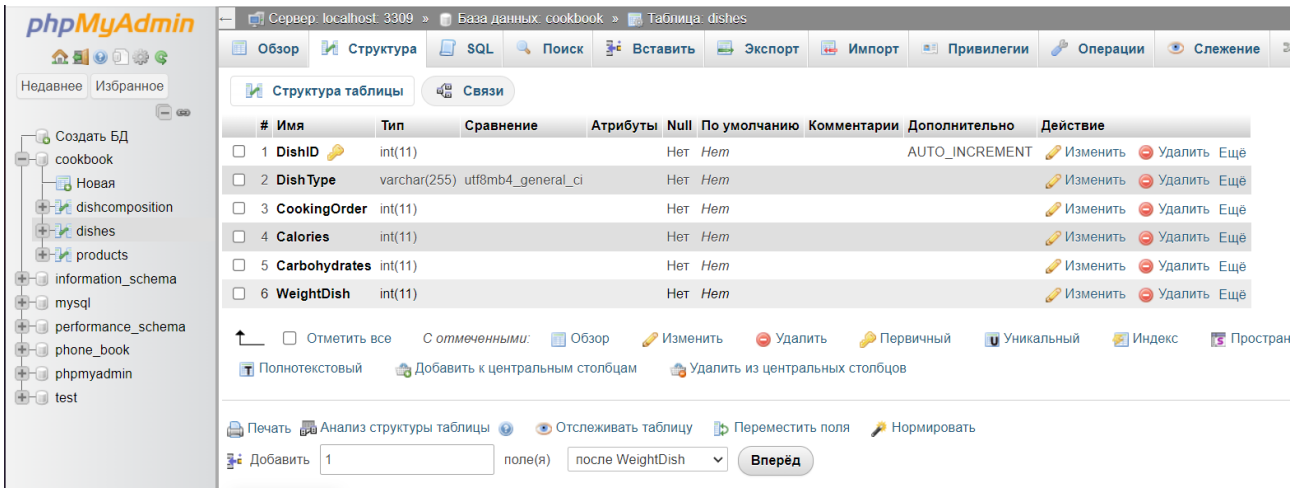
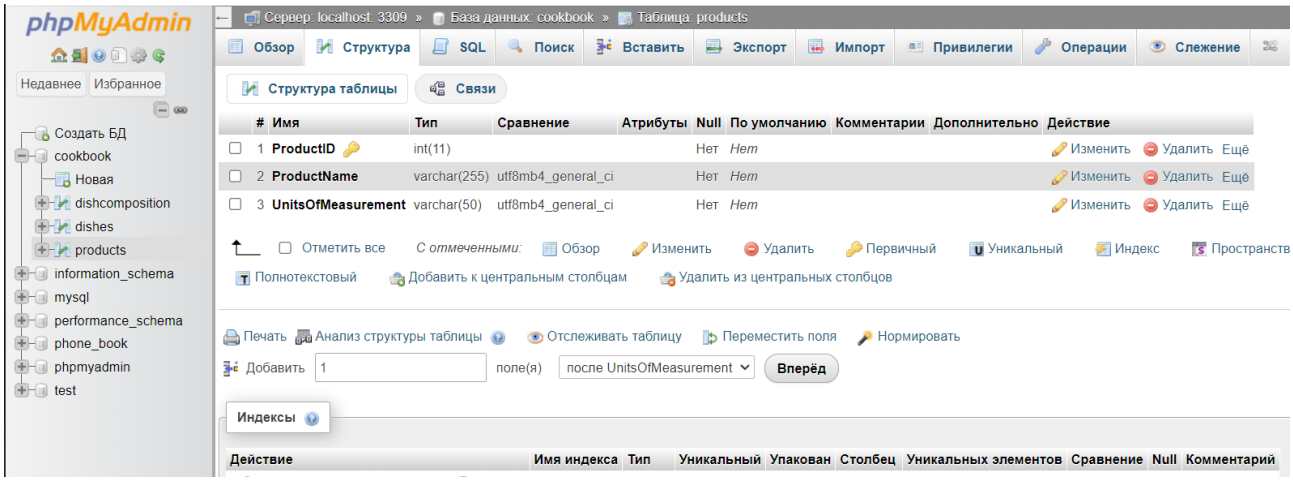
1. Создаём БД

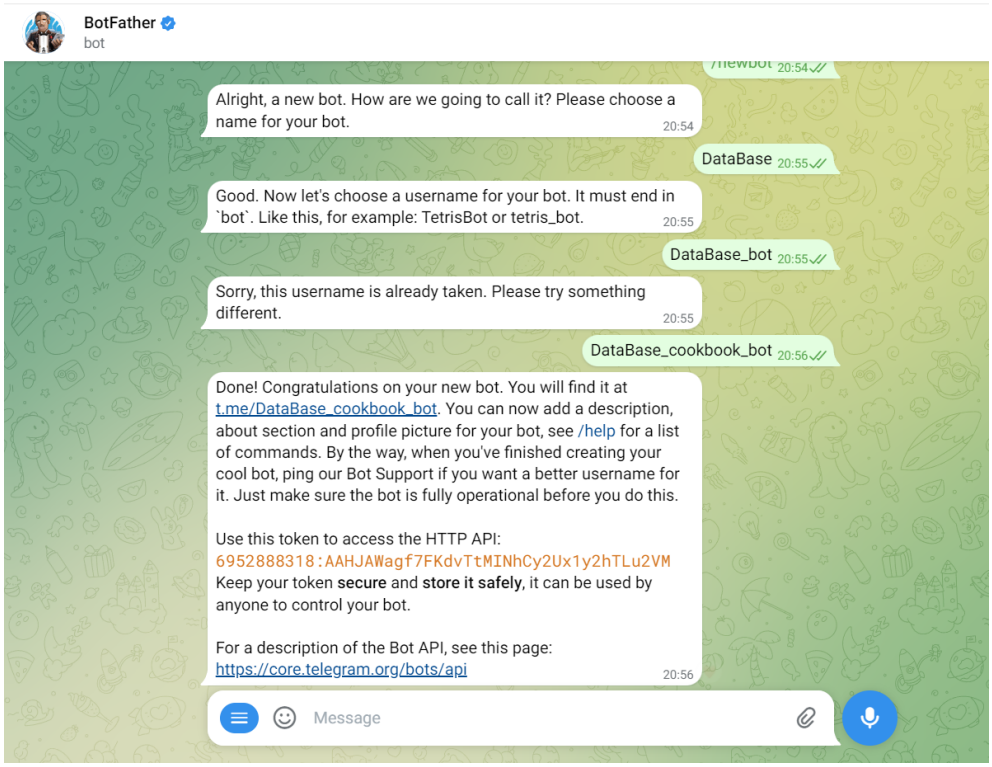

phpMyAdmin
Недавнее Избранное
Создать БД
cookbook
Новая
dishcomposition
dishes
products
information_schema
mysql
performance_schema
phone_book
phpmyadmin
test
Сервер: localhost:3309 » База данных: cookbook
Структура SQL Поиск Запрос по шаблону Экспорт Импорт Операции Привилегии Процедуры События
Фильтры
Содержит слово:
Таблица Действие Строки Тип Сравнение Размер Фрагментировано
dishcomposition Обзор Структура Поиск Вставить Очистить Удалить 0 InnoDB utf8mb4_general_ci 48,0 КиБ -
dishes Обзор Структура Поиск Вставить Очистить Удалить 0 InnoDB utf8mb4_general_ci 16,0 КиБ -
products Обзор Структура Поиск Вставить Очистить Удалить 0 InnoDB utf8mb4_general_ci 16,0 КиБ -
3 таблицы Всего 0 InnoDB utf8mb4_general_ci 80,0 КиБ 0 Байт
Отметить все С отмеченными:
Печать Словарь данных
Создать новую таблицу
Имя таблицы Количество столбцов
4
Создать


phpMyAdmin
Недавнее Избранное
Создать БД
cookbook
Новая
dishcomposition
dishes
products
information_schema
mysql
performance_schema
phone_book
phpmyadmin
test
Сервер: localhost:3309 » База данных: cookbook » Таблица: dishcomposition
Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Слежение
Структура таблицы Связи
Имя Тип Сравнение Атрибуты Null По умолчанию Комментарии Дополнительно Действие
1 DishCompositionID int(11) Нет Нем AUTO_INCREMENT Изменить Удалить Ещё
2 DishID int(11) Нет Нем Изменить Удалить Ещё
3 ProductID int(11) Нет Нем Изменить Удалить Ещё
4 ProductVolume int(11) Нет Нем Изменить Удалить Ещё
Отметить все С отмеченными: Обзор Изменить Удалить Первичный Уникальный Индекс
Полнотекстовый Добавить к центральным столбцам Удалить из центральных столбцов
Печать Анализ структуры таблицы Отслеживать таблицу Переместить поля Нормировать
Добавить 1 поле(я) после ProductVolume Вперёд


phpMyAdmin
Недавнее Избранное
Создать БД
cookbook
Новая
dishcomposition
dishes
products
information_schema
mysql
performance_schema
phone_book
phpmyadmin
test
Сервер: localhost:3309 » База данных: cookbook » Таблица: dishes
Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Слежение
Структура таблицы Связи
Имя Тип Сравнение Атрибуты Null По умолчанию Комментарии Дополнительно Действие
1 DishID int(11) Нет Нем AUTO_INCREMENT Изменить Удалить Ещё
2 DishType varchar(255) utf8mb4_general_ci Нет Нем Изменить Удалить Ещё
3 CookingOrder int(11) Нет Нем Изменить Удалить Ещё
4 Calories int(11) Нет Нем Изменить Удалить Ещё
5 Carbohydrates int(11) Нет Нем Изменить Удалить Ещё
6 WeightDish int(11) Нет Нем Изменить Удалить Ещё
Отметить все С отмеченными: Обзор Изменить Удалить Первичный Уникальный Индекс
Полнотекстовый Добавить к центральным столбцам Удалить из центральных столбцов
Печать Анализ структуры таблицы Отслеживать таблицу Переместить поля Нормировать
Добавить 1 поле(я) после WeightDish Вперёд



2. Создаем телеграмм бота. Через BotFather получаем токен.



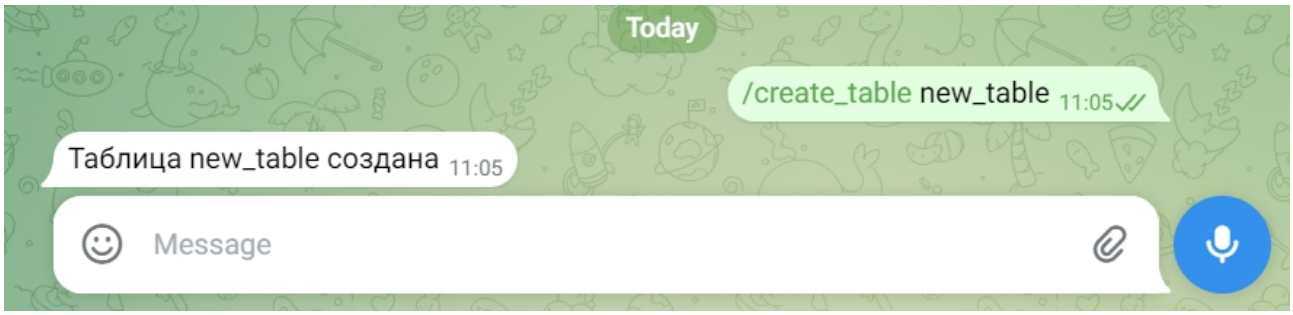
3. Прописываем код на python, который будет взаимодействовать с нашей БД

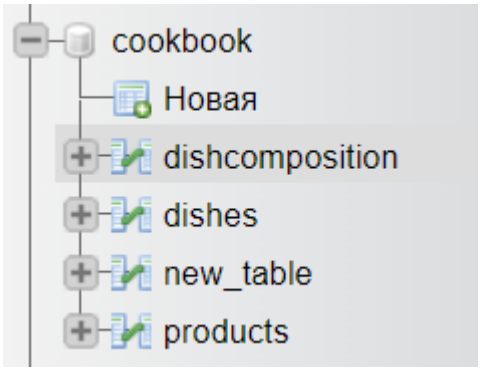
```python
import mysql.connector
from telegram import Update
from telegram.ext import Updater, CommandHandler, CallbackContext

# Подключение к базе данных
db_connection = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="cookbook",
    port='3309'
)

# Функция для создания таблицы с заданным именем
def create_table(update: Update, context: CallbackContext):
```

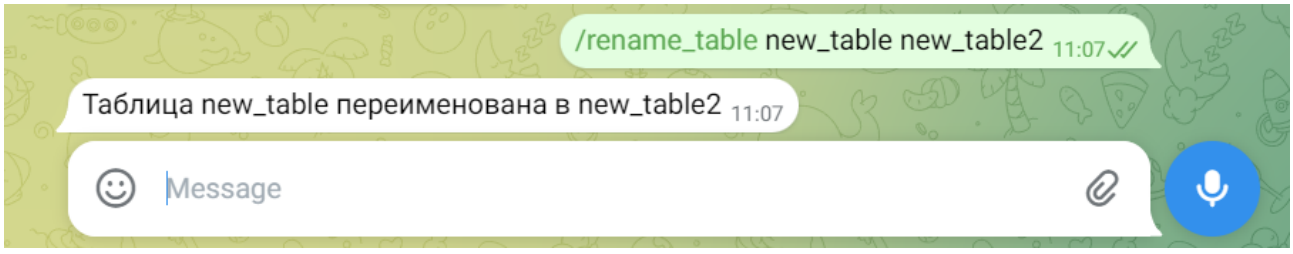
```
    table_name = "example_table"  # Имя таблицы по умолчанию
    if context.args:  # Если в сообщении есть аргументы, используем их как имя таблицы
        table_name = ' '.join(context.args)
    cursor = db_connection.cursor()
    cursor.execute(f"CREATE TABLE IF NOT EXISTS {table_name} (id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255), age INT)")
    db_connection.commit()
    update.message.reply_text(f'Таблица {table_name} создана')

# Функция для удаления заданной таблицы
def delete_table(update: Update, context: CallbackContext):
    if context.args:
        table_name = ' '.join(context.args)
        cursor = db_connection.cursor()
        cursor.execute(f"DROP TABLE IF EXISTS {table_name}")
        db_connection.commit()
        update.message.reply_text(f'Таблица {table_name} удалена')
    else:
        update.message.reply_text('Пожалуйста, укажите имя таблицы для удаления')

# Функция для переименования таблицы
def rename_table(update: Update, context: CallbackContext):
    if len(context.args) >= 2:
        old_table_name = context.args[0]
        new_table_name = context.args[1]
        cursor = db_connection.cursor()
        cursor.execute(f"ALTER TABLE {old_table_name} RENAME TO {new_table_name}")
        db_connection.commit()
        update.message.reply_text(f'Таблица {old_table_name} переименована в {new_table_name}')
    else:
        update.message.reply_text('Пожалуйста, укажите имя текущей таблицы и новое имя')



# Функция для вставки данных в таблицу
def insert_data(update: Update, context: CallbackContext):
    if len(context.args) >= 3:
        table_name = context.args[0]
        columns = context.args[1]
        values = context.args[2:]
        placeholders = ', '.join(['%s'] * len(values))
        cursor = db_connection.cursor()
        cursor.execute(f"INSERT INTO {table_name} ({columns}) VALUES ({placeholders})", values)
        db_connection.commit()
        update.message.reply_text(f'Данные вставлены в таблицу {table_name}')
    else:
        update.message.reply_text('Использование: /insert_data table_name column1, column2 value1, value2')




# Функция для выборки данных из таблицы
def select_data(update: Update, context: CallbackContext):
    if context.args:
        table_name = context.args[0]
        cursor = db_connection.cursor(prepared=True)
        cursor.execute(f"SELECT * FROM {table_name}")
        data = cursor.fetchall()
        update.message.reply_text(f'Данные из таблицы {table_name}: {data}')
    else:
        update.message.reply_text('Пожалуйста, укажите имя таблицы для выборки данных')
```

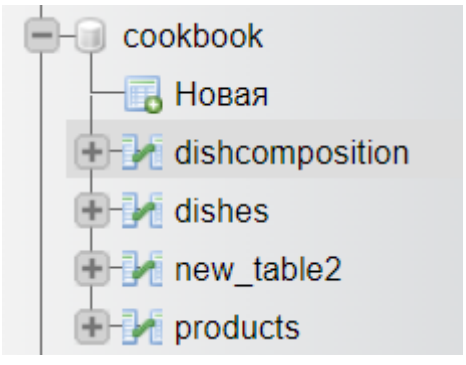
```
def main():
    updater = Updater("6952888318:AAHJAWagf7FKdvTtMINhCy2Ux1y2hTLu2VM", use_context=True)
    dispatcher = updater.dispatcher

    dispatcher.add_handler(CommandHandler("create_table", create_table, pass_args=True))
    dispatcher.add_handler(CommandHandler("delete_table", delete_table, pass_args=True))
    dispatcher.add_handler(CommandHandler("rename_table", rename_table, pass_args=True))
    dispatcher.add_handler(CommandHandler("insert_data", insert_data, pass_args=True))
    dispatcher.add_handler(CommandHandler("select_data", select_data, pass_args=True))

    updater.start_polling()
    updater.idle()

if __name__ == '__main__':
    main()
```

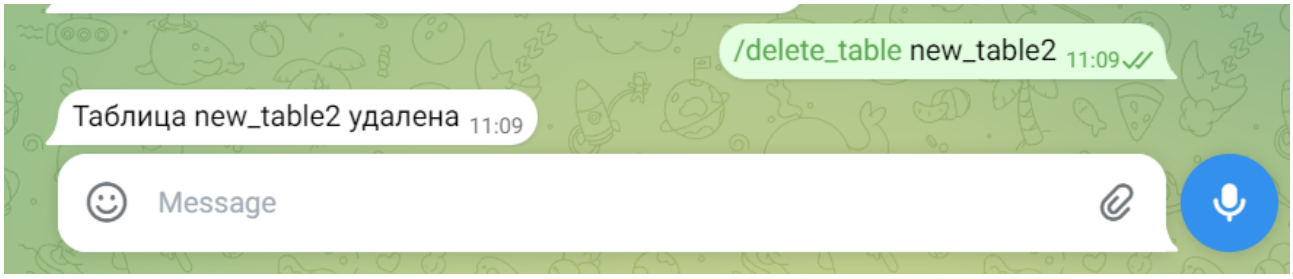
4. Запускаем код и проверяем работоспособность бота.

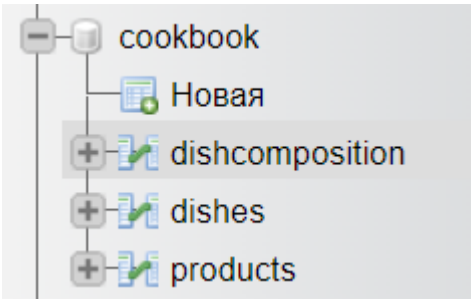


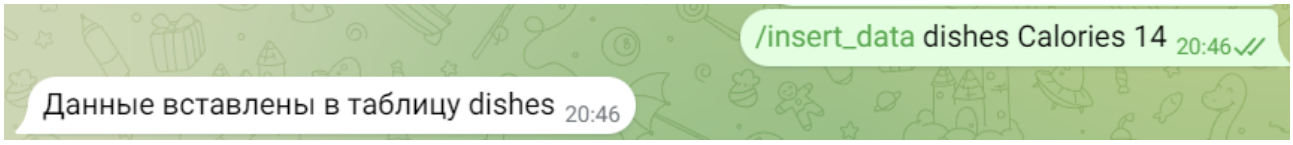


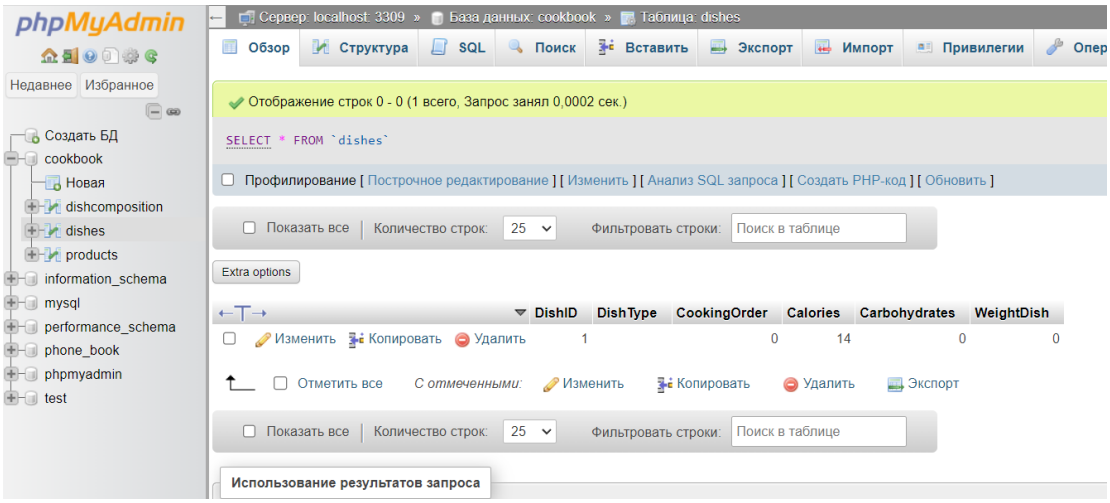


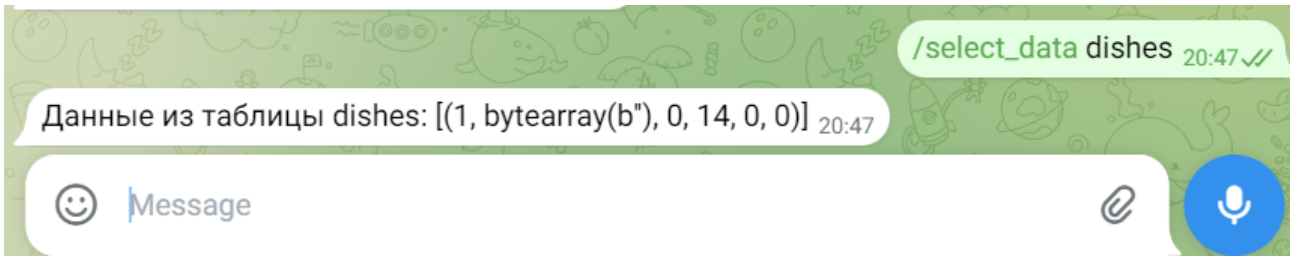



/delete_table new_table2 11:09
Таблица new_table2 удалена 11:09
Message


cookbook
Новая
dishcomposition
dishes
products


/insert_data dishes Calories 14 20:46
Данные вставлены в таблицу dishes 20:46


phpMyAdmin
Недавнее
Избранное
Создать БД
cookbook
Новая
dishcomposition
dishes
products
information_schema
mysql
performance_schema
phone_book
phpmyadmin
test
Сервер: localhost:3309 » База данных: cookbook » Таблица: dishes
Обзор
Структура
SQL
Поиск
Вставить
Экспорт
Импорт
Привилегии
Отображение строк 0 - 0 (1 всего, Запрос занял 0,0002 сек.)
SELECT * FROM `dishes`
Профилирование [Построчное редактирование] [Изменить] [Анализ SQL запроса] [Создать PHP-код] [Обновить]
Показать все
Количество строк: 25
Фильтровать строки: Поиск в таблице
Extra options
DishID
DishType
CookingOrder
Calories
Carbohydrates
WeightDish
Изменить
Копировать
Удалить
1
0
14
0
0
Отметить все
С отмеченными:
Изменить
Копировать
Удалить
Экспорт
Показать все
Количество строк: 25
Фильтровать строки: Поиск в таблице
Использование результатов запроса


/select_data dishes 20:47
Данные из таблицы dishes: [(1, bytearray(b''), 0, 14, 0, 0)] 20:47
Message