Цель: Создать бота, который бы сообщал расписание.

Ход работы:

Импортируем необходимые библиотеки для обращения к БД, создания бота и вычисления даты.

import psycopg2

import telebot

from telebot import types

import datetime

from datetime import date, datetime

Задаём параметры обращения к БД.

conn = psycopg2.connect(database="schedular\_db",

user="postgres",

password='1',

host="localhost",

port="5432")

cursor = conn.cursor()

Задаём параметры для подключения к боту.

token = '5033454002:AAExGn47yblfMHoV5B5ohtPYdya9JiCiW7g'

bot = telebot.TeleBot(token)

Используем библиотеку Datatime для оперделения нынешней недели.

current\_date = date.today()

first\_week = date(2021, 9, 3)

dellta = current\_date - first\_week

parity = (dellta.days // 7) % 2

if parity == 0:

parity = 't'

parity1 = 'f'

else:

parity = 'f'

parity1 = 't'

Чтобы получить порядок нынешней недели нужно к результату деления delta.days на 7 нацело прибавить два, так как в счёте нет первой недели, а нынешняя уже началась с понедельника, но так как на чётность это не влияет, мы не берём этого в расчёт.

Делаем реакцию бота на команду старт.

@bot.message\_handler(commands=['start'])

def start(message):

keyboard = types.ReplyKeyboardMarkup()

keyboard.row("/Понедельник", "/Вторник", "/Среда", "/Четверг", "/Пятница", "/Расписание\_на\_текущую\_неделю",

"/Расписание на следующую неделю")

bot.send\_message(message.chat.id, 'Привет! Хотите узнать расписание?', reply\_markup=keyboard)

Делаем реакцию на понедельник.

@bot.message\_handler(commands=['Понедельник'])

def monday(message):

keyboard = types.ReplyKeyboardMarkup()

keyboard.row("/Понедельник", "/Вторник", "/Среда", "/Четверг", "/Пятница", "/Текущая",

"/Следующая")

cursor.execute("SELECT timetable.subject, timetable.room\_numb, timetable.start\_time, teacher.name FROM timetable, teacher WHERE day='Понедельник' AND chet=%s AND timetable.subject=teacher.subject ORDER BY timetable.start\_time ASC;", parity)

records = list(cursor.fetchall())

day = 'Понедельник.'

for i in range(len(records)):

cl = str(records[i])

cl = cl.replace("(","")

cl = cl.replace(")", "")

cl = cl.replace("'", "")

day += '\n' + cl + '.'

bot.send\_message(message.chat.id, (day),reply\_markup=keyboard)

Делаем запрос на две таблицы из БД – учителя и расписание. Из них берём необходимые данные на вывод, согласно чётности недели. Сортируем по времени. Затем, вывод считываем построчно, избавляясь от ненужных скобок и кавычек. Выводим пользователю. То же самое с другими днями недели.

В расписание текущей недели делаем каждый день недели, не забывая ставить после каждого дня ‘\n’ дважды, чтобы визуально было проще различить дни.

@bot.message\_handler(commands=['Текущая'])

def t\_week(message):

print(parity)

keyboard = types.ReplyKeyboardMarkup()

keyboard.row("/Понедельник", "/Вторник", "/Среда", "/Четверг", "/Пятница", "/Текущая",

"/Следующая")

cursor.execute("SELECT timetable.subject, timetable.room\_numb, timetable.start\_time, teacher.name FROM timetable, teacher WHERE day='Понедельник' AND chet=%s AND timetable.subject=teacher.subject ORDER BY timetable.start\_time ASC;", parity)

records = list(cursor.fetchall())

day = 'Понедельник.'

for i in range(len(records)):

cl = str(records[i])

cl = cl.replace("(","")

cl = cl.replace(")", "")

cl = cl.replace("'", "")

day += '\n' + cl + '.'

day += '\n' + '\n'

cursor.execute("SELECT timetable.subject, timetable.room\_numb, timetable.start\_time, teacher.name FROM timetable, teacher WHERE day='Вторник' AND chet=%s AND timetable.subject=teacher.subject ORDER BY timetable.start\_time ASC;",parity)

records = list(cursor.fetchall())

day += 'Вторник.'

for i in range(len(records)):

cl = str(records[i])

cl = cl.replace("(", "")

cl = cl.replace(")", "")

cl = cl.replace("'", "")

day += '\n' + cl + '.'

day += '\n' + '\n'

cursor.execute(

"SELECT timetable.subject, timetable.room\_numb, timetable.start\_time, teacher.name FROM timetable, teacher WHERE day='Среда' AND chet=%s AND timetable.subject=teacher.subject ORDER BY timetable.start\_time ASC;",

parity)

records = list(cursor.fetchall())

day += 'Среда.'

for i in range(len(records)):

cl = str(records[i])

cl = cl.replace("(", "")

cl = cl.replace(")", "")

cl = cl.replace("'", "")

day += '\n' + cl + '.'

day += '\n' + '\n'

cursor.execute(

"SELECT timetable.subject, timetable.room\_numb, timetable.start\_time, teacher.name FROM timetable, teacher WHERE day='Четверг' AND chet=%s AND timetable.subject=teacher.subject ORDER BY timetable.start\_time ASC;",

parity)

records = list(cursor.fetchall())

day += 'Четверг.'

for i in range(len(records)):

cl = str(records[i])

cl = cl.replace("(", "")

cl = cl.replace(")", "")

cl = cl.replace("'", "")

day += '\n' + cl + '.'

day += '\n' + '\n'

cursor.execute(

"SELECT timetable.subject, timetable.room\_numb, timetable.start\_time, teacher.name FROM timetable, teacher WHERE day='Пятница' AND chet=%s AND timetable.subject=teacher.subject ORDER BY timetable.start\_time ASC;",

parity)

records = list(cursor.fetchall())

day += 'Пятница.'

for i in range(len(records)):

cl = str(records[i])

cl = cl.replace("(", "")

cl = cl.replace(")", "")

cl = cl.replace("'", "")

day += '\n' + cl + '.'

bot.send\_message(message.chat.id, day, reply\_markup=keyboard)

Для следующей недели мы берём parity1, который обратен обычной parity, так как следующая неделя будет другая по чётности, в остальном код такой же, как и для текущей недели.

@bot.message\_handler(commands=['Следующая'])

def n\_week(message):

keyboard = types.ReplyKeyboardMarkup()

keyboard.row("/Понедельник", "/Вторник", "/Среда", "/Четверг", "/Пятница", "/Текущая",

"/Следующая")

cursor.execute(

"SELECT timetable.subject, timetable.room\_numb, timetable.start\_time, teacher.name FROM timetable, teacher WHERE day='Понедельник' AND chet=%s AND timetable.subject=teacher.subject ORDER BY timetable.start\_time ASC;",

parity1)

records = list(cursor.fetchall())

day = 'Понедельник.'

for i in range(len(records)):

cl = str(records[i])

cl = cl.replace("(", "")

cl = cl.replace(")", "")

cl = cl.replace("'", "")

day += '\n' + cl + '.'

day += '\n' + '\n'

cursor.execute(

"SELECT timetable.subject, timetable.room\_numb, timetable.start\_time, teacher.name FROM timetable, teacher WHERE day='Вторник' AND chet=%s AND timetable.subject=teacher.subject ORDER BY timetable.start\_time ASC;",

parity1)

records = list(cursor.fetchall())

day += 'Вторник.'

for i in range(len(records)):

cl = str(records[i])

cl = cl.replace("(", "")

cl = cl.replace(")", "")

cl = cl.replace("'", "")

day += '\n' + cl + '.'

day += '\n' + '\n'

cursor.execute(

"SELECT timetable.subject, timetable.room\_numb, timetable.start\_time, teacher.name FROM timetable, teacher WHERE day='Среда' AND chet=%s AND timetable.subject=teacher.subject ORDER BY timetable.start\_time ASC;",

parity1)

records = list(cursor.fetchall())

day += 'Среда.'

for i in range(len(records)):

cl = str(records[i])

cl = cl.replace("(", "")

cl = cl.replace(")", "")

cl = cl.replace("'", "")

day += '\n' + cl + '.'

day += '\n' + '\n'

cursor.execute(

"SELECT timetable.subject, timetable.room\_numb, timetable.start\_time, teacher.name FROM timetable, teacher WHERE day='Четверг' AND chet=%s AND timetable.subject=teacher.subject ORDER BY timetable.start\_time ASC;",

parity1)

records = list(cursor.fetchall())

day += 'Четверг.'

for i in range(len(records)):

cl = str(records[i])

cl = cl.replace("(", "")

cl = cl.replace(")", "")

cl = cl.replace("'", "")

day += '\n' + cl + '.'

day += '\n' + '\n'

cursor.execute(

"SELECT timetable.subject, timetable.room\_numb, timetable.start\_time, teacher.name FROM timetable, teacher WHERE day='Пятница' AND chet=%s AND timetable.subject=teacher.subject ORDER BY timetable.start\_time ASC;",

parity1)

records = list(cursor.fetchall())

day += 'Пятница.'

for i in range(len(records)):

cl = str(records[i])

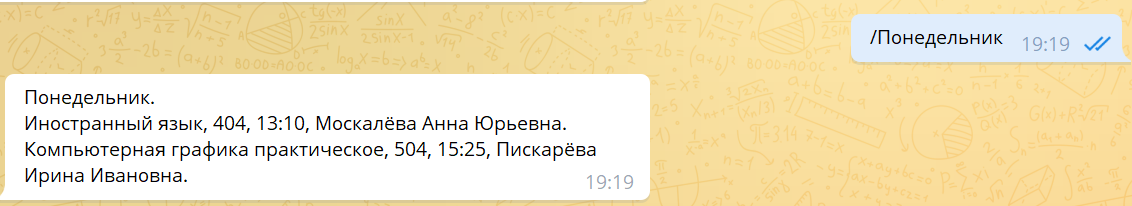
cl = cl.replace("(", "")

cl = cl.replace(")", "")

cl = cl.replace("'", "")

day += '\n' + cl + '.'

bot.send\_message(message.chat.id, (day), reply\_markup=keyboard)





**Вывод:** В ходе проделанной работы мы научились составлять Базы данных, взаимодействовать между несколькими её таблицами, а также отработали взаимодействие Телеграм-бота и БД Postgres