МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение   
высшего образования «Самарский национальный исследовательский университет   
имени академика С.П. Королева»

(Самарский университет)

Институт информатики и кибернетики

Кафедра технической кибернетики

**ОТЧЕТ**

По лабораторной работе №1

**Проектирование приложения**

Выполнили студенты

группы 6301-010302D

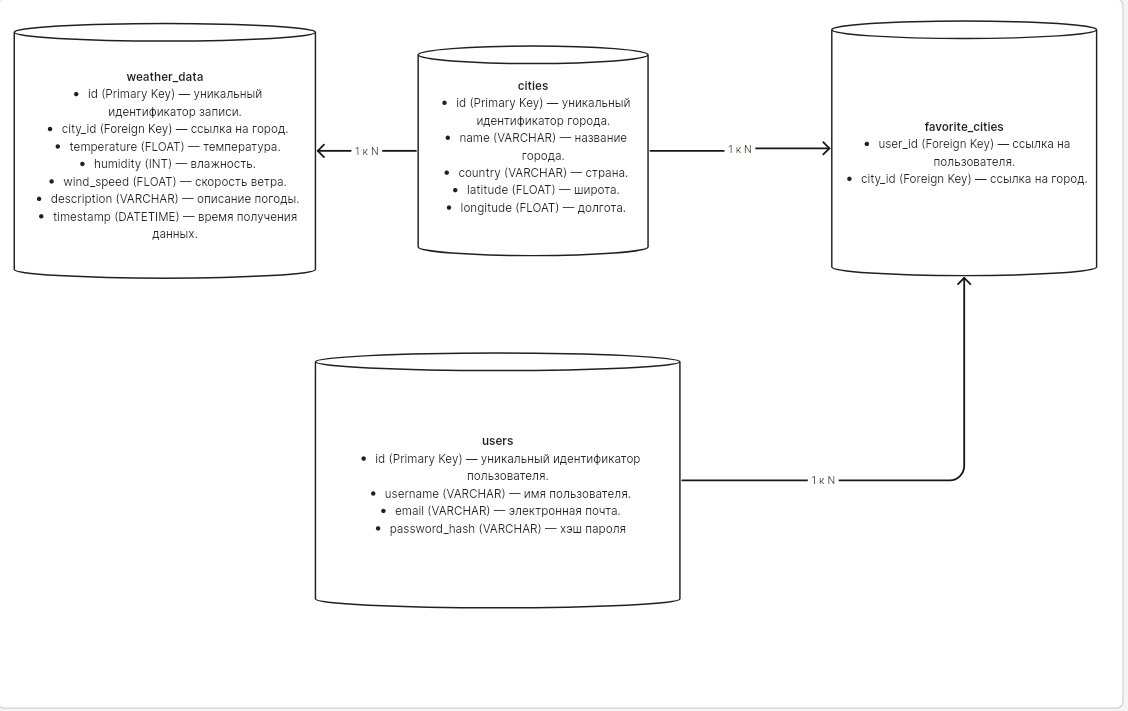
Астафьева А.В.

и 6304-010302D

Серебряков В.С.

САМАРА 2025

# Описание БД



# Скрипты для демонстрации работы с данными

from backend.src.run import db  
from werkzeug.security import generate\_password\_hash  
  
  
class Users(db.Model):  
 \_\_tablename\_\_ = 'users'  
 id = db.Column(db.Integer, primary\_key=True)  
 username = db.Column(db.String(64), unique=True, nullable=False)  
 email = db.Column(db.String(64), unique=True, nullable=False)  
 password\_hash = db.Column(db.String(256), nullable=False)  
  
 def set\_password(self, password):  
 self.password\_hash = generate\_password\_hash(password)  
  
  
class Cities(db.Model):  
 \_\_tablename\_\_ = 'cities'  
 id = db.Column(db.Integer, primary\_key=True)  
 name = db.Column(db.String(64), nullable=False)  
 country = db.Column(db.String(64), nullable=False)  
 latitude = db.Column(db.Float, nullable=False)  
 longitude = db.Column(db.Float, nullable=False)  
  
  
class WeatherData(db.Model):  
 \_\_tablename\_\_ = 'weather\_data'  
 id = db.Column(db.Integer, primary\_key=True)  
 city\_id = db.Column(db.Integer, db.ForeignKey('cities.id'), nullable=False)  
 temperature = db.Column(db.Float, nullable=False)  
 humidity = db.Column(db.Integer, nullable=False)  
 wind\_speed = db.Column(db.Float, nullable=False)  
 description = db.Column(db.String(255), nullable=False)  
 timestamp = db.Column(db.DateTime, nullable=False)  
  
  
class FavoriteCities(db.Model):  
 \_\_tablename\_\_ = 'favorite\_cities'  
 user\_id = db.Column(db.Integer, db.ForeignKey('users.id'), primary\_key=True)  
 city\_id = db.Column(db.Integer, db.ForeignKey('cities.id'), primary\_key=True)  
  
  
def create\_test\_user():  
 *"""Создает тестового пользователя"""* user = Users(username='test\_user2', email='test2@example.com')  
 user.set\_password('password123')  
 db.session.add(user)  
 db.session.flush()  
 return user  
  
  
def create\_test\_city():  
 *"""Создает тестовый город"""* city = Cities(  
 name='Moscow',  
 country='Russia',  
 latitude=55.7558,  
 longitude=37.6173  
 )  
 db.session.add(city)  
 db.session.flush()  
 return city  
  
  
def create\_weather\_data(city\_id):  
 *"""Создает тестовые данные о погоде"""* weather = WeatherData(  
 city\_id=city\_id,  
 temperature=25.5,  
 humidity=60,  
 wind\_speed=5.3,  
 description='Sunny',  
 timestamp='2023-10-01 12:00:00'  
 )  
 db.session.add(weather)  
 return weather  
  
  
def create\_favorite\_city(user\_id, city\_id):  
 *"""Добавляет город в избранное"""* favorite = FavoriteCities(user\_id=user\_id, city\_id=city\_id)  
 db.session.add(favorite)  
 return favorite  
  
  
def initialize\_data():  
 *"""Инициализирует тестовые данные"""* try:  
 # Создаем пользователя (если его нет)  
 user = create\_test\_user()  
 city = create\_test\_city()  
  
 if user and city:  
 create\_weather\_data(city.id)  
 create\_favorite\_city(user.id, city.id)  
 db.session.commit()  
 print("Тестовые данные успешно созданы")  
 else:  
 print("Тестовые данные уже существуют")  
  
 except Exception as e:  
 db.session.rollback()  
 print(f"Ошибка при создании тестовых данных: {str(e)}")  
 raise