*Лабораторная работа №1*

**Реляционные данные. Исследовательский анализ данных.**

**Построение визуализаций данных OLAP**

**Цель:** провести первичный анализ реляционной базы данных с помощью python, pandas и библиотеки для подключения к БД.

**Содержание**

[Задание для самостоятельного выполнения 1](#_Toc187660413)

[Требования к проекту 1](#_Toc187660414)

[Сложность выполнения 2](#_Toc187660415)

[Варианты 3](#_Toc187660416)

[Отчёт 5](#_Toc187660417)

[Формат выполнения задания 5](#_Toc187660418)

[Справка. Работа с языком SQL 5](#_Toc187660419)

# Задание для самостоятельного выполнения

1. Создать базу из файлов или скриптов согласно варианту. Сложность программы выбрать из представленных ниже.
2. Подключиться к базе данных из python.
3. Сделать описание данных. Из каких таблиц и полей состоят данные таблиц? Какие из данных являются признаками? К какому типу данных и к какой шкале относятся признаки?
4. Одномерный анализ. Построить гистограммы распределения количественных признаков, которые важны для задачи. Сделать вывод после построения. Какое распределение для каждого из признаков? Почему, по вашему мнению, признаки важны для задачи?
5. Многомерный анализ. Построить графики из 3-4 признаков. Выбрать категориальные (номинальные, порядковые или бинарные) признаки и количественные. Что получилось на каждом графике? Почему, по вашему мнению, признаки важны для задачи?

# Требования к проекту

* Проект реализован локально и выгружен на github или прикреплен в архиве к заданию в elearning
* Проект состоит из файлов программы (py или ipynb), выгрузки (dump) базы данных и файла ReadMe.md
* Проект реализован на python с применением библиотеки pandas.
* README.md должен содержать:
* Название вашего приложения
* Описание
* Инструкции по запуску проекта и базы данных

# Сложность выполнения

**Сложность: Rare**

* Реализовать Jupiter Notebook или консольное приложение для выполнения задания.
* В качестве базы данных выбрать sqlite3. Создать базу из файлов или скриптов согласно варианту.
* Подключиться к базе данных из python.
* Сделать описание данных и выводы по заданию.
* Соединить признаки в 1 таблицу pandas для анализа
* Одномерный анализ. Построить 2 гистограммы распределения количественных признаков, которые важны для задачи и сделать их описание по заданию.
* Многомерный анализ. Построить хотя бы 1 график из 3-4 признаков и сделать его описание по заданию.

**Сложность: Medium**

* Реализовать Jupiter Notebook или консольное приложение для выполнения задания.
* В качестве базы данных выбрать PostgreSQL. Создать базу из файлов или скриптов согласно варианту.
* Подключиться к базе данных из python.
* Сделать описание данных и выводы по заданию.
* Соединить признаки в 1 таблицу pandas для анализа.
* Одномерный анализ. Построить 2 гистограммы распределения количественных признаков, которые важны для задачи и сделать их описание по заданию.
* Многомерный анализ. Построить 2 графика из 3-4 признаков и сделать их описание по заданию.

**Сложность: Well-done**

* Реализовать приложение с визуальным интерфейсом на pyside6 или аналогах для выполнения задания. Приложение должно подключаться к выбранной базе данных и давать возможность выбора признаков для общей таблицы построения визуализаций.
* В качестве базы данных выбрать PostgreSQL. Создать базу из файлов или скриптов согласно варианту.
* Подключиться к базе данных из python.
* Сделать описание данных и выводы по заданию.
* Соединить признаки в 1 таблицу pandas для анализа.
* Одномерный анализ. Построить 2 гистограммы распределения количественных признаков, которые важны для задачи и сделать их описание по заданию.
* Многомерный анализ. Построить 2 графика из 3-4 признаков и сделать их описание по заданию.

# Варианты

|  |  |  |
| --- | --- | --- |
| № | Вариант | Ссылка на kaggle |
| 1 | SQL Murder Mystery Database | <https://www.kaggle.com/datasets/johnp47/sql-murder-mystery-database> |
| 2 | SQLite Sakila Sample Database | <https://www.kaggle.com/datasets/atanaskanev/sqlite-sakila-sample-database?select=SQLite3+Sakila+Sample+Database+ERD.png> |
| 3 | Bike Store Relational Database | SQL | <https://www.kaggle.com/datasets/dillonmyrick/bike-store-sample-database> |
| 4 | Indian Premier League SQLite Database | <https://www.kaggle.com/datasets/harsha547/ipldatabase> |
| 5 | Housing - SQL Project | <https://www.kaggle.com/datasets/bvanntruong/housing-sql-project> |
| 6 | NBA Database | <https://www.kaggle.com/datasets/wyattowalsh/basketball> |
| 7 | Formula 1 Race Data (SQLite) | <https://www.kaggle.com/datasets/davidcochran/formula-1-race-data-sqlite> |
| 8 | League of Legends Ranked Matches | <https://www.kaggle.com/datasets/paololol/league-of-legends-ranked-matches> |
| 9 | Sample databases for the SQL Server | <https://www.kaggle.com/datasets/emrahaydemr/sample-databases-for-the-sql-server-course> |
| 10 | UEFA Champions League 2016-2022 Data | <https://www.kaggle.com/datasets/cbxkgl/uefa-champions-league-2016-2022-data> |
| 11 | Kensho Derived Wikimedia Dataset | <https://www.kaggle.com/datasets/kenshoresearch/kensho-derived-wikimedia-data> |
| 12 | SQL Murder Mystery Database | <https://www.kaggle.com/datasets/johnp47/sql-murder-mystery-database> |
| 13 | SQLite Sakila Sample Database | <https://www.kaggle.com/datasets/atanaskanev/sqlite-sakila-sample-database?select=SQLite3+Sakila+Sample+Database+ERD.png> |
| 14 | Bike Store Relational Database | SQL | <https://www.kaggle.com/datasets/dillonmyrick/bike-store-sample-database> |
| 15 | Indian Premier League SQLite Database | <https://www.kaggle.com/datasets/harsha547/ipldatabase> |
| 16 | Housing - SQL Project | <https://www.kaggle.com/datasets/bvanntruong/housing-sql-project> |
| 17 | NBA Database | <https://www.kaggle.com/datasets/wyattowalsh/basketball> |
| 18 | Formula 1 Race Data (SQLite) | <https://www.kaggle.com/datasets/davidcochran/formula-1-race-data-sqlite> |
| 19 | League of Legends Ranked Matches | <https://www.kaggle.com/datasets/paololol/league-of-legends-ranked-matches> |
| 20 | Sample databases for the SQL Server | <https://www.kaggle.com/datasets/emrahaydemr/sample-databases-for-the-sql-server-course> |
| 21 | UEFA Champions League 2016-2022 Data | <https://www.kaggle.com/datasets/cbxkgl/uefa-champions-league-2016-2022-data> |
| 22 | Kensho Derived Wikimedia Dataset | <https://www.kaggle.com/datasets/kenshoresearch/kensho-derived-wikimedia-data> |
| 23 | Housing - SQL Project | <https://www.kaggle.com/datasets/bvanntruong/housing-sql-project> |

# Отчёт

Отчёт должен содержать:

* Титульный лист
* Формулировка задач, описание условий
* Вариант и условия задач. Указать какой сложности выполняется задание.
* Ссылка на репозиторий с программной реализацией
* Описание проделанной работы
* Краткий вывод по работе. Описание реализованной программы и её функций.
* Ссылки на используемые материалы. Документация
* Листинг кода

# Формат выполнения задания

1. Код в репозитории github | gitlab | bitbucket. Для отчета по работе выполнить задание в файле.py или .ipynb.

Желательна реализация в файлах py. Сохранить задачи (сделать коммиты для каждой) в локальном git и опубликовать в удаленном репозитории.

2. Отчет в форматах google docs, docx или tex (+ pdf).

3. База данных в формате db или dump базы данных.

# Справка. Работа с языком SQL

Перед выполнением задания необходимо:

1. Установить базу данных PostgreSQL или MySQL Community (под администратором системы):

* для Windows (инсталлятор и документация) : <https://postgrespro.ru/windows>
* для Linux (инсталлятор и документация) : <https://www.postgresql.org/download/>
* для Windows (инсталлятор и документация) : <https://dev.mysql.com/downloads/installer/>
* для Linux (инсталлятор и документация) : <https://dev.mysql.com/doc/mysql-installation-excerpt/8.0/en/linux-installation.html>

После установки и настройки загрузить библиотеки psycopg2, pymysql, sqlalchemy.

Упрощенная версия работы :

* Для работы с данными выбрать sqlite3 вместо postgreSQL. Библиотека встроена в python, дополнительных установок не нужно. Руководство (русский язык) по URL: <https://pythonru.com/osnovy/sqlite-v-python>
* Руководство (англ.) по URL: <https://docs.python.org/3/library/sqlite3.html>

Для работы с базой используется модуль psycopg2 для Python3. SqlAlchemy

устанавливается и настраивается дополнительно (ORM для python3).

3. Для работы с таблицами и входными файлами использовать также pandas,

numpy.