

## Финальный проект

### Первый блок:

Ваше первое задание заключается в том, чтобы правильно написать запросы к базе данных **sakila**.  
Файл для загрузки базы данных sakila в приложении «**sakila\_db.sql**», вы открываете файл и запускаете все запросы.

Также задачи на написание запросов в приложении «**sql\_questions.sql**», открываете файл и под каждой задачей напишите запрос в соответствии с требованиями.

Описание таблиц лежит тут: <https://dev.mysql.com/doc/sakila/en/sakila-structure-tables.html>

### Второй блок:

1. Создайте базу данных «final\_project» (в MySQL, там же где мы работали с employees, sakila), а в ней таблицу «**uk\_bank**» с такой же структурой как в CSV файле «**UK-Bank-Customers.csv**» в приложении. Это данные искусственного банка в Великобритании.
2. Импортируйте CSV файл в таблицу **uk\_bank** (проверьте все ли строки загрузились правильно).
3. Теперь создайте отчет в Power BI Desktop, где источником данных будет **uk\_bank**.
4. В данном отчете отобразите всю необходимую информацию по клиентам данного банка, так чтобы видеть полную картину по всем клиентам. Средний возраст, мин/макс баланс на счету, соотношение пола и другие любые показатели, которые были бы полезны банку. Максимум 1 лист, аккуратное оформление, подписи для каждого визуального элемента.

### Третий блок:

1. Откройте уже спарсенный и готовый для работы датасет «**data.csv**». Это спарсенные данные с сайта **head hunter**. URL: <http://hh.kz/search/resume>.
2. Мы будем работать только со столбцами ['job\_title', 'age', 'salary', 'exp. period']. Сохраните только их.
3. Создайте переменную X, которая содержит данные датасета без столбца 'job\_title'.
4. Создайте переменную y, которая содержит только данные столбца 'job\_title'.
5. Разделите датасеты X и y на тренировочные и тестовые, где train\_size будет равен 80%. На забудьте использовать функцию stratify.
6. Обучите сначала модель с помощью алгоритма GaussianNB.
7. Протестируйте обученную модель на тестовых данных, и выведите на экран точность модели.
8. Используйте перекрестную проверку модели. Разбейте данные на десять групп и по очереди используйте каждую из них для оценки обучения модели. Выведите на экран среднюю точность модели.
9. Затем обучите модель с помощью алгоритма KNeighborsClassifier с количеством n\_neighbors равным 3.
10. Протестируйте обученную модель на тестовых данных, и выведите на экран точность модели.
11. Используйте перекрестную проверку модели. Разбейте данные на десять групп и по очереди используйте каждую из них для оценки обучения модели. Выведите на экран среднюю точность модели.

### Итого:

По завершению всех блоков, вы отправляете одним **zip** файлом на почту [academica.hw@gmail.com](mailto:academica.hw@gmail.com):

- SQL файл с вашими запросами
- PBIX файл отчета, а также PDF версию отчета
- IPYNB файл с вашим кодом для третьего блока