**Задание для пересдачи по курсу «ИТ для финансистов»**

**Трек: Риски**

**Задание 1**

1. SQL запросами загрузить данные из xls файла, содержащего данные о рейтинговых действиях. (**1 балл**)
2. Соединить Python с базой данных по протоколу ODBC. (**1 балл**)
3. Запросом(-ами) SQL из Python запросить из базы данные о рейтингах:
   1. актуальных на/присвоенных с 01.07.2010 г.
   2. присвоенных НЕ агентствам АМБэст Компани и Интерфакс;
   3. присвоенных банкам;
   4. являющихся рейтингами компании (НЕ финансового инструмента);
   5. присвоены в национальной шкале или не имеют указатель шкалы.

(**1 балл**)

1. В Python закодировать поле grade, заменив буквенные значения грейдов на числовые, а также создать таблицы-словари для расшифровки; кодировка должна быть монотонной, больше число должно соответствовать более низкому кредитному качеству, число 0 должно соответствовать наивысшему кредитному качеству.   
   (**1 балл**)
2. Запросом из Python отправить перекодированную таблицу рейтингов в базу данных, а также создать в базе таблицы-словари для разкодировки, связанные с основной таблицей. (**1 балл**)

**Решение**

**Устанавливаем необходимое программное обеспечение:**

* Бесплатный выпуск Microsoft SQL Server [тут](https://go.microsoft.com/fwlink/?linkid=853017) (далее – MS SQL)
* СУБД [тут](https://go.microsoft.com/fwlink/?linkid=2043154) (далее – SSMS)
* Python [тут](https://www.python.org/ftp/python/3.7.2/python-3.7.2.exe)
* Дистрибутив для Python Anaconda [тут](https://repo.continuum.io/archive/Anaconda3-2018.12-Windows-x86_64.exe)
* PYODBC драйвер для БД [тут](https://www.microsoft.com/en-us/download/confirmation.aspx?id=56567)
* Выполняем стандартную установку, следуя указаниям установщиков
* Запускаем SSMS.
  + Подключаемся к БД с использованием *проверки подлинности Windows*
  + Переходим *Безопасность – Имена для входа – Создать имя для входа…*
  + *Выбираем проверку подлинности SQL Server*
  + *Задаем имя для входа и пароль*
  + Во вкладке *Роли сервера* отмечаем *sysadmin*
  + Подключаемся к БД с использованием *проверки подлинности SQL Server*
    - Возможно появление диалогового окна о смене пароля
    - Производим изменение пароля на новый
  + Переходим Базы данных – Создать базу данных…
  + Вводим имя БД и жмем «ОК»
  + Переходим Таблицы – Таблица
    - Вводим *имя столбца, тип данных и NULL/ NOT NULL*
  + В БД в таблице *dbo.Table\_1* выбираем изменить первые 200 строк
    - Вводим любые данные (необходимо для выполнения подключения)
* Запускаем Anaconda – Spyder.
  + Запускаем скрипт *TestConnection.py*
  + Результатом выполнения скрипта должно быть получение всех строк таблицы *dbo.Table\_1,* которую мы создали вручную с помощью СУБД
* Импорт данных из XLSX файлов.
  + Создаем на рабочем столе папку *Scripts*
  + Скачиваем необходимый файл *(Рейтинги…) и сохраняем его в эту папку*
  + Нам понадобятся:
    - Из файла *Рейтинги* *– листы ratings, base*
  + Сохраняем каждый лист из документов в отдельную книгу
    - *Лист – переместить или скопировать…*
    - Вводим имя книги, такое, как и было у листа и **ОБЯЗАТЕЛЬНО** прописываем имя листа *Sheet1*
  + Перемещаем созданные книги в папку *Scripts*
    - *Rating – файл Рейтинги, лист ratings*
    - *Base – файл Рейтинги, лист base*
* Для правильного импорта данных необходимо выбрать тип полей для таблиц БД
  + Rating:
    - Все столбцы (большинство) представляют собой строку
      * *В скрипте укажем для всех столбцов тип – NVARCHAR(50)*
      * *NVARCHAR – строка переменной длины с поддержкой Юникода*
      * *Размер в 50 символов определен опытным путем*
      * *ID – указываем тип данных INT*
      * *При импорте поля «For\_Instrument», «For\_Company» изменят свое значение с ИСТИНА/ЛОЖЬ на 1/0 соответственно. Без указания типа BOOL при импорте. В скрипте используется тип NVARCHAR*
  + Base:
    - Все столбцы (большинство) представляют собой строку
      * *В скрипте укажем для всех столбцов тип – NVARCHAR(50)*
      * *NVARCHAR – строка переменной длины с поддержкой Юникода*
      * *Размер в 50 символов определен опытным путем*
      * *Столбец «Date\_str» можно указать тип DATE, однако импорт завершится ошибкой, потому что БД поддерживает определенный формат ГГГГ-ММ-ДД чч:мм:сс, что отличается от формата даты в XLSX. Оставляем тип NVARCHAR*
      * *Столбцы «Okpo», «Ogrn», «Inn» имеют тип FLOAT. Если указывать тип NVARCHAR, то при импорте данные в этих столбцах отобразятся в экспоненциальной форме*
* *Создадим новую базу данных для импорта*
  + - Запускаем скрипт *CreateBase.py*
    - В качестве проверки скрипт выведет список всех БД
  + Импортируем данные из соотношения 1 лист (книга) – 1 таблица
    - Запускаем скрипт *Import\_ratings.py*
    - Запускаем скрипт *Import\_base.py*
    - В качестве проверки скрипт будет возвращать число импортированных строк. Если их количество больше 0, то программа выдаст сообщение: *«Import successful»*. В случае каких-либо ошибок пользователь увидит ошибку *«DataError»* с ее подробным описанием
* *В скриптах используется единая структура:*
  + *единое подключение к БД*
  + *создание таблицы с определенными полями (столбцами) и их форматом*
  + *открытие и чтение XLSX файла*
  + *изменение имен столбцов, если в них есть пробел*
  + *присваивание каждому столбцу своей переменной*
  + *запуск цикла построчного чтения в документе и перенос данных в БД*
* Выполнение запросов:
  + Запускаем скрипт *Query\_a.py*

SELECT Ag\_name, Rat\_name, Grade, Outlook, Change, Date\_str, Ent\_name, Okpo, Ogrn, Inn, Finst, Ag\_id, Rat\_Sctr, Rat\_type, Horizon, Scale\_Typer, Currency, Backed\_flag INTO [dbo].[Query\_a] FROM [dbo].[Base] WHERE Date\_str != '01.07.2010' AND Grade !='Приостановлен' AND Grade !='Снят'

**Примечание!**

**Открывая исходный файл с данными, пользователь видит дату в поле «Date\_str»**

**Excel «видит» строку «ДД.ММ.ГГГГ» «Date\_str»**

**База данных «видит» числовой формат даты - количество дней с 01.01.1900**

**Другими словами, при импорте данных дата преобразуется в числовой формат и сохранится в таком виде.**

**Если нужно сделать дату в виде даты, то нужно преобразовывать «Date\_str» в XLSX документе и приводить к виду ГГГГ-ММ-ДД чч:мм:сс**

**При выполнении первого запроса будут выбраны все даты меньше 01.07.2010 или же меньше 40360**

* + Запускаем скрипт *Query\_b.py*

SELECT Ag\_name, Rat\_name, Grade, Outlook, Change, Date\_str, Ent\_name, Okpo, Ogrn, Inn, Finst, Ag\_id, Rat\_Sctr, Rat\_type, Horizon, Scale\_Typer, Currency, Backed\_flag INTO [dbo].[Query\_b] FROM [dbo].[Base] WHERE Ag\_name != 'АМБест Компани' AND Ag\_name != 'Интерфакс'

* + Запускаем скрипт *Query\_c.py*

SELECT Ag\_name, Rat\_name, Grade, Outlook, Change, Date\_str, Ent\_name, Okpo, Ogrn, Inn, Finst, Ag\_id, Rat\_Sctr, Rat\_type, Horizon, Scale\_Typer, Currency, Backed\_flag INTO [dbo].[Query\_c] FROM [dbo].[Base] WHERE Rat\_sctr = 'B'

* + Запускаем скрипт *Query\_d.py*

SELECT ID, Code\_name, Agency, Full\_Name\_Rus, Full\_Name\_Eng, For\_Instrument, For\_Company INTO [dbo].[Query\_d] FROM [dbo].[Ratings] WHERE For\_Company = '1'

* + Запускаем скрипт *Query\_e.py*

SELECT Ag\_name, Rat\_name, Grade, Outlook, Change, Date\_str, Ent\_name, Okpo, Ogrn, Inn, Finst, Ag\_id, Rat\_Sctr, Rat\_type, Horizon, Scale\_Typer, Currency, Backed\_flag INTO [dbo].[Query\_e] FROM [dbo].[Base] WHERE Scale\_typer != 'Isc'

* + - Каждый скрипт выполняет следующие операции (+ проверка):
      * Создает отдельную таблицу в БД
      * Выполняет требуемый в задании запрос

**Задание 2**

В Python написать код, который:

1. По данным, запрошенным в п. 3 задания 1, и заданной пользователем дате определяет количество актуальных на заданную дату рейтингов, присвоенных каждым агентством, и строит по этим данным pie chart, на котором указаны дата, названия агентств, структура присвоенных рейтингов по агентствам в шт. и %.
2. По данным, запрошенным в п. 3 задания 1, и заданным пользователем [дате, рейтинговому агентству] определяет распределение актуальных на заданную дату рейтингов заданного агентства по грейдам и строит bar plot, на котором указаны дата, агентство и буквенные наименование грейдов.
3. По данным, запрошенным в п. 3 задания 1, и заданным пользователем [дате, горизонте и рейтинговому агентству] строит матрицу переходов и представляет из в виде таблицы вида:

**Решение**

1. Запускаем скрипт *Pie\_Chart.py*
   * *Пользователю необходимо ввести дату в числовом формате (40932, 42339 и т.д.)*
   * *Скрипт делает выборку данных по указанной дате, и перемещает их в отдельную таблицу.*
   * *Создает дополнительную таблицу с числом повторений*
   * *Строит Pie-chart диаграмму*

*Результаты работы скрипта можно отследить в самой БД (таблицы) и диаграмме*