**Домашняя работа 1.**

**Речмединой Светланы**

1. Краткая инструкция:

a. подготовительные действия для работы с данными (где и в каком виде хранить данные, какой софт установить и т.д.) - установить необходимо PostgreSQL, данные брались из таблиц, чей формат был изменен на csv. Также в колонке с названием date был изменен формат с числового на “дата”.

b. краткое содержания процедур, заложенных в скриптах/запросах, а также порядок их запуска;

c. описание ожидаемых результатов каждого этапа.

2. Скрипты с запросами в формате sql

**1 задание:** **Запросами SQL загрузить1 в БД данные из файлов ratings\_task.xlsx, credit\_events\_task.xls и scale\_EXP\_task.csv. Выбрать форматы полей, создать таблицы (наложить разумные ограничения на поля), импортировать данные. (1 балл)**

**1 таблица**

#Сначала удаляем предыдущую таблицу, которая получалась из-за работы над кодом в процессе его написания. Затем создаем новую.

DROP TABLE if exists public.actions

;

CREATE TABLE public.actions

(

#Устанавливаем первичный ключ, это понадобится в следующих заданиях

Rating\_record\_id serial primary key,

#выбираем форматы полей, соответствующие содержанию и назначению данных, также ставим ограничение по размеру. Если необходимо пишем, что в ячейке не может отсутствовать значение (NOT NULL).

rat\_id varchar(25) NOT NULL,

grade varchar(25) NOT NULL,

outlook varchar(25),

change varchar(25) NOT NULL,

"date" char(15) NOT NULL,

ent\_name varchar(200) NOT NULL,

okpo varchar(10) NOT NULL,

ogrn varchar(15),

inn varchar(10),

finst varchar(10),

agency\_id varchar(10) NOT NULL,

rat\_industry varchar(20),

rat\_type varchar(10) NOT NULL,

horizon varchar(10),

scale\_typer varchar(50),

currency varchar(10),

backed\_flag varchar(30)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.actions

OWNER to postgres;

#устанавливаем разумные ограничения на поля, чтобы при использовании неподходящего по логике таблицы значения данные не вводились, данные ограничения взяты из описания данных в предложенных для работы файлах.

ALTER TABLE public.actions ADD CONSTRAINT agency\_id\_con CHECK (agency\_id='AKM' OR agency\_id ='EXP' OR agency\_id ='FCH' OR agency\_id ='MDS' OR agency\_id ='NRA' OR agency\_id ='RUS' OR agency\_id ='SNP');

ALTER TABLE public.actions ADD CONSTRAINT horizon\_con CHECK (horizon='LT' OR horizon='ShT');

ALTER TABLE public.actions ADD CONSTRAINT scale\_typer\_con CHECK (scale\_typer='Isc' OR position('sc' in scale\_typer) = char\_length(scale\_typer) - 1);

ALTER TABLE public.actions ADD CONSTRAINT currency\_con CHECK (currency='Icur' OR currency='Ncur');

#Импортируем данные в таблицу, учитывая правильный пусть файла, его формат и необходимость в первой строке выводить данные, а заголовок помещать отдельно primary key)

copy public.actions (rat\_id, grade, outlook, change, "date", ent\_name, okpo, ogrn, inn, finst, agency\_id, rat\_industry, rat\_type, horizon, scale\_typer, currency, backed\_flag) FROM E'C:\\Users\\Public\\Data\\Data\\Risks\\ratings\_task.csv' (format csv, delimiter ';', header True)

**Ожидаемые результаты:** ожидается получить описанную выше таблицу с 38252 значениями.

**2 таблица**

#Описание процесса аналогично пункту «а».

DROP TABLE if exists public.credit\_events\_task

;

CREATE TABLE public.credit\_events\_task

(

"inn" varchar(10) NOT NULL,

"date" date NOT NULL,

event varchar(3) NOT NULL

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.credit\_events\_task

OWNER to postgres;

ALTER TABLE public.credit\_events\_task ADD CONSTRAINT event\_con CHECK (event='dft' OR event='lqd');

copy public.credit\_events\_task FROM 'C:\\Users\\Public\\Data\\Data\\Risks\\credit\_events\_task.csv'(format csv, delimiter ';', header True)

**Ожидаемые результаты:** ожидается получить описанную выше таблицу с 188 значениями.

**3 таблица**

#Описание процесса аналогично пункту «а», однако в конце появляется следующий код: encoding 'WIN 1251' – он помогает работать с кириллицей в ячейках таблицы.

DROP TABLE if exists public.scale\_EXP\_task

;

CREATE TABLE public.scale\_EXP\_task

(

grade varchar(15) NOT NULL UNIQUE,

grade\_id integer NOT NULL PRIMARY KEY

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.scale\_EXP\_task

OWNER to postgres;

copy public.scale\_EXP\_task FROM 'C:\\Users\\Public\\Data\\Data\\Risks\\scale\_EXP\_task.csv'(format csv, delimiter ';', header True,encoding 'WIN 1251')

**Ожидаемые результаты:** ожидается получить описанную выше таблицу с 13 значениями.

**2 номер: Запросами SQL вынести из таблицы actions информацию о рейтингах (тип, срочность и т.д.) и информацию о рейтингуемом лице (текстовое название и лишние числовые коды) в отдельные таблицы. (1 балл)**

#Удаляем данные от предыдущей версии в процессе работы и создаем новую таблицу

DROP TABLE if exists rat\_info

;

CREATE TABLE rat\_info AS

# Выбираем из таблицы actions информацию о рейтингах и выносим ее в отдельную таблицу, группируя записи по набору параметров

SELECT distinct agency\_id, rat\_industry, rat\_type, horizon, scale\_typer, currency, backed\_flag,rating\_record\_id, rat\_id

FROM public.actions GROUP BY agency\_id, rat\_industry, rat\_type, horizon, scale\_typer, currency, backed\_flag,rating\_record\_id, rat\_id;

#устанавливаем внешний ключ для дальнейшей работы

ALTER TABLE rat\_info ADD PRIMARY KEY (rating\_record\_id)

#Удаляем данные от предыдущей версии в процессе работы и создаем новую таблицу

DROP TABLE if exists info\_comp

;

CREATE TABLE info\_comp AS

# Выбираем из таблицы ratings\_task информацию о компаниях и выносим ее в отдельную таблицу, группируя записи по набору параметров

SELECT ent\_name, okpo, ogrn, inn, finst

FROM ratings\_task GROUP BY ent\_name, okpo, ogrn, inn, finst;

#устанавливаем внешний ключ для дальнейшей работы

ALTER TABLE info\_comp ADD PRIMARY KEY (ent\_name)

**Ожидаемые результаты:** ожидается получить описанные выше таблицы.

**3 номер: Связать таблицы в базе, задав внешние ключи, где это целесообразно. (1 балл)**

#Внешний ключ связывает поле подчинённой таблицы с уникальным полем главной таблицы  
По тому, как получены таблицы info\_comp и rat\_info, очевидно, что из таблицы actions можно удалить сообветствующие столбцы, оставив ссылки на новые таблицы. Для этих ссылок нужны внешние ключи, которые соединяют содержащиеся в actions поля ent\_name и rating\_record\_id с одноимёнными первичными ключами таблиц. Логично было бы, если бы первичным ключом rat\_info был rat\_id, но он не обладает уникальностью. Не удалось подобрать набор полей, который не содержал бы искусственного ключа rating\_record\_id и был бы уникальным для данной таблицы. Попытка создания внешнего ключа на поле grade, ссылающегося на таблицу scale\_EXP\_task, не работает, поскольку значения grade в таблице actions принимают не только значения, имеющие id и находящиеся в scale\_EXP\_task, но и многие другие. Таблица credit\_events, содержащая inn в роли идентификатора фирм, не может быть соединенинена с actions или info\_comp по внешнему ключу inn, потому что даже в info\_comp поле inn не уникально.

ALTER TABLE actions ADD CONSTRAINT entfk FOREIGN KEY (ent\_name) REFERENCES info\_comp (ent\_name);

ALTER TABLE actions ADD CONSTRAINT ratfk FOREIGN KEY (rating\_record\_id) REFERENCES rat\_info (rating\_record\_id);

ALTER TABLE actions ADD CONSTRAINT gradefk FOREIGN KEY (grade) REFERENCES scale\_EXP\_task (grade)

**4 номер:** **В сформированной базе данных составьте запрос, который выводит для выбранных вида рейтинга (rat\_id) и даты (date) в таблице actions, все актуальные рейтинги. Актуальным рейтингом признается значение, присвоенное или подтвержденное последним рейтинговым действием до исследуемой даты, при условии, что это действие было не снятием и не приостановкой рейтинга. Таблица должна содержать поля ent\_name, grade иassign\_date (дату присвоения актуального рейтинга). (2 балла)**  
#Делаем вложенный запрос, который для каждой пары значений rat\_id и date находит требуемые последние рейтинги. И если они актуальны (не закрыты), то присоединяет их к выборке.

SELECT distinct ent\_name, grade, assign\_date

FROM actions, (SELECT actions2.rating\_record\_id, actions2.ent\_name, actions2.grade, actions2."date" as assign\_date

from actions actions2

WHERE actions2."date" =

(SELECT max("date")

from actions actions2

WHERE actions2.rat\_id = actions.rat\_id

AND actions.date > actions2."date"))

as f

WHERE actions.rating\_record\_id = f.rating\_record\_id

AND change != 'снят'

AND change != 'приостановлен'