**1. Инструкция**

1. Для работы с данными необходимо установить СУБД (система управления базами данных). В данной работе будет использована PostgreSQL. Также для выполнения определенных команд потребуется терминал SQL Shell.

Для удобства в дальнейшей работе в исходной таблице ratings\_task.xlsx в столбце, содержащем информацию о дате действия, был изменен формат с числового на формат даты. Далее файлы credit\_events\_task.xls и ratings\_task.xlsx необходимо сохранить в формате .csv. Для дальнейшего импорта данных из таблиц credit\_events\_task.csv, ratings\_task.csv и scale\_EXP\_task.csv через PostgreSQL проверяем, что файлы находятся в папке с публичным доступом (~\User\Public) или перемещаем их туда.



*Скрипт 1.*

**CREATE TABLE public.credit\_events\_task**

**("inn" bigint NOT NULL,**

**"date" date NOT NULL,**

**"event" text NOT NULL)**

**WITH (OIDS = FALSE)**

**TABLESPACE pg\_default;**

**ALTER TABLE public.credit\_events\_task**

**OWNER to postgres;**

**copy public.credit\_events\_task FROM 'C:\Users\Public\credit\_events\_task.csv' DELIMITER ';' CSV HEADER;**

*Описание:*

1.1 На первом этапе создаем выше рассмотренные таблицы в базе данных и указываем нужный формат полей. Первая таблица credit\_events\_task – это таблица, содержащая информацию о кредитных событиях, связанных с компаниями. Она содержит 3 столбца, значения в которых не могут быть пропущены:

* inn - идентификационный номер налогоплательщика компании
* date - дата события
* event - кредитное событие компании (dft - дефолт, lqd - ликвидация компании без банкротства)

**CREATE TABLE public.credit\_events\_task**

**("inn" bigint NOT NULL,**

**"date" date NOT NULL,**

**"event" text NOT NULL)**

**WITH (OIDS = FALSE)**

**TABLESPACE pg\_default;**

1.2 Далее назначаем владельца таблицы

**ALTER TABLE public.credit\_events\_task**

**OWNER to postgres;**

1.3 После создания таблицы импортируем в нее данные. Необходимо убедиться, что полученный файл находится в папке с публичным доступом (~\User\Public). Используем файл в формате .csv для импорта данных в таблицу credit\_events\_task в БД.

**copy public.credit\_events\_task FROM 'C:\Users\Public\credit\_events\_task.csv' DELIMITER ';' CSV HEADER;**

По итогу мы получаем таблицу в БД, состоящую из 3 столбцов и 188 строк.

*Скрипт 2.*

**CREATE TABLE public.****scale\_exp\_task**

**("grade" text NOT NULL,**

**"grade\_id" smallint NOT NULL,**

**CONSTRAINT scale\_exp\_task\_pkey PRIMARY KEY ("grade\_id"))**

**WITH (OIDS = FALSE)**

**TABLESPACE pg\_default;**

**ALTER TABLE public.scale\_exp\_task**

**OWNER to postgres;**

**\! chcp 1251**

**\copy public.scale\_exp\_task FROM 'C:\Users\Public\scale\_EXP\_task.csv' DELIMITER ';' CSV HEADER;**

*Описание:*

2.1 Создаем вторую таблицу scale\_exp\_task – это таблица, содержащая информацию о числовой кодификации одной из рейтинговых шкал. Она содержит 2 столбца, значения в которых не могут быть пропущены:

* grade – рейтинговая шкала
* grade\_id – числовая кодификация.

Также назначаем столбец grade\_id в качестве первичного ключа.

**CREATE TABLE public.scale\_exp\_task**

**("grade" text NOT NULL,**

**"grade\_id" smallint NOT NULL,**

**CONSTRAINT scale\_exp\_task\_pkey PRIMARY KEY ("grade\_id"))**

**WITH (OIDS = FALSE)**

**TABLESPACE pg\_default;**

2.2 Далее назначаем владельца таблицы

**ALTER TABLE public.scale\_exp\_task**

**OWNER to postgres;**

2.3 После создания таблицы импортируем данные в таблицу scale\_exp\_task с помощью терминала SQL Shell, предварительно установив русскую раскладку. Используем файл в формате .csv для импорта данных в таблицу scale\_exp\_task в БД.

**\! chcp 1251**

**\copy public.scale\_exp\_task FROM 'C:\Users\Public\scale\_EXP\_task.csv' DELIMITER ';' CSV HEADER;**

По итогу мы получаем таблицу в БД, состоящую из 2 столбцов и 13 строк.

*Скрипт 3.*

**CREATE TABLE public.ratings\_task**

**("rat\_id" smallint NOT NULL,**

**"grade" text NOT NULL,**

**"outlook" text,**

**"change" text NOT NULL,**

**"date" date NOT NULL,**

**"ent\_name" text NOT NULL,**

**"okpo" bigint NOT NULL,**

**"ogrn" bigint,**

**"inn" bigint,**

**"finst" bigint,**

**"agency\_id" text NOT NULL,**

**"rat\_industry" text,**

**"rat\_type" text NOT NULL,**

**"horizon" text,**

**"scale\_typer" text,**

**"currency" text,**

**"backed\_flag" text)**

**WITH (OIDS = FALSE)**

**TABLESPACE pg\_default;**

**ALTER TABLE public.ratings\_task**

**OWNER to postgres;**

**copy public.****ratings\_task FROM 'C:\Users\Public\ratings\_task.csv' DELIMITER ';' CSV HEADER;**

*Описание:*

3.1 Создаем третью таблицу ratings\_task – это таблица, содержащая информацию о присвоении рейтинговыми агентствами рейтингов российским компаниям. Она содержит 17 столбцов, значения в некоторых из них могут быть пропущены:

* rat\_id - числовой код рейтинга
* grade - присвоенная рейтинговая категория
* outlook - прогноз по рейтингу
* change - изменение рейтинга по отношению предыдущему рейтинговому действию
* date - дата действия
* ent\_name - наименование рейтингуемой компании
* okpo - ОКПО рейтингуемой компании
* ogrn - ОГРН рейтингуемой компании
* inn - ИНН рейтингуемой компании
* finst - код референтного инструмента рейтингуемой компании
* agency\_id - код рейтингового агентства (AKM - AK&M, EXP - Эксперт РА, FCH - Fitch, MDS - Moodys, NRA - национальное рейтинговое агентство, RUS - РусРейтинг, SNP - Standard & Poors)
* rat\_industry - отраслевая группа рейтинга
* rat\_type - тип рейтинга (например, Dflt и Cr&D обозначают рейтинги кредитоспособности)
* horizon - горизонт рейтинга (LT - долгосрочный, ShT - краткосрочный)
* scale\_typer - тип рейтинговой шкалы (Isc - международная, Nsc - национальная)
* currency - валюта рейтингуемых обазательств (Icur - доллары, Ncur - национальная валюта)
* backed\_flag- флаг поддерживаемого рейтинга

**CREATE TABLE public.ratings\_task**

**("rat\_id" smallint NOT NULL,**

**"grade" text NOT NULL,**

**"outlook" text,**

**"change" text NOT NULL,**

**"date" date NOT NULL,**

**"ent\_name" text NOT NULL,**

**"okpo" bigint NOT NULL,**

**"ogrn" bigint,**

**"inn" bigint,**

**"finst" bigint,**

**"agency\_id" text NOT NULL,**

**"rat\_industry" text,**

**"rat\_type" text NOT NULL,**

**"horizon" text,**

**"scale\_typer" text,**

**"currency" text,**

**"backed\_flag" text)**

**WITH (OIDS = FALSE)**

**TABLESPACE pg\_default;**

3.2 Далее назначаем владельца таблицы

**ALTER TABLE public.ratings\_task**

**OWNER to postgres;**

3.3 После создания таблицы импортируем в нее данные. Необходимо убедиться, что полученный файл находится в папке с публичным доступом (~\User\Public). Используем файл в формате .csv для импорта данных в таблицу ratings\_task в БД.

**copy public.****ratings\_task FROM 'C:\Users\Public\ratings\_task.csv' DELIMITER ';' CSV HEADER;**

По итогу мы получаем таблицу в БД, состоящую из 17 столбцов и 38252 строк.

4.1 Далее создается отдельная таблица - ratings\_info, содержащая информацию о рейтингах, со столбцом rat\_key, который содержит порядковый номер вида рейтинга и является первичным ключом.

**CREATE TABLE public.ratings\_info**

**("rat\_key" bigint NOT NULL,**

**"rat\_id" smallint NOT NULL,**

**"agency\_id" text NOT NULL,**

**"rat\_industry" text,**

**"rat\_type" text NOT NULL,**

**"horizon" text,**

**"scale\_typer" text,**

**"currency" text,**

**"backed\_flag" text,**

**CONSTRAINT some\_name PRIMARY KEY ("rat\_key"))**

**WITH ( OIDS = FALSE)**

**TABLESPACE pg\_default;**

**ALTER TABLE public.ratings\_info**

**OWNER to postgres;**

4.2 Из исходной таблицы ratings\_task выносится информации о каждом виде рейтинга (его описание) в созданную выше таблицу rating\_info, а также заполняется столбец rat\_key числами по порядку

**insert into ratings\_info select count(\*) over (order by "rat\_id", "agency\_id", "rat\_industry", "rat\_type", "horizon", "scale\_typer", "currency", "backed\_flag") as rate\_key, "rat\_id", "agency\_id", "rat\_industry", "rat\_type", "horizon", "scale\_typer", "currency", "backed\_flag"**

**from (select distinct "rat\_id", "agency\_id", "rat\_industry", "rat\_type", "horizon", "scale\_typer", "currency", "backed\_flag" from public.ratings\_task) as tbl;**

По итогу мы получаем таблицу в БД, состоящую из 8 столбцов и содержащую информацию о 336 различных рейтингах.

4.3 В таблицу ratings\_task добавляется поле с кодами-ссылками на таблицу ratings\_info

**alter table ratings\_task add column "rat\_key" bigint;**

4.4 Заполнение поля с кодами-ссылками в таблице ratings\_task на таблицу ratings\_info

**update ratings\_task**

**set rat\_key=ratings\_info.rat\_key**

**from ratings\_info**

**where ratings\_task."rat\_id"=ratings\_info."rat\_id" AND ratings\_task."agency\_id"=ratings\_info."agency\_id"**

**AND ratings\_task."rat\_industry"=ratings\_info."rat\_industry"**

**AND ratings\_task."rat\_type"=ratings\_info."rat\_type"**

**AND ratings\_task."horizon"=ratings\_info."horizon"**

**AND ratings\_task."scale\_typer"=ratings\_info."scale\_typer"**

**AND ratings\_task."currency"=ratings\_info."currency";**

4.5 Присваиваем полю rat\_key в таблице ratings\_task ограничения внешнего ключа

**ALTER TABLE public.ratings\_task**

**ADD CONSTRAINT fr\_key\_1 FOREIGN KEY (rat\_key) REFERENCES public.ratings\_info (rat\_key);**

По итогу мы получаем в таблице ratings\_task столбец rat\_key, который связывает две таблицы ratings\_info и ratings\_task.

5.1 Вынос из таблицы ratings\_task информации о рейтингуемом лице в таблицу ent\_info и добавление столба ent\_id, содержащего порядковый номер компании, в качестве первичного ключа

**CREATE TABLE public.ent\_info**

**("ent\_id" bigint NOT NULL,**

**"ent\_name" text NOT NULL,**

**"okpo" bigint NOT NULL,**

**"ogrn" bigint,**

**"inn" bigint,**

**"finst" bigint,**

**CONSTRAINT somename PRIMARY KEY ("ent\_id"))**

**WITH (OIDS = FALSE)**

**TABLESPACE pg\_default;**

**ALTER TABLE public.emitent**

**OWNER to postgres;**

5.2 Из исходной таблицы ratings\_task выносится информации о рейтингуемом лице в созданную выше таблицу ent\_info, а также заполняется столбец ent\_id числами по порядку

**insert into ent\_info select count(\*) over (order by "ent\_name", "okpo", "ogrn", "inn", "finst") as ent\_id,**

**"ent\_name", "okpo", "ogrn", "inn", "finst"**

**from (select distinct "ent\_name", "okpo", "ogrn", "inn", "finst"**

**from public.ratings\_task) as num;**

5.3 В таблицу ratings\_task добавляется поле с кодами-ссылками на таблицу ent\_info

**alter table ratings\_task add column "ent\_id" bigint;**

5.4 Заполнение поля с кодами-ссылками в таблице ratings\_task на таблицу ent\_info

**update ratings\_task**

**set ent\_id=ent\_info.ent\_id**

**from ent\_info**

**where ratings\_task."ent\_name"=ent\_info."ent\_name";**

5.5 Присвоение полю ent\_id в таблице ratings\_task ограничения внешнего ключа

**ALTER TABLE public.ratings\_task**

**ADD CONSTRAINT fr\_key\_2 FOREIGN KEY (ent\_id) REFERENCES public.ent\_info (ent\_id);**

По итогу мы получаем в таблице ratings\_task столбец ent\_id, который связывает две таблицы ent\_info и ratings\_task.

5.6 Добавление в таблицу ratings\_task поля с кодами-ссылками на таблицу scale\_exp\_task

**alter table ratings\_task add column "grade\_id" smallint;**

5.7 Заполнение поля с кодами-ссылками на таблицу scale\_exp\_task

**update ratings\_task**

**set grade\_id=scale\_exp\_task."grade\_id"**

**from scale\_exp\_task**

**where ratings\_task."grade"=scale\_exp\_task."grade";**

5.8 Присвоение полю grade\_id в таблице ratings\_task ограничения внешнего ключа

**ALTER TABLE public.ratings\_task**

**ADD CONSTRAINT fr\_key\_3 FOREIGN KEY (grade\_id) REFERENCES public.scale\_exp\_task (grade\_id);**

5.9 Добавление в таблицу credit\_events\_task поля с кодами-ссылками на таблицу ent\_info

**alter table credit\_events\_task add column "ent\_id" smallint;**

5.10 Заполнение поля с кодами-ссылками на таблицу ent\_info

**update credit\_events\_task**

**set ent\_id=ent\_info."ent\_id"**

**from ent\_info**

**where credit\_events\_task."inn"=ent\_info."inn";**

5.11 Присвоение полю ent\_id в таблице credit\_events\_task ограничения внешнего ключа

**ALTER TABLE public.credit\_events\_task**

**ADD CONSTRAINT fr\_key\_4 FOREIGN KEY (ent\_id) REFERENCES public.ent\_info (ent\_id);**

5.12 Удаление вынесенной информации из исходной таблицы ratings\_task, а именно столбцов "rat\_id", "agency\_id", "rat\_industry", "rat\_type", "horizon", "scale\_typer", "currency", "backed\_flag", "ent\_name", "okpo", "ogrn", "inn", "finst"

**alter table public.ratings\_task**

**drop column "rat\_id",**

**drop column "agency\_id",**

**drop column "rat\_industry",**

**drop column "rat\_type",**

**drop column "horizon",**

**drop column "scale\_typer",**

**drop column "currency",**

**drop column "backed\_flag",**

**drop column "ent\_name",**

**drop column "okpo",**

**drop column "ogrn",**

**drop column "inn",**

**drop column "finst";**

По итогу мы получаем таблицу в БД, состоящую из 6 столбцов и содержащей информацию о 2048 компаний.

6. Выведение актуальных рейтингов на 11.11.2014 для вида рейтинга с числовым кодом 47.

Данный запрос предполагает соединение таблиц ent\_id, ratings\_task и ratings\_info с помощью внешнего ключа в таблице ratings\_task: столбцы ent\_id и rat\_key. Ограничения в запросе представляют собой входные данные: актуальные рейтинги на дату 11.11.2014, т.е. на дату равную или предшествующую ей, а также наиболее позднюю при этом ограничении (date <= '11.11.2014' и max(date)), вывод только рейтингов с числовым кодом 47 (rat\_id=47). В качестве еще одного ограничения выступает исключение такого действия как снятие и приостановление (change" <> 'снят' and "change" <> 'приостановлен').

**select ent\_name, grade, assign\_date**

**from public.ent\_info inner join**

**(select \***

**from public.ratings\_info inner join**

**(select public.ratings\_task."ent\_id", public.ratings\_task."rat\_key", grade, change, assign\_date**

**from public.ratings\_task inner join**

**(select max(date) as assign\_date, ent\_id**

**from public.ratings\_task**

**WHERE "date" <= '11.11.2014'**

**group by ent\_id) as tb1**

**on public.ratings\_task."date" = tb1."assign\_date"**

**and public.ratings\_task."ent\_id" = tb1."ent\_id") as tb2**

**on public.ratings\_info."rat\_key"=tb2."rat\_key") as tb3**

**on public.ent\_info."ent\_id"=tb3."ent\_id"**

**where "change" <> 'снят'**

**and "change" <> 'приостановлен'**

**and rat\_id = 47;**

Выведенная таблица содержит информацию о названии рейтингуемой компании, присвоенной рейтинговой категории и дате присвоения актуального рейтинга. Для выбранных характеристик выборка составила 24 рейтинга ввиду того, что в один день компании могли быть присвоены разные рейтинговые категории.