

# Вариант 1. Кредитные рейтинги

## Краткая инструкция к работе

### 0 Ознакомление с заданием

Файлы содержат данные 1) о присвоении рейтинговыми агентствами рейтингов российским компаниям (рейтинговых действиях); 2) о кредитных событиях, связанных с компаниями; 3) о числовой кодификации одной из рейтинговых шкал.

1. Запросами SQL загрузить<sup>1</sup> 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)
2. 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)
3. Связать таблицы в базе, задав внешние ключи, где это целесообразно. (1 балл)
4. В сформированной базе данных составить 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** и **assign\_date** (дату присвоения актуального рейтинга). (2 балла)

### 1 Установка необходимого софта

1) Скачиваем PostgreSQL 12 с официального сайта дистрибутива. Ссылка на скачивание: <https://www.postgresql.org/download/>. Далее будет описана процедура установки данного софта для Windows x86-64.

1.1) Переходим по ссылке, представленной в пункте 1), и выбираем свою операционную систему.

#### Binary packages

Pre-built binary packages are available for a number of different operating systems:

- BSD
  - FreeBSD
  - OpenBSD
- Linux
  - Red Hat family Linux (including CentOS/Fedora/Scientific/Oracle variants)
  - Debian GNU/Linux and derivatives
  - Ubuntu Linux and derivatives
  - SuSE and OpenSuSE
  - Other Linux
- macOS
- Solaris
- Windows

1.2) Далее нажимаем на «Download the Installer».

## Windows installers

### Interactive installer by EnterpriseDB

**Download the installer** certified by EnterpriseDB for all supported PostgreSQL versions.

This installer includes the PostgreSQL server, pgAdmin; a graphical tool for managing and developing your databases, and StackBuilder; a package manager that can be used to download and install additional PostgreSQL tools and drivers. Stackbuilder includes management, integration, migration, replication, geospatial, connectors and other tools.

This installer can run in graphical or silent install modes.

The installer is designed to be a straightforward, fast way to get up and running with PostgreSQL on Windows.

Advanced users can also download a **zip archive** of the binaries, without the installer. This download is intended for users who wish to include PostgreSQL as part of another application installer.

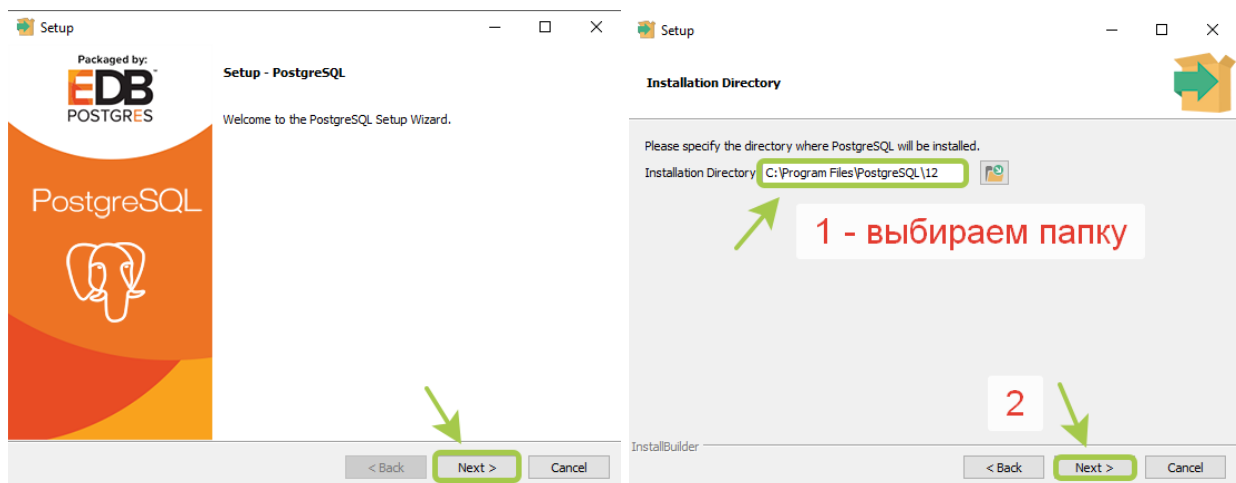
1.3) В появившемся окне нажимаем на поле «Download» напротив нужной нам операционной системы. Должно начаться скачивание файла.

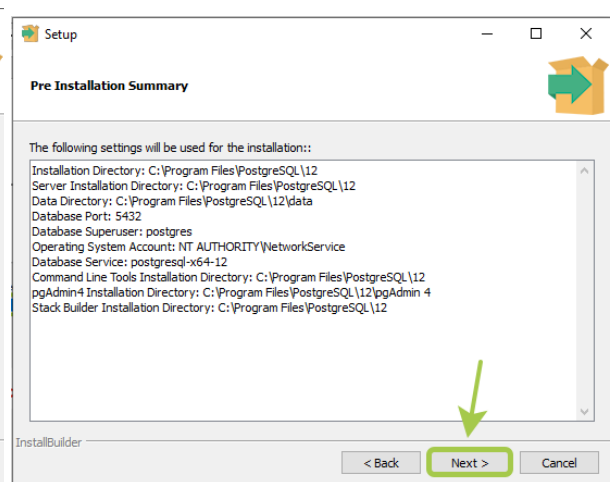
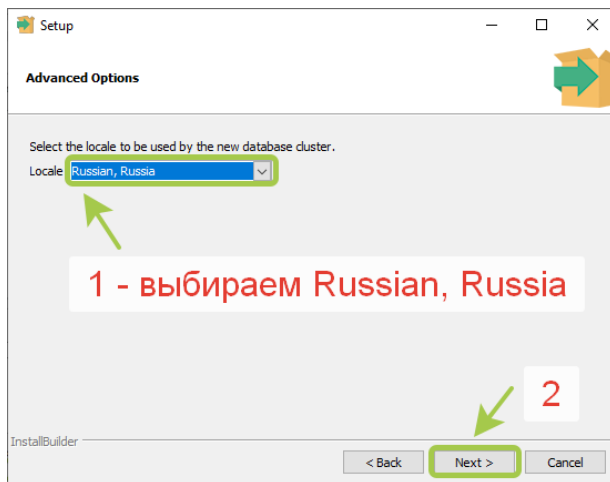
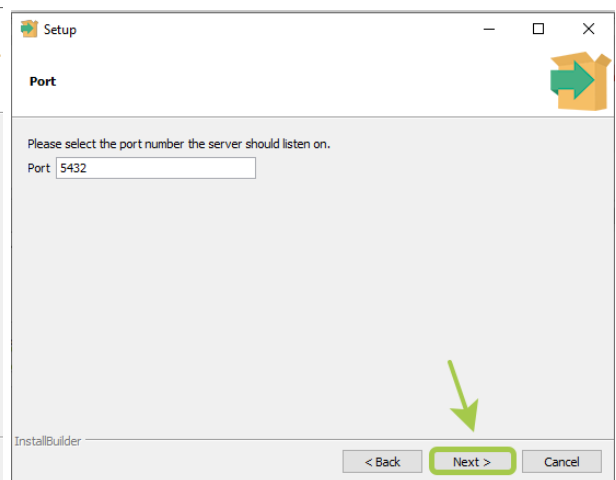
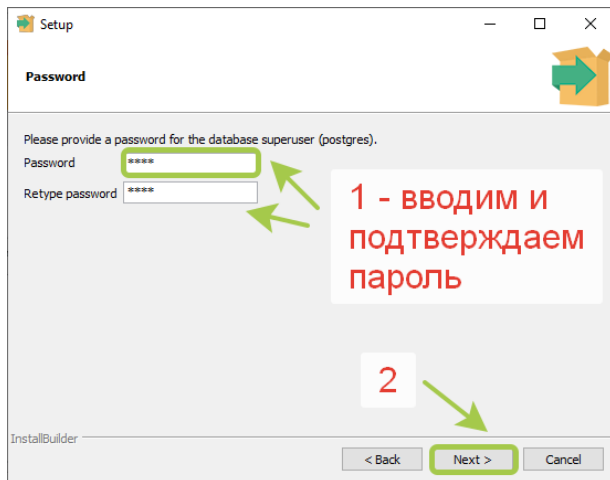
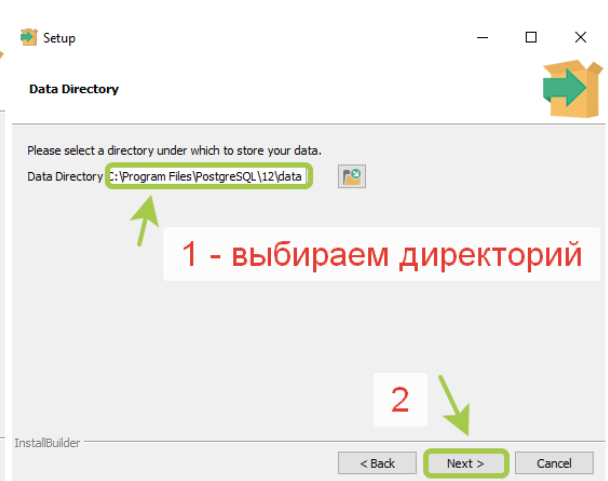
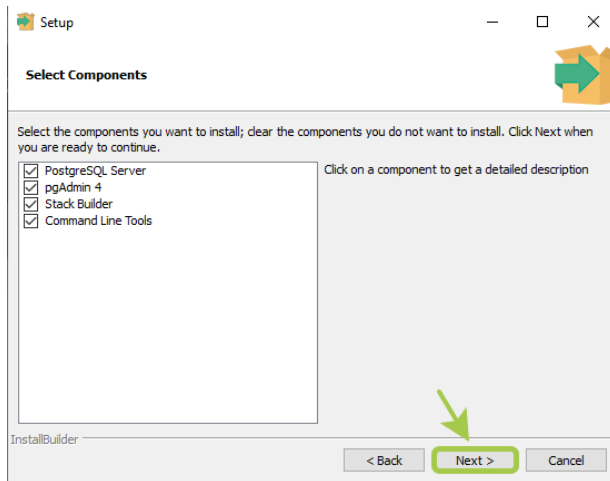
PostgreSQL Database Download					
PostgreSQL Version	Linux x86-64	Linux x86-32	Mac OS X	Windows x86-64	Windows x86-32
12.1	N/A	N/A	Download	Download	N/A
11.6	N/A	N/A	Download	Download	N/A
10.11	Download	Download	Download	Download	Download
9.6.16	Download	Download	Download	Download	Download
9.5.20	Download	Download	Download	Download	Download
9.4.25	Download	Download	Download	Download	Download
9.3.25 (Not Supported)	Download	Download	Download	Download	Download

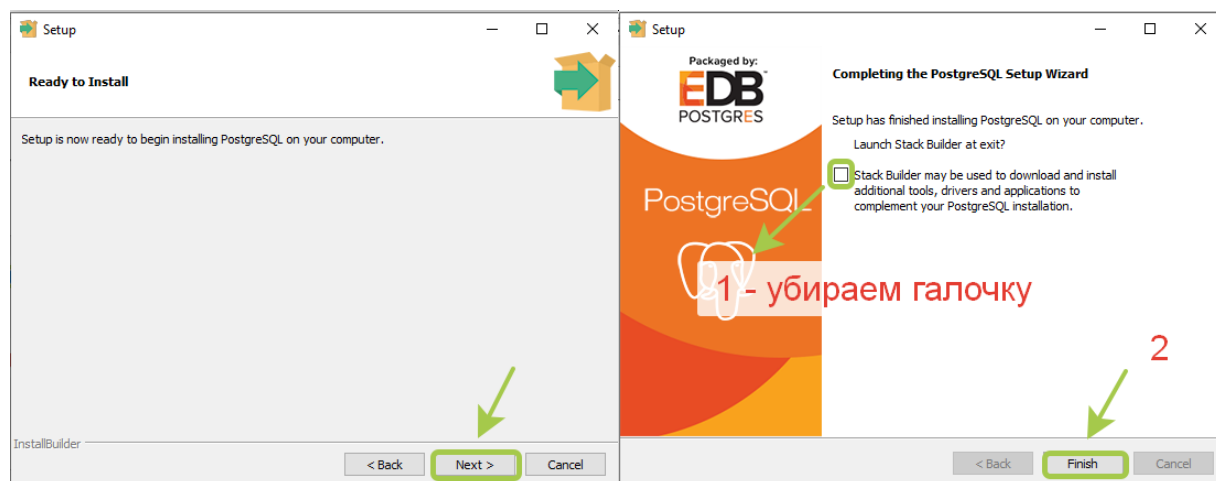
2) Устанавливаем PostgreSQL 12.

2.1) В папке «Загрузки» запускаем скачанный файл «postgresql-12.0-1-windows-x64».

2.2) Далее следуем указаниям на картинках.







Установка завершена.

## 2 Скачивание данных

1) Переходим по следующей ссылке:

<https://yadi.sk/d/osSdoLhMuTFXdw/%D0%A0%D0%B5%D0%B9%D1%82%D0%B8%D0%BD%D0%B3%D0%B8>

2) Нажимаем «Скачать всё». Данные сохранятся в виде архива в «Загрузки».

3) Распаковываем архив в ту же папку. Данную папку можно переместить в удобное для вас место.

## 3 Подготовка данных

1) Открываем файл credit\_events\_task.xls с помощью Microsoft Office Excel.

2) Переходим во вкладку «Файл». Нажимаем «Сохранить как». Выбираем текущую папку. Имя файла оставляем прежним, а тип файла выбираем «CSV (разделители – запятые)».

3) Закрываем файл credit\_events\_task.xls.

4) Открываем файл ratings\_task.xlsx с помощью Microsoft Office Excel.


5) На листе actions кликаем на ячейку E2 и зажимаем Ctrl + Shift + ↓. Таким образом мы выделяем весь столбец значений date. Изменяем формат данного выделенного столбца на «Дата».

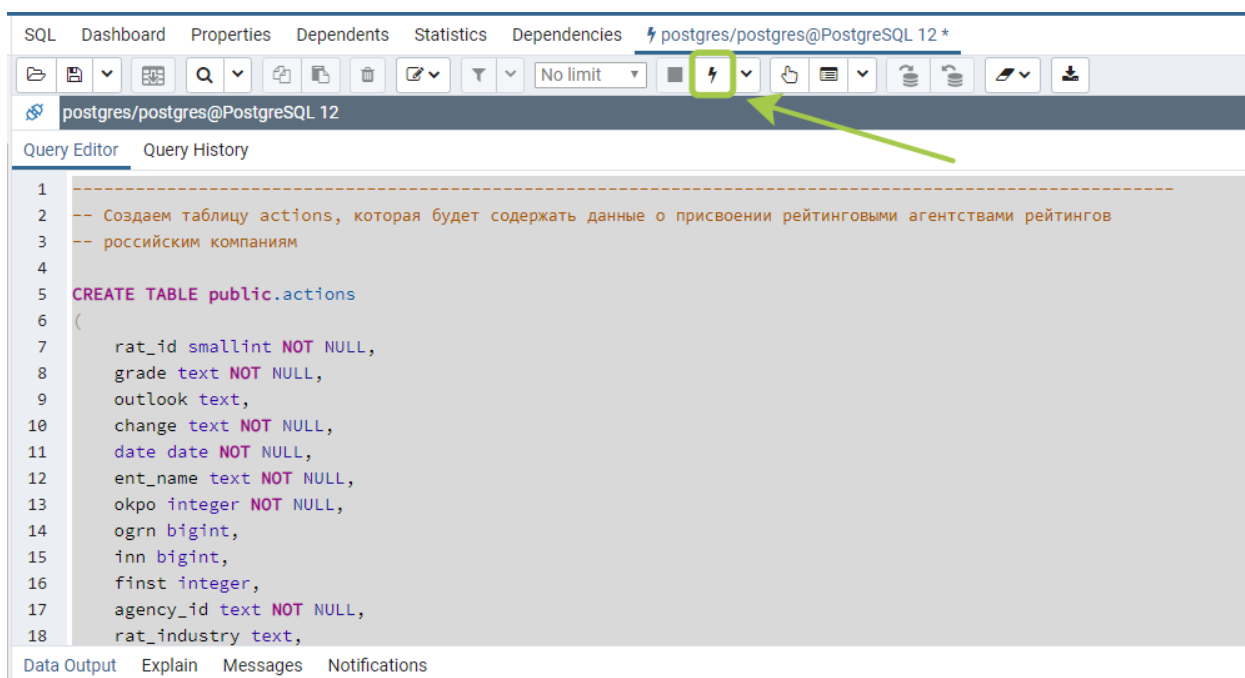
6) Переходим во вкладку «Файл». Нажимаем «Сохранить как». Выбираем текущую папку. Имя файла оставляем прежним, а тип файла выбираем «CSV (разделители – запятые)».

7) Закрываем файл ratings\_task.xlsx.

## 4 Работа с данными

1) Открываем приложение «pgAdmin 4».

- 2) Вводим пароль, придуманный на этапе установки программы, и нажимаем «ОК».
- 3) Кликаем с левой части экрана на «Servers» (при запросе пароля на данном этапе вводим его повторно). Далее кликаем «PostgreSQL 12» → «Databases» → «postgres» → «Schemas» → «public» → «Tables».
- 4) Нажимаем на значок  (Query Tool) в левой верхней части экрана. Открывается Query Editor.
- 5) Переходим по следующей ссылке: [https://vk.com/morozov\\_15](https://vk.com/morozov_15) и пишем в личные сообщения, чтобы я скинул вам файл с кодом.
- 6) Копируем содержимое данного файла в Query Editor (файл можно открыть с помощью блокнота).
- 7) В дальнейшем чтобы исполнить код, будем использовать кнопку «Execute». Ее положение на панели показано на рисунке.



- 8) Последовательно исполняем по частям вставленный код (чтобы исполнить код частично, нужно его выделить и нажать «Execute»). В Query Editor также представлены пояснения к каждому запросу. Перед исполнением кода желательно ознакомиться с тем, для чего он предназначен. Если нет необходимости тщательно разбирать смысл написанных запросов, то можно исполнить часть кода, представленную в данном пункте, за один раз (выделив его до того момента, где необходимо импортировать данные в таблицы, и нажав «Execute»).

```

5 CREATE TABLE public.actions
6 (
7     rat_id smallint NOT NULL,
8     grade text NOT NULL,
9     outlook text,
10    change text NOT NULL,
11    date date NOT NULL,
12    ent_name text NOT NULL,
13    okpo integer NOT NULL,
14    ogrn bigint,
15    inn bigint,
16    finst integer,
17    agency_id text NOT NULL,
18    rat_industry text,
19    rat_type text,
20    horizon text,
21    scale_typer text,
22    currency text,
23    backed_flag text
24 )
25
26 TABLESPACE pg_default;
27
28 ALTER TABLE public.actions OWNER to postgres;
29 COMMENT ON TABLE public.actions
30     IS 'Таблица содержит данные о присвоении рейтинговыми агентствами рейтингов российским компаниям';

35 CREATE TABLE public.events
36 (
37
38     inn bigint NOT NULL,
39     date date NOT NULL,
40     event text NOT NULL
41 )
42
43 TABLESPACE pg_default;
44
45 ALTER TABLE public.events OWNER to postgres;
46 COMMENT ON TABLE public.events
47     IS 'Таблица содержит данные о кредитных событиях, связанных с компаниями';
--

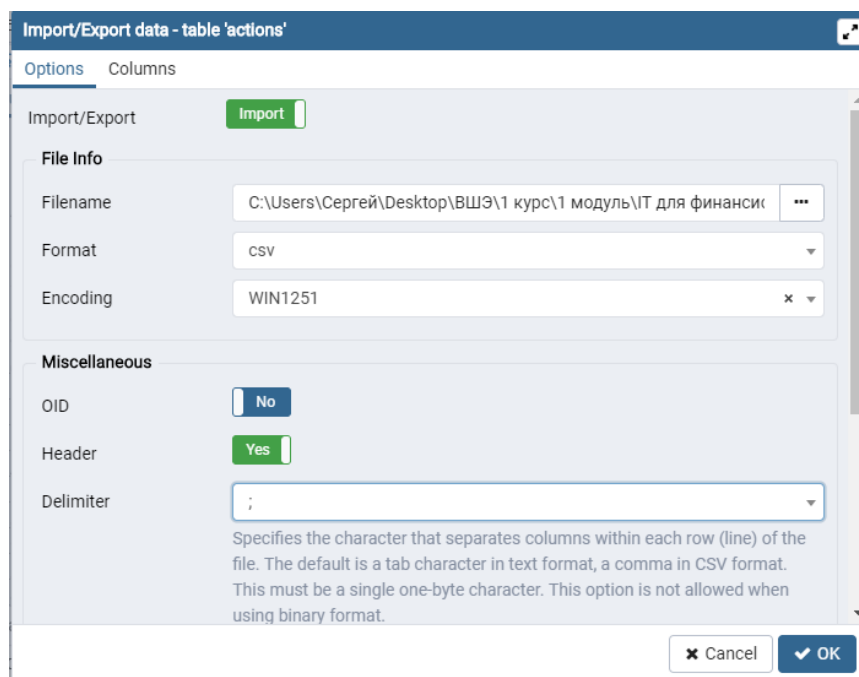
52 CREATE TABLE public.scale_exp
53 (
54     grade text COLLATE pg_catalog."default" NOT NULL,
55     grade_id smallint NOT NULL,
56     CONSTRAINT scale_exp_pkey PRIMARY KEY (grade)
57 )
58
59 TABLESPACE pg_default;
60
61 ALTER TABLE public.scale_exp OWNER to postgres;
62 COMMENT ON TABLE public.scale_exp
63     IS 'Таблица содержит данные о числовой кодификации агентства Эксперт РА (EXP)';

```

9) На панели слева нажимаем правой кнопкой мыши (ПКМ) на «Tables» → «Refresh...». Во вкладке «Tables» появятся созданные нами таблицы actions, events и scale\_exp.

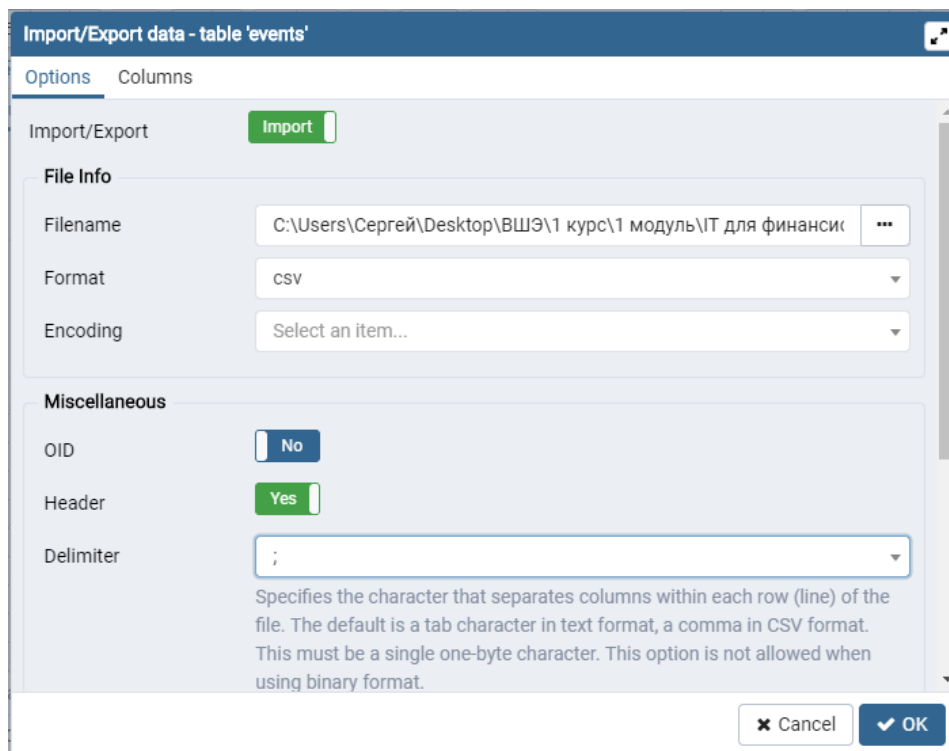
10) Импортируем данные в таблицу actions. Кликаем ПКМ на таблицу actions на панели слева. Далее нажимаем «Import/Export...». В поле Filename указываем путь к файлу

ratings\_task.csv, сохраненному ранее. Остальные настройки устанавливаем так, как показано на картинке, и нажимаем «ОК».



Получаем подтверждение, что данные были успешно загружены в таблицу.

11) Импортируем данные в таблицу events. Кликаем ПКМ на таблицу events на панели слева. Далее нажимаем «Import/Export...». В поле Filename указываем путь к файлу credit\_events\_task.csv, сохраненному ранее. Остальные настройки устанавливаем так, как показано на картинке, и нажимаем «ОК».



Получаем подтверждение, что данные были успешно загружены в таблицу.

12) Импортируем данные в таблицу scale\_exp. Кликаем ПКМ на таблицу scale\_exp на панели слева. Далее нажимаем «Import/Export...». В поле Filename указываем путь к файлу scale\_EXP\_task.csv. Остальные настройки устанавливаем так, как показано на картинке, и нажимаем «ОК».

Import/Export data - table 'events'

Options Columns

Import/Export **Import**

**File Info**

Filename C:\Users\Cepрей\Desktop\ВШЭ\1 курс\1 модуль\ИТ для финанси...

Format CSV

Encoding Select an item...

**Miscellaneous**

OID No

Header Yes

Delimiter ;

Specifies the character that separates columns within each row (line) of the file. The default is a tab character in text format, a comma in CSV format. This must be a single one-byte character. This option is not allowed when using binary format.

Cancel OK

Получаем подтверждение, что данные были успешно загружены в таблицу.

13) Продолжаем последовательно по частям (или целиком всю часть, представленную в данном пункте, если нет необходимости по шагам разбирать исполняемые запросы) исполнять код в Query Editor:

```
75 CREATE TABLE public.rat_info
76 (
77     rat_id smallint NOT NULL,
78     agency_id text,
79     rat_industry text,
80     rat_type text,
81     horizon text,
82     scale_typer text,
83     currency text,
84     backed_flag text,
85     CONSTRAINT rat_info_pkey PRIMARY KEY (rat_id)
86 )
87
88 TABLESPACE pg_default;
89
90 ALTER TABLE public.rat_info OWNER to postgres;
--
96 ALTER TABLE actions DROP COLUMN rat_id;
```



```

103 INSERT INTO rat_info
104 SELECT COUNT(*) OVER
105 (ORDER BY agency_id, rat_industry, rat_type, horizon, scale_typer, currency, backed_flag) AS rat_id,
106 agency_id, rat_industry, rat_type, horizon, scale_typer, currency, backed_flag
107 FROM (SELECT DISTINCT agency_id, rat_industry, rat_type, horizon, scale_typer, currency, backed_flag
108 FROM actions) as sample;

113 ALTER TABLE actions ADD COLUMN rat_id smallint;

118 UPDATE actions
119 SET rat_id = rat_info.rat_id
120 FROM rat_info
121 WHERE actions.agency_id = rat_info.agency_id AND
122 (actions.rat_industry = rat_info.rat_industry
123 OR (actions.rat_industry is null and rat_info.rat_industry is null)) AND
124 (actions.rat_type = rat_info.rat_type
125 OR (actions.rat_type is null and rat_info.rat_type is null)) AND
126 (actions.horizon = rat_info.horizon
127 OR (actions.horizon is null and rat_info.horizon is null)) AND
128 (actions.scale_typer = rat_info.scale_typer
129 OR (actions.scale_typer is null and rat_info.scale_typer is null)) AND
130 (actions.currency = rat_info.currency
131 OR (actions.currency is null and rat_info.currency is null)) AND
132 (actions.backed_flag = rat_info.backed_flag
133 OR (actions.backed_flag is null and rat_info.backed_flag is null));

138 ALTER TABLE public.actions
139 ADD CONSTRAINT fr_key_2 FOREIGN KEY (rat_id) REFERENCES public.rat_info (rat_id);
...

144 alter table public.actions
145 drop column agency_id,
146 drop column rat_industry,
147 drop column rat_type,
148 drop column horizon,
149 drop column scale_typer,
150 drop column currency,
151 drop column backed_flag;

157 CREATE TABLE public.company
158 (
159     "ent_name" text NOT NULL,
160     "okpo" integer NOT NULL,
161     "ogrn" bigint,
162     "inn" bigint,
163     "finst" integer,
164     CONSTRAINT company_pkey PRIMARY KEY (ent_name)
165 )
166
167 TABLESPACE pg_default;
168
169 ALTER TABLE public.company OWNER to postgres;

175 INSERT INTO company
176 SELECT DISTINCT ent_name, okpo, ogrn, inn, finst
177 FROM actions
178 ORDER BY ent_name;
...

183 ALTER TABLE public.actions
184 ADD CONSTRAINT fr_key_1 FOREIGN KEY (ent_name) REFERENCES public.company (ent_name);
185

```

```

189 alter table public.actions
190 drop column okpo,
191 drop column ogrn,
192 drop column inn,
193 drop column finst;

201 select ent_name, grade, assign_date
202 from public.actions inner join
203 (select max(date) as assign_date, ent_name as ent2_name
204 from public.actions
205 where rat_id = 50
206 AND date <= '12-01-2014'
207 group by ent_name) as zapros1
208 on public.actions.date = zapros1.assign_date
209 and public.actions.ent_name = zapros1.ent2_name
210 where change <> 'снят'
211 and change <> 'приостановлен'
212 and rat_id = 50;

```

14) Итогом исполнения запросов является таблица, которая служит ответом на пункт 4 задания:

	ent_name text	grade text	assign_date date
1	Bank of Tokyo...	C	2013-07-03
2	Kazkommerts ...	E	2013-10-02
3	Абсолют Банк	E+	2013-05-31
4	Авангард АКБ	E+	2013-05-23
5	Автоторгбанк	E+	2012-12-10
6	Агропромкре...	E+	2009-07-13
7	Азиатско-Тих...	E+	2013-10-29
8	АйМаниБанк	E+	2013-02-15

15) Для контроля выполнения остальных пунктов можно воспользоваться следующим алгоритмом: «Tables» (ПКМ) → «Refresh...» → «ИМЯ ПРОВЕРЯЕМОЙ ТАБЛИЦЫ» (ПКМ) → «View/Edit Data» → «First 100 Rows».