**Лабораторная работа № 6**

**Создание объектов БД PostgreSQL**

**Цель работы**. Научиться в среде SQL PowerArchitect для физической модели данных формировать сценарий SQL по созданию объектов БД, создавать подключение к серверу баз данных, выполнять сценарийSQLиз программной среды SQL PowerArchitect и с использованием сценария в pgAdmin. Научиться подготавливать и загружать данные в таблицы базы данных, а также выгружать данные из таблиц базы данных. Освоить приемы проверки согласованности данных.

1. **Теоретические сведения**

**1.1 Создание объектов БД**

После создания физической модели данных в SQL PowerArchitect таблицы физической модели необходимо перенести в базу данных.

Для этого в SQL PowerArchitect формируется пакет команд SQL по созданию таблиц и других объектов базы данных, который называют сценарием SQL. Данный сценарий можно использовать двумя способами:

1. сохранить в файле для выполнения в pgAdmin;
2. выполнить команды сценария непосредственно из среды SQL PowerArchitect, используя подключение к серверу БД.

Первый способ используется при отсутствии соединения с БД на компьютере, на котором выполняется моделирование данных.

**1.1.1 Создание сценария SQL схемы БД**

Для формирования команд создания таблиц БД необходимо нажать кнопку*Сконструировать сценарий SQL*или выполнить команду меню *Инструменты – Сконструировать* (рис.1).

|  |  |
| --- | --- |
|  |  |

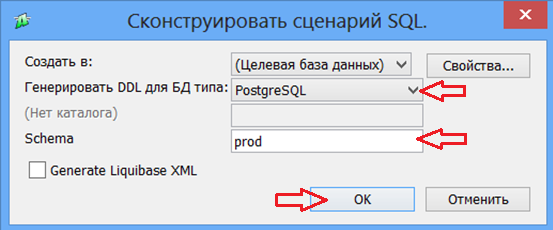
**Рис. 1 Формирование сценария SQL**

Откроется окно «Сконструировать сценарий SQL» (рис.2).

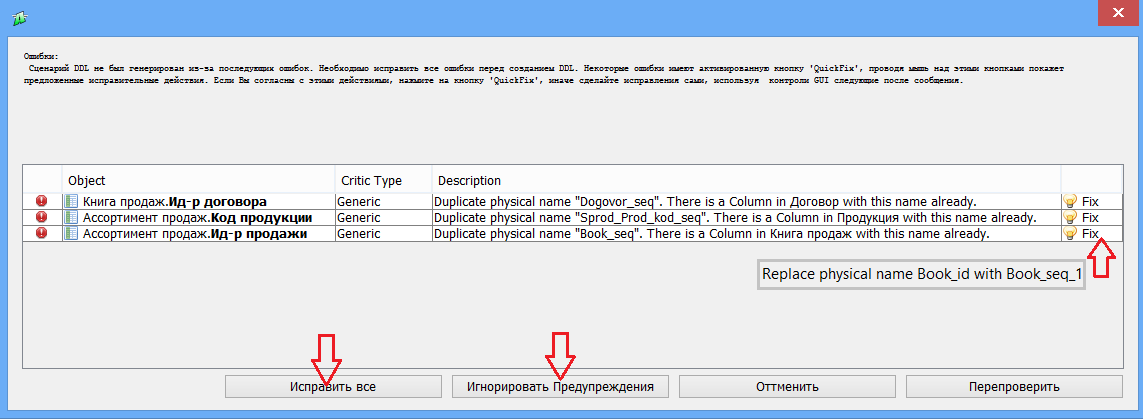
В окне необходимо:

* выбрать тип БД в списке *Генерировать DDL для БД типа* (в нашем случае PostgreSQL)*;*
* Задать имя *Схемы*, созданной на предыдущем лабораторном занятии (в нашем случае prod). В данной схеме БД будут созданы таблицы;
* нажать кнопку ОК.

Если при формировании сценария будут ошибки в модели (дублирование имен объектов и т.п.) то будет выведено окно с перечнем таких ошибок (Рис. 3). Ошибки можно попытаться исправить. Варианты исправления показываются во всплывающем окне, если щелкнуть мышью в строке с предупреждением по колонке *Fix*. Для исправления ошибок надо нажать кнопку *Исправить все*. Можно также игнорировать ошибки, нажав кнопку *Игнорировать предупреждения*.

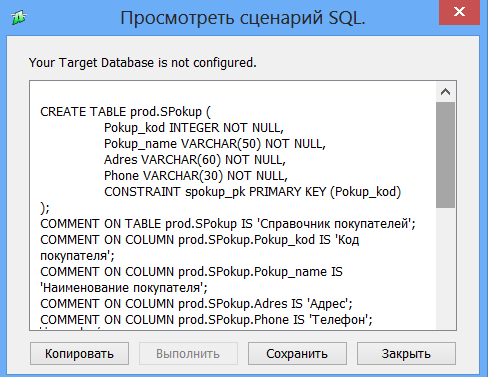


**Рис.2 Задание параметров сценария**



**Рис.3 Окно с перечнем ошибок**

Если ошибок при формирования сценария не обнаружено (также при игнорировании имеющихся ошибок), откроется окно с командами SQL (рис.4).

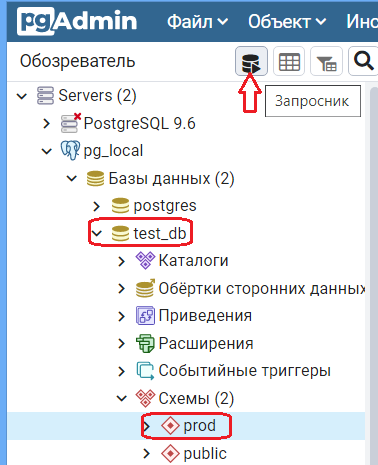


**Рис. 4. Текст сгенерированного сценария SQL**

Для сохранения команд сценария в файле необходимо нажать кнопку *Сохранить*. Далее надо выбрать папку и имя файла для команд сценария. Сгенерированный сценарий можно выполнить на целевом сервере БД. Необходимо гарантировать, что кодировка файла со сценарием UTF8. В этом можно убедиться, открыв сохраненный файл сценария в текстовом редакторе NotePad++. Если кодировка файла ANSI, то кодировку в редакторе надо преобразовать в UTF8.

Для выполнения сценария будем использовать программу pgAdmin. Запустим программу pgAdmin и выполним следующие действия (рис. 5):

* активизируем подключение pg\_local под учетной записью суперпользователя admin, созданной на предыдущем лабораторном занятии;
* выделим в браузере в базе данных test\_db схему prod;
* откроем окно ввода SQL команд.



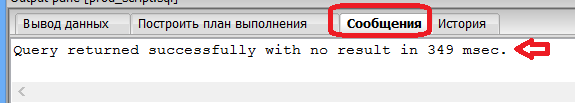
**Рис.5 Вывод окна выполнения запросов**

В окне ввода команд необходимо загрузить файл с командами сценария SQL, полученный при создании физической модели (рис. 6а). Для выполнения команд необходимо нажать кнопку *Выполнить запрос (Execute)* либо нажать кнопку F5 (рис. 6б).

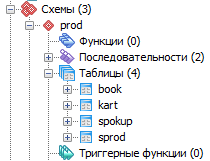
|  |  |
| --- | --- |
|  |  |
| а) | б) |

**Рис. 6 Загрузка и выполнение сценария SQL в pgAdmin 4**

Если сценарий будет выполнен успешно, то в нижней части окна запросов на вкладке Сообщения будет выведен текст *Query returned successfully* (рис. 7), а в дереве объектов в схеме prod отобразятся таблицы и последовательности (рис. 8).



**Рис. 7 Успешное выполнение сценария**



**Рис. 8 Созданные таблицы в схеме prod**

**1.1.2 Выполнение сценария SQL в целевой БД**

**из среды SQL PowerArchitect**

Другим вариантом создания схемы базы данных является выполнение сценария в целевой БД непосредственно из среды SQL PowerArchitect. Данный вариант можно использовать при наличии драйвера для подключения к БД.

Для этого необходимо создать подключение к серверу БД PostgreSQL, выполнив пункт меню

*Подключения – Добавить подключение источника – Новое подключение*

В окне создания нового подключения выполним следующие действия (рис.9):

1. Зададим имя подключения (например,pg\_local);
2. Выберем тип сервера БД (PostgreSQL);
3. Зададим параметры подключения

- имя хоста (*Hostname*), на котором запущен сервер БД. В нашем случае зададим *localhost*, т.к. это локальный компьютер, на котором выполняется работа;

- номер порта (*Port*). *Порт* -это некоторое число, которое используется для идентификации процесса (программы), который должен обработать данные. Хотя работа выполняется на локальном компьютере, а не в сети, сервер БД подчиняется логике сетевой программы. Для всех сетевых программ существуют стандартные номера портов. Для сервера БД PostgreSQL таким стандартным номером порта является 5432. Данный номер будет подставлен автоматически при выборе сервера;

- имя БД (Database). Введем имя созданной на предыдущем лабораторном занятии БД test\_db;

4) Зададим имя пользователя (*Username*). Введем имя суперпользователя admin, созданного на прошлом лабораторном занятии;

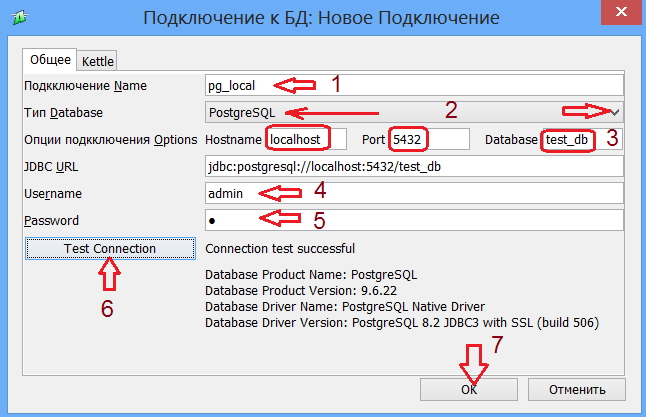
5) Зададим пароль (*Password*). Будем использовать пароль созданного суперпользователя – admin. Символы пароля при вводе будут замещаться символами-заменителями;

6) Нажмем кнопку *TestConnection* для проверки работоспособности подключения. Если рядом с кнопкой появится текст

*Connection test successful*

то это означает, что подключение создано;

7) Нажмем кнопку ОК для сохранения подключения.

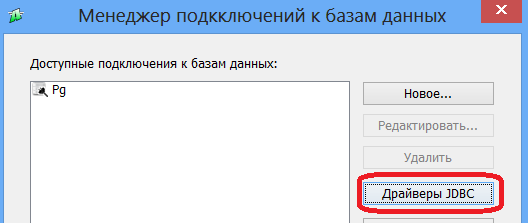


**Рис.9 Создание подключения к серверу PostgreSQL**

***Замечание.*** При проверке подключения после нажатия кнопки *TestConnection* может быть выведено сообщение об ошибке. Оно может быть связано с неправильным заданием параметров подключения либо с отсутствием драйвера подключения к БД. Драйверы для различных БД устанавливаются автоматически, если при установке SQL PowerArchitect был выбран пакет установки, включающий эти драйверы (в имени пакета присутствует «jdbc»). Если использовался другой пакет, то драйвер надо скачать в интернете (найти самостоятельно) и подключить к программе в окне менеджера подключений, вызвав его в пункте меню

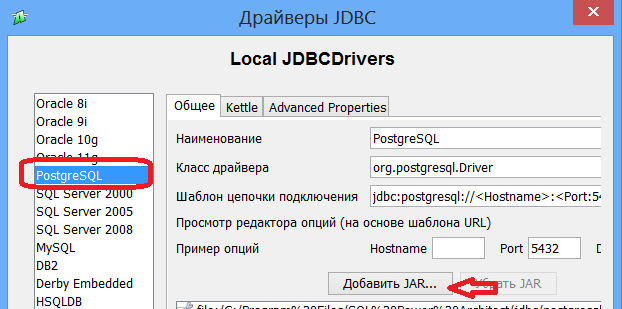
*Подключения – Менеджер Подключений к Базам Данных*

В окне менеджера необходимо нажать кнопку *Драйверы JDBC* (рис.10)

******

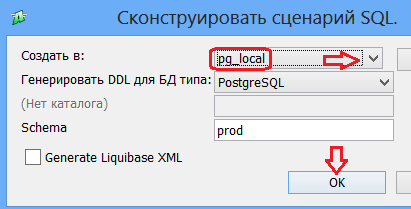
**Рис. 10 Окно подключения драйверов JDBC**

В открывшемся окне (рис.11) необходимо выбрать тип сервера БД (PostgreSQL) и, нажав кнопку Добавить JAR, найти скачанный драйвер. Подключенный драйвер отобразиться в окне.



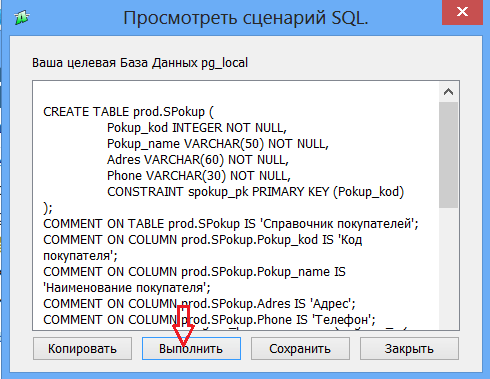
**Рис. 11 Подключение драйвера JDBC для PostgreSQL**

После создания подключения запустим формирование сценария, нажав кнопку *Сконструировать сценарий SQL* или выполнив команду меню *Инструменты – Сконструировать* (рис.1). Но в отличие от первого варианта в поле «*Создать в»* выберем подключение к целевой БД, созданное выше, и нажмем кнопку ОК (рис.12).



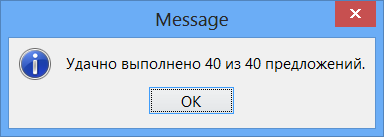
**Рис. 12 Создание сценария с подключением к целевому серверу БД**

В окне сценария SQL нажмем кнопку «Выполнить» (рис.13).



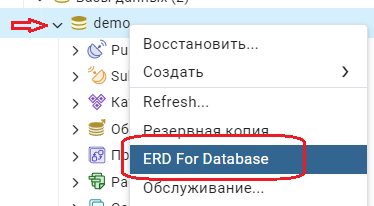
**Рис.13 Выполнение сценария в целевой БД**

Если все выполнено правильно, то получим сообщение об успешном выполнении операторов сценария (рис. 14).

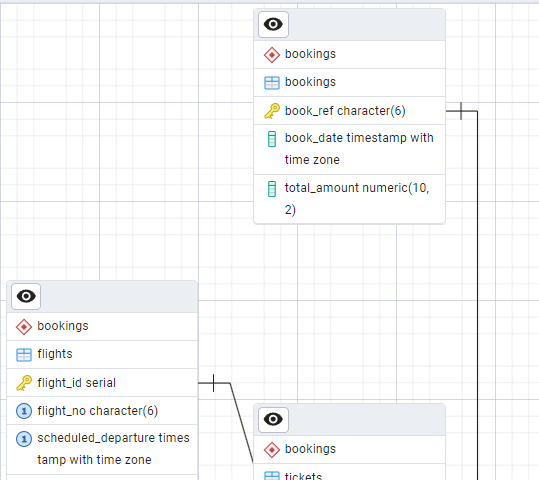
****

**Рис. 14 Сообщение о выполнении команд файла**

Можно вывести ERD-диаграмму для текущей схемы БД. Для этого необходимо на узле БД выбрать в контекстном меню пункт **ERD For Dtabase** (рис. 15). Диаграмма будет выведена на правой панели (рис. 16).



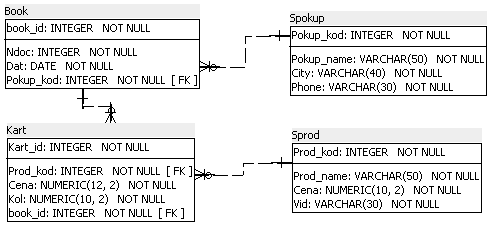
**Рис. 15 Вывод ERD диаграммы модели данных**



**Рис. 16 ERD диаграмма модели данных**

* 1. **Загрузка данных из текстовых файлов**

Загрузку и выгрузку данных будем рассматривать на примере БД Продаж (рис. 17).



**Рис. 17 База данных Продажи**

Последовательность загрузки данных должна быть следующая. Вначале загружаются данные в таблицы, не имеющие внешних ключей, затем в таблицы, которые ссылаются на таблицы с уже загруженными данными и т.д.

Для рассматриваемой базы данных Продаж последовательность загрузки должна быть следующей:

* Продукция (таблица Sprod);
* Покупатель (таблица Spokup);
* Книга продаж (таблица Book);
* Ассортимент продаж (таблица Kart).

Данные могут быть подготовлены в текстовых файлах формата CSV.

Файлы CSV содержат данные в текстовом формате, разделенные запятыми. Одна строка — это одна запись об одном объекте. Однотипные свойства в каждой строке занимают фиксированные места. Некоторые значения могут быть пропущены. В месте пропуска будут подряд следовать символы разделители. В первой строке файла могут быть указаны имена свойств объекта, разделенные запятыми. Имена свойств обычно соответствуют именам колонок таблицы БД. Файлы формата CSV удобно подготавливать и просматривать в Microsoft Excel.

**1.2.1 Подготовка данных в Excel**

Откройте программу Microsoft Excel и введите данные для таблиц (рис.18-21). Обратите внимание на правильность заполнения внешних ключей.



**Рис. 18 Данные для таблицы sprod (Продукция)**



**Рис. 19 Данные для таблицы spokup (Покупатели)**

****

**Рис. 20 Данные для таблицы book (Книга продаж)**

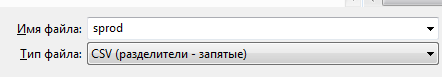


**Рис. 21 Данные для таблицы kart (Ассортимент продаж)**

Введенные данные необходимо сохранить в формате CSV (текстовые данные с разделителями между колонками значений). Для этого необходимо выполнить команду меню

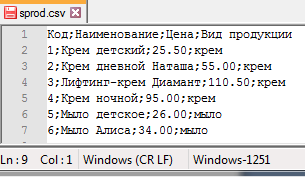
*Сохранить как - Другие форматы*

В открывшемся окне необходимо выбрать формат сохранения CSV, а имя файла на английском (рис.22). Для удобства выберем имя, совпадающее с именем таблицы, для которой данные предназначены.



**Рис. 22 Сохранение данных в формате CSV**

Откроем один из сохраненных файлов в текстовом редакторе (рис.23). Первая строка содержит название атрибутов. Это надо учитывать при загрузке данных в таблицы.



**Рис. 23 Данные в формате CSV**

Обратите внимание, что кодировка файла должна совпадать с кодировкой созданной в лабораторной работе 5 БД – Windows-1251 (ANSI).

**1.2.2 Загрузка данных в таблицы БД**

Для загрузки данных из подготовленных файлов в таблицы базы данных используется команда COPY, имеющая следующую структуру (в квадратные скобки заключается элемент, который может быть опущен):

COPY *имя\_таблицы* [ ( *имя\_столбца* , ... ) ]

FROM '*имя\_файла*'

[*параметр* , ...]

В команде могут быть использованы *параметры (ниже указаны основные)*:

*имя\_формата* данных (по умолчанию txt). Следует выбрать формат CSV (нужно указывать явно);

DELIMITER '*символ\_разделитель*' – символ-разделитель значений полей (для CSV по умолчанию запятая).

Если текстовые строки содержат запятые, то в качестве разделителя можно использовать символы ';' или '|';

NULL - '*маркер\_NULL*' определяет строку, задающую значение NULL. По умолчанию в текстовом формате это \N (обратная косая черта и N), а в формате CSV — пустая строка без кавычек;

HEADER [ *boolean* ] - Указывает, что файл в 1-й строке содержит строку заголовка с именами столбцов. Этот параметр допускается только для формата CSV. При отсутствии значения *boolean* подразумевается *true;*

QUOTE '*символ\_кавычек*' - Указывает символ кавычек, используемый для заключения значений в кавычки. Этот параметр поддерживается только для формата CSV;

ENCODING '*имя\_кодировки*' - Указывает, что файл имеет заданную*, например* WIN1251.

Для выполнения команды необходимо открыть окно ввода SQL-запросов. Пример записи команды загрузки в таблицу sprod из файла sprod.csv, находящегося в каталоге d:/data:

SET SEARCH\_PATH TO prod; // Установка текущей схемы

COPY sprod

FROM 'd:/data/sprod.csv' CSV

DELIMITER ';' // разделитель «точка с запятой»

HEADER; // первая строка содержит имена атрибутов

Если в файле использована кодировка 1251, а в БД используется кодировка UTF8, то при экспорте будет выведено сообщение об ошибке:

*ОШИБКА: неверная последовательность байт для кодировки "UTF8"*

В этом случае надо указать кодировку исходного файла в параметре:

ENCODING 'WIN1251'

И команда загрузки будет такой:

COPY sprod

FROM 'c:/data/sprod.csv' CSV

DELIMITER ';'

HEADER

ENCODING 'WIN1251';

Если все данные соответствуют тем типам, которые определены в таблице БД, то ошибок в процессе загрузки данных не будет. Если же ошибки появляются, необходимо скорректировать подготовленные данные.

Полный пакет команд загрузки данных в базу данных Продажи (последовательность команд должна быть именно такой):

SET SEARCH\_PATH TO prod; -- Установка текущей схемы

COPY sprod FROM 'd:/data/sprod.csv' CSV

DELIMITER ';' HEADER ENCODING 'WIN1251';

COPY spokup FROM 'd:/data/spokup.csv' CSV

DELIMITER ';' HEADER ENCODING 'WIN1251';

COPY book FROM 'd:/data/book.csv' CSV

DELIMITER ';' HEADER ENCODING 'WIN1251';

COPY kart FROM 'd:/data/kart.csv' CSV

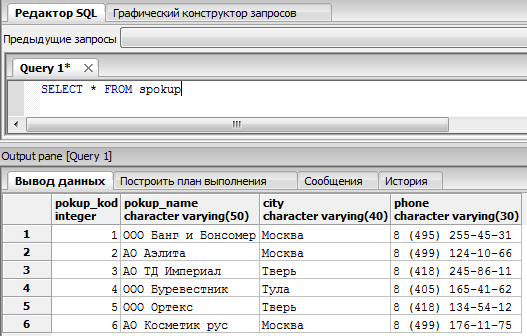
DELIMITER ';' HEADER ENCODING 'WIN1251';

***Замечание.*** Путь к файлам рассматривается относительно сервера. Поэтому при подключении к удаленному серверу необходимо использовать сетевое имя. Для сервера БД на локальном компьютере задаем обычные пути к файлу. Желательно размещать данные не в личных папках текущего пользователя, а в обычной папке на диске, как показано впримере.

Можно проверить загрузку данных, выполнив запрос к таблице spokup (Покупатели):

SELECT \* FROM prod.spokup

Результат запроса приведен на рисунке 24.



**Рис. 24 Результат запроса выборки данных**

При загрузке данных мы явно задавали значения суррогатным первичным ключам (идентификаторам), которые при добавлении данных должны заполняться с использованием последовательностей. Чтобы механизм автоувеличения далее работал корректно необходимо изменить текущие значения последовательностей на последнее значение, заданное в файлах данных. Это выполняется оператором:

setval (имя\_последовательности, последнее\_значение, true)

Максимальные значения первичного ключа в таблицах (по файлам данных):

book: book\_id=5

kart: kart\_id=10

Поэтому для изменения текущего значения последовательностей для этих колонок необходимо выполнить операторы:

SELECT setval('prod.book\_seq', 5, true);

SELECT setval('prod.kart\_seq', 10, true);

**1.2.3 Выгрузка данных в файл**

Можно не только загружать данные в таблицы, но также и выгружать данные из таблицы во внешние файлы. Это также выполняется командой COPY:

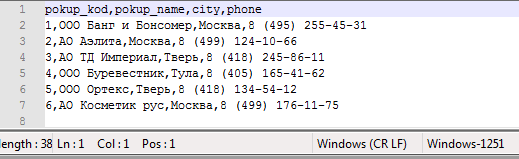
COPY имя\_таблицы TO 'путь и имя файла' [параметры];

Необязательные параметры такие же, как и при загрузке данных. Для примера выведем данные из таблицы Покупатель (spokup) в формате CSV с названием колонок в первой строке, с разделителем «запятая», в кодировки WIN-1251 в файл c:\data\spokup.csv

COPY prod.spokup TO 'c:/data/spokup.csv' DELIMITER ','

CSV HEADER ENCODING 'WIN1251';

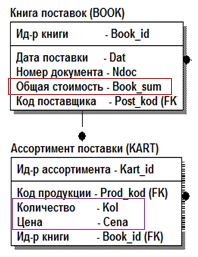
Содержимое файла в текстовом редакторе приведено на рисунке 25.



**Рис. 25 Результат выгрузки данных**

* 1. **Проверка согласованности загруженных данных**

Загруженные данные должны отвечать требованиям согласованности, в соответствии с моделью данных в предметной области. В индивидуальных заданиях принцип согласованности данных должен выполняться для вычисляемой (расчетной) колонки, присутствующей в одной или нескольких таблицах. Пример такой вычисляемой колонки приведен для модели Поставки (рис. 26).



**Рис. 26. Расчетная колонка в модели Поставки**

Расчетная колонка ***Общая стоимость*** в таблице **Книга поставок** определяется по экземплярам-строкам (их может быть несколько) таблицы **Ассортимент поставки**, по колонкам ***Цена*** и ***Количество***

**по сущности Ассортимент поставки**

Общая стоимость **=**∑ (Цена \* Кол-во)

**фильтр по Ид-ру книги**

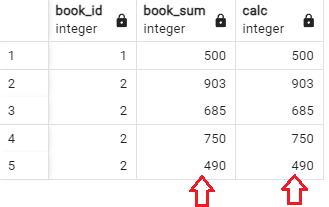
Для проверки правильности загрузки значений вычисляемой колонки необходимо выполнить запрос. Для приведенного примера текст запроса:

SELECT book\_id, book\_sum,

(SELECT SUM(kol\*cena) FROM kart.book\_id=b.book\_id) calc

FROM book b

В список вывода оператора запроса включаются колонки главной таблицы Книга поставок (таблица book): первичный ключ (book\_id), вычисляемая колонка (book\_sum). Также выводится значение подзапроса к подчиненной таблице Ассортимент поставки (таблица kart). Подзапрос выбирает строки, в которых значение внешнего ключа подчиненной таблицы kart (колонка book\_id) совпадает со значением первичного ключа главной таблицы (колонка book\_id). В отобранных строках суммируется выражение произведения колонок объема (колонка kol) и цены (колонка cena). Данные считаются согласованными, если в результате запроса совпадают значения второй и третьей колонок во всех его строках (рис. 27).

.

**Рис. 27. Результат запроса проверки расчетной колонки**

1. **Порядок выполнения лабораторной работы**

Для базы данных созданной на предыдущем лабораторном занятии:

1. Создать объекты (таблицы и другие объекты) в схеме целевой БД для модели данных, созданной на лабораторном занятии Физическое проектирование БД.
2. Подготовить файлы данных
3. Выполнить загрузку данных в таблицы БД из внешних файлов
4. Проверить согласованность данных по вычисляемой колонке, выполнив запрос
5. Выполнить выгрузку данных из таблиц БД во внешний файл
6. **Содержание отчета**

* Скриншот ERD-диаграммы модели данных
* Скриншоты подготовленных данных
* Текст запроса проверки вычисляемой колонки и скриншот результата выполнения запроса
* Скриншот выгрузки данных