Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ.**

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Программист

ОТЧЕТ

ОСНОВЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ

АДМИНИСТРИРОВАНИЕ POSTGRESQL

Листов: 17

|  |  |
| --- | --- |
| Выполнил студент  Группы П50-1-22  А.А. Себежко  «1» мая 2024 года | Проверил преподаватель  \_\_\_\_\_\_\_\_\_\_\_Е.М.Парамонова  «\_\_\_\_» \_\_\_\_\_\_\_2024 года |

Москва 2024

Цель работы**:** Научиться создавать на диалекте POSTGRES различные процедуры, функции, триггеры, представления, пользователей и их группы. Освоить навык экспорта и импорта данных из таблиц, а также научится делать бэкапы и ресторы БД.

Ход работы:

Создаем таблицы

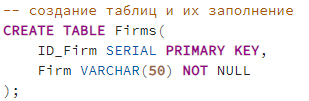


Рисунок 1 - создание таблиц

Заполняем таблицы

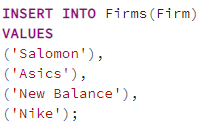


Рисунок 2 - заполнение таблицы

Создаем представления. Первое представление – StaffDetails, которое выводит информацию о сотрудниках: полное имя, их пол и корпус, в котором работают. С помощью CONCAT соединяем три столбца в один «Полное имя».

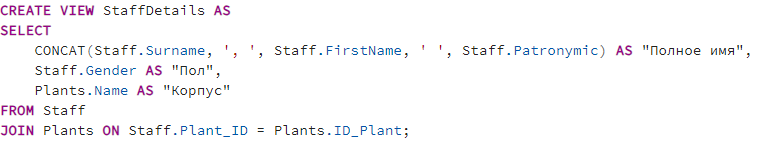


Рисунок 3 - StaffDetails

Вывод виртуальной таблицы StaffDetails.

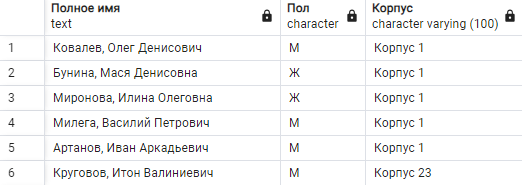


Рисунок 4 - вывод таблицы StaffDetails

Создается еще одна виртуальная таблица, которая возвращает таблицу, где можно узнать какую модель в какой сезон и на каком заводе производили.

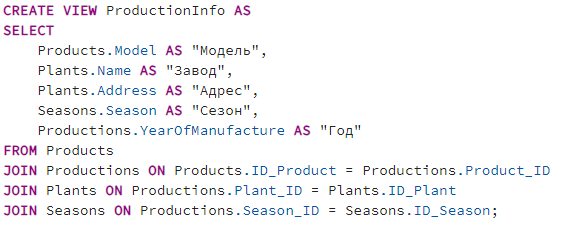


Рисунок 5 - ProductionInfo

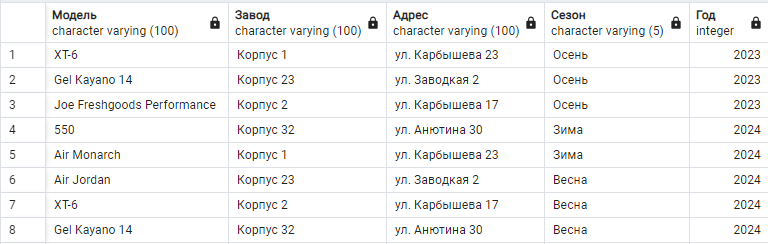


Рисунок 6 - вывод таблицы ProductionInfo

Третье представление помогает узнать кол-во кросовок на складе, их полную цену и среднюю. Для суммы используется SUM, для округления ROUND, кол-во уникальных продуктов подсчитывается через DISCTINCT и COUNT.

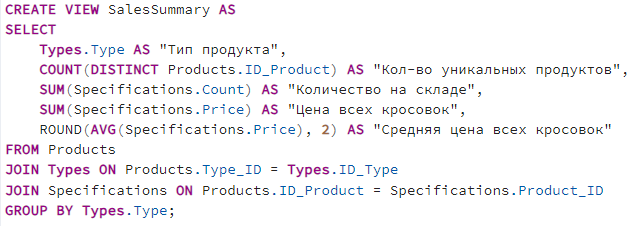


Рисунок 7 - SalesSummary

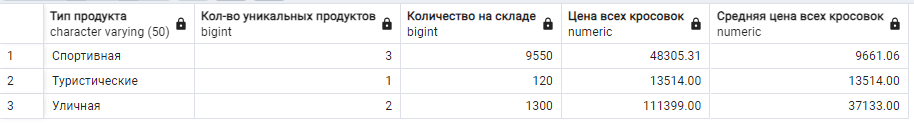


Рисунок 8 - вывод SalesSummary

Создаем первую процедуру, которой передается параметр название фирмы кроссовка. Процедура проверяет не NULL ли значение, а также есть ли уже такое в таблице. После всех проверок фирма добавится в таблицу.

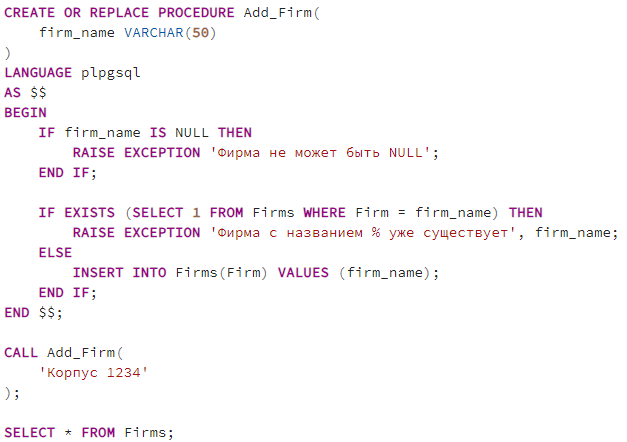


Рисунок 9 – Add\_Firm

Создаем вторую процедуру, которая обновляет название фирмы. Теперь она проверяет есть ли такая же фирма на которую хотят изменить название, после проверки изменяет.

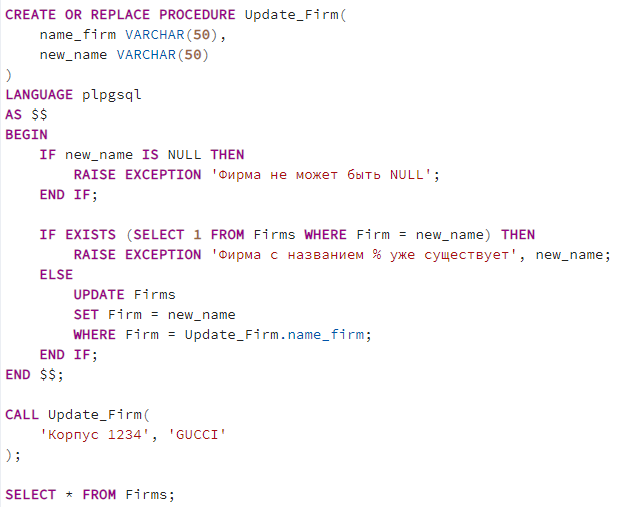


Рисунок 10 - Update\_firm

Создаем третью процедуру, которая удаляет фирму.

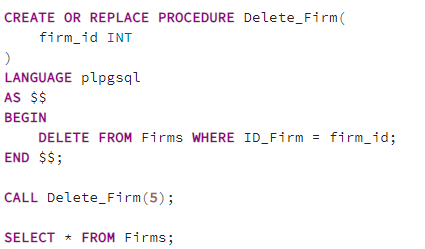


Рисунок 11 – Delete\_Firm

Создаем первую функцию, которая считает среднюю цену по типу кроссовок через SELECT AVG по столбцу с ценой.

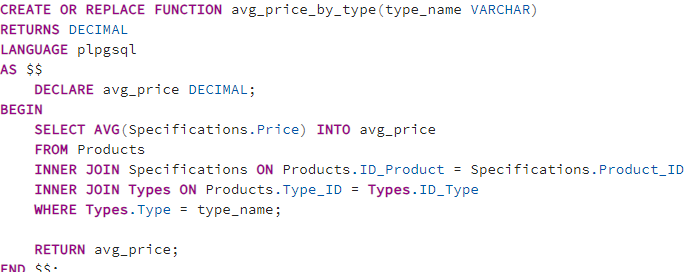


Рисунок 12 - avg\_price\_by\_type

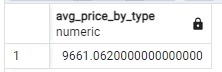


Рисунок 13 – средняя цена для спортивной обуви

Создаем вторую функцию, которой передается название фирмы и название корпуса, после чего та выводит кол-во моделей с этой фирмой, которое производит этот корпус.

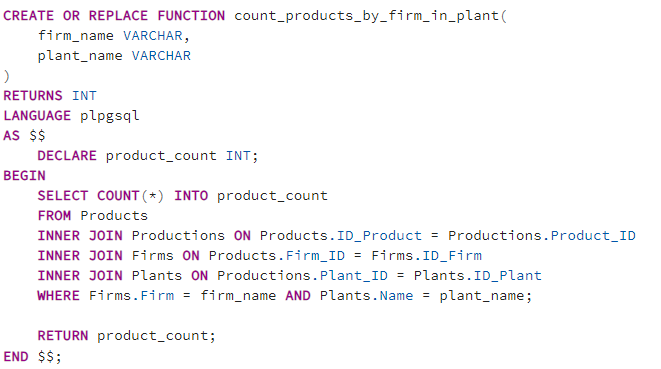


Рисунок 14 - count\_products\_by\_firm\_in\_plant

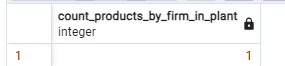


Рисунок 15 - производство моделей Nike 1 корпусом

Создаем третью функцию, которой передается пол и название корпуса, после чего функция вернет сколько рабочих того или иного пола трудится в этом корпусе.

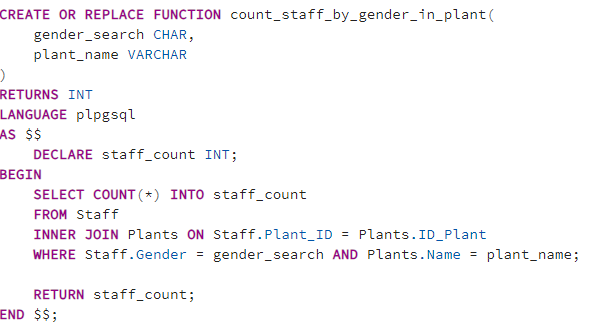


Рисунок 16 - count\_staff\_by\_gender\_in\_plant



Рисунок 17 - две девушки работают в 1 корпусе

Создается первый триггер, который при добавлении в таблицу Types будет автоматически вычислять среднюю цену продуктов данного типа.

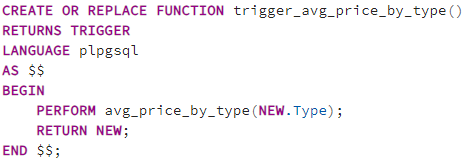


Рисунок 18 - первый триггер

Создадим второй триггер, который при добавлении новой записи в Products будет автоматически подсчитает количество продуктов данной фирмы в указанном корпусе.

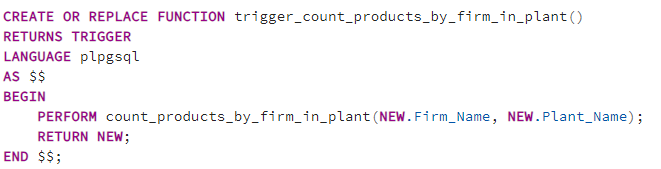


Рисунок 19 – второй триггер

Создается третий триггер, который реагирует на добавление в Staff, после чего автоматически подсчитает количество сотрудников данного пола в указанной плантации.

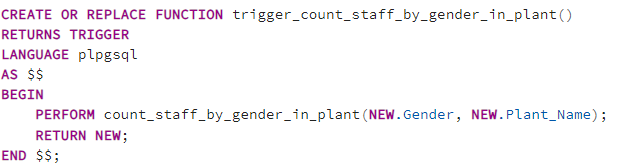


Рисунок 20 – третий триггер

Создаем два пользователя и зададим пароль.



Рисунок 21 - login

Создадим роль (группу) Base (базовая-группа) и добавим в нее наших пользователей.



Рисунок 22 - роль

Выдадим права базовой группе через GRANT, где разрешим данной группе делать все (SCHEMA). А для секретарши выдадим только селект.



Рисунок 23 - выдача прав

Оператор LIKE использует только «\_» и «%». Нижнее подчеркивание «бронирует» символ. А процент говорит о том, что все что после него – допустимо. Поэтому достаточно применить их, для того чтобы показать весь функционал LIKE. Данный SELECT выведет модели, у которых вторая буква – T.

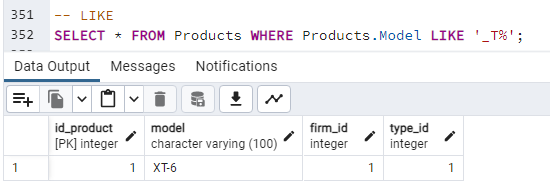


Рисунок 24 - LIKE

Начнем экспорт данных с настройки. Указываем какую таблицу экспортируем и куда.

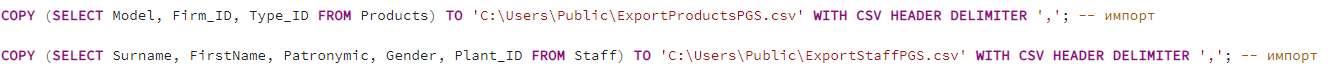


Рисунок 25 - экспортируем

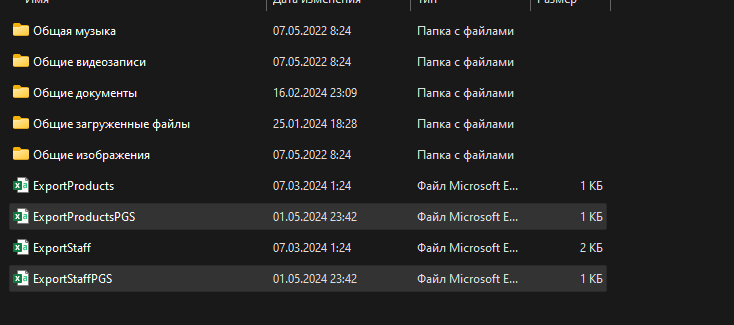


Рисунок 26 - файлы

После чего импортируем данные в эти же таблицы.



Рисунок 27 - импорт

Теперь если вывести таблицу Products, то можно увидеть, что она продублировалась.

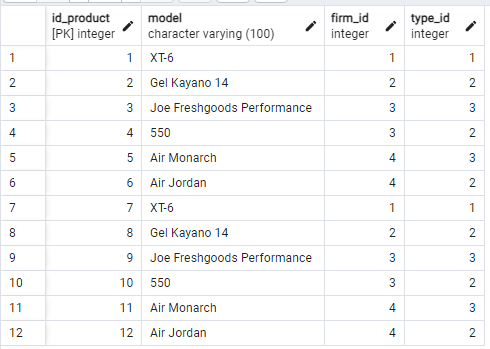


Рисунок 28 - дублирование после экспорта

Создадим бэкап в дефолтную папку, через консоль, предварительно разрешив права и все остальное. После ввода команд консоль запросит пароль, вводит везде «1».

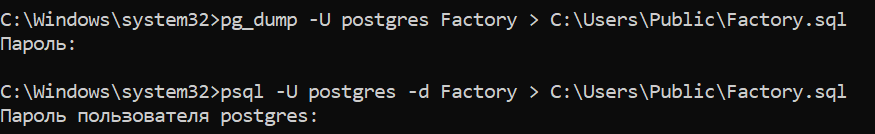


Рисунок - бэкап

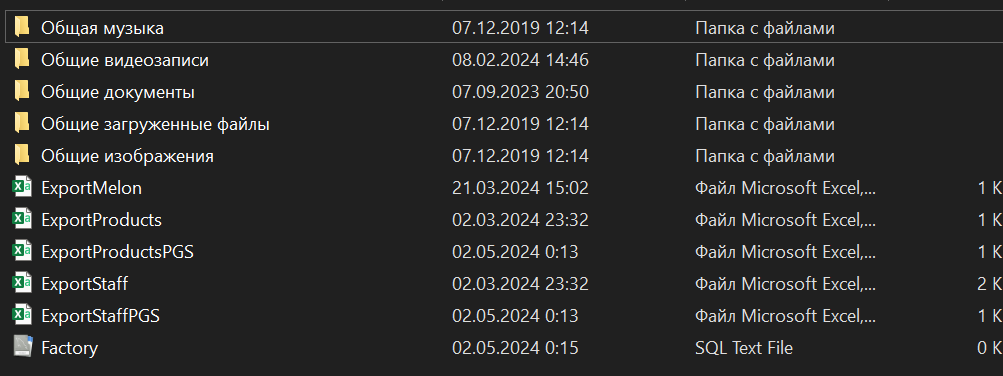


Рисунок 30 - файлы бэкапа в дефолтной папке

Вывод**:** Было изучено как создавать на диалекте POSTGRES различные процедуры, функции, триггеры, представления, пользователей и их группы. Был освоен навык экспорта и импорта данных из таблиц, а также навык создания бэкапы и выполнения рестора БД.