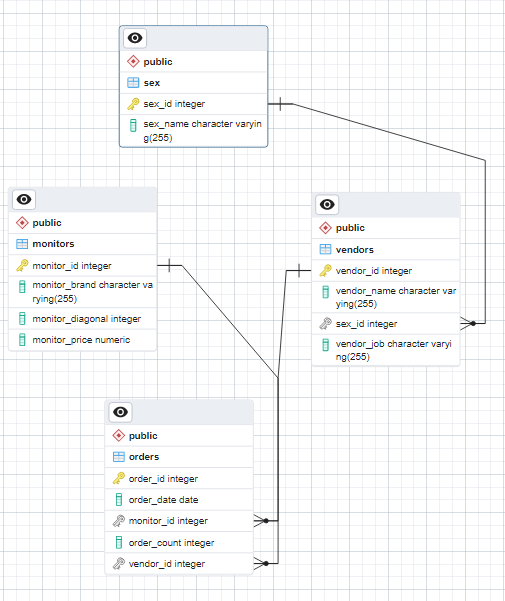
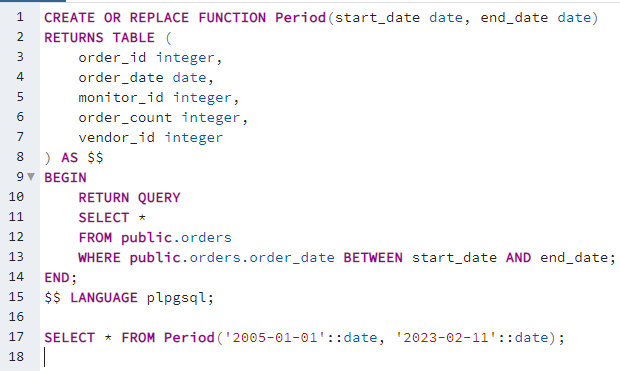
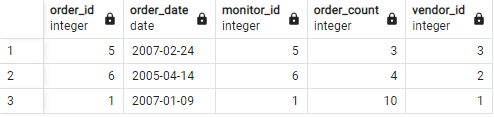
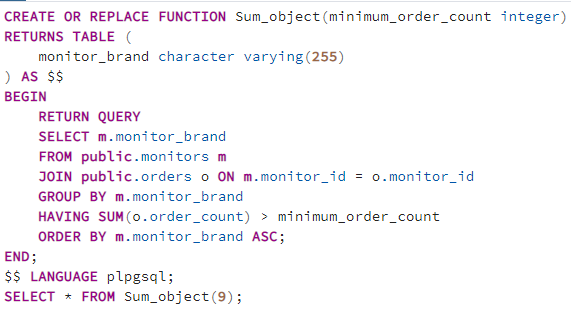
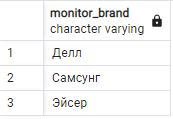
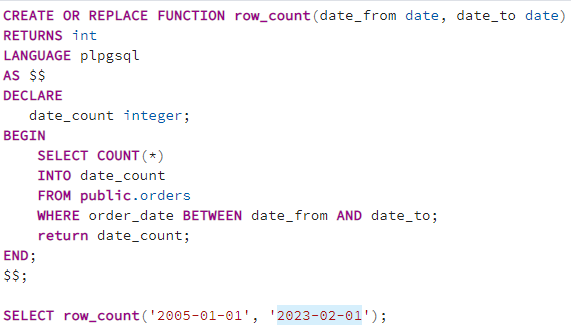
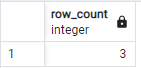
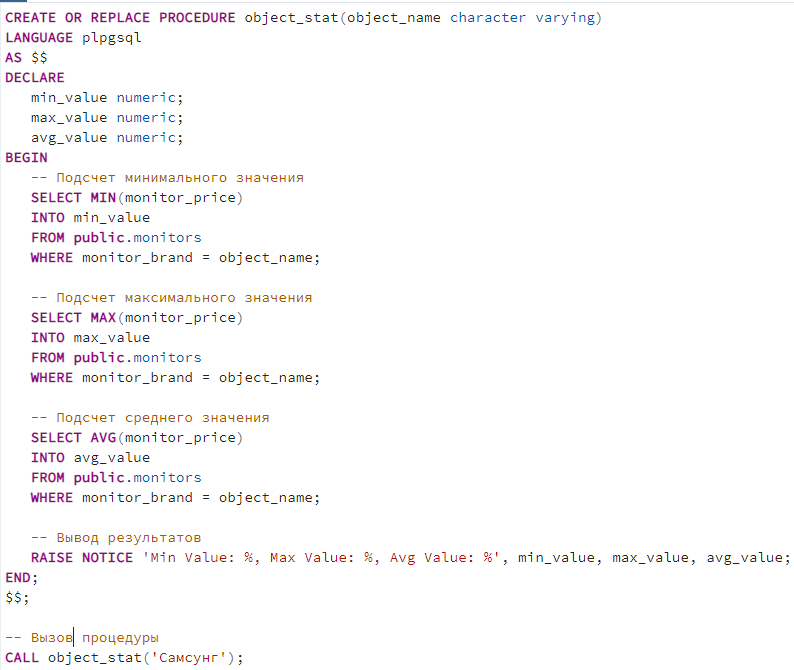
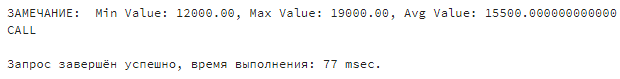
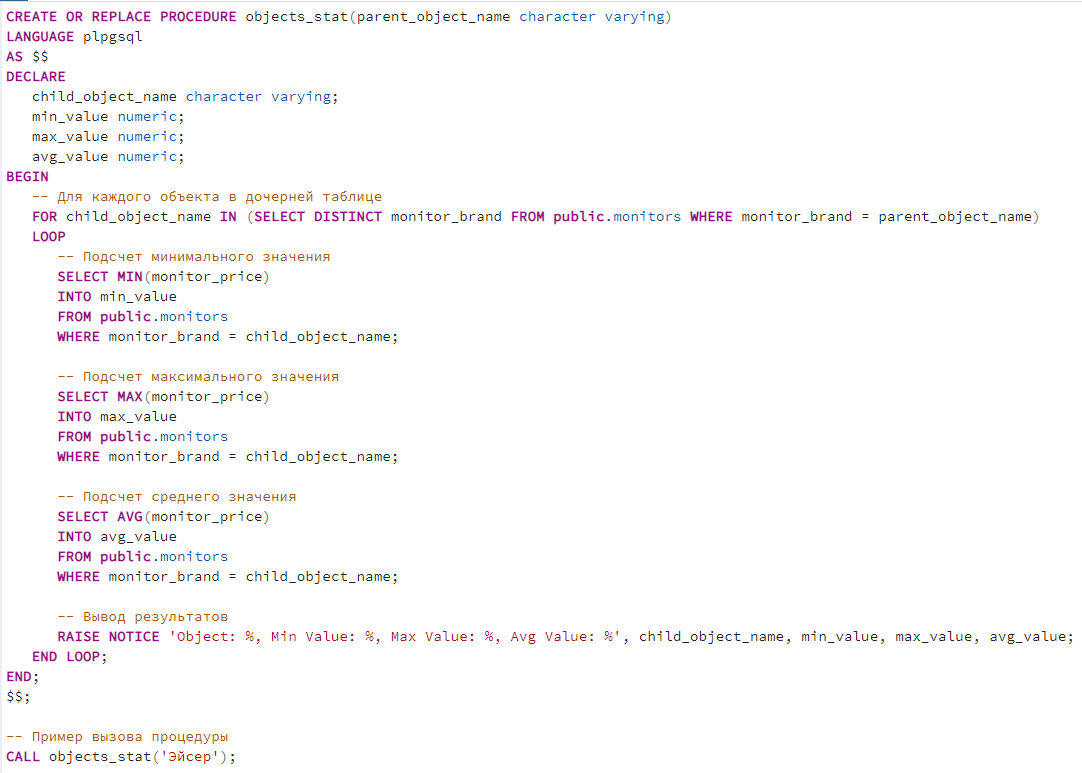
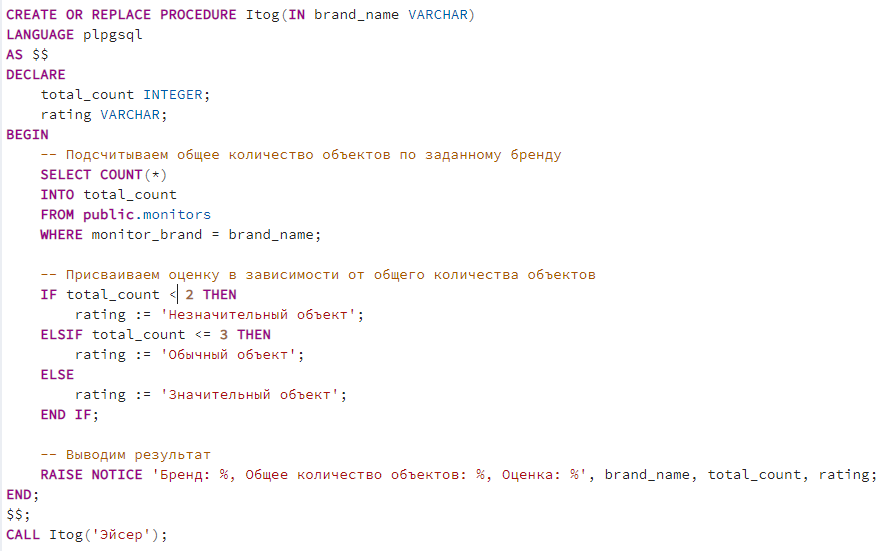
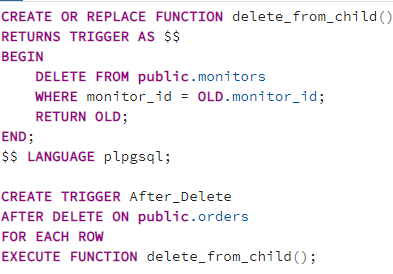
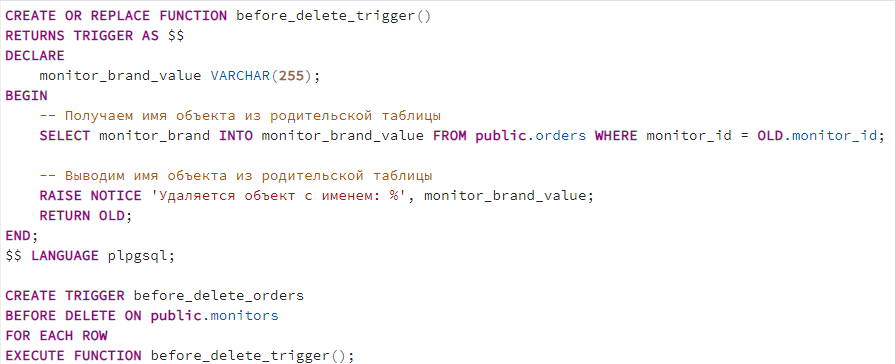
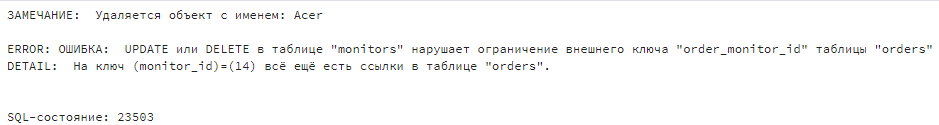
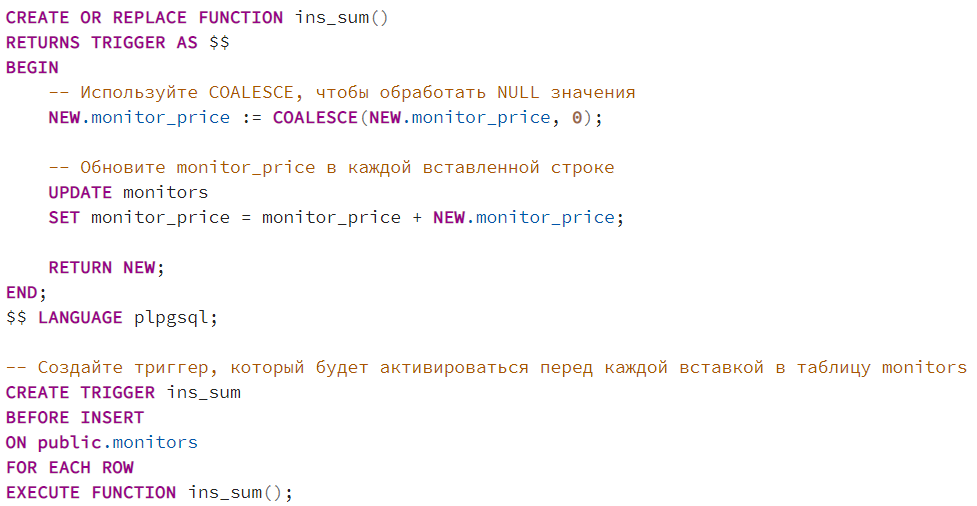
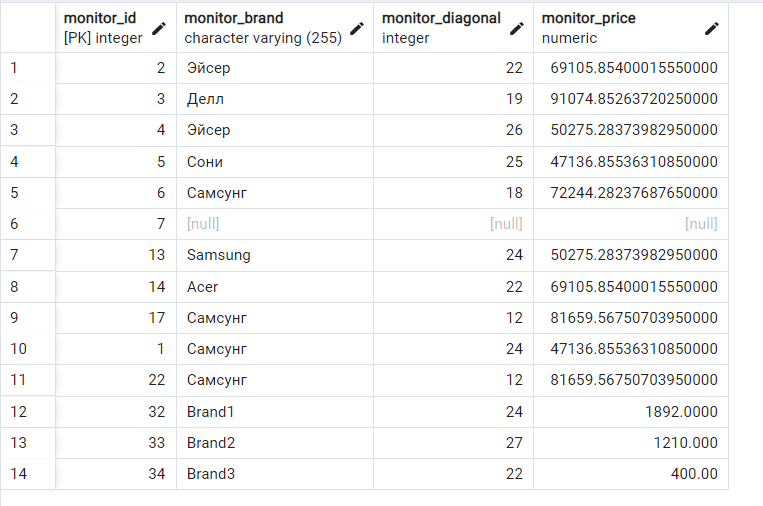
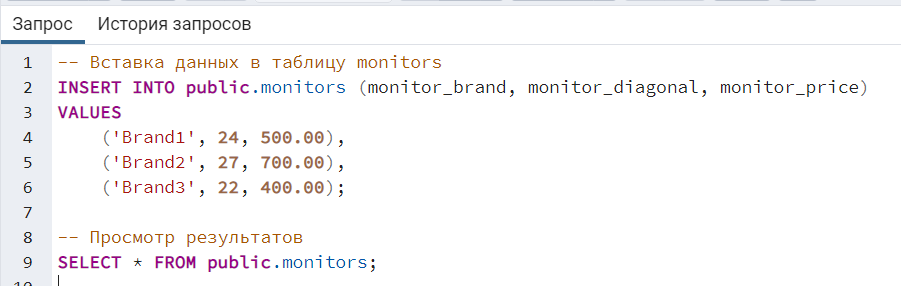
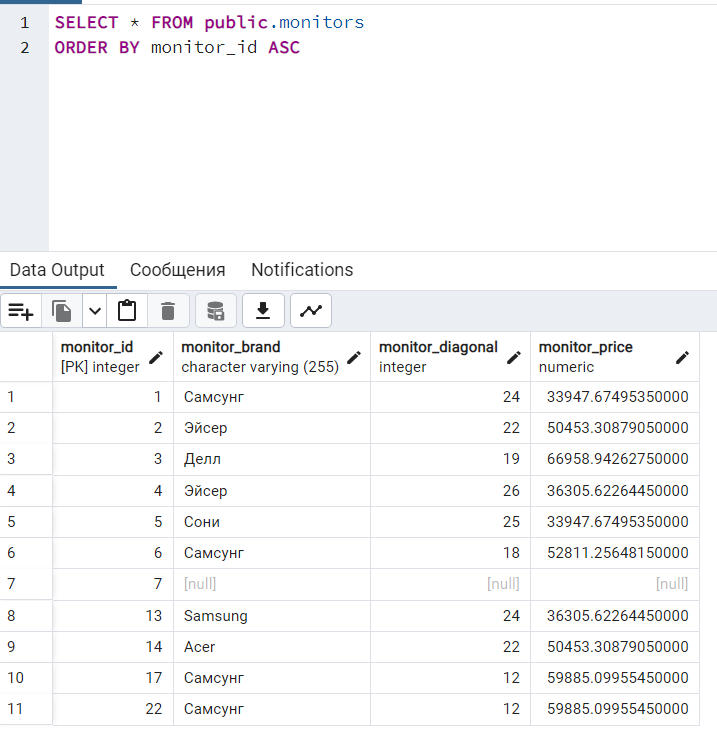
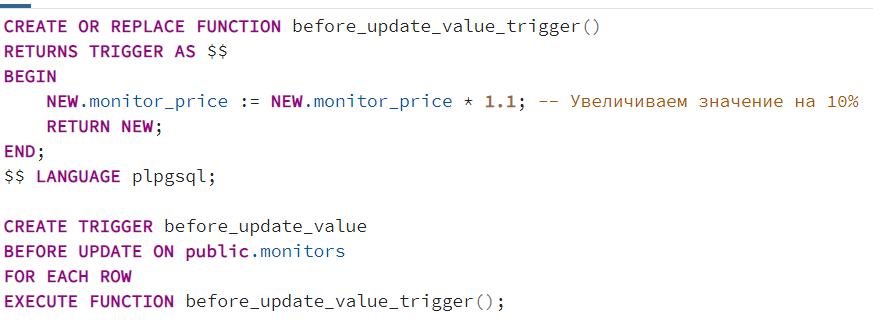
Отчет по лабораторной работе 5

ERD:  


1. Создать функцию Period(…, …) c двумя входными параметрами типа date, которая выберет строки из дочерней таблицы в диапазоне дат, указанных первым и вторым аргументами при вызове функции Period(…, …)  
   Запрос: Вывести данные о покупках совершенных с 2005-01-01 по 2023-02-11.  
   SQL: Данные:  
   
2. Создать функцию Sum\_object(…)(с одним параметром), которая возвращает список имен объектов из родительской таблицы на основании данных дочерней таблицы. Список объектов определяется значением параметра, исходя из условия, что суммарное количество объектов должно быть больше, чем заданное значение в параметре.  
   Запрос: вывести список имен объектов из таблицы monitors, для которых суммарное количество объектов в таблице orders больше 9.  
   SQL: Данные:  
   
3. Создать функцию row\_count(…), которая подсчитывает количество строк дочерней таблицы, даты которых находятся между параметрами date\_from и date\_to.  
   Запрос: отображает количество заказов(строк), которые были сделаны между 2005-01-01 и 2023-02-01.  
   SQL: Данные:  
   
4. Создать хранимую процедуру object\_stat(…), которая подсчитывает минимальное, максимальное и среднее значение объектов в дочерней таблице, входным параметром является имя объекта.   
   Запрос: Вывести минимальную, максимальную, среднюю цену на мониторы Самсунг.  
   SQL:  
     
   Данные:  
   
5. Создать хранимую процедуру objects\_stat(…), которая подсчитывает минимальное, максимальное и среднее значение каждого объекта в дочерней таблице и выводит имя объекта, входным параметром является имя объекта из родительской таблицы.  
   Запрос: Мониторы с диагональю больше 20 и затем исключает мониторы с брендом Эйсер. Результатом будет список мониторов, удовлетворяющих первому условию, но не удовлетворяющих второму условию.  
   SQL:  
     
   Данные:  
   
6. Создать хранимую процедуру Itog(…) с одним входным параметром, которая выводит наименование объекта по суммарному количеству объектов:   
   а) Оценка «Незначительный объект», если число объектов меньше 2   
   б) Оценка «Обычный объект», если число объектов больше 2 и меньше или равно 3   
   с) Оценка «Значительный объект», если число объектов больше 3  
   Запрос: Выводим для бренда Эйсер оценку по количеству уникальных мониторов.  
   SQL:   
     
   Данные:  
   
7. Создать триггер After\_Delete, который при удалении записи из родительской таблицы удалял бы все связанные записи из дочерней таблицы. Показать результат работы триггера.  
   Запрос: Создает тригер, который реагирует на удаление записи из таблицы заказов путем удаления монитора из таблицы Мониторы.  
   SQL:  
     
   Данные:  
   
8. Создать триггер Before\_Delete, который при удалении записи из дочерней таблицы выводил бы имя объекта родительской таблицы. Показать результат работы триггера.  
   Запрос: Создает Тригер, который реагирует на удаление записи из таблицы Мониторы  
   SQL:  
     
   Данные:   
   
9. Создать триггер ins\_sum, который связывает триггер с таблицей для инструкций INSERT. Это действует как сумматор, чтобы суммировать значения, вставленные в один из столбцов дочерней таблицы. Триггер должен активироваться перед каждой строкой, вставленной в таблицу. Показать результат работы триггера.  
   Запрос: Этот триггер обновляет столбец **цены** в каждой вставленной строке, прибавляя новое значение к существующему значению перед вставкой в таблицу **Мониторы**.  
   SQL:   
     
   Данные:  
   
10. Создать триггер Before\_Update\_Value на событие UPDATE, который увеличивает значение числового поля дочерней таблицы на 10%. Показать результат работы триггера.  
    Запрос: создать тригер на обновление поля цены монитора в дочерней таблицы Мониторов, так, чтобы цена увеличилась на 10%.  
    SQL:  
      
    Данные:  
      
    До   
      
    После  
    