

**Занятие 10. Программные блоки**

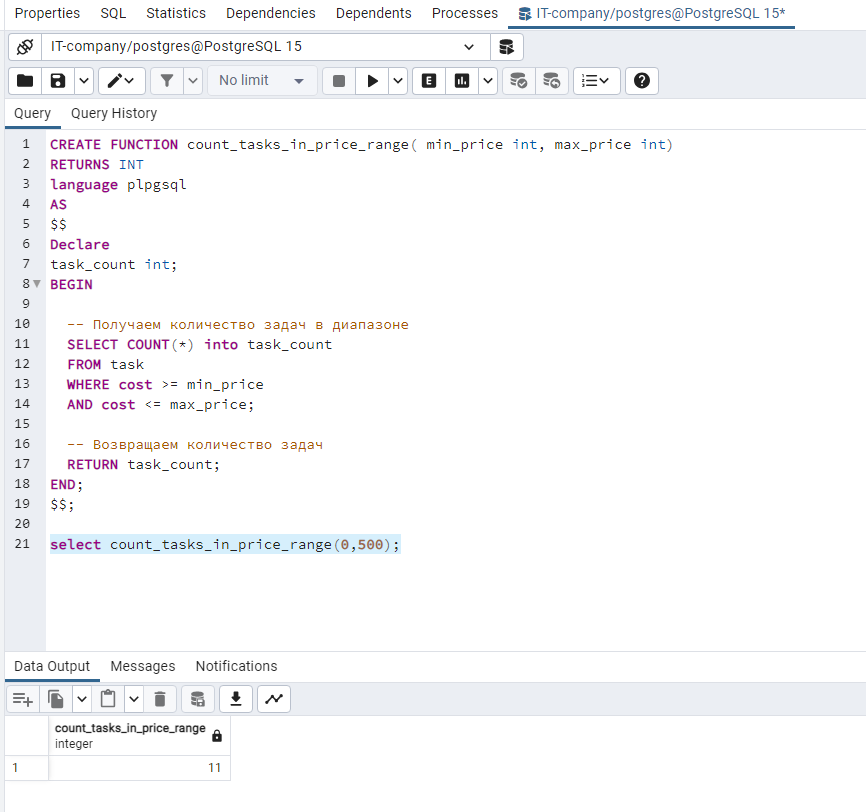
**Практическое задание**

Практическое задание сдается в виде отчета, в котором для каждого задания, подразумевающего практическую работу с экземпляром Postgres, приводятся снимки экрана, на которых видны фрагменты соответствующих экранных форм, комментированные исходные коды запросов, а также результатов их выполнения на экземпляре. Также сдаётся текстовый файл \*.sql, в который включены комментированные исходные коды всех запросов по всему заданию. В случае, если задание требует использования возможности, поддержка которой в Postgres (или используемой версии Postgres) отсутствует, то необходимо отразить это в отчете и использовать альтернативный путь решения задачи.

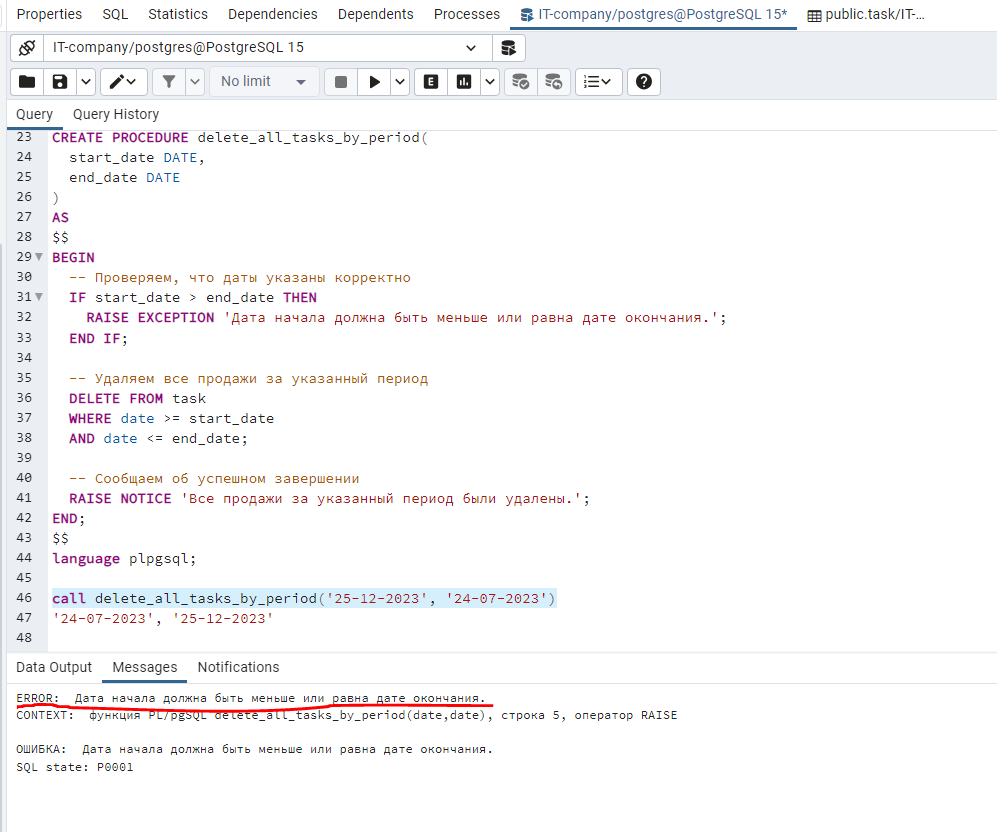
В качестве «полигона» используйте таблицы (если необходимо), созданные в рамках подготовки ДЗ с одного из предыдущих занятий.

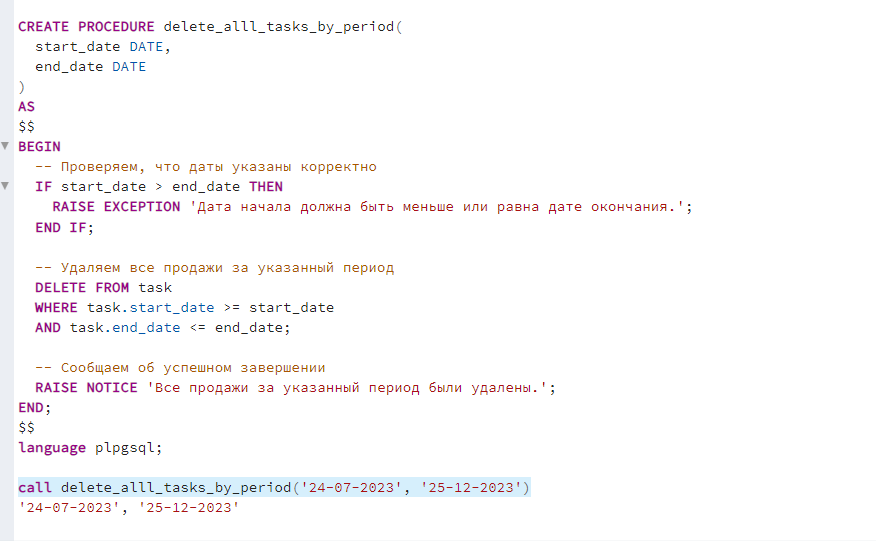
1. На уровне схемы реализуйте функцию, которая возвращает некий статистический показатель, основанный на существующих данных в таблицах. Это может быть, например, некий «коэффициент доступности» услуги, который вычисляется как отношение количества свободных слотов времени всех мастеров (за указанный период) к количеству оказаний услуг (за указанный период). Или, например, «коэффициент рентабельности», вычисляющийся как отношение суммарной выручки (за указанный период) к общему фонду оплаты труда продавцов (за тот же период) в процентах. Период и прочие необходимые данные (кроме тех, что рассчитываются по данным в таблицах) передаются в функцию в качестве входных параметров.

рассчитаем функцию, которая будет рассчитывать количество задач в определенном ценовом диапазоне



1. Реализуйте процедуру, удаляющие все продажи за указанный период. Продумайте и обработайте исключительные ситуации (exceptions).





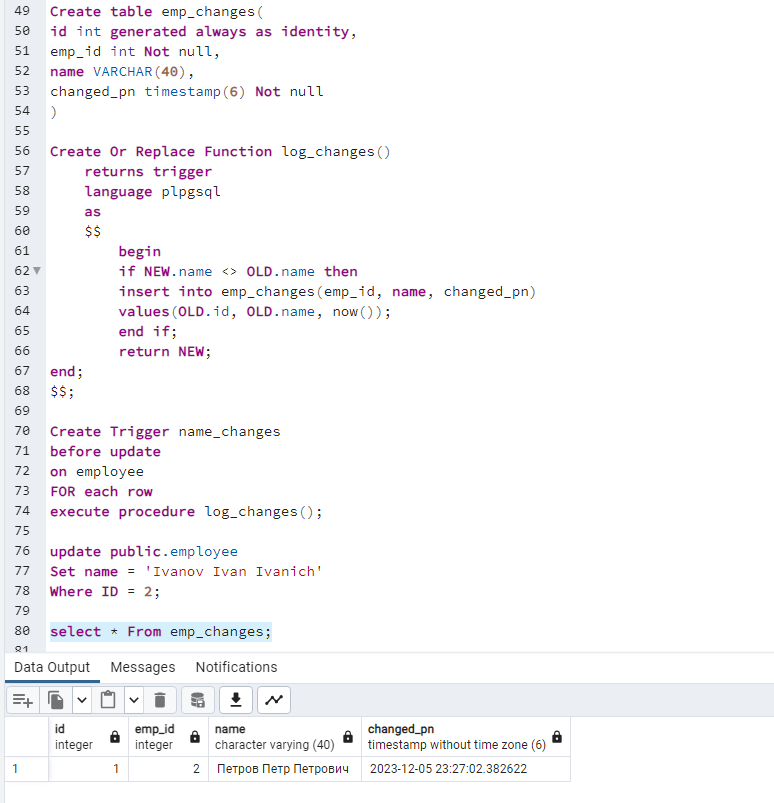
1. FO (или таблицах). Например, это может быть триггер, который при регистрации новой продажи проверяет наличие необходимого товара на складе (в таблице «товары») и если количества достаточно, то производит его актуализацию (вычитает количество проданного). Если же количества товара для данной регистрируемой продажи недостаточно, то генерируется исключение и вся транзакция откатывается.

Создаем таблицу с изменениями работников, в которую будут вноситься фамилии в конкретное время

Функция вставляет старую фамилию в таблицу emp\_changes, включая идентификатор сотрудника, имя и время изменения, если имя сотрудника меняется.

Old представляет строку до обновления, в то время как NEW представляет новую строку, которая будет обновлена. OLD.name возвращает имя до обновления, а NEW.name возвращает новую имя.

привязываем триггерную функцию к таблице mechanic Имя триггера - name\_changes. Перед обновлением значения столбца name автоматически вызывается функция запуска для регистрации изменений.



1. \* Самостоятельно посмотрите, что такое compound-триггеры и для чего нужны. Попробуйте реализовать такой триггер.

**составные триггеры не поддерживаются в postgres**

1. Немного отдохните. Это сложная и интересная тема. Вы проделали большую работу ☺ Но мы, как обычно, затронули только лишь верхушку. Курс близится к завершению…