Саратовский государственный технический университет им.

Гагарина Ю.А.

Институт прикладных информационных технологий и коммуникаций

Кафедра прикладные информационные технологии

Практическая работа №7

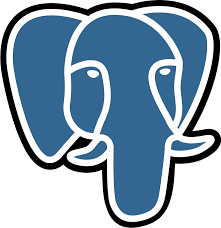
Выполнил студент 3 курса

Группы б1ИФСТ-31

Песчанов Арсений Андреевич

Преподаватель: Кузьмин Алексей Константинович

Саратов 2023



**Занятие 7. Физическая организация таблиц и индексов. Секционирование**

**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. Попробуйте реализовать различные виды секционирования таблиц, рассмотренных на лекции (списочное, диапазонное, по хешу) и не рассмотренных на лекции (интервальное, ссылочное, системное). При этом постарайтесь сделать так, чтобы выбор способа и критерия секционирования был обоснован с точки зрения производительности и/или обслуживания.

В PostgreSQL существует два типа секционных таблиц:

* Секционные таблицы по списку. Секционные таблицы по списку делятся на подтаблицы на основе списка значений столбца.
* Секционные таблицы по диапазону. Секционные таблицы по диапазону делятся на подтаблицы на основе диапазона значений столбца.

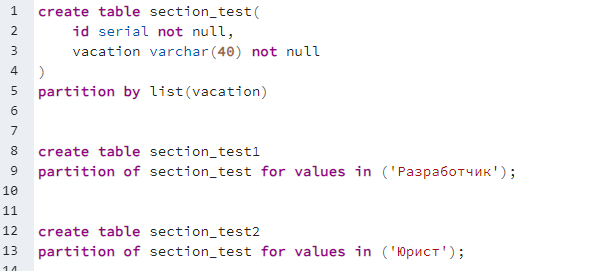
Секционные таблицы имеют следующие преимущества:

* Улучшенная производительность. Секционные таблицы могут улучшить производительность запросов, которые обращаются к подмножеству данных.
* Улучшенная масштабируемость. Секционные таблицы могут улучшить масштабируемость таблиц, которые содержат большое количество данных.

Секционные таблицы имеют следующие недостатки:

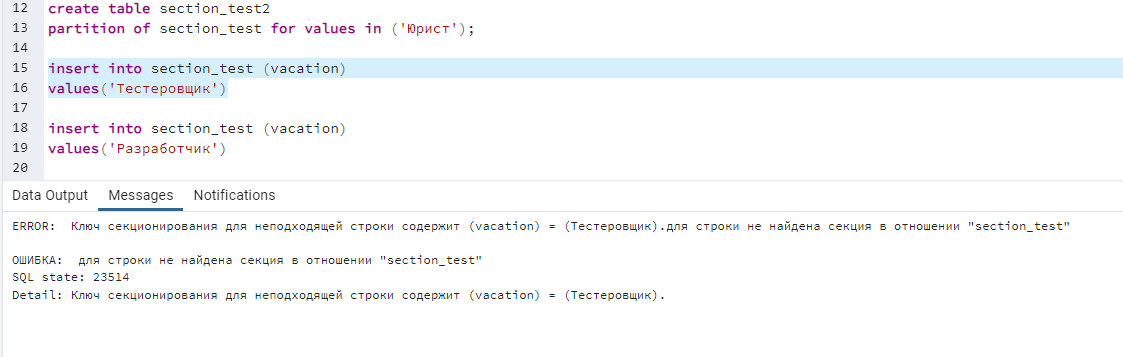
* Дополнительная сложность. Секционные таблицы могут усложнить создание и управление таблицами.
* Дополнительные затраты памяти. Секционные таблицы могут потребовать дополнительных затрат памяти для хранения метаданных.

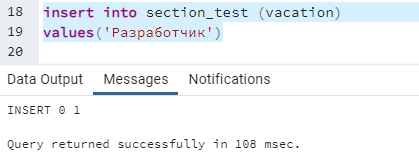
1. **Списочное создаем таблицу для работников и делаем разбиение по секциям их должностей**



Partition by list - это один из типов partitioning в PostgreSQL. С помощью partition by list таблица делится на несколько подтаблиц, называемых разделами, на основе списка значений столбца.

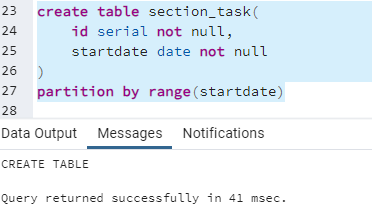
Пробуем добавить новых работников



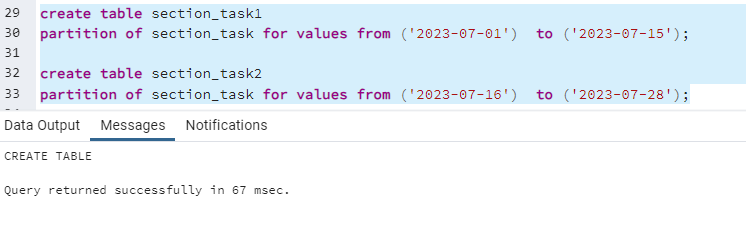


1. **диапазонное**

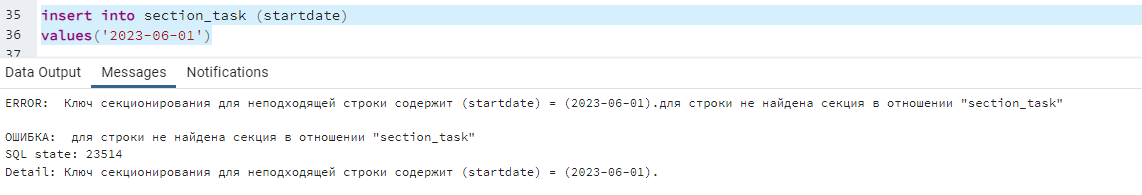
создаем таблицу

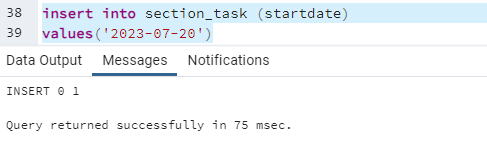


Partition by range - это один из типов partitioning в PostgreSQL. С помощью partition by range таблица делится на несколько подтаблиц, называемых разделами, на основе диапазона значений столбца.



добавляем значения





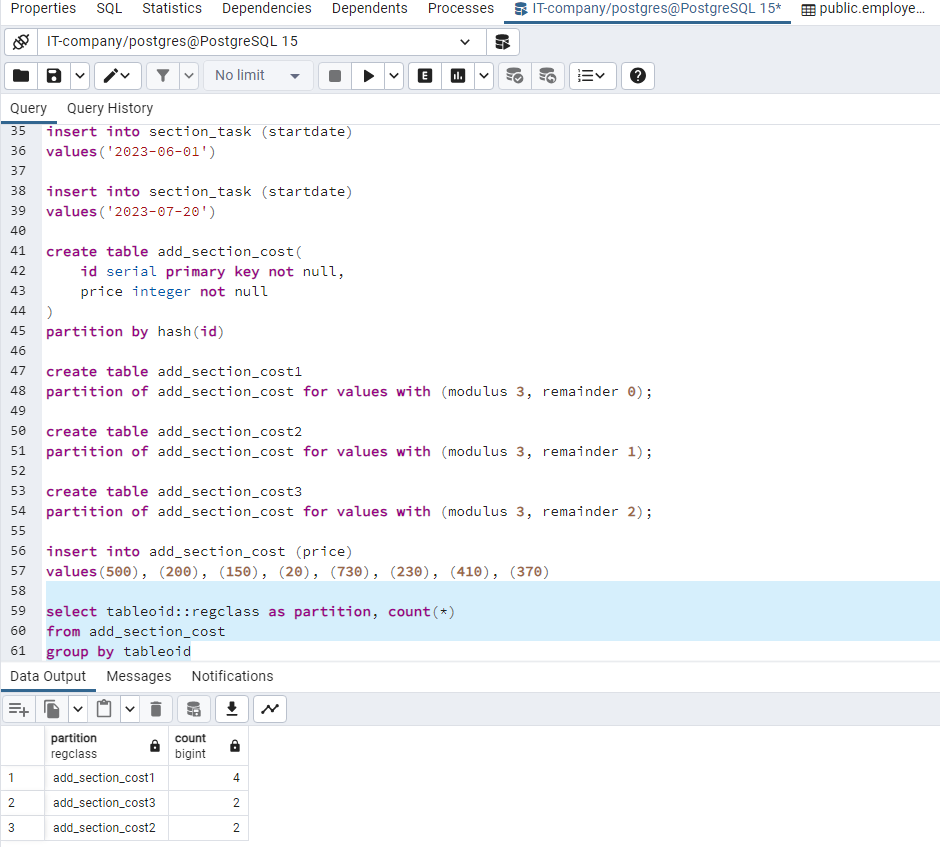
Получаем различные секции по разным промежуткам, например, по дате выполнения той или иной услуги.

1. **по хешу**

в таком варианте id - ключ секционирвоания, его сразу можно объявлять первичным ключом

tableoid - это уникальный идентификатор таблицы. Он представляет собой целое число, которое присваивается таблице при создании.

Tableoid и regclass тесно связаны друг с другом. Каждый regclass содержит tableoid таблицы, на которую он ссылается. Tableoid таблицы можно получить из regclass, используя оператор regclass().



Modulus 3 означает, что данные будут разделены на основе остатка от деления значения столбца на 3. А remainder 0(1,2) указывает, что данные с остатком 0 будут помещены в этот раздел.

1. Немного отдохните. Это сложная и интересная тема. Вы проделали большую работу ☺