

Лабораторная работа №1-4: «Переход на PostgreSQL»

Введение

Выбор между компактной, встраиваемой СУБД и большим сервером СУБД зависит от количества клиентов базы данных, сложности запросов и количества хранящихся в БД сведений. При разработке встраиваемых баз данных предпочтение отдаётся простоте и компактности, поэтому некоторые функции не реализуются.

Организация одновременного доступа нескольких пользователей к общим ресурсам (не только на чтение) всегда была одной из самых сложных задач в области компьютерных технологий. Добавление такой возможности несоизмеримо увеличивает сложность любого программного продукта. SQLite3 не поддерживает одновременную работу: она может только сериализовать доступ для нескольких пользователей, тем самым обеспечив целостность данных.

Многие другие функции тоже приносятся в жертву простоте и компактности реализации. Среди них встречаются неожиданные примеры. Одна из таких возможностей — COLLATION. Под этим термином подразумевается определение полного порядка на множестве символов. В контексте базы данных речь обычно идёт о текстовых символах в строке. COLLATION нужен для таких задач, как сортировка строк в алфавитном порядке.

Сложность этой задачи проявляется уже в тот момент, когда нужно верно разделить строчные и заглавные буквы. Затем, то, что в таблице ASCII порядок символов соответствует алфавитному — скорее случайность, чем закономерность. Для других языков и кодовых таблиц это часто не так. Наконец, последний удар по проблеме наносит сложность Unicode: один и тот же символ может быть записан несколькими способами. В итоге, для того, чтобы решить примитивную на первый взгляд проблему сортировки строк по алфавиту, приходится использовать специальные таблицы. Суммарный размер этих таблиц в разы превосходит размер всей библиотеки SQLite3: неудивительно, что эта СУБД не выполняет COLLATION, а при сортировке строк просто упорядочивает их побайтово).

Есть и более неожиданные сюрпризы. Например, SQLite3 не умеет вычислять медиану числового множества.

Поэтому чем крупнее организация, тем больше вероятность, что увеличенный размер, сложность и потребление ресурсов, связанные с эксплуатацией крупного сервера СУБД, окупятся. Ключевым фактором при принятии решения является наличие нескольких активных клиентов, затем наличие нужных функций. Требования заданий к лабораторным работам также могут сделать переход на более сложные модели СУБД неизбежным.

Важно помнить о том, что крупные клиент-серверные СУБД обладают своими недостатками в сравнении с мелкими встраиваемыми:

1. их движок не может быть интегрирован в прикладное программное обеспечение (но можно наоборот — интегрировать в них клиентскую часть). Так, приложения, написанные для PostgreSQL всегда будут требовать доступности сервера PostgreSQL, локально или по сети. Приложения, написанные с SQLite3, могут обслуживать сами себя;
2. принцип хранения данных на дисках более сложный, базу данных сложнее копировать и переносить с места на место;
3. сложные СУБД требуют установки в систему и дополнительной работы системного/сетевого администратора;
4. сложные СУБД требуют больше системных ресурсов, особенно при простое.

Важные в рамках данного курса отличия PostgreSQL:

1. PostgreSQL рассчитана на многопользовательский доступ. Отдельные учётные записи, характеризующие права пользователя при подключении к серверу, называются ролями (ROLES) (rem.: это отличается от понятия роли в Oracle Database). При работе с базой данных, управляемой PostgreSQL, изменения вносятся от лица той или иной роли. На данный момент можно ограничиться использованием роли по умолчанию, создаваемой при установке СУБД;
2. PostgreSQL (как и большинство РСУБД) жёстко типизирует данные. В отличие от SQLite3, она не допустит хранения в одном столбце данных разных типов, а также хранение данных, тип которых не соответствует типу столбца;
3. PostgreSQL — сервер системы управления базами данных. Клиентскую часть можно выбрать на свой вкус (но в ней должна быть поддержка протокола PostgreSQL). Сервер PostgreSQL при установке комплектуется стандартным клиентом: psql.

Цель работы

Изучить процедуру установки и первичной настройки сервера PostgreSQL. Изучить процедуру подключения к серверу PostgreSQL. Применить PostgreSQL и SQLite3.

Ход работы

1. Выполнить установку сервера PostgreSQL на реальную или виртуальную машину;
2. Выбрать клиент (psql,...) и осуществить подключение к установленному серверу;
3. Реализовать базу данных, аналогичную реализованной в рамках работ 1-1 ... 1–3. Попробовать повторно выполнить все запросы. Отразить в отчёте все моменты, в которых поведение PostgreSQL не соответствует поведению SQLite3 (если они будут);
4. Оформить отчёт.

Оформление отчёта

1. Титульный лист: название института, название лабораторной работы, имя, фамилия, номер группы, год,...
2. Список выполненных шагов настройки и запросов SQL с описаниями их назначения и смысла;
3. Приложение: Листинг использованных инструкций SQL. В тексте отчёта должна быть ссылка на приложение;
4. Заключение: краткое описание проделанной работы.

Справочные материалы

1. Install PostgreSQL on Windows. <https://www.postgresqltutorial.com/postgresql-getting-started/install-postgresql/> Пошаговое руководство по установке РСУБД PostgreSQL.
2. PostgreSQL Tutorial. <https://www.postgresqltutorial.com/> Справочник по возможностям диалекта SQL, используемого PostgreSQL, с примерами.