

Лабораторная работа №2-3: «Базовые задачи администрирования СУБД»

Цель работы

Изучить базовые механизмы, предоставляемые СУБД для управления базами данных: создание, удаление, запуск, остановку.

Подготовка к работе

Тенденцией последних лет является переход к концепции “контейнерной” базы данных (по терминологии Oracle; у всех это называется по-своему). В 70-е, 80-е годы прошлого века производительности компьютеров с трудом хватало для обработки баз данных средних по размеру организаций. Но в наши дни возможности вычислительной техники возросли непропорционально объёмам обрабатываемой информации, и один компьютер может легко справиться с управлением несколькими средними или несколькими десятками-сотнями небольших баз данных. Можно запускать на одной машине несколько совершенно отдельных экземпляров СУБД, каждая из которых управляет своей базой данных. Но очевидно, что это породит множество идентичных процессов, которые не только избыточны, но и могут создавать дополнительные трудности при доступе к ресурсам компьютера (самыми важными из которых являются операции ввода-вывода и процессорное время).

Поэтому появление программной поддержки такого варианта использования в СУБД было вопросом времени. Oracle уже перешёл на контейнерные базы данных по умолчанию, — это «сломало» большинство существовавших на тот момент старых обучающих курсов по Oracle Database.

В PostgreSQL также можно создавать несколько изолированных баз данных, находящихся под управлением одной СУБД. В совокупности эти базы данных образуют «кластер». Важно отметить, что такое значение термин «кластер» имеет только для PostgreSQL, для других СУБД значение этого термина может в корне отличаться (например, в Oracle Database кластером называется множество таблиц, хранящееся заранее соединёнными по каким-то ключевым полям). Существует небольшое множество объектов БД и ролей (это удобно для администрирования и настройки), но основное множество (включая все пользовательские данные) хранятся в конкретной базе данных, и не существуют за её пределами. С точки зрения конечного пользователя отдельная база данных в кластере — полноценный экземпляр PostgreSQL.

Физически, кластер PostgreSQL — просто директория на диске, где хранятся служебные и пользовательские данные. Примером такой директории по умолчанию может служить `/usr/local/pgsql/data`. Для начала работы экземпляра PostgreSQL достаточно запустить демон, указав на эту директорию: `postgres -D /usr/local/pgsql/data`.

При установке по умолчанию создаётся кластер, состоящий из нескольких баз данных, основная называется `postgres`. Она считается «базой данных по умолчанию». Также создаются две базы данных: `template0` и `template1`. Эти базы данных — образцы (шаблоны) для создания новых баз данных. Если необходимо создать несколько новых баз данных с аналогичной структурой, можно создать эту структуру в `template1`, и потом создать любое количество баз данных. `template0` служит чистым, неиспорченным образцом, из которого можно восстановить испорченную, загрязнённую или удалённую `template1`.

Инструкция `SELECT datname FROM pg_database`; позволяет получить список баз данных, которые были созданы и управляются этим экземпляром PostgreSQL. Новую базу данных можно создать при помощи `CREATE DATABASE` (по умолчанию получается копия `template1`), удалить при помощи `DROP DATABASE`. При создании можно также указать «владельца» новой базы данных. Это пользова-

тель (вернее: роль), которая будет обладать административными привилегиями в новой БД. Ключевое слово: **OWNER**.

Остановка экземпляра PostgreSQL обычно производится путём отправки сигнала Linux базовому демону **postgres**. В зависимости от «жёсткости» сигнала, СУБД выполнит штатное или нештатное завершение работы. Важно, чтобы процесс **postgres** получил хоть какую-то возможность корректного завершения, т.к. для обработки пользовательских подключений создаются служебные процессы, — кто-то должен их завершить и освободить разделяемые ресурсы.

Вопросы для самоконтроля и самостоятельного изучения

1. Что такое «контейнерная база данных»?
2. Как объяснить тенденцию к внедрению контейнерных баз данных?
3. Что такое кластер PostgreSQL?
4. Как устроен кластер PostgreSQL по умолчанию после установки?
5. Зачем служит база данных **template0**?
6. Зачем служит база данных **template1**?
7. Что будет, если внести изменения в базу данных **template1**?
8. Можно ли (если можно, то как?) удалить базу данных?
9. Можно ли (если можно, то как?) удалить кластер PostgreSQL?
10. Как запустить экземпляр СУБД PostgreSQL?
11. Какие существуют способы остановки экземпляра СУБД PostgreSQL?
12. Каковы основные последствия отказа или немедленного завершения базового процесса PostgreSQL **postgres**?
13. Может ли в системе функционировать два демона **postgres**? Если да, то зачем это нужно? Если нет, то почему?
14. Можно ли создать новую базу данных в кластере, используя инструкции SQL? Как? Какие параметры можно указать?
15. Можно ли создать новую базу данных в кластере, не используя инструкции SQL? Как? Какие параметры можно указать?
16. Могут ли роли PostgreSQL существовать во всём кластере (в каждой из БД кластера)?
17. Могут ли роли PostgreSQL существовать в отдельной БД кластера, но быть невидимыми для остального кластера?
18. Кто является администратором вновь создаваемой базы данных?
19. Можно ли создать новый кластер баз данных, используя инструкции SQL? Как? Какие параметры можно указать?
20. Можно ли создать новый кластер баз данных, не используя инструкции SQL? Как? Какие параметры можно указать?
21. Как выяснить, к какой базе данных из кластера подключена текущая сессия пользователя?
22. Можно ли подключиться к другой базе данных в том же кластере, не прерывая текущей сессии? Если да, то как?
23. Каким образом надо подключиться к PostgreSQL, чтобы попасть в желаемую БД в кластере, а не в базу данных по умолчанию **postgres**?

Ход работы

1. Выяснить, в каком месте файловой системы расположен установленный в предыдущих работах кластер баз данных PostgreSQL;
2. Выяснить, какие файлы хранятся в директории кластера базы данных;
3. Выяснить, какой командной строкой запущен экземпляр PostgreSQL;
4. Выполнить штатное завершение работы сервера PostgreSQL;
5. Вновь запустить экземпляр PostgreSQL вручную;

6. Подключиться к экземпляру и проверить его работоспособность;
7. Создать новую базу данных в кластере. Кто её владелец? Какие объекты в ней содержатся?
8. Подключиться к новой базе данных и создать в ней несколько пробных объектов;
9. Убедиться, что из новой базы данных нет доступа к исходной;
10. Убедиться, что из исходной базы данных нет доступа к новой;
11. Оформить отчёт.

Оформление отчёта

1. Титульный лист: название института, название лабораторной работы, имя, фамилия, номер группы, год,...
2. Стенограмма всех сессий работы с СУБД с комментариями и ответами на поставленные в ходе работы вопросы;
3. Заключение: краткое описание проделанной работы;