Содержание

Лабораторная работа №4
Аутентификация и управление пользователями в СУБД Oracle

Цель задания

Целью выполнения лабораторной работы является обучение методам и средствам аутентификации и управления пользователями в СУБД *Oracle*.

Краткие теоретические сведения

Санкционированный (разрешенный) доступ к ресурсам является одним из важнейших механизмов обеспечения безопасности в распределенных базах данных.

Доступ к ресурсам предусматривает выполнение трех процедур: идентификации, аутентификации и авторизации.

Идентификация - назначение субъектам, объектам, процессам уникальных имен (идентификаторов).

Аутентификация – проверка и подтверждение подлинности идентифицированного субъекта, объекта, процесса.

Авторизация – это определение набора возможных операций с данными, которые может осуществлять пользователь.

В СУБД *Oracle* реализована поддержка принципа безопасности по умолчанию или принципа минимальных привилегий. Суть принципа состоит в том, что пользователь может получить доступ к объекту СУБД (например, таблице или представлению) или выполнить определенные действия в системе (например, создать новую таблицу или нового пользователя) только если это явно разрешено. Поэтому пользователь, успешно прошедший аутентификацию, по сути ничего не может делать до тех пор, пока уполномоченный администратор не определит перечень возможных для данного пользователя операций. В частности, корректно созданный пользователь после успешной аутентификации (ввода

правильного пароля) не сможет даже присоединиться к серверу баз данных, т.е. успешно выполнить команду *CONNECT*.

Каждый пользователь СУБД *Oracle* должен иметь специальный идентификатор: имя или точку входа. Создание нового идентификатора осуществляется уполномоченным пользователем или администратором выполнением предложения *CREATE USER.*

С позиций системы источники, предъявившие идентификатор, неразличимы. То есть, хотя пользователем может быть как реальный человек, сидящий за терминалом, так и прикладной процесс, для системы оба объекта тождественны.

Когда пользователь пытается подключиться к СУБД, сервер СУБД *Oracle* обеспечивает выполнение стандартной процедуры подтверждения подлинности или аутентификации. Обычно для подтверждения подлинности пользователь должен ввести пароль.

СУБД *Oracle* предлагает четыре метода аутентификации – аутентификация на уровне ОС, аутентификация на уровне сети, аутентификация на уровне СУБД и многоуровневая аутентификация.

Система аутентификации пользователей в СУБД *Oracle* может состоять из любого набора предлагаемых СУБД *Oracle* методов аутентификации.

Внешняя аутентификация – аутентификация на уровне ОС (*Authentication by the Operating System*) и аутентификация на уровне сети (*Authentication by the Network*) – предполагают использование средств операционной системы или сетевых средств аутентификации. Тем самым контроль выносится за пределы средств управления паролями и идентификации пользователя СУБД *Oracle*, хотя пользователь будет по-прежнему опознаваться СУБД. Для этого типа регистрации пароль СУБД не требуется. При вынесении процедуры аутентификации на уровень ОС, сервер определяет идентификатор пользователя из информации сеанса его

работы с операционной системой и на основе этой информации разрешает или запрещает пользователю подключение.

Чтобы использовать эту опцию, необходимо установить в файле СУБД *init.ora* параметр *OS_AUTHENT_PREFIX*. Это укажет СУБД *Oracle,* что пользователь, имя которого имеет тот же префикс, должен рассматриваться как подлежащий внешней идентификации. Например, если для параметра *OS_AUTHENT_PREFIX* установлено значение *ops*$ и имеются два пользователя – *ops*$*Ivanov* и *Petrov*, то для системы СУБД *Oracle* не нужен пароль от пользователя *ops*$*Ivanov*, но нужен – от *Petrov.* Данным параметром может быть установлен любой желаемый префикс (включая нулевую строку). В таком случае указывается пустое значение в двойных кавычках. При этом для параметра *REMOTE_OS_AUTHENT* в файле *init*.ora должно быть установлено значение *true* (значение по умолчанию – *false*), чтобы система СУБД *Oracle* могла использовать имя пользователя из незащищенного соединения.

При использовании аутентификации на уровне СУБД (*Authentication by* СУБД *Oracle Database*), имена и пароли этом хранятся не в файлах операционной системы, а в СУБД *Oracle*. Применение этого метода повышает степень защиты данных, так как универсальный сервер данных СУБД *Oracle* обеспечивает более детализированное управление доступом, чем операционная система. Администратор конфигурирует именную область, которая состоит или из строк связей с удаленными базами данных, или из *SID* СУБД *Oracle* (идентификатор локальной СУБД *Oracle*). Кроме того, такая область может содержать роль из СУБД, которая доступна только пользователям базы, имеющим привилегию присваивать себе аутентификационную роль.

Идентификация в СУБД используется при первичной регистрации (создании учетной записи) пользователя и указании пароля. Этот подход вполне удовлетворителен для небольших групп пользователей и в тех случаях, когда отсутствуют другие средства обеспечения безопасности.

Для других типов идентификации вместо пароля необходимо использовать зарезервированное слово *external* (внешний).

Если в СУБД *Oracle* используется идентификация на уровне СУБД, система предоставляет возможность управления паролями.

Многоуровневой (*Multitier Authentication*), или промежуточной, аутентификацией называется регистрация программного обеспечения промежуточного уровня от своего имени для выполнения в СУБД каких-либо действий по поручению пользователя. Это позволяет создавать промежуточные приложения, использующие собственную схему аутентификации, например, с помощью сертификатов *X*509 или другого процесса однократной регистрации и регистрироваться от имени пользователя, не зная его пароля в СУБД. Хотя регистрация выполнена не от имени пользователя, а от имени промежуточного ПО, для СУБД он зарегистрирован.

Многоуровневая аутентификация обеспечивает одноразовую аутентификацию – на сервере приложений – и обеспечивает доступ сервера приложений ко всем необходимым базам данных от имени пользователя, не передавая пароли для каждой СУБД.

Соответствующий оператор *ALTER USER* имеет следующий базовый синтаксис:

ALTER USER <имя пользователя> *GRANT CONNECT THROUGH*

<промежуточный пользователь><, промежуточный пользователь>...

Это дает возможность пользователям, перечисленным в списке промежуточных пользователей, подключаться от имени указанного после *ALTER USER* пользователя. По умолчанию для этих пользователей будут устанавливаться все роли данного пользователя. Другая разновидность этого оператора:

ALTER USER <имя пользователя> *GRANT CONNECT THROUGH*

<промежуточный пользователь> *WITH NONE*;

позволяет промежуточной учетной записи подключаться от имени указанного пользователя, но только с ее базовыми привилегиями – роли включаться не будут. Кроме того, можно использовать:

ALTER USER <имя пользователя> *Grant Connect Through*

<промежуточный пользователь> *ROLE* имя_роли,имя_роли,...

или:

ALTER USER <имя пользователя> *Grant Connect Through*

<промежуточный пользователь> *ROLE ALL EXCEPT*
имя_роли,имя_роли,...

Два представленных выше оператора дают промежуточной учетной записи возможность подключаться в качестве пользователя, но при этом включены будут только определенные роли. Необязательно давать учетной записи сервера приложений все привилегии – достаточно предоставить роли, необходимые для выполнения его функций. По умолчанию сервер СУБД *Oracle* пытается включить все стандартные роли пользователя и роли *PUBLIC*. Вполне допустимо разрешить серверу приложений использовать только роль *HR* данного пользователя, и никакие другие прикладные роли этого пользователя.

Разумеется, можно и отобрать соответствующую привилегию:

ALTER USER <имя пользователя> *REVOKE CONNECT THROUGH*
<промежуточный пользователь><,промежуточный пользователь>...

Есть административное представление, *PROXY_USERS*, которое можно использовать для получения информации обо всех промежуточных учетных записях.

Синтаксис оператора *AUDIT* позволяет настроить аудит действий, выполняемых указанными промежуточными пользователями от имени некоторых или всех учетных записей:

AUDIT <действие> *BY* <промежуточный пользователь>,

<промежуточный пользователь> ... *ON BEHALF OF* <клиент>,<клиент>...;

или:

AUDIT <действие> *BY* <промежуточный пользователь>,

<промежуточный пользователь> *ON BEHALF OF ANY*;

Для **администраторов СУБД** требуется более надежная схема опознания, соответствующая привилегированному характеру их задач (например, закрытие и запуск СУБД). Дополнительное опознание может быть выполнено с помощью операционной системы и (или) файла пароля.

Если операционная система предоставляет способ объединения пользователей в группы (это возможно в таких операционных системах, как *Unix* и *NT*), то в документации к СУБД *Oracle* рекомендуется объединить администраторов СУБД в специальную группу. Это позволяет СУБД *Oracle* дополнительно проверять с помощью идентификатора группы, является ли пользователь администратором СУБД.

Файл пароля для администраторов СУБД необязателен и может быть установлен с помощью утилиты *ORAPWD*. Файл пароля ограничивает привилегии администрирования только тех пользователей, которые знают пароль и имеют специальные роли – *SYSOPER* и *SYSDBA.*

Роль *SYSOPER* предоставляет возможность выполнять операторы *STARTUP, SHUTDOWN, ALTER DATABASE OPEN/MOUNT, ALTER DATABASE BACKUP, ARCHIVE LOG* и *RECOVER*, а также включает привилегию *RESTRICTED SESSION*.

Роль *SYSDBA* включаются все системные привилегии с помощью опции *ADMIN OPTION*, а системная привилегия *SYSOPER* допускает привилегию *CREATE DATABASE* и привилегию восстановления через определенные промежутки времени.

Задание

1. Изучить возможности аутентификации и управления пользователями с помощью запуска нескольких сеансов соединения с СУБД (например, *SQL*Plus*). Один сеанс работает от имени

администратора СУБД (*SYSTEM*), другие – от имени пользователей, созданных администратором (на одном компьютере могут быть запущены несколько сеансов).

2. Создать пользователя *Ivanov* с паролем *SKY*. Обеспечить, чтобы объекты и временные сегменты, создаваемые пользователем *Ivanov*, не принадлежали табличному пространству *SYSTEM*. Обеспечить также пользователю *Ivanov* доступ к табличным пространствам *DATA*01 и *INDX*01 и возможность использования в них пространства размером до одного мегабайта для создания своих объектов. Для этого: назначить пользователю временное табличное пространство, табличное пространство по умолчанию и указать квоты на использование табличных пространств *DATA*01 и *INDX*01. Например:

   SQL> CREATE USER Ivanov
   2> *IDENTIFIED BY SKY*
   3> *DEFAULT TABLESPACE data*01
   4> *TEMPORARY TABLESPACE temp*
   5> *QUOTA* 1M *ON data*01
   6> *QUOTA* 1M *ON indx*01;
   SQL> GRANT create session TO Ivanov;

3. Создать пользователя *Petrov* с паролем *OCEAN*. Обеспечить, чтобы в табличном пространстве *SYSTEM* не было объектов и сегментов сортировки, создаваемых пользователем *Petrov*. Например:

   SQL> CREATE USER Petrov
   2> *IDENTIFIED BY OCEAN*
   3> *DEFAULT TABLESPACE data*01
   4> *TEMPORARY TABLESPACE temp*;

4. Скопировать таблицу *ORDERS* из схемы *SYSTEM* в схему пользователя *Petrov*. Прежде чем пользователь *Petrov* сможет

создавать объекты в своей схеме, ему необходимо предоставить квоту на его табличное пространство по умолчанию.

*SQL> ALTER USER Petrov QUOTA UNLIMITED ON data*01,

SQL> CREATE TABLE Petrov.orders AS

2> *SELECT * FROM system.orders*;

5. Вывести на экран информацию словаря данных о пользователях *Ivanov* и *Petrov*. Эту информацию можно получить, выполнив запрос к представлению *DBA_USERS*.

   SQL> SELECT username, default_tablespace,

   2> *temporary_tablespace*

   3> *FROM dba_users*

   4> *WHERE username IN* ('*Ivanov*', '*Petrov*');

   USERNAME DEFAULT_TABLESP TEMPORARY_TABLE

   ---------- ---------- ----------

   Petrov *DATA*01 *TEMP*

   Ivanov *DATA*01 *TEMP*

   2 *rows selected*.

6. Выведите на экран информацию словаря данных об объеме пространства в табличных пространствах, которое может использовать *Ivanov*. Эту информацию можно получить, выполнив запрос к представлению *DBA_TS_QUOTAS*.

   *SQL> SELECT * FROM dba_ts_quotas WHERE username = '*Ivanov';

   TABLESPACE_N USERNAME BYTES MAX_BYT BLOCKS MAX_BLO

   *INDX*01 *Ivanov* 0 1048576 0 512

   *DATA*01 *Ivanov* 0 1048576 0 512

   2 *rows selected*.

7. Как пользователь *Ivanov* произвести попытку изменить назначенное ему временное табличное пространство.

   SQL> CONNECT Ivanov/*SKY*;

   Connected.

SQL> ALTER USER Ivanov

2> *TEMPORARY TABLESPACE data*01;

ALTER USER Ivanov

ORA-01031: *insufficient privileges*

8. Как пользователь *Ivanov* изменить свой пароль на *SAM*.

   SQL> CONNECT Ivanov/*SKY*;

   Connected.

   SQL> ALTER USER Ivanov

   2> *IDENTIFIED BY sam*;

9. Как пользователь *SYSTEM* отменить для пользователя *Ivanov* квоту на его табличное пространство по умолчанию.

   SQL> CONNECT system/*manager*

   Connected.

   SQL> ALTER USER Ivanov QUOTA 0 *ON data*01;

10. Удалить пользователя СУБД *Petrov*. Так как пользователь *Petrov* является владельцем таблиц, нужно использовать режим *CASCADE*.

    SQL> DROP USER Petrov CASCADE;

11. Пользователь *Ivanov* забыл свой пароль. Назначить ему пароль *OLINK* и потребовать, чтобы *Ivanov* изменил пароль при следующем входе в систему.

    SQL> ALTER USER Ivanov

    2> *IDENTIFIED BY olink*

    3> *PASSWORD EXPIRE*;

12. Перехватить на рабочей станции *WS*2 данные, переданные при аутентификации пользователя. Попытаться найти в перехвате параметры аутентификации. Результаты перехвата привести в отчете.

13. Оформить отчет о выполнении лабораторной работы.

Содержание отчета о выполненной работе

1. Цель работы.
2. Описание комплекса программно-технических средств, использованного для выполнения лабораторной работы.
3. Результаты выполнения команд языка *SQL*.
4. Выводы по результатам выполнения лабораторной работы.

Контрольные вопросы по лабораторной работе № 4:

1. Дать определения следующим понятиям: идентификация, аутентификация, авторизация.
2. Описать 4 метода аутентификации.
3. Объяснить отличия пользователей и администраторов базы данных при авторизации.

Список дополнительной литературы

1. Технологии и средства консолидации информации: *Учебное пособие. Деревянко А.С., Солощук М.Н. - Харьков: НТУ "ХПИ", 2008. - 432с.*
2. Организация баз данных. 1 часть*: Курс лекций / Е.В. Сопченко, К.А. Кудрин.* Самарский гос. аэрокосмический ун-т. Самара, 2000, 71 с.
3. Сергей Кузнецов. Базы данных. Вводный курс. *www.cityforum.ru*
4. Сергей Кузнецов. Основы современных баз данных. *www.cityforum.ru*
5. Дейт К.Д. Введение в системы баз данных, 6-е издание. -М: Вильямс. 1999 г. -848 с.
6. Бобровски С. Oracle 8. Архитектура. – М: Издательство «Лори», 1998, 210 с.

7. Методические рекомендации к выполнению лабораторных работ по дисциплине «Серверные системы управления базами данных» для студентов специальности 230102 «Автоматизированные системы обработки информации и управления» всех форм обучения /*Сост.: М.В. Додонов, А.Ю. Павлов*. –Самара: СамГУПС, 2007. – 16 стр.