Федеральное государственное образовательное бюджетное учреждение
высшего образования
**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»**
Новороссийский филиал
Кафедра «Информатики, математики и общегуманитарные науки»

И.Г.Рзун

Методические рекомендации

Корпоративные информационные системы на базе ORACLE

Направление подготовки: 38.03.05 Бизнес-информатика
Направленность (профиль): ИТ- менеджмент в бизнесе
Форма обучения: очная/заочная/очно-заочная
Квалификация (степень) выпускника: бакалавр

Новороссийск 2019

Методические указания предназначены для методического сопровождения дисциплины «Корпоративные информационные системы на базе ORACLE». Цель методических рекомендаций - обеспечить обучающемуся оптимальную организацию процесса изучения дисциплины, а также выполнения различных форм контактной и самостоятельной работы. Методические указания содержат набор заданий с пояснениями и могут быть использованы для аудиторного или самостоятельного изучения.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ

1. Цели и задачи

Цель дисциплины — Целью дисциплины является ознакомление студентов с возможностью управления информационными потоками между всеми хозяйственными подразделениями (бизнес-функциями) внутри предприятия и информационной поддержки связей с другими предприятиями с помощью бизнес-приложений компании Oracle – Oracle e-Business Suite (OeBS).

Задачи дисциплины

Выработка способности к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям;

способность критически переосмысливать накопленный опыт, изменять при необходимости вид и характер своей профессиональной деятельности;

способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения.

Цель курса - изучение основ программирования и администрирования баз данных в СУБД Oracle в объеме, достаточном для выполнения разработки и сопровождения корпоративных информационных систем, включая настройку SQL.

Задачи курса - изучение архитектуры СУБД Oracle, освоение языка SQL и процедурного языка PL/SQL. Рассматривается динамический SQL и объектно-реляционная компонента Oracle

2. Место дисциплины в структуре основной образовательной программы.

Знания и умения, полученные в рамках данного курса, могут быть полезны при написании выпускной квалификационной работы и дальнейшего обучения в магистратуре и аспирантуре.

3. Методические указания и порядок изучения дисциплины.

Система обучения основывается на рациональном сочетании нескольких видов учебных занятий (в первую очередь, лекций и практических (лабораторных) занятий), работа на которых обладает определенной спецификой.

Подготовка к лекциям.

Знакомство с дисциплиной происходит уже на первом занятии, где требуется не просто внимание, но и самостоятельное оформление конспекта. Конспектирование лекций – сложный вид аудиторной работы, предполагающий интенсивную умственную деятельность студента. Конспект является полезным тогда, когда записано самое существенное. Не надо стремиться записать дословно всю лекцию. Такое «конспектирование» приносит больше вреда, чем пользы. Целесообразно вначале понять основную мысль, излагаемую лектором, а затем записать ее. Желательно запись осуществлять на одной странице листа или оставляя поля, на которых позднее, при самостоятельной работе с конспектом, можно сделать дополнительные записи, отметить непонятные места.

Конспект лекции лучше подразделять на пункты, соблюдая красную строку. Этому в большой степени будут способствовать вопросы плана лекции, предложенные преподавателям. Следует обращать внимание на акценты, выводы, которые делает лектор, отмечая наиболее важные моменты в лекционном материале замечаниями «важно», «хорошо запомнить» и т.п. Можно делать это и с помощью

разноцветных маркеров или ручек, подчеркивая термины и определения.

Работая над конспектом лекций, Вам всегда необходимо использовать не только учебник, но и ту литературу, которую дополнительно рекомендовал лектор. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть теоретическим материалом.

Подготовка к практическим (лабораторным) занятиям.

Подготовку к каждому практическому занятию необходимо начать с ознакомления с планом практического занятия, который отражает содержание предложенной темы. Тщательное продумывание и изучение вопросов плана основывается на проработке текущего материала лекции, а затем изучения обязательной и дополнительной литературы, рекомендованной к данной теме. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса.

Подготовка к лабораторным занятиям и практикумам носит различный характер, как по содержанию, так и по сложности исполнения. Проведение прямых и косвенных измерений предполагает детальное знание измерительных приборов, их возможностей, умение вносить своевременные поправки для получения более точных результатов. Многие лабораторные занятия требуют большой исследовательской работы, изучения дополнительной научной литературы.

В процессе подготовки к практическим занятиям, необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы. При всей полноте конспектирования лекции в ней невозможно изложить весь материал. Поэтому самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс

овладения информацией, способствует более глубокому усвоению изучаемого материала.

Защита лабораторных работ должна происходить, как правило, в часы, отведенные на лабораторные занятия. Студент может быть допущен к следующей лабораторной работе только в том случае, если у него не защищено не более двух предыдущих работ.

Рекомендации по работе с литературой.

Работу с литературой целесообразно начать с изучения общих работ по теме, а также учебников и учебных пособий. Далее рекомендуется перейти к анализу монографий и статей, рассматривающих отдельные аспекты проблем, изучаемых в рамках курса, а также официальных материалов и неопубликованных документов (научно-исследовательские работы, диссертации), в которых могут содержаться основные вопросы изучаемой проблемы.

Работу с источниками надо начинать с ознакомительного чтения, т.е. просмотреть текст, выделяя его структурные единицы. При ознакомительном чтении закладками отмечаются те страницы, которые требуют более внимательного изучения.

В зависимости от результатов ознакомительного чтения выбирается дальнейший способ работы с источником. Если для разрешения поставленной задачи требуется изучение некоторых фрагментов текста, то используется метод выборочного чтения. Если в книге нет подробного оглавления, следует обратить внимание ученика на предметные и именные указатели.

Избранные фрагменты или весь текст (если он целиком имеет отношение к теме) требуют вдумчивого, неторопливого чтения с «мысленной проработкой» материала. Такое чтение предполагает выделение: 1) главного в тексте; 2) основных аргументов; 3) выводов. Особое внимание следует обратить на то, вытекает тезис из аргументов или нет.

Необходимо также проанализировать, какие из утверждений автора носят проблематичный, гипотетический характер, и уловить скрытые вопросы.

Понятно, что умение таким образом работать с текстом приходит далеко не сразу. Наилучший способ научиться выделять главное в тексте, улавливать проблематичный характер утверждений, давать оценку авторской позиции – это сравнительное чтение, в ходе которого Вы знакомитесь с различными мнениями по одному и тому же вопросу, сравниваете весомость и доказательность аргументов сторон и делаете вывод о наибольшей убедительности той или иной позиции.

Если в литературе встречаются разные точки зрения по тому или иному вопросу из-за сложности прошедших событий и правовых явлений, нельзя их отвергать, не разобравшись. При наличии расхождений между авторами необходимо найти рациональное зерно у каждого из них, что позволит глубже усвоить предмет изучения и более критично оценивать изучаемые вопросы. Знакомясь с особыми позициями авторов, нужно определять их схожие суждения, аргументы, выводы, а затем сравнивать их между собой и применять из них ту, которая более убедительна.

Следующим этапом работы с литературными источниками является создание конспектов, фиксирующих основные тезисы и аргументы.

Таким образом, при работе с источниками и литературой важно уметь:

- сопоставлять, сравнивать, классифицировать, группировать, систематизировать информацию в соответствии с определенной учебной задачей;
- обобщать полученную информацию, оценивать прослушанное и прочитанное;
- фиксировать основное содержание сообщений; формулировать, устно и письменно, основную идею сообщения; составлять план, формулировать тезисы;
- готовить и презентовать развернутые сообщения типа доклада;
- работать в разных режимах (индивидуально, в паре, в группе), взаимодействуя друг с другом;

- пользоваться реферативными и справочными материалами;
- контролировать свои действия и действия своих товарищей, объективно оценивать свои действия;
- обращаться за помощью, дополнительными разъяснениями к преподавателю, другим студентам;
- пользоваться лингвистической или контекстуальной догадкой, словарями различного характера, различного рода подсказками, опорами в тексте (ключевые слова, структура текста, предваряющая информация и др.);
- использовать при говорении и письме перифраз, синонимичные средства, слова-описания общих понятий, разъяснения, примеры, толкования, «словотворчество»;
- повторять или перефразировать реплику собеседника в подтверждении понимания его высказывания или вопроса;
- обратиться за помощью к собеседнику (уточнить вопрос, переспросить и др.);
- использовать мимику, жесты (вообще и в тех случаях, когда языковых средств не хватает для выражения тех или иных коммуникативных намерений).

Подготовка к промежуточной аттестации.

При подготовке к промежуточной аттестации целесообразно:

- внимательно изучить перечень вопросов и определить, в каких источниках находятся сведения, необходимые для ответа на них;
- внимательно прочитать рекомендованную литературу;
- составить краткие конспекты ответов (планы ответов).

Лекционные занятия проводятся в соответствии с тематическим ланом, при изложении материала рекомендуется использовать презентации в среде PowerPoint и фрагменты печатных материалов по теме лекции.

В ходе интерактивных занятий следует проводить разбор конкретных примеров, максимально приближенных к реальным данным, соответствующих экономической и финансовой информации.

Основное внимание при проведении практических занятий следует уделять развитию навыков формирования рациональных схем данных предметной области, реализации этих схем в среде современных аналитических систем, формирования сложных содержательных запросов по выбору данных, использования методов и алгоритмов анализа данных.

При этом задача состоит в обучении профессиональным навыкам разработки и использования современных аналитических систем.

Тема Корпоративные информационные системы (ERP)

Содержание тем: Инсталляция Oracle 11g. (Структура каталогов. Файлы данных, журнальные, управляющие. Службы в Windows. Соединение с базой. Oracle XE. SQL*Plus.). Архитектура базы данных Oracle 11g. (Экземпляр. SGA. PGA. Серверные и пользовательские процессы. Фоновые процессы. Блоки, экстенты, сегменты, табличные пространства. Запуск и останов. Особенности Oracle XE.). Сетевая среда Oracle. (Сетевые службы Oracle Net Services. Прослушиватели. Соединение с удалённой базой.)

Методические рекомендации

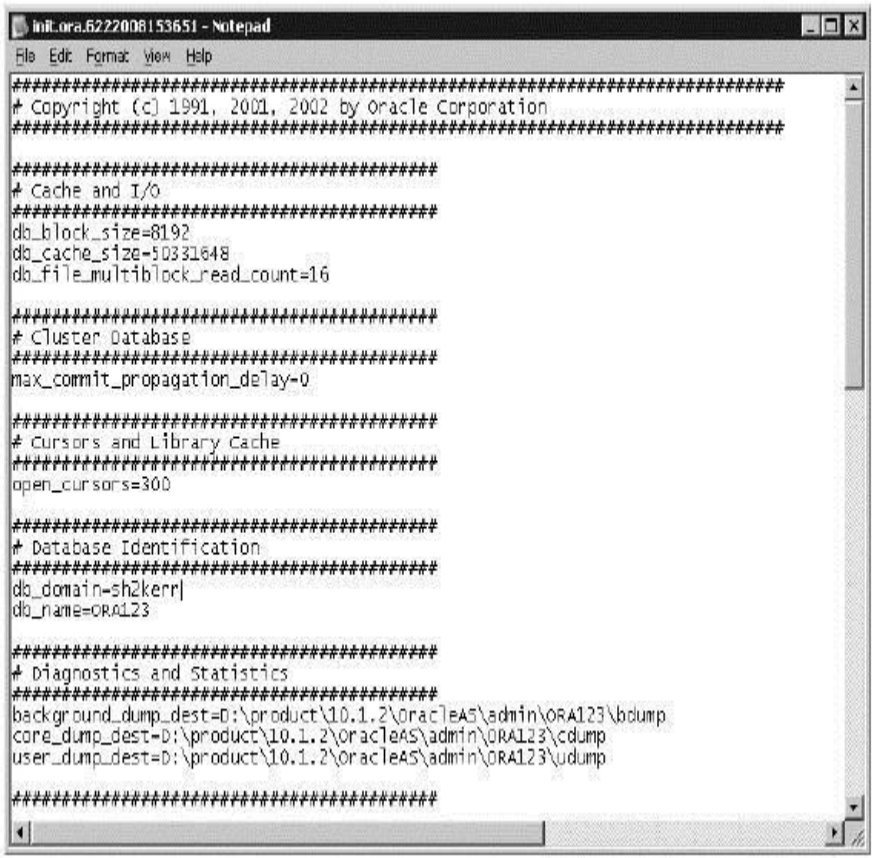
Система управления базами данных (СУБД) Oracle предназначена для одновременного доступа к большим объемам хранимой информации и манипуляции с ними. В СУБД есть два основных понятия, которые необходимо усвоить для понимания некоторых последующих моментов с точки зрения безопасности и защиты СУБД, – это база данных и экземпляр. Если в двух словах, то база данных – это набор файлов в ОС, а экземпляр – процессы и память, причем одна база данных может быть доступна в нескольких экземплярах, а экземпляр единовременно обеспечивает доступ только к одной базе данных. Теперь рассмотрим эти понятия подробнее.

База данных Oracle

В базе данных Oracle есть два уровня представления данных: физический и логический. Физический уровень включает файлы баз данных, которые хранятся на диске, а логический уровень включает в себя табличное пространство, схемы пользователей. Рассмотрим эти уровни более подробно.

Физический уровень базы данных

База данных и экземпляр на физическом уровне представлены шестью типами файлов. К экземпляру относятся файлы параметров, в которых прописываются его характеристики. Основной файл – это файл init.ora, отвечающий за параметры инициализации экземпляра, такие как имя базы данных, ссылку на управляющие файлы и пр. Пример файла инициализации представлен на рисунке



Файлы базы данных

База данных как таковая представлена набором файлов разных типов, в которых собственно хранятся различные данные. Ниже кратко рассказано о том, что представляют собой эти типы файлов и чем файлы каждого типа могут быть нам полезны:

- *Файлы данных*. В этих файлах хранятся собственно сами данные в виде таблиц, индексов, триггеров и прочих объектов. Файлы данных являются наиболее важными во всей базе данных. В стандартной базе должно присутствовать минимум два файла данных: для системных данных (табличное пространство SYSTEM) и для пользовательских данных (табличное пространство USER). В табличном пространстве

SYSTEM хранятся пароли всех пользователей в зашифрованном виде.

- *Файлы журнала повторного выполнения (redo logs)*. Файлы журнала повторного выполнения очень важны для базы данных Oracle. В них записываются все транзакции базы данных. Они используются только для восстановления данных в самой базе при сбое экземпляра. В журналах повторного выполнения можно обнаружить множество критичной информации, о существовании которой рядовой администратор мог и не задуматься, в том числе и пароли пользователей.
- *Управляющие файлы*. В этих файлах определено местонахождение файлов данных и другая информация о состоянии базы данных. Управляющие файлы должны быть хорошо защищены. Наиболее важным является файл параметров инициализации экземпляра, потому что без него не удастся запустить экземпляр. Остальные файлы, такие как **LISTENER.ORA, SQLNET.ORA, PROTOCOL.ORA, NAMES.ORA** и пр., связаны с поддержкой сети и так же очень важны. В этих файлах можно обнаружить множество полезной информации для проникновения в СУБД.
- *Временные файлы*. Временные файлы используются для хранения промежуточных результатов действий над большим объемом данных в случае, если в оперативной памяти для этого не хватает места. Во временных файлах можно обнаружить содержимое временных таблиц и построенных по ним индексов. Временные файлы могут оказаться полезными в процессе расследования инцидентов или при восстановлении важной информации, удаленной из базы данных.
- *Файлы паролей*. Используются для аутентификации пользователей, выполняющих удаленное администрирование СУБД по сети. Более детально о них мы будем говорить позже.

Как видно, с точки зрения безопасности каждый приведенный выше тип файлов имеет большое значение.

Логический уровень базы данных

На логическом уровне находятся табличные пространства и схема БД, состоящая из таблиц, индексов, представлений, хранимых процедур и пр.

База данных разделяется на несколько логических частей, называемых табличными пространствами. Табличные пространства используются для логической группировки данных между собой для упрощения администрирования.

Каждое табличное пространство состоит из одного или более файлов данных, которые физически могут располагаться на разных дисках.

В табличных пространствах, в свою очередь, находятся схемы – это своеобразные контейнеры хранимых в БД объектов. Каждая схема однозначно ассоциируется с определенным пользователем – владельцем этой схемы. В этих схемах уже находятся такие логические единицы, как таблицы, индексы, представления и хранимые процедуры.

Вопросы и задания для самопроверки:

1. Архитектура СУБД Oracle.
2. Режимы остановки и запуска экземпляра базы данных.
3. Пользователи и схемы в Oracle.
4. Управление привилегиями и ролями в Oracle.

Литература

Основная

1. Базы данных.Практическое применение СУБД SQL и NoSOL-типа для применения проектирования информационных систем: Учебное пособие / Мартишин С.А., Симонов В.Л., Храпченко М.В. - М.:ИД ФОРУМ, НИЦ ИНФРА-М, 2017. - 368 с.: 60x90 1/16. - (Высшее образование) (Переплёт 7БЦ) ISBN 978-5-8199-0660-6 [Электронный ресурс] http://znanium.com/bookread2.php?book=556449, 05.10.2017.

2. Базы данных: учебник / Л.И. Шустова, О.В. Тараканов. - М.: НИЦ ИНФРА-М, 2016. [Электронный рессурс] - http://znanium.com/bookread2.php?book=491069, 05.10.2017.

Дополнительная:

1. Поляков, Александр Михайлович. Безопасность Oracle глазами аудитора: нападение и защита [Текст] / А. М. Поляков ; под ред. И. Медведовского. - Москва : ДМК Пресс, 2014. - 334 с. : ил. – ISBN.

2. Балдин, Константин Васильевич. Информационные системы в экономике [Текст] : учебник для студентов вузов, обучающихся по специальностям "Прикладная информатика (по областям)" и другим междисциплинарным специальностям / В. Б. Уткин, К. В. Балдин. - 7-е изд. - Москва : Дашков и К°, 2015. - 394 с. - Библиогр.: с. 390-394. - ISBN 978-5-394-01449-9

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Российское образование, федеральный портал [Официальный сайт] — URL: http://www.edu.ru
2. Образовательный портал «Учеба» [Официальный сайт] URL: http://www.ucheba.com/
3. Портал «Российское образование» [Официальный сайт] URL: http://www.edu.ru/

Тема Семейство полнофункциональных бизнес-приложений Oracle E-Business Suite.

Содержание тем: Управление структурами хранения данных. (Табличные пространства и файлы данных. Работа с табличными пространствами. Управление заполнением блока. Работа с экстентами.). Управление пользователями. (Пользователи и схемы. Децентрализованная истема защиты данных. Системные и объектные привилегии. Профили и роли. Создание пользователей и управление ими.). Управление хранимыми объектами. (Таблицы. Их виды создание, удаление и изменение таблиц. Временные таблицы. Темпоральные таблицы. Представления. Последовательности.)

Методические рекомендации

Привилегии

Привилегии пользователей назначаются им администратором базы данных и определяют, какие действия над данными и над объектами схемы являются разрешенными. При контроле привилегий используется имя пользователя базы данных, называемое иногда идентификатором авторизации (Autorization ID). Некоторые СУБД идентифицируют понятие "пользователь" с понятием "учетная запись".

Все объекты пользователя БД входят в его схему. На практике один пользователь, как правило, ассоциируется с одной схемой, хотя стандарт подразумевает, что одному пользователю может принадлежать несколько схем, содержащих взаимосвязанные объекты.

После успешного завершения процедуры идентификации открывается сеанс пользователя и устанавливается соединение с базой данных.

Существуют привилегии двух типов:

- системные привилегии (system privileges), контролирующие общий доступ к базе данных;
- объектные привилегии (object privileges), контролирующие доступ к конкретным объектам базы данных.

Синтаксис, используемый для работы с привилегиями, на практике значительно шире стандарта, но в значительной степени зависит от архитектуры конкретной БД.

Для управления привилегиями определены следующие правила:

- объект принадлежит пользователю, его создавшему (если синтаксисом не указано создание объекта другого пользователя, конечно, при соответствующих полномочиях);
- владелец объекта, согласно стандарту, может изменять привилегии своего объекта (в коммерческих СУБД, таких как Oracle, уровни полномочий представляют собой более сложную иерархию);
- объектная привилегия всегда соотносится с конкретным объектом, а системная - с объектами вообще.

Язык SQL поддерживает следующие привилегии:

- ALTER - позволяет выполнять оператор ALTER TABLE;
- SELECT - позволяет выполнять оператор запроса;
- INSERT - позволяет выполнять добавление строк в таблицу;
- UPDATE - позволяет изменять значения во всей таблице или только в некоторых столбцах;
- DELETE - позволяет удалять строки из таблицы;
- REFERENCES - позволяет устанавливать внешний ключ с использованием в качестве родительского ключа любых столбцов таблицы или только некоторых из них;
- INDEX - позволяет создавать индексы (не входит в стандарт SQL-92);
- DROP - позволяет удалять таблицу из схемы базы данных.

Предоставление и снятие привилегий

Предоставление привилегии выполняется SQL-оператором GRANT , который имеет в стандарте SQL-92 следующее формальное описание:

```
 ON { [TABLE] table_name
      | DOMAIN domain_name
      | COLLATION collation_name
      | CHARACTER SET set_name
      | TRANSLATION translation_name }
TO { user_name .,: } | PUBLIC
[ WITH GRANT OPTION ]
где privilege определяется как

    { ALL PRIVELEGES }
    | SELECT
    | DELETE
    | INSERT [(field .,:)]
    | UPDATE [(field .,:)]
    | REFERENCES [(field .,:)]
    | USAGE
```

После фразы GRANT через запятую можно перечислить список всех назначаемых привилегий.

Фраза ON определяет объект, для которого устанавливается привилегия.

Фраза TO указывает пользователя или пользователей, для которых устанавливается привилегия.

Так, оператор GRANT SELECT ON tbl1 TO PUBLIC; предоставляет доступ к выполнению оператора SELECT для таблицы tbl1 не только всем существующим пользователям, но и тем, которые позднее будут добавлены в базу данных.

Оператор GRANT UPDATE ON tbl1 TO user1; предоставляет пользователю user1 привилегию UPDATE на всю таблицу, а оператор GRANT UPDATE (f1,f2) ON tbl1 TO user1 предоставляет привилегию UPDATE для изменения только столбцов f1 и f2.

Фраза WITH GRANT OPTION предоставляет получающему привилегию пользователю дополнительную привилегию GRANT OPTION, позволяющую выполнять передачу полученных привилегий.

Отмена привилегии выполняется SQL-оператором REVOKE, который имеет в стандарте SQL-92 следующее формальное описание:

```
REVOKE [ GRANT OPTION FOR ]
    { ALL PRIVILEGES } | privilege
    ON { [TABLE] table_name
        | DOMAIN domain_name
        | COLLATION collation_name
        | CHARACTER SET set_name
        | TRANSLATION translation_name }
    FROM { PUBLIC | user_name .,: }
    [ CASCADE | RESTRICT ]
```

После фразы REVOKE через запятую можно перечислить список всех отменяемых привилегий.

Фраза ON определяет объект, для которого отменяется привилегия.

Фраза FROM указывает пользователя или пользователей, для которых отменяется привилегия.

Фраза GRANT OPTION FOR определяет отмену не самих привилегий, а только права их передачи другим пользователям.

Если одна привилегия вместе с опцией WITH GRANT OPTION была последовательно передана от одного пользователя другому несколько раз, то образуется цепочка зависимых привилегий. Фразы CASCADE и RESTRICT определяют, что будет происходить с этими привилегиями при отмене одного из звеньев этой цепочки.

Если при отмене зависимой привилегии для объекта не остается ни одной существующей привилегии, то такой объект называется несостоявшимся. Например, подобное может произойти с представлением, созданным как запрос к таблице, привилегия на которую была утрачена.

Если при отмене привилегии появляется несостоявшийся объект, то фраза RESTRICT предотвратит выполнение оператора REVOKE, и никакие привилегии отменены не будут.

Если указана фраза CASCADE и при отмене привилегии появляется несостоявшийся объект, то все несостоявшиеся

объекты (представления) удаляются, а при наличии несостоявшихся ограничений в таблицах они отменяются автоматически выполнением оператора ALTER TABLE несостоявшиеся ограничения в доменах отменяются автоматически выполнением оператора ALTER DOMAIN.

Роли

Ролью называется именованный набор привилегий. Объединение привилегий в роли значительно упрощает процесс назначения и снятия привилегий. Если СУБД поддерживает управление ролями, то в SQL-операторах GRANT и REVOKE вместо имени пользователя можно указывать имя роли.

Многоуровневый контроль доступа в БД Oracle

Среди современных коммерческих СУБД базу данных Oracle можно считать одной из самых продвинутых в области контроля доступа. Все привилегии делятся на системные и объектные.

Синтаксис оператора GRANT, выполняющего предоставление пользователям или ролям системных полномочий и ролей, может быть представлен следующей схемой:

предоставление пользователям или ролям системных полномочий и ролей,

Предоставление пользователям или ролям привилегий над обычными объектами БД Oracle также может быть показано на примере следующей схемы:

Предоставление пользователям или ролям привилегий над обычными объектами БД Oracle

system_priv - предоставляемое системное полномочие.

role - предоставляемая роль.

TO -определяет пользователей или роли, которым предоставляются системные или объектные полномочия.

PUBLIC - указывает, что системные или объектные полномочия, определяемые оператором, предоставляются всем пользователям.

WITH ADMIN OPTION - позволяет пользователю, получившему системные или объектные полномочия или роль,

предоставлять их в дальнейшем другим пользователям или ролям. Такое разрешение включает и возможность изменения или удаления роли. (Синтаксически данная опция отличается от стандарта SQL-92.)

object_priv - определяет предоставляемую привилегию, которая может быть указана одним из следующих значений:

- ALTER
- DELETE
- EXECUTE (только для процедур, функций и пакетов)
- INDEX (только для таблиц)
- INSERT
- REFERENCES (только для таблиц)
- SELECT
- UPDATE.

column - определяет столбец таблицы или представления, на который распространяется предоставляемая привилегия.

ON - определяет объект (таблицу, вид, хранимую процедуру, снимок), на который предоставляется привилегия.

Например:

```
GRANT SELECT, UPDATE ON tbl1 TO PUBLIC
GRANT REFERENCES  (f1), UPDATE (f1, f2, f3)
    ON user1.tbl1 TO user2
```

Приведем список предоставляемых оператором GRANT системных полномочий, который характеризует систему контроля доступа БД Oracle. К системным полномочиям относятся следующие:

ALTER ANY CLUSTER - разрешает получившему эти полномочия изменение любого кластера в любой схеме;

ALTER ANY INDEX - разрешает изменение любого индекса в любой схеме;

ALTER ANY PROCEDURE - разрешает изменение любой хранимой функции, процедуры или пакета в любой схеме;

ALTER ANY ROLE - разрешает изменение в базе данных любой роли;

ALTER ANY SEQUENCE - разрешает изменение в базе данных любой последовательности;

ALTER ANY SNAPSHOT - разрешает изменение в базе данных любого снимка;

ALTER ANY TABLE - разрешает изменение в схеме любой таблицы или вида;

ALTER ANY TRIGGER - позволяет разрешать, запрещать или компилировать любой триггер базы данных в любой схеме; изменение в базе данных любой роли;

ALTER DATABASE- разрешает изменение базы данных;

ALTER PROFILE - разрешает изменение профилей;

ALTER RESOURCE COST - разрешает устанавливать цену ресурсов сеанса работы пользователя;

ALTER ROLLBACK SEGMENT - разрешает изменение сегментов отката;

ALTER SESSION - разрешает выполнение оператора ALTER SESSION;

ALTER SYSTEM - разрешает выполнение оператора ALTER SYSTEM;

ALTER TABLESPACE - разрешает изменение табличных пространств;

ALTER USER - разрешает изменение параметров для любого пользователя (пароль, количество доступного табличного пространства, назначенный профиль и т.п.);

ANALYZE ANY - разрешает анализировать таблицу, кластер или индекс в любой схеме;

AUDIT ANY - разрешает выполнять аудит любого объекта в любой схеме;

AUDIT SYSTEM - разрешает выполнение SQL-оператора AUDIT;

BACKUP ANY TABLE - позволяет выполнять экспорт объектов из схемы других пользователей;

BECOME USER - позволяет становиться другим пользователем (требуется при импорте БД);

COMMENT ANY TABLE - разрешает получившему эти полномочия комментарий для любой таблицы, вида или столбца в любой схеме;

CREATE ANY CLUSTER ;

CREATE ANY INDEX ;

CREATE ANY LIBRARY ;

CREATE ANY PROCEDURE ;

CREATE ANY SEQUENCE ;

CREATE ANY SNAPSHOT ;

CREATE ANY SYNONYM ;

CREATE ANY TABLE ;

CREATE ANY TRIGGER ;

CREATE ANY VIEW ;

CREATE CLUSTER - разрешает создавать кластер в своей схеме (системное полномочие, не содержащее в названии фразу ANY, распространяется только на собственную схему пользователя);

CREATE DATABASE LINK - разрешает создавать линк базы данных в своей схеме;

CREATE PROCEDURE ;

CREATE PROFILE ;

CREATE PUBLIC DATABASE LINK - разрешает создавать общедоступные линки базы данных;

CREATE PUBLIC SYNONYM ;

CREATE ROLE - разрешает создание ролей;

CREATE ROLLBACK SEGMENT ;

CREATE LIBRARY ;

CREATE SEQUENCE;

CREATE SESSION - разрешает соединение с базой данных;

CREATE SNAPSHOT;

CREATE SYNONYM;

CREATE TABLE;

CREATE TABLESPACE ;

CREATE TRIGGER;

CREATE USER ;

CREATE VIEW ;
DELETE ANY TABLE ;
DROP ANY CLUSTER ;
DROP ANY INDEX - разрешает удаление любого индекса;
DROP ANY LIBRARY ;
DROP ANY PROCEDURE ;
DROP ANY ROLE ;
DROP ANY SEQUENCE ;
DROP ANY SNAPSHOT ;
DROP ANY SYNONYM ;
DROP ANY TABLE ;
DROP ANY TRIGGER ;
DROP ANY VIEW ;
DROP LIBRARY ;
DROP PROFILE ;
DROP PUBLIC DATABASE LINK ;
DROP PUBLIC SYNONYM ;
DROP ROLLBACK SEGMENT ;
DROP TABLESPACE ;
DROP USER ;
EXECUTE ANY PROCEDURE - разрешает выполнение процедур и функций, а также ссылки на общедоступные переменные пакетов в любой схеме;
FORCE ANY TRANSACTION - позволяет выполнять фиксацию или откат любой сомнительной распределенной транзакции на локальной базе данных, а также определять сбой распределенной транзакции;
FORCE TRANSACTION - позволяет выполнять фиксацию или откат собственной сомнительной распределенной транзакции на локальной базе данных;
GRANT ANY PRIVILEGE ;
GRANT ANY ROLE ;
INSERT ANY TABLE ;
LOCK ANY TABLE ;

MANAGE TABLESPACE - разрешает переключение табличного пространства из автономного режима в оперативный или обратно, а также разрешает выполнять копирование табличного пространства;

RESTRICTED SESSION ;

SELECT ANY SEQUENCE ;

SELECT ANY TABLE ;

UNLIMITED TABLESPACE - разрешает неограниченное использование любого табличного пространства. Предоставление этого полномочия перекрывает любые ограничения на количество доступного табличного пространства, ранее установленные для пользователя;

UPDATE ANY TABLE - разрешает изменение строк в таблицах и видах любой схемы.

При создании базы данных Oracle некоторые роли создаются автоматически. Следующая таблица содержит названия автоматически создаваемых Oracle ролей и список предоставляемых ими системных полномочий. Эти роли создают иерархию предоставляемых полномочий.

Роль	**Предоставляемые системные полномочия и роли**
CONNECT	ALTER SESSION
	CREATE CLUSTER
	CREATE DATABASE LINK
	CREATE SEQUENCE
	CREATE SESSION
	CREATE SYNONYM
	CREATE TABLE
	CREATE VIEW
RESOURCE	CREATE CLUSTER
	CREATE PROCEDURE
	CREATE SEQUENCE
	CREATE TABLE

DBA	CREATE TRIGGER Все системные полномочия WITH ADMIN OPTION
	EXP_FULL_DATABASE (роль)
EXP_FULL_DATABASE	IMP_FULL_DATABASE (роль) SELECT ANY TABLE
	BACKUP ANY TABLE
	INSERT, UPDATE, DELETE ON sys.incexp, sys.incvid, sys.incfil

Для просмотра предоставленных привилегий администратор базы данных может использовать следующие системные представления словаря данных:

Системное представление	**Описание**
ALL_COL_PRIVS	Содержит список привилегий, предоставленных для столбцов таблицы другим пользователям или PUBLIC. ;
	Содержит столбцы: GRANTOR (кто предоставляет привилегию)
	GRANTEE (кому предоставляется привилегия) TABLE_SCHEMA (схема объекта)
	TABLE_NAME
	COLUMN_NAME PRIVILEGE
ALL_COL_PRIVS_MADE	Содержит список привилегий столбцов, которыми

	владеет пользователь или предоставляет на них привилегии.
	Содержит столбцы:
	GRANTEE
	OWNER
	GRANTOR
	TABLE_NAME
	COLUMN_NAME
	PRIVILEGE
ALL_TAB_PRIVS	Содержит список привилегий, предоставленных для таблицы другим пользователям или PUBLIC.
	Содержит столбцы:
	GRANTOR
	GRANTEE
	TABLE_NAMEPRIVILEGE
DBA_PROFILES	Содержит описания всех профилей базы данных и определяемых ими ограничений.
	Содержит столбцы:
	PROFILE
	RESOURCE_NAME
	LIMIT
DBA_ROLES	Содержит список имен всех существующих в базе данных ролей.
	Содержит столбцы:
	ROLE
	PASSWORD_REQUIRED

DBA_ROLE_PRIVS	Содержит список ролей, предоставляемых другим пользователям или ролям.
	Содержит столбцы:
	GRANTEE (кто получает полномочия)
	GRANTED_ROLE (предоставляемая роль)
DBA_SYS_PRIVS	Содержит список системных полномочий, предоставленных пользователям и ролям.
	Содержит столбцы:
	GRANTEE (кто получает полномочия)
	PRIVILEGE (название системного полномочия)
	ADMIN_OPTION
DBA_TAB_PRIVS	Содержит список всех предоставленных полномочий для объектов базы данных.
	Содержит столбцы:
	GRANTEE
	OWNER
	TABLE_NAME
	GRANTOR PRIVILEGE
DBA_USERS	Содержит информацию обо всех пользователях базы данных.
	Содержит столбцы:
	USERNAME
	USER_ID
	PASSWORD

	DEFAULT_TABLESPACE
	PROFILE
ROLE_SYS_PRIVS	Содержит информацию о предоставленных ролям системных полномочиях.
	Содержит столбцы:
	ROLE
	PRIVILEGE
ROLE_TAB_PRIVS	Содержит информацию о предоставленных ролям привилегиях для таблиц и столбцов.
	Содержит столбцы:
	ROLE
	OWNER
	TABLE_NAME
	COLUMN_NAME
	PRIVILEGE
SYSTEM_PRIVILEGE_MAP	Содержит список всех системных полномочий
TABLE_PRIVILEGE_MAP	Содержит информацию о кодах привилегий доступа к таблицам и столбцам.
USER_ROLE_PRIVS	Содержит список ролей, предоставленных пользователю.
	Содержит столбцы:
	USERNAME
	GRANTED_ROLE
USER_SYS_PRIVS	Содержит список всех системных полномочий, предоставленных пользователю.

	Содержит столбцы: USERNAME PRIVILEGE
USER_TAB_PRIVS	Содержит список привилегий для объектов, где пользователь является владельцем, получателем или лицом, предоставляющим привилегии.
	Содержит столбцы:
	GRANTEE (кому предоставляется привилегия)
	OWNER TABLE_NAME (имя объекта) GRANTOR (кто предоставляет привилегию)
USER_TAB_PRIVS_MADE	Содержит список всех предоставлений привилегий для объектов, принадлежащих пользователю. Содержит столбцы:
	GRANTEE
	GRANTOR TABLE_NAME
	PRIVILEGE
USER_TAB_PRIVS_RECD	Содержит список всех привилегий для объектов, где пользователь является получателем привилегии.
	Содержит столбцы: OWNER (владелец объекта)
	TABLE_NAME (имя объекта)

	GRANTOR (имя пользователя, предоставившего привилегию)
	PRIVILEGE

Эти системные представления позволяют администратору БД Oracle полностью контролировать назначение, передачу и взаимозависимость системных и объектных привилегий.

Вопросы и задания для самопроверки:

1. Структуры хранения данных в Oracle (табличные пространства, сегменты, экстенты, блоки данных).
2. Таблицы в Oracle. Создание таблиц.
3. Ограничения в Oracle. Создание ограничений.
4. Индексы в Oracle. Создание и использование индексов.

Литература

Основная

1. Базы данных: учебник / Л.И. Шустова, О.В. Тараканов. - М.: НИЦ ИНФРА-М, 2016. [Электронный рессурс] - http://znanium.com/bookread2.php?book=491069, 05.10.2017.

2. Тузовский, А. Ф. Объектно-ориентированное программирование : учебное пособие для прикладного бакалавриата / А. Ф. Тузовский. — М. : Издательство Юрайт, 2017. [Электронный ресурс] https://www.biblio-online.ru/viewer/BDEEFB2D-532D-4306-829E-5869F6BDA5F9#page/1, 05.10.2017.

Дополнительная:

1. Балдин, Константин Васильевич. Информационные системы в экономике [Текст] : учебник для студентов вузов, обучающихся по специальностям "Прикладная информатика (по областям)" и другим междисциплинарным специальностям / В. Б. Уткин, К. В. Балдин. - 7-е изд. - Москва :

Дашков и К°, 2015. - 394 с. - Библиогр.: с. 390-394. - ISBN 978-5-394-01449-9
2. Информационные системы и технологии в экономике и управлении [Текст] : учебник для бакалавров : учебник по направлению "Менеджмент" / [В. В. Трофимов и др.] ; под ред. В. В. Трофимова ; С.-Петербург. гос. эконом. ун-т. - 4-е изд., перераб. и доп. - Москва : Юрайт, 2014. - 542 с. - (Бакалавр. Базовый курс). - Библиогр. в конце глав. - ISBN 978-5-9916-3608-7

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

4. Единое окно доступа к образовательным ресурсам «Единое окно» [Официальный сайт] URL: http://window.edu.ru/

5. Федеральная университетская компьютерная сеть России [Официальный сайт] URL: http://www.runnet.ru/

6. Служба тематических толковых словарей[Официальный сайт] URL: http://www.glossary.ru/

Тема Функциональные подсистемы Oracle E'Business Suite

Содержание тем: Транзакции. (Команды COMMIT, ROLLBACK. Точки останова SAVEPOINT. Транзакции read only, read write, serialyzable и автономные.). PL\SQL. (Типы данных. Блоки. Разветвления и циклы. Курсоры явные и неявные. Работа с курсорами. Курсорные ссылки. Хранимые процедуры и функции. Исключительные ситуации.). Манипулирование данных (Команды INSERT, UPDATE, DELETE. ROWID

Методические рекомендации

Создание хранимой процедуры PL/SQL

Для того чтобы написать собственную программу на PL/SQL, нужно воспользоваться одной из инструкций *SQL CREATE*. Например, если вы хотите создать хранимую функцию именем wordcount для подсчета количества слов в строке, выполните инструкцию:

CREATE FUNCTION:

```
CREATE FUNCTION wordcount (str IN VARCHAR2)
RETURN PLS_INTEGER
AS
здесь объявляются локальные переменные
BEGIN
здесь реализуется алгоритм
END;
/
```

Как и в случае с простыми блоками *BEGIN-END*, приводившимися ранее, код этой инструкции в SQL*Plus должен завершаться символом косой черты, который размещается в отдельной строке.

Если администратор базы данных предоставил вам привилегию создания процедур *CREATE PROCEDURE* (которая также включает привилегию создания функций), эта инструкция заставит Oracle откомпилировать и сохранить в схеме заданную хранимую функцию. И если код будет откомпилирован успешно, вы увидите следующее сообщение:

```
Function created.
```

Если в схеме Oracle уже имеется объект (таблица или пакет) с именем *wordcount*, выполнение инструкции *CREATE FUNCTION* завершится сообщением об ошибке:

```
ORA-00955: name is already used by an existing object.
```

По этой причине Oracle поддерживает инструкцию *CREATE OR REPLACE FUNCTION* — вероятно, вы будете использовать ее в 99 случаях из 100:

```
CREATE OR REPLACE FUNCTION wordcount (str IN
VARCHAR2)
RETURN PLS_INTEGER
AS то же, что в приведенном выше примере
```

Связка *OR REPLACE* позволяет избежать побочных эффектов, вызванных удалением и повторным созданием программ; она сохраняет все привилегии на объект, предоставленные другим пользователям или ролям. При этом она заменяет только объекты одного типа и не станет

автоматически удалять таблицу с именем *wordcount* только потому, что вы решили создать функцию с таким же именем.

Программисты обычно сохраняют подобные команды (равно как и анонимные блоки, предназначенные для повторного использования) в файлах операционной системы. Например, для хранения рассматриваемой функции можно было бы создать файл *wordcount.fun*, а для его запуска применить команду SQL*Plus @:

```
SQL> @wordcount.fun
Function created.
```

Как упоминалось ранее, SQL*Plus по умолчанию не выводит содержимое сценария на экран. Для того чтобы исходный код сценария, включая присвоенные Oracle номера строк, отображался на экране, воспользуемся командой *SET ECHO ON*. Особенно полезна эта команда в ходе диагностики. Давайте намеренно допустим в программе ошибку, закомментировав объявление переменной:

```
SQL> /* Файл в Сети: wordcount.fun */
SQL> SET ECHO ON
SQL> @wordcount.fun
SQL> CREATE OR REPLACE FUNCTION wordcount (str
IN VARCHAR2)
  2 RETURN PLS_INTEGER
  3 AS
  4 /* words PLS_INTEGER := 0; ***Намеренное внесение
ошибки*** */
  5 len PLS_INTEGER := NVL(LENGTH(str),0);
  6 inside_a_word BOOLEAN;
  7 BEGIN
  8 FOR i IN 1..len + 1
  9 LOOP
 10 IF ASCII(SUBSTR(str, i, 1)) < 33 OR i > len
 11 THEN
 12 IF inside_a_word
 13 THEN
 14 words := words + 1;
```

```
15 inside_a_word := FALSE;
16 END IF;
17 ELSE
18 inside_a_word := TRUE;
19 END IF;
20 END LOOP;
21 RETURN words;
22 END;
23 /
Warning: Function created with compilation errors.
SQL>
```

Предупреждение сообщает нам о том, что функция была создана, но из-за ошибок компиляции ее выполнение невозможно. Нам удалось сохранить исходный код в базе данных; теперь нужно извлечь подробную информацию об ошибке из базы данных. Проще всего это сделать с помощью команды SQL*Plus *SHOW ERRORS*, которую можно сократить до *SHO ERR*:

```
SQL> SHO ERR
Errors for FUNCTION WORDCOUNT:
LINE/COL ERROR
-------- -----------------------------------------------
14/13 PLS-00201: identifier 'WORDS' must be declared
14/13 PL/SQL: Statement ignored
21/4 PL/SQL: Statement ignored
21/11 PLS-00201: identifier 'WORDS' must be declared
```

Вывод других ошибок

Многие программисты Oracle знают только одну форму команды SQL*Plus:

```
SQL> SHOW ERRORS
```

Они ошибочно полагают, что для получения дополнительной информации об ошибках, не встречавшихся при последней компиляции, необходимо обращаться с запросом к представлению *USER_ERRORS*. Однако если указать в команде

SHOW ERRORS категорию и имя объекта, вы получите информацию о последних связанных с ним ошибках:

SQL> SHOW ERRORS категория [схема.]объект

Например, чтобы просмотреть информацию о последних ошибках в процедуре *wordcount*, выполните такую команду:

SQL> SHOW ERRORS FUNCTION wordcount

Будьте внимательны при интерпретации выходного сообщения:

No errors.

Оно выводится в трех случаях: (1) когда код объекта откомпилирован успешно; (2) вы задали неверную категорию (скажем, функцию вместо процедуры); и (3) объект с заданным именем не существует.

Полный список категорий, поддерживаемых этой командой, зависит от версии СУБД, но в него как минимум входят следующие категории:

DIMENSION
FUNCTION
JAVA SOURCE
JAVA CLASS
PACKAGE
PACKAGE BODY
PROCEDURE
TRIGGER
TYPE
TYPE BODY
VIEW

Компилятор обнаружил оба вхождения переменной и сообщил точные номера строк и столбцов. Более подробную информацию об ошибке можно найти по идентификатору (в данном случае PLS-00201) в документации Oracle Database Error Messages.

Во внутренней реализации команда *SHOW ERRORS* обращается с запросом к представлению Oracle *USER_ERRORS* из словаря данных. В принципе вы можете обращаться к этому

представлению и самостоятельно, но обычно это просто не нужно (см. врезку «Вывод других ошибок»).

Команда *SHOW ERRORS* часто добавляется послед каждой инструкции *CREATE*, создающей хранимую программу PL/SQL. Поэтому типичный шаблон для построения хранимых процедур в SQL*Plus может начинаться так:

```
CREATE OR REPLACE тип_программы
AS
ваш код
END;
/
SHOW ERRORS
```

(Обычно я не включаю команду *SET ECHO ON* в сценарий, а просто ввожу ее в командной строке, когда это потребуется.)

Если ваша программа содержит ошибку, которая может быть обнаружена компилятором, инструкция *CREATE* сохранит эту программу в базе данных, но в нерабочем состоянии. Если же вы неверно используете синтаксис *CREATE*, то Oracle не поймет, что вы пытаетесь сделать, и не сохранит код в базе данных.

Выполнение хранимой процедуры PL/SQL

Мы рассмотрели два способа вызова хранимой программы: заключение ее в простом блоке PL/SQL и использование команды *EXECUTE* среды SQL*Plus. Одни хранимые процедуры также можно использовать в других. Например, функция *wordcount* может использоваться в любом месте, где может использоваться целочисленное выражение. Короткий пример тестирования функции *wordcount* с входным значением *CHR(9)*, которое является ASCII-кодом символа табуляции:

```
BEGIN
DBMS_OUTPUT.PUT_LINE('Введенная строка содержит ' 
|| wordcount(CHR(9)) || '
слов');
END;
```

/

Вызов функции *wordcount* включен в выражение как аргумент процедуры *DBMS_OUTPUT. PUT_LINE*. В таких случаях PL/SQL автоматически преобразует целое число в строку, чтобы соединить его с двумя другими литеральными выражениями. Результат получается следующим:

Введенная строка содержит 0 слов

Многие функции PL/SQL можно вызывать и из SQL-инструкций. Несколько примеров использования функции *wordcount*:

- Включение в список выборки для вычисления количества слов в столбце таблицы:

SELECT isbn, wordcount(description) FROM books;

- Использование в ANSI-совместимой инструкции *CALL* для привязки выходных данных функции к переменной SQL*Plus и вывода результата:

VARIABLE words NUMBER

CALL wordcount('некоторый_ текст') INTO :words;

PRINT :words

То же, но с выполнением функции из удаленной базы данных, определяемой ссылкой *test.newyork.ora.com:*

Выполнение функции, принадлежащей схеме *bob*, при подключении к любой схеме с соответствующей привилегией:

SELECT bob.wordcount(description) FROM books WHERE id = 10007;

Вывод хранимых процедур PL/SQL

Рано или поздно вам потребуется просмотреть список имеющихся хранимых процедур и последние версии их исходного кода, которые Oracle хранит в словаре данных. Эту задачу намного проще выполнить в графических служебных программах, но если у вас такой программы нет, можно написать несколько SQL-инструкций, извлекающих из словаря данных нужную информацию.

Так, чтобы просмотреть полный список программ (а также таблиц, индексов и других элементов), запросите информацию представления *USER_OBJECTS*:

```
SELECT * FROM USER_OBJECTS;
```

Представление содержит сведения о каждом объекте: его имя, тип, время создания, время последней компиляции, состояние работоспособности и другую полезную информацию.

Если вам нужно получить данные об интерфейсе программы в SQL*Plus, проще всего воспользоваться командой *DESCRIBE*:

```
SQL> DESCRIBE wordcount
FUNCTION wordcount RETURNS BINARY_INTEGER
Argument Name                  Type                    In/Out Default?
------------------------------ ---------------------- ------ --------
STR                            VARCHAR2                IN
```

Команда *DESCRIBE* также работает с таблицами, объектными типами, процедурами и пакетами. Чтобы просмотреть полный исходный код хранимых процедур, обратитесь с запросом к представлению *USER_SOURCE* или *TRIGGER_SOURCE*.

Управление привилегиями и создание синонимов хранимых процедур

Созданную вами программу на PL/SQL обычно не может выполнять никто, кроме вас или администратора базы данных. Предоставить право на ее применение другому пользователю можно с помощью инструкции *GRANT*:

```
GRANT EXECUTE ON wordcount TO scott;
```

Инструкция *REVOKE* лишает пользователя этой привилегии:

```
REVOKE EXECUTE ON wordcount FROM scott;
```

Привилегия выполнения *EXECUTE* также может быть представлена роли:

```
GRANT EXECUTE ON wordcount TO all_mis;
```

а также всем пользователям Oracle:

```
GRANT EXECUTE ON wordcount TO PUBLIC;
```

Если привилегия *EXECUTE* представляется отдельному пользователю (например, с идентификатором *scott*), затем — роли, в которую входит этот пользователь (например, *all_mis*), и наконец, — всем пользователям, Oracle запомнит все три

варианта ее предоставления. Любой из них позволит пользователю *scott* выполнять программу. Но если вы захотите лишить данного пользователя этой возможности, то сначала следует отменить привилегию пользователя с идентификатором *scott*, а затем аннулировать привилегию на выполнение функции для всех пользователей (*PUBLIC*) и роли (или же исключить пользователя из этой роли).

Для просмотра списка привилегий, предоставленных другим пользователям и ролям, можно запросить информацию представления *USER_TAB_PRIVS_MADE*. Имена программ в этом представлении почему-то выводятся в столбце *table_name*:

```
SQL> SELECT table_name, grantee, privilege
2 FROM USER_TAB_PRIVS_MADE
3 WHERE table_name = 'WORDCOUNT';
TABLE_NAME                     GRANTEE
PRIVILEGE
------------------------------ ------------------------------ ----------
WORDCOUNT                      PUBLIC
EXECUTE
WORDCOUNT                      SCOTT
EXECUTE
WORDCOUNT                      MIS_ALL
EXECUTE
```

Если пользователь *scott* имеет привилегию *EXECUTE* на выполнение программы *wordcount*, он, возможно, захочет создать для нее синоним, чтобы ему не приходилось указывать перед именем программы префикс с именем схемы:

```
SQL> CONNECT scott/tiger
Connected.
SQL> CREATE OR REPLACE SYNONYM wordcount FOR
bob.wordcount;
```

Теперь пользователь может выполнять программу, ссылаясь на ее синоним:

```
IF wordcount(localvariable) > 100 THEN...
```

Так удобнее, потому что в случае изменения владельца программы достаточно будет изменить только ее синоним, а не все те хранимые процедуры, из которых она вызывается.

Синоним можно определить для процедуры, функции, пакета или пользовательского типа. В синонимах процедур, функций и пакетов может скрываться не только схема, но и база данных; синонимы для удаленных программ создаются так же просто, как и для локальных. Однако синонимы могут скрывать только идентификаторы схем и баз данных; синоним не может использоваться вместо пакетной подпрограммы.

Созданный синоним удаляется простой командой:

DROP SYNONYM wordcount;

Удаление хранимой программы (процедуры) PL/SQL

Если вы твердо уверены в том, что какая-либо хранимая программа вам уже не понадобится, удалите ее с помощью команды SQL *DROP*. Например, следующая команда удаляет хранимую функцию *wordcount*:

DROP FUNCTION wordcount;

Полное удаление пакета, который может состоять из двух элементов (спецификации и тела):

DROP PACKAGE pkgname;

Также можно удалить только тело пакета без отмены соответствующей спецификации:

DROP PACKAGE BODY pkgname;

При удалении программы, которая вызывается из других программ, последние помечаются как недействительные (*INVALID*).

Сокрытие исходного кода хранимой программы (процедуры) PL/SQL

При создании программы PL/SQL описанным выше способом ее исходный код сохраняется в словаре данных в виде обычного текста, который администратор базы данных может просмотреть и даже изменить. Для сохранения профессиональных секретов и предотвращения постороннего вмешательства в программный код перед распространением его следует зашифровать или скрыть иным способом.

Oracle предлагает приложение командной строки *wrap*, которое преобразует серию команд *CREATE* в комбинацию обычного текста и шестнадцатеричных кодов. Это действие не является шифрованием в прямом смысле слова, но все же направлено на сокрытие кода.

Но если вам понадобится полноценное шифрование (скажем, для передачи такой секретной информации, как пароль), полагаться на возможности *wrap* не следует.

Вопросы и задания для самопроверки:

1. Запросы SQL в Oracle. Простые запросы, условная выборка данных.
2. Использование стандартных функций в запросах.
3. Выборка из нескольких таблиц. Способы соединения таблиц в запросах.
4. Использование групповых операций в запросах.

Литература

Основная

1. Базы данных.Практическое применение СУБД SQL и NoSOL-типа для применения проектирования информационных систем: Учебное пособие / Мартишин С.А., Симонов В.Л., Храпченко М.В. - М.:ИД ФОРУМ, НИЦ ИНФРА-М, 2017. - 368 с.: 60x90 1/16. - (Высшее образование) (Переплёт 7БЦ) ISBN 978-5-8199-0660-6 [Электронный ресурс] http://znanium.com/bookread2.php?book=556449, 05.10.2017.

2. Базы данных: учебник / Л.И. Шустова, О.В. Тараканов. - М.: НИЦ ИНФРА-М, 2016. [Электронный рессурс] - http://znanium.com/bookread2.php?book=491069, 05.10.2017.

Дополнительная:

1. Поляков, Александр Михайлович. Безопасность Oracle глазами аудитора: нападение и защита [Текст] / А. М. Поляков ; под ред. И. Медведовского. - Москва : ДМК Пресс, 2014. - 334 с. : ил. – ISBN.

2. Балдин, Константин Васильевич. Информационные системы в экономике [Текст] : учебник для студентов вузов, обучающихся по специальностям "Прикладная информатика (по областям)" и другим междисциплинарным специальностям / В. Б. Уткин, К. В. Балдин. - 7-е изд. - Москва : Дашков и К°, 2015. - 394 с. - Библиогр.: с. 390-394. - ISBN 978-5-394-01449-9

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Образовательный портал [Официальный сайт] URL: «Академик» http://dic.academic.ru/
2. Web of Sciense (архив с 2002 года) рефераты [Официальный сайт] URL: http://webofknowledge.com.
3. Лекториум “(Минобрнауки РФ) единая Интернет-библиотека лекций [Официальный сайт] URL http://www.lektorium.tv/

Тема Модуль «Oracle: Главная Книга» (Oracle General Ledger

Содержание тем: Триггеры, их типы. События. Триггеры Instead of. Работа с триггерами. Пакеты PL\SQL. (Стандартные пакеты DBMS OUTPUT. DBMS_METADATA.). Объектно-реляционная модель данных. (Введение в объекты Oracle. Объектные типы. Объектные таблицы. Зависимости. Объектные ссылки. Коллекции (nested tables и varray)).

Методические рекомендации

Триггер - это выполняемый модуль, привязанный к *объекту базы данных* и событию, связанному с этим объектом. Триггер вызывается неявно при возникновении события над этим объектом. *Триггеры* имеют следующие характеристики -

- Тип триггера - *DDL* или *DML*
- Объект - таблица, *VIEW*, *системный объект* для *DDL триггеров*
- Событие - *insert*, *update*, *delete* для таблицы и *DML*, instead of для *VIEW* или *системное событие* для *DDL триггеров*.

- Способ активации - для всего *оператора* или для каждой строки
- Время активации - до или после выполнения *оператора*.

Триггеры в T-*SQL* по функциональности беднее *триггеров* в *Oracle*. В *SQL Server* существуют только after или instead of *триггеры*, вызываемые для всего *оператора*.

Триггеры sql Server Создание триггеров

```
create trigger trg
on my_table
for insert, update, delete
as
select "this is trigger"
```

Синтаксис команды создания триггера creat trigger

```
creat trigger [владелец.] название_триггера
on [владелец.] название_таблицы
for {insert, update, delete}
as SQL_операторы
creat trigger [владелец.] название_триггера
on [владелец.] название_таблицы
for {insert, update}
as
[if update (название_столбца) [{and | or}
update (название_столбца)] ... ]
SQL_операторы
[if update (название_столбца) [{and | or}
update (название_столбца)] ...
SQL_операторы] ...
```

```
CREATE trigger dbo.TD_Inscribe on Inscribe for DELETE
as begin
delete PP_S_ImageObject
where PP_S_ImageObject.ImageObjectID in
(select deleted.PhotoID from deleted)
delete from
ImageObject
```

where
ImageObject.ImageObjectID in
(select deleted.PhotoID from deleted)
end

Удаление триггеров

Команда *удаления триггера drop trigger* имеет следующий вид:

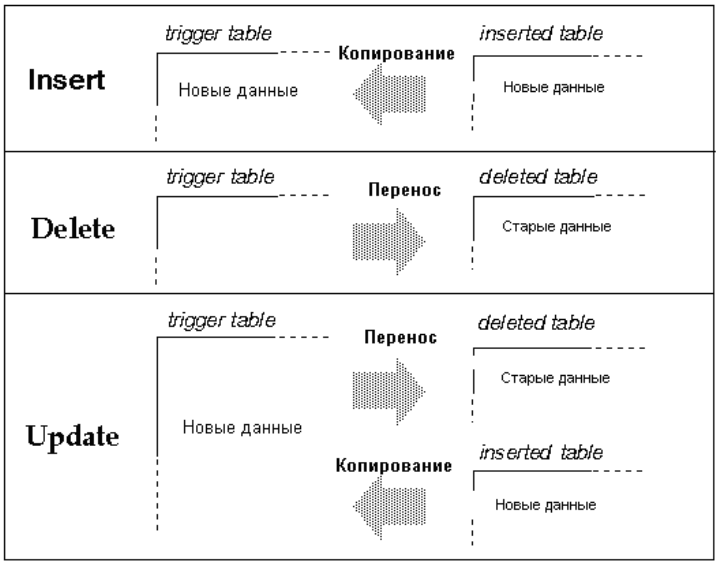
drop trigger [владелец.] название_триггера
[, [владелец.]название_триггера] ...

Таблицы INSERTED и DELETED

При вызове *триггеров* используются две специальные таблицы: таблица удаления (*deleted table*) и таблица добавления (*inserted table*). Они используются для проверки *операторов* модификации данных и создания условий для работы *триггеров*. Пользователь не может непосредственно изменять данные в этих таблицах, но может использовать находящуюся в них информацию для проверки последствий выполнения *операторов insert,update* или *delete.*

В таблице *deleted* сохраняются копии строк, которые удаляются *операторами* update или *delete*. В процессе выполнения этих *операторов* строки удаляются из триггерной таблицы и помещаются в таблицу удаления. Обычно триггерная таблица и таблица удаления не имеют общих строк.

В таблице *inserted* сохраняются копии строк, которые вставляются *операторами*
вставляются *операторами insert* или *update*. В процессе выполнения этих *операторов* новые строки вставляются в таблицу добавления и триггерную таблицу одновременно. Таким образом, таблица добавления всегда содержит копии новых строк, которые были добавлены в триггерную таблицу.




```
CREATE trigger dbo.TU_Inscribe on Inscribe for UPDATE
as begin
    if exists (select 1 from deleted, inserted where
    deleted.IdentifyDocumentID = inserted.IdentifyDocumentID
    and deleted.InscribeID = inserted.InscribeID and
    deleted.PhotoID is not null and inserted.PhotoID is null
    )
    Begin delete PP_S_ImageObject where
PP_S_ImageObject.ImageObjectID in
    (select deleted.PhotoID from deleted, inserted where
    deleted.IdentifyDocumentID = inserted.IdentifyDocumentID
    and deleted.InscribeID = inserted.InscribeID and
    deleted.PhotoID is not null and inserted.PhotoID is null
    )
    Delete ImageObject from ImageObject where
ImageObject.ImageObjectID in
    (select deleted.PhotoID from deleted, inserted where
    deleted.IdentifyDocumentID = inserted.IdentifyDocumentID
    and deleted.InscribeID = inserted.InscribeID and
    deleted.PhotoID is not null and
    inserted.PhotoID is null
```

```
)
end
end
```

Вопросы и задания для самопроверки:

1. Создание представлений в Oracle.
2. Создание и использование триггеров в Oracle.
3. Последовательности. Использование последовательностей для создания суррогатных ключей в Oracle.

Литература

Основная

1. Базы данных: учебник / Л.И. Шустова, О.В. Тараканов. - М.: НИЦ ИНФРА-М, 2016. [Электронный рессурс] - http://znanium.com/bookread2.php?book=491069, 05.10.2017.

2. Тузовский, А. Ф. Объектно-ориентированное программирование : учебное пособие для прикладного бакалавриата / А. Ф. Тузовский. — М. : Издательство Юрайт, 2017. [Электронный ресурс] https://www.biblio-online.ru/viewer/BDEEFB2D-532D-4306-829E-5869F6BDA5F9#page/1, 05.10.2017.

Дополнительная:

1. Балдин, Константин Васильевич. Информационные системы в экономике [Текст] : учебник для студентов вузов, обучающихся по специальностям "Прикладная информатика (по областям)" и другим междисциплинарным специальностям / В. Б. Уткин, К. В. Балдин. - 7-е изд. - Москва : Дашков и К°, 2015. - 394 с. - Библиогр.: с. 390-394. - ISBN 978-5-394-01449-9

2. Информационные системы и технологии в экономике и управлении [Текст] : учебник для бакалавров : учебник по направлению "Менеджмент" / [В. В. Трофимов и др.] ; под ред. В. В. Трофимова ; С.-Петербург. гос. эконом. ун-т. - 4-е изд., перераб. и доп. - Москва : Юрайт, 2014. - 542 с. - (Бакалавр.

Базовый курс). - Библиогр. в конце глав. - ISBN 978-5-9916-3608-7

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Служба тематических толковых словарей[Официальный сайт] URL: http://www.glossary.ru/
2. Образовательный портал [Официальный сайт] URL: «Академик» http://dic.academic.ru/
3. Web of Sciense (архив с 2002 года) рефераты [Официальный сайт] URL: http://webofknowledge.com.

Перечень примерных вопросов для подготовки к промежуточной аттестации

1. Определите критерии оценки информационных систем.
2. Охарактеризуйте назначение и основные функциональные блоки ERP-систем.
3. Перечислите типовые модули современной ERP-системы.
4. Каким образом производится выбор ERP-системы? На какие ключевые вопросы следует обратить особое внимание?
5. Какие основные проблемы возникают при внедрении и использовании ERP-систем?
6. Перечислите типы автоматизированных систем предприятия (АСУ).
7. Что такое "единое информационное пространство" современного предприятия и с помощью каких технологий и систем оно формируется?
8. Какие функциональные модули входят в состав типовой КИС?

9. Назовите классы задач в управлении предприятием, решаемые с помощью ИС.
10. Приведите примеры специализированных информационных систем и укажите области применения таких ИС.
11. Каким образом производится выбор ERP-системы? На какие ключевые вопросы следует обратить особое внимание?
12. Какие основные проблемы возникают при внедрении и использовании ERP-систем?
13. Опишите основные принципы различных методологий внедрения
14. КИС
15. Опишите основные решения компании Oracle.
16. Опишите стратегию компании Oracle на рынке бизнес-приложений.
17. Перечислите основные модули семейства OeBS.
18. Опишите принципы интеграции программных продуктов OeBS.
19. Опишите структуру работы модуля «Финансы» OeBS.
20. Опишите процесс работы модуля «Главная книга» OeBS.
21. Опишите процесс работы модуля «Кредиторы» OeBS.
22. Опишите процесс работы модуля «Дебиторы» OeBS.
23. Опишите процесс работы модуля «Основные средства» OeBS.
24. Перечислите основные элементы наборов книг модуля “Главная книга” OeBS.
25. Использование нескольких валют в системе OeBS.
26. Перечислите основных представителей рынка ERP-систем в структуре квадранта Gartner.
27. Перечислите этапы процесса выбора ERP-системы.
28. Опишите этапы реализации проекта по внедрению КИС.

29. Опишите, как в проектах внедрения ERP-систем производится оценка рисков.