

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
для самостоятельной работы студентов
по дисциплине «Администрирование баз данных»

по направлению подготовки
09.03.01 Информатика и вычислительная техника,
Направленность (профиль) Программное обеспечение робототехнических
и интеллектуальных систем

Ставрополь, 2025

Методические указания по организации самостоятельной работы студентов составлено в соответствии с требованиями программы дисциплины «Администрирование баз данных» для бакалавров направления подготовки 09.03.01 Информатика и вычислительная техника. Учебно-методическое пособие посвящено планируемой учебной работе студентов, выполняемой во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия, при частичном, непосредственном участии преподавателя.

Составитель: Братченко Н.Ю.

СОДЕРЖАНИЕ

1. Самостоятельная работа как важнейшая форма учебного процесса	5
2. Цели и основные задачи СРС	7
3. Виды самостоятельной работы	8
4. Организация СРС	9
5. Общие рекомендации по организации самостоятельной работы	11
6. Самостоятельная работа студента - необходимое звено становления исследователя	20
7. Методические рекомендации для студентов по отдельным формам самостоятельной работы	21
8. Учебно-методическое и информационное обеспечение дисциплины	26

ВВЕДЕНИЕ

Цель и задачи освоения дисциплины (модуля)

Цель освоения дисциплины: формирование у студентов профессиональных компетенций, основанных на освоении современных навыков администрирования баз данных, удовлетворяющих требованиям профессиональных стандартов и организаций-работодателей.

Задачи освоения дисциплины:

Задачи дисциплины – предоставить студентам в доступной для понимания форме основные сведения о:

1. Обязанностях администратора базы данных (АБД)
2. Подключении в режиме INTERNAL
3. Утилитах АБД (Import, Export, Loader)
4. Пользователях базы данных и схемы
5. Табличных пространствах и файлах данных
6. Схемах и объектах схемы
7. Блоках данных, экстентах и сегментах.
8. Структурах памяти и процессах
9. Примеры работы Oracle.
10. Журнале Повторений
11. Транзакции
12. Обеспечении защиты базы данных
13. Представлении словаря данных.
14. Привилегии (Grant, role).
15. Управлении пользователями базы данных.
16. Аудите базы данных
17. Обеспечении целостности базы данных
18. Создании базы данных. (файлы параметров)
19. Запуске и останове базы данных
20. Различных режимах работы базы данных
21. Резервном копировании базы данных
22. Динамический SQL
23. Объектно-ориентированных базах данных.

Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Администрирование баз данных» относится к части дисциплин, формируемых участниками образовательных отношений.

Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
ПК-5. Способен обеспечивать предотвращение потерь и повреждений данных при сбоях технического характера.	ИД-1 _{ПК-5} использует инструменты обеспечения безопасности БД, средства и инструменты восстановления безопасности на уровне БД; понимает методы и средства обеспечения безопасности данных при работе с установленной БД.	Понимает угрозы безопасности БД и способы их предотвращения. Применяет инструменты обеспечения безопасности БД и их возможности. Реализует средства и инструменты восстановления безопасности на уровне БД.
	ИД-2 _{ПК-5} выявляет угрозы безопасности на уровне БД; планирует и осуществляет меры по устранению последствий нарушения регламентов обеспечения безопасности на уровне БД; настраивает параметры инструментов системы безопасности в соответствии с установленными критериями; оценивает степень защиты данных от угроз безопасности на уровне БД.	Определяет угрозы безопасности на уровне БД. Имеет навыки конфигурирования БД. Осуществляет резервное копирования и восстановления БД. Демонстрирует методы администрирования БД и повышения производительности запросов СУБД.

1. Самостоятельная работа как важнейшая форма учебного процесса

Самостоятельная работа - планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Технологическая карта самостоятельной работы обучающегося

Коды реализуемых	Вид деятельности студентов	Средства и технологии	Объем часов, в том числе		
			СРС	Кон-	Всего

компетен- ций, ин- дикатора (ов)		оценки		тактная работа с препо- давате- лем	
7 семестр					
ПК-5 ИД-1 _{ПК-5} ИД-2 _{ПК-5}	Подготовка практи- ческой работе	Собеседо- вание	10,00	2,00	12,00
ПК-5 ИД-1 _{ПК-5} ИД-2 _{ПК-5}	Самостоятельное изучение литерату- ры	Собеседо- вание	40,00	2,00	42,00
Итого за 7 семестр			50,00	4,00	54,00
Итого			50,00	4,00	54,00

Самостоятельная работа студентов в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения. Обучение в ВУЗе включает в себя две, практически одинаковые по объему и взаимовлиянию части – процесса обучения и процесса самообучения. Поэтому СРС должна стать эффективной и целенаправленной работой студента.

Концепцией модернизации российского образования определены основные задачи профессионального образования - "подготовка квалифицированного работника соответствующего уровня и профиля, конкурентоспособного на рынке труда, компетентного, ответственного, свободно владеющего своей профессией и ориентированного в смежных областях деятельности, способного к эффективной работе по специальности на уровне мировых стандартов, готового к постоянному профессиональному росту, социальной и профессиональной мобильности".

Решение этих задач невозможно без повышения роли самостоятельной работы студентов над учебным материалом, усиления ответственности преподавателей за развитие навыков самостоятельной работы, за стимулирование профессионального роста студентов, воспитание творческой активности и инициативы.

К современному специалисту общество предъявляет достаточно широкий перечень требований, среди которых немаловажное значение имеет наличие у выпускников определенных способностей и умения самостоятельно добывать знания из различных источников, систематизировать полученную информацию, давать оценку конкретной финансовой ситуации. Формирование такого умения происходит в течение всего периода обучения через участие студентов в прак-

тических занятиях, выполнение контрольных заданий и тестов, написание курсовых и выпускных квалификационных работ. При этом самостоятельная работа студентов играет решающую роль в ходе всего учебного процесса.

Формы самостоятельной работы студентов разнообразны. По данной дисциплине «Администрирование баз данных» предусмотрены следующие:

Использование материала учебно-методического комплекса дисциплины

На первом этапе необходимо ознакомиться с обязательным минимумом содержания основной образовательной программы по изучаемой дисциплине. Далее необходимо изучить рабочую программу дисциплины, в которой рассмотрено содержание тем дисциплины лекционного курса, взаимосвязь тем лекций с практическими занятиями, темы и виды самостоятельной работы. По каждому виду самостоятельной работы предусмотрены определённые формы отчетности, представленные в учебной программе по дисциплине «Администрирование баз данных».

Работа с литературой

Для успешного освоения дисциплины, необходимо самостоятельно детально изучить представленные темы по рекомендуемым источникам информации, которые представлены в учебной программе.

Самостоятельная работа приобщает студентов к научному творчеству, поиску и решению актуальных современных проблем.

2. Цели и основные задачи СРС

Ведущая цель организации и осуществления СРС должна совпадать с целью обучения студента – подготовкой бакалавра с высшим образованием. При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности.

Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;

- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

3. Виды самостоятельной работы

В образовательном процессе высшего профессионального образовательного учреждения выделяется два вида самостоятельной работы – аудиторная, под руководством преподавателя, и внеаудиторная. Тесная взаимосвязь этих видов работ предусматривает дифференциацию и эффективность результатов ее выполнения и зависит от организации, содержания, логики учебного процесса (межпредметных связей, перспективных знаний и др.):

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется студентом по заданию преподавателя, но без его непосредственного участия.

Основными видами самостоятельной работы студентов без участия преподавателя по данной дисциплине являются:

- формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);
- подготовка к практическим работам, их оформление.

Основными видами самостоятельной работы студентов с участием преподавателей являются:

- текущие консультации;
- прием и защита практических работ (во время проведения л/р).

4. Организация СРС

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей дисциплины «Администрирование баз данных», объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);
- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

Текущая аттестация студентов проводится преподавателями, ведущими практические и лекционные занятия по дисциплине, в следующих формах:

- собеседование по вопросам к практическим работам;
- собеседование по вопросам для самостоятельного изучения литературы.

К практическому занятию студент должен подготовить отчет с помощью технических и программных средств.

Максимальное количество баллов студент получает, если он полностью выполнил задания по практической работе, подготовил отчет и активно участвует в работе, владеет материалом, умеет логично и четко излагать мысли, творчески подходит к решению задач, показывает самостоятельность мышления.

Основанием для снижения оценки являются:

- не качественное выполнение отчета, не соблюдение требований, отраженных в методических указаниях;
- не верное выполнение поставленных задач в практических работах.

Критерии оценивания результатов самостоятельной работы по курсу «Администрирование баз данных» приведены в Фонде оценочных средств по дисциплине.

Деятельность студентов по формированию и развитию навыков учебной самостоятельной работы

В процессе самостоятельной работы студент приобретает навыки самоорганизации, самоконтроля, самоуправления, саморефлексии и становится активным самостоятельным субъектом учебной деятельности.

Выполняя самостоятельную работу под контролем преподавателя студент должен:

- освоить минимум содержания, выносимый на самостоятельную работу студентов и предложенный преподавателем в соответствии с образовательными стандартами высшего образования;
- планировать самостоятельную работу в соответствии с графиком самостоятельной работы, предложенным преподавателем;
- самостоятельную работу студент должен осуществлять в организационных формах, предусмотренных учебным планом и рабочей программой преподавателя;
- выполнять самостоятельную работу и отчитываться по ее результатам в соответствии с графиком представления результатов, видами и сроками отчетности по самостоятельной работе студентов.

студент может:

сверх предложенного преподавателем (при обосновании и согласовании с ним) и минимума обязательного содержания, определяемого программой по данной дисциплине:

- самостоятельно определять уровень (глубину) проработки содержания материала;
- предлагать темы и вопросы для самостоятельной проработки;
- в рамках общего графика выполнения самостоятельной работы предлагать обоснованный индивидуальный график выполнения и отчетности по результатам самостоятельной работы;
- предлагать свои варианты организационных форм самостоятельной работы;
- использовать для самостоятельной работы методические пособия, учебные пособия, разработки сверх предложенного преподавателем перечня;
- использовать не только контроль, но и самоконтроль результатов самостоятельной работы в соответствии с методами самоконтроля, предложенными преподавателем или выбранными самостоятельно.

Самостоятельная работа студентов должна оказывать важное влияние на формирование личности будущего бакалавра, она планируется студентом самостоятельно. Каждый студент самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием по каждой дисциплине. Он выполняет внеаудиторную работу по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

5. Общие рекомендации по организации самостоятельной работы

Основной формой самостоятельной работы студента является изучение конспекта лекций, их дополнение, рекомендованной литературы, активное участие на практических занятиях. Но для успешной учебной деятельности, ее интенсификации, необходимо учитывать следующие субъективные факторы:

1. Знание школьного программного материала, наличие прочной системы знаний, необходимой для усвоения основных вузовских курсов. Это особенно важно для математических дисциплин. Необходимо отличать пробелы в знаниях, затрудняющие усвоение нового материала, от малых способностей. Затратив силы на преодоление этих пробелов, студент обеспечит себе нормальную успеваемость и поверит в свои способности.

2. Наличие умений, навыков умственного труда:

- а) умение конспектировать на лекции и при работе с книгой;
- б) владение логическими операциями: сравнение, анализ, синтез, обобщение, определение понятий, правила систематизации и классификации.

3. Специфика познавательных психических процессов: внимание, память, речь, наблюдательность, интеллект и мышление. Слабое развитие каждого из них становится серьезным препятствием в учебе.

4. Хорошая работоспособность, которая обеспечивается нормальным физическим состоянием. Ведь серьезное учение - это большой многосторонний и разнообразный труд. Результат обучения оценивается не количеством сообщаемой информации, а качеством ее усвоения, умением ее использовать и развитием у себя способности к дальнейшему самостоятельному образованию.

5. Соответствие избранной деятельности, профессии индивидуальным способностям. Необходимо выработать у себя умение саморегулировать свое эмоциональное состояние и устранять обстоятельства, нарушающие деловой настрой, мешающие намеченной работе.

6. Овладение оптимальным стилем работы, обеспечивающим успех в деятельности. Чередование труда и пауз в работе, периоды отдыха, индивидуально обоснованная норма продолжительности сна, предпочтение вечерних или утренних занятий, стрессоустойчивость на экзаменах и особенности подготовки к ним,

7. Уровень требований к себе, определяемый сложившейся самооценкой.

Адекватная оценка знаний, достоинств, недостатков - важная составляющая самоорганизации человека, без нее невозможна успешная работа по управлению своим поведением, деятельностью.

Одна из основных особенностей обучения в высшей школе заключается в том, что постоянный внешний контроль заменяется самоконтролем, активная

роль в обучении принадлежит уже не столько преподавателю, сколько студенту.

Зная основные методы научной организации умственного труда, можно при наименьших затратах времени, средств и трудовых усилий достичь наилучших результатов.

Эффективность усвоения поступающей информации зависит от работоспособности человека в тот или иной момент его деятельности.

Работоспособность - способность человека к труду с высокой степенью напряженности в течение определенного времени. Различают внутренние и внешние факторы работоспособности.

К внутренним факторам работоспособности относятся интеллектуальные особенности, воля, состояние здоровья.

К внешним:

- организация рабочего места, режим труда и отдыха;
- уровень организации труда - умение получить справку и пользоваться информацией;
- величина умственной нагрузки.

Выдающийся русский физиолог Н. Е. Введенский выделил следующие условия продуктивности умственной деятельности:

- во всякий труд нужно входить постепенно;
- мерность и ритм работы. Разным людям присущ более или менее разный темп работы;
- привычная последовательность и систематичность деятельности;
- правильное чередование труда и отдыха.

Отдых не предполагает обязательного полного бездействия со стороны человека, он может быть достигнут простой переменой дела. В течение дня работоспособность изменяется. Наиболее плодотворным является *утреннее время* (с 8 до 14 часов), причем максимальная работоспособность приходится на период с 10 до 13 часов, затем *послеобеденное* - (с 16 до 19 часов) и *вечернее* (с 20 до 24 часов). Очень трудный для понимания материал лучше изучать в начале каждого отрезка времени (лучше всего утреннего) после хорошего отдыха. Через 1-1,5 часа нужны перерывы по 10 - 15 мин, через 3 - 4 часа работы отдых должен быть продолжительным - около часа.

Составной частью научной организации умственного труда является овладение техникой умственного труда.

Время, которым располагает студент для выполнения учебного плана, складывается из двух составляющих: одна из них - это аудиторная работа в вузе по расписанию занятий, другая - внеаудиторная самостоятельная работа. Зада-

ния и материалы для самостоятельной работы выдаются во время учебных занятий по расписанию, на этих же занятиях преподаватель осуществляет контроль за самостоятельной работой, а также оказывает помощь студентам по правильной организации работы.

Самостоятельные занятия потребуют интенсивного умственного труда, который необходимо не только правильно организовать, но и стимулировать. При этом очень важно уметь поддерживать устойчивое внимание к изучаемому материалу. Выработка внимания требует значительных волевых усилий. Именно поэтому, если студент замечает, что он часто отвлекается во время самостоятельных занятий, ему надо заставить себя сосредоточиться. Подобную процедуру необходимо проделывать постоянно, так как это является тренировкой внимания. Устойчивое внимание появляется тогда, когда человек относится к делу с интересом.

Следует правильно организовать свои занятия по времени: 50 минут - работа, 5-10 минут - перерыв; после 3 часов работы перерыв - 20-25 минут. Иначе нарастающее утомление повлечет неустойчивость внимания. Очень существенным фактором, влияющим на повышение умственной работоспособности, являются систематические занятия физической культурой. Организация активного отдыха предусматривает чередование умственной и физической деятельности, что полностью восстанавливает работоспособность человека.

Необходимо подготовиться так, чтобы уверенно ответить на вопросы собеседования с преподавателем.

Так, вопросы для собеседования по результатам выполнения практических работ, то они могут быть следующими:

ПОДКЛЮЧЕНИЕ В РЕЖИМЕ INTERNAL

1. Охарактеризуйте режим INTERNAL (CONNECT INTERNAL)?
2. Каковы требования образуют дополнительный слой защиты, препятствующий несанкционированному запуску или остановке баз данных ORACLE.?
3. Опишите использование пароля для INTERNAL.
4. Дайте характеристику режима INTERNAL и незащищенных соединений.
5. Что представляет собой SQL*Loader?
Каковы основные компоненты SQL*Loader?

ОСНОВНЫЕ КОМПОНЕНТЫ SQL*LOADER

1. Дайте определение домена защиты.
2. Что такое Привилегии?
3. Что представляет собой это логическая коллекция объектов базы данных?
4. Какие примеры привилегий можете привести?

5. Что представляют собой системные привилегии?

ТАБЛИЧНОЕ ПРОСТРАНСТВО SYSTEM

1. Дайте характеристику табличным пространствам.
2. Табличное пространство SYSTEM?
3. Файлов данных?
4. Что представляет собой Объекты?
5. Как называются логические элементы БД?
6. Онлайнновые и офлайнновые табличные пространства?

РАСПРЕДЕЛЕНИЕ ПРОСТРАНСТВА БАЗЫ ДАННЫХ

1. В чем назначение отношений между сегментами, экстендами и блоками данных?
2. Каковы Блоки данных?
3. Каков Формат блока данных ORACLE ?
4. Что такое Экстенды данных?
5. Что такое Сегменты данных?
6. Каковы типы сегментов?
7. ?

ОДНОПРОЦЕССОРНАЯ (ТАКЖЕ НАЗЫВАЕМАЯ ОДНОПОЛЬЗОВАТЕЛЬСКОЙ) СИСТЕМА ORACLE

1. Что представляет собой Структуры памяти?
2. Каковы задачи, работающие в памяти?
3. Какова структура процесса такой системы, как ORACLE?
4. Каковы цели структуры процесса?
5. Что такое однопроцессная (также называемая однопользовательской) система ORACLE?

МНОГОПРОЦЕССНЫЙ ORACLE (НАЗЫВАЕМЫЙ ТАКЖЕ МНОГОПОЛЬЗОВАТЕЛЬСКИМ)

1. Что такое многопроцессная ORACLE ?
2. Что представляет собой глобальная область системы (SGA)
3. Введение в словарь данных.
4. Структура словаря данных.
5. Каковы наборы представлений?

ПРИМЕР РАБОТЫ ORACLE

1. Что представляет собой Структуры памяти?
2. Каковы задачи, работающие в памяти?
3. Какова структура процесса такой системы, как ORACLE?
4. Каковы цели структуры процесса?
5. Что такое однопроцессная (также называемая однопользовательской) система ORACLE?

6. Что такое многопроцессная ORACLE ?
7. Что представляет собой глобальная область системы (SGA)
8. Введение в словарь данных.
9. Структура словаря данных.
10. Каковы наборы представлений?

ЖУРНАЛ ПОВТОРЕНИЙ

1. Что такое методы защиты данных?
2. Что такое Безопасность?
3. Что представляет собой Секретность?
4. Дайте определение идентификации пользователя?
5. Управление доступом?
6. В чем суть физической защиты?

ТРАНЗАКЦИЯ (TRANSACTION)

1. Дайте определение блокировки таблиц, строк, столбцов.
2. Обоснуйте внешние блокировки.
3. Управление доступом?
4. В чем суть физической защиты?

СОЗДАНИЕ УЧЕТНОЙ ЗАПИСИ ПОЛЬЗОВАТЕЛЯ БАЗЫ ДАННЫХ ORACLE

1. Учётная запись пользователя базы данных Oracle.
2. Роли CONNECT, RESOURCE и DBA.
3. Какова команда SQL, которая может использоваться для определения учётной записи Oracle в базе данных?
4. Для чего используется фраза DEFAULT TABLESPACE?
5. Как применяется команда CREATE USER?
6. Когда используется команда DROP USER
7. Что такое аудит базы данных?
8. Установка опций аудита?

РАЗРАБОТКА БАЗЫ ДАННЫХ (ФАЙЛЫ ПАРАМЕТРОВ)

1. Каковы групповые обозначения для опций аудита?
2. Включение и выключение аудита базы данных?
3. Очистка аудиторских записей из аудиторского журнала?
4. Аудит с помощью триггеров базы данных?
5. С какой целью применяется параметр INCREMENT BY?
6. Что такое синонимы?

Важно знать критерии оценки. Которые по данному виду опроса будут следующими:

Оценка «отлично» выставляется, если студент продемонстрировал точное выполнение заданий практической работы, правильно выполнил отчет, активно участвует в работе, владеет материалом, умеет логично и четко излагать мысли,

отвечая на задаваемые вопросы, творчески подходит к решению представленных индивидуальных заданий, показывает самостоятельность мышления.

Оценка «хорошо» выставляется, если студент продемонстрировал точное выполнение заданий практической работы, правильно выполнил отчет, активно участвует в работе, владеет материалом, умеет логично и четко излагать мысли, отвечая на задаваемые вопросы, творчески подходит к решению представленных индивидуальных заданий, показывает самостоятельность мышления., при этом допустил не более одной ошибки.

Оценка «удовлетворительно» выставляется, если студент продемонстрировал точное выполнение заданий практической работы, правильно выполнил отчет, активно участвует в работе, владеет материалом, умеет логично и четко излагать мысли, отвечая на задаваемые вопросы, творчески подходит к решению представленных индивидуальных заданий, показывает самостоятельность мышления, при этом допускает более двух ошибок при ответе на поставленные вопросы.

Оценка «неудовлетворительно» выставляется, если студент не продемонстрировал точное выполнение заданий практической работы, неправильно выполнил отчет, не участвует в работе, не владеет материалом, не умеет логично и четко излагать мысли, отвечая на задаваемые вопросы, не справляется с индивидуальными заданиями, демонстрирует отсутствие самостоятельности мышления и допускает серьезные ошибки при ответах на поставленные вопросы.

Если рассматривать вопросы собеседования по вопросам для самостоятельного изучения литературы, то в качестве таких вопросов могут быть следующие:

ТЕМА 1. ОБЯЗАННОСТИ АДМИНИСТРАТОРА БАЗЫ ДАННЫХ (АБД)

1. Охарактеризуйте установку и обновление версий сервера ORACLE и прикладных инструментов.

2. Дайте описание процессу распределение дисковой памяти и планирование будущих требований системы к памяти.

3. В чем суть обязанности администратора: создание первичных структур памяти в базе данных (табличных пространств) по мере проектирования приложений разработчиками приложений?

4. В чем суть обязанности администратора: создание первичных объектов (таблиц, представлений, индексов) по мере проектирования приложений разработчиками.

5. В чем суть обязанности администратора: модификация структуры базы данных в соответствии с потребностями приложений?

6. Что представляет собой процесс зачисления пользователей и поддержание защиты системы?

7. Что представляет собой процесс соблюдения лицензионного соглашения ORACLE?

8. Как вы думаете, управление и отслеживание доступа пользователей к базе данных входят в обязанности администратора?

ТЕМА 2. ПОДКЛЮЧЕНИЕ В РЕЖИМЕ INTERNAL

1. Охарактеризуйте режим INTERNAL (CONNECT INTERNAL)?

2. Каковы требования образуют дополнительный слой защиты, препятствующий несанкционированному запуску или остановке баз данных ORACLE.?

3. Опишите использование пароля для INTERNAL.

4. Дайте характеристику режима INTERNAL и незащищенных соединений.

5. Что представляет собой SQL*Loader?

6. Каковы основные компоненты SQL*Loader?

ТЕМА 3. ПОЛЬЗОВАТЕЛИ БАЗЫ ДАННЫХ И СХЕМЫ

1. Дайте определение домена защиты.

2. Что такое Привилегии?

3. Что представляет собой это логическая коллекция объектов базы данных?

4. Какие примеры привилегий можете привести?

5. Что представляют собой системные привилегии?

ТЕМА 4. ТАБЛИЧНЫЕ ПРОСТРАНСТВА И ФАЙЛЫ ДАННЫХ

1. Дайте характеристику табличным пространствам.

2. Табличное пространство SYSTEM?

3. Файлов данных?

4. Что представляет собой Объекты?

5. Как называются логические элементы БД?

6. Онлайнные и офлайнные табличные пространства

ТЕМА 5. БЛОКИ ДАННЫХ, ЭКСТЕНТЫ И СЕГМЕНТЫ

1. В чем назначение отношений между сегментами, экстендами и блоками данных?

2. Каковы Блоки данных?

3. Каков Формат блока данных ORACLE?

4. Что такое Экстенды данных?

5. Что такое Сегменты данных?

6. Каковы типы сегментов?

ТЕМА 6. СТРУКТУРЫ ПАМЯТИ И ПРОЦЕССЫ

1. Что представляет собой Структуры памяти?
2. Каковы задачи, работающие в памяти?
3. Какова структура процесса такой системы, как ORACLE?
4. Каковы цели структуры процесса?
5. Что такое однопроцессная (также называемая однопользовательской) система ORACLE?

6. Что такое многопроцессная ORACLE?
7. Что представляет собой глобальная область системы (SGA)
8. Введение в словарь данных.
9. Структура словаря данных.
10. Каковы наборы представлений?
11. Представления с префиксом USER.
12. Представления с префиксом ALL.

ТЕМА 7 ОБЕСПЕЧЕНИЕ ЗАЩИТЫ БАЗЫ ДАННЫХ.

1. Что такое методы защиты данных?
2. Что такое Безопасность?
3. Что представляет собой Секретность?
4. Дайте определение идентификации пользователя?
5. Управление доступом?
6. В чём суть физической защиты?
7. В чём заключается восстановление данных?
8. Каковы особенности защиты данных средствами Oracle?

ТЕМА 8 УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ БАЗЫ ДАННЫХ.

1. Учётная запись пользователя базы данных Oracle.
2. Роли CONNECT, RESOURCE и DBA.
3. Какова команда SQL, которая может использоваться для определения учётной записи Oracle в базе данных?
4. Для чего используется фраза DEFAULT TABLESPACE?
5. Как применяется команда CREATE USER?
6. Когда используется команда DROP USER
7. Что такое аудит базы данных?
8. Установка опций аудита?
9. Каковы групповые обозначения для опций аудита?
10. Включение и выключение аудита базы данных?
11. Очистка аудиторских записей из аудиторского журнала?
12. Аудит с помощью триггеров базы данных?
13. С какой целью применяется параметр INCREMENT BY?
14. Что такое синонимы?

При этом критерии оценки:

оценка «отлично» выставляется, если студент безупречно отвечает на заданные вопросы, в полной мере понимает угрозы безопасности БД и способы их предотвращения. Без особого труда применяет инструменты обеспечения безопасности БД и их возможности. Реализует необходимые средства и инструменты восстановления безопасности на уровне БД. Самостоятельно определяет угрозы безопасности на уровне БД. Имеет прочные навыки конфигурирования БД. Самостоятельно и грамотно осуществляет резервное копирования и восстановления БД. В полном объеме демонстрирует методы администрирования БД и повышения производительности запросов СУБД.

оценка «хорошо» выставляется, если студент отвечает на заданные вопросы с некоторыми неточностями, при этом понимает угрозы безопасности БД и способы их предотвращения. Применяет инструменты обеспечения безопасности БД и их возможности. Реализует средства и инструменты восстановления безопасности на уровне БД. Определяет угрозы безопасности на уровне БД. Имеет навыки конфигурирования БД. Осуществляет резервное копирования и восстановления БД. Демонстрирует методы администрирования БД и повышения производительности запросов СУБД.

оценка «удовлетворительно» выставляется, если студент неточно отвечает на заданные вопросы, частично понимает угрозы безопасности БД и способы их предотвращения. Частично применяет инструменты обеспечения безопасности БД и их возможности. В недостаточной степени реализует средства и инструменты восстановления безопасности на уровне БД. Частично определяет угрозы безопасности на уровне БД. Имеет частичные навыки конфигурирования БД. Недостаточно осуществляет резервное копирования и восстановления БД. Демонстрирует некоторые методы администрирования БД и повышения производительности запросов СУБД.

оценка «неудовлетворительно» выставляется, если студент не отвечает на заданные вопросы, недостаточно понимает угрозы безопасности БД и способы их предотвращения. С трудом применяет инструменты обеспечения безопасности БД и их возможности. Не реализует средства и инструменты восстановления безопасности на уровне БД. Не определяет угрозы безопасности на уровне БД. Не имеет навыков конфигурирования БД. Не осуществляет резервное копирование и восстановление БД. С трудом демонстрирует методы администрирования БД и повышения производительности запросов СУБД.

6. Самостоятельная работа студента - необходимое звено становления исследователя

Прогресс науки и техники, информационных технологий приводит к значительному увеличению научной информации, что предъявляет более высокие требования не только к моральным, нравственным свойствам человека, но и в особенности, постоянно возрастающие требования в области образования – обновление, модернизация общих и профессиональных знаний, умений специалиста.

Всякое образование должно выступать как динамический процесс, присущий человеку и продолжающийся всю его жизнь. Овладение научной мыслью и языком науки является необходимой составляющей в самоорганизации будущего специалиста исследователя. Под этим понимается не столько накопление знаний, сколько овладение научно обоснованными способами их приобретения. В этом, вообще говоря, состоит основная задача вуза.

Специфика вузовского учебного процесса, в организации которого самостоятельной работе студента отводятся все больше места, состоит в том, что он является как будто бы последним и самым адекватным звеном для реализации этой задачи. Ибо во время учебы в вузе происходит выработка стиля, навыков учебной (познавательной) деятельности, рациональный характер которых будет способствовать постоянному обновлению знаний высококвалифицированного выпускника вуза.

Однако до этого пути существуют определенные трудности, в частности, переход студента от синтетического процесса обучения в средней школе, к аналитическому в высшей. Это связано как с новым содержанием обучения (расширение общего образования и углубление профессиональной подготовки), так и с новыми, неизвестными до сих пор формами: обучения (лекции, семинары, практические занятия и т.д.). Студент получает не только знания, предусмотренные программой и учебными пособиями, но он также должен познакомиться со способами приобретения знаний так, чтобы суметь оценить, что мы знаем, откуда мы это знаем и как этого знания мы достигли. Ко всему этому приходят через собственную самостоятельную работу.

Это и потому, что самостоятельно приобретенные знания являются более оперативными, они становятся личной собственностью, а также мотивом поведения, развивают интеллектуальные черты, внимание, наблюдательность, критичность, умение оценивать. Роль преподавателя в основном заключается в руководстве накопления знаний (по отношению к первокурсникам), а в последующие годы учебы, на старших курсах, в совместном установлении проблем и заботе о самостоятельных поисках студента, а также контролирования за их де-

тельностью. Отметим, что нельзя ограничиваться только приобретением знаний предусмотренных программой изучаемой дисциплины, надо постоянно углублять полученные знания, сосредотачивая их на какой-нибудь узкой определенной области, соответствующей интересам студента. Углубленное изучение всех предметов, предусмотренных программой, на практике является возможным, и хорошая организация работы позволяет экономить время, что создает условия для глубокого, систематического, заинтересованного изучения самостоятельно выбранной студентом темы.

Конечно, все советы, примеры, рекомендации в этой области, даваемые преподавателем, или определенными публикациями, или другими источниками, не гарантируют никакого успеха без проявления собственной активности в этом деле, т. е. они не дают готовых рецептов, а должны способствовать анализу собственной работы, ее целей, организации в соответствии с индивидуальными особенностями. Учитывая личные возможности, существующие условия жизни и работы, навыки, на основе этих рекомендаций, возможно, выработать индивидуально обоснованную совокупность методов, способов, найти свой стиль или усовершенствовать его, чтобы изучив определенный материал, иметь время оценить его значимость, пригодность и возможности его применения, чтобы, в конечном счете, обеспечить успешность своей учебы с будущей профессиональной деятельности

7. Методические рекомендации для студентов по отдельным формам самостоятельной работы

Система вузовского обучения подразумевает значительно большую самостоятельность студентов в планировании и организации своей деятельности.

Работа с литературой

При работе с литературой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Различают два вида чтения; первичное и вторичное. *Первичное* - это внимательное, неторопливое чтение, при котором можно остановиться на трудных местах. После него не должно остаться ни одного непонятного слова. Содержание не всегда может быть понятно после первичного чтения.

Задача *вторичного* чтения - полное усвоение смысла целого (по счету это чтение может быть и не вторым, а третьим или четвертым).

Чтение научного текста является частью познавательной деятельности. Ее цель - извлечение из текста необходимой информации. От того на сколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют **четыре основные установки в чтении научного текста:**

1. информационно-поисковый (задача - найти, выделить искомую информацию)
2. усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения излагаемые автором, так и всю логику его рассуждений)
3. аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)
4. творческая (создает у читателя готовность в том или ином виде - как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. - использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

С наличием различных установок обращения к научному тексту связано существование и нескольких **видов чтения**:

1. библиографическое – просматривание карточек каталога, рекомендательных списков, сводных списков журналов и статей за год и т.п.;
2. просмотрное – используется для поиска материалов, содержащих нужную информацию, обычно к нему прибегают сразу после работы со списками литературы и каталогами, в результате такого просмотра читатель устанавливает, какие из источников будут использованы в дальнейшей работе;
3. ознакомительное – подразумевает сплошное, достаточно подробное прочтение отобранных статей, глав, отдельных страниц, цель – познакомиться с характером информации, узнать, какие вопросы вынесены автором на рассмотрение, провести сортировку материала;
4. изучающее – предполагает доскональное освоение материала; в ходе такого чтения проявляется доверие читателя к автору, готовность принять изложенную информацию, реализуется установка на предельно полное понимание материала;
5. аналитико-критическое и творческое чтение – два вида чтения близкие между собой тем, что участвуют в решении исследовательских задач. Первый из них предполагает направленный критический анализ, как самой информации, так и способов ее получения и подачи автором; второе – поиск тех суждений, фактов, по которым или в связи с которыми, читатель считает нужным высказать собственные мысли.

Из всех рассмотренных видов чтения основным для студентов является изучающее – именно оно позволяет в работе с учебной литературой накапливать знания в различных областях. Вот почему именно этот вид чтения в рамках учебной деятельности должен быть освоен в первую очередь. Кроме того, при овладении данным видом чтения формируются основные приемы, повышающие эффективность работы с научным текстом.

Основные виды систематизированной записи прочитанного:

1. Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;
2. Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;
3. Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;
4. Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;

5. Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

Методические рекомендации по составлению конспекта:

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта.

2. Выделите главное, составьте план.

3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора.

4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.

5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

Практические занятия

Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью выполнения заданий практической работы, поставленных задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также

получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном выполнении заданий нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что выполнение каждого учебного задания должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Выполнение заданий данного типа нужно продолжать до приобретения твердых навыков в их решении.

Самопроверка

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих заданий на практических занятиях и самостоятельно студенту рекомендуется, используя лист опорных материалов, воспроизвести по памяти определения, выводы формул, формулировки основных положений и доказательств.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала - умение решать задачи или пройти тестирование по пройденному материалу. Однако следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

Консультации

Если в процессе самостоятельной работы над изучением теоретического материала или при выполнении заданий у студента возникают вопросы, разрешить которые самостоятельно не удастся, необходимо обратиться к преподавателю для получения у него разъяснений или указаний. В своих вопросах студент должен четко выразить, в чем он испытывает затруднения, характер этого затруднения. За консультацией следует обращаться и в случае, если возникнут сомнения в правильности ответов на вопросы самопроверки.

Подготовка к зачету с оценкой по дисциплине «Администрирование баз данных».

Изучение многих общепрофессиональных и специальных дисциплин завершается зачетом. Подготовка к зачету способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к зачету, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания.

Для получения зачета студенту необходимо полностью выполнить задания практических работ занятий, подготовить отчет по каждой практической работе и отчет по результатам самостоятельного изучения тем, согласно учебной программе. При наличии задолженностей по текущей аттестации по данной дисциплине студент не получает зачет.

Систематическая подготовка к занятиям в течение семестра позволит использовать время экзаменационной сессии для систематизации знаний.

Текущая аттестация студентов проводится преподавателями, ведущими практические занятия по дисциплине в следующих формах: собеседование при защите практической работы с предоставлением письменного отчета. Допуск к практическим работам происходит при наличии у студентов печатного варианта отчета. Защита отчета проходит в форме собеседования. Максимальное количество баллов студент получает, если оформление отчета соответствует установленным требованиям, а отчет полностью раскрывает суть работы. Основанием для снижения оценки являются:

- не качественно выполненная работа на персональном компьютере,
- письменный отчет выполнен в небрежной форме и с ошибками,
- студент не дает исчерпывающий ответ на вопрос преподавателя,
- путается или затрудняется в демонстрации работы на компьютере.

Отчет может быть отправлен на доработку в следующих случаях:

- небрежное выполнение и с ошибками,
- сокращенное содержание отчета.

Критерии оценивания собеседования приведены в Фонде оценочных средств по дисциплине «Администрирование баз данных».

8. Учебно-методическое и информационное обеспечение дисциплины

По данной дисциплине преподаватель рекомендует следующую литературу:

В качестве основной литературы рассматриваются:

Култыгин, О. П. Администрирование баз данных. СУБД MS SQL Server : Учебное пособие / Култыгин О. П. - Москва : Московский финансово-промышленный университет «Синергия», 2012. - 232 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-4257-0026-1

В качестве дополнительной литературы рассматриваются:

1. Федотов, Е. А. Администрирование программных и информационных систем : Учебное пособие / Федотов Е. А. - Белгород : Белгородский государственный технологический университет им. В.Г. Шухова, ЭБС АСВ, 2012. - 136 с. - Книга находится в базовой версии ЭБС IPRbooks.

2. Гудов, А. М. Базы данных и системы управления базами данных. Программирование на языке PL/SQL / А.М. Гудов ; С.Ю. Завозкин ; Т.С. Рейн. - Кемерово : Кемеровский государственный университет, 2010. - 134 с. - ISBN 978-5-8353-1005-0

3. Братченко, Н. Ю. (СевКавГТУ). Базы данных : учеб. пособие / Н. Ю. Братченко ; Сев.-Кав. гос. техн. ун-т. – Ставрополь : СевКавГТУ, 2011. – 195 с. – Библиогр.: с. 194 (Электронная библиотека Фолиант) <http://catalog.ncfu.ru/>

Методическая литература

1. Практикум по администрированию программного обеспечения : лабораторный практикум : Направление подготовки 10.05.03 Информационная безопасность автоматизированных систем. Профиль подготовки «Защищенные автоматизированные системы управления» / сост. И. В. Анзин ; Сев.-Кав федер. ун-т. - Ставрополь : СКФУ, 2017. - 85 с., экземпляров неограничено

Интернет ресурсы:

Официальный сайт фирмы «1С» <http://1c.ru/>

Единое окно доступа к образовательным ресурсам: [портал]. URL: <http://window.edu.ru/>

Национальный открытый университет ИНТУИТ. URL: <http://www.intuit.ru/>

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
для выполнения практических работ
по дисциплине «Администрирование баз данных»

по направлению подготовки
09.03.01 Информатика и вычислительная техника,
Направленность (профиль) Программное обеспечение робототехнических
и интеллектуальных систем

Ставрополь, 2025

СОДЕРЖАНИЕ

Практическая работа. Подключение в режиме INTERNAL.

Практическая работа. Основные компоненты SQL*Loader.

Практическая работа. Табличное пространство SYSTEM.

Практическая работа. Распределение пространства базы данных.

Практическая работа. Однопроцессная (также называемая однопользовательской) система ORACLE.

Практическая работа. Многопроцессный ORACLE (называемый также многопользовательским).

Практическая работа. Пример работы Oracle.

Практическая работа. Журнал Повторений.

Практическая работа. Транзакция (Transaction) .

Практическая работа. Разработка базы данных (файлы параметров).

Практическая работа. Создание учётной записи пользователя базы данных Oracle.

ПРАКТИЧЕСКАЯ РАБОТА.

ПОДКЛЮЧЕНИЕ В РЕЖИМЕ INTERNAL

Цель и содержание

Цель работы: работа с БД в режиме INTERNAL средствами ORACLE.

Теоретическое обоснование

Запуск и останов базы данных - это мощные административные возможности. В угоду поддержания корректной работоспособности базы данных, функции (команды **STARTUP** или **SHUTDOWN**) остановки и запуска разрешены, только для администратора подключенного к ORACLE в режиме INTERNAL (**CONNECT INTERNAL**), а для возможности подключиться в режиме INTERNAL, вы должны соответствовать одному из ниже следующих условий:

- Ваше учетное имя в операционной системе имеет привилегии операционной системы, позволяющие вам соединяться в режиме INTERNAL.
- Вы имеете полномочия соединяться в режиме INTERNAL.
- Ваша база данных имеет пароль для INTERNAL, и вы знаете этот пароль.

Все эти требования образуют дополнительный слой защиты, препятствующий несанкционированному запуску или остановке баз данных ORACLE. Для систем, имеющих пароль для INTERNAL, существуют дополнительные, ниже описываемые соображения.

Использование пароля для INTERNAL

Некоторые операционные системы позволяют устанавливать пароль для соединений в режиме INTERNAL. Можно установить пароль для INTERNAL во время инсталляции сервера ORACLE, Oracle предоставляет утилиту для управления этим паролем (создания, изменения и удаления его).

INTERNAL и незащищенные соединения

Если используется незащищенное соединение(как большинство сетевых соединений), то ДОЛЖНО использовать пароль для INTERNAL, для последующего подключения в режиме INTERNAL; это требование подразумевает, что в системе должен быть установлен пароль для INTERNAL.

В некоторых О.С. можно либо включить, либо полностью отключить возможность соединений **CONNECT INTERNAL** для незащищенных соединений. Выбор делается во время инсталляции ORACLE, и может быть изменен позднее.

Использование пароля для INTERNAL.

Одной из многих проблем, с которыми часто сталкиваются администраторы базы данных, является перемещение данных из внешних источников в базу данных Oracle. Сложность этой задачи возрастает с появлением хранилищ данных, приходится перемещать уже не мегабайты данных, а гигабайты, а в некоторых случаях – терабайты. Oracle предусматривает для решения этой задачи SQL*Loader – универсальное инструментальное средство, которое загружает внешние данные в таблицы базы данных Oracle. Утилита SQL*Loader является гибкой и настраиваемой до такой степени, что часто удастся обойтись без процедур на языке третьего поколения с внедренными операторами SQL. Каждый раз, сталкиваясь с задачей преобразования инородных данных в формат Oracle, вначале рассмотрите возможность применения SQL*Loader, прежде чем обращаться к другим средствам.

Для утилиты SQL*Loader необходимы входные данные 2-ух типов: внешние данные, которые могут находиться на диске или ленте, и управляющая информация (содержащаяся в управляющем файле), которая описывает характеристики входных данных. Выходные данные, часть которых является необязательной, включает таблицы Oracle, журналы, файлы некорректных записей и файлы отвергнутых записей.

Утилита SQL*Loader может обрабатывать файлы данных практически любого типа и поддерживает собственные типы данных почти любой платформы. Данные обычно считываются из одного или нескольких файлов данных, однако они могут быть также внесены в управляющий файл после управляющей информации. Файл данных может находиться:

В файлах с переменным форматом данные находятся в записях, которые могут изменяться по длине, в зависимости от размеров данных в полях. Поля имеют длину, необходимую для размещения данных. Поля в файлах с переменным форматом могут быть разделены завершающими символами (такими как запятые и пробелы), а так же заключены в ограничительные символы.

Прежде чем утилита SQL*Loader сможет обработать данные в файлах данных, необходимо знать определения данных для SQL*Loader. Используйте управляющий файл для указания физических определений файла данных, а также формата данных в файлах. Управляющий файл – это файл произвольного формата, который также содержит дополнительные управляющие данные, указывающие SQL*Loader, как обрабатывать эти данные.

Журнал

После выполнения утилита SQL*Loader создает журнал, содержащий подробную информацию о загрузке, включая, следующие сведения:

- Имена фалов входных данных, управляющего файла, файлов некорректных записей и файлов отвергнутых записей.
- Входные данные и связанные с ними определения таблиц
- Ошибки SQL*Loader
- Результаты работы SQL*Loader
- Итоговую статистику

Import и Export

Import и Export — две дополнительные утилиты, поставляемые корпорацией Oracle. Они в основном применяются для копирования и восстановления данных и для перемещения данных либо в другую базу данных Oracle, либо из более старой версии Oracle в более новую. Ниже приведены другие возможности утилит Import и Export :

- Хранение данных в файлах операционной системы для архивирования
- Выборочное резервное копирование частей базы данных
- Перемещение данных из одной пользовательской схемы Oracle в другую
- Перемещение данных с одной аппаратной платформы или операционной системы в другую
- Экономия пространства и повышение производительности за счет уменьшения фрагментации

Работа с утилитами Import и Export весьма проста. Утилита Export записывает информацию о таблицах или объектах базы данных, такую как операторы создания таблицы, операторы создания индекса, разрешения на таблицу, информация о размерах и т.д., а также данные из самих таблиц Oracle. Затем утилита Export сохраняет эту информацию в именованных файлах операционной системы. Файлы операционной системы, создаваемые утилитой Export, известны как файлы дампа. Файлы дампа, которые представлены в двоичном формате Oracle, применяются главным образом только в утилите Import. Можно назвать Файл дампа любым именем, допустимым в операционной системе. Если вы не укажете имя выходного файла для утилиты Export, то по умолчанию будет принято имя EXPDAT.DMP.

Затем можно сохранить выходные файлы, созданные Export, на диске или записать на съемный носитель для дальнейшего хранения, либо воспользоваться утилитой Import для воссоздания экспортируемых данных в целях восстановления или ведения базы данных.

Export

Иногда можно обнаружить, что вы не сделали в свое время то, в чем сейчас остро нуждаетесь. Возьмем, например, утилиты Import и Export. Утилита

Export — это самый удобный способ застраховаться от возможных неприятностей. Экспорт — это универсальная утилита, поставляемая корпорацией Oracle. При всей заложенной в ней гибкости ею довольно легко пользоваться на основе обширного списка параметров. Разнообразие параметров дает возможность воспользоваться утилитой Export для решения сложных проблем управления данными. Утилита Export может записывать файлы операционной системы, которые затем можно перемещать в другую операционную систему или версию Oracle.

Проверьте, чтобы для хранения экспортного файла на запоминающем устройстве было достаточно свободного пространства. Можно использовать представление `user_segments` для оценки необходимого дискового пространства.

В следующем коде показан пример использования утилиты Export:

```
exp userid=system/manager OWNER=scott... [прочие опции]
```

Использование файла параметров

Можно использовать файл параметров как для утилиты Export, так и для утилиты Import. Файл параметров помогает выполнять операции импорта и экспорта, обеспечивая при этом непротиворечивость и простоту. Его удобно использовать для экспорта в ночное время. Файлы параметров гарантируют целостный экспорт для полной уверенности в том, что все необходимые таблицы действительно будут экспортированы. Можно вызывать эти утилиты, задавая параметры в командной строки или в сценарии операционной системы, но использование командной строки может не позволить задать все необходимые параметры. Сценарий `export_ts` записывает необходимый ему файл параметров. Имена таблиц в кавычках являются чувствительными к регистру.

Режим экспорта таблиц

Режим экспорта таблиц используется для экспорта одной таблицы или перечня таблиц, а не всей базы данных. По умолчанию он экспортирует все таблицы, которые принадлежат пользователю, выполняющему экспорт. Пользователи, имеющие доступ к другой схеме, могут экспортировать таблицы из этой схемы, указав имя схемы.

Режим экспорта пользователя

Режим экспорта пользователя в основном используется для экспорта всех таблиц и индексов конкретного пользователя или перечня пользователей. Этот режим работает хорошо при создании пользователя, который является владельцем всех объектов приложения. Например, если существует пользователь с именем `sales`, который является владельцем всех таблиц и

индексов и других объектов в приложении sales, экспорт приложения может выглядеть следующим образом:

```
exp VSERID=system/manager OWNER=sales
```

Режим экспорта всей базы данных

Режим экспорта всей базы данных используется для экспорта всех объектов базы данных, за исключением объектов, которые обычно создаются и поддерживаются учетной записью SYS. Эту опцию могут применять только пользователи, которым назначена роль EXP_FULL_DATABASE. Здесь можно упомянуть несколько других интересных возможностей. По умолчанию Oracle выполняет полный экспорт при указании режима экспорта всей базы данных (INCTYPE= COMPLETE). Если указана опция INCTYPE= INCREMENTAL, Oracle будет экспортировать только таблицы, содержащие какие-либо изменившиеся строки, начиная с последнего полного экспорта любого типа. Если указана опция INCTYPE=CUMULATIVE, Oracle будет экспортировать только таблицы, содержащие какие-либо измененные строки, начиная с последнего полного или кумулятивного экспорта.

Типы экспорта:

- Полный экспорт
- Инкрементный экспорт
- Кумулятивный экспорт

Import

Утилита Import противоположна утилите Export. Она отвечает за чтение экспортных файлов в целях воссоздания объектов базы данных, а также любого состояния, в котором они экспортировались первоначально. Утилита Import может также преобразовывать данные предоставленные с разных платформ к примеру с UNIX машины в ASCII кодах, на мейнфрем с кодировкой EBCDIC и наоборот, что позволяет перемещать данные с одной платформы на другую. Утилита Import может работать в интерактивном режиме или в режиме командной строки. При использовании интерактивного режима утилита Import запрашивает у пользователя параметры, необходимые для выполнения импорта. Обычно проще задать параметры в командной строке или в файле параметров. Утилита Import, как и Export, использует файлы параметров.

Оборудование и материалы

Для выполнения практической работы необходим персональный компьютер с установленной СУБД Oracle10g/11g ExpressEdition и пакетом Microsoft Office. Инсталляция пакета представлена в приложении.

Указания по технике безопасности

Необходимо выполнять требования по технике безопасности, предусмотренные для компьютерного класса

1. Не включать компьютер без разрешения администратора класса и преподавателя.
2. Запрещается самостоятельно вскрывать корпус ПК.
3. Не касаться одновременно экрана монитора и клавиатуры (возможен разряд повышенного электростатического потенциала).
4. Не загромождать верхние панели устройств бумагами и посторонними устройствами.
5. Не прикасаться одновременно к металлическим частям ПК и устройствам, имеющим естественное заземление.
6. Запрещается производить отключение питания во время выполнения активной задачи.
7. Не допускать попадание влаги на поверхность системного блока, монитора, клавиатуру, принтеры и другие устройства.

Порядок выполнения работы

1. На основе созданной БД установите пароль для соединений в режиме INTERNAL.
2. Установите пароль для INTERNAL во время инсталляции сервера .
3. Загрузите внешние данные и таблицы БД Oracle средствами SQL*Loader.
4. Используйте управляющий файл для указания физических определений файла данных.
5. Выполните утилиту SQL*Loader.
6. Отследите ошибки SQL*Loader.
7. Поясните результаты работы SQL*Loader
8. Охарактеризуйте итоговую статистику.
9. Выполните выборочное резервное копирование частей базы данных.

Содержание отчета и его форма

Отчет должен содержать

- название практической работы;
- цель работы;
- порядок выполнения работы;
- результаты выполнения;
- вывод о проделанной работе.

Контрольные вопросы и защита работы

Контрольные вопросы

6. Охарактеризуйте режим NTERNAL (CONNECT INTERNAL)?
7. Каковы требования образуют дополнительный слой защиты, препятствующий несанкционированному запуску или остановке баз данных ORACLE.?

8. Опишите использование пароля для INTERNAL.
9. Дайте характеристику режима INTERNAL и незащищенных соединений.
10. Что представляет собой SQL*Loader?
11. Каковы основные компоненты SQL*Loader?

Защита работы

Защита работы заключается в:

- выполнении заданий;
- ответах на контрольные вопросы;
- предоставлении отчета.

ПРАКТИЧЕСКАЯ РАБОТА.

ОСНОВНЫЕ КОМПОНЕНТЫ SQL*LOADER

Цель и содержание

Цель работы: работа с утилитой SQL*Loader.

Теоретическое обоснование

Каждая база данных ORACLE имеет список имен пользователей. Чтобы получить доступ к базе данных, пользователь должен использовать приложение базы данных, и попытаться соединиться с базой данных, предоставив действительное имя пользователя. С каждым именем пользователя связан пароль, чтобы предотвратить несанкционированный доступ.

С каждым именем пользователя ассоциирована SCHEMA(SCHEMA), имеющая такое же имя. SCHEMA(схема) - это логическая коллекция объектов базы данных (таблиц, представлений, последовательностей, синонимов, индексов, кластеров, процедур, функций, пакетов и связей баз данных). По умолчанию, каждый пользователь базы данных создает и имеет доступ ко всем объектам в соответствующей схеме. Команда CREATE SCHEMA полезна для гарантирования успешного создания всех нужных объектов и грантов за одну операцию; если индивидуальный объект внутри этой операции не может быть создан, все предложение откатывается и аннулируются все его результаты.

Каждый пользователь имеет ДОМЕН ЗАЩИТЫ - набор свойств, которые определяют такие вещи, как:

- действия (привилегии и роли), доступные пользователю
- квоты табличных пространств (доступной дисковой памяти)
- лимиты на системные ресурсы (например, время процессора) для данного пользователя

Каждое свойство, вносящее вклад в домен защиты пользователя, обсуждается ниже.

ПРИВИЛЕГИЯ(PRIVILEGE) - это право выполнять определенный тип предложений SQL. Некоторые примеры привилегий включают:

- право соединяться с базой данных (создавать сессию)
- право создавать таблицу в вашей схеме
- право выбирать строки из чьей-либо таблицы
- право выполнять чью-либо хранимую процедуру

Привилегии в базе данных ORACLE могут быть разделены на две различные категории: системные привилегии и объектные привилегии.

СИСТЕМНЫЕ ПРИВИЛЕГИИ(SYSTEM_PRIVILEGE) позволяют пользователям выполнять конкретное действие на уровне системы, или конкретное действие над конкретным типом объектов. Таковы, например, привилегия создавать табличное пространство или привилегия удалять строки любой таблицы в базе данных. Большинство системных привилегий доступны только администраторам и разработчикам приложений, поскольку такие привилегии весьма мощны.

ОБЪЕКТНЫЕ ПРИВИЛЕГИИ позволяют пользователям выполнять конкретные действия на конкретном объекте. Такова, например, привилегия удалять строки в указанной таблице. Объектные привилегии назначаются пользователям, так что они могут использовать приложения базы данных для выполнения конкретных задач.

Назначение привилегий

Привилегии назначаются пользователям с тем, чтобы эти пользователи могли обращаться к данным и модифицировать эти данные в базе данных. Пользователь может получить привилегию двумя различными способами:

- Привилегии могут назначаться пользователям явно. Например, привилегия вставлять записи в таблицу EMP может быть явно назначена пользователю SCOTT.
- Привилегии могут назначаться РОЛЯМ (именованным группам привилегий), а затем эта роль может быть назначена одному или нескольким пользователям. Например, привилегия вставлять записи в таблицу EMP может быть назначена роли с именем CLERK, а эта роль, в свою очередь, может быть назначена пользователям SCOTT и BRIAN.

ORACLE предоставляет легкое и контролируемое управление привилегиями через использование ролей. РОЛИ (ROLE) - это поименованные группы взаимосвязанных привилегий, которые назначаются пользователям или другим ролям.

Оборудование и материалы

Для выполнения практической работы необходим персональный компьютер с установленной СУБД Oracle10g/11g ExpressEdition и пакетом Microsoft Office. Инсталляция пакета представлена в приложении.

Указания по технике безопасности

Необходимо выполнять требования по технике безопасности, предусмотренные для компьютерного класса

1. Не включать компьютер без разрешения администратора класса и

преподавателя.

2. Запрещается самостоятельно вскрывать корпус ПК.
3. Не касаться одновременно экрана монитора и клавиатуры (возможен разряд повышенного электростатического потенциала).
4. Не загромождать верхние панели устройств бумагами и посторонними устройствами.
5. Не прикасаться одновременно к металлическим частям ПК и устройствам, имеющим естественное заземление.
6. Запрещается производить отключение питания во время выполнения активной задачи.
7. Не допускать попадание влаги на поверхность системного блока, монитора, клавиатуру, принтеры и другие устройства.

Порядок выполнения работы

1. Произведите преобразование данных перед или прямо во время самой их загрузки (правда, ограниченным образом).
2. Выполнять загрузку данных из нескольких видов источников: с дисков, с лент и из именованных каналов, а также использовать несколько файлов данных в одном и том же сеансе загрузки.
3. Выполните загрузку данных по сети.
4. Осуществите выборочную загрузку данных из входного файла на основании различных условий.
5. Произведите загрузку как всей таблицы, так и только какой-то конкретной ее части, а также загрузите данные одновременно в несколько таблиц.
6. Проведите одновременные операции по загрузке данных.
7. Автоматизируйте процесс загрузки, так что бы он выполнялся в запланированное время.
8. Загрузите сложные объектно-реляционные данные.

Содержание отчета и его форма

Отчет должен содержать

- название практической работы;
- цель работы;
- порядок выполнения работы;
- результаты выполнения;
- вывод о проделанной работе.

Контрольные вопросы и защита работы

Контрольные вопросы

6. Дайте определение домена защиты.
7. Что такое Привилегии?
8. Что представляет собой это логическая коллекция объектов базы данных?

9. Какие примеры привилегий можете привести?
10. Что представляют собой системные привилегии?

Защита работы

Защита работы заключается в:

- выполнении заданий;
- ответах на контрольные вопросы;
- предоставлении отчета.

ПРАКТИЧЕСКАЯ РАБОТА.

ТАБЛИЧНОЕ ПРОСТРАНСТВО SYSTEM

Цель и содержание

Цель работы: работа с табличными пространствами.

Теоретическое обоснование

Используемые данные базы данных ORACLE логически хранятся в **ТАБЛИЧНЫХ ПРОСТРАНСТВАХ (TABLESPACE)**, а физически располагаются в **ФАЙЛАХ ДАННЫХ**, ассоциированных с соответствующим табличным пространством. Эту взаимосвязь иллюстрирует следующий рисунок:

Файлы данных - физические структуры, каждая из которых связана с одним табличным пространством

Объекты - Хранятся в табличных пространствах, и могут располагаться в нескольких файлах данных

Каждая база данных ORACLE подразделяется на логические элементы, называемые **ТАБЛИЧНЫМИ ПРОСТРАНСТВАМИ (TABLESPACE)**. Администратор базы данных может использовать табличные пространства для:

- управления распределением памяти для объектов базы данных
- установления квот памяти для пользователей базы данных
- управления доступностью данных, путем перевода отдельных табличных пространств в состояния online или offline
- копирования и восстановления данных
- распределения данных по устройствам для повышения производительности

Каждая база данных ORACLE содержит табличное пространство SYSTEM, которое создается автоматически при создании базы данных. Табличное пространство SYSTEM всегда содержит таблицы словаря данных для всей базы данных. Небольшой базе данных может оказаться достаточным одного табличного пространства SYSTEM; однако рекомендуется создать по меньшей мере одно дополнительное пространство, чтобы хранить данные пользователей отдельно от информации словаря данных. Это позволит более гибко осуществлять разнообразные операции администрирования.

Размер табличного пространства - это размер его файла данных или суммарный размер всех файлов данных, составляющих это табличное пространство.

АБД может перевести любое табличное пространство в базе данных ORACLE в состояние **ОНЛАЙН** (т.е. доступно) или **ОФЛАЙН** (недо-

ступно), если база данных открыта. Единственным исключением является то, что табличное пространство SYSTEM всегда находится в онлайн, ибо словарь данных должен быть всегда доступен ORACLE. Обычное состояние табличного пространства - онлайн, так что данные, содержащиеся в нем, доступны пользователям базы данных. Однако администратору может понадобиться перевод табличного пространства в офлайн по одной из следующих причин:

- чтобы сделать часть базы данных недоступной, сохраняя в то же время нормальный доступ к остальной части
- чтобы выполнить резервное копирование офлайнового табличного пространства (хотя такое копирование можно осуществлять и в онлайн, одновременно с использованием табличного пространства)
- чтобы сделать приложение вместе с его группой таблиц временно недоступным на время обновления или сопровождения этого приложения

Когда табличное пространство переводится в офлайн, ORACLE не позволяет последующим предложениям SQL обращаться к объектам этого табличного пространства.

Табличное пространство не может быть переведено в офлайн, если оно содержит активные сегменты отката. Табличное пространство может быть переведено в офлайн лишь в том случае, если все сегменты отката, содержащиеся в нем, не используются.

Табличное пространство в базе данных ORACLE состоит из одного или нескольких физических **ФАЙЛОВ ДАННЫХ**. Файлы данных, ассоциированные с табличным пространством, хранят все данные этого табличного пространства. Любой файл данных может ассоциироваться только с одним табличным пространством и только с одной базой данных. При создании файла данных для табличного пространства ORACLE распределяет ему указанное количество дисковой памяти. Когда файл данных создается, операционная система несет ответственность за очистку старой информации и за установку должных режимов доступа к файлу, прежде чем он будет распределен ORACLE. Если файл велик, этот процесс может потребовать значительного времени. Поскольку первым табличным пространством в любой базе данных всегда является SYSTEM, первые файлы данных любой базы данных автоматически распределяются табличному пространству SYSTEM во время создания базы данных.

После первоначального создания файла данных соответствующее дисковое пространство еще не содержит никаких данных; однако это

пространство ЗАРЕЗЕРВИРОВАНО за будущими сегментами ассоциированного табличного пространства - оно не может содержать каких-либо иных данных. Когда сегмент (например, сегмент данных таблицы) будет создан и начнет увеличиваться в размерах, ORACLE использует свободное место в соответствующих файлах данных, чтобы распределять экстенды для этого сегмента. Данные в сегментах объектов (сегментах данных, сегментах индексов, сегментах отката и т.д.) в табличном пространстве физически хранятся в одном или нескольких файлах данных, составляющих это табличное пространство.

Табличные пространства в любой момент можно переводить в ОФЛАЙН (т.е. делать недоступными) или в ОНЛАЙН (делать доступными). Поэтому все файлы данных, составляющие табличное пространство, переводятся в офлайн или онлайн одновременно, всей группой. Индивидуальные файлы данных также могут быть переведены в офлайн; однако это обычно делается лишь при некоторых процедурах восстановления базы данных.

СХЕМА(SCHEMA) - это коллекция объектов. ОБЪЕКТЫ СХЕМЫ - это логические структуры, непосредственно относящиеся к данным базы данных. Объекты схемы включают такие структуры как: таблицы, представления, последовательности, хранимые процедуры, синонимы, индексы, кластеры и связи баз данных. (Не существует взаимосвязи между табличным пространством и схемой; объекты одной и той же схемы могут находиться в разных табличных пространствах, и одно и то же табличное пространство может содержать объекты из разных схем.)

ТАБЛИЦА(TABLE) - это основная единица хранения данных в базе данных ORACLE. Таблицы базы данных хранят все данные, доступные пользователям.

Данные таблицы хранятся в виде СТРОК и СТОЛБЦОВ. Каждая таблица определяется с ИМЕНЕМ ТАБЛИЦЫ и набором столбцов. Каждому столбцу дается ИМЯ СТОЛБЦА, ТИП ДАННЫХ (такой как CHAR, DATE или NUMBER), а также ШИРИНА (которая может быть предопределена типом данных, как в случае DATE) или МАСШТАБ и ТОЧНОСТЬ (только для типа данных NUMBER). После того, как таблица создана, в нее могут быть вставлены строки действительных данных. После этого строки таблицы можно опрашивать, удалять или обновлять. Чтобы учредить организационные правила для данных таблицы, для таблицы можно также определить ограничения целостности и триггеры.

ПРЕДСТАВЛЕНИЯ(VIEW) - это настроенное представление данных из одной или нескольких таблиц. представлений можно рассматривать как "хранимый запрос". Представления в действительности не содержат данных;

вместо этого они доставляют данные из тех таблиц, на которых они основаны (так называемых **БАЗОВЫХ ТАБЛИЦ** представлений). Базовые таблицы, в свою очередь, могут быть как таблицами, так и представлениями.

Как и таблицы, представления можно опрашивать, обновлять и осуществлять в них вставки и удаления, с некоторыми ограничениями. Все операции, осуществляемые над представлениями, в действительности затрагивают базовые таблицы этого представления.

ПОСЛЕДОВАТЕЛЬНОСТЬ(SEQUENCE) - генерирует уникальные порядковые номера, которые могут использоваться как значения числовых столбцов таблиц базы данных. Последовательности упрощают прикладное программирование, автоматически генерируя уникальные числовые значения для строк одной или нескольких таблиц. Номера, генерируемые последовательностью, независимы от таблиц, так что одну и ту же последовательность можно использовать для нескольких таблиц. После ее создания, к последовательности могут обращаться различные пользователи, чтобы получать действительные порядковые номера.

Термином "программная единица" обозначаются хранимые процедуры, функции и пакеты.

ПРОЦЕДУРА(PROCEDURE) или **ФУНКЦИЯ(FUNCTION)** - это совокупность предложений SQL и PL/SQL, сгруппированных вместе как выполняемая единица, исполняющая специфическую задачу.

Процедуры и функции сочетают легкость и гибкость SQL с процедурными возможностями языка структурного программирования. С помощью PL/SQL такие процедуры и функции можно определять и сохранять в базе данных для продолжительного использования.

Процедуры и функции похожи друг на друга, с той разницей, что функция всегда возвращает вызывающей программе единственное значение, тогда как процедура в общем случае не возвращает значения, однако существуют специфические методики для возвращения значения процедуры.

ПАКЕТЫ(PACKAGE) дают метод инкапсулирования и хранения взаимосвязанных процедур, функций и других конструкторов пакета как единицы в базе данных. Предоставляя администратору базы данных или разработчику приложений организационные преимущества, пакеты в то же время расширяют функциональные возможности, и увеличивают производительность базы данных.

СИНОНИМ(SYNONYM) - это алиас (дополнительное имя) для таблицы, представлений, последовательности или программной единицы. Синоним не есть объект, но он является прямой ссылкой на объект. Синонимы используются для:

- 1) маскировки действительного имени и владельца объекта;
- 2) обеспечения общего доступа к объекту
- 3) достижения прозрачности местоположения для таблиц, представлений или программных единиц удаленной базы данных
- 4) упрощения кодирования предложений SQL для пользователей базы данных

Синоним может быть общим или личным. Индивидуальный пользователь может создать **ЛИЧНЫЙ СИНОНИМ**, который доступен только этому пользователю. Администраторы баз данных чаще всего создают **ОБЩИЕ СИНОНИМЫ**, благодаря которым объекты базовых схем становятся доступными для общего пользования всем пользователям базы данных.

Индексы, кластеры и хешированные кластеры - это необязательные структуры, ассоциированные с таблицами, которые можно создавать для повышения производительности операций извлечения данных. Так же, как индекс в книге позволяет вам быстрее отыскивать нужную информацию, индекс ORACLE предоставляет быстрый путь доступа к данным таблицы. При обработке запроса ORACLE может использовать некоторые или все имеющиеся индексы для эффективного отыскания запрашиваемых строк. Индексы полезны, когда приложения часто опрашивают интервалы строк таблицы либо отдельные строки.

Индексы создаются по одному или нескольким столбцам таблицы. Однажды созданный, индекс автоматически поддерживается и используется ORACLE. Изменения в данных таблицы автоматически отражаются во всех соответствующих индексах при полной прозрачности для пользователей.

КЛАСТЕРЫ (CLUSTER) предоставляют необязательный способ хранения данных таблиц. Кластер - это группа из одной или нескольких таблиц, физически хранящихся вместе.

Взаимосвязанные столбцы таблиц в кластере называются **КЛЮЧОМ КЛАСТЕРА**. Ключ кластера индексируется, так что строки кластера могут извлекаться с минимальными затратами на ввод-вывод. Поскольку данные ключа кластера в индексированном (не кэшированном) кластере хранятся в одном экземпляре для всех таблиц кластера, достигается экономия пространства по сравнению с обычными (некластеризованными) таблицами.

Кластеризованные таблицы: Связанные данные хранятся вместе, более эффективно. **Некластеризованные таблицы:** Связанные данные хранятся отдельно, занимая больше места.

Кластеры могут также повысить эффективность извлечения данных, в зависимости от распределения данных и от того, какие операции SQL

наиболее часто выполняются на кластеризованных данных. В частности, кластеризованные таблицы, опрашиваемые через соединения, выигрывают за счет кластеров, потому что строки, общие для объединяемых таблиц, извлекаются за одну операцию ввода-вывода. Как и индексы, кластеры не влияют на проектирование приложений. Является ли таблица частью кластера или нет, остается прозрачным для пользователей и приложений. Данные, хранящиеся в кластеризованной таблице, доступны через те же операции SQL, как если бы они не были кластеризованы.

ХЭШИРОВАННЫЕ КЛАСТЕРЫ похожи на обычные, индексированные, кластеры. Однако в хэшированных кластерах строки записываются не на основе ключа кластера, а на основе значения **ФУНКЦИИ ХЭШИРОВАНИЯ**, применяемой к ключу кластера. Все строки с одинаковым значением такого хэш-ключа хранятся на диске вместе. Хэшированные кластеры выигрывают по сравнению с индексированной таблицей и индексированным кластером, когда таблица часто опрашивается на равенство. Для таких запросов значения указанного ключа кластера хэшируются, и результирующие значения хэш-ключа прямо указывают на участок диска, в котором хранятся соответствующие строки.

СВЯЗЬ БАЗ ДАННЫХ - это именованный объект, который описывает "путь" от одной базы данных к другой. Связи баз данных неявно используются при обращении к **ГЛОБАЛЬНОМУ ИМЕНИ ОБЪЕКТА** в распределенной базе данных.

Оборудование и материалы

Для выполнения практической работы необходим персональный компьютер с установленной СУБД Oracle10g/11g ExpressEdition и пакетом Microsoft Office. Инсталляция пакета представлена в приложении.

Указания по технике безопасности

Необходимо выполнять требования по технике безопасности, предусмотренные для компьютерного класса

1. Не включать компьютер без разрешения администратора класса и преподавателя.
2. Запрещается самостоятельно вскрывать корпус ПК.
3. Не касаться одновременно экрана монитора и клавиатуры (возможен разряд повышенного электростатического потенциала).
4. Не загромождать верхние панели устройств бумагами и посторонними устройствами.
5. Не прикасаться одновременно к металлическим частям ПК и устройствам, имеющим естественное заземление.
6. Запрещается производить отключение питания во время выполнения

активной задачи.

7. Не допускать попадание влаги на поверхность системного блока, монитора, клавиатуру, принтеры и другие устройства.

Порядок выполнения работы

1. Переведите табличное пространство БД с состояние онлайн и офлайн.
2. Исследуйте процесс определения системного табличного пространства, созданного в процессе создания БД.
3. Сделайте часть базы данных недоступной.
4. Выполните резервное копирование офлайнного табличного пространства.
5. Рассмотрите физические файлы данных., они хранят все данные табличного пространства.

Содержание отчета и его форма

Отчет должен содержать

- название практической работы;
- цель работы;
- порядок выполнения работы;
- результаты выполнения;
- вывод о проделанной работе.

Контрольные вопросы и защита работы

Контрольные вопросы

1. Дайте характеристику табличным пространствам.
2. Табличное пространство SYSTEM?
3. Файлов данных?
4. Что представляет собой Объекты?
5. Как называются логические элементы БД?
6. Онлайнновые и офлайнновые табличные пространства

Защита работы

Защита работы заключается в:

- выполнении заданий;
- ответах на контрольные вопросы;
- предоставлении отчета.

ПРАКТИЧЕСКАЯ РАБОТА.

РАСПРЕДЕЛЕНИЕ ПРОСТРАНСТВА БАЗЫ ДАННЫХ

Цель и содержание

Цель работы: распределение пространства базы данных для всех ее данных.

Теоретическое обоснование

ORACLE распределяет пространство базы данных для всех ее данных. Единицами логического распределения являются блоки данных, экстененты и сегменты.

На самом низком уровне рассмотрения, данные базы данных ORACLE хранятся в **БЛОКАХ ДАННЫХ** (называемых также логическими блоками, блоками ORACLE или страницами). Один блок данных соответствует фиксированному числу байт физического пространства базы данных на диске. Размер блока данных специфически устанавливается для каждой базы данных ORACLE при ее создании. Этот размер кратен размеру блока операционной системы, но не превышает определенный максимум. Важно помнить, что база данных, на ее самом низком уровне, использует и распределяет свободное пространство в базе данных блоками данных ORACLE. По контрасту с этим, все данные на физическом уровне, т.е. на уровне операционной системы, распределяются в байтах. Каждая операционная система имеет то, что называется **РАЗМЕРОМ БЛОКА**, который определяется как специфическое число байт на диске.

ORACLE управляет пространством в файлах данных базы данных в единицах, называемых **БЛОКАМИ ДАННЫХ**. Блок данных - это наименьшая единица ввода-вывода, используемая базой данных. Блок данных соответствует физическому блоку на диске с размером, совпадающим с размером блока данных ORACLE. Этот размер блока может отличаться от стандартного размера блока ввода-вывода операционной системы, в которой выполняется ORACLE.

Формат блока данных ORACLE один и тот же, независимо от того, содержит ли блок данные таблицы, индекса или кластера.

Следующий уровень логического пространства базы данных называется **ЭКСТЕНТОМ(EXTENTS)**. Экстент - это специфическое число смежных блоков данных, распределяемых для хранения специфического типа информации.

Экстент - это логическая единица распределения пространства базы данных, состоящая из определенного числа непрерывных блоков данных. Каждый тип сегмента состоит из одного или нескольких экстенентов. Когда

существующее пространство в сегменте полностью использовано, ORACLE распределяет для сегмента новый экстен-

том, называется СЕГМЕНТОМ (SEGMENTS). Сегмент - это совокупность экстен-
тентов, распределенных для специфического типа структуры данных, и нахо-
дящихся в одном и том же табличном пространстве. Например, данные каждой
таблицы хранятся в ее собственном СЕГМЕНТЕ ДАННЫХ, а данные каждого
индекса хранятся в его собственном СЕГМЕНТЕ ИНДЕКСА.

ORACLE распределяет пространство для сегментов экстен-тами. Поэтому,
когда существующие экстен-ты сегмента заполнены, ORACLE распределяет
очередной экстен-т для этого сегмента. Поскольку экстен-ты распределяются при
необходимости, экстен-ты сегмента не обязательно смежные на диске, и могут
быть распределены между различными файлами. Каждый экстен-т, однако, не
может находиться в нескольких файлах.

Сегмент - это набор экстен-тов, содержащих все данные для конкретного
типа структуры логического пространства внутри табличного пространства.
Например, для каждой таблицы ORACLE распределяет один или несколько
экстен-тов, чтобы сформировать сегмент данных этой таблицы; для каждого
индекса ORACLE распределяет один или несколько экстен-тов, чтобы
сформировать сегмент индекса для этого индекса.

База данных ORACLE может содержать четыре различных типа
сегментов:

- сегмент данных
- сегмент индекса
- сегмент отката
- временный сегмент

Оборудование и материалы

Для выполнения практической работы необходим персональный компью-
тер с установленной СУБД Oracle10g/11g ExpressEdition и пакетом Microsoft
Office. Инсталляция пакета представлена в приложении.

Указания по технике безопасности

Необходимо выполнять требования по технике безопасности, предусмот-
ренные для компьютерного класса

1. Не включать компьютер без разрешения администратора класса и
преподавателя.
2. Запрещается самостоятельно вскрывать корпус ПК.
3. Не касаться одновременно экрана монитора и клавиатуры (возможен

разряд повышенного электростатического потенциала).

4. Не загромождать верхние панели устройств бумагами и посторонними устройствами.

5. Не прикасаться одновременно к металлическим частям ПК и устройствам, имеющим естественное заземление.

6. Запрещается производить отключение питания во время выполнения активной задачи.

7. Не допускать попадание влаги на поверхность системного блока, монитора, клавиатуру, принтеры и другие устройства.

Порядок выполнения работы

1. Расщепляйте все файлы по всем дискам с шириной полосы расщепления, равной 1 Мб.

2. Зеркалируйте данные для обеспечения высокой доступности.

3. Распределяйте данные по секциям, а не по дискам.

4. Размещайте данные, к которым возможен частый доступ, на внешних половинах пространства дисков.

Содержание отчета и его форма

Отчет должен содержать

- название практической работы;
- цель работы;
- порядок выполнения работы;
- результаты выполнения;
- вывод о проделанной работе.

Контрольные вопросы и защита работы

Контрольные вопросы

7. В чем назначение отношений между сегментами, экстендами и блоками данных?

8. Каковы Блоки данных?

9. Каков Формат блока данных ORACLE ?

10. Что такое Экстенды данных?

11. Что такое Сегменты данных?

12. Каковы типы сегментов?

Защита работы

Защита работы заключается в:

- выполнении заданий;
- ответах на контрольные вопросы;
- предоставлении отчета.

ПРАКТИЧЕСКАЯ РАБОТА.

ОДНОПРОЦЕССОРНАЯ (ТАКЖЕ НАЗЫВАЕМАЯ ОДНОПОЛЬЗОВАТЕЛЬСКОЙ) СИСТЕМА ORACLE

Цель и содержание

Цель работы: работа с однопользовательской системой ORACLE

Теоретическое обоснование

Механизмы ORACLE работают через использование структур памяти и процессов. Все структуры памяти располагаются в основной памяти (иногда называемой виртуальной памятью или памятью произвольного доступа) компьютеров, составляющих систему базы данных.

ПРОЦЕССЫ(PROCESS) - это задания или задачи, работающие в памяти этих компьютеров. ПРОЦЕСС - это "канал управления", или механизм в операционной системе, способный выполнять последовательность шагов. Некоторые операционные системы используют термины "задание" или "задача". Процесс обычно имеет свою собственную личную область памяти, в которой он выполняется. (V\$PROCESS - информация об активных и V\$BGPROCESS – информация о фоновых процессах)

Структура процесса такой системы, как ORACLE, существенна, потому что она определяет, как осуществляется параллельная деятельность и как она управляется. Например, двумя целями структуры процесса могут быть: 1) симуляция личных окружений для нескольких одновременно работающих процессов, так, как будто каждый процесс имеет свое собственное личное окружение 2) обеспечение разделения между процессами ресурсов компьютера, необходимых каждому процессу, но не на долгое время

Однопроцессная (также называемая однопользовательской) система ORACLE - это система базы данных, в которой весь код ORACLE выполняется одним процессом. Не используются разные процессы, чтобы разграничить выполнение компонент ORACLE и прикладной программы клиента. Вместо этого весь код ORACLE и единственное приложение базы данных выполняются как единственный процесс. Единственный процесс исполняет весь код, ассоциированный как с приложением базы данных, так и с ORACLE.

Многопроцессный ORACLE (называемый также многопользовательским) использует несколько процессов для исполнения различных частей ORACLE, а также отдельный процесс для каждого присоединенного пользователя. Каждый процесс в многопроцессном ORACLE выполняет специфическую задачу. Благодаря разделению работы ORACLE и приложений базы данных на

несколько процессов, несколько пользователей и приложений могут одновременно присоединяться к единственной инстанции базы данных, в то время как система поддерживает отличную производительность. Большинство систем баз данных - многопользовательские, ибо одним из основных преимуществ СУБД является управление данными, с которыми много пользователей работают одновременно. Каждый присоединенный пользователь имеет отдельный пользовательский процесс, а для выполнения ORACLE используются несколько фоновых процессов. Много процессной системе все процессы можно разделить на две группы: пользовательские процессы и процессы ORACLE.

ORACLE создает и использует свои структуры памяти для выполнения некоторых задач. Например, память используется для размещения исполняемого программного кода и данных, разделяемых между пользователями. С ORACLE ассоциируются несколько базовых структур памяти: глобальная область системы (которая включает буфера базы данных и журнала повторения, а также разделяемый пул) и глобальные области программ. Следующие секции подробно объясняют каждую из этих видов областей.

ГЛОБАЛЬНАЯ ОБЛАСТЬ СИСТЕМЫ (SGA) - это область памяти разделяемой между процессами. Область SGA и фоновые процессы ORACLE составляют инстанцию ORACLE. SGA распределяется при запуске инстанции и освобождается при закрытии инстанции. Для оптимальной производительности SGA должна быть максимально большой (пока позволяет реальная память), чтобы держать как можно больше данных в памяти и минимизировать дисковые операции. Информация, хранящаяся в SGA, подразделяется на несколько типов структур памяти, включая: буфера базы данных, буфера журнала повторения и разделяемый пул. Эти области имеют фиксированные размеры и создаются при запуске инстанции. Буферный кэш базы данных **БУФЕРЫ БАЗЫ ДАННЫХ** в SGA хранят наиболее недавно использовавшиеся блоки данных из базы данных; все множество буферов базы данных в инстанции составляет **БУФЕРНЫЙ КЭШ БАЗЫ ДАННЫХ**.

Оборудование и материалы

Для выполнения практической работы необходим персональный компьютер с установленной СУБД Oracle10g/11g ExpressEdition и пакетом Microsoft Office. Инсталляция пакета представлена в приложении.

Указания по технике безопасности

Необходимо выполнять требования по технике безопасности, предусмотренные для компьютерного класса

1. Не включать компьютер без разрешения администратора класса и преподавателя.
2. Запрещается самостоятельно вскрывать корпус ПК.
3. Не касаться одновременно экрана монитора и клавиатуры (возможен разряд повышенного электростатического потенциала).
4. Не загромождать верхние панели устройств бумагами и посторонними устройствами.
5. Не прикасаться одновременно к металлическим частям ПК и устройствам, имеющим естественное заземление.
6. Запрещается производить отключение питания во время выполнения активной задачи.
7. Не допускать попадание влаги на поверхность системного блока, монитора, клавиатуру, принтеры и другие устройства.

Порядок выполнения работы

1. Охарактеризуйте однопроцессорный режим работы Oracle. Объясните возможное поведение системы при работе с вашей БД.
2. Обоснуйте, что весь код Oracle и единственное приложение базы данных выполняются как единственный процесс.

Содержание отчета и его форма

Отчет должен содержать

- название практической работы;
- цель работы;
- порядок выполнения работы;
- результаты выполнения;
- вывод о проделанной работе.

Контрольные вопросы и защита работы

Контрольные вопросы

13. Что представляет собой Структуры памяти?
14. Каковы задачи, работающие в памяти?
15. Какова структура процесса такой системы, как ORACLE?
16. Каковы цели структуры процесса?
17. Что такое однопроцессная (также называемая однопользовательской) система ORACLE?

Защита работы

Защита работы заключается в:

- выполнении заданий;
- ответах на контрольные вопросы;
- предоставлении отчета.

ПРАКТИЧЕСКАЯ РАБОТА.

МНОГОПРОЦЕССНЫЙ ORACLE (НАЗЫВАЕМЫЙ ТАКЖЕ МНОГО-ПОЛЬЗОВАТЕЛЬСКИМ)

Цель и содержание

Цель работы: работа с многопроцессорной системой ORACLE

Теоретическое обоснование

Механизмы ORACLE работают через использование структур памяти и процессов. Все структуры памяти располагаются в основной памяти (иногда называемой виртуальной памятью или памятью произвольного доступа) компьютеров, составляющих систему базы данных.

ПРОЦЕССЫ(PROCESS) - это задания или задачи, работающие в памяти этих компьютеров. ПРОЦЕСС - это "канал управления", или механизм в операционной системе, способный выполнять последовательность шагов. Некоторые операционные системы используют термины "задание" или "задача". Процесс обычно имеет свою собственную личную область памяти, в которой он выполняется. (V\$PROCESS - информация об активных и V\$BGPROCESS – информация о фоновых процессах)

Структура процесса такой системы, как ORACLE, существенна, потому что она определяет, как осуществляется параллельная деятельность и как она управляется. Например, двумя целями структуры процесса могут быть: 1) симуляция личных окружений для нескольких одновременно работающих процессов, так, как будто каждый процесс имеет свое собственное личное окружение 2) обеспечение разделения между процессами ресурсов компьютера, необходимых каждому процессу, но не на долгое время

Однопроцессная (также называемая однопользовательской) система ORACLE - это система базы данных, в которой весь код ORACLE выполняется одним процессом. Не используются разные процессы, чтобы разграничить выполнение компонент ORACLE и прикладной программы клиента. Вместо этого весь код ORACLE и единственное приложение базы данных выполняются как единственный процесс. Единственный процесс исполняет весь код, ассоциированный как с приложением базы данных, так и с ORACLE.

Многопроцессный ORACLE (называемый также многопользовательским) использует несколько процессов для исполнения различных частей ORACLE, а также отдельный процесс для каждого присоединенного пользователя. Каждый процесс в многопроцессном ORACLE выполняет специфическую задачу. Благодаря разделению работы ORACLE и приложений базы данных на

несколько процессов, несколько пользователей и приложений могут одновременно присоединяться к единственной инстанции базы данных, в то время как система поддерживает отличную производительность. Большинство систем баз данных - многопользовательские, ибо одним из основных преимуществ СУБД является управление данными, с которыми много пользователей работают одновременно. Каждый присоединенный пользователь имеет отдельный пользовательский процесс, а для выполнения ORACLE используются несколько фоновых процессов. Много процессной системе все процессы можно разделить на две группы: пользовательские процессы и процессы ORACLE.

ORACLE создает и использует свои структуры памяти для выполнения некоторых задач. Например, память используется для размещения исполняемого программного кода и данных, разделяемых между пользователями. С ORACLE ассоциируются несколько базовых структур памяти: глобальная область системы (которая включает буфера базы данных и журнала повторения, а также разделяемый пул) и глобальные области программ. Следующие секции подробно объясняют каждую из этих видов областей.

ГЛОБАЛЬНАЯ ОБЛАСТЬ СИСТЕМЫ (SGA) - это область памяти разделяемой между процессами. Область SGA и фоновые процессы ORACLE составляют инстанцию ORACLE. SGA распределяется при запуске инстанции и освобождается при закрытии инстанции. Для оптимальной производительности SGA должна быть максимально большой (пока позволяет реальная память), чтобы держать как можно больше данных в памяти и минимизировать дисковые операции. Информация, хранящаяся в SGA, подразделяется на несколько типов структур памяти, включая: буфера базы данных, буфера журнала повторения и разделяемый пул. Эти области имеют фиксированные размеры и создаются при запуске инстанции. Буферный кэш базы данных **БУФЕРЫ БАЗЫ ДАННЫХ** в SGA хранят наиболее недавно использовавшиеся блоки данных из базы данных; все множество буферов базы данных в инстанции составляет **БУФЕРНЫЙ КЭШ БАЗЫ ДАННЫХ**.

Оборудование и материалы

Для выполнения практической работы необходим персональный компьютер с установленной СУБД Oracle10g/11g ExpressEdition и пакетом Microsoft Office. Инсталляция пакета представлена в приложении.

Указания по технике безопасности

Необходимо выполнять требования по технике безопасности, предусмотренные для компьютерного класса

8. Не включать компьютер без разрешения администратора класса и преподавателя.

9. Запрещается самостоятельно вскрывать корпус ПК.

10. Не касаться одновременно экрана монитора и клавиатуры (возможен разряд повышенного электростатического потенциала).

11. Не загромождать верхние панели устройств бумагами и посторонними устройствами.

12. Не прикасаться одновременно к металлическим частям ПК и устройствам, имеющим естественное заземление.

13. Запрещается производить отключение питания во время выполнения активной задачи.

14. Не допускать попадание влаги на поверхность системного блока, монитора, клавиатуру, принтеры и другие устройства.

Порядок выполнения работы

1. Охарактеризуйте многопроцессный режим работы Oracle. Объясните возможное поведение системы при работе с вашей БД.

2. Обоснуйте, какое множество процессов выполняется в системе при работе с вашей БД.

Содержание отчета и его форма

Отчет должен содержать

- название практической работы;
- цель работы;
- порядок выполнения работы;
- результаты выполнения;
- вывод о проделанной работе.

Контрольные вопросы и защита работы

Контрольные вопросы

1. Что такое многопроцессная ORACLE ?
2. Что представляет собой глобальная область системы (SGA)
3. Введение в словарь данных.
4. Структура словаря данных.
5. Каковы наборы представлений?

Защита работы

Защита работы заключается в:

- выполнении заданий;
- ответах на контрольные вопросы;
- предоставлении отчета.

ПРАКТИЧЕСКАЯ РАБОТА.

ПРИМЕР РАБОТЫ ORACLE

Цель и содержание

Цель работы: исследование работы конфигурации ORACLE

Теоретическое обоснование

Следующее описание иллюстрирует работу конфигурации ORACLE, при которой пользователь и ассоциированный с ним процесс сервера находятся на разных машинах, соединенных через сеть.

1. **ЭКЗЕМПЛЯР** - работает на том компьютере, на котором выполняется ORACLE (часто называемом **ХОСТОМ** или **СЕРВЕРОМ БАЗЫ ДАННЫХ**, информацию о текущем состоянии сервера БД, экземпляре, можно получить по V\$INSTANCE)

ЭКЗЕМПЛЯР (INSTANCE она же инстанция) СУБД Oracle – это набор серверных процессов, объединенных единой глобальной системной областью и связанных с общим набором совместно используемых файлов данных. К примеру если на сервере работает две независимые базы данных, и каждая обладает своей глобальной системной областью и независимым набором сервисных процессов то у нас будет два экземпляра базы данных Oracle; каждому из экземпляров присваивается системный идентификатор SID (system identifier)

2. Компьютер, используемый для приложений (**ЛОКАЛЬНАЯ МАШИНА** или **РАБОЧАЯ СТАНЦИЯ КЛИЕНТА**) выполняет приложение в рамках пользовательского процесса. Приложение-клиент пытается установить соединение с сервером, используя подходящий драйвер SQL*Net.

3. Сервер также выполняет соответствующий драйвер SQL*Net. Сервер обнаруживает запрос на соединение от приложения и создает (выделенный) процесс сервера от имени пользовательского процесса.

4. Пользователь выполняет предложение SQL и подтверждает свою транзакцию. Например, пользователь изменяет данные в строке таблицы.

5. Процесс сервера принимает это предложение и проверяет разделяемый пул, чтобы отыскать в нем разделяемую область SQL, содержащую идентичное предложение SQL. Если такая разделяемая область SQL найдена, процесс сервера проверяет пользовательские привилегии доступа к запрашиваемым данным и использует существующую разделяемую область SQL для обработки предложения; если нет, то для предложения распределяется новая разделяемая область SQL, в которой осуществляется разбор и исполнение предложения.

6. Процесс сервера извлекает все необходимые значения данных из файла данных (таблицы) или получает их из SGA.

7. Процесс сервера модифицирует данные в SGA. Процесс DBWR записывает модифицированные данные на диск для постоянного хранения, если сочтет это необходимым. Поскольку транзакция подтверждена, процесс LGWR немедленно регистрирует транзакцию в онлайн-файле журнала повторения.

8. Если транзакция успешна, процесс сервера посылает сообщение через сеть приложению. В противном случае передается соответствующее сообщение об ошибке.

9. Во время всей описанной процедуры работают другие фоновые процессы, наблюдая за условиями, которые могут потребовать вмешательства. Кроме того, сервер базы данных управляет транзакциями других пользователей и предотвращает соперничество между транзакциями, запрашивающими одни и те же данные.

Эти шаги описывают лишь самый основной уровень операций, которые осуществляет ORACLE.

Оборудование и материалы

Для выполнения практической работы необходим персональный компьютер с установленной СУБД Oracle10g/11g ExpressEdition и пакетом Microsoft Office. Инсталляция пакета представлена в приложении.

Указания по технике безопасности

Необходимо выполнять требования по технике безопасности, предусмотренные для компьютерного класса

1. Не включать компьютер без разрешения администратора класса и преподавателя.
2. Запрещается самостоятельно вскрывать корпус ПК.
3. Не касаться одновременно экрана монитора и клавиатуры (возможен разряд повышенного электростатического потенциала).
4. Не загромождать верхние панели устройств бумагами и посторонними устройствами.
5. Не прикасаться одновременно к металлическим частям ПК и устройствам, имеющим естественное заземление.
6. Запрещается производить отключение питания во время выполнения активной задачи.
7. Не допускать попадание влаги на поверхность системного блока, монитора, клавиатуру, принтеры и другие устройства.

Порядок выполнения работы

Примените выше описанные шаги как самый основной уровень операций, которые осуществляет ORACLE.

Содержание отчета и его форма

Отчет должен содержать

- название практической работы;
- цель работы;
- порядок выполнения работы;
- результаты выполнения;
- вывод о проделанной работе.

Контрольные вопросы и защита работы

Контрольные вопросы

1. Что представляет собой Структуры памяти?
2. Каковы задачи, работающие в памяти?
3. Какова структура процесса такой системы, как ORACLE?
4. Каковы цели структуры процесса?
5. Что такое однопользовательская (также называемая однопользовательской) система ORACLE?
6. Что такое многопользовательская ORACLE ?
7. Что представляет собой глобальная область системы (SGA)
8. Введение в словарь данных.
9. Структура словаря данных.
10. Каковы наборы представлений?

Защита работы

Защита работы заключается в:

- выполнении заданий;
- ответах на контрольные вопросы;
- предоставлении отчета.

ПРАКТИЧЕСКАЯ РАБОТА. ЖУРНАЛ ПОВТОРЕНИЙ

Цель и содержание

Цель работы: исследование работы файлов журнала повторений

Теоретическое обоснование

Каждая база данных ORACLE имеет набор из двух или более ФАЙЛОВ ЖУРНАЛА ПОВТОРЕНИЯ РАБОТЫ. Комплект файлов журнала повторения работы для одной базы данных совместно называется ЖУРНАЛОМ ПОВТОРЕНИЯ (redo log).

В базе данных Oracle имеется файлы двух типов:

- Файлы данных, сгруппированные в табличные пространства.
- Файлы данных, относящиеся к классу журнала повтора.

В базе данных как минимум должно быть не менее двух оперативных журналов повтора.

Журнал повтора часто называют журналом транзакций. Основная функция журнала повторения - регистрация всех изменений, осуществляемых в данных. Все изменения, выполняемые в базе данных, записываются в журнал повторения. Если в результате сбоя модифицированные данные не удастся постоянно записать в файлы данных, эти изменения можно получить из журнала повторения, так что результаты работ никогда не теряется, т.е. можно произвести не только восстановление старой копии БД, но и повторить все транзакции, выполненные к моменту сбоя.

Файлы журнала повторения критичны в вопросе защиты базы данных от сбоев. Чтобы защититься от таких сбоев, которые затрагивают сам журнал повторения, ORACLE допускает ЗЕРКАЛЬНЫЙ ЖУРНАЛ ПОВТОРЕНИЯ, так что две или более копий журнала повторения можно поддерживать одновременно на разных дисках.

Информация в файле журнала повторения используется только для восстановления базы данных после сбоя системы или носителя, в результате которого данные базы данных не могут быть записаны в файлы данных.

Процесс применения журнала повторения в процессе операции восстановления базы данных называется ПРОКРУТКОЙ ВПЕРЕД. База данных может работать в двух режимах. В режиме ARCHIVELOG и в режиме NOARCHIVELOG, соответственно либо, создается архивная копия всех журналов транзакций, либо содержимое транзакций не сохраняется. Для определения текущего режима работы экземпляра выдают команду archive log list, из под Server* Manager. Многие аспекты процесса архивирования указаны

параметрами файла INIT.ORA Восстановление после сбоя экземпляра, с использованием только журнала обновления, называется - оперативным восстановлением.

Чтобы добавить новые члены в существующую группу журнала, используйте либо диалог Add Online Redo Log Member в SQL*DBA, либо команду SQL ALTER DATABASE с параметром ADD LOGFILE MEMBER. Для удаления группы онлайн-журнала повторения вы должны иметь системную привилегию ALTER DATABASE.

Число онлайн-файлов журнала базы данных ограничивается тремя параметрами:

- Параметр MAXLOGFILES, который использовался в предложении CREATE DATABASE при создании базы данных, определяет максимальное число групп онлайн-журнала на базу данных; номера групп могут изменяться от 1 до MAXLOGFILES. Единственный способ перекрыть эту верхнюю границу - заново создать базу данных или ее управляющий файл; поэтому важно рассмотреть это значение перед созданием базы данных. Если в предложении CREATE DATABASE не указан параметр MAXLOGFILES, то ORACLE использует умалчиваемое значение, зависящее от операционной системы.

- Параметр LOG_FILES в файле параметров может временно уменьшить максимальное число групп файлов журнала на время исполнения текущего экземпляра. Значение этого параметра не может превышать значение MAXLOGFILES. Если в файле параметров не указан параметр LOG_FILES, то ORACLE использует умалчиваемое значение, зависящее от операционной системы.

- Параметр MAXLOGMEMBERS, который использовался в предложении CREATE DATABASE при создании базы данных, определяет максимальное число членов в группе. Как и для MAXLOGFILES, единственный способ перекрыть эту верхнюю границу - заново создать базу данных или ее управляющий файл; поэтому важно рассмотреть это значение перед созданием базы данных. Если в предложении CREATE DATABASE не указан параметр MAXLOGMEMBERS, то ORACLE использует умалчиваемое значение, зависящее от операционной системы.

Журнал повтора можно разделить на две категории: оперативные и архивные. Оперативные журналы повтора – это два и более журналов, записываемые во время работы базы данных; При работе в режиме ARCHIVELOG перед началом перезаписи Oracle копирует содержимое очередного оперативного журнала повтора в соответствующее место на диске, где хранятся архивные

журналы повтора. В момент переключения оперативных журналов повтора, каждому создаваемому журналу присваивается определенный порядковый номер, в соответствие с коим происходит процесс переключения журналов повтора.

К примеру: Уже после второго переключения журнала повтора копируется журнал повтора из первой группы, после чего созданная копия будет называться архивным журналом повтора.

Можно создать группы онлайн-журнала и как часть создания базы данных, и позднее. Если возможно, спланируйте журнал и создайте все необходимые группы файлов журнала во время создания базы данных. Для создания новой группы онлайн-журнала используйте команду SQL: ALTER DATABASE с параметром ADD LOGFILE. Более подробно см Резервное копирование и восстановление.

Оборудование и материалы

Для выполнения практической работы необходим персональный компьютер с установленной СУБД Oracle10g/11g ExpressEdition и пакетом Microsoft Office. Инсталляция пакета представлена в приложении.

Указания по технике безопасности

Необходимо выполнять требования по технике безопасности, предусмотренные для компьютерного класса

1. Не включать компьютер без разрешения администратора класса и преподавателя.
2. Запрещается самостоятельно вскрывать корпус ПК.
3. Не касаться одновременно экрана монитора и клавиатуры (возможен разряд повышенного электростатического потенциала).
4. Не загромождать верхние панели устройств бумагами и посторонними устройствами.
5. Не прикасаться одновременно к металлическим частям ПК и устройствам, имеющим естественное заземление.
6. Запрещается производить отключение питания во время выполнения активной задачи.
7. Не допускать попадание влаги на поверхность системного блока, монитора, клавиатуру, принтеры и другие устройства.

Порядок выполнения работы

1. Добавьте новые члены в существующую группу журнала.
2. Используйте диалог Add Online Redo Log Member в SQL*DBA.
3. Используйте команду SQL ALTER DATABASE с параметром ADD LOGFILE MEMBER.

4. Для удаления группы онлайнового журнала повторения примените системную привилегию ALTER DATABASE.

5. Параметр MAXLOGFILES, который использовался в предложении CREATE DATABASE при создании базы данных, определяет максимальное число групп онлайнового журнала на базу данных; номера групп могут изменяться от 1 до MAXLOGFILES. Примените на практике.

6. Если в предложении CREATE DATABASE не указан параметр MAXLOGFILES, то ORACLE использует умалчиваемое значение, зависящее от операционной системы. Покажите на примере.

7. Параметр LOG_FILES в файле параметров может временно уменьшить максимальное число групп файлов журнала. Примените.

Содержание отчета и его форма

Отчет должен содержать

- название практической работы;
- цель работы;
- порядок выполнения работы;
- результаты выполнения;
- вывод о проделанной работе.

Контрольные вопросы и защита работы

Контрольные вопросы

1. Что такое методы защиты данных?
2. Что такое Безопасность?
3. Что представляет собой Секретность?
4. Дайте определение идентификации пользователя?
5. Управление доступом?
6. В чем суть физической защиты?

Защита работы

Защита работы заключается в:

- выполнении заданий;
- ответах на контрольные вопросы;
- предоставлении отчета.

ПРАКТИЧЕСКАЯ РАБОТА.

ТРАНЗАКЦИЯ (TRANSACTION)

Цель и содержание

Цель работы: изучение работы транзакций

Теоретическое обоснование

Транзакция (модуль фиксации) – логический модуль, который состоит из набора изменений (вставок, обновлений и удалений). Транзакции либо должны быть сохранены в базе данных, либо должен быть выполнен их откат. Фиксируются либо все изменения в транзакции, либо ни одно из них. Для фиксации в базе данных результатов транзакции используется команда COMMIT, для восстановления изменённых данных используется команда ROLLBACK.

Транзакция начинается:

1. Когда пользователь подключается к базе данных и начинает с ней работать.
2. После выполнения оператора COMMIT.
3. После выполнения оператора ROLLBACK.

Откат на уровне оператора – данное понятие означает, что для конкретного оператора либо будут сохранены все внесённые им изменения, либо не будет выполнено ни одно из них.

БЛОКИРОВКИ(LATCH) - это механизмы, используемые для того, чтобы предотвратить деструктивные взаимодействия между пользователями, обращающимися к одному и тому же ресурсу, будь то вся таблица или одна строка в таблице.(V\$LATCH)

В многопользовательской базе данных могут применяться два уровня блокировок:

1. Монопольная блокировка – запрещает разделение ассоциированного ресурса. Первая транзакция, получившая монопольную блокировку ресурса, становится единственной транзакцией, которая может изменять этот ресурс до снятия монопольной блокировки.
2. Разделяемая блокировка – позволяет совместно использовать ассоциированный ресурс, в зависимости от того, какие операции выполняются (например, несколько пользователей могут читать данные одновременно). Несколько транзакций могут получить разделяемые блокировки для одного и того же ресурса.

Захват – ситуация, которая возникает, когда два или более пользователей ожидают данных, заблокированных друг другом.

ORACLE автоматически распознает ситуации захватов, и автоматически разрешает такие ситуации, осуществляя откат одного из предложений, вовлеченных в захват и тем самым освобождая одно множество конфликтующих блокировок строк.

Многоверсионная модель согласованности данных.

1. ORACLE обеспечивает для всех запросов согласованность по чтению на уровне предложения, которая гарантирует, что данные, возвращаемые запросом, согласованы по отношению к моменту начала этого запроса.

2. ORACLE также предоставляет необязательную возможность согласованности по чтению на уровне транзакции, которая гарантирует, что данные, которые видят все запросы внутри транзакции, согласованы по отношению к одной точке времени (моменту начала транзакции). ORACLE обеспечивает согласованность по чтению на уровне транзакции с помощью двух методов:

➤ Транзакции read-only – могут содержать только запросы и не могут содержать никаких предложений DML (языка манипулирования данными). В течение транзакции её доступны данные, которые были подтверждены к моменту её начала.

➤ Монопольные блокировки таблиц и строк. Если повторяемость чтений необходимо обеспечить в транзакциях, содержащих предложения DML, транзакция может явно запросить разделяемые блокировки по таблицам или монопольные блокировки по тем строкам, которые будут считываться неоднократно.

Где:

DDL – Data Definition Language – язык содержащий набор операторов для определения данных.

DML – Data Modifying Language – язык содержащий набор операторов для модификации данных.

По умолчанию Oracle автоматически блокирует строки, на которые воздействуют операторы INSERT, UPDATE или DELETE; причём блокируются только те строки, на которые фактически оказано воздействие, а не вся таблица или весь блок данных.

Назначение блокировок данных (блокировок DML) - защитить данные таблицы, гарантируя их целостность при одновременном доступе к данным нескольких пользователей. Блокировки данных предотвращают деструктивное взаимодействие одновременных конфликтующих операций DML и операций

DDL. Операции DML могут получать блокировки на двух различных уровнях: для конкретных строк и для целых таблиц.

Монопольная блокировка данных запрашивается для индивидуальной строки от имени транзакции, если эта строка модифицируется одним из следующих предложений: INSERT, UPDATE, DELETE или SELECT с фразой FOR UPDATE. Строка всегда блокируется монопольно, так что другие пользователи не могут модифицировать эту строку до тех пор, пока транзакция, удерживающая блокировку, не будет подтверждена или отменена. Блокировки строк всегда запрашиваются ORACLE автоматически как результат приведенных выше предложений.

Транзакция запрашивает блокировку таблицы, когда таблица модифицируется следующими предложениями: INSERT, UPDATE, DELETE, SELECT с фразой FOR UPDATE, либо блокируется предложением LOCK TABLE. Блокировка таблицы может удерживаться в одном из следующих режимов: разделяемая для строк (RS), монопольная для строк (RX), разделяемая для таблицы (S), разделяемая для строк монопольная (SRX) и монопольная (X). Степень ограничения доступа, налагаемая блокировкой, определяет, какие режимы блокировок по этой же таблице могут быть получены другими одновременными транзакциями.

Эта блокировка указывает, что транзакция, удерживающая блокировку по таблице, заблокировала строки в этой таблице и намерена обновить их. Эта блокировка автоматически запрашивается для ТАБЛИЦЫ, модифицируемой следующими предложениями:

SELECT ... FROM таблица ... FOR UPDATE OF ...;

LOCK TABLE таблица IN ROW SHARE MODE;

Блокировка RS - это наименее ограничительный режим блокировки таблицы, предоставляющий наибольшую степень одновременного доступа к таблице.

Блокировка RS, удерживаемая транзакцией, запрещает другим транзакциям получать монопольный доступ по записи к данной таблице, который запрашивается следующим предложением (и только им):

LOCK TABLE таблица IN EXCLUSIVE MODE;

Эта блокировка обычно указывает, что транзакция, удерживающая блокировку по таблице, уже выполнила одно или несколько обновлений в строках таблицы. Эта блокировка автоматически запрашивается для ТАБЛИЦЫ, модифицируемой следующими предложениями:

- INSERT INTO таблица ...;
- UPDATE таблица ...;

- DELETE FROM таблица ...;
- LOCK TABLE таблица IN ROW EXCLUSIVE MODE;

Блокировка RX, удерживаемая транзакцией, запрещает другим транзакциям вручную блокировать таблицу для монопольного чтения или записи; поэтому другие транзакции не могут одновременно использовать следующие предложения:

- LOCK TABLE таблица IN SHARE MODE;
- LOCK TABLE таблица IN SHARE EXCLUSIVE MODE;
- LOCK TABLE таблица IN EXCLUSIVE MODE;

Блокировка S, удерживаемая транзакцией, позволяет другим транзакциям одновременно лишь опрашивать эту таблицу, блокировать выбираемые строки с помощью команды SELECT ... FOR UPDATE, или успешно выполнять предложения LOCK TABLE ... IN SHARE MODE; никакие обновления они не могут осуществлять. Несколько транзакций могут одновременно удерживать блокировки S для одной и той же таблицы.

Разделяемая блокировка таблицы, удерживаемая транзакцией, также запрещает другим транзакциям выполнять следующие предложения:

- LOCK TABLE таблица IN SHARE EXCLUSIVE MODE;
- LOCK TABLE таблица IN EXCLUSIVE MODE;
- LOCK TABLE таблица IN ROW EXCLUSIVE MODE;

Блокировка SRX, удерживаемая транзакцией, позволяет другим транзакциям одновременно лишь опрашивать эту таблицу или блокировать выбираемые строки с помощью команды SELECT ... FOR UPDATE, но не обновлять эту таблицу.

Блокировка SRX, удерживаемая транзакцией, запрещает другим транзакциям получать блокировки SRX по этой таблице и модифицировать эту таблицу. Транзакция не может вставлять, обновлять или удалять строки в таблице, если какая-то другая транзакция имеет блокировку SRX по этой таблице. Блокировка SRX, удерживаемая транзакцией, также запрещает другим транзакциям получать блокировки SRX, S и RX по этой таблице; иными словами, другие транзакции не могут успешно выполнять следующие предложения:

- LOCK TABLE таблица IN SHARE MODE;
- LOCK TABLE таблица IN SHARE EXCLUSIVE MODE;
- LOCK TABLE таблица IN ROW EXCLUSIVE MODE;
- LOCK TABLE таблица IN EXCLUSIVE MODE;

Эта блокировка вводит самый ограничительный режим, который обеспечивает транзакции, удерживающей эту блокировку по таблице,

возможность монопольной записи в таблицу. Эта блокировка запрашивается для таблицы следующим предложением LOCK TABLE таблица IN EXCLUSIVE MODE;

Все блокировки данных, получаемые транзакцией, включая все блокировки строк и таблиц, освобождаются при подтверждении или откате этой транзакции. Блокировки данных, полученные после точки сохранения, освобождаются, если транзакция откатывается к этой точке сохранения.

ORACLE автоматически конвертирует блокировку таблицы из более слабой в необходимую более строгую степень ограничений. Например, предположим, что транзакция использует предложение SELECT с фразой FOR UPDATE, чтобы заблокировать строки в таблице. Как следствие, она получает монопольные блокировки строк и разделяемую для строк блокировку таблицы. Если эта транзакция позднее обновляет одну или несколько заблокированных строк, блокировка таблицы автоматически конвертируется из режима RS в режим RX.

ORACLE никогда не прибегает к эскалации блокировок, когда СУБД автоматически заменяет многочисленные блокировки, полученные на одном уровне другой блокировкой на более высоком уровне.

Замки (latches) - это простые, низкоуровневые механизмы очередизации, которые защищают структуры разделяемых данных в SGA. Например, замки защищают список пользователей, обращающихся к базе данных в текущий момент, а также структуры данных, описывающие блоки в буферном кэше. Серверный или фоновый процесс получает замок на очень короткое время, пока он манипулирует или просматривает одну из таких структур. Реализация замков зависит от операционной системы, особенно в вопросе о том, ожидает ли процесс замка и сколь долго.

Внутренние блокировки - это более сложные механизмы, чем замки, и они служат разнообразным целям. Рассмотрим их назначение ниже для трех различных категорий внутренних блокировок:

1. **Блокировки кэша словаря.** Эти блокировки на очень короткое время удерживаются для записей словаря при использовании или модификации этих записей. Они гарантируют, что предложения SQL во время их разбора видят согласованные определения объектов. Блокировки кэша словаря могут быть разделяемыми и монопольными. Разделяемые блокировки освобождаются по окончании синтаксического разбора. Монопольные блокировки освобождаются по концу операции DDL.

2. **Блокировки управления файлами и журналом.** Эти блокировки защищают различные файлы. Например, одна блокировка защищает управляющий файл, чтобы его мог модифицировать лишь один процесс в каждый мо-

мент времени. Другая блокировка координирует использование и архивирование файлов журнала повторения. Файлы данных блокируются для того, чтобы гарантировать, что база данных монтируется несколькими экземплярами в разделяемом режиме или одного экземпляра в монопольном режиме. Поскольку блокировки файлов и журнала отражают состояние файлов, эти блокировки по необходимости удерживаются продолжительное время.

3. Блокировки табличных пространств и сегментов отката. Эти блокировки защищают табличные пространства и сегменты отката. Например, все экземпляры, имеющие доступ к базе данных, должны согласованно отражать онлайнное или офлайнное состояние табличного пространства. Сегменты отката блокируются для того, чтобы лишь один экземпляр мог писать в сегмент отката.

Во всех случаях ORACLE автоматически осуществляет блокировки для обеспечения одновременного доступа, целостности и согласованности данных на уровне предложения. Однако имеются возможности, которые позволяют вам перекрывать умалчиваемые механизмы блокировок ORACLE. Такое перекрытие полезно в следующих ситуациях:

- Приложениям требуется согласованность по чтению на уровне транзакций или "повторяемость чтений"; транзакциям необходимо опрашивать множество данных, согласованное на все время выполнения транзакции, с гарантией, что эти данные не будут изменены другими транзакциями в системе.
- Приложениям требуется, чтобы транзакция имела монопольный доступ к ресурсу; такая транзакция может выполняться, не ожидая завершения других транзакций.

Оборудование и материалы

Для выполнения практической работы необходим персональный компьютер с установленной СУБД Oracle10g/11g ExpressEdition и пакетом Microsoft Office. Инсталляция пакета представлена в приложении.

Указания по технике безопасности

Необходимо выполнять требования по технике безопасности, предусмотренные для компьютерного класса

8. Не включать компьютер без разрешения администратора класса и преподавателя.

9. Запрещается самостоятельно вскрывать корпус ПК.

10. Не касаться одновременно экрана монитора и клавиатуры (возможен разряд повышенного электростатического потенциала).

11. Не загромождать верхние панели устройств бумагами и посторонними устройствами.

12. Не прикасаться одновременно к металлическим частям ПК и устрой-

ствам, имеющим естественное заземление.

13. Запрещается производить отключение питания во время выполнения активной задачи.

14. Не допускать попадание влаги на поверхность системного блока, монитора, клавиатуру, принтеры и другие устройства.

Порядок выполнения работы

1. Примените все перечисленные в данной практической работе виды блокировок.

2. Обоснуйте результаты.

Содержание отчета и его форма

Отчет должен содержать

- название практической работы;
- цель работы;
- порядок выполнения работы;
- результаты выполнения;
- вывод о проделанной работе.

Контрольные вопросы и защита работы

Контрольные вопросы

1. Дайте определение блокировки таблиц, строк, столбцов.
2. Обоснуйте внешние блокировки.
3. Управление доступом?
4. В чем суть физической защиты?

Защита работы

Защита работы заключается в:

- выполнении заданий;
- ответах на контрольные вопросы;
- предоставлении отчета.

ПРАКТИЧЕСКАЯ РАБОТА.

СОЗДАНИЕ УЧЕТНОЙ ЗАПИСИ ПОЛЬЗОВАТЕЛЯ БАЗЫ ДАННЫХ ORACLE

Цель и содержание

Цель работы: создать и исследовать учетную запись пользователя базы данных oracle

Теоретическое обоснование

Словарь данных - не только центральное хранилище в каждой базе данных ORACLE. Это также важный инструмент для всех пользователей, от конечных пользователей до разработчиков приложений и администраторов базы данных. Даже начинающие пользователи могут извлекать выгоду из понимания и использования словаря данных.

Словарь данных является одной из важнейших частей базы данных ORACLE. Словарь данных - это набор таблиц, используемых как справочник только для чтения, который предоставляет информацию об ассоциированной с ним базе данных. Например, словарь данных может предоставлять следующую информацию:

- имена пользователей ORACLE
- привилегии и роли, которые были предоставлены каждому пользователю
- имена объектов схем (таблиц, представлений, снимков, индексов, кластеров, синонимов, последовательностей, процедур, функций, пакетов, триггеров и т.д.)
- информацию об ограничениях целостности
- умалчиваемые значения для столбцов
- сколько пространства было распределено и в настоящее время используется объектами в базе данных
- информацию аудита, например, кто обращался к различным объектам и обновлял их
- другую общую информацию о базе данных

Словарь данных структурирован через таблицы и представления, как и любые другие данные в базе данных. Для обращения к словарю данных вы используете SQL. Так как словарь данных можно только читать, пользователи могут выдавать лишь запросы (предложения SELECT) по таблицам и представлениям словаря данных.

Все базовые таблицы и представления словаря данных принадлежат пользователю ORACLE с учетным именем SYS. Поэтому ни один пользователь

ORACLE не должен изменять никаких объектов, содержащихся в схеме SYS, а администратор безопасности должен строго контролировать использование этого центрального учетного имени.

Данные в базовых таблицах словаря данных необходимы для функционирования ORACLE. Поэтому только ORACLE должен записывать или изменять информацию словаря данных.

Во время операций по базе данных ORACLE читает словарь данных, чтобы удостовериться, что нужные объекты существуют, и что пользователи имеют к ним должный доступ. ORACLE также непрерывно обновляет словарь данных, чтобы отражать происходящие изменения в структурах базы данных, аудите, грантах и данных.

Вы можете добавлять в словарь данных новые таблицы или представления. Если вы добавляете новые объекты в словарь данных, их владельцем должен быть SYSTEM или какой-либо третий пользователь ORACLE. Когда включен режим аудита, эта таблица может неограниченно расти. Хотя пользователи не должны удалять эту таблицу, администратор безопасности может удалять из нее данные, потому что строки этой таблицы служат лишь для информации и не являются необходимыми для работы ORACLE.

Представления словаря данных выступают как справочники для всех пользователей базы данных. Доступ к этим представлениям осуществляется через SQL. Некоторые представления доступны всем пользователям, тогда как некоторые другие предназначены лишь для администраторов.

Словарь данных всегда доступен при открытой базе данных. Он размещается в табличном пространстве SYSTEM, которое всегда находится в состоянии онлайн, когда база данных открыта.

Словарь данных состоит из нескольких наборов представлений. Во многих случаях такой набор состоит из трех представлений, содержащих аналогичную информацию и отличающихся друг от друга своими префиксами:

Столбцы в каждом представлении в наборе идентичны, со следующими исключениями:

- В представлениях с префиксом USER обычно нет столбца с именем OWNER (владелец); в представлениях USER под владельцем подразумевается пользователь, выдавший запрос.
- Некоторые представления DBA имеют дополнительные столбцы, которые содержат информацию, полезную для АБД.

Представления с префиксом USER:

- отражают собственное окружение пользователя в базе данных, включая информацию об объектах, созданных этим пользователем, грантах, предоставленных им, и т.д.
- выдают лишь строки, имеющие отношение к пользователю
- имеют столбцы, идентичные с другими представлениями, с тем исключением, что столбец OWNER подразумевается (текущий пользователь)
- возвращают подмножество информации, предоставляемой представлениями ALL могут иметь сокращенные общие синонимы для удобства

Представления с префиксом ALL отражают общее представление о базе данных со стороны пользователя. Эти представления возвращают информацию об объектах, к которым пользователь имеет доступ через общие или явные гранты привилегий и ролей, помимо тех объектов, которыми владеет этот пользователь. Представления с префиксом DBA показывают общее представление о базе данных, так что они предназначены только для администраторов базы данных. Точнее, опрашивать представления словаря с префиксом DBA может любой пользователь, имеющий системную привилегию SELECT ANY TABLE.

Существует две основные группы привилегий системного уровня: имеющие в составе имён слово ANY и не имеющие его. Привилегии со словом ANY в имени позволяют пользователю

Привилегии системного уровня могут быть даны либо непосредственно пользователю Oracle, либо роли, которая будет назначена пользователю Oracle.

Если нужно будет дать привилегии системного уровня каждому пользователю Oracle (включая любых пользователей Oracle, которые могут быть созданы в будущем), можно дать системную привилегию пользователю PUBLIC, что означает любого пользователя Oracle.

Привилегии системного уровня могут быть также предоставлены пользователю Oracle с использованием WITH ADMIN OPTION в конце оператора GRANT. Это позволяет пользователю Oracle, получившему такую системную привилегию, передавать её любому другому пользователю Oracle. По сути дела он становится ещё одним администратором этой привилегии.

Для изъятия привилегии или роли системного уровня у пользователя Oracle, можно использовать команду REVOKE. Ею можно определить сразу несколько системных привилегий и имён пользователей.

На любые объекты, созданные в то время, когда действовала эта привилегия, её отмена не повлияет. Кроме того, если системная привилегия изъята у пользователя PUBLIC, не будет затронут другой пользователь, которому эта системная привилегия была предоставлена непосредственно.

Любой пользователь предоставивший системную привилегию посредством WITH ADMIN OPTION, может предоставить эту системную привилегию другому пользователю Oracle – он, по сути дела, является дополнительным администратором этой системной привилегии.

Если пользователь Oracle владеет объектом наподобие таблицы, другим пользователям Oracle может быть разрешено использовать этот объект путём предоставления им одной или нескольких привилегий объектного уровня. Например, если пользователь user_1 намеревается позволить пользователю user_2 использовать его таблицу, но только для вставки и обновления строк с помощью команд INSERT или UPDATE, то пользователю user_2 могут быть даны привилегии UPDATE и INSERT.

- SELECT – позволяет другому пользователю Oracle сделать запрос к данным таблицы.
- INSERT – позволяет другому пользователю Oracle вставлять строки в таблицу с помощью команды INSERT.
- UPDATE – позволяет пользователю Oracle обновлять строки в таблице вне зависимости от того, были ли эти строки созданы этим пользователем или нет. Привилегия UPDATE
- DELETE – привилегия объектного уровня DELETE позволяет удалять из таблицы любые существующие строки. С использованием представления можно ограничить то, какие строки будут удалены.
- EXECUTE – даёт возможность пользователю Oracle, владеющему процедурным кодом базы данных (процедурами, функциями или пакетами), позволить другому пользователю Oracle вызывать его процедурные объекты.
- ALTER – даёт возможность пользователю Oracle изменить определение таблицы или последовательности.
- INDEX – позволяет пользователю Oracle создавать индексы на таблицы владельца.
- REFERENCES – позволяет пользователю Oracle обращаться к объектам другого пользователя.

В конце оператора GRANT для привилегии объектного уровня может быть определена фраза WITH GRANT OPTION, которая позволяет пользователю Oracle получившему эту привилегию, передать её другому пользователю Oracle.

Имеется ряд представлений словаря данных, которые показывают, какие привилегии объектного уровня были даны непосредственно пользователю и какие привилегии были предоставлены ему другими пользователями:

- USER_TAB_PRIVS

- USER_COL_PRIVS
- TABLE_PRIVILEGES
- COLUMN_PRIVILEGES

Представления словаря данных, начинающиеся с ALL_ и DBA_.

Роль(role) – совокупность системных привилегий и привилегий объектного уровня, которые были сгруппированы под одним именем, чтобы облегчить предоставление и отмену этих привилегий. Когда пользователю даётся роль, он получает все привилегии, связанные с ней.

Если пользователь Oracle удалён из базы данных командой DROP USER, можно сохранить присвоенные ему привилегии, если эти привилегии были назначены посредством роли. Затем можно переназначить эти привилегии любому новому пользователю. Если привилегии первоначально не были назначены посредством роли, то при уничтожении первого пользователя будет потеряна вся относящаяся к нему информация.

С помощью ролей можно расширять и сужать набор привилегий, назначенных пользователю, разрешая и запрещая роли.

Новой роли могут быть назначены привилегии уже существующей роли (роль более высокого уровня наследует все привилегии роли более низкого уровня).

Как только пользователю Oracle был дан набор ролей, предусмотренные ими привилегии могут включаться и выключаться (обычно прикладными программами) посредством разрешения и запрещения ролей.

Вместе с новой базой данных автоматически создаются пять ролей. Это – CONNECT, RESOURCE, DBA, EXP_FULL_DATABASE и IMP_FULL_DATABASE. Первые три предусмотрены для совместимости с предыдущими версиями программного обеспечения Oracle и обычно не требуются.

Центральное место в средствах защиты занимает учётная запись пользователя базы данных Oracle.

CREATE USER – это команда SQL, которая может использоваться для определения учётной записи Oracle в базе данных. После создания учётной записи пользователя Oracle она не может использоваться, пока пользователь не получит, по меньшей мере одну системную привилегию. Системная привилегия CREATE SESSION позволяет пользователю создавать сеанс по отношению к базе данных Oracle. Это – необходимая привилегия, которую должна иметь учётная запись пользователя, без неё учётная запись пользователя Oracle не может использоваться.

При первоначальном создании пользователя Oracle можно определить заданное по умолчанию табличное пространство, в котором будут создаваться объекты пользователя. Если заданное по умолчанию табличное пространство не определено, пользователю будет назначено табличное пространство SYSTEM в качестве заданного по умолчанию, которое будут использовать объекты базы данных. В составе оператора CREATE USER может использоваться фраза DEFAULT TABLESPACE для определения того, что объекты пользователя должны быть помещены в табличное пространство, отличное от SYSTEM. Пользователю Oracle также должна быть назначена квота, которая определяет, сколько памяти он может использовать в табличном пространстве.

Другой способ создания пользователя состоит в том, чтобы предоставить пользователю роли CONNECT, RESOURCE и DBA. Хотя это и быстрый метод, он включён, прежде всего, для совместимости с предыдущими версиями программного обеспечения Oracle. Команда CREATE USER – более предпочтительный метод, поскольку в одной команде можно указать и квоту и другие установки.

Можно использовать команду ALTER USER для изменения таких параметров пользователя, как пароль, заданные по умолчанию временные табличные пространства и квота памяти.

Для удаления пользователя из базы данных используется команда DROP USER, которая удаляет запись пользователя из словаря данных Oracle. Если пользователь Oracle владеет какими-либо объектами базы данных, можно либо удалить каждый из объектов перед использованием команды DROP USER, либо использовать в DROP USER опцию CASCADE для автоматического уничтожения всех объектов при удалении учётной записи пользователя.

Словарь данных каждой базы данных содержит таблицу с именем SUS.AUD\$, обычно называемую АУДИТОРСКИМ ЖУРНАЛОМ базы данных. Аудиторские записи, генерируемые как результат отслеживания предложений, привилегий или объектов, можно помещать как в аудиторскую таблицу базы данных, так и в аудиторский журнал операционной системы. Использование аудиторского журнала дает возможность просматривать выбранные порции аудиторского журнала с помощью predefined представлений словаря данных, а также использовать инструменты ORACLE, такие как SQL*ReportWriter, для генерации аудиторских отчетов.

Использование аудиторского журнала операционной системы помогает консолидировать аудиторские записи из многих источников, включая ORACLE и другие приложения. Поэтому исследование активности системы может быть более эффективным, так как аудиторские записи сосредоточены в одном месте.

Аудиторский журнал базы данных (SYS.AUD\$) - это единственная таблица в словаре данных каждой базы данных ORACLE. Если вы решили использовать аудит, создайте аудиторские представления, подключившись как SYS и запустив скрипт CATAUDIT.SQL. Если вы решили отключить аудит и больше не нуждаетесь в аудиторских представлениях, удалите их, подключившись как SYS и запустив скрипт CATNOAUD.SQL. Точное имя и местоположение этого скрипта зависит от операционной системы.

Все опции аудита генерируют следующую общую информацию:

- имя пользователя, выполнявшего отслеживаемое предложение;
- код действия, указывающий выполненное предложение;
- объекты, адресуемые в отслеживаемом предложении;
- дату и время выполнения отслеживаемого предложения;

Аудиторский журнал не сохраняет информации о каких-либо значениях данных, которые могли быть вовлечены в отслеживаемое предложение; например, при аудите предложения UPDATE не сохраняются старые и новые значения данных. Однако, такой специализированный тип аудита можно осуществить для предложений DML, работающих с таблицами, с помощью триггеров базы данных.

ORACLE позволяет устанавливать опции аудита на трех уровнях:

- предложение аудита базируется на типе предложений SQL, например, на любых предложениях SQL по таблицам (что регистрирует каждое предложение CREATE, TRUNCATE и DROP TABLE)
- привилегия отслеживает использование конкретной системной привилегии, такой как CREATE TABLE
- объект отслеживает конкретные типы предложений на конкретных объектах, например, ALTER TABLE по таблице EMP

Для удобства спецификации часто встречающихся групп связанных опций аудита предоставляются специальные обозначения. Эти обозначения сами не являются опциями; они просто позволяют указать одним словом целую группу опций в предложении AUDIT или NOAUDIT.

Любой пользователь базы данных ORACLE может в любой момент установить опции аудита предложений, привилегий или объектов, но ORACLE не генерирует аудиторских записей и не помещает их в аудиторский журнал, если не включен режим аудита базы данных. Обычно за эту операцию отвечает администратор. Аудит базы данных включается и выключается параметром инициализации AUDIT_TRAIL в файле параметров базы данных. Этот параметр может быть установлен в следующие значения:

- DB - включает аудит базы данных и направляет все аудиторские записи в аудиторский журнал базы данных
- OS - включает аудит базы данных и направляет все аудиторские записи в аудиторский журнал операционной системы
- NONE - выключает аудит (умолчание)

Администратор защиты обязан контролировать рост аудиторского журнала и его размер. Когда аудит включен и генерируются аудиторские записи, аудиторский журнал растет за счет двух факторов:

- числа включенных опций аудита
- частоты выполнения отслеживаемых предложений

Для контроля за ростом аудиторского журнала вы можете использовать следующие методы:

- Включать и выключать аудит базы данных. Когда аудит включен, аудиторские записи генерируются и поступают в журнал; когда аудит выключен, аудиторские записи не генерируются.

- Жестко контролировать возможности осуществлять аудит объектов. Это можно делать двумя различными способами:

- Всеми объектами владеет администратор защиты, привилегия AUDIT ANY никогда не назначается никаким другим пользователям. Все объекты схемы могут принадлежать схеме, соответствующий пользователь которой не имеет привилегии CREATE SESSION.

- Все объекты содержатся в схемах, которые не соответствуют реальным пользователям базы данных (т.е. привилегия CREATE SESSION не назначена пользователям, одноименным со схемами), и администратор защиты является единственным лицом, имеющим системную привилегию AUDIT ANY.

- Все объекты содержатся в схемах, которые не соответствуют реальным пользователям базы данных (т.е. привилегия CREATE SESSION не назначена пользователям, одноименным со схемами), и администратор защиты является единственным лицом, имеющим системную привилегию AUDIT ANY.

После того, как аудит был включен в течение некоторого времени, администратор защиты может удалить записи из аудиторского журнала, - как для того, чтобы освободить память, так и для облегчения управления этим журналом. Например, чтобы удалить BCE записи из аудиторского журнала, введите следующее предложение:

```
DELETE FROM sys.aud$;
```

Если информация аудиторского журнала должна архивироваться для целей накопления истории, администратор защиты может скопировать соответствующие записи в нормальную таблицу базы данных или

экспортировать аудиторскую таблицу в файл операционной системы. Удалять записи из аудиторского журнала базы данных может лишь пользователь SYS, т.е. пользователь, имеющий привилегию DELETE ANY TABLE (или пользователь, которому SYS передал привилегию DELETE по таблице SYS.AUD\$).

Как и в любой таблице базы данных, после удаления записей из аудиторского журнала базы данных все экстенды, которые были распределены этой таблице, будут по-прежнему существовать. Если аудиторский журнал имеет слишком много экстендов, большинство из которых не используются, то можно уменьшить размер этого журнала, выполнив следующие шаги:

1. Если вы хотите сохранить информацию из аудиторского журнала, скопируйте ее в другую таблицу базы данных, или экспортируйте ее с помощью утилиты экспорта.

2. Соединитесь с базой данных как INTERNAL.

3. Выполните усечение таблицы SYS.AUD\$ с помощью команды TRUNCATE.

4. Перезагрузите аудиторские записи, сохраненные вами на шаге 1.

Осуществляя отслеживание подозрительной деятельности в базе данных, защищайте целостность записей аудиторского журнала, чтобы гарантировать точность и полноту аудиторской информации. Чтобы защитить аудиторский журнал от несанкционированных удалений, назначайте системную привилегию DELETE ANY TABLE только администраторам защиты. Чтобы отслеживать изменения, выполняемые над самим аудиторским журналом, организуйте аудит аудиторского журнала с помощью следующего предложения:

```
AUDIT INSERT, UPDATE, DELETE
ON sys.aud$
BY ACCESS;
```

Аудит с помощью триггеров базы данных

Вы можете использовать триггеры, чтобы дополнить встроенные средства аудита ORACLE. Хотя вы можете писать триггеры, которые регистрировали бы информацию, аналогичную той, которую генерирует команда AUDIT, старайтесь использовать триггеры лишь в том случае, когда вам необходима более детальная аудиторская информация. Например, с помощью триггеров вы можете отслеживать изменения значений по строкам таблицы.

Целостность данных определением правил проверки достоверности данных гарантирующих, что недействительные данные не попадут в ваши таблицы. Oracle позволяет определять и хранить эти правила для объектов БД, которых они касаются, таким образом, чтобы кодировать их только однажды. При этом они активируются всякий раз, когда какой-либо вид изменение

проводится в таблице, независимо от того, какая программа выполняет вставки, модификации или удаления. Этот контроль осуществляется в форме ограничений и триггеров БД. Ограничения – это правила, применимые к таблицам во времена или после создания, распространяемые на то, как эти таблицы могут заполняться.

Ограничение целостности устанавливает правила на уровне БД, определяя набор проверок для таблиц системы. Эти проверки автоматически выполняются всякий раз, когда вызывается оператор вставки, модификации или удаления данных в таблице. Если какие либо ограничения нарушены, операторы отменяются. Поскольку ограничения условности проверяются на уровне БД, они выполняются независимо от того, откуда были инициированы операторы вставки, модификации или удаления. Для таблиц можно задавать следующие типы ограничений целостности:

- NOT NULL
- PRIMARY KEY
- UNIQUE KEY
- FOREIGN KEY (REFERENCES)
- CHECK
- INDEX (ИНДЕКСЫ)
- TRIGGERS и PROCEDURES

NOT NULL. Это ограничение устанавливается для столбца, чтобы указать, что столбец должен иметь значение в каждой строке, т.е. некоторое непустое значение.

PRIMARY KEY (первичный ключ). Ограничение определяет столбец или группу столбцов, которую можно использовать для уникальной идентификации строки. Никакие две строки в таблице не могут иметь одинаковые значения столбцов первичного ключа. Кроме того, столбцы первичного ключа должны всегда содержать значение. Все эти условия гарантируют то, что в нашем распоряжении будет одна и только одна строка, соответствующая критериям связывания. Первичные ключи могут быть или именованные (пользователем) или неименованные (Oracle составляет имя сам). В первичных ключах не могут использоваться столбцы типа: raw, long, long raw.

UNIQUE (уникальный). Ограничение UNIQUE используется для определения того, что значения в столбце не должно повторяться в другой строке этой таблицы, определяет вторичный ключ для таблицы. Это столбец или группа столбцов, которые можно использовать как уникальную идентификацию строки. Никакие две строки не могут иметь одинаковые значения для столбца или столбцов ключа UNIQUE. Столбцы для ограничения

UNIQUE не обязательно NOT NULL. Можно сформировать ограничение таблицы, указав, что в таблице не должна повторяться комбинация столбцов. К примеру: можно в начале объявить стандартно emp_id number(5), person_id date а под конце объявить что: unique(emp_id, person_id) – и получится, что сочетание значений этих полей, должно быть уникальным в каждой строке.

FOREIGN KEY (внешний ключ). FOREIGN KEY, устанавливает отношение целостности между таблицами. Оно требует, чтобы столбец или набор столбцов в одной таблице совпадал с первичным или вторичным ключом другой таблицы. С момента создания внешнего ключа ссылающегося на первичный ключ некой таблицы удаление таблицы будет – запрещено. И обойти это ограничение можно только удалив ограничение. Внешние ключи могут быть именованные или неименованные.

CHECK. Ограничение CHECK определяет логику проверки, которая должна жать результат true (истина) для оператора вставки, модификации или удаления из таблицы. Ограничение CHECK гарантирует, что значение в измененной строке удовлетворяют заданному набору проверок правильности.

ИНДЕКСЫ (INDEX). Ограничения PRIMARY KEY и UNIQUE автоматически создают индексы на столбцах, для которых они определены, если ограничение активизируется при создании. Если индекс уже существует на столбцах, которые составляют ограничение PRIMARY KEY и UNIQUE, то использует именно этот индекс и Oracle не может создать новый.

TRRIGERS(Триггеры) – с программный элемент хранимый в БД выполняемый автоматически, в определенных ситуациях, не имеющий входных или выходных параметров, что в конечном итоге и является причиной невозможности вызвать его явно, непосредственно, его вызывает только сама база данных Oracle. Выходные данные триггера должны быть также применимы к БД, а не возвращены вызывающей программе или отображены на экране. Процедуры и функции расписаны в вопросе № 6.Схемы и объекты схемы

Таким образом, целостность базы данных может быть рассмотрена на трех уровнях:

1. На уровне типа данных (т.е. соответствия типов данных)
2. На уровне ключей (к примеру, соответствие первичных и внешних ключей)
3. На уровне триггеров, процедур и/или функций(к примеру, триггера отвечают только за свои области данных).

Оборудование и материалы

Для выполнения практической работы необходим персональный компьютер с установленной СУБД Oracle10g/11g ExpressEdition и пакетом Microsoft Office. Инсталляция пакета представлена в приложении.

Указания по технике безопасности

Необходимо выполнять требования по технике безопасности, предусмотренные для компьютерного класса

1. Не включать компьютер без разрешения администратора класса и преподавателя.
2. Запрещается самостоятельно вскрывать корпус ПК.
3. Не касаться одновременно экрана монитора и клавиатуры (возможен разряд повышенного электростатического потенциала).
4. Не загромождать верхние панели устройств бумагами и посторонними устройствами.
5. Не прикасаться одновременно к металлическим частям ПК и устройствам, имеющим естественное заземление.
6. Запрещается производить отключение питания во время выполнения активной задачи.
7. Не допускать попадание влаги на поверхность системного блока, монитора, клавиатуру, принтеры и другие устройства.

Порядок выполнения работы

1. Создайте учетную запись пользователя.
2. Изучите возможности работы с БД.
3. Добавьте пользователя под другой учетной записью.
4. Введите ограничения на используемые данные БД.
5. Удалите учетную запись..

Содержание отчета и его форма

Отчет должен содержать

- название практической работы;
- цель работы;
- порядок выполнения работы;
- результаты выполнения;
- вывод о проделанной работе.

Контрольные вопросы и защита работы

Контрольные вопросы

1. Учётная запись пользователя базы данных Oracle.
2. Роли CONNECT, RESOURCE и DBA.
3. Какова команда SQL, которая может использоваться для определения

учётной записи Oracle в базе данных?

4. Для чего используется фраза DEFAULT TABLESPACE?
5. Как применяется команда CREATE USER?
6. Когда используется команда DROP USER
7. Что такое аудит базы данных?
8. Установка опций аудита?

Защита работы

Защита работы заключается в:

- выполнении заданий;
- ответах на контрольные вопросы;
- предоставлении отчета.

ПРАКТИЧЕСКАЯ РАБОТА.

РАЗРАБОТКА БАЗЫ ДАННЫХ (ФАЙЛЫ ПАРАМЕТРОВ)

Цель и содержание

Цель работы: разработать и исследовать базу данных

Теоретическое обоснование

При создании базы данных необходимо подготовить несколько файлов данных операционной системы, которые будут использоваться вместе как единая база данных. База данных создается один раз, независимо от того, сколько файлов данных она имеет, и сколько экземпляров будут обращаться к ней. Процедуру создания базы данных можно также использовать для того, чтобы стереть информацию в существующей базе данных и создать новую базу данных с тем же именем и физической структурой.

Создание базы данных включает следующие операции:

- создание новых файлов данных, или стирание данных, хранившихся в предыдущих файлах данных
- создание структур, требующихся ORACLE для доступа и работы с базой данных (словаря данных)
- создание и инициализация управляющих файлов и файлов журнала повторения для базы данных

База данных создается с помощью предложения, включающего команду SQL CREATE DATABASE; однако, прежде чем выдавать такое предложение, рассмотрите следующие вопросы:

- Спланируйте ваши таблицы и индексы, и оцените, сколько пространства они потребуют.
- Спланируйте проблемы защиты вашей базы данных, включая конфигурацию ее онлайн-журнала и архивного журналов (с учетом занимаемого ими пространства) и стратегию резервного копирования.
- Выберите набор символов базы данных. Вы должны указать набор символов при создании базы данных, и не сможете изменить его до повторного создания базы данных. Все символьные данные в базе данных, включая данные в словаре данных, хранятся в наборе символов базы данных. Если пользователи предполагают обращаться к базе данных, используя другие наборы символов, то набор символов базы данных должен быть надмножеством всех используемых наборов символов.

Для создания новой базы данных вы должны иметь:

- привилегии операционной системы, соответствующие полноправному администратору базы данных

- достаточное количество памяти для запуска экземпляра ORACLE
- достаточный объем дискового пространства на компьютере, выполняющем ORACLE

Для создания каждой новой базы данных ORACLE необходимо последовательно выполнить следующие шаги:

1. Скопировать все существующие базы данных.
2. Создать файлы параметров.
3. Отредактировать новые файлы параметров.
4. Проверить идентификатор экземпляра ORACLE.
5. Запустить SQL*DBA и соединиться с ORACLE как INTERNAL.
6. Запустить экземпляр.
7. Создать базу данных.
8. Скопировать базу данных.

Запуск и останов базы данных

Для запуска базы данных или инстанции(экземпляр) используйте либо диалоговое окно Start Up Instance, либо команду STARTUP (после того, как соединитесь с ORACLE как INTERNAL). Вы можете запустить экземпляр и базу данных различными способами:

- запустить экземпляр без монтирования базы данных
- запустить экземпляр и смонтировать базу данных, но оставить ее закрытой
- запустить экземпляр, смонтировать и открыть базу данных в одном из следующих режимов:
 - неограниченном режиме (доступна всем пользователям)
 - ограниченном режиме RESTRICTED (доступна только АБД)

Кроме того, вы можете форсировать запуск экземпляра, либо заставить экземпляр начать немедленно после запуска полное восстановление носителя.

Прежде, чем запускать экземпляр, нужно подключиться как INTERNAL; также может понадобиться указать, для какой базы данных вы запускаете экземпляр, и специфицировать файл параметров.

Вы также должны подключиться как INTERNAL. Это условие обязательно, независимо от того, используете ли вы графический интерфейс SQL*DBA или команды SQL.

Можно запустить экземпляр без монтирования базы данных; обычно это требуется лишь при создании базы данных. Чтобы сделать это, используйте одну из следующих опций SQL*DBA:

- кнопку Nomount в диалоговом окне Start Up Instance
- команду STARTUP с опцией NOMOUNT

Вы можете запустить экземпляр с монтированием базы данных, но не открывать базу данных, чтобы выполнить специфические операции сопровождения. Например, база данных должна быть смонтирована, но не открыта, во время выполнения следующих задач:

- переименования файлов данных
- добавления, удаления или переименования файлов журнала
- включения и выключения опций архивирования журнала
- полного восстановления базы данных

Для запуска экземпляра и монтирования базы данных без ее открытия используйте одну из следующих опций SQL*DBA:

- кнопку Mount в диалоговом окне Start Up Instance
- команду STARTUP с опцией MOUNT

При использовании команды STARTUP с параметром NOMOUNT производится только запуск фоновых процессов Oracle и распределение SGA в памяти. В состоянии NOMOUNT базу данных может использовать только DBA. Опция NOMOUNT обычно используется только при создании базы данных.

Нормальная работа базы данных означает, что экземпляр запущен, а база данных смонтирована и открыта. Это позволяет всем действительным пользователям соединяться с базой данных и выполнять типичные операции, требующие доступа к данным. Для запуска экземпляра с монтированием базы данных и ее открытием используйте одну из следующих опций SQL*DBA:

- кнопку Open в диалоговом окне Start Up Instance
- команду STARTUP с опцией OPEN

При запуске экземпляра базы данных специфицируйте имя базы данных, которая будет монтироваться, одним из следующих способов:

- введите имя базы данных в поле Database Name в диалоговом окне Start Up Instance
- укажите имя базы данных в команде STARTUP

При запуске экземпляра базы данных выберите файл параметров для инициализации характеристик экземпляра, одним из следующих способов:

- введите имя файла параметров в поле Parameter File в диалоговом окне Start Up Instance
- укажите полное имя файла параметров в опции PFILE команды STARTUP

Спецификации имен файлов зависят от операционной системы. Если имя файла не указано, ORACLE использует умалчиваемое имя файла.

Форсированный запуск, описанный ниже, следует применять ТОЛЬКО в следующих случаях:

- Текущую работающую инстанцию не удастся закрыть с помощью опций Normal или Immediate меню Shut Down (или эквивалентных опций предложения SHUTDOWN)

- Экземпляр не удастся запустить обычным способом

Если вы попали в такую ситуацию, обычно удастся решить проблему путем запуска нового экземпляра в форсированном режиме, используя одну из следующих опций SQL*DBA:

- переключатель Force в диалоговом окне Start Up Instance
- команду STARTUP с опцией FORCE

Эти опции в действительности сначала останавливают текущую инстанцию, а затем запускают новую инстанцию (возможно, с монтированием и открытием базы данных).

Если вы знаете, что необходимо восстановление носителя, то вы можете запустить инстанцию так, чтобы она автоматически начала процесс восстановления, используя одну из следующих опций SQL*DBA:

- переключатель Recover в диалоговом окне Start Up Instance
- команду STARTUP с опцией RECOVER

Если ваш сервер ORACLE позволяет обращаться к одной базе данных из нескольких инстанций, то вы должны выбрать монтирование базы данных в монопольном или параллельном режимах.

На многих установках одна или несколько инстанций и баз данных ORACLE автоматически запускаются вслед за загрузкой ОС. Процедуры, позволяющие организовать такой режим, специфичны для каждой операционной системы.

Если ваш локальный сервер ORACLE является частью распределенной базы данных, то вам может понадобиться запускать удаленную инстанцию и базу данных. Процедуры запуска и останова удаленных инстанций широко варьируются в зависимости от коммуникационного протокола и операционной системы.

Чтобы инициировать останов базы данных, используйте либо меню Shut Down, либо команду SHUTDOWN в SQL*DBA.

Нормальный останов базы данных протекает следующим образом:

- После получения команды на останов не допускаются новые соединения с базой данных.

- Прежде чем остановить базу данных, ORACLE ждет отсоединения от нее всех текущих соединенных пользователей.

- Очередной запуск базы данных не потребует никаких процедур восстановления инстанции.

Для останова базы данных в нормальных обстоятельствах используйте одну из следующих опций SQL*DBA:

- опцию Normal меню Shut Down
- команду SHUTDOWN с опцией NORMAL

В чрезвычайных обстоятельствах вы можете остановить базу данных немедленно. Используйте этот способ останова лишь в случаях, подобных следующим:

- Скоро произойдет отключение питания.
- База данных или одно из ее приложений работает неверно.
- Немедленный останов базы данных протекает следующим образом:
- Обработка текущих предложений SQL от клиентов немедленно прекращается.
- Все неподтвержденные транзакции откатываются. (Если есть длинные неподтвержденные транзакции, этот способ останова может оказаться достаточно продолжительным, несмотря на свое название.)
- ORACLE не ждет отключения текущих соединенных пользователей; ORACLE неявно откатывает активные транзакции и разрывает все пользовательские соединения.
- Очередной запуск базы данных может потребовать восстановления инстанции (которое ORACLE выполнит автоматически).

Для немедленного останова базы данных используйте одну из следующих опций SQL*DBA:

- опцию Immediate меню Shut Down
- команду SHUTDOWN с опцией IMMEDIATE

Эта секция приводит примеры останова базы данных и инстанции через интерфейс меню и команды SQL*DBA. Во всех примерах предполагается, что АБД уже подключен как INTERNAL.

Меню Shut Down останавливает базу данных.

Команда SHUTDOWN эквивалентна меню Shut Down. Например, следующее предложение является командным эквивалентом меню Shut Down.

В большинстве операционных систем вы можете запускать инстанцию ORACLE либо в однопроцессном, либо в многопроцессном режиме, независимо от того, как ORACLE был установлен или запускался последний раз. Если компьютер, на котором выполняется сервер ORACLE, поддерживает многопроцессность, то инстанции баз данных обычно запускаются в многопроцессном режиме, так что много пользователей могут

одновременно обращаться к разделяемой базе данных; однопроцессные же экземпляры поддерживают лишь одного пользователя в каждый момент. Однако, в некоторых экспериментальных ситуациях, вы можете найти полезным запустить экземпляр в однопроцессном режиме. Некоторые операционные системы (такие как MS-DOS) не поддерживают многопроцессности или разделяемой памяти; в таких системах однопроцессная экземпляра является единственной возможностью.

Если над базой данных производят любое из ниже перечисленных структурных изменений, базы данных, непосредственно перед изменениями и после делается соответствующее копирование базы данных:

- Создание или удаление табличного пространства
- Добавление или переименование (перемещение) файла данных в существующем табличном пространстве
- Добавление, переименование(перемещение) или удаление группы или члена онлайн-журнала повторения.
- Если база данных работает в режиме ARCHIVELOG, то до и после структурного изменения базы данных требуется лишь резервное копирование управляющего файла базы данных (с помощью команды ALTER DATABASE с опцией BACKUP CONTROLFILE). Можно скопировать и другие части базы данных.
- Если база данных работает в режиме NOARCHIVELOG, то непосредственно перед и после изменения базы данных требуется сделать полное копирование файла базы данных, включая все файлы данных, файлы журнала повторения и управляющие файлы.

Существует, по большому счету, два вида резервного копирования :

1. Непротиворечивое (холодное) резервное копирование, ситуация когда, копии создаются, в случае закрытой БД (close) для пользователей. Копия базы данных, созданной в автономном режиме, содержит: все файлы данных, журналы повторов и управляющие файлы. После останова БД, все файлы базы данной по средствам ОС копируются на один из backup дисков.

Этапы:

- Остановка экземпляра БД Oracle – в режиме shutdown normal (игнорирование, новых подключений и ожидание отключения все зарегистрированных пользователей) или shutdown immediate (немедленное прерывание всех соединений, выполнение операции отката на всех транзакциях ожидающих обработки)

- Копирование всех физических файлов, относящихся к базе данных, управляющие файлы, файлы журнала обновления и файлы базы данных.

- Закончить работу, перезагрузить базу данных

2. Резервное(горячее) копирование в оперативном режиме, к примеру, когда БД работает в архивном режиме ARCHIVELOG, БД все время находиться в оперативном режиме таким образом доступна пользователям.

Этапы:

- Перевод табличного пространства в режим резервного копирования.

- Копирование всех файлов базы данных, связанных с табличным пространством.

- Выведение табличное пространство из режима резервного копирования.

- Повторение действий с первого по третье, пока не будет выполнено резервное копирование всех табличных пространств.

- Копирование управляющего файла.

- Копирование оперативного журнала обновления.

В режиме ARCHIVELOG:

- Требуется дополнительное дисковое пространство

- Управление архивными журналами влечет за собой дополнительные административные непроизводительные затраты.

- Применимо горячее резервное копирование.

- В случае отказа носителя может быть выполнено полное восстановление базы данных.

В режиме NOARCHIVELOG:

- Не требуется дополнительное дисковое пространство или непроизводительные затраты.

- Может использоваться только холодное резервное копирование.

- В случае отказа теряется вся работа, выполненная со времени последнего резервного копирования.

Вы устанавливаете первоначальный режим архивирования базы данных во время ее создания. В большинстве случаев, во время создания базы данных вы можете выбрать режим по умолчанию NOARCHIVELOG, потому что нет необходимости архивировать информацию, генерируемую за этот период. После того как база данных создана, решите, нужно ли изменить первоначальный режим архивирования. После того, как база данных создана, всегда можно пе-

включать режим архивирования базы данных. Однако, как правило, следует выбрать постоянный режим работы базы данных.

Чтобы автоматическое архивирование заполненных групп было включено установите в TRUE значение параметра LOG_ARCHIVE_START в файле параметров базы данных INIT.ORA:

LOG_ARCHIVE_START = TRUE Это значение будет иметь эффект при очередном запуске базы данных.

Вы можете выключить автоматическое архивирование журнала в любой момент. Однако, выключив автоматическое архивирование, вы должны вручную, периодически и своевременно, архивировать заполняемые группы журнала. Если база данных работает в режиме ARCHIVELOG, автоматическое архивирование выключено, а группы журнала заполняются, но не архивируются, то процесс LGWR не сможет повторно использовать неактивные группы журнала, чтобы продолжать запись информации повторения. Поэтому работа базы данных будет временно приостановлена до тех пор, пока не будет выполнено необходимое архивирование. Автоматическое архивирование может быть выключено как до, так и после запуска инстанции.

Если база данных работает в режиме ARCHIVELOG, то можно копировать индивидуальное табличное пространство или даже индивидуальный файл. Эта возможность полезна, если одна часть базы данных используется более интенсивно, чем другие, - например, табличное пространство SYSTEM или табличные пространства, содержащие сегменты отката. Если сбой диска повреждает один из таких файлов данных, для его реставрации может быть использована ранняя копия, и меньшее число изменений необходимо применить при прокрутке вперед, чтобы восстановить файл к состоянию на момент сбоя, т.е. время, затрачиваемое на восстановление, сокращается.

В принципе, существуют более упрощенные методики резервного копирования, с использованием программных средств администрирования баз данных - копирования с помощью Oracle Enterprise Manager(OEM), но эти методики уступают по критерию надежности SQL*Plus'у.

Для более подробного ознакомления с процессами восстановления и копирования следует дополнительно ознакомиться с вопросом №10 Журнал повторения(redo log)

Описанный в стандарте SQL набор операторов SQL предназначен для встраивания в программу на обычном языке программирования. Поэтому в этом наборе перемешаны операторы "истинного" реляционного языка запросов (например оператор удаления из таблицы части строк, удовлетворяющих

заданному условию) и операторы работы с курсорами, позволяющими обеспечить строчный доступ к таблице-результату запроса.

В диалоговом режиме набор операторов SQL и их синтаксис должен быть несколько другим. Весь вопрос состоит в том, как реализовывать такую диалоговую программу. Правила встраивания стандартного SQL в программу на обычном языке программирования предусматривают, что вся информация, касающаяся операторов SQL, известна (за исключением значений переменных, используемых в качестве констант в операторах SQL). Не предусмотрены стандартные средства компиляции с последующим выполнением операторов, которые становятся известными только во время выполнения. Поэтому, опираясь только на стандарт, невозможно реализовать диалоговый монитор взаимодействия с БД на языке SQL или другую прикладную программу, в которой текст операторов SQL возникает во время выполнения, т. е. фактически так или иначе стандарт необходимо расширять.

Один из возможных путей расширения состоит в использовании специальной группы операторов, обеспечивающих динамическую компиляцию (во время выполнения прикладной программы) базового подмножества операторов SQL и поддерживающих их корректное выполнение. В дополнительный набор операторов, поддерживающих динамическую компиляцию базовых операторов SQL, входят операторы: PREPARE, DESCRIBE и EXECUTE.

Оператор подготовки

Во время выполнения оператора PREPARE символьная строка, содержащаяся в host-string-variable, передается компилятору SQL, который обрабатывает таким же образом, как если бы получил в статике. Построенный при выполнении оператора PREPARE код остается действующим до конца транзакции или до повторного выполнения данного оператора PREPARE в пределах этой же транзакции.

Оператор DESCRIBE предназначен для того, чтобы определить тип ранее подготовленного оператора, узнать количество и типы формальных параметров(если они есть) и количество и типы столбцов результирующей таблицы, если подготовленный оператор является оператором выборки (SELECT). Действие оператора DESCRIBE состоит в том, что в указанную область памяти прикладной программы (структура этой области фиксирована и известна пользователям) помещается информация, характеризующая ранее подготовленный оператор с заданным именем.

Оператор EXECUTE служит для выполнения ранее подготовленного оператора SQL типа "N" (не требующего применения курсора) или для совмещенной подготовки и выполнения такого оператора.

Для выполнения подготовленного оператора служит первый вариант оператора EXECUTE. Число и типы фактических параметров должны соответствовать числу и типам формальных параметров подготовленного оператора.

Для использования таких операторов используется расширение механизма курсоров стандарта SQL. Во-первых, при определении курсора можно указывать не только литеральную спецификацию курсора, но и имя оператора, вводимое с помощью оператора PREPARE (в этом случае оператор PREPARE должен текстуально находиться выше оператора DECLARE). Тем самым полный синтаксис оператора DECLARE становится следующим:

```
<declare cursor> ::=
DECLARE <cursor name> CURSOR
FOR { <cursor specification> | <statement-name> }
```

Далее, поскольку для такого курсора в статике неизвестна информация о входных и выходных переменных включающей программы, то используется другая форма операторов OPEN и FETCH.

Полный синтаксис этих операторов становится следующим:

```
<open statement> ::=
OPEN <cursor name>
[USING { <host-vars-list> | DESCRIPTOR <descr-name> }]
<fetch statement> ::=
FETCH <cursor name>
{ INTO <fetch target list> |
USING <host-vars-list> |
USING DESCRIPTOR <descr-name> }
```

Как видно, предлагается два способа задания фактических входных и выходных параметров: прямое с указанием в операторах OPEN и/или FETCH списков имен переменных включающей программы и косвенное, когда число параметров и их адреса сообщаются через дополнительную структуру-дескриптор.

Первый способ предлагается использовать для работы с операторами выборки, для которых фиксирован набор формальных входных и выходных параметров.

Второй способ работы с динамически откомпилированными операторами, требующими использования курсоров, состоит в использовании дескрипторов динамически формируемых списков параметров.

Объектно-ориентированные Базы Данных.

Направление объектно-ориентированных баз данных (ООБД) возникло сравнительно давно. Публикации появлялись уже в середине 1980-х. Однако наиболее активно это направление развивается в последние годы. С каждым годом увеличивается число публикаций и реализованных коммерческих и экспериментальных систем.

Общие понятия объектно-ориентированного подхода и их преломление в ООБД

В наиболее общей и классической постановке объектно-ориентированный подход базируется на концепциях:

- объекта и идентификатора объекта;
- атрибутов и методов;
- классов;
- иерархии и наследования классов.

Объектный тип — это расширение типа, определяемого пользователем, позволяющее инкапсулировать методы с элементами данных в едином логическом модуле. Определение объектного типа служит в качестве шаблона, но не распределяет память. Объекты хранятся физически как строки или столбцы таблицы. Также в Oracle введены объекты трёх новых встроенных типов: объекты типа `var`, типа `ref`, и ‘большие объекты’ типа LOB.

- Объекты типа `var` представляют собой один из примеров коллекции Oracle8, которая примерно соответствует массиву. Объекты этого типа полезны при обработке небольших коллекций в коде PL/SQL.
- Объект типа `ref` представляет собой указатель на объектный тип, определяемый пользователем.
- Объект типа LOB предназначен для работы с ‘большими’ объектами, такими как полные документы, видеоизображения, цифровые звукозаписи и т.д.

Любая сущность реального мира в объектно-ориентированных языках и системах моделируется в виде объекта. Любой объект при своем создании получает генерируемый системой уникальный идентификатор, который связан с объектом во все время его существования и не меняется при изменении состояния объекта.

Каждый объект имеет состояние и поведение. Состояние объекта - набор значений его атрибутов. Поведение объекта - набор методов (программный код), оперирующих над состоянием объекта. Значение атрибута объекта - это тоже некоторый объект или множество объектов. Состояние и поведение обь-

екта инкапсулированы в объекте; взаимодействие между объектами производится на основе передачи сообщений и выполнении соответствующих методов.

Множество объектов с одним и тем же набором атрибутов и методов образует класс объектов. Объект должен принадлежать только одному классу (если не учитывать возможности наследования, см. следующий абзац). Допускается наличие примитивных предопределенных классов, объекты - экземпляры которых не имеют атрибутов: целые, строки и т.д. Класс, объекты которого могут служить значениями атрибута объектов другого класса, называется доменом этого атрибута.

Допускается порождение нового класса на основе уже существующего класса - наследование. В этом случае новый класс, называемый подклассом существующего класса (суперкласса) наследует все атрибуты и методы суперкласса. В подклассе, кроме того, могут быть определены дополнительные атрибуты и методы. Различаются случаи простого и множественного наследования. В первом случае подкласс может определяться только на основе одного суперкласса, во втором случае суперклассов может быть несколько. При поддержании множественного наследования классы связаны в ориентированный граф с корнем, называемый решеткой классов. Объект подкласса считается принадлежащим любому суперклассу этого класса.

Одной из более поздних идей объектно-ориентированного подхода является идея возможного переопределения атрибутов и методов суперкласса в подклассе (перегрузки методов). Эта возможность увеличивает гибкость, но порождает дополнительную проблему: при компиляции объектно-ориентированной программы могут быть неизвестны структура и программный код методов объекта, хотя его класс (в общем случае - суперкласс) известен. Для разрешения этой проблемы применяется так называемый метод позднего связывания, означающий, по сути дела, интерпретационный режим выполнения программы с распознаванием деталей реализации объекта во время выполнения посылки сообщения к нему.

Метод - программный код, привязанный к конкретному классу и применимый к объектам этого класса. Определение метода в ООБД производится в два этапа. Сначала объявляется сигнатура метода, т.е. его имя, класс, типы или классы аргументов и тип или класс результата. Методы могут быть публичными (доступными из объектов других классов) или приватными (доступными только внутри данного класса). На втором этапе определяется реализация класса на одном из языков программирования ООБД.

В модели ООБД поддерживается множественное наследование классов на основе отношения суперттип/подтип. В подклассе допускается добавление и/или переопределение атрибутов и методов. Возможные при множественном насле-

довании двусмысленности (по именованию атрибутов и методов) разрешаются либо путем переименования, либо путем явного указания источника наследования. Объект подкласса является объектом каждого суперкласса, на основе которого порожден данный подкласс.

Специфической особенностью модели ООБД является возможность объявления дополнительных "исключительных" атрибутов и методов для именованных объектов. Это означает, что конкретный именованный объект-представитель класса может обладать типом, являющимся подтипом типа класса. Конечно, с такими атрибутами не работают стандартные методы класса, но специально для именованного объекта могут быть определены дополнительные (или переопределены стандартные) методы, для которых дополнительные атрибуты уже доступны. Подчеркивается, что дополнительные атрибуты и методы привязываются не к конкретному объекту, а к имени, за которым в разные моменты времени могут стоять вообще говоря разные объекты.

Оборудование и материалы

Для выполнения практической работы необходим персональный компьютер с установленной СУБД Oracle10g/11g ExpressEdition и пакетом Microsoft Office. Инсталляция пакета представлена в приложении.

Указания по технике безопасности

Необходимо выполнять требования по технике безопасности, предусмотренные для компьютерного класса

1. Не включать компьютер без разрешения администратора класса и преподавателя.
2. Запрещается самостоятельно вскрывать корпус ПК.
3. Не касаться одновременно экрана монитора и клавиатуры (возможен разряд повышенного электростатического потенциала).
4. Не загромождать верхние панели устройств бумагами и посторонними устройствами.
5. Не прикасаться одновременно к металлическим частям ПК и устройствам, имеющим естественное заземление.
6. Запрещается производить отключение питания во время выполнения активной задачи.
7. Не допускать попадание влаги на поверхность системного блока, монитора, клавиатуру, принтеры и другие устройства.

Порядок выполнения работы

Разработайте базу данных, согласно задания преподавателя.

Выполните действия с БД, описанные выше.

Содержание отчета и его форма

Отчет должен содержать

- название практической работы;
- цель работы;
- порядок выполнения работы;
- результаты выполнения;
- вывод о проделанной работе.

Контрольные вопросы и защита работы

Контрольные вопросы

7. Каковы групповые обозначения для опций аудита?
8. Включение и выключение аудита базы данных?
9. Очистка аудиторских записей из аудиторского журнала?
10. Аудит с помощью триггеров базы данных?
11. С какой целью применяется параметр INCREMENT BY?
12. Что такое синонимы?

Защита работы

Защита работы заключается в:

- выполнении заданий;
- ответах на контрольные вопросы;
- предоставлении отчета.

УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Перечень основной и дополнительной литературы, необходимой для освоения дисциплины (модуля)

Перечень основной литературы:

1. Култыгин, О. П. Администрирование баз данных. СУБД MS SQL Server : Учебное пособие / Култыгин О. П. - Москва : Московский финансово-промышленный университет «Синергия», 2012. - 232 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-4257-0026-1. Режим доступа: <http://www.iprbookshop.ru/17009.html>.

Перечень дополнительной литературы:

1. Федотов, Е. А. Администрирование программных и информационных систем : Учебное пособие / Федотов Е. А. - Белгород : Белгородский государственный технологический университет им. В.Г. Шухова, ЭБС АСВ, 2012. - 136 с. - Книга находится в базовой версии ЭБС IPRbooks. Режим доступа: <http://www.iprbookshop.ru/27280.html>
2. Михайлов, В. В. Администрирование информационных систем Электронный ресурс : Учебное пособие / В. В. Михайлов. - Белгород : Белгородский

государственный технологический университет им. В.Г. Шухова, ЭБС АСВ, 2017. - 112 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 2227-8397. Режим доступа: <http://www.iprbookshop.ru/80407.html>

3. Братченко, Н. Ю. (Северо-Кавказский федеральный университет). Базы данных: учебник. Направление подготовки 230100 – Информатика и вычислительная техника / Н. Ю. Братченко ; Сев.-Кав. федер. ун-т. – Ставрополь : СКФУ, 2021. – 160 с. [электронный вариант]

Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине (модулю)

Братченко Н. Ю. Администрирование баз данных: методические рекомендации для самостоятельной работы студентов. – Ставрополь: Изд-во СКФУ, 2022. – 20 с.

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

www.biblioclub.ru - Электронная библиотечная система «Университетская библиотека онлайн»

www.eLibrary.ru - Научная электронная библиотека

www.iprbookshop.ru - Электронно-библиотечная система IPRbooks