МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский государственный гуманитарный университет»
(ФГБОУ ВО «РГГУ»)

ИНСТИТУТ ИНФОРМАЦИОННЫХ НАУК И ТЕХНОЛОГИЙ БЕЗОПАСНОСТИ
Факультет информационных систем и безопасности
Кафедра информационных технологий и систем

СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

01.03.04 Прикладная математика
Код и наименование направления подготовки/специальности

Математика информационных сред
Наименование направленности (профиля)/ специализации

Уровень высшего образования: *Бакалавриат*

Форма обучения: *очная*

РПД адаптирована для лиц
с ограниченными возможностями
здоровья и инвалидов

Москва 2023

СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

Рабочая программа дисциплины

Составитель: к.с.-х.н., доц. Н.Ш.Шукенбаева

УТВЕРЖДЕНО
Протокол заседания кафедры
информационных технологий и систем
№ 8 от 15.04.2023

ОГЛАВЛЕНИЕ

1. Пояснительная записка

1.1. Цель и задачи дисциплины

Цель дисциплины: обеспечить студентов теоретическими знаниями о современных профессиональных системах управления базами данных, познакомить с историей развития и типологией СУБД, моделями архитектур, а также дать практические навыки по разработке ИС под современными СУБД.

Задачи:

- рассмотреть этапы разработки ИС и их характеристики,
- изучить архитектуры реализации корпоративных информационных систем,
- познакомится с различными реляционными СУБД промышленного класса и сравнить их характеристики,
- изучить принципы архитектуры СУБД, встроенный и динамический SQL,
- получить практические навыки разработки, управления и администрирования проектов БД.

1.2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Компетенция (код и наименование)	**Индикаторы компетенций** (код и наименование)	**Результаты обучения**
ПК-1. Способен проводить систематизацию, алгоритмизацию конкретных информационных потоков по месту научных исследований, производственной деятельности	ПК-1.2. Выделяет динамические, статистические структуры для представления их математическими моделями.	Знать: типологию и методологию проектирования многопользовательских баз данных, выделения динамических и статистических структур для представления их различными моделями. Уметь: конфигурировать и администрировать СУБД для работы в многопользовательском режиме транзакционной обработки, используя SQL разрабатывать проекты БД, обеспечивающие автоматизированную обработку информации в корпоративных ИС, выделять динамические и статистические структуры для представления их математическими моделями баз данных. Владеть: навыками работы в групповых проектах, навыками, связанными с разработкой технологической документации, сопровождающей процесс создания баз данных.

1.3. Место дисциплины в структуре образовательной программы

Дисциплина «СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ» относится к части, формируемой участниками образовательных отношений блока дисциплин учебного плана.

Для освоения дисциплины необходимы знания, умения и владения, сформированные в ходе изучения дисциплины: «Базы данных».

В результате освоения дисциплины формируются знания, умения и владения необходимые для изучения следующих дисциплин: «Управление информационными системами».

2. Структура дисциплины

Общая трудоёмкость дисциплины составляет 3 з.е., 108 академических часа (ов).

Структура дисциплины для очной формы обучения

Объем дисциплины в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Семестр	Тип учебных занятий	Количество часов
6	Лекции	**18**
6	Практические занятия	**24**
	Всего:	**42**

Объем дисциплины в форме самостоятельной работы обучающихся составляет 66 академических часа(ов).

3. Содержание дисциплины

№	Наименование раздела дисциплины	Содержание
1	Введение. Архитектуры реализации корпоративных ИС	Основные отличительные особенности корпоративных СУБД. Этапы разработки АИС. Основные классы задач, решаемых с использованием баз данных. Применение профессиональных СУБД для построения корпоративных информационных систем. Архитектура клиент-сервер. Режимы работы с базой данных. Распределение процесса выполнения функций приложения между клиентом и сервером: Presentation logic, Business logic, database logic, database Manager System, служебные функции. Структура типового интерактивного приложения, работающего с БД. Двухуровневые модели (модель файлового сервера, модель удаленного доступа к данным, модель сервера баз данных). Модель сервера приложений.
2	Архитектура и основные объекты схемы сервера	Конфигурации. Типы пользователей. Обзор архитектуры– физический (экземпляр, файлы журналирования операций, управляющие файлы) и логический (табличные пространства, схема БД) уровни. Логические структуры для контроля над дисковым пространством (блоки данных, экстенты, сегменты). Структуры памяти (SGA, PGA), буфер, кэш. Процессы (нити): пользовательские (клиентские) и процессы (серверные и фоновые). Типы данных. Создание объектов с помощью команд SQL (таблицы, индексы, представления, последовательности, определенные пользователем типы, синонимы, кластеры, ограничения, табличные пространства, триггеры, процедуры и функции).
3	Основные конструкции языка SQL	Введение в SQL. Типы данных SQL. Объявление программных данных и преобразование типов данных. Типы данных для больших объектов и предопределенные объектные типы. SQL и управление транзакциями. Выборка данных с помощью курсоров. Динамический SQL и динамический SQL. Хранимые процедуры и функции.

		Пакеты. Триггеры. Управление приложениями SQL. Механизм выполнения программ SQL сервером. Подключение внешних библиотек.
4	Физические модели данных. Транзакции. Безопасность данных.	Организация внешней памяти. Хранение отноше-ний. Индексы. Журнальная информация. Служебная информация. Классификация файлов и файловых структур. Файлы прямого и последовательного доступа. Доступ к файлу с использованием методов хеширования. По-нятие коллизий. Методы разрешения коллизий. Поня-тие индексного файла. Плотный индекс. Понятие ин-дексного файла. Неплотный индекс. Индексы в виде B-деревьев. Кластеризованный индекс. Некластеризованный индекс. Представление связи 1:M в структурах хранения. Однонаправленные указатели. Двунаправленные указатели. Алгоритм поиска записи в связях 1:M. Инвертированные списки Страничная организация данных. Экстенты. Типы экстентов. Типы страниц. Параметры страниц. Страницы размещения. Страницы данных. Строки данных. Индексные страницы. Текстовые страницы. Структура файлов в Ms SQL Server. Разделяемая память. Понятие операции транзакции и основные харак-теристики. Основные свойства транзакций – атомар-ность, согласованность, изолированность и долговеч-ность. Фиксация и откат транзакций. Транзакции и целостность баз данных. Изолированность транзакций. Сериализация транзакций. Методы сериализации транзакций Назначение и использование журнала транзакций. Индивидуальные откаты транзакций, восстановление БД после мягкого и жесткого сбоев. Параллельное выполнение транзакций. Захваты и блокировки. Гранулированные и предикатные синхронизационные захваты.
5	Администрирование СУБД	Администраторы и конечные пользователи: права и функции DBA. Пример систем управления БД. Основные функции и компоненты. Особенности работы в гетерогенной среде. Стандартизация программных интерфейсов.
6	Сравнительные характеристики SQL СУБД. Современные направления развития СУБД	Сравнительные характеристики Oracle 10i, MS SQL Server, IBM DB2, Informix, Postgress на каждом из основных этапов функционирования: конфигурирование системы, мониторинг, настройка, обработка запросов, разработка серверных и клиентских модулей. Принимаемые во внимание факторы: максимальное число пользователей одновременно обращающихся к базе, характеристики клиентского ПО; аппаратные компоненты сервера, серверная операционная система. Интеллектуальный анализ данных. Многомерное представление данных. Взаимодействие WEB-технологий и баз данных. Системы информационной поддержки принятия решений. Корпоративные порталы.

4. Образовательные технологии

№ п/	Наименование раздела	Виды учебных занятий	Образовательные технологии

п			
1.	Введение. Архитектуры реализации корпоративных ИС	Лекция. Практическая работа Самостоятельная работа	Вводная лекция с использованием видеоматериалов Занятия с использованием специализированного ПО Изучение материала по теме
2.	Архитектура и основные объекты схемы сервера	Лекция. Практическая работа Самостоятельная работа	Лекция-визуализация с применением проектора Занятия с использованием специализированного ПО Подготовка к занятию с использованием ЭБС
3.	Основные конструкции языка SQL	Лекция. Практическая работа Самостоятельная работа	Лекция-визуализация с применением проектора Занятия с использованием специализированного ПО Подготовка к занятию с использованием ЭБС
4.	Физические модели данных. Транзакции. Безопасность данных.	Лекция. Практическая работа. Самостоятельная работа	Проблемная лекция Занятия с использованием специализированного ПО Подготовка к занятию с использованием ЭБС
5.	Администрирование СУБД	Лекция. Практическая работа. Самостоятельная работа	Проблемная лекция Занятия с использованием специализированного ПО Подготовка к занятию с использованием ЭБС
6.	Сравнительные характеристики SQL СУБД. Современные направления развития СУБД	Лекция. Практическая работа. Самостоятельная работа	Проблемная лекция Занятия с использованием специализированного ПО Подготовка к занятию с использованием ЭБС

В период временного приостановления посещения обучающимися помещений и территории РГГУ для организации учебного процесса с применением электронного обучения и дистанционных образовательных технологий могут быть использованы следующие образовательные технологии:

– видео-лекции**;**

– онлайн-лекции в режиме реального времени;

– электронные учебники, учебные пособия, научные издания в электронном виде и доступ к иным электронным образовательным ресурсам;

– системы для электронного тестирования;

– консультации с использованием телекоммуникационных средств.

5. Оценка планируемых результатов обучения

5.1 Система оценивания

Форма контроля	Макс. количество баллов	
	За одну работу	Всего
Текущий контроль: - защита практических работ	10 баллов	60 баллов
Промежуточная аттестация - зачет (Ответы на вопросы)		40 баллов
Итого за семестр		100 баллов

Полученный совокупный результат конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

100-балльная шкала	Традиционная шкала		Шкала ECTS
95 – 100	отлично	зачтено	A
83 – 94			B
68 – 82	хорошо		C
56 – 67	удовлетворительно		D
50 – 55			E
20 – 49	неудовлетворительно	не зачтено	FX
0 – 19			F

5.2 Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	**Оценка по дисциплине**	**Критерии оценки результатов обучения по дисциплине**
100-83/ A,B	зачтено	Выставляется обучающемуся, если он глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации. Обучающийся исчерпывающе и логически стройно излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения. Свободно ориентируется в учебной и профессиональной литературе. Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».
82-68/ C	зачтено	Выставляется обучающемуся, если он знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей. Обучающийся правильно применяет теоретические положения при решении практических задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами. Достаточно хорошо ориентируется в учебной и профессиональной литературе. Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».

Баллы/ Шкала ECTS	**Оценка по дисциплине**	**Критерии оценки результатов обучения по дисциплине**
67-50/ D,E	зачтено	Выставляется обучающемуся, если он знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации. Обучающийся испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами. Демонстрирует достаточный уровень знания учебной литературы по дисциплине. Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».
49-0/ F,FX	не зачтено	Выставляется обучающемуся, если он не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации. Обучающийся испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами. Демонстрирует фрагментарные знания учебной литературы по дисциплине. Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции на уровне «достаточный», закреплённые за дисциплиной, не сформированы.

При оценивании *защиты практической работы* учитывается:

- полнота выполненной работы (задание выполнено не полностью и/или допущены две и более ошибки или три и более неточности) – 1-4 балла;

- обоснованность содержания и выводов работы (задание выполнено полностью, но обоснование содержания и выводов недостаточны, но рассуждения верны) – 5-8 баллов;

- работа выполнена полностью, в рассуждениях и обосновании нет пробелов или ошибок, возможна одна неточность -9-10 баллов.

Затем баллы конвертируются в количество баллов в семестре согласно таблице.

Промежуточная аттестация (зачет)

При проведении *промежуточной аттестации* студент должен ответить на 2 вопроса теоретического характера.

При оценивании ответа на вопрос теоретического характера учитывается:

- теоретическое содержание не освоено, знание материала носит фрагментарный характер, наличие грубых ошибок в ответе (1-10 баллов);

- теоретическое содержание освоено частично, допущено не более двух-трех недочетов (11-20 баллов);

- теоретическое содержание освоено почти полностью, допущено не более одного-двух недочетов, но обучающийся смог бы их исправить самостоятельно (21-30 баллов);

- теоретическое содержание освоено полностью, ответ построен по собственному плану (31-40 баллов).

5.3 Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

Вопросы к текущей аттестации

В чем суть использования механизма транзакций?
изменения в базу данных вносятся каждой операцией
изменения в базу данных вносятся только после выполнения всей последовательности операций , входящих в транзакцию, если они выполнены успешно
изменения в базу данных вносятся только администратором базы данных
изменения в базу данных вносятся только при определенных условиях
При каких условиях система меняет данные в базе данных?
по завершению транзакции и по оператору commit
по завершению каждой операции транзакции
по указанию администратора
по оператору модификации данных
Для чего ведется журнал транзакций?
для анализа действий с базой данных
для использования прикладными программами
для проверки правильности данных
для восстановления базы данных
Индексирование в базах данных используется для …
уменьшения объема базы данных
защиты базы данных от несанкционированного доступа
ускорения операций поиска и сортировки
корректности выполнения запросов к базе данных
Выберите Неверную рекомендацию при создании объекта "индекс".
При создании индекса необходимо проанализировать, какие запросы направляются к данной таблице. Если это в основном запросы SELECT, то нужно создать индексы для столбцов, по которым производится выборка данных для ускорения выполнения запросов.
Если таблица постоянно обновляется (выполняются операторы INSERT, UPDATE, DELETE), то количество индексов необходимо свести к минимуму, т.к. значительная часть ресурсов сервера будет затрачиваться на обновление индексов.
Так как индексы позволяют ускорить операции вставки данных в таблицу, необходимо создавать индексы для часто изменяемых таблиц.
Выберите объекты, которые непосредственно не относятся к объектам базы данных во внешней памяти базы данных.
Индексы
Журнальная информация
Служебная информация
Строки отношений
Модули прикладных программ
Какое высказывание несправедливо по отношению термина «Транзакция»
Каждая транзакция начинается при целостном состоянии БД и должна оставить это состояние целостными после своего завершения
Для поддержания ограничений целостности не допускается их нарушение внутри транзакции.
Транзакции разных пользователей не должны мешать друг другу. Иначе говоря, транзакция должна работать только с непротиворечивыми данными, не имея доступа к промежуточным результатам.
Результаты работы выполненной транзакции должны сохраниться в базе данных, даже если по завершении транзакции произойдет сбой системы.
Транзакции разных пользователей не должны мешать друг другу.

Какого уровня изолированности транзакций не существует
Отсутствие записи «грязных данных»
Отсутствие потерянных изменений.
Отсутствие чтения «грязных данных».
Отсутствие неповторяющихся чтений.
Для чего используется обьект Trigger?
Для задания значения столбца по умолчанию (если оно явно не определено при вставке строки)
Для работы с выборкой значений из базы данных по запросу как с виртуальной клиентской таблицей
Для ограничения диапазона значений, хранимых в столбце таблицы
Для работы «запись за записью» с выборками за базы данных
Для контроля правильности ввода и модификаций данных в БД
Для чего используется обьект View
Для задания значения столбца по умолчанию (если оно явно не определено при вставке строки)
Для работы с выборкой значений из базы данных по запросу как с виртуальной клиентской таблицей
Для ограничения диапазона значений, хранимых в столбце таблицы
Для работы «запись за записью» с выборками за базы данных
… - это специальный вид процедуры, который выполняется сервером баз данных
Хранимая процедура
Триггер
Индекс
Ограничение
… - это специальный вид процедуры, который нельзя вызвать из клиентского приложения и связан с некоторым событием, возникающем при редактировании объекта БД
Хранимая процедура
Триггер
Индекс
Представление
… - это специальная таблица, представляющая список номеров записей и указывающая в каком порядке их нужно представлять
Хранимая процедура
Триггер
Индекс
Представление
… - это виртуальная таблица, предоставляющая данные из одной или нескольких реальных таблиц
Хранимая процедура
Триггер
Индекс
Представление
Что такое SQL?
Язык разметки базы данных
Структурированный язык запросов
Язык программирования низкого уровня
Язык программирования высокого уровня
SQL предназначен для работы с
Иерархическими моделями данных
Сетевыми моделями данных
Реляционными моделями данных
Как удалить из таблицы data все записи, поле info_id которых не заполнено?

DROP TABLE dat
DELETE FROM data
DELETE FROM data WHERE info_id LIKE»
DELETE FROM data WHERE info_id IS NULL
DELETE FROM data WHERE info_id NOT NULL

Какая из представленных команд служит для изменения структуры существующей таблицы?

ALTER TABLE
SELECT
CREATE TABLE
DROP TABLE
CREATE VIEW
UPDATE
INSERT INTO
DELETE FROM

Какая из представленных команд служит для модификации данных в таблицах?

ALTER TABLE
SELECT
CREATE TABLE
DROP TABLE
CREATE VIEW
INSERT INTO

Какие из представленных команд служат для удаления таблицы?

ALTER TABLE
SELECT
CREATE TABLE
DROP TABLE
CREATE VIEW
UPDATE
INSERT INTO
DELETE FROM

Для чего используется команда GRANT?

Для назначения пользователям прав доступа к объектам базы данных
Для получения эксклюзивного права доступа к базе данных
Для запрещения доступа пользователей к объектам базы данных
Для отклонения прав роли пользователя на доступ к объектам базы данных

Для чего используется команда REVOKE?

Для назначения пользователям прав доступа к объектам базы данных
Для получения эксклюзивного права доступа к базе данных
Для запрещения доступа пользователей к объектам базы данных
Для отклонения прав пользователя на доступ к объектам базы данных

Какой из перечисленных операторов относится к языку управления данными (DCL)?

Update - изменение значений в полях таблицы
Grant – создание в системе безопасности разрешающей записи для пользователя
Select –выборка строк, удовлетворяющих заданным условиям
Create – создание таблицы, индекса
Drop – удаление таблицы
Alter – изменение структуры таблицы
Insert – вставка строк в таблицу
Delete – удаление строк из таблицы

Какой из перечисленных операторов относится к языку определения данных (DDL)?

Update - изменение значений в полях таблицы

Grant – создание в системе безопасности разрешающей записи для пользователя
Select –выборка строк, удовлетворяющих заданным условиям
Create – создание таблицы, индекса и т.д.
Insert – вставка строк в таблицу
Delete – удаление строк из таблицы
Deny - создание в системе безопасности запрещающей записи для пользователя

Какой из перечисленных операторов относится к языку манипулирования данными (DML)?

Update - изменение значений в полях таблицы
Grant – создание в системе безопасности разрешающей записи для пользователя
Create – создание таблицы, индекса
Drop – удаление таблицы
Alter – изменение структуры таблицы
Insert – вставка строк в таблицу

Что делает оператор INSERT?

вставляет строку с заданными значениями элементов в таблицу
вставляет столбец с заданными значениями элементов в таблицу
вставляет строку с заданными значениями элементов и значениями по умолчанию в таблицу
вставляет столбец с заданными значениями элементов и значениями по умолчанию в таблицу

В каком предложении оператора INSERT указываются вставляемые в таблицу значения?

INSERT
VALUES
FROM
WHERE

Какие служебные слова могут использоваться в операторе INSERT?

FROM
WHERE
VALUES
GROUP BY

Какие служебные слова могут использоваться в операторе DELETE?

WHERE
INTO
VALUES
GROUP BY

Какие СУБД относятся к клиент-серверным?

ACCESS
ORACLE
DB2
BORLAND PARADOX

Какие СУБД относятся к файл-серверным?

ACCESS
MS SQL-сервер
ORACLE
INTERBASE

Какие СУБД используются для организации баз данных в крупных организациях (относятся к промышленным)?

ACCESS
MS SQL SERVER
FoxPro
PARADOX

Вопросы к зачету

1. Перечислить персональные и профессиональные СУБД. Какие возможности обеспечивает использование профессиональных СУБД.
2. Этапы разработки АИС.
3. Архитектура «клиент-сервер». Основной принцип технологии «клиент-сервер»
4. Понятие презентационной логики, бизнес-логики и логики обработки данных
5. Модель удаленного управления данными FS (модель файлового сервера). Достоинства и недостатки
6. Модель удаленного доступа к данным (RDA). Достоинства и недостатки
7. Модель (активного) сервера баз данных. Достоинства и недостатки
8. Модель сервера приложений. Достоинства и недостатки
9. Архитектура сервера баз данных. Централизованная архитектура и модель 1:1. Недостатки. Архитектура выделенного сервера (многопотоковая серверная архитектура). Недостатки
10. Архитектура виртуального сервера. Недостатки. Многонитиевая мультисерверная архитектура
11. Типы параллелизма.
12. Требования к безопасности реляционных СУБД
13. Пользователи СУБД. Объекты доступа. Привилегии. Операторы SQL для установки привилегий. Примеры.
14. Основные способы определения групп пользователей. Роль. Механизм ролей. Создание роли. Пример
15. Понятие транзакции. Свойства транзакции. Варианты завершения транзакции.
16. Расширенная модель транзакции. Смесь транзакций. График запуска набора транзакций.
17. Проблемы параллельной работы транзакций. Проблема потери результатов обновления. Неаккуратное считывание
18. Проблемы параллельной работы транзакций. Проблема несовместимого анализа.
19. Конкурирующие транзакции. Конфликты между транзакциями. Графики запуска набора транзакции. Способы разрешения конкуренций между транзакциями.
20. Типы блокировок. Протокол доступа к данным. Уровни блокирования.
21. Управление транзакциями. Команды управления транзакциями.
22. Явные транзакции. Вложенные транзакции
23. Управление блокировками
24. Тупиковые блокировки
25. Уровни изоляции
26. Режимы SQL. Процесс выполнения операторов sql
27. Классификация файлов и файловых структур
28. Файлы прямого и последовательного доступа
29. Доступ к файлу с использованием методов хеширования
30. Понятие коллизий. Методы разрешения коллизий
31. Понятие индексного файла. Плотный индекс. Неплотный индекс.
32. Индексы в виде B-деревьев.
33. Кластеризованный индекс
34. Некластеризованный индекс
35. Представление связи 1:M в структурах хранения. Однонаправленные указатели. Пример. Двунаправленные указатели. Пример. Алгоритм поиска записи в связях 1:M
36. Инвертированные списки
37. Страничная организация данных. Экстенты. Типы экстентов. Типы страниц. Параметры страниц
38. Страницы размещения. Страницы данных. Строки данных. Индексные страницы. Текстовые страницы.

39. Структура файлов в Ms SQL Server. Разделяемая память
40. Сравнительные характеристики SQL СУБД: Oracle, MS SQL Server, IBM DB2, Informix
41. Конфигурации Типы пользователей.
42. Архитектура (физический и логический уровень)
43. Табличные пространства. Сегменты, экстенты и блоки данных.
44. Экземпляр. SGA, PGA
45. Процессы. 7 основных фоновых процессов
46. Функции СУБД. Инсталляция
47. Программные средства Oracle. Основные функции управления. Создание пользователя и настройка прав доступа.
48. Объекты БД
49. Создание таблиц. Типы данных
50. Создание индексов
51. Создание представлений
52. Создание последовательностей
53. Определенные пользователем типы данных. Создание синонимов
54. Создание ограничений
55. Создание табличных пространств
56. Основные понятия и конструкции SQL. Архитектура SQL
57. Поддерживаемый набор символов SQL. Арифметические операторы и операторы отношения
58. Структура программы и переменные SQL
59. Условные операторы
60. Циклы
61. Курсоры
62. Хранимые процедуры
63. Функции
64. Триггеры

6. Учебно-методическое и информационное обеспечение дисциплины

6.1 Список источников и литературы

Литература

Основная

1. Базы данных: учебник / Л.И. Шустова, О.В. Тараканов. - М.: НИЦ ИНФРА-М, 2016. - 336 с.: 60х90 1/16. - (Высшее образование: Бакалавриат) (Переплёт 7БЦ) ISBN 978-5-16-010485-0, 500 экз http://znanium.com/catalog.php?bookinfo=491069
2. Проектирование информационных систем и баз данных/СтасышинВ.М. - Новосиб.: НГТУ, 2012. - 100 с.: ISBN 978-5-7782-2121-5 http://znanium.com/catalog.php?bookinfo=548234

Дополнительная

1. Агальцов В.П. Базы данных. В 2-х кн. Книга 2. Распределенные и удаленные базы данных : учебник / В.П. Агальцов. — М. : ИД «ФОРУМ» : ИНФРА-М, 2017. — 271 с.: ил. — (Высшее образование). http://znanium.com/catalog.php?bookinfo=652917
2. Базы данных. Практическое применение СУБД SQL и NoSOL-типа для применения проектирования информационных систем: Учебное пособие / Мартишин С.А., Симонов В.Л., Храпченко М.В. - М.:ИД ФОРУМ, НИЦ ИНФРА-М, 2017. - 368 с.: 60х90 1/16. - (Высшее образование) (Переплёт 7БЦ) ISBN 978-5-8199-0660-6 http://znanium.com/catalog.php?bookinfo=556449

3. Методы, модели, средства хранения и обработки данных: учебник / Э.Г. Дадян, Ю.А. Зеленков. — М.: Вузовский учебник: ИНФРА-М, 2017. — 168 с. http://znanium.com/catalog.php?bookinfo=543943
4. Култыгин, О. П. Администрирование баз данных. СУБД MS SQL Server [Электронный ресурс] : учеб. пособие / О. П. Култыгин. - М.: МФПА, 2012. - 232 с. - (Университетская серия). - ISBN 978-5-4257-0026-1 . http://znanium.com/catalog.php?bookinfo=451114

6.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет».

1. Электронно-библиотечная система «Знаниум» Режим доступа: http://znanium.com
2. Национальный открытый университет «ИНТУИТ». Режим доступа: https://www.intuit.ru/
3. Сайт Microsoft Режим доступа: https://msdn.microsoft.com/ru-ru/library/
4. Научная библиотека РГГУ Режим доступа: http://liber.rsuh.ru/
5. «CITFORUM»: Аналитическая информация в сфере IT. Режим доступа: http://citforum.ru/

Национальная электронная библиотека (НЭБ) www.rusneb.ru
ELibrary.ru Научная электронная библиотека www.elibrary.ru
Cambridge University Press
SAGE Journals

6.3 Профессиональные базы данных и информационно-справочные системы

Доступ к профессиональным базам данных: https://liber.rsuh.ru/ru/bases

Информационные справочные системы:
1. Консультант Плюс
2. Гарант

7. Материально-техническое обеспечение дисциплины

Для обеспечения дисциплины используется материально-техническая база образовательного учреждения:
- *для лекций:* учебные аудитории, оснащённые доской, компьютером или ноутбуком, проектором (стационарным или переносным) для демонстрации учебных материалов.

Состав программного обеспечения:
1. Windows
2. Microsoft Office
3. Kaspersky Endpoint Security

- *для практических занятий:* компьютерный класс или лаборатория, оснащённые доской, компьютером или ноутбуком для преподавателя, компьютерами для обучающихся, проектором (стационарным или переносным) для демонстрации учебных материалов.

Состав программного обеспечения:
1. Windows
2. Microsoft Office
3. Microsoft SQL Server 2008
4. Mozilla Firefox
5. Kaspersky Endpoint Security

8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- для слепых и слабовидящих: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением или могут быть заменены устным ответом; обеспечивается индивидуальное равномерное освещение не менее 300 люкс; для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств; письменные задания оформляются увеличенным шрифтом; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.
- для глухих и слабослышащих: лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования; письменные задания выполняются на компьютере в письменной форме; экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.
- для лиц с нарушениями опорно-двигательного аппарата: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- для слепых и слабовидящих: в печатной форме увеличенным шрифтом, в форме электронного документа, в форме аудиофайла.
- для глухих и слабослышащих: в печатной форме, в форме электронного документа.
- для обучающихся с нарушениями опорно-двигательного аппарата: в печатной форме, в форме электронного документа, в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

- для слепых и слабовидящих: устройством для сканирования и чтения с камерой SARA CE; дисплеем Брайля PAC Mate 20; принтером Брайля EmBraille ViewPlus;
- для глухих и слабослышащих: автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих; акустический усилитель и колонки;

- для обучающихся с нарушениями опорно-двигательного аппарата: передвижными, регулируемыми эргономическими партами СИ-1; компьютерной техникой со специальным программным обеспечением.

9. Методические материалы

9.1 Планы практических занятий

Практическая работа № 1 Язык Transact-SQL. Компоненты SQL

Цель: Рассмотреть основные объекты и базовые операторы языка Transact-SQL

- Основные объекты SQL
- Типы данных
- Функции языка SQL
- Скалярные операторы
- Значения NULL

Упражнения

Упражнение 1

Какая разница между числовыми типами данных INT, SMALLINT и TINYINT?

Упражнение 2

Какая разница между типами данных CHAR и VARCHAR? Когда следует использовать первый, а не второй, и наоборот?

Упражнение 3

Как настроить столбец типа данных DATE для ввода значений в формате 'гггг/мм/дд'?

В следующих двух упражнениях используйте инструкцию SELECT в окне редактора запросов средства Management Studio для вывода результатов всех системных функций и глобальных переменных. (Например, инструкция SELECT host_id () выводит идентификационный номер текущего хоста.)

Упражнение 4

Используя системные функции, узнайте идентификационный номер базы данных test, которую создали на прошлой лабораторной работе.

Упражнение 5

Используя системные переменные, узнайте текущую версию программного обеспечения системы базы данных и используемый в программном обеспечении язык.

Упражнение 6

Используя битовые операторы &, | и ^, выполните следующие операции над битовыми строками:

(11100101) & (01010111)

(10011011) | (11001001)

(10110111) ^ (10110001)

Упражнение 7

Какими будут результаты следующих выражений? (Выражение A — числовое, а B — логическое.)

A + NULL

NULL = NULL

B OR NULL

B AND NULL

Упражнение 8

В каких случаях можно использовать как одинарные, так и двойные кавычки для определения строковых и временных констант?

Упражнение 9

Что такое идентификатор с ограничителями и когда требуется использовать идентификаторы этого типа?

Практическая работа № 2 Язык описания данных

Цель:

Рассмотреть все инструкции Transact-SQL, связанные с языком описания данных

- ♦ Создание объектов баз данных
- ♦ Модифицирование объектов баз данных
- ♦ Удаление объектов баз данных

Упражнения

Упражнение 1

Используя инструкцию CREATE DATABASE, создайте новую базу данных test_db, задав явные спецификации для файлов базы данных и журнала транзакций. Файл базы данных с логическим именем test_db_dat сохраняется в физическом файле C:\tmp\test_db.mdf, его начальный размер — 5 Мбайт, автоувеличение по 8%, максимальный размер не ограничен. Файл журнала транзакций с логическим именем test_db_log сохраняется в физическом файле C:\tmp\test_db_log.ldf, его начальный размер — 2 Мбайта, автоувеличение по 500 Кбайт, максимальный размер 10 Мбайт.

Упражнение 2

Используя инструкцию ALTER DATABASE, добавьте новый файл журнала в базу данных test_db. Файл сохраняется в физическом файле C:\tmp\emp_log.ldf, его начальный размер — 2 Мбайта, автоувеличение по 2 Мбайта, максимальный размер не ограничен.

Упражнение 3

Используя инструкцию ALTER DATABASE, измените начальный размер файла базы данных test_db на 10 Мбайт.

Упражнение 4

В примере 4 для некоторых столбцов четырех созданных таблиц запрещены значения NULL. Для каких из этих столбцов это определение является обязательным, а для каких нет?

Упражнение 5

Почему в примере 4 тип данных для столбцов dept_no и project_no определен как CHAR, а не как один из целочисленных типов?

Упражнение 6

Создайте таблицы customers и orders, содержащие перечисленные в следующей таблице столбцы. Не объявляйте соответствующие первичный и внешние ключи.

customers orders

customerid char(5) not null orderid integer not null

companyname varchar(40) not null customerid char(5) not null

contactname char(30) null orderdate date null

address varchar(60) null shippeddate date null

city char(15) null freight money null

phone char(24) null shipname varchar(40) null

fax char(24) null shipaddress varchar(60) null

quantity integer null

Упражнение 7

Используя инструкцию ALTER TABLE, добавьте в таблицу orders новый столбец shipregion. Столбец должен иметь целочисленный тип данных и разрешать значения NULL.

Упражнение 8

Используя инструкцию ALTER TABLE, измените тип данных столбца shipregion с целочисленного на буквенно-цифровой длиной 8 символов. Столбец может содержать значения NULL.

Упражнение 9

Удалите созданный ранее столбец shipregion.

Упражнение 10

Дайте точное описание происходящему при удалении таблицы с помощью инструкции DROP TABLE.

Упражнение 11

Создайте заново таблицы customers и orders, усовершенствовав их определение всеми ограничениями первичных и внешних ключей.

Упражнение 12

Используя среду SQL Server Management Studio, попробуйте вставить следующую новую строку в таблицу orders:

(10, 'ord0l', getdate(), getdate(), 100.0, 'Windstar', 'Ocean', 1).

Почему система отказывается вставлять эту строку в таблицу?

Упражнение 13

Используя инструкцию ALTER TABLE, определите значение по умолчанию столбца orderdate таблицы orders в виде текущей даты и времени системы.

Упражнение 14

Используя инструкцию ALTER TABLE, создайте ограничение для обеспечения целостности, ограничивающее допустимые значения столбца quantity таблицы orders диапазоном значений от 1 до 30.

Упражнение 15

Отобразите все ограничения для обеспечения целостности таблицы orders.

Упражнение 16

Попытайтесь удалить первичный ключ таблицы customers. Почему это не удается?

Упражнение 17

Удалите ограничение для обеспечения целостности prim_empl, определенное в примере 7.

Упражнение 18

В таблице customers измените имя столбца city на town.

Практическая работа № 3 Запросы

Цель:

Рассмотреть операторы части DML языка SQL

- ♦ Инструкция SELECT. Ее предложения и функции
- ♦ Подзапросы
- ♦ Временные таблицы
- ♦ Операторы соединения
- ♦ Связанные подзапросы
- ♦ Табличные выражения

Упражнения

Упражнение 1

Выполните выборку всех строк из таблицы works_on.

Упражнение 2

Выполните выборку табельных номеров всех сотрудников с должностью клерк (clerk).

Упражнение 3

Выполните выборку табельных номеров всех сотрудников, которые работают над проектом p2 и чей табельный номер меньше, чем 10 000. Решите эту задачу, используя два различных, но эквивалентных запроса с инструкцией select.

Упражнение 4

Выполните выборку табельных номеров всех сотрудников, которые не приступили к работе над проектом в 2007 г.

Упражнение 5

Выполните выборку табельных номеров всех сотрудников проекта p1 с ведущими должностями (т. е. аналитик analyst и менеджер manager).

Упражнение 6

Выполните выборку всех сотрудников проекта p2, чья должность еще не определена.

Упражнение 7

Выполните выборку табельных номеров и фамилий сотрудников, чьи имена содержат две буквы "t".

Упражнение 8

Выполните выборку табельных номеров и имен всех сотрудников, у которых вторая буква фамилии "o" или "a" (буквы английские) и последние буквы фамилии "es".

Упражнение 9

Выполните выборку табельных номеров сотрудников, чьи отделы расположены в Сиэтле (Seattle).

Упражнение 10

Выполните выборку фамилий и имен сотрудников, которые приступили к работе над проектами 4 января 2007 г.

Упражнение 11

Сгруппируйте все отделы по их местонахождению.

Упражнение 12

Объясните разницу между предложениями DISTINCT и GROUP BY.

Упражнение 13

Как предложение GROUP BY обрабатывает значения NULL? Подобна ли эта обработка обычной обработке этих значений?

Упражнение 14

Объясните разницу между агрегатными функциями COUNT(*) и COUNT(column) .

Упражнение 15

Выполните выборку наибольшего табельного номера сотрудника.

Упражнение 16

Выполните выборку должностей, занимаемых больше, чем двумя сотрудниками. Упражнение 17

Выполните выборку табельных номеров сотрудников, которые или имеют должность клерк clerk, или работают в отделе d3.

Упражнение 18

Объясните, почему следующий запрос неправильный.

```
SELECT project name FROM project WHERE project_no =
(SELECT project_no FROM works_on WHERE Job = 'Clerk')
```

Исправьте синтаксис запроса.

Упражнение 19

Каково практическое применение временных таблиц?

Упражнение 20

Объясните разницу между глобальными и локальными временными таблицами.

ПРИМЕЧАНИЕ

В решениях всех последующих упражнений по операции соединения используйте явный синтаксис.

Упражнение 21

Создайте следующие соединения таблиц project и works_on, выполнив:

- естественное соединение;
- декартово произведение.

Упражнение 22

Сколько условий соединения необходимо для соединения в запросе n таблиц?

Упражнение 23

Выполните выборку табельного номера сотрудника и должности для всех сотрудников, работающих над проектом Gemini.

Упражнение 24

Выполните выборку имен и фамилий всех сотрудников, работающих в отделе Research или Accounting.

Упражнение 25

Выполните выборку всех дат начала работы для всех клерков (clerk), работающих в отделе d1.

Упражнение 26

Выполните выборку всех проектов, над которыми работают двое или больше сотрудников с должностью клерк (clerk).

Упражнение 27

Выполните выборку имен и фамилий сотрудников, которые имеют должность менеджер (manager) и работают над проектом Mercury.

Упражнение 28

Выполните выборку имен и фамилий всех сотрудников, которые начали работать над проектом одновременно, по крайней мере, еще с одним другим сотрудником.

Упражнение 29

Выполните выборку табельных номеров сотрудников, которые живут в том же городе, где находится их отдел. (Используйте расширенную таблицу enployee_ enh базы данных sample.)

Упражнение 30

Выполните выборку табельных номеров всех сотрудников, работающих в отделе маркетинга marketing. Создайте два равнозначных запроса, используя:

- оператор соединения;
- связанный подзапрос.

Практическая работа № 4 Хранимые процедуры и определяемые пользователем функции

Цель:

Познакомиться с пакетами и подпрограммами SQL

- ♦ Процедурные расширения
- ♦ Хранимые процедуры
- ♦ Определяемые пользователем функции

Упражнения

Упражнение 1

Создайте пакет для вставки 3000 строк в таблицу employee. Значения столбца emp_no должны быть однозначными в диапазоне от 1 до 3000. Всем ячейкам столбцов emp_lname, emp_fname и dept_no присваиваются значения "Jane", "Smith" и "d1" соответственно.

Упражнение 2

Измените пакет из упражнения 1 таким образом, чтобы генерировать случайные значения для столбца emp_no, используя функцию RAND. (Подсказка: для получения случайных значений используйте системные функции DATEPART и GETDATE.)

Практическая работа № 5 Индексы. Представления

Цель:

Рассмотретьиндексы и их роль в оптимизации времени выполнения запросов, а также объекты базы данных, которые называются представлениями

- ♦ Общие сведения
- ♦ Язык Transact-SQL и индексы

- Рекомендации по созданию и использованию индексов
- Специальные типы индексов
- Инструкции языка DDL и представления
- Инструкции языка DML и представления
- Индексированные представления

Упражнения

Упражнение 1

Создайте некластеризованный индекс для столбца enter_date таблицы works_on, с заполнением пространства каждой страницы листьев индекса на 70%.

Упражнение 2

Создайте однозначный составной индекс для столбцов emp_lname и emp_fname таблицы employee. Будет ли какая-либо разница, если изменить порядок столбцов в составном индексе?

Упражнение 3

Как удалить индекс, который был неявно создан для первичного ключа таблицы?

Упражнение 4

Изложите достоинства и недостатки индексов.

В следующих четырех упражнениях создайте индексы, которые повысят производительность запросов. Предполагается, что все таблицы базы данных sample, используемые в этих упражнениях, имеют большое количество строк.

Упражнение 5

SELECT emp no, emp fname, emp lname FROM employee WHERE emp_lname = 'Smith'

Упражнение 6

SELECT emp no, emp fname, emp lname FROM employee WHERE emp lname = 'Hansel' AND emp fname = 'Elke'

Упражнение 7

SELECT job FROM works on, employee
WHERE employee.emp no = works on.emp no

Упражнение 8

SELECT emp lname, emp fname FROM employee, department
WHERE employee.dept no = department.dept no AND dept name = 'Research'

Упражнение 9

Создайте представление, содержащее данные для всех сотрудников, которые работают в отделе d1.

Упражнение 10

Создайте представление для таблицы project, чтобы сотрудники могли просматривать все данные этой таблицы, за исключением столбца budget.

Упражнение 11

Создайте представление, содержащее имя и фамилии всех сотрудников, которые начали работать над своими проектами во второй половине 2007 г.

Упражнение 12

Выполните упражнение 11, переименовав исходные столбцы emp_fname и
emp_lname в first и last соответственно.

Упражнение 13

Используя представление в упражнении 9, отобразите полные сведения для всех сотрудников, чья фамилия начинается с буквы "M".

Упражнение 14

Создайте представление, содержащее полные сведения для всех проектов, над которыми работает сотрудник по фамилии Smith.

Упражнение 15

Модифицируйте условие представления в упражнении 9 (используя инструкцию ALTER VIEW), чтобы представление возвращало данные всех сотрудников, которые работают в отделе d1 или d2 или в обоих отделах.

Упражнение 16

Удалите представление, созданное в упражнении 11. Что произойдет вследствие этого с представлением, созданным в упражнении 12?

Упражнение 17

Используя представление из упражнения 10, вставьте данные для нового проекта, номер которого p2, а имя - Moon.

Упражнение 18

Создайте представление (с предложением with check option), содержащее имена и фамилии всех сотрудников, чей табельный номер меньше, чем 10000. Используя это представление, вставьте данные для нового сотрудника по фамилии Kohn, табельный номер которого 22123 и который работает в отделе d3.

Упражнение 19

Выполните упражнение 18, не используя предложение WITH CHECK OPTION, и определите разницу относительно вставки новых данных.

Упражнение 20

Создайте представление (с предложением WITH CHECK OPTION), содержащее полные сведения из таблицы works_on для всех сотрудников, которые начали работать над своими проектами в 2007 и 2008 гг. Затем для сотрудника с табельным номером 29346 измените дату начала работы над проектом на 06/01/2006.

Упражнение 21

Выполните упражнение 20, не используя предложение WITH CHECK OPTION, и определите разницу касательно модифицирования данных.

Практическая работа № 6 Транзакции

Цель:

Рассмотреть модели одновременного конкурентного доступа. Реализация трнзакций. Изучить понятие блокировок и уровней изоляций

Теоретическая часть

Модели одновременного конкурентного доступа

Компонент Database Engine поддерживает две разные модели одновременного конкурентного доступа:

- ♦ пессимистический одновременный конкурентный доступ;
- ♦ оптимистический одновременный конкурентный доступ.

В модели пессимистического одновременного конкурентного доступа для предотвращения одновременного доступа к данным, которые используются другим процессом, применяются блокировки. Иными словами, система баз данных, использующая модель пессимистического одновременного конкурентного доступа, предполагает, что между двумя или большим количеством процессов в любое время может возникнуть конфликт и поэтому блокирует ресурсы (строку, страницу, таблицу), как только они потребуются в течение периода транзакции. Как мы увидим в разд. "Блокировка" этой работы, модель пессимистического одновременного конкурентного доступа устанавливает блокировку с обеспечением разделяемого доступа, иначе немонопольную блокировку (shared lock) на считываемые данные, чтобы никакой другой процесс не мог изменить эти данные. Кроме этого, механизм пессимистического одновременного конкурентного доступа устанавливает монопольную блокировку (exclusive lock) на изменяемые данные, чтобы никакой другой процесс не мог их считывать или модифицировать.

Работа оптимистического одновременного конкурентного доступа основана на предположении маловероятности изменения данных одной транзакцией одновременно с другой. Компонент Database Engine применяет оптимистический одновременный конкурентный доступ, при котором сохраняются старые версии строк, и любой процесс при чтении данных использует ту версию строки, которая была активной, когда он начал чтение. Поэтому процесс

может модифицировать данные без каких-либо ограничений, поскольку все другие процессы, которые считывают эти же данные, используют свою собственную сохраненную версию. Конфликтная ситуация возможна только при попытке двух операций записи использовать одни и те же данные. В таком случае система выдает ошибку, которая обрабатывается клиентским приложением.

ПРИМЕЧАНИЕ

Понятие оптимистического одновременного конкурентного доступа обычно определяется в более широком смысле. Работа управления оптимистического одновременного конкурентного доступа основана на предположении маловероятности конфликтов между несколькими пользователями, поэтому разрешается исполнение транзакций без установки блокировок. Только когда пользователь пытается изменить данные, выполняется проверка ресурсов, чтобы определить наличие конфликтов. Если таковые возникли, то приложение требуется перезапустить.

Транзакции

Транзакция задает последовательность инструкций языка Transact-SQL, применяемую программистами базы данных для объединения в один пакет операций чтения и записи для того, чтобы система базы данных могла обеспечить согласованность данных. Существует два типа транзакций.

- Неявная транзакция — задает любую отдельную инструкцию insert, update или delete как единицу транзакции.
- Явная транзакция — обычно это группа инструкций языка Transact-SQL, начало и конец которой обозначаются такими инструкциями, как begin transaction,

COMMIT и ROLLBACK.

Свойства транзакций

Транзакции обладают следующими свойствами, которые все вместе обозначаются сокращением ACID (Atomicity, Consistency, Isolation, Durability):

- атомарность (Atomicity);
- согласованность (Consistency);
- изолированность (Isolation);
- долговечность (Durability).

Повторите по лекционному материалу, что означают эти свойства.

Инструкции Transact-SQL и транзакции

Для работы с транзакциями язык Transact-SQL предоставляет следующие шесть инструкций:

- BEGIN TRANSACTION;
- BEGIN DISTRIBUTED TRANSACTION;
- COMMIT [WORK];
- ROLLBACK [WORK];
- SAVE TRANSACTION;
- SET IMPLICIT_TRANSACTIONS.

Инструкция BEGIN TRANSACTION запускает транзакцию. Синтаксис этой инструкции выглядит следующим образом:

BEGIN TRANSACTION [{transaction_name | @trans_var}

[WITH MARK ['description ']]]

В параметре transaction_name указывается имя транзакции, которое можно использовать только в самой внешней паре вложенных инструкций BEGIN TRANSACTION/COMMIT или BEGIN TRANSACTION/ROLLBACK. В параметре @trans_var указывается имя определяемой пользователем переменной, содержащей действительное имя транзакции. Параметр with mark указывает, что транзакция должна быть отмечена в журнале. Аргумент description — это строка, описывающая эту отметку. В случае использования параметра with mark требуется указать имя транзакции.

Инструкция BEGIN DISTRIBUTED TRANSACTION запускает распределенную транзакцию, которая управляется Microsoft Distributed Transaction Coordinator (MS DTC — координатором распределенных транзакций Microsoft). Распределенная транзакция — это транзакция, которая используется на нескольких базах данных и на нескольких серверах. Поэтому для таких транзакций требуется координатор для согласования выполнения инструкций на всех вовлеченных серверах. Координатором распределенной транзакции является сервер, запустивший инструкцию BEGIN DISTRIBUTED TRANSACTION, и поэтому он и управляет выполнением распределенной транзакции.

Инструкция COMMIT WORK успешно завершает транзакцию, запущенную инструкцией BEGIN TRANSACTION. Это означает, что все выполненные транзакцией изменения фиксируются и сохраняются на диск. Инструкция COMMIT WORK является стандартной формой этой инструкции. Использовать предложение WORK не обязательно.

В противоположность инструкции COMMIT WORK, инструкция ROOLBACK WORK сообщает о неуспешном выполнении транзакции. Программисты используют эту инструкцию, когда они полагают, что база данных может оказаться в несогласованном состоянии. В таком случае выполняется откат всех произведенных инструкциями транзакции изменений. Инструкция ROLLBACK WORK является стандартной формой этой инструкции. Использовать предложение WORK не обязательно.

Инструкция save transation устанавливает точку сохранения внутри транзакции. Точка сохранения (savepoint) определяет заданную точку в транзакции, так что все последующие изменения данных могут быть отменены без отмены всей транзакции. (Для отмены всей транзакции применяется инструкция rollback.)

Как вы уже знаете, каждая инструкция Transact-SQL всегда явно или неявно принадлежит к транзакции. Для удовлетворения требований стандарта SQL компонент Database Engine предоставляет поддержку неявных транзакций. Когда сеанс работает в режиме неявных транзакций, выполняемые инструкции неявно выдают инструкции BEGIN TRANSACTION. Это означает, что для того чтобы начать неявную транзакцию, пользователю или разработчику не требуется ничего делать. Но каждую неявную транзакцию нужно или явно зафиксировать или явно отменить, используя инструкции COMMIT или ROLLBACK соответственно. Если транзакцию явно не зафиксировать, то все изменения, выполненные в ней, откатываются при отключении пользователя.

Для разрешения неявных транзакций параметру SET IMPLICIT_TRANSACTIONS необходимо присвоить значение ON. Это установит режим неявных транзакций для текущего сеанса. Когда для соединения установлен режим неявных транзакций и соединение в данный момент не используется в транзакции, выполнение любой из следующих инструкций запускает транзакцию:

ALTER TABLE FETCH REVOKE
CREATE TABLE GRANT SELECT
DELETE INSERT TRUNCATE TABLE
DROPTABLE OPEN UPDATE

Иными словами, если имеется последовательность инструкций из предыдущего списка, то каждая из этих инструкций будет представлять транзакцию.

Начало явной транзакции помечается инструкцией BEGIN TRANSACTION, а окончание — инструкцией COMMIT или ROLLBACK. Явные транзакции можно вкладывать друг в друга. В таком случае, каждая пара инструкций BEGIN TRANSACTION/COMMIT или BEGIN TRANSACTION/ROLLBACK используется внутри каждой такой пары или большего количества вложенных транзакций. (Вложенные транзакции обычно используются в хранимых процедурах, которые сами содержат транзакции и вызываются внутри другой транзакции.) Глобальная переменная @@trancount содержит число активных транзакций для текущего пользователя.

Инструкции BEGIN TRANSACTION, COMMIT и ROLLBACK могут использоваться с именем заданной транзакции. (Именованная инструкция ROLLBACK соответствует или именованной транзакции, или инструкции SAVE TRANSACTION с таким же именем.) Именованную транзакцию можно применять только в самой внешней паре вложенных инструкций BEGIN TRANSACTON/COMMIT или BEGIN TRANSACTION/ROLLBACK.

Журнал транзакций

Реляционные системы баз данных создают запись для каждого изменения, которые они выполняют в базе данных в процессе транзакции. Это требуется на случай ошибки при выполнении транзакции. В такой ситуации все выполненные инструкции транзакции необходимо отменить, осуществив для них откат. Как только система обнаруживает ошибку, она использует сохраненные записи, чтобы возвратить базу данных в согласованное состояние, в котором она была до начала выполнения транзакции.

Компонент Database Engine сохраняет все эти записи, в особенности значения до и после транзакции, в одном или более файлов, которые называются журналами транзакций (transaction log). Для каждой базы данных ведется ее собственный журнал транзакций. Таким образом, если возникает необходимость отмены одной или нескольких операций изменения данных в таблицах текущей базы данных, компонент Database Engine использует записи в журнале транзакций, чтобы восстановить значения столбцов таблиц, которые существовали до начала транзакции.

Журнал транзакций применяется для отката или восстановления транзакции. Если в процессе выполнения транзакции еще до ее завершения возникает ошибка, то система использует все существующие в журнале транзакций исходные значения записей (которые называются исходными образами записей (before image)), чтобы выполнить откат всех изменений, выполненных после начала транзакции. Процесс, в котором исходные образы записей из журнала транзакций используются для отката всех изменений, называется операцией отмены записей (undo activity).

В журналах транзакций также сохраняются преобразованные образы записей (after image). Преобразованные образы — это модифицированные значения, которые применяются для отмены отката всех изменений, выполненных после старта транзакции. Этот процесс называется операцией повторного выполнения действий (redo activity) и применяется при восстановлении базы данных.

Каждой записи в журнале транзакций присваивается однозначный идентификатор, называемый порядковым номером журнала транзакции (log sequence number или LSN). Все записи журнала, являющиеся частью определенной транзакции, связаны друг с другом, чтобы можно было найти все части этой транзакции для операции отмены или повтора.

Блокировка

Одновременный конкурентный доступ может вызывать разные отрицательные эффекты, например чтение несуществующих данных или потерю модифицированных данных. Рассмотрим следующий практический пример, иллюстрирующий один из этих отрицательных эффектов, называемый грязным чтением. Пользователь U из отдела кадров получает извещение, что сотрудник Jim Smith поменял место жительства. Он вносит соответствующее изменение в базу данных для данного сотрудника, но при просмотре другой информации об этом сотруднике он понимает, что изменил адрес не того человека. (В компании работают два сотрудника по имени Jim Smith.) К счастью, приложение позволяет отменить это изменение одним нажатием кнопки. Он нажимает эту кнопку, уверенный в том, что данные после отмены операции изменения адреса уже не содержат никакой ошибки.

В то же самое время пользователь U2 в отделе проектирования обращается к данным второго сотрудника с именем Jim Smith, чтобы отправить ему домой последнюю техническую документацию, поскольку этот служащий редко бывает в офисе. Однако пользователь U2 обратился к базе данных после того, как адрес этого второго сотрудника с именем Jim Smith был ошибочно изменен, но до того, как он был исправлен. В результате письмо отправляется не тому адресату.

Чтобы предотвратить подобные проблемы в модели пессимистического одновременного конкурентного доступа, каждая система управления базами данных должна обладать механизмом для управления одновременным доступом к данным всеми пользователями. Для обеспечения согласованности данных в случае одновременного обращения к данным несколькими пользователями компонент Database Engine, подобно всем СУБД, применяет блокировки. Каждая прикладная программа блокирует требуемые ей данные, что гарантирует, что никакая другая программа не сможет модифицировать эти данные. Когда другая прикладная программа пытается получить доступ к заблокированным данным для их модификации, то система или завершает эту попытку ошибкой, или заставляет программу ожидать снятия блокировки.

Блокировка имеет несколько разных свойств:

- ♦ длительность блокировки;
- ♦ режим блокировки;
- ♦ гранулярность блокировки.

Длительность блокировки — это период времени, в течение которого ресурс удерживает определенную блокировку. Длительность блокировки зависит, среди прочего, от режима блокировки и выбора уровня изоляции.

Режимы блокировки и уровень гранулярности блокировки рассматриваются в следующих двух разделах.

ПРИМЕЧАНИЕ

Последующее обсуждение относится к модели пессимистического одновременного конкурентного доступа. Модель оптимистического одновременного конкурентного доступа основана на управлении версиями строк и рассматривается в конце этой работы.

Режимы блокировки

Режимы блокировки определяют разные типы блокировок. Выбор определенного режима блокировки зависит от типа ресурса, который требуется заблокировать. Для блокировок ресурсов уровня строки и страницы применяются следующие три типа блокировок:

- ♦ разделяемая (shared, S);
- ♦ монопольная (exclusive, X);
- ♦ обновления (update, U).

Разделяемая блокировка (shared lock) резервирует ресурс (страницу или строку) только для чтения. Другие процессы не могут изменять заблокированный таким образом ресурс, но, с другой стороны, несколько процессов могут одновременно накладывать разделяемую блокировку на один и тот же ресурс. Иными словами, чтение ресурса с разделяемой блокировкой могут одновременно выполнять несколько процессов.

Монопольная блокировка (exclusive lock) резервирует страницу или строку для монопольного использования одной транзакции. Блокировка этого типа применяется инструкциями DML (insert, update и delete), которые модифицируют ресурс. Монопольную блокировку нельзя установить, если на ресурс уже установлена разделяемая или монопольная блокировка другим процессом, т. е. на ресурс может быть установлена только одна монопольная блокировка. На ресурс (страницу или строку) с установленной монопольной блокировкой нельзя установить никакую другую блокировку.

ПРИМЕЧАНИЕ

Система баз данных автоматически выбирает соответствующий режим блокировки, в зависимости от типа операции (чтение или запись).

Блокировка обновления (update lock) может быть установлена на ресурс только при отсутствии на нем другой блокировки обновления или монопольной блокировки. С другой стороны, этот тип блокировки можно устанавливать на объекты с установленной разделяемой блокировкой. В таком случае блокировка обновления накладывает на объект другую разделяемую блокировку. Если транзакция, которая модифицирует объект, подтверждается, и у объекта нет никаких других блокировок, блокировка обновления преобразовывается в монопольную блокировку. У объекта может быть только одна блокировка обновления.

ПРИМЕЧАНИЕ

Блокировка обновления применяется для предотвращения определенных распространенных типов взаимоблокировок. (Взаимоблокировки рассматриваются в конце этого раздела.)

Возможность совмещения разных типов блокировок приводится в табл. 1.

Таблица 1. Матрица совместимости разных типов блокировок

	Разделяемая	Обновления	Монопольная
Разделяемая	Да	Да	Нет
Обновления	Да	Нет	Нет
Монопольная	Нет	Нет	Нет

Таблица 1 интерпретируется следующим образом: предположим транзакция Т имеет блокировку, указанную в заголовке соответствующей строки таблицы, а транзакция Т2 запрашивает блокировку, указанную в соответствующем заголовке столбца таблицы. Значение "Да" в ячейке на пресечении строки и столбца означает, что транзакция Т2 может иметь запрашиваемый тип блокировки, а значение "Нет", что не может.

ПРИМЕЧАНИЕ

Компонент Database Engine также поддерживает и другие типы блокировок, такие как кратковременные блокировки (latch lock) и взаимоблокировки (spin lock).

На уровне таблицы существует пять разных типов блокировок:

- разделяемая (shared, S);
- монопольная (exclusive, X);
- разделяемая с намерением (intent shared, IS);
- монопольная с намерением (intent exclusive, IX);
- разделяемая с монопольным намерением (shared with intent exclusive, SIX).

Разделяемые и монопольные типы блокировок для таблицы соответствуют одноименным блокировкам для строк и страниц. Обычно блокировка с намерением (intent lock) означает, что транзакция намеревается блокировать следующий нижележащий в иерархии объектов базы данных ресурс. Таким образом, блокировка с намерением помещаются на уровне иерархии объектов, который выше того объекта, который этот процесс намеревается заблокировать. Это является действенным способом узнать, возможна ли подобная блокировка, а также устанавливается запрет другим процессам блокировать более высокий уровень, прежде чем процесс может установить требуемую ему блокировку.

Возможность совмещения разных типов блокировок на уровне таблиц базы данных приведена в табл. 2. Эта таблица интерпретируется точно таким же образом, как и табл. 1.

Таблица 2. Возможность совмещения разных типов блокировок на уровне таблиц базы данных

	S	X	IS	SIX	IX
S	Да	Нет	Да	Нет	Нет
X	Нет	Нет	Нет	Нет	Нет
IS	Да	Нет	Да	Да	Да
SIX	Нет	Нет	Да	Нет	Нет
IX	Нет	Нет	Да	Нет	Да

S — разделяемая, X — монопольная, IS — разделяемая с намерением, SIX — монопольная с намерением, IX — разделяемая с монопольным намерением блокировки.

Гранулярность блокировки

Гранулярность блокировки определяет, какой ресурс блокируется в одной попытке блокировки. Компонент Database Engine может блокировать следующие ресурсы:

- строки;
- страницы;
- индексный ключ или диапазон индексных ключей;

- ♦ таблицы;
- ♦ экстент;
- ♦ саму базу данных.

Строка является наименьшим ресурсом, который можно заблокировать. Блокировка уровня строки также включает как строки данных, так и элементы индексов. Блокировка на уровне строки означает, что блокируется только строка, к которой обращается приложение. Поэтому все другие строки данной таблицы остаются свободными и их могут использовать другие приложения. Компонент Database Engine также может заблокировать страницу, на которой находится подлежащая блокировке строка.

Блокироваться также могут единицы дискового пространства, которые называются экстентами и имеют размер 64 Кбайт. Экстенты блокируются автоматически, и когда растет таблица или индекс, то для них требуется выделять дополнительное дисковое пространство.

Гранулярность блокировки оказывает влияние на одновременный конкурентный доступ. В общем, чем выше уровень гранулярности, тем больше сокращается возможность совместного доступа к данным. Это означает, что блокировка уровня строк максимизирует одновременный конкурентный доступ, т.к. она блокирует всего лишь одну строку страницы, оставляя все другие строки доступными для других процессов. С другой стороны, низкий уровень блокировки увеличивает системные накладные расходы, поскольку для каждой отдельной строки требуется отдельная блокировка. Блокировка на уровне страниц и таблиц ограничивает уровень доступности данных, но также уменьшает системные накладные расходы.

Укрупнение блокировок

Если в процессе транзакции имеется большое количество блокировок одного уровня, то компонент Database Engine автоматически объединяет эти блокировки в одну уровня таблицы. Этот процесс преобразования большого числа блокировок уровня строки, страницы или индекса в одну блокировку уровня таблицы называется укрупнением блокировок (lock escalation). Порогом укрупнения называется граница, на которой система баз данных применяет укрупнение блокировок. Пороги укрупнения устанавливаются динамически системой и не требуют настройки. (В настоящее время пороговым значением укрупнения блокировок является 5000 блокировок.)

Основной проблемой, касающейся укрупнения блокировок, является то обстоятельство, что решение, когда осуществлять укрупнение, принимает сервер баз данных, и это решение может не быть оптимальным для приложений, имеющих различные требования. Механизм укрупнения блокировок можно модифицировать с помощью инструкции ALTER TABLE. Эта инструкция поддерживает параметр TABLE и имеет следующий синтаксис:

SET (LOCK_ESCALATION = {TABLE | AUTO | DISABLE})

Параметр TABLE является значением по умолчанию и задает укрупнение блокировок на уровне таблиц. Параметр AUTO позволяет компоненту Database Engine самому выбирать уровень гранулярности, который соответствует схеме таблицы. Наконец, параметр DISABLE отключает укрупнение блокировок в большинстве случаев. (В некоторых случаях компоненту Database Engine требуется наложить блокировку на уровне таблиц, чтобы предохранить целостность данных.)

Взаимоблокировки

Взаимоблокировка (deadlock) — это особая проблема одновременного конкурентного доступа, в которой две транзакции блокируют друг друга. В частности, первая транзакция блокирует объект базы данных, доступ к которому хочет получить другая транзакция, и наоборот. (В общем, взаимоблокировка может быть вызвана несколькими транзакциями, которые создают цикл зависимостей.) В примере 5 показана взаимоблокировка двумя транзакциями.

Практическая часть

Понятие транзакции лучше всего объяснить на примере. В базе данных sample сотруднику Ann Jones требуется присвоить новый табельный номер. Этот номер нужно

одновременно изменить в двух разных таблицах. В частности, требуется одновременно изменить строку в таблице employee и соответствующие строки в таблице works_on. Если обновить данные только в одной из этих таблиц, данные базы данных sample будут несогласованны, поскольку значения первичного ключа в таблице employee и соответствующие значения внешнего ключа в таблице works_on не будут совпадать. Реализация этой транзакции посредством инструкций языка Transact-SQL показана в примере 1.

Пример 1. Реализация транзакции

```
USE sample;
BEGIN TRANSACTION /* Начало транзакции */
UPDATE employee
SET emp_no = 39831 WHERE emp_no = 10102
IF (@@error <> 0)
ROLLBACK /*Откат транзакции */
UPDATE works_on
SET emp_no = 39831 WHERE emp_no = 10102
IF (@@error <> 0)
ROLLBACK
COMMIT /*Завершение транзакции */
```

Согласованность данных, обрабатываемых в примере 1, можно обеспечить лишь в том случае, если выполнены обе инструкции update либо обе не выполнены. Успех выполнения каждой инструкции update проверяется посредством глобальной переменной @@error. В случае ошибки этой переменной присваивается отрицательное значение и выполняется откат всех выполненных на данный момент инструкций транзакции.

Пример 2. Создание и использование точки сохранения

```
BEGIN TRANSACTION;
INSERT INTO department (depy_no, dept_name) VALUES (’d4’, ’Sales’);
SAVE TRANSACTION a;
INSERT INTO department (depy_no, dept_name) VALUES (’d5’, ’Research’);
SAVE TRANCACTION b;
INSERT INTO department (depy_no, dept_name) VALUES (’d6’, ’Management’);
ROLLBACK TRANCACTION b;
INSERT INTO department (depy_no, dept_name) VALUES ('d7 'Support');
ROLLBACK TRANSACTION a;
COMMIT TRANSACTION;
```

Добейтесь, чтобы этот запрос выполнялся безошибочно. Если возникают конфликты, исправьте их. Какой будет результат? Сколько строк вставилось в таблицу?

Пример 3. Отмена возможности укрупнения блокировок для таблицы

```
USE sample;
ALTER TABLE employee SET (LOCK_E S CALATION = DISABLE);
```

Пример 4. Взаимоблокировка двух процессов

```
USE sample;
BEGIN TRANSACTION
UPDATE works_on
SET job = 'Manager'
WHERE emp_no = 18316 AND project_no = 'p2'
WAITFOR DELAY '00:00:10'
UPDATE employee
SET emp_lname = ’Green’
WHERE emp_no = 9031
COMMIT
BEGIN TRANSACTION
UPDATE employee
```

```
SET dept_no = ’d 4’
WHERE emp_no = 9031
WAITFOR DELAY ’00:00:10’
DELETE FROM works_on
WHERE emp_no = 18316 AND project_no = ’p2’
COMMIT
```

Если обе транзакции в примере будут выполняться в одно и то же время, то возникнет взаимоблокировка и система возвратит следующее сообщение об ошибке:

Как можно видеть по результатам выполнения примера 4, система баз данных обрабатывает взаимоблокировку, выбирая одну из транзакций (на самом деле, транзакцию, которая замыкает цикл в запросах блокировки) в качестве "жертвы" и выполняя ее откат. После этого выполняется другая транзакция. На уровне прикладной программы взаимоблокировку можно обрабатывать посредством реализации условной инструкции, которая выполняет проверку на возврат номера ошибки (1205), а затем снова выполняет инструкцию, для которой был выполнен откат.

Вы можете повлиять на то, какая транзакция будет выбрана системой в качестве "жертвы" взаимоблокировки, присвоив в инструкции set параметру deadlock_ priority один из 21 (от -10 до 10) разных уровней приоритета взаимоблокировки. Константа low соответствует значению -5, normal (значение по умолчанию) — значению 0, а константа high — значению 5. Сеанс "жертва" выбирается в соответствии с приоритетом взаимоблокировки сеанса.

Упражнения

Упражнение 1

Какая цель использования транзакций?

Упражнение 2

В чем заключается разница между локальной и распределенной транзакцией?

Упражнение 3

В чем заключается разница между явным и неявным режимом транзакции?

Упражнение 4

Какие типы блокировок совместимы с монопольной блокировкой?

Упражнение 5

Как можно проверить, было ли успешным выполнение каждой инструкции Transact-SQL?

Упражнение 6

В каких случаях следует использовать инструкцию SAVE TRANSACTION?

Упражнение 7

В чем заключается разница между блокировкой уровня строк и блокировкой уровня страниц?

Упражнение 8

Может ли пользователь явно влиять на реализацию блокировок системой?

Упражнение 9

В чем состоит разница между основными типами блокировки (разделяемой и монопольной) и блокировкой намерения?

Упражнение 10

Что означает понятие укрупнения блокировки?

Упражнение 11

Изложите разницу между уровнями изоляции READ UNCOMMITTED и SERIALIZABLE.

Упражнение 12

Что такое взаимоблокировка?

Упражнение 13

Какой процесс в качестве "жертвы" в случае взаимоблокировки? Может ли пользователь повлиять на решение системы в этом вопросе?

Приложение 1. Аннотация
рабочей программы дисциплины

АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Дисциплина «СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ» реализуется на факультете Информационных систем и безопасности кафедрой Информационных технологий и систем.

Цель дисциплины: обеспечить студентов теоретическими знаниями о современных профессиональных системах управления базами данных, познакомить с историей развития и типологией СУБД, моделями архитектур, а также дать практические навыки по разработке ИС под современными СУБД.

Задачи:

- рассмотреть этапы разработки ИС и их характеристики,
- изучить архитектуры реализации корпоративных информационных систем,
- познакомится с различными реляционными СУБД промышленного класса и сравнить их характеристики,
- изучить принципы архитектуры СУБД, встроенный и динамический SQL,
- получить практические навыки разработки, управления и администрирования проектов БД.

Дисциплина направлена на формирование следующих компетенций:

ПК-1. Способен проводить систематизацию, алгоритмизацию конкретных информационных потоков по месту научных исследований, производственной деятельности.

В результате освоения дисциплины обучающийся должен:

Знать: типологию и методологию проектирования многопользовательских баз данных, выделения динамических и статистических структур для представления их различными моделями.

Уметь: конфигурировать и администрировать СУБД для работы в многопользовательском режиме транзакционной обработки, используя SQL разрабатывать проекты БД, обеспечивающие автоматизированную обработку информации в корпоративных ИС, выделять динамические и статистические структуры для представления их математическими моделями баз данных.

Владеть: навыками работы в групповых проектах, навыками, связанными с разработкой технологической документации, сопровождающей процесс создания баз данных.

По дисциплине предусмотрена промежуточная аттестация в форме зачета.

Общая трудоемкость освоения дисциплины составляет 3 зачетные единицы.

Приложение 2

ЛИСТ ИЗМЕНЕНИЙ[1]

№	Текст актуализации или прилагаемый к РПД документ, содержащий изменения	Дата	№ протокола

[1] Для ОП ВО магистратуры изменения только за 2020 г.