МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

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

БАЗЫ ДАННЫХ

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

38.03.02 Менеджмент
Код и наименование направления подготовки/специальности

Менеджмент и цифровая трансформация бизнес-процессов компании
Наименование направленности (профиля)/ специализации

Уровень высшего образования: бакалавриат

Форма обучения: очно-заочная

РПД адаптирована для лиц
с ограниченными возможностями
здоровья и инвалидов

Москва 2024

БАЗЫ ДАННЫХ
Рабочая программа дисциплины
Составитель:
к.с.-х.н., доцент, заведующая кафедрой информационных технологий и систем
Н.Ш. Шукенбаева

УТВЕРЖДЕНО:

Протокол заседания кафедры информационных технологий и систем

№ 8 от 04.03.2024 года

ОГЛАВЛЕНИЕ

1 Пояснительная записка

1.1 Цель и задачи дисциплины

Цель дисциплины: профессиональная подготовка студентов, необходимая для освоения методов и технологий формирования современных баз данных, являющихся основой любой информационной системы, создаваемой в любой сфере человеческой деятельности.

Задачи:

- изучить типологии и методологии баз данных, современные модели баз данных;
- усвоить методы классификации и моделирования предметных областей, методы проектирования баз данных с помощью современных технологий;
- получить навыки работы с инструментальными средствами проектирования баз данных, использования стандартов информационных технологий, разработки технологической документации, сопровождающей процесс создания баз данных.

1.2 Формируемые компетенции, соотнесённые с планируемыми результатами обучения по дисциплине:

Компетенция	Индикаторы компетенций	Результаты обучения
ОПК-2 Способен осуществлять сбор, обработку и анализ данных, необходимых для решения поставленных управленческих задач, с использованием современного инструментария и интеллектуальных информационно-аналитических систем	ОПК-2.1. Знает источники, способы и методы аккумуляции информации, необходимой для решения поставленных управленческих задач	**Знать** модели данных; архитектуру БД; системы управления БД и информационными хранилищами; методы и средства проектирования БД. **Уметь** проводить сравнительный анализ и выбор ИКТ для решения прикладных задач и создания ИС, выбирать инструментальные средства и технологии проектирования ИС. **Владеть** навыками оценки инструментальных средств (в том числе отечественного производства) моделирования предметной области, прикладных и информационных процессов, проектирования баз данных.
	ОПК-2.2. Эффективно собирает, обрабатывает, анализирует данные и применяет их при решении управленческих задач, используя современный информационно-технологический инструментарий	**Знать** модели данных; архитектуру БД; системы управления БД и информационными хранилищами; методы и средства проектирования БД. **Уметь** проводить анализ предметной области, выявлять

		информационные потребности и разрабатывать требования к ИС; разрабатывать концептуальную модель прикладной области; проводить формализацию и реализацию решения прикладных задач. **Владеть** навыками работы с инструментальными средствами (в том числе отечественного производства) моделирования предметной области, прикладных и информационных процессов, проектирования баз данных.
ОПК-6 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК-6.1. Знает принципы работы современного инструментария сбора и анализа данных, необходимых для решения поставленных управленческих задач.	**Знать** принципы работы СУБД для решения поставленных управленческих задач. **Уметь** выполнять работы с базой данных на всех стадиях жизненного цикла проекта ИС. **Владеть** навыками использования стандартов информационных технологий.
	ОПК-2.2. Использует принципы работы информационных технологий и эффективно применяет при решении управленческих задач	**Знать** особенности администрирования БД в глобальных и локальных сетях. **Уметь** оценивать качество и затраты проекта для работы с БД. **Владеть** разработки технологической документации, сопровождающей процесс работы с ИС.

1.3 Место дисциплины в структуре основной образовательной программы

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

2 Структура дисциплины

Общая трудоёмкость дисциплины составляет 6 з.е., 216 академических часа.

Структура дисциплины для очно-заочной 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
4	Лекции	**12**
4	Семинары	**12**
5	Лекции	**12**
5	Семинары	**12**
	Всего:	**48**

Объем дисциплины в форме самостоятельной работы обучающихся составляет 168 академических часов, включая 18 часов на контроль.

3 Содержание дисциплины

№	Наименование раздела дисциплины	Содержание
1	Концепция баз данных. Система управления базами данных. Введение в банки данных. Ранние подходы к организации БД. Уровни представления моделей данных. Этапы проектирования БД	История возникновения БД. Информация, данные и информационные системы. Концепция файловой системы и концепция баз данных. Определение базы данных. Понятие системы управления базами данных (СУБД). Этапы развития СУБД и БД. Особенности каждого этапа Основные функции СУБД. Управление данными во внешней памяти. Буферизация данных в оперативной памяти. Управление транзакциями. Журнализация. Поддержка языков БД. Архитектура СУБД. Понятие банка данных (БнД). Требования к БнД. Компоненты БнД. Пользователи БнД. Администраторы БнД (АБД) и их функции. Преимущества и недостатки БнД. Классификация. Ранние подходы к организации БД. Особенности СУБД, построенных на основе инвертированных файлов. Иерархические системы. Сетевые системы. Структуры данных. Манипулирование данными. Ограничения целостности. Уровни представления моделей данных. Трехуровневая архитектура ANSI-SPARC. Три уровня абстракции: логический уровень, физический уровень, внешний уровень. Причины разделения на уровни. Логическая и физическая независимость от данных. Отображения. Схемы БД. Модели данных Этапы проектирования баз данных.
2	Инфологическое моделирование предметной области.	Предметная область. Способы описания предметной области. Требования, предъявляемые к инфологической модели. Компоненты инфологической модели. Основные принципы инфологического моделирования баз данных. Анализ и декомпозиция предметной области. Моделирование локальных представлений. Модель “сущность-связь” (ER). Сущности, атрибуты, уникальные идентификаторы, связи, сущности-связи. Отношения и мощности отношений. Конкретизации и обобщения. Агрегации.

		Построение набора концептуальных моделей локальных представлений предметной области. Синтез концептуальной схемы предметной области. Проверка концептуальной схемы на адекватность. Модификации концептуальной схемы.
3	Даталогическое проектирование	Понятие даталогического проектирования. Определение состава баз данных. Критерии оценки баз данных. Проектирование логической модели данных. Отображение концептуальной модели базы данных на выбранную модель данных. Принципы и особенности отображения на реляционную модель. Правила отображения.
4	Формализация реляционной модели данных. Нормализация схем отношений	Общие положения реляционного подхода. Базовые понятия реляционных баз данных. Тип данных. Домен. Отношение. Атрибут. Схема отношения. Кортеж. Схема базы данных. Первичный ключ. Внешний ключ. Связи. Типы связей. Фундаментальные свойства отношений. Получение реляционной схемы из ER-модели. Объекты реляционных баз данных. Общее понятие нормализации. Скорость операций манипулирования данных. Пример 1НФ. Аномалии при выполнении операций над данными. Определение функциональной зависимости. Функциональные зависимости отношений и математическое понятие функциональной зависимости 2НФ. Анализ декомпозированных отношений. Оставшиеся аномалии вставки, обновления и удаления. 3НФ. Алгоритм нормализации (приведение к 3НФ). Анализ критериев для нормализованных и ненормализованных моделей данных. Сравнение нормализованных моделей OLTP и OLAP–системы. Корректность процедуры нормализации – декомпозиция без потерь. Теорема Хеза. Нормальные формы высоких порядков. НФБК (нормальная форма Бойса–Кодда). 4НФ. 5НФ. Продолжение алгоритма нормализации (приведение к 5НФ).
5	Базовые возможности языка SQL	Язык SQL. История. Первые разработки. Стандартизация. Вопросы совместимости. Средства определения данных. Типы данных SQL. Операторы создания схемы базы данных. Создание и удаление БД. Создание, удаление и изменения структуры таблицы. Операторы создания, удаления и изменения индексов. Использование представлений. Другие возможности SQL. Средства манипулирования данными. Добавление новой записи в таблицу. Модификация записей. Удаление записей. Выборка данных. Задание условий для выборки. Агрегатные функции. Группировки. Сортировки. Вычисляемые поля. Выборка данных из нескольких таблиц. Подзапросы. Операция объединения. Средства управления доступом к данным. Определение прав доступа. Права пользователя на уровне таблицы. Отмена прав доступа.
6	Реляционная алгебра	Обзор реляционной алгебры. Замкнутость реляционной алгебры. Отношения, совместимые по типу. Оператор

		переименования атрибутов. Теоретико-множественные операторы. Объединение. Пересечение. Разность. Декартово произведение. Выборка (ограничение и селекция). Проекция. Соединение. Общая операция соединения. Тэта–соединение. Экви–соединение. Естественное соединение. Деление. Примеры использования реляционных операторов. Зависимые реляционные операторы. Оператор соединения. Оператор пересечения. Оператор деления. Примитивные реляционные операторы. Оператор декартова произведения. Оператор проекции. Оператор выборки. Операторы объединения и вычитания. Запросы, невыразимые средствами реляционной алгебры. Плохая нормализация отношений. Невыразимость транзитивного замыкания реляционными операторами. Кросс–таблицы.
7	Многопользовательский режим работы с БД. Модели "клиент сервер" в системах БД.	Архитектура «клиент-сервер». Основной принцип технологии «клиент-сервер». Понятие презентационной логики, бизнес-логики и логики обработки данных. Модель удаленного управления данными FS (модель файлового сервера). Достоинства и недостатки. Модель удаленного доступа к данным (RDA). Достоинства и недостатки. Модель (активного) сервера баз данных. Достоинства и недостатки. Модель сервера приложений. Достоинства и недостатки Архитектура сервера баз данных. Централизованная архитектура и модель 1:1. Недостатки. Архитектура выделенного сервера (многопотоковая серверная архитектура). Недостатки. Архитектура виртуального сервера. Недостатки. Многонитиевая мультисерверная архитектура
8	Физические модели данных. Транзакции. Безопасность данных.	Организация внешней памяти. Хранение отношений. Индексы. Журнальная информация. Служебная информация. Классификация файлов и файловых структур. Файлы прямого и последовательного доступа. Доступ к файлу с использованием методов хеширования. Понятие коллизий. Методы разрешения коллизий. Понятие индексного файла. Плотный индекс. Понятие индексного 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 характеристики. Основные свойства транзакций – атомарность, согласованность, изолированность и долговечность. Фиксация и откат транзакций. Транзакции и целостность баз данных. Изолированность транзакций. Сериализация тран-

		закций. 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.

4 Информационные и образовательные технологии

Для проведения учебных занятий по дисциплине используются различные образовательные технологии. Для организации учебного процесса может быть использовано электронное обучение и (или) дистанционные образовательные технологии.

5 Оценка планируемых результатов обучения

5.1 Система оценивания

Форма контроля	Макс. количество баллов	
	За одну работу	Всего
4 семестр		
Текущий контроль:		
- защита работ на семинарах	15 баллов	60 баллов
Промежуточная аттестация (зачет с оценкой)		40 баллов
Итого за 4 семестр		100 баллов
5 семестр		
Текущий контроль:		
- защита работ на семинарах	20 баллов	60 баллов
Промежуточная аттестация (экзамен)		40 баллов
Итого за 5 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
91 – 100	отлично	зачтено	A
83 – 90	хорошо		B
75 – 82			C
61 – 74	удовлетворительно		D
51 – 60			E
31 – 50	неудовлетворительно	не зачтено	FX
0 – 30			F

5.2 Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	**Оценка по дисциплине**	**Критерии оценки результатов обучения по дисциплине**
100-83/ A,B	«отлично»/ «зачтено (отлично)»	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	«хорошо»/ «зачтено (хорошо)»	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».
67-50/ D,E	«удовлетворительно»/ «зачтено (удовлетворительно»	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».

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
49-0/ F,FX	«неудовлетворительно» / 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 работы на семинаре 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 (экзамен)

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

1. Иерархическая модель БД является:
   а) структурированной моделью
   б) неструктурированной моделью
   в) частично-структурированной моделью
2. Сетевая модель БД является:

а) структурированной моделью
б) неструктурированной моделью
в) частично-структурированной моделью

3. Документальная модель БД является:
а) структурированной моделью
б) неструктурированной моделью
в) частично-структурированной моделью

4. Реляционная модель БД является:
а) теоретико-графовой моделью
б) теоретико-множественной моделью
в) объектно-ориентированной моделью

5. Иерархическая модель БД является:
а) теоретико-множественной моделью
б) теоретико-графовой моделью
в) объектно-ориентированной моделью

9. В реляционной модели БД кортежем называется:
а) атрибут домена
в) строка отношения
г) таблица данных
б) столбец отношения

10. В реляционной модели БД первичный ключ отношения однозначно определяет:
а) множество кортежей данного отношения
б) отношение
в) связь между отношениями
г) единственный кортеж отношения

16. Возможным ключом отношения называется:
а) атрибут, однозначно определяющий кортеж отношения
б) набор атрибутов, однозначно определяющий кортеж отношения
в) набор кортежей, однозначно определяющий отношение

17. В реляционной модели БД отношение – это:
а) подмножество декартова произведения множества доменов
б) полное декартово произведение множества доменов
в) единственный экземпляр подмножества декартова произведения множества доменов

28. Сущностью называется:
а) экземпляр класса объектов
б) класс однотипных объектов
в) набор атрибутов объекта

29. В модели “сущность-связь” ключевым атрибутом называется:
а) единственный атрибут, однозначно идентифицирующий экземпляр сущности
б) набор атрибутов, однозначно идентифицирующий экземпляр сущности
в) экземпляр сущности

30. Агрегированный объект является:
а) сложным объектом
б) условным объектом
в) простым объектом

31. Если степень связи между двумя сущностями 1:1, а класс принадлежности связи обязательный со стороны обеих сущностей, то каким в этом случае должно быть количество отношений при переходе от модели “сущность-связь” к реляционной модели?
а) одно отношение
б) два отношения
в) три отношения

32. Если степень связи между двумя сущностями 1:1, а класс принадлежности связи обязательный со стороны одной сущности и необязательный со стороны другой сущности, то каким в этом случае должно быть количество отношений при переходе от модели “сущность-связь” к реляционной модели?

а) одно отношение

б) два отношения

в) три отношения

33. Если степень связи между двумя сущностями 1:1, а класс принадлежности связи необязательный со стороны обеих сущностей, то каким в этом случае должно быть количество отношений при переходе от модели “сущность-связь” к реляционной модели?

а) одно отношение

б) два отношения

в) три отношения

34. Если степень связи между двумя сущностями 1:M (M:1), а класс принадлежности связи обязательный со стороны M-связной сущности, то каким в этом случае должно быть количество отношений при переходе от модели “сущность-связь” к реляционной модели?

а) одно отношение

б) два отношения

в) три отношения

35. Если степень связи между двумя сущностями 1:M (M:1), а класс принадлежности связи необязательный со стороны M-связной сущности, то каким в этом случае должно быть количество отношений при переходе от модели “сущность-связь” к реляционной модели?

а) одно отношение

б) два отношения

в) три отношения

42. Инфологическое (концептуальное) проектирование БД подразумевает:

а) формализованное описание объектов предметной области в терминах модели “сущность-связь”

б) описание схемы БД в терминах выбранной СУБД

в) выбор эффективного размещения БД на внешних носителях для обеспечения наиболее эффективной работы приложения

43. Даталогическое проектирование БД подразумевает:

а) формализованное описание объектов предметной области в терминах модели “сущность-связь”

б) описание схемы БД в терминах выбранной СУБД

в) выбор эффективного размещения БД на внешних носителях для обеспечения наиболее эффективной работы приложения

44. Физическое проектирование БД подразумевает:

а) формализованное описание объектов предметной области в терминах модели “сущность-связь”

б) описание схемы БД в терминах выбранной СУБД

в) выбор эффективного размещения БД на внешних носителях для обеспечения наиболее эффективной работы приложения

Вопросы к зачету с оценкой

1. История возникновения БД. Файловые системы. Достоинства и недостатки
2. Основные черты концепции БД
3. Этапы развития СУБД и БД. Особенности каждого этапа
4. СУБД как независимый системный компонент (схема). Определение СУБД. Типовая организация современной СУБД

5. Основные функции СУБД.
6. Понятие БД, БнД, Предметная область
7. Требования к БнД
8. Пользователи БнД
9. Состав БнД (компоненты). Краткое описание всех компонентов.
10. Информационная компонента БнД.
11. Языковые средства БнД
12. Программные, технические и организационно-методические средства БнД
13. Схема взаимодействия компонент БнД
14. Схема выполнения запроса к БнД
15. Жизненный цикл БД
16. Классификация БД
17. Классификация моделей в системах БД
18. Уровни представления данных в БД
19. Архитектура ANSI-SPARC. Три уровня абстракции. Основное назначение трехуровневой архитектуры. Типы независимостей от данных. Причины разделения на три уровня.
20. Этапы разработки и создания БД.
21. Предметная область. Понятие инфологической модели
22. Требования, предъявляемые к ИЛМ
23. Компоненты ИЛМ
24. Основные понятия ER-диаграмм
25. Виды связей. Модальность связей. Чтение связей.
26. Общие понятия даталогического проектирования
27. Подход к даталогическому проектированию
28. Особенности даталогических моделей (внутризаписная и внезаписная структура)
29. Ранние подходы к организации БД. Общие характеристики ранних систем
30. Иерархические системы. Общие понятия. Организация.
31. Иерархические системы. Манипулирование данными. Достоинства и недостатки
32. Сетевые системы. Общие понятия. Организация.
33. Сетевые системы. Манипулирование данными. Достоинства и недостатки
34. Достоинства и недостатки ранних моделей

Вопросы к экзамену

1. Общие положения реляционного подхода. Части реляционной модели (по Дейту). Достоинства и недостатки реляционной модели данных.
2. Базовые понятия реляционных БД (тип данных, домен, атрибут, отношение, схема отношения, схема БД, кортеж). Привести примеры.
3. Понятие отношения. Запись схемы отношения. Степень и мощность отношения. Привести примеры.
4. Фундаментальные свойства отношения (с пояснениями). Понятие первичного ключа. Простой и составной ключ. Суррогатный и естественный ключ. Привести примеры.
5. Связи и связанные отношения. Внешний ключ. Типы связей. Привести примеры.
6. Общее понятие нормализации. Нормальные формы. Свойства нормальных форм. Первая нормальная форма (1NF). Аномалии обновления (удаления, добавления и модификации) (показать на примере).
7. Понятие функциональной зависимости (FD). Примеры FD. Транзитивная FD. Примеры.

8. Минимально зависимые атрибуты. Минимальные FD. Пример. Вторая нормальная форма (2NF) Алгоритм перехода ко 2NF. Пример. Аномалии обновления (удаления, добавления и модификации) (показать на примере).
9. Нетранзитивные FD. Пример. Аномалии обновления (удаления, добавления и модификации) (показать на примере). Третья нормальная форма (3NF) Алгоритм перехода ко 3NF. Пример.
10. Реляционная алгебра. Замкнутость реляционной алгебры. Отношения, совместимые по типу. Показать на примерах. Оператор переименования атрибутов. Показать на примерах.
11. Объединение. Пересечение. Показать на примерах.
12. Вычитание. Декартово произведение. Показать на примерах.
13. Выборка (ограничение, селекция). Проекция. Показать на примерах.
14. Соединение. Виды соединения. Показать на примерах.
15. Деление. Показать на примерах. Зависимые и примитивные реляционные операции. Как зависимые операции выводятся из примитивных.
16. Запросы, невыразимые средствами реляционной алгебры. Привести примеры.
17. Архитектура «клиент-сервер». Основной принцип технологии «клиент-сервер»
18. Понятие презентационной логики, бизнес-логики и логики обработки данных
19. Модель удаленного управления данными FS (модель файлового сервера). Достоинства и недостатки
20. Модель удаленного доступа к данным (RDA). Достоинства и недостатки
21. Модель (активного) сервера баз данных. Достоинства и недостатки
22. Модель сервера приложений. Достоинства и недостатки
23. Архитектура сервера баз данных. Централизованная архитектура и модель 1:1. Недостатки. Архитектура выделенного сервера (многопотоковая серверная архитектура). Недостатки
24. Архитектура виртуального сервера. Недостатки. Многонитиевая мультисерверная архитектура
25. Типы параллелизма.
26. Требования к безопасности реляционных СУБД
27. Пользователи СУБД. Объекты доступа. Привилегии. Операторы SQL для установки привилегий. Примеры.
28. Основные способы определения групп пользователей. Роль. Механизм ролей. Создание роли. Пример
29. Понятие транзакции. Свойства транзакции. Варианты завершения транзакции.
30. Расширенная модель транзакции. Смесь транзакций. График запуска набора транзакций.
31. Проблемы параллельной работы транзакций. Проблема потери результатов обновления. Неаккуратное считывание
32. Проблемы параллельной работы транзакций. Проблема несовместимого анализа.
33. Конкурирующие транзакции. Конфликты между транзакциями. Графики запуска набора транзакции. Способы разрешения конкуренций между транзакциями.
34. Типы блокировок. Протокол доступа к данным. Уровни блокирования.
35. Управление транзакциями. Команды управления транзакциями.
36. Явные транзакции. Вложенные транзакции
37. Управление блокировками
38. Тупиковые блокировки
39. Уровни изоляции
40. Режимы SQL. Процесс выполнения операторов sql
41. Классификация файлов и файловых структур
42. Файлы прямого и последовательного доступа
43. Доступ к файлу с использованием методов хеширования

44. Понятие коллизий. Методы разрешения коллизий
45. Понятие индексного файла. Плотный индекс. Неплотный индекс.
46. Индексы в виде B-деревьев.
47. Кластеризованный индекс
48. Некластеризованный индекс
49. Представление связи 1:M в структурах хранения. Однонаправленные указатели. Пример. Двунаправленные указатели. Пример. Алгоритм поиска записи в связях 1:M
50. Инвертированные списки
51. Страничная организация данных. Экстенты. Типы экстентов. Типы страниц. Параметры страниц
52. Страницы размещения. Страницы данных. Строки данных. Индексные страницы. Текстовые страницы.
53. Структура файлов в Ms SQL Server. Разделяемая память

6 Учебно-методическое и информационное обеспечение дисциплины

6.1 Список источников литературы

Основная литература

1. Карпова, И. П. Базы данных: учебное пособие / И. П. Карпова. - Санкт-Петербург : Питер, 2021. - 240 с. - (Серия «Учебное пособие»). - ISBN 978-5-4461-9681-4. - Текст : электронный. - URL: https://znanium.com/catalog/product/1857026
2. Шустова, Л. И. Базы данных : учебник / Л.И. Шустова, О.В. Тараканов. — Москва : ИНФРА-М, 2023. — 304 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — DOI 10.12737/11549. - ISBN 978-5-16-010485-0. - Текст : электронный. - URL: https://znanium.com/catalog/product/1986697
3. Голицына, О. Л. Базы данных : учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. — 4-е изд., перераб. и доп. — Москва : ФОРУМ : ИНФРА-М, 2023. — 400 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-00091-516-5. - Текст : электронный. - URL: https://znanium.com/catalog/product/1937956.

Дополнительная литература

1. Копырин, А. С. Базы данных: практикум : учебно-практическое пособие / А. С. Копырин. - Москва : ФЛИНТА, 2021. - 106 с. - ISBN 978-5-9765-4752-0. - Текст : электронный. - URL: https://znanium.com/catalog/product/1851992 (дата обращения: 06.10.2023).
2. Мартишин, С. А. Базы данных. Практическое применение СУБД SQL и NoSQL-типа для проектирования информационных систем : учебное пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. — Москва : ФОРУМ : ИНФРА-М, 2024. — 368 с. — (Высшее образование). - ISBN 978-5-8199-0946-1. - Текст : электронный. - URL: https://znanium.com/catalog/product/2096940
3. Агальцов, В. П. Базы данных : в 2 кн. Книга 2. Распределенные и удаленные базы данных : учебник / В. П. Агальцов. — Москва : ФОРУМ : ИНФРА-М, 2020. — 271 с. — (Высшее образование: Бакалавриат). - - Текст : электронный. - URL: https://znanium.com/catalog/product/1093648. – Режим доступа: по подписке.

4. Полищук, Ю. В. Базы данных и их безопасность : учебное пособие / Ю. В. Полищук, А. С. Боровский. — Москва : ИНФРА-М, 2020. — 210 с. — (Высшее образование: Специалитет). - - Текст : электронный. - URL: https://znanium.com/catalog/product/1011088. – Режим доступа: по подписке.
5. Полищук, Ю. В. Базы данных и их безопасность : учебное пособие / Ю.В. Полищук, А.С. Боровский. — Москва : ИНФРА-М, 2023. — 210 с. — (Высшее образование: Специалитет). — DOI 10.12737/1011088. - ISBN 978-5-16-014924-0. - Текст : электронный. - URL: https://znanium.com/catalog/product/1905717

6.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Электронно-библиотечная система «Знаниум» - Режим доступа: http://znanium.com
2. - Информационная система «Единое окно доступа к образовательным ресурсам». - Режим доступа: http://window.edu.ru
3. Онлайн-энциклопедия. - Режим доступа: http://encyclopedia.ru
4. Электронный справочник «Информио» для высших учебных заведений. - Режим доступа: http://www.informio.ru
5. КонсультантПлюс. Правовая поддержка. - Режим доступа: http://www.consultant.ru/
6. Национальный открытый университет «ИНТУИТ». - Режим доступа: https://www.intuit.ru/
7. Сайт Microsoft - Режим доступа: https://msdn.microsoft.com/ru-ru/library/
8. Научная библиотека РГГУ - Режим доступа: http://liber.rsuh.ru/
9. «CITFORUM»: Аналитическая информация в сфере IT. - Режим доступа: http://citforum.ru/

6.3 Профессиональные базы данных и информационно-справочные системы

Доступ к профессиональным базам данных: https://www.rsuh.ru/liber/resources.php

Информационные справочные системы:
1. Консультант Плюс
2. Гарант

7 Материально-техническое обеспечение дисциплины

Для материально-технического обеспечения дисциплины необходимы:

- *для лекций:*
- учебная аудитория,
- доска,
- проектор (стационарный или переносной),
- компьютер или ноутбук,
- программное обеспечение (ПО).

- *для семинаров:*
- лаборатория,
- доска,
- проектор (стационарный или переносной),
- компьютер или ноутбук для преподавателя,
- компьютеры для обучающихся,
- выход в Интернет,

- программное обеспечение (ПО).

Перечень программного обеспечения (ПО)

- для лекций:

№п/п	Наименование ПО	Способ распространения
1	Microsoft Office	лицензионное
2	Windows	лицензионное
3	Kaspersky Endpoint Security	лицензионное

- для семинаров занятий:

Наименование ПО	Способ распространения
Windows	лицензионное
Microsoft Office	лицензионное
Mozilla Firefox	свободно распространяемое
Kaspersky Endpoint Security	лицензионное
Microsoft SQL Server 2008	лицензионное

8 Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья

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 семинарских занятий

Семинар № 1

Начальные навыки работы в среде конкретной СУБД. Реализация основных функций СУБД. Разработка и реализация первой базы данных в среде конкретной СУБД "

Цель работы

Познакомиться с СУБД Microsoft Access с помощью создания небольшой Базы данных.

Задачи

- Создать Базу данных и заполнить ее данными.
- Сформировать SQL запросы, исходя из предоставленных условий.
- Создать самостоятельно базу данных по предложенному варианту

Ход Работы

Теоретические сведения

База данных — это средство сбора и организации информации. В базах данных могут содержаться сведения о людях, продуктах, заказах и т. д. Многие базы данных изначально представляют собой список в текстовом процессоре или электронной таблице. По мере того как список разрастается, в нем накапливаются излишние и противоречивые данные. В форме списка эти данные становится все труднее понять, а возможности поиска или извлечения подмножеств данных для просмотра весьма ограничены. Когда возникают подобные проблемы, полезно перенести информацию в базу данных, созданную с помощью системы управления базами данных (СУБД), например Office Access 2007.

Компьютерная база данных представляет собой хранилище объектов. В одной базе данных может содержаться несколько таблиц. Например, система складского учета, в которой используются три таблицы, — это не три базы данных, а одна, содержащая три таблицы. В базе данных Access таблицы сохраняются в одном файле вместе с другими объектами, такими как формы, отчеты, макросы и модули, если только база данных не предназначена специально для использования данных или кода из другого источника. Базы данных, созданные в формате Access 2007, имеют расширение имени файла ACCDB, а базы данных, созданные в более ранних форматах Access, — расширение MDB. Приложение Access 2007 можно использовать для создания файлов в более ранних форматах файлов (например, Access 2000 и Access 2002-2003).

Приложение Access предоставляет следующие возможности:

- добавление новых данных в базу данных (например, новой позиции в складскую опись);
- изменение существующих данных в базе данных (например, изменение текущего размещения позиции на складе);
- удаление сведений (например, если позиция продана или отбракована);
- организация и просмотр данных различными способами;
- совместное использование данных посредством отчетов, сообщений электронной почты, внутренней сети или Интернета.

Access, при работе с базой данных, иначе взаимодействует с жёстким (или гибким) *диском*, нежели другие программы.

В других программах, файл-документ, при открытии, полностью загружается в оперативную память, и новая редакция этого файла (изменённый файл) целиком записывается на *диск* только при нажатии кнопки «сохранить».

В Access новая редакция содержимого изменённой ячейки таблицы записывается на диск (*сохраняется*) сразу, как только курсор клавиатуры будет помещён в другую ячейку (или новая редакция изменённой *записи* записывается на диск сразу, как только курсор клавиатуры будет поставлен в другую *запись* (строку)). Таким образом, если внезапно отключат электричество, то пропадёт только изменение той *записи*, которую не успели покинуть.

Целостность данных в Access обеспечивается также за счёт механизма транзакций.

Кнопка «Сохранить» в Access тоже есть, но в Access в *режиме просмотра данных* она нужна, в первую очередь, для сохранения изменённого режима показа таблицы или другого объекта — то есть, для сохранения таких изменений, как:

- изменение ширины столбцов и высоты строк,
- перестановка столбцов в режиме просмотра данных, «*закрепление*» столбцов и освобождение *закреплённых* столбцов,
- изменение *сортировки*,
- применение нового *фильтра*,
- изменение шрифта; цвета текста, сетки и фона,
- и т. п.

Кроме того, в Access эта кнопка нужна в режиме «Конструктор» для сохранения изменений структуры объекта базы данных, сделанных в этом режиме.

Даже если в процессе работы с файлом базы данных не применялся режим «Конструктор» и новые данные в базу данных не добавлялись (то есть если база данных только просматривалась), то всё равно файл базы данных имеет тенденцию со временем, в процессе работы с ним, всё больше и больше увеличиваться в размере. Очень способствует увеличению размера файла применение новых *сортировок* и *фильтров* (особенно если было применено несколько разных, сильно отличающихся друг от друга сортировок/фильтров).

Это приращение размера файла является, фактически, пустотой, но эта пустота лежит внутри файла, увеличивая его объём.

Чтоб вернуть файлу базы данных нормальный (минимальный) объём (то есть, чтоб убрать из файла пустоту), в Access есть кнопка «Сжать и восстановить базу данных» — эту кнопку нужно время от времени нажимать (при нажатии этой кнопки никакая информация, никакие данные из файла базы данных не удаляются). Так же базу данных можно запустить с параметром `/compact`, что выполнит сжатие автоматически и закроет базу по окончании процесса.

Создание базы данных «Отдел кадров»

Создание и заполнение базы данных данными десяти сотрудников.

Первоначально создаем таблицу и в ее конструкторе создаем несколько полей: Номер дела, Имя, Фамилия, Отчество, Дата приема на работу, Образование, Зарплата, Количество детей, Военная обязанность, Фото, Резюме, Приказ о приеме.

Далее настраиваем тип данных для каждого поля в отдельности и производим настройку полей: уменьшаем размер поля, добавляем индексирование, обязательность заполнения, подпись, возможность наличия пустых строк, значения по умолчанию. Заполняем все поля.

На рис.1 показано как выглядит Полная таблица.

Sotrudniki

Номер	Фамилия	Имя	Отчество	Дата приема	Образование	Зарплата	Количество детей	Фото	Военная обязанность	Резюме	При
1	Лямцева	Юлия	Николаевна	07.09.2015	Высшее	76 000,00р.	1	Package	☑	Есть	https://www
2	Воропаева	Виктория	Викторовна	06.05.2015	Начальное	21 000,00р.	0	Package	☐		
3	Волкова	Мира	Егоровна	05.07.2015	Среднее	34 000,00р.	3		☐		https://www
4	Воропаев	Андрей	Григорьевич	01.03.2015	Высшее	68 000,00р.	2		☑	Есть	
5	Рогов	Николай	Сергеевич	01.05.2015	Средне-специали	31 000,00р.	0	Package	☑		
6	Кондратьева	Ирина	Андреевна	13.09.2015	Среднее	45 687,00р.	1		☐	Есть	
7	Карпов	Антон	Егорович	07.04.2015	Средне-специали	32 421,00р.	2	Package	☐		https://www
8	Аниченко	Влад	Макарович	11.06.2015	Начальное	10 200,00р.	3		☐		
9	Суляева	Анна		21.05.2015	Средне-специали	17 000,00р.	0	Package	☐	Есть	https://www
10	Макаров	Анатолий		13.03.2015	Начальное	13 498,00р.	2	Package	☑	Есть	https://www

Рис. 1 База данных сотрудников

Переходим во вкладку создание и выбираем конструктор запросов. В нем добавляем нашу таблицу и переходим в режим SQL.

Создание запросов

Теоретические сведения и задания

Запросы манипулирования или выборки предназначены для выборки нужной информации из таблиц или других запросов или для изменения, удаления или добавления данных в таблицы. Запросы реализуются с помощью специального языка баз данных SQL.

Запросы на выборку. Оператор SELECT.

SELECT …
FROM …
WHERE …
GROUP BY …
ORDER BY …;

Строки SELECT и FROM обязательны. Остальные добавляются по мере надобности. В конце оператора ставится точка с запятой.

SELECT содержит список выводимых полей запроса, либо выражение (для вычисляемого поля), либо обращение к агрегатной функции, либо знак *, который обозначает, что выводить нужно все поля таблицы. Все элементы этой строки перечисляются через запятую. Если имя поля содержит пробел, то имя поля нужно заключить в квадратные скобки. Если в БД существуют два поля с одинаковыми именами (например, в двух разных таблицах), то поле выводится так: ***Имя_таблицы.Имя_поля***, то есть перед именем поля ставится имя таблицы и они разделяются точкой.

FROM содержит список таблиц, используемых в запросе, перечисленных через запятую.

Например:

1. Вывести все поля из таблицы Сотрудники.

*SELECT **
FROM Сотрудник;

2. Вывести список всех сотрудников.

SELECT Фамилия, Имя, Отчество
FROM Сотрудник;

WHERE предназначен для отображения только тех записей таблицы, которые удовлетворяют некоторому условию. Условие записывается в виде предиката, которое представляет собой логическое выражение (возвращающее true или false). Предикат может содержать в себе операции отношения (<, >, =, <=, >=, <> не равно), логические операции (and, or, not) и другие логические операции.

Например:

3. Вывести всех сотрудников, у которых нет детей.

SELECT Фамилия, Имя, Отчество
FROM Сотрудники
WHERE КолДетей=0;

4. Вывести всех сотрудников с высшим образованием.

SELECT Фамилия, Имя, Отчество
FROM Сотрудники
WHERE Образование = ”Высшее”;

5. Вывести табельный номер, фамилию и зарплату сотрудников, поступивших на работу после 1.01.2011 года.

SELECT ТабНомер, Фамилия, Зарплата
FROM Сотрудники
WHERE ДатаНайма>#01/01/2011#;

Обратите внимание, что текстовые константы заключаются в кавычки, даты обрамляются с двух сторон знаками #.

6. Вывести список сотрудников, имеющих более 2-ух детей и получающих зарплату менее 15000 рублей.

Самостоятельное задание:

7.1. Вывести всех сотрудников, которые получают зарплату от 15000 до 20000 рублей включительно.

Специальные логические операции Access табл.1.

Таблица 1.

Оператор	Описание	Пример
Like	Определяет, начинается ли строковое значение с заданного образца.	Like "Доц*"
In	Определяет, является ли строковое значение элементом списка значений.	In("Доцент", "Профессор")
Between … And …	Определяет, находится ли числовое значение в определенном диапазоне значений	Between 3 And 7
Is Null	Определяет, является ли значение неопределенным	

Все эти операции также употребляются с операцией NOT: Not Between, Not In, Not Like, Is Not Null.

Самостоятельное задание:
7.2. Переписать предыдущий запрос при помощи операции Between.
8. Вывести всех сотрудников, фамилия которых начинается на букву К.
9. Вывести всех сотрудников, имеющих начальное и среднее образование.
10. Вывести всех сотрудников, которые принесли фото.
11. Вывести всех военнообязанных сотрудников.
12. Вывести всех сотрудников, которые поступили на работу до 1.01.2000 и после 31.12.2010.

Использование выражений для создания вычисляемых полей

Вычисляемые поля отображают данные, рассчитанные на основе значений других полей из той же строки таблицы запроса.

Например, Если мы хотим определить чистую зарплату за вычетом подоходного налога, то соответствующее вычисляемое поле будет иметь следующий вид: *Зарплата * 0.87.* Вычисляемое поле не существует в БД, а генерируется во время выполнения запроса. Оно не имеет имени, поэтому его надо назвать. Имя полю присваивается при помощи конструкции as.

Например:

13. Вывести Табельный номер, Фамилию, Зарплаты и Чистую зарплату для каждого сотрудника. Чистую зарплату назвать *К выдаче*.

*SELECT ТабНомер, Фамилия, Зарплата, Зарплата*0.87 as [К выдаче]*
FROM Сотрудник;

С помощью *as* можно переименовывать любые поля таблицы в запросе:

*SELECT ТабНомер, Фамилия, Зарплата as Начислено, Зарплата*0.87 as [К выдаче]*

FROM Сотрудник;

Самостоятельное задание:

14. Определить стаж в годах каждого сотрудника.

15. Вывести Фамилию, Имя, Отчество и Зарплату каждого сотрудника. Фамилию, Имя, Отчество объединить в одном столбце и назвать ФИО.

Примечание: Операция слияния текстовых данных +.

Запросы с параметрами

Параметр – это типа переменной, вместо которой можно подставлять любые значения. Параметры используются для того, чтобы вывести данные для конкретного значения поля (полей). Параметр имеет вид: [текст параметра]. Параметры используются в выражениях в условиях отбора или вычисляемых полях. При выполнении запроса, если он содержит параметр (параметры) на экране появляется диалоговое окно с запросом значения параметра. Это значение подставляется в выражение, содержащее параметр. Текст параметра отражается в диалоговом окне, поэтому он должен нести информацию о запрашиваемом параметре. Текст параметра не должен совпадать с именами полей таблицы.

Например:

16. Вывести данные на конкретного сотрудника

*SELECT **

FROM Сотрудник

WHERE Фамилия=[Введите фамилию:];

Самостоятельное задание:

17. Вывести всех сотрудников, зарплата которых находится в диапазоне.

18 . Модифицируйте запрос 16, так, чтобы можно было вводить только часть фамилии.

19. Вывести Фамилию, Имя, Отчество и Дату найма всех сотрудников по определенному в параметре году найма.

Сортировка значений в запросе

Для того чтобы отсортировать запрос по значению какого-либо поля, используется строка ORDER BY, которая должна быть всегда последней в запросе SELECT. При этом в списке ORDER BY можно перечислять имена полей через запятую или ставить их номера (номер в строке SELECT). Если данные надо отсортировать по возрастанию, после имени поля или номера через пробел ставится слово ASC (не обязательно), по убыванию – DESC.

Например:

Исправьте запрос 1: отсортируйте данные запроса по возрастанию фамилий.

*SELECT **

FROM Сотрудник

ORDER BY Фамилия ASC;

Самостоятельное задание: Исправьте запрос 13: отсортируйте данные запроса по убыванию поля *К выдаче*.

Запросы с группировкой

Данные в запросе можно сгруппировать по определенному полю (полям), содержащие одинаковые значения, которые образуют группу. Для остальных полей запроса данные группируются по одной из агрегатной функций:

SUM – выводится сумма значений этого поля в группе;

AVG – выводится среднее значений этого поля в группе;

MIN – выводится минимальное значение этого поля в группе;

MAX – выводится максимальное значение этого поля в группе;

COUNT – выводится количество значений этого поля в группе.

Все поля группировки перечисляются в строке GROUP BY в порядке группировки.

Например:

20. Вывести минимальную, максимальную и среднюю зарплату по видам образования. В этом запросе поле группировки – *Образование*. К полю *Зарплата* применяется трижды агрегатные функции: MIN, MAX и AVG.

SELECT Образование, MIN(Зарплата) as Минимум, MAX(Зарплата) as Максимум, AVG(Зарплата) as Средняя

FROM Сотрудник

GROUP BY Образование;

21. Подсчитать количество сотрудников, имеющих 0, 1 и т.д. детей.

SELECT КолДетей, COUNT(ТабНомер) as [Количество сотрудников]

FROM Сотрудник

GROUP BY КолДетей;

Использование конструкции HAVING в группировке.

Если в запросе нужно применить условия группировки, то к конструкции GROUP BY добавляется предложение HAVING, за которым следует предикат (условие группировки). HAVING употребляется только с предложением GROUP BY.

Например:

22. Подсчитать количество сотрудников, имеющих высшее, среднее и начальное образование, и вывести только те строки, количество сотрудников в которых больше 3.

SELECT Образование, Count(ТабНомер) AS Количество

FROM Сотрудники

GROUP BY Образование

HAVING Count(ТабНомер)>3;

Оператор вставки (добавления) данных в таблицу

INSERT INTO Имя_таблицы (Список полей)

VALUES (Список значений);

Список полей и список значений должны совпадать по следующим параметрам:

.a Их количество должно быть одинаковым;

.b Их порядок должен совпадать;

.c Типы данных должны совпадать.

Если вставляются значения всех полей, то список полей писать не обязательно. Тогда оператор выглядит следующим образом:

INSERT INTO Имя_таблицы

VALUES (Список значений);

Например:

23. Вставить нового сотрудника (только Фамилию, Дата_рождения, Образование и Зарплату)

INSERT INTO Сотрудники (Фамилия, ДатаНайма, Образование, Зарплата)

VALUES ("Гаврилов", #3/05/2009#, "высшее", 56000);

Оператор обновления данных в таблице

Синтаксис:

UPDATE Имя_таблицы

SET Имя_поля = значение или выражение для вычисления значения
WHERE предикат;
Например:
24. Сотрудник поменял фамилию. Поменять его фамилию в БД при помощи оператора UPDATE.
UPDATE Сотрудники
SET Фамилия = "Самойлов"
WHERE Фамилия="Гаврилов";
25. Увеличить зарплату на 5000р. для сотрудников с высшим образованием.

Оператор удаления записей из таблицы

Синтаксис:
DELETE FROM Имя_таблицы
WHERE Предикат;
Оператор DELETE, записанный без строки WHERE удалит все записи таблицы.
Например:
26. Удалить всех сотрудников с начальным образованием.
DELETE FROM Сотрудники
WHERE Образование="Начальное";
27. Удалить сотрудника с конкретной фамилией (фамилия вводится по параметру).

Самостоятельная работа

1. Создать однотабличную БД по предложенному варианту. Определить первичный ключ, настроить индексы и свойства полей, в том числе подстановки. Заполнить таблицу (6-7 записей).
2. Создать запросы:
    a. На выборку, которые обязательно содержат условия отбора, сортировки, группировки, вычисляемые поля, параметры.
    b. На удаление (с параметрами).
    c. На добавление (с параметрами).
    d. На обновление (с параметрами).

Оформить отчет по работе

Содержание отчета

1. Описание реализованной базы данных
2. Рисунки, иллюстрирующие процесс создания базы данных и настройки параметров
3. Запросы, записанные на языке SQL

Семинар № 2

"Инфологическое моделирование предметной области"

Цель работы

Построение инфологической модели по выбранной предметной области.

Задачи

- Проанализировать предметную область.
- Построить ER-модель.
- Определить задачи, которые должна выполнять данная предметная область.

Ход работы

Теоретическая часть

База данных – именованная совокупность данных, отражающая совокупность объектов и их отношений в рассматриваемой предметной области.

Предметная область базы данных – область применения конкретной БД или область реального мира, которая является предметом исследования.

Семантическое моделирование – моделирование структуры данных, опираясь на смысл этих данных. В качестве инструментов используются различные диаграммы «сущность-связь».

Диаграмма ER-модели состоит из нотаций графического отображения сущности, атрибутов и взаимосвязи между сущностями.

Сущность – класс однотипных объектов, информация о которых должна быть учтена в данной модели. Должна иметь наименование, выраженное в единственном числе (например, сотрудник, накладная, студент).

Экземпляр сущности – конкретный представитель данной сущности. Экземпляры должны быть различимы, т.е. иметь уникальные свойства для каждого экземпляра. Экземпляры сущности в ER- модели не отображаются.

Атрибут сущности – именованная характеристика, являющаяся некоторым свойством сущности. Имя атрибута сущности должно быть существительным в ед. числе, возможно с характеризующим прилагательным. Отображается внутри прямоугольника (см. рис. Выше)

Ключ сущности – не избыточный набор атрибутов, значение которых, в совокупности, является уникальными для каждого экземпляра сущности. Не избыточность заключается в том, что удаление любого атрибута из ключа нарушает его уникальность. Сущность может иметь несколько различных ключей. Ключевые атрибуты в диаграмме подчеркиваются.

Связь – некая ассоциация между сущностями. Одна сущность может быть связана с другой или сама с собой. Связи определяются глаголом в неопределенной форме.

Модальные связи ---------- - экземпляр одной сущности может быть связан с 1 или несколькими экземплярами другой сущности. А может быть не связан ни с одной.

Мод~~альные свя~~зи - экземпляр должен/обязан быть связан хотя бы с 1 экземпляром другой сущности. Может иметь связь с разных концов.

При разработке данной модели мы должны получить следующую информацию о предметной области (список сущностей предметной области, список атрибутов сущностей, описание взаимосвязей между сущностями).

Практическая часть

Практическая часть выполняется по одному из предложенных вариантов (предметных областей):

1. Проектирование БД «Абитуриент»
2. Проектирование БД «Автобусный парк»
3. Проектирование БД «Автовокзал»
4. Проектирование БД «Автозаправочный комплекс»»
5. Проектирование БД «Автомагазин»
6. Проектирование БД «Автоматизация учета и контроля успеваемости студентов»»
7. Проектирование БД «Автосервис»
8. Проектирование БД «Автошкола»
9. Проектирование БД «Агентство недвижимости»
10. Проектирование БД «Аптека»
11. Проектирование БД «Аренда жилых помещений »
12. Проектирование БД «Аренда офисных помещений »
13. Проектирование БД «Аукцион»
14. Проектирование БД «Аэропорт»
15. Проектирование БД «Банно-оздоровительный комплекс»
16. Проектирование БД «Библиотека»
17. Проектирование БД «Больница. Работа с пациентами»

18. Проектирование БД «Видеопрокат»
19. Проектирование БД «Гостиница. Работа с клиентами»
20. Проектирование БД «Дачный кооператив»
21. Проектирование БД «Деканат»
22. Проектирование БД «Жилищно-коммунальное хозяйство»
23. Проектирование БД «Издательство. Работа с авторами»
24. Проектирование БД «Издательство. Служба маркетинга»
25. Проектирование БД «Интернет-магазин»
26. Проектирование БД «Кадры для вуза»
27. Проектирование БД «Касса железнодорожного вокзала (бронирование билетов)»
28. Проектирование БД «Кинотеатр»
29. Проектирование БД «Компьютерные курсы»
30. Проектирование БД «Контроль выполнения нагрузки преподавателей вуза»
31. Проектирование БД «Контроль успеваемости студентов вуза»
32. Проектирование БД «Курьерские служба»
33. Проектирование БД «Магазазин спортивных товаров»
34. Проектирование БД «Магазин бытовой техники»
35. Проектирование БД «Мебельный магазин»
36. Проектирование БД «Модельное агентство»
37. Проектирование БД «Начисление квартплаты»
38. Проектирование БД «Обмен валюты»
39. Проектирование БД «Оборудование компьютерных классов учебного заведения»
40. Проектирование БД «Обувная мастерская»
41. Проектирование БД «Пожарная часть»
42. Проектирование БД «Полиграфическое оборудование»
43. Проектирование БД «Поликлиника. Планирование и учет работы медицинского персонала»
44. Проектирование БД «Поликлиника. Работа с пациентами»
45. Проектирование БД «Поликлиника. Учет льготных лекарств»
46. Проектирование БД «Поставка товаров и расчет с поставщиками в магазине стройматериалов»
47. Проектирование БД «Почта Учет изданий»
48. Проектирование БД «Провайдерская компания»
49. Проектирование БД «Продажа авиабилетов»
50. Проектирование БД «Продажа ж/д билетов»
51. Проектирование БД «Продажа земельных участков»
52. Проектирование БД «Продажа компьютерной техники»
53. Проектирование БД «Продажа легковых автомобилей»
54. Проектирование БД «Продвижение сайтов»
55. Проектирование БД «Пункт проката автомобилей»
56. Проектирование БД «Расписание движения поездов»
57. Проектирование БД «Расписание занятий»
58. Проектирование БД «Регистрация и учёт юридических и физических лиц в налоговых органах РФ»
59. Проектирование БД «Ресторанный бизнес»
60. Проектирование БД «Риэлтерская фирма»
61. Проектирование БД «Салон красоты»
62. Проектирование БД «Социолог. Анкетирование, тестирование»
63. Проектирование БД «Социологическое исследование»
64. Проектирование БД «Страховая компания»
65. Проектирование БД «Строительство дач»

66. Проектирование БД «Строительство новостроек»
67. Проектирование БД «Таксопарк»
68. Проектирование БД «Телевидение»
69. Проектирование БД «Телефонная станция. Учет расчетов с клиентами»
70. Проектирование БД «Тестирование»
71. Проектирование БД «Трансагентство»
72. Проектирование БД «Тренажерный зал»
73. Проектирование БД «Туристическая фирма»
74. Проектирование БД «Управление заказами. АРМ менеджера по работе с клиентами»
75. Проектирование БД «Учет автоперевозок на предприятии. АРМ автодиспетчера»
76. Проектирование БД «Учет техники на предприятии»
77. Проектирование БД «Учет товаров на складе»
78. Проектирование БД «Учет услуг юридической консультационной фирмы»
79. Проектирование БД «Хозяйственный магазин»
80. Проектирование БД «Чемпионат мира по футболу»
81. Проектирование БД «Читательный зал»
82. Проектирование БД «Экскурсионная фирма»

Пример построения ER-модели.

Предметная область агентство недвижимости: «Агентство недвижимости»

Цель модели: показать схему взаимодействия агентства недвижимости, собственника и покупателя при свободной продаже за наличные деньги.

Ограничения:

В данной модели не участвует банк: комиссия и стоимость квартиры на сделке не закладываются в ячейку. Реализована только свободная продажа

Агентство недвижимости занимается:

- покупка и продажа жилой, загородной и коммерческой недвижимости;
- аренда жилой, загородной и коммерческой недвижимости;
- регистрация сделок по Москве и Московской области;
- страхование сделок;
- юридическое сопровождение.

Приходя в агентство, клиент заключает с агентством договор об оказании услуг, но основе которого агентство развивает дальнейшую деятельность, в зависимости от необходимой услуги (данная модель реализует свободную продажу). Клиент предоставляет агентству пакет документов, в который входят: паспортные данные, свидетельство о собственности, передаточный акт, расписка о получении денежных средств (если имеется), выписка из домовой книги, единый жилищный документ (далее ЕЖД), справка об отсутствии долгов, справка НД и ПНД, нотариальное согласие супруга (супруги).

Агентство закрепляет за предоставленной квартирой сотрудника, который начинает курировать этот объект. Данный сотрудник занимается размещением рекламы, показом квартиры, подготовкой пакета документов, в том числе договора купли продажи и юридически сопровождает сделку.

Если появляется покупатель, то для проведения сделки он должен предоставить свои паспортные данные.

Сделка представляет собой заключение договора купли продажи (далее ДКП), подписание передаточного акты и написании расписки о получении денежных средств. В ДКП прописываются данные обеих сторон (собственника и покупателя), адрес объекта, стоимость объекта, наличие согласия супруги, отсутствие обременений и задолженностей по квартире. В передаточном акте указываются данные обеих сторон,

адрес квартиры, стоимость, срок передачи квартиры после регистрации в государственном реестре. Расписка состоит из даты, паспортных данных собственника, номера ДКП.

Далее ДКП регистрируется государственном реестре. После этого новый собственник получает документы о собственности. После этого он может распоряжаться квартирой по своему усмотрению.

Данная модель должна отвечать на такие вопросы как:

- Как называется агентство и какие его реквизиты (Физический, Юридический адрес, БИК, ИНН, расчетный счет, КПП, ОГРН, уполномоченный банк).
- Какие сотрудники, в каком агентстве состоят.
- Какой сотрудник закреплен за данной сделкой.
- Кто покупатель, его паспортные данные.
- Кто собственник и его паспортные данные.
- Какие документы подал собственник (Номер документа и его атрибуты)
- Когда и на какие услуги заключен договор оказания услуг.
- Какая комиссия выплачивается агентству за сделку.
- Какой срок действия договора оказания услуг.
- Какой адрес квартиры, ее жилая и общая площадь.
- Какая стоимость квартиры.
- Когда заключался договор ДКП.
- Какая дата регистрации в гос. Реестре.
- Когда проходила сделка.
- Какие услуги может оказывать агентство.
- С помощью вычисляемого поля можно вычислить эффективность работы сотрудника: какое количество объектов за определенный промежуток продал сотрудник.

В данной модели можно выделить несколько сущностей:

1. Сущность «Агентство недвижимости», которая характеризуется следующими атрибутами: Индивидуальный номер (далее id) агентства, название, физический адрес, юридический адрес, ИНН, КПП, ОГРН, БИК, расчетный счет, уполномоченный банк.
2. Сущность «Сотрудники», которая характеризуется следующими атрибутами: id сотрудника, id агентства, фамилия, имя, отчество, стаж сотрудника.
3. Сущность «Услуги», которая характеризуется следующими атрибутами: id агентства, вид услуги.
4. Сущность «Покупатель», которая характеризуется следующими атрибутами: id покупателя, фамилия, имя, отчество, серия и номер паспорта, кем выдан, дата выдачи, адрес прописки.
5. Сущность «Договор на оказание услуг», которая характеризуется следующими атрибутами: id договора, id собственника, id агентства, дата договора, комиссия, срок действия.
6. Сущность «Собственник», которая характеризуется следующими атрибутами: id собственника, id квартиры, id документа, фамилия, имя, отчество, серия и номер паспорта, кем выдан, дата выдачи, адрес прописки.
7. Сущность «Документы», которая характеризуется следующими атрибутами: id документа, кадастровый номер квартиры, id квартиры, стоимость квартиры, дата ДКП.
8. Сущность «Сделка», которая характеризуется следующими атрибутами: id собственника, id квартиры, id договора, id покупателя, id ДКП, id сотрудника, дата сделки.

9. Сущность «Квартира», которая характеризуется следующими атрибутами: id квартиры, адрес, жилая и общая площадь, этаж, количество комнат.

10. Сущность «ДКП», которая характеризуется следующими атрибутами: id ДКП, стоимость, id документа.

11. Сущность «Регистрация», которая характеризуется следующими атрибутами: id записи в реестре, дата регистрации, id ДКП.

12. Сущность «Свидетельство о собственности», которая характеризуется следующими атрибутами: id документа, дата выдачи, документы основания, субъект права, вид права, объект права, кадастровый номер, обременения права.

13. Сущность «Передаточный акт», которая характеризуется следующими атрибутами: id документа, id собственника, id предыдущего продавца, дата заключения акта.

14. Сущность «Предыдущий продавец», которая характеризуется следующими атрибутами: id предыдущего продавца, фамилия, имя, отчество, серия и номер паспорта, кем выдан, дата выдачи, адрес прописки.

15. Сущность «Выписка из домовой книги», которая характеризуется следующими атрибутами: id документа, номер выписки, дата запроса, дата рассмотрения, наименование объекта, назначение объекта, вид и номер права, дата регистрации права.

16. Сущность «Расписка о получении денежных средств», которая характеризуется следующими атрибутами: id документа, id предыдущего продавца, id собственника, дата расписки.

17. Сущность «ЕЖД», которая характеризуется следующими атрибутами: id документа, id собственника, характеристика помещения, количество комнат, этажность, этаж, материал стен, год постройки, лифт, электроснабжение, ванна, наличие приборов учета, задолженность, срок действия.

18. Сущность «Справка об отсутствии задолженностей», которая характеризуется следующими атрибутами: id документа, id собственника, дата выдачи, оплата счетов (с и до).

19. Сущность «НД», которая характеризуется следующими атрибутами: id документа, id собственника, название диспансера, дата осмотра, адрес проживания, фамилия, имя, отчество врача, противопоказания, вид сделки.

20. Сущность «ПНД», которая характеризуется следующими атрибутами: id документа, id собственника, название диспансера, дата осмотра, адрес проживания, фамилия, имя, отчество врача, противопоказания, вид сделки.

21. Сущность «Нотариальное согласие супруга», которая характеризуется следующими атрибутами: фамилия, имя, отчество, серия и номер паспорта, кем выдан, дата выдачи, адрес прописки, стоимость услуги, фамилия нотариуса, имя нотариуса, отчество нотариуса.

Данные отношения связаны связями:

1. Агентство недвижимости может заключать несколько договоров на оказание услуг. Каждый договор на оказание услуг должен быть заключен агентством недвижимости («один-ко-многим»).

2. В Агентстве недвижимости может работать несколько сотрудников. Каждый сотрудник должен работать в одном агентстве («один-ко-многим»).

3. Агентство недвижимости может оказывать несколько услуг. Каждая услуга должна быть оказана агентством («один-ко-многим»).

4. Собственник может заключить несколько договоров на оказание услуг. Каждый договор на оказание услуг должен быть заключен с собственником («один-ко-многим»).

5. Собственник может иметь несколько квартир. Каждая квартира должна иметь собственника («один-ко-многим»).

6. Собственник должен иметь несколько документов на квартиру. Каждый документ на квартиру должен быть у одного собственника («один-ко-многим»).

7. ДКП должен заключаться на нескольких сделках. На каждой сделке должен быть заключен один ДКП («один-ко-многим»).

8. Покупатель может заключить несколько сделок. Каждая сделка должна быть заключена с покупателем («один-ко-многим»).

9. Сотрудник может вести несколько сделок. Каждая сделка должна вестись сотрудником («один-ко-многим»).

10. Договор на оказание услуг может участвовать в нескольких сделках. В каждой сделке должен участвовать один договор на оказание услуг («один-ко-многим»).

11. Квартира может проводиться по нескольким сделкам. По каждой сделке должна проводиться квартира.

12. Собственник может заключить несколько сделок. Каждая сделка должна быть заключена с покупателем («один-ко-многим»).

13. Один ДКП регистрируется в одном реестре («один-к-одному»).

14. Квартира может иметь несколько документов. Каждый документ должен иметь одну квартиру(«один-ко-многим»).

15. Передаточный акт может иметь одного предыдущего продавца. Один предыдущий продавец может иметь один передаточный акт («один-к-одному»).

16. Тип документы имеет следующие подтипы: нотариальное согласие супруга, НД, ПНД, ЕЖД, справка об отсутствии долгов, расписка о получении денежных средств, выписка из домовой книги, передаточный акт, свидетельство о собственности.

Построенная ER-модель реализована на рис. 2.

Содержание отчета

1. Описание предметной области.
2. Описание ограничений.
3. Описание требований к БД
4. Описание основных сущностей, атрибутов сущностей и связей
5. ER-модель.

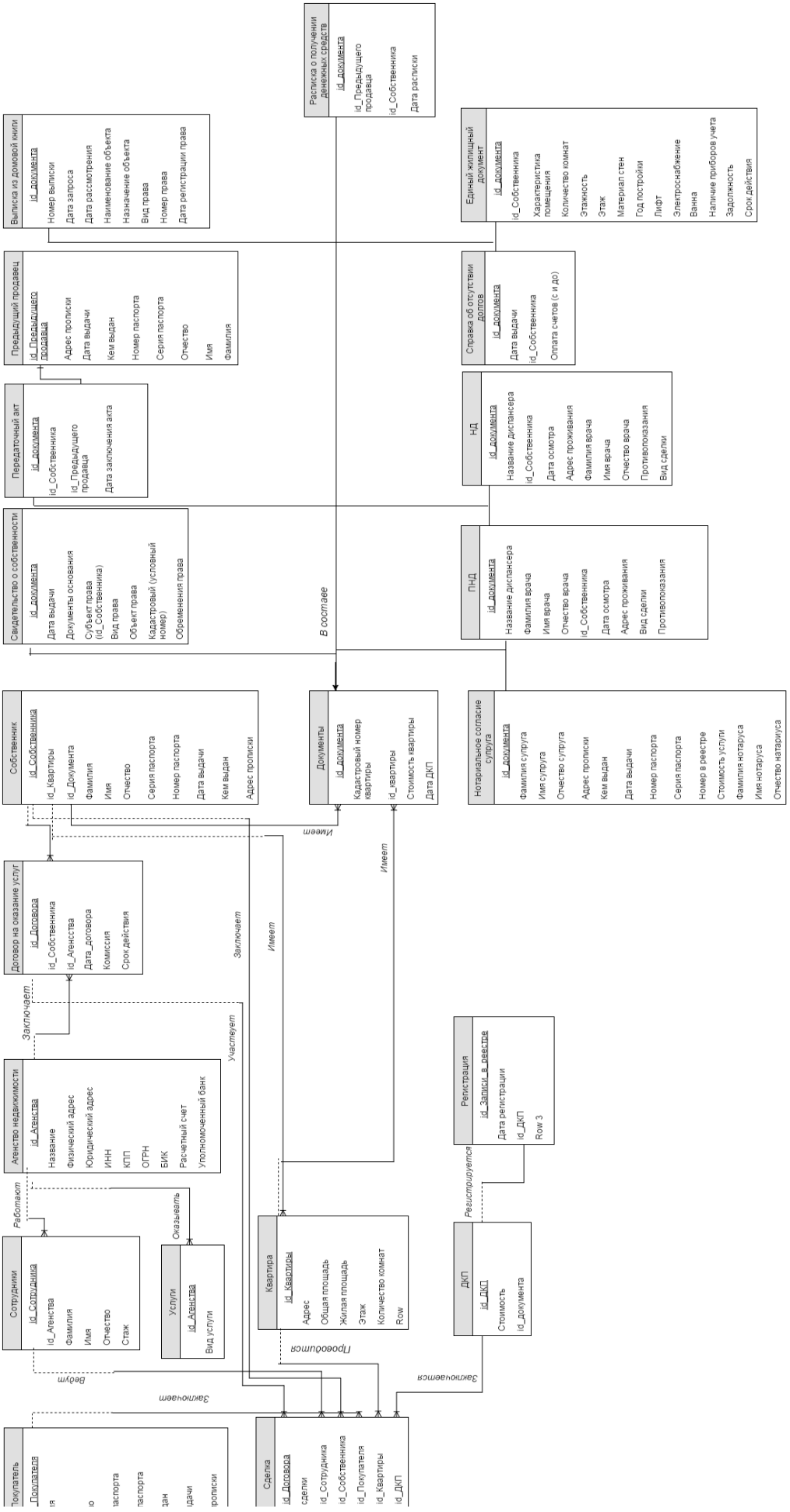

Рис.2 ER-модель «Агентство недвижимости»

Семинар № 3

«Даталогическое проектирование и реализация БД»

Цель работы

Построение даталогической модели по инфологической модели.

Задачи

- Проанализировать инфологическую модель.
- Перевести инфологическую модель в даталогическую по алгоритму.
- Представить даталогическую модель в графическом виде.
- Реализовать модель в Access.
- Определить структуру каждой таблицы, свойства и связи, заполнить небольшим количеством данных.

Ход работы

Теоретическая часть

Под даталогической понимается модель, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации. При этом даталогическая модель разрабатывается с учётом конкретной реализации СУБД, также с учётом специфики конкретной предметной области на основе ее инфологической модели.

Правила преобразования ER-модели в даталогическую (реляционную) модель:

1. Каждой сущности ставится в соответствие отношение реляционной модели данных.

Ограничение:

- Уникальность имени в рамках модели.
- Некоторые ограничения накладываются определенной СУБД.

2. Каждый атрибут сущности становится атрибутом соответствующего отношения. Переименования происходят с теми же правилами, что и в п.1. Для атрибута задается определенный тип данных СУБД. Обязательность или не обязательность атрибутов помечается допустимостью/недопустимостью нового значения.

3. Первичный ключ становится Primer key (PK). Атрибуты, входящие в первичный ключ, получают свойство обязательности Not Null.

4. В каждое отношение, соответствующее подчиненной сущности, добавляется набор атрибутов основной сущности, являющийся первичным ключом основной сущности. В отношении, соответствующем подчиненной сущности, этот набор атрибутов становится внешним ключом (Foreing Key - FK).

5. Для моделирования необязательного типа связи у атрибутов, соответствующего внешнему ключу, устанавливается свойство допустимости неопределенных значений (Null). При обязательном типе связи атрибуты получают свойство отсутствия неопределенных значений (Not Null).

6. Для отражения категоризации сущности при переходе к реляционной модели возможно несколько вариантов представления. Возможно создать только одно отношение подтипа для всего супертипа (объединяются атрибуты всех подтипов). Для ряда экземпляров значение атрибутов не будут иметь смысла, мы вынуждены ставить их в Null. Потребуются правила различения одних подтипов от других. Достоинство такого подхода то, что создается одно отношение.

Второй подход для каждого типа: создается отдельное отношение для супертипа.

Недостаток: создание многих отношений.

Достоинства: работа ведется со значимыми атрибутами подтипа. Для возможности перехода к подтипам от супертипа необходимо в супертип включать идентификатор связи.

Дополнительно, при описании можно указать тип дискриминатора. Дискриминатор

может быть взаимоисключающим или нет. Если установлен данный тип дискриминатора, то один экземпляр сущности супер типа связан с одним один экземпляр сущности подтипа. Как правило, наследование устанавливается только идентификатором.

7. В реляционной модели не поддерживается связь многие-ко-многим, а в ER – допустима такая связь через дополнительное отношение, которое связано с каждым исходным связью "один-ко-многим", атрибутами этого отношения являются первичные ключи связываемых отношений. При этом каждый из атрибутов нового отношения является внешним ключом (FOREIGN KEY), а вместе они образуют первичный ключ (PRIMARY KEY). Теория нормализации, которую мы рассматривали ранее применительно к реляционной модели, применима и к модели "сущность—связь". Поэтому нормализацию можно проводить и на уровне инфологической (семантической) модели и

смысл ее аналогичен нормализации реляционной модели.

8. Индексы создаются для первичного ключа, для внешних ключей (1:1 – уникальный индекс, 1:M – не уникальный индекс) и для атрибутов, которые часто используются в запросах.

Практическая часть

Предметная область: «Агентство недвижимости». В предыдущей практической работе была построена инфологическая модель.

В данной модели можно выделить следующие отношения:

1. Отношение «Agenstvo_Nedvigimosti», которая характеризуется следующими атрибутами: id_Agenstva (обязательное поле; индексированное, не допускает совпадения), Nazvanie(String: 50;обязательное поле; индексированное, не допускает совпадения), Fiz_Adres(String: 50;обязательное поле; индексированное, не допускает совпадения), Yur_Adres(String: 50;), INN(обязательное поле; индексированное, не допускает совпадения), KPP(обязательное поле; индексированное, не допускает совпадения), OGRN(обязательное поле; индексированное, не допускает совпадения), BIK(обязательное поле; индексированное, не допускает совпадения), Rassch_schet(обязательное поле; индексированное, не допускает совпадения), Upoln_bank.

2. Отношение «Sotrudniki», которая характеризуется следующими атрибутами: id_sotrudnika(обязательное поле; индексированное, не допускает совпадения), id_Agenstva, Fam (string: 50), FName (string: 50), Otch (string: 50), Stag.

3. Отношение «Uslugi», которая характеризуется следующими атрибутами: id_Agenstva (обязательное поле; допускаются совпадения), Vid_Uslugi(string 50).

4. Отношение «Pokupatel», которая характеризуется следующими атрибутами: id_Pokupatelya (обязательное поле; индексированное, не допускает совпадения), Fam (string: 50), FName (string: 50), Otch (string: 50), Seriya, Nomer, Kem_vidan, Data_vidachi, Adres_propiski (string: 50).

5. Отношение «Dogovor_na_okazanie_uslug», которая характеризуется следующими атрибутами: id Dogovora (обязательное поле; индексированное, не допускает совпадения), id Sobstvennika (обязательное поле; индексированное, не допускает совпадения), id Agenstva (обязательное поле; индексированное, не допускает совпадения), Date_zakl_dogovora, Komissiya, Srok deystviya.

6. Отношение «Sobstvennik», которая характеризуется следующими атрибутами: id_sobstvennika(обязательное поле; индексированное, не допускает совпадения), id_kv, id_dokumenta, Fam (string: 50), FName (string: 50), Otch (string: 50), Seriya, Nomer, Kem_vidan, Data_vidachi, Adres_propiski (string: 50).

7. Отношение «Documents», которая характеризуется следующими атрибутами: id_documents и id_sobstvennika (обязательное поле; индексированное, не допускает совпадения), Kadastr, id_kvart, Stoimost_kvart, Date_DKP.

8. Отношение «Sdelka», которая характеризуется следующими атрибутами: id_dogovora(обязательное поле; индексированное, не допускает совпадения), Date_sdeilki, id_sotrudnika, id_sobstvennika, id_pokupatelya, id_kvart, id_DKP(обязательное поле; индексированное, не допускает совпадения).

9. Отношение «Kvartira», которая характеризуется следующими атрибутами: id_Kv(обязательное поле; индексированное, не допускает совпадения), Adres (String: 50), Obchaya_p, Gil_pl , Etag, Kol_komnat.

10. Отношение «DKP», которая характеризуется следующими атрибутами: id_DKP (обязательное поле; индексированное, не допускает совпадения), Stoimost_kvart, id_documents.

11. Отношение «Registration», которая характеризуется следующими атрибутами: id_zapisi_v_reestre(обязательное поле; индексированное, не допускает совпадения), Date_registration, id_DKP(обязательное поле; индексированное, не допускает совпадения).

12. Отношение «Svideiestvo o sobstvennosti», которая характеризуется следующими атрибутами: id_documents(обязательное поле; индексированное, не допускает совпадения), Date_vidachi, Date_vidachi (string: 20), documets_osnov (string: 30), id_sobstvennika, vid_prava (string: 20), object_prava (string: 20), kadastr, Obremeneniya (string: 100).

13. Отношение «Peredatochn_act», которая характеризуется следующими атрибутами: id_documents, id_cobstvennika, id_pred_cobstvennika, Date_zakl_acta (string: 20).

14. Отношение «Pred_sobstvennik», которая характеризуется следующими атрибутами: id_pred_sobstv (обязательное поле; индексированное, не допускает совпадения), Fam (string: 50), FName (string: 50), Otch (string: 50), Seriya, Nomer, Kem_vidan, Data_vidachi, Adres_propiski (string: 50).

15. Отношение «vipiska_dom_kniga», которая характеризуется следующими атрибутами: id_documents(обязательное поле; индексированное, не допускает совпадения), Nomer_vipiski(обязательное поле; индексированное, не допускает совпадения), Data_zaprosa, Data_rassmotreniya, Name-object (string: 100), Naznach_object (string: 20), Vid-prava (string: 20), Nom_prava, Date_registr_prava.

16. Отношение «Raspiska», которая характеризуется следующими атрибутами: id_documents(обязательное поле; индексированное, не допускает совпадения), id_pred_sobstvennika, id_sobstvennika, Date_raspiski.

17. Отношение «EGD», которая характеризуется следующими атрибутами: id_documents (обязательное поле; индексированное, не допускает совпадения), id_sobstvennika(обязательное поле; индексированное, не допускает совпадения), Charact_pomecheniya (string: 100), Kolich_komnat, Etagnost, Etag, Material_sten, God_postroyki, Lift, Electr (string: 5), Vanna (string: 5), Pribori_ucheta (string: 50), Zadolgnost (string: 50), Srok_deystv (string: 20).

18. Отношение «Spravka_ob_otsutstvii_dolgov», которая характеризуется следующими атрибутами: id_documents (обязательное поле; индексированное, не допускает совпадения), date_vidachi, id_sobstvennika, оплата счетов (с и до), Oplata_schetov (string: 20).

19. Отношение «ND», которая характеризуется следующими атрибутами: id_documents (обязательное поле; индексированное, не допускает совпадения), id_sobstvennika, Name_dispanc (string: 50), Date_osmotra (string: 10), Adres_progiv (string: 50), Fam_vrach (string: 50), Name_vrach(string: 50), Otch_vrach(string: 50), protivopokaz (string: 50), Vid_sdelki (string: 50).

20. Отношение «PND», которая характеризуется следующими атрибутами: id_documents (обязательное поле; индексированное, не допускает совпадения), id_sobstvennika, Name_dispanc (string: 50), Date_osmotra (string: 10), Adres_progiv (string: 50), Fam_vrach (string: 50), Name_vrach(string: 50), Otch_vrach(string: 50), protivopokaz (string: 50), Vid_sdelki (string: 50).

21. Отношение «Notar_soglas», которая характеризуется следующими атрибутами: id_documents (обязательное поле; индексированное, не допускает совпадения), Fam (string: 50), FName (string: 50), Otch (string: 50), Seriya, Nomer, Kem_vidan, Data_vidachi, Adres_propiski (string: 50), Stoimost_uslugi, Fam_not (string: 50), FName_not (string: 50), Otch_not (string: 50).

Данные отношения связаны связями:

1. Агентство недвижимости может заключать несколько договоров на оказание услуг. Каждый договор на оказание услуг должен быть заключен агентством недвижимости («один-ко-многим»).
2. В Агентстве недвижимости может работать несколько сотрудников. Каждый сотрудник должен работать в одном агентстве («один-ко-многим»).
3. Агентство недвижимости может оказывать несколько услуг. Каждая услуга должна быть оказана агентством («один-ко-многим»).
4. Собственник может заключить несколько договоров на оказание услуг. Каждый договор на оказание услуг должен быть заключен с собственником («один-ко-многим»).
5. Собственник может иметь несколько квартир. Каждая квартира должна иметь собственника («один-ко-многим»).
6. Собственник должен иметь несколько документов на квартиру. Каждый документ на квартиру должен быть у одного собственника («один-ко-многим»).
7. ДКП должен заключаться на нескольких сделках. На каждой сделке должен быть заключен один ДКП («один-ко-многим»).
8. Покупатель может заключить несколько сделок. Каждая сделка должна быть заключена с покупателем («один-ко-многим»).
9. Сотрудник может вести несколько сделок. Каждая сделка должна вестись сотрудником («один-ко-многим»).
10. Договор на оказание услуг может участвовать в нескольких сделках. В каждой сделке должен участвовать один договор на оказание услуг («один-ко-многим»).
11. Квартира может проводиться по нескольким сделкам. По каждой сделке должна проводиться квартира.
12. Собственник может заключить несколько сделок. Каждая сделка должна быть заключена с покупателем («один-ко-многим»).
13. Один ДКП регистрируется в одном реестре («один-к-одному»).
14. Квартира может иметь несколько документов. Каждый документ должен иметь одну квартиру(«один-ко-многим»).
15. Передаточный акт может иметь одного предыдущего продавца. Один предыдущий продавец может иметь один передаточный акт («один-к-одному»).
16. Тип документы имеет следующие подтипы: нотариальное согласие супруга, НД, ПНД, ЕЖД, справка об отсутствии долгов, расписка о получении денежных средств, выписка из домовой книги, передаточный акт, свидетельство о собственности.

Содержание отчета

1. Описание предметной области.
2. Описание отношений и их реализация.
3. Описание связей между отношениями
4. Описание атрибутов отношений, выделение первичных ключей, связывание отношений. Реализация механизма связывания отношений типа 1:M.
5. Построение даталогической модели.
6. Реализация даталогической модели в среде конкретной СУБД

Семинар № 4
Формализация реляционной модели данных. Нормализация отношений

Часть 1

Цель:

Создание реляционных баз данных в среде конкретной реляционной СУБД.

Содержание занятия:

1. Проектирование таблиц БД в среде конкретной СУБД.
2. Определение первичных ключей.
3. Определение связей и их типов.
4. Создание схемы данных.
5. Создание индексов.
6. Заполнение таблиц.
7. Схема данных по выбранной предметной области, содержащая не менее 5 основных таблиц.

Работа в среде конкретной СУБД.

Вопросы для самостоятельной работы студентов:

1. Особенности реляционного подхода.
2. Свойства отношений.
3. Понятие первичного и внешнего ключа.

Решение практических задач.

1. Создать БД которая содержит сведения о континентах и странах(5-6 атрибутов в каждой таблице). Определить ключи и связи. Основной вопрос: какие страны на каких континентах находятся.
2. БД поликлиника(врачи и пациенты 5-6 атрибутов). Вопрос: какие врачи каких пациентов принимали, когда и какой диагноз был поставлен.
3. Бюро переводов (создать БД которая отражает сведения о сотрудниках бюро переводов). Личные данные отделить от общедоступных в отельной таблице. Основные вопросы:

• какой сотрудник, в каком отделе работает;

• какой сотрудник, какими языками и в какой степени владеет.

4. Школа (создать БД которая отражает сведения об основных сущностях предметной области Школа: учителях, классах, предметах, школьниках и т.д.). Основные вопросы:

• какие ученики, в каких классах учатся;

• какой учитель классным руководителем, какого класса является (не может быть руководителем у двух классов);

• какой учитель, какие предметы может вести;

• какой учитель в каком классе в каком кабинете какой предмет и когда ведет (расписание).

Часть 2

Цель:

Научиться приводить отношения к третьей нормальной форме.

Содержание занятия:

1. Использование формального аппарата для оптимизации схем отношений.
2. Переход к первой нормальной форме.
3. Выявление функциональных зависимостей. Переход ко второй нормальной форме.
4. Выявление транзитивных зависимостей. Третья нормальная форма.

Работа в среде конкретной СУБД.

Вопросы для самостоятельной работы сту-дентов:

1. Общее понятия традиционного подхода к проектированию реляционных моделей с помощью нормальных форм.
2. Функциональные зависимости

3. Свойства нормальных форм.

Решение практических задач.

1. Выделение атрибутного состава и построение функциональных зависимостей
2. Переход к нормализованной схеме отношений

Варианты заданий:

Примените алгоритм получения отношений в ЗНФ, если ЗНФ не соблюдается. Постройте реализацию полученных отношений средствами СУБД. Проверьте, выполняется ли свойство соединения без потерь.

1. Атрибуты:
   - ФИО вкладчика (фио)
   - Номер сберкнижки (номер)
   - Дата
   - Приход
   - Расход
   - Остаток
1. Атрибуты
   - Цех
   - Год
   - Код_станка
   - Количество станков (кол-во)
   - Код_детали
   - План производства деталей
2. Атрибуты
   - ФИО служащего (фио)
   - Должность
   - Дата
   - Зарплата
   - Имя_ребенка
   - Возраст ребенка
3. Атрибуты
   - Табельный номер (таб. №) Фамилия рабочего (фио) Цех
   - Участок
   - Дата
   - Сумма зарплаты (сумма)
4. Атрибуты
   - Магазин
   - Книга
   - Цена
   - Издательство
   - Дата
   - Количество продано (кол-во)
5. Атрибуты
   - Отправитель
   - Получатель
   - Адрес получателя (адрес)
   - Изделие
   - Цена
   - Кол-во на месяц (кол-во)
6. Атрибуты

- Аэропорт отправления
- Номер рейса (№ рейса)
- Количество мест (кол-во)
- Бортовой № самолета
- Пункт назначения
- Дата вылета (дата)

7. Атрибуты
   - Фио студента
   - Дата поступления в вуз
   - Факультет
   - Группа
   - Место работы
   - Зарплата
8. Атрибуты
   - № больницы
   - № палаты
   - Пациент
   - Домашнии адрес
   - Лечащии врач (врач)
   - Диагноз
9. Атрибуты
   - Дата матча (дата)
   - Команда-хозяин
   - Команда-гость
   - Счет матча
   - Число очков в чемпионате
10. Атрибуты
    - ФИО студента (фио)
    - Дата поступления в вуз (дата)
    - Факультет
    - Группа
    - Общественная работа
11. Атрибуты
    - Фио служащего (фио)
    - Тема работы
    - Источник финансирования темы (фин)
    - Отдел
    - Учреждение
12. Атрибуты
    - Порт
    - Судно
    - Грузоподъемность
    - Дата отплытия
    - Порт назначения
13. Атрибуты
    - Дисциплина
    - Преподаватель
    - Время занятий
    - Аудитория

- ФИО студента (фио)

14. Атрибуты
    - Учреждение
    - Отдел
    - Тема
    - Код оборудования
    - ФИО сотрудника (фио)
    - Продолжительность работы
15. Атрибуты
    - Отдел
    - ФИО сотрудника (фио)
    - Номер комнаты
    - Телефон
    - Тема работы
    - Продолжительность работы

Семинар № 5
Базовые возможности языка SQL

Цель: Научиться использовать язык SQL в среде конкретной СУБД

Содержание занятия:

При помощи языка SQL:

1. Создание новой БД.
1. Создание структуры таблицы.
2. Удаление таблицу.
3. Изменение структуры таблицы.
4. Создание многотабличную БД со связями.
5. Создание, удаление и изменение индексов.
6. Заполнение базы данных. Создание представления.
7. Создание внешних процедур и триггеров.
8. Отработка оператора выборки данных на готовой базе данных (созданной в предыдущих пунктах).
9. Отработка оператора добавления записей в таблицу.
10. Отработка оператора удаления записей из таблицы.
11. Отработка оператора обновления записей.
12. Отработка оператора определения прав доступа и отмены прав доступа.

Работа в среде конкретной СУБД.

Вопросы для самостоятельной работы студентов:

1. История развития SQL
1. Подмножества языка SQLю
2. Типы данных SQL.

Решение практических задач.

1. Создание таблиц БД с помощью оператора CREATE TABLE.
1. Создание индексов.
2. Выборка данных.

Варианты заданий:

Запрос 1

Рассмотрим пример создания таблицы базы данных, учитывающей сотрудников предприятия. Положим, что таблица Employees (Сотрудники) будет содержать следующие поля: код сотрудника (E_KOD), фамилия (E_FAM), имя (E_NAM), дата рождения (E_DATE) и стаж работы (E_STAG). Создадим описание таблицы Employees.

Выйдем на окно редактора запросов на языке SQL. В редакторе запишем запрос на языке SQL на создание таблицы Employees:

```
CREATE TABLE Employees (
        E_KOD           CHAR (4) NOT NULL,
        E_FAM           CHAR (30),
        E_NAM       CHAR (30),
        E_DATE      DATE,
        E_STAG      INTEGER );
```

Выполним созданный запрос и убедимся, что в списке таблиц появилась новая таблица (Employees). Вызвав режим «Конструктор таблиц», проверим правильность описания полей. Отметим, что в созданной таблице не указано ключевое поле.

Запросы на удаление таблиц

Запрос 2

Удалите таблицу Employees из базы данных. Убедитесь, что таблица удалена

Запрос 3

Усложним запрос на создание таблицы Employees, введя ограничение, которое определяет поле E_KOD как ключевое («первичный ключ»):

```
CREATE TABLE Employees
        (E_KOD CHAR (4) NOT NULL PRIMARY KEY ,
        E_FAM CHAR (30),
        E_NAM CHAR (30),
        E_DAT DATE,
        E_STAG INTEGER);
```

Пример описания базы даны «Автосалон»

База данных «Автосалон» (avtoshop.mdb) предназначена для регистрации продаж в автосалоне. База данных avtoshop.mdb состоит из 4-х таблиц: «Продавцы» (salespeople), «Покупатели» (customers), «Цена» (price) и «Заказы»(orders). Описание каждой таблицы на языке SQL приведено ниже:

Запрос 4

```
create table Salespeople
( snum          char(4) not null primary key,
  sname         char(30) not null,
  saddress      char(60),
  comm          float ,
  stel      char(20),
  semail        char(40) );
```

Запрос 5

```
create table Customers
( cnum          char(4) not null primary key,
  cname         char(30) not null,
  caddress      char(60),
  rating    int ,
  ctel      char(20),
  cemail        char(40) );
```

Запрос 6

```
create table Price
( pnun      char(4) not null primary key,
  pname         char(60) not null,
  peddim        char(10),
```

```
 price      float ,
 pdate      datetime Not null );
```

Запрос 7

```
create table Orders
( onum              char(4) not null primary key,
 odate      datetime Not null,
 amount             float ,
 snum               char(4) not null,
 cnum               char(4) not null,
 pnun       char(4) not null );
```

Запросы на создание индексов в таблицах

Запрос 8 (запросы 8.1-8.4)

Создайте в каждой таблице по одному индексу. Обоснуйте свой выбор индексов. Убедитесь, что индексы созданы! Для этого посмотрите таблицы в режиме конструктора.

Запрос 9

Удалите один индекс.

Описание связей между таблицами

Модификация структуры таблиц

Запрос 10

Добавьте столбец NewPrice в таблицу Price:

Запрос 11

Удалите столбец NewPrice из таблицы Price:

```
ALTER TABLE Price DROP COLUMN NewPrice
```

Запрос 12

Создайте связь между таблицами Price и Orders.

Запрос 13

Создайте связь между таблицами SalesPeople и Orders.

Запрос 14

Создайте связь между таблицами Customers и Orders.

II Запросы DML – манипулирование данными (повторение)

Запросы на ввод данных в таблицы

Запросы 15.1, 15.2 и т.д.

При помощи операторов INSERT INTO заполните все таблицы данными.

Запрос 16

Добавьте нового продавца (т.е. вставить новую запись в таблицу Salespeople) со следующими параметрами: код продавца - 1010, фамилия - Бояринов, адрес - Москва, комиссионные - 0.12, телефона - нет, почта - bojar @yandex.ru.

Запрос 17

```
INSERT INTO Customers (cname, cnum)
        VALUES ("Синичкин", "2007");
```

В результате выполнения такого запроса в таблицу покупателей добавится новая строка с кодом «2007» (cnum ="2007"), у которой будет заполнен только один столбец cname (cname="Синичкин").

Запросы на изменение данных в таблицах

Запрос 18

Всем покупателям, которые живут в городе «Москва», изменить рейтинг на 125.

Запросы на удаление данных из таблиц

Аналогичным образом строятся и запросы на удаление записей из таблицы. Общая форма запроса на удаления имеет вид:

```
DELETE *
FROM <имя таблицы>
```

WHERE <условия отбора записей>

Запрос 19

Вставить новую сроку в таблицу Price

Запрос 20

Изменить цены в таблице Price для автомобилей определенной марки.

Запрос 21

Удалить все заказы с определенной датой.

Запрос 22

Вставить запись для нового заказа.

Запрос 23

Удалить всех продавцов, которые не обслужили ни одного заказа (увольнение неработающих продавцов).

III Запросы DQL – выборка данных

Построение запросов на выборку на SQL к одной таблице (повторение)

Запрос 24

Вывести список продавцов, которые живут в городе Москве.

Запрос 25

Вывести номер, имя и адрес эл. почты продавцов. В результате вместо имен полей должны стоять русские названия столбцов (переименование имен столбцов - конструкция AS).

Замечание. Использование синонимов столбцов (с помощью ключевого слово AS), позволяет разработчикам баз данных не использовать кириллицу в именах столбцов на этапе физического проектирования базы данных.

Запрос 26

Отобрать все товары (т.е. строки из таблицы price), которые имеют цену больше или равную 3000 и зарегистрированы позднее 14 декабря 2005г..

Запрос 27

Надо отобрать все строки из прайса для автомобилей «Волга» и «Ауди»

Запрос 28

Например, надо отобрать строки из таблицы Price, от продавцов 3001 и 3003, зарегестрированных между 14 и 16 декабрем 2005г, цена которых превышает 3000.

Запрос 29

В запросы можно вставлять текстовые столбцы, которые облегчают понимание результатов. Например, в запросе ниже вставлено два текстовых столбца: «Наименование» и «цена». Этот текст будет повторяться в каждой строке результирующей таблицы.

Пример SQL запроса с текстовыми столбцами:

```
SELECT pnum AS НОМЕР, 'Наименование' , pname AS НАЗВАНИЕ, 'Цена' , price AS ЦЕНА
FROM Price
WHERE pnum ='3003' ;
```

Запрос 30

Вычислить стоимость товара с НДС (налог на добавленную стоимость), который равен 20% от цены. Иначе говоря [СТОИМОСТЬ С НДС]=ЦЕНА*1.2..

Запрос 31

Вывести все столбцы из таблицы Customers, для тех покупателей, которые не живут в городе Москва и имеют рейтинг меньше 300.

Запрос 32

Упростите и проинтерпретируете запрос:

```
SELECT orders.onum, *
FROM orders
```

WHERE ((((orders.odate = #1/1/2006#)OR (orders.odate =#1/2/2006#))) AND (orders.onum="4001"));

Запрос 33

Приведите пример запроса с измененными названиями столбцов и с вставленными текстовыми столбцами.

Запрос 34

Напишите запрос с вычисляемыми столбцами и с использованием функций

Семинар № 6

Манипулирование данными в реляционной модели. Реляционная алгебра

Цель:

Научиться использовать операторы реляционной алгебры.

Содержание занятия:

1. Применение операторов реляционной алгебры на придуманных таблицах.
2. Виды соединений и их отличия.
3. Перевод основных операторов реляционной алгебры в запросы SQL.

Работа в среде конкретной СУБД.

Вопросы для самостоятельной работы студентов:

1. Перечислите все операторы реляционной алгебры.
2. Замкнутость реляционной алгебры.
3. Запросы, нереализуемые средствами реляционной алгебры

Решение практических задач.

1. Теоретико-множественные операторы реляционной алгебры.
2. Специальные реляционные операторы

Примеры практических задач:

Теоретико-множественные операции реляционной алгебры

Пример 1.

Исходные отношения R_1 и R_2 содержат перечни товаров, находящихся соответственно на первом и втором складах.

R1	
Артикул	Товар
01	Intel i3
02	Intel i5
03	Intel i7
04	HDD 320GB
05	HDD 750GB
06	DDR3 1Gb
07	DDR3 4Gb

R2	
Артикул	Товар
01	Intel i3
03	Intel i7
08	Intel Core Duo
09	HDD 500GB
05	HDD 750GB

Задание 1.1 Объединение

Построить отношение R_3 содержащее общий перечень товара на складах, то есть характеризует общую номенклатуру складов.

Задание 1. 2. Пересечение

Построить отношение R_4 содержащее перечень товара, который есть в наличии одновременно на двух складах.

Задание 1.3. Разность

Построить отношение R_5 содержащее перечень товара, находящегося только на складе 1 и отношение R_6 содержащее перечень товара, находящегося только на складе 2 и написать соответствующие формулы.

Пример 2. Рассмотрим пример из другой предметной области. Исходными являются три отношения R21, R22 и R23- Все они имеют эквивалентные схемы.

R21= (ФИО, Паспорт, Школа);

R22= (ФИО, Паспорт, Школа);

R23= (ФИО, Паспорт, Школа).

Ситуация была характерна для периода, когда были разрешены так называемые репетиционные вступительные экзамены, которые сдавались раньше основных вступительных экзаменов в вуз. Отношение R21 содержит список абитуриентов, сдававших репетиционные экзамены. Отношение R22 содержит список абитуриентов, сдававших экзамены на общих условиях. И наконец, отношение R23 содержит список абитуриентов, принятых в институт. При неудачной сдаче репетиционных экзаменов абитуриент мог делать вторую попытку и сдавать экзамены в общем потоке, поэтому некоторые абитуриенты могут присутствовать как в первом, так и во втором отношении.

Задание. Записать формулы, дающие ответы на следующие вопросы:

2.1. Список абитуриентов, которые поступали два раза и не поступили в вуз.

2.2. Список абитуриентов, которые поступили в вуз с первого раза, то есть они сдавали экзамены только один раз и сдали их так хорошо, что сразу были зачислены в вуз.

2.3. Список абитуриентов, которые поступили в вуз только со второго раза.

2.4. Список абитуриентов, которые поступали только один раз и не поступили.

Задание 1.4. Расширенное декартово произведение отношений

Пусть в отношении R7 задана обязательная номенклатура товаров для всех складов, а в отношении R8 дан перечень всех складов.

R7	
Артикул	Товар
01	Intel i3
02	Intel i5
03	Intel i7
04	HDD 320GB
05	HDD 750GB
06	DDR3 1Gb
07	DDR3 4Gb
08	Intel Core Duo
09	HDD 500GB
11	HDD 1TB
10	DDR3 2Gb

R8
Склад
Склад 1
Склад 2
Склад 3

Построить отношение R9, которое соответствует ситуации, когда каждый склад хранит *все* товары из перечня (Какая это операция?)

Задание 1.5

Пусть отношение R10, характеризует реальное хранение товаров на каждом складе.

В отношении R11 отобразить какие товары на каких складах из общей обязательной номенклатуры не хранятся.

R10		
Артикул	Товар	Склад
01	Intel i3	Склад 1
02	Intel i5	Склад 1
03	Intel i7	Склад 1
04	HDD 320GB	Склад 1
05	HDD 750GB	Склад 1
06	DDR3 1Gb	Склад 1

07	DDR3 4Gb	Склад 1
08	Intel Core Duo	Склад 1
09	HDD 500GB	Склад 1
05	HDD 750GB	Склад 2
05	DDR3 1Gb	Склад 2
07	DDR3 4Gb	Склад 2
08	Intel Core Duo	Склад 2
09	HDD 500GB	Склад 2
05	HDD 750GB	Склад 2
10	DDR3 2Gb	Склад 2
01	Intel i3	Склад 3
02	Intel i5	Склад 3
03	Intel i7	Склад 3
04	HDD 320GB	Склад 3
05	HDD 750GB	Склад 3
06	DDR3 1Gb	Склад 3
07	DDR3 4Gb	Склад 3
08	Intel Core Duo	Склад 3
11	HDD 1TB	Склад 3
05	HDD 750GB	Склад 3
11	HDD 1TB	Склад 1
05	HDD 750GB	Склад 1
10	DDR3 2Gb	Склад 1

Задание 1.6 . Группа теоретико-множественных операций избыточна. Как можно записать операцию пересечения через объединение и разность? (или показать взаимосвязь любых других операций)

Специальные операции реляционной алгебры

Задание 1.7 . В отношение R12 выбрать из R10 детали с шифром «05».

Задание 1.8. Выбрать все склады, которые хранят деталь «HDD 320GB». (Построить отношения R13, R14)

Задание 1.9-1.11. Даны отношения RD1-RD4

RD1 Задания			
№	Текст	Пользо ватель	Срочное
1	Задание 1	00011076	1
2	Задание 2	00011075	0
3	Задание 3	00011073	0
4	Задание 4	00013062	1
5	Задание 5	00011003	1
6	Задание 6	00013062	1
7	Задание 7	00011075	0

RD4 Часовые пояса	
Город	Часовой пояс (UTC+X)
Москва	4
Санкт-Петербург	4
Лондон	0
Нью-Йорк	-5
Владивосток	11

Задание 1.9. Найти подразделения, находящиеся западнее Москвы.

Задание 1.10. Найти пользователей из Москвы, имеющих задания.

Задание 1.11. Найти пользователей, имеющих срочные задания, у которых в данный момент столько же времени, что и у вас.

Операция условного соединения (бинарная операция)

Задание 1.12. Пусть отношение R_{15} содержит перечень товаров с указанием упаковки. Получить перечень товара, которые находятся на складе 1 в упаковке «OEM»

R15		
Артикул	Товар	Упаковка
0001	Intel i3	OEM
0002	Intel i5	ODM
0003	Intel i7	OEM
0004	HDD 320GB	Full
0005	HDD 750GB	Full
0006	DDR3 1Gb	OEM
0007	DDR3 4Gb	OEM
0008	Intel Core Duo	ODM
0009	HDD 500GB	Full
0011	HDD 1TB	Full
0010	DDR3 2Gb	OEM

Операция деления.

Задание 1.10. Используя отношения R_7, и R_{10} определить перечень складов (отношение R17), в которых хранится вся номенклатура деталей.

Семинар № 7
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 ДИСЦИПЛИНЫ

Цель дисциплины: профессиональная подготовка студентов, необходимая для освоения методов и технологий формирования современных баз данных, являющихся основой любой информационной системы, создаваемой в любой сфере человеческой деятельности.

Задачи:

- изучить типологии и методологии баз данных, современные модели баз данных;
- усвоить методы классификации и моделирования предметных областей, методы проектирования баз данных с помощью современных технологий;
- получить навыки работы с инструментальными средствами проектирования баз данных, использования стандартов информационных технологий, разработки технологической документации, сопровождающей процесс создания баз данных.

В результате освоения дисциплины обучающийся должен:

Знать модели данных; архитектуру БД; системы управления БД и информационными хранилищами; методы и средства проектирования БД; особенности администрирования БД в глобальных и локальных сетях.

Уметь проводить анализ предметной области, выявлять информационные потребности и разрабатывать требования к ИС; проводить сравнительный анализ и выбор ИКТ для решения прикладных задач и создания ИС; разрабатывать концептуальную модель прикладной области; выбирать инструментальные средства и технологии проектирования ИС; проводить формализацию и реализацию решения прикладных задач; выполнять работы на всех стадиях жизненного цикла проекта ИС, оценивать качество и затраты проекта.

Владеть навыками работы с инструментальными средствами (в том числе отечественного производства) моделирования предметной области, прикладных и информационных процессов, проектирования баз данных, использования стандартов информационных технологий, разработки технологической документации, сопровождающей процесс создания и инсталляции информационных систем.