МИНИСТЕРСТВО НАУКИ, ВЫСШЕГО ОБРАЗОВАНИЯ И ИННОВАЦИЙ КР
КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. И.АРАБАЕВА
ОСПО ИНСТИТУТА НОВЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

«УТВЕРЖДАЮ»
Директор ИНИТ
КГУ им. И. Арабаева
к.т.н., доц. Керимов У.Т.
_____
_____ 2025г.

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС

*по дисциплине :*_____**База данных**_

для студентов специальности: _230109 «Программное обеспечение вычислительной техники и автоматизированных систем», 230701 «Прикладная информатика (по отраслям)»

форма обучения: очное\ заочное

Учебно-методический комплекс составлен на основе Государственного Образовательного Стандарта среднего профессионального образования КР

Учебно-методический комплекс разработала, преподаватель отделения СПО ИНИТ КГУ имени И. Арабаева Осмонова Назира Шамшыбековна

Соавторы: Омурбек кызы Асель

Бишкек 2025г.

СОДЕРЖАНИЕ УМК

МИНИСТЕРСТВО НАУКИ, ВЫСШЕГО ОБРАЗОВАНИЯ И ИННОВАЦИЙ КР
КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. И.АРАБАЕВА
ОСПО ИНСТИТУТА НОВЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

«УТВЕРЖДАЮ»
Директор ИНИТ
КГУ им. И. Арабаева
к.т.н., и. о. доц. У.Т. Керимов
«01» 09 2025г.

РАБОЧАЯ ПРОГРАММА

по дисциплине: **«База данных»**
для студентов специальности: 230109 «Программное обеспечение вычислительной техники и автоматизированных систем», 230701 «Прикладная информатика (по отраслям)».
форма обучения: очное/заочное
институт: ИНИТ
отделение: ОСПО ИНИТ
курс: 2,3
семестр: 4/5
экзамен (семестр): 3/5
всего часов по учебному плану: 90
из них:
-лекции: 54
-практические: 36
-самостоятельная работа: 24

Рабочая программа составлена в соответствии с требованиями Государственного Образовательного Стандарта среднего профессионального образования КР

Рабочую программу разработала: преподаватель отделения СПО ИНИТ КГУ имени И. Арабаева Осмонова Назира Шамшыбековна
Соавторы: Омурбек кызы Асель

Рассмотрена и утверждена на заседании
ОСПО ИНИТ КГУ им. И. Арабаева
Протокол № 1
от «01» 09 2025г.

Зав. отделением: Н.С. Сейтказиева

Одобрено учебно-методическим советом
ИНИТ КГУ им. И. Арабаева
Протокол № 1
от «04» 09 2025г.

Председатель УМС ИНИТ: _____

Бишкек 2025г.

1.Цели и задачи изучения дисциплины, ее значение в учебном процессе

1.1. Цели и задачи изучения дисциплины

владеть технологией проектирования и управления базами данных;

- владеть основными методами и способами разработки алгоритмов поставленных задач, а также средствами разработки программных приложений;
- использовать информационно-коммуникационные технологии в профессиональной деятельности;

1.2. Ожидаемые результаты изучения дисциплины

В результате изучения курса «Базы данных» и по его успешному завершению в стенах колледжа, студент должен приобрести следующие компетенции: В результате изучения дисциплины студент должен:

знать:

- основные положения теории баз данных, хранилищ данных, баз знаний; − основные принципы построения концептуальной, логической и физической модели данных;
- современные инструментальные средства разработки схемы базы данных; − структуры данных СУБД, общий подход к организации представлений, таблиц и индексов;
- методы организации целостности данных;
- способы контроля доступа к данным и управления привилегиями; модели и структуры информационных систем.

уметь:

- создавать объекты базы данных в современных системах управления базами данных и управлять доступом к этим объектам;
- формировать и настраивать схему базы данных;
- применять стандартные методы для защиты объектов баз данных.

владеть:

- работы с объектами базы данных в конкретной системе управления базами данных;
- использования средств заполнения базы данных;
- использования стандартных методов защиты объектов баз данных.

2.Компетенции по Госстандарту.

Выпускник в соответствии с целями основной профессиональной образовательной программы и задачами профессиональной деятельности, указанными в пунктах 11 и 15 настоящего Государственного образовательного стандарта, должен обладать следующими компетенциями:

а) общими (ОК):

ОК-1. Уметь организовывать собственную деятельность, выбирать методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК-2. Решать проблемы, принимать решения в стандартных и нестандартных ситуациях, проявлять инициативу и ответственность.

ОК-3. Осуществлять поиск, интерпретацию и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК-4. Использовать информационно – коммуникационные технологии в профессиональной деятельности.

ОК-5. Уметь работать в команде, эффективно общаться с коллегами, руководством, клиентами.

ОК-6. Брать ответственность за работу членов команды (подчиненных), за результат выполнения заданий.

ОК-7. Управлять собственным личностным и профессиональным развитием, адаптироваться к изменениям условий труда и технологий в профессиональной деятельности.

ОК-8. Быть готовым к организационно – управленческой работе с малыми коллективами. ОК-9. Способен приобретать новые знания, с большой степенью самостоятельности, с использованием современных образовательных и информационных технологий.

ОК-10. Способен на научной основе оценить свой труд, оценивать с большой степенью самостоятельности, результаты своей деятельности.

230109 – «Программное обеспечение вычислительной техники и автоматизированных систем»

б) профессиональными, соответствующими основным видам профессиональной деятельности (ПК):

производственно-технологическая деятельность:

ПК-1. Владеет знаниями об архитектуре и технических характеристиках персональных компьютеров;

ПК-8. Способен осуществлять модификацию, адаптацию и настройку программных продуктов;

ПК-11. Владеет знаниями о правилах и нормах охраны труда, техники безопасности, промышленной санитарии и противопожарной защиты.

ПК-13. Способен реализовать функции сопровождения программных продуктов;

ПК-16. Способен обеспечивать эффективное применение пакетов прикладных программ;

230701 – «Прикладная информатика (по отраслям)»

ПК-6 - осуществлять сбор и анализ информации для определения потребностей клиента.

ПК-12 - осуществлять продвижение и презентацию программного обеспечения отраслевой направленности.

3.Межпредметные связи. Перечень дисциплин и их разделов, усвоение которых необходимо при изучении данной дисциплины.

Пререквизиты - это дисциплины, содержащие знания, умения и навыки, необходимые для освоения изучаемой дисциплины, соответственно до изучения данной дисциплины определяют ее преемственность. Пререквизитами данного курса являются знания студентов базовых основ информатики.

Постреквизиты: Постреквизиты — это дисциплины, для изучения которых требуются знания, умения и навыки, приобретаемые по завершении курса. Постреквизитами данного курса являются знания основных понятий информатики, технические и программные средства реализации информационные процессов, а также теоретические дисциплины согласно учебному плану.

4.Структура дисциплины с разбивкой по видам занятий, часам и модулям

№	Наименование раздела, тем лекционных занятий	Кол-во часов	Примечание
	Модуль 1		
1	**Введение**. Основные понятие теории баз данных	2	
2	Архитектура базы данных	2	
3	Введение в банки данных. OLAP и OLTP-системы	4	
4	Принцип построение БД. Инфологическое (концептуальное) моделирование предметной области. Проектирование реляционных БД на основе принципов нормализации	6	
5	Информационные модели. Взаимосвязи в моделях и реляционный подход к построению моделей.	6	
6	**Нормализация отношений.** Этапы проектирования баз данных	4	
7	**Работа с объектами**(таблицы, запросы, формы, отчеты и макросы)	6	
8	**Классификация и характеристика** СУБД. Обзор современных СУБД.	4	
9	Интерфейс программы. Основы работы в Microsoft Access	2	
10	Создание таблиц и форм в Access	2	
11	Работа с запросами в Access	2	
12	Таблицы и связи в Access	2	
13	Создание отчетов и форм в Access	2	
14	Практическое проектирование базы данных в Access	4	
15	Безопасность данных в Access	2	
16	Введение в SQL. Краткая характеристика языка SQL.	4	
	ВСЕГО:	**54**	
	ИТОГО за год	**54**	

№	Наименование тем практических занятий, содержание	Кол-во часов	Примечание
	Модуль 2		
1.	Создание новой базы данных в Microsoft Access	***2***	
2.	Создание таблиц и определение типов полей.	***2***	
3.	Создание таблиц в режиме конструктора. Создание форм. Сортировка и отбор данных.	4	
4.	Многотабличные базы данных Access. Связывание данных таблиц	4	
5.	Создание запросов и поиск информации.	4	
6.	Многотабличные базы данных Access. Многотабличные запросы.	4	
7.	Сложные (подчиненные) формы Access. Отчеты для многотабличные баз данных Access	4	
8.	Создание отчетов и печать данных.	4	
9.	Мини-проект: “Студенческая база данных”, “База данных колледжа”, “База данных библиотека”, или “База рабочих кадров"	8	
	Всего за I полугодие	**36**	
	ИТОГО за год	**36**	

№	Наименование раздела, тем лекционных занятий	Кол-во часов	Примечание
	Модуль 1		
1	**Введение**. Основные понятие теории баз данных	2	
2	Архитектура базы данных	2	
3	Введение в банки данных. OLAP и OLTP-системы	2	
4	Принцип построение БД. Инфологическое (концептуальное) моделирование предметной области. Проектирование реляционных БД на основе принципов нормализации	4	
5	Информационные модели. Взаимосвязи в моделях и реляционный подход к построению моделей.	4	
6	**Нормализация отношений.** Этапы проектирования баз данных	2	
7	**Работа с объектами**(таблицы, запросы, формы, отчеты и макросы)	4	

8	**Классификация и характеристика** СУБД. Обзор современных СУБД.	2	
	ВСЕГО:	**22**	
	ИТОГО за год	**22**	

№	**Наименование тем практических занятий, содержание**	**Кол-во часов**	**Примечание**
	Модуль 2		
1.	Создание новой базы данных в Microsoft Access	***2***	
2.	Создание таблиц и определение типов полей.	***2***	
3.	Создание таблиц в режиме конструктора. Создание форм. Сортировка и отбор данных.	2	
4.	Многотабличные базы данных Access. Связывание данных таблиц	2	
5.	Создание запросов и поиск информации.	2	
6.	Многотабличные базы данных Access. Многотабличные запросы.	2	
7.	Мини-проект: “Студенческая база данных”, “База данных колледжа”,	2	
	Всего за I полугодие	**14**	
	ИТОГО за год	**14**	

№	**Наименование раздела, тем лекционных занятий**	**Кол-во часов**	**Примечание**
	Модуль 1		
1	**Введение**. Основные понятие теории баз данных	2	
2	Архитектура базы данных	2	
3	Введение в банки данных. OLAP и OLTP-системы	4	
4	Принцип построение БД. Инфологическое (концептуальное) моделирование предметной области. Проектирование реляционных БД на основе принципов нормализации	6	
5	Информационные модели. Взаимосвязи в моделях и реляционный подход к построению моделей.	6	
6	**Нормализация отношений.** Этапы проектирования баз данных	4	
7	**Работа с объектами**(таблицы, запросы, формы, отчеты и макросы)	6	
8	**Классификация и характеристика** СУБД. Обзор современных СУБД.	4	
	ВСЕГО:	**34**	

	ИТОГО за год	**34**	

№	Наименование тем практических занятий, содержание	Кол-во часов	Примечание
	Модуль 2		
1.	Создание новой базы данных в Microsoft Access	***2***	
2.	Создание таблиц и определение типов полей.	***2***	
3.	Создание таблиц в режиме конструктора. Создание форм. Сортировка и отбор данных.	2	
4.	Многотабличные базы данных Access. Связывание данных таблиц	2	
5.	Создание запросов и поиск информации.	4	
6.	Многотабличные базы данных Access. Многотабличные запросы.	4	
7.	Мини-проект: “Студенческая база данных”, “База данных колледжа”, “База данных библиотека”, или “База рабочих кадров"	4	
	Всего за I полугодие	**20**	
	ИТОГО за год	**20**	

Структура дисциплины заочного отделения

№	Наименование раздела, тем лекционных занятий для заочной формы обучен	Кол-во часов	Примечание
1	**Введение**. Основные понятие теории баз данных	2	
2	Архитектура базы данных	2	
3	Введение в банки данных. OLAP и OLTP-системы	2	
4	**Классификация и характеристика** СУБД. Обзор современных СУБД.	2	
	ВСЕГО:	**8**	
	ИТОГО за год	**8**	

№	Наименование тем практических занятий для заочной формы обучение	Кол-во часов	Примечание
1.	Создание новой базы данных в Microsoft Access	***2***	
2.	Создание запросов и поиск информации.	***2***	
3.	Мини-проект: “База данных колледжа”	2	
	Всего за I полугодие	**6**	
	ИТОГО за год	**6**	

5.Темы для самостоятельной работы студентов.

Студент выполняет самостоятельную работу для закрепления теоретических знаний. Суть работы – письменные иллюстрированные схемами ответы на вопросы. Объем 5-25 страниц.

СРС по модулям

	Название тем	**Часы**	**Сроки выполения**
МОДУЛЬ 1			
1	Средства автоматизации проектирование	6	2,3 неделя
2	Реляционная алгебра	9	4-6 неделя
3	Публикация баз данных в Internet	9	6,7 неделя
4	Современные СУБДи их применение(MS Access, MS SQLServer и др.)	9	8 неделя
МОДУЛЬ 2			
6	Объектно-ориентированное программированные СУБД	9	8-11 неделя
7	Системы баз данных, основанные направления	9	12-14-неделя
8	Основные категории команд SQL	9	15 –неделя
	ВСЕГО:	60	

6.Распределение баллов по модулям и видам учебных занятий

№	**Этапы проверки**	**Вид средства проверки**	**Баллы**
1	1 модуль	Проверка практических заданий. Устный, тестирование. Посещение занятий.	100
2	2 модуль	Проверка практических заданий. Тестирование. Посещение занятий.	100
3	Итоговый контроль: • Практическое занятие; • СРС.	Контрольные и графические работы, рефераты, презентации, СРС, практические задания. Тестирование. Посещение занятий.	100
	Итого средний балл		**100**

Итоговое распределение баллов по модулям

		Удовлетворительно	Хорошо	Отлично
Модуль 1 – 100 б.		60-79	80-89	90-100
Модуль 2 – 100 б.		60-79	80-89	90-100
Практическое занятие – 50 б.	Итоговый контроль	60-79	80-89	90-100
СРС – 50 б.				

7.Список литературы

Список используемой литературы **Основная:**

№	**Библиографическое описание издания (автор, наименование, вид, место и год издания, кол. стр.)**
1	Введение в системы баз данных, 8-е издание К. Дж. Дейт, 1328 стр., с ил.; ISBN 5-8459-0788-8, 0-321-19784-4;
2	Базы данных. Учебник для высших учебных заведений, под редакцией проф. А. Д. Хомоненко, Санкт Петербург, Корона, 2004г.
3	Access 2003. базы данных и приложения, лекции и упражнения. Борис Послед, изд-во. «ДиаСофт», 2004.
4	SQL Server 2008. Степанов А.Н. - СПб. 2010.
5	Базы данных. Модели данных. Учебник – М.: ООО. С.Д.Кузнецов. «Бином-Пресс», 2018 г.
6	Базы данных. Язык SQL для студента. 2-е издание. В.В.Дунаев– СПб.: БХВПетербург, 2019.
7	Базы данных: разработка и управление. Гэри Хансен, Джеймс Хансен. Москва, ЗАО «Издательство БИНОМ», 1999
8	Основы современных баз данных. Информационно-аналитические материалы. С. Д. Кузнецов
9	Структурированный язык запросов (SQL). Учебное пособие. В. В. Кириллов, Г. Ю. Громов, WWW. CITFORUM. RU
10	Основы использования www - технологий для доступа к существующим базам данных. Е. Фадденков
11	Основы проектирования реляционных баз ДАННЫХ. Учебное пособие. В. В. Кириллов

7.1.Интернет-ресурсы

1. Сайт Национального открытого университета ИНТУИТ www.intuit.ru
2. Документация по MySQL http://www.mysql.ru/docs/
3. MySQL - справочное руководство на русском https://phpclub.ru/mysql/doc/
4. Открытое образование, курс «Базы данных» https://openedu.ru/course/spbu/DTBS/

КОНТРОЛЬНЫЕ ВОПРОСЫ:

1) Основные понятия теории баз данных

2) Виды информационных моделей: информационная модель данных и информационная модель предприятия.

3) Информационная модель данных. Концептуальная, логическая, внешняя и физическая модели. Условия первого и второго уровней независимости данных

4) Типы логических моделей: иерархическая, сетевая и реляционная

5) Основы реляционной алгебры

6) Взаимосвязи в моделях и реляционный подход к построению моделей

7) Нормализация отношений. Условия 1,2 и 3 НФ

8) Требования, предъявляемые к базе данных

9) Этапы проектирования баз данных

10) Классификация СУБД

11) Основные характеристики СУБД

12) Современные системы управления базами данных: Access, SQL-Server

13) Основные направления совершенствования баз данных. Постреляционные СУБД.

14) Способы создания таблицы

15) Открытие таблицы, модификация структуры таблицы

16) Добавление новой записи в таблицу

17) Удаление записи из таблицы

18) Создание и открытие базы данных

19) Добавление таблиц в базу данных, освобождение таблиц

20) Установление взаимосвязи «один-к-одному»

21) Установление взаимосвязи «один-ко-многим»

22) Управление данными в базах данных

23) Типы реляционных языков

24) Язык запросов SQL

1.

Тест по дисциплине БАЗЫ ДАННЫХ

Вариант №1 ***Тесты***
по дисциплине «Базы данных»

1.Уровни проектирования БД

А. Первый, второй и третий

Б. Проектировочный и пользовательский

В. Реляционный, иерархический и сетевой

Г. Конструкторский и мастер

2.База данных – это

А. Совместно используемое единое хранилище для некоторого набора логически связанных данных по определенной предметной области.

Б. это представление реальности, отражающего лишь избранные детали.

В. хранения специально организованной информации по определенной предметной области

Г. это значение, которое однозначно определяет элемент объектного множества.

3. Внешний ключ –это

А. Древовидные иерархические структуры широко используются в повседневной человеческой деятельности Б. Идентификатор

В. Система баз данных

Г. Набор лексических атрибутов, значения которых всегда однозначно определяют элемент объектного множества

4. Сетевые модели данных

А. организует и представляет данные в виде таблиц или реляций.

Б. базируются на использовании графовой формы представления данных. Вершина графа используются для интерпретации типов сущностей

В. приведения реляционных таблиц к стандартному виду.Правила нормализации: Г. Противоречивость данных, вызванная их избыточностью и частичным обновлением.

5. язык DDL (Data Definition Language) это

А. предназначенный, для выборки и обновления данных.

Б. разделения таблицы на несколько таблиц в целях избавления от аномалий и поддержания целостности данных

В. Повторение данных в базе данных

Г предназначенный для определения структур базы данных;

6. В базе данных используется следующие способы создания таблиц в MS ACCESS: А.через Конструктор;

Б Через Мастер;

В через поле вводе данных; Г все вышеперечисленные

Группа_____ФИО_____

1.

7. Запросы служат:

А. Для выбора более чем одной таблицы одновременно
Б. служат для селекции и фильтрации набора данных
В. выполняет роль хранилища данных
Г. Вывода данных на экран

8. Типы данных MS ACCESS: А. Запросы, таблицы и формы
Б. Отчеты, макросы и модули
В. Тестовый, числовой, логический, дата/время, OLE, счетчик
Г. Конструктор и мастер

9. Макросы - это

А. последовательность операций, записанных в виде инструкций на специальном языке.
Б. MS ACCESS
В. Базы данных
Г. совокупность объявлений (деклараций) и последовательностей исполняемых команд

10. Нормализация.

А. нормальные формы
Б. Согласованность данных в базе данных
В. избавления от аномалий и поддержания целостности данных
Г. Процесс приведения реляционных таблиц к стандартному виду.

Тесты по дисциплине «Базы данных»

ВАРИАНТ№2

1. Главными элементами модели данных

А. запросы и таблицы
Б. объекты и отношения.
В. Сетевая и реляционная модели
Г. Текстовый и логический

2. Системы базы данных состоят из главных компонента:

А. Аномалия обновления, удаления и ввода
Б. один к одному, один ко многим, и многие ко многим
В. данные, аппаратное обеспечение, программное обеспечение и пользователи. Г. DDL (Data Definition Language) и DML (Data Manipulation Language)

3. Реляционная модель данных

А. Количество атрибутов реляции
Б. это термин, пришедший из математики
В. организует и представляет данные в виде таблиц или реляций.
Г. Количество НЕ атрибутов реляции

4. Модель *«Клиент-сервер»* – технология

1.

А. Графические возможности оболочки производят большое впечатление при изготовлении высококачественных отчетов и распечаток.

Б. технология, разделяющая приложение- СУБД на две части: клиентскую и сервер,

В. Они позволяют выбрать из базы только необходимую информацию, т.е. ту, которая соответствует определенному критерию

Г. когда размеры баз данных велики

5. В данном примере поле адрес ул. Душанбинская №45 относится к типу данных MS Access: А.дата/время

Б.числовой

В.текстовый

Г.логический

6. Сколько форм нормализации:

А.5

Б.4

В.2

Г.3

7.Что означает данная запись: 1-∞

А. связи

Б. один – к - одному

В.связь одни – ко - многим

Г. Многие – ко - многим

8. Реляционная таблица находится в первой нормальной форме (1НФ)

А. если никакие неключевые атрибуты не являются функционально зависимыми лишь от части ключа

Б. если для любой ФЗ: X —> Y X является ключом

В. условие, обеспечивающее независимость атрибутов путем обязательного повторения значений Г. если значения в таблице являются атомарными для каждого атрибута таблицы

9. Последовательная организация файла означает:

А. Записи обычным образом физически упорядочены по значениям первичного ключа

Б. Записи обычным образом физически упорядочены по значениям не первичного ключа

В. что записи упорядочены по значению ключа

Г. в качестве способа адресации записей избавляет от необходимости поддерживать и просматривать индексы **10. SQL – запросы это:**

А. Представляют Ваши данные в компактном суммированном формате.

Б. Пользователь формулирует их с использованием инструкций и функций, выстраивая описание

В. Находят дубликатные записи в выбранной Вами таблице или запросе.

Г. Копируют записи из существующей таблицы в новую и затем удаляют (по желанию) эти записи из таблицы оригинала

Группа_____ФИО_____

1.
Группа_____ФИО_____

Тесты по дисциплине «Базы данных»
ВАРИАНТ№3

1. Рекурсивное отношение это
А. Столбец реляции
Б. значение, которое однозначно определяет элемент объектного множества
В. Древовидные иерархические структуры
Г. Отношение, связывающее объектное множество с ним самим.

2. Составной ключ
А. Любой набор атрибутов, который может быть выбран в качестве ключа таблицы.
Б. Ключ, содержащий два или более атрибута.
В. Потенциальный ключ, выбранный для преимущественного использования в целях однозначного определения строк таблицы Г. Потенциальный ключ

3. Ограничительное условие.
А. Строки реляционной таблицы представляют в базе данных элементы конкретных объектов реального мира или категорий
Б. Список, содержащий имена реляционных таблиц, имена атрибутов, ключевые атрибуты и внешние ключи
В. Правило, ограничивающее значения данных в базе данных. Г. Никакой ключевой атрибут строки не может быть пустым.

4. Процесс приведения реляционных таблиц к стандартному виду - это
А. Стандартизация
Б. Типы данных
В.Схема данных
Г. Нормализация.

5. Повторение данных в базе данных
А. Целостность данных
Б. Избыточность данных
В. Противоречивость данных, вызванная их избыточностью и частичным обновлением
Г. Аномалия удаления

6. Функциональная зависимость.
А. Атрибут (ы) в левой части функциональной зависимости
Б. Никакие неключевые атрибуты не являются функционально зависимыми лишь от части ключа
В. Значение атрибута в кортеже однозначно определяет значение другого атрибута в кортеже Г. Таблица, состоящая из нескольких выбранных атрибутов

1.

7. Транзитивная зависимость.
А. Первая нормальная форма запрещает таблицам иметь неатомарные
Б. если таблица удовлетворяет НФБК
В. Неключевой атрибут функционально зависит от одного или более неключевых атрибутов
Г.если связи между независимыми атрибутами можно исключить

8. Реляционная модель данных
А. Основана на логических отношениях данных
Б. Модель данных, представляющая данные в виде таблиц.
В. Основана на концептуальной модели данных
Г. Реляционная модель данных организует и представляет данные

9.Объекты MS Access:
А.Таблицы, запросы, формы, отчеты
Б. Текстовое поле, логическое поле
В. Атрибуты
Г. Выше перечисленные
10.Создание новых объектов базы данных: объекта формы
А. Лот №Наименование товара, Единица измерения, Склад №
Б. Текстовое поле, логическое поле
В. В столбец, Ленточная, Табличная, Сводная таблица
Г. Выше перечисленные

Тесты по дисциплине «Базы данных»
ВАРИАНТ№4

В данной таблице «Крупа манная» имеет тип данных

Лот №	Наименование товара	Единица измерения	Склад №
101	Сахарный песок	50	1
102	Крупа манная	25	2
103	Крупа гречневая	50	1

А.логический
Б.текстовый
В. числовой
Г. Наименование товара

2.Какое поле может быть ключевым (верхняя таблица):
А.Наименование
Б. Единица измерения
В. Лот№

Группа_____ФИО_____

1.
Г. Склад№

3. Разбиение — это процесс

А. значения в таблице являются атомарными для каждого атрибута таблицы

Б. разделения таблицы на несколько таблиц в целях избавления от аномалий и поддержания целостности данных

В. введенные в атрибут BLDG-ID

Г. идентификатор конкретного
здания **4. Запрос: Отобразить
продажи в октябре**

А. >=01.10.2005 __ And __ <=31.11.2005

Б. Is __ not __ null

В. LIKE __ [введите название]

Г. LIKE __ [введите наименование товара]

5 Запрос: Создать запрос на вычисление суммы к оплате

А. = 0

Б. [Цена отгрузки]*1,05

В. LIKE __ [введите название]

Г. К оплате: [Количество продажи]*[Цена отгрузки]

6.Таблица «Клиенты» включает поля:

А. Лот №, Наименование товара, Единица измерения, Склад №, Дата поставки, Поставщик, Количество поставки, Цена поставки, Цена отгрузки

Б. Название, ИНН, Обращаться к, Должность, Адрес, Телефон, Факс

В. Накладная № , Дата продажи, ИНН , Лот №, Количество продажи

Г. Параметрический, выборка **7. Поле
«Количество поставки» имеет тип
данных:**

А. текстовый

Б. числовой

В. денежный

Г. Дата/ время

8. Составное объектное множество

А. Максимальное количество элементов одного объектного множества, связанных с одним элементом другого объектного множества.

Б. Мощность отношения конкретизации

В. Отношение, рассматриваемое как объектное множество.

Г. Отношение, максимальная мощность которого как минимум в одном направлении равна одному

9.Атрибуты это-

А. запросы и
формы

Б. базы данных

В. имя, дата рождения, номер страховки, рост, вес, пол,

Г. таблицы

10. Мощность отношения обозначает

1.

А. число элементов одного множества, связанных посредством отношения с одним элементом другого объектного множества

Б.Модель данных состоит из объектных множеств

В. Отношение устанавливает связи между элементами двух объектных множеств

Г. Объектные множества представляют категории

Тесты по дисциплине «Базы данных»

ВАРИАНТ№5

Запрос с параметрами -

А. это запрос, создаваемый при помощи языка SQL

Б. это запрос, при выполнении отображающий в собственном диалоговом окне приглашение ввести данные

В. объекты, которые идут на печать

Г. С помощью макросов можно выполнить практически все действия над объектами

2. База данных - это

А. - совокупность языковых и программных средств, предназначенных для создания, ведения и эксплуатации баз данных;

Б.- Банк данных (БнД) − совокупность баз данных и системы управления базами данных.

В. − формализованное описание, отражающее состав и типы данных, а также взаимосвязь между ними. Г.-совокупность структурированных данных.

3.**Кортеж в реляционной модели это -**

А. -строка; Б.
-столбец:

В. –таблица

Г. отношения

4. Язык T-SQL делиться на:

А. DML, DDL, DOL,

Б. DDL, DKL, DML

В. DCL,DML, DDL,

Г. DOL, DML,DDL

5. Select - это команда

А. DML

Б. DDL

В. DCL

Г. KDL

6. Предложение order by

a. сортирует

b.группирует

c. выбирает

7. Главными элементами модели данных

Группа_____ФИО_____

1.
А. запросы и
таблицы Б. объекты
и отношения.
В. Сетевая и реляционная модели
Г. Текстовый и логический

8. Системы базы данных состоят из главных компонента:

А. Аномалия обновления, удаления и ввода
Б. один к одному, один ко многим, и многие ко многим

В. данные, аппаратное обеспечение, программное обеспечение и

пользователи. Г. DDL (Data Definition Language) и DML (Data

Manipulation Language)

9. Реляционная модель данных
А. Количество атрибутов реляции
Б. это термин, пришедший из математики
В. организует и представляет данные в виде таблиц или реляций.
Г. Количество НЕ атрибутов реляции

10. Модель *«Клиент-сервер»* – технология
А. Графические возможности оболочки производят большое впечатление при изготовлении высококачественных отчетов и распечаток.
Б. технология, разделяющая приложение- СУБД на две части: клиентскую и сервер,
В. Они позволяют выбрать из базы только необходимую информацию, т.е. ту, которая соответствует определенному критерию
Г. когда размеры баз данных велики

Тесты по дисциплине «Базы данных»
ВАРИАНТ№6

1. В данном примере поле адрес ул. Душанбинская №45 относится к типу данных MS Access:
А.дата/время
Б.числовой
В.текстовый
Г.логический

2. Сколько форм нормализации:
А.5
Б.4
В.2
Г.3

3. Процесс приведения реляционных таблиц к стандартному виду - это

1.
А. Стандартизация
Б. Типы данных
В.Схема данных
Г. Нормализация.

4. Повторение данных в базе данных
А. Целостность данных
Б. Избыточность данных
В. Противоречивость данных, вызванная их избыточностью и частичным обновлением
Г. Аномалия удаления

5. Функциональная зависимость.
А. Атрибут (ы) в левой части функциональной зависимости
Б. Никакие неключевые атрибуты не являются функционально зависимыми лишь от части ключа
В. Значение атрибута в кортеже однозначно определяет значение другого
атрибута в кортеже Г. Таблица, состоящая из нескольких выбранных атрибутов

6. Транзитивная зависимость.
А. Первая нормальная форма запрещает таблицам иметь неатомарные
Б. если таблица удовлетворяет НФБК
В. Неключевой атрибут функционально зависит от одного или более неключевых атрибутов
Г.если связи между независимыми атрибутами можно исключить

7. Реляционная модель данных
А. Основана на логических отношениях данных
Б. Модель данных, представляющая данные в виде таблиц.
В. Основана на концептуальной модели данных
Г. Реляционная модель данных организует и представляет данные

8.Объекты MS Access:
А.Таблицы, запросы, формы, отчеты
Б. Текстовое поле, логическое поле
В. Атрибуты
Г. Выше перечисленные

9.Создание новых объектов базы данных: форм
А. Лот №Наименование товара, Единица измерения, Склад №
Б. Текстовое поле, логическое поле
В. В столбец, Ленточная, Табличная, Сводная таблица
Г. Выше перечисленные

10. Внешний ключ –это
А. Древовидные иерархические структуры широко используются в повседневной человеческой деятельности Б. Идентификатор
В. Система баз данных

Группа_____ФИО_____

1.
Г. Набор лексических атрибутов, значения которых всегда однозначно определяют элемент объектного множества

Группа_____ФИО_____
_____

КОНСПЕКТ ЛЕКЦИЙ

Лекция №1. Введение. Основные понятия теории баз данных.

ВВЕДЕНИЕ

В истории вычислительной техники можно проследить развитие двух основных областей ее использования. Первая область — применение вычислительной техники для выполнения численных расчетов, которые слишком долго или вообще невозможно производить вручную. Развитие этой области способствовало интенсификации методов численного решения сложных математических задач, появлению языков программирования, ориентированных на удобную запись численных алгоритмов, становлению обратной связи с разработчиками новых архитектур ЭВМ. Характерной особенностью данной области применения вычислительной техники является наличие сложных алгоритмов обработки, которые применяются к простым по структуре данным, объем которых сравнительно невелик.

Вторая область — это использование средств вычислительной техники в автоматических или автоматизированных информационных системах. Информационная система представляет собой программно-аппаратный комплекс, обеспечивающий выполнение следующих функций:

- надежное хранение информации в памяти компьютера;
- выполнение специфических для данного приложения преобразований информации и вычислений;
- предоставление пользователям удобного и легко осваиваемого интерфейса.

Обычно такие системы имеют дело с большими объемами информации, имеющей достаточно сложную структуру. Классическими примерами информационных систем являются банковские системы, автоматизированные системы управления предприятиями, системы резервирования авиационных или железнодорожных билетов, мест в гостиницах и т. д.

Основные понятия и определения

Банк данных (БнД) — это система специальным образом организованных данных — баз данных, программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных.

База данных (БД) —это набор информации, которая хранится упорядоченно в электронном виде.

СУБД — это система управления базами данных. Первые СУБД появились ещё в 1970-х, и сегодня их используют в каждой второй компании: от небольших интернет-магазинов до Facebook, Google и Amazon.

Чтобы управлять базами данных и находить нужную информацию, запросы к ним пишут на специальных языках. Самый популярный из них **— SQL (от англ. Structured Query Language — «язык структурированных запросов»).**

Система управления базами данных (СУБД) — совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

Программы, с помощью которых пользователи работают с базой данных, называются ***приложениями***. В общем случае с одной базой данных могут работать множество различных приложений. Например, если база данных моделирует некоторое предприятие, то для работы с ней может быть создано приложение, которое обслуживает подсистему учета кадров, другое приложение может быть посвящено работе подсистемы расчета заработной платы сотрудников, третье приложение работает как подсистемы складского учета, четвертое

приложение посвящено планированию производственного процесса. При рассмотрении приложений, работающих с одной базой данных, предполагается, что они могут работать параллельно и независимо друг от друга, и именно СУБД призвана обеспечить работу множества приложений с единой базой данных таким образом, чтобы каждое из них выполнялось корректно, но учитывало все изменения в базе данных, вносимые другими приложениями.

БД используются для хранения информации об объектах какой-либо предметной области.

Предметная область - часть реального мира, подлежащая изучению с целью автоматизации. Предметную область можно представить как множество взаимосвязанных объектов.

Объект (сущность) - это выделенный элемент предметной области, подлежащий хранению в БД. Другими словами - это «нечто, о чем мы хотим хранить информацию в БД». Объект может быть реальным (человек, населенный пункт, какой-либо предмет) и абстрактным (событие, счет покупателя).

Для каждого объекта выделяют набор признаков (характеристик, свойств или атрибутов) которые позволяют описать объект в рамках выбранной предметной области.

Если рассматривать человека как объект, о котором мы хотим хранить информацию в БД, то можно заметить что для предметных областей связанных с медициной наиболее значимыми наборами характеристик человека могут оказаться: рост, вес, пол и т.д. Для производства набор значимых характеристик человека иной: возраст, должность, рейтинг и т.д.

Характеристики (свойства, атрибуты) – набор признаков определяющих объект для выбранной предметной области.

Данные (в концепции БД) – это набор конкретных значений, параметров, характеризующих объект.

Не следует путать характеристики и данные, например, «ВЕС» – это характеристика объекта, а 120кг – это конкретное значение (данные).

Класс объектов - совокупность объектов, обладающих одинаковым набором свойств (характеристик).

Пример. Секретарь учебного заведения должен учитывать контактную информацию об учащихся (адрес, телефон и т.д.). Эту информацию удобно расположить в таблице (таблица 1.1). Столбцы таблиц обычно называют ***полями***, а строки –***записями***. Каждая запись таблицы (строка) содержит индивидуальные ***данные*** конкретного ученика – ***объекта***. Заголовок таблицы представляет собой набор ***свойств*** (***характеристик***). Все однотипные объекты (ученики) составят ***класс объектов***. Можно сказать, что мы получили однотабличную ***БД***.

Таблица 1 – Контактная информация учащихся

№ п/п	**Фамилия**	**Имя**	**Телефон**	**Адрес**
1	Александров	Саша	5-55-57	пр. Ленина, д. 25, кв. 17
2	Андреев	Степан	5-15-23	ул. Чапаева, д. 18, кв. 8
3	Башкирова	Ольга	5-50-44	ул. Северная, д. 117
…	…	…	…	…

Система управления базами данных (СУБД) – это совокупность языковых и программных средств, предназначенных для создания, ведения и использования БД.

СУБД позволяют структурировать, систематизировать и организовывать данные для компьютерного хранения и обработки. Системы управления базами данных (DataBase Management System – DBMS) являются основами практически любой информационной системы. Использование современных СУБД позволяет дать следующие преимущества:

1. Может быть сокращена избыточность в хранимых данных.
2. Может быть устранена возможность возникновения противоречивости хранимых данных.
3. Централизованное управление обеспечивает соблюдение стандартов в представлении данных, принятых в данной предметной области.
4. Могут быть выполнены условия безопасности данных.
5. Может поддерживаться целостность данных. Благодаря централизованному управлению, могут быть определены процедуры проверки, выполняющиеся при операциях запоминания.
6. Может быть обеспечена независимость данных и приложений, т.е. возможность расширения приложений независимо от базы данных и наоборот возможность расширения базы данных без влияния на приложения.

Информационная система (ИС) - взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели. В большинстве случаев сегодня под ИС понимают автоматизированные информационные системы (АИС).

АИС позволяют автоматизировать деятельность, повысить качество и достоверность обрабатываемой информации. Информационную основу ИС составляют хранящиеся в ней данные. В большинстве случаев это БД, а для управления данными используют СУБД.

Тема 2: Архитектура базы данных.

В процессе научных исследований, посвященных тому, как именно должна быть устроена СУБД, предлагались различные способы реализации. Самым жизнеспособным из них оказалась предложенная американским комитетом по стандартизации ANSI (American National Standards Institute) трехуровневая система организации БД, изображенная на рис. 1.

1. *Уровень внешних моделей* — самый верхний уровень, где каждая модель имеет свое «видение» данных. Этот уровень определяет точку зрения на БД отдельных приложений. Каждое приложение видит и обрабатывает только те данные, которые необходимы именно этому приложению. Например, система распределения работ использует сведения о квалификации сотрудника, но ее не интересуют сведения об окладе, домашнем адресе и телефоне сотрудника, и наоборот, именно эти сведения используются в подсистеме отдела кадров.
2. *Концептуальный уровень* — центральное управляющее звено, здесь база данных представлена в наиболее общем виде, который объединяет данные, используемые всеми приложениями, работающими с данной базой данных. Фактически концептуальный уровень отражает обобщенную модель предметной области (объектов реального мира), для которой создавалась база данных. Как любая модель, концептуальная модель отражает только существенные, с точки зрения обработки, особенности объектов реального мира.

Рис. 1 Архитектура базы данных

3. *Физический уровень* — собственно данные, расположенные в файлах или в страничных структурах, расположенных на внешних носителях информации.

Эта архитектура позволяет обеспечить *логическую* (между уровнями 1 и 2) и *физическую* (между уровнями 2 и 3) независимость при работе с данными. ***Логическая независимость*** предполагает возможность изменения одного приложения без корректировки других приложений, работающих с этой же базой данных. ***Физическая независимость*** предполагает возможность переноса хранимой информации с одних носителей на другие при сохранении работоспособности всех приложений, работающих с данной базой данных.

Выделение концептуального уровня позволило разработать аппарат централизованного управления базой данных.

Тема 3: Введение в банки данных. OLAP и OLTP-системы

Пользователи банков данных

Как любой программно-организационно-технический комплекс, банк данных существует во времени и в пространстве. Он имеет определенные стадии своего развития:

1. Проектирование.
2. Реализация.
3. Эксплуатация;
4. Модернизация и развитие.
5. Полная реорганизация.

На каждом этапе своего существования с банком данных связаны разные категории пользователей.

Определим основные категории пользователей и их роль в функционировании банка данных:

Конечные пользователи. Это основная категория пользователей, в интересах которых и создается банк данных. В зависимости от особенностей создаваемого банка данных круг его

конечных пользователей может существенно различаться. Это могут быть случайные пользователи, обращающиеся к БД время от времени за получением некоторой информации, а могут быть регулярные пользователи. В качестве случайных пользователей могут рассматриваться, например, возможные клиенты вашей фирмы, просматривающие каталог вашей продукции или услуг с обобщенным или подробным описанием того и другого. Регулярными пользователями могут быть ваши сотрудники, работающие со специально разработанными для них программами, которые обеспечивают автоматизацию их деятельности при выполнении своих должностных обязанностей. Например, менеджер, планирующий работу сервисного отдела компьютерной фирмы, имеет в своем распоряжении программу, которая помогает ему планировать и распределять текущие заказы, контролировать ход их выполнения, заказывать на складе необходимые комплектующие для новых заказов. Главный принцип состоит в том, что от конечных пользователей не должно требоваться каких-либо специальных знаний в области вычислительной техники и языковых средств.

Администраторы банка данных. Это группа пользователей, которая на начальной стадии разработки банка данных отвечает за его оптимальную организацию с точки зрения одновременной работы множества конечных пользователей, на стадии эксплуатации отвечает за корректность работы данного банка информации в многопользовательском режиме. На стадии развития и реорганизации эта группа пользователей отвечает за возможность корректной реорганизации банка без изменения или прекращения его текущей эксплуатации.

Разработчики и администраторы приложений. Это группа пользователей, которая функционирует во время проектирования, создания и реорганизации банка данных. Администраторы приложений координируют работу разработчиков при разработке конкретного приложения или группы приложений, объединенных в функциональную подсистему. Разработчики конкретных приложений работают с той частью информации из базы данных, которая требуется для конкретного приложения.

Современный уровень развития аппаратных и программных средств с некоторых пор сделал возможным повсеместное ведение баз данных оперативной информации на разных уровнях управления. В процессе своей деятельности промышленные предприятия, корпорации, ведомственные структуры, органы государственной власти и управления накопили большие объемы данных. Они хранят в себе большие потенциальные возможности по извлечению полезной аналитической информации, на основе которой можно выявлять скрытые тенденции, строить стратегию развития, находить новые решения.

В области информационных технологий существуют два взаимно дополняющих друг друга направления:

• технологии, ориентированные на оперативную (транзакционную) обработку данных. Эти технологии лежат в основе экономических информационных систем, предназначенных для оперативной обработки данных. Называются подобные системы - **OLTP** (online transaction processing) системы;

• технологии, ориентированные на анализ данных и принятие решений. Эти технологии лежат в основе экономических информационных систем, предназначенных для анализа накопленных данных. Называются подобные системы - **OLAP** (online analytical processing) системы .

1. Понятие и основное назначение ОЛТП систем

OLTP - системы, являясь высокоэффективным средством реализации оперативной обработки, оказались мало пригодны для задач аналитической обработки. Это вызвано следующим:

1. средствами традиционных OLTP -систем можно построить аналитический отчет и даже прогноз любой сложности, но заранее регламентированный. Любой шаг в сторону, любое

нерегламентированное требование конечного пользователя, как правило, требует знаний о структуре данных и достаточно высокой квалификации программиста;
2. многие необходимые для оперативных систем функциональные возможности являются избыточными для аналитических задач и в то же время могут не отражать предметной области. Для решения большинства аналитических задач требуется использование внешних специализированных инструментальных средств для анализа, прогнозирования и моделирования. Жесткая же структура баз не позволяет достичь приемлемой производительности в случае сложных выборок и сортировок и, следовательно, требует больших временных затрат для организации шлюзов.
3. в отличие от транзакционных, в аналитических системах не требуются и, соответственно, не предусматриваются развитые средства обеспечения целостности данных, их резервирования и восстановления. Это позволяет не только упростить сами средства реализации, но и снизить внутренние накладные расходы и, следовательно, повысить производительность при выборке данных.

2. Понятие и основное назначение ОЛАП систем

Основное назначение OLAP -систем - динамический многомерный анализ исторических и текущих данных, стабильных во времени, анализ тенденций, моделирование и прогнозирование будущего. Такие системы, как правило, ориентированы на обработку произвольных, заранее не регламентированных запросов. В качестве основных характеристик этих систем можно отметить следующие :
• поддержка многомерного представления данных, равноправие всех измерений, независимость производительности от количества измерений;
• прозрачность для пользователя структуры, способов хранения и обработки данных;
• автоматическое отображение логической структуры данных во внешние системы;
• динамическая обработка разряженных матриц эффективным способом.
Термин OLAP является сравнительно новым и в разных литературных источниках трактуется иногда по разному. Этот термин часто отождествляют с поддержкой принятия решений (DSS (Decision Support Systems)- системы поддержки принятия решения. А в качестве синонима для последнего термина используют Data Warehousing -хранилища (склады) данных, понимая под этим набор организационных решений, программных и аппаратных средств для обеспечения аналитиков информацией на основе данных из систем обработки транзакций нижнего уровня и других источников
“Склады данных” позволяют обрабатывать данные, накопленные за длительные периоды времени. Эти данные являются разнородными (и не обязательно структурированными). Для “складов данных” присущ многомерный характер запросов. Огромные объемы данных, сложность структуры как данных, так и запросов требует использования специальных методов доступа к информации.
В других источниках понятие Системы Поддержки Принятия Решений (СППР) считается более широким. Хранилища данных и средства оперативной аналитической обработки могут служить одними из компонентов архитектуры СППР.
OLAP всегда включает в себя интерактивную обработку запросов и последующий многопроходный анализ информации, который позволяет выявить разнообразные, не всегда очевидные, тенденции, наблюдающиеся в предметной области.
Иногда различают " OLAP в узком смысле" - это системы которые обеспечивают только выборку данных в различных разрезах, и " OLAP в широком смысле", или просто OLAP , включающей в себя:
- поддержку нескольких пользователей, редактирующих БД.
- функции моделирования, в том числе вычислительные механизмы получения производных результатов, а также агрегирования и объединения данных;
- прогнозирование, выявление тенденций и статистический анализ.

Естественно, что каждый из этих типов ИС требует специфической организации данных, а так же специальных программных средств, обеспечивающих эффективное выполнение стоящих задач.
OLAP - средства обеспечивают проведение анализа деловой информации по множеству параметров, таких как вид товара, географическое положение покупателя, время оформления сделки и продавец, каждый из которых допускает создание иерархии представлений. Так, для времени можно пользоваться годовыми, квартальными, месячными и даже недельными и дневными промежутками; географическое разбиение может проводиться по городам, штатам, регионам, странам или, если потребуется, по целым полушариям.
Для ИТ-практиков OLAP — это технология быстрого создания интерактивных отчётов. Как правило, OLAP-отчёт может быть разработан практически без программирования, методом drag & drop, т. е. переноса групп данных из одного места в другое при помощи мыши. Проектирования и программирования требует только этап извлечения и подготовки данных для отчётов из корпоративных систем предприятия. Серия отчётов для определённой предметной области может быть настроена буквально за несколько часов.
Для конечных пользователей OLAP — это табличный отчёт с промежуточными и окончательными итогами. Он выпускается практически мгновенно, и можно самостоятельно изменить его форму — расположение колонок и строк, порядок фильтров и сортировки и т. д. Это позволяет взглянуть на одни и те же данные под разными углами зрения. Например, получив перечень по дебиторской задолженности, углубиться в архивные данные и посмотреть, были ли у должника проблемы с платежами в прошлом квартале, полгода назад, в ушедшем году. Разносторонний анализ позволяет быстро оценить ситуацию и получить максимум информации о проблеме, чтобы на этой основе наметить пути её решения.
С появлением и использованием OLAP-инструмента традиционная задача отдела автоматизации — разработка новых отчётов, в которых одни данные показаны в разрезе других, — превращается в другую задачу, суть которой заключается в том, чтобы предоставить пользователю исходные данные. После этого экономист или бухгалтер не тратит время на написание служебных записок с требованием создать новую разновидность отчёта и не ждёт, пока программисты спроектируют, разработают и протестируют его. Пользователь сам конструирует нужный вид отчёта, выбирая из настроенной для него заготовки нужные поля, меняя порядок их следования, устанавливая фильтры и др. Созданный отчёт может быть проиллюстрирован графиками и диаграммами, распечатан на бумагу или выгружен в Excel.

Лекция 4: Принцип построение БД. Инфологическое (концептуальное) моделирование предметной области. Проектирование реляционных БД на основе принципов нормализации

Проектирование реляционной БД - это набор взаимосвязанных отношений, в которых определены все атрибуты, заданы первичные ключи отношений и заданы еще некоторые дополнительные свойства отношений, которые относятся к принципам поддержки целостности. Этапы жизненного цикла базы данных изображены на рис.

Процесс проектирования БД представляет собой последовательность переходов от неформального словесного описания информационной структуры предметной области к формализованному описанию объектов предметной области в терминах некоторой модели. В общем случае можно выделить следующие этапы проектирования:

1. Системный анализ и словесное описание информационных объектов предметной области.
2. Проектирование инфологической модели предметной области — частично формализованное описание объектов предметной области в терминах некоторой семантической модели.
3. Даталогическое или логическое проектирование БД, то есть описание БД в терминах принятой диалогической модели данных.

Физическое проектирование БД, то есть выбор эффективного размещения БД на внешних носителях для обеспечения наиболее эффективной работы приложения.

Системный анализ предметной области

С точки зрения проектирования БД в рамках системного анализа, необходимо осуществить первый этап, то есть провести подробное словесное описание объектов предметной области и реальных связей, которые присутствуют между описываемыми объектами. Желательно, чтобы данное описание позволяло корректно определить все взаимосвязи между объектами предметной области.

В общем случае существуют два подхода к выбору состава и структуры предметной области:

- *Функциональный подход* — он реализует принцип движения «от задач» и применяется тогда, когда заранее известны функции некоторой группы лиц и комплексов задач, для обслуживания информационных потребностей которых создается рассматриваемая БД. В этом случае мы можем четко выделить минимальный необходимый набор объектов предметной области, которые должны быть описаны.
- *Предметный подход* — когда информационные потребности будущих пользователей БД жестко не фиксируются. Они могут быть многоаспектными и весьма динамичными. Мы не можем точно выделить минимальный набор объектов

предметной области, которые необходимо описывать. В описание предметной области в этом случае включаются такие объекты и взаимосвязи, которые наиболее характерны и наиболее существенны для нее. БД, конструируемая при этом, называется предметной, то есть она может быть использована при решении множества разнообразных, заранее не определенных задач. Конструирование предметной БД в некотором смысле кажется гораздо более заманчивым, однако трудность всеобщего охвата предметной области с невозможностью конкретизации потребностей пользователей может привести к избыточно сложной схеме БД, которая для конкретных задач будет неэффективной.

Чаще всего на практике рекомендуется использовать некоторый компромиссный вариант, который, с одной стороны, ориентирован на конкретные задачи или функциональные потребности пользователей, а с другой стороны, учитывает возможность наращивания новых приложений.

Системный анализ должен заканчиваться подробным описанием информации об объектах предметной области, которая требуется для решения конкретных задач и которая должна храниться в БД, формулировкой конкретных задач, с кратким описанием алгоритмов их решения, описанием выходных документов, которые должны генерироваться в системе, описанием входных документов, которые служат основанием для заполнения данными БД.

Инфологическая модель предметной области.

Модель «Сущность - связь»

Одной из наиболее популярных средств формализованного представления предметной области является модель «сущность — связь» (ER-модели).

Семантическую основу ER-модели составляют следующие предположения:

- та часть реального мира (совокупность взаимосвязанных объектов), сведения о которых должны быть помещены в базу данных, может быть *представлена* как совокупность *сущностей;*
- каждая сущность обладает характеристическими свойствами (атрибутами), отличающими ее от других сущностей и позволяющими ее *идентифицировать;*
- сущности можно классифицировать по типам сущностей: каждый экземпляр сущности (представляющий некоторый объект) может быть отнесен к классу — *типу сущностей,* каждый экземпляр которого обладает общими для них и отличающими их от сущностей других классов свойствами;
- систематизация представления, основанная на классах, в общем случае предполагает иерархическую зависимость типов: сущность типа является *подтипом* сущности *B,* если каждый экземпляр типа *A* является экземпляром сущности типа *B;*
- взаимосвязи объектов могут быть представлены как *связи* — сущности, которые служат для фиксирования (представления) взаимозависимости двух или нескольких сущностей.

ER-модель должна определить объекты и взаимосвязи между ними, т. е. установить связи следующих двух типов.

1.Связи между объектами и наборами характеристических свойств, и таким образом определить сами объекты.

2.Связи между объектами, задающие характер и функциональную природу их взаимозависимости.

ER-моделирование предметной области базируется на использовании графических диаграмм.

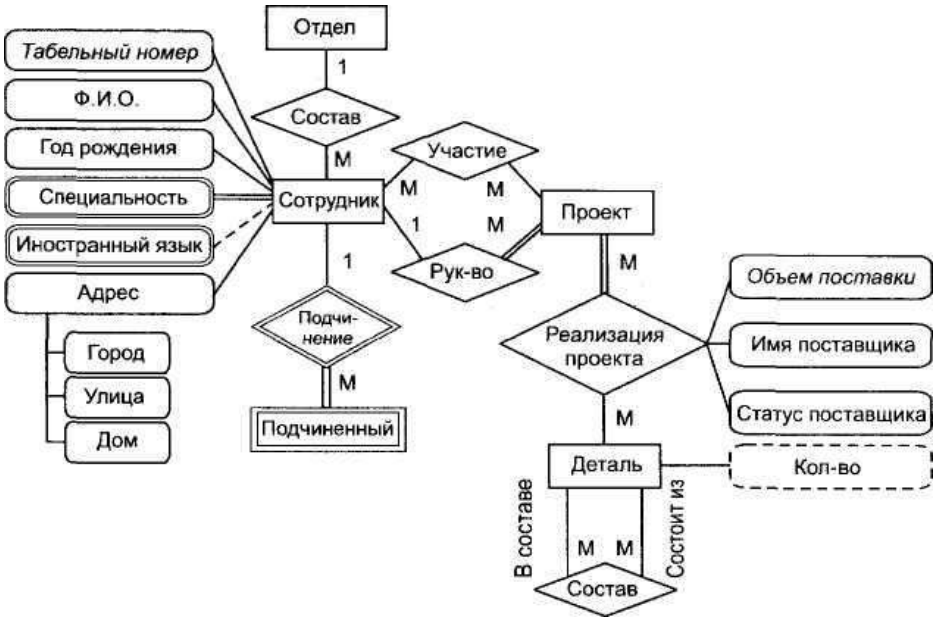


Рис. 10 Пример ER-диаграммы

Сущность. Сущность, с помощью которой моделируется класс однотипных объектов, определяется как «предмет, который может быть четко идентифицирован». Сущность должна *определяться* таким *набором атрибутов,* который позволял бы различать отдельные экземпляры сущности. Каждый экземпляр сущности должен быть отличим от любого другого экземпляра той же сущности. Например, для однозначной идентификации каждого экземпляра сущности «Сотрудник» вводится атрибут «Табельный номер», который вследствие своей природы будет всегда иметь уникальное значение в рамках предприятия. Уникальным идентификатором сущности может являться атрибут, комбинация атрибутов, комбинация связей или комбинация связей и атрибутов, однозначно отличающая любой экземпляр сущности от других экземпляров сущности того же типа.

Сущность имеет *имя,* уникальное в пределах модели. При этом *имя сущности* — это *имя типа,* а не некоторого конкретного экземпляра.

Сущности подразделяются на *сильные* и *слабые.* Сущность является слабой, если ее существование зависит от другой сущности — сильной по отношению к ней. Например, сущность «Подчиненный» является слабой по отношению к сущности «Сотрудник»: если будет удалена запись, соответствующая некоторому сотруднику, имеющему подчиненных, то сведения о подчинении также должны быть удалены.

Свойства. Свойство может быть *множественным* или *единичным* — т. е. атрибут, задающий свойство, может одновременно иметь несколько значений или, соответственно, только одно. Например, сотрудник может иметь несколько специальностей, но единственное значение — «Табельный номер».

Свойство может быть *простым* (не подлежащим дальнейшему делению с точки зрения прикладных задач) или *составным* — если его значение составляется из значений простых свойств. Например, свойство «Год рождения» является простым, а свойство «Адрес» — составным, так как включает значения простых свойств «Город», «Улица», «Дом».

В некоторых случаях полезно различать *базовые* и *производные* свойства. Например, «Поставщик» может иметь свойство «Общее количество поставляемых деталей», которое вычисляется суммированием количества деталей, поставляемых им по проекту.

Если наличие некоторого свойства для всех экземпляров сущности не является обязательным, то такое свойство называется *условным.* Например, не все сотрудники обладают свойством «ученая степень».

Значения свойств могут быть *статическими* или *динамическими,* т. е. меняться со временем. Например, свойство «Табельный номер» является статическим, а «Адрес» — динамическим. Свойство может быть *неопределенным,* если оно является динамическим, но его текущее значение еще не задано.

Свойство может рассматриваться как *ключевое,* если его значение уникально и, возможно, в определенном контексте, однозначно идентифицирует сущность. Например, подчиненный некоторого определенного сотрудника.

Связи. Кроме связей между объектом и его свойствами, модель отражает связи между объектами разных классов. *Связь* определяется как «ассоциация, объединяющая несколько сущностей». Эта ассоциация всегда может существовать между разными сущностями или между сущностью и ею же самой (рекурсивная связь).

Сущности, объединяемые связью, называются *участниками. Степень связи* определяется количеством участников связи.

Если каждый экземпляр сущности участвует, по крайней мере, в одном экземпляре связи, то такое участие этой сущности называется *полным* (или *обязательным);* в противном случае — *неполным* (или *необязательным).*

Количественный характер участия экземпляров сущностей задается *типом связи* (или *мощностью связи).* Возможны следующие типы: *«один к одному»* (1:1), *«один ко многим»* (1:M), *«многие к одному»* (M:1), *«многие ко многим»* (M:M).

Следует отметить, что инструмент связей — это средство представления *сложных объектов,* каждый из которых может рассматриваться как множество некоторым образом взаимосвязанных *простых объектов.* Деление на простые и сложные объекты, также как и характер взаимосвязи, является условным и определяется особенностями анализа предметной области, т. е. в конце концов — характером использования данных о предметах в решаемых прикладных задачах. При этом с точки зрения, например, конструктора, ДЕТАЛЬ является сложным объектом, а с точки зрения Поставщика — простым.

Среди многих разновидностей взаимосвязей наиболее частыми являются такие отношения иерархического типа, как «часть — целое», «род — вид».

Отношение «часть — целое» используются для представления *составных объектов.* Например, МАШИНЫ состоят из УЗЛОВ, УЗЛЫ состоят из ДЕТАЛЕЙ. Здесь возможны как отношения *«один ко многим»,* так и *«многие ко многим».*

Отношение «род — вид» — для представления *обобщенных объектов.* Например, СОТРУДНИКИ подразделяются по профессии на КОНСТРУКТОРОВ, ПРОГРАММИСТОВ, РАБОЧИХ; ПРОГРАММИСТЫ - на ПРИКЛАДНЫХ ПРОГРАММИСТОВ и СИСТЕМНЫХ ПРОГРАММИСТОВ. Иерархические отношения, и в частности — «родовидовые», обычно используются как основа классификации объектов по наборам характеристических признаков. Причем, «видовые» объекты *наследуют* свойства «родовых».

Другой широко используемой разновидностью взаимосвязи является агрегирование — объединение простых объектов в сложный по принципу их принадлежности *агрегату* или их совместного участия в некотором процессе. Агрегирование, рассматриваемое здесь как более общий случай иерархических отношений, объединяет объекты разной природы с единственным общим свойством «совместное участие». Агрегированные объекты именуются обычно отглагольными существительными, например, *«Состав»:* ПОДРАЗДЕЛЕНИЕ *состоит из* СОТРУДНИКОВ; *«Поставка»:* ПОСТАВЩИК *поставляет* ДЕТАЛИ.

Супертипы и подтипы. Сущность может быть расщеплена на два или более взаимоисключающих *подтипов,* каждый из которых включает общие атрибуты и/или связи. Эти общие атрибуты и/или связи явно определяются один раз на более высоком уровне. В подтипах могут определяться собственные атрибуты и/или связи. В принципе выделение подтипов может продолжаться на более низких уровнях, но в большинстве случаев оказывается достаточно двух-трех уровней.

Сущность, на основе которой определяются подтипы, называется *супертипом.* Подтипы должны образовывать полное множество, т. е. любой экземпляр супертипа должен

относиться к некоторому подтипу. Иногда для полноты множества надо определять дополнительный подтип, например, ПРОЧИЕ.

Подтип наследует свойства и связи супертипа. Например, тип сущности ПРОГРАММИСТ является подтипом сущности СОТРУДНИК. Программисты обладают всеми свойствами сотрудников и участвуют во всех связях, однако обратные утверждения неверны.

Тип сущности, его подтипы, подтипы этих подтипов и т. д. образуют *иерархию типов сущности,* пример которой приведен на рис. 11.

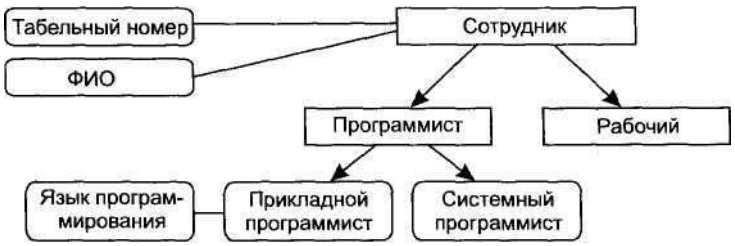


Рис. 11 Пример иерархии типов сущности

ER-диаграмма

ER-диаграмма является очень удачным решением моделирования. В ней сочетаются функциональный и информационный подходы, что позволяет представлять как совокупность выполняемых функций, так и отношения между элементами системы, задаваемые структурами данных.

Сущности. Каждый тип сущности в ER-диаграммах представляется в виде прямоугольника, содержащего имя сущности. В качестве имени обычно используются существительные (или обороты существительного) в единственном числе. Для отражения сущностей слабых типов используются прямоугольники, стороны которых рисуются двойными линиями. Например, в рассматриваемой далее ER-диаграмме, приведенной на рис. 5.4, ПОДЧИНЕННЫЙ — сущность слабого типа.

Свойства. Свойства служат для уточнения, идентификации, характеристики или выражения состояния сущности или связи. Свойства отображаются в виде эллипсов, содержащих имя свойства. Эллипс соединяется с соответствующей сущностью или связью линией.

Имена ключевых свойств подчеркиваются, например, свойство «Табельный номер» сущности СОТРУДНИК.

Контур эллипса рисуется двойной линией, если свойство многозначное, например, свойство «Специальность» сущности СОТРУДНИК.

Контур эллипса рисуется штриховой линией, если свойство производное, например, свойство «Кол-во» сущности ПОСТАВЩИК.

Эллипс соединяется пунктирной линией, если свойство условное, например, свойство «Иностранный язык» сущности СОТРУДНИК.

Если свойство составное, то составляющие его свойства отображаются другими эллипсами, соединенными с эллипсом составного, например, свойство «Адрес» сущности СОТРУДНИК состоит из простых свойств «Город», «Улица», «Дом».

Связи. Связь — это графически изображаемая ассоциация, устанавливаемая между сущностями. Каждый тип связи на ER-диаграмме отображается в виде ромба с именем связи внутри. В качестве имени обычно используются отглагольные существительные.

Стороны ромба рисуют двойными линиями, если это связь сущности слабого типа с сущностью, от которой она зависит. Например, связь «Подчинение», связывающая сущность слабого типа ПОДЧИНЕННЫЙ с сущностью СОТРУДНИК, от которой она зависит.

Участники связи соединены со связью линиями. Двойная линия обозначает полное участие сущности в связи с данной стороны. Например, связь «Подчинение» со стороны сущности ПОДЧИНЕННЫЙ.

Связь может быть модифицирована указанием роли. Например, для рекурсивной связи «Состав» указаны роли: «Деталь *состоит из* ...» и «Деталь входит *в состав* ...».

Тип связи указывается индексами «1» или «М» над соответствующей линией. Например, связь «Руководство» имеет тип «один ко многим»: один сотрудник может руководить многими проектами; связь «Участие» имеет тип «многие ко многим»: один сотрудник может участвовать во многих проектах, и в проекте могут участвовать многие сотрудники.

Нормальные формы ER-диаграмм

В *первой нормальной форме* ER-диаграммы устраняются повторяющиеся атрибуты или группы атрибутов, т. е. производится выявление неявных сущностей, «замаскированных» под атрибуты.

Во *второй нормальной форме* устраняются атрибуты, зависящие только от части уникального идентификатора. Эта часть уникального идентификатора определяет отдельную сущность.

В *третьей нормальной форме* устраняются атрибуты, зависящие от атрибутов, не входящих в уникальный идентификатор. Эти атрибуты являются основой отдельной сущности.

На рис. 12 представлена ER-диаграмма рис. 10 в третьей нормальной форме.

Рис. 12 Пример ER-диаграммы в третьей нормальной форме

Концептуальная модель данных

Концептуальная модель данных отображает обобщающее представление о данных, не зависимое от типа выбранной СУБД. Она описывает то, какие данные хранятся в базе данных, а также связи, существующие между ними. Фактически это полное представление требований к данным со стороны организации, у которой работают пользователи.

Концептуальная модель данных состоит из сущностей со своими атрибутами и n-парных связей и используется как средство построения и представления информационных потребностей предприятия.

Сущность: информационный объект, относящийся к деятельности предприятия

Атрибут: характеристика сущности

Связь: связь сущностей между собой, обычно между двумя сущностями, а в общем – между n сущностями; осуществляется через связь экземпляров одной сущности с экземплярами другой сущности

Роль: определяется с каждой стороны связи. Определяет смысл участия соответствующей сущности в данной связи (например, родительская сущность, дочерняя сущность)

Кардинальность связи: максимальное количество экземпляров одной сущности, связанных с одним экземпляром другой сущности

Управляемость ролью: показывает, что данная сущность является дочерней сущностью родительской сущности

Ограничения роли: механизм поддержания целостности связей

Ключ: может быть первичным или потенциальным.

Зависимость (подчиненность) ключа: определяется для первичных ключей и суперключей.

Сущность представляет собой любой абстрактный или конкретный объект организации, чьи характеристики определяются с помощью атрибутов.

При создании сущности, *Open ModelSphere* автоматически присваивает ей имя «Сущность». Во избежание неприятностей при создании сложных моделей, настоятельно рекомендуется присваивать сущностям смысловые имена как можно раньше, сразу после создания сущностей. Для этого следует использовать функцию редактирования содержимого созданной сущности.

Атрибут обычно содержит одно значение. Каждая сущность должна иметь, по крайней мере, один атрибут.

Open ModelSphere имеет функцию редактирования, которая позволяет добавлять атрибуты непосредственно в графическое представление сущности. При создании атрибута, *Open ModelSphere* автоматически присваивает атрибуту имя «Атрибут».

Первичные ключи можно создавать с помощью двух способов: используя панель инструментов, или, непосредственно вводя информацию в окне свойств сущностей модели.

Например: BOOKS ID, FIRMS ID, PAYMENTS ID, NAKLS ID.

В *Open ModelSphere*, первичные ключи подчеркнуты, чтобы отличить их от других атрибутов.

В рассматриваемом примере, все первичные ключи состоят из одного атрибута. Например, в сущности FIRMS использован атрибут FIRMS ID для идентификации каждой фирмы, атрибут BOOKS ID однозначно идентифицирует каждую книгу сущности BOOKS.

Open ModelSphere предоставляет два типа связей: с использованием прямых углов и использованием любых углов  . Оба типа играют одну и ту же роль, различаются только их изображения на диаграмме.

При создании связи устанавливаются кардинальности по умолчанию: показатель кардинальности равен **0,N** для родительской сущности, от которой начинается связь, и **1,1** – для дочерней сущности, на которой эта связь заканчивается. Если дочерняя сущность является слабой сущностью, то ее кардинальность должна быть подчеркнутой (такая связь называется **идентифицирующей**).

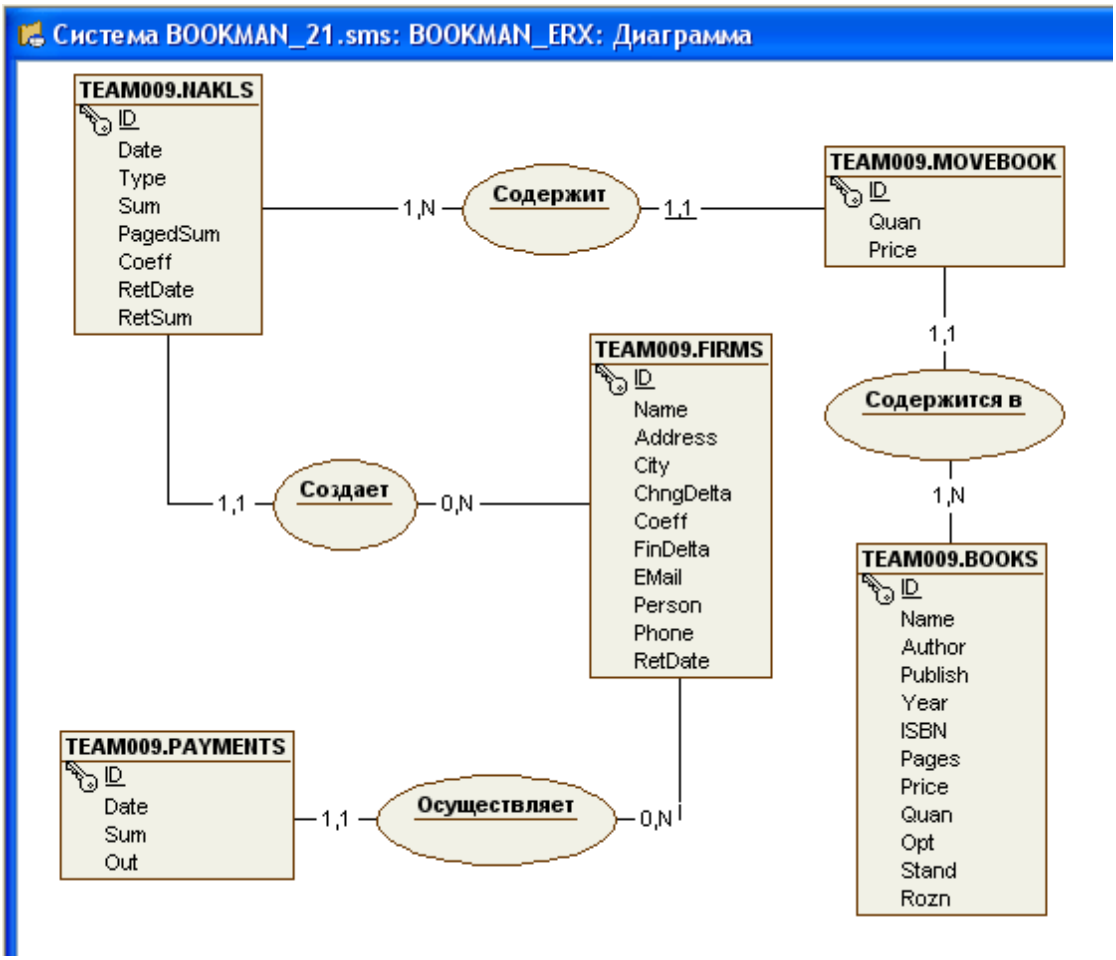


Рисунок 2. Окончательный вид концептуальной модели

2.3. Логическая модель данных

Логическая модель данных – это модель данных логического уровня не привязанная ни к какой конкретной СУБД.

Конкретные СУБД (Oracle, Firebird и т. д.) и такие специфические понятия баз данных как индексы, триггеры и т.д. будут рассмотрены в дальнейшем.

Перед созданием логической модели данных необходимо изучить такие понятия логической модели данных, как: таблицы, столбцы; первичные, потенциальные и внешние ключи; нормальные формы и правила ссылочной целостности.

Сначала ознакомимся с некоторыми основными терминами реляционных баз данных и моделирования логических структур данных (см. табл. 2).

Таблица 2. Основные термины

Термин	Определение
1	**2**
Таблица (Table)	Основной контейнер хранения данных в базе данных. Реляционную таблицу можно представить в виде плоскости, разделенной на строки и столбцы
Строка (row)	Соответствует одному объекту реального мира. Таким объектом может быть счет-фактура, запись в телефонной книге и т.д. Строки - это основа основ базы данных. Часто строки называют записями. Каждая строка таблицы должна содержать данные определенного типа. Таблица - всего лишь средство для организации строк
Столбец (column)	Элемент строки. Каждый столбец представляет собой определенную характеристику объекта, представленного строкой таблицы. Часто столбцы называют полями
Первичный ключ (primary key)	Столбец или набор столбцов таблицы, который однозначно идентифицирует каждую строку. Например, номера счетов в таблице счетов в каждой строке являются уникальными. Столбец, являющийся первичным ключом, обычно используется для создания индекса таблицы, предназначенного для ускорения доступа к ее строкам
Внешний ключ (foreign key)	Столбец или набор столбцов , импортированный из другой таблицы. Обычно внешний ключ является первичным ключом своей таблицы. Кроме того он может в таблице, где он используется как внешний ключ, быть и первичным ключом
Ограничение (constraint)	Механизм, обеспечивающий невозможность попадания неправильных данных в базу данных. Существует два основных типа ограничений: ограничения ссылочной целостности (referential integrity) и ограничения целостности доменов (domain integrity). Ограничения первого типа обеспечивают соблюдение целостности связей между таблицами. Ограничения второго типа не допускают попадания в базу данных значений неправильного типа, выходящих за заданные диапазоны и т.п.
Индекс (index)	Механизм физического хранения информации, который позволяет ускорить операции поиска строк в таблице. Использование индексов дает возможность отказаться от необходимости последовательного перебора всех строк таблицы - строки сортируются таким образом, чтобы обеспечить максимально быстрый поиск

Далее мы, используя в качестве исходных данных созданную концептуальную модель, преобразуем ее с помощью системы Open ModelSphere в завершенную логическую модель, а затем на этапе физического моделирования получим сценарий SQL, который можно использовать для создания объектов базы данных.

Концептуальную модель данных можно преобразовать в логическую модель. Процесс преобразования изменяет связи многие-ко-многим (N:N), переопределяет атрибуты связей и учитывает зависимости ключей для идентифицирующих связей. Логическая модель данных для реляционной базы данных в Open ModelSphere называется **реляционной моделью**.

При проектировании БД основной целью является выбор подходящей логической структуры для заданного массива данных, который требуется поместить в БД. Нужно решить, какие необходимы отношения и какой выбор атрибутов они должны включать.

При этом такой процесс называется концептуальным проектированием БД (относится к внешнему уровню представления данных).

Для определения подходящего набора отношений используется метод, называемый нормализацией. Нормализация представляет собой один из вариантов восходящего подхода к проектированию БД, который начинается с установления связей между атрибутами и заканчивается созданием необходимых отношений, определяющих логическую структуру БД.

Нормализация – это метод создания набора отношения с заданными свойствами на основе требований к данным.

Нормализация часто выполняется в виде последовательности тестов, выполняемых над некоторым отношением и необходимых для проверки его соответствия требованиям заданной нормальной формы.

Основная цель нормализации заключается в минимизации избыточности данных и сокращений объема памяти, необходимого для физического хранения БД.

При наличии избыточности возникает ряд проблем, осложняющих процесс функционирования СУБД.

Служащий

StaffNo	SName	SAddress	Position	Salary	Br.No
SL21					B5
SG37					B3
SG14					B7

Отделение

Br.No	BAddress	TellNo
B5		
B3		
B7		

Альтернативное представление данных с помощью одного отношения

Служащий - Отделение

StaffNo	SName	SAddress	Position	Salary	Br.No	BAddress	TellNo
SL21					B5		
SG37					B3		
SG14					B7		
CH2					B5		

Избыточные данные: сведения об отделении повторяются в кортежах, относящихся к каждому сотруднику.

Проблемы, возникающие при наличии избыточности в отношениях, называются аномалиями обновления и подразделяются на аномалии вставки, удаления и модификации.

Существуют два основных типа аномалий вставки.

•При вставке сведений о новых сотрудниках необходимо указывать и сведения об отделении, в которых эти сотрудники работают, которые должны соответствовать сведениям об этом же отделении в других строках отношения «Служащий-отделение». Первые отношения не могут пострадать от такого потенциального несоответствия, так как требуется вводить только номер отделения. Сведения об отделении, кроме того, заносятся однократно.

• Для вставки сведении о новом отделении, которое еще не имеет собственных сотрудников, потребуется присвоить значение NULL всем атрибутам, включая номер сотрудника StaffNo. Однако, этот атрибут является ключом и попытка ввода NULL нарушит целостность сущностей. Первые отношения позволяют избегать возникновения этой проблемы.

При удалении из отношения строки с информации о последнем сотрудники некоторого отделения, сведения об этом отделении будут полностью удалены из БД.

Первые два отношения позволяют избежать возникновения этой проблемы.

При попытке изменения значения одного из атрибутов для некоторого отделения в отношении необходимо обновить соответствующие значения в строках для всех сотрудников этого отделения. Если такой модификации будут подвергнуты не все требуемые строки, то база данных будет содержать противоречивые сведения.

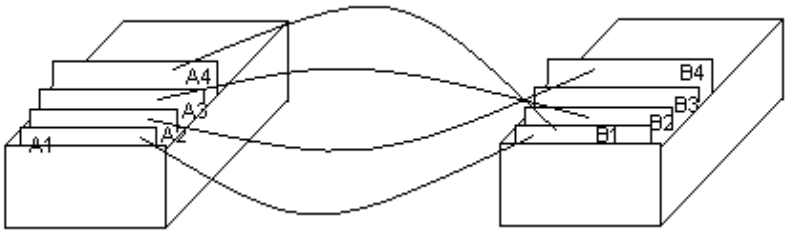
Приведенные примеры иллюстрируют, что первые два отношения обладают более приемлемыми свойствами, чем отношение Staff-Branch.

Нормализация и служит для получения правильно спроектированных отношений.

Но вначале следует познакомиться с концепцией функциональной зависимости.

Функциональная зависимость описывает связь между атрибутами и является одним из основных понятий нормализации.

Если в отношении R, содержащем атрибуты A и B, атрибут B функционально зависит от атрибута A (что обозначается A B), то каждое значение атрибута A связано только с одним значением атрибута B. (При этом каждый из атрибутов A и B может состоять из одного или нескольких атрибутов).



Детерминантом функциональной зависимости называется атрибут или группа атрибутов, расположенная на диаграмме функциональной зависимости слева от символа строки.

Staff – может быть несколько сотрудников с одинаковыми должностями.

Связь между атрибутами StaffNo и Position относится к типу 1:1, поскольку для каждого номера сотрудника имеется только одна должность. А связь между атрибутами Position и StaffNo имеет тип 1:N, так существует несколько номеров сотрудников, занимающих одну и ту же должность.

StaffNo – детерминант функциональной зависимости StaffNo Pjsition. →

Для выявления ключей отношения Staff - Brunch необходимо найти атрибут (или группу атрибутов), который уникальным образом идентифицирует каждую строку этого отношения. Это атрибут StaffNo.

В отношении Branch потенциальными ключами являются TelNo и Branch-No. В отношении Staff – атрибут StaffNo.

Нормализация - это формальный метод анализа отношения на основе их первичного ключа (или потенциальных ключей) и существующих функциональных зависимостей. Он включает ряд правил, которые могут использоваться для проверки отдельных отношений таким образом, чтобы вся БД могла быть нормализована до желаемой степени нормализации. Если некоторое требование не удовлетворяется, то нарушающее данное требование отношение необходимо декомпозировать на несколько отношений, каждое из которых удовлетворяет всем требованиям нормализации.

При работе с реляционной моделью данных важно понимать, что только удовлетворение требований 1НФ (каждый кортеж содержит ровно одно значение для каждого атрибута) обязательно для создания отношений приемлемого качества. Все остальные формы могут использоваться по желанию проектировщиков. Однако, для того чтобы избежать аномалий обновления нормализацию рекомендуется выполнять как минимум до 3НФ.

Отношение, в котором на пересечении каждой строки и каждого столбца содержится только одно значение (отсутствуют повторяющиеся группы данных).

BranchNo	StaffNo
B5	SL10, CF2

Ненормализованная форма.

Повторяющейся группой называется группа, состоящая из атрибута таблицы, в котором возможно наличие нескольких значений для единственного значения ключевого атрибута таблицы.

Существует два подхода исключения повторяющихся групп – выравнивание и декомпозиция.

BranchNo	StaffNo
B5	SL10
B5	CF2

…

Это выравнивание.

BranchNo	…
B5	

Это декомпозиция.

StaffNo	BranchNo
SL10	B5
CF2	B5

При использовании выравнивания дальнейшая нормализация все равно приводит к необходимости декомпозиции исходного отношения.

Вторая нормальная форма основана на понятии полной функциональной зависимости.

В некотором отношении атрибут B называется полностью функционально зависимым от атрибута A, если атрибут B функционально зависит от полного значения атрибута A и не зависит ни от какого подмножества полного значения атрибута A.

StaffNo, SName → BranchNo

(м. б. связано только с одним значением).

Эта зависимость не является полной, т. к. существует зависимость

StaffNo → BranchNo

Отношение, которое находится в первой нормальной форме и каждый атрибут которого, не входящий в состав первичного ключа, характеризуется полной функциональной зависимостью от этого первичного ключа.

Если в отношении между атрибутами существует частичная зависимость, то функционально – зависимые атрибуты удаляются из него и помещаются в новое отношение вместе с копией их детерминанта.

FIRST

Поставщик Рейтинг Город Поставки Количество

SNo	STATUS	CITY	PNO	QTY
S1	20	London	P1	300
S1	20	London	P2	200
S1	20	London	P3	400
S2	10	Paris	P1	300
S3	10	Paris	P2	200
S4	20	London	P2	200
S4	20	London	P4	300

SECOND

SNo	STATUS	CITY
S1	20	London
S2	10	Paris
S3	10	Paris
S4	20	London

SP

SNo	PNO	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S2	P1	300
S3	P2	200
S4	P2	200
S4	P4	300

В отношении FIRST первичный ключ SNo, PNo. Однако, только атрибут QTY (количество) полностью зависит от этого составного ключа. Атрибуты STATUS, CITY частично зависят от указанного ключа, т.к. существуют зависимости

SNo → Status

SNo → City

Поэтому отношение First декомпозировано на два отношения Second и SP.

В отношении Second помещены атрибуты STATUS и CITY, имеющие частичную зависимость от ключа SNo, PNo вместе с их детерминантом SNo.

Если в отношении нет составного первичного ключа и оно находится в 1НФ, то оно автоматически также находится и в 2НФ.

Третья нормальная форма основана на понятии транзитивной зависимости.

Транзитивная зависимость. Если для атрибутов A, B и C некоторого отношения существуют зависимости вида A → B и B → C, то говорят, что атрибут C транзитивно зависит от атрибута A через атрибут B (при условии., что атрибут A функционально не зависит ни от атрибута B, ни от атрибута C).

Рассмотрим отношение “Служащий – Отделение”. В нем существуют следующие функциональные зависимости:

StaffNo(номер следующего) → BranchNo (номер отделения)

и

BranchNo → BAddress (адрес отделения)

SNo → City

City → Status

SNo → Status через City

В этом случае транзитивная зависимость StaffNo → Address осуществляется через атрибут BranchNo. Данное утверждение справедливо, поскольку атрибут StaffNo не зависит функционально от атрибутов BranchNo и BAddress.

Третья нормальная форма. Отношение, которое находится в первой и второй нормальных формах и не имеет находящих в первичный ключ атрибутов, которые находились бы в транзитивной функциональной зависимости от этого первичного ключа.

Пусть в отношении FIRST существует функциональная зависимость

CITY → STATUS,

то есть статус поставщика определяется его местонахождением, например, все поставщики из Лондона должны иметь статус 20.

Тогда отношение SECOND не находится в третьей нормальной форме, так как в нем существует транзитивная зависимость

SNo → STATUS через атрибут CITY.

(SNo → CITY и CITY → STATUS)

Отношение, которое находиться в 2НФ, но не находится в 3НФ, всегда может быть преобразовано в эквивалентный набор (двух) отношений, находящихся в 3НФ. Для этого транзитивно зависимые атрибуты удаляются из такого отношения и помещаются в новое отношение вместе с копией их детерминанта.

SC

SNo	CITY
S1	London
S2	Paris
S3	Paris
S4	London

CS

CITY	STATUS
London	20
Paris	10

Уровень нормализации отношения определяется связями атрибутов, а не их конкретными значениями в некоторое время. Нельзя с первого взгляда на набор данных некоторого отношения однозначно определить, находится ли оно, например, в 3НФ. Для этого необходимо представлять себе существующие между атрибутами зависимости.

Таким образом, процесс нормализации заключается в декомпозиции исходного отношения посредством последовательного выполнения нескольких операции проекции реляционной алгебры. Полученные в результате декомпозиции отношения обеспечивают выполнение их соединения без потерь данных, поэтому данную процедуру называют беспроигрышной. Результаты декомпозиции можно обратить посредством операции (естественного) соединения.

Для реляционной модели необходимо сгенерировать внешние ключи, а также необходимо определить физические имена для всех элементов модели.

Физические имена – это имена объектов сущностей, атрибутов и т.д., приведенные к сокращенной форме для обеспечения надежности реализации. Это можно сделать вручную или с помощью генератора физических имен системы Open ModelSphere.

Однако возможности автоматической генерации физических имен довольно ограничены, генерируемые таким образом имена далеки от совершенства. Поэтому воспользуемся ручным способом. Будем все атрибуты формировать из начальной буквы имени сущности и второго слова имени атрибута. Например, атрибут Books Pages надо представить в форме BPAGES.

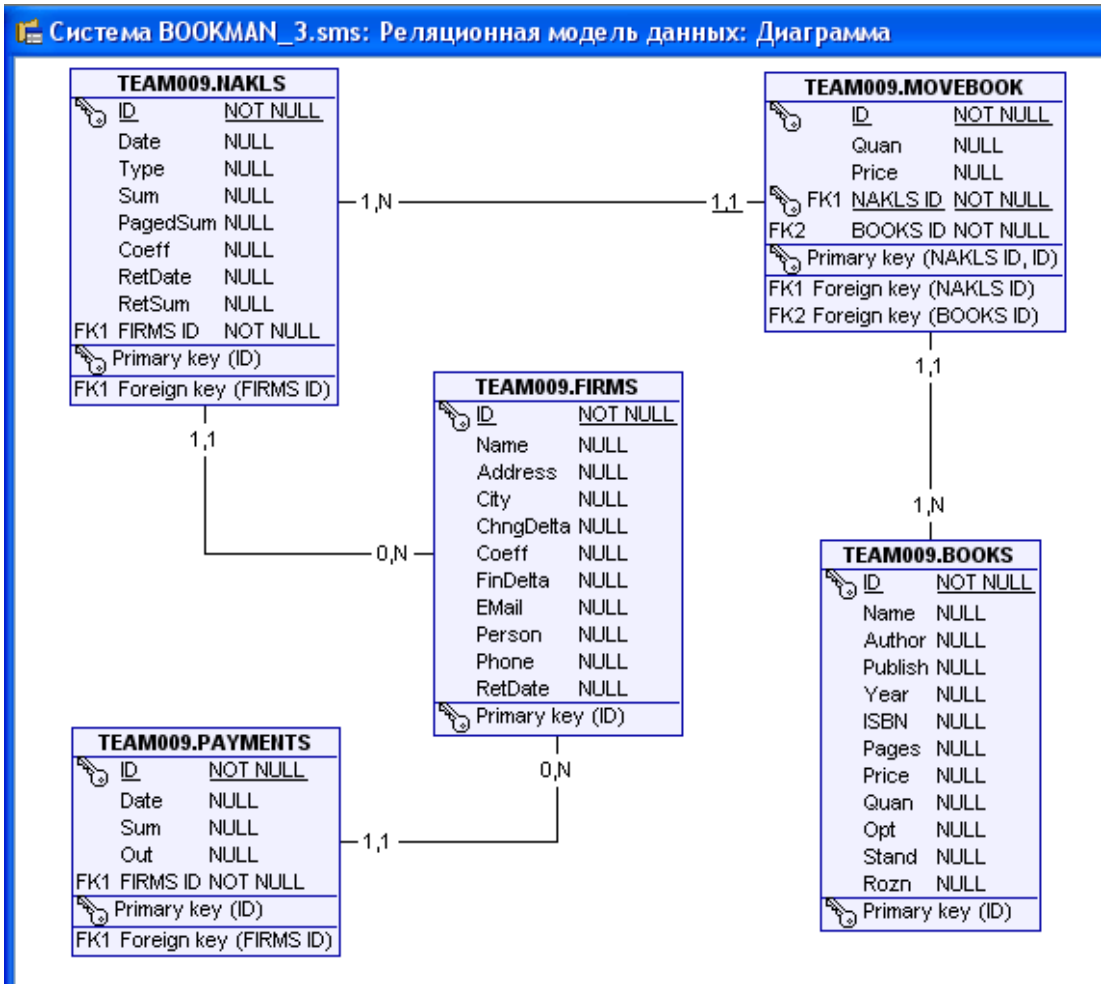


Рис.3 Логическая модель данных

Даталогические модели

Задачей следующей стадии проектирования системы базы данных является выбор подходящей СУБД и отображение в ее среду (структуру данных) спецификаций инфологической модели предметной области. Результатом даталогического проектирования базы данных является концептуальная схема базы данных, включающая определение всех информационных элементов (единиц) и связей, в том числе задание типов, характеристик и имен.

Даталогическое проектирование оперирует логическими понятиями, связанными со структурой базы данных, но особенности представления данных, правила и языки агрегирования и манипулирования данными имеют определяющее влияние. Не все виды связей, например, «многие ко многим», могут быть непосредственно отображены в логической модели.

Может быть много вариантов отображения инфологической модели предметной области в даталогическую модель базы. Следует учитывать влияние двух факторов.

Во-первых, связи предметной области могут отображаться двумя путями: как декларативным — в логической схеме, так и процедурным — отработкой связей через программные модули, обрабатывающие (связывающие) соответствующие хранимые данные.

Во-вторых, существенным фактором может оказаться характер обработки информации. Например, частые обращения к совместно обрабатываемым данным, очевидно, предполагают их совместное хранение, а данные (особенно большого объема), к которым обращаются редко, целесообразно хранить отдельно от часто используемых.

Рассмотрим по шагам общий подход к построению реляционной базы данных на основе инфологической модели, представленной ER-диаграммой.

Получение реляционной схемы из ER-диаграммы

1.Каждая простая сущность превращается в таблицу (отношение). Имя сущности становится именем таблицы.

2.Каждый атрибут становится возможным столбцом с тем же именем. Столбцы, соответствующие необязательным атрибутам, могут содержать неопределенные значения; столбцы, соответствующие обязательным атрибутам, — не могут. Если атрибут является множественным, то для него строится отдельное отношение.

3.Компоненты уникального идентификатора сущности превращаются в первичный ключ. Если имеется несколько возможных уникальных идентификаторов, выбирается наиболее используемый. Если в состав уникального идентификатора входят связи, то к числу столбцов первичного ключа добавляется копия уникального идентификатора сущности, находящейся на дальнем конце связи (этот процесс может продолжаться рекурсивно). Для именования этих столбцов используются имена концов связей и/или имена сущностей.

4.Связи «многие к одному» и «один к одному» становятся внешними ключами. Т.е. создается копия уникального идентификатора с конца связи «один», и соответствующие столбцы составляют внешний ключ.

5. Индексы создаются для первичного ключа (уникальный индекс), а также внешних ключей и тех атрибутов, которые будут часто использоваться в запросах.

6. Если в концептуальной схеме присутствуют подтипы, то возможны два варианта.

Все подтипы хранятся в одной таблице, которая создается для самого внешнего супертипа, а для подтипов создаются представления. В таблицу добавляется по крайней мере один столбец, содержащий код ТИПА, и он становится частью первичного ключа.

Во втором случае для каждого подтипа создается отдельная таблица (для более нижних — представления) и для каждого подтипа первого уровня супертип воссоздается следующим образом: из всех таблиц подтипов выбираются общие столбцы — столбцы супертипа.

7. Если остающиеся внешние ключи все принадлежат одному домену, т. е. имеют общий формат, то создаются два столбца: идентификатор связи и идентификатор сущности. Столбец идентификатора связи используется для различения связей. Столбец идентификатора сущности используется для хранения значений уникального идентификатора сущности на дальнем конце соответствующей связи.

Если результирующие внешние ключи не относятся к одному домену, то для каждой связи, покрываемой дугой исключения, создаются явные столбцы внешних ключей.

Физические модели

Стадия физического проектирования базы данных в общем случае включает:

•выбор способа организации базы данных;

•разработку спецификации внутренней схемы средствами модели данных ее внутреннего уровня;

•описание отображения концептуальной схемы во внутреннюю.

Важно заметить, что в отличие от ранних СУБД, многие современные системы не предоставляют разработчику какого-либо выбора на этой стадии. Реально к вопросам проектирования физической модели можно отнести выбор схемы размещения данных.

Способ хранения базы данных определяется механизмами СУБД автоматически «по умолчанию» на основе спецификаций концептуальной схемы базы данных, и внутренняя схема в явном виде в таких системах не используется.

Внешние схемы базы данных обычно конструируются на стадии разработки приложений.

Лекция №5. Информационные модели. Взаимосвязи в моделях и реляционный подход к построению моделей.

Понятие и классификация моделей данных

Ядром любой базы данных является модель данных. С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.
Модель данных – совокупность структур данных и операций их обработки.
Рассмотрим 3 основных типа моделей данных: **иерархическую, сетевую и реляционную.**

а) иерархическая модель данных

Иерархическая модель базы данных представляет собой совокупность элементов, расположенных в порядке их подчинения от общего к частному и образующих перевернутое дерево (граф).

К основным понятиям иерархической структуры относятся: уровень, элемент (узел), связь.

Узел – это информационная модель элемента, находящегося на данном уровне иерархии. На схеме иерархического дерева узлы представляются вершинами графа.

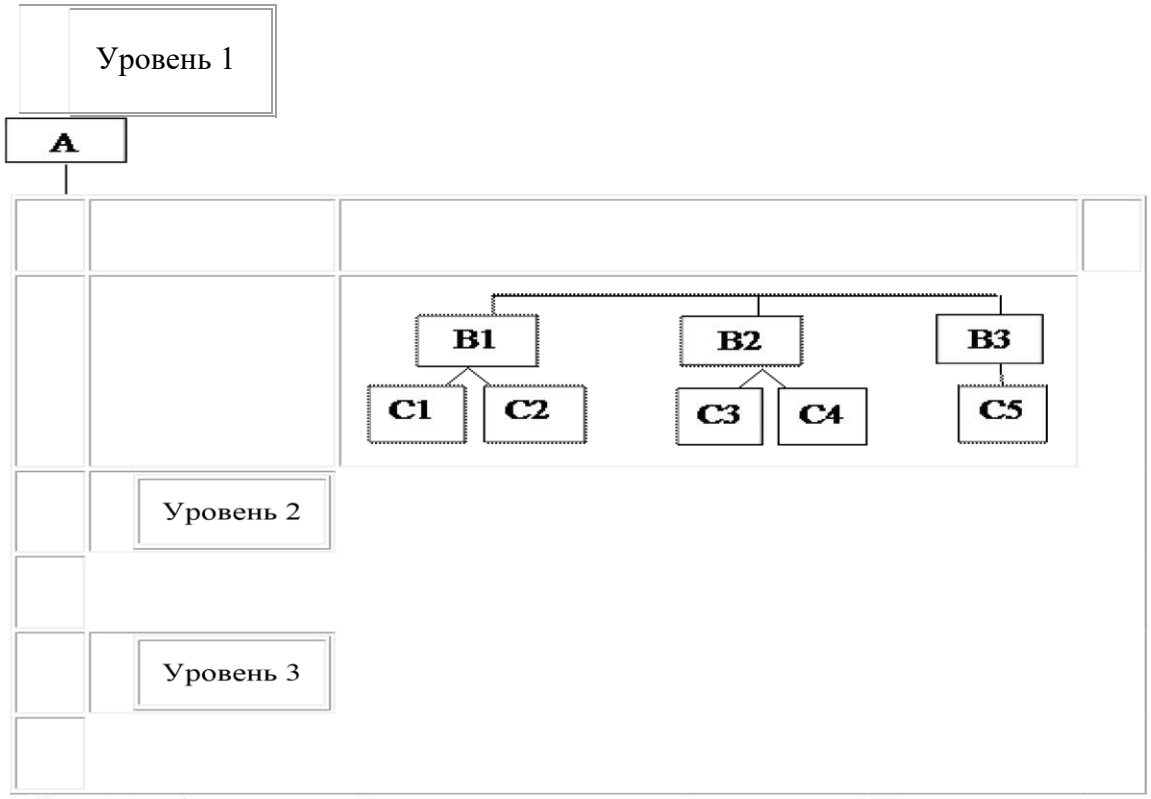
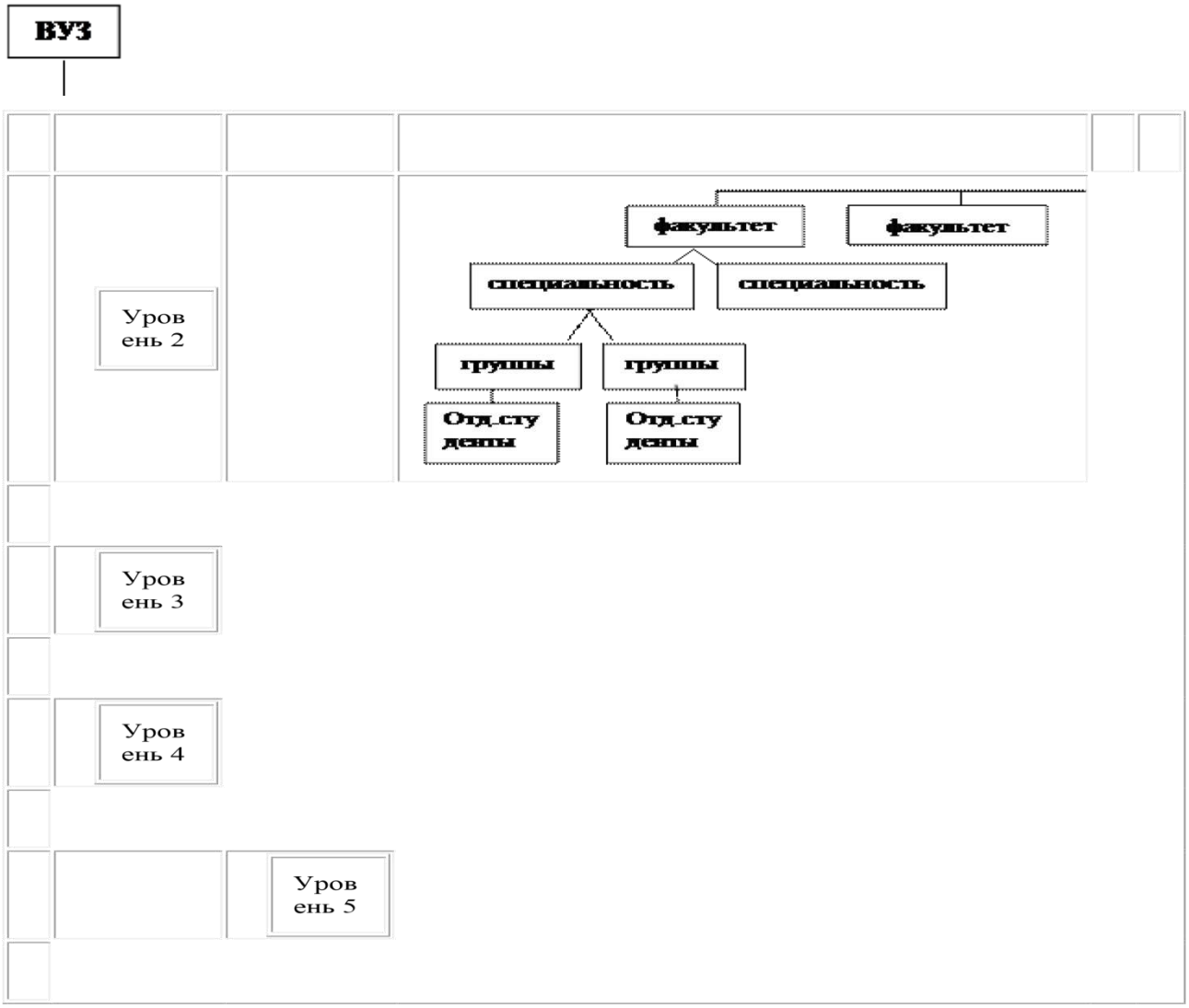


Рис. 1 Графическое изображение иерархической структуры БД

Пример. Иерархическая модель «ВУЗ».



Свойства иерархической модели:

- несколько узлов низшего уровня связано только с одним узлом высшего уровня; - иерархическое дерево имеет только одну вершину (корень дерева), не подчиненный никакой другой вершине и находящуюся на самом верхнем (первом) уровне.

Зависимые (подчиненные) узлы находятся на втором, третьем и т.д. уровнях. - каждый узел имеет свое имя (идентификатор).

- количество деревьев в базе данных определяется числом корневых записей;

- **б) сетевая модель данных:**
- Сетевая модель имеет те же основные составляющие (узел, уровень, связь). Однако в ней принята свободная связь между элементами разных уровней, т.е. каждый элемент может быть связан с любым другим элементом.

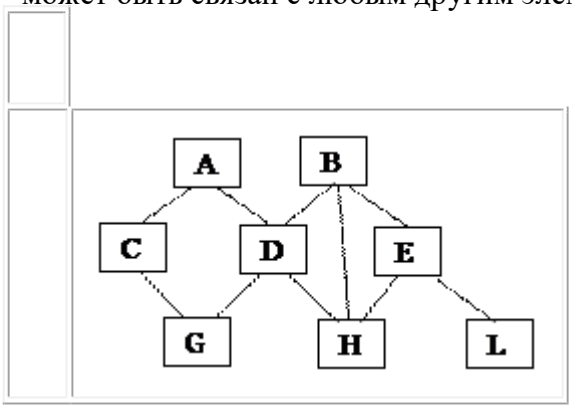


Рис. 2 Графическое изображение сетевой структуры БД

Пример. Сетевая модель «Профессорско-преподавательский состав»

в) реляционная модель данных (табличная)

Термин «реляционный» произошел от англ. слова relation – отношение.

Отношение – математическое понятие, но в терминологии моделей данных отношения удобно изображать в виде таблицы.

Теоретической основой этой модели стала теория отношений американца Чарльза Пирса и немца Эрнеста Шредера. Ими было показано, что множество отношений замкнуто относительно некоторых специальных операций и образует вместе с ними

абстрактную алгебру. Американский математик Э.Ф. Кодд в 1970 г. впервые сформулировал основные понятия и ограничения реляционной модели, ограничив набор операций в ней семью основными и одной дополнительной.

Реляционная модель хранения данных построена на взаимоотношении составляющих ее частей. В простейшем случае она представляет собой двухмерный массив или двухмерную таблицу, а при создании сложных информационных моделей составляет совокупность взаимосвязанных таблиц.

Пример реляционной таблицы:

№ личного дела	Фамилия	Имя	Отчество	Дата рождения	Группа
	Костин	Владимир	Владимирович	01.03.78	БУА
	Антонов	Юрий	Петрович	18.09.80	ФК

Реляционная модель базы данных имеет следующие свойства:

1) каждый элемент таблицы – один элемент данных;
2) все столбцы в таблице являются однородными, т.е. имеют один тип (числа, текст, дата и т.д.)
3) каждый столбец (поле) имеет уникальное имя;
4) одинаковые строки в столбце отсутствуют;
5) порядок следования строк и столбцов может быть произвольным.

Отношения представлены в виде таблиц, строки которых соответствуют кортежам или записям, а столбцы – атрибутам отношений, доменам, полям.

Если реляционная модель данных состоит из нескольких таблиц, то они связываются между собой ключами.

Ключ – поле, которое однозначно определяет соответствующую запись (ключевое поле).

В данном примере в качестве ключа может служить номер личного дела студента.

План лекции:

1.Понятие «информационный объект»

2. Нормализация отношений

3. Построение инфологической модели

Информационный объект – это описание некоторой сущности (реального объекта, явления, процесса, события) в виде совокупности логически связанных реквизитов (информационных элементов). Такими сущностями для информационных объектов могут служить: цех, склад, вуз, студент и т.д.

Информационный объект определенного реквизитного состава и структуры образует класс (тип), которому присваивается уникальное имя (символьное обозначение), например Студент, Сессия, Стипендия.

Информационный объект имеет множество реализаций – экземпляров, каждый из которых представлен совокупностью конкретных значений реквизитов и идентифицируется значением ключа (простого – один реквизит или составного –

несколько реквизитов). Остальные реквизиты информационного объекта являются описательными.

Пример. 1. На рис. 1 представлен пример структуры и экземпляров информационного объекта Студент

В информационном объекте Студент ключом является реквизит Номер (№ личного дела), к описательным реквизитам относятся: Фамилия (фамилия студента), Имя (имя студента), Отчество (отчество студента), Дата (дата рождения), Группа (№ группы). Если отсутствует реквизит Номер, то для однозначного определения характеристик конкретного студента необходимо использование составного ключа из трех реквизитов: Фамилия + Имя + Отчество.

Структура	№ личного дела	Фамилия	Имя	Отчество	Дата	Группа
Экземпляры инф. объекта Студент		Костин	Владимир	Владимирович	01.03.78	БУА
	Антонов	Юрий	Петрович	18.09.80	ФК	

Рис. 1 - Пример структуры и экземпляров информационного объекта

Лекция №6. Нормализация отношений. Этапы проектирования баз данных

При создании базы данных следует определить количество таблиц (отношений) БД и их атрибутный состав, который должен быть рациональным. Суть рациональности заключается в минимизации дублирования данных и упрощении процедуры их обработки и обновления. Классическая технология создания реляционных БД связана с теорией нормализации, которая позволяет улучшить характеристики БД и основана на анализе функциональной зависимости между атрибутами отношений. Процесс нормализации заключается в разложении (декомпозиции) исходных отношений БД на более простые отношения. Каждая ступень этого процесса приводит схему отношений в последовательные нормальные формы. Для каждой ступени нормализации имеются наборы ограничений, которым должны удовлетворять отношения БД. Нормализация позволяет удалить из таблиц базы избыточную неключевую информацию.

При этом можно выделить следующую последовательность нормальных форм:

- первая нормальная форма (1 N.1*);
- вторая нормальная форма (2ИР);
- третья нормальная форма (ЭТУ/);
- нормальная форма Бойса—Кодда (ВСИР)',
- четвертая нормальная форма (4 ИР);
- пятая нормальная форма, или нормальная форма проекции- соединения (5Л^или *В//ИР).*

Основные свойства нормальных форм состоят в том, что каждая следующая нормальная

форма лучше предыдущей и при переходе к следующей нормальной форме свойства предыдущих сохраняются.

Первая нормальная форма (1НФ). Отношение называется нормализованным или приведенным к первой нормальной форме, если все его атрибуты простые (атомарные или неделимые). Примеры

1. Отношение Товар = (Код товара[1]: Название; Цвет; Вес; Цена_за_единицу) находится в 1НФ, так как все атрибуты в данном отношении являются простыми.

2. Отношение Сотрудник = (Таб номер: ФИОсотрудника; Дата рождения; Дети) не находится в 1НФ, так как атрибуты «ФИО сотрудника», «Дата_рождения», «Дети» могут быть сложными:

• атрибут «ФИО_сотрудника» содержит три части — фамилию, имя и отчество сотрудника фирмы, поэтому должен быть разбит на три атомарных атрибута, каждый из которых может использоваться самостоятельно;

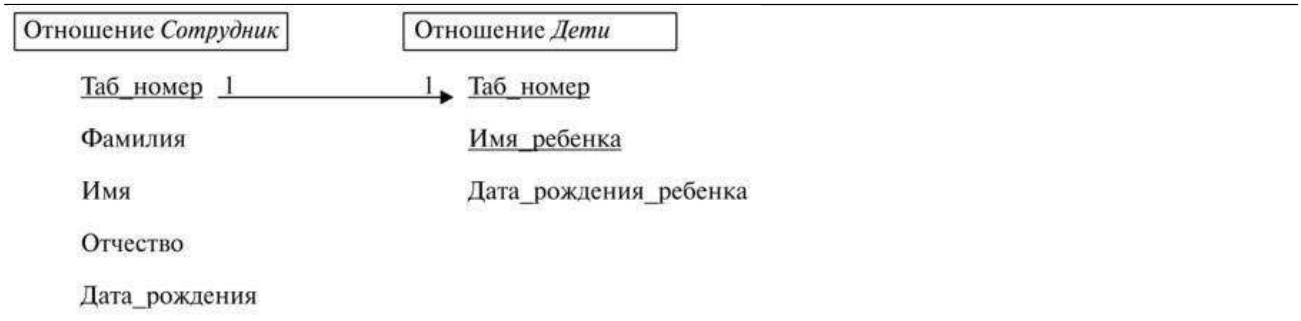
• атрибут «Дата_рождения», как и атрибут «ФИО_сотрудника», не является атомарным, поскольку включает в себя три понятия — «Год», «Месяц» и «День». Однако в БД существуют типы данных — Date и Date Time,которые позволяют выполнять над датами специальные операции, например, <дата_1> — <дата_2> = <количество_дней_ между_ними>. В связи с этим атрибуты дат в базах данных считаются атомарными; • атрибут «Дети» также является сложным, так как может содержать сведения об имени ребенка и его дату рождения.

Для приведения отношения Сотрудник к 1НФ необходимо декомпозировать его на следующие два отношения:

Сотрудник = ГГаб номер: Фамилия; Имя; Отчество; Дата_рож- дения)

Дети = Паб номер: Имя ребенка: Дата_рождения_ребенка)

При этом связь между этими отношениями осуществляется через ключевое поле «Таб_номер», тип связи «один к одному». атрибутов добавляет еще одно — каждый неключевой атрибут должен функционально полно зависеть от первичного ключа (не должен зависеть от части составного ключа).



Вторая нормальная форма (2НФ). Вторая нормальная форма к требованию атомарности Отношение находится во второй нормальной форме, если оно удовлетворяет определению 1НФ и все ее поля, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом.

Примеры

1. Отношение Товар = (Код товара: Название; Цвет; Вес; Цена_ за единицу) находится в 1НФ и во 2НФ одновременно, так как все атрибуты в данном отношении однозначно определены и функционально зависят от ключа «Код_товара».

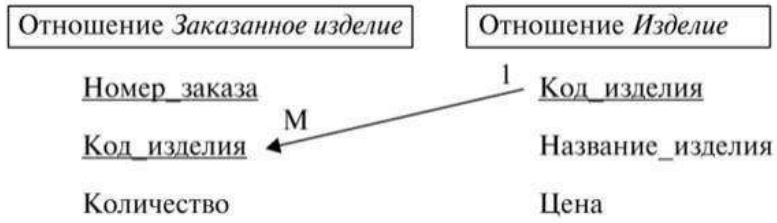
2. Отношение Заказанное изделие = (Номер заказа: Код изделия: Название_изделия; Цена; Количество) не находится во 2НФ, так как атрибут «Количество» зависит от составного ключа (Номер заказа и Код изделия), а атрибуты «Название_изделия» и «Цена» зависят только от части составного ключа «Код_изделия».

Для приведения отношения Заказанное изделие к 2НФ необходимо декомпозировать его на следующие два отношения:

Заказанное_изделие = (Номер заказа: Код изделия: Количество)

Изделие = (Код изделия: Название_изделия; Цена)

Связь между этими отношениями осуществляется через ключевое поле «Код_изделия».



Третья нормальная форма (ЗНФ) подразумевает атомарность и функционально полную зависимость атрибутов каждой сущности от ее первичного ключа. Кроме того, между неключевыми атрибутами сущности должны отсутствовать транзитивные зависимости, т.е. они должны быть взаимно независимы.

Отношение находится в ЗНФ, если оно находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа. Поясним понятие транзитивной зависимости.

Имеем отношение О с атрибутами А, В, С. Если С зависит от В, а В, в свою очередь, зависит от А, то считается, что С транзитивно зависит от А (рис. 2.13).

Преобразование в ЗНФ состоит в декомпозиции исходного отношения на два отношения (рис. 2.14).

Пример

Отношение Сотрудник = (Таб номер: Фамилия; Должность; Оклад; Дата_зачисления; Дата_увольнения) не находится в ЗНФ, так

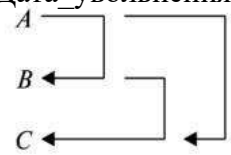


Рис. 2.13. Транзитивная зависимость

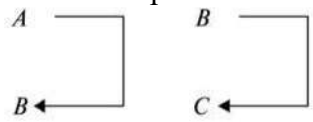


Рис. 2.14. Декомпозиция отношения

как атрибуты «Должность» и «Оклад» находятся в транзитивной зависимости. В этом случае несколько человек могут работать в одной и той же должности и, следовательно, при изменении должностного оклада необходимо менять данные в каждой записи. Также если имеются уникальные должности, то при увольнении сотрудника потеряется информация об окладе по этой должности. Кроме того, сотрудник может переходить с одной должности на другую.

Для приведения отношения Сотрудник к ЗНФ необходимо декомпозировать его на следующие отношения:

Сотрудник = (Таб номер: Фамилия)

Должность = (Код должности: Должность; Оклад)

Должностные_перемещения = (Таб номер: Код должности:

Датазачисления; Дата_увольнения)

Отношение Сотрудник Отношение Должность

Связь между отношениями Сотрудник и Должностные перемещения осуществляется через ключевое поле «Таб_номер», между отношениями Должость и Должностные_ перемещения — через ключевое поле «Код_должности».

Лекция №7. Работа с объектами(таблицы, запросы, формы, отчеты и макросы)

Компьютерная база данных — это хранилище объектов. В одной базе данных может быть больше одной таблицы. Например, система отслеживания складских запасов, в которой используются три таблицы, — это не три базы данных, а одна. В базе данных Access (если ее специально не настраивали для работы с данными или кодом, принадлежащими другому источнику) все таблицы хранятся в одном файле вместе с другими объектами, такими как формы, отчеты, макросы и модули. Для файлов баз данных, созданных в формате Access 2007 (который также используется в Access 2016, Access 2013 и Access 2010), используется расширение ACCDB, а для баз данных, созданных в более ранних версиях Access, — MDB. С помощью Access 2016, Access 2013, Access 2010 и Access 2007 можно создавать файлы в форматах более ранних версий приложения (например, Access 2000 и Access 2002–2003).

Использование Access позволяет:

- добавлять новую информацию в базу данных, например новый артикул складских запасов;
- изменять информацию, уже находящуюся в базе, например перемещать артикул;
- удалять информацию, например если артикул был продан или утилизирован;
- упорядочивать и просматривать данные различными способами;
- обмениваться данными с другими людьми с помощью отчетов, сообщений электронной почты, внутренней сети или Интернета.

Элементы базы данных Access:

Таблицы

Таблица базы данных похожа на электронную таблицу — и там, и там информация расположена в строках и столбцах. Поэтому импортировать электронную таблицу в таблицу базы данных обычно довольно легко. Основное различие заключается в том, как данные структурированы.

Чтобы база данных была как можно более гибкой и чтобы в ней не появлялось излишней информации, данные должны быть структурированы в виде таблиц. Например, если речь идет о таблице с информацией о сотрудниках компании, больше одного раза вводить данные об одном и том же сотруднике не нужно. Данные о товарах должны храниться в отдельной таблице, как и данные о филиалах компании. Этот процесс называется *нормализацией*.

Строки в таблице называются записями. В записи содержатся блоки информации. Каждая запись состоит по крайней мере из одного поля. Поля соответствуют столбцам в таблице. Например, в таблице под названием "Сотрудники" в каждой записи находится информация об одном сотруднике, а в каждом поле — отдельная категория информации, например имя, фамилия, адрес и т. д. Поля выделяются под определенные типы данных, например текстовые, цифровые или иные данные.

Записи и поля можно описать по-другому. Представьте старый библиотечный карточный каталог. Каждой карточке в шкафу соответствует *запись* в базе данных. Блоки информации на карточке (автор, название книги и т. д.) соответствуют *полям* в базе данных.

Формы

С помощью форм создается пользовательский интерфейс для ввода и редактирования данных. Формы часто содержат кнопки команд и другие элементы управления, предназначенные для выполнения различных функций. Можно создать базу данных, не используя формы, если просто отредактировать уже имеющуюся информацию в таблицах Access. Тем не менее, большинство пользователей предпочитает использовать формы для просмотра, ввода и редактирования информации в таблицах.

С помощью кнопок команд задаются данные, которые должны появляться в форме, открываются прочие формы и отчеты и выполняется ряд других задач. Например, есть "Форма клиента", в которой вы работаете с данными о клиентах. И в ней может быть кнопка, нажатием которой открывается форма заказа, с помощью которой вы вносите информацию о заказе, сделанном определенным клиентом.

Формы также дают возможность контролировать взаимодействие пользователей с информацией базы данных. Например, можно создать форму, в которой отображаются только определенные поля и с помощью которой можно выполнять только ограниченное число операций. Таким образом обеспечивается защита и корректный ввод данных.

Отчеты

Отчеты используются для форматирования, сведения и показа данных. Обычно отчет позволяет найти ответ на определенный вопрос, например "Какую прибыль в этом году принесли нам наши клиенты?" или "В каких городах живут наши клиенты?" Отчеты можно форматировать таким образом, чтобы информация отображалась в наиболее читабельном виде.

Отчет можно сформировать в любое время, и в нем всегда будет отображена текущая информация базы данных. Отчеты обычно форматируются таким образом, чтобы их можно было распечатать, но их также можно просматривать на экране, экспортировать в другие программы или вкладывать в сообщения электронной почты.

Запросы

Запросы могут выполнять множество функций в базе данных. Одна из их основных функций — находить информацию в таблицах. Нужная информация обычно содержится в нескольких таблицах, но, если использовать запросы, ее можно просматривать в одной. Кроме того, запросы дают возможность фильтровать данные (для этого задаются критерии поиска), чтобы отображались только нужные записи.

Используются и так называемые "обновляемые" запросы, которые дают возможность редактировать данные, найденные в основных таблицах. При работе с обновляемым запросом помните, что правки вносятся в основные таблицы, а не только в таблицу запроса.

Есть два основных вида запросов: запросы на выборку и на изменение. Запрос на выборку только находит данные и предоставляет к ним доступ. Результаты такого запроса можно просмотреть на экране, распечатать или скопировать в буфер обмена, а также использовать в качестве источника записей для формы или отчета.

С помощью запроса на изменение, как видно из названия, можно выполнять определенные операции с найденными данными: создавать таблицы, добавлять информацию в уже существующие таблицы, а также обновлять или удалять данные.

Макросы

Макросы в Access — это нечто вроде упрощенного языка программирования, с помощью которого можно сделать базу данных более функциональной. Например, если к кнопке команды в форме добавить макрос, то он будет запускаться всякий раз при нажатии этой кнопки. Макросы состоят из команд, с помощью которых выполняются определенные задачи: открываются отчеты, выполняются запросы, закрывается база данных и т. д. Используя макросы, можно автоматизировать большинство операций, которые в базе данных вы делаете вручную, и, таким образом, значительно сэкономить время.

Модули

Подобно макросам, модули — это объекты, с помощью которых базу данных можно сделать более функциональной. Но если макросы в Access составляются путем выбора из списка макрокоманд, модули создаются на языке Visual Basic для приложений (VBA). Модули представляют собой наборы описаний, инструкций и процедур. Существуют модули класса и стандартные модули. Модули класса связаны с конкретными формами или отчетами и обычно включают в себя процедуры, которые работают только с этими формами или отчетами. В стандартных модулях содержатся общие процедуры, не связанные ни с каким объектом. Стандартные модули, в отличие от модулей класса, перечисляются в списке **Модули** в области навигации.

5-семестр

Лекция №8. Классификация и характеристика СУБД.
Обзор современных СУБД.

Системы управления базами данных можно классифицировать:

по используемому языку общения:

- замкнутые, имеющие собственные самостоятельные языки общения пользователей с БД. Они обеспечивают непосредственное общение с системой в режиме диалога, позволяют работать без программистов;
- открытые, в которых для общения с БД используется язык программирования, «расширенный» операторами языка манипулирования данными (ЯМД). В этом случае необходимо присутствие квалифицированного программиста.

по числу поддерживаемых СУБД уровней моделей данных: одно-, двух-, трехуровневые системы. Теоретически обоснован выбор трехуровневой архитектуры данных, однако на

практике СУБД для персональных ЭВМ часто объединяют концептуальный и внутренний уровни представления. по выполняемым функциям:

- операционные, предполагающие иные виды обработки по получению информации, не хранящейся в явном виде в БД;
- информационные, позволяющие организовать хранение данных, поиск и выдачу нужных данных из БД и поддерживать их целесообразность и актуальность; по сфере применения:
- универсальные, настраиваемые на любую предметную область путѐм создания соответствующей БД и прикладных программ;
- проблемно-ориентированные на определенные процедуры обработки данных, присущих конкретной области применения; по допустимым режимам работы:
- пакетные;
- с использованием телеобработки. Программные средства баз данных

Оболочки информационных систем (системы программирования ИС) представляют гибкие программные комплексы, настраиваемые на задачи пользователя. Наиболее распространѐенными классами данных программных средств являются СУБД и оболочки автоматизированных информационно-поисковых систем (АИПС).

Информационно-поисковые системы

В узком смысле под АИПС принято понимать открытый (обычно) или замкнутый (реже) программный продукт, предназначенный для реализации практически большинства функций (процессов): ввод, обработка, хранение, поиск, представление данных (организованных в записи или документы, находящиеся в БД). В этом смысле часто отождествляют АИПС с АИС, и это трудно оспаривать.

Среди АИПС в узком смысле принято выделять:

- фактографические системы с фиксированной структурой данных или записей, для разработки которых, как правило, используются СУБД, поддерживающие табличные (реляционные) БД;
- документальные системы с неопредѐленной или переменной структурой данных или документов, для разработки которых часто (но не обязательно) применяют оболочки АИПС.

В более широком смысле под АИПС подразумеваются также программные оболочки, ориентированные на разработку продуктов типа АИПС. Это связанно с тем фактом, что первые системы типа СУБД и оболочек АИПС были предложены в 1960–1970-е гг. фирмой IBM (и сотрудничавшими с ней организациями) и включали в себя: ● IMS/360 (Information Management System) – по-видимому, первая реальная СУБД, поддерживавшая т.н. иерархическую модель данных(понятие появилось позже, в связи с необходимостью систематизации СУБД), нашедшая достаточно широкое применение, в частности, для информационного обеспечения проекта Apollo, завершившегося, как известно, высадкой граждан США на Луну в 1969 г.;

- DPS/360 (Document Processing System) – первый промышленный ППП, предназначенный для реализации документальных АИПС. В дальнейшем путѐм развития принципов DPS, фирма в 1972 г. выпустила пакет STAIRS (STorage And Information Retrieval System),

предназначенный для диалогового обслуживания множества (удалѐнных) пользователей;

- IRMS (Information Retrieval and Management System), TEXT-

PAC идругиеаналогичныепакеты.

Как следует из наименований продуктов, разработчики понимали под АИПС именно ППП оболочки.

Системы управления базами данных и программирования АИС

Среди различных программных средств данного класса различают три типа:

- СУБД в «чистом виде» (IMS, CETOP и пр.);
- СУБД с элементами систем программирования АИС (ADABAS/NATURAL, ORACLE);
- системы программирования АИС с элементами СУБД (FoxBase / FoxPro, Clipper).

Первый тип фактически относится к начальному этапу развития систем второго (реже – третьего) типов. В этом случае СУБД состоит только из системы интерпретации вызовов (обращений) из пользовательской программы (call-interface) на выборку (корректировку, занесение) информации из/в БД, причѐм программа написана на одном из таких универсальных языков программирования (ЯП), как Кобол, Фортран, Паскаль и пр., получивших название включающие языки СУБД. Данная система в последующих СУБД (второй тип) получила наименование ядра. Соглашения о форматах и структурах такого взаимодействия обычно пытаются оформить в виде некоторого формального языка (языка ядра).

Второй тип представляет расширение первого в направлении создания универсальной системы разработчика АИС, включающей также специализированные языковые средства. В этом случае СУБД представляет совокупность специализированных программных средств, вспомогательных файлов и управляющих таблиц (иногда находящихся в составе БД, реже это файлы ОС), обеспечивающих доступ пользователей к БД при соблюдении следующих существенных критериев: целостность и непротиворечивость данных, описывающих различные аспекты объектов реального мира, защиту информации от несанкционированного доступа на чтение/обновление содержимого БД, установление и поддержание связей между зависимыми данными, удобство использования данных.

Третий тип представляют (разработанные обычно для ПК) системы, содержащие как элементы непроцедурного типа (язык запросов), так и процедурного (язык программирования) во входном языке, предназначенном для управления данными и обработки информации. Элементы СУБД здесь также заключаются в наличии простейшего словаря данных, возможности создания модели предметной области в форме совокупности таблиц, связанных между собой простейшим образом, а также в наличии средств генерации отчетов и управления доступом пользователей.

Лекция №9. Введение в SQL. Краткая характеристика языка SQL. Операторы языка SQL для работы с реляционной базой данных.

SQL (**Structured Query Language**) — язык структурированных запросов.

SQL (**Structured Query Language**, **язык структурированных запросов**) — это специальный язык, используемый для определения данных, доступа к данным и их обработки.

Язык SQL относится к непроцедурным (nonprocedural) языкам — он лишь описывает нужные компоненты (например, таблицы) и желаемые результаты, не указывая, как именно эти результаты должны быть получены. Каждая реализация **SQL** является надстройкой над процессором базы данных (database engine), который интерпретирует операторы **SQL** и определяет порядок обращения к структурам БД для корректного и эффективного формирования желаемого результата.

Стандарт **SQL** определяется ANSI — American National Standarts Institute (Американским Национальным Институтом Стандартов) и в настоящее время принят ISO — International Standarts Organization (Международной Организацией по Стандартизации).

SQL — непроцедурный **язык**: серверу базы данных сообщается, что нужно сделать и каким образом. Для обработки запроса сервер базы данных транслирует команды **SQL** во внутренние процедуры. Благодаря тому, что **SQL** скрывает детали обработки данных, его легко использовать.

Что можно делать с помощью SQL?

- **SQL** позволяет создавать таблицы данных.
- **SQL** позволяет хранить данные.
- **SQL** позволяет получать данные.
- **SQL** позволяет изменять данные.
- **SQL** позволяет изменять структуру таблиц.
- **SQL** позволяет объединять данные.
- **SQL** позволяет выполнять вычисления.
- **SQL** позволяет обеспечивать защиту данных.

Команды SQL

Команды SQL разделяются на следующие группы:

- Команды языка определения данных — **DDL (Data Definition Language).**
  Эти **SQL** команды можно использовать для создания, изменения и удаления различных объектов базы данных.
- Команды языка управления данными — **DCL (Data Control Language).** С помощью этих **SQL** команд можно управлять доступом пользователей к базе данных и использовать конкретные данные (таблицы, представления и т.д.).
- Команды языка управления транзакциями — **TCL (Transaction Control Language).**
  Эти **SQL** команды позволяют определить исход транзакции.
- Команды языка манипулирования данными — **DML (Data Manipulation** Language).
  Эти **SQL** команды позволяют пользователю перемещать данные в базу данных и из нее.

Определение структур базы данных

- **Определение структур базы данных (DDL)**
  Язык определения данных (**Data Definition Language, DDL**) позволяет создавать и изменять структуру объектов *базы данных*, например, создавать и удалять *таблицы*. Основными командами языка DDL являются следующие: **CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX, ALTER INDEX, DROP INDEX.**

- **Манипулирование данными (DML)**
  Язык манипулирования данными (Data Manipulation Language, DML) используется для манипулирования информацией внутри объектов *реляционной базы данных* посредством трех основных команд: **INSERT, UPDATE, DELETE.**

- **Выборка данных (DQL)**
  Язык *запросов* DQL наиболее известен пользователям *реляционной базы данных*, несмотря на то, что он включает всего одну команду **SELECT**. Эта команда вместе со своими многочисленными опциями и предложениями используется для формирования *запросов* к *реляционной базе данных*.

- **Язык управления данными (DCL - Data Control Language)**
  Команды управления данными позволяют управлять доступом к информации, находящейся внутри *базы данных*. Как правило, они используются для создания объектов, связанных с доступом к данным, а также служат для контроля над распределением привилегий между пользователями. Команды управления данными следующие: **GRANT,REVOKE.**

- **Команды администрирования данных**
  С помощью команд администрирования данных пользователь осуществляет контроль за выполняемыми действиями и анализирует операции *базы данных* ; они также могут оказаться полезными при анализе производительности системы. Не следует путать администрирование данных с администрированием *базы данных*, которое представляет собой общее управление *базой данных* и подразумевает использование команд всех уровней.

- **Команды управления транзакциями**
  Существуют следующие команды, позволяющие управлять транзакциями *базы данных*: **COMMIT, ROLLBACK, SAVEPOINT, SET TRANSACTION.**

ГЛОССАРИЙ

OLAP (Online Analytical Processing) - оперативный анализ данных. Этот метод обработки применяется с целью ускорения обработки запросов и предусматривает предварительный расчет часто запрашиваемых данных (например, сумм или значений счетчика).

OLTP (Online Transaction Processing) - обработка транзакций в реальном времени. Способ организации БД, при котором система работает с небольшими по размерам транзакциями, и при этом клиенту требуется от системы максимально быстрое время ответа.

PHP (Hypertext Preprocessor) – препроцессор гипертекста. Язык сценариев с открытым исходным кодом, операторы которого могут встраиваться в код HTML. В качестве операторов могут использоваться конструкции SQL.

SQL (Structured Query Language) - язык структурированных запросов, язык S0L. Является принятым в отрасли баз данных стандартом для выполнения операций выборки данных из реляционных БД.

QBE (Query by example) — способ создания запросов к базе данных, с использованием образцов в виде текстовой строки, названия документа или списка документов.

XML (extended markup language) – расширяемый язык разметки (язык для описания других языков). Текстовый формат представления данных. Используется для оформления Web документов.

Агрегат данных - поименованная совокупность элементов данных внутри записи, которую можно рассматривать как единое целое.

Активная база - БД называется активной, если СУБД по отношению к ней выполняет не только те действия, которые явно указывает пользователь, но и дополнительные действия в соответствии с правилами, заложенными в саму БД.

Апплет – Web-программа, передаваемая клиенту сервером и выполняемая на клиенте.

Атрибут – поименованный столбец отношения.

Аудит — одна из форм контроля работоспособности БД.

Аутентификация – это процедура, проверяющая, имеет ли пользователь с предъявленным идентификатором право на доступ к ресурсу.

Авторизация - предоставление лицу, прошедшего аутентификацию, некоторых прав или проверка их наличия (как правило — следующий шаг системы после аутентификации).

База данных – совместно используемый набор логически связанных данных (и описание этих данных), предназначенный для удовлетворения информационных потребностей организации.

Банк данных – совокупность нескольких баз данных с программами управления ими и совместимыми аппаратными средствами.

Блокировка - временное ограничение (захват ресурса) на выполнение некоторых операций обработки данных. Блокировка представляет собой метод управления параллельными процессами, при котором объект БД не может быть модифицирован без ведома транзакции, т.е. происходит блокирование доступа к объекту со стороны других транзакций, чем исключается непредсказуемое изменение объекта.

Браузер – программа, позволяющая просматривать документы в стандарте HTML. Веб-брамузер программное обеспечение для просмотра веб-сайтов.

Веб-сервер — это сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, обычно вместе с HTML-страницей.

Взаимоблокировка – тупиковая ситуация, которая может возникнуть, когда две (или более) транзакции находятся во взаимном ожидании освобождения блокировок, удерживаемых друг другом.

Внешний ключ – атрибут или множество атрибутов внутри отношения, которое соответствует потенциальному ключу некоторого (может быть того же самого) отношения. Внешний ключ должен быть определен на тех же доменах, что и соответствующий первичный ключ родительского отношения.

Внешний уровень – представление базы данных с точки зрения пользователей. Этот уровень описывает часть базы данных, которая относится к определенному пользователю.

Внутренний уровень – физическое представление базы данных в компьютере. Этот уровень описывает, как информация хранится в базе данных.

Время доступа – одна из основных характеристик запоминающих устройств, показывающая время, необходимое для извлечения информации из ЗУ или записи информации в него; определяется по-разному для различных типов запоминающих устройств.

Временная отметка - уникальный идентификатор, создаваемый СУБД с целью обозначения относительного момента времени запуска транзакции.

Данные – информация, записанная (закодированная) на "языке машины".

Двухфазная блокировка – транзакция выполняется по протоколу двухфазной блокировки, если в ней все операции блокирования предшествуют первой операции разблокирования.

Декларативная поддержка ограничений целостности - заключается в определении ограничений средствами языка определения данных (DDL - Data Definition Language). Обычно средства декларативной поддержки целостности (если они имеются в СУБД) определяют ограничения на значения доменов и атрибутов, целостность сущностей (потенциальные ключи отношений) и ссылочную целостность (целостность внешних ключей).

Дескриптор – одно слово или словосочетание, заменяющее в определенном контексте множество связанных по смыслу слов и словосочетаний, выражающих одну и ту же мысль.

Домен – подмножество значений данного типа, допустимых для данного атрибута.

Доступ, адресный – способ доступа к запоминающему устройству, при котором местоположение извлекаемой или записываемой в ЗУ информации определяется ее адресом (обычно некоторым положительным целым числом).

Запись – одна строка в хранимой таблице базы данных.

Защита баз данных - понимается способ предотвратить несанкционированный доступ к информации, хранимой в таблицах.

Идентификация – отождествление, признание тождественности по совокупности общих и частных признаков.

Избирательное управление доступом (Discretionary access control, DAC) — управление доступом субъектов к объектам на основе списков управления доступом.

Иерархия - это разновидность сети, являющаяся совокупностью деревьев (лесом), в которой все связи направлены от отца к сыну.

Иерархическая модель данных – совокупность элементов базы данных, образующих граф (дерево).

Индекс - структура данных (и механизмы доступа), которая помогает СУБД быстрее обнаружить отдельные записи в файле и сократить время выполнения запросов пользователей.

Индексирование базы данных – процесс формирования индекса. Один из способов повышения производительности СУБД.

Индексированные файлы базы данных – способ организации файла записей с помощью которого осуществляется поддержание файла в логически отсортированном (по значениям атрибута (ов)) порядке.

Инкапсуляция - ограничивает область видимости имени атрибута пределами того объекта, в котором оно определено. Смысл этого атрибута будет определяться тем объектом, в котором оно инкапсулировано.

Интероперабельность - Возможность упрощения комплексирования новых программных систем на основе использования готовых компонентов со стандартными интерфейсами.

Интерфейс – сопряжение средств объектов информатики (информации, данных, программ, аппаратуры, конечного пользователя), в котором все информационные, логические, физические и электрические параметры отвечают предварительно выработанным соглашениям (стандартизованным протоколам), для обеспечения программно-аппаратной и эргономической совместимости.

Информационная система (ИС) – система, реализующая сбор, обработку и манипулирование данными и включающая технические средства обработки данных, программное обеспечение и персонал.

Информационная безопасность - состояние (качество) определённого объекта (в качестве объекта может выступать данные и ресурсы автоматизированной системы).

Информационная совокупность – группа данных, характеризующих объект, процесс, операцию.

Информационная технология – совокупность методов, производственных процессов и программно-технических средств, объединенных в технологическую цепочку, обеспечивающих сбор, хранение, обработку, вывод и распространение информации для снижения трудоемкости процессов использования информационного ресурса, повышения их надежности и оперативности.

Информационный запрос – текст на естественном языке, выражающий определенную потребность в информации.

Информационный поиск – процесс извлечения информации из информационной системы в соответствии с признаками этой информации.

Инфологическая модель предметной области – модель формализованного интерфейса между специалистами, владеющими предметной стороной предметной области, и специалистами, осуществляющими проектирование базы данных, безотносительно к их содержанию и среде хранения.

Кардинальность – Количество кортежей отношения.

Классификация - разделение множества на подмножества по неформально предложенному признаку.

Ключ (в базе данных) – одно или несколько полей, по которым можно однозначно найти любую запись.

Ключевое слово – слово естественного языка, выражающее в заданном контексте смысл существа излагаемого вопроса.

Контекстный поиск – возможность поиска информации и любых понятий в наборе документов, в отдельном документе или его фрагменте, а также в базе данных при контекстном индексировании последних.

Концептуальный уровень – обобщенное представление базы данных с точки администратора базы (обобщение и агрегирование точек зрения пользователей). Этот уровень описывает данные и связи базы данных, которые относятся к автоматизируемому объекту.

Концептуальное проектирование БД – описание и синтез информационных требований пользователей в терминах целевой СУБД.

Кортеж – строка отношения.

Кэш - это временное хранилище (оперативная память) часто используемых данных.

Логическая независимость от данных – означает защищенность концептуальной схемы от изменений, вносимых во внутреннюю схему. Представление данных в приложении не должно зависеть от структуры реляционных таблиц.

Логическое проектирование БД – преобразование формализованных данных в структуру СУБД.

Локальные БД – БД, располагающаяся на одном компьютере.

Мандатное управление доступом (Mandatory access control, MAC) — разграничение доступа субъектов к объектам, основанное на назначении метки.

Масштабируемость – гибкость системы по отношению к числу одновременно работающих пользователей.

Метаданные – данные, описывающие данные. Метаданные включают описания элементов данных, типов данных, атрибутов/свойств, подчиненности/месторасположения, процессов/методов и др.

Метод доступа - действия, выполняемые при сохранении или извлечении записей из файла.

Метод физического доступа - совокупность программных средств, обеспечивающих возможность хранения и выборки данных, расположенных на физических устройствах.

Метод временных меток – протокол управления параллельным выполнением транзакций, основная цель которого состоит в установлении глобальной очередности выполнения транзакций, при которой более старые транзакции (транзакции с меньшим значением временной отметки) имеют более высокий приоритет при разрешении возникающих конфликтов.

Множество - это неопределяемое понятие, представляющее некоторую совокупность данных. Элементы множества должны отличаться друг от друга.

Мобильность - сравнительная простота переноса программной системы в широком спектре аппаратно-программных средств, соответствующих стандартам.

Моделирование (лат. modulus – мера, образец, норма) – метод исследования объектов различной природы на их аналогах (моделях) для определения или уточнения характеристик существующих или вновь конструируемых объектов. Модель может выступать гносеологическим заместителем оригинала на четырех уровнях: элементов, структур, поведения (или функций), результатов.

Модель данных – интегрируемый набор понятий для описания данных, связей между ними и ограничений, накладываемых на данные в некоторой предметной области, с указанием операций над объектами модели.

Монопольная блокировка (X-блокировки, X-locks - eXclusive locks) - блокировки без взаимного доступа (блокировка записи).

Мощность отношения - это мощность множества кортежей отношения (аналог количества строк в таблице).

Наследование. - процесс порождения нового класса на основе уже существующего класса. В этом случае новый класс, называемый подклассом существующего класса, наследует все атрибуты и методы класса. В подклассе, кроме того, могут быть определены дополнительные атрибуты и методы.

Независимость данных - возможность изменения логической и физической структуры БД без изменения представлений пользователей.

Независимость контроля целостности - вся информация, необходимая для поддержания целостности, должна находиться в словаре данных. Язык для работы с данными должен выполнять проверку входных данных и автоматически поддерживать целостность данных.

Непроцедурный язык DML – язык, который позволяет указать лишь то, какие данные требуются, но не то, как их следует извлекать.

Неупорядоченный файл (который иногда называют кучей) - записи размещаются в файле в том порядке, в котором они в него вставляются. Каждая новая запись помещается на последнюю страницу файла, а если на последней странице для нее не хватает места, то в файл добавляется новая страница.

Нормализация отношений в базах данных – формальный аппарат установления ограничений целостности, позволяющий устранить дублирование и противоречивость хранимой информации.

Объект – конкретная материализация абстракции; сущность с хорошо определенными границами, в которой инкапсулированы состояние и поведение.

Ограничение целостности - некоторое утверждение, которое может быть истинным или ложным в зависимости от состояния базы данных. Любое ограничение целостности является семантическим понятием, т.е. появляется как следствие определенных свойств объектов предметной области и/или их взаимосвязей.

Ограничения целостности проверяемые немедленно - проверяются непосредственно в момент выполнения операции, могущей нарушить ограничение. Например, проверка уникальности потенциального ключа проверяется в момент вставки записи в таблицу. Если ограничение нарушается, то такая операция отвергается. Транзакция, внутри которой произошло нарушение немедленно проверяемого утверждения целостности, обычно откатывается.

Ограничения целостности с отложенной проверкой - проверяется в момент фиксации транзакции оператором COMMIT. Внутри транзакции ограничение может не выполняться. Если в момент фиксации транзакции обнаруживается нарушение ограничения с отложенной проверкой, то транзакция откатывается.

Ограничение целостности атрибута - представляют собой ограничения, накладываемые на допустимые значения атрибута вследствие того, что атрибут основан на каком-либо домене. Ограничение атрибута в точности совпадают с ограничениями соответствующего домена. Отличие ограничений атрибута от ограничений домена в том, что ограничения атрибута проверяются.

Ограничения целостности домена - представляют собой ограничения, накладываемые только на допустимые значения домена.

Ограничения целостности кортежа - представляют собой ограничения, накладываемые на допустимые значения отдельного кортежа отношения, и не являющиеся ограничением целостности атрибута.

Ограничение PRIMARY KEY. - ограничение PRIMARY KEY для таблицы или столбца означает, что группа из одного или нескольких столбцов образуют потенциальный ключ таблицы. Это означает, что комбинация значений в PRIMARY KEY должна быть уникальной для каждой строки таблицы. Дублированные значения или значения, содержащие NULL, будут отвергнуты. Для одной таблицы может быть определено единственное ограничение PRIMARY KEY. В терминах стандарта SQL это называется первичным ключом таблицы.

Ограничение UNIQUE. -ограничение UNIQUE для таблицы или столбца означает, что группа из одного или нескольких столбцов образуют потенциальный ключ таблицы, в котором допускаются значения NULL. Это означает, что две строки, содержащие одинаковые и не равные NULL-значения, считаются нарушающими уникальность и не допускаются. Две строки, содержащие NULL-значения считаются различными и допускаются. Для одной таблицы может быть определено несколько ограничений UNIQUE.

Организация файла - физическое распределение данных файла по записям и страницам на вторичном устройстве хранения. Распределение данных подразделяют на: последовательное и прямое (хешированное - с помощью функции хеширования).

Открытый код – программное обеспечение (ПО) с открытым кодом означает, что применять и модифицировать его может любой желающий. Такое ПО можно получать по Internet и использовать бесплатно.

Отношение – подмножество декартова произведения. Плоская таблица, состоящая из столбцов и строк, недопускающая дублированных строк.

Первичный ключ – потенциальный ключ, который выбран для уникальной идентификации кортежей внутри отношения.

Поле (в базе данных) – один столбец таблицы (для реляционной базы данных).

Полиморфизм – способность одного и того же программного кода работать с разнообразными данными. Другими словами, он допускает возможность в объектах разных типов иметь методы (процедуры или функции) с одинаковыми именами. Во время выполнения объектной программы одни и те же методы оперируют с разными объектами в зависимости от типа аргумента.

Постреляционная модель данных – расширенная реляционная модель, снимающая ограничение неделимости (атомарности) данных, хранящихся в таблицах БД.

Потенциальный ключ – суперключ, который не содержит подмножества, так же являющегося суперключом данного отношения. Потенциальный ключ отношения - это набор атрибутов отношения, обладающий свойствами уникальности и неизбыточности. Доступ к конкретному кортежу можно получить, лишь зная значение потенциального ключа для этого кортежа.

Привилегия доступа - возможность управлять тем, как пользователь может обращаться к таблице БД или к базе данных.

Процедурная поддержка ограничений целостности - заключается в использовании триггеров и хранимых процедур.

Процедурный язык DML – язык, который позволяет сообщить системе о том, какие данные необходимы, и точно указать, как их можно извлечь.

Разделяемая блокировка (S-блокировки, S-locks - Shared locks) - блокировки с взаимным доступом (блокировка чтения).

Распределенные БД - БД, располагающаяся на нескольких компьютерах.

Распределенные СУБД – программный комплекс, предназначенный для управления распределенными базами данных и обеспечивающий прозрачный доступ пользователей к распределенной информации.

Реляционная модель данных – модель данных в виде нормализованных отношений (двумерных таблиц).

Репликация базы данных – это тиражирование и синхронизация объектов базы данных по разным узлам информационной системы, с целью повышения ее эффективности.

Сервер - в информационных сетях - компьютер или программная система, предоставляющие удаленный доступ к своим службам или ресурсам с целью обмена информацией.

Сервер баз данных - сервер, выполняющий обработку запросов, направляемых базе данных.

Сервер приложений - сервер, предназначенный для выполнения прикладных процессов. Сервер приложений: - взаимодействует с клиентами, получая задания; и взаимодействует с базами данных, выбирая данные, необходимые для обработки.

Сетевая модель данных – модель данных в виде произвольной сети.

Степень отношения – количество атрибутов отношения или количество элементов в каждом кортеже отношения (аналог количества столбцов в **таблице).**

Система управления базами данных (СУБД) – совокупность языковых, программных и технических средств для работы с базами данных.

Сортировка - данные требуется представить в соответствии с некоторым заданным критерием: в порядке возрастания, убывания либо в алфавитном порядке.

Сохранность баз данных – комплекс мер обеспечивающих, минимальное время простоя системы после сбоя, с выполнением требования сохранить всю информацию или, в крайнем случае, минимизировать потери информации.

Страница данных - это минимальная единица для дисковых операций СУБД. Страница данных это набор записей, хранимых в некоторой области жесткого диска на сервере (клиенте). Все страницы имеют один и тот-же размер, который определяется конфигурацией сервера и базы данных.

Структурирование в базах данных – введение соглашений о способах представления данных.

Структурная связь – наличие отношений между двумя множествами объектов (элементами системы).

Структурированность систем – наличие установленных связей и отношений между элементами системы.

Суперключ (superkey) – атрибут или множество атрибутов, которое единственным образом идентифицирует кортеж данного отношения.

Сущность – отдельный тип объекта организации (человек, место или вещь, понятие или событие), который нужно представить в базе данных.

Схема базы данных - структура, описанная на формальном языке, поддерживаемом системой управления базами данных, предназначена для контроля целостности данных.

Таблица – объект (компонент) базы данных, предназначенный для хранения однородной информации, представленный в табличной форме, разделенной на строки (записи) и столбцы (поля), в которых содержаться данные или значения Null.

Текстовый файл – файл в котором содержаться символы с кодами, попадающими в диапазон печатных символов. Строки текста в таком файле отделены символами новой строки (chr(13) и chr(10)).

Тезаурус (греч. thesauros – сокровище, запас) – нормативный словарь, в котором понятие определяется логически упорядоченным множеством синонимичных или близких по значению слов.

Тип данных – характеристика, определяющая способ интерпретации компьютером определенного элемента данных.

Транзакция (Transaction) - совокупность операций базы данных, выполнение которых не может быть прервано. Для того чтобы изменения, внесенные в БД в ходе выполнения любой из входящих в транзакцию операций, были зафиксированы в базе данных, все операции должны завершиться успешно.

Транзакции конкурирующие - если транзакции пересекаются по времени и обращаются к одним и тем же данным.

Третья нормальная форма – правило для реляционных баз данных, требующее, чтобы столбец, который не является ключевым, не зависел ни от одного другого столбца, который также не является ключевым.

Триггер (Trigger) - программа базы данных, вызываемая всякий раз при вставке, изменении или удалении строки таблицы. Триггеры обеспечивают проверку любых изменений на корректность, прежде чем эти изменения будут приняты.

Упорядоченный файл – файл записи которого отсортированы по некоторому признаку (полю, группе полей).

Управление параллельным доступом – процесс организации одновременного выполнения в базе данных различных операций доступа, гарантирующий предотвращение их влияния друг на друга.

Уровень изоляции - представляет собой степень независимости одной транзакции от всех остальных. Наивысшим уровнем является сериализуемость (SERIALIZABLE), которая обеспечивает полную независимость транзакций. Причем, каждый последующий уровень изоляции обеспечивает защиту, предлагаемую предыдущим уровнем изоляции, и добавляет новые требования к защите. Согласно стандарту SQL ANSI-92 выделяют четыре уровня изоляции транзакций. Для каждого уровня изоляции определены действия, выполнение которых запрещено.

Файловые системы – набор программ, которые выполняют для пользователей некоторые операции. Каждая программа определяет собственные данные и управляет ими. В случае баз данных - данные отделены от программ их использующих.

Файл данных – поименованная совокупность данных на физическом носителе в информационной системе.

Файл последовательного доступа – файл в котором одна запись следует за другой.

Файловый сервер - узел вычислительной сети, реализующий начальный уровень архитектуры клиент-сервер. Обычно файловый сервер работает под управлением развитой многозадачной сетевой операционной системы. Файловый сервер:

- обеспечивает управление доступом к файлам;

- предоставляет в общее пользование дисковое пространство, принтеры, модемы и другие ресурсы.

Физическая независимость данных - приложения не должны зависеть от используемых способов хранения данных на носителях, от аппаратного обеспечения компьютеров, на которых находится реляционная база данных.

Физическая организация базы данных - структуры хранения и методы доступа к хранимым данным.

Физическое проектирование БД – решение вопросов, связанных с производительностью СУБД, определением структуры хранимых данных, методов доступа к ним.

Фрейм – метод представления знаний, когда свойства связываются с вершинами, представляющими концепции или объекты. Свойства описываются в терминах атрибутов, называемых слотами.

Хеширование – метод обеспечения быстрого (прямого) доступа к информации, хранящейся во вторичной памяти.

Хешированные файлы – файлы прямого доступа. Структура файла предполагает поиск записи по ключу, значение которого является аргументом хеш функции. Значение хеш функции определяет адрес (местоположение) записи в структуре хранения.

Хранимая процедура (Stored procedure) - программа, которая выполняется внутри базы данных и может предпринимать сложные действия на основе информации, задаваемой пользователем. Поскольку хранимые процедуры выполняются непосредственно на сервере базы данных, обеспечивается более высокое быстродействие, нежели при выполнении тех же операций средствами клиента БД.

Целостность данных – правила, предотвращающие случайное или умышленное нарушение достоверности и полноты содержащейся в базе данных информации. Другими словами - это механизм поддержания соответствия базы данных предметной области.

Целостность сущностей – ни один атрибут первичного ключа отношения не может содержать отсутствующих значений, обозначаемых определителем NULL.

Целостность по ссылкам – если в отношении существует внешний ключ, то значение внешнего ключа должно либо соответствовать значению потенциального ключа некоторого кортежа в его базовом отношении, либо задаваться определителем NULL. Для каждого значения внешнего ключа, появляющегося в дочернем отношении, в родительском отношении должен найтись кортеж с таким же значением первичного ключа. Часто вместо выражения "целостность по ссылкам" употребляют его синонимы "ссылочная целостность", "целостность связей" или "требование внешнего ключа".

Четвертая нормальная форма – правило для реляционных баз данных, требующее, чтобы в отдельную таблицу включались только относящиеся к ней информационные объекты, при этом такие таблицы не должны содержать данные, связанные с несколькими информационными объектами при наличии между этими объектами отношения “многие к одному”.

Элемент данных – значение, которое описывает какое-нибудь одно свойство информационного объекта, например, имя человека, фамилию, пол. В данном случае информационным объектом является человек.

Экстент — это коллекция, состоящая из группы физически непрерывных страниц данных; они используются для эффективного управления страницами. Все страницы хранятся в экстентах (например в MS SQL экстент содержит 8 страниц по 8 кб в каждой странице).

Язык DDL – описательный язык, который позволяет администратору описывать и именовать сущности (связи) и атрибуты, необходимые для работы некоторого приложения, с указанием ограничением целостности и защиты.

Язык DML – язык, содержащий набор операторов для поддержки основных операций манипулирования данными содержащихся в базе данных.