

базы данных

4.09.03г.

Брешиков Александр Владимирович

1. Курсач

27 ср. ≈ 10-15 ср

2. Зад.

Тема:

2. Физ. зачет

2. Зад.

Физик, контра-

3. Экзамен

2. Зад.

Физик, контра-

лабн - окт - нояб. до 15 окт. курс

2. Зад.

Физик, контра-

Харитонов, Михеев "Access 2000"

Информация о

компьютерах.

Термин inf. с-м - качественное аспекты inf.

Данные - inf. фиксированная в  
отд. форме

Элемент данных - кач. семанти-  
ческая единица данных

Информация - содержание данных

Предметная область - часть реального  
мира, представляющая интерес с  
точкой зрения к.-л. задачи, проблемы.

Инф. системы - любые сов-ти данных  
и их обработки (inf. поисковая с-ма)  
соединение сов-ти поток и управл.  
изменениями.

Пример: БД, процессор.

Базы данных - сов-ти спец. образом  
отр. данных, предназначенных для  
использования в разн. целях.

СДБД - инструмент моделир БД.

11.09.03г.

есть 2 языка  
Во всех БД ЯОД, ЯМД, в соотв. с ними  
наз-ся языки запросов

ЯМД - языки работы с данными  
(алгоритмические)

Языки делются на процедурные и не-  
(обработка по SQL  
интерпретации)

Типы интр-ых систем

- 1) документальное - статьи, книги  
библиографические  
накопительные  
реферативные
- 2) фактографические - содержат факты  
упре в производстве  
испыт в будущем
- 3) лексикографические - словари в эл. виде
- 4) гибридные

Типы баз данных

- 1) локальные (персональные) - на одном ПК
- 2) общие (интегрированные) - с одной БД  
поиск раб. всех  
пользователей
- 3) распределенные  
(БД распределена по неск. машинам)

Основы построения БД  
Этапы проектирования БД

- 1) формулировка и анализ требований
- 2) концептуальное проектирование
- 3) проектирование реализации
- 4) физическое проектирование

1) этап заканчив-ся ТЗ.

На этом этапе  
Необх-мо опр-ть цели, провести  
экспертирование, опр-ть семантику  
(смысл), выделить интр-ые потоки.  
На этом этапе опр-ся сферы применения  
собирается интр-я об использовании  
данных, преобр-ся требования в форму  
уровня для анализа. Строится словарь  
задачи, правила, формируются  
требования

2) Концепт. (интерлогич.) пр-ц-е этапа -  
создание модели предметной области.  
Создается описание предм. обл., абстра-  
гированное от СУБД. Такое опис-е  
наз-ся интерлогической моделью. Для  
этого существуют спец. ЯОД. Трудн. треб-ти  
модели:

- адекватность
- непротиворечивость
- расширяемость
- восприимчивость

По сути на этом этапе проводится  
структуризация данных в упор. виде.

Модель - это описание объекта кот.  
в полной мере удовлетворяет за-  
дачам, которые реш-ся с этим объектом  
и абстрагированное от несуществующих  
с точки зрения задачи характеристик

Возм-ть удовлетворения треб-я задачи -  
проблема, кот. решает модель.

3) Датологическое проектирование.  
Создается на основе интерлог. мод.  
разр-ся модель данных и связей  
между эл-тами данных в терминах  
языка СУБД.

На этом этапе также пров-ся

- адекватность
- состав БД
- можно выделить эррективность БД.

4. Привязка <sup>детальной</sup> модели к среде хранения, проектируется организация данных, выбор носителя, формат.

18.09.03г.

Концептуальное или информационное проектирование.

Модель р-е предметной области (часть реального мира)

На этом этапе выделяется сущность или объект реального мира.  
Сущность - объект, с кот. оперирует будущая БД (человек, предмет, событие)  
Тип сущ-ти - понятие, к кот. относится набор однородных объектов.

Экземпляр сущ-ти - конкретный объект в наборе.

Св-ва объекта - атрибуты.

Некот. эл-ты р-е могут различать такими св-ва, что значение можно идентифицировать значения, кот-ли обладают группой эл-ментов данных объекта (ключи)

применяются - опр-ют к сущ-ти (напр.)  
собираются в сущности (напр.)

У всех атрибутов должно быть характеристики (наим-е, описание, роль, тип, допуст. значения и т.р.)

Связь сущностей показывает как сущности взаимодействуют (объект, таблица)

# сущностей, зарплата (связь - "наименование")

Связи бывают обязатель, необяз, возможны

Важное понятие: степень связности показывает сколько эл-тов одной сущ-ти связано со ск-ми эл-тами другой (1:1, студент-курс, зачет. книжка, группа студента, студент-группа, преподаватель)

Информационный подход к проектированию БД.

БД - целевая модель предметной области. (по)

Необходимо опр-ть границы по которым анализ фактов, факты интерпретируются. Этих фактов по-сущ-ные объекты, св-ва, отношения.

Проектир-е БД начинается с структуризации объектов по объектам классифицируются

Выражаются в виде БД.

Для объектов фиксируются их св-ва, отн-ся в отношении к другим объектам (отношения - не всегда).  
Суть инд-ции. проект-я - это уст-е, состоящее из частей, по его восприятию и представлению в БД, объекту, св-во. связь.

Объект - то, о чем каковы inf (бывают атомарные и составные).  
Вост-е зависит от времени.

Св-ва бывают независимые от объектов (атомарные / локальные) и зависящие (реляционные).

Связи хар-ся степенью связанности.

Мн-ва объектов с <sup>одинаковыми</sup> св-вами наз-ся объектной группой.

Принцип модели в проектир. ав-ся, модели сущность - связь, раз-ка.

Это формальная модель предметной об-ти.

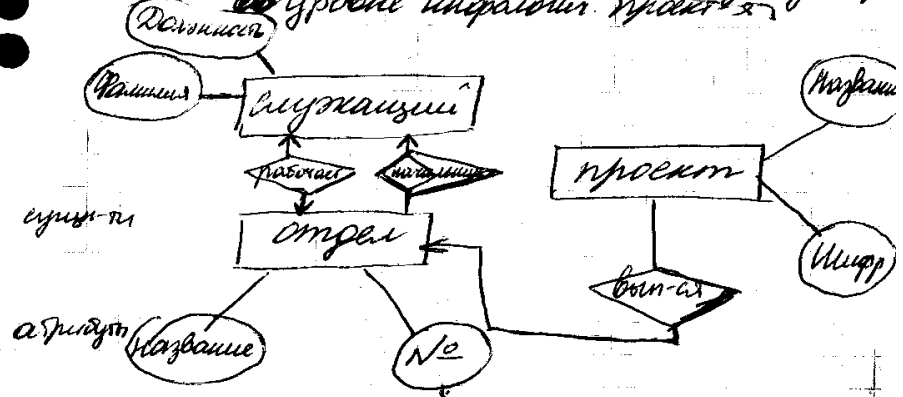
3. конструктивный эл-т:

сущность  
атрибут  
связь

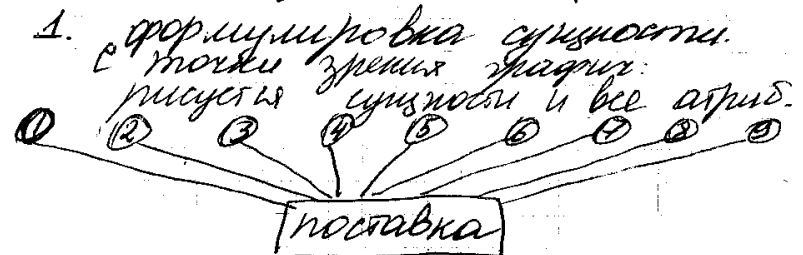
Сущность - обобщенное понятие реального мира (материального или немат.)

(6)

Для сущности определяется набор атрибутов. Пример - конкретный объект в наборе. Тип данных быть определен, значение можно иметь идентифицировать с ним. Атрибут может быть исп-н для прет-я связи м/у сущностями. Часто бывает, чтобы не сам объект, а его св-ва, например, цвет. Мн-ва <sup>представляется</sup> в виде рисунка, уровня инд-ции, проекта.



Инд-ция, проект, по сути, модель локальных представлений (то основная задача).  
Состоит из 6 этапов (шагов):

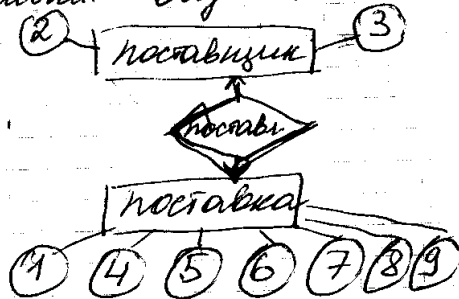


(7)

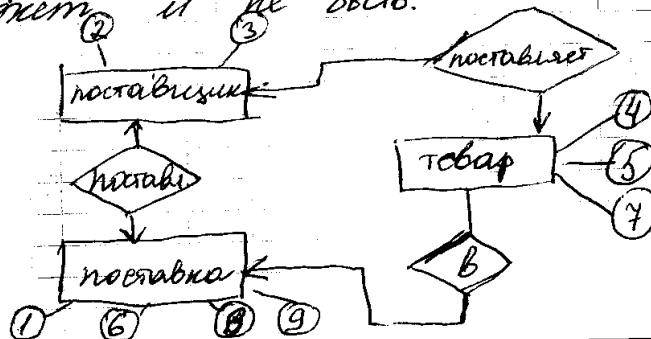
- 1 индекс поставки
- 2 индекс поставщика
- 3 адрес поставщика
- 4 индекс товара
- 5 название товара
- 6 кол-во товаров в пост-ке
- 7 цена
- 8 номер склада
- 9 дата поставки

надо проаб. несколько вариантов  
модели и выбрать наиболее  
покой вариант предст-я inf.

В этой модели нет inf. о поставке,  
кот. не поставл. товар  
или simply. выбрать пост. в отг. изд.



Товар может быть в пост-ке, а  
может и не быть.



2. Необх-но выбрать атрибут,  
идентифицирующий сущность (ключ).

Требования к ключу:

- с одной стороны
- минимальность, оптимальность
- полностью должен хвать эссенцию сущности

3. Назначение единицы измерения  
для атрибутов (объектов)

Это значит тип, зн-е по умолчанию,  
длина, маска и т.д.  
ограничения, мин-во, макс-во и т.д.

На самом деле проработку: - SQL 25.09.03.

- реляционная алгебра
- распределенные БД
- Oracle, DB2, MS-SQL, MySQL
- клиент-серверные технологии.

4. назначаются внешние ключи для  
выбора объектов с одинаков.  
атрибутами

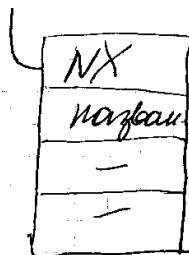
Как правило вторичн. ключ. не з-ся  
в най. таб. же первичным  
и тогда возн-на связь 1 ко многим

возн-на связь 1:1 - также.

№
Ф.
И
О
№

первич. ключ

номер хвосты (вторичн.) 9



первичный

с типа связи

эквивалентное (связыв-ся 2 таблицы при условии, что ключевые поля в них совпадают)  
 левое объединение (выбираются все записи левой табл. и только те записи правой, ключевые поля кот-х совп.)  
 правое объединение

5. Модель спецификации связей, позволяющая изобразить связи

Объект-ситуаций  
 Атрибуты: ① ФИО

- ② дата рождения
- ③ назв-е ВУЗа
- ④ адрес ВУЗа
- ⑤ специальность
- ⑥ имя ребенка
- ⑦ дата рождения ребенка

таблицы номер первичный ключ.  
 1) дополнительная связь: ребенок, дата рождения  
 2) же непосредств. связь с ключ. ВУЗ-адрес ВУЗа

Ситуаций

① ② ⑤

N ВУЗа

③

④

N Ребенка

⑥

⑦

Рассм-ть ск-ко детей

⑩

6. Объединение моделей локальных представлений.

- объединяются фрагментарные
- устраняются различия представлений
- вводятся новые объектные понятия
- образуются производные типы объектов
- формируются связи м/у объектами (надо продумать степень связанности, тип связей)

Комплексный на этом этапе:

- 1) идентификация - если одинаков символ знач-е
- 2) агрегация - группировка атрибутов и именования группы одним именем
- 3) обобщение (позволяет трактовать класс подобных объектов как один поименованный общий объект)

Работологическое проектирование.

В кач-ве модели данных вы- 2.10.03.  
 тупают сами данные, в том числе их структурная композиция, правила построения.

В каждой с-ме свои форматы данных, свой состав операций, с их помощью строятся более сложные модели данных для реш-я специфич. задач.

Существует 2 типа языков: декларативные и процедурные.  
 Декларативные - с точки зрения БД, объявление процедур и правил.

⑪

1. и процедурн. языки - в частности SQL.

Совок-ть операций языка определяет модель данных, являясь формализацией для описания стр-ры данных и процессов изменений их составных с целью моделирования реальных процессов.

Формализация - формально описываемая и машинная.

Эти стр-ры называются логическими (альтернативно - физическими).

Можно выделить четко выделяется ЯОД, ЯМД.

При датологич. проектир-и строится схема данных. Опис. статистически свойства данных опис-е на формальном языке.

Структуризация данных базируется на агрегации и обобщении. Здесь видятся логические стр-ры:

поле.  
записи  
файлы  
набор файлов

Агрег. - кодирование полей в записи  
обобщение - предмет-е записей файлов.

Очень важно задать атрибуты полей!

Поля - элемент данных, наименьшая именованная единица данных

Записи - повторяющаяся совок-ть эл-тов данк.

(12) Файл - именованная совок-ть записей одного типа

БД - типичн. совок-ть файлов.

2.10.03.

Основные операции над данными.

Отражают динамич. св-ва модели. Эти св-ва мн-во допуст. операций, кот. переводят БД из одного сост. в др.

центральной операцией БД явл-ся селекция (выборка), более сложные запросы, выборка условий. Запросы дел-ся на: идентифик., включение, удаление, модификацию, выборка и т.д.

Навигационные и спецификационные операции.

система запросов. Необходимо иметь процедуры для более сложных БД (проц. обработки данных), работ транзакц. - ер-ва для с таблицами в многопользоват. режиме, програм., связ. с таблицами.

Пример на удаление  
добавление  
изменения

Этап проектир-я БД. На этом этапе много проблем.

Проблема ограничения целостности (непротиворечивости данных).

На этом этапе осуществляется выбор модели БД (базовая модель построения БД), иерархическая, сетевая, реляционная и объектно-реляционная модели БД.

В конечном итоге в прог. построения мод. надо учесть след. св-ва:  
модель данных

(13)

1) полнота и трудоемкость описания данных и написания программы.

2) простота и компактность минимальное число типов базовых структур

3) сложность модели для изучения пользователями.

4) наглядность предмет-а структуры данных

СУБД выбирает исходя из ваших треб-ий с одной стор, а с другой исходя из модели и требований.

Иерархическая модель данных

это такая модель, когда данные в бд представлены в виде дерева.

сетевая БД - в виде графа. (связи между сущностями).

Реляционная модель. Реляцион. бд - все объекты представл. в виде таблиц и связей м/у табл.)

Реляц. мод. - опирается на строгий подход к описанию и управлению данными.

Этот подход основан на наборе математич. правил, обеспечивающих точность и целостность данных.

Внешн. предст-е данных и легкость изуч-я зная запросов к базам данных. Образуется, когда данные представл. в виде отношений (реляционное представление)

Реляция с атриб. - отношение

⑭ Отношение - это некотор. связь м/у

элементами множества. ( $>$ ,  $<$ ,  $=$ )

Синонимы: таблица, объект, сущность, файл

Отношения строятся на данных (мн-во отношений одного типа)  
Аналоги: данные-атрибут, столбец.

Каждому атриб. ставится в соотв-ие его значение - данные атрибута.

Реляционные отношения имеют смысл м/у значениями из одного дм-ка

Отнош-е состоит из заголовка и тела.

Заголовок состоит из фиксированного мн-ва атрибутов. Каждому атрибуту соотв-ет один файл.

Тело отношения состоит из конечного мн-ва ~~элементов~~ картетей (мн-во пар атрибут, значение атрибута).

Картет - это набор, запись.  
<sup>Тель</sup>

Строение отнош-я - число атрибутов отношений.

Множественность отнош-ий - кол-во картетей в отношении.

Для отнош-ий стр-ся ключ отношения.

Ключ отнош-ий - мн-во атрибутов, удовл-ющих след. требованиям:

1. уникальность (хотя бы 1 атрибут в паре картетей должен различаться)

2. минимальность, но полностью ракурсивная неопределенность относятся к другим требованиям

⇒ всякое отнош-е имеет хотя бы один ключ. ⑮



Если есть атрибуты отношения  $R_2$ , которые совпадают с первичными ключами атрибута  $R_1$ , то эти атрибуты входят в первичный ключ  $R_2$ .

Реляц. исчисление над столбцами и строками.

Реляц. предст-е данных ~ их предст-е в таблице

столбцы - аргументы отнош.  
строки - связи

но реляц. табл. это не просто таблица.

Представ. собой двумерн. массив, обладающ. некотор. св-вами.

- каждый э-т таблицы - один элемент данных
- все столбцы однородные т.е. все э-ты в столбце имеют одинаков. тип
- каждый столбец имеет уникальное имя
- одинаковые строки в таблице отсутствуют
- порядок следов-я строк и столбцов может быть произвольным

и т.д.

Ревенков, Григорьев

21.03.21

Реляционная БД - набор взаимосвязанных отношений  
кажд. табл. представл-ся в виде графа. записей.  
Отношение - бинарн. математич. объект ~~таблицы~~

В Таблицах бывают пустые клетки, для математич. объектов это недопустимо

Чтобы не было пуст. клеток вводятся зна-е NULL, как помин-е этого мн-ва, завис. от семантики табл. (# int отсутств., еб-во не применимо и т.д.)

Реляционные операторы составляют реляционную алгебру.

Сост-е отношений со временем меняется. Изм-е отнош-я описывается набором операторов, наз-ся операторами реляц. алгебры.

Операторами могут операт. явл-ся одно или 2 атрибута (унарные или бинарные операторы соотв.)

Рез-там применения оператора явл-ся единственное реляционное отнош-е

2 группы операторов:

- 1) описывает обычные текн. операции и их комб-ции
- 2) специальные операторы

В 1-м. имеют место только эти  
первые операторы (активные):  
- объединение двух отношений есть  
лишь-во всех карточек,  $\in$  каждому  
из исходных отношений. Формально  
создается третья таблица, кот.  
включает все записи 1-й табл. и недостающие  
записи 2-й таблицы. Пример: студент 1  
и 17, П17, Ф19, У19 - таблица 1  
Фам. Возраст.

студент 2 и 17, У17, Р21 - табл. 2

таблица 3: и 17, П17, У17, Ф19, Р21  
объединение!

Операнды должны быть совместимыми;  
т.е. они должны базироваться на одинак.  
и там где были одни и те же данные.

пересечение  
- ~~пересечение~~ двух совместимых по  
объединению отношений - есть лишь-во  
карточек,  $\in$  обоим отношениям.  
Воз-ся 3-я таблица, включающая те  
записи первой табл., кот-е есть  
во второй.

#1 таблица	2 табл.	3 табл.
и	и	и
П	У	У
С	С	С
Ф		

совмещ. табл. это 3-я табл., кот. включает  
- разность - ~~разность~~ ~~таблица~~ ~~таблица~~  
только те записи 1-й табл., кот. не в 2-й.

#	и	и	П
	П	У	С
	С	Ф	
18	У		

- произведение 2х отношений -  
есть лишь-во всех карточек, по-  
лученных объединением каждого  
карт. 1-го отн-ия со всеми картами  
2-го отн-ия. Степень резуль-  
тирующего отношения = сумме  
степеней отношений ~~операндов~~  
Мощность = произведению их ~~мощ-~~  
ностей (мощ-ть - это кол-во записей)

#1 табл.	2 табл.	3 табл.
Иванов	Физика	и-физ.
Петров	математ.	и-м
1 студент)	(экзам-н)	П-Ф
		П-м
		степень 2
		мощн. 4.

2) Специальные реляционные операц. 9.10.03

1. селекция - унарный оператор. Рез-тат  
оператора явл-ся вычеркив-е из табл.  
тех строк, для кот-х не был-ся ~~выпол-~~  
выпол-е. Селекция - лишь-во карточек ~~табл.~~  
для каждого из кот. был-ся поставл-  
усл-е.

#	Фам.	Возраст	Рез-т
	и	17	и 17
	П	16	
	У	18	У 18
	Ф	12	

Усл-е: возраст > 16

2. Проекция - это срез по столбцам.  
Унарный оператор. Результатирующее  
отношение состоит из карточек,  
каждая из кот-х явл. подмножеством  
исходного набора карточек с за- 19

данным набором атрибутов, по которому осуществляется проекция. с ор. икар. Рез-т оператора - вычеркивание из исходн. набора тех атрибутов, кот. не входят в указанный набор

Если после вычеркивания оказались совпадающие строки, то дубликаты вычеркиваются.

Имя	Группа	Вид спорта	Рез-т
И	1	Плав.	17
П	2	Шахи	2 И
С	1	Бокс	15
Ф	2	Шахи	
Г	1	Бокс	

Проекция по атрибутам группы и вид спорта.

- 3) Соединение - бинари. В кач-ве отнош. выдел-ся атрибут, по кот-му произв. соединение. Выделенные атриб. базируются на имени и там же данены, для кот. имеет смысл условие соединения. Соед-е - это мн-во всех картетей, кот-е из кот-х получены объединением 1 картеты из 1-го отношения и одного картеты из второго, таких, что вып-ся условие соединения. Если условие соед-я - равенство, то соед-е наз-ся эквисоединением. Рез-татом эквисоединения явл-ся отн-е, кот-е состоит из картетей, кот-ые содержат 2 одинаковых значения атрибутов из разных исходных отношений.

# 1 таблица - исходная

Имя	Группа	Возраст
П	1	17
И	4	19
С	9	18
Ф	2	17
Г	3	12

2 табл. - срезы

Вид спорта	Группа
Плавание	4
Шахматы	3
Футбол	1

В кач-ве ун-я соед-я берем рав-во номеров групп

Результативная табл.

Имя	Гр.	Возраст	Вид сп.
И	4	19	Пл.
Г	3	12	Шахи
П	1	17	Футбол

- 4) Деление - атрибуты отн-я дел-ся на 2 группы. Данные должны содержаться в отн-и, кот-е делится на 2 и базироваться на делителе. Результативное отн-е состоит только из атрибутов делителя, в кот-х нет делителя.

#	Фам	Дисциплина	Оценка
	И	вт	5
	С	вт	4
	Ф	вт	5
	Ф	физика	4
	Б	физика	5
Делители			
	вт	5	
	физ	4	
Рез-т	С, Б		

### Нормализация отношений

Одни и те же данные могут группироваться в разн. отнош., но сред. котор. отнош. обладает лучшими св-вами при манипулировании с БД, если они отвечают требованиям нормализации отношений.

Нормализация отношений - формальный аппарат ограничений на структурирование отнош.-й.

Он позволяет устранить дублирование, обеспечить непротиворечивость хранения данных, ↓ затраты на ведение БД. Выделяют 3 гр 7 нормальных форм.

- 1) Отнош.-е приведено к 1<sup>ой</sup> норм. форме, если из одного из его атрибутов не вытекает отношение т.е. все его атрибуты простые, неделимые, атомарные.

Спортсмены - Напр, отношение (фамилия, группа, спорт). атрибут Спорт = (вид, разряд). это неформально в БД. Надо привести к 1<sup>ой</sup> норм. форме.

Спортсмены = (Ф, гр, вид спорта, разр.)

- 2) 2<sup>ая</sup> норм. форма: отнош.-е приведено к 1<sup>ой</sup> н.ф., если каждый ключевой атрибут функционально полно зависит от ключевого атрибута.

Функциональная зависимость - это завис-ть, при которой определенному значению ключ. атриб. соответствует только одно зн-е ключевого атриб. В случ. составного ключа говорят о функционально полной зависимости.

Студент = (Инициалы, Фам, И, В, группа)

отнош.-е нах-ся и в 1<sup>ой</sup> и во 2<sup>ой</sup> н.ф.  
все атриб. неделимые      все атриб. составные

Ведомость = (ФИО, дисциплина, лектор, ауд.)

Пусть у этого отнош.-я ФИО и дисциплина - ключ, тогда оно не нах-ся во 2<sup>ой</sup> н.ф., т.к. лектор не нах-ся в полной функционал. зависимости от этого ключа.

Поэтому лектор опред-ся атрибутами дисциплина. ⇒ отнош.-е надо разбить на 2 независимых.

1. ~~Ведомость~~ = (ФИО, дисц., оценка)
2. Лектор = (дисциплина, лектор)

30.10.03г.

3) 3-я норм. форма - отношения приведено к 3-ей норм. форме, если устранены транзитивные зависимости между атрибутами отношений. Т.е. каждый атрибут отношения, не являющийся первичным, не транзитивно зависит от каждого ключевого ключа транзитивной зависимости, когда 1 из ключевых атрибутов зависит от другого.

# Отношение Предметы = (Дисциплины, кафедра, теледран, лектор) 3-ей норм. форме:  
(Каф, теледран) = Отношение 2

(Дисциплины, кафедра, лектор) = Отношение 1

связать 2 таблицы с помощью атрибута кафедры

4) 4-я норм. форма, если у объекта имеются множественные св-ва, то некоторые атрибуты могут оказаться связанными многозначной зависимостью.

# 2 отношения:

авторы  
публикации.

1) каждый автор может быть несколько публикаций и наоборот.  
Отношение: ин:ин. Такая связь в БД не реализуется.

Устранение многозначных зависимостей называется приведение отношения к 4-й норм. форме

Каждому из множественных св-в ставится в соответствие отдельное отношение. Оно связывает исходные 2

②1) отношение с детальной хар-кой объектов.

#

Авторы = (Фамилия, Год Род и т.д.)

Публикации = (Название, Год, Типаж, Изд-во, язык, полетит и т.д.)

Эти табл. надо связать через 3-ю табл. в самом простом случае в ней должно быть 2 столбца с первичными ключами первых двух таблиц

Авторы

Публик.

Код А
ФИО
Г.Р.
...

Код. П
Назван.
Изд-во
...

A	Ключев. поля (КА)	КА КП	КП П
1	1	1	1
2	1	2	2
3	2	2	3
4	2	3	4

Реляционная модель - это концепция построения логической структуры данных. (до этапа реализации БД мы можем предусмотреть много вещей)

Нормализация

Включает в себя постулаты:

- логическая модель данных представляется набором двурядных таблиц;
- отношения связаны через согласующие атрибуты, при этом выполняются условия уникальности и минимальности первич- ②5

кого языка;

- на отклики-а определена разница с-ми реляционных операторов а также реляционно-посредный язык запросов;
- базовые и производные отклики-а удовлетворяют правилам (ограничениям) целостности.

Не все табличные данные реляционные.

Решая технологию упрощает проектирование БД за счет:

1) упрощается процесс отображения реального мира за счет высокой степени разделения логического и физического уровней информации; за счет приближения логического уровня к объектам реального мира.

2) концептуальная простота реляционной модели позволяет реализовать небольшие (неизбыточные) и простые БД.

3) теория нормализации позволяет избежать проблем, если отношение неструктурировано естественно. Иными словами.

5 проблемы:

1) пустые значения вызывают трудности при манипулировании данными;

2) множественные значения с одной стороны нежелат. иметь много карточек для одного объекта, а с другой операции сведения к единому сведению об одном объекте затруднены.

3) реляционная модель не является структурой данных в традиционном для программирования смысле.

## SQL

Разработан ANSI - Институт стандартов. Протокол разработан IBM в 1972. Потом Oracle выпустила СБД, Oracle и только в 1986 ANSI приняла этот язык как стандарт. В 1987, междунар. организация станд. приняла его. Недавние стандарты SQL 92.

Преимущества стандартизации: не надо изучать разные языки, хвалит однообразие.

Недост.: сужаются рамки. Фирмы стараются уйти от станд.

Наиб. извест. фирм-производ:

Oracle, Micros, IBM и т.д. Во всех БД присутствует SQL.

Конкретные реализации SQL отличаются друг от друга, используя расширение стандарта (разные фирменные нотации). Для фирм выгодно использовать фирменн., близк. к станд.

Типы команд SQL.

Выделяют 4 основных языка в рамках SQL: DDL - дизайн

- DML - яз. манипулир-я данных
- DQL - яз. запросов
- DCL - яз. упр-я данными

Отдельно стоят:

Команды упр-я транзакциями, команды администрир-я данных

Команды языка определения данных <sup>исп-ся для определения структуры БД</sup> позволяют создавать и изменять структуру ее объектов.

# создав./удалить таблицы

Основн. команды DDL:

create table - созд. табл.

alter table - измен-ть стр-у табл.

drop - удалить

Объектами явл-ся также индексы, <sup>отд. файлы (для скорости)</sup> для них  $\exists$  отдельн. команды

create  
alter } index  
drop

Команды DML:

исп-ся для манипулир-я <sup>внутри</sup> объектами реляционной БД.

таблицы представления (синонимы запроса)

3 ср-ва манипулир-я:

insert - добавить

update - обнов.

delete - удалить

DQL - служит для выборки данных. Он наиболее известен пользователям.

Включает единств. команду:

select - выбрать.

исп-ся для формиру-я запросов (на выборку) <sup>каждо своим специалн. предложением в БД</sup>

Триггеры - проги, связан. непосред-но с табл.

$\exists$  команд. табл. по измен.  $\exists$  тригг. - добавл., обновл-е, удален.

Это реакции таблиц на действия по поводу этих вещей.

DCL. Его команды позвол. управл. доступом к инф, находя. внутри БД.

Эти ком. исп-ся для созд. объектов, связ-х с доступом к данным.

Кроме того, они служат для контроля

кажд. распредел-ся привилегий м/у пользователями.

# Alter password - созд. пароль

Как администрируя данных позвол.  
наблюдателю и админу осущ. контроль  
за выполнением действий, ад-  
министрировать операции БД, анализ-  
производительность системы.

# START AUDIT, STOP AUDIT.

Команды упр-я транзакциями:

Транзак-совокуп-ть действий, направл. на  
применение данных в БД.  
(заказы, удалять, ...)

Транз. оправдан, когда предусмотрен  
минимальный режим работы (клиент-  
сервер)

Важная задача при работе с БД, когда много  
пользоват. обращ. к данным и тем же  
мех-изм. транз. позвол. сделать, чтобы все  
было хорошо.

commit - заверш. для сохр-я рез-тов  
транзакции

rollback - для отмены рез-тов транз.

savepoint - созд. внутри группы  
транзакции точки вкл-е вназад  
или можно быть соверш. откат

SET TRANSACTION - задать имя  
транзакции

(чтобы к ним можно было  
обратиться из прилож-я)

Структура данных

Осн-е стр-ры данных

Осн-е типы данных:

- символьные строки

- числовые знач-я

- дата/время

- Символьные бывают фиксир-ой и  
переменной длины  
character (fix) - фиксир.  
character varying (n) - перемен.

fix (fix) - фиксир.  
fix (fix) - фиксир.

- Числовые знач-я - фиксир. верен. др.

bit (n - fix)

bit varying

decimal (p, s)

integer

smallint

float (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

real (fix)

- Data / br

date

time

interval

interval

interval

interval

interval

interval

interval

interval

interval

interval

interval

interval

interval

interval

interval

interval

interval

interval

interval

interval

interval

Не алгебраический язык

Литерное строки - алгебраико-  
цифровая инт. записываемая  
цифровые типы данных есть: NULL

(или одиноч.  
кавычки)

Объекты управления БД



Объект: таблица  
представление (запросы)  
классификатор (ср-во разделения табл.)  
индексы (ключи)  
ежедневные  
SELECT <имя> S <состав>

Создание таблицы в SQL

CREATE TABLE <ИТ.>

(К. п.с. 1, тип [NOT NULL]);  
( ) STORAGE

таблица с вог-ся 36 (INITIAL 36)  
если не хватало мере кетх. 20 (NEXT 20);

Alter TABLE - модифицир. стр-ру  
сущ-ств. табл

Add - добав.

Drop - удаля.

можно созд. табл. на основе сущ-ств.  
ИТ. новое

CREATE TABLE AS

SELECT [\*] ст. 1, ..., ст. N

FROM ИТ. старое

[WHERE];

Удаление таблицы

DROP TABLE ИТ.

[restrict]

[cascade]

Назн-е ключей - стр. команда.

В опис-ии к.п.с. - ключевые ключи

primary key  
foreign key  
unique

манипулир-е данными - язык SQL  
язык SQL, предназначен для  
пользователя (язык) в inf, содержащ.  
бд.

INSERT  
DELETE  
UPDATE

insert ИТ. VALUES

("зн. 1", ..., "зн. N")

изм-е сущ-ств. данных

update ИТ. set ст. 1, ..., ст. N

where

удаля.

DELETE FROM ИТ.

where

20.11.032.

Важный аспект БД. Управление транзакциями - ст.

Обесп-во бездефектного упр-е?

Транзакция это логич. операция над данными. Они могут быть в ручную или автоматически (приложением). Транзакция состоит с помощью языка SQL, в частности insert, update, delete.

Транзакция может состоять из 1 или более операторов.

Особенности транзакции

- все они имеют начало и конец
- транзакция может быть сохранена или отменена
- если произошли свои транз. в процессе её выполнения, то в БД не может быть сохранена никакая-либо её часть.

Под управлением транз. процессора спос-ть  
управлять различными операциями над  
данными.

После боя транзакции немедленно из  
нельзя целевой таблицы не происходит.  
Для оконч. завершения транзакции  
существует команда упр-я транзакц.  
С их помощью можно либо отменить  
в базе все измен-я, либо полностью их  
отменить.

commit  
rollback

Решение запросов SQL

Команда всего одна `SELECT`

организовывает  
просмотр информации.

Основные эт-пы календы:

FROM  
WHERE  
ORDER BY  
GROUP BY

## Синтаксис SELECT

select <sup>operator</sup> ~~[\*]~~ <sup>key/obscure-as operator</sup> all <sup>operator</sup> / distinct / <sup>CT1</sup> CT1 <sup>CTN</sup> CTN ]  
from T1, ..., TN;

select \* from

T1  
where  $p1 > 5$ ;  
значения  $> 5$

Фильм по сценарию

Сортировка выводимых данных  
 \* после where - order by <name>

.. where  $P1 > 5$   
 orderby P1 ASC/DESC;  
 -- <sup>Boyle</sup> <sup>Yable</sup>

Можно посчитать кол-во записей табл.

select <sup>count</sup> from Table 1

Псевдоним. служит для переим-а статьи  
(или даже главы).

Операторы языка SQL:

Опер-ры - жаргизированное слова,  
предназначен. для выт-е операций /сравн-и/ (35)  
Формирует операции.

Группа операторов - сравнение:  
Арифметич., логич. связи, отриц-е, арифм.  
операции.  
сравнения:

=, <, >

логич. св.

IS NULL, BETWEEN, IN, LIKE, ANY, ALL

Эти опер-ры могут быть задействованы  
в конструкции where.

EXIST - для опред-я, присутств. ли строка  
в таблице.  
в where может быть задейств. SELECT -  
сложный запрос

WHERE

exists

(select cos

from p2

where cos < 10)

Отрицание - ! = <sup>станд.</sup>

Арифметич.: + - \* /

Агрегативные функции: count, sum,  
min, max, avg

Группировка - процесс объединения  
в один порядок столбцов с повторяющ.  
значениями

получая послед-ть:

where

group by

order by

Если группир. по столбцу, то они должны  
быть указаны в опер-ре select

(56)

Группирующие функции то же самое, что  
и агрегативные.

Сортировка: нал. через запятую

27.11.03

Если группируем, то соотв. столбец  
должен быть указан в опер-ре SELECT  
Group by (sum) - группировка по  
сумме

HAVING использ-ся совместно с group by  
GROUP BY HAVING

HAVING сообщает какие группы будут  
включены в результирующий набор

Пр. SELECT city, avg/pay, avg/salary)  
from emp\_pay where <> 'London'

GROUP BY CITY HAVING avg(salary) > 200  
ORDER BY 3;

третий столбец результирующей  
запроса

Соединения.

Естественные: исключаются дублирующиеся  
столбцы.

SELECT T1.\*, T2.C1 FROM T1, T2

WHERE T1.C1 = T2.C2 - все е. рав-и  
T1.\* - все нал. из 1ой таблицы

(37)

соед-е пер-ва:

<> !=

Словные соединения:

LEFT, RIGHT, FULL, AUTO, JOIN

все записи  
левой и  
только пра-  
вой те,  
которые  
совпадают  
по ключевым  
назв

акало

только  
те записи,  
которые  
совпадают  
по ключевым  
назв

все записи  
обеих таб-  
лиц, где  
совпадают  
и не совпа-  
дают по ключевым  
назв

A		A
B		C
D		F

Внешнее соединение

A A (AUTO)

B — пусто

D —

— C

— F

38

FULL: A/A (внутр. соединение)

Подзапросы:

обычно в подзапр помещаются

WHERE, ( )

SELECT подзапрос, как правило, содержит 1 столбец

ORDER BY может быть только в главном запросе.

Подзапросы возвращающие более одной строки данных, могут использоваться только в многозначных операторах

(N)

Главный запрос не может использовать BETWEEN

SELECT NC1[UC2]

FROM T1[T2]

WHERE UC1

(SELECT UC1[, UC2]

FROM T1[T2] [WHERE]);

Подзапрос может быть использован в INSERT, UPDATE, DELETE

??

криво

39

UNION исп-ся для объединения результатов нескольких SELECT'ов

UNION ALL - вывод дублирующихся строк

4.18.032 Введение в VBA.  
Visual Basic Access  
(ссылка Visual Basic Application)

VBA - абв. алгоритмическим расширением Access

Дост-ва из самых первых языков - (Access, Word, Excel - везде исп-ся)

Из визуальных средств прог-я. Формы и отчеты еще дано. Все-то не реализ-ся например. Макросы - не реализованы

Прога на VBA не треб. запуска. Ее вып-е инициализируется наступлением события.

Словно, VBA - расширение средств макросов

Можно макрос преобразовать в прог-у

Дост-ва VBA перед макросами:

1) в отличие от макр. можно пер-ть переименовать и передать парам-ры

2) VBA не ограничен встроенными конструкциями

3) возможна встроичная обр-ка таблиц

4) возм-на обр-ка транзакций (not в SQL)

б) возможен перехват ошибок и включение ср-в отладки

в) возм. создав. БД во время работы прог-ы

VBA исп-ся для вып-я сложных вычислений, ввода данных во время вып-я прог-ы, обработки транзакций, реализации циклических прог-ов обр-ки данных, вып-я операций Visual Access (напр. проверка наличия памяти)

Основы разработки программ

Основы VBA-модуль, включение процедуры

2 типа модулей:

1) стандартные

2) формы и отчеты

В 1) хранятся прог-ы и ср-ца, дост-ры для модиф. других объектов.

2) - абв-ся неотъемлемой частью существующих форм и отчетов. Созда-ся автоматически по мере созд-я форм и отчетов.

Эти модули, нап-мо общей обв., включают процедуры обраб. событ.

Эти прог-ы исп-ся для каталога существующих форм и отчетов

Область: примор-я  
Земельная  
11.7.9  
У края обл. есть водохранилище. (используются  
для выпаса скота, то вывозят 1-2)  
и-то делают. (проблема, как  
можно сделать из красивое примор-я

Пример: общей области  $\lambda$  и  $\mu$  нет.  
 Если  $\lambda$  и  $\mu$  так и  $\lambda$  и  $\mu$  так  
 Тогда  $\lambda$  и  $\mu$  — произв. обр-ки событий  
 в  $\mathcal{F}$

Временное

с псих. конструкц. DSM - созд-но явное  
или неявное - по мере исп-я

DIM  $\leq$  числ. перем.  $\geq$  AS  $\leq$  числ. перем.  $\geq$

Boolean [private/public] (neg neg DIM)

перем. доступна  
в других пред-вах  
данного мес.

Типов данных.

explain

cyte.

indeed

long

CUZKREKLS

double single

double.

data-

Stina

Object, variant

Если тип не указан - то считается  
variant - любой.

Интерпретируется в зависимости от контекста  
При умелом использовании это мощное средство!!!

В базисе гомоген. урав. объектов  
различ-х

Тип объекта для явл-ся сложным  
объединяет несколько типов.

Напр, объекти. тлины Access:

Application, Form, Report 47g.

Объемы типов данных:

Tablets, Filed

DIM BA AS Database

DIM TABA AS TABLEDEF NOW.

AS FIELD

Объект. Типа данных предметной  
IM только эти-есть <sup>для</sup> <sup>книжки</sup> <sup>тема</sup> <sup>прем-е</sup>, <sup>ка</sup> <sup>каф.</sup>  
ссылка на объект

Дим Талочко емигрантски е, кај кој  
својот е објект

Чтобы перемен. сослалась на реальный объект, нужно присвоить объект этой переменной

SET  $\delta_A = \text{current DB}$  ✓

SET TAB = 5 Δ, Create TABLEDEF

(73)

SET ПОЛЕ ТАБ. Create Field  
(начинание, object)

Константа или-ли CONSTANT <sup>или-ли</sup> <sub>или-ли</sub>

Тип конст. опред-ся <sup>или-ли</sup> <sub>или-ли</sub> буквами  
AS-буква-ка Address <sup>или-ли</sup> <sub>или-ли</sub>  
as-form  
as-table

Системные конст. TRUE, FALSE

IF [FORMS]! [FORMA]! [ПОЛЕ] = ?  
THEN

Процедура

3 типа: FUNCTION  
SUB

1) SUB <sup>или-ли</sup> <sub>или-ли</sub> <sup>или-ли</sup> <sub>или-ли</sub> <sup>или-ли</sup> <sub>или-ли</sub>

[PRIVATE / Public / Static] SUB или-ли

[(или-ли парам-б)] <sup>или-ли</sup> <sub>или-ли</sub> <sup>или-ли</sup> <sub>или-ли</sub>

[или-ли]

[EXIT SUB]

[или-ли]

End sub

Функция

2) [P/P/S] FUNCTION или-ли

[или-ли парам]

AS или-ли типа

[или-ли]

[или-ли = возвраще]

[или-ли]

End Function

3) произ. обр-ки событий.

вып-ся авт-ки в ответ на событие

Может быть вызв. с-мой, програм-  
ми или действиями пользователя.

или-ли - не возвращает

Разработка сетевых

11.12.03г.

или-ли <sup>или-ли</sup> <sub>или-ли</sub>

Сетевое прил-е - общий доступ к БД  
(т.е. неск. пользо-ль могут работать в БД)

БД-лимитируется <sup>или-ли</sup> <sub>или-ли</sub>

Предоставить <sup>или-ли</sup> <sub>или-ли</sub>

1) самый простой

Предоставить <sup>или-ли</sup> <sub>или-ли</sub> доступ к сети

2) -||-||-||- к таблицам БД в сети

3) -||-||-||- к БД через и-ли

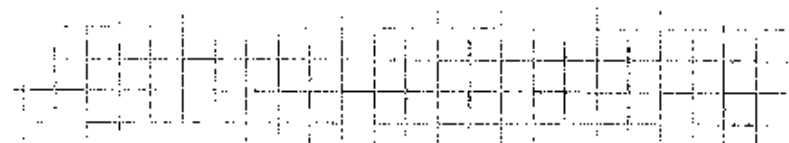
4) <sup>или-ли</sup> <sub>или-ли</sub> <sup>или-ли</sup> <sub>или-ли</sub> <sup>или-ли</sup> <sub>или-ли</sub> <sup>или-ли</sup> <sub>или-ли</sub> <sup>или-ли</sup> <sub>или-ли</sub>

5) <sup>или-ли</sup> <sub>или-ли</sub> <sup>или-ли</sup> <sub>или-ли</sub> <sup>или-ли</sup> <sub>или-ли</sub> <sup>или-ли</sup> <sub>или-ли</sub> <sup>или-ли</sup> <sub>или-ли</sub>

- 1) - применение принципа 3-х уровней: клиент-сервер или клиент-сервер-сервер.  
Подходит, когда нет-я один и те же объекты БД.
- 2) очень часто используются, работает быстрее.  
БД делится на группы объектов БД и групп приложений (грубо говоря, 2 БД - в одной группе, в другой во второй).
- 3) объекты БД предоставляются в соответствии с требованиями. В интернете на основе объектов создаются страницы html (динамичные или статичные).  
можно и то, и другое просто отображает
- 4) Аккумуляция информации - копирование информации из баз в базу.  
Скорости никакой нет, и есть удобства: в каждой базе ретрансляция или интеграция информации.
- 5) Если на сервере нет-я SQL-сервер, то можно использовать для своих работ с БД, в том же Access.  
Затем все это перенести в Microsoft SQL-сервер.  
То есть выстраивать некую клиент-серверную архитектуру.  
Можно использовать Microsoft Access в клиент-серверном режиме.

46.

- 1) Основные свойства данных: удобство, безопасность, непротиворечивость, оптимальность.
  - 2) Структура объекта БД, описанная на сервере на стороне клиента только сервер-визуализация запрашивает инф, а так же сервер-доступ к данным, находится на сервере.
- Применение-сервер проектирует клиент-сервер. Приложение в Access.  
С его помощью создается связь с приложением на сервере.
- При описании проекта можно дать физическую связь к СУБД или построив ее с БД.
- Получено описания, таблицы и т.д. описано:  
- представлено  
- описание данных (матрица или матрица)  
- процедура (или в Access)  
Представлено - запросы, которые хранятся на сервере. Доступ на клиентской стороне.
- Примеры: проект, который привязан к таблицам



47.