

Контрольные вопросы

1. Что такое модель данных?
2. Что такое схема данных?
3. Что такое база данных?
4. Назовите основные объекты реляционной модели .
5. Основные принципы технологии баз данных.

Проектирование баз данных

2 семестр Лекция 1
Дисциплина «Технологии бд и СУБД»

Вопросы для рассмотрения

1. Концептуальное проектирование
2. Проектирование схемы БД

1 Проектирование БД

- Одна из наиболее трудоемких и сложных задач при создании АИС – проектирование базы данных, как основы подсистемы представления и обработки информации.
- Проектирование:
 - концептуальное
 - схемно-структурное
- В группу разработчиков входят специалисты:
 - по формализации предметной области (постановщики задач, формализаторы)
 - по программному обеспечению СУБД
 - технические дизайнеры и специалисты по эргономике

Проектирование БД

концептуальное проектирование

Концептуальное проектирование базы данных – процесс во многом *эвристический*, а адекватность построенной инфологической схемы предметной области проверяется *эмпирически* по анализу и проверке удовлетворения информационных потребностей пользователей.

Проектирование БД

концептуальное проектирование

Этапы концептуального проектирования:

- обзор и изучение области использования БД для формирования общего представления о предметной области
- формирование и анализ круга функций обработки данных
- определение основных объектов-сущностей предметной области и отношений между ними
- формализованное описание предметной области

- **Обзор и изучение области использования БД для формирования общего представления о предметной области** выполняется разработчиком в тесном взаимодействии с заказчиком
- Разработчик изучает *организационно-распорядительную документацию* ⇒ определяет:
 - Процессы обработки
 - участники
 - информационные потоки

предметная область
- Принципиальный момент – **фрагментирование предметной области** – разделение на организационные, технологические, функциональные и др. блоки

При фрагментировании предметной области формализатор должен ответить на такие *вопросы*:

- выделить перечень *фрагментов*, подлежащих отражению в БД
 - лица, принимающие решения
 - функционально-технологические структуры (подразделения)
- определить информационные *потребности* и информационные *результаты* деятельности фрагментов
 - какая информация
 - в каком виде
 - в какие сроки
- определить общие характеристики и содержание *процессов* потребления и обработки информации фрагментами
 - содержание информации
 - технологии обработки, передачи, использования

Ответы на эти вопросы помогают сформировать представление о существующей («*как есть*») технологии формирования, накопления, обработки и использования информации в рамках БД и проанализировать (вместе с заказчиком) «*узкие места*» и *недостатки* в существующей технологии

- После формирования общего представления о предметной области производится ***определение функций обработки данных.***
- Это делается на основе *декомпозиции*:
 - определяется предварительный перечень *пользователей* системы
 - уточняются *информационные потребности* пользователей

Проектирование БД

определение основных объектов предметной области

- Главный итоговый результат концептуального проектирования – ***определение основных объектов-сущностей предметной области и отношений между ними.***
- ***Отношения:***
 - организационные
 - технологические
- Отношения предметной области выражаются в ***документах:***
 - организационно-распорядительных
 - информационно-справочных
 - других нормативно-служебных документах

Проектирование БД

определение основных объектов предметной области

- Анализ «бумажной» документации \Rightarrow *перечень атрибутов*, характеризующих объекты и отношения предметной области
- В одном документе могут быть отражены атрибуты *разных объектов и отношений*.
- Два подхода формирования перечня объектов предметной области и их атрибутов:
 - дедуктивный
 - индуктивный

Проектирование БД

определение основных объектов предметной области

- Дедуктивный подход:
 - выделяются *основные понятия и категории*, которыми выражаются фрагменты предметной области
 - эти понятия и категории принимаются за основу *списка объектов-сущностей предметной области*
 - на основе анализа документации и взаимодействия с заказчиком формируются *атрибуты* выделенных объектов
- При определении списка атрибутов каждого объекта руководствуются соображениями *минимальной достаточности* – принцип «**бритвы Оккама**»:
 - перечень объектов и их атрибутов должен быть *достаточным*
 - перечень объектов и их атрибутов *не должен быть избыточным*

Проектирование БД

определение основных объектов предметной области

- Индуктивный подход:
 - формируется *общий перечень атрибутов предметной области*
 - на основе эвристического анализа проводится *агрегация атрибутов в группы, образующие объекты предметной области*
- Часть атрибутов и понятий предметной области выражают ***процессы-отношения*** между объектами.
- Такие атрибуты выделяются и анализируются ***параметры и характер связей***, которые они выражают:
 - структурность
 - направленность
 - множественность
 - обязательность

Проектирование БД

определение основных объектов предметной области

- Чаще всего выделение объектов-сущностей, их атрибутов и отношений-связей осуществляется **комбинированным способом на итерационной основе**
- Распространенный прием – «**обобщение**» некоторых понятий и атрибутов – объединение в одну сущность близких или однотипных понятий, категорий, атрибутов на основе анализа их частных проявлений и вариантов.

Проектирование БД

формализованное описание предметной области

- Формализованное описание концептуальной схемы базы данных осуществляется средствами одной из *семантических моделей данных*
- В большинстве случаев семантические модели применяются на стадии концептуального проектирования с последующим *преобразованием концептуальной схемы в структуру соответствующей реляционной БД.*
- Наиболее известные семантические модели:
 - *ER-модели* – диаграммы Бахмана (Пин-Шен-Чен, 1976)
 - *UML*- диаграммы классов

Проектирование БД

формализованное описание предметной области

- *Формализованное описание концептуальной схемы базы данных – основа эскизного проекта создания БД информационной системы*
- *Следующий шаг – построение средствами СУБД схемы БД, соответствующей концептуальной схеме*
- *Адекватность реализации концептуальной схемы БД определяется эвристически и эмпирически в ходе отладки и пилотной эксплуатации БД*

Проектирование схем реляционных БД

- При **проектировании** **схемы реляционной БД** можно выделить такую **последовательность** процедур:
 - определение перечня таблиц и их связей
 - определение перечня полей, типов полей, ключевых полей таблиц, установление связей между таблицами через внешние ключи
 - определение и установление индексов для полей таблиц
 - разработка списков (словарей) для полей с перечислимым характером значений данных
 - установление ограничений целостности по полям таблиц и связям
 - нормализация таблиц, доработка перечня таблиц и их связей

Первые пять процедур называют **процессом предварительного проектирования таблиц и связей между ними**

Проектирование и создание таблиц

- Для каждого объекта-сущности в реляционных СУБД проектируют соответствующую *таблицу*.
- **Поля таблиц** соответствуют *атрибутам* информационных объектов концептуальной схемы БД
- Основные базисные характеристики:
 - домен
 - поле-атрибут
 - кортеж
 - отношение
 - ключ
 - внешний ключ

- Кроме этого указывается **тип поля**. *Понятие типа поля в СУБД = тип в ЯП.*

- *Традиционно поддерживаемые СУБД простые типы:*
 - числовой
 - символьный
 - темпоральный (дата/время)
 - булевский (логический)

- **Современные СУБД:**
 - специализированные типы полей
 - денежный
 - OLE
 - MEMO
 - и др.
 - Сложные типы, позаимствованные из ЯП высокого уровня

- Домен \neq тип!
- **Домен** – подмножество базисного типа данных с определенной смысловой нагрузкой.
- *Пример* – множество всех имен из множества всевозможных значений символьного типа.

Проектирование и создание таблиц

- Требование уникальности кортежей \Rightarrow определение и установление **ключевых полей** таблиц реляционных СУБД
- Определение ключевого поля выполняется на основе *смыслового эвристического анализа* тематики таблицы + *принцип минимальной достаточности* \Rightarrow количество полей, образующих ключ таблицы, должно быть минимальным
- Часто как ключевые поля используются поля типа «AUTOINCREMENT» (AUTOINC, Счетчик)

Проектирование и создание таблиц

- Реляционная модель обеспечивает лишь два типа связей-отношений:
 - *Один-ко-многим*
 - *создание внешнего ключа*
 - *Установление связи*
 - *Один-к-одному*
 - *связывание таблиц по одноименным и одностипным полям*

■ Пример:

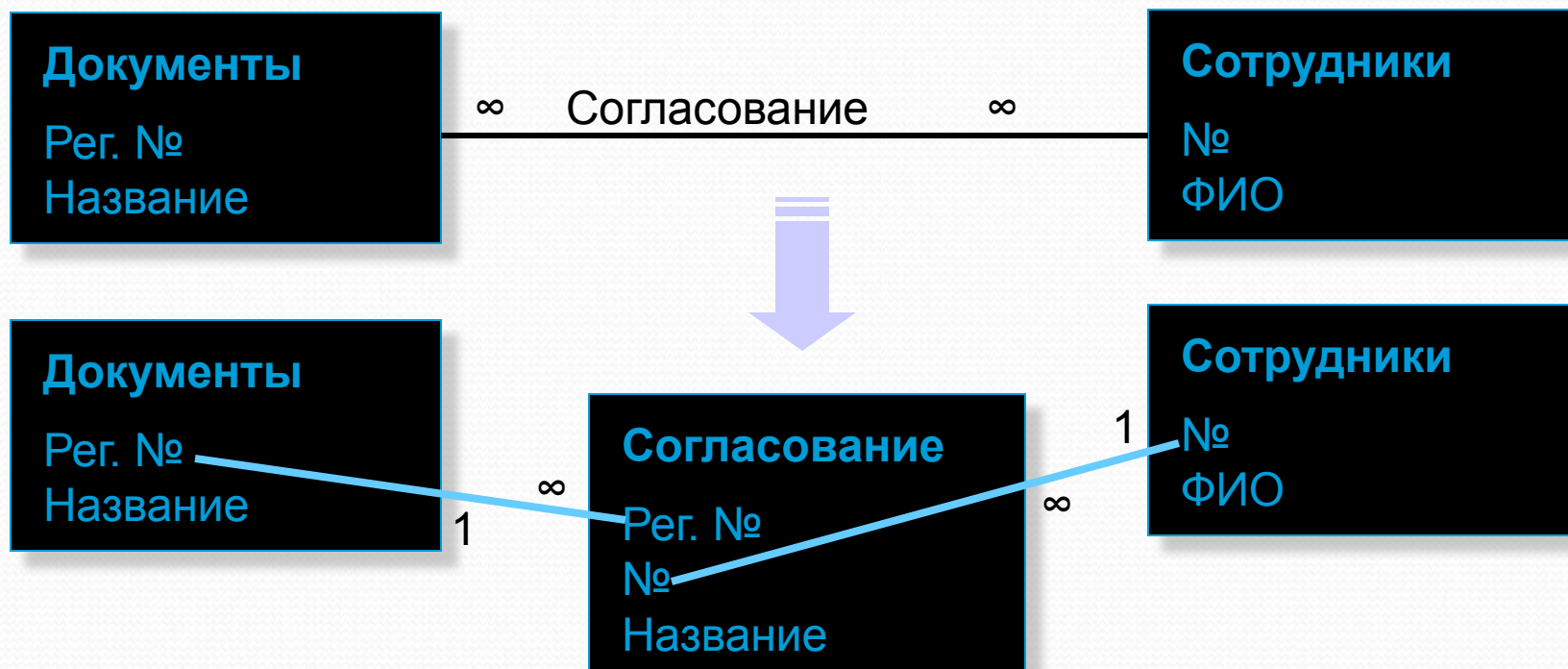
№ зачетки	ФИО	Группа	Комната в общежитии
567	Иванов П.С.	М-21	
568	Петров С.И.	М-23	77
569	Сидоров И.П.	М-21	

№ зачетки	ФИО	Группа
567	Иванов П.С.	М-21
568	Петров С.И.	М-23
569	Сидоров И.П.	М-21

№ зачетки	ФИО	Комната в общежитии
568	Петров С.И.	77

Проектирование и создание таблиц

- Связи типа **«Многие-ко-многим»** в реляционных СУБД реализуются через создание двух связей **«Один-ко-многим»** - введение **связной** таблицы
- **Пример:**



Проектирование и создание таблиц

■ Пример – реализация:

Рег. №	Название

№	ФИО

Один документ согласован двумя сотрудниками

<i>Согласование</i>		
Рег. №	№	Дата
1497с	132	15.02.04
1497с	134	17.02.04
4774	132	16.02.04

Один сотрудник согласовал два документа

Проектирование и создание таблиц

- Важный момент проектирования таблиц – определение необходимости **индексирования** тех или иных полей таблиц
- Если в одной таблице установлено более 10 индексов, то:
 - недостаточно продумана структура БД (таблицы)
или
 - неправильно выбраны поля, по которым чаще всего производится поиск
- ключевые поля индексируются автоматически
- внутреннее устройство индексных массивов обычно остается скрытым и для пользователей и для разработчиков

Проектирование и создание таблиц

- Также важное значение имеет выделение полей с **перечислимым** (перечислительным, словарным, списковым) **характером значений**.
- Значения таких полей определяются из некоторого унифицированного **списка-словаря**
- Словари (списки):
 - *фиксированные*
 - «привязываются» к соответствующим полям БД и размещаются в каталоге БД
 - *динамические*
 - реализуются через создание дополнительных одностолбцовых таблиц

Проектирование и создание таблиц

- В практическом плане важным является ***установление ограничений целостности по полям и связям***
- ***Ограничения по значениям полей***
 - уникальность кортежей \Rightarrow уникальность значений ключевых полей
 - UNIQUE
 - NOT NULL
 - допустимые диапазоны значений полей
 - относительные соотношения значений по полям таблицы
- Ограничения целостности данных отражают ту часть ***правил и особенностей предметной области, которая не формализуется в реляционной модели (бизнес-правила)***

Проектирование и создание таблиц

- Три подхода реализации *требования целостности по ссылкам*:
 - *запрет удаления записи*, если на нее существуют ссылки из других таблиц
 - при удалении записи значения внешних ключей всех связанных записей становятся *неопределенными*
 - *каскадное удаление* всех записей, связанных с удаляемой
- Разделение процесса проектирования таблиц на этапы является условным, а сам процесс предварительного проектирования (создания) таблиц реализуется инструкциями SQL

Нормализация таблиц

- **Нормализация** реляционных таблиц-отношений – следствие:
 - требования *атомарности значений* полей
 - требования *рациональности группировки* полей-атрибутов по различным таблицам
- **Нормализация таблиц** – доработка концептуальной схемы данных
- **Нормализация таблиц** – последовательный процесс разбиения и преобразования некоторого небольшого исходного набора таблиц для построения набора взаимосвязанных таблиц в *нормальных формах*

Нормализация таблиц

Наиболее простая – ***первая нормальная форма*** =
требование атомарности полей и единственности
значений по полям в реляционной модели

Нормализация таблиц

- Пример приведения к первой нормальной форме:

Личный №	Фамилия	Звание	Мероприятия в кот. принимал участие		Кабинет	Телефон
			Кодовое название	Награда		



Личный №	Кодовое название мероприятия	Награда	Фамилия	Звание	Кабинет	Телефон

Нормализация таблиц

- Таблицы в первой нормальной форме могут содержать:
 - ситуации дублирования данных
 - аномалии схемы таблиц-отношений



- Е. Кодд разработал механизм разбиения таблиц для приведения к более совершенным нормальным формам – **функциональная зависимость полей-атрибутов**

Нормализация таблиц

Поле-атрибут Y **функционально зависит** от поля атрибута X , если любому значению X всегда соответствует *одно и только одно* значение Y

В первой нормальной форме все *неключевые атрибуты функционально зависят от ключа* таблицы.

Нормализация таблиц

- ***Вторая нормальная форма*** основана на понятии *полной функциональной зависимости*
- Функциональная зависимость неключевого атрибута от составного ключа таблицы называется *полной*, если он зависит от составного ключа в целом, но не зависит от любой его части
- Таблица находится во *второй нормальной форме*, если она находится в *первой нормальной форме* и все ее неключевые атрибуты функционально *полно* зависят от составного ключа

Нормализация таблиц

- Пример приведения во вторую нормальную форму:

Личный №	Кодовое название мероприятия	Награда	Фамилия	Звание	Кабинет	Телефон



Личный №	Кодовое название мероприятия	Награда

Личный №	Фамилия	Звание	Кабинет	Телефон

Нормализация таблиц

- В таблицах, находящихся во второй нормальной форме *большинство аномалий*, присущих первой нормальной форме, *устранено*.
- Вместе с тем, могут сохраниться ситуации дублирования данных \Leftarrow ***транзитивная зависимость атрибутов***

Нормализация таблиц

- Таблица-отношение находится в **третьей нормальной форме**, если она находится во *второй нормальной форме* и каждое ее неключевое поле-атрибут *нетранзитивно* зависит от первичного ключа
- Третья нормальная форма = *взаимная независимость неключевых атрибутов и их полная функциональная зависимость от первичного ключа*

Нормализация таблиц

- Пример приведения в третью нормальную форму:

Личный №	Фамилия	Звание	Кабинет	Телефон



Личный №	Фамилия	Звание	Кабинет

Кабинет	Телефон

Нормализация таблиц

- Третья нормальная форма **устраняет**:
 - большинство *аномалий схем таблиц-отношений*
 - ситуации *дублирования данных*
- В некоторых случаях третью нормальную форму можно «улучшить» приведением в *нормальную форму Бойса-Кодда* \Leftarrow **детерминанты** – совокупность атрибутов, от которых функционально полно зависят другие атрибуты
- «Улучшения» нормальной формы Бойса-Кодда связаны с **многозначной зависимостью атрибутов**
- Существуют также **четвертая** и **пятая** нормальные формы

Нормализация таблиц

- *Нормализация* исходных таблиц при проектировании БД проводится для рационализации группировки полей-атрибутов в схемах таблиц с целью *устранения аномалий и дублирования данных*
- *Нормализация* = декомпозиция исходных таблиц на множество связанных и не связанных более простых таблиц ⇒ при обработке данных выполняется множество операций соединения таблиц ⇒ *обычно ограничиваются третьей нормальной формой*

Нормализация таблиц

- Результат проектирования и нормализации таблиц – законченная *схема* (логическая структура) *БД*
- Технологически описание схемы БД помещается в *каталог БД (системную таблицу)*
- Обычно каталог БД хранится в *файле БД вместе с данными*

Нормализация таблиц

- Для повышения эффективности схемно-структурного проектирования БД применяются **CASE-системы**:
 - Designer (Oracle)
 - ERWin/BPWin
 - PowerBuilder
 - UML tools
- Визуализированная концептуальная схема БД *транслируется* CASE-системой в схему соответствующей реляционной БД
- Подобные тенденции наблюдаются во многих современных СУБД, в т.ч. *персональных*
- Появляются также специальные анализаторы структуры таблиц БД

ИТОГИ

- Создание БД – сложный многоэтапный процесс, требующий привлечения различных категорий специалистов – программистов, инженеров, управленческих и др. работников
- Проектирование БД:
 - концептуальное
 - схемно-структурное
- Существует пять нормальных форм, но чаще всего останавливаются на третьей
- CASE-системы позволяют визуально построить концептуальную схему БД и оттранслировать ее в схему реляционной БД



Спасибо за внимание