

Лекция

Проектирование баз данных

Требования к проекту базы данных

Основные требования, которым должен удовлетворять

проект базы данных (БД):

- 1. Корректность схемы БД.
- 2. Обеспечение ограничений на ресурсы вычислительной системы.
- 3. Эффективность функционирования.
- 4. Обеспечение защиты данных.
- 5. Гибкость.
- 6. Простота и удобство эксплуатации.

Удовлетворение первых 4-х требований обязательно для принятия проекта.

Этапы проектирования АИС

В создании АИС (автоматизированной информационной системы)

можно выделить следующие этапы:

- • Предпроектная подготовка.
- • Проектирование базы данных.
- • Реализация (создание базы данных и прикладного программного обеспечения, ППО).

Этапы проектирования АИС

Специалисты, необходимые для выполнения этой работы:

- Аналитики (специалисты исследуемой предметной области).
- Пользователи – те работники, для которых создаётся АИС.
- Проектировщики (разработчики базы данных).
- Администраторы (системные, базы данных, безопасности и др.)
- Программисты (разработчики программного обеспечения).

Определение общих требований к системе

1. Предварительный анализ ПО.
2. Рассмотрение и принятие результатов анализа.
3. Определение критических факторов успеха.
4. Оценка системных ограничений.
5. Определение целевой архитектуры.

Определение общих требований к системе

6. Определение требований к производительности.

Требования к производительности зависят от режима, в котором будет функционировать система:

- 1) Интерактивный режим.
- 2) Пакетный режим.
- 3) Режим реального времени.

7. Согласование стандартов проектирования

8. Выбор программных средств для проектирования и реализации системы

Определение требований пользователей

Собственно процесс проектирования БД включает в себя следующие основные этапы:

- 1. Информационно-логическое (инфологическое) проектирование.
- 2. Определение требований к операционной обстановке, в которой будет функционировать информационная система.
- 3. Выбор СУБД и других инструментальных программных средств.
- 4. Логическое проектирование БД.
- 5. Физическое проектирование БД.

После того, как проект базы данных создан, наступает этап реализации проекта. Он разбивается на следующие шаги:

- 1. Создание прототипа БД и его отладка.
- 2. Разработка и отладка приложений.
- 3. Конвертирование и загрузка данных в БД.

Определение требований пользователей

- 4. Тестирование работы базы данных и АИС в целом. Различают такие виды тестов, как:
 - *автономные;*
 - *тесты связей;*
 - *регрессивные;*
 - *нагрузочные;*
 - *системные;*
 - *приёмо-сдаточные.*
- 5. Эксплуатация и сопровождение АИС.

Этапы проектирования БД

- I. Информационно-логическое (инфологическое) проектирование
- II. Определение требований к операционной обстановке:
- III. Выбор СУБД и других инструментальных программных средств.
- IV. Логическое проектирование БД (даталогическое):
- V. Физическое проектирование БД:

I. Инфологическое проектирование

Инфологическая модель ПрО включает описание структуры и динамики ПрО, характера информационных потребностей пользователей системы

Обратите внимание: инфологическая модель ПрО не должна зависеть от модели данных, которая будет использована при создании БД.

- 1. Определение границ предметной области (ПрО).
- 2. Анализ ПрО.

І. Інфологічне проектування

- 3. Методи аналізу:
 - функціональний,
 - предметний;
 - метод сутність-св'язь – entity-relation method, ER-метод.
- 4. ER-метод (сутність-св'язь), основні поняття:
 - **сутність;**
 - **атрибут;**
 - **св'язь.**

Анализ ПрО с помощью ER-метода

Сущности:

- базовые
- зависимые

Обычно описание ПО выражается в терминах не отдельных сущностей и связей между ними, а их типов, связанных с ними ограничений целостности и тех процессов, которые приводят к переходу ПО из одного состояния в другое.

Выделяют понятия ***тип сущности и экземпляр сущности***.

Тип позволяет выделить из всего множества сущностей ПрО группу

сущностей, однородных по структуре и поведению (относительно рамок рассматриваемой ПрО).

Данные в БД представлены экземплярами сущностей.

Анализ ПрО с помощью ER-метода

Атрибуты сущностей:

- 1. Идентифицирующие и описательные атрибуты.*
- 2. Составные и простые атрибуты.*
- 3. Однозначные и многозначные атрибуты.*
- 4. Основные и производные атрибуты.*
- 5. Обязательные и необязательные.*

Для каждого атрибута необходимо определить **название, указать тип данных и**

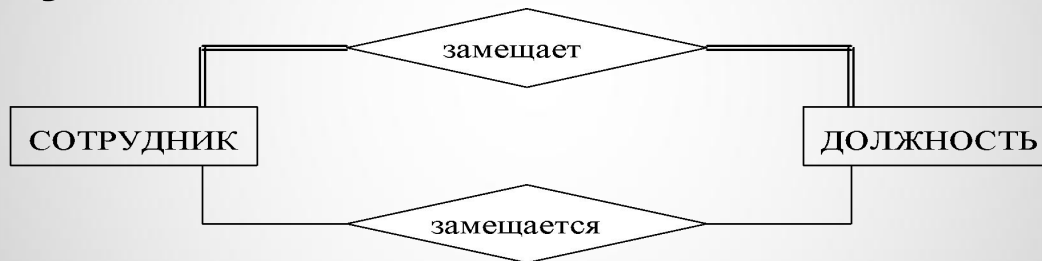
описать ограничения целостности – множество значений, которые может принимать данный атрибут.

Анализ ПрО с помощью ER-метода

Связи между сущностями:

Для связи указывается:

- название,
- тип (факультативная или обязательная),



- кардинальность (1:1, 1:n или m:n),
- степень (унарная, бинарная, тернарная или n-арная).
- Различают *тип связи* и *экземпляр связи*.

Анализ ПрО с помощью ER-метода

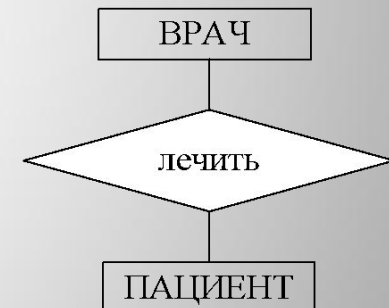
- **Кардинальность связей между сущностями:**
 - один-к-одному (1:1);
 - один-ко-многим (1:n);
 - многие-ко-многим (m:n).

Анализ ПрО с помощью ER-метода

- Степень связей между сущностями:
- унарная:
- бинарная:
- тернарная:



а) унарная связь



б) бинарная связь



в) тернарная связь

Модель предметной области

Совокупность типов сущностей и типов связей между ними характеризует

структуру предметной области.

Собственно данные представлены **экземплярами сущностей и связей**

между ними. Данные экземпляров сущностей и связей хранятся в базе

данных информационной системы, а описание типов сущностей и связей

является **метаданными.**

Множества экземпляров сущностей, значения атрибутов сущностей и

экземпляры связей между ними могут изменяться во времени.

Поэтому

каждому моменту времени можно сопоставить некоторое **состояние**

предметной области.

Ограничения целостности – это правила, которым должны удовлетворять значения данных в БД.

Моделирование локальных представлений

Если ПрО содержит много сущностей (10 и более), то она разбивается на ряд локальных областей (**локальных представлений**) по 6-7 сущностей.

Каждое локальное представление включает в себя информацию, достаточную для обеспечения информационных потребностей одной группы будущих пользователей.

Каждое локальное представление моделируется отдельно.

При объединении локальных представлений используют концепции:

- **Идентичность..**
- **Агрегация.**
- **Обобщение.**

На этапе объединения локальных представлений необходимо устранить все противоречия.

Объединение локальных представлений

- **Использование обобщения:**
- Например, пусть в объединяемых представлениях присутствуют
- следующие сущности:
- *ДЕТАЛИ СОБСТВЕННОГО ПРОИЗВОДСТВА*
- *ДЕТАЛИ ПОКУПНЫЕ*
- *СБОРОЧНЫЕ ЕДИНИЦЫ ПОКУПНЫЕ*
- *СБОРОЧНЫЕ ЕДИНИЦЫ СОБСТВЕННОГО ПРОИЗВОДСТВА*
- Их можно объединить так :

Объединение локальных представлений

- Элементы изделий предприятия
 1. Покупные
 - а) Сборочные единицы
 - б) Детали
 2. Собственного производства
 - а) Сборочные единицы
 - б) Детали

Результаты инфологического проектирования

Концептуальная инфологическая модель ПрО.

Она фиксируется в виде общей ER-диаграммы предметной области.

Модели локальных представлений .

На этапе анализа ПрО решаются следующие задачи:

Правила (ограничения) целостности.

Перечень групп пользователей системы.

Внешние спецификации функций (процессов).

Определение требований к операционной обстановке

На этом этапе производится:

1. оценка требований к вычислительным ресурсам, необходимым для функционирования системы;
2. выбор типа и конфигурации ЭВМ;
3. выбор типа и версии операционной системы (ОС).

Выбор зависит от таких показателей, как:

- примерный объём данных в БД;
- динамика роста объёма данных;
- характер запросов к данным;
- интенсивность запросов к данным по типам запросов;
- требования ко времени отклика системы по типам запросов;
- режим работы.

Выбор СУБД

Наиболее важные критерии выбора СУБД:

- тип модели данных,
 - адекватность модели данных структуре рассматриваемой ПО;
 - характеристики производительности СУБД;
 - запас функциональных возможностей для дальнейшего развития
- информационной системы;
- степень оснащённости СУБД инструментарием для персонала
- администрирования данными;
- удобство и надежность СУБД в эксплуатации;
 - наличие специалистов по работе с конкретной СУБД;
 - стоимость СУБД и дополнительного программного обеспечения.

Логическое проектирование РБД

- Преобразование ER-диаграммы в схему базы данных.

Правила преобразования:

- 1. Каждый тип сущности преобразуется в таблицу БД.
- 2. Бинарная связь 1:n (между сущностями разных типов) реализуется с помощью внешнего ключа между двумя таблицами

Логическое проектирование РБД

- Преобразование ER-диаграммы в схему базы данных.

Правила преобразования:

- 3. Каждая связь со степенью больше двух и связь, имеющая атрибуты, преобразуется в таблицу БД.

Логическое проектирование РБД

- Преобразование ER-диаграммы в схему базы данных.

Правила преобразования:

- 4. Связь 1:1 реализуется в рамках одной таблицы.
- 5. Унарная связь 1:n (между сущностями одного типа) реализуется с помощью внешнего ключа, определённого в той же таблице, что и первичный ключ.

Логическое проектирование РБД

- Преобразование ER-диаграммы в схему базы данных.

Правила преобразования:

- 6. Бинарная связь типа $n:m$ реализуется с помощью промежуточной таблицы.

Логическое проектирование РБД

- Преобразование ER-диаграммы в схему базы данных.

Правила преобразования:

- 7. Унарная связь $n:m$ реализуется с помощью промежуточной таблицы.
- На этом этапе возможно ещё выявление нереализуемых и необычных связей (связи $1:n$, обязательные в обе стороны; взаимоисключающие связи и др.).

Логическое проектирование РБД

- Составление схем отношений.
- Определение первичных ключей (ПК):
 1. При наличии потенциальных ключей ПК выбирается из них.
 2. Если потенциальных ключей нет, назначается суррогатный ПК
 3. Составной ПК назначается в том случае, если необходимо реализовать ограничение целостности "уникальность".

Логическое проектирование РБД

Определение типов данных атрибутов. **Общие рекомендации:**

- Для коротких символьных значений и символьных строк фиксированной длины следует выбирать тип CHAR.
- Для символьных строк переменной длины нужно выбирать тип VARCHAR с указанием максимально возможной длины хранимого значения.
- Для числовых атрибутов, не участвующих в сложных расчётах, нужно использовать основной числовой тип реляционных СУБД – тип NUMBER.
- Для числовых атрибутов, которые участвуют в сложных расчётах, следует использовать такие числовые типы, которые хранят данные в машинном (двоичном) представлении.
- Для числовых атрибутов, имеющих ведущие нули, следует выбирать тип CHAR, а не числовой тип, иначе ведущие нули будут потеряны.
- Для хранения дат нужно выбирать тип DATE или его варианты (DATETIME, например).
- Для хранения больших объектов (графических, звуковых и т.п.) следует выбирать специальные типы данных, перечень которых зависит от СУБД.
- Для семантически одинаковых полей разных таблиц нужно выбирать одинаковые типы данных.

Логическое проектирование РБД

Определение и реализация ограничений целостности:

Рассмотрим различные типы ограничений целостности в языке SQL:

- Уникальность значения первичного ключа (PRIMARY KEY).
- Уникальность ключевого поля или комбинации значений ключевых полей (UNIQUE).
- Обязательность/необязательность значения (NOT NULL/NULL).
- Задание условия на значения атрибутов (CHECK).
- Определение домена атрибута на основе значений другого атрибута:
(внешний ключ, FOREIGN KEY).

Логическое проектирование РБД

- Определение и реализация ограничений целостности.
- Если какое-либо ограничение целостности (ОЦ) нельзя реализовать средствами DCL, то возможны следующие способы его реализации:
 - С помощью процедурных объектов БД .
 - Программно (т.е. через приложение).
 - Вручную.
- Необходимо обратить особое внимание на поля таблиц, для которых домен определён как список возможных значений. Это ограничение целостности можно реализовать в виде: CHECK(<поле> IN (<список значений>)).
- Но такой подход имеет следующий недостаток: добавление нового значения в список потребует изменения схемы отношения.
- Можно поступить до-другому: вынести этот список значений в отдельное отношение.

Физическое проектирование РБД

- При использовании СУБД примерная последовательность создания объектов БД следующая:
- 1. Создание БД.
- 2. Создание пользователей .
- 3. Создание пользовательских типов.
- 4. Создание кластеров и таблиц.
- 5. Создание представлений.
- 6. Создание синонимов.
- 7. Создание последовательностей.
- 8. Назначение прав доступа.
- 9. Заливка данных.
- 10. Создание индексов.
- 11. Создание процедур и функций.
- 12. Создание триггеров.

Лекция

**Перспективы
развития технологии
базы данных**

Вступление

- Вот уже более 30-и лет базы данных являются одной из одной из наиболее широко востребованных информационных технологий. Некоторые авторы утверждают, что появление баз данных стало самым важным достижением в области программного обеспечения. Системы баз данных коренным образом изменили работу многих организаций, и практически нет такой области деятельности, которую они не затронули.

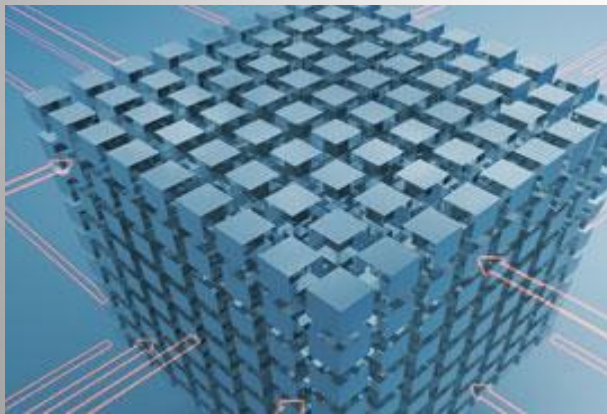
К числу наиболее важных и перспективных направлений развития БД следует отнести следующие:

- Хранилища данных и OLAP-обработка.
- Работа с неточными данными.
- Новые пользовательские интерфейсы
- Проблемы оптимизации запросов
- Интеграция разнородных и слабо формализованных данных
- Организация доступа к базам данных через Internet.
- Самоадаптация.
- Использование GRID.
- Сохранность данных
- Технологии разработки данных и знаний



Хранилища данных и OLAP-обработка

- Хранилище данных – это предметно-ориентированный, интегрированный, привязанный ко времени и неизменяемый набор данных, предназначенный для поддержки принятия решений. Хранилище данных позволяют сохранять исторические данные с целью анализа и прогнозирования развития ситуаций. При правильном проектировании хранилище данных даёт высокую отдачу за счёт более качественного управления работой организации.



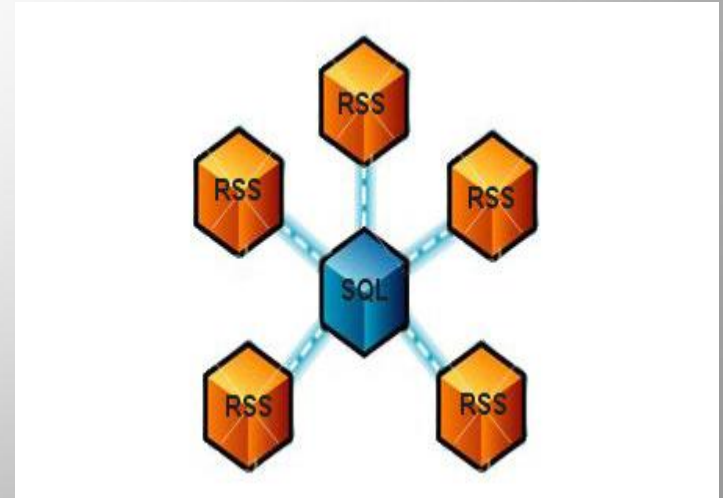
Работа с неточными данными

- Информация в базах данных часто содержит ошибки или является неполной. Результаты запроса по такой БД могут сильно отличаться от реального положения дел.



Новые пользовательские интерфейсы

- Это одно из наиболее актуальных направлений современных информационных технологий. Конечные пользователи не знают язык запросов (SQL), и для получения информации из БД вынуждены пользоваться интерфейсами, которые для них создают программисты.
- Для того, чтобы воспользоваться конструктором, пользователь должен знать структуру базы данных и хорошо разбираться в предложенном ему формализме ПО.



Новые пользовательские интерфейсы

- Наиболее естественным видом является запрос к БД, сформулированный на естественном языке (ЕЯ). Но для таких запросов характерны неточности и неоднозначность. Решение этой задачи невозможно без использования знаний о предметной области и о структуре языка. Одним из вариантов решения этой проблемы являются онтологии. Интеграция онтологий и баз данных позволит пользователям задавать запросы в собственной терминологии с использованием ограниченного естественного языка.

Проблемы оптимизации запросов

Помимо остающейся актуальной задачи поиска новых способов оптимизации, можно выделить ещё две серьёзные проблемы оптимизации:

- обработка неструктурированных запросов,
- оптимизация группы запросов.

Работа с неструктурированными запросами особенно актуальна в свете использования баз данных в поисковых системах (в том числе, при поиске в Internet).



Интеграция разнородных и слабо формализованных данных

- Изначально базы данных предназначались для хранения и обработки фактографических хорошо структурированных данных. Но огромное количество данных представлено в различных графических и мультимедийных форматах. Включение в СУБД способов обработки подобных данных позволяет использовать технологии баз данных в разных сферах.

Организация доступа к базам данных через Internet

- Многие web-сайты содержат динамическую информацию, например, о товарах и ценах в Internet-магазинах. В локальных системах такая информация традиционно хранится в базах данных.



Организация доступа к базам данных через Internet

Основные задачи:

Организация
эффективного
интерфейса;

Оптимизация запросов;

Повышение
производительности
СУБД в
многопользовательском
режиме работы.



Самоадаптация

- Современные СУБД имеют широкие возможности по настройке баз данных под конкретную предметную область и аппаратные средства. Но использование этих возможностей – достаточно сложная задача, которая требует наличия высококвалифицированного администратора БД.

Использование GRID.

- GRID – это концепция объединения. Так же при возникновении потребности в вычислениях пользователь должен просто подключаться к GRID и получать вычислительные ресурсы. Преимущества этого подхода очевидны: возможность решать более ресурсоёмкие задачи и перераспределять



Использование GRID.

Тем не менее, первые промышленные GRID-системы уже существуют, но поддерживают они только базы данных: это системы Oracle 10G и Oracle 11G (G – это сокращение от GRID). Они динамически выделяют ресурсы для выполнения задач пользователя по доступу к БД.

Сохранность данных.

- Количество накопленных цифровых данных в мире огромно. Но со временем устаревают и форматы хранения данных, и средства доступа к ним.
- Даже архивированные данные могут стать недоступными, особенно если нет устройства для чтения устаревшего носителя. Решить эту проблему могут средства, обеспечивающие миграцию данных в новые форматы с сохранением их описания (т.е. метаданных).

Технологии разработки данных и знаний

- Технологии разработки данных предназначены для поиска неочевидных тенденций и скрытых закономерностей в больших объёмах данных. Knowledge mining – это извлечение знаний из баз данных. Здесь используются как формальные методы, так и методы интеллектуальной обработки данных, основанные на моделировании познавательных механизмов – индукции, дедукции, абдукции.

Выводы

- Технологическая среда во всем мире меняется очень быстро, и вместе с этим расширяются наши представления о сферах применимости баз данных. Растущие информационные потребности общества отчетливо выявляют ограничения существующих технологий СУБД, и задача исследовательского сообщества – самым энергичным образом устремить свои усилия на эти новые направления. Спектр возможностей и потребностей здесь широк, как никогда, – от сугубо теоретических изысканий в области создания новых моделей и алгоритмических основ до реализации прототипов новаторских систем.