СИСТЕМЫ БАЗЫ ДАННЫХ

**Основы систем баз данных: понятие, характеристика, структура. Модели данных. Представление в нормальной форме. Ограничение целостности данных. Запросы и их обработка. Основы международного стандартного языка для определения и доступа к реляционным базам данных. Параллельная обработка данных и их восстановление. Проектирование и разработка баз данных. Технология программирования объектно-реляционного отображения. Распределенные, параллельные и гетерогенные базы данных.**

<https://www.tutorialspoint.com/dbms/dbms_overview.htm>

<https://people.inf.elte.hu/miiqaai/elektroModulatorDva.pdf>

<https://theswissbay.ch/pdf/Gentoomen%20Library/Databases/Molina%2CUllman%20-%20Database%20Systems%20The%20Complete%20Book.pdf>

**ОСНОВЫ БАЗ ДАННЫХ: ПОНЯТИЕ, ХАРАКТЕРИСТИКА, СТРУКТУРА.**

Весьма вероятно, что вы слышали термин «база данных». Если у вас есть какие-то данные, вам нужно где-то их хранить. Данные могут быть любыми: данные клиента, данные о продукте, данные о сотрудниках и т. д. Также данные могут быть представлены в разных формах. Некоторые из них: текст, цифры, мультимедиа, модели, аудио, код, программное обеспечение, видео и инструмент.

База данных - это набор компьютеризированных файлов данных, который управляется СУБД (системой управления базами данных). Пользователи системы могут выполнять различные операции:

* Добавление новых файлов в базу данных
* Ввод данных в существующие файлы
* Получение данных обратно из существующих файлов
* Удаление данных из существующих файлов
* Изменение данных в существующих файлах
* Удаление существующих файлов из базы данных

Система управления базами данных (СУБД) представляет собой программный пакет, предназначенный для выполнения всех перечисленных операций в базе данных. Прикладная программа обращается к данным, хранящимся в базе данных, отправляя запросы к СУБД. (рисунок)



Первые системы управления базами данных произошли от файловых систем, которые обеспечивают хранение больших объемов данных, файловые системы хранят данные в течение длительного периода времени и позволяют хранить большие объемы данных. Тем не менее, файловые системы, как правило, не гарантируют, что данные не могут быть потеряны, если они не сохранены, и не поддерживают эффективный доступ к элементам данных, местоположение которых неизвестно. Современная СУБД имеет следующие характеристики:

1. Обмен данными - система поддерживает многопользовательскую среду и позволяет им параллельно получать доступ к данным и управлять ими.

2. Уменьшение избыточности - СУБД следует правилам нормализации, которая разделяет взаимосвязь, когда любые из ее внешние данные имеют избыточность в значениях.

3. Предотвращение несовместимости - СУБД может обеспечить большую согласованность по сравнению с более ранними формами приложений для хранения данных, таких как системы обработки файлов

4. Обеспечение приема и передачи информации - СУБД следует концепциям элементарности, согласованности, изоляции и долговечности (автоматическая классификация и интерпретация данных). Свойства автоматической классификации и интерпретации данных помогают базе данных оставаться работоспособной в мультитранзакционных средах и в случае сбоя.

5. Поддержание целостности - СУБД может предотвратить ввод неверной информации в таблицы с помощью ограничений целостности.

6. Обеспечение безопасности - СУБД предлагает множество различных функций безопасности, что позволяет пользователям иметь разные представления о разных функциях.

Теперь мы можем представить структуру системы баз данных. Структура СУБД объясняет, как пользователь просматривает данные в базах данных. Сегодня большая часть структуры СУБД основана на структуре базы данных ANSI-SPARC.

Структура ANSI-SPARC разделена на три уровня, обычно называемые внутренним, внешним и концептуальным уровнями.

* Внутренний уровень (также известный как уровень хранения) является наиболее близким к реальной памяти, и связан с тем, как хранятся данные в системе.
* Внешний уровень (также известный как логический уровень пользователя) является наиболее близким к пользователям, и связан с тем, как данные воспринимаются отдельными пользователями.
* Концептуальный уровень (также известный как логический уровень сообщества) - это уровень использования косвенной адресации между двумя другими уровнями.



Структура СУБД

Ранее мы обсуждали системы баз данных с точки зрения так называемой структуры ANSI / SPARC, то сейчас мы предложим несколько иной взгляд.

Общая цель системы баз данных заключается в поддержке разработки и работы приложений баз данных. Таким образом, с точки зрения высокого уровня, такую систему можно рассматривать как очень простую двухкомпонентную структуру, состоящую из сервера, также называемого внутренним сервером, и набора клиентов, также называемых внешними интерфейсами.

Сервер - это сама СУБД. Он поддерживает все основные функции СУБД - определение данных, управление данными, безопасность и целостность данных и так далее.



Как видите, структура сервера отвечает за выполнение текущих операций в базах данных.

Разработка базы данных

Цели разработки базы данных:

* Удовлетворить требования к содержанию информации указанных пользователей и приложений
* Обеспечить естественное и легкое для понимания структурирование информации
* Поддержка требований к обработке и любых целей производительности, таких как время отклика, время обработки и объем памяти.



Также можно определить основные этапы разработки и реализации базы данных:

 1. Сбор и анализ требований

 2. Концептуальная разработка базы данных

 3. Выбор СУБД

 4. Отображение модели данных (логическая разработка базы данных)

 5. Физическая разработка базы данных

 6. Внедрение и настройка системы баз данных

Процесс разработки, как можно увидеть (рисунок), состоит из двух действий: первое включает в себя разработку содержимого данных, структуры и ограничений базы данных; вторая относится к разработке приложений баз данных.

Процесс разработки состоит из двух параллельных действий, как показано на рисунке 10.1. Первое действие включает в себя разработку содержимого данных, структуры и ограничений базы данных; второе действие относится к разработке приложений баз данных. Чтобы не усложнять фигуру, мы старались не показывать большинство взаимодействий между сторонами, но эти два вида деятельности тесно взаимосвязаны. Например, анализируя приложения базы данных, мы можем определить «элементы данных, которые будут храниться в базе данных». Кроме того, этап разработки физической базы данных, на котором мы выбираем структуры хранения и пути доступа к файлам базы данных, зависит от приложений, которые будут использовать эти файлы для запросов и обновления. С другой стороны, мы обычно указываем разработку приложений базы данных, ссылаясь на схемы базы данных, которые указываются во время выполнения первого действия. Очевидно, что эти два вида деятельности сильно влияют друг на друга. Традиционно методологии разработки баз данных были сосредоточены в первую очередь на первом из этих видов деятельности, тогда как разработка программного обеспечения была сосредоточена на втором; это можно назвать управляемый данными, а не управляемый данными процесса проектированием. Сейчас разработчики баз данных и разработчики программного обеспечения признают, что эти два действия должны идти рука об руку, и инструменты разработки все чаще объединяют их.

**МОДЕЛЬ ДАННЫХ**

Понятие «модель данных» является одним из самых фундаментальных в исследовании систем баз данных. Модель данных - это представление для описания данных или информации. Оно определяет, как данные связаны друг с другом и как они обрабатываются и хранятся в системе.

Описание модели данных обычно состоит из трех частей:

1. Структура данных - это совокупность типов структуры данных (компоновочные блоки любой базы данных, которая соответствует модели)
2. Операции над данными - это совокупность операторов или правил вывода, которые могут применяться к любым действительным экземплярам типов данных.
3. Ограничения данных - это совокупность общих правил целостности, которые неявно или явно задают набор согласованных положений базы данных.

Система управления базами данных может быть классифицирована в соответствии с моделью данных на:

**Иерархическая.** Иерархическая модель данных представляет данные в виде иерархических древовидных структур. Каждая иерархия представляет ряд связанных записей.



**Родственная**. Родственная модель данных представляет базу данных в виде набора таблиц, где каждая таблица храниться в виде отдельного файла.



**Сетевая**. Сетевая модель данных представляет данные как типы записей, а также представляет ограниченный тип отношения 1: N, называемый типом набора.

****

**Объектно-ориентированная**. Объектно-ориентированная модель данных определяет базу данных с точки зрения объектов, их свойств и операций.



(Объяснение схемы <https://stackoverflow.com/questions/7815560/query-on-an-object-oriented-database>)

**Основы международного стандартного языка для определения и доступа к реляционным базам данных.**

Если вы создаете приложения, выполняете административные задачи, генерируете отчеты, вам необходимо знать, как управлять данными в вашей базе данных и взаимодействовать с ними. Несмотря на то, что сегодня существует много типов систем управления базами данных (СУБД), родственная модель баз данных является наиболее популярной, и системы международного стандартного языка для определения и доступа к реляционным базам данных стали доминировать на рынке СУБД.

Международный стандартный язык для определения и доступа к реляционным базам данных - это язык для генерации, обработки и извлечения данных из родственной модели базы данных. Одна из причин популярности родственной модели баз данных заключается в том, что правильно спроектированные родственные модели базы данных могут обрабатывать огромные объемы данных.

Международный стандартный язык для определения и доступа к реляционным базам данных включает в себя как компонент языка определения данных (DDL), так и компонент языка манипулирования данными (DML). Язык манипулирования данными международного стандартного языка для определения и доступа к реляционным базам данных может работать как на внешнем (в представлениях), так и на концептуальном (в базовых таблицах) уровнях. Язык определения данных международного стандартного языка для определения и доступа к реляционным базам данных может использоваться для определения объектов на внешнем уровне (представления), концептуальном уровне (базовые таблицы) и интервальном уровне (индексы или другие вспомогательные структуры).

Определив базу данных, теперь можно начать работать с ней с помощью манипулятивных SQL-операций SELECT (ВЫБОР), INSERT (ВВОД), DELETE (УДАЛЕНИЕ) и UPDATE (УТОЧНЕНИЕ).

SELECT (ВЫБОР) - оператор, который выбирает данные из таблиц базы данных

INSERT (ВВОД) - оператор, который вставляет новые данные в таблицы базы данных

DELETE (УДАЛЕНИЕ) - оператор, который удаляет существующие данные из таблиц базы данных

UPDATE (УТОЧНЕНИЕ) - оператор, который обновляет существующие данные в таблицы базы данных

Операторы языка определения данных

CREATE (СОЗДАНИЕ) - оператор, который создает базу данных или ее объекты

DROP (ПРОСМОТР) - оператор, который удаляет объекты из базы данных

ALTER (ИЗМЕНЕНИЕ) - оператор, который изменяет структуру базы данных

**Ограничение целостности данных.**

Как мы уже говорили, модель данных состоит из двух частей: манипулятивной части, определяющей типы операций, которые разрешены для данных, и набора ограничений целостности, которые обеспечивают точность данных.

Есть два важных правила целостности, которые являются ограничениями или запретами, которые применяются ко всем экземплярам базы данных. Два основных правила для родственной модели: сущность и ссылочная целостность. Другими типами ограничения целостности являются множественность и общие ограничения.

*Целостность объекта* гарантирует, что в таблице нет повторяющихся записей и что поле, которое идентифицирует каждую запись в таблице, уникально и никогда не равно нулю.

*Ссылочная целостность* гарантирует, что взаимосвязь между таблицами в базе данных остается точной, применяя ограничения для запрета пользователям или приложениям вводить неточные данные или указывать на данные, которые не существуют.

*Множественность*

*Общие ограничения*

**Нормализация**

Когда мы разрабатываем базу данных, основной целью является создание точного представления данных, взаимосвязей между данными и их содержания. Для достижения данной цели, мы можем использовать больше методов проектирования баз данных. Одной из них является нормализация.

Нормализация - это метод организации данных в базе данных. Нормализация использует серию испытаний (описанных как нормальные формы), чтобы помочь определить оптимальную группировку для этих атрибутов, чтобы в конечном итоге идентифицировать набор подходящих отношений, которые поддерживают требования к данным.

Характеристики подходящего набора отношений включают в себя:

* минимальное количество атрибутов, необходимых для поддержки требований к данным;
* атрибуты с тесной логической связью (описываются как функциональная зависимость)
* минимальная избыточность, каждый атрибут представлен только один раз, за исключением важных атрибутов, которые образуют все части внешних ключей, которые необходимы для объединения связанных отношений



Шаги нормализации:

* Выбрать источник данных и преобразовать в ненормализованную таблицу (UNF)
* Преобразовать ненормализованные данные в первую нормальную форму (1NF)
* Преобразовать данные в первой нормальной форме (1NF) во вторую нормальную форму (2NF)
* Преобразование данных во второй нормальной форме (2NF) в третью нормальную форму (3NF)

Иногда данные могут все еще подвергаться аномалиям в третьей нормальной форме. В таком случае нам, возможно, придется выполнить дальнейшие преобразования.

* Преобразовать третью нормальную форму в нормальную форму Бойса-Кодда (BCNF)
* Преобразовать нормальную форму Бойса-Кодда в четвертую нормальную форму (4NF)
* Преобразовать четвертую нормальную форму в пятую нормальную форму (5NF)

**Технология программирования объектно-реляционного отображения (1200)**

Объектно-реляционное отображение (ORM) - это метод хранения, извлечения, обновления и удаления из объектно-ориентированной программы в реляционной базе данных.

Объектно-реляционное отображение использует «уровень данных» для управления переводом между объектно-ориентированным и реляционным

Уровень данных, как правило, представляет собой библиотеку, написанную на объектно-ориентированных языках, таких как Java или C#, которая является частью или работает в сочетании с вашей веб-инфраструктурой.



****

**Распределенные, параллельные и гетерогенные базы данных.**