

7. СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

Оглавление

7. СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ	1
7.1. Классификация СУБД	1
7.2. Правила Кодда для реляционной СУБД (РСУБД)	2
7.3. Основные функции реляционной СУБД	4
7.4. Администрирование базы данных	5
7.5. Словарь-справочник данных	6

Система управления базами данных (СУБД) – это важнейший компонент АИС, основанной на базе данных. СУБД необходима для создания и поддержки базы данных информационной системы в той же степени, как для разработки программы на алгоритмическом языке – транслятор. Программные составляющие СУБД включают в себя ядро и сервисные средства (утилиты).

Ядро СУБД – это набор программных модулей, необходимый и достаточный для создания и поддержания БД, то есть универсальная часть, решающая стандартные задачи по информационному обслуживанию пользователей. **Сервисные программы** предоставляют пользователям ряд дополнительных возможностей и услуг, зависящих от описываемой предметной области и потребностей конкретного пользователя.

Системой управления базами данных называют программную систему, предназначенную для создания на ЭВМ общей базы данных для множества приложений, поддержания её в актуальном состоянии и обеспечения эффективного доступа пользователей к содержащимся в ней данным в рамках предоставленных им полномочий.

Принципиально важное свойство СУБД заключается в том, что она позволяет различать и поддерживать два независимых взгляда на БД: "взгляд" пользователя, воплощаемый в "логическом" представлении данных, и "взгляд" системы – "физическое" представление (организация хранимых данных).

Для инициализации базы данных разработчик средствами конкретной СУБД описывает логическую структуру БД, её организацию в среде хранения и пользовательские представления данных (соответственно концептуальную схему БД, схему хранения и внешние схемы). Обработывая эти схемы, СУБД создаёт пустую БД требуемой структуры и предоставляет средства для наполнения её данными предметной области и дальнейшей эксплуатации.

7.1. Классификация СУБД

По степени универсальности СУБД делят на два класса: СУБД **общего назначения** (СУБД ОН) и **специализированные** СУБД (СпСУБД).

СУБД ОН не ориентированы на какую-либо предметную область или на конкретные информационные потребности пользователей. Каждая система такого рода является универсальной и реализует функционально избыточное множество операций над данными. СУБД ОН имеют в своём составе средства настройки на конкретную предметную область, условия эксплуатации и требования пользователей. Производство этих систем поставлено на широкую коммерческую основу.

Специализированные СУБД создаются в тех случаях, когда ни одна из существующих СУБД общего назначения не может удовлетворительно решить задачи, стоящие перед разработчиками. Причин может быть несколько:

- не достигается требуемого быстродействия обработки данных;
- необходима работа СУБД в условиях жёстких аппаратных ограничений;
- требуется поддержка специфических функций обработки данных.

СпСУБД предназначены для решения конкретной задачи, а приемлемые параметры этого решения достигаются следующим образом:

- 1) за счёт знания особенностей конкретной предметной области,
- 2) путём сокращения функциональной полноты системы.

Создание СпСУБД – дело весьма трудоёмкое, поэтому для того, чтобы выбрать этот путь, надо иметь действительно веские основания. В дальнейшем будут рассматриваться только СУБД общего назначения.

По методам организации хранения и обработки данных СУБД делят на **централизованные** и **распределённые**. Первые работают с БД, которая физически хранится в одном месте (на одном компьютере). Это не означает, что пользователь может работать с БД только за этим же компьютером: доступ может быть удалённым (в режиме клиент–сервер). Большинство централизованных СУБД перекладывает задачу организации удалённого доступа к данным на сетевое обеспечение, выполняя только свои стандартные функции, которые усложняются за счёт одновременности доступа многих пользователей к данным.

По модели данных различают **иерархические, сетевые, реляционные, объектно-реляционные** и **объектно-ориентированные** СУБД.

Для реляционных СУБД Э.Ф. Кодд предложил и обосновал 12 правил, которым должна удовлетворять реляционная СУБД данных (РСУБД).

7.2. Правила Кодда для реляционной СУБД (РСУБД)

1. Явное представление данных (The Information Rule). Информация должна быть представлена в виде данных, хранящихся в ячейках. Данные, хранящиеся в ячейках, должны быть атомарны. Порядок строк в реляционной таблице не должен влиять на смысл данных.
2. Гарантированный доступ к данным (Guaranteed Access Rule). К каждому элементу данных должен быть гарантирован доступ с помощью комбинации имени таблицы, первичного ключа строки и имени столбца.
3. Обработка неизвестных значений (Systematic Treatment of Null Values). Неизвестные значения NULL, отличные от любого известного значения,

должны поддерживаться для всех типов данных при выполнении любых операций. Например, для числовых данных неизвестные значения не должны рассматриваться как нули, а для символьных данных – как пустые строки.

4. Динамический каталог данных, основанный на реляционной модели (Dynamic On-Line Catalog Based on the Relational Model). Каталог (или *словарь-справочник*) данных должен сохраняться в форме реляционных таблиц, и РСУБД должна поддерживать доступ к нему при помощи стандартных языковых средств, тех же самых, которые используются для работы с реляционными таблицами, содержащими пользовательские данные.
5. Полнота подмножества языка (Comprehensive Data Sublanguage Rule). РСУБД должна поддерживать единственный язык, который позволяет выполнять все операции над данными: определение данных (DDL, Data Definition Language), манипулирование данными (DML, Data Manipulation Language), управление доступом пользователей к данным, управление транзакциями.
6. Поддержка обновляемых представлений (View Updating Rule). Представление (view) – это хранимый запрос к таблицам базы данных. Обновляемое представление должно поддерживать все операции манипулирования данными, которые поддерживают реляционные таблицы: операции вставки, модификации и удаления данных.
7. Наличие высокоуровневых операций управления данными (High-Level Insert, Update, and Delete). Операции вставки, модификации и удаления данных должны поддерживаться не только по отношению к одной строке таблицы, но по отношению к любому множеству строк произвольной таблицы.
8. Физическая независимость данных (Physical Data Independence). Приложения не должны зависеть от используемых способов хранения данных на носителях, от аппаратного обеспечения компьютера, на котором находится БД. РСУБД должна предоставлять некоторую свободу модификации способов организации базы данных в среде хранения, не вызывая необходимости внесения изменений в логическое представление данных. Это позволяет оптимизировать среду хранения данных с целью повышения эффективности системы, не затрагивая созданных прикладных программ, работающих с БД.
9. Логическая независимость данных (Logical Data Independence). Это свойство позволяет сконструировать несколько различных логических взглядов (представлений) на одни и те же данные для разных групп пользователей. При этом пользовательское представление данных может сильно отличаться не только от физической структуры их хранения, но и от концептуальной (логической) схемы данных. Оно может синтезироваться динамически на основе хранимых объектов БД в процессе обработки запросов.

10. Независимость контроля целостности (Integrity Independence). 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. Это реализуется с помощью ограничений целостности и механизма транзакций.
11. Независимость от распределённости (Distribution Independence). База данных может быть распределённой (может находиться на нескольких компьютерах), и это не должно оказывать влияние на приложения. Перенос базы данных на другой компьютер не должен оказывать влияние на приложения.
12. Согласование языковых уровней (Non-Subversion Rule). Не должно быть иного средства доступа к данным, отличного от стандартного языка для работы с данными. Если используется низкоуровневый язык доступа к данным, он не должен игнорировать правила безопасности и целостности, которые поддерживаются языком более высокого уровня.

7.3. Основные функции реляционной СУБД

Основные функции реляционной СУБД определяются правилами Кодда. Но потребности пользователей обуславливают также следующие функции:

1. Поддержка многопользовательского режима доступа.

База данных создаётся для решения многих задач многими пользователями. Это подразумевает возможность одновременного доступа многих пользователей к данным. Данные в БД являются разделяемым ресурсом, и РСУБД должна обеспечивать разграничение доступа к ним.

2. Обеспечение физической целостности данных.

Проблема обеспечения физической целостности данных обусловлена возможностью разрушения данных в результате сбоев и отказов в работе вычислительной системы или в результате ошибок пользователей. Развитые РСУБД позволяют в большинстве случаев восстановить потерянные данные. Восстановление данных чаще всего основано на периодическом создании резервных копий БД и ведении журнала регистрации изменений (журнала транзакций).

3. Управление доступом.

Для многопользовательских систем актуальна проблема защиты данных от несанкционированного доступа. Каждый пользователь этой системы в соответствии со своим уровнем (приоритетом) имеет доступ либо ко всей совокупности данных, либо только к её части. Управление доступом также подразумевает предоставление прав на проведение отдельных операций над отношениями или другими объектами БД.

4. Настройка РСУБД.

Настройка РСУБД обычно выполняется администратором БД, отвечающим за функционирование системы в целом. В частности, она может включать в себя следующие операции:

- подключение внешних приложений к БД;
- модификация параметров организации среды хранения данных с целью повышения эффективности системы;
- изменение структуры хранимых данных или их размещения в среде хранения (**реорганизация БД**) для повышения производительности системы или повторного использования освободившейся памяти;
- модификацию концептуальной схемы данных (**реструктуризация БД**) при изменении предметной области и/или потребностей пользователей.

Задачи администратора БД (АБД) достаточно важны, поэтому на них следует остановиться несколько подробнее.

7.4. Администрирование базы данных

Основные задачи администрирования базы данных – обеспечение надежного и эффективного функционирования системы БД, адекватности содержания БД информационным потребностям пользователей, отображения в БД актуального состояния ПО.

Администрирование БД возлагается на администратора (или персонал администрирования, если система БД велика). В задачи администратора входит выполнение нескольких групп функций:

1. Администрирование предметной области: поддержка представления БД на концептуальном уровне архитектуры СУБД (общем для всех приложений); адекватное отображение в БД изменений, происходящих в ПО. Последнее требование может подразумевать реструктуризацию (изменение схемы) БД и последующее приведение содержимого БД в соответствие с новой схемой.
2. Администрирование БД: поддержка представления БД в среде хранения, эффективная и надежная эксплуатация системы БД. Если на этом уровне проводится реорганизация БД (с целью повышения эффективности работы), то она заключается в следующем:
 - изменения в структуре хранимых данных, например, выведение в отдельную таблицу редко используемых данных;
 - изменения способов размещения данных в памяти, например:
 - разбиение таблицы на части для распределения её по различным физическим носителям с целью распараллеливания доступа к ней;
 - построение кластеров ;
 - изменение физических параметров среды хранения, например, размера блока данных в пространстве памяти.
 - изменения используемых методов доступа к данным, например, построение индексов или введение хеширования.
3. Администрирование приложений: поддержка представлений БД для различных групп пользователей механизмами внешнего уровня СУБД. При

изменении концептуальной схемы БД или схемы хранения может потребоваться внесение соответствующих изменений в приложения.

4. Администрирование безопасности данных: предоставление пользователям прав на доступ к БД и настройка системных средств защиты от несанкционированного доступа.

В состав СУБД обычно включаются вспомогательные средства (различные утилиты), упрощающие администрирование БД.

7.5. Словарь-справочник данных

Словарь-справочник данных (ССД) – это программная система, предназначенная для централизованного хранения и использования описания объектов БД (метаданных). Иногда ССД называют *каталогом данных*. Эта система содержит сведения:

- о владельцах объектов данных, пользователях ресурсов данных и полномочиях их доступа;
- о составе и структуре базы данных;
- об ограничениях целостности;
- о вспомогательных объектах и компонентах информационной системы.

ССД обеспечивает непротиворечивость метаданных, единую точку зрения на базу данных всего персонала разработчиков, администраторов и пользователей системы. Метаданные в словаре-справочнике реляционной СУБД обычно организованы в виде набора таблиц и представлений.

Словарь БД служит для поддержки функционирования компонентов программного обеспечения – СУБД и прикладных программ, работающих с БД. Словарь содержит сведения об организации БД, её составе и структуре, описание данных: форматы представления, структуру, методы доступа, способы размещения данных в памяти и т.п. Информация в словаре представлена в виде, удобном для программного использования.

Справочник БД содержит сведения о семантике данных, способах их идентификации, источниках данных и т.п. Справочник предназначен главным образом для документирования разработки БД и справочного обслуживания её пользователей. Информация в справочнике представлена в виде, удобном для восприятия человеком.

Множества метаданных словаря и справочника в значительной мере пересекаются. Более того, они могут реализовываться совместно: во многих РСУБД словарь состоит из таблиц (table), содержащих описание объектов БД, а справочник реализуется с помощью представлений (view) над таблицами словаря.

Далее мы познакомимся с теми механизмами, с помощью которых в СУБД организуется хранение данных и доступ к ним.