

Информатика и программирование

Основные понятия баз данных

Доцент кафедры ИВТ, к.т.н.
Проскурин Александр Викторович

Файловые системы

Файловые системы — это набор программ, которые выполняют для пользователей некоторые операции (например, создание отчета).

Каждая программа определяет свои собственные данные и управляет ими.

Ограничения файловых систем:

- разделение и изоляция данных;
- дублирование данных;
- зависимость от данных;
- несовместимость файлов;
- фиксированные запросы и быстрое увеличение количества приложений.

Системы баз данных

Все перечисленные выше ограничения файловых систем являются следствием двух факторов:

1. Определение данных содержится внутри приложений, а не хранится отдельно и независимо от них.
2. Помимо приложений не предусмотрено никаких других инструментов доступа к данным и их обработки.

Для повышения эффективности работы необходимо использовать новый подход, а именно **базу данных и систему управления базами данных.**

Базы данных

База данных (БД, database) – это информационная модель предметной области, совокупность взаимосвязанных, хранящихся вместе данных при наличии такой минимальной избыточности, которая допускает их использование оптимальным образом для одного или нескольких приложений.

Система управления базами данных (СУБД, Database Management System) – это программа, с помощью которой реализуется централизованное управление данными, хранящимися в базе, доступ к ним, поддержка их в актуальном состоянии.

История развития СУБД

Считается, что развитие СУБД началось в 60-е годы XX века, когда разрабатывался проект полета корабля Apollo на Луну.

Специалисты основного подрядчика – фирмы North American Aviation – разработали программное обеспечение под названием **GUAM** (Generalized Update Access Method).

Идея GUAM была построена на том, что малые компоненты объединяются вместе как части более крупных компонентов до тех пор, пока не будет собран воедино весь проект.

Эта соответствующая инвертированному дереву структура называется **иерархической структурой** (hierarchical structure).

Модели данных

Классическими являются три типа моделей описания баз данных:

1. иерархическая,
2. сетевая,
3. реляционная.

Основное различие между ними состоит в характере описания взаимосвязей и взаимодействия между объектами и атрибутами базы данных.

Иерархическая модель

Предполагает использование для описания базы данных **древовидных структур**, состоящих из определенного числа уровней.

«Дерево» представляет собой иерархию элементов, называемых **узлами**.

Под элементами понимается список, совокупность, набор атрибутов, элементов, описывающих объекты.

Пример иерархической структуры



Иерархическая модель

Достоинства:

- простота построения;
- легкость понимания принципа иерархии;
- наличие промышленных СУБД, поддерживающих данную модель.

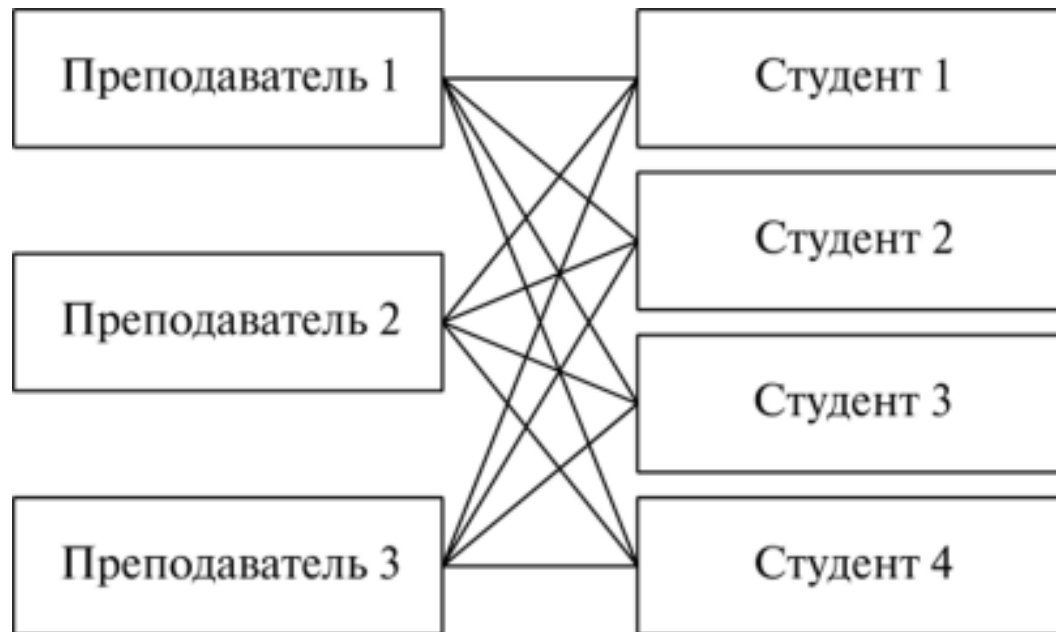
Недостатки:

- сложность включения информации о новых видах данных в модель;
- сложность удаления устаревшей информации.

Сетевая модель

Описывает элементарные данные и отношения между ними в виде **ориентированной сети**.

Это такие отношения между объектами, когда каждый порожденный элемент имеет более одного исходного и может быть связан с любым другим элементом структуры.



Сетевая модель

Достоинства:

- высокая эффективность при описании сложных моделей данных (отношения многие ко многим, структуры знаний).

Недостатки:

- потеря независимости данных при модификации базы данных;
- сложность понимания модели для пользователя.

Реляционная модель

В реляционных базах данных вся информация представляется в виде **двумерных таблиц**.

С ее созданием начинается новый этап в эволюции СУБД.

Эдгар Ф. Кодд в 1970 издал работу «A Relational Model of Data for Large Shared Data Banks», которая считается первой работой по реляционной модели данных.

Простота и гибкость модели привлекли к ней внимание разработчиков и снискали ей множество сторонников.

Реляционная модель данных стала доминирующей, а реляционные СУБД стали промышленным стандартом “де-факто”.

Реляционная модель

Реляционная модель опирается на систему понятий реляционной алгебры, важнейшими из которых являются:

- **“таблица” (“отношение”),**
- **“строка” (“кортеж”),**
- **“столбец” (“атрибут”),**
- **“первичный ключ”,**
- **“внешний ключ”.**

Все операции над реляционной базой данных сводятся к манипуляциям с таблицами.

Реляционная модель

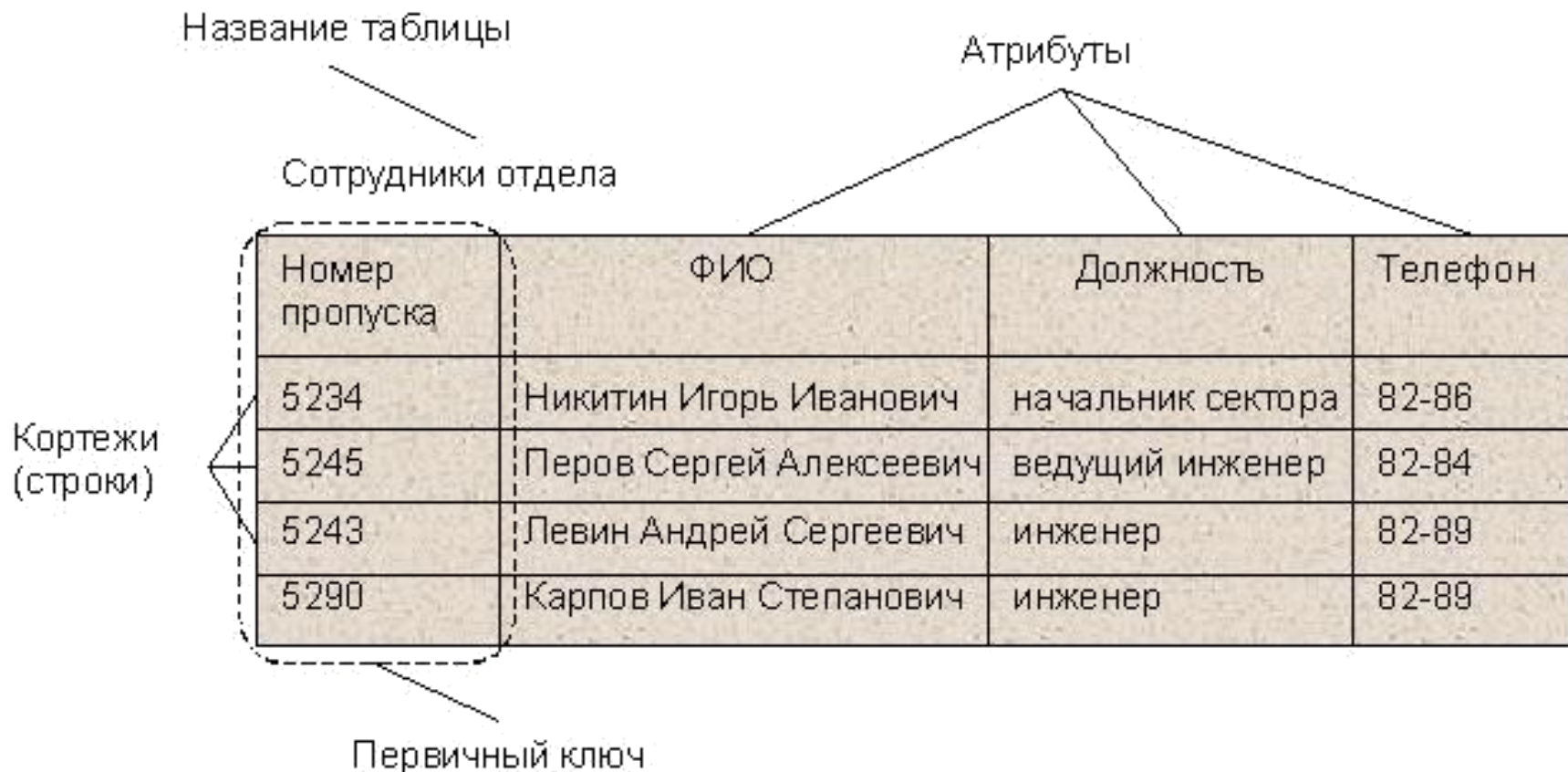


Рис. 2.7. Отношение реляционной базы данных

Реляционная модель

Номер пропуска	Фамилия Имя Отчество	Должность	Телефон
5234	Никитин Игорь Александрович	Начальник сектора	82-86
5245	Перов Сергей Алексеевич	Ведущий инженер	82-84
5243	Карпов Иван Степанович	Инженер	82-55
5290	Левин Андрей Сергеевич	Инженер	82-47

Сотрудники:

Номер пропуска	Фамилия Имя Отчество	Номер должности	Телефон
5234	Никитин Игорь Александрович	1	82-86
5245	Перов Сергей Алексеевич	2	82-84
5243	Карпов Иван Степанович	3	82-55
5290	Левин Андрей Сергеевич	3	82-47

Внешний ключ

Должности:

Номер должности	Название должности
1	Начальник сектора
2	Ведущий инженер
3	Инженер

Первичный ключ

Нормализация данных

Нормализация – это процесс организации данных в базе данных, включающий создание таблиц и установление отношений между ними в соответствии с правилами, которые обеспечивают **защиту данных** и делают базу данных более гибкой, **устраняя избыточность** и **несогласованные зависимости**.

Существует несколько правил нормализации баз данных. Каждое правило называется «нормальной формой».

Реляционная модель

Достоинства:

- простота построения;
- доступность понимания модели (естественное представление данных в виде двумерных таблиц);
- наличие богатого математического аппарата.

Недостатки:

- относительно низкая производительность;
- сложность программного обеспечения;
- избыточность данных.

Постреляционные модели

Классическая реляционная модель предполагает неделимость данных, хранящихся в полях записей таблиц.

Постреляционная модель данных представляет собой расширенную реляционную модель, снимающую ограничение неделимости данных, хранящихся в записях таблиц.

Постреляционная модель данных допускает многозначные поля — поля, значения которых состоят из подзначений.

Сопоставление реляционной и постреляционной моделей данных

Таблицы реляционной модели

Заказы:

Номер накладной	Код покупателя
21	3241
18	4075
43	2459

Товары:

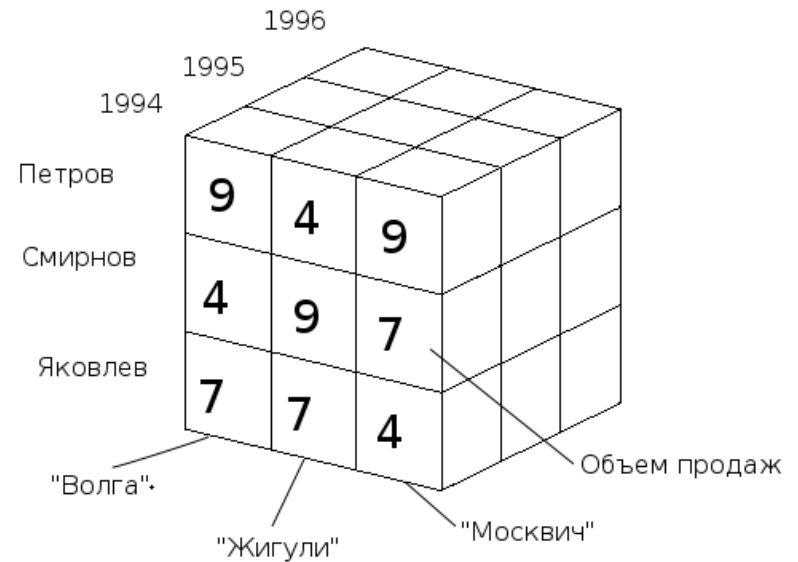
Номер накладной	Наименование товара	Количество
21	Соль	5
21	Сыр	7
18	Мед	3
43	Сок	10
43	Рыба	20
43	Мясо	30

Таблица постреляционной модели

Номер накладной	Код покупателя	Наименование товара	Количество
21	3241	Соль	5
		Сыр	7
18	4075	Мед	3
43	2459	Сок	10
		Рыба	20
		Мясо	30

Постреляционные модели

- объектная модель;
- объектно-реляционная модель;
- многомерные модели данных.



Централизованные и распределенные БД

Централизованная база данных обеспечивает простоту управления, улучшенное использование данных на местах при выполнении дистанционных запросов, более высокую степень одновременности обработки, меньшие затраты на обработку.

Распределенная база данных предполагает хранение и выполнение функций управления данными в нескольких узлах и передачу данных между этими узлами в процессе выполнения запросов.

Архитектуры централизованных баз данных

Системы централизованных баз данных с сетевым доступом предполагают несколько архитектур:

- **Централизованная архитектура.** СУБД и БД располагаются и функционируют на центральном компьютере (мэйнфрейме), а пользователи получают доступ к БД при помощи обычных терминалов (устройства ввода-вывода).
- **Архитектура файл-сервер.** Предполагает выделение одной из машин сети в качестве центральной (главный сервер файлов), где хранится совместно используемая централизованная база данных.

Архитектуры централизованных баз данных

- **Архитектура клиент-сервер.** Каждый из подключенных к сети и составляющих эту архитектуру компьютеров играет свою роль: сервер владеет и распоряжается информационными ресурсами системы, клиент имеет возможность пользоваться ими.
- **Трехуровневая архитектура.** Клиентская часть представляет устройство ввода-вывода, вся деловая логика вынесена на сервер приложений, который собственно и обеспечивает формирование запросов к БД, передаваемых на выполнение серверу БД.

Функциональные возможности СУБД

- производительность;
- обеспечение целостности данных на уровне баз данных;
- обеспечение безопасности данных;
- возможность работы в многопользовательских средах;
- возможность импорта и экспорта данных;
- обеспечение доступа к данным с помощью языка SQL;
- возможность составления запросов;
- наличие инструментальных средств разработки прикладных программ.

Обеспечение целостности данных

Обеспечение целостности данных подразумевает наличие средств, позволяющих удостовериться, что информация в базе данных всегда остается корректной и полной.

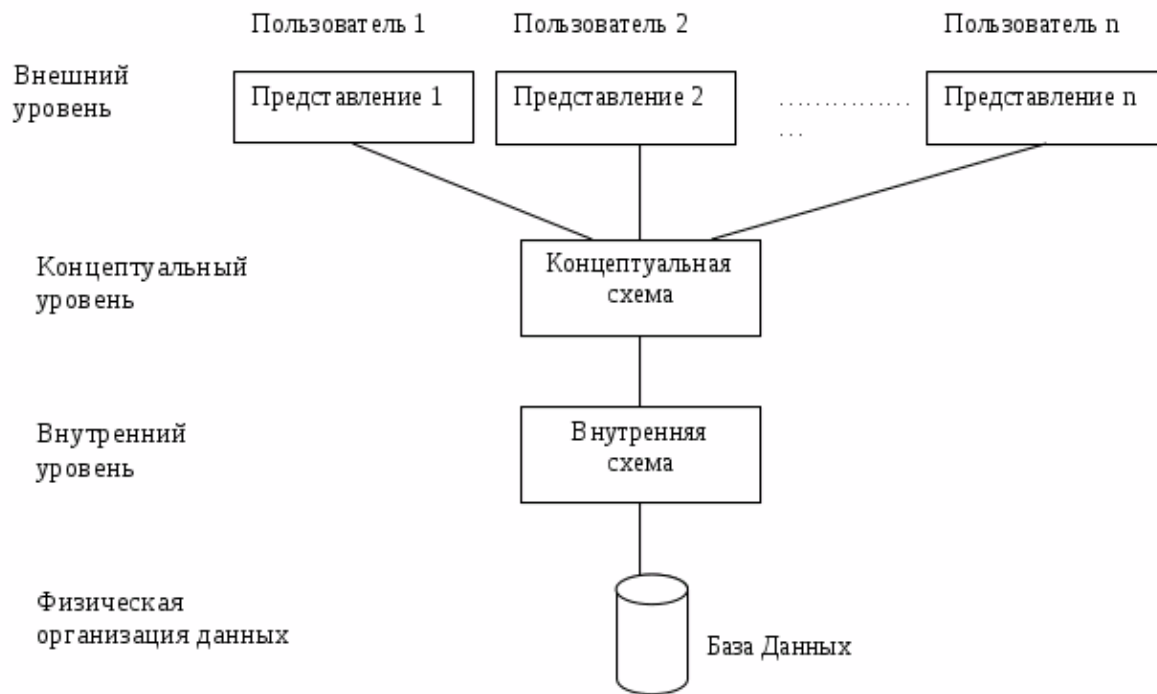
Система управления базами данных управляет данными во внешней памяти, обеспечивает надежное хранение данных и поддержку соответствующих языков базы данных.

Обеспечение безопасности данных

Безопасность данных достигается:

- шифрованием данных;
- защитой прикладных программ паролем;
- ограничением доступа к базе данных (к таблице, к словарю и т.д.).

Трехуровневая архитектура БД



➤ **Внешний уровень** — это представление базы данных с точки зрения пользователей. Этот уровень описывает ту часть базы данных, которая относится к каждому пользователю.

➤ **Концептуальный уровень** — это обобщающее представление базы данных. Этот уровень описывает

то, какие данные хранятся в БД, а также связи, существующие между ними.

➤ **Внутренний уровень** — это физическое представление базы данных в компьютере. Этот уровень описывает как информация хранится в базе данных.

Представления данных

В СУБД имеется механизм создания **представления** (view), который позволяет любому пользователю иметь собственный взгляд на базу данных.

Представления обладают следующими **достоинствами**:

- обеспечивают дополнительный уровень безопасности;
- предоставляют механизм настройки внешнего интерфейса базы данных;
- позволяют сохранять внешний интерфейс базы данных непротиворечивым и неизменным даже при внесении изменений в ее структуру.

Проектирование базы данных в СУБД

Этапами работы в СУБД являются:

- 1. создание структуры базы данных, т.е. определение перечня полей, из которых состоит каждая запись таблицы, типов и размеров полей;**
- 2. ввод и редактирование данных в таблицах баз данных;**
- 3. обработка данных, содержащихся в таблицах, на основе запросов и на основе программы (создание пользовательского интерфейса);**
- 4. вывод информации из ЭВМ с использованием отчетов и без использования отчетов.**

Преимущества СУБД

- контроль за избыточностью данных.
- непротиворечивость данных.
- совместное использование данных.
- поддержка целостности данных.
- повышенная безопасность.
- улучшение показателей производительности.
- упрощение сопровождения системы за счет независимости от данных.
- развитые службы резервного копирования и восстановления.

Недостатки СУБД

- Сложность.
- Размер.
- Стоимость СУБД.
- Дополнительные затраты на аппаратное обеспечение.
- Затраты на преобразование.
- Производительность.
- Более серьезные последствия при выходе системы из строя.