

Лекция 2

Архитектура Базы Данных

Независимость БД от приложений

Программы, с помощью которых пользователи работают с базой данных, называются приложениями. В общем случае с одной базой данных могут работать множество различных приложений.

Например, если база данных моделирует некоторое предприятие, то для работы с ней может быть создано приложение, которое обслуживает подсистему учета кадров, другое приложение может быть посвящено работе подсистемы расчета заработной платы сотрудников, третье приложение работает как подсистемы складского учета, четвертое приложение посвящено планированию производственного процесса.

Под **независимостью данных от приложений** и наоборот, приложений от данных, подразумевается, что изменение одних не приводит к изменению других.

При рассмотрении приложений, работающих с одной базой данных, предполагается, что они могут работать параллельно и независимо друг от друга, и **именно СУБД** призвана обеспечить работу множества приложений с единой базой данных таким образом, чтобы каждое из них выполнялось корректно, но учитывало все изменения в базе данных, вносимые другими приложениями.

Достижение независимости

Задача обеспечения независимости данных от приложений – совместная задача проектировщиков БД, разработчиков приложений и СУБД.

С точки зрения прикладного программирования, независимость данных является не техникой, а дисциплиной программирования. Например, для того чтобы при любом изменении избежать перекомпиляции, допустимо не определять константы в программе. Лучшее решение состоит в передаче программе значений в качестве параметров.

Одним из средств достижения независимости , реализуемым СУБД, является **трехуровневая архитектура БД**



Трехуровневая схема БД

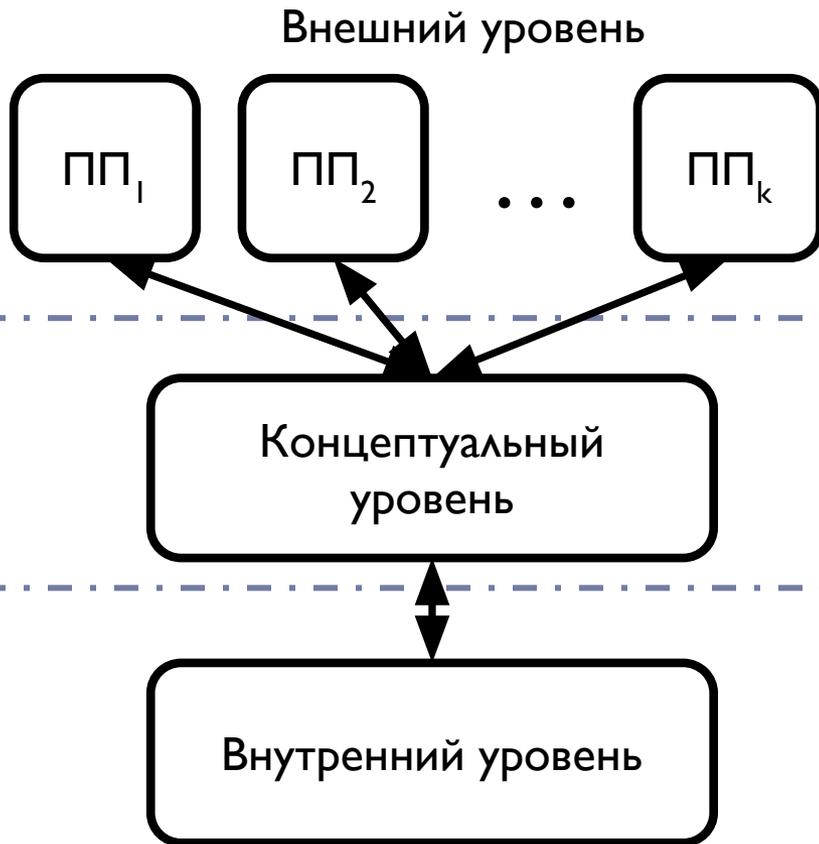
Одним из важнейших аспектов развития СУБД является идея **отделения логической структуры** БД и манипуляций данными, необходимыми пользователям, от **физического представления**, требуемого компьютерным оборудованием. Разграничение пользовательского и системного уровней.

В процессе научных исследований, посвященных тому, как именно должна быть устроена СУБД, предлагались различные способы реализации. Самым жизнеспособным из них оказалась предложенная американским комитетом по стандартизации ANSI (American National Standards Institute) трехуровневая система организации БД (1978)

В соответствии с принятой концепцией выделяют три уровня абстракции представления данных

1. Внешний - точка зрения на базу приложений и пользователей
 2. Концептуальный - общий вид, объединяющий точки зрения всех приложений, отображение внешнего уровня на внутренний уровень. Концептуальный уровень отражает обобщенную модель предметной области
 3. Внутренний – на котором СУБД и операционная система воспринимают данные ;
-

Трёхуровневая схема БД



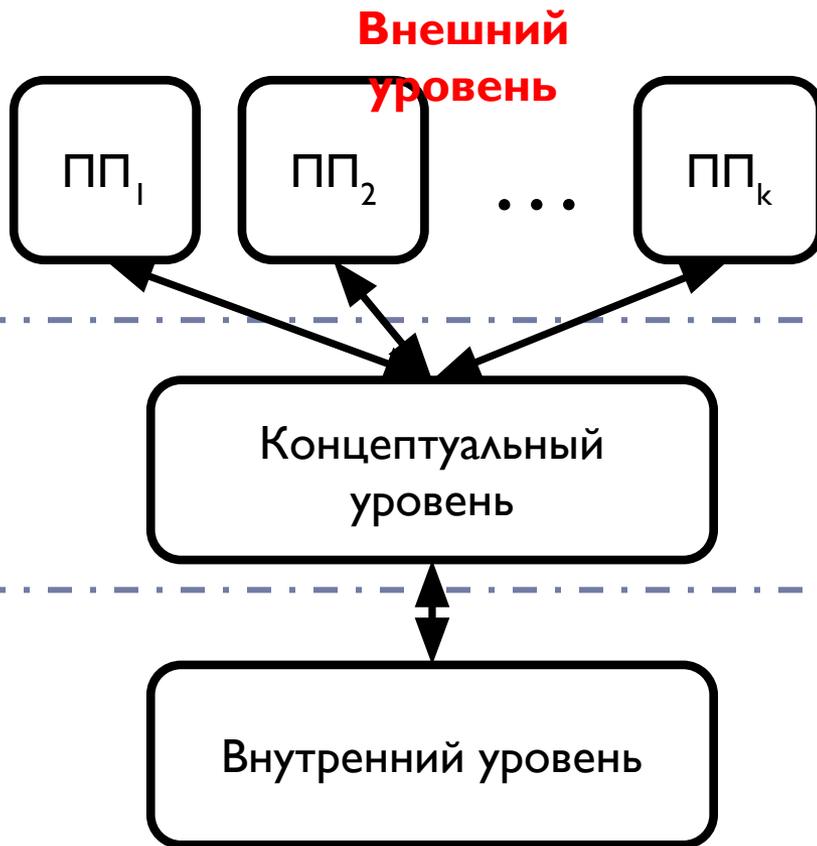
Трёхуровневая архитектура позволяет обеспечить **независимость хранимых данных** от использующих их программ.

Администратор БД (АБД) может при необходимости переписать хранимые данные на другие носители информации и реорганизовать их структуру, изменив лишь внутреннюю модель данных.

АБД может подключить к системе любое число новых пользователей (приложений), дополнив, если надо, концептуальную модель.

Указанные изменения не будут замечены существующими пользователями системы. Следовательно, независимость данных обеспечивает возможность развития системы баз данных без разрушения существующих приложений.

Трёхуровневая схема БД



На **внешнем уровне** пользователи и приложения воспринимают данные, где отдельные группы пользователей имеют свое представление пользователя (ПП) на базу данных.

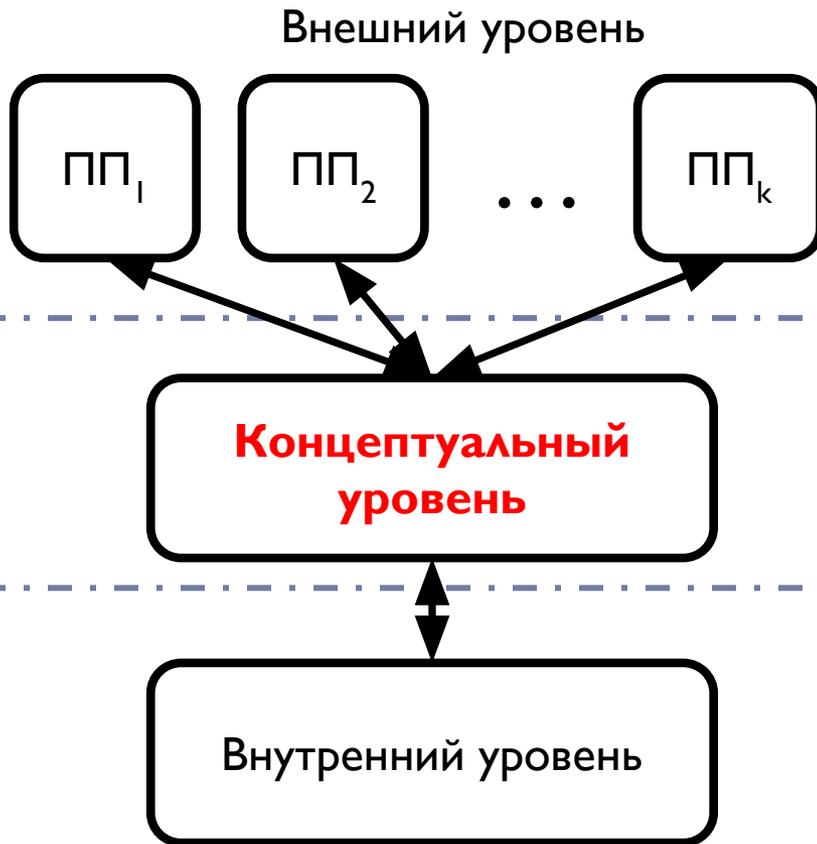
Каждый пользователь (приложение) видит и обрабатывает только те данные, которые необходимы именно этому приложению., например:

Отдел кадров - возраст, адрес

Система учета ЗП – квалификация, стаж

Каждый пользователь (приложение) может применять для работы с БД свой язык общения. Конечные пользователи употребляют либо язык запросов, либо язык приложений. Прикладные программисты чаще применяют либо языки высокого уровня, например, С, Pascal и так далее, либо специальные языки СУБД.

Трехуровневая схема БД

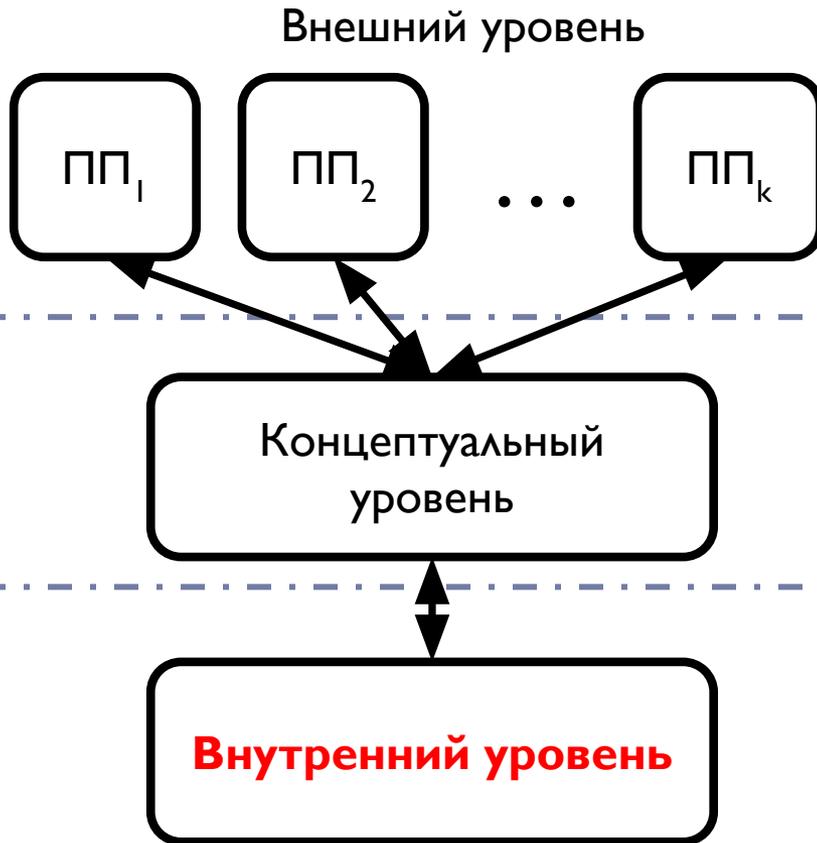


Концептуальный уровень отражает интересы всех пользователей и представляет собой единое логическое описание всех элементов данных и отношений между ними, образуя логическую структуру всей БД

Это полное представление требований к данным со стороны организации, которое не зависит от любых соображений относительно способа их хранения. На концептуальном уровне представлены следующие компоненты:

- Все сущности, их атрибуты и связи;
- Накладываемые на данные ограничения;
- Семантическая информация о данных
- Информация о мерах обеспечения безопасности и поддержки целостности данных.

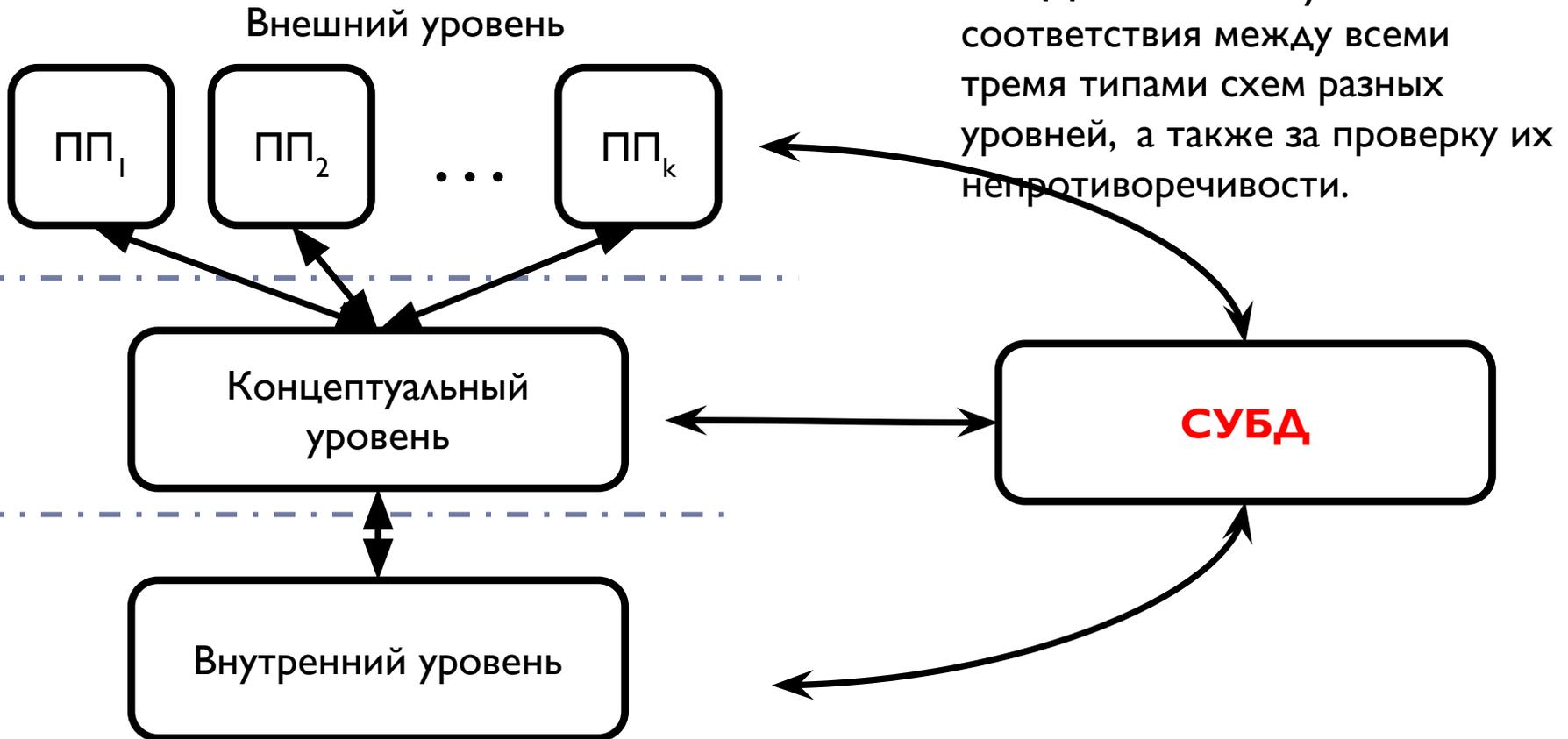
Трёхуровневая схема БД



Внутренний уровень описывает физическую реализацию и предназначен для достижения оптимальной производительности и обеспечения экономного использования дискового пространства. На внутреннем уровне хранится следующая информация:

- распределение дискового пространства для хранения данных и индексов;
- описание подробностей сохранения записей (типы, размеры элементов данных и др.);
- сведения о размещении записей;
- сведения о сжатии записей и выбранных методах шифрования.

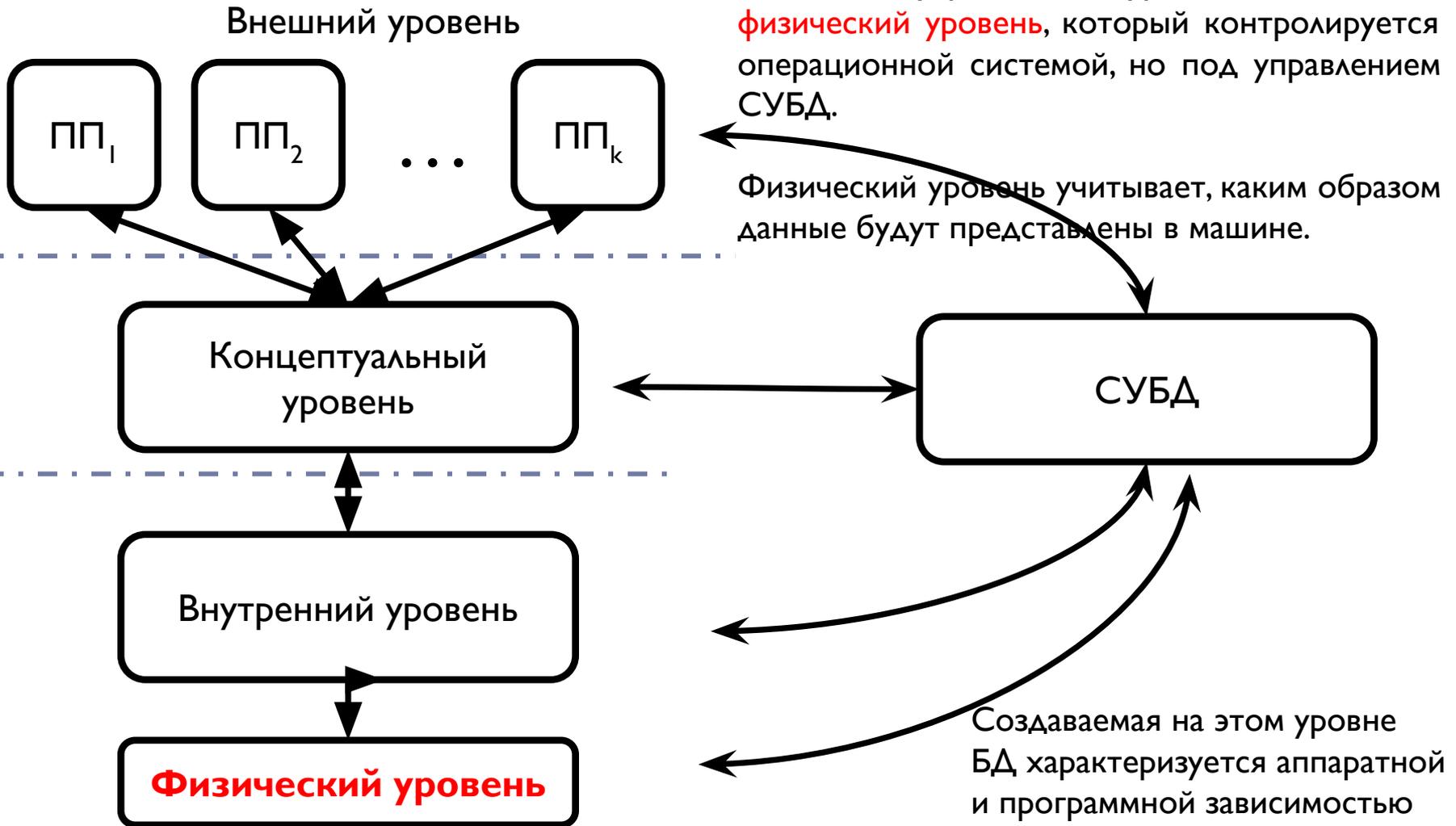
Трёхуровневая схема БД



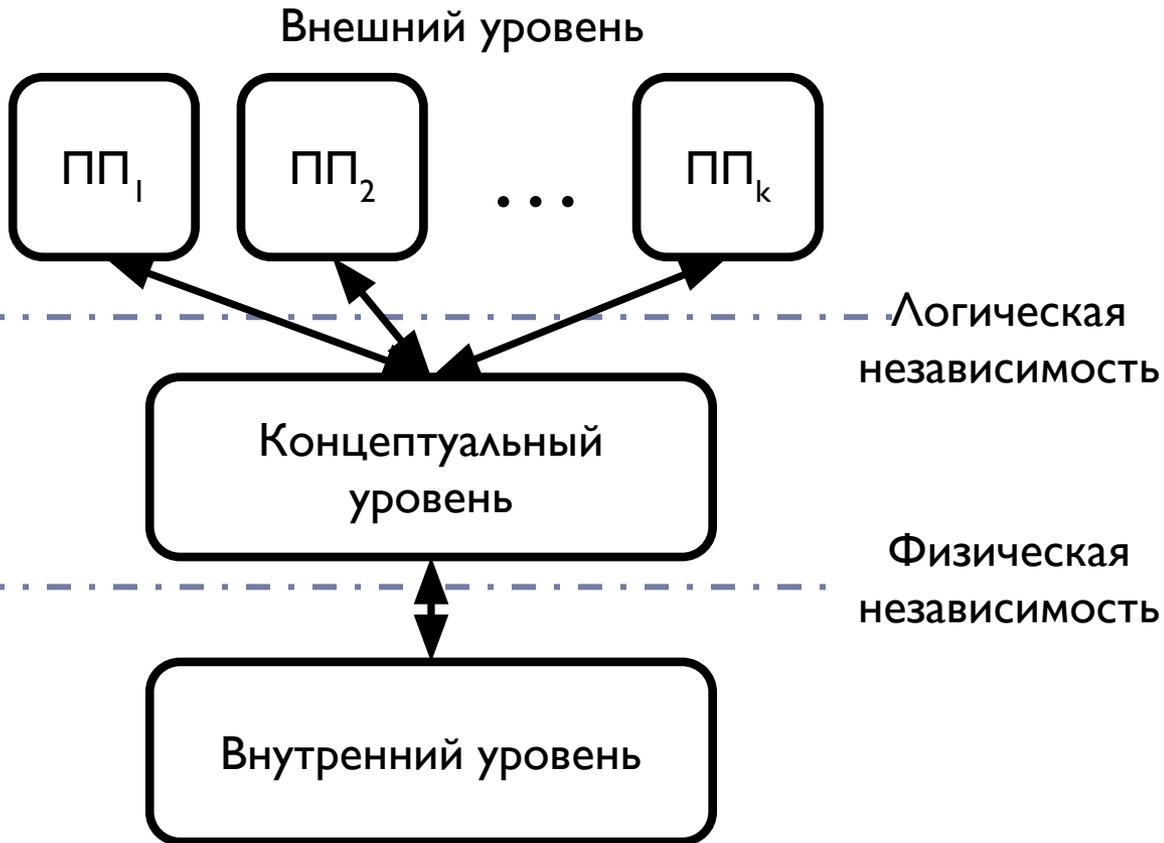
СУБД отвечает за установление соответствия между всеми тремя типами схем разных уровней, а также за проверку их непротиворечивости.

СУБД преобразовывает адреса и указатели в соответствующие логические имена и отношения и наоборот

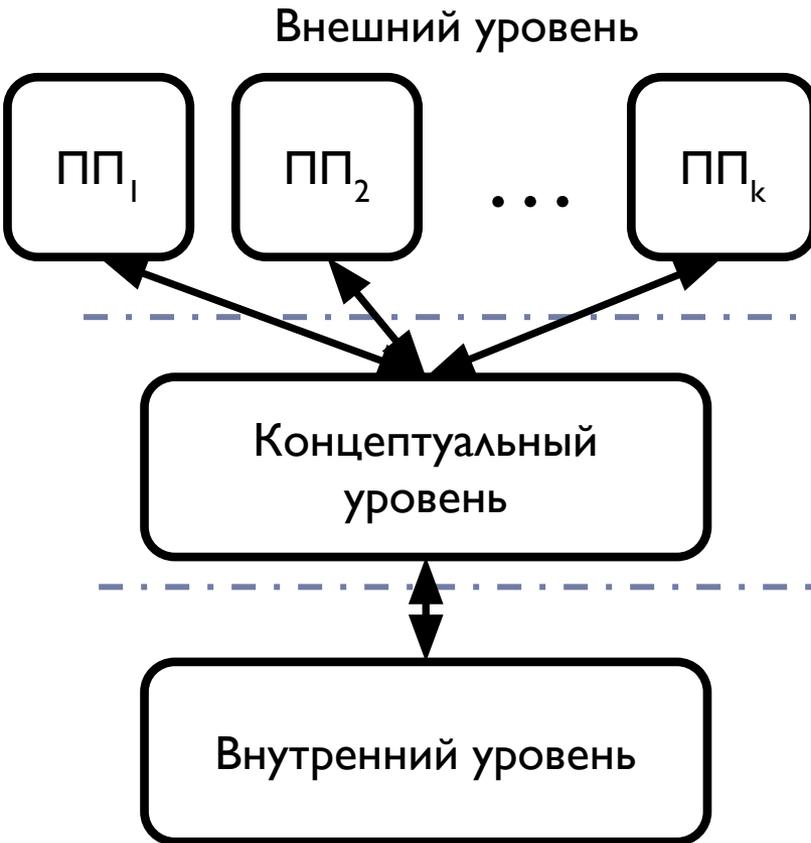
Трёхуровневая схема БД



Независимость данных



Независимость данных



Логическая независимость

ь

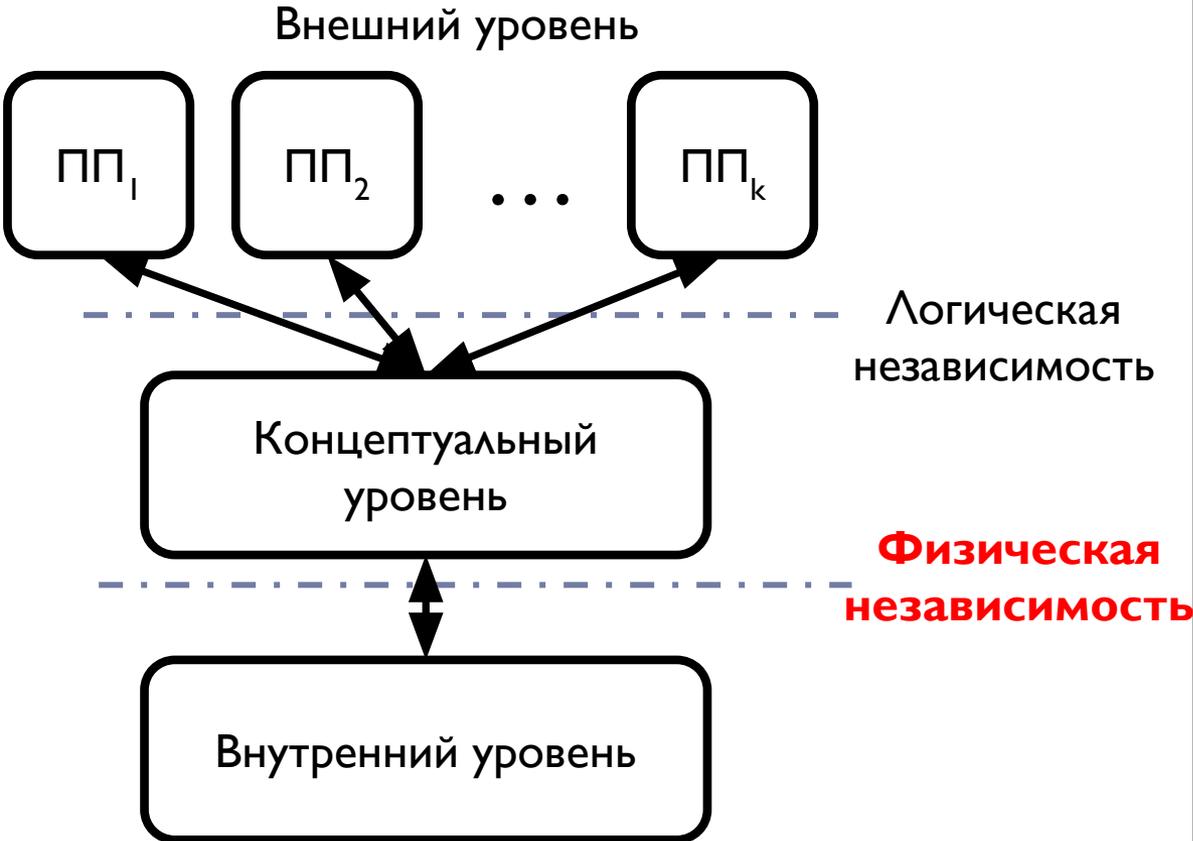
Физическая независимость

Логическая независимость

от данных означает полную защищенность внешних схем от изменений, вносимых в концептуальную схему. Такие изменения концептуальной схемы, как добавление или удаление новых сущностей, атрибутов или связей, должны осуществляться без необходимости внесения изменений в уже существующие внешние схемы для других групп пользователей. Также обеспечивается возможность изменить одно приложение без корректировки других.



Независимость данных



Физическая

независимость от данных означает защищенность концептуальной схемы от изменений, вносимых во внутреннюю схему. Такие изменения внутренней схемы, как использование различных файловых систем или структур хранения, разных устройств хранения, должны осуществляться без необходимости внесения изменений в концептуальную или внешнюю схемы. Перенос информации на другие носители сохраняет работоспособность приложений.

Лекция 2

Жизненный цикл БД

Жизненный цикл БД

Как и любой программный продукт, база данных обладает собственным жизненным циклом (ЖЦБД). Главной составляющей в жизненном цикле БД является создание единой базы данных и программ , необходимых для ее работы.



Планирование разработки БД

Содержание данного этапа — разработка стратегического плана. Планирование разработки базы данных состоит в определении объема работ, ресурсов и стоимости проекта. Важной частью разработки стратегического плана является проверка осуществимости проекта, состоящая из нескольких частей.



Планирование разработки БД

Содержание данного этапа — разработка стратегического плана. Планирование разработки базы данных состоит в определении объема работ, ресурсов и стоимости проекта. Важной частью разработки стратегического плана является проверка осуществимости проекта, состоящая из нескольких частей.

1. проверка технологической осуществимости. Она состоит в выяснении вопроса, существует ли оборудование и программное обеспечение, удовлетворяющее информационным потребностям фирмы.
2. проверка операционной осуществимости — выяснение наличия экспертов и персонала, необходимых для работы БД.
3. проверка экономической целесообразности осуществления проекта. При исследовании этой проблемы весьма важно дать оценку ряду факторов, в том числе и таким:
 - целесообразность совместного использования данных разными отделами;
 - ожидаемая выгода от внедрения подлежащих созданию приложений;
 - время окупаемости внедренной БД;
 - влияние системы управления БД на реализацию долгосрочных планов организации.



Определение требований к системе

На данном этапе необходимо определить диапазон действия приложения базы данных, состав его пользователей и области применения. Определение требований:

- Цели БД,
- информационные потребности различных структурных подразделений и их руководителей
- требования к оборудованию и требования программному обеспечению.



Сбор и анализ требований пользователей

На данном этапе необходимо создать для себя модель движения важных материальных объектов и уяснить **процесс документооборота**. По каждому документу необходимо установить периодичность использования, определить данные, необходимые для выполнения выделенных функций (анализируя существующую и планируемую документацию, выясняют, как получается каждый элемент данных, кем получается, где в дальнейшем используется, кем контролируется).



Проектирование БД

Полный цикл разработки базы данных включает концептуальное, логическое и физическое ее проектирование.



Проектирование БД



Концептуальное проектирование БД.

Или инфологическое,

Семантическое моделирование. Связано со смысловым содержанием данных, независимо от их представления в ЭВМ

На этом этапе создаются подробные модели пользовательских представлений данных предметной области. Затем они интегрируются в концептуальную модель, которая фиксирует все элементы корпоративных данных подлежащих загрузке в БД

Проектирование сложных баз данных с большим количеством атрибутов осуществляется использованием, так называемого, нисходящего подхода. Этот подход начинается с разработки моделей данных, которые содержат несколько высокоуровневых сущностей и связей, затем работа продолжается в виде серии нисходящих уточнений низкоуровневых сущностей, связей и относящихся к ним атрибутов.

Нисходящий подход демонстрируется в концепции модели "сущность — связь" (Entity-Relationship model — ER-модель) — самой популярной технологии высокоуровневого моделирования данных, предложенной П. Ченом.



Концептуальное проектирование БД.

В построении общей концептуальной модели данных выделяют ряд этапов.

1. Выделение локальных представлений, соответствующих обычно относительно независимым данным. Каждое такое представление проектируется как подзадача.
2. Формулирование сущностей, описывающих локальную предметную область проектируемой БД, и описание атрибутов, составляющих структуру каждой сущности.
3. Выделение ключевых атрибутов.
4. Спецификация связей между сущностями. Удаление избыточных связей.
5. Анализ и добавление неключевых атрибутов.
6. Объединение локальных представлений.



Проектирование БД



Логическое проектирование БД.

Или дательное
является этапом синтаксического моделирования

Цель второй фазы проектирования базы данных состоит в создании логической модели данных для исследуемой части предприятия.

На этом этапе осуществляется выбор типа модели данных. Концептуальная модель отображается в логическую, основанную уже на структурах моделей данного типа

Модель данных определяется типом предполагаемой для реализации информационной системы СУБД.

Концептуальное и логическое проектирование — это итеративные процессы, которые включают в себя ряд уточнений, продолжающиеся до тех пор, пока не будет получен наиболее соответствующий структуре предприятия продукт.



Проектирование БД



Физическое проектирование БД.

является этапом синтаксического моделирования

Определяет структуру БД в терминах языка описания данных выбранной СУБД.

Дает ответ на вопрос «Как хранить данные?»

Здесь указываются носители, методы доступа и способы защиты данных, требуемого объема памяти, правил сопровождения БД ..др.



Проектирование БД

Концептуальное

-Только человек способен построить в голове семантическую модель

Логическое

Физическое

-Создание синтаксических моделей данных можно частично автоматизировать, применив средства автоматизации проектирования - CASE-средства

Концептуальное и логическое проектирование — это итеративные процессы. Решение о возврате на требуемый этап принимает человек.



Разработка приложений

Главные составляющие данного процесса — это проектирование транзакций и пользовательского интерфейса.



Разработка приложений

Главные составляющие данного процесса — это проектирование транзакций и пользовательского интерфейса.

Проектирование транзакций

Транзакции представляют некоторые события реального мира. Транзакция может состоять из нескольких операций, однако с точки зрения пользователя эти операции представляют собой единое целое, переводящее базу данных из одного непротиворечивого состояния в другое. Реализация транзакций опирается на тот факт, что СУБД способна обеспечивать сохранность внесенных во время транзакции изменений в БД и непротиворечивость базы данных даже в случае возникновения сбоя.

Проектирование транзакций заключается в определении:

- данных, которые используются транзакцией;
 - функциональных характеристик транзакции;
 - выходных данных, формируемых транзакцией;
 - степени важности и интенсивности использования транзакции.
-



Разработка приложений

Главные составляющие данного процесса — это проектирование транзакций и пользовательского интерфейса.

Проектирование пользовательского интерфейса

Интерфейс должен быть удобным и обеспечивать все функциональные возможности, предусмотренные в спецификациях требований пользователей. Специалисты рекомендуют при проектировании пользовательского интерфейса использовать следующие основные элементы и их характеристики:

- • содержательное название;
- • ясные и понятные инструкции;
- • логически обоснованные группировки и последовательности полей;
- • визуально привлекательный вид окна формы или поля отчета;
- • легко узнаваемые названия полей;
- • визуальное выделение пространства и границ полей ввода данных;
- • средства исправления отдельных ошибочных символов и целых полей;
- • средства вывода сообщений об ошибках при вводе недопустимых значений;
- • особое выделение необязательных для ввода полей;
- • средства вывода пояснительных сообщений с описанием полей;
- • средства вывода сообщения об окончании заполнения формы.

Реализация



Реализация

На данном этапе осуществляется физическая реализация базы данных и разработанных приложений.

База данных описывается на языке определения данных выбранной СУБД. В результате компиляции его команд и их выполнения создаются схемы и **пустые файлы** базы данных.

Прикладные программы реализуются с помощью языков третьего или четвертого поколений.



Конвертирование и загрузка данных

На этом этапе созданные в соответствии со схемой базы данных пустые файлы, предназначенные для хранения информации, должны быть заполнены данными.



Тестирование

Для оценки законченности и корректности выполнения приложения базы данных может использоваться несколько различных стратегий тестирования:

- нисходящее тестирование;
- восходящее тестирование;
- тестирование потоков;
- интенсивное тестирование.



Тестирование

Нисход
предста
интерф
предста



Стратегия
возраст



ЭКСПЛУАТАЦИЯ И СОПРОВОЖДЕНИЕ

Основные действия, связанные с этим этапом, сводятся к наблюдению за созданной системой и поддержке ее нормального функционирования по окончании развертывания. Поддержка БД предполагает разрешение проблем, возникающих в процессе эксплуатации БД и связанных как с ошибками реализации БД, так и с изменениями в самой предметной области, созданием дополнительных программных компонентов или модернизацией самой БД.

