**Лекция 5-6:**

**Проектирование баз данных**

**Аннотация:**В лекции рассматриваются вопросы проектирования БД и СУБД. Подходы к проектированию СУБД. Архитектура СУБД. Методология проектирования БД. Основные этапы разработки БД.

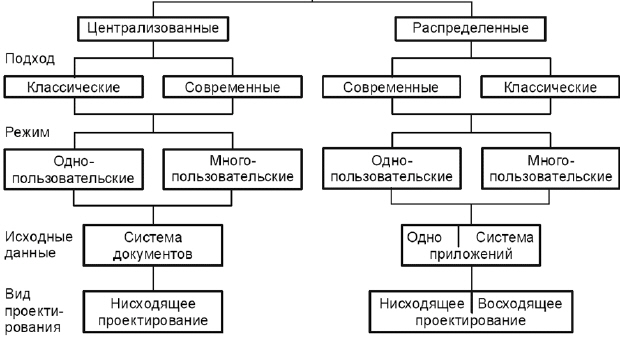
**Цель лекции:** Ознакомиться с подходами и методологией проектирования *БД* и *СУБД*. Понять архитектуру *СУБД*. Разобрать основные этапы разработки *БД*.

**Подходы к проектированию**

В конце 70-х годов появились современные *СУБД*, обеспечивающие физическую и логическую независимость, *безопасность данных*, обладающие развитыми языками *БД*. Последнее десятилетие характеризуется появлением распределенных и *объектно-ориентированных баз данных*, характеристики которых определяются приложениями *средств автоматизации* проектирования и *интеллектуализации* *БД* ([рис. 3.1](https://intuit.ru/studies/courses/3439/681/lecture/14021?page=1#image.3.1)).

Существует два подхода к построению *БД*, базирующихся на двух подходах к созданию *автоматизированной системы управления* (*АСУ*).

Первый из них, широко использовался в 80-е годы и потому получил название классического (традиционного), связано это с автоматизацией документооборота (совокупность документов, движущихся в процессе работы предприятия). Исходными и выходными координатами являлись документы.



**Рис. 3.1.**Характеристики БД

К 90-м годам сформировался второй, современный подход, связанный с автоматизацией управления. Он предполагает первоначальное выявление стандартных алгоритмов приложений (алгоритмов бизнеса в зарубежной терминологии), под которые определяются данные, а стало быть, и *база данных*. *Объектно-ориентированное программирование* только усилило *значимость* этого подхода. Состав *БД* для различных подходов представлен на [рис. 3.2](https://intuit.ru/studies/courses/3439/681/lecture/14021?page=1#image.3.2).

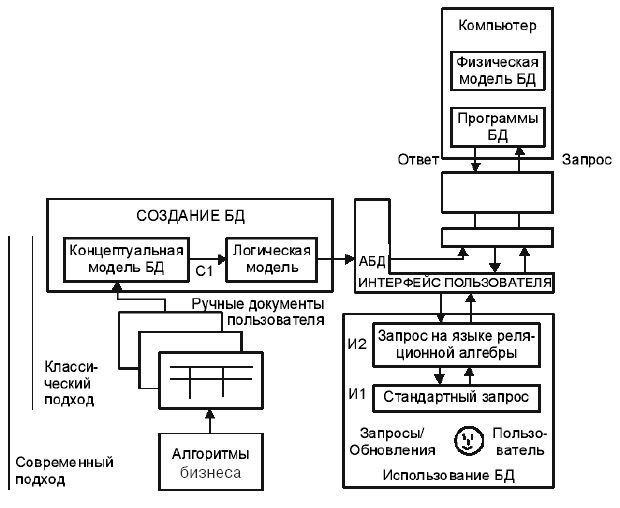


**Рис. 3.2.**Схемы (а) классического и (б) современного подходов при построении БД

В работе *БД* возможен одно- и многопользовательский (несколько пользователей подключаются к одному компьютеру через разные порты) режимы.

Используют восходящее и *нисходящее проектирование* *БД*. Первое применяют в распределенных *БД* при интеграции спроектированных локальных баз данных, которые могут быть выполнены с использованием различных моделей данных. Более характерным для централизованных *БД* является *нисходящее проектирование*.

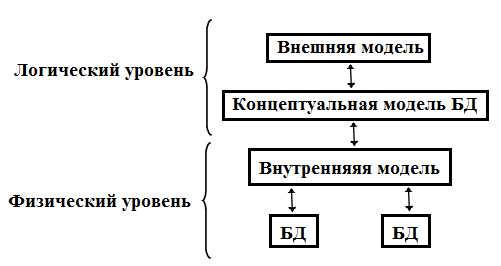
Работа с базами данных может быть представлена в виде схемы, показанной на [рис. 3.3](https://intuit.ru/studies/courses/3439/681/lecture/14021?page=1#image.3.3). Из нее видно, что следует выделять методологию создания и методологию использования *БД*. Методология *БД* определяется в процедуре проектирования, но проявляется и в процедуре использования.



**Рис. 3.3.**Схема создания использования БД

**Архитектура СУБД**

*СУБД* имеет многоуровневую структуру, в которой реализуется принцип относительной независимости *логической* и *физической* организации данных ([рис. 3.4](https://intuit.ru/studies/courses/3439/681/lecture/14021?page=1#image.3.4)).



**Рис. 3.4.**Структура СУБД

Различают *концептуальный*, *внутренний* и *внешний* *уровни представления данных* *БД*, которым соответствуют модели аналогичного назначения.

*Концептуальная модель* состоит из *множества* экземпляров различных типов данных, имеющих структуру в соответствии с требованиями *СУБД* к *логической структуре* *БД*.

*СУБД* имеет два режима работы:

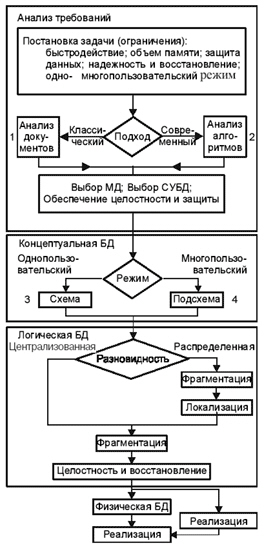
* *проектировочный* - предназначен для создания или изменения структуры базы и создания её объектов;
* *пользовательский* - использование ранее подготовленных объектов для наполнения базы или получения данных из нее.

Проектирование *БД* состоит в построении комплекса взаимосвязанных моделей данных.

**Методология проектирования баз данных**

Существует много разновидностей методологии рассмотрения баз данных в классическом подходе, однако чаще всего придерживаются методологии ANSI/SPARC, схема которой представлена на [рис. 3.5](https://intuit.ru/studies/courses/3439/681/lecture/14021?page=2#image.3.5).

На [рис. 3.5](https://intuit.ru/studies/courses/3439/681/lecture/14021?page=2#image.3.5) показана совокупность процедур проектирования централизованной *БД*, которые можно объединить в четыре этапа.



**Рис. 3.5.**Схема этапов проектирования БД

На этапе формулирования и *анализа требований* устанавливаются цели организации, определяются требования к *БД*. Они состоят из общих требований, определенных выше, и специфических требований. Для формирования специфических требований обычно используется методика интервьюирования персонала различных уровней управления. Все требования документируются в форме, доступной конечному пользователю и проектировщику *БД*.

Этап концептуального проектирования заключается в описании и синтезе информационных требований пользователей в первоначальный проект *БД*. Исходными данными могут быть совокупность документов пользователя ([рис. 3.3](https://intuit.ru/studies/courses/3439/681/lecture/14021?page=1#image.3.3)) при классическом подходе или алгоритмы приложений (алгоритмы бизнеса) при современном подходе. Результатом этого этапа является высокоуровневое *представление* (в виде системы таблиц *БД*) информационных требований пользователей на основе различных подходов.

Сначала выбирается модель *БД*. Затем с помощью ЯОД создается структура *БД*, которая заполняется данными с помощью команд ЯМД, систем *меню*, *экранных форм* или в режиме просмотра таблиц *БД*. Здесь же обеспечивается защита и *целостность* (в том числе ссылочная) данных с помощью *СУБД* или путем построения триггеров.

В процессе *логического проектирования* высокоуровневое *представление* данных преобразуется в структуру используемой *СУБД*. Основной целью этапа является устранение избыточности данных с использованием специальных правил нормализации.

Цель нормализации - минимизировать повторения данных и возможные структурные изменения *БД* при процедурах обновления. Это достигается разделением (декомпозицией) одной таблицы в две или несколько с последующим использованием при запросах *операции* навигации. Полученная *логическая структура* *БД* может быть оценена количественно с помощью различных характеристик (число обращений к логическим записям, объем данных в каждом приложении, общий объем данных). На основе этих оценок *логическая структура* может быть усовершенствована с целью достижения большей эффективности.

Специального обсуждения заслуживает процедура управления *БД*. Она наиболее проста в однопользовательском режиме. В многопользовательском режиме и в распределенных *БД* процедура сильно усложняется. При одновременном доступе нескольких пользователей без принятия специальных мер, возможно, *нарушение целостности*. Для устранения этого явления используют систему транзакций и режим блокировки таблиц или отдельных записей.

**Транзакция** - процесс изменения файла, записи или *базы данных*, вызванный передачей одного входного сообщения.

На этапе физического проектирования решаются вопросы, связанные с производительностью системы, определяются *структуры хранения* данных и методы доступа.

Взаимодействие между этапами проектирования и словарной системой необходимо рассматривать отдельно. Процедуры проектирования могут использоваться независимо в случае отсутствия словарной системы. Сама словарная система может рассматриваться как элемент автоматизации проектирования.

Средства проектирования и оценочные критерии используются на всех стадиях разработки. В настоящее время неопределенность при выборе критериев является наиболее слабым местом в проектировании *БД*. Это связано с трудностью описания и идентификации большого числа альтернативных решений.

Проще обстоит дело при работе с количественными критериями, к которым относятся время ответа на *запрос*, *стоимость* модификации, *стоимость* памяти, время на создание, *стоимость* на реорганизацию. Затруднение может вызывать противоречие критериев друг другу.

В то же время существует много *критериев оптимальности*, являющихся неизмеримыми свойствами, трудно выразимыми в количественном представлении или в виде *целевой функции*.

К качественным критериям могут относиться гибкость, адаптивность, доступность для новых пользователей, совместимость с другими системами, возможность конвертирования в другую вычислительную среду, возможность восстановления, возможность распределения и расширения.

Процесс проектирования является длительным и трудоемким и обычно продолжается несколько месяцев. Основными ресурсами проектировщика *БД* являются его собственная интуиция и *опыт*, поэтому качество решения во многих случаях может оказаться низким.

Основными причинами низкой эффективности проектируемых *БД* могут быть:

* недостаточно глубокий анализ требований (начальные этапы проектирования), включая их семантику и взаимосвязь данных;
* большая длительность процесса структурирования, делающая этот процесс утомительным и трудно выполняемым при ручной обработке.

В этих условиях вопросы автоматизации разработки становятся первостепенными.

**Основные этапы разработки БД**

**Этап 1. Уточнение задач**

На первом этапе составляется список всех основных задач, которые в принципе должны решаться этим приложением, - включая и те, которые не нужны сегодня, но могут появиться в будущем. Под "основными" задачами понимаются функции, которые должны быть представлены в формах или отчетах приложения.

**Этап 2. Последовательность выполнения задач**

Для того, чтобы приложение работало логично и удобно, лучше всего объединить основные задачи в тематические группы и затем упорядочить задачи каждой группы так, чтобы они располагались в порядке их выполнения. Может получиться так, что некоторые задачи будут связаны с разными группами или, что выполнение некоторой задачи должно предшествовать выполнению другой, принадлежащей к иной группе.

**Этап 3. Анализ данных**

После формирования списка задач, наиболее важным этапом является составление подробного перечня всех данных, необходимых для решения каждой задачи. Некоторые данные понадобятся в качестве исходных и меняться не будут. Другие данные будут проверяться и изменяться в ходе выполнения задачи. Некоторые элементы данных могут быть удалены или добавлены. И наконец, некоторые данные будут получены с помощью вычислений: их вывод будет частью задачи, но в базу данных вноситься они не будут.

**Этап 4. Определение структуры данных**

После предварительного анализа всех необходимых элементов данных нужно упорядочить их по объектам и соотнести объекты с таблицами и запросами базы данных. Для реляционных баз данных типа Access используется процесс, называемый нормализацией, в результате которого вырабатывается наиболее эффективный и гибкий способ хранения данных.

**Этап 5. Разработка макета приложения и пользовательского интерфейса**

После задания структуры таблиц приложения, в Microsoft Access легко создать его макет с помощью форм и связать их между собой, используя несложные макросы или процедуры обработки событий. Предварительный рабочий макет легко продемонстрировать заказчику и получить его одобрение еще до детальной реализации задач приложения.

**Этап 6. Создание приложения**

В случае очень простых задач созданный макет является практически законченным приложением. Однако довольно часто приходится писать процедуры, позволяющие полностью автоматизировать решение всех намеченных в проекте задач. Поэтому, понадобится создать специальные связующие формы, которые обеспечивают переход от одной задачи к другой.

**Этап 7. Тестирование и усовершенствование**

После завершения работ по отдельным компонентам приложения необходимо проверить функционирование приложения в каждом из возможных режимов. Необходимо проверить работу макросов, для этого использовав пошаговый *режим отладки*, при котором будет выполняться одна конкретная *макрокоманда*. При использовании Visual Basic для приложений в вашем распоряжении имеются разнообразные средства отладки, позволяющие проверить работу приложения, выявить и исправить ошибки.

По мере разработки автономных разделов приложения желательно передать их заказчику для проверки их функционирования и получения мнения о необходимости внесения тех или иных изменений. После того как заказчик ознакомится с работой приложения, у него практически всегда возникают дополнительные предложения по усовершенствованию, какой бы тщательной не была предварительная проработка проекта. Пользователи часто обнаруживают, что некоторые моменты, о которых в процессе постановки задач, они говорили как об очень важных и необходимых, на самом деле не играют существенной роли при практическом использовании приложения. Выявление необходимых изменений на ранних стадиях разработки приложения позволяет существенно сократить время на последующие переделки.

**Краткие итоги**

Рассмотрены вопросы проектирования *БД*. Существуют два подхода при построении *БД* классический и современный.

*Архитектура* *СУБД*. *СУБД* имеет многоуровневую структуру, в которой реализуется принцип относительной независимости *логической* и *физической* организации данных.

*СУБД* имеет два режима работы проектировочный и пользовательский.

Рассматривается совокупность процедур проектирования централизованной *БД*, которые можно объединить в четыре этапа. Этап формулирования и *анализа требований*, этап концептуального проектирования, этап *логического проектирования* и этап физического проектирования.

Основные этапы разработки *БД*. Разработка *БД* состоит из следующих этапов: уточнение задач, последовательность выполнения задач, *анализ* данных, *определение* структуры данных, разработка макета приложения и пользовательского интерфейса, создание приложения и тестирования.

**Вопросы для самопроверки**

* Назовите два подхода применяющиеся при построении БД и дайте их характеристику.
* Назовите уровни архитектуры СУБД.
* Какие режимы работы СУБД Вы знаете?
* Этапы проектирования БД.
* Кратко расскажите, что происходит на каждом этапе проектирования БД.
* Понятие транзакции.
* Основные причины низкой эффективности проектируемых БД.
* Чем отличается проектирование БД от разработки БД.
* Назовите этапы разработки БД.
* Кратко расскажите, что происходит на каждом этапе разработки БД.