

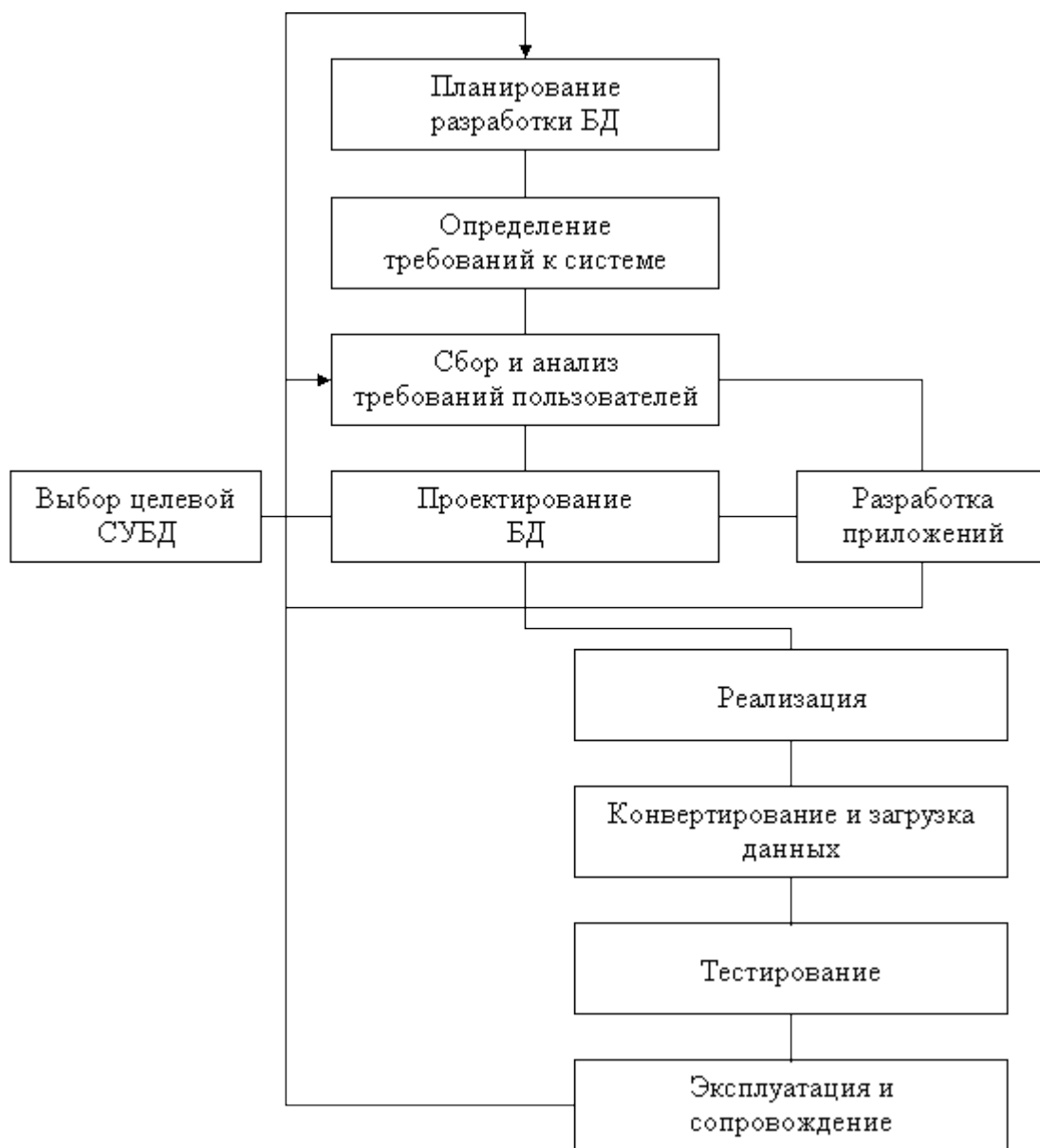
## Тема 5. Концепции проектирования БД

3.1. 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.

ЖЦБД включает в себя следующие основные этапы:

1. планирование разработки базы данных;
2. определение требований к системе;
3. сбор и анализ требований пользователей;
4. проектирование базы данных:
  - концептуальное проектирование базы данных;
  - логическое проектирование базы данных;
  - физическое проектирование базы данных;
5. разработка приложений:
  - проектирование транзакций;
  - проектирование пользовательского интерфейса;
6. реализация;
7. загрузка данных;
8. тестирование;
9. эксплуатация и сопровождение:
  - анализ функционирования и поддержка исходного варианта БД;
  - адаптация, модернизация и поддержка переработанных вариантов.

Рис. 3.1. Жизненный цикл БД



### 3.1.1. Планирование разработки базы данных

Содержание данного этапа — разработка стратегического плана, в процессе которой осуществляется предварительное планирование конкретной системы управления базами данных. Планирование разработки базы данных состоит в определении трех основных компонентов: объема работ, ресурсов и стоимости проекта.

Важной частью разработки стратегического плана является проверка осуществимости проекта, состоящая из нескольких частей.

Первая часть — проверка технологической осуществимости. Она состоит в выяснении вопроса, существует ли оборудование и программное обеспечение, удовлетворяющее информационным потребностям фирмы.

Вторая часть — проверка операционной осуществимости — выяснение наличия экспертов и персонала, необходимых для работы БД.

Третья часть — 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;
- величина риска, связанного с реализацией системы базы данных;
- ожидаемая выгода от внедрения подлежащих созданию приложений;
- время окупаемости внедренной БД;
- влияние системы управления БД на реализацию долговременных планов организации.

### 3.1.2. Определение требований к системе

На данном этапе необходимо определить диапазон действия приложения базы данных, состав его пользователей и области применения. Определение требований включает выбор целей БД, выяснение информационных потребностей различных отделов и руководителей фирмы и требований к оборудованию и программному обеспечению.

### 3.1.3. 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. Собранная информация о каждой важной области применения приложения и пользовательской группе должна включать следующие компоненты: исходную и генерируемую документацию, подробные сведения о выполняемых транзакциях, а также список требований с указанием их приоритетов. Формализация собранной на этом этапе информации может быть повышена с помощью методов составления спецификаций требований, к числу которых относятся, например, технология структурного анализа и проектирования, диаграммы потоков данных и графики "вход — процесс — выход".

### 3.1.4. Проектирование базы данных

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

Концептуальное проектирование базы данных Первая фаза процесса проектирования базы данных заключается в создании для анализируемой

части предприятия концептуальной модели данных. 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.

Модель "сущность — связь" относится к семантическим моделям. Семантическое моделирование данных, связанное со смысловым содержанием данных, независимо от их представления в ЭВМ.

В построении общей концептуальной модели данных выделяют ряд этапов.

- Выделение локальных представлений, соответствующих обычно относительно независимым данным. Каждое такое представление проектируется как подзадача.
- Формулирование сущностей, описывающих локальную предметную область проектируемой БД, и описание атрибутов, составляющих структуру каждой сущности.
- Выделение ключевых атрибутов.
- Спецификация связей между сущностями. Удаление избыточных связей.
- Анализ и добавление неключевых атрибутов.
- Объединение локальных представлений. Созданная концептуальная модель данных предприятия является источником информации для фазы логического проектирования базы данных.

### Логическое проектирование базы данных

Цель второй фазы проектирования базы данных состоит в создании логической модели данных для исследуемой части предприятия. Логическая модель, отражающая особенности представления о функционировании предприятия одновременно многих типов пользователей, называется глобальной логической моделью данных. Процесс проектирования БД должен опираться на определенную модель данных (реляционная, сетевая, иерархическая), которая 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.

### Физическое проектирование базы данных

Целью проектирования на данном этапе является создание описания СУБД ориентированной модели БД. Действия, выполняемые на этом этапе, слишком специфичны для различных моделей данных, поэтому их сложно обобщить. Остановимся на реляционной модели данных. В этом случае под физическим проектированием подразумевается:

- создание описания набора реляционных таблиц и ограничений для них на основе информации, представленной в глобальной логической модели данных;
- определение конкретных структур хранения данных и методов доступа к ним, обеспечивающих оптимальную производительность системы с базой данных;
- разработка средств защиты создаваемой системы.

### 3.1.5. Разработка приложений

Параллельно с проектированием системы базы данных выполняется 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.

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;
- согласованную терминологию и сокращения;
- согласованное использование цветов;
- визуальное выделение пространства и границ полей ввода данных;
- удобные средства перемещения курсора;
- 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.

### 3.1.6. Реализация

На данном этапе осуществляется физическая реализация базы данных и разработанных приложений, позволяющих пользователю формулировать требуемые запросы к БД и манипулировать данными в БД. База данных описывается на языке определения данных выбранной СУБД. В результате компиляции его команд и их выполнения создаются схемы и пустые файлы базы данных. На этом же этапе определяются и все специфические пользовательские представления.

Прикладные программы реализуются с помощью языков третьего или четвертого поколений. Кроме того, на этом этапе создаются другие компоненты проекта приложения — например, экраны меню, формы ввода данных и отчеты. Реализация этого, а также и более ранних этапов проектирования БД может осуществляться с помощью инструментов автоматизированного проектирования и создания программ, которые принято называть CASE-инструментами (Computer-Aided Software Engineering).

### 3.1.7. Загрузка данных

На этом этапе созданные в соответствии со схемой базы данных пустые файлы, предназначенные для хранения информации, должны быть заполнены данными. Наполнение базы данных может протекать поразному, в зависимости от того, создается ли база данных вновь или новая база данных предназначена для замены старой.

### 3.1.8. Тестирование

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

нисходящее тестирование;  восходящее тестирование;  тестирование потоков;  интенсивное тестирование. Нисходящее тестирование начинается на уровне подсистем с модулями, которые представлены заглушками, т. е. простыми компонентами, имеющими такой же интерфейс, как модуль, но без функционального кода. Каждый модуль низкого уровня представляется заглушкой. Постепенно все программные компоненты заменяются фактическим кодом и после каждой замены снова тестируются. Восходящее тестирование выполняется в противоположном направлении по отношению к нисходящему. Оно начинается с тестирования модулей на самых низких уровнях иерархии системы, продолжается на более высоких уровнях и заканчивается на самом высоком уровне. Тестирование потоков осуществляется при тестировании работающих в реальном масштабе времени систем, которые обычно состоят из большого количества взаимодействующих процессов, управляемых с помощью прерываний. Стратегия тестирования потоков направлена на слежение за отдельными процессами. Стратегия интенсивного тестирования часто включает серию тестов с постепенно возрастающей нагрузкой и продолжается до тех пор, пока система не выйдет из строя.

### 3.1.9. 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 компонент или модернизацией самой БД.