**ЛЕКЦИИ ПО ТЕОРИИ БАЗ ДАННЫХ**

[**ЛЕКЦИЯ №1. БАЗЫ ДАННЫХ В СТРУКТУРЕ ИНФОРМАЦИОННЫХ СИСТЕМ** 2](#_Toc120524927)

[**ЛЕКЦИЯ №2. СИСТЕМЫ. ХАРАКТЕРИСТИКИ. СВОЙСТВА** 12](#_Toc120524928)

[**ЛЕКЦИЯ №3. РАЗРАБОТКА МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ.** 15](#_Toc120524929)

[**ЛЕКЦИЯ №4-5. РАЗРАБОТКА МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ ТИПА «СУЩНОСТЬ-СВЯЗЬ»** 17](#_Toc120524930)

[**ЛЕКЦИЯ №6-7. РАЗРАБОТКА РЕЛЯЦИОННОЙ МОДЕЛИ ДАННЫХ** 24](#_Toc120524931)

[**ЛЕКЦИЯ №8. СОВРЕМЕННЫЕ ПРОБЛЕМЫ ЭКСПЛУАТАЦИИ И РАЗВИТИЯ ТЕХНОЛОГИИ БАЗ ДАННЫХ** 37](#_Toc120524932)

# **ЛЕКЦИЯ №1. БАЗЫ ДАННЫХ В СТРУКТУРЕ ИНФОРМАЦИОННЫХ СИСТЕМ**

***Информационная система*** – взаимосвязанная совокупность определенной идеологии работы с информацией, методов , технологий, технических средств, используемых для сбора, хранения и выдачи информации потребителю в интересах достижения поставленной цели.

***Информационный процесс*** – процесс сбора, обработки, накполения, хранения, поиска и распространения информации (ФЗ 149)

***Информационный процесс*** – совокупность последовательных действий (операций), производимых над информацией (в виде данных, сведений, фактов, идей, гипотез, торий и пр.) для получения какого-либо результата(достижения цели)

**Типы информационного процесса**

|  |  |
| --- | --- |
| Общий информационный ресурс | Сбор |
| Преобразование |
| Использование информации |
| Основной информционный ресурс | Поиск |
| Отбор |
| Хранение |
| Передача |
| Обработка |
| Защита информации |

**Структура ИС**

* ***Функциональные подсистемы*** – составляют содержательную часть ИС, в функциональных подсистемах сосредотачиваются задачи, реализующие конкретные функции системы управления
* ***Обеспечивающие подсистемы*** – создают условия для работы функциональных подсистем

Виды обеспечивающих подсистем:

* + Информ. Обесспечение
  + Тех. Обеспечение
  + Программное обеспечение
  + Организационно-правовое обеспечение

**Информационное обеспечение**

***Целью информационного обеспечения*** яв-ся своевременнаая выдача необходимой и достоверной информации для выработки и принятия управленческих решений

**К основным задачам ИО** отн-ся:

* Классификация, формировани оптимального перечня и состава информации, ее кодирование
* Разработка классификаторов
* Создание информ. модели объекта автоматизации
* Формировнаие информационной справочно-нормативной базы управления

**Состав ИО:**

* Состав инфы
* Структура инфы и закономерности ее преобразования
* Характеристики движения информации
* Характеристики качества инфы
* Способы преобразования инфы

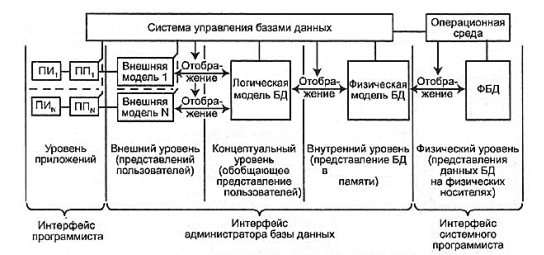
**Структура ИО:**

* Внемашинное ИО
* Внутримашинное ИО

**Определение БД**

Под ***базой данных*** понимается «совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимая от прикладных программ. Эти данные относятся к определенной предметной области и организованы таким образом, что могут быть использованы для решения многих задач многими пользователями»

***БД*** – структурированный органимзованный набор данных, объединенных в соответствии с некоторой выбранной моделью и описывающих характеристики какой–либо физической или виртуальной системы.



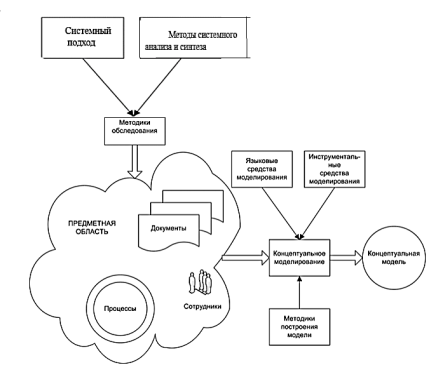
**Предметная область**

***Предметная область*** какой-либо деятельности – часть реального мира, подлежащая изучению с целью организации управления процессами и объектами для получения результата

Ккаждый фрагмент предметной области характеризуется множеством объектов и процессов, использующих объекты, а также множеством поользователей , характеризщуемых различными взгляжами на предметную область и данными, которые описывают указанные составляющие предметной области . Эти данные отражают динамичную внешнюю и внутреннюю среды поедприятия, поэтому в сппециальных разделах ИС необхожимо создаватб динамически обновляемые модели отражения внешнего мира с использованием единого хранилища – ***базы данных***

***БД*** – яв-ся информ. моделью предметной области и к ней предъявляются требования адекватности (отражение основных свойств и характеристик процессов, релизуемых в предметной области) и актуальности (обновление БД в соотвествии с изменением состояния ПрО)

***БД*** – некоторая целевая модель ПрО, т.е. в базе данных находят отражение только те факты о ПО, которые необходимы для функционирования автоматизированной системы, в состав которой входит БД. При приоектировании БД проектировщик должен выделить и описать эти ожидаемые факты, тем самым определяются границы ПО, заетм необхожимо выпрлнить интерпретацию описание этих фактов с помощью жопустимых конкретной СУБД струтур данных.



**Понятие модели данных**

Под ***данными*** понимается инфа, фиксированная в определнной форме, пригодной для последующей обработки, хранения и передачи.

Понятие ***данные в концепции БД*** – набор конкретных значений, параметров, характеризующих объект, условие, ситуацию или любые др. факторы состояния предметной области

Основная ***задача построения моделей данных*** – адекватное описание предметной области для последующего перевода на язык информационных систем

***Модель данных*** – некоторая абстракция, которая, будучи приложима к конкретным данным, позволяет пользователям и разработчикам трактовать их как сведения, содержащие не только данные, но и взаимосвязь между ними.

**Инфологическая модель**

***Инфологическая модель*** яв-ся объединением частных представлений о содержимом БД, полученных в результате опроса пользователей и своих представления о данных, которые могут потребоваться в будущих приложениях и отображает реальный мир в некоторые, понятные человеку концепции, полностью независымые от параметров среды хранения данных

***Цель*** – обеспечение наиболее естественных для человека способов сбора и представления той инфы., которую предполагается хранить в создаваемой БД. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком и является семантической моделью.

***Выражает*** инфу о ПрО в виде, независимом от используемой СУБД

**Даталогическая модель**

Под ***даталогической понимается модель***, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации

Разрабатывается с конкертным учетом конкретной реализации СУБД, также с учетом специфики конкретной предметной области на основе ее инфологической модели

**Классификация:**

* Докуменальные модели
* Фактографические модели
  + Теоретико-графовые: Иерархические; Сетевые
  + Теоретико-мнодественные: реляционные; бинарных отношений
  + Объектно-ориентированные

**Основные операции над данными**:

* Идентификация одного данного и нахождения его положения в БД
* Выборка (чтение) из БД
* Включение (запись) в БД
* Удаление из БД
* Модификация(изменение) в БД

Операции над данными отражают динамические свойства модели данных

Операции над данными должны соотносится с ЯМД СУБД

Логические ограничения, которые накладываются на данные – ***ограничение целостности***.

**Физические модели баз данных**

***Физические модели БД*** определяют способы размещения данных в среде хранения и способы доступа к этим данным, которые поддерживаются на физическом уровне.

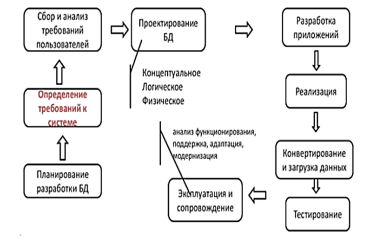
Физическая модель данных оперирует категориями, касающимисяорганизации внешней памяти и структур хранения, используемых в данной операционной среде

Для каждого файла в файловой системе хранится:

* имя файла
* тип файла
* размер записи
* количество занятых физических блоков
* базовый начальный адрес
* ссылка на сегмент расширения
* способ доступа (код защиты)

**Жизненный цикл БД:**

* предварительное планирование
* проверка осуществимости
* определение требований
* концептуальное проектирование
* реализация
* оценка работы и поддержка БД
* снятие с эксплуатации





**СУБД**

***СУБД*** – специализированная программа или комплекс программ, предназначенные для маннипулирования БД

**Основные функции СУБД**:

* управление данными в оперативной памяти
* управление данными во внешней памяти
* журналирование изменений и восстановление БД после сбоев
* контроль пользовательских запросов и выполнение правил бкзопасности и целостности
* поддрежка языкорв БД

**Классификация:**

***По типу управляемой БД***:

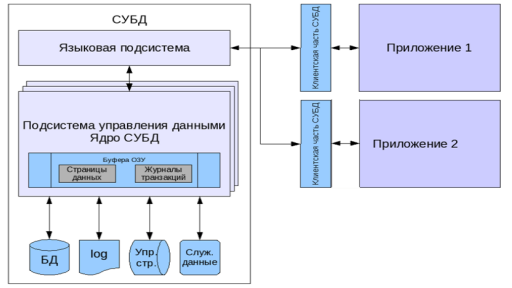
* сетевые
* иерархические
* реляционные
* объектно-ориентиованные
* объектно-реляционные

***По архитектуре организации хранения данных:***

* локальные СУБД(все части СУБД на одном компьютере)
* распределнные СУБД (части СУБД могут размещаться на 2х и более компьютерах)

***По способу доступа к БД:***

* файл-серверные
* клиент-серверные
* трехзвенные
* встраиваемые

****

**Функции админа БД**

***Админ БД*** – лицо (или группа лиц), на которое возложено управление средствами БД организации

* координация всех действий по проектированию, реализации и ведению БД
* анализ существующих программных средств и возможности их испоьзования в БД, разработка программно-технических мероприятий по развитию БД
* разработка и реализация мер по обеспечению защиты данных от некомпетентного их использования, от сбоев технических средств
* выполнение работ по ведению словаря данных
* обеспечение удовлетворения БД требованиям по производительности
* изменение необходимости методов хранения данных, путей доступа к ним,связей между данными, форматов данных
* координация работы технического обеспечения системы аппаратными средствами исходя из требований, предъявляемых БД у оборудованию
* координация работы системных программистов
* координация работы прикладных программистов
* работа с конечными пользователями

# **ЛЕКЦИЯ №2. СИСТЕМЫ. ХАРАКТЕРИСТИКИ. СВОЙСТВА**

**Общее определение системы**

***Система*** – множество составляющих единство элементов, связей и взаимодействий между ними и внешней средой, образующие присущую данной системе целостность, качественную определнность и целенаправленность.

***Системный анализ*** – совокупность понятий, методов, процедур и технологий для изучения, описания, реаллизации явлений и процессов различной природы и характера, междисциплинарных проблем.

Это совокупность общих законов, методов, приемов исследования систем.

**Элемент системы**

***Элемент* –** это составная часть сложного целого.

*Сложное целое* – это система, которая представляет собой целостный комплекс взаимосвязанных элементов.

***Элемент*** – часть системы, обладающая самостоятельностью по отношению ко всей системем неделимая при данном способе выделения частей.

*Неделимость* элемента рассматривается как нецелесообразность учета в пределах модели данной системы его внутреннего строения

Сам элемент харктеризуется только его внешними проявлениями в виде связей и взаимосвязей с остальными элементами и внешней средой.

Состояние элемента, в зависимости от различных факторов (времени, пространства, внешней среды и т.д.), может изменяться.

Последовательные изменения состояния элементабудем называть ***движением элемента.***

**Связи**

***Связь*** – совокупность зависимостей свойств одного элемента от свойств других элементов системы.

***Установить связь между двумя элементами*** – это значит выявить наличие зависимостей их свойств.

Зависимость свойств элементов может иметь односторонний и двусторонний характер.

***Взаимосвязи*** – совокупность двухсторонних зависимостей свойств одного элемента от свойств других элементов системы.

***Взаимодействие*** – совокупность взаимосвязей и взаимоотношений между свойствами элементов, когда они приобретают характер взаимосодействия друг другу.

**Внешняя среда**

***Внешняя среда*** – набор существующих в пространстве и во времени объектов (систем), которые как предполагается, оказывают действие на систему.

Для данной системы внешняя среда (окружение) есть множество предметов вне системы:

1. изменение признаков которых влияет на систему;
2. признаки которых изменяются вследствие поведения системы

Решение задачи отнесения предметов к самой системе или к ее окружению является в значительной мере произвольной и *зависит от целей изучения системы*. Для того, чтобы указать окружение полностью, необходимо все факторы, воздействующие на систему или испытывающие воздействие с ее стороны.

При определении границ системы и ее окружения часто используют *метод абстрагирования или идеализации*. При использовании этого метода в систему и ее окружение включают те предметы, которые кажутся наиболее важным, описывают связи между ними возможно точнее и исследуют наиболее интересные признаки, пренебрегая теми, которые не играют

# **ЛЕКЦИЯ №3. РАЗРАБОТКА МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ.**

***Модель типа «сущность – связь»*** – неформальная модель предметной области, которая используется на этапе инфологического проектирования БД.

***Основные назначения этой модели*** – семантическое описание предметной области и представление информации для обоснования выбора видов моделей и структур данных.

Модель разрабатывается на *естественном языке*.

При построении модели используется **3 основных конструктивных элементов**:

* ***Сущность*** – собирательное понятие, некоторая абстракция реально существующего объекта, процесса или явления, о котором необходимо хранить информацию в системе.
* ***Атрибут*** –поименованная характеристика сущности, принимаемая значения некоторого множества значений.
* ***Связь***

Информация объединяется с помощью графических диаграмм.

***Тип сущности*** определяет набор однородных элементов.

***Экземпляр сущности*** – конкретный объект в наборе.

*Каждый тип сущности должен иметь имя*;

Атрибуты выявляются на основе реквизитов, относящихся к рассматриваемому объекту указанных во множестве документов предметной области.

***В модели атрибут*** – средство, с помощью которого моделируется свойство сущностей.

Атрибут (или набор), однозначно идентифицирующий экземпляр сущности – первичный ключ.

**Классификация**:

* ***Сильная (regular)*** – существует сама по себе;
* ***Слабая (weak)*** – существует в зависимости от сильной;
* ***Независимая*** – идентифицируется без определения его отношений с другими сущностями;
* ***Зависимая*** – однозначная идентификация экземпляра сущности зависит от ее отношения с экземпляром другой сущности.

**Классификация атрибутов**:

* Идентифицируемое/относительное
* Составное/простое
* Однозначное/многозначное
* Основные/производные
* Обязательные/необязательные

***Первичный ключ*** – атрибут ил набор атрибутов, однозначно идентифицирующие экземпляр сущности.

***Альтернативный ключ*** – атрибут или набор атрибутов, которые однозначно идентифицируют экземпляр сущности, но не выбранные в качестве первичного ключа.

# **ЛЕКЦИЯ №4-5. РАЗРАБОТКА МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ ТИПА «СУЩНОСТЬ-СВЯЗЬ»**

***Связи*** – выступают в модели в качестве средства, с помощью которого представляют отношения между сущностями, имеющими место в ПрО.

Тип связи рассматривается между типами сущностей.

Существуют следующие виды бинарных связей:

* бинарные
* тернарные
* n-арные.

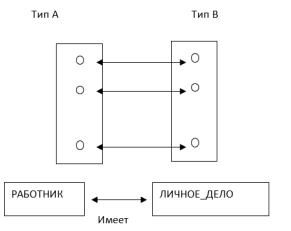
Классификация бинарных связей:

* Отображение 1:1
* Отображение 1:М
* Отображение М:1
* Отображение М:М

**Отображение 1:1 (Связь один к одному)**

С помощью отображения 1:1 определяют такой тип связи между типами сущностей А и В, когда каждому экземпляру сущности А соответствует один и только один экземпляр сущности В и наоборот.

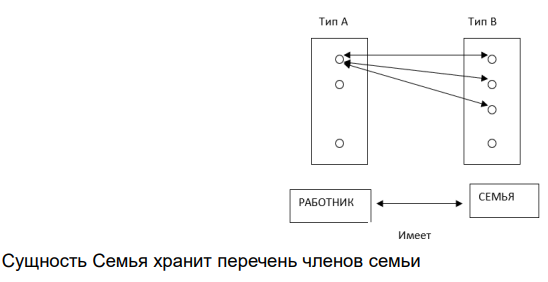
Т.е. А однозначно идентифицирует В и, наоборот, В однозначно идентифицирует А.



**Отображение 1:М (Связь один ко многим)**

С помощью отображения 1:М определяют такой тип связи между типами сущностей А и В, когда одному экземпляру сущности А соответствует несколько или ни одного экземпляра сущности В, а каждому экземпляру сущности В соответствует только один экземпляр А.

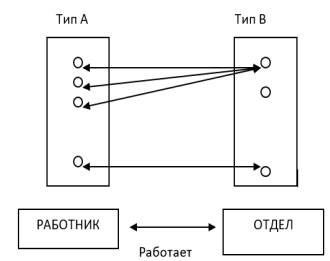
Связь можно охарактеризовать как «имеет в составе».



**Отображение М:1 (Связь многие к одному)**

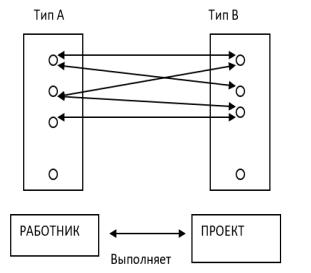
Связь обратная 1:М.

Характеризуется как «входит в состав»



**Отображение М:М (Связь многие ко многим)**

С помощью отображения М:М определяют такой тип связи между типами сущностей А и В, когда каждому экземпляру сущности А соответствует несколько или ни одного экземпляра сущности В и наоборот.



**Модальность связи**

Связь характеризуется свойством модальности:

* Может
* Должен.

Если экземпляр одной сущности может быть связан с одним или несколькими экземплярами другой сущности, а может быть и не связан ни с одним экземпляром связь имеет модальность ***может***.

Если экземпляр одной сущности всегда связан не менее чем с одним экземпляром другой сущности связь имеет модальность ***должен***.

Связь может иметь разную модальность с разных концов.

Тип связи в совокупности с характеристикой модальности называется ***кардинальностью связи***.

**Класс принадлежности сущности**

Через модальность связи определяется класс принадлежности сущности.

Если экземпляры данной сущности должны участвовать в связи, то класс принадлежности сущности называется ***обязательным*** (*Mandatory*).

Если экземпляры данной сущности могут не участвовать в связи, то класс принадлежности сущности называется ***необязательным*** (*Optional*).

**Нормализация схемы сущности**

Нормализации подлежат схемы сущностей, т.е. нормализуется перечень атрибутов. Нормализация выполняется путем выделения в отдельную сущность аномальных атрибутов.

***Первая нормальная форма:*** устраняются повторяющиеся атрибуты или группы атрибутов, т.е. производится выявление неявных сущностей, «замаскированных» под атрибуты.

***Вторая нормальная форма:*** устраняются атрибуты, зависящие только от части уникального идентификатора.

***Третья нормальная форма:*** устраняются атрибуты, зависящие от атрибутов, не входящих в уникальный идентификатор (первичный ключ).

**Первая нормальная форма**

❖Невозможно использование в виде атрибута массива данных

❖Частично структурированные атрибуты не являются аномалией по первой нормальной форме.

При анализе алгоритмов манипулирования данными возможны неструктурированные атрибуты Адрес, Паспорт и т.п.

**Вторая нормальная форма**

Сущность: Знание языков

Атрибуты: Табельный номер, Код ОКИН Языка, Наименование языка, Код ОКИН Знания ин.яз, Наименование степени знания

Первичный ключ: Табельный номер, Код ОКИН Языка

Наименование языка зависит от элемента первичного ключа

Код ОКИН Языка – это аномалия по 2-ой НФ.

***Нормализация:***

Сущность: ОКИН языки

Атрибуты: код ОКИН языка, Наименование языка

Первичный ключ: код ОКИН языка

Сущность: Знание языков

Атрибуты: Табельный номер, Код ОКИН Языка, Код ОКИН Знания ин.яз, Наименование степени знания

Первичный ключ: Табельный номер, Код ОКИН Языка

**Третья нормальная форма**

Сущность: Знание языков

Атрибуты: Табельный номер, Код ОКИН Языка, Код ОКИН Знания ин.яз,

Наименование степени знания

Первичный ключ: Табельный номер, Код ОКИН Языка

Наименование степени знания зависит от неосновного атрибута Код

ОКИН Знания ин.яз – это аномалия по 3-ой НФ.

***Нормализация:***

Сущность: ОКИН Степени знания ин. яз

Атрибуты: код ОКИН, Наименование степени знания

Первичный ключ: код ОКИН Степени знания ин. Яз

Сущность: Знание языков

Атрибуты: Табельный номер, Код ОКИН Языка, Код ОКИН Знания ин.яз

Первичный ключ: Табельный номер, Код ОКИН Языка

**Этапы разработки модели предметной области «Сущность-Связь»**

1. Определение сущностей, их атрибутов, а также первичных

и альтернативных ключей.

2. Определение связей между сущностями и указание видов.

3. Определение классов принадлежности сущностей.

4. Нормализация схем сущностей

**Оформление модели предметной области типа «Сущность - Связь»**

Спецификация сущностей:

❖Имя сущности

❖Назначение сущности

Спецификация атрибутов сущности:

❖Имя атрибута

❖Назначение атрибута

❖Тип атрибута

❖Формат атрибута

❖Допустимые значения

❖Примечание – отнесение атрибута к первичному или альтернативному (возможному) и другие замечания

Спецификация связей (для каждой бинарной связи):

❖Сущности участвующие в связи

❖Кардинальность связи

❖Классы принадлежности сущностей в бинарной связи

# **ЛЕКЦИЯ №6-7. РАЗРАБОТКА РЕЛЯЦИОННОЙ МОДЕЛИ ДАННЫХ**

**Таблица реляционного отношения:**

Таблица удовлетворяющая требованиям

* Каждый элемент таблицы представляет собой элемент данных, повторяющихся группы отсутствующих
* Все столбцы в таблицы однородны
* Столбцам присвоены имена
* В таблице нет двух одинаковых строк
* В операциях с таблицами ее строки и столбцы могут просматриваться в любом порядке и в любой последовательности безотносительно к их информационному содержанию

Называется ***реляционным* *отношением*.**

**Реляционные отношения в БД**

Отношение принимает вил таблицы или файла, где кортежи – строки таблицы или записи в файле.

Имена столбцов называются ***атрибутами***, а индивидуальное значение, появляющееся в отдельных кортежах – ***значением атрибута***.

Число столбцов в отношении называется ***степенью***.

Число кортежей – ***мощностью***.

Реляционная БД определяется как совокупность отношений, содержащих всю инфу хранимую в БД.

Внешний ключ реляционного отношения

Для связей между отношениями используются внешние ключи.

***Внешний ключ (foreign key)*** – это атрибут подчиненного(дочернего) отношения, который является копией первичного (primary key) или уникального (unique) ключа родительского отношения.

Фактически внешние ключи логически связывают экземпляры сущностей разных типов (родительской и подчиненной сущностей).

***Внешний ключ*** – это ограничение целостности, в соответствии с которым множество значений внешнего ключа является подмножеством значений первичного или уникального ключа родительской таблицы.

**Ограничение целостности по внешнему ключу**

Ограничение целостности по внешнему ключу проверяется в двух случаях:

* При добавлении записи в подчиненную таблицу СУБД проверяет, что в родительской таблице есть запись с таким же значением первичного ключа
* При удалении записи из родительского таблицы СУБД проверяет, что в подчиненной таблице нет записей с таким же значением внешнего ключа.

Ограничения целостности должны поддерживаться СУБД.

Для соблюдения целостности сущностей достаточно гарантировать отсутствие в отношении кортежей с одним и тем же значением первичного ключа.

**Ссылочная целостность**

Необходимо следить за тем, чтобы не появлялись некорректные значения внешних ключей при обновлении отношений (например, заказы несуществующих клиентов).

При удалении кортежа существует три подхода, позволяющие поддерживать ссылочную целостность:

* Запрещается производить удаление кортежа, на который существуют ссылки (либо сначала удалить ссылающиеся кортежи, либо изменить значения их внешнего ключа)
* При удалении кортежа, на который имеются ссылки, во всех ссылающихся кортежах значение внешнего ключа становится неопределенным
* При удалении кортежа из отношения, на которое ведется ссылка из ссылающегося отношения автоматически удаляются все ссылающиеся кортежи (каскадное удаление)

**Правило 1 (преобразования реляционных отношений и привидения их к нормальной норме)**

Если степень бинарной связи равна **1:1** и класс принадлежности **обеих** сущностей является **обязательным**, то требуется только ***одно*** отношение, объединяющее атрибуты двух сущностей. Первичным ключом этого отношения может быть ключ любой из двух сущностей.

При реализации этого правила закладывается аномалия в схеме реляционного отношения по третьей нормальной форме. Такие реляционные отношения должны быть более внимательно проанализированы на этапе нормализации

**Правило 2 (преобразования реляционных отношений и привидения их к нормальной норме)**

Если степень бинарной связи равна **1:1** и класс принадлежности **одной** сущности является **обязательным**, а другой **необязательным**, то необходимо построение ***двух*** отношений. Под каждую сущность необходимо выделить по одному отношению, при этом ключ сущности должен служить первичным ключом для соответствующего отношения. Ключ сущности, для которой класс принадлежности является необязательным, добавляется в качестве атрибута в отношение, выделенное для сущности с обязательным классом принадлежности и описывается как ***внешний ключ.***

**Правило 3 (преобразования реляционных отношений и привидения их к нормальной норме)**

Если степень бинарной связи равна **1:1** и класс принадлежности **обеих** сущностей является **необязательным**, то необходимо построение ***трех*** отношений: по одному для каждой сущности, ключи которых служат в качестве первичных ключей соответствующих отношений, и одно для связи. Среди своих атрибутов отношение, выделенное для связи, будет иметь ключи обеих сущностей, также они будут специфицироваться как внешние ключи. Первичным ключом третьего отношения может быть ключ любой из двух сущностей.

**Правило 4 (преобразования реляционных отношений и привидения их к нормальной норме)**

При построении отношений для связи 1:М учитывается класс принадлежности М-связанной сущности. Класс принадлежности 1-связаянной сущности наконечный результат не влияет.

Если степень бинарной связи **1:М** и класс принадлежности **М-связанной** сущности является **обязательным**, то достаточным является использование **двух** отношений, по одному на каждую сущность. При условии, что ключ каждой сущности служит в качестве первичного ключа для соответствующего отношения. Дополнительно ключ 1-связанной сущности должен быть добавлен как атрибут в отношение, отведенной для М-связанной сущности в качестве внешнего ключа.

**Правило 5 (преобразования реляционных отношений и привидения их к нормальной норме)**

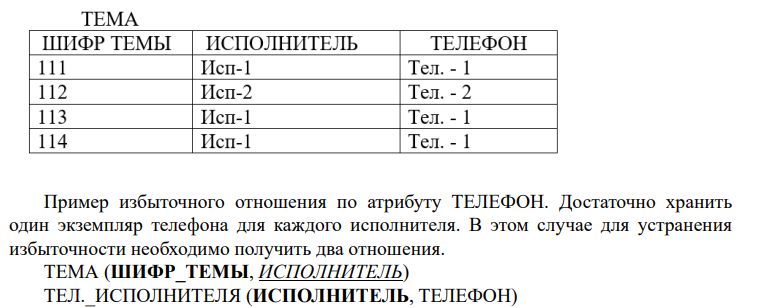
Если степень бинарной связи **1:М** и класс принадлежности **М-связанной** сущности является **необязательным**, то достаточным является использование **трех** отношений, по одному на каждую сущность, причем ключ каждой сущности служит первичным ключом соответствующего отношения, и одного отношения для связи. Отношение для связи должно иметь среди своих атрибутов первичный ключ каждой сущности (внешние ключи). Первичным ключом третьего отношения выбирается первичным ключом **М-связанной** сущности.

**Правило 6 (преобразования реляционных отношений и привидения их к нормальной норме)**

При построении отношений для связи М:М класс принадлежности сущностей на конечный результат не влияет.

Если степень бинарной связей равна **М:М**, то необходимо три отношения: по одному на каждую сущность с первичными ключами от соответствующих сущностей, и одно отношение для связи. Отношение для связи должно иметь среди своих атрибутов ключ каждой сущности как внешние ключи. Первичный ключ третьего отношения составной из первичных ключей двух сущностей.

**Аномалии в БД. Определение избыточности данных**

****

**Аномалии вставки, удаления и обновления**

*Вставка*. При добавлении новых экземпляров отношения может возникнуть ситуация, когда некоторые из атрибутов не имеют значения (они могут принимать значение 0 для арифметических данных или пробелы для символьных данных, если событие еще не наступило). При обработке запросов по таким данным будут выдаваться искаженные результаты.

*Обновление*. Если имеет место большая избыточность данных, то к одному и тому же объекту может относиться несколько кортежей. В этом случае изменения должны вноситься во все записи, относящиеся к этому объекту.

*Удаление*. Аномалия сводится к тому, что при необходимости удаления некоторых значение атрибутов в случае определенной избыточности данных, удаляется вся запись.

**Нормализация**

Процесс разбиения отношения с целью уменьшения вероятности возникновения аномалий манипулирования данными называется ***нормализацией***. Нормализация осуществляется методом декомпозиции.

Логический и методический основной декомпозиции является концепция функциональной зависимости между атрибутами в рассматриваемом отношении.

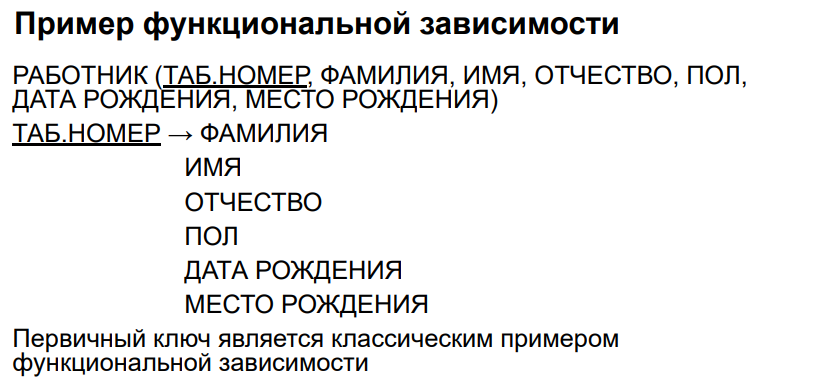
**Функциональная зависимость**

Если даны два атрибута A и , то говорят, что B функционально зависит от A, если для каждого значения A существует ровно одно связанное с ним значение B в любой момент времени: 

При этом A и B могут быть не только атомарными, но и составными.

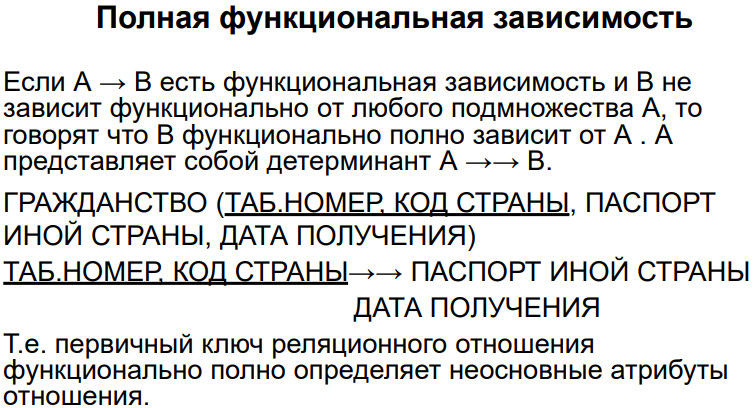
На практике это сводится к тому, что функциональная зависимость B от A означает, что если в любой момент времени известно значение A, то можно одновременно найти значение атрибута B. Примером функциональной зависимости может служить первичный ключ, который определяет значения неосновных атрибутов.

Отсутствие функциональной зависимости 



**Определение функциональной зависимости через термины реляционной алгебры**

Пусть R(A1, A2, …, An) – схема отношения, X и Y – подмножества (A1, A2, …, An). Говорят, что X функционально определяет Y , если в любом отношении R, являющемся текущим значением R, не может содержаться два кортежа, компоненты которых, совпадают по всем атрибутам , принадлежащим множеству X, но не совпадают по одному или более атрибутам принадлежащим множеству Y.



**Способ определения функциональной зависимости**

Единственный способ определения функциональной зависимости для схемы отношения заключается в тщательном анализе **семантики атрибутов** этого отношения. В этом смысле зависимости являются фактически отображением связей, существующих в реальном мире.

Исходя из функциональной зависимости можно определить как первичные, так и возможные ключи.

**Нормальные формы**

Наличие тех или иных зависимостей в схеме отношения определяет степень ее нормализации.

Для реляционных моделей данных выделяют наиболее значимые 4 нормальные формы (1НФ, 2НФ, 3НФ, 4НФ) и еще одну доп НФ форму Бойса-Кодда (НФБК).

Другие более высокие НФ являются значимыми для объектно-ориентированных моделей.

Для приведения реляционного отношения к НФ требуется выполнить его декомпозицию в соответствии с выявленными функциональными зависимостями.

Практически выявленные аномалии свидетельствуют о недостаточно внимательной нормализации сущностей на этапе разработки модели предметной области типа «Сущность-Связь».

**Определение 1НФ**

Отношение находится в 1НФ, если каждый его элемент имеет и всегда будет иметь атомарное строение.

В этом случае реляционное отношение представлено в виде множества неповторяющихся кортежей.

Эта форма является необходимой для работы языков запросов СУБД. Формально само определение реляционного отношения подразумевает удовлетворение 1 НФ.

Представление атрибута как частично структурированного, например АДРЕС, не является аномалией по 1НФ.

**Определение 2НФ**

Отношение задано во 2НФ, если оно является отношением в 1НФ, и каждый атрибут не являющийся первичным атрибутом в этом отношении, функционально полно зависит от первичного и любого возможного (альтернативного) ключа этого отношения.

***Первичным*** называется атрибут, который входит в хотя бы один из возможных ключей.

Если все возможные ключи отношения содержат по одному атрибуту, т.е. являются несоставными, то это отношение задано во 2НФ, т.к. в этом случае все атрибуты не являющиеся первичными полностью зависят от возможных ключей.

Если ключи составные, отношение заданное в 1НФ может не быть отношением во 2НФ.

**Определение 3НФ**

Отношение задано в 3НФ, если оно задано во 2НФ и каждый атрибут из отношения, не являющийся первичным, нетранзитивно зависит от каждого возможного ключа.

Пусть A, B, C – атрибуты реляционного отношения. Если C зависит от B, а B – от A, то C зависит от A. Если при этом обратное соответствие неоднозначно (т.е. A не зависит от B или B не зависит от C), то говорят, что C транзитивно от A.

|  |  |
| --- | --- |
|  | Преобразование к 3НФ состоит в разбиении исходного отношения на два:    Функциональная зависимость  исключается.  3НФ является достаточной для нормализации реляционной модели данных. |

**Определение НФ Бойса-Кодда**

Отношение находится в НФБК, если каждый детерминант отношения является возможным ключом.

НФБК является дополнительной проверкой правильной нормализации по предыдущим нормальным формам.

Также используется для нормализации универсального реляционного отношения.

**Определение 4НФ**

Схема реляционного отношения задана в 4НФ, если при существовании многозначной зависимости X определяет Y, где Y не является пустым множеством, Y не принадлежит X, причем XY состоит не из всех атрибутов отношения, также существует зависимость  для любого атрибута A отношений.



**Концепция Алгоритма декомпозиции без потерь.**

Предполагаемый алгоритм целесообразно использовать, если в БД учитывается порядка 20-30 атрибутов. Для баз данных используемых в реальных информационных системах такое практически невозможно.

В основу алгоритма полагается удовлетворение НФБК.

1. Разработка универсального отношения для БД.

Универсальное реляционное отношение представляет собой совокупность атомарных атрибутов составляющих БД. В случае повторяющихся групп, они оформляются путем введения избыточности данных.

2. Определение всех функциональных зависимостей между атрибутами отношения.

3. Определение того, что находится ли отношение в НФБК. Если «ДА» проектирование завершается, если «НЕТ», отношение должно быть разложено на отношения.

**Алгоритм декомпозиции без потерь**

Пусть отношение R(A, B, C, D, E… ) не приведено к НФБК.

Определяется функциональная зависимость , про которую известно, что она является причиной того, что отношение не находится в НФБК. С является детерминантом, но не является возможным ключом. Создаются 2 новых отношения.

R1(A, B, C, E…)

R2 (C, D…)

Отношение R2 называется проекцией отношения R.



# **ЛЕКЦИЯ №8. СОВРЕМЕННЫЕ ПРОБЛЕМЫ ЭКСПЛУАТАЦИИ И РАЗВИТИЯ ТЕХНОЛОГИИ БАЗ ДАННЫХ**

**Концепция хранилища данных**

***Хранилище данных*** — разновидность систем хранения, ориентированная на поддержку процесса анализа данных, обеспечивающая целостность, непротиворечивость и хронологию данных, а также высокую скорость выполнения аналитических запросов.

***Хранилище данных*** – предметно-ориентированные, интегрированные, неизменные, поддерживающие хронологию наборы данных, организованные для целей поддержки управления, призванные вступать в роли единого и единственного источника истины, обеспечивающего менеджеров и аналитиков достоверной информацией, необходимой для оперативного анализа и принятия решений.

**Концепция хранилища данных:**

* Интеграция ранее разъединенных детализированных данных (исторические архивы, данные из традиционных систем обработки документов, разрозненных БД, данные из внешних источников) в едином хранилище данных;
* Тематическое и временное структурирование, согласование и агрегирование;
* Разделение наборов данных, используемых для операционной (производственной) обработки, и наборов данных, используемых дял решения задач анализа.

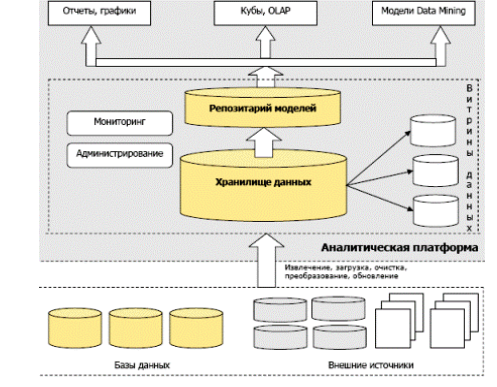
**Достоинства архитектуры классического хранилища данных:**

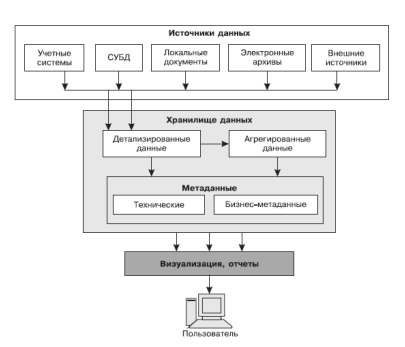
* Согласованный набор процессов извлечения и бизнес-логики использования;
* Непротиворечивость содержащейся информации;
* Легко создаваемые по шаблонам и напоминаемые витрины данных;
* Централизованная, управляемая среда;
* Единый репозиторий метаданных;
* Многообразие механизмов обработки и представления данных
* Общая семантика

**Хранилища данных (Datawarehouse) и оперативный анализ данных (OnLineAnalyticalProcessing, OLAP) – новые информационные технологии**, которые обеспечивают аналитика возможность изучать большие объемы взаимосвязанных данных при помощи быстрого интерактивного отображения информации на разных уровнях детализации с различных точек зрения в соответствии с представлениями пользователя о предметном пространстве.

**Компоненты хранилища:**

* Подсистемы загрузки данных – в соответствии с определенным регламентом извлекает данные из источников и приводит их к единому формату, определенному для хранилища. Отвечает за формализованную логическую согласованность, качество и интеграцию данных;
* Подсистемы обработки запросов и представления данных – реализует обработку регламентированных и нерегламентированных запросов и извлечение закономерностей из хранящихся данных
* Подсистемы администрирования хранилища реализует задачи, связанные с поддерживанием системы и обеспечением ее устойчивой работы и расширения





**Распределенные БД**

***Распределенная БД (Distributed Database, DDB)*** – это совокупность нескольких взаимосвязанных БД, которые физически распределены по различным местам, которые обмениваются данными через компьютерную сеть.

**Принципы построения РБД:**

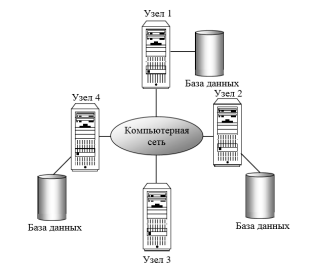
* Минимизация интенсивности обмена данными (сетевого трафика)
* Оптимальное размещение серверных и клиентских приложений в сети
* Декомпозиция данных на часто и редко используемые сегменты (для правильной настройки репликации – размещение наиболее часто используемых данных на АРМ конченых пользователей)
* Периодическое сохранение копий данных выполнение действий по поддержке целостности распределенной ИС.

**Достоинства РБД:**

* Отражение структуры организации;
* Разделяемость и локальная автономность;
* Повышение доступности данных;
* Повышение надежности;
* Повышение производительности;
* Экономические выгоды;
* Модульность системы

**Недостатки РБД:**

* Повышение сложности;
* Увеличение стоимости;
* Проблемы с защитой;
* Усложнение контроля за целостностью данных;
* Отсутствие стандартов;
* Недостаток опыта;
* Усложнение процедуры разработки БД.

****

**Распределенные реестры**

***Технология распределенных реестров (Distributed ledger technology, DLT***) – то подход к обмену и хранению информации, **при котором:**

* Каждый участник может обладать полноценной копией реестра;
* Синхронизация копий реестра происходит на основе протокола достижения распределенного консенсуса, то есть, соглашения среди участников на добавление новой информации;
* Каждый участник взаимодействуя может иметь доступ к истории транзакций.

**Достоинства DTL:**

* ***Надежность.*** DTL-системы отличаются большим количеством узлов, которые достигают соглашения при обработки операций. Помимо этого распределенные реестры не имеют слабого места, в отличие от традиционных БД. Такая неуязвимость помогает им успешно отражать атаки хакеров.
* ***Прозрачность и доступность*.** Все авторизованные пользователи могут отслеживать и проверять информацию, а также обмениваться зашифрованными данными, что формирует высокий уровень доверия. Кроме того, такая система способствует быстрому обнаружению манипуляций на рынке.
* ***Безопасность и конфиденциальность.*** DTL обеспечивает высокий уровень безопасности и секретности хранящийся информации
* ***Цифровое представление данных.*** С помощью системы распределенных реестров любая информация ил актив могут быть представлены в «цифре».
* ***Ускорение процессов за счет отсутствия посредников. Благодаря*** многоуровневому механизму мониторинга DTL нуждается в третьих сторонах.
* ***Сокращение операционных затрат***. Благодаря отсутствию посредников и высокому уровню автоматизации удается существенно сократить издержки.

**Недостатки DTL:**

* Неоднозначное правовое регулирование.
* Необходимость в высокой вычислительной мощности.
* Необходимость реформирования бизнес-стратегии.
* Недостаточная осведомленность пользователей

**Потоковые базы данных**

Обработка в режиме реального времени выполняется для потоков данных, получаемых в реальном времени и обрабатываемых с минимальной задержкой для создания отчетов или автоматизированного реагирования в режиме реального времени (или приближенном к реальному времени).

Принцип работы PipelineDB можно сформулировать так: «постоянные запросы, кратковременные данные». В реляционных СУБД всё обстоит ровно наоборот: «кратковременные запросы, постоянные данные. В PipelineDB данные не хранятся, а поступают в потоке; их обработка происходит «на лету», в движении.

**Архитектура обработки в режиме реального времени**

* + ***Прием сообщений в реальном времени***. Архитектура должна включать средства сбора и сохранения сообщений в режиме реального времени, доступные для объекта-получателя, обрабатывающего поток. В простых случаях роль такой службы может выполнять обычное хранилище данных, в одной из папок которого размещаются новые сообщения. Но чаще для этого компонента нужны специализированные брокеры обмена сообщениями, например Центры событий Azure, которые выполняют роль буфера для входящих сообщений. Брокер обмена сообщениями должен поддерживать масштабируемую обработку и надежную доставку.
  + ***Потоковая обработка***. Сохранив сообщения, поступающие в режиме реального времени, система выполняет для них фильтрацию, статистическую обработку и другие процессы подготовки данных к анализу.
  + ***Хранилище аналитических данных***. Многие решения по обработке больших данных спроектированы так, чтобы подготавливать данные к анализу и предоставлять их в структурированном формате для запросов через средства аналитики.
  + ***Анализ и создание отчетов***. Большинство решений по обработке больших данных предназначены для анализа и составления отчетов, что позволяет получить важную информацию.