**1. Информация, Данные, Знания. Терминология.**

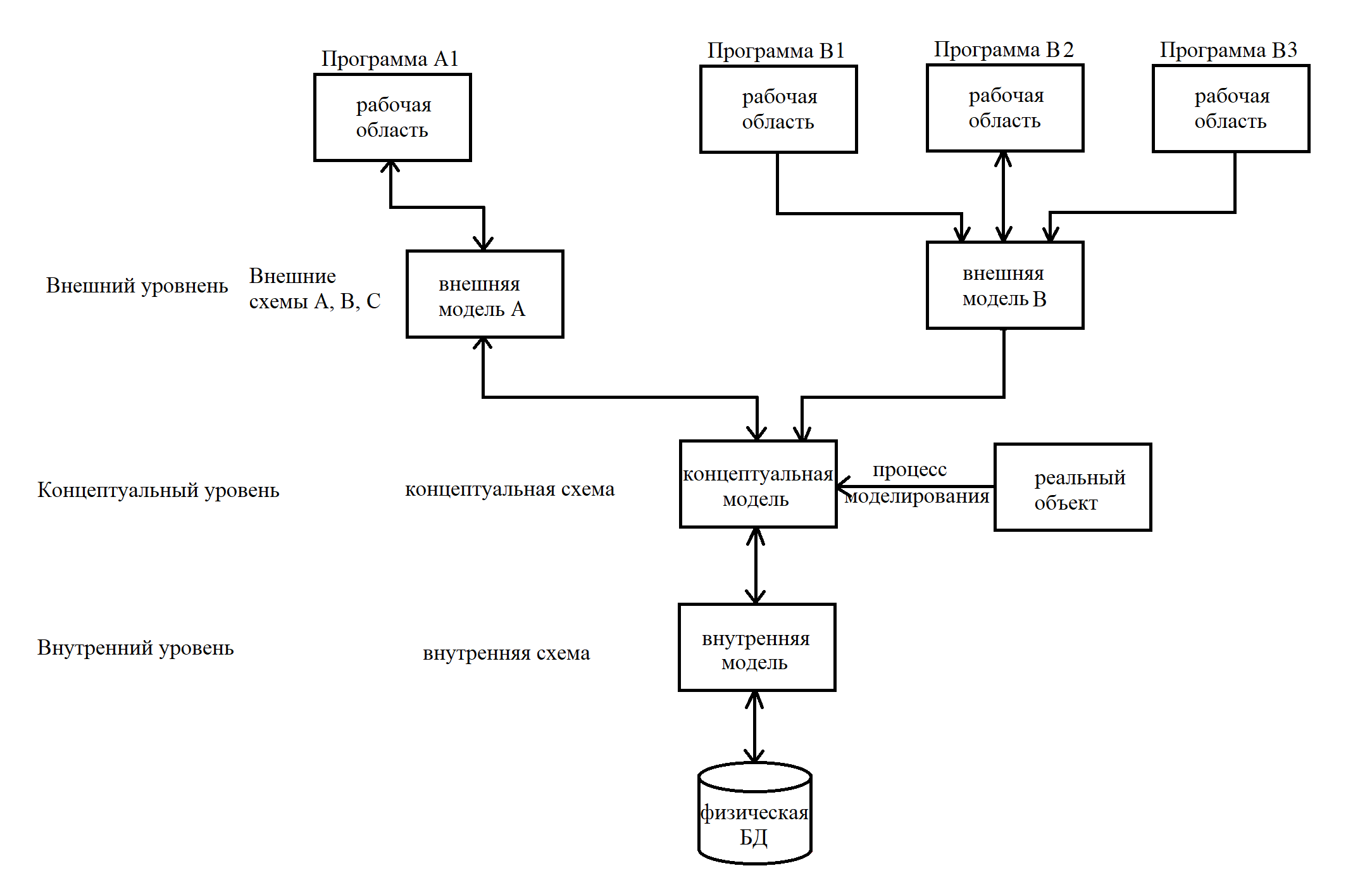
**Информация** — любые сведения о каком-либо событии, сущности, процессе. Информация является результатом определенных операций: восприятия, передачи, преобразования, хранения и использования.  
**Данные** — это информация, зафиксированная в форме пригодной для последующей обработки, передачи, хранения, находящейся в памяти ЭВМ или подготовленное для ввода в ЭВМ. Подготовка информации состоит в ее формализации, сборе, переносе на машинные носители (перфоленты, дискеты, жесткий диск). **Обработка данных** — это совокупность задач, осуществляющих преобразование данных. Она включает в себя ввод данных в ЭВМ, отбор или выборку по каким-то критериям, преобразование структуры данных и вывода данных. Вывод осуществляется в удобном виде для пользователя (таблица, график). **Система обработки данных СОД** — набор аппаратных и программных средств осуществляющих выполнение задач по управлению данными. **Управление данными** — совокупность функция, обеспечивающих требуемое представление данных, их накопление, хранение, обновление, удаление, поиск по критерию и выдачи данных. **Предметная область ПрО** — часть реального мира подлежащая изучению с целью организации управления и в конечном итоге автоматизации. **База данных БД** — совокупность данных по определенным правилам предусматривающая общие принципы описания данных, хранения, манипулирование данными и БД является независимой от прикладных программ. **Ведение БД** — деятельность по обновлению, восстановлению, изменению структуры БД с целью обеспечить ее целостности, сохранности и эффективности использования. **Система управления БД СУБД** —совокупность программ и языковых средств, предназначенных для управления данными в БД и для обеспечения взаимодействия ее с прикладными программами. **Банк данных БнД** — автоматизированная информационная система, включающая свойства БД и СУБД.

**2. Предметная область информационной системы.**

Предметную область информационной системы рассмотрим, как совокупность реальных процессов и объектов. Эти процессы и объекты называются **сущности**. Каждая из сущностей ПрО обладает определенным набором свойств или атрибутов. Для управления процедуры описывающей ПрО прибегают к определению типов сущности. **Тип сущности** позволяет выделить группы сущностей однородных по структуре и поведению, например, возьмем ПрО университет, где можно выделить следующие типы сущностей: студенты, преподаватели, дисциплины. Все данные ПрО представляются экземплярами сущностей, например, студент — Леваневская Н.И., преподаватель — Савонюк В.А., дисциплина — ПБЗ. Экземпляры сущности одного типа обладают одинаковыми свойствами либо у них одинаковый набор атрибутов, но экземпляры сущностей должны отличаться значением хотя-бы одного атрибута, чтобы их можно было различить. Существуют атрибуты существенные и малозначимые. **Атрибуты** **классифицируются** следующим образом: **1)** описательные (ФИО, пол) и идентифицирующие (имеют уникальное значение). **2)** составные (например, ФИО) и простые/атомарные (состоят из 1 компонента). **3)** однозначные и многозначные. **4)** основные (значения не зависят от других атрибутов) и производные (вычисляются на основе значений других атрибутов). **5)** обязательные (всегда устанавливаются при вводе в БД) и необязательные (могут быть пропущены при вводе в БД).

**3. Трехуровневая архитектура БД**

Современная технология БД основана на концепции многоуровневой СУБД.



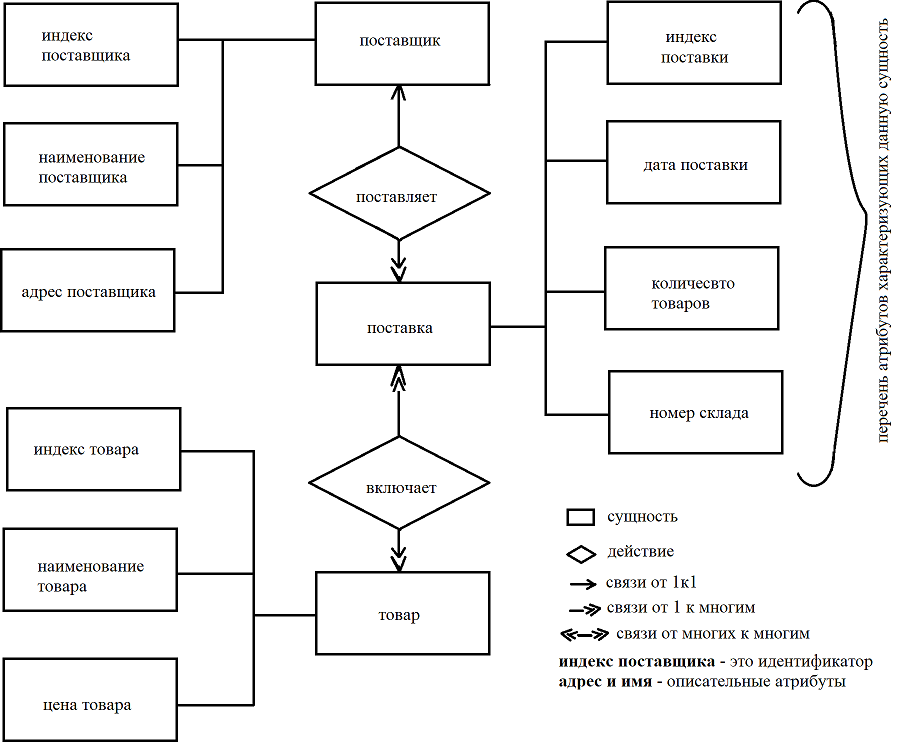
На основании реального объекта реализуется концептуальная модель. Она описывается концептуальной схемой. Эта модель не зависит от прикладного ПО. На ее основании для пользователя формируется внешний уровень, на каждом внешнем уровне формируется внешние модели. Внешний уровень связан с приложением. На основе концептуальной схемы формируется внутренняя модель, соответствующая физической организации БД.

**4. Концептуальный уровень архитектуры.**

**Концептуальная модель** — это описание БД в терминах конкретной модели данных. Иногда называют диаграммой модели ПрО.  
Основные свойства и назначение концептуальной модели: **1)** концептуальная схема объединена специфическими требованиями, предъявленными пользователю к информации и является связью между внутренней и внешней схемой. **2)** концептуальная схема должна быть стабильной, т.е. не зависеть от данных во внутренней схеме. **3)** внутренняя схема, полученная из концептуальной с помощью параметров конкретного объекта. Эти параметры со временем могут меняться и влиять на внутреннюю схему, но концептуальная схема остается той же. **4)** концептуальная схема, реализуемая во внутренней должна быть совместима с различными существующими моделями данных. **5)** изменения или добавления во внешних схемах не должны влиять на концептуальную схему, но могут изменять параметры, которые используются для разработки внутренней схемы. Это может привести к реорганизации базы данных.

**5. Пример концептуальной схемы.**

Пусть у нас есть какие-то поставщики, например, швейное производство, у которых есть потребители, поставщики оформляют поставку. Предметной областью является поставка товара.



После нарисованной схемы формируют спецификации на эту схему. Их 3 вида: **1)** спецификация сущностей строится в виде таблицы **2)** спецификация типов связи: поставщик <- поставляет это связь один к многим 1:N, связь поставки ->> товар M:N **3)** спецификация атрибутов.

**6. Спецификация атрибутов.**

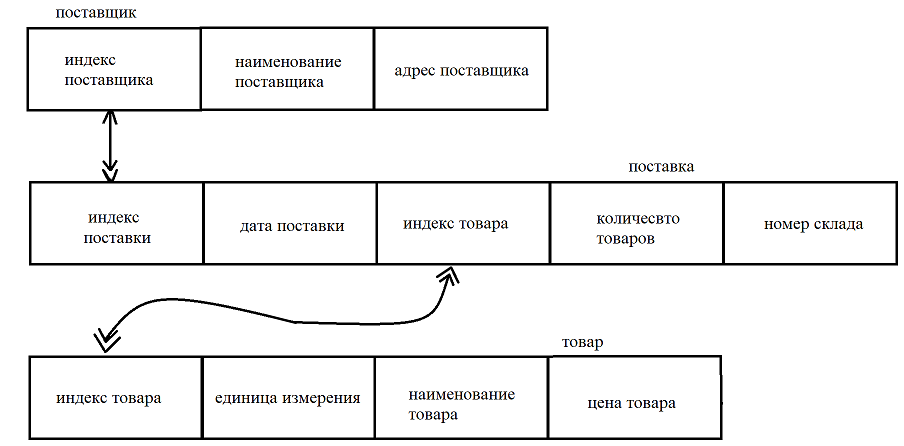
**Поставщики**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Название атрибута** | **Идентификатор** | **Тип** | **Размерность** | **Примечание** |
| **1.** | Индекс поставщика | INP | INT | 6 |  |
| **2.** | Наименование поставщика | NP | символьный | 30 |  |
| **3.** | Адрес | ADRS | символьный | 60 |  |

**Товар**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Название атрибута** | **Идентификатор** | **Тип** | **Размерность** | **Примечание** |
| **1.** | Код товара | KT | цифирный | 11 |  |
| **2.** | Наименование товара | NT | символьный | 60 |  |
| **3.** | Единица товара | E | символьный | 5 |  |
| **4.** | Цена | CENA | цифирный | 5 |  |

После расписывания спецификаций формируется логическая схема БД



**7. Внешний уровень архитектуры.**

Предназначен для групп пользователей. Наличие внешнего уровня позволяет поддерживать различное представление одних и тех же данных для различных пользователей или задач. На внешний уровень БД описывается внешней моделью, является частным представлением БД пользователем. В связи с этим она включает классы объектов и связей, соответствующие конкретным пользователем. Внешняя схема рассматривается как подсхема концептуальной или логической модели. Функциональные возможности для пользователя определяются независима.

**8. Внутренний уровень архитектуры.**

Он поддерживает представление данных в среде хранения и в пути доступа к этим данным. Описание данных на **внутреннем уровне** называется внутренней схемой хранения. На внутреннем уровне БД представляет собой набор данных или файлов. Файл или набор данных состоит из записей. Эти записи для эффективного использования памяти и времени доступа могут блокироваться. Основными характеристика записи набора данных является: **1)** тип организации. **2)** метод доступа. **3)** формат внутренней записи. **4)** записи переменной длины. **5)** записи неопределенной длины. **6)** длина записи и размер блока, если записи блокируются. На концептуальном уровне каждой записи соответствует внутренняя запись на внутреннем уровне. Внутренняя запись состоит из 2 частей: **1)** информационная часть состоит из наборов данных. **2)** Служебная часть пользователь не имеет доступа к служебной информации, доступ имеет только СУБД.

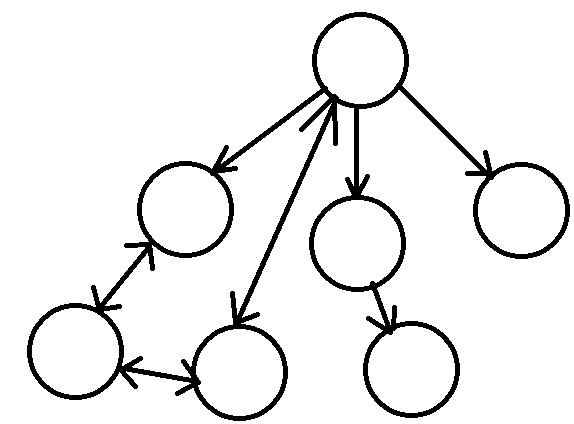
**9. Понятие модели данных.**

Модель данных служит инструментов моделирования произвольной ПрО. **Модель данных** — это совокупность правил, порождение структур данных в БД, операций над ними, а также ограничений целостности (правило определяющее достоверность информации, а также определяющее правильность данных между связью). Она состоит из 3 частей: **1)** набор типов структур данных, с которыми работает данная модель. **2)** набор операторов или правил вывода, которые могут быть применены к любым правильным примерам типов данных, чтобы осуществлять поиск, преобразовывать информацию, выводить информацию, содержащуюся в любых частях этих структур и в любых комбинациях. **3)** набор общих правил целостности, которое прямо или косвенно определяет множество не противоположных состояний БД и/или множество изменений ее состояний.

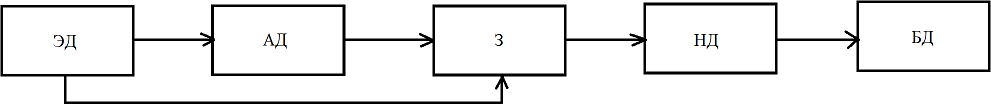
**10. Сетевая модель данных.**

Сетевые БД опираются на математику графов, т.е. сетевую модель можно представить в виде ориентированного графа.

Пример сетевой модели



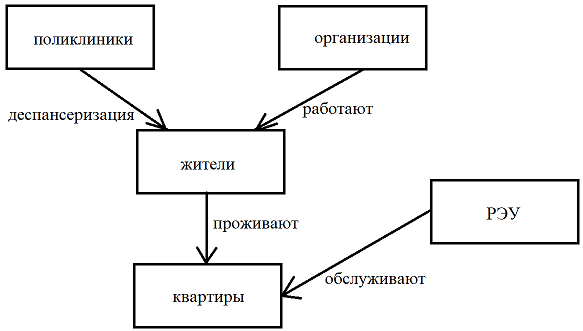
Ориентированный граф состоит из узлов и ребер, ребра имеют направление, указываемое стрелкой. Узлы направленного графа — это объекты сетевой БД, а ребра показывают связи между объектами сетевой БД причем ребра показывают не только сами связи, но и тип связи. Сетевые базы имеют простую структуру. В ней используется 4 типа структур данных (2 главные и 2 неглавные). **Главные типы структур** — запись и набор. **Не** **главные** — элементы данных и агрегат данных.



**Элемент данных** — наименьшая информационная именованная единица данных доступная для пользователя**. Агрегат данных** — представляет собой обобщенный элемент сетевой модели. Именованная совокупность данных внутри 1 записи, состоящая из нескольких элементов. В реляционных БД отсутствует аналогия с АД. **Запись** — конечный уровень обобщения данных.

**11. Набор или схема Бакмана.**

Это наименованная совокупность записей, образующая двух уровневую иерархическую структуру. Каждый тип набора представляет собой связь между 2 или более записями. Один тип, объявленный владельцем набора, остальные членами набора. С помощью наборов формируют групповые отношения и эти отношения строятся с помощью схемы Бакмана.



Групповые отношения чаще всего описываются связью 1к1, 1кМ, но может и NкМ.

**12. Иерархическая модель.**

Иерархическая модель данных позволяет строить БД с иерархической древовидной структурой. Запись в иерархической структуре называется **сегментом**. В основе информационной модели иерархического типа было дерево.

|  |  |
| --- | --- |
|  | **Дерево** — ориентированный граф без циклов, имеется одна единственная особенная вершина, которая называется **корнем** и в которую не заходит не одно ребро, а исходит произвольное количество. Те вершины, из которых не выходит ни одно ребро называются **листьями** дерева. Количество вершин на пути от корня к листьям может быть различно. В иерархической модели используется ориентация от корня к листьям. Иерархические структуры поддерживают отношения один к одному и один к многим*. Есть* ***проблемы****, связанные с корректировкой данных, так как нужно пройти от корня к измененному сегменту. Необходим контроль, чтобы при таких операциях не появились висячие вершины.* Корневой сегмент должен содержать ключ с уникальным значением, а сегменты должны иметь уникальный ключ только в своём сегменте, это объясняется тем, что каждая запись идентифицируется соединением ключей родительских записей. Например, студенты — листья и идентифицируются по номеру зачетной книжки, Ключ будет представлять собой следующую группу: код факультета, номер группы, номер зачетной книжки. Недостаток: некоторая избыточность информации за счёт дублирования данных. |

**13. Реляционная модель.**

Основное понятия. В реляционной модели информация хранится в одной или нескольких связанных таблицах, отдельная таблица представляет собой совокупность либо реальных объектов, либо некоторых абстрактных, событий одного типа. Каждая запись идентифицирует один объект группы. Таблицы обладают следующими свойствами:  
**1)** Каждый элемент таблицы представляет один элемент данных, то есть группы значений в одной строке одном столбце недопустима.   
**2)** Все столбцы в таблице однородные. Это означает, что элементы столбца имеют одинаковую природу и столбцам присваиваются имена.  
**3)** В таблице нет двух одинаковых строк.  
**4)** Порядок размещения строк в таблице и столбцов может быть произвольным. В оперировании с такой таблицей строки и столбцы могут просматриваться в любом порядке.  
**5)** Все операции производятся с учетом **реляционной алгебры** — это процедурный язык, так как отношения являются результатом запроса к БД, вычисляемого при выполнении последовательности реляционных операторов. Результатом реляционных операторов является отношение.  
**6)** К БД реляционного типа можно применять **реляционные исчисления** — это непроцедурный язык описательного или декларативного характера, который содержит информацию о желаемом результате. Процесс вычисления скрыт от пользователя. К языкам такого типа относится SQL.  
**Отношение** — двумерное множество или таблица, которая удовлетворяет вышеперечисленным свойствам.  
**Атрибут** — свойство, которое характеризует объект, в структуре таблицы каждый атрибут имеет имя, в нашем случае это П1, П2, П3, то есть все атрибуты данного типа — столбец. **Количество атрибутов называется степенью отношений.   
Кортеж** — строка таблицы, в общем в общем случае это атрибут-значение. Каждое значение должно быть атомарным. **Атомарным** значит не может быть многозначным или составным. **Количество кортежей** — это кардинальное число.  
**Домен** — множество всех возможных значений определяющее атрибуты отношений.  
**Первичный ключ** — атрибут отношений однозначно идентифицирующий каждый кортеж, ключ может быть составным, то есть состоять из нескольких атрибутов.  
**Потенциальный ключ** — подмножество атрибутов отношений, которые обладают следующими 2 свойствами: **1)** свойство уникальности (нет одинаковых кортежей с теми же значениями). **2)** свойство неизбыточности.  
**Внешний ключ** — атрибут, подчиненного отношения, который используется для установления связи с базовым отношением.

**14. Отношения.**

Теоретическим фундаментом реляционных БД является математическая теория отношений. Всякая система связанна с множеством объектов. Например, математика связана с множествами чисел: натуральных, целых, комплексных. Помимо этих элементов между ними существуют отношения. Эти отношения называют **бинарными**. Бывают отношения с тремя объектами, их называют **тернарными**. Также есть один к многим — **n-нарные**. Бинарные отношения между множествами A и B обозначаются R(A,B) или если A состоит из {a1, a2, … an}, а B состоит из {b1, b2, … bk}, то a1 R b3 обозначает, что a1 находится в отношении с b3. **Например**, Есть семья в семье трое детей: Ольга, Павел и Иван. Отношение A – является братом B будет представлять следующее множество   
R = { (Павел, Ольга), (Павел, Иван), (Иван, Ольга), (Иван, Павел) }.

**15. Связи в РБ.**

В реляционных базах отношения между сущностями определяют связи между таблицами. Они могут быть 4 типов:

|  |  |  |
| --- | --- | --- |
| **1)** один к одному | **2)** один к многим **3)** многие к одному | **4)** многие к многим |

Отношение многие к многим обозначает, что каждой записи из одной таблицы может соответствовать одна и более запись из другой таблицы.

**16. Реляционная алгебра. Объединение. Разность.**

Реляционная алгебра представляет собой совокупность операций высокого уровня над отношениями, эти операции делятся на 2 группы. **Группа A** — основные операции: объединение, разность, декартово произведение, выбор. **Группа Б** — дополнительные операции, которые обеспечивают более краткую запись. Операндами операций реляционной алгебры являются постоянные или переменные отношения. Для операций объединения, разность, пересечение должны быть: совместимы по объединению, иметь одинаковую арность и атрибуты, по которым осуществляются операции.  
**Пример**: пусть R(ABC) = { p1a, p2b, q2c }, S(DEF) = { p2b, q3a }, тогда:  
Операция **объединение** множеств R и S — отношение T, включающее в себя множество кортежей принадлежащих R или S или R и S. Все кортежи результирующего множества имеют одинаковую арность, так как арность отношений R и S одинакова. T = R U S = { p1a, p2b, q2c, q3a }.  
Операция **разность** двух отношений R и S называется отношение T, которое включает множество кортежей, принадлежащих R, но не принадлежащих S, отношения имеют одинаковую арность. T = R – S = { p1a, q2c }.

**17. Реляционная алгебра. Декартово произведение. Пересечение.**

Реляционная алгебра представляет собой совокупность операций высокого уровня над отношениями, эти операции делятся на 2 группы. **Группа A** — основные операции: объединение, разность, декартово произведение, выбор. **Группа Б** — дополнительные операции, которые обеспечивают более краткую запись. Операндами операций реляционной алгебры являются постоянные или переменные отношения. Для операций объединения, разность, пересечение должны быть: совместимы по объединению, иметь одинаковую арность и атрибуты, по которым осуществляются операции.  
**Пример**: пусть R(ABC) = { p1a, p2b, q2c }, S(DEF) = { p2b, q3a }, тогда:  
Операция **декартово произведение** над отношениями R и S называется отношение T, которое содержит результат — множество кортежей длиной k1 + k2, первые k1 компонентов относятся к отношению R, а вторые k2 — к отношению S. T = R \* S = { p1ap2b, p1aq3a, p2bp2b, p2bq3a, q2cp2b, q2cq3a }.  
Операция **пересечения** множеств R и S называется отношение, которое включает кортежи принадлежащие как R, так и S. T = R ∩ S = { p2b }.

**18. Реляционная алгебра. Проекция. Выбор.**

Реляционная алгебра представляет собой совокупность операций высокого уровня над отношениями, эти операции делятся на 2 группы. **Группа A** — основные операции: объединение, разность, декартово произведение, выбор. **Группа Б** — дополнительные операции, которые обеспечивают более краткую запись. Операндами операций реляционной алгебры являются постоянные или переменные отношения. Для операций объединения, разность, пересечение должны быть: совместимы по объединению, иметь одинаковую арность и атрибуты, по которым осуществляются операции.  
**Пример**: пусть R(ABC) = { p1a, p2b, q2c }, S(DEF) = { p2b, q3a }, тогда:  
Операция **проекция xR на Y** называется отношение R[y], которое является вертикальным подмножеством отношения R. R[y] включает все кортежи, полученные путём отбора компонент соответствующих **y** и удалению одинаковых кортежей кроме 1 из множества, полученных в результате проекции. **Пх(R)** — проекция. **[y]** — подмножество столбцов из отношения R. R[AC](AC)S[EF](EF) = { pa2b, pb3a, qc }.  
Операция **выбор** строит горизонтальное подмножество кортежей отношения R, удовлетворяющих определенному предикату, под предикатом подразумевается одна из бинарных операций >, <, >=, <=, =, ≠ . R[C != b] (ABC) = { p1a, q2c }.

**19. Реляционная алгебра. Соединение. Слияние.**

Реляционная алгебра представляет собой совокупность операций высокого уровня над отношениями, эти операции делятся на 2 группы. **Группа A** — основные операции: объединение, разность, декартово произведение, выбор. **Группа Б** — дополнительные операции, которые обеспечивают более краткую запись. Операндами операций реляционной алгебры являются постоянные или переменные отношения. Для операций объединения, разность, пересечение должны быть: совместимы по объединению, иметь одинаковую арность и атрибуты, по которым осуществляются операции.  
**Пример**: пусть R(ABC) = { p1a, p2b, q2c }, S(DEF) = { p2b, q3a }, тогда:  
Операция **соединение** R и S работает как соединение кортежей для которых выполняется определенное условие. T = R(A=D)S = { p1ap2b, p2bp2b, q2cq3a }. Арность T = k = k1 + k2. Туда войдут кортежи, у которых атрибут A из R совпадает с атрибутом D из S.  
Операция **слияния** R ᐅᐊ S выполняется точно также как операция соединения, при использовании операции = между атрибутами, при этом в результате удаляются одинаковые столбцы. R[A\*D] S(ABCEF) = { p1ap2b, p2bp2b, q2cq3a } .

**20. Реляционная алгебра. Декартово произведение, деление.**

Реляционная алгебра представляет собой совокупность операций высокого уровня над отношениями, эти операции делятся на 2 группы. **Группа A** — основные операции: объединение, разность, декартово произведение, выбор. **Группа Б** — дополнительные операции, которые обеспечивают более краткую запись. Операндами операций реляционной алгебры являются постоянные или переменные отношения. Для операций объединения, разность, пересечение должны быть: совместимы по объединению, иметь одинаковую арность и атрибуты, по которым осуществляются операции.  
**Пример**: пусть R(ABC) = { p1a, p2b, q2c }, S(DEF) = { p2b, q3a }, тогда:  
Операция **декартово произведение** над отношениями R и S называется отношение T, которое содержит результат — множество кортежей длиной k1 + k2, первые k1 компонентов относятся к отношению R, а вторые k2 — к отношению S. T = R \* S = { p1ap2b, p1aq3a, p2bp2b, p2bq3a, q2cp2b, q2cq3a }.  
Пусть задано отношение P(BC) = { 1a, 2c }.   
Арность R = 3, P = 2. R — делимое, P — делитель. Операция R делить на P — отношение, включающее компоненты, значения BC которых совпадает с делителем. T = R÷P = { p, q }. Арность T = k = k1 – k2.