**Лекция №9**

**Концептуальные модели данных**

В отличие от инфологической модели ПрОбл, описывающей по некоторым правилам сведения об объектах материального мира и связи между ними, которые следует иметь в БД, *концептуальная модель описывает хранимые в ЭВМ данные и связи. В силу этого каждая модель данных неразрывно связана с языком описания данных*конкретной СУБД (см. Лекция №8 рис. 3).   
По существу модель данных — это совокупность трех составляющих: 

* *типов (структур)*данных;
* *операций*над данными;
* *ограничений*целостности.

Другими словами, модель данных представляет собой некоторое интеллектуальное средство проектировщика, позволяющее реализовать интерпретацию сведений о ПрОбл в виде формализованных данных в соответствии с определенными требованиями, т.е. средство абстракции, которое дает возможность увидеть "лес" (информационное содержание данных), а не отдельные "деревья" (конкретные значения данных). 

**Типы структур данных**

Среди широкого множества определений, обозначающих типы структур данных, наиболее распространена терминология КОДАСИЛ (Conference of DAta SYstems Language) – международной ассоциации по языкам систем обработки данных, созданной в 1959 г.   
В соответствии с этой терминологией используют пять типовых структур (в порядке усложнения): 

* *элемент данных*;
* *агрегат данных*;
* *запись*;
* *набор*;
* *база данных*.

Дадим краткие определения этих структур.   
*Элемент данных*— наименьшая поименованная единица данных, к которой СУБД может адресоваться непосредственно и с помощью которой выполняется построение всех остальных структур данных.   
*Агрегат данных*— поименованная совокупность элементов данных, которую можно рассматривать как единое целое. Агрегат может быть простым или составным (если он включает в себя другие агрегаты).   
*Запись*— поименованная совокупность элементов данных и (или) агрегатов. Таким образом, запись – это агрегат, не входящий в другие агрегаты. Запись может иметь сложную иерархическую структуру, поскольку допускает многократное применение агрегации.   
*Набор*— поименованная совокупность записей, образующих двухуровневую иерархическую структуру. Каждый тип набора представляет собой связь между двумя типами записей. Набор определяется путем объявления одного типа записи "записью-владельцем", а других типов записей — "записями-членами". При этом каждый экземпляр набора должен содержать один экземпляр "записи-владельца" и любое количество "записей-членов". Если запись представляет в модели данных сущность, то набор — связь между сущностями. Например, если рассматривать связь "учится" между сущностями "учебная группа" и "студент", то первая из сущностей объявляется "записью-владельцем" (она в экземпляре набора одна), а вторая – "записью-членом" (их в экземпляре набора может быть несколько).   
*База данных*поименованная совокупность экземпляров записей различного типа, содержащая ссылки между записями, представленные экземплярами наборов.   
Отметим, что структуры БД строятся на основании следующих основных композиционных правил: 

* БД может содержать *любое количество типов записей и типов наборов*;
* между *двумя типами записей*может быть определено *любое количество наборов*;
* тип записи может быть *владельцем*и *одновременно членом*нескольких типов наборов.

Следование данным правилам позволяет моделировать данные о сколь угодно сложной ПрОбл с требуемым уровнем полноты и детализации.   
Рассмотренные типы структур данных могут быть представлены в различной форме — графовой; табличной; в виде исходного текста языка описания данных конкретной СУБД. 

**Операции над данными**

Операции, реализуемые СУБД, включают *селекцию*(поиск) данных; *действия*над данными.   
Селекция данных выполняется с помощью критерия, основанного на использовании либо логической позиции данного (элемента; агрегата; записи), либо значения данного, либо связей между данными.   
Селекция на основе логической позиции данного базируется на упорядоченности данных в памяти системы. При этом критерии поиска могут формулироваться следующим образом: 

* найти следующее данное (запись);
* найти предыдущее данное;
* найти *n*-ое данное;
* найти первое (последнее) данное. Этот тип селекции называют *селекцией посредством текущей,*в качестве которой используется индикатор текущего состояния, автоматически поддерживаемый СУБД и, как правило, указывающий на некоторый экземпляр записи БД.

Критерий селекции по значениям данных формируется из простых или булевых условий отбора. Примерами простых условий поиска являются: 

* ВУС = 200100;
* ВОЗРАСТ > 20;
* ДАТА < 19.04.2002 и т. п.

Булево условие отбора формируется путем объединения простых условий с применением логических операций, например: 

* (ДАТА\_РОЖДЕНИЯ < 28.12.1963) И (СТАЖ> 10);
* (УЧЕНОЕ\_ЗВАНИЕ=ДОЦЕНТ:) ИЛИ (УЧЕНОЕЗВАНИЕ-ПРОФЕССОР).

Если модель данных, поддерживаемая некоторой СУБД, позволяет выполнить селекцию данных по связям, то можно найти данные, связанные с текущим значением какого-либо данного. Например, если в модели данных реализована двунаправленная связь "учится" между сущностями "студент" и "учебная группа", можно выявить учебные группы, в которых учатся юноши (если в составе описания студента входит атрибут "пол").   
Как правило, большинство современных СУБД позволяет осуществлять различные комбинации описанных выше видов селекции данных. 

**Ограничения целостности**

*Ограничения целостности*— логические ограничения на данные — используются для обеспечения непротиворечивости данных некоторым заранее заданным условиям при вы­полнении операций над ними. По сути ограничения целостности — это набор правил, используемых при создании конкретной модели данных на базе выбранной СУБД.   
Различают *внутренние*и *явные*ограничения.   
*Ограничения, обусловленные возможностями конкретной СУБД, называют внутренними ограничениями целостности.*Эти ограничения касаются типов хранимых данных (например, "текстовый элемент данных может состоять не более чем из 256 символов" или "запись может содержать не более 100 полей") и допустимых типов связей (например, СУБД может поддерживать только так называемые функциональные связи, т.е. связи типа 1:1, 1:М или М:1). Большинство существующих СУБД поддерживают, прежде всего, именно внутренние ограничения целостности, нарушения которых приводят к некорректности данных и достаточно легко контролируются.   
*Ограничения, обусловленные особенностями хранимых данных о конкретной ПрОбл, называют явными ограничениями целостности.*Эти ограничения также поддерживаются средствами выбранной СУБД, но они формируются обязательно с участием разработчика БД путем определения (программирования) специальных процедур, обеспечивающих непротиворечивость данных. Например, если элемент данных "зачетная книжка" в записи "студент" определен как ключ, он должен быть уникальным, т.е. в БД не должно быть двух записей с одинаковыми значениями ключа. Другой пример: пусть в той же записи предусмотрен элемент "военно-учетная специальность" и для него отведено 6 десятичных цифр. Тогда другие представления этого элемента данных в БД невозможны. С помощью явных ограничений целостности можно организовать как "простой" контроль вводимых данных (прежде всего, на предмет принадлежности элементов данных фиксированному и заранее заданному множеству значений: например, элемент "ученое звание" не должен принимать значение "почетный доцент", если речь идет о российских ученых), так и более сложные процедуры (например, введение значения "профессор" элемента данных "ученое звание" в запись о преподавателе, имеющем возраст 25 лет, должно требовать, по крайней мере, дополнительного подтверждения).   
Разнообразие существующих моделей данных соответствует разнообразию областей применения и предпочтений пользователей.   
В специальной литературе встречается описание довольно большого количества различных моделей данных. Хотя наибольшее распространение получили иерархическая, сетевая и, бесспорно, реляционная модели, вместе с ними следует упомянуть и некоторые другие.   
Используя в качестве классификационного признака особенности логической организации данных, можно привести следующий перечень известных моделей данных: 

* иерархическая;
* сетевая;
* реляционная;
* бинарная;
* семантическая сеть.

**Иерархическая модель данных**

Наиболее давно используемой (можно сказать классической) является модель данных, в основе которой лежит *иерархическая структура типа дерева.*Дерево — орграф, вкаждую вершину которого, кроме первой (корневой), входит только одна дуга, а из любой вершины (кроме конечных) может исходить произвольное число дуг. В иерархической структуре подчиненный элемент данных всегда связан только с одним исходным.   
На рис. 1 показан фрагмент объектной записи в иерархической модели данных. Часто используется также "упорядоченное дерево", в котором значим относительный порядок поддеревьев.   
*Достоинства*такой модели несомненны: простота представления предметной области, наглядность, удобство анализа структур и простота их описания. К *недостаткам*следует отнести сложность добавления новых и удаления существующих типов записей, невозможность отображения отношений, отличающихся от иерархических, громоздкость описания и информационную избыточность.

[](http://mirea.kremlina.ru/uploads/posts/2017-12/1512415023_1.png)

Рис. 1. Фрагмент иерархической модели данных.

Характерные примеры реализации иерархических структур — язык Кобол и СУБД семейства IMS (создана в рамках проекта высадки на Луну – "Аполлон") и System 2000 (S2K). 

**Сетевая модель данных**

В системе баз данных, предложенных CODASYL, за основу была взята сетевая структура. Существенное влияние на разработку этой модели оказали ранние сетевые системы — IDS и Ассоциативный ПЛ/1. Необходимость в процессе получения одного отчета обрабатывать несколько файлов обусловила целесообразность установления перекрестных ссылок между файлами, что, в конце концов, и привело к сетевым структурам.   
Сетевая модель данных основана на представлении информации в виде орграфа, в котором в каждую вершину может входить *произвольное число дуг.*Вершинам графа сопоставлены типы записей, дугам — связи между ними. На рис. 2. представлен пример структуры сетевой модели данных.   
По сравнению с иерархическими сетевые модели обладают рядом существенных *преимуществ*:возможность отображения практически всего многообразия взаимоотношений объектов предметной области, непосредственный доступ к любой вершине сети (без указания других вершин), малая информационная избыточность. Вместе с тем в сетевой модели невозможно достичь полной независимости данных — с ростом объема информации сетевая структура становится весьма сложной для описания и анализа.

[](http://mirea.kremlina.ru/uploads/posts/2017-12/1512415080_2.png)

Рис. 2. Фрагмент сетевой модели данных.

Известно, что применение на практике иерархических и сетевых моделей данных в некоторых случаях требует разработки и сопровождения значительного объема кода приложения, что иногда может стать для информационной системы непосильным бременем. 

**Реляционная модель данных**

В основе реляционной модели данных лежат не графические, а *табличные методы и средства представления данных*и манипулирования ими (рис. 3). В реляционной модели для отображения информации о предметной области используется таблица, называемая "отношением". Строка такой таблицы называется кортежем, столбец — атрибутом. Каждый атрибут может принимать некоторое подмножество значений из определенной области — домена.   
Таблица организации БД позволяет реализовать ее важнейшее преимущество перед другими моделями данных, а именно — возможность использования точных математических методов манипулирования данными, и прежде всего — аппарата реляционной алгебры и исчисления отношений, К другим достоинствам реляционной модели можно отнести наглядность, простоту изменения данных и организации разграничения доступа к ним.   
Основным недостатком реляционной модели данных является информационная избыточность, что ведет к перерасходу ресурсов вычислительных систем. Однако именно реляционная модель данных находит все более широкое применение в практике автоматизации информационного обеспечения профессиональной деятельности.

[](http://mirea.kremlina.ru/uploads/posts/2017-12/1512415104_3.png)

Рис. 3. Фрагмент реляционной модели данных.

Подавляющее большинство СУБД, ориентированных на персональные ЭВМ, являются системами, построенными на основе реляционной модели данных — реляционными СУБД. 

**Бинарная модель данных**

Бинарная модель данных — это графовая модель, в которой *вершины являются представлениями простых однозначных атрибутов, а дуги представлениями бинарных связей между атрибутами*(см. рис. 4).

[http://mirea.kremlina.ru/uploads/posts/2017-12/thumbs/1512415099_4.png](http://mirea.kremlina.ru/uploads/posts/2017-12/1512415099_4.png)

Рис. 4. Пример бинарного отношения.

Бинарная модель не получила особо широкого распространения, но в ряде случаев находит практическое применение. Так, в области искусственного интеллекта уже давно ведутся исследования с целью представления информации в виде бинарных отношений. Рассмотрим триаду (тройку) "объект—атрибут—значение". Триада "Кузнецов—возраст—20" означает, что возраст некоего Кузнецова равен 20 годам. Эта же информация может быть выражена, например, бинарным отношением ВОЗРАСТ. Понятие бинарного отношения положено в основу таких моделей данных, как, например, Data Semantics (автор Абриал) и DIAM II (автор Сенко).   
Бинарные модели данных обладают возможностью представления связки любой сложности (и это их несомненное преимущество), но вместе с тем их ориентация на пользователя недостаточна. 

**Семантическая сеть**

Семантические сети как модели данных были предложены исследователями, работавшими над различными проблемами *искусственного интеллекта*. Так же, как в сетевой и бинарной моделях, базовые структуры семантической сети могут быть представлены графом, множество вершин и дуг которого образует сеть. Однако семантические сети предназначены для представления и систематизации знаний самого общего характера.   
Таким образом, семантической сетью можно считать любую графовую модель (например — помеченный бинарный граф) если изначально четко определено, что обозначают вершины и дуги и как они используются.   
Семантические сети являются богатыми источниками идей моделирования данных, чрезвычайно полезных в плане решения проблемы представления сложных ситуаций. Они могут быть использованы независимо или совместно с идеями, положенными в основу других моделей данных. Их интересной особенностью является то, что расстояние, измеренное на сети (семантическое расстояние или метрика), играет важную роль, определяя близость взаимосвязанных понятий. При этом предусмотрена возможность в явной форме подчеркнуть, что семантическое расстояние велико. Как показано на рис. 5, СПЕЦИАЛЬНОСТЬ соотносится с личностью ПРЕПОДАВАТЕЛЬ, и в то же время ПРЕПОДАВАТЕЛЮ присущ РОСТ. Взаимосвязь личности со специальностью очевидна, однако из этого не обязательно следует взаимосвязь СПЕЦИАЛЬНОСТИ и РОСТА.

[http://mirea.kremlina.ru/uploads/posts/2017-12/thumbs/1512415056_5.png](http://mirea.kremlina.ru/uploads/posts/2017-12/1512415056_5.png)

Рис. 5. Соотношение понятий в семантической сети.

Следует сказать, что моделям данных типа семантической сети при всем присущем им богатстве возможностей при моделировании сложных ситуаций присуща усложненность и некоторая неэкономичность в концептуальном плане.