

Модели данных

Сами по себе данные не обладают определенной структурой, данные становятся информацией тогда, когда пользователь задает им определенную структуру, то есть, осознает их смысловое содержание. Поэтому центральным понятием в области БД является понятие модели.

Модель данных, будучи приложена к конкретным данным, позволяет пользователям и разработчикам трактовать их уже как информацию, то есть сведения, содержащие не только данные, но и взаимосвязь между ними.

СУБД имеют свою **архитектуру**. В процессе разработки и совершенствования СУБД предлагались различные архитектуры, но самой удачной оказалась трехуровневая архитектура, предложенная исследовательской группой ANSI/SPARC американского комитета по стандартизации ANSI (American National Standards Institute). Упрощенная схема архитектуры СУБД приведена на рис. 1.

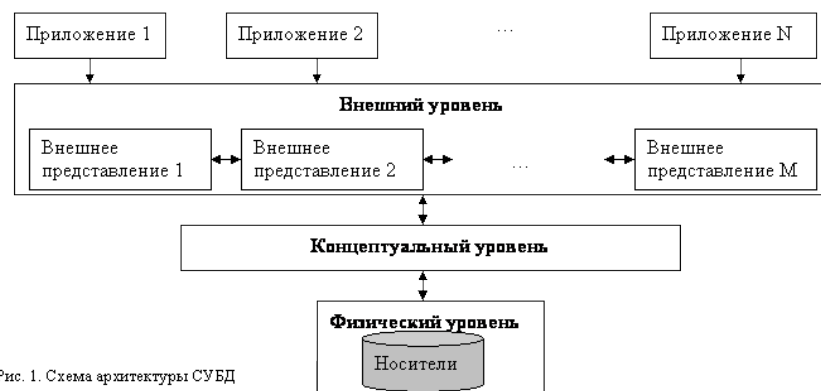


Рис. 1. Схема архитектуры СУБД

Внешний уровень – это уровень пользователя. По сути, это совокупность внешних представлений данных, которые обрабатывают приложения и какими их видит пользователь на экране. Это может быть таблица с отсортированными данными, с примененным фильтром, форма, отчет, результат запроса. Внешние представления взаимосвязаны, т.е. из одного внешнего представления можно получить другое.

Концептуальный уровень – центральный. Здесь БД представлена в наиболее общем виде, который объединяет данные, используемые всеми приложениями. Т.е. это обобщенная *модель* предметной области, для которой созданы БД. Можно сказать, что концептуальный уровень формируется при создании таблиц (определение их полей, типов, свойств), связей, а так же при заполнении таблиц.

Физический уровень – собственно данные, расположенные на внешних носителях.

Классификация моделей данных

Ядром любой БД является модель данных.

Модель данных – это совокупность структур данных и операций их обработки.

Т.к. СУБД имеет 3-х уровневую архитектуру, то понятие модели данных связано с каждым уровнем.

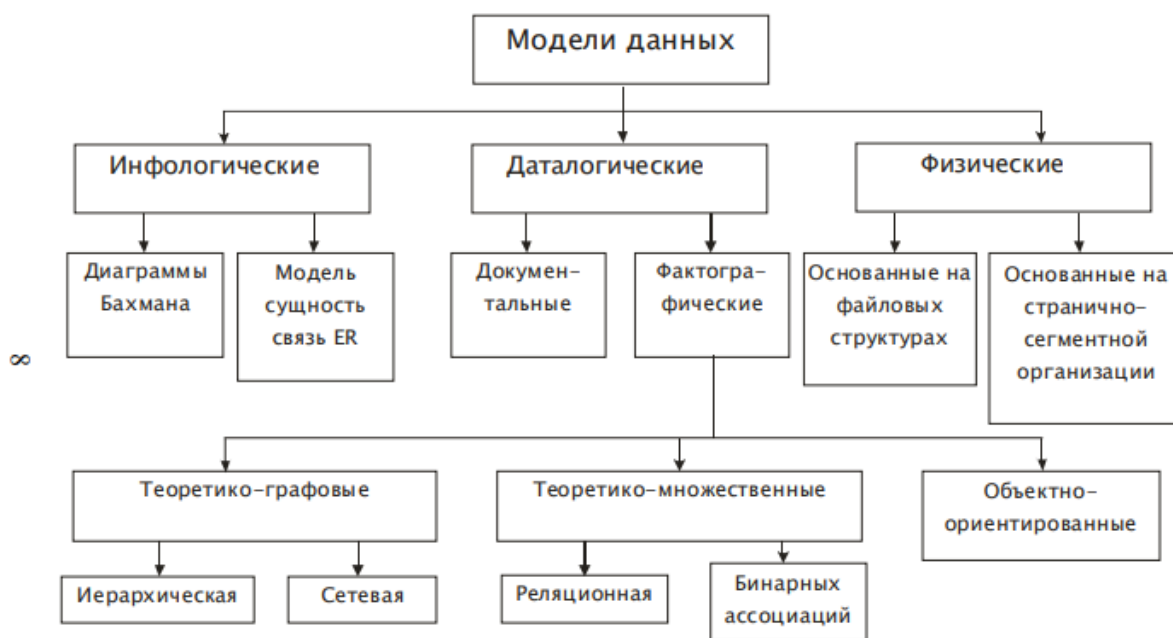
Физическая модель данных связана с организацией внешней памяти и структур хранения, используемых в данной операционной среде.

На *концептуальном* уровне модели данных наиболее важны для разработчиков БД, т.к. именно ими определяется тип СУБД.

Для *внешнего* уровня отдельных моделей данных нет, они лишь являются подсхемами концептуальных моделей данных.

Кроме моделей данных, соответствующих трем уровням архитектуры СУБД, существуют *предшествующие* им, не связанные с компьютерной реализацией. Они служат переходным звеном от реального мира к БД. Это класс инфологических (семантических) моделей.

Общая классификация моделей данных приведена на рисунке



Инфологические (семантические) модели данных используются на ранних стадиях проектирования БД.

Даталогические модели данных уже поддерживаются конкретной СУБД.

Физические модели данных связаны с организацией данных на носителях.

Документальные модели данных соответствуют слабоструктурированной информации, ориентированной на свободные форматы документов на естественном языке.

Модели данных, *ориентированные на формат документа*, связаны со стандартным общим языком разметки SGML (Standart Generaliset Markup

Language), а также HTML, предназначенным для управления процессом вывода содержимого документа на экран.

Дескрипторные модели данных – самые простые, широко использовались раньше. В них каждому документу соответствует дескриптор – описатель, который имеет жёсткую структуру и описывает документ в соответствии с заранее определенными характеристиками.

Тезаурусные модели данных основаны на принципе организации словарей. Содержат языковые конструкции и принципы их взаимодействия в заданной грамматике. Эти модели используются, например, в системах-переводчиках.

Объектно-ориентированная модель перекликается с семантическими моделями данных. Принципы похожи на принципы объектно-ориентированных языков программирования. Структура таких моделей графически представима в виде дерева, узлами которого являются объекты. Свойства объектов описываются типом.

Объекты **иерархической** модели данных связаны иерархическими отношениями и образуют ориентированный граф. Основные понятия иерархических структур: уровень, узел (совокупность свойств данных, описывающих объект), связь.

В **сетевой** модели данных при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом.

В **реляционной** модели данных данные представлены только в виде таблиц.

Инфологические модели используются на ранних стадиях проектирования для описания структур данных в процессе разработки приложения, а даталогические модели уже поддерживаются конкретной СУБД.

В семидесятых годах было предложено несколько моделей данных, названных семантическими. В настоящий момент наиболее широкое распространение получила модель Чена (Chen), предложенная в 1976 году, которая получила название «сущность-связь» или ER-модель (“Entity Relationship”).

Модель «сущность-связь»

В основе этой модели лежат следующие базовые понятия. Сущность – это класс однотипных объектов. Сущность имеет уникальное имя. Предполагается, что в системе существует множество экземпляров данной сущности.

Пример.

Сущность – «студент», «преподаватель». Экземпляры сущности «студент» – студенты вуза (Иванов, Петров, Степанов и пр.), «преподаватель» - преподаватели вуза (Захаров И.С., Бронштейн И.А. и пр.).

Объект имеет свой набор атрибутов – свойств объекта.

Пример.

Для сущности «студент» можно выделить такие атрибуты как «номер зачетной книжки», «фамилия», «имя», «отчество», «номер групп». Для сущности «преподаватель» – «табельный номер», «ФИО», «кафедра».

Выделение тех или иных сущностей и их атрибутов определяется требованиями, накладываемыми на содержание базы данных, то есть задается теми параметрами, которые необходимо учитывать при проектировании базы данных.

Атрибут, однозначно идентифицирующий конкретный экземпляр сущности, называется ключевым.

Пример.

Для сущности «студент» в качестве ключевого атрибута можно обозначить «номер зачетной книжки», поскольку для всех студентов данного вуза номера зачетных книжек будут различны.

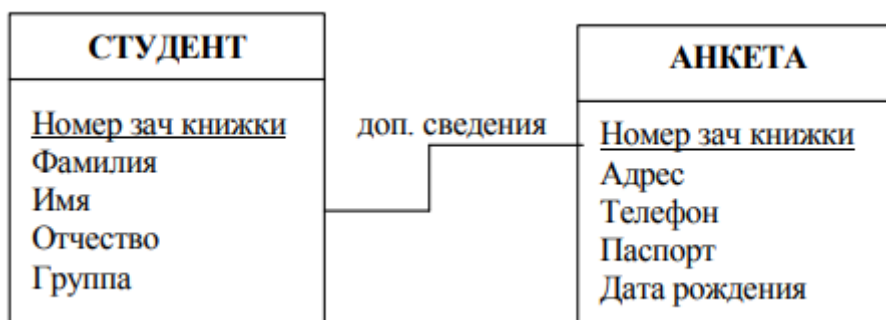
Для сущности «преподаватель» можно выделить «табельный номер».

Между сущностями могут быть установлены связи. Связи – это бинарные ассоциации, показывающие, каким образом сущности соотносятся или взаимодействуют между собой. По множественности связи делятся на три типа:

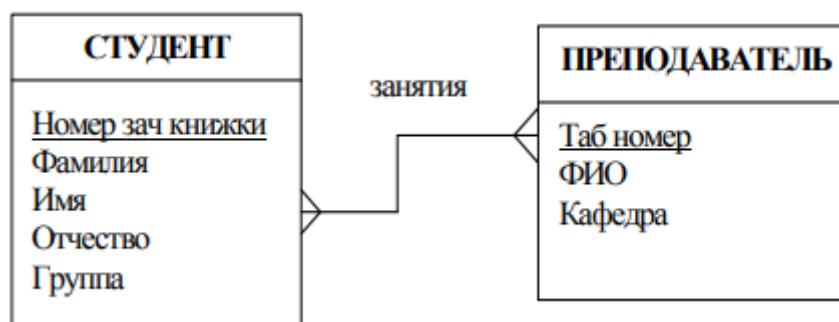
- «один-ко-многим» (1:M) - один экземпляр одной сущности может быть связан с несколькими экземплярами другой сущности;



- «один-к-одному» (1:1) - один экземпляр одной сущности связан только с одним экземпляром другой сущности;



- «многие-ко-многим» (M:M) - один экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности или один экземпляр второй сущности может быть связан несколькими экземплярами первой сущности.



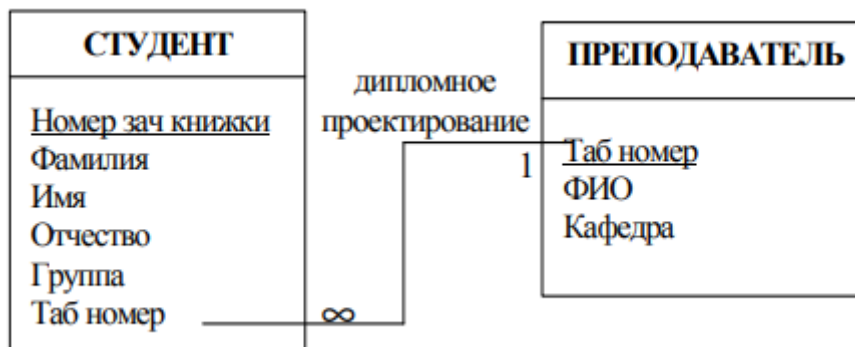
Сама по себе ER-модель представляет графическое изображение связанных друг с другом прямоугольников, в верхней части которых записываются имена сущностей, а ниже перечисляются атрибуты. Ключевой атрибут выделяется подчеркиванием и, как правило, располагается на первом месте

Реляционная модель данных

Теоретической основой этой модели стала теория отношений и реляционная алгебра. Американский математик Э.Ф.Кодд в 1970 году сформулировал основные понятия реляционной модели. Основной структурой данных в модели является отношение, именно поэтому модель получила название реляционной (от англ. relation - отношение). Отношение представляет собой таблицу. Заголовок таблицы играет ту же роль, что и имя сущности. Столбцы таблицы соответствуют атрибутам сущности и в реляционной модели называются атрибутами (вхождениями доменов), строки – экземплярам сущности и называются кортежами.

Реляционная модель представляет базу данных в виде множества взаимосвязанных отношений. Между отношениями существуют связи. Связь любого типа может быть обязательной, если в данной связи должен участвовать каждый экземпляр сущности, и необязательной. Связь может быть обязательной с одной стороны и необязательной с другой. В каждой связи одно отношение может выступать как основное, а другое отношение выступает в роли подчиненного. Это означает, что один кортеж основного отношения может быть связан с несколькими кортежами подчиненного отношения. Для поддержания связей оба отношения должны содержать наборы атрибутов, по которым они связаны. В подчиненном отношении для моделирования связи должен присутствовать атрибут, соответствующий первичному ключу основного отношения. Такой атрибут в подчиненном отношении является вторичным ключом и называется внешний ключ.

Пример.



Связь «дипломное проектирование» с точки зрения реляционной модели. Основным отношением будет «преподаватель», подчиненным – «студент». Первичным ключом основного отношения – «таб номер» отношения «преподаватель», внешним ключом – «таб номер» отношения «студент» (которое уже не является 15 ключевым!). Этот атрибут отсутствовал в ER модели у сущности «студент». Он нужен для организации связи в реляционной модели, так как в ней, в отличие от ER модели, связь осуществляется не между таблицами целиком, а между конкретными атрибутами отношений. Как правило, эти атрибуты имеют одинаковые названия («таб номер» - «таб номер»).

Одним из основополагающих понятий в технологии БД является понятие целостности. **Под целостностью понимают соответствие информационной модели предметной области, хранимой в БД, объектам реального мира и их взаимодействиям в каждый момент времени.** Поддержка целостности включает в себя три аспекта:

- структурная целостность – допустимыми являются только данные, представленные в виде отношений реляционной модели;
- языковая целостность - допускаются языки манипулирования данными высокого уровня (не ниже стандарта SQL);
- ссылочная целостность - обеспечение поддержки непротиворечивого состояния БД в процессе модификации данных при выполнении операций добавления или удаления.

Кроме указанных ограничений целостности, вводится понятие семантической поддержки целостности. Структурная, языковая и ссылочная целостность определяют правила работы СУБД с реляционными структурами данных. Но эти аспекты никак не касаются содержания БД. Принципы семантической поддержки целостности позволяют накладывать ограничения на содержание БД (например, в библиотеке записаны читатели не моложе 16 лет).