

ТЕМА: Основные принципы проектирования информационных систем.

Перед началом детального обсуждения способов проектирования баз данных необходимо отметить, что любая база данных является составной частью некой информационной системы (ИС), которая подразумевает не только хранение данных, но и их обработку. Поэтому, проектированию данных всегда сопутствует (а чаще предшествует) проектирование алгоритмов их использования. Здесь мы рассмотрим все этапы проектирования информационной системы: от функционального моделирования предметной области, до построения структуры реляционной базы данных.

Проектирование базы данных (БД) – одна из наиболее сложных и ответственных задач, связанных с созданием информационной системы (ИС). В результате её решения должны быть определены содержание БД, эффективный для всех её будущих пользователей способ организации данных и инструментальные средства управления данными.

Основная цель процесса проектирования БД состоит в получении такого проекта, который удовлетворяет следующим требованиям:

1. Корректность схемы БД, т.е. база должна наиболее полно отображать моделируемую предметную область, т.е. каждому объекту предметной области должны соответствовать данные в памяти ЭВМ, а каждому процессу – адекватные процедуры обработки данных.
2. Обеспечение ограничений (на объёмы внешней и оперативной памяти и другие ресурсы вычислительной системы).
3. Эффективность функционирования (соблюдение ограничений на время реакции системы на запрос и обновление данных).
4. Защита данных (от аппаратных и программных сбоев и несанкционированного доступа).
5. Простота и удобство эксплуатации.
6. Гибкость, т.е. возможность развития и адаптации к изменениям предметной области и/или требований пользователей.

Этапы проектирования данных

Предметная область - часть реального мира, подлежащая изучению с целью организации управления и, в конечном счете, автоматизации. Предметная область представляется множеством *фрагментов*, например, предприятие - цехами, дирекцией, бухгалтерией и т.д. Каждый фрагмент предметной области характеризуется множеством *объектов* и *процессов*, использующих объекты, а также множеством *пользователей*, характеризующихся различными взглядами на предметную область.

В теории проектирования информационных систем предметную область (или, если угодно, весь реальный мир в целом) принято рассматривать в виде трех представлений:

1. представление предметной области в том виде, как она реально существует
2. как ее воспринимает человек (имеется в виду проектировщик базы данных)
3. как она может быть описана с помощью символов.

Данные, используемые для описания предметной области, представляются в виде трехуровневой схемы (так называемая модель ANSI/SPARC):



Внешнее представление (внешняя схема) данных является совокупностью требований к данным со стороны некоторой конкретной функции, выполняемой пользователем.

Концептуальная схема является полной совокупностью всех требований к данным, полученной из пользовательских представлений о реальном мире.

Внутренняя схема - это сама база данных.

Отсюда вытекают основные этапы, на которые разбивается процесс проектирования базы данных информационной системы:

1. **Концептуальное проектирование**(инфологическое) - сбор, анализ и редактирование требований к данным. Для этого осуществляются следующие мероприятия:

- обследование предметной области, изучение ее информационной структуры
- выявление всех фрагментов, каждый из которых характеризуется пользовательским представлением, информационными объектами и связями между ними, процессами над информационными объектами
- моделирование и интеграция (соединение) всех представлений

По окончании данного этапа получаем концептуальную модель, инвариантную (независимую от) к структуре базы данных. Часто она представляется в виде модели "сущность-связь".

Концептуальная модель – модель, представляющая объекты и их взаимосвязи без указания способов их физического хранения. При проектировании концептуальной модели структурируют данные и выявляют взаимосвязи между ними, без рассмотрения особенностей реализации и вопросов эффективности обработки.

2. **Логическое проектирование** - преобразование требований к данным в структуры данных. На выходе получаем СУБД-ориентированную структуру базы данных и спецификации прикладных программ. На этом этапе часто моделируют базы данных применительно к различным СУБД и проводят сравнительный анализ моделей.

Логическая модель – модель, отражающая логические связи между элементами данных вне зависимости от их содержания и среды хранения.

Логическая модель может быть реляционной, иерархической или сетевой.

Результатом выполнения этого этапа являются схемы БД концептуального и внешнего уровней архитектуры, составленные на языках определения данных (DDL, Data Definition Language), поддерживаемых данной СУБД.

3. **Физическое проектирование** - определение особенностей хранения данных, методов доступа и т.д.

Внутренняя модель системы – это физическая модель, определяющая размещение данных, методы доступа и технику индексирования.

Различие уровней представления данных на каждом этапе проектирования представлено в следующей таблице:

КОНЦЕПТУАЛЬНЫЙ УРОВЕНЬ <ul style="list-style-type: none">• сущности• атрибуты• связи	Представление аналитика
ЛОГИЧЕСКИЙ УРОВЕНЬ <ul style="list-style-type: none">• записи• элементы данных• связи между записями	Представление программиста
ФИЗИЧЕСКИЙ УРОВЕНЬ <ul style="list-style-type: none">• группирование данных• индексы• методы доступа	Представление администратора

Основные подходы при создании концептуальной модели предметной области.

1. Функциональный подход к проектированию БД

Этот метод реализует принцип "от задач" и применяется тогда, когда известны функции некоторой группы лиц и/или комплекса задач, для обслуживания информационных потребностей которых создаётся рассматриваемая БД.

2. Предметный подход к проектированию БД

Предметный подход к проектированию БД применяется в тех случаях, когда у разработчиков есть чёткое представление о самой ПО и о том, какую именно информацию они хотели бы хранить в БД, а структура запросов не определена или определена не полностью. Тогда основное внимание уделяется исследованию ПО и наиболее адекватному её отображению в БД с учётом самого широкого спектра информационных запросов к ней.

3. Проектирование с использованием метода "сущность-связь"

Метод "сущность-связь" (entity-relation, ER-method) является комбинацией двух предыдущих и обладает достоинствами обоих. Этап концептуального (инфологического) проектирования начинается с моделирования ПО. Проектировщик разбивает её на ряд локальных областей, каждая из которых (в идеале) включает в себя информацию, достаточную для обеспечения запросов отдельной группы будущих пользователей или решения отдельной задачи (подзадачи). Каждое локальное представление моделируется отдельно, затем они объединяются.

Выбор локального представления зависит от масштабов ПО. Обычно она разбивается на локальные области таким образом, чтобы каждая из них соответствовала отдельному внешнему приложению и содержала 6-7 сущностей.

Сущность – это объект, о котором в системе будет накапливаться информация. Сущности бывают как физически существующие (например, СОТРУДНИК или АВТОМОБИЛЬ), так и абстрактные (например, ЭКЗАМЕН или ДИАГНОЗ).

Для сущностей различают тип сущности и экземпляр. Тип характеризуется именем и списком свойств, а экземпляр – конкретными значениями свойств.

Типы сущностей можно классифицировать как сильные и слабые. Сильные сущности существуют сами по себе, а существование слабых сущностей зависит от существования сильных. Например, читатель библиотеки – сильная сущность, а абонемент этого читателя – слабая, которая зависит от наличия соответствующего читателя. Слабые сущности называют подчинёнными (дочерними), а сильные – базовыми (основными, родительскими).

Для каждой сущности выбираются свойства (атрибуты). Различают:

1. *Идентифицирующие и описательные атрибуты*. Идентифицирующие атрибуты имеют уникальное значение для сущностей данного типа и являются *потенциальными ключами*. Они позволяют однозначно распознавать экземпляры сущности. Из потенциальных ключей выбирается один первичный ключ (ПК). В качестве ПК обычно выбирается потенциальный ключ, по которому чаще происходит обращение к экземплярам записи. Кроме того, ПК должен включать в свой состав минимально необходимое для идентификации количество атрибутов. Остальные атрибуты называются описательными и включают в себе интересующие свойства сущности.

2. *Составные и простые атрибуты*. Простой атрибут состоит из одного компонента, его значение неделимо. Составной атрибут является комбинацией нескольких компонентов, возможно, принадлежащих разным типам данных (например, ФИО или адрес). Решение о том, использовать составной атрибут или разбивать его на компоненты, зависит от характера его обработки и формата пользовательского представления этого атрибута.

3. *Однозначные и многозначные атрибуты*, т.е. могут иметь, соответственно, одно или много значений для каждого экземпляра сущности.

4. *Основные и производные атрибуты*. Значение основного атрибута не зависит от других атрибутов. Значение производного атрибута вычисляется на основе значений других атрибутов (например, возраст студента вычисляется на основе даты его рождения и текущей даты).

Спецификация атрибута состоит из его названия, указания типа данных и описания ограничений целостности – множества значений (или домена), которые может принимать данный атрибут.

Далее осуществляется спецификация связей внутри локального представления. Связи могут иметь различный содержательный смысл (семантику). Различают связи типа "сущность-сущность", "сущность-атрибут" и "атрибут-атрибут" для отношений между атрибутами, которые характеризуют одну и ту же сущность или одну и ту же связь типа "сущность-сущность".

Каждая связь характеризуется именем, обязательностью, типом и степенью. Различают **факультативные** и **обязательные** связи. Если вновь порождённый объект одного типа оказывается по необходимости связанным с объектом другого типа, то между этими типами объектов существует обязательная связь (обозначается двойной линией). Иначе связь является факультативной.

По типу различают множественные связи "один к одному" (1:1), "один ко многим" (1:n) и "многие ко многим" (m:n). ER–диаграмма, содержащая различные типы связей, приведена на рис. 1. Обратите внимание, что обязательные связи на рис. 1 выделены двойной линией.

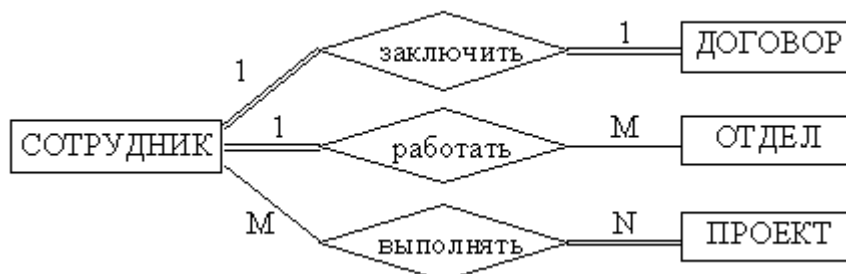


Рис.1. ER–диаграмма с примерами типов множественных связей

Степень связи определяется количеством сущностей, которые охвачены данной связью. Пример бинарной связи – связь между отделом и сотрудниками, которые в нём работают. Примером тернарной связи является связь типа экзамен между сущностями ДИСЦИПЛИНА, СТУДЕНТ, ПРЕПОДАВАТЕЛЬ. Из последнего примера видно, что связь также может иметь атрибуты (в данном случае это Дата проведения и Оценка). Пример ER–диаграммы с указанием сущностей, их атрибутов и связей приведен на рис. 2.

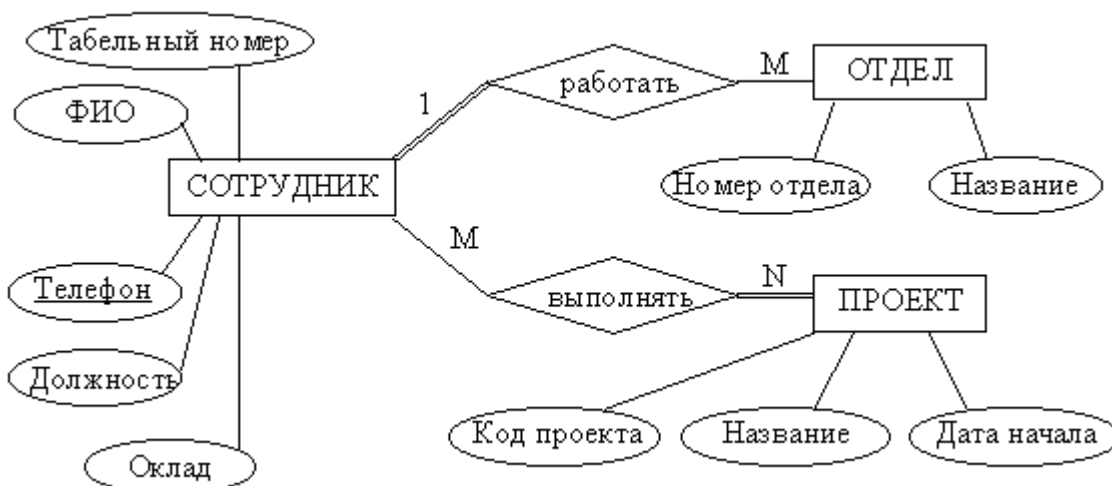


Рис.2. Пример ER–диаграммы с однозначными и многозначными атрибутами (подчеркнутые)

После того, как созданы локальные представления, выполняется их объединение. При небольшом количестве локальных областей (не более пяти) они объединяются за один шаг. В противном случае обычно выполняют бинарное объединение в несколько этапов.

При объединении проектировщик может формировать конструкции, производные по отношению к тем, которые были использованы в локальных представлениях. Такой подход может преследовать следующие цели:

- объединение в единое целое фрагментарных представлений о различных свойствах одного и того же объекта;
- введение абстрактных понятий, удобных для решения задач системы, установление их связи с конкретными понятиями, использованными в модели;
- образование классов и подклассов подобных объектов (например, класс "изделие" и подклассы типов изделий, производимых на предприятии).

На этапе объединения необходимо выявить и устранить все противоречия. Например, одинаковые названия семантически различных объектов или связей или несогласованные

ограничения целостности на одни и те же атрибуты в разных приложениях. Устранение противоречий вызывает необходимость возврата к этапу моделирования локальных представлений с целью внесения в них соответствующих изменений.

По завершении объединения результаты проектирования являют собой концептуальную (инфологическую) модель предметной области. Модели локальных представлений – это внешние инфологические модели.

Замечание: после создания концептуальной модели необходимо определить требования к операционной обстановке, в которой будет функционировать информационная система. На этом этапе производится:

- 1) оценка требований к вычислительным ресурсам, необходимым для функционирования системы;
- 2) определение типа и конфигурации конкретной ЭВМ;
- 3) выбор типа и версии операционной системы.

Объём вычислительных ресурсов зависит от предполагаемого объёма проектируемой базы данных и от интенсивности их использования. Если БД будет работать в многопользовательском режиме, то требуется подключение её к сети и наличие соответствующей многозадачной операционной системы.

Далее необходимо осуществить выбор системы управления базой данных (СУБД) и других инструментальных программных средств.

Физическое проектирование БД

Этап физического проектирования заключается в увязке логической структуры БД и физической среды хранения с целью наиболее эффективного размещения данных, т.е. отображении логической структуры БД в структуру хранения. Решается вопрос размещения хранимых данных в пространстве памяти, выбора эффективных методов доступа к различным компонентам "физической" БД. Результаты этого этапа документируются в форме схемы хранения на языке определения данных (DDL). Принятые на этом этапе решения оказывают определяющее влияние на производительность системы.

Одной из важнейших составляющих проекта базы данных является разработка средств защиты БД. Защита данных имеет два аспекта: защита от сбоев и защита от несанкционированного доступа. Для защиты от сбоев разрабатывается стратегия резервного копирования. Для защиты от несанкционированного доступа каждому пользователю доступ к данным предоставляется только в соответствии с его правами доступа.

Особенности проектирования реляционной базы данных

Проектирование реляционной базы данных проходит в том же порядке, что и проектирование БД других моделей данных, но имеет свои особенности.

Проектирование схемы БД должно решать задачи минимизации дублирования данных и упрощения процедур их обработки и обновления. При неправильно спроектированной схеме БД могут возникнуть аномалии модификации данных. Они обусловлены отсутствием средств явного представления типов множественных связей между объектами ПО и неразвитостью средств описания ограничений целостности на уровне модели данных.

Для решения подобных проблем проводится нормализация отношений.