

ГЛАВА 7. ТЕНДЕНЦИИ РАЗВИТИЯ БАЗ ДАННЫХ

7.1. Объектно-ориентированные базы данных

В начале 90-х годов XX века начались активные попытки по внедрению объектно-ориентированных технологий в отрасль проектирования и разработки баз данных. Бытовала точка зрения о том, что соответствующие технологии быстро вытеснят все остальные, так же как и во многих других программистских отраслях, но ничего подобного не произошло.

Объектно-ориентированное программирование

Рассмотрим термин «объектно-ориентированное программирование». Заметим, что это термин, принятый преимущественно в российской литературе. В западной литературе [2] под этим понимается сразу три аспекта:

- Объектно-ориентированный анализ – ООА, *object-oriented analysis*. Объектно-ориентированный анализ – это методология, при которой требования к системе воспринимаются с точки зрения классов и объектов, выявленных в предметной области.
- Объектно-ориентированное проектирование – ООД, *object-oriented design*. Объектно-ориентированное проектирование – это методология проектирования, соединяющая в себе процесс объектной декомпозиции и приемы представления логической и физической, а также статической и динамической моделей проектируемой системы.
- Объектно-ориентированное программирование – ООП, *object-oriented programming*. Объектно-ориентированное программирование – это методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.

Здесь и далее по тексту условимся не отступать от традиций и понимать под объектно-ориентированным программированием (ООП) сразу три указанных выше аспекта.

Итак, основой объектно-ориентированной технологии является так называемая объектная модель, которая возникает как результат объектно-ориентированной декомпозиции. Она выделяет основные абст-

рации предметной области, определяет классы абстракций и выясняет, какими данными (атрибутами) описывается каждая абстракция, какую функциональность эти абстракции должны обеспечивать. В отличие от традиционных технологий программирования объектно-ориентированная технология представляет программу как совокупность классов и объектов, взаимодействующих друг с другом.

Объект – конкретная материализация абстракции; сущность с хорошо определенными границами, в которой инкапсулированы состояние и поведение.

Объект ООП – инкапсулированная структура, имеющая атрибуты и методы.

Термин «инкапсулированная структура» означает, что объект является самодостаточным, программы, внешние по отношению к объекту, ничего «не знают» о его структуре и такое «знание» им не требуется. «Внешний» вид объекта называется его интерфейсом.

В таком понимании объект – это черный ящик, нам неизвестно, что у него внутри, мы лишь можем вызвать его методы и только через них взаимодействовать с ним. Кроме этого, объекты могут принадлежать иерархии «от общего к частному», которая реализуется путем наследования. Инкапсулированные состояния объекта могут быть как простыми типами данных, так и другими объектами, или даже массивами объектов. Каждый объект содержит определенную совокупность методов, классы взаимодействуют друг с другом посредством механизма сообщений. Объекты идентифицируются с помощью специальных указателей – дескрипторов. Методы объектов ООП представляют собой последовательности инструкций, выполняемых объектом. Например, у объекта может быть метод, отображающий данный объект, создающий данный объект и изменяющий его.

Предметная область моделируется как множество классов взаимодействующих объектов. Объект характеризуется набором свойств, которые являются как бы его пассивными характеристиками, и набором методов работы с этим объектом. Работать с объектом можно только с использованием его методов. Атрибуты объекта могут принимать множество допустимых значений, набор конкретных значений атрибутов определяет состояние объекта. Используя методы работы с объектом можно изменять значение его атрибутов и тем самым как бы изменить состояние самого объекта. Множество объектов с одним и тем же набором атрибутов и методов образует класс объектов. Класс,

объекты которого могут служить значениями атрибута объектов другого класса, называется доменом этого атрибута.

К числу основных идей объектно-ориентированной технологии, как правило, относят [2]: абстрагирование, инкапсуляцию, модульность, иерархичность, типизацию, полиморфизм, наследование.

Инкапсуляция ограничивает область видимости имени атрибута пределами того объекта, в котором оно определено. Смысл этого атрибута будет определяться тем объектом, в котором оно инкапсулировано.

Полиморфизм – способность одного и того же программного кода работать с разнообразными данными. Другими словами, он допускает возможность в объектах разных типов иметь методы (процедуры или функции) с одинаковыми именами. Во время выполнения объектной программы одни и те же методы оперируют с разными объектами в зависимости от типа аргумента.

Наследование. Допускается порождение нового класса на основе уже существующего класса, и этот процесс называется наследованием. В этом случае новый класс, называемый подклассом существующего класса, наследует все атрибуты и методы класса. В подклассе, кроме того, могут быть определены дополнительные атрибуты и методы. Различают случаи простого и множественного наследования. В первом случае подкласс может определяться только на основе одного класса, во втором случае – на основе нескольких классов. Набор классов образует иерархическую структуру.

Объектно-ориентированные базы данных

К настоящему моменту терминология еще не устоялась, существует много разных определений и трактовок. Представляется, что объектно-ориентированная база данных (ООБД) – база данных, основанная на принципах объектно-ориентированной технологии. К основным описательным моментам, связанным с ООБД, в литературе [8] относят:

- объекты (в ООБД любая сущность – объект и обрабатывается как объект); отметим, что здесь используется понятие «объект» объектно-ориентированного программирования, которое отличается от понятия «объект», рассматриваемого в главе 2;
- классы (понятие «тип данных» из реляционной модели заменяется понятиями «класс» и «подкласс»);
- наследование (классы образуют иерархию наследования, заимствуя свойства друг друга);

- атрибуты (характеристики объекта моделируются его атрибутами);
- сообщения и методы (каждый класс имеет определенную совокупность методов, классы взаимодействуют друг с другом посредством механизма сообщений);
- инкапсуляция (внутренняя структура объектов скрыта);
- идентификаторы объектов – дескрипторы.

Система управления объектно-ориентированной базой данных называется объектно-ориентированной СУБД (ООСУБД). Цель ООСУБД – обеспечение постоянного хранения объектов, причем в отличие от традиционной СУБД ООСУБД должна хранить в составе объекта данные и программы.

Поскольку каждый объект данного класса имеет один и тот же набор методов, методы сохраняются только один раз – как методы класса (данные каждого экземпляра объекта хранятся отдельно).

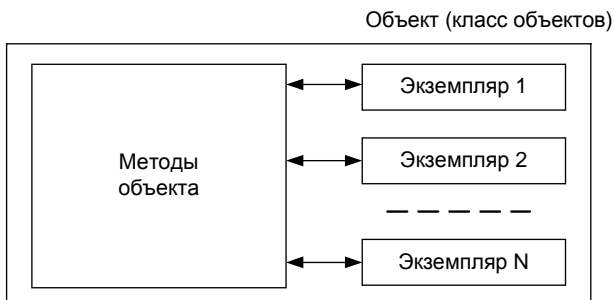


Рис. 54. Схема представления объекта

Используя наследование всем объектам ПОДРАЗДЕЛЕНИЕ на рис. 55 можно приписать свойство объекта-родителя (ФАКУЛЬТЕТ) – название факультета, номер факультета.

Выделим основные положительные черты ООБД по сравнению с реляционной базой данных (РБД).

Класс объектов ФАКУЛЬТЕТ

МЕТОДЫ	АТРИБУТЫ
Создать объект	Название
Модифицировать	Номер
Удалить	Декан
Вывести на экран	Подразделение - класс
Выдать значения атрибутов _____ у объекта _____	

Класс объектов ПОДРАЗДЕЛЕНИЕ

МЕТОДЫ	АТРИБУТЫ
Создать объект	Название
Модифицировать	Зав. кафедрой
Удалить	Декан
Вывести на экран	Сотрудник - класс
Выдать значения атрибутов _____ у объекта _____	

Рис. 55. Фрагменты представления классов объектов

Сравнивая объектно-ориентированный и реляционный подходы к БД, можно отметить следующие особенности. В реляционных БД реальные объекты представляются как структуры, состоящие из набора элементарных типов данных. Такое представление имеет понятную интерпретацию – строка в плоской таблице. В том случае когда специфика предметной области позволяет работать с такого рода приближением реальных объектов, РБД отлично справляются со своей задачей. Довольно часто реляционная модель и ее способ описания предметной области в виде набора плоских таблиц не отражают внутренней структуры для многих предметных областей, являются искусственными и становятся совершенно непонятными при увеличении количества таблиц. Основная причина несостоятельности реляционного подхода заключается в слишком сильной абстракции реального объекта, что ведет к потере семантики.

В отличие от реляционных баз данных объектно-ориентированные базы данных обладают простой и естественной связью с предметной областью, представляя ее структуру и состав, что облегчает проектирование и положительно сказывается на понимании принципов функционирования программ. Так, в сложных неоднородных предметных областях использование ООБД должно действительно упростить процесс проектирования и разработки. Опыт одного из авторов по созданию ООБД для моделирования физических процессов в прочностной механике показывает: все удобство применения соответствующей технологии – именно в простоте и естественности описания предметной области. Однако нельзя не остановиться и на недостатках. К сожалению, в ООБД существуют свои проблемы. Например, там отсутствует мощная математическая база, лежащая в основе реляционной модели. Кроме того, в ООБД обычно обходятся без интерпретируемых языков запросов, таких, как SQL, что автоматически приводит к трудностям при работе с данными.

ООСУБД отличаются от реляционных СУБД тем, что программный интерфейс создания приложения либо очень слаб, либо вообще отсутствует. Это означает, что при написании приложения на основе ООБД не существует конструкторов (не считая, например, конструктора создания списка полей в объекте, который поставляется вместе с ООСУБД ObjectStore). Поэтому разработчик создает приложения на одном из языков.

Один из ведущих аналитиков в этой области С. Кузнецов характеризует состояние дел в данной отрасли следующим образом: «Обсуждение затрудняло то обстоятельство, что в области ООБД отсутствовал общепринятый набор терминов и определений (эта ситуация сохраняется и сегодня). Сторонники ООБД выдвигали два основных аргумента. Во-первых, они утверждали, что ООБД представляют хорошую основу расширяемых СУБД. Во-вторых, идея наследования может быть упрощена до уровня, понимаемого разработчикам приложений. У противников тоже имелись два аргумента: 1) ООБД по своей природе не слишком пригодны для непредвиденных запросов и поэтому не подходят для систем поддержки принятия решений; 2) типичный для ООБД навигационный интерфейс доступа возвращает нас во времена CODASYL. Однако реальной проблемой являлось то, что термин «объектная ориентированность» относился к слишком большому числу различных вещей. В общем-то за десять лет почти ничего не изменилось. Существует ряд коммерческих ООСУБД, они довольно актив-

но используются, но по-прежнему общего согласия по поводу понятий и терминов нет» [41].

Для перехода к объектно-ориентированным БД стандарт объектно-го программирования был дополнен стандартизованными средствами доступа к базам данных (стандарт ODMG 93; Object Database Management Group – группа управления объектно-ориентированными базами данных). К настоящему времени этот стандарт не реализован. Состояние проблемы подробно описано также в работах [10, 18, 27, 42 и др.]. Мы здесь лишь подведем краткие итоги:

- ООБД используются, но пока не стали реальной альтернативой реляционным базам данных;
- объектно-ориентированные возможности появляются в ведущих современных СУБД, таких, как, например, Oracle;
- предпринимаются попытки внесения изменений в стандарты языка SQL с целью его частичной адаптации к ООБД. Так, новый стандарт SQL-3 включает большой раздел, посвященный этому вопросу.

Объектно-реляционные базы данных

Общая и вполне разумная идея заключается, с нашей точки зрения, в применении гибких подходов, позволяющих не ограничивать собственные возможности догматическими положениями, а, напротив, применяя различные приемы, адаптироваться к любой возникшей ситуации.

Именно эту идею и проповедают разработчики объектно-реляционных баз данных. Так, соответствующие базы данных соединяют в себе лучшие качества реляционных и объектно-ориентированных баз данных. Рассмотрим кратко основные элементы, осуществляющие объектные расширения в реляционных базах данных, уже используемые на сегодняшний день:

- **Хранение больших объемов данных.** Наряду с теми данными, которые хранились в БД традиционно, современные объектно-реляционные базы данных позволяют хранить в столбцах таблицы картинки, видеоролики и другие большие документы.
- **Списки в столбцах.** Более того, объектно-реляционные базы данных позволяют хранить в столбцах целые списковые структуры.
- **Пользовательские расширения.** В объектно-реляционных базах данных пользователи имеют возможность вмешиваться в изна-

чально предоставляемый СУБД инструментарий, создавая, в частности, новые пользовательские типы данных.

- **Хранимые процедуры.** В определенном смысле хранимые процедуры также являются объектным расширением, осуществляя необходимые пользователю воздействия на данные (стандартный для ООП процедурный подход).

В качестве примеров реализации такого подхода можно указать доработку СУБД DB2, произведенную фирмой IBM, а также добавление объектной надстройки над реляционным ядром систем Oracle, произведенное фирмой Oracle.

7.2. Распределенные базы данных

База данных – интегрированная совокупность данных, с которой работают много пользователей. Изложение всех предыдущих разделов предполагало единую базу данных, размещаемую на одном компьютере. Напомним основные принципы, положенные в основу теории баз данных:

- централизованное хранение данных;
- централизованное обслуживание данных (ввод, корректировка, чтение, контроль целостности).

Заметим, что базы данных появились в период господства больших ЭВМ. База данных велась на одной ЭВМ, все пользователи работали именно на ЭВМ (возможные режимы работы описаны в разделе 1.6). Других вариантов использования вычислительной техники в то время просто не существовало.

Если проанализировать работу пользователей с данными в компаниях, организациях, предприятиях в «докомпьютерное» время, то нетрудно заметить, что на отдельных участках пользователи работали со «своими» данными (осуществляли сбор определенных данных, их хранение, обработку, передачу обработанных данных на другие участки или уровни управления).

У такой технологии были существенные недостатки, которые уже отмечались в предыдущих разделах: дублирование некоторых данных, отсутствие возможности сравнительного анализа данных всех участков. Однако у этой технологии были и существенные достоинства: данные вводились и хранились в местах их порождения; с этими данными работал пользователь, являющийся специалистом именно по этим данным, что позволяло ему вести эффективный контроль пра-

вильности данных на всех стадиях обработки; данные находились непосредственно у пользователя, что давало возможность их оперативной обработки.

Централизация данных на одной ЭВМ, несомненно, дающая эффективные возможности хранения и обработки данных, не позволяла реализовывать вышеназванные достоинства.

Развитие вычислительных компьютерных сетей обусловило новые возможности в организации и ведении данных, позволяющие каждому пользователю иметь на своем компьютере свои данные и работать с ними и в то же время позволяющие работать всем пользователям со всей совокупностью данных как с единой централизованной базой данных. Соответствующая совокупность данных называется распределенной базой данных.

Термин «распределенная база данных» достаточно часто встречается в литературе [18, 27, 31]. Однако в разных источниках под этим термином понимаются совершенно разные вещи. Часть авторов понимают под распределенной базой данных то, что имеется удаленный сервер, на котором расположены данные, а также клиентские компьютеры, расположенные территориально в другом месте. Такая трактовка нам представляется неправильной. Настоящая распределенная база данных располагается на нескольких компьютерах. При этом часть файлов расположена на одном компьютере, часть на другом и т.д. Более того, возможна и даже часто встречается ситуация, когда информация на этих компьютерах пересекается, дублируется.

Распределенная база данных – это совокупность логически взаимосвязанных разделяемых данных (и описаний их структур), физически распределенных в компьютерной сети.

Система управления распределенной базой данных – это программная система, обеспечивающая работу с распределенной базой данных и позволяющая пользователю работать как с его локальными данными, так и со всей базой данных в целом.

Система управления распределенной базой данных (PaСУБД) является распределенной системой.

Каждый фрагмент базы данных работает под управлением отдельной СУБД, которая осуществляет доступ к данным этого фрагмента. Пользователи взаимодействуют с распределенной базой данных через локальные и глобальные приложения. Локальные приложения дают пользователю возможность работать со своими локальными данными и не требуют доступа к другим фрагментам. Глобальные приложения

дают пользователю возможность работать с другими фрагментами базы данных, расположенными на других компьютерах сети.

Объединение данных организуется виртуально. Соответствующий подход, по сути, отражает организационную структуру предприятия (и даже общества в целом), состоящего из отдельных подразделений. Причем, хотя каждое подразделение обрабатывает свой набор данных (эти наборы, как правило, пересекаются), существует необходимость доступа к этим данным как к единому целому (в частности, для управления всем предприятием).

Одним из примеров реализации такой модели может служить сеть Интернет: данные вводятся и хранятся на разных компьютерах по всему миру, любой пользователь может получить доступ к этим данным, не задумываясь о том, где они физически расположены.

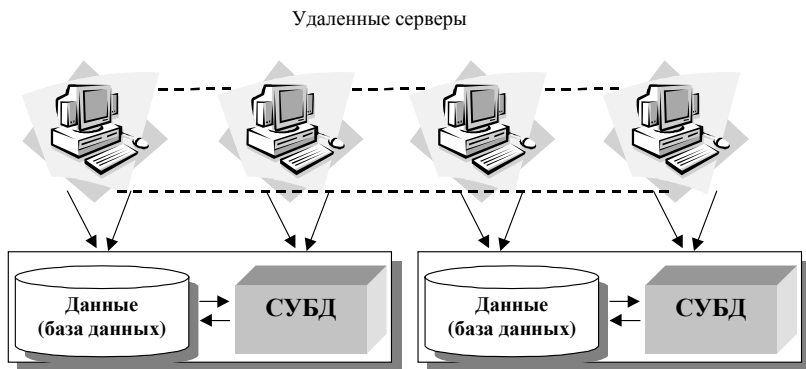


Рис. 56. Распределенная база данных

К.Дж. Дейт провозглашает следующий фундаментальный принцип распределенной базы данных [10]. Для пользователя распределенная система должна выглядеть точно так же, как нераспределенная. Из этого принципа следует ряд правил [10]:

1. Локальная автономия.
2. Независимость от центрального узла.
3. Непрерывное функционирование.
4. Независимость от расположения.
5. Независимость от фрагментации.
6. Независимость от репликации.

7. Обработка распределенных запросов.
8. Управление распределенными транзакциями.
9. Независимость от аппаратного обеспечения.
10. Независимость от операционной системы.
11. Независимость от сети.
12. Независимость от СУБД.

Заметим, что понятие распределенной базы данных можно интерпретировать как следующий шаг в развитии понятий о данных (см. раздел 1.1), обусловленный распределенностью данных в реальных предметных областях, а также новым этапом развития средств вычислительной техники – широким использованием вычислительных сетей. В этой интерпретации распределенную базу данных можно понимать как совокупность логически взаимосвязанных распределенных по разным компьютерам баз данных.

Перечислим основные проблемы создания распределенной базы данных.

1. Фрагментация данных и распределение по компьютерам.
2. Составление глобального каталога, содержащего информацию о каждом фрагменте БД и его местоположении в сети. (Каталог может храниться на одном узле или быть распределенным.)
3. Организация обработки запросов (синхронизация нескольких запросов к одним и тем же данным, исключение аномалий удаления и обновления одних и тех же данных, расположенных на различных узлах, оптимизация последовательности шагов при обработке запроса и т.д.).

Значительным достоинством этой модели является приближение данных к месту их порождения, что позволяет существенно повысить их достоверность, недостатком – достаточно высокая сложность управления данными как единым целым.

К сожалению, процесс создания и обслуживания распределенных баз данных связан и с техническими трудностями, среди которых можно выделить жесткие требования к пропускной способности каналов связи, а также низкую производительность, обусловленную значительными затратами коммуникационных и вычислительных ресурсов при их синхронизации во время выполнения транзакций (особенно при интенсивных обращениях из разных узлов к одному фрагменту).

В задачу данного учебника не входит подробное изучение принципов построения распределенных баз данных. Интересующимся ре-

комендуем обратиться к соответствующей литературе, например [8, 10, 18 и др.]. Здесь мы хотим лишь обрисовать проблему и сделать некоторые выводы по перспективам ее решения. Технология, связанная с использованием распределенных баз данных, в наибольшей степени соответствует организационной человеческой деятельности (информация распределена по месту деятельности людей, и они обмениваются ей в процессе работы) и позволяет наиболее успешно решать важнейшие проблемы, ведения баз данных:

- повысить достоверность информации (информация вводится в месте ее порождения лицом, которое лучше всех понимает ее смысловое значение);
- повысить оперативность локальной обработки информации (соответствующие вопросы решаются на локальном компьютере с фрагментом базы данных).

Поэтому очевидно, что задача проектирования, создания и функционирования распределенных баз данных является весьма существенной, активно изучается в настоящее время и будет решаться и далее.