Лекция 5

Модели архитектуры БД

Понятия о СУБД

Все структуры хранения начинались с файловых систем. Файловые структуры хранения являются естественными для машинной реализации, однако с ростом числа пользователей появляется потребность в абстрагировании представления данных и отдельных видов представления для админа, юзера и т.д.

Эти потребности реализуются с помощью многоуровневой архитектурой БД.

Модель CODASYL (1971) – двухуровневая схема данных, это схема (сетевая схема БД, т.е. логическая организация БД в целом), сюда входит определение имени, типа каждой записи и компонента записи. В 1975 появилась модель ANSI/SPARC – трехуровневая модель данных.

Цели трехуровневой архитектуры:

1. Отделение пользовательского представления БД от физического представления. Причины реализации через трехуровневую модель: каждый пользователь должен иметь возможность обращения к БД с собственным представлением, которое может иметь возможность изменять, не меняя других пользователей.
2. Пользователям не предоставляется подробность физического хранения данных.
3. Администратор базы должен изменять структуру хранения без влияния на пользовательские представления.
4. Внутренняя структуры базы не должна меняться при миграции на новое устройство хранения.
5. Администратор должен менять концептуальную структуру базы без влияния на всех пользователей.

Внешний уровень – представление базы с точки зрения пользователей, состоит из набора view (пользовательских представлений). Каждое view является подмножеством БД. Они могут храниться статически или формироваться динамически.

Концептуальный уровень – содержит логическую структуру базы с точки зрения администратора. На концептуальном уровне представлены следующие компоненты:

1. Все сущности, их атрибуты и связи.
2. Ограничения на данные.
3. Семантическая информация о данных.
4. Информация о мерах обеспечения безопасности и поддержке целостности.

Уровень не содержит сведений о методах хранения данных. Пример: описание сущности содержит сведения о типах данных атрибута и о их длине, но не содержит сведений об объеме занятого пространства.

Внутренний уровень – физическое представление базы в компьютере:

1. Распределение дискового пространства для хранения данных и индексов.
2. Описание подробностей сохранения записей, в том числе указание реальных размеров сохраняемых элементов.
3. Сведения о сжатии и методах шифрования.
4. Сведения о размещениях данных.

Физический уровень – ниже, контролируется совместно с операционной системой и СУБД.

СУБД – система управление базой данных – программное обеспечение, взаимодействующее с прикладными программами пользователя и базой данных, с помощью которого пользователи могут создавать, управлять и поддерживать БД, а также осуществлять к ней контролируемый доступ.

СУБД обладает следующими возможностями:

1. Создавать БД (делается с помощью Data Definition Language).
2. Вставлять, обновлять, удалять информацию из БД с помощью Data Manipulation Language = SQL.
3. Предоставлять контролируемый доступ к базе средствами:
   1. Система обеспечения защиты предотвращает несанкционированный доступ к БД со стороны пользователей.
   2. Система поддержки целостности данных, обеспечивает непротиворечивое состояние хранения данных.
   3. Система управление параллельной работы приложения, контроль совместного доступа.
   4. Система восстановления до предыдущего непротиворечивого состояния.
4. Доступный пользователям каталог, описывающий состояние и структуру БД.

Определения базы:

1. Набор нормализованных отношений, которые отличаются по именам.
2. (Математическое определение) Совокупность данных, организованных в соответствии с концептуальной моделью данных, которая описывает характеристики этих данных и взаимоотношения между соответствующими им реалиями, предназначенная для информационного обеспечения одного или более приложения.
3. Машинная реализация рекурсивно вычислимого отношения (перечислимое – завершаемое за определенное количество шагов. Рекурсивное – возможно потребуется вложение одного в другое).

Нормализация БД. Теоретико-множественное описание отношений

Семейство множеств – множества, состоящие из необязательно различных элементов. Введем конечное семейство множеств D = {di}, i = 1...n. отношением R = R {D1, D2, ... Dn}. Отношением R на D называется заданное подмножество декартово произведения множеств Di. D\* = D1 x D2 x ... x Dn = {(d1, d2, ..., dn)}. Каждый из кортежей множества d1...dn содержит элементы множества D1...Dn. Множества Di называется доменами отношения. Подмножества доменов отношений, выделенные задание D\* называются атрибутами отношений. Тогда отношение R можно представить в виде табличной формулы.

Математически строгое определение атрибута отношений – выделенное определением отношение, подмножество некоторых членов семейства доменов отношения. При этом каждый атрибут отношения однозначно характеризуется позицией в перечне аргументных мест отношения и критерием принадлежности их значений к определенному кортежу.

Атрибут реляционной таблицы – атрибут отношения, представленный данными в реляционной таблице, при этом каждый атрибут характеризуется:

1. Локализованным уникальным именем.
2. Однозначно ассоциированным с ним доменом базы данных (в общем случае домен – абстрактный тип данных).
3. Ассоциированным с ним набором условий целостности, относящихся к этому атрибуту.

Математически строгое условие целостности реляционной таблицы – особые отдельно хранящиеся данные, которыми на концептуальном уровне в табличной или предикатной форме представлен критерий принадлежности кортежей отношению в БД.

Разница между отношением и реляционной таблицей аналогична разнице между понятием информация и данными, которыми она представлена. Множество кортежей отношения может быть бесконечным. Количество строк в реляционной таблице всегда ограничено. Существуют вычислительные ограничения на точность представления данных в таблице, в отношении на это не обращают внимание.

Табличную форму отношения можно эквивалентно представить в виде функции: F (x1, x2, ...xn) = 1 или 0 (истина или ложь) на любом наборе строк в таблице.

Функция F называется характеристической функцией отношения. Между функцией F и отношением имеется взаимно однозначное соответствие. Если множество значений задается как 0 1, то F является функцией в математическом смысле. Если как true или false, то предикатом.