



Кафедра
географической
экологии

Модели данных и СУБД в геоэкологии

Раздел 2. Реляционные базы данных

Современные направления развития баз данных

Лекция 3

Преподаватель – Воробьёв Д.С.



План лекции

- Ограничения реляционных баз данных
- Основные понятия и термины построения реляционной, многомерной и объектно-ориентированной модели представления данных
- Особенности управления транзакциями в объектно-ориентированной БД
- Достоинства и недостатки современных СУБД



Реляционная модель ограничена в представлении данных:

- Реляционная модель данных не допускает естественного представления данных со сложной (иерархической) структурой, поскольку в ее рамках возможно моделирование лишь с помощью плоских отношений (таблиц). Все отношения принадлежат одному уровню, многие значимые связи между данными либо теряются, либо их поддержку приходится осуществлять в рамках конкретной прикладной программы.
- По определению в реляционной модели поля кортежа могут содержать лишь атомарные значения. Однако, в таких приложениях как САПР, ГИС, искусственный интеллект системы оперируют со сложно - структурированными объектами. Кроме того, даже в том случае, когда сложный объект удается "уложить" в реляционную базу данных, его данные распределяются, как правило, по многим таблицам. Соответственно, извлечение каждого такого объекта требует выполнения многих операций соединения (join), что значительно замедляет работу СУБД. Обойти это и предыдущее ограничения можно было бы в том случае, если бы реляционная модель допускала
 - возможность определения новых типов данных
 - определение наборов операций, связанных с данными определенного типа

Имеются определенные недостатки и в реализации тех возможностей, которые прямо не предусматриваются реляционной моделью, но стали непременным атрибутом всех современных СУБД:

- Цикл существования реляционной базы данных состоит в переходе от одного целостного состояния к другому. Однако, нельзя избежать такой ситуации, когда пользователь вводит данные, формально удовлетворяющие ограничениям целостности, но не соответствующие реальному состоянию предметной области. В этом случае предыдущее "истинное" значение данных будет утеряно.
- Реляционная СУБД выполняет над данными не только те действия, которые задает пользователь, но и дополнительные операции в соответствии с правилами, заложенными в базу данных. Этот механизм реализуется с помощью триггеров, однако аппарат триггеров весьма сложен в отладке и полностью не реализован ни в одной системе.



Постреляционная модель представления данных является расширенной версией реляционной модели данных и позволяет устранить ограничение неделимости данных, хранящихся в записях таблиц. Именно поэтому хранение данных в постреляционной модели по сравнению с реляционной считается более эффективным.

Достоинство постреляционной модели состоит в том, что она дает возможность формирования совокупности связанных реляционных таблиц через одну постреляционную таблицу, что обеспечивает высокую наглядность представления информации и эффективность ее обработки.

Недостаток такой модели заключается в сложности решения проблемы обеспечения целостности и непротиворечивости хранимых данных.

Примерами постреляционных СУБД являются системы UniVers, Budda и Dasdb.

Многомерные модели – это узкоспециализированные СУБД, которые используются для интерактивной аналитической обработки информации. Многомерная организация данных отличается более высокой наглядностью и информативностью в сравнении с реляционной моделью.

Основным недостатком многомерной модели данных является ее громоздкость для решения простейших задач обычной оперативной обработки информации.

Примерами СУБД на основе таких моделей являются Ess-base фирмы Arbor Software, Oracle Express Server фирмы Oracle и др.

Объектно-ориентированные модели представления данных позволяют идентифицировать отдельные записи базы. Между записями базы данных и функциями их обработки формируются определенные взаимосвязи с помощью механизмов, похожих на соответствующие средства в объектно-ориентированных языках программирования.

Достоинствами объектно-ориентированной модели данных являются:

- 1) возможность показа информации о сложных взаимосвязях объектов;
- 2) способность идентификации отдельной записи базы данных и определения функции ее обработки.

К недостаткам объектно-ориентированной модели данных относятся:

- 1) трудность в понимании ее деятельности непрофессиональным пользователем;
- 2) неудобство обработки данных;
- 3) небольшая скорость выполнения запросов.

Среди объектно-ориентированных СУБД можно выделить системы POET фирмы POET Software, Versant фирмы Versant Technologies и др.



- Дела с управлением транзакциями в системах ООБД обходятся примерно аналогично ситуации с оптимизацией запросов: как правило, используются слегка модифицированные традиционные методы сериализации транзакций, журнализации изменений объектов, индивидуальных откатов транзакций и восстановления состояния БД после сбоев. Фактически, как и в случае оптимизации запросов, такое управление транзакциями предполагает частичное нарушение инкапсуляции объектов: синхронизация основывается на знании внутренней структуры объектов, журнализация и восстановление - на знании природы методов, изменяющих состояние объекта и т.д.
- Какой-либо подход, в котором предлагался бы полный набор средств управления транзакциями, полностью согласующийся с парадигмой объектной ориентированностью, неизвестен.
- В основе управления транзакциями в системе VODAK находится разработанный ранее в контексте инженерных СУБД механизм транзакций со вложенными подтранзакциями: вызов любого действия в объекте равносителен образованию новой подтранзакции. В отличие от традиционного механизма вложенных подтранзакций в данном случае заранее не определена максимальная вложенность транзакций. При синхронизации транзакций используется знание о семантике объектов, в том числе информация о коммутативности операций.
- Не очень понятно, как при таком подходе поступать с журнализацией изменений. В принципе только сам объект знает, какая информация может понадобиться для его восстановления, и только сам объект может выполнить такое восстановление. Может быть, следует применять технику темпоральных БД, и при каждом изменении состояния объекта заводить его новую версию. В общем, как нам представляется, проблема журнализации и восстановления в ООБД пока остается открытой.