Пример общей организации СУБД. Физическое представление реляционных баз данных во внешней памяти. Индексные структуры

План (1)

- Введение
- Основные понятия, цели и общая организация System R
  - Используемая терминология
  - Цели System R и их связь с общей организацией системы
  - Организация внешней памяти в базах данных System R
  - Интерфейс RSS

План (2)

- Общие принципы организации данных во внешней памяти в SQL-ориентированных СУБД
  - Хранение таблиц
  - Индексы
    - B+-деревья
    - Хэширование
  - Журнальная информация
  - Служебная информация
- Заключение

Введение (1)

- В 1975-1979 г.г. в исследовательской лаборатории компании IBM разрабатывалась система управления реляционными базами данных System R
  - Эта работа оказала революционизирующее влияние на развитие теории и практики реляционных систем во всем мире
  - Именно System R практически доказала жизнеспособность реляционного подхода к управлению базами данных
- После успешного завершения работ по созданию этой системы и получения экспериментальных результатов ее использования был разработан целый ряд коммерчески доступных реляционных систем,
  - В том числе и на основе непосредственного развития System R

Введение (2)

- Исключительно важен опыт, приобретенный при разработке этой системы
- Практически во всех более поздних реляционных СУБД в той или иной степени используются методы, примененные в System R
- Поэтому лекции, посвященные внутренней организации SQL-ориентированных СУБД, во многом опираются на материалы статей, посвященных System R

Основные понятия, цели и общая организация System R (1)

Используемая терминология (1)

- Несмотря на то, что при реализации System R использовался подход, несколько отличающийся от реляционного подхода Кодда (отсюда и пошли расхождения между реляционной моделью данных и моделью данных SQL), мы будем активно пользоваться терминами реляционной модели
  - К таким терминам относятся, например, названия реляционных операций – ограничение, проекция, соединение; названия теоретико-множественных операций – объединение, пересечение, взятие разности и т.д.
- В тех случаях, когда терминология System R расходится с реляционной терминологией, предпочтение будет отдаваться терминологии System R

Основные понятия, цели и общая организация System R (2)

Используемая терминология (2)

- В частности, это касается использования термина «поле таблицы» вместо термина «атрибут отношения»
- В самой System R при переходе к коммерческим системам также произошла некоторая смена терминологии
  - В частности, появилась тенденция к употреблению терминов, более привычных в среде пользователей IBM: файл, запись и т.д.
- Здесь будут использоваться термины System R, более близкие реляционным системам
- Опишем некоторые основные термины System R,
  - стремясь отразить практические аспекты соответствующих понятий

Основные понятия, цели и общая организация System R (3)

Используемая терминология (3)

- Базовым понятием System R является понятие *таблицы*
  - приближенный к реализации аналог основного понятия реляционного подхода *отношения*
  - иногда, в зависимости от контекста, мы будем использовать и этот термин
- Таблица – это регулярная структура данных, состоящая из конечного набора однотипных записей – кортежей.
- Каждый кортеж одного таблицы состоит из конечного (и одинакового) числа полей кортежа, причем
  - *i*-тое поле каждого кортежа одной таблицы может содержать данные только одного типа, и
  - набор допустимых типов данных в System R предопределен и фиксирован

Основные понятия, цели и общая организация System R (4)

Используемая терминология (4)

- В силу регулярности структуры таблицы понятие поля кортежа расширяется до понятия поля таблицы
- Тогда *i*-тое поле таблицы можно трактовать как набор одноместных кортежей, полученных выборкой *i*-тых полей из каждого кортежа этой таблицы,
  - т.е. в общепринятой терминологии как проекцию таблицы на *i*-тый атрибут
- В терминологию System R не входит понятие домена,
  - оно заменяется здесь понятием типа поля,
    - т.е. типом данных, хранение которых в данном поле допускается

Основные понятия, цели и общая организация System R (5)

Используемая терминология (5)

- Таблицы, составляющие базу данных System R, могут физически храниться в одном или нескольких сегментах, которые проще всего понимать как файлы внешней памяти
- Сегменты разбиваются на страницы, в которых располагаются кортежи таблиц и вспомогательные служебные структуры данных – индексы
- Соответственно, каждый сегмент содержит две группы страниц – страницы данных и страницы индексной информации
- Страницы каждой группы имеют фиксированный размер, но страницы с индексной информацией меньше по размеру, чем страницы данных
- В страницах данных могут располагаться кортежи более чем одной таблицы

Основные понятия, цели и общая организация System R (6)

Цели System R и их связь с общей организацией системы (1)

- При выполнении проекта System R преследовались следующие основные цели:
  - обеспечить ненавигационный интерфейс высокого уровня пользователя с системой,
    - позволяющий достичь независимости данных и дать возможность пользователям работать максимально эффективно;
  - обеспечить многообразие допустимых способов использования СУБД,
    - включая программируемые транзакции, диалоговые транзакции и генерацию отчетов;
  - поддерживать динамически изменяемую среду баз данных,
    - в которой таблицы, индексы, представления, транзакции и другие объекты могут легко добавляться и уничтожаться без приостановки нормального функционирования системы;

Основные понятия, цели и общая организация System R (7)

Цели System R и их связь с общей организацией системы (2)

- обеспечить возможность параллельной работы с одной базой данных многих пользователей
  - с возможностью параллельной модификации объектов базы данных при наличии необходимых средств защиты целостности базы данных;
- обеспечить средства восстановления согласованного состояния баз данных после разного рода сбоев аппаратуры или программного обеспечения;
- обеспечить гибкий механизм, позволяющий определять различные представления хранимых данных и ограничивать этими представлениями доступ пользователей к базе данных по выборке и модификации на основе механизма авторизации;
- обеспечить производительность системы при выполнении упомянутых функций, сопоставимую с производительностью существующих СУБД низкого уровня

Основные понятия, цели и общая организация System R (8)

Цели System R и их связь с общей организацией системы (3)

- Основой System R является «реляционный» язык SQL
  - разработчики System R искренне считали созданный ими язык реляционным
- Иногда его называют языком запросов или языком манипулирования данными, но на самом деле возможности SQL гораздо шире
- Средствами SQL (с соответствующей системной поддержкой) решаются многие из поставленных целей

Основные понятия, цели и общая организация System R (9)

Цели System R и их связь с общей организацией системы (4)

- Язык SQL включает средства динамической компиляции запросов,
  - на основе чего возможно построение диалоговых систем обработки запросов
- Допускается динамическая параметризация статически откомпилированных запросов,
  - в результате чего возможно построение эффективных (не требующих динамической компиляции) диалоговых систем со стандартными наборами (параметризуемых) запросов
- Средствами SQL определяются все доступные пользователю объекты баз данных:
  - таблицы, индексы, представления
- Имеются средства уничтожения любого такого объекта
- Соответствующие операторы языка могут выполняться в любой момент, и
  - возможность выполнения операции данным пользователем зависит от ранее предоставленных ему прав

Основные понятия, цели и общая организация System R (10)

Цели System R и их связь с общей организацией системы (5)

- В System R под целостным состоянием базы данных понимается состояние, удовлетворяющее набору сохраняемых при базе данных предикатов целостности
- Эти предикаты, называемые в System R *утверждениями целостности* (assertion), также задаются средствами языка SQL
- Любой оператор языка выполняется в границах некоторой *транзакции* – последовательности операторов языка, неделимой в смысле состояния базы данных
- Неделимость означает, что все изменения базы данных, произведенные в пределах одной транзакции,
  - либо целиком отображаются в состоянии базы данных,
  - либо полностью в нем отсутствуют
- Последняя возможность возникает при *откате* транзакции, который может произойти
  - по инициативе пользователя (при выполнении соответствующего оператора SQL) или
  - по инициативе системы

Основные понятия, цели и общая организация System R (11)

Цели System R и их связь с общей организацией системы (6)

- Одной из причин отката транзакции по инициативе системы является как раз нарушение целостности базы данных в результате действий данной транзакции
- Язык SQL System R содержит средство установки так называемых *точек сохранения* (savepoint)
  - При инициируемом пользователем откате транзакции можно указать номер точки сохранения, выше которого откат не распространяется
  - Инициируемый системой откат транзакции производится до ближайшей точки сохранения, в которой условие, вызвавшее откат, уже отсутствует
  - В частности, откат транзакции, инициированный по причине нарушения условия целостности, производится до ближайшей точки сохранения, в которой условия целостности соблюдены

Основные понятия, цели и общая организация System R (12)

Цели System R и их связь с общей организацией системы (7)

- Естественно, для реального выполнения отката транзакции необходимо запоминать некоторую информацию о выполнении транзакции
- В System R для этих и других целей используется специальный набор данных – *журнал*, в который
  - помещаются записи обо всех операциях всех транзакций, изменяющих состояние базы данных
- При откате транзакции происходит процесс *обратного выполнения* транзакции (undo), в ходе которого
  - в обратном порядке выполняются все изменения, запомненные в журнале

Основные понятия, цели и общая организация System R (13)

Цели System R и их связь с общей организацией системы (8)

- В языке SQL System R имеется средство определения так называемых *триггеров* (trigger), позволяющих автоматически поддерживать целостность базы данных при модификациях ее объектов
- В SQL System R триггер – это каталогизированная операция модификации, для которой задано условие ее автоматического выполнения
  - Особенно существенно наличие такого механизма в связи с наличием представлений базы данных, которыми может быть ограничен доступ к базе данных для ряда пользователей
  - Возможна ситуация, когда такие пользователи просто не могут соблюдать целостность базы данных без автоматического выполнения условных воздействий, поскольку они просто «не видят» всей базы данных и, в частности, не могут представить всех ограничений ее целостности

Основные понятия, цели и общая организация System R (14)

Цели System R и их связь с общей организацией системы (9)

- Язык SQL содержит средства определения представлений
- Представление – это каталогизированный именованный запрос на выборку данных (из одной или нескольких таблиц)
  - Поскольку SQL – это «реляционный» язык, результатом выполнения любого запроса на выборку является таблица, и поэтому концептуально можно относиться к любому представлению как к таблице
    - при определении представления можно, в частности, присвоить имена полям этой таблицы
  - В языке допускается использование ранее определенных представлений практически везде, где допускается использование таблиц
    - с некоторыми ограничениями по поводу возможностей модификации через представления
  - Наличие возможности определять представления в совокупности с развитой системой авторизации позволяет ограничить доступ некоторых пользователей к базе данных выделенным набором представлений

Основные понятия, цели и общая организация System R (15)

Цели System R и их связь с общей организацией системы (10)

- Авторизация доступа к базе данных также основана на средствах SQL
- При создании любого объекта базы данных пользователь, выполняющий эту операцию, становится полновластным владельцем этого объекта, т.е.
  - может выполнять по отношению к этому объекту любую допустимую операцию SQL
- Далее этот пользователь может выполнить оператор SQL, означающий передачу всех его прав на этот объект (или их подмножества) любому другому пользователю
- В частности, этому пользователю может быть передано право на передачу всех переданных ему прав (или их части) третьему пользователю и т.д.
- Одним из прав пользователя по отношению к объекту является право на изъятие у других пользователей всех или некоторых прав, которые ранее им были переданы
- Эта операция распространяется транзитивно на всех дальнейших наследников этих прав

Основные понятия, цели и общая организация System R (16)

Цели System R и их связь с общей организацией системы (11)

- Наличие в языке средств определения представлений и авторизации в принципе позволяет обойтись при эксплуатации System R без традиционного администратора баз данных, поскольку практически все системные действия производятся на основе средств SQL
  - Тем не менее, если организационно администратор баз данных требуется, то его работа достаточно упрощается за счет унифицированного набора средств управления
  - Кроме того, в System R каталоги баз данных поддерживаются также в виде таблиц, и к ним применены все запросы языка SQL
- Заметим, что в более поздних SQL-ориентированных СУБД появился ряд дополнительных утилит, не связанных с языком SQL (например, утилиты сбора статистики или массовой загрузки базы данных), и в этих системах, видимо, без администратора базы данных не обойтись

Основные понятия, цели и общая организация System R (17)

Цели System R и их связь с общей организацией системы (12)

- Что касается обеспечения параллельной работы многих пользователей с одной базой данных, основной подход System R состоит в том, что
  - пользователь не обязан знать о наличии других пользователей, конкурирующих с ним за доступ к базе данных, т.е.
  - система ответственна за обеспечение изолированности пользователей с гарантией отсутствия их взаимного влияния в пределах транзакций
- Из этого следует,
  - во-первых, что в интерфейсе пользователя с системой (т.е. в языке SQL) не должно быть средств регулирования взаимодействий с другими пользователями и,
  - во-вторых, что система должна обеспечить автоматическую сериализацию набора транзакций, т.е.
    - обеспечить режим выполнения этого набора транзакций, эквивалентный по конечному результату некоторому последовательному выполнению этих транзакций
    - эта проблема решается в System R за счет автоматического выполнения синхронизационных блокировок всех изменяемых объектов базы данных

Основные понятия, цели и общая организация System R (18)

Цели System R и их связь с общей организацией системы (13)

- Одним из основных требований к СУБД вообще и к System R, в частности, является обеспечение надежности баз данных по отношению к различного рода сбоям
- К таким сбоям могут относиться программные ошибки прикладного и системного уровня, сбои процессора, поломки внешних носителей и т.д.
- В частности, к одному из видов сбоев можно отнести упоминавшиеся выше нарушения целостности базы данных, и
  - автоматический инициируемый системой откат транзакции – это системное средство восстановления базы данных после сбоев такого рода
  - как уже отмечалось, такое восстановление происходит путем обратного выполнения транзакции на основе информации о внесенных ею изменениях, запомненной в журнале
- На информации журнала также основано восстановление базы данных и после сбоев другого рода

Основные понятия, цели и общая организация System R (19)

Цели System R и их связь с общей организацией системы (14)

- Что касается естественных требований к эффективности системы, то здесь основные решения связаны
  - со спецификой физической организации баз данных на внешней памяти,
  - использованием техники индексированного доступа к данным,
  - буферизацией используемых страниц базы данных в основной памяти и
  - развитой техникой оптимизации SQL-запросов, производимой на стадии их компиляции
- Структурная организация System R согласуется с поставленными при ее разработке целями и выбранными решениями
- Основными структурными компонентами System R являются система управления реляционными данными (Relational Data System – RDS), состоящая, по существу,
  - из компилятора языка SQL и подсистемы поддержки откомпилированных операторов, и
  - системы управления реляционной памятью (Relational Storage System – RSS)

Основные понятия, цели и общая организация System R (20)

Цели System R и их связь с общей организацией системы (15)

- RSS обеспечивает интерфейс довольно низкого, но достаточного для реализации SQL уровня для доступа к хранимым в базе данным
  - этот внутренний интерфейс System R напоминает внешний интерфейс систем, основанных на модели инвертированных таблиц
- Синхронизация транзакций, журнализация изменений и восстановление баз данных после сбоев также относятся к числу функций RSS.
- Компилятор запросов
  - использует интерфейс RSS для доступа к разнообразной справочной информации (каталоги таблиц, индексов, прав доступа, условий целостности, условных воздействий и т.д.) и
  - производит рабочие программы, выполняемые в дальнейшем также с использованием интерфейса RSS

Основные понятия, цели и общая организация System R (21)

Цели System R и их связь с общей организацией системы (16)

- Таким образом, система естественно разделяется на два уровня
  - уровень управления памятью и синхронизацией, фактически, не зависящий от базового языка запросов системы, и
  - языковый уровень (уровень SQL), на котором решается большинство проблем System R
- Заметим, что эта независимость скорее условная, чем абсолютная:
  - язык SQL можно в принципе заменить каким-либо другим языком, но он должен обладать примерно такой же семантикой

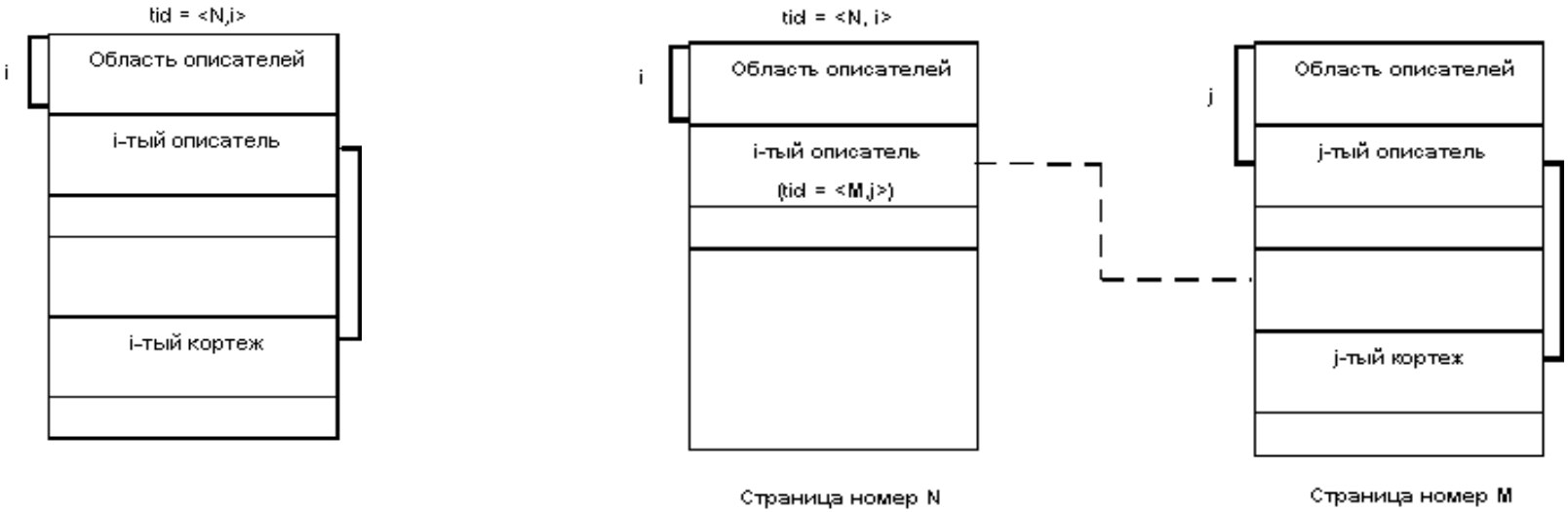
Основные понятия, цели и общая организация System R (22)

Организация внешней памяти в базах данных System R (1)

- Как уже говорилось, база данных System R располагается в одном или нескольких сегментах внешней памяти
- Каждый сегмент состоит из страниц данных и страниц индексной информации
- Размер страницы данных в сегменте может быть выбран равным либо 4, либо 32 килобайтам;
  - размер страницы индексной информации равен 512 байтам
- Кроме того, при работе RSS поддерживается дополнительный набор данных для ведения журнала
  - Для повышения надежности журнала этот набор данных дублируется на двух внешних носителях

Основные понятия, цели и общая организация System R (23)

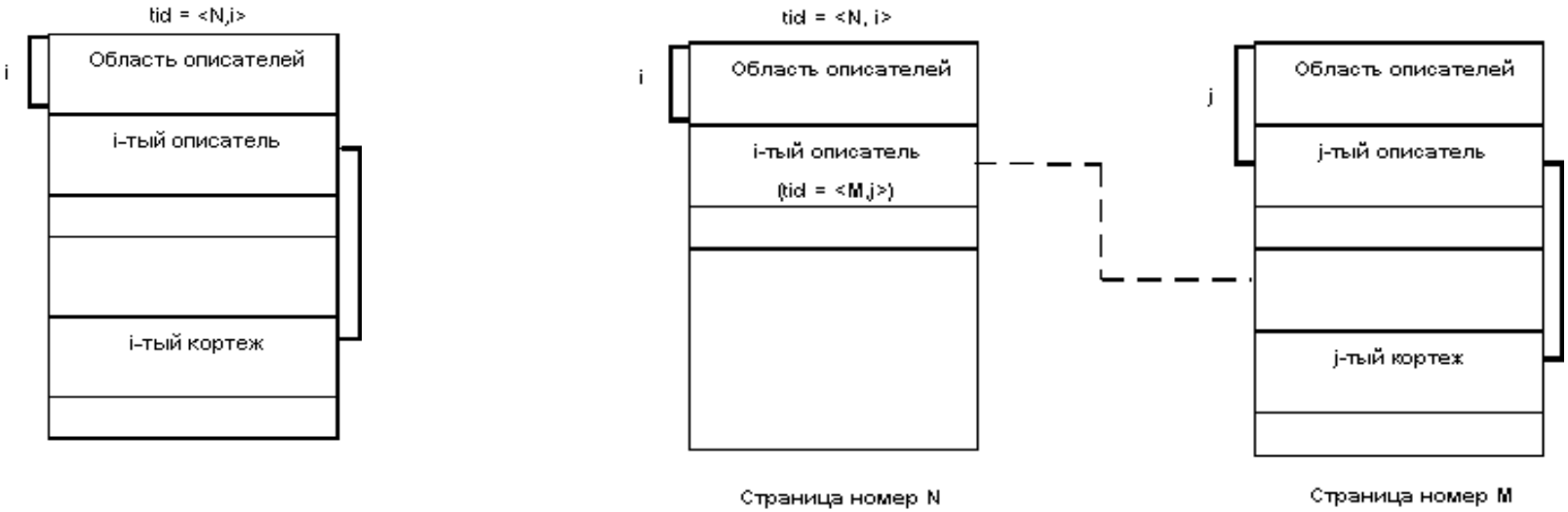
Организация внешней памяти в базах данных System R (2) Страницы данных (1)



- В каждой странице данных хранятся кортежи одного или нескольких таблиц
- Фундаментальным понятием RSS является *идентификатор кортежа* (tuple identifier – tid)
- Гарантируется неизменяемость tid'а во все время существования кортежа в базе данных независимо от перемещений кортежа внутри страницы и даже при перемещении кортежа в другую страницу

Основные понятия, цели и общая организация System R (24)

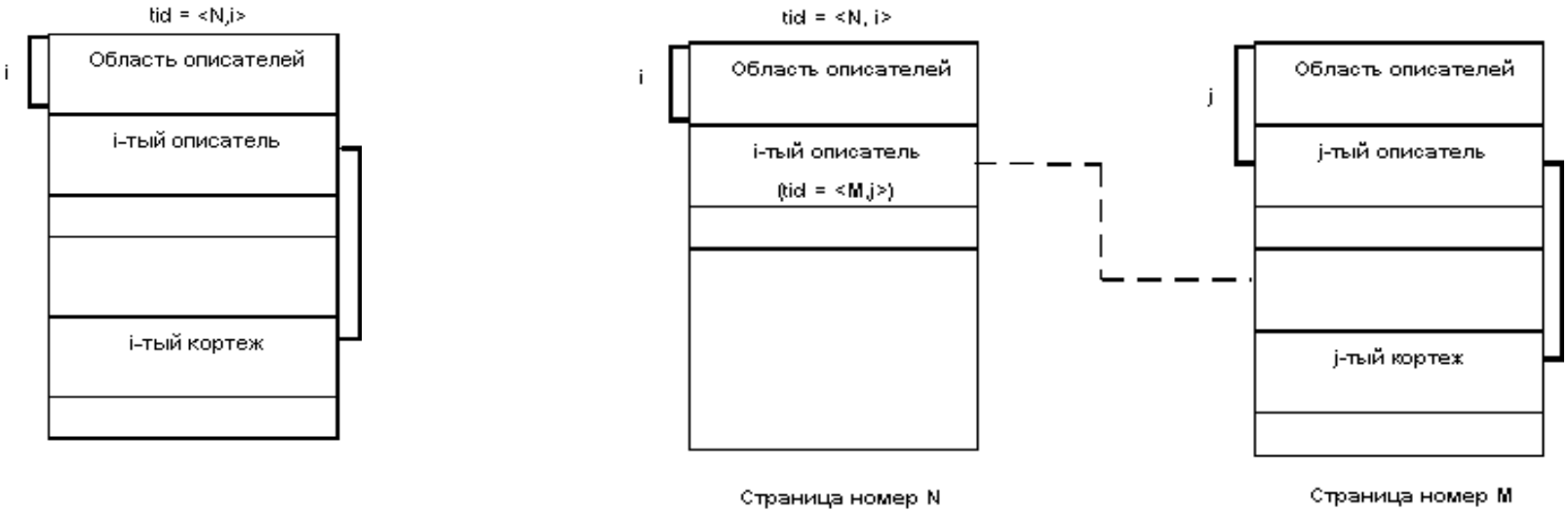
Организация внешней памяти в базах данных System R (3) Страницы данных (2)



- Потребность в перемещении кортежей возникает по той причине, что кортеж, занесенный в некоторую таблицу базы данных, вообще говоря, во время своего существования может увеличиваться в размерах
  - ✓ если к этой таблице добавляется новое поле, или
  - ✓ если в ней имеется хотя бы одно поле, типом данных которого являются строки символов переменного размера

Основные понятия, цели и общая организация System R (25)

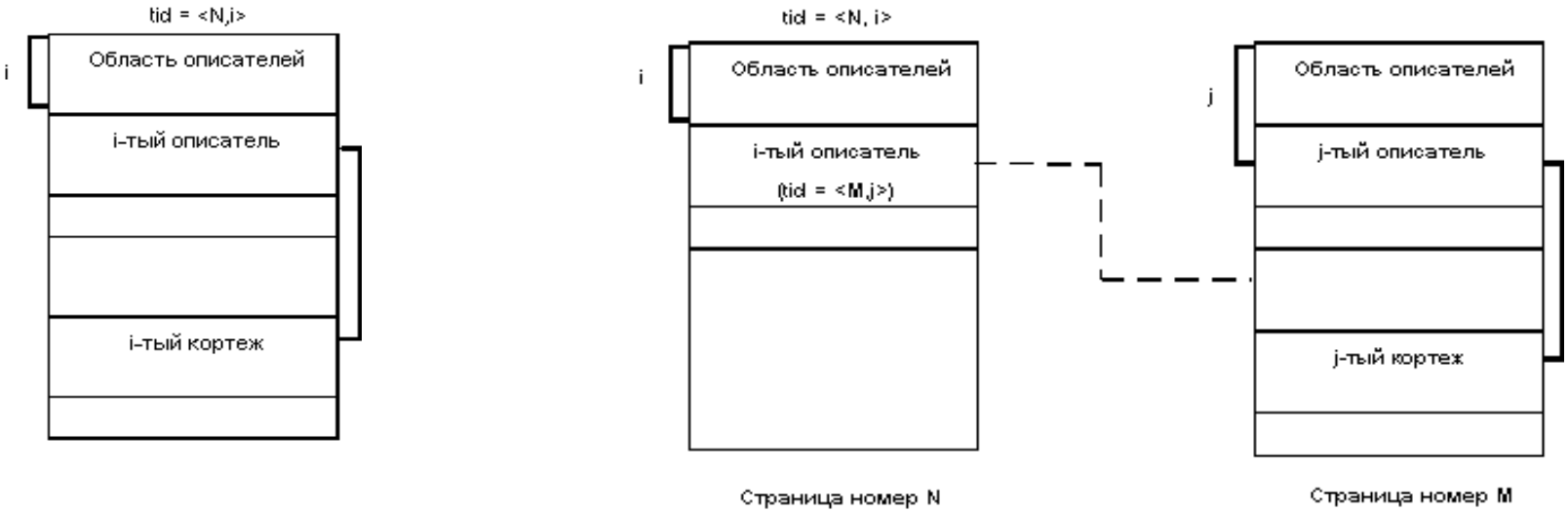
Организация внешней памяти в базах данных System R (4) Страницы данных (3)



- Реально tid представляет собой пару <номер страницы, индекс описателя кортежа в странице>
- При этом кортеж может реально располагаться в данной странице (рис. слева) или в другой странице (рис. справа)

Основные понятия, цели и общая организация System R (26)

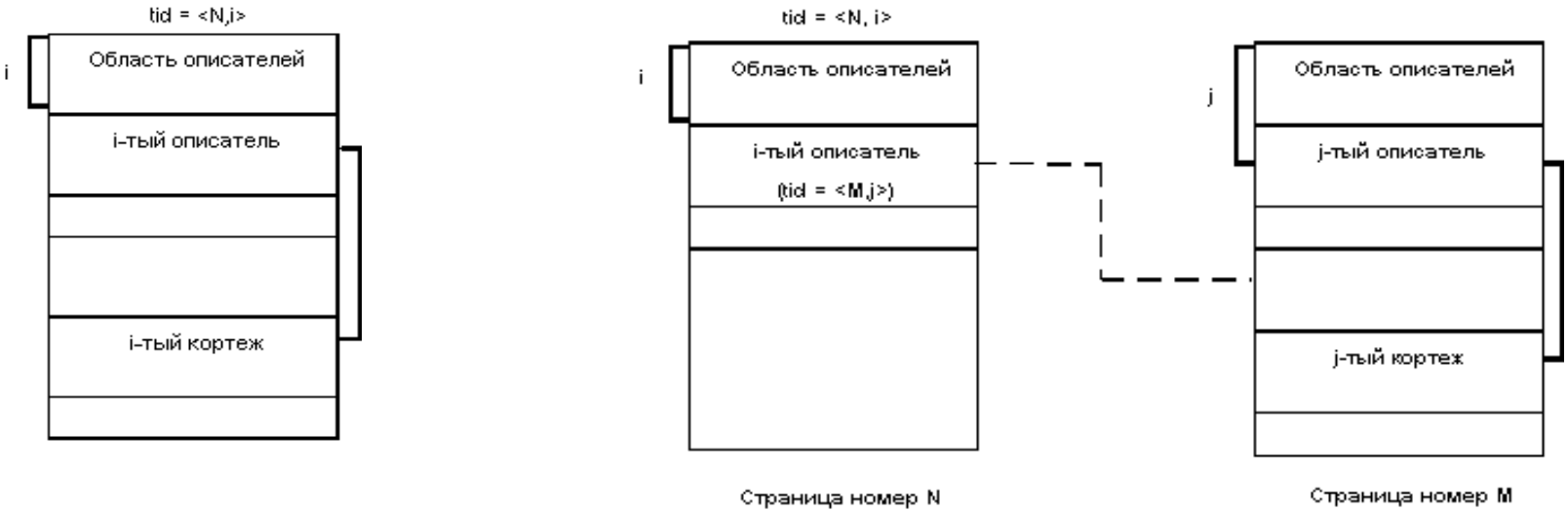
Организация внешней памяти в базах данных System R (5) Страницы данных (4)



- Как показывает рис. слева, в каждой странице данных имеются две области:
  - ✓ область хранения описателей кортежей и
  - ✓ область хранения самих кортежей
- Обе эти области являются динамическими, т.е. в странице данных заранее не резервируется место под описатели кортежей

Основные понятия, цели и общая организация System R (27)

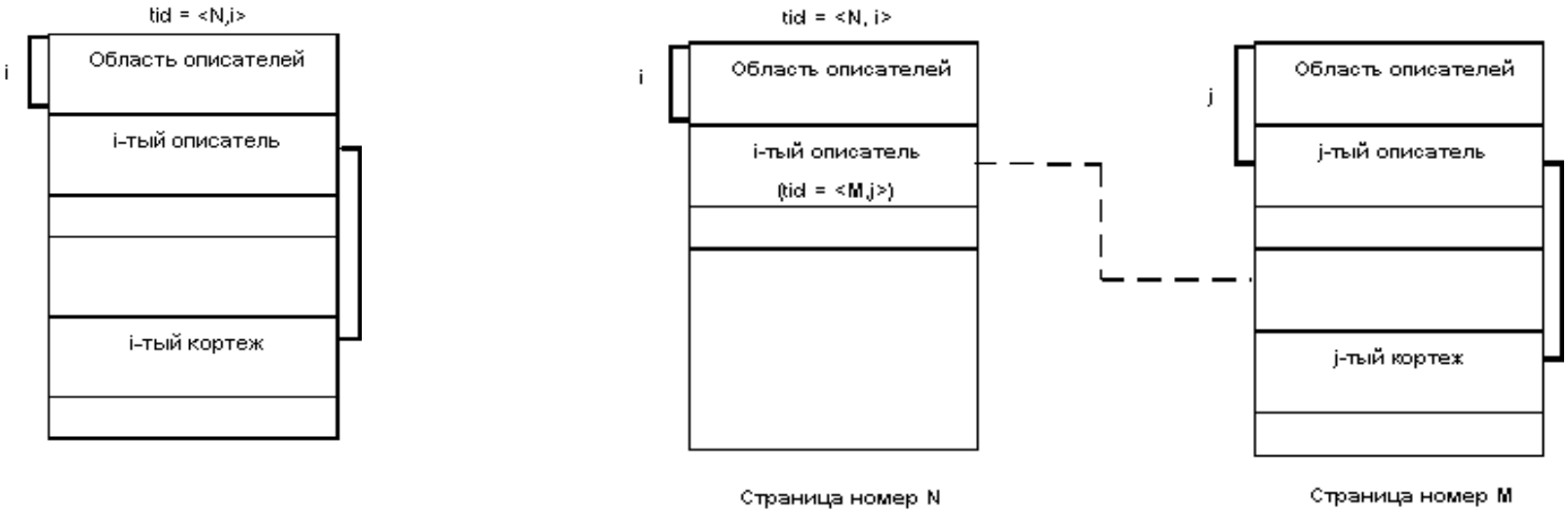
Организация внешней памяти в базах данных System R (6) Страницы данных (5)



- Выделение фиксированной части страницы данных под описатели кортежей (вмещающей, скажем, *k* описателей) потенциально привело бы к потери памяти в этой странице, поскольку при размещении в ней *k* кортежей очень маленького размера пропадало бы место в области хранения кортежей, а при размещении *p*<*k* крупных кортежей полностью заполнялась бы область хранения кортежей, но пропадало бы место в области описателей

Основные понятия, цели и общая организация System R (28)

Организация внешней памяти в базах данных System R (7) Страницы данных (6)

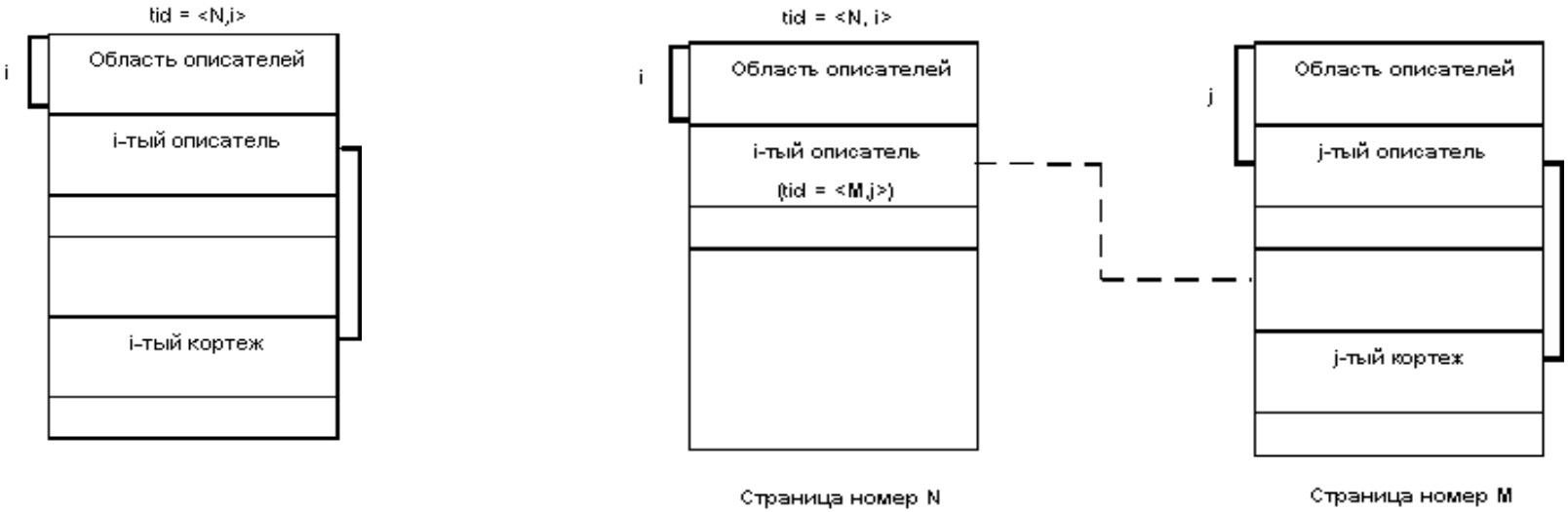


- Для динамического распределения памяти внутри страницы память под описатели кортежей выделяется
  - ✓ вниз от начала страницы,

  а память для хранения кортежей –
  - ✓ вверх от конца страницы

Основные понятия, цели и общая организация System R (29)

Организация внешней памяти в базах данных System R (8) Страницы данных (7)



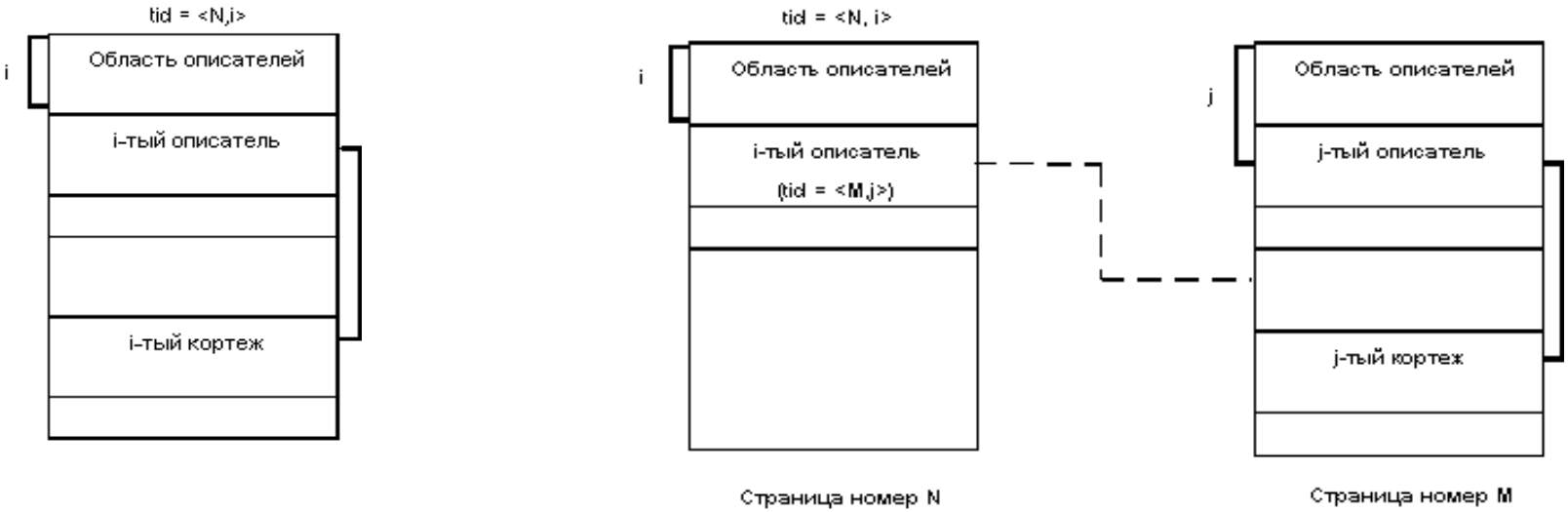
- Второй вариант хранения кортежей возникает в том случае, когда некоторый кортеж после своего создания был размещен системой в
  - ✓ странице с номером N,

а после обновления с увеличением размера перестал помещаться в этой странице, и система была вынуждена разместить его в

- ✓ странице с номером M

Основные понятия, цели и общая организация System R (30)

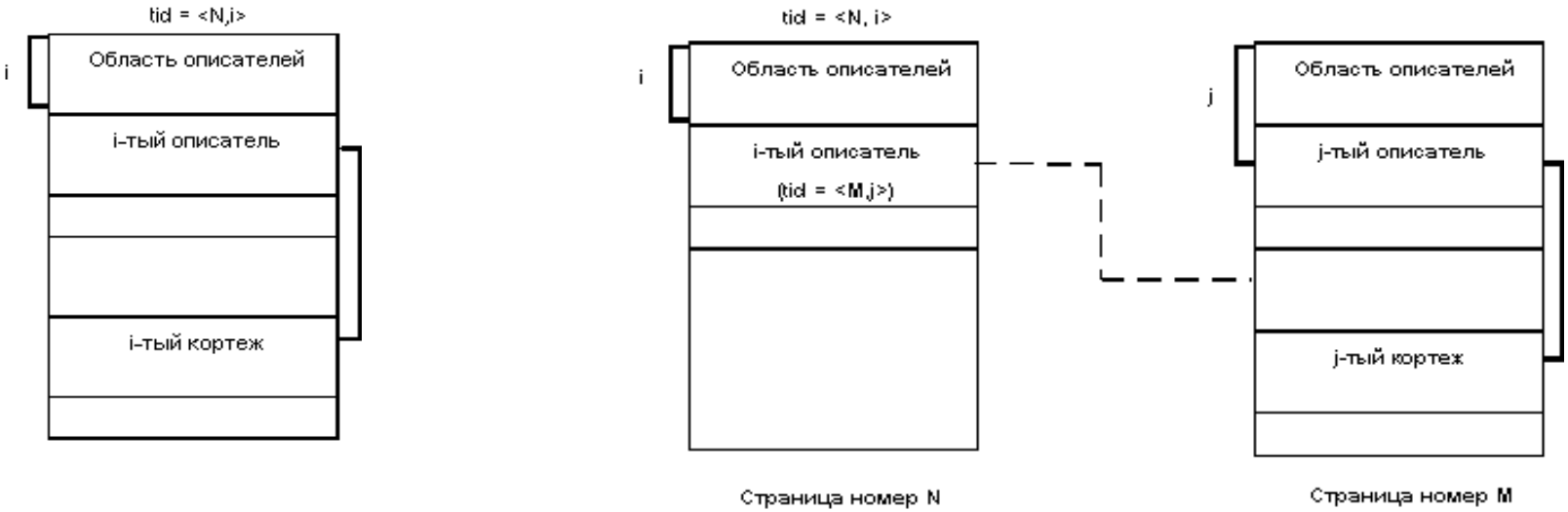
Организация внешней памяти в базах данных System R (9) Страницы данных (8)



- Тогда исходный tid этого кортежа не изменится, но его описатель в странице N будет содержать не координаты кортежа в данной странице, а
- ✓ новый tid, указывающий на реальное положение кортежа в странице M

Основные понятия, цели и общая организация System R (31)

Организация внешней памяти в базах данных System R (10) Страницы данных (9)



- Легко видеть, что применение такого подхода позволяет ограничиться максимум одним уровнем косвенности
- ✓ если данный кортеж в какой-то момент времени перестанет помещаться в странице M, и система переместит его в страницу P, то достаточно будет изменить косвенную ссылку на этот кортеж в странице N, и его исходный tid не изменится

Основные понятия, цели и общая организация System R (32)

Организация внешней памяти в базах данных System R (11) Страницы данных (10)

- Поскольку допускается нахождение в одной странице данных кортежей разных таблиц, каждый кортеж должен, кроме содержательной части, включать
  - служебную информацию, идентифицирующую таблицу, которому принадлежит данный кортеж
- Кроме того, в System R (точнее, в языке SQL) допускается динамическое добавление полей к существующим таблицам
- При этом реально происходит лишь модификация описателя таблицы в таблице-каталоге таблиц

Основные понятия, цели и общая организация System R (33)

Организация внешней памяти в базах данных System R (12) Индексы и кластеризация таблиц

- (1) На основе наличия уникальных, обеспечивающих почти прямой доступ к кортежам и не изменяемых во время существования кортежей tid'ов в System R поддерживаются
  - дополнительные управляющие структуры – индексы
- Каждый индекс определяется на одном или нескольких полях таблицы, значения которых составляют его ключ, и позволяет производить
  - прямой поиск по ключу кортежей (их tid'ов) и
  - последовательное сканирование таблицы по индексу, начиная с указанного ключа, в порядке возрастания или убывания значений ключа

Основные понятия, цели и общая организация System R (34)

Организация внешней памяти в базах данных System R (13) Индексы и кластеризация таблиц

- (2) Некоторые индексы при их создании могут обладать атрибутом уникальности
  - В таком индексе не допускаются дубликаты ключа
  - Это единственное средство SQL System R указания системе первичного ключа таблицы (фактически, набора первичного и всех возможных ключей таблицы).
- Для организации индексов в System R применяется техника B+-деревьев
- более подробно B+-деревья рассматриваются в этой лекции позже

Основные понятия, цели и общая организация System R (35)

Организация внешней памяти в базах данных System R (14) Индексы и кластеризация таблиц

- (3) Каждый индекс занимает отдельный набор страниц, номер корневой страницы запоминается в описателе индекса
- Использование B+-деревьев позволяет достичь эффективности при прямом поиске, поскольку они в силу своей сильной ветвистости обладают небольшой глубиной
- Кроме того, B+-деревья сохраняют порядок ключей в листовых блоках иерархии, что позволяет производить последовательное сканирование таблицы в порядке возрастания или убывания значений полей, на которых определен индекс

Основные понятия, цели и общая организация System R (36)

Организация внешней памяти в базах данных System R (15) Индексы и кластеризация таблиц

- (4) Фундаментальное свойство B+-деревьев
  - автоматическая балансировка дерева

  допускает произведение лишь локальных модификаций индекса при переполнениях и опустошениях страниц индекса
- System R была первой системой, в которой для организации индексов использовались B+-деревья
- Эту традицию соблюдает большинство реляционных систем, возникших после System R

Основные понятия, цели и общая организация System R (37)

Организация внешней памяти в базах данных System R (16) Индексы и кластеризация таблиц

(5)

- Видимо, наиболее важной особенностью физической организации баз данных в System R является возможность обеспечения кластеризации связанных кортежей одной или нескольких таблиц
- Под кластеризацией кортежей понимается физически близкое расположение (в пределах одной страницы данных) логически связанных кортежей
- Обеспечение соответствующей кластеризации позволяет добиться высокой эффективности системы при выполнении выделенного класса запросов
- В силу большой важности понятия кластеризации в System R и ее развитиях рассмотрим историю вопроса более подробно

Основные понятия, цели и общая организация System R (38)

Организация внешней памяти в базах данных System R (17) Индексы и кластеризация таблиц

- (6) В окончательном варианте System R существует только одно средство определения условий кластеризации таблицы
  - объявить до заполнения таблицы один (и только один) индекс, определенный на полях этой таблицы, кластеризованным
- Тогда, если заполнение таблицы кортежами производится в порядке возрастания или убывания значений полей кластеризации (в зависимости от атрибутики индекса),
  - система физически располагает кортежи в страницах данных в том же порядке

Основные понятия, цели и общая организация System R (39)

Организация внешней памяти в базах данных System R (18) Индексы и кластеризация таблиц

(7)

- Кроме того, в каждой странице данных кластеризованной таблицы оставляется некоторое
  - резервное свободное пространство
- При последующих вставках кортежей в такую таблицу система стремится поместить каждый кортеж в одну из страниц данных, в которых уже находятся кортежи этой таблицы
  - с такими же (или близкими) значениями полей кластеризации

Основные понятия, цели и общая организация System R (40)

Организация внешней памяти в базах данных System R (19) Индексы и кластеризация таблиц

(8) Естественно, поддерживать идеальную кластеризацию таблицы можно только до определенного предела,

- пока не исчерпается резервная память в страницах

Далее этого предела степень кластеризации таблицы начинает уменьшаться, и для восстановления идеальной кластеризации таблицы требуется

- физическая реорганизация таблицы
- ее можно произвести средствами SQL

Основные понятия, цели и общая организация System R (41)

- (9) Очевидным преимуществом кластеризации таблицы является то, что при последовательном сканировании кластеризованной таблицы с использованием кластеризованного индекса потребуется
  - ровно столько чтений страниц данных с внешней памяти,
  - сколько страниц занимают кортежи этой таблицы
- Следовательно, при правильно выбранных критериях кластеризации запросы, связанные с заданием условий на полях кластеризации можно выполнить почти оптимально

Основные понятия, цели и общая организация System R (42)

Организация внешней памяти в базах данных System R (21) Индексы и кластеризация таблиц

(10)
- В ранних версиях System R существовал еще один способ физического доступа к кортежам таблицы и, соответственно, еще один способ указания условия кластеризации с использованием так называемых связей (links)
- На уровне физического представления связь – это физическая ссылка (tid) из одного кортежа на другой кортеж
  - не обязательно одной таблицы
- В языке SEQUEL (до того момента, когда его стали называть SQL) существовали средства определения связей в иерархической манере:
  - можно было объявить некоторую таблицу родительской по отношению к той же или другой таблице-потомку

Основные понятия, цели и общая организация System R (43)

- (11) При этом указывались поля родительской таблицы и таблицы-потомка, в соответствии со значениями которых образовывалась иерархия
- Правила построения были очень простыми
  - проводились связи от кортежа родительской таблицы ко всем кортежам таблицы-потомка с теми же значениями полей связывания
- На самом деле, все кортежи таблицы-потомка с общим значением полей связывания образовывали кольцевой список, на который проводилась одна связь из соответствующего кортежа родительской таблицы

Основные понятия, цели и общая организация System R (44)

- (12) Следует заметить, что этот способ использования механизма связей поддерживался в ранних версиях SEQUEL
- В интерфейсе RSS System R этого периода допускалась возможность произвольного проведения связей без учета совпадения значений полей связывания
- Тем самым, в системе в целом не использовались все возможности RSS, которые
  - с избытком превосходили потребности организации иерархических бинарных связей по совпадению полей связывания

Основные понятия, цели и общая организация System R (45)

Организация внешней памяти в базах данных System R (24) Индексы и кластеризация таблиц

- (13) Для одной таблицы допускалось создание многих связей: кортеж таблицы мог быть родителем нескольких иерархий и входить в несколько других иерархий в качестве потомка
- При этом одна связь могла быть объявлена кластеризованной
  - Тогда система стремилась поместить в одну страницу данных все кортежи одной иерархии
  - При этом, естественно, использовалась возможность размещения в одной странице данных кортежей нескольких таблиц
- Основной смысл такой кластеризации заключался в возможности оптимизации выполнения некоторых запросов,
  - включающих (экви)соединение двух связанных таблиц в соответствии со значениями полей связывания

Основные понятия, цели и общая организация System R (46)

- (14) В более поздних публикациях, посвященных System R, упоминания о механизме связей исчезли, из чего можно заключить, что разработчики отказались от его использования
- Думается, что основными причинами отказа от использования связей были следующие
- Во-первых, средства построения связей, обеспечиваемые RSS, были очень низкого уровня, гораздо более низкого, чем средства поддержания индексов
  - Если при занесении, удалении или обновлении кортежа RSS обеспечивала автоматическую коррекцию всех индексов, то
    - для коррекции связей требовалось выполнить ряд дополнительных обращений к RSS, из-за чего время выполнения этих операций, конечно

Основные понятия, цели и общая организация System R (47)

- (15) Во-вторых, при реализации этого механизма возникают дополнительные синхронизационные проблемы нижнего уровня (уровня совместного доступа к страницам данных)
  - В частности, наличие прямых ссылок между страницами данных увеличивает вероятность возникновения синхронизационных тупиков.
- Наконец, в-третьих, все эти дополнительные накладные расходы не окупались преимуществами, предоставляемыми механизмом связей
  - Действительно, максимального эффекта от использования связей можно достичь только при выполнении операции соединения двух таблиц, кластеризованных по этой связи, если поле соединения совпадает с полем связывания и
    - условия, накладываемые на родительскую таблицу, выделяют в нем ровно один кортеж
  - Очевидно, что такие запросы на практике редки

Основные понятия, цели и общая организация System R (48)

(16)
- Кроме таблиц и индексов при работе System R во внешней памяти могут располагаться еще и временные объекты – списки (list)
- *Список* – это временная структура данных, создаваемая с целью оптимизации выполнения SQL-запроса,
  - содержащая некоторые кортежи хранимой таблицы базы данных,
  - не имеющая имени и, следовательно,
  - не видимая на уровне интерфейса SQL
- Кортежи списка могут быть упорядочены по возрастанию или убыванию полей соответствующей таблицы
- Средства работы со списками имеются в интерфейсе RSS, но их, естественно, нет в SQL

Основные понятия, цели и общая организация System R (49)

(17)
- Соответственно, эти средства используются только внутри системы при выполнении запросов
  - в частности, один из наиболее эффективных алгоритмов выполнения соединений основан на использовании отсортированных списков кортежей
- Публикации по System R не дают точного представления о структурах данных, используемых при организации списков, но исходя из здравого смысла можно предположить, что
  - они устроены не так, как таблицы (например, для кортежа, входящего в список, не требуется адресация через tid), и что
  - располагаются они во временных файлах (в случае сбоя системы все временные объекты пропадают

Основные понятия, цели и общая организация System R (50)

Организация внешней памяти в базах данных System R (28) Интерфейс RSS (1)

- Описываемый интерфейс RSS не соответствует в точности ни одной из публикаций, посвященных System R, а является скорее некоторой компиляцией, согласующейся с завершающими публикациями
- На уровне RSS отсутствует именование объектов базы данных, употребляемое на уровне SQL
  - Вместо имен объектов используются их уникальные идентификаторы, являющиеся прямыми или косвенными адресами внутренних описателей объектов на внешней памяти для постоянных объектов или в основной памяти для временных объектов
  - Замена имен объектов базы данных на их идентификаторы производится компилятором SQL на основе информации, черпаемой им из системных таблиц-каталогов

Основные понятия, цели и общая организация System R (51)

Организация внешней памяти в базах данных System R (29) Интерфейс RSS (2)

- Можно выделить следующие группы операций:
  - операции сканирования таблиц и списков;
  - операции создания и уничтожения постоянных и временных объектов базы данных;
  - операции модификации таблиц и списков;
  - операция добавления поля к таблицы;
  - операции управления прохождением транзакций;
  - операция явной синхронизации

Основные понятия, цели и общая организация System R (52)

Организация внешней памяти в базах данных System R (30) Интерфейс RSS (3)

- **Операции сканирования таблиц и списков**
- Операции группы сканирования позволяют последовательно, в порядке, определяемом типом сканирования, прочитать кортежи таблицы или списка, удовлетворяющие требуемым условиям
- Группа включает операции OPEN, NEXT и CLOSE, означающие, соответственно,
  - начало сканирования,
  - требование чтения следующего кортежа, удовлетворяющего условиям, и
  - конец сканирования
- Для таблицы возможны два режима сканирования: прямое сканирование и сканирование через индекс

Основные понятия, цели и общая организация System R (53)

- При прямом сканировании единственным параметром операции OPEN является идентификатор таблицы
  - включающий и идентификатор сегмента, в котором эта таблица хранится
- По причине того, что в System R допускается размещение в одной странице данных кортежей нескольких таблиц, прямое сканирование предполагает
  - последовательный просмотр всех страниц сегмента с выделением в них кортежей, входящих в данную таблицу;
  - это очень дорогой способ сканирования таблицы
- При этом порядок выборки кортежей определяется их физическим размещением в страницах сегмента, т.е. предопределен системой

Основные понятия, цели и общая организация System R (54)

Организация внешней памяти в базах данных System R (32) Интерфейс RSS (5)

- При начале сканирования таблицы через индекс в число параметров операции OPEN входит идентификатор какого-либо индекса, определенного ранее на полях этой таблицы
  - Кроме того, можно указать диапазон сканирования в терминах значений поля (полей), составляющего ключ индекса
- При открытии сканирования через индекс производится начальная установка указателя сканирования в позицию листа B-дерева индекса, соответствующую левой границе заданного диапазона
- Процесс сканирования состоит в последовательном продвижении по листовым вершинам индекса до достижения правой границы диапазона сканирования с выборкой идентификаторов кортежей и чтением соответствующих кортежей

Основные понятия, цели и общая организация System R (55)

Организация внешней памяти в базах данных System R (33) Интерфейс RSS (6)

- Легко видеть, что в худшем случае может потребоваться столько чтений страниц данных из внешней памяти, сколько идентификаторов кортежей было встречено, т.е.
  - эффективность сканирования по индексу определяется «широтой» заданного диапазона сканирования
- При этом, конечно, имеется то преимущество, что порядок сканирования соответствует порядку возрастания или убывания значений ключа индекса

Основные понятия, цели и общая организация System R (56)

Организация внешней памяти в базах данных System R (34) Интерфейс RSS (7)

- Наконец, при сканировании списка, как и при прямом сканировании таблицы, единственным параметром операции OPEN является идентификатор списка, но,
  - в отличие от прямого сканирования таблицы это сканирование максимально эффективно:
    - читаются только страницы, содержащие кортежи из данного списка, и
    - порядок сканирования совпадает с порядком занесения кортежей в список или порядком списка, если он упорядочен

Основные понятия, цели и общая организация System R (57)

Организация внешней памяти в базах данных System R (35) Интерфейс RSS (8)

- Наконец, при сканировании списка, как и при прямом сканировании таблицы, единственным параметром операции OPEN является идентификатор списка, но,
  - в отличие от прямого сканирования таблицы это сканирование максимально эффективно:
    - читаются только страницы, содержащие кортежи из данного списка, и
    - порядок сканирования совпадает с порядком занесения кортежей в список или порядком списка, если он упорядочен
- В результате успешного выполнения операции открытия сканирования вырабатывается и возвращается идентификатор сканирования, который используется в качестве аргумента других операций этой группы

Основные понятия, цели и общая организация System R (58)

Организация внешней памяти в базах данных System R (36) Интерфейс RSS (9)

- Операция NEXT выполняет чтение следующего кортежа указанного сканирования, удовлетворяющего условию данной операции
- Условие представляет собой дизъюнктивную нормальную форму простых условий, накладываемых на значения указанных полей таблицы
- Простое условие – это условие вида номер-поля op константа, где op – операция сравнения
  - <, <=, >, >=, =, !=
- Общее условие является параметром операции NEXT

Основные понятия, цели и общая организация System R (59)

Организация внешней памяти в базах данных System R (37) Интерфейс RSS (10)

- Семантика операции NEXT следующая:
- начиная с текущей позиции сканирования выбираются кортежи таблицы в порядке, определяемом типом сканирования, до тех пор, пока не встретится кортеж, значения полей которого удовлетворяют указанному условию
  - Этот кортеж и является результатом операции
- Если при выборке кортежа достигается правая граница диапазона сканирования
  - правая граница значения ключа при сканировании через индексу или
  - последний кортеж таблицы или списка при прямом сканировании, то вырабатывается особый признак результата
- После этого единственным разумным действием является закрытие сканирования – операция CLOSE

Основные понятия, цели и общая организация System R (60)

- Операция CLOSE может быть выполнена в данной транзакции по отношению к любому ранее открытому сканированию независимо от его состояния,
  - т.е. независимо от того, достигнута ли при сканировании правая граница диапазона сканирования
- Параметром операции является идентификатор сканирования, и
- ее выполнение приводит к тому, что этот идентификатор становится недействительным
  - и, соответственно, уничтожаются служебные структуры памяти RSS, относящиеся к данному сканированию

Основные понятия, цели и общая организация System R (61)

Организация внешней памяти в базах данных System R (38) Интерфейс RSS (11)

- **Операции создания и уничтожения постоянных и временных объектов базы данных**
- Группа операций создания и уничтожения постоянных и временных объектов базы данных включает операции создания
  - таблиц (CREATE TABLE),
  - списков (CREATE LIST),
  - индексов (CREATE IMAGE)

  и уничтожения любого из подобных объектов
  - (DROP TABLE, DROP LIST и DROP IMAGE)
- Входным параметром операций создания таблиц и списков является спецификатор структуры объекта,
  - т.е. число полей объекта и спецификаторы их типов

Основные понятия, цели и общая организация System R (62)

Организация внешней памяти в базах данных System R (39) Интерфейс RSS (12)

- Кроме того, при спецификации полей таблицы указывается разрешение или запрещение наличия неопределенных значений полей в кортежах этой таблицы или списка
- Неопределенные значения кодируются специальным образом
- Любая операция сравнения константы данного типа с неопределенным значением по определению вырабатывает значение *false*, кроме операции сравнения на совпадение со специальной литеральной константой NULL

Основные понятия, цели и общая организация System R (63)

Организация внешней памяти в базах данных System R (40) Интерфейс RSS (13)

- В результате выполнения этих операций заводится описатель в служебной таблице описателей таблиц или основной памяти
  - в зависимости от того, создается ли постоянный объект или временный,
- и вырабатывается идентификатор объекта, который служит входным параметром других операций, относящихся к соответствующему объекту
  - в частности, параметром операции OPEN при открытии сканирования объекта

Основные понятия, цели и общая организация System R (64)

Организация внешней памяти в базах данных System R (41) Интерфейс RSS (14)

- Входными параметрами операции CREATE IMAGE являются
  - идентификатор таблицы, для которой создается индекс,
  - список номеров полей, значения которых составляют ключ индекса, и
  - признаки упорядочения по возрастанию или убыванию для всех полей, составляющих ключ
- Кроме того, может быть указан признак уникальности индекса
  - т.е. запрещения наличия в данном индексе ключей-дубликатов
- Если операция выполняется по отношению к пустой в этот момент таблице, то выполнение операции такое же простое, как и для операций создания таблиц и списков:
  - создается описатель в служебной таблице описателей индексов и возвращается идентификатор индекса
    - который, в частности, используется в качестве аргумента операции открытия сканирования таблицы через индекс

Основные понятия, цели и общая организация System R (65)

Организация внешней памяти в базах данных System R (42) Интерфейс RSS (15)

- Если же к моменту создания индекса соответствующая таблица не пуста (а это допускается), то операция становится существенно более дорогостоящей,
  - поскольку при ее выполнении происходит реальное создание B-дерева индекса, что требует, по меньшей мере, одного последовательного просмотра таблицы
- При этом, если создаваемый индекс имеет признак уникальности, то это контролируется при создании B-дерева,
  - и если уникальность нарушается, то операция не выполняется (т.е. индекс не создается)
- Из этого следует, что хотя создание индексов в динамике не запрещается, более эффективно создавать все индексы на данной таблице до ее заполнения
- Заметим, что создание кластеризованного индекса для непустой таблицы запрещено,
  - поскольку соответствующую кластеризацию таблицы без ее реструктуризации получить невозможно

Основные понятия, цели и общая организация System R (66)

Организация внешней памяти в базах данных System R (43) Интерфейс RSS (16)

- Операции DROP TABLE, DROP LIST и DROP IMAGE могут быть выполнены в любой момент независимо от состояния объектов
- Выполнение операции приводит к уничтожению соответствующего объекта и, вследствие этого,
  - недействительности его идентификатора.
- Следует отметить, что массовые операции над постоянными объектами (CREATE IMAGE и DROP TABLE) требуют дополнительных накладных расходов в связи с необходимостью
  - обеспечения возможности откатов транзакции, для чего требуется выполнение массовых обратных действий
- Особенно сильно это затрагивает операцию уничтожения непустых таблиц,
  - поскольку требует журнализации всех кортежей, содержащихся в них к моменту уничтожения
- Поэтому, хотя уничтожение непустых таблиц и не запрещено, нужно иметь в виду, что это очень дорогостоящая операция

Основные понятия, цели и общая организация System R (67)

Организация внешней памяти в базах данных System R (44) Интерфейс RSS (17)

- **Операции модификации таблиц и списков**
- Группа операций модификации таблиц и списков включает операции
  - вставки кортежа в таблицу или список (INSERT),
  - удаления кортежа из таблицы (DELETE) и
  - обновления кортежа в таблице (UPDATE)
- Параметрами операции вставки кортежа являются
  - идентификатор таблицы или списка и
  - набор значений полей кортежа
- Среди значений полей могут быть литеральные неопределенные значения NULL
  - Естественно, при выполнении операции контролируется допустимость неопределенных значений в соответствующих полях

Основные понятия, цели и общая организация System R (68)

Организация внешней памяти в базах данных System R (45) Интерфейс RSS (18)

- При занесении кортежа в кластеризованную таблицу поиск места в сегменте под кортеж производится с использованием кластеризованного индекса:
  - система пытается вставить кортеж в страницу данных, уже содержащую кортежи с теми же или близкими значениями полей кластеризации
- При занесении кортежа в некластеризованную таблицу место под кортеж выделяется в первой подходящей странице данных
- Наконец, при вставке кортежа в список он помещается в конец списка

Основные понятия, цели и общая организация System R (69)

Организация внешней памяти в базах данных System R (46) Интерфейс RSS (19)

- При занесении кортежа в таблицу производится коррекция всех индексов, определенных на этой таблице
- Реально это выражается во вставке новой записи во все B-деревья индексов
- При этом могут произойти переполнения одной или нескольких страниц индекса, что вызовет
  - переливание части записей в соседние страницы или
  - расщепление страниц
- Если индекс определен с атрибутом уникальности, то проверяется соблюдение этого условия, и
  - если оно нарушено, операция вставки считается невыполненной
- Из этого видно, что операция вставки кортежа тем более накладна, чем больше индексов определено для данной таблицы
  - это относится и к операциям удаления и модификации кортежей

Основные понятия, цели и общая организация System R (70)

Организация внешней памяти в базах данных System R (47) Интерфейс RSS (20)

- В результате успешного выполнения операции вставки кортежа в таблицу вырабатывается идентификатор нового кортежа, который
  - выдается в качестве результата операции и
  - может быть в дальнейшем использован как прямой параметр операций удаления и модификации кортежей таблицы
- При занесении кортежа в список значение идентификатора кортежа не вырабатывается
  - для списков допускается только последовательное сканирование и добавление новых кортежей в конец списка;
  - над ними нельзя определить индексов, и поэтому косвенная адресация кортежей списков через их идентификаторы не требуется

Основные понятия, цели и общая организация System R (71)

- Операции удаления и модификации кортежей допускаются только для кортежей таблиц
- Естественно, что для выполнения этих операций необходимо идентифицировать соответствующий кортеж
- В интерфейсе RSS допускаются два способа такой идентификации:
  - с помощью идентификатора кортежа (явная адресация) и
  - с использованием идентификатора открытого к этому времени сканирования
- Первый вариант возможен, поскольку идентификатор кортежа сообщается как ответный параметр операции занесения кортежа в постоянную таблицу

Основные понятия, цели и общая организация System R (72)

Организация внешней памяти в базах данных System R (49) Интерфейс RSS (22)

- При идентификации кортежа с помощью идентификатора сканирования имеется в виду кортеж, прочитанный с помощью последней операции NEXT
- Если при такой идентификации выполняется операция DELETE или операция UPDATE, задевающая порядок сканирования
  - т.е. сканирование ведется по индексу и операция модификации меняет поле кортежа, входящее в состав ключа этого индекса,

  то текущий кортеж сканирования теряется, и
  - его идентификатор нельзя использовать для идентификации кортежа до выполнения следующей операции NEXT

Основные понятия, цели и общая организация System R (73)

Организация внешней памяти в базах данных System R (50) Интерфейс RSS (23)

- Единственным параметром операции DELETE является идентификатор кортежа или идентификатор сканирования
- Параметры операции UPDATE включают, кроме этого, спецификацию изменяемых полей кортежа
  - список номеров полей и их новых значений
- Среди значений могут находиться литеральные изображения неопределенных значений,
  - если соответствующие поля таблицы допускают хранение неопределенных значений
- При выполнении операции DELETE производится коррекция всех индексов, определенных на данной таблице
- Операция UPDATE также может повлечь коррекцию индексов, если затрагивает поля, входящие в состав их ключей

Основные понятия, цели и общая организация System R (74)

Организация внешней памяти в базах данных System R (51) Интерфейс RSS (24)

- Кроме описанных «атомарных» операций сканирования и модификации таблиц и списков, интерфейс RSS включает одну «макрооперацию» BUILDLIST,
  - позволяющую за одно обращение к RSS построить список, отсортированный в соответствии со значениями заданных полей
- Эта операция включает
  - сканирование заданной таблицы или списка,
  - создание нового списка, в который включаются указанные поля выбираемых кортежей, и
  - сортировку построенного списка в соответствии со значениями указанных полей
- Идентификатор заново построенного отсортированного списка является ответным параметром операции

Основные понятия, цели и общая организация System R (75)

Организация внешней памяти в базах данных System R (52) Интерфейс RSS (25)

- Соответственно, параметрами операции BUILDLIST являются
  - набор параметров для открытия сканирования
  - допускается любой способ сканирования,
  - список номеров полей, составляющих кортежи нового списка, и
  - список номеров полей, по которым нужно производить сортировку
- Как и в случае создания нового индекса, можно отдельно для каждого из этих полей указать требование к сортировке по возрастанию или убыванию значений данного поля
- Отдельным параметром операции BUILDLIST является признак, в соответствии со значением которого
  - в новом списке допускаются или не допускаются кортежи-дубликаты

Основные понятия, цели и общая организация System R (76)

Организация внешней памяти в базах данных System R (53) Интерфейс RSS (26)

- **Операция добавления поля к существующей таблице**
- Операция RSS добавления поля к существующей таблице позволяет в динамике изменять схему таблицы
- Параметрами операции CHANGE являются
  - идентификатор существующей таблицы и
  - спецификация нового поля (его тип)
- При выполнении операции изменяется только описатель данной таблице в служебной таблице описателей таблиц
- До выполнения первой операции UPDATE, затрагивающей новое поле таблицы, реально ни в одном кортеже таблицы память под новое поле выделяться не будет
- По умолчанию значения нового поля во всех кортежах таблицы, в которые еще не производилось явное занесение значения, считаются неопределенными
  - Тем самым, ни для одного поля, динамически добавленного к существующей таблице, не может быть запрещено хранение неопределенных значений

Основные понятия, цели и общая организация System R (77)

Организация внешней памяти в базах данных System R (54) Интерфейс RSS (27)

- **Операции управления прохождением транзакций**
- Каждая операция RSS выполняется в пределах некоторой транзакции
- Интерфейс RSS включает набор операций управления прохождением транзакции:
  - начать транзакцию (BEGIN TRANSACTION),
  - закончить транзакцию (END TRANSACTION),
  - установить точку сохранения (SAVE) и
  - выполнить откат до указанной точки сохранения или до начала транзакции (RESTORE)
- Это не отмечалось раньше, но на самом деле при вызове любой операции функции RSS, кроме BEGIN TRANSACTION, должен указываться еще один параметр – идентификатор транзакции
  - Этот идентификатор и вырабатывается при выполнении операции BEGIN TRANSACTION, которая сама входных параметров не требует

Основные понятия, цели и общая организация System R (78)

Организация внешней памяти в базах данных System R (55) Интерфейс RSS (28)

- В любой точке транзакции до выполнения операции END TRANSACTION может быть выполнен откат данной транзакции,
  - т.е. обратное выполнение всех изменений, произведенных в данной транзакции, и
  - восстановление состояния позиций сканирования
- Откат может быть произведен
  - до начала транзакции
  - в этом случае о восстановлении позиций сканирования говорить бессмысленно

  или
  - до установленной ранее в транзакции точки сохранения

Основные понятия, цели и общая организация System R (79)

Организация внешней памяти в базах данных System R (56) Интерфейс RSS (29)

- Точка сохранения устанавливается с помощью операции SAVE
- При выполнении этой операции запоминаются
  - состояние сканов данной транзакции, открытых к моменту выполнения SAVE, и
  - координаты последней записи об изменениях в базе данных в журнале, произведенной от имени данной транзакции
- Ответным параметром операции SAVE
  - а прямых параметров, кроме идентификатора транзакции, она не требует

является идентификатор точки сохранения

Основные понятия, цели и общая организация System R (80)

Организация внешней памяти в базах данных System R (57) Интерфейс RSS (30)

- Этот идентификатор в дальнейшем может быть использован как аргумент операции RESTORE,
  - при выполнении которой производится восстановление базы данных по журналу
    - с использованием записей о ее изменениях от данной транзакции

    до того состояния, в котором находилась база данных к моменту установки указанной точки сохранения
- Кроме того, по локальной информации в оперативной памяти, привязанной к транзакции, восстанавливается состояние ее сканов
- Откат к началу транзакции инициируется также вызовом операции RESTORE, но
  - с указанием некоторого предопределенного идентификатора точки сохранения

Основные понятия, цели и общая организация System R (81)

Организация внешней памяти в базах данных System R (58) Интерфейс RSS (31)

- При выполнении своих транзакций пользователи System R изолированы один от другого,
  - т.е. не ощущают того, что система функционирует в многопользовательском режиме
- Это достигается за счет наличия в RSS механизма неявной синхронизации
- До конца транзакции никакие изменения базы данных, произведенные в пределах этой транзакции, не могут быть использованы в других транзакциях
  - попытка использования таких данных приводит к временным синхронизационным блокировкам этих транзакций

Основные понятия, цели и общая организация System R (82)

Организация внешней памяти в базах данных System R (59) Интерфейс RSS (32)

- При выполнении операции END TRANSACTION происходит "фиксация" изменений, произведенных в данной транзакции,
  - т.е. они становятся видимыми в других транзакциях
- Реально это означает снятие синхронизационных блокировок с объектов базы данных, изменявшихся в транзакции
- Из этого следует, что после выполнения END TRANSACTION невозможны индивидуальные откаты данной транзакции
  - RSS просто делает недействительным идентификатор данной транзакции, и
    - после выполнения операции окончания транзакции отвергает все операции с таким идентификатором

Основные понятия, цели и общая организация System R (83)

Организация внешней памяти в базах данных System R (60) Интерфейс RSS (33)

- **Операция явной синхронизации**
- Последняя операция интерфейса RSS – операция явной синхронизации LOCK
- Эта операция позволяет установить явный синхронизационную блокировку указанной таблицы
- параметром операции является идентификатор таблицы
- Выполнение операции LOCK гарантирует, что никакая другая транзакция до конца данной не сможет
  - изменить данную таблицу
    - вставить в него новый кортеж,
    - удалить или
    - модифицировать существующий,
    - если установлена ее блокировка в режиме чтения,
  - или даже прочитать любой кортеж этой таблицы,
    - если установлена монопольная блокировка

Основные понятия, цели и общая организация System R (84)

Организация внешней памяти в базах данных System R (61) Интерфейс RSS (34)

- Из всего, что говорилось раньше по поводу подхода к синхронизации в System R и соответствующего разбиения системы на уровни, следует нелогичность наличия этой операции в интерфейсе RSS
- На самом деле, логически эта операция избыточна,
  - т.е. если бы ее не было, можно вполне реализовать SQL с использованием оставшейся части операций
- Предварительно
  - до подробного обсуждения средств управления транзакциями

  заметим, что операция LOCK введена в интерфейс RSS для возможности оптимизации выполнения запросов

Основные понятия, цели и общая организация System R (85)

Организация внешней памяти в базах данных System R (62) Интерфейс RSS (35)

- Дело в том, что, как видно из описания интерфейса RSS, этот интерфейс является покортежным
  - Следовательно, и информация для синхронизации носит достаточно узкий характер
- В то же время на уровне SQL имеется более полная информация
  - Например, если обрабатывается предложение SQL DELETE FROM EMP, то известно, что будут удалены все кортежи указанной таблицы
- Понятно, что как бы не реализовывался механизм синхронизации в RSS, в данном случае выгоднее сообщить сразу, что изменения касаются всей таблицы

Основные понятия, цели и общая организация System R (86)

Организация внешней памяти в базах данных System R (63) Интерфейс RSS (35)

- Дело в том, что, как видно из описания интерфейса RSS, этот интерфейс является покортежным
  - Следовательно, и информация для синхронизации носит достаточно узкий характер
- В то же время на уровне SQL имеется более полная информация
  - Например, если обрабатывается предложение SQL DELETE FROM EMP, то известно, что будут удалены все кортежи указанной таблицы
- Понятно, что как бы не реализовывался механизм синхронизации в RSS, в данном случае выгоднее сообщить сразу, что изменения касаются всей таблицы
- Но ситуации, в которых очевидна выгода от использования явной синхронизации, достаточно редки
- Пользоваться этим средством можно только очень осмотрительно, потому что неоправданные захваты таких крупных объектов могут
  - резко ограничить степень асинхронности выполнения транзакций

- Обсудим основные подходы к организации данных во внешней памяти, принятые в современных SQL-ориентированных СУБД
- В большинстве случаев они основаны на идеях, заложенных в System R, хотя, конечно, в любой развитой системе имеются собственные приемы, которые здесь обсуждаться не будут
- SQL-ориентированные СУБД обладают рядом особенностей, влияющих на организацию внешней памяти
- Наиболее важными являются следующие особенности

- Наличие двух уровней системы:
  - уровня непосредственного управления данными во внешней памяти
    - а также обычно управления буферами оперативной памяти,
    - управления транзакциями и
    - журнализацией изменений БД и
  - языкового уровня уровня, реализующего язык SQL
  - При такой организации подсистема нижнего уровня должна поддерживать во внешней памяти набор базовых структур, конкретная интерпретация которых входит в число функций подсистемы верхнего уровня.
- Поддержка таблиц-каталогов
  - Информация, связанная с именованием объектов базы данных и их конкретными свойствами
    - например, структура ключа индекса

    поддерживается подсистемой языкового уровня
  - С точки зрения структур внешней памяти таблица-каталог ничем не отличается от обычной таблицы базы данных

- Регулярность структур данных
  - Поскольку основным объектом модели данных SQL является плоская таблица, основной набор объектов внешней памяти может иметь очень простую регулярную структуру.
  - Необходимость обеспечения возможности эффективного выполнения операторов языкового уровня
    - как над одной таблицей (простые селекция и проекция),
    - так и над несколькими таблицами
      - наиболее распространено и трудоемко соединение нескольких таблиц
  - Для этого во внешней памяти должны поддерживаться дополнительные «управляющие» структуры – индексы.
- Наконец, для выполнения требования надежного хранения баз данных необходимо поддерживать избыточность хранения данных,
  - что обычно реализуется в виде журнала изменений базы данных

- Соответственно возникают следующие разновидности объектов во внешней памяти базы данных:
  - строки таблиц
    - основная часть базы данных, большей частью непосредственно видимая пользователям;
  - управляющие структуры
    - индексы, создаваемые по инициативе пользователя (администратора) или верхнего уровня системы из соображений повышения эффективности выполнения запросов и обычно автоматически поддерживаемые нижним уровнем системы;
  - журнальная информация,
    - поддерживаемая для удовлетворения потребности в надежном хранении данных;
  - служебная информация,
    - поддерживаемая для удовлетворения внутренних потребностей нижнего уровня системы (например, информация о свободной памяти)

Хранение таблиц (1)

- Существуют два принципиальных подхода к физическому хранению таблиц
- Наиболее распространенным является покортежное хранение таблиц
  - единицей физического хранения является кортеж
- Естественно, это обеспечивает быстрый доступ к целому кортежу, но при этом
  - во внешней памяти дублируются общие значения разных кортежей одной таблицы и, вообще говоря,
  - могут потребоваться лишние обмены с внешней памятью, если нужна часть кортежа
- Альтернативным (менее распространенным) подходом является хранение таблицы по столбцам,
  - т.е. единицей хранения является столбец таблицы с исключенными дубликатами

Хранение таблиц (2)

- Естественно, что при такой организации суммарно в среднем тратится меньше внешней памяти, поскольку
  - дубликаты значений не хранятся;
  - за один обмен с внешней памятью в общем случае считывается больше полезной информации
- Дополнительным преимуществом является возможность использования значений столбца таблицы для оптимизации выполнения операций соединения
- Но при этом требуются существенные дополнительные действия для сборки целого кортежа (или его части)

Хранение таблиц (4)

- Поскольку гораздо более распространено хранение по строкам, рассмотрим немного более подробно этот способ хранения таблиц
- Типовой, унаследованной от System R, структурой страницы данных является та, которая показана на рисунке
- Эту организацию хранения кортежей можно в целом охарактеризовать следующим образом:
  - ✓ Каждый кортеж обладает уникальным идентификатором (tid), не изменяемым во все время существования кортежа и позволяющим выбрать кортеж в основную память не более чем за два обращения к внешней памяти
  - ✓ Обычно каждый кортеж хранится целиком в одной странице
    - o Из этого следует, что максимальная длина кортежа любой таблицы ограничена размерами страницы
    - o Возникает вопрос: как быть с «длинными» данными, которые в принципе не помещаются в одной странице?

Хранение таблиц (5)

- Наиболее простым решением является хранение таких данных в отдельных (вне базы данных) файлах с заменой «длинного» данного в кортеже на имя соответствующего файла
- В некоторых системах такие данные хранились внутри базы данных в отдельном наборе страниц внешней памяти, связанном физическими ссылками
- Оба эти решения сильно ограничивают возможность работы с длинными данными
  - как, например, удалить несколько байт из середины 2-мегабайтной строки?
- В настоящее время все чаще используется метод, предложенный много лет тому назад в проекте Exodus, когда «длинные» данные организуются в виде B-деревьев последовательностей байтов

Хранение таблиц (6)

- Как правило, в одной странице данных хранятся кортежи только одной таблицы
  - ✓ Существуют, однако, варианты с возможностью хранения в одной странице кортежей нескольких таблиц
  - ✓ Это вызывает некоторые дополнительные расходы по части служебной информации
    - при каждом кортеже нужно хранить информацию о соответствующей таблице,

    но зато иногда позволяет резко сократить число обменов с внешней памятью при выполнении соединений
- Изменение схемы хранимой таблицы с добавлением нового поля не вызывает потребности в физической реорганизации таблицы
  - ✓ Достаточно лишь изменить информацию в описателе таблицы и расширять кортежи только при занесении информации в новое поле

Хранение таблиц (7)

- Поскольку таблицы могут содержать неопределенные значения, необходима соответствующая поддержка на уровне хранения
  - ✓ Обычно это достигается путем хранения соответствующей шкалы при каждом кортеже, который в принципе может содержать неопределенные значения
- Проблема распределения памяти в страницах данных связана с проблемами синхронизации и журнализации и не всегда тривиальна
  - ✓ Например, если в ходе выполнения транзакции некоторая страница данных опустошается, то ее нельзя перевести в статус свободных страниц до конца транзакции,
    - поскольку при откате транзакции удаленные при прямом выполнении транзакции и восстановленные при ее откате кортежи должны получить те же самые идентификаторы

Хранение таблиц (8)

- Распространенным способом повышения эффективности СУБД является кластеризация таблицы по значениям одного или нескольких столбцов
  - Полезной для оптимизации соединений является совместная кластеризация нескольких таблиц.
- С целью использования возможностей распараллеливания обменов с внешней памятью иногда применяют схему декластеризованного хранения таблиц:
  - кортежи с общим значением столбца декластеризации размещают на разных дисковых устройствах, обмены с которыми можно выполнять в параллель

Хранение таблиц (9)

- Что же касается хранения таблицы по столбцам, то основная идея состоит в совместном хранении всех значений одного (или нескольких) столбцов
- Для каждого кортежа таблицы хранится кортеж той же степени, состоящий из ссылок на места расположения соответствующих значений столбцов

Индексы (1)

- Как бы не были организованы индексы в конкретной СУБД, их основное назначение состоит в обеспечении эффективного прямого доступа к кортежу таблицы по ключу
- Обычно индекс определяется для одной таблицы, и ключом является значение ее поля (возможно, составного)
- Если ключом индекса является возможный ключ таблицы, то индекс должен обладать свойством уникальности,
  - т.е. не содержать дубликатов ключа
- На практике ситуация выглядит обычно противоположно:
  - при объявлении первичного ключа таблицы автоматически заводится уникальный индекс, а
  - единственным способом объявления возможного ключа, отличного от первичного, является явное создание уникального индекса
- Это связано с тем, что для проверки сохранения свойства уникальности возможного ключа, так или иначе, требуется индексная поддержка

Индексы (2)

- Поскольку при выполнении многих операций уровня SQL требуется сортировка кортежей таблиц в соответствии со значениями некоторых полей, полезным свойством индекса является
  - обеспечение последовательного просмотра кортежей таблицы в заданном диапазоне значений ключа в порядке возрастания или убывания значений ключа.
- Наконец, одним из способов оптимизации выполнения эквисоединения таблиц
  - наиболее распространенная из числа дорогостоящих операций

  является организация так называемых мультииндексов для нескольких таблиц, обладающих общими атрибутами
- Любой из этих атрибутов (или их набор) может выступать в качестве ключа мультииндекса
- Значению ключа сопоставляется набор кортежей всех связанных мультииндексом таблиц, значения выделенных атрибутов которых совпадают со значением ключа

Индексы (3)

- Общей идеей любой организации индекса, поддерживающего прямой доступ по ключу и последовательный просмотр в порядке возрастания или убывания значений ключа является
  - хранение упорядоченного списка значений ключа с привязкой к каждому значению ключа списка идентификаторов кортежей
- Одна организация индекса отличается от другой главным образом в способе поиска ключа с заданным значением

Индексы (4) B+-деревья (1)

- Наиболее популярным подходом к организации индексов в базах данных является использование техники B+-деревьев
- Техника B- и B+-деревьев была предложена в начале 1970-х гг. Рудольфом Байером (Rudolf Bayer) и Эдом Маккрейтом (Ed McCreight)
- С точки зрения внешнего логического представления B-дерево – это сбалансированное сильно ветвистое дерево во внешней памяти
- Сбалансированность означает, что длина пути от корня дерева к любому его листу одна и та же
- Ветвистость дерева – это свойство каждого узла дерева ссылаться на большое число узлов-потомков
- С точки зрения физической организации B-дерево представляется как мультисписочная структура страниц внешней памяти,
  - т.е. каждому узлу дерева соответствует блок внешней памяти (страница)
- В B+-дереве внутренние и листовые страницы обычно имеют разную структуру

Индексы (5) B+-деревья (2)

N_1 ключ$_1$ N_2 ключ$_2$ N_3 ключ$_3$ . . . N_m ключ$_m$ N_{m+1}

- Типовая структура внутренней страницы B+-дерева
- Выдерживаются следующие свойства:
- ✓ ключ$_1$ ≤ ключ$_2$ ≤ ... ≤ ключ$_m$;
- ✓ в странице дерева N_m находятся ключи k со значениями ключ$_m$ <= k <= ключ$_{m+1}$

ключ$_1$ список$_1$ ключ$_2$ список$_2$ . . . ключ$_k$ список$_k$

- Структура листовой страницы B+-дерева.
- Листовая страница обладает следующими свойствами:
- ✓ ключ$_1$ < ключ$_2$ < ... < ключ$_k$;
- ✓ список$_r$ – упорядоченный список идентификаторов кортежей (tid), включающих значение ключ$_r$;
- ✓ листовые страницы связаны одно- или двунаправленным списком

- Поиск в B+-дереве – это прохождение от корня к листу в соответствии с заданным значением ключа
- Заметим, что поскольку B+-деревья являются сильно ветвистыми и сбалансированными,
  - для выполнения поиска по любому значению ключа потребуется одно и то же (и обычно небольшое) число обменов с внешней памятью
- Более точно, в сбалансированном дереве, где длины всех путей от корня к листу одни и те же, если во внутренней странице помещается n ключей, то при хранении m записей требуется дерево глубиной $\log_n(m)$
- Если n достаточно велико (обычный случай), то глубина дерева невелика, и производится быстрый поиск

- Основной «изюминкой» B+-деревьев является автоматическое поддержание свойства сбалансированности
- Рассмотрим, как это делается при выполнении операций занесения и удаления записей.
- При занесение новой записи выполняются следующие действия
  - Поиск листовой страницы
    - Фактически, производится обычный поиск по ключу
    - Если в B+-дереве не содержится ключ с заданным значением, то будет получен номер страницы, в которой ему надлежит содержаться, и соответствующие координаты внутри страницы
  - Помещение записи на место
    - Естественно, что вся работа производится в буферах оперативной памяти
    - Листовая страница, в которую требуется занести запись, считывается в буфер, и в нем выполняется операция вставки
    - Размер буфера должен превышать размер страницы внешней памяти

- Если после выполнения вставки новой записи размер используемой части буфера не превосходит размера страницы, то
  - на этом выполнение операции занесения записи заканчивается
  - Буфер может быть немедленно вытолкнут во внешнюю память, или
  - временно сохранен в основной памяти в зависимости от политики управления буферами
- Если же возникло переполнение буфера (т.е. размер его используемой части превосходит размер страницы), то
  - выполняется расщепление страницы
  - Для этого запрашивается новая страница внешней памяти,
  - используемая часть буфера разбивается, грубо говоря, пополам (так, чтобы вторая половина также начиналась с ключа), и
  - вторая половина записывается во вновь выделенную страницу, а
  - в старой странице модифицируется значение размера свободной памяти
  - Естественно, модифицируются ссылки по списку листовых страниц

- Чтобы обеспечить доступ от корня дерева к заново заведенной странице, необходимо
  - ✓ соответствующим образом модифицировать внутреннюю страницу, являющуюся предком ранее существовавшей листовой страницы,
    - o т.е. вставить в нее соответствующее значение ключа и ссылку на новую страницу
  - ✓ При выполнении этого действия может снова произойти переполнение теперь уже внутренней страницы, и она будет расщеплена на две
  - ✓ В результате потребуется вставить значение ключа и ссылку на новую страницу во внутреннюю страницу-предка выше по иерархии и т.д.
- Предельным случаем является переполнение корневой страницы B+-дерева
  - ✓ В этом случае она тоже расщепляется на две, и заводится новая корневая страница дерева, т.е. его глубина увеличивается на единицу

- При удалении записи выполняются следующие действия
  - Поиск записи по ключу
    - Если запись не найдена, то, значит, удалять ничего не нужно.
  - Реальное удаление записи в буфере, в который прочитана соответствующая листовая страница
  - Если после выполнения этой подоперации размер занятой в буфере области оказывается таковым, что
    - его сумма с размером занятой области в листовых страницах, являющихся левым или правым братом данной страницы, больше, чем размер страницы,

    операция завершается

- Иначе производится слияние с правым или левым братом,
  - т.е. в буфере производится новый образ страницы, содержащей общую информацию из данной страницы и ее левого или правого брата
  - Ставшая ненужной листовая страница заносится в список свободных страниц
  - Соответствующим образом корректируется список листовых страниц
- Чтобы устранить возможность доступа от корня к освобожденной странице, нужно удалить соответствующее значение ключа и ссылку на освобожденную страницу из внутренней страницы – ее предка
  - При этом может возникнуть потребность в слиянии этой страницы с ее левым или правыми братьями и т.д.

- Предельным случаем является полное опустошение корневой страницы дерева, которое возможно после слияния последних двух потомков корня
  - В этом случае корневая страница освобождается, а глубина дерева уменьшается на единицу
- Как видно, при выполнении операций вставки и удаления свойство сбалансированности B+-дерева сохраняется, а внешняя память расходуется достаточно экономно

- Проблемой является то, что при выполнении операций модификации слишком часто могут возникать расщепления и слияния
- Чтобы добиться эффективного использования внешней памяти с минимизацией числа расщеплений и слияний, применяются более сложные приемы, в том числе:
  - упреждающие расщепления,
    - т.е. расщепления страницы не при ее переполнении, а несколько раньше, когда степень заполненности страницы достигает некоторого уровня;
  - переливания,
    - т.е. поддержание равновесного заполнения соседних страниц;
  - слияния 3-в-2,
    - т.е. порождение двух листовых страниц на основе содержимого трех соседних

- Следует заметить, что при организации мультидоступа к B+-деревьям, характерного при их использовании в СУБД, приходится решать ряд нетривиальных проблем
- Конечно, грубые решения очевидны, например, возможен монопольный захват B+-дерева (т.е. его корневого блока) на все выполнение операции модификации
- Но существуют и более тонкие решения, рассмотрение которых выходит за пределы материала этой лекции

Индексы (14) Хэширование (1)

- Альтернативным и достаточно популярным подходом к организации индексов является использование техники хэширования
- Это очень обширная тема, которая заслуживает отдельного рассмотрения
- Ограничимся здесь лишь несколькими замечаниями
- Общей идеей методов хэширования является применение к значению ключа некоторой функции свертки (хэш-функции), вырабатывающей значение меньшего размера
- Значение хэш-функции затем используется для доступа к записи

- В самом простом, классическом случае свертка ключа используется как адрес в таблице, содержащей ключи и записи
- Основным требованием к хэш-функции является равномерное распределение значение свертки
  - одним из распространенных видов «хороших» хэш-функций являются функции, выдающие остаток от деления значения ключа на некоторое простое число
- При возникновении коллизий
  - одна и та же свертка для нескольких значений ключа

  образуются цепочки переполнения

- Главным ограничением этого метода является фиксированный размер таблицы
- Если таблица заполнена слишком сильно или переполнена, но возникнет слишком много цепочек переполнения, и главное преимущество хэширования
  - доступ к записи почти всегда за одно обращение к таблице

  будет утрачено
- Расширение таблицы требует ее полной переделки на основе новой хэш-функции
  - со значением свертки большего размера

- Идея доступа к данным на основе хэширования настолько привлекательна
  - потенциальная возможность за одно обращение к памяти получить требуемые данные,

  что от нее невозможно отказаться при работе с данными во внешней памяти
- Исходная идея кажется очевидной:
- если при управлении данными на основе хэширования в основной памяти хэш-функция вырабатывает
  - адрес требуемого элемента,

  то при обращении к внешней памяти необходимо
  - генерировать номер блока дискового пространства, в котором находится запрашиваемый элемент данных
- Основная проблема относится к коллизиям

- Если при работе в основной памяти потенциально возникающими потребностями дополнительного поиска информации при возникновении коллизий можно, вообще говоря, пренебречь
  - поскольку время доступа к основной памяти мало,

  то при использовании внешней памяти любое дополнительное обращение вызывает существенные накладные расходы
- Основные методы хэширования для поиска информации во внешней памяти направлены на решение именно этой задачи

- В основе подхода расширяемого хэширования (Extendible Hashing) лежит принцип использования деревьев цифрового поиска в основной памяти
- В основной памяти поддерживается справочник, организованный на основе бинарного дерева цифрового поиска,
  - ключами которого являются значения хэш-функции, а
  - в листовых вершинах хранятся номера блоков записей во внешней памяти
- В этом случае любой поиск в дереве цифрового поиска является «успешным», т.е. ведет к некоторому блоку внешней памяти
- Входит ли в этот блок искомая запись, обнаруживается уже после прочтения блока в основную память

- Проблема коллизий переформулируется следующим образом
- Как таковых, коллизий не существует
- Может возникнуть лишь ситуация переполнения блока внешней памяти
  - Значение хэш-функции указывает на этот блок, но места для включения записи в нем уже нет
- Эта ситуация обрабатывается так
  - Блок расщепляется на два, и дерево цифрового поиска переформируется соответствующим образом
  - Конечно, при этом может потребоваться расширение самого справочника
- Расширяемое хэширование хорошо работает в условиях динамически изменяемого набора записей в хранимом файле,
  - но требует наличия в основной памяти справочного дерева

- Идея линейного хэширования (Linear Hashing) состоит в том, чтобы можно было обойтись без поддержания справочника в основной памяти
- Основой метода является то, что для адресации блока внешней памяти всегда используются младшие биты значения хэш-функции
- Если возникает потребность в расщеплении, то записи перераспределяются по блокам так, чтобы адресация осталась правильной

- Структура журнала обычно является сугубо частным делом конкретной реализации
- Отметим только самые общие свойства.
- Журнал обычно представляет собой чисто последовательный файл с записями переменного размера, которые можно просматривать в прямом или обратном порядке
- Обмены производятся стандартными порциями (страницами) с использованием буфера оперативной памяти
- В грамотно организованных системах структура (и тем более, смысл) журнальных записей известна только компонентам СУБД, ответственным за журнализацию и восстановление
- Поскольку содержимое журнала является критичным при восстановлении базы данных после сбоев, к ведению файла журнала предъявляются особые требования по части надежности
- В частности, обычно стремятся поддерживать две идентичные копии журнала на разных устройствах внешней памяти

Индексы (23) Служебная информация (1)

- Для корректной работы подсистемы управления данными во внешней памяти необходимо поддерживать информацию, которая используется только этой подсистемой и не видна подсистеме языкового уровня
- Набор структур служебной информации зависит от общей организации системы, но обычно требуется поддержание следующих служебных данных:
  - Внутренние каталоги, описывающие физические свойства объектов базы данных, например,
    - ✓ число атрибутов таблицы,
    - ✓ их размер и, возможно, типы данных;
    - ✓ описание индексов, определенных для данной таблицы и т.д.

- Описатели свободной и занятой памяти в страницах данных
  - Такая информация требуется для нахождения свободного места при занесении кортежа
  - Отдельно приходится решать задачу поиска свободного места в случаях некластеризованных и кластеризованных таблиц
    - в последнем случае приходится дополнительно использовать кластеризованный индекс
  - Как уже отмечалось, нетривиальной является проблема освобождения страницы в условиях мультидоступа

- Связывание страниц одной таблицы
  - Если в одном файле внешней памяти могут располагаться страницы нескольких таблиц (обычно к этому стремятся), то нужно каким-то образом связать страницы одной таблицы
  - Тривиальный способ использования прямых ссылок между страницами часто приводит к затруднениями при синхронизации транзакций
    - например, особенно трудно освобождать и заводить новые страницы таблицы
  - Поэтому стараются использовать косвенное связывание страниц с использованием служебных индексов
  - В частности, известен общий механизм для описания свободной памяти и связывания страниц на основе B-деревьев