

Типовая организация современной системы управления базами данных

Цель: сформировать знание типовой организации современной системы управления базами данных

Файл с точки зрения прикладной программы это именованная область внешней памяти, в которую можно записывать и из которой можно считывать данные.

Правила именования файлов, способ доступа к данным, хранящимся в файле, и структура этих данных зависят от конкретной системы управления файлами и от типа файла. Система управления файлами берет на себя функции распределения внешней памяти, отображения имен файлов в соответствующие адреса во внешней памяти, обеспечения доступа к данным. Однако этого оказывается недостаточно для построения даже простых информационных систем. Системы управления файлами не обеспечивают возможности поддержания логически согласованного набора файлов, обеспечения языка манипулирования данными, восстановления информации после разного рода сбоев, параллельной работы нескольких пользователей. Прикладная программа должна опираться на некоторую систему управления данными, обладающую этими свойствами. Такой системой и является система управления базами данных.

Системы обработки данных включают в себя собственно данные, систему управления базами данных и прикладное программное обеспечение, которое обращается к данным через систему управления БД.

Можно выделить следующие функции систем управления базами данных:

1. управление данными во внешней памяти;
2. управление данными в оперативной памяти;
3. управление транзакциями;
4. журнализация, резервное копирование и восстановление;
5. поддержка языков БД.

Управление данными во внешней памяти (на дисках). СУБД должна предоставлять пользователям возможность:

1. сохранять, извлекать и обновлять данные в базе данных;
2. контролировать доступ к данным;
3. обеспечивать параллельную работу нескольких пользователей;
4. поддерживать целостность данных.

Возможность сохранять, извлекать и обновлять данные в базе данных самая фундаментальная функция СУБД. Причем способ реализации этой функции должен быть скрыт от конечного пользователя (использует ли СУБД файловую систему, как организованы файлы и т.п.).

Контроль доступа к данным это возможность обеспечить только санкционированный доступ к базе данных, используя защиту паролем, поддержку уровней доступа к базе данных и отдельным ее элементам и т.д. Каждый пользователь должен иметь возможность работы только с теми данными из БД, которые доступны для него в соответствии с его пользовательскими правами. Таким образом, обеспечивается безопасность в системах управления данными.

Управление параллельностью заключается в том, что СУБД имеют механизм, который гарантирует корректное обновление данных многими пользователями при одновременном доступе. Коллизии при совместной работе могут привести к нарушению логической целостности данных, поэтому система должна предусматривать меры, не допускающие обновление данных пользователем, пока эти данные используются кем-то еще. В описанных случаях используются так называемые «блокировки». Существуют разные типы блокировок — табличные, страничные, строчные и другие, которые отличаются друг от друга количеством заблокированных записей.

Поддержка целостности данных осуществляется инструментальными средствами контроля для того, чтобы данные и их изменения соответствовали заданным правилам. Целостность БД есть свойство базы данных, означающее, что в ней содержится полная, непротиворечивая и адекватно отражающая предметную область информация. Поддержание целостности БД включает проверку целостности и ее восстановление в случае обнаружения противоречий в базе данных.

Целостное состояние БД описывается с помощью ограничений целостности в виде условий, которым должны удовлетворять хранимые в базе данные. Примером таких условий может служить ограничение диапазонов возможных значений атрибутов объектов, сведения о которых хранятся в БД, или отсутствие повторяющихся записей в таблицах реляционных БД.

Управление данными в оперативной памяти. Размер базы данных, с которой работает СУБД, достаточно велик обычно существенно больше доступного объема оперативной памяти. Понятно, что если при обращении к любому элементу данных будет производиться обмен с внешней памятью, то вся система будет работать со скоростью устройства внешней памяти. Практически единственным способом реального увеличения этой скорости является буферизация данных в оперативной памяти.

Буферы представляют собой области оперативной памяти, предназначенные для ускорения обмена между внешней и оперативной памятью. В буферах временно хранятся фрагменты БД, данные из которых предполагается использовать при обращении к СУБД или планируется записать в базу после обработки.

В развитых СУБД поддерживается собственный набор буферов оперативной памяти с собственными правилами замены буферов. Возможно,

что в будущем объем оперативной памяти компьютеров будет настолько велик, что позволит не беспокоиться о буферизации.

Управление транзакциями. *Транзакция* это совокупность действий над базой данных, рассматриваемых СУБД как единое целое, т.е. последовательность операций с данными, когда выполняются все операции либо ни одна из них (принцип «все или ничего»).

Если транзакция выполняется успешно, то СУБД фиксирует изменения БД, произведенные этой транзакцией, во внешней памяти, либо все изменения в рамках транзакции отменяются и ни одно из них никак не отражается на состоянии БД.

Понятие транзакции необходимо для поддержания логической целостности БД. Примером транзакции является операция перевода денег с одного счета на другой в банковской системе. Нужно либо совершить все действия (увеличить счет одного клиента и уменьшить счет другого), либо не выполнить ни одно из этих действий. Нельзя уменьшить сумму денег на одном счете, но не увеличить сумму денег на другом.

Предположим, что после выполнения первого из действий (уменьшения суммы денег на счете) произошел сбой. Например, могла прерваться связь клиентского компьютера с базой данных или на клиентском компьютере мог произойти системный сбой, что привело к перезагрузке операционной системы. Что в этом случае стало с базой данных? Команда на уменьшение денег на счете первого клиента была выполнена, а вторая команда — на увеличение денег на другом счете — нет, что привело бы к противоречивому, неактуальному состоянию базы данных. Сначала снимают деньги с одного счета, затем добавляют их к другому счету. Если хотя бы одно из действий не выполнится успешно, результат операции окажется неверным и будет нарушен баланс между счетами. Поэтому в этом случае система должна вернуться в предыдущее состояние до начала цепочки операций.

Контроль транзакций важен в однопользовательских и многопользовательских СУБД, где транзакции могут быть запущены параллельно.

При параллельном выполнении нескольких транзакций возможно возникновение конфликтов, разрешение которых является также функцией СУБД. При обнаружении таких случаев обычно производится «откат» транзакции, т. е. отмена изменений, произведенных одной или несколькими транзакциями. Более подробно механизм транзакций и управление транзакциями будет рассмотрен при изучении вопросов обеспечения целостности данных.

Журнализация, резервное копирование и восстановление. При работе ЭВМ возможны сбои (например, из-за отключения электропитания) и повреждение машинных носителей данных. При этом могут быть нарушены данные, что приводит к невозможности дальнейшей работы. Одним из основных требований к СУБД является надежность хранения данных во

внешней памяти, в том числе защита физической и логической целостности. Под надежностью хранения понимается то, что СУБД должна иметь возможность восстановить последнее согласованное состояние БД после любого аппаратного или программного сбоя (логических или физических отказов).

Защита физической целостности включает в себя журнализацию изменений, резервное копирование и восстановление базы данных после сбоев.

Журнализация изменений в простейшем случае заключается в последовательной записи во внешнюю память всех изменений, выполняемых в базе данных. Записывается следующая информация: порядковый номер, тип и время изменения; идентификатор транзакции; объект, подвергшийся изменению (номер хранимого файла и номер блока данных в нем, номер строки внутри блока); предыдущее и новое состояние объекта.

Журнал изменений базы данных это и есть формируемая таким образом информация. Журнал содержит отметки начала и завершения транзакции и отметки принятия контрольной точки. Он является особой частью базы данных, недоступной пользователям СУБД. Ведение журнала поддерживается с особой тщательностью, к которую поступают записи обо всех изменениях основной части базы данных. Иногда создаются и поддерживаются две копии журнала, располагаемые на разных физических дисках.

Резервное копирование базы данных процесс создания копии данных на носителе, предназначенном для восстановления данных в оригинальном или новом месте их расположения в случае их повреждения или разрушения.

Восстановление базы данных это функция СУБД, которая в случае логических и физических сбоев приводит базу данных в актуальное состояние. Как уже отмечалось выше, сбой может произойти в результате выхода из строя системы или запоминающего устройства, возможны ошибки аппаратного и программного обеспечения, которые могут привести к остановке работы СУБД. Во всех подобных случаях СУБД должна предоставить механизм восстановления базы данных и возврата к ее непротиворечивому состоянию.

В случае физического отказа, если повреждены как журнал изменений, так и сама база данных, то восстановление возможно только на момент выполнения резервной копии.

Поддержка языков БД. Для работы с базами данных используются специальные языки, в целом называемые языками баз данных. В современных СУБД обычно поддерживается единый интегрированный язык, содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс с базами данных. Стандартным языком наиболее распространенных в настоящее время реляционных СУБД является язык SQL (Structured Query Language).

Кроме основных функций СУБД еще предоставляет некоторый набор различных вспомогательных служб. Вспомогательные утилиты обычно предназначены для эффективного администрирования базы данных, что включает в себя:

- ✓ экспорт/импорт данных;

- ✓ мониторинг базы данных отслеживание характеристик функционирования и использования базы данных;
- ✓ статистический анализ, позволяющий оценить производительность или степень использования базы данных;
- ✓ реорганизация индексов;
- ✓ сборка «мусора» (неиспользуемых записей) и перераспределение памяти для физического устранения удаленных записей с запоминающих устройств, объединение освобожденного пространства и перераспределение памяти в случае необходимости.

Организация типичной системы управления базой данных и состав ее компонентов соответствует рассмотренному выше набору функций.

В современной системе управления базами данных можно выделить **ядро СУБД, процессор языка базы данных, подсистему поддержки времени выполнения, а также сервисные программы (внешние утилиты), обеспечивающие дополнительные возможности по обслуживанию БД.** В некоторых системах эти части выделяются явно, в других нет, но логически такое разделение можно провести во всех системах управления базами данных.

Ядро СУБД отвечает за управление данными во внешней памяти, управление буферами оперативной памяти, управление транзакциями и журнализацию. Иными словами, ядро СУБД это набор программных модулей, необходимый и достаточный для создания и поддержания БД универсальная часть, решающая стандартные задачи по информационному обслуживанию пользователей.

Сервисные программы предоставляют пользователям ряд дополнительных возможностей и услуг, зависящих от описываемой предметной области и потребностей конкретного пользователя.

Процессор языка БД компилирует операторы языка баз данных в некоторую выполняемую программу, представляемую в машинных кодах.

Наконец в отдельные утилиты БД обычно выделяют такие процедуры, как например, загрузка и выгрузка БД, сбор статистики, глобальная проверка целостности БД и т.д. Утилиты программируются с использованием интерфейса ядра СУБД, а иногда даже с проникновением внутрь ядра

Контрольные вопросы.

1. Назовите основные функции систем управления базами данных.
2. Дайте определение понятию «транзакция». Приведите примеры транзакций.
3. Поясните суть надежности хранения данных.
4. Поясните, что включает в себя защита целостности базы данных.
5. Поясните суть журнализации изменений.
6. Перечислите, что хранится в журнале изменений.
7. Поясните суть резервного копирования базы данных.
8. Поясните суть восстановления базы данных.
9. Назовите стандартный язык наиболее распространенных в настоящее время реляционных СУБД.

10. Поясните необходимость вспомогательных служб СУБД.

11. Назовите типовую организацию современной системы управления базами данных.

12. Поясните необходимость каждого компонента типовой организации СУБД.

Преподаватель

С.В. Банцевич