Транзакции и целостность баз данных

Понятие транзакции

- С точки зрения воздействия на СУБД - это неделимая последовательность операций манипулирования данными.
- Для пользователя - выполняется по принципу "*все или ничего*", т.е. либо транзакция выполняется целиком и переводит базу данных из одного *целостного состояния* в другое , либо, если по каким-либо причинам, одно из действий транзакции невыполнимо, или произошло какое-либо нарушение работы системы, база данных возвращается в исходное состояние, которое было до начала транзакции.

Транзакции в системах

однопользовательские

- это логические единицы работы, после выполнения которых база данных остается *в целостном состоянии*
- являются *единицами восстановления* данных после сбоев - восстанавливаясь, система ликвидирует следы транзакций, не успевших успешно завершиться в результате программного или аппаратного сбоя

многопользовательские

- служат для обеспечения *изолированной* работы отдельных пользователей (пользователям, одновременно работающим с одной базой данных, кажется, что они работают как бы в однопользовательской системе и не мешают друг другу)

Пример нарушения целостности базы

- Ограничение целостности БД состоит в том, что поле Dept_Kol не может заполняться произвольными значениями, т. к. должно содержать количество сотрудников, реально числящихся в подразделении.
- => вставка нового сотрудника в таблицу *не может быть выполнена одной операцией,* т.к. необходимо одновременно увеличить значение поля Dept_Kol:
  - Шаг 1. Вставить сотрудника в таблицу PERSON: INSERT INTO PERSON (6, Муфтахов, 1)
  - Шаг 2. Увеличить значение поля Dept_Kol: UPDATE DEPART SET Dept=Dept+1 WHERE Dept_Id=1

Таблица 1 DEPART

Dept_Id	Dept_Name	Dept_Kol
1	Кафедра алгебры	3
2	Кафедра программирования	2

Таблица 2 PERSON

Pers_Id	Pers_Name	Dept_Id
1	Иванов	1
2	Петров	2
3	Сидоров	1
4	Пушников	2
5	Шарипов	1

Понятие транзакции

- ***Транзакция*** - это последовательность операторов манипулирования данными, выполняющаяся *как единое целое* (все или ничего) и переводящая базу данных *из одного целостного состояния в другое целостное состояние*.

Свойства транзакции (АСИД)

- ***(А) Атомарность***. Транзакция выполняется как атомарная операция - либо выполняется вся транзакция целиком, либо она целиком не выполняется.
- ***(С) Согласованность***. Транзакция переводит базу данных из одного согласованного (целостного) состояния в другое согласованное (целостное) состояние. Внутри транзакции согласованность базы данных может нарушаться.
- ***(И) Изоляция***. Транзакции разных пользователей не должны мешать друг другу (например, как если бы они выполнялись строго по очереди).
- ***(Д) Долговечность***. Если транзакция выполнена, то результаты ее работы должны сохраниться в базе данных, даже если в следующий момент произойдет сбой системы.

Начало транзакции

Транзакция обычно начинается автоматически с момента присоединения пользователя к СУБД и продолжается до тех пор, пока не произойдет одно из следующих событий:

- Подана команда COMMIT WORK (зафиксировать транзакцию).
- Подана команда ROLLBACK WORK (откатить транзакцию).
- Произошло отсоединение пользователя от СУБД.
- Произошел сбой системы.

Ограничения целостности

- ***Ограничение целостности*** **-** это некоторое утверждение, которое может быть истинным или ложным в зависимости от состояния базы данных.
- Примеры ограничений целостности:

Пример 1. Возраст сотрудника не может быть меньше 18 и больше 65 лет.

Пример 2. Каждый сотрудник имеет уникальный табельный номер.

Пример 3. Сотрудник обязан числиться в одном отделе.

Пример 4. Сумма накладной обязана равняться сумме произведений цен товаров на количество товаров для всех товаров, входящих в накладную.

- Любое ограничение целостности является *семантическим* понятием, т.е. появляется как следствие определенных свойств объектов предметной области и/или их взаимосвязей.

Согласованность БД

- База данных находится в ***согласованном (целостном) состоянии***, если выполнены (удовлетворены) все ограничения целостности, определенные для базы данных.
- ***Реакция системы на попытку нарушения целостности***: система должна не только проверять, не нарушаются ли ограничения в ходе выполнения различных операций, но и должным образом реагировать, если операция приводит к нарушению целостности.
- Имеется два типа реакции на попытку нарушения целостности:
  - *Отказ* выполнить "незаконную" операцию
  - Выполнение *компенсирующих* действий

Работа системы по проверке ограничений

Классификация ограничений целостности по способам реализации

- **Декларативная** поддержка ограничений целостности (заключается в определении ограничений средствами языка определения данных (DDL - Data Definition Language). Обычно средства декларативной поддержки целостности (если они имеются в СУБД) определяют ограничения на значения доменов и атрибутов, целостность сущностей (потенциальные ключи отношений) и ссылочную целостность (целостность внешних ключей).)
- **Процедурная** поддержка ограничений целостности заключается в использовании триггеров и хранимых процедур.

Классификация ограничений целостности по времени проверки

- Немедленно проверяемые ограничения
  - ***Немедленно проверяемые ограничения*** проверяются непосредственно в момент выполнения операции, могущей нарушить ограничение (Если ограничение нарушается, то такая операция отвергается. Транзакция, внутри которой произошло нарушение немедленно проверяемого утверждения целостности, обычно откатывается)
- Ограничения с отложенной проверкой
  - ***Ограничения с отложенной проверкой*** проверяется в момент фиксации транзакции оператором COMMIT WORK. Внутри транзакции ограничение может не выполняться. Если в момент фиксации транзакции обнаруживается нарушение ограничения с отложенной проверкой, то транзакция откатывается.

Классификация ограничений целостности по области действия

1. Ограничения домена
2. Ограничения атрибута
3. Ограничения кортежа
4. Ограничения отношения
5. Ограничения базы данных

1. Ограничения целостности домена

- Представляют собой ограничения, накладываемые только на допустимые значения домена. Фактически, ограничения домена обязаны являться частью определения домена
- Например, ограничением домена "Возраст сотрудника" может быть условие "Возраст сотрудника не менее 18 и не более 65".
- Проверка ограничения. Ограничения домена сами по себе не проверяются. Если на каком-либо домене основан атрибут, то ограничение соответствующего домена становится ограничением этого атрибута.

2. Ограничение целостности атрибута

- Представляют собой ограничения, накладываемые на допустимые значения атрибута вследствие того, что атрибут основан на каком-либо домене. Ограничение атрибута в точности совпадают с ограничениями соответствующего домена. Отличие ограничений атрибута от ограничений домена в том, что ограничения атрибута *проверяются*.
- Если логика предметной области такова, что на значения атрибута необходимо наложить дополнительные ограничения, помимо ограничений домена, то такие ограничения переходят в следующую категорию.
- Проверка ограничения. Ограничение атрибута является *немедленно проверяемым* ограничением. Действительно, ограничение атрибута не зависит ни от каких других объектов базы данных, кроме домена, на котором основан атрибут. Поэтому никакие изменения в других объектах не могут повлиять на истинность ограничения.

2. Ограничения целостности атрибута

Пример 2.1. Атрибут "Возраст сотрудника" в таблице "Спецподразделение", может иметь дополнительное ограничение "Возраст сотрудника не менее 25 и не более 45", помимо того, что этот атрибут уже имеет ограничение, определяемое доменом - "Возраст сотрудника не менее 18 и не более 65".

- Приведенное ограничение кортежа является дополнительным ограничением на значения *одного* атрибута.
- В этом случае допустимы два решения:
  - Можно объявить новый домен "Возраст сотрудника спецподразделения" и тогда ограничение кортежа становится ограничением домена и атрибута
  - Можно рассматривать это ограничение именно как ограничение кортежа.
- Оба решения имеют свои положительные и отрицательные стороны.

3. Ограничения целостности кортежа

- Представляют собой ограничения, накладываемые на допустимые значения *отдельного* кортежа отношения, и *не являющиеся* ограничением целостности атрибута. Требование, что ограничение относится к *отдельному* кортежу отношения, означает, что для его проверки *не требуется* никакой информации о других кортежах отношения
- Проверка ограничения. К моменту проверки ограничения кортежа должны быть проверены ограничения целостности атрибутов, входящих в этот кортеж.
- Ограничение кортежа является *немедленно проверяемым* ограничением (ограничение кортежа не зависит ни от каких других объектов базы данных, кроме атрибутов, входящих в состав кортежа, поэтому никакие изменения в других объектах не могут повлиять на истинность ограничения).

3. Ограничения целостности кортежа

Пример 3.1. Для отношения "Сотрудники" можно сформулировать следующее ограничение: если атрибут "Должность" принимает значение "Директор", то атрибут "Зарплата" содержит значение не менее 1000$.

- Это ограничение связывает два атрибута одного кортежа.

Пример 3.2. В накладной можно установить следующую взаимосвязь атрибутов - "Цена*Количество=Сумма", связывающую атрибуты "Цена", "Количество", "Сумма".

- Сумма является явно избыточным атрибутом, значение которого просто выводятся из значений других атрибутов.
- Нужно ли хранить в отношении только базовые атрибуты, или желательно хранить все атрибуты, пересчитывая значения вычислимых атрибутов каждый раз при изменении базовых?

Пример 3.2 – решение 1

В отношении решено хранить *только базовые атрибуты*.

- *Достоинства решения:*
  - Структура отношения полностью неизбыточна
  - Не требуется дополнительного программного кода для поддержания целостности кортежа
  - Экономится дисковое пространство
  - Уменьшается трафик сети
- *Недостатки решения:*
  - Имеется риск в разных местах вычислять одни и те же данные по *разным формулам*
  - При изменении логики вычислений необходимо изменить одни и те же фрагменты кода *во всех местах*, где они встречаются. Это сильно затрудняет модификацию приложений
  - Если возникает нерегламентированный запрос, то человек, формулирующий запрос должен *помнить* все эти формулы

Пример 3.2 – решение 2

В отношении решено хранить *все атрибуты*, в том числе и вычислимые.

- *Достоинства решения:*
  - Код, поддерживающий целостность кортежа (и содержащий формулы для вычислимых атрибутов), хранится в одном месте, например в триггере, связанном с данным отношением
  - При изменении логики вычислений, изменения в формулы требуется внести только в одном месте (в триггере)
  - Запросы к базе данных содержат меньше формул и поэтому более просты
  - Легче формулировать нерегламентированные запросы, т.к. в запросе используются атрибуты, имеющие для бухгалтера конкретный смысл
- *Недостатки решения:*
  - При изменении логики расчета надобность в некоторых атрибутах может исчезнуть, зато может появиться потребность в новых атрибутах => потребуется *перестройка структуры* отношения (болезненная операция для работающей системы)
  - Структура отношения становится более *сложной и запутанной*
  - *Увеличивается объем* базы данных
  - *Увеличивается трафик* сети

Пример 3.2 – другие решения

- Можно хранить в базовом отношении только базовые атрибуты, а для работы бухгалтерии использовать заранее подготовленные представления (динамические отношения, задаваемые оператором SQL). Тогда логика расчетов будет храниться в одном SQL-операторе, определяющем это представление.
- Другим вариантом может быть сохранение формул в виде хранимых процедур и функций базы данных.

4. Ограничения целостности отношения

- представляют ограничения, накладываемые только на допустимые значения *отдельного* отношения, и *не являющиеся* ограничением целостности кортежа. Требование, что ограничение относится к отдельному отношению, означает, что для его проверки не требуется информации о других отношениях (в том числе не требуется ссылок *по внешнему ключу* на кортежи *этого же* отношения)
- Проверка ограничения. К моменту проверки ограничения отношения должны быть проверены ограничения целостности кортежей этого отношения.
- Ограничение отношения может быть как *немедленно проверяемым* ограничением, так и ограничением *с отложенной проверкой*.

4. Ограничения целостности отношения

- **Пример 4.1**. Ограничение целостности сущности, задаваемое потенциальным ключом отношения, является ограничением отношения, т.к. для его проверки необходимо иметь информацию обо всех кортежах отношения (более точно, обо всех занятых в данный момент значениях потенциального ключа).
- **Пример 4.2**. Ограничение целостности, определяемые наличием функциональных, многозначных зависимостей и зависимостей соединения, являются ограничениями отношения.
- **Пример 4.3**. Предположим, что в отношении PERSON задано следующее ограничение - в каждом отделе должно быть не менее двух сотрудников. Это ограничение можно сформулировать так - количество строк с одинаковым значением Dept_Id должно быть не меньше 2.
- **Пример 4.4**. Ограничение целостности, определяемое требованием, что некоторая таблица должна быть не пуста, являются ограничениями отношения.

4. Ограничения целостности отношения

- Ограничение отношения может быть как *немедленно проверяемым* ограничением, так и ограничением *с отложенной проверкой*.
- Ограничение отношения, являющееся ограничением потенциального ключа (пример 4.1) является немедленно проверяемым ограничением.
- Ограничение, определенное наличием функциональной зависимости атрибутов также является немедленно проверяемым ограничением.
- Ограничения же, определенные многозначной зависимостью или зависимостью соединения являются ограничениями с отложенной проверкой. Эти ограничения требуют, чтобы кортежи вставлялись и удалялись *целыми группами,* что невозможно сделать при выполнении проверки после каждой одиночной вставки или удаления кортежа.
- В примере 4.3 невозможно вставить ни один новый кортеж для нового отдела. В новый отдел необходимо вставить сразу не менее двух сотрудников. Таким образом, это ограничение с отложенной проверкой.
- Ограничение из примера 4.4 имеет смысл проверять только при удалении кортежей из отношения. Это ограничение может быть как немедленно проверяемым, так и отложенным.

5. Ограничения целостности базы данных

- Представляют ограничения, накладываемые на значения двух или более связанных между собой отношений (в том числе отношение может быть связано само с собой)
- **Пример 5.1**. Ограничение целостности ссылок, задаваемое внешним ключом отношения, является ограничением базы данных.
- **Пример 5.2**. Ограничение на таблицы DEPART и PERSON из примера 1 является отношением базы данных, т.к. оно связывает данные, размещенные в различных таблицах.
- Проверка ограничения. К моменту проверки ограничения базы данных должны быть проверены ограничения целостности отношений.
- Ограничение базы данных может быть как *немедленно проверяемым* ограничением, так и ограничением *с отложенной проверкой*.