СРСП 12. Обработка транзакций

Ранее рассматривался пример, в котором требовалось изменить номер продукта ПР = 13 на ПР = 20 и для этого пришлось проводить последовательное изменение четырех таблиц:

```
UPDATE   Продукты          UPDATEСостав
SET      ПР = 20   SET     ПР = 20
WHERE    ПР = 13;  WHERE   ПР = 13;
UPDATE   Поставки          UPDATEНаличие
SET      ПР = 20   SET     ПР = 20
WHERE    ПР = 13;  WHERE   ПР = 13;
```

Этот пример приведен здесь для иллюстрации того, что единственная, с точки зрения пользователя, операция может потребовать нескольких операций над базой данных. Более того, между этими операциями может даже нарушаться непротиворечивость базы данных. Например, в ней могут временно содержаться некоторые записи поставок, для которых не имеется соответствующих записей поставляемых продуктов. Положение не спасает и перестановка последовательности обновляемых таблиц. Противоречивость исчезнет только после выполнения всех обновлений, т.е. выполнения логической единицы работы - полной замены номера продукта в базе данных (независимо от количества таблиц, в которых встречается номер продукта).

Теперь можно дать определение транзакции. Транзакция или логическая единица работы, - это в общем случае последовательность ряда таких операций, которые преобразуют некоторое непротиворечивое состояние базы данных в другое непротиворечивое состояние, но не гарантируют сохранения непротиворечивости во все промежуточные моменты времени.

Никто кроме пользователя, генерирующего ту или иную последовательность SQL-предложений, не может знать о том, когда может возникнуть противоречивое состояние базы данных и после выполнения каких SQL-предложений оно исчезнет, т.е. база данных вновь станет актуальной. Поэтому в большинстве СУБД создается механизм обработки транзакций, при инициировании которого все изменения данных будут рассматриваются как предварительные до тех пор, пока пользователь (реже система) не выдаст предложения:

COMMIT (фиксировать), превращающее все предварительные обновления в окончательные ("зафиксированные");

ROLLBACK (откат), аннулирующее все предварительные обновления.

Таким образом, транзакцией можно назвать последовательность SQL-предложений, расположенных между "точками синхронизации", учреждаемых в начале выполнения программы и издании COMMIT или ROLLBACK и только в этих случаях. При этом следует иметь в виду, что возможен неявный COMMIT (существует режим AUTOCOMMIT, в котором система издает COMMIT после выполнения каждого SQL-предложения) и ROLLBACK (выполняемый при аварийном завершении программы).

Ясно теперь, что пользователь должен сам решать, включать ли механизм обработки транзакций и если включать, то где издавать COMMIT (ROLLEBACK), т.е. какие последовательности SQL-предложений являются транзакциями.

Теперь о проблемах, связанных с параллельным использованием базы данных множеством разнообразных пользователей.

Большинство СУБД позволяют любому числу транзакций одновременно осуществлять доступ к одной и той же базе данных и в них существуют те или иные механизмы управления параллельными процессами, предотвращающие нежелательные воздействия одних транзакций на другие. По сути это механизм блокирования, главная идея которого достаточно проста. Если транзакции нужны гарантии, что некоторый объект (база данных, таблица, строка или поле), в котором она заинтересована, не будет изменен каким-либо непредсказуемым образом в течение требуемого промежутка времени, она устанавливает блокировку этого объекта. Результат блокировки заключается в том, чтобы изолировать этот объект от других транзакций и, в частности, предотвратить его изменение средствами этих транзакций. Для первой транзакции, таким образом, имеется возможность выполнять предусмотренную в ней обработку, располагая

определенными знаниями о том, что объект в запросе будет оставаться в стабильном состоянии до тех пор, пока данная транзакция этого пожелает.

Задание к СРСП 12:

1. Определить, какие транзакции могут быть созданы по индивидуальному варианту задания.
2. Написать скрипт для создания транзакции в соответствии с индивидуальным вариантом.