**Понятие транзакции. Свойства транзакций. Работа транзакций. Проблемы при работе транзакций.**

**Транзакция** – последовательность операторов манипулирования данными, выполняющаяся как единое целое и переводящая БД из одного целостного состояния в другое. Это логическая единица работы, единицы восстановления данных после сбоя.

4 свойства транзакций:

* **атомарность** – транзакция выполняется как атомарная, либо выполняется целиком, либо не выполняется
* **согласованность** – транзакция переводит БД из одного согласованного состояния в другое
* **изоляция** - транзакции разных пользователей не должны мешать друг другу
* **долговечность** - если транзакция выполнена, то результат ее работы должен сохраниться в БД, даже если в следующий момент произойдет сбой системы

Продолжается транзакция до наступления одного из 4 событий.

* Commit– завершить транзакцию
* Rollback – откатить
* Произошло отсоединение пользователя от СУБД
* Сбой системы

При последовательном запуске системы происходит анализ транзакций, которые можно восстановить.

**Проблемы при параллельной работе транзакции.**

1. **Проблема потери результатов обновления**. Две транзакции по очереди делают запись в одну и туже строку и фиксируют изменения. После окончания транзакции P имеет значение Р2. Транзакция А ничего не знает о транзакции В и, следовательно, транзакция А потеряла результаты своей работы.
2. **Проблема незафиксированной зависимости** (чтения грязных данных). Транзакция В изменяет данные в строке, затем транзакция А читает измененные данные и работает с ними. Транзакция В откатывается и восстанавливает измененные данные.
3. **Проблема несовместимого анализа**.
   1. **Неповторяемое считывание**. Транзакция А дважды читает одну и туже строку, между считываниями вклинивается транзакция В, изменяющая значения в строке. Транзакция А работает с данными, которые с точки зрения транзакций самопроизвольно изменяются
   2. **Фантомы**. Транзакция А выполняет выборку строк, удовлетворяющих некоторому условию
   3. **Несовместный анализ**. Длинная транзакция выполняет анализ по таблице, подсчитывает сумму денег на счетах клиентов банка. Пусть на всех счетах сумма 100 денежных единиц. Короткая транзакция выполняет перевод 50 денежных единиц со счета на счет так, чтобы сумма на счетах не уменьшилась

Транзакции называются **конкурирующими** если они пересекаются во времени и обращаются к одним и тем же данным. В результате конкуренции транзакции возникают конфликты:

1. WRITE - WRITE. Первая транзакция изменила объект и не закончилась. Вторая транзакция пытается изменить этот объект. В результате происходит конфликт обновления.
2. READ - WRITE. Первая транзакция прочитала объект и не закончилась. Вторая транзакция пытается изменить объект.
3. WRITE-READ. Первая транзакция изменила объект и не закончилась. Вторая транзакция пытается прочитать объект. В результате чтение грязных данных

Имеется 2 способа разрешить конкуренцию между поступающими в произвольные моменты времени транзакциями:

1. Обеспечить чтобы конкурирующие транзакции выполнялись в разное время. Реализуется с помощью временных меток.
2. Обеспечить чтобы конкурирующие транзакции работали с разными версиями данных. Реализуется с помощью использования журнала транзакции