**Транзакция** − это механизм базы данных, позволяющий таким образом объединять несколько операторов, изменяющих базу данных, чтобы при выполнении этой совокупности операторов они или все выполнились или все не выполнились.

Основные свойства транзакции: *атомарность* (операторы изменения БД, включенные в транзакцию, либо выполнятся все, либо не выполнится ни один); *согласованность* (транзакция должна фиксировать новое согласованное состояние БД); *изолированность* (транзакции физически обрабатываются последовательно, изолированно друг от друга, результирующий набор 2 транзакций не влияет друг на друга ); *долговечность* (первая транзакция обновило состояние бд, чтобы отменить, изменить еще раз эту транзакцию, нам понадобится новая транзакция.).

Режим *неявной транзакции* может быть включен для текущего соединения с сервером БД с помощью специальной инструкции:

**SET IMPLICIT\_TRANSACTIONS ON**

Обратное переключение осуществляется с использованием ключевого слова OFF вместо ON.

Неявная транзакция начинается, если выполняется один из следующих операторов:

CREATE, DROP; ALTER TABLE; INSERT, DELETE, UPDATE, SELECT, TRUNCATE TABLE; OPEN, FETCH; GRANT (выдача разрешений), REVOKE (запрещение разрешений).

Неявная транзакция продолжается до тех пор, пока не будет выполнен оператор фиксации (COMMIT) или оператор отката (ROLLBACK) транзакции.

Переключение в режим *явной транзакции* осуществляется с помощью оператора BEGIN TRANSACTION. Транзакцию должен завершать один из операторов: COMMIT TRAN или ROLLBACK TRAN.

Системная функция @@TRANCOUNT возвращает уровень вложенности транзакции. (если значение больше нуля, то транзакция не завершена).

Встроенная функция PATINDEX определяет в строке позицию первого символа подстроки, заданную шаблоном. С помощью этой функции в тексте сообщения об ошибке отыскивается имя ограничения целостности.

Оператор SAVE TRANSACTION, формирует *контрольную точку* транзакции.

**Проблемы при параллельных транзакциях**:

Неподтвержденное чтение – это такое чтение, при котором мы имеем две транзакции, первая транзакция началась, но мы ее не зафиксировали, тогда вторая транзакция сможет считывать данные из первой транзакции.

Неповторяющееся чтение – это такое чтение, при котором одна транзакция имеет определенные данные, эти данные считываются и изменяются во второй транзакции, и третья транзакция уже считывает эти измененные данные.

Фантомное чтение – тоже самое, что и в неповтор. чтении, только еще идет добавление несуществующих строк.

Явная транзакция - обычно это группа инструкций языка Transact-SQL, начало и конец которой обозначаются такими инструкциями, как BEGIN TRANSACTION, COMMIT и ROLLBACK.

Неявная транзакция - задает любую отдельную инструкцию INSERT, UPDATE или DELETE как единицу транзакции.

Существует четыре уровня изоляции транзакций — Read uncommited (Чтение незафиксированных данных), Read committed (Чтение зафиксированных данных), Repeatable read (Повторяемое чтение) и Serializable (Сериализуемость).

Read uncommitted(**Все разрешает**)  
Уровень, имеющий самую плохую согласованность данных, но самую высокую скорость выполнения транзакций. Название уровня говорит само за себя — каждая транзакция видит незафиксированные изменения другой транзакции (феномен **грязного чтения**)

Read committed(Запрещает Неподтвержденное чтение)  
Для этого уровня параллельно исполняющиеся транзакции видят только зафиксированные изменения из других транзакций. Таким образом, данный уровень обеспечивает защиту от **грязного чтения**.

Repeatable read(**Запрещает Неподтвержденное чтение и Неповторяющееся чтение**)  
Уровень, позволяющий предотвратить феномен **неповторяющегося чтения**. Т.е. мы не видим в исполняющейся транзакции измененные и удаленные записи другой транзакцией. Но все еще видим вставленные записи из другой транзакции. **Чтение фантомов** никуда не уходит.

Serializable(**Запрещает все**)  
Уровень, при котором транзакции ведут себя как будто ничего более не существует, никакого влияния друг на друга нет. В классическом представлении этот уровень избавляет от эффекта **чтения фантомов**.

Транзакция, выполняющаяся в рамках другой транзакции, называется *вложенной*.

При работе с вложенными транзакциями нужно учитывать следующее:

− оператор COMMIT вложенной транзакции действует только на внутренние операции вложенной транзакции;

− оператор ROLLBACK внешней транзакции отменяет зафиксированные операции внутренней транзакции;

− оператор ROLLBACK вложенной транзакции действует на операции внешней и внутренней транзакции, а также завершает обе транзакции;