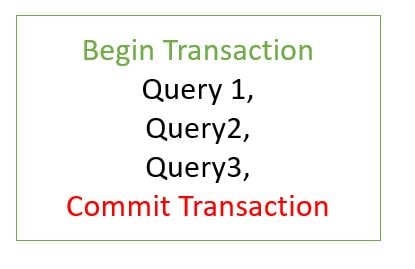
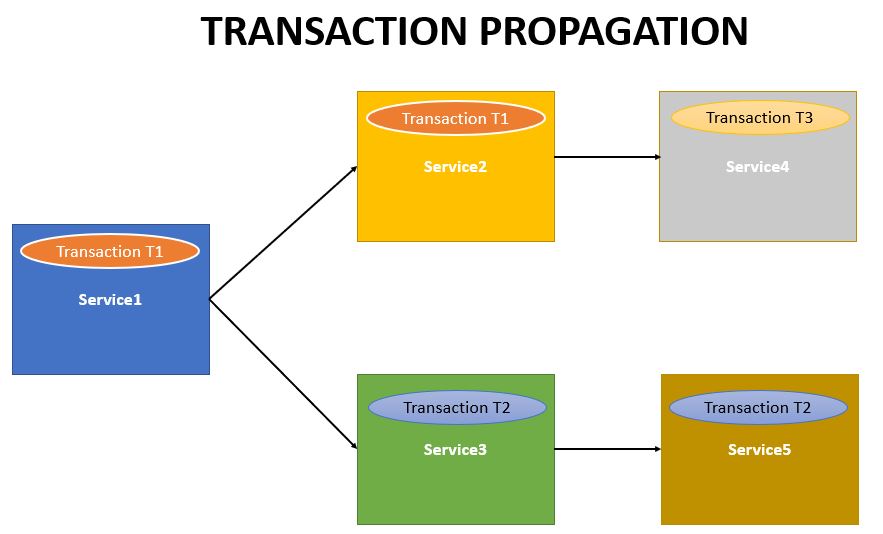
What are Database Transactions?  
A Database transaction is a **single logical unit of work which accesses and possibly modifies the contents of a database.**  
  


What is Transaction Propagation?  
Any application involves a number of services or components making a call to other services or components. Transaction Propagation indicates if any component or service will or will not participate in transaction and how will it **behave if the calling component/service already has or does not have a transaction created already.**

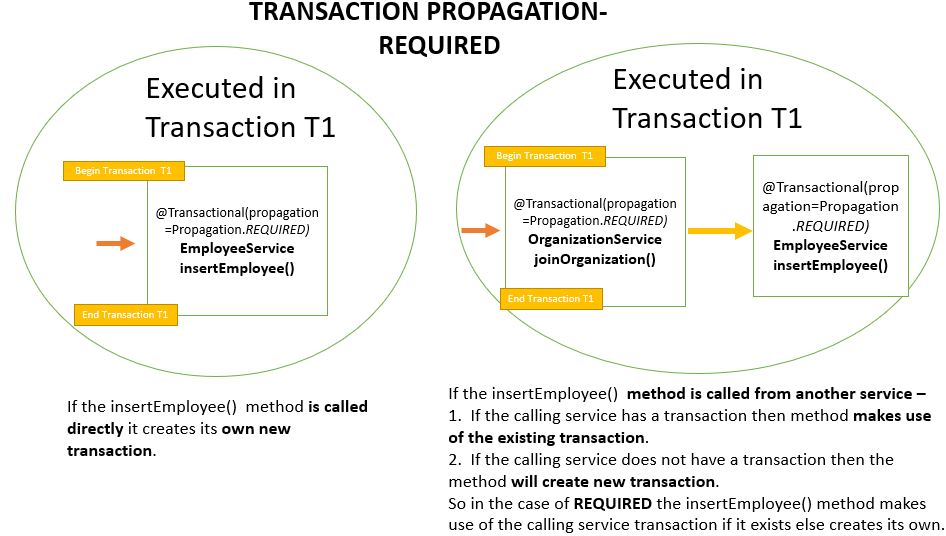


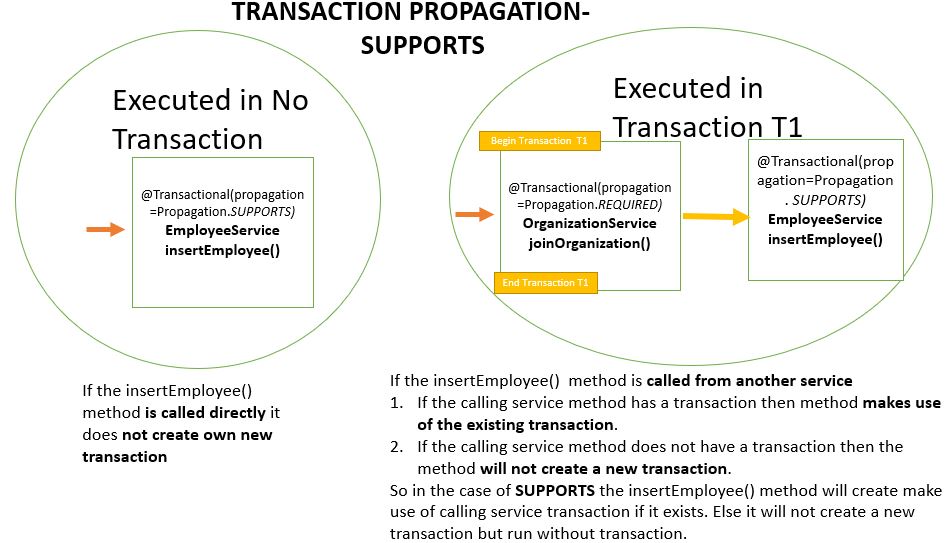
.

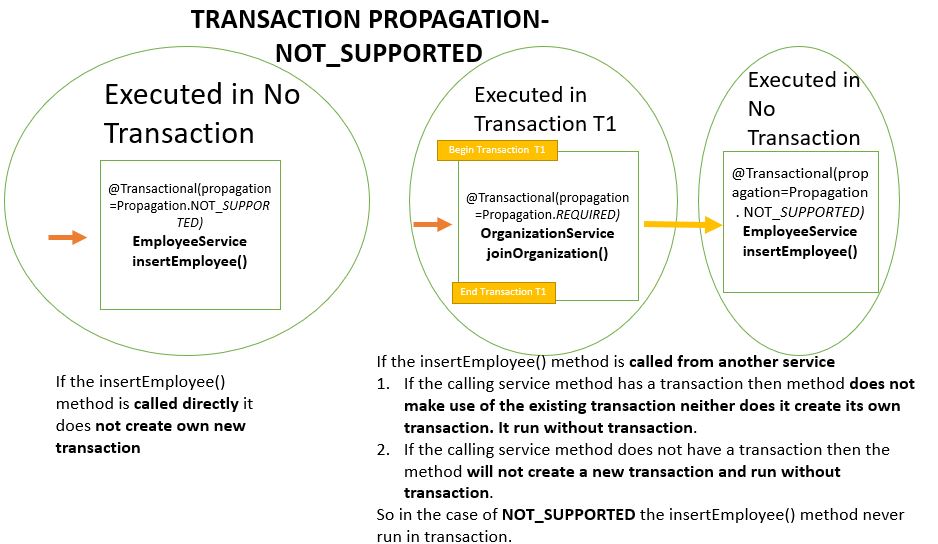
There are six types of Transaction Propagations-

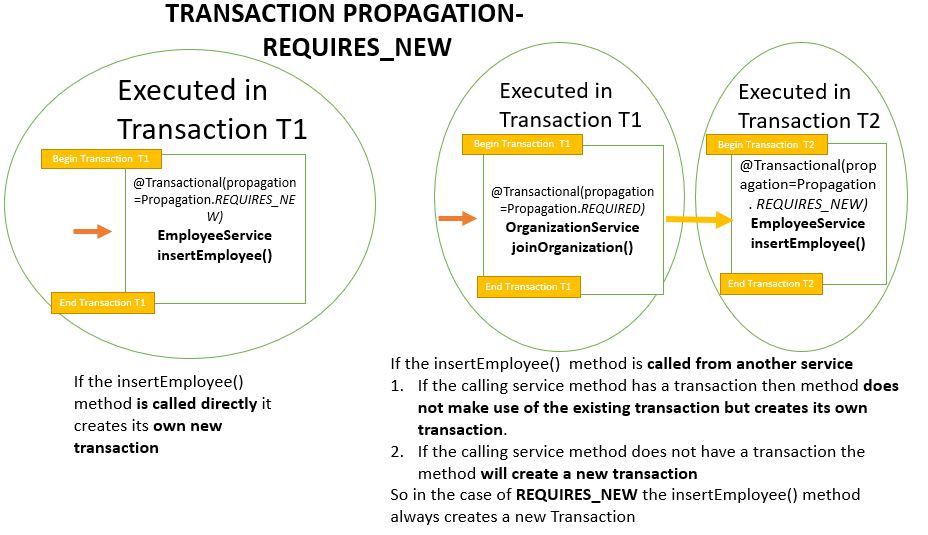
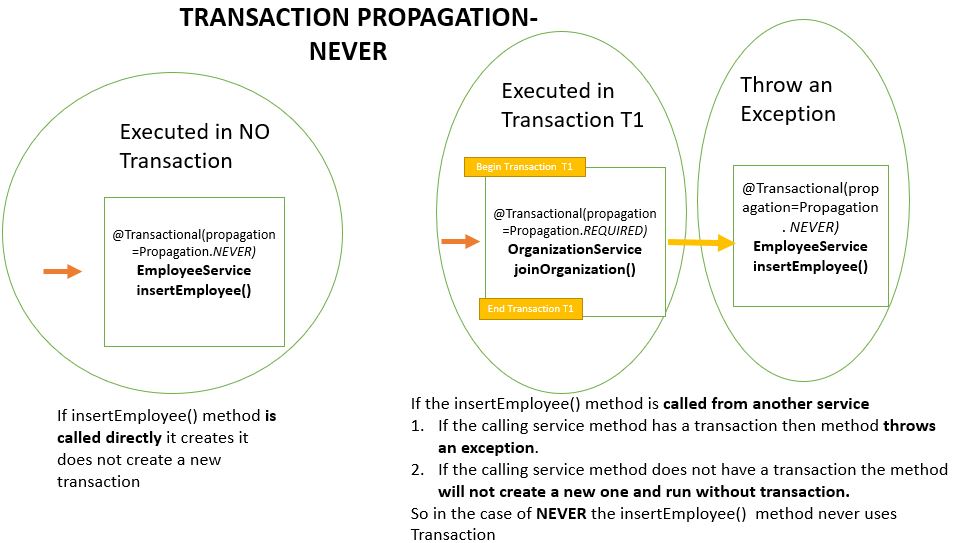
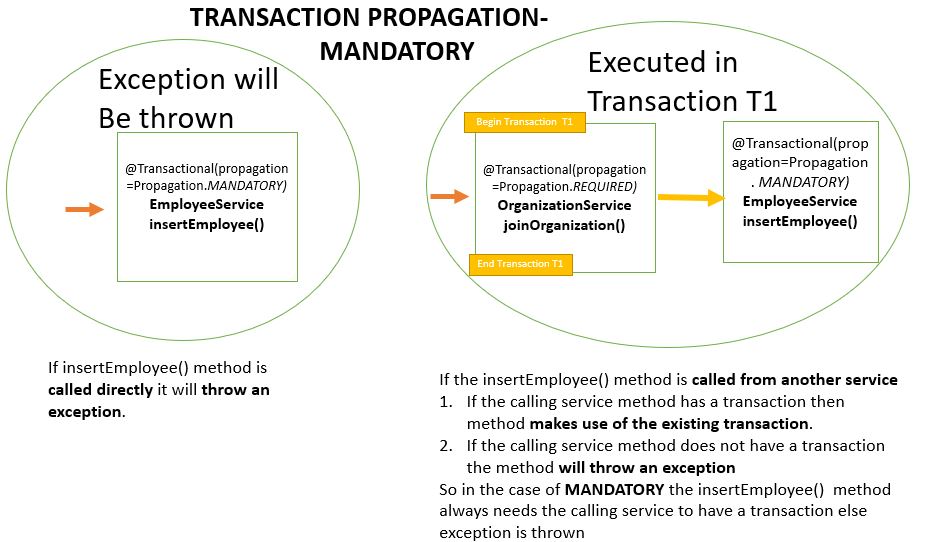
* **REQUIRED**
* **SUPPORTS**
* **NOT\_SUPPORTED**
* **REQUIRES\_NEW**
* **NEVER**
* **MANDATORY**

Transaction Propagation - **REQUIRED** (Default Transaction Propagation)







|  |  |
| --- | --- |
| **Propagation** | **Behavior** |
| **REQUIRED** | **Always executes in a transaction.** If there is any existing transaction it uses it. If none exists then only a new one is created |
| **SUPPORTS** | **It may or may not run in a transaction.** If current transaction exists then it is supported. If none exists then gets executed without transaction. |
| **NOT\_SUPPORTED** | **Always executes without a transaction.** If there is any existing transaction it gets suspended |
| **REQUIRES\_NEW** | **Always executes in a new transaction.** If there is any existing transaction it gets suspended |
| **NEVER** | **Always executes without any transaction.** It throws an exception if there is an existing transaction |
| **MANDATORY** | **Always executes in a transaction.** If there is any existing transaction it is used. If there is no existing transaction it will throw an exception. |

Transaction is a cross cutting concern and it is implemented using AOP in Spring Boot.  


Spring Boot implicitly creates a **proxy for the transaction annotated methods.** So for such methods the proxy acts like a wrapper which takes care of creating a transaction at the beginning of the method call and committing the transaction after the method is executed.  
