WHY ATTATCH IS BETTER IN DISCONNECTED SCENARIO IN EFCORE

->Attach puts all entities in the graph into the Unchanged state.

->However, entities will be put in the Added state if they have store-generated keys (e.g. Identity column)

and no key value has been set.

->This means that when exclusively using store-generated keys, Attach can be used to start tracking a mix of new and

existing entities where the existing entities have not changed.

->The new entities will be inserted while the existing entities will not be saved other than to update any necessary FK values.

| Attach() | Root entity with Key value | Root Entity with Empty or CLR default value | Child Entity with Key value | Child Entity with empty or CLR default value |
| --- | --- | --- | --- | --- |
| context.Attach(entityGraph).State = EntityState.Added | Added | Added | Unchanged | Added |
| context.Attach(entityGraph).State = EntityState.Modified | Modified | Exception | Unchanged | Added |
| context.Attach(entityGraph).State = EntityState.Deleted | Deleted | Exception | Unchanged | Added |

If we use update then

->entities are put in the Modified state instead of the Unchanged state.

->This means that when exclusively using store-generated keys,

Update can be used to start tracking a mix of new and existing entities where the existing entities may have some modifications.

->The new entities wil

| Update() | Root entity with Key value | Root Entity with Empty or CLR default value | Child Entities with Key value | Child Entities with Empty Key value |
| --- | --- | --- | --- | --- |
| DbContext.Update(entityGraph) or DbSet.Update(entityGraph) | Modified | Added | Modified | Added |