A volte può succedere che non ho le righe di alcune, questa cosa si risolvere con le transazioni: (Capire cos’è)  
  
In java c’è un acronimo molto usato “JPA” è un concetto che sta attorno e permette di definire la persistenza dei dati.  
  
Per quanto riguarda JPA nell’AS abbiamo due tecnologie:  
  
JPA transazionali:  
La transazione si può implementare con la scrittura di codice, il BPM sarebbe l’apertura della transazione svolta dal programmatore ed il container di una transazione in questo caso sarà l’AS (non direttamente ma con plugin)  
  
JPA non transazionali:

‘’’’’’’’’’’’’’’’’’’

Una connessione si può definire dentro wildfly, in quello che è JNDI.  
  
Apro wildfly / su App.html#home.  
  
Dobbiamo configurare datasources, il XA sono quelle tipologie di datasource con una lieve differenza, sono ds che obbligano il programmatore a gestire la transazione di proprio conto.  
  
con la XA una volta configurato il datosource siamo noi obbligati nel capire quando chiudere o aprire la transazione.  
  
con la non XA siamo più liberi, li dobbiamo configurare entrambi.  
  
Vado su Non XA e poi ADD > Selezionare un datasource > MySQL Datasource > lo rinominiamo come vogliamo.  
  
Adesso mi va a chiedere il JNDI name, lo possiamo scegliere ma ce lo dobbiamo ricordare però.  
  
Andiamo su next > il driver: wildfly necessita dei driver, faccio

È meglio creare l’ear quando creo EJB project,

metodi che useremo più spesso sono:  
persist e merge.

Se voglio salvare un metodo uso entityManager.persist(model);

Entity Transaction transaction = entityManager.getTransaction();  
transaction.begin();

entityManager.persist(model);

if (1-1) {

Le fetch nel OneToMany, fetch = FetchType.EAGER (o LAZY)