Improve Performance by Avoiding Null Values

In your SOQL and SOSL queries, explicitly filtering out null values in the WHERE clause allows Salesforce to improve query performance. In the following example, any records where the `Thread__c` value is null are eliminated from the search.

```
Public class TagWS {

/* getThreadTags
*
* a quick method to pull tags not in the existing list
*
*/
   public static webservice List<String>
   getThreadTags(String threadId, List<String> tags) {

      system.debug(LoggingLevel.Debug,tags);

      List<String> retVals = new List<String>();
      Set<String> tagSet = new Set<String>();
      Set<String> origTagSet = new Set<String>();
      origTagSet.addAll(tags);

// Note WHERE clause optimizes search where Thread__c is not null

      for(CSO_CaseThread_Tag__c t :
         [SELECT Name FROM CSO_CaseThread_Tag__c
         WHERE Thread__c = :threadId AND
         Thread__c != null])

      {
         tagSet.add(t.Name);
      }
      for(String x : origTagSet) {
   // return a minus version of it so the UI knows to clear it
         if(!tagSet.contains(x)) retVals.add('-' + x);
      }
      for(String x : tagSet) {
   // return a plus version so the UI knows it's new
         if(!origTagSet.contains(x)) retvals.add('+' + x);
      }

      return retVals;
   }
}
```