Question 1:
You are given the data below. You have to analyse the execution of following query on RDF triple store. “Give the title of the book which has author whose name has the keyword with ‘Gekhre’.
Now, evaluate the query on a vertically partitioned approach suggested in the paper, compare the two queries and state which strategy is better and why?

RDF triple store

Vertically partitioned

Ans1:

**RDF triple row oriented**
Select t5.obj
From table 1 as t1, table 1 as t2, table 1 as t3, table 1 as t4, table 1 as t5
Where
T1.prop = ‘isnamed’ and t1.obj like ‘%Gekhre%’
AND t1.subj=t2.subj AND t2.prop =’type’
AND t2.obj=’author’ AND t3.prop=’hasAuth’
AND t3.subj=t2.subj AND t4.subj=t3.obj
AND t4.prop=type AND t4.obj=book
AND t4.subj=t5.subj AND t5.prop=’title’

**Vertically partitioned**
Select t.obj
From Title t, Type ty, Type ty1, IsNamed I, IsNamed is, HasAuth h
Where i.obj=’Gekhre” AND i.subj=ty.subj
AND ty.obj=’Author’ AND is.subj=h.subj
AND h.obj=ty1.subj AND ty.obj=’book’
AND h.obj=t.subj

The query in normal RDF triple store will lead 5 way self join on the same table, which can be huge as it a single table. On the other hand the query with vertically partitioned approach has normal subject–object or subject-subject joins which can be implemented as merge joins. Thus performance of the second query is better than the first approach.
Question 2
Below is a set of 4 tables. One is a subject table and rest represent object tables. Note that these have one to one mapping as proposed in SW-store. Please highlight the most important problem in this case and why is the resolution of that problem important from the paper’s perspective. Hint: Recall the discussion during class presentation.

Further classify which column oriented sparse compression scheme (Tuple ID Range, bitmap or list of tuple ids) would you use in Title, Copyright and Author above.

Answer 2:

The problem here is of NULLs in the data and their affect on the performance. This is important as there is one to one mapping between the subject table and the object tables and we cannot eliminate the NULL values. This is important from the paper perspective because one of the key limitations of the related work (Property tables) which was highlighted was NULL values.

Classification
- **Course**: we should use tuple id Range scheme her because the data is quite dense and there is only one NULL here. We prefer range when the data is quite dense or when NULLs occur in long runs.
- **Student Count**: This table can be classified as neither dense nor sparse as the number of values and number of NULLs is almost equal. We should use Bitmap scheme here which indicates assign a 1 to a valid value and a 0 to a NULL value.
- **Instructor Rating**: This table is classified as sparse due to large number of NULL values. We would use tuple ID list in this case. This assigns a list of tuple ids which have value.