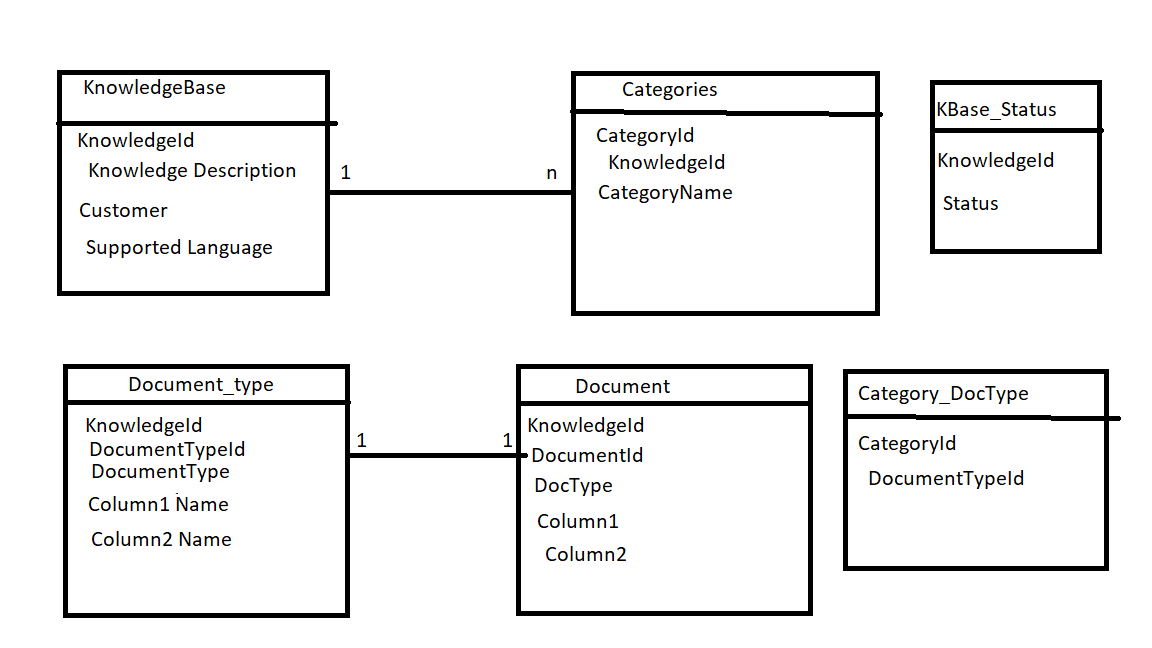
Knowledge Base

ER Diagram :



Entities:

* KnowledgeBase
* Categories
* Document\_type
* Document
* KBase\_Status

Relationship:

* Category\_DocType

Database Design:

* All the primary key(knowledgeId,documentTypeId etc.) and foreign key would be of integer type. Each primary key and foreign key column should have a trigger which would be triggered before inserting a row. It would generate the next Unique Identifier.
* Trigger on KnowledgeBase so that after insert of row, it would insert a row into KBase\_status with status “CREATED”.
* Partitions : All tables would be partitioned on knowledgeId,categoryId.
* Indexes: Document table should be indexed on Column1 and Column2.

API Endpoints:

**Categories:**

**GET:**

/categories/{knowledgeId} - get all the categories for a knowledgeId

**POST :**

Post request:

{

“knowledgeId” : ,

“categoryId” : ,

“categoryName” : “”

}

/categories – add a category for a knowledgeId

/categories/categoryId – update a category name for a categoryId.

**KnowledgeBase:**

**GET:**

/knowledgeBase/{knowledgeId} - get all the categories for a knowledgeId

**POST :**

Post request:

{

“knowledgeId” :

“KnowledgeDescription” :

“customer” :

“supportedLang” :

}

/knowledgeBase– add a category for a knowledgeId

/ knowledgeBase/knowledgeId – update a category name for a categoryId.

**Permissions and Privileges for customer and author:**Custom annotations can be created in Jersey framework. These can be used on the entire resource class or the individual api.