# Data Warehousing & Mining using MongoDB

Final year Project by Koustuv Sinha & Hitesh Agarwal, Institute of Engineering & Management, under the guidance of Prof. Ee-Kian Wong

## Assignment 1: Library Management System

Our first assignment enables us to learn more about the NoSQL system of MongoDB, which will enable us to proceed further in our Project.

This is a simple Library Management System, having the following entities:

Book, Author, Publisher – core classes

*Book*

id integer

name String

pub Publisher

isbn String

authors ArrayList<Integer>

category String

price Double

status String

*Author*

id integer

name String

address String

email String

phone String

books ArrayList<Integer>

## **Problems encountered**

**Many to Many Relationship Management:** Book and Author class has many to many relationship. A Book class points to multiple Authors, as well as an Author class points to multiple Books. The problem lies that, what is the optimal way of saving both the classes, so that this relationship is maintained.

**Answer:** We have made an integer array in both class, containing the id’s of the other classes. Book has an Integer array containing Author Id’s and vice versa.

**Transaction ACID Property handling:** MongoDB doesn’t support transaction controls like “Rollback” or “Commit” as in Relational Database. So, to ensure atomicity and concurrency, we have to devise a unique way to handle database transactions.

**Answer:** We have created commit & rollback equivalent in MongoDB as follows:

Two tables : backup\_insert and backup\_update.

Whenever any data is inserted in the database, it is also inserted in the backup\_insert table. When “rollback” is called, the data in the backup\_insert is extracted and deleted from the main table, else when “commit” is called, the data stored in backup\_insert or backup\_update is deleted.

Similar operations for backup\_insert table.

**Encapsulation**: library\_manager class handles all the operations with the MongoDB database, thus encapsulates the details from the library\_clerk class, who calls the methods available. The library\_manager class supports Generics, so can be invoked with reference to core classes.