Managing & Programming Database MCDA5540

Master of Science in Computing and Data Analytics Project Report

Submitted by:

Vivekanand Boopathy	A00425792
Sreeraj Punnoli	A00429404
Sri Akhil Reddy Kovvuri	A00428260
Narasimha Rao Durgam	A00429601

Submitted to:

Trishla Shah

SAINT MARY'S UNIVERSITY SINCE 1802

One University. One World. Yours.

Introduction

The Halifax Science Library (HSL) maintains an SQL database containing tables about various publications. HSL now has the following immediate plans:

a. To sell library items and record information about these sales (transactions) in their database.
b. To record, for each scientific journal (magazine), data about all articles in that magazine.
c. To record data about monthly expenses.

This document describes the relational database schema structure to implement the same.

Tools Required:

- MySQL Workbench
- MySQL Server
- PHP
- PHP My Admin
- Mongo DB
- Sublime Text

Operating System:

- Windows 10
- Linux

Enhanced Entity Relational (EER) Diagram

The first step to create the database schema is to create an Enhanced Entity Relationship (EER) diagram which describes the attributes of each entity and relationship with other entities. Below screenshot represent the EER of HSL data.

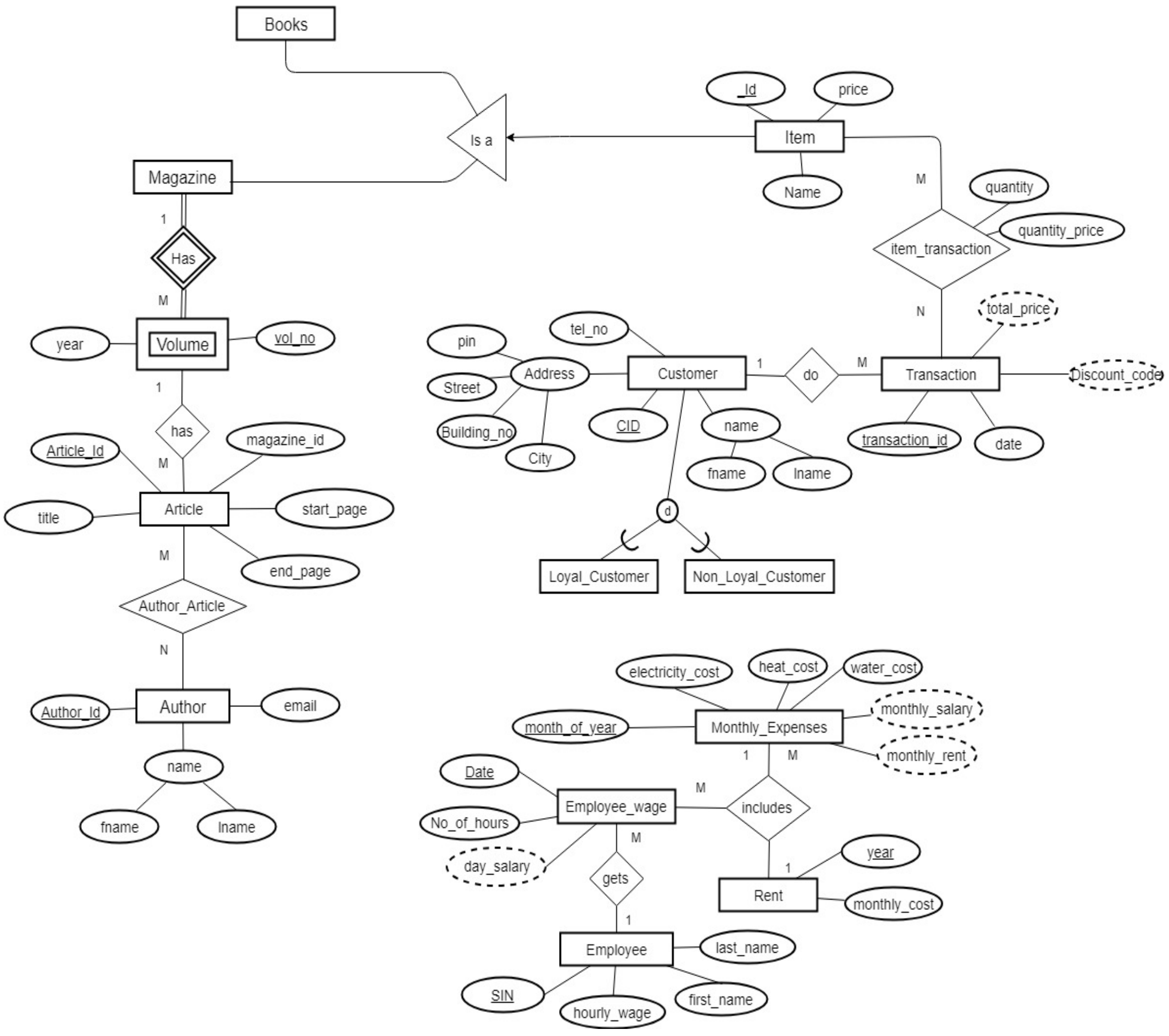


Fig 1: Enhanced Entity Relationship Diagram

Relational Database Schema

Based on the created EER diagram, designed relational database schema to create tables and its attributes.

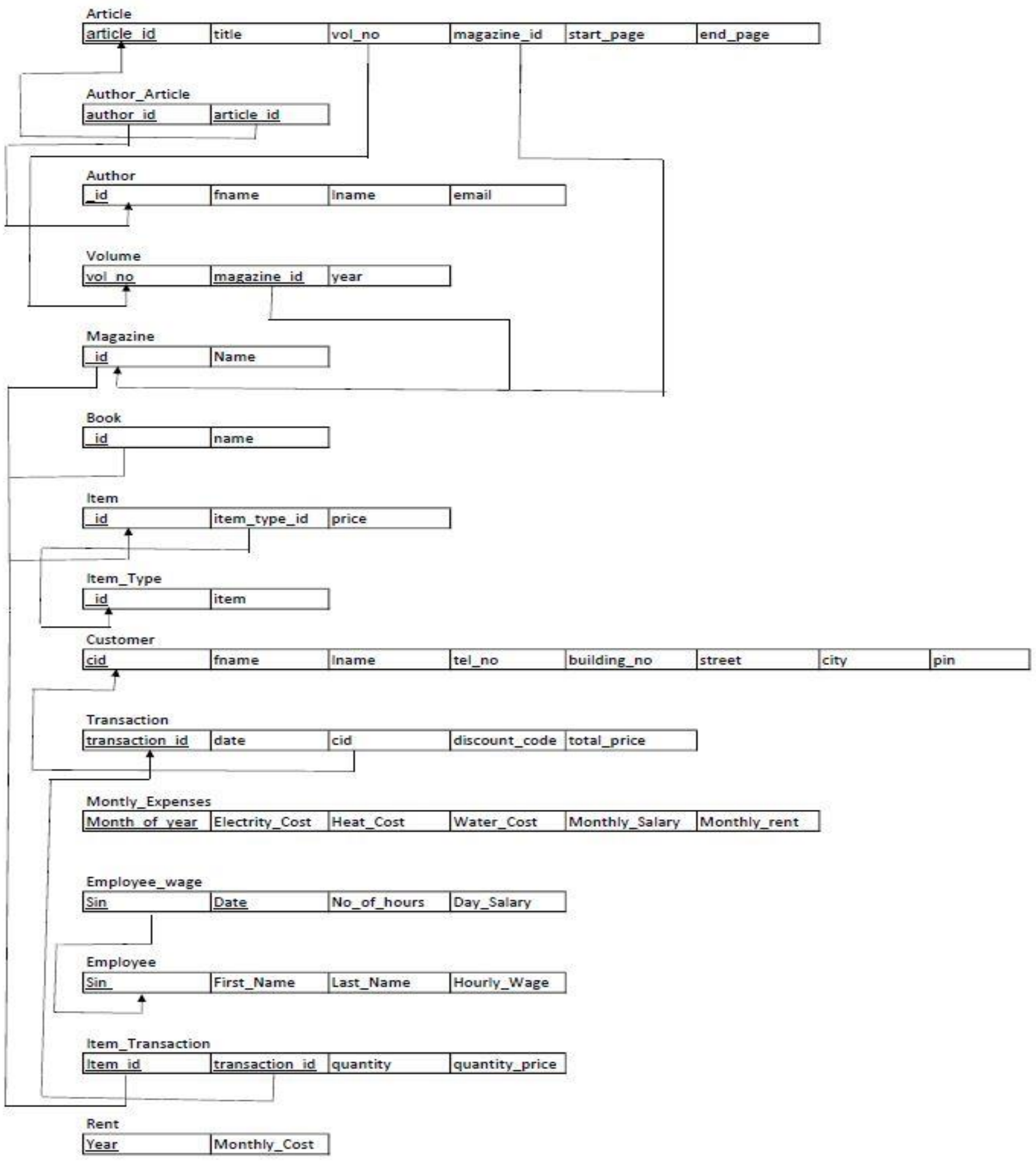

Article
article_id | title | vol_no | magazine_id | start_page | end_page
Author_Article
author_id | article_id
Author
_id | fname | lname | email
Volume
vol_no | magazine_id | year
Magazine
_id | Name
Book
_id | name
Item
_id | item_type_id | price
Item_Type
_id | item
Customer
cid | fname | lname | tel_no | building_no | street | city | pin
Transaction
transaction_id | date | cid | discount_code | total_price
Montly_Expenses
Month_of_year | Electrity_Cost | Heat_Cost | Water_Cost | Monthly_Salary | Monthly_rent
Employee_wage
Sin | Date | No_of_hours | Day_Salary
Employee
Sin | First_Name | Last_Name | Hourly_Wage
Item_Transaction
Item_id | transaction_id | quantity | quantity_price
Rent
Year | Monthly_Cost

Figure2: Relational database schema for Halifax Science Library

Normalization

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly. For HSL database designing we have considered the 3NF normalization form.
A database is in third normal form if it satisfies the below conditions:

- It is in second normal form
- There is no transitive functional dependency

By transitive functional dependency, we mean we have the following relationships in the table: A is functionally dependent on B, and B is functionally dependent on C. In this case, C is transitively dependent on A via B.

3rd Normal Form Example:

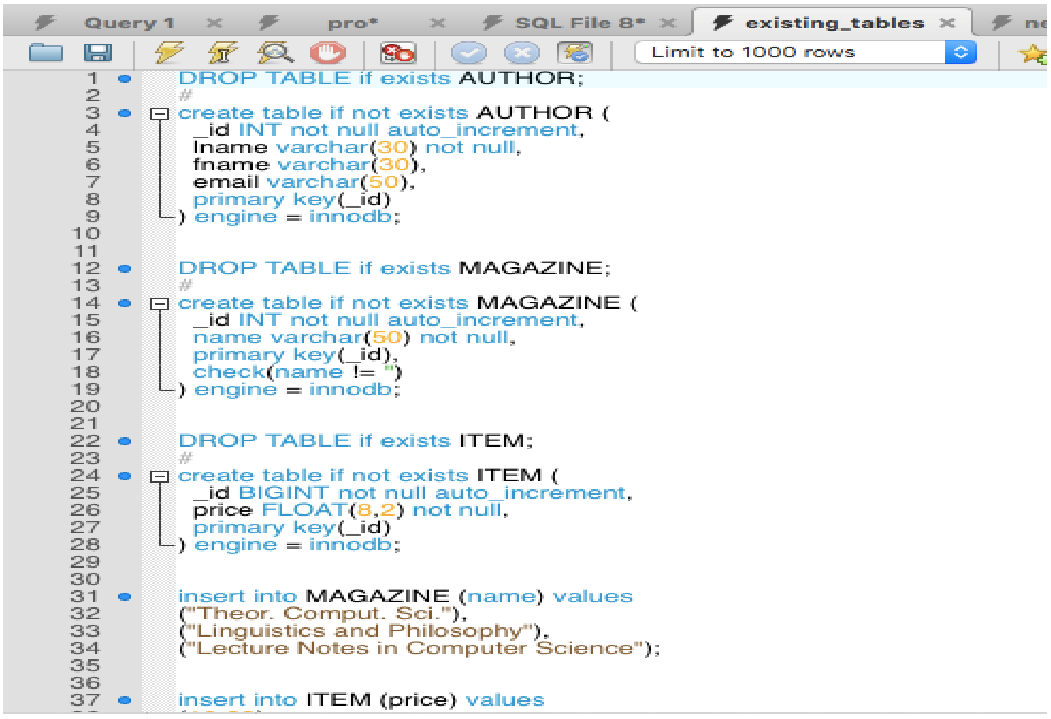
Author Table:

_id	fname	Lname	email

1. In the author table, _id determines fname(first name), lname(last name) and email of the author.
2. There is no transitive dependency in the table.
3. Author table satisfy 1NF conditions.

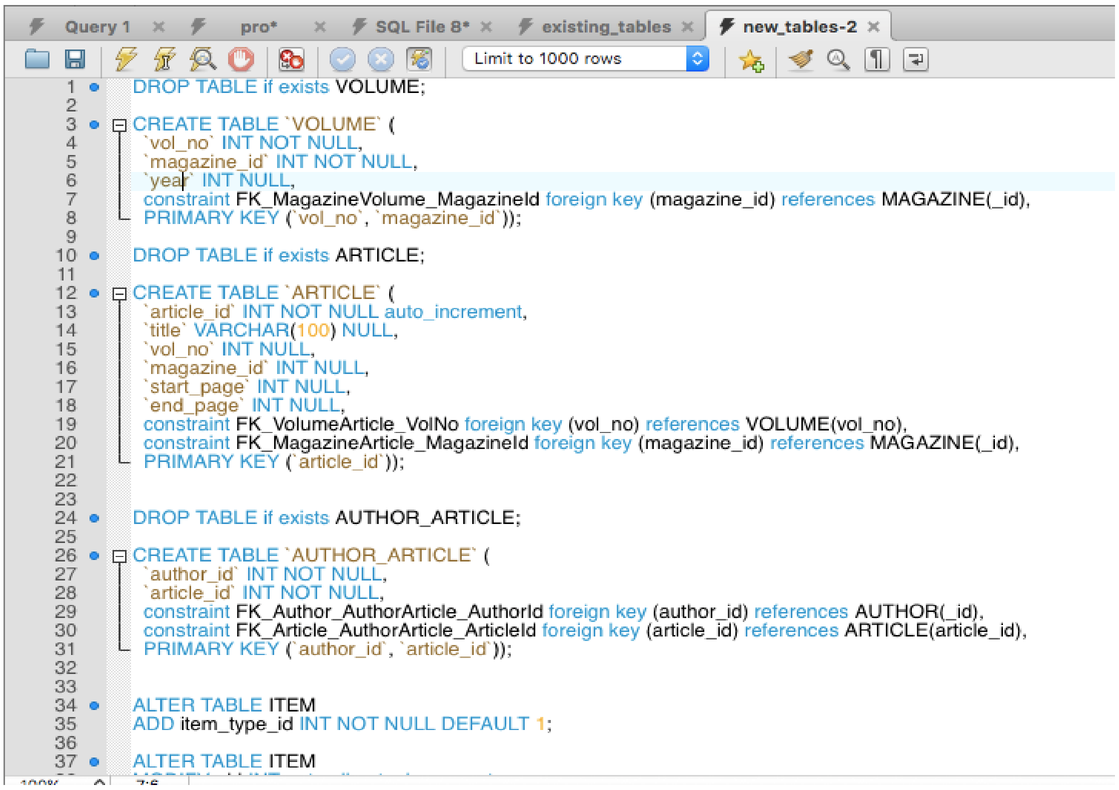
Creating tables in MySQL:

Based on the normalised database schema, some enhancements are done on existing tables and other new tables are created by considering the primary key and foreign key constraints.



```sql
DROP TABLE if exists AUTHOR;
#
create table if not exists AUTHOR (
  _id INT not null auto_increment,
  lname varchar(30) not null,
  fname varchar(30),
  email varchar(50),
  primary key(_id)
) engine = innodb;


DROP TABLE if exists MAGAZINE;
#
create table if not exists MAGAZINE (
  _id INT not null auto_increment,
  name varchar(50) not null,
  primary key(_id),
  check(name != '')
) engine = innodb;


DROP TABLE if exists ITEM;
#
create table if not exists ITEM (
  _id BIGINT not null auto_increment,
  price FLOAT(8,2) not null,
  primary key(_id)
) engine = innodb;


insert into MAGAZINE (name) values
("Theor. Comput. Sci."),
("Linguistics and Philosophy"),
("Lecture Notes in Computer Science");

insert into ITEM (price) values
```

Figure 3: Enhanced Existing tables

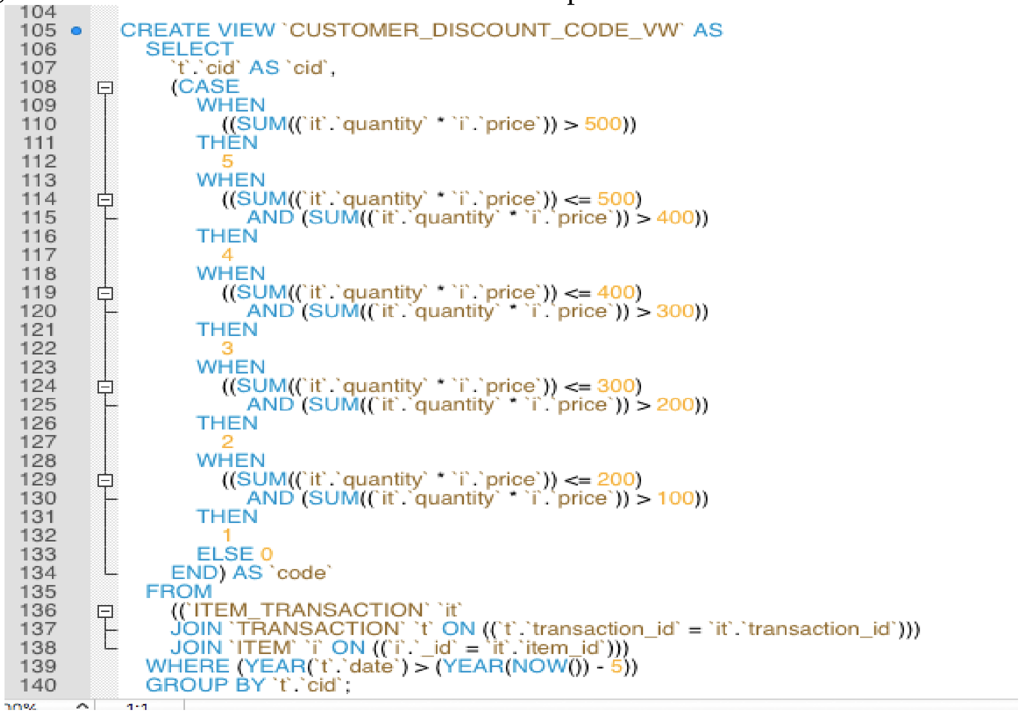


```sql
DROP TABLE if exists VOLUME;

CREATE TABLE `VOLUME` (
  `vol_no` INT NOT NULL,
  `magazine_id` INT NOT NULL,
  `year` INT NULL,
  constraint FK_MagazineVolume_MagazineId foreign key (magazine_id) references MAGAZINE(_id),
  PRIMARY KEY (`vol_no`, `magazine_id`));

DROP TABLE if exists ARTICLE;

CREATE TABLE `ARTICLE` (
  `article_id` INT NOT NULL auto_increment,
  `title` VARCHAR(100) NULL,
  `vol_no` INT NULL,
  `magazine_id` INT NULL,
  `start_page` INT NULL,
  `end_page` INT NULL,
  constraint FK_VolumeArticle_VolNo foreign key (vol_no) references VOLUME(vol_no),
  constraint FK_MagazineArticle_MagazineId foreign key (magazine_id) references MAGAZINE(_id),
  PRIMARY KEY (`article_id`));


DROP TABLE if exists AUTHOR_ARTICLE;

CREATE TABLE `AUTHOR_ARTICLE` (
  `author_id` INT NOT NULL,
  `article_id` INT NOT NULL,
  constraint FK_Author_AuthorArticle_AuthorId foreign key (author_id) references AUTHOR(_id),
  constraint FK_Article_AuthorArticle_ArticleId foreign key (article_id) references ARTICLE(article_id),
  PRIMARY KEY (`author_id`, `article_id`));


ALTER TABLE ITEM
ADD item_type_id INT NOT NULL DEFAULT 1;

ALTER TABLE ITEM
```

Figure 3: New tables of Schema

DISCOUNT PRICE CALCULATION:

A view,which finds the discount code of the customer according to his/her transaction history,is created for the PHP to calculate discount pricefor the transaction.



```sql
CREATE VIEW `CUSTOMER_DISCOUNT_CODE_VW` AS
    SELECT
        `t`.`cid` AS `cid`,
        (CASE
            WHEN
                ((SUM((`it`.`quantity` * `i`.`price`)) > 500))
            THEN
                5
            WHEN
                ((SUM((`it`.`quantity` * `i`.`price`)) <= 500)
                    AND (SUM((`it`.`quantity` * `i`.`price`)) > 400))
            THEN
                4
            WHEN
                ((SUM((`it`.`quantity` * `i`.`price`)) <= 400)
                    AND (SUM((`it`.`quantity` * `i`.`price`)) > 300))
            THEN
                3
            WHEN
                ((SUM((`it`.`quantity` * `i`.`price`)) <= 300)
                    AND (SUM((`it`.`quantity` * `i`.`price`)) > 200))
            THEN
                2
            WHEN
                ((SUM((`it`.`quantity` * `i`.`price`)) <= 200)
                    AND (SUM((`it`.`quantity` * `i`.`price`)) > 100))
            THEN
                1
            ELSE 0
        END) AS `code`
    FROM
        ((`ITEM_TRANSACTION` `it`
        JOIN `TRANSACTION` `t` ON ((`t`.`transaction_id` = `it`.`transaction_id`)))
        JOIN `ITEM` `i` ON ((`i`.`_id` = `it`.`item_id`)))
    WHERE (YEAR(`t`.`date`) > (YEAR(NOW()) - 5))
    GROUP BY `t`.`cid`;
```

Figure 4: View for Discount Price

Data2mongo.sh Code Explanation:

- Prompting the user and read the DB details through printf and read commands.

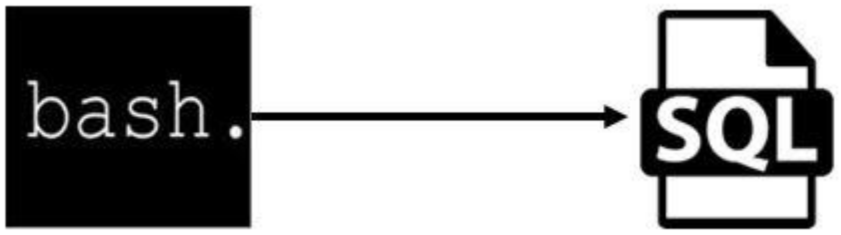


Figure 5: Execution of new_tables, existing_tables SQL

- Following SQL files will be run:

    1. rolback.sql contains the SQL queries to drop all the tables required.

    2. existing_tables.sql contains the create queries for pre-requisite tables.

3. new_tables.sql contains the create queries for the tables as per our schema and also the views need for the Step 6.

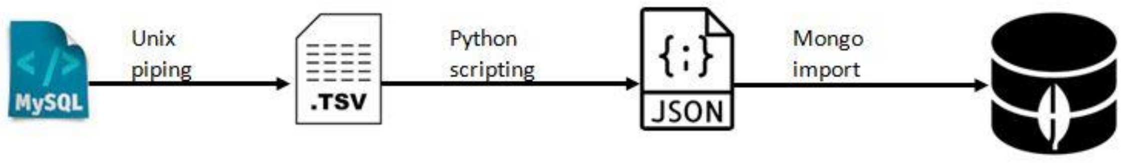


Figure 5: Execution flow for importing data from MYSQL to Mongo

- Get the contents of the AUTHOR table using SELECT * command and redirecting the output to a TSV file.

- In order to load the data into mongodb using the mongoimport utility , the data needs to be in JSON.

- We convert the TSV data to JSON by running the convert_TSV_to_JSON.py, where we load the entire TSV data into a Pandas Dataframe and convert to a JSON array using the tojson() function.

- Now that we have JSON array we use the mongoimport utility directly on the JSON file.

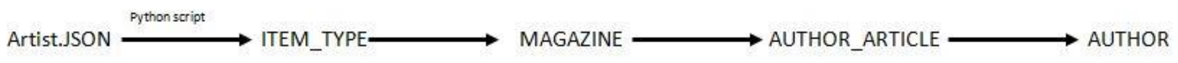


Figure 5: Execution flow of inserting articles JSON into the Mongo collections

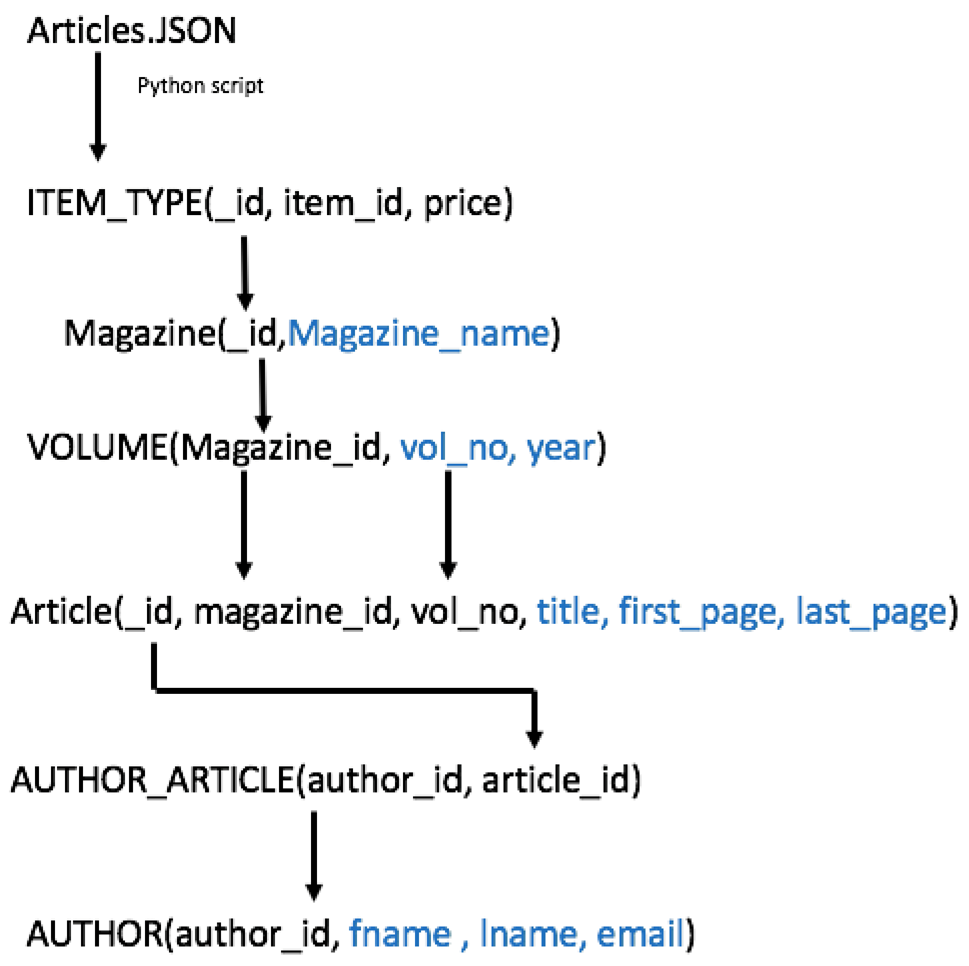
- In order to insert the articles.json into mongo collection, we run the python file "4C.py"

    1. For each line in the Articles JSON, we first create a record of its magazine if not there already in "magazine" and "item" collections.

    2. Then inserts the volume of the article in the volume collection if not already present.

    3. Then inserts the authors of the article in author_article and author collections if not already present.

- Finally inserts all the article information in article collection.
- Finally we remove the unwanted files.



Articles.JSON
Python script
ITEM_TYPE(_id, item_id, price)
Magazine(_id,Magazine_name)
VOLUME(Magazine_id, vol_no, year)
Article(_id, magazine_id, vol_no, title, first_page, last_page)
AUTHOR_ARTICLE(author_id, article_id)
AUTHOR(author_id, fname , lname, email)

Figure 5: Data Dependencies for inserting articles JSON into the Mongo collections

- The fields in blue are directly extracted from the articles.json

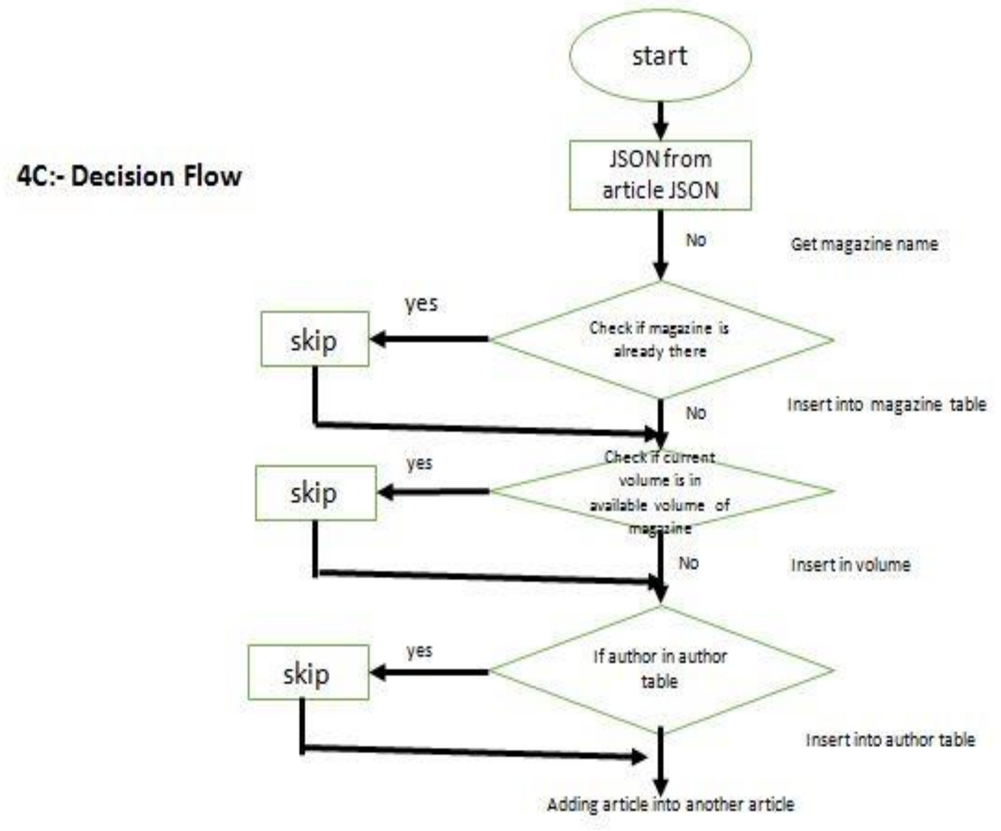
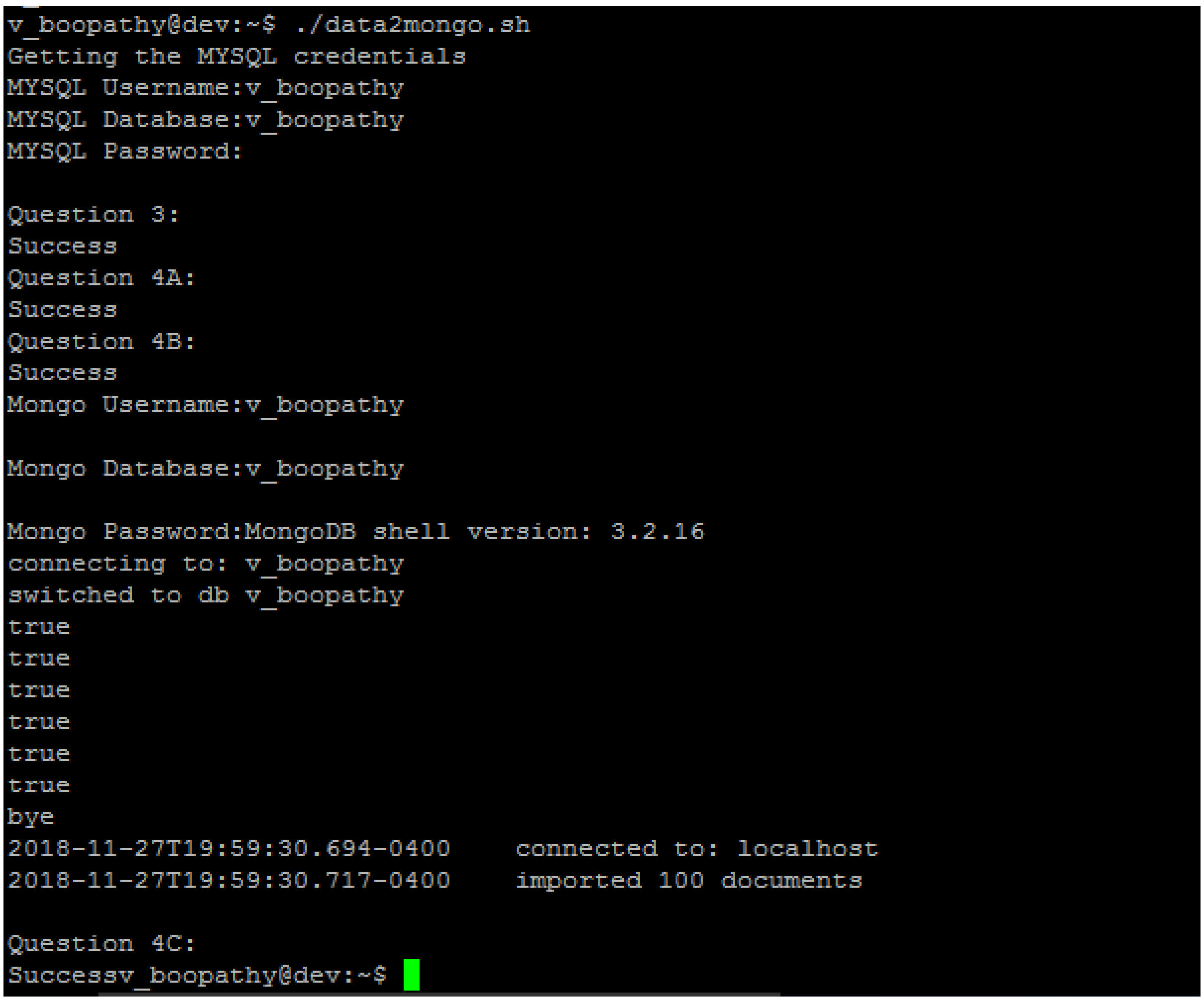

4C:- Decision Flow
start
JSON from article JSON
No
Get magazine name
yes
skip
Check if magazine is already there
No
Insert into magazine table
yes
skip
Check if current volume is in available volume of magazine
No
Insert in volume
yes
skip
If author in author table
Insert into author table
Adding article into another article

Figure 5: Decision tree for inserting articles JSON into the Mongo collections

Instructions for data2mongo:

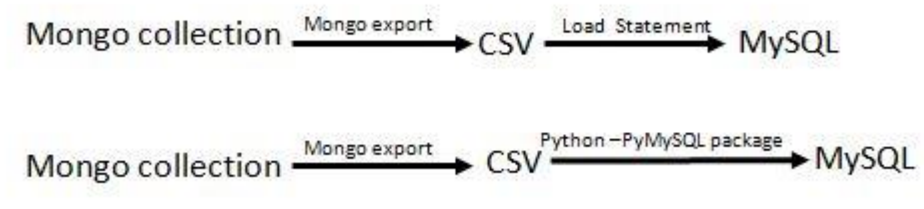
- Run the shell program with the command ./data2mongo.sh.
- Enter your MYSQL Username, Database name and Password when asked by the prompt.
- Enter your MongoDB Username, Database name and Password when asked by the prompt.
- It also shows which part of the questions it is running followed by a success message on success.



```
v_boopathy@dev:~$ ./data2mongo.sh
Getting the MYSQL credentials
MYSQL Username:v_boopathy
MYSQL Database:v_boopathy
MYSQL Password:

Question 3:
Success
Question 4A:
Success
Question 4B:
Success
Mongo Username:v_boopathy

Mongo Database:v_boopathy

Mongo Password:MongoDB shell version: 3.2.16
connecting to: v_boopathy
switched to db v_boopathy
true
true
true
true
true
true
bye
2018-11-27T19:59:30.694-0400    connected to: localhost
2018-11-27T19:59:30.717-0400    imported 100 documents

Question 4C:
Successv_boopathy@dev:~$
```

Figure 6: Execution of data2mongo script

Mongo2sql Code Explanation:



- We first export the mongo collections - volume, magazine, author and article as a CSV using mongo export utility.

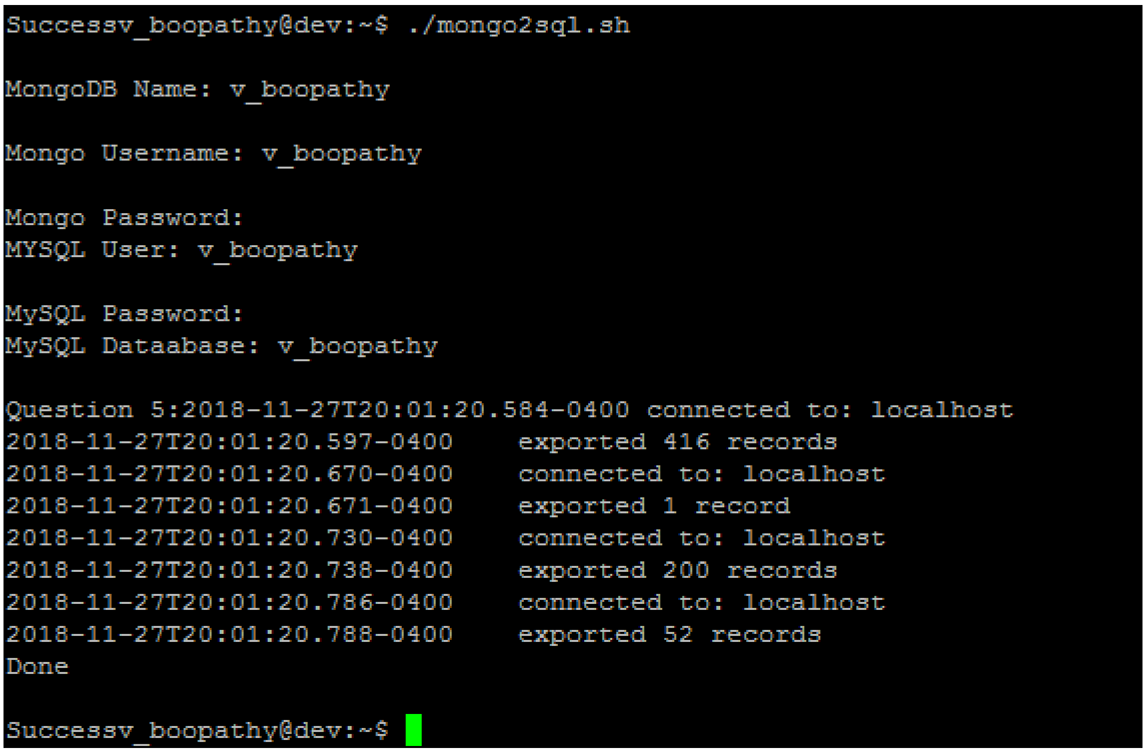
- We load the CSV files created using the LOAD statements in MYSQL.

- We also use "articles_importer.py" python file to load the CSV file into MYSQL using "pymsql" package.

- Finally remove the unwanted files.

Instructions for mongo2sql:

- Run the shell program with the command ./mongo2sql.sh.
- Enter your MYSQL Username, Database name and Password when asked by the prompt.
- Enter your MongoDB Username, Database name and Password when asked by the prompt.
- It shows "Done" message on success.



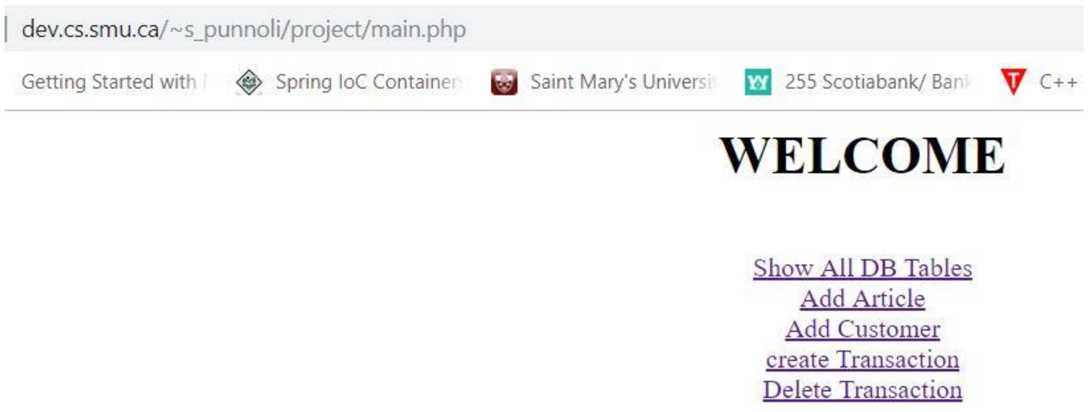
```
Successv_boopathy@dev:~$ ./mongo2sql.sh

MongoDB Name: v_boopathy

Mongo Username: v_boopathy

Mongo Password:
MYSQL User: v_boopathy

MySQL Password:
MySQL Dataabase: v_boopathy

Question 5:2018-11-27T20:01:20.584-0400 connected to: localhost
2018-11-27T20:01:20.597-0400    exported 416 records
2018-11-27T20:01:20.670-0400    connected to: localhost
2018-11-27T20:01:20.671-0400    exported 1 record
2018-11-27T20:01:20.730-0400    connected to: localhost
2018-11-27T20:01:20.738-0400    exported 200 records
2018-11-27T20:01:20.786-0400    connected to: localhost
2018-11-27T20:01:20.788-0400    exported 52 records
Done

Successv_boopathy@dev:~$
```

Figure 7: Execution of data2mongo script

PHP Web Application:

The complete HSL management system developed by PHP.

- **Home Screen:**
  This is home screen for the application. It consists of all the modules like showing all the database tables, adding article, adding customer, creating transaction, and deleting transaction of application.



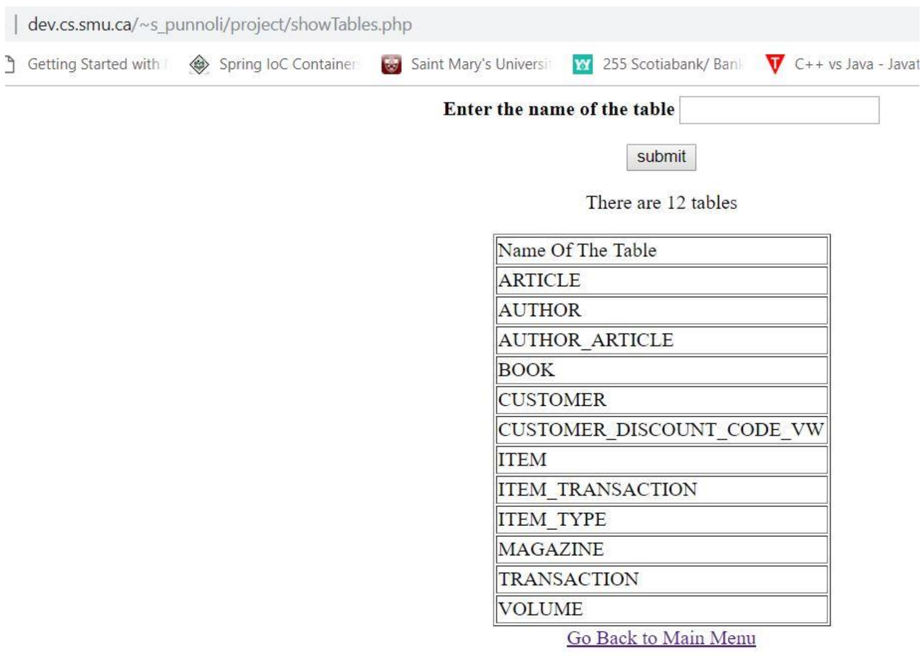
dev.cs.smu.ca/~s_punnoli/project/main.php

Getting Started with | Spring IoC Container | Saint Mary's Universit | 255 Scotiabank/ Bank | C++

WELCOME

Show All DB Tables
Add Article
Add Customer
create Transaction
Delete Transaction

- **Show Tables:**
  This screen displays all the tables in the database. It allows users to enter the name of the table and clicking on submit button displays content of the table.



dev.cs.smu.ca/~s_punnoli/project/showTables.php

Getting Started with | Spring IoC Container | Saint Mary's Universit | 255 Scotiabank/ Bank | C++ vs Java - Javat

Enter the name of the table

submit

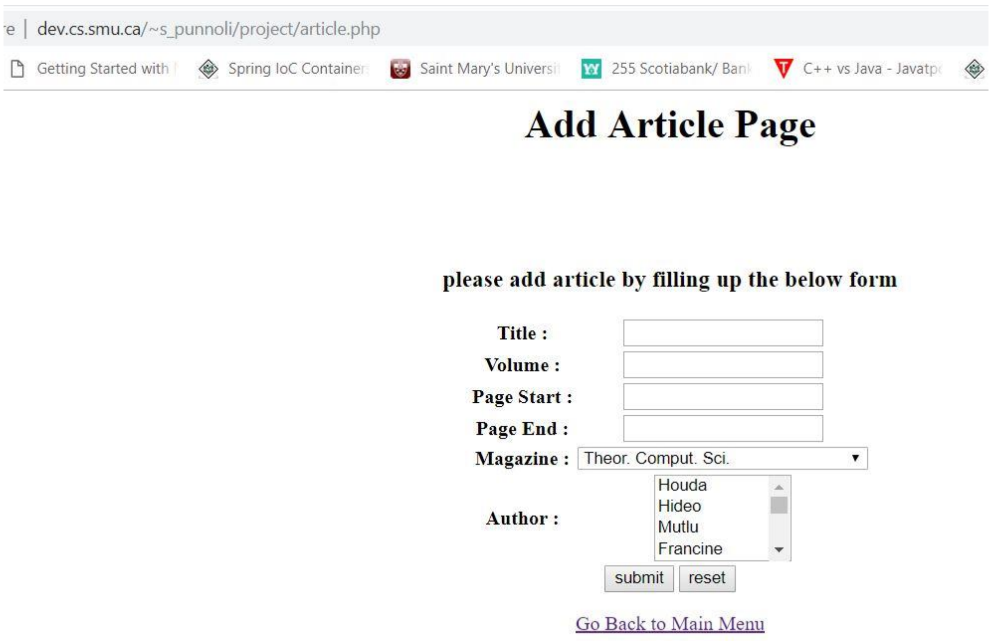
There are 12 tables

Name Of The Table
ARTICLE
AUTHOR
AUTHOR_ARTICLE
BOOK
CUSTOMER
CUSTOMER_DISCOUNT_CODE_VW
ITEM
ITEM_TRANSACTION
ITEM_TYPE
MAGAZINE
TRANSACTION
VOLUME

Go Back to Main Menu

- **Add Article**

In this screen the user enters the details of an article. This PHP webpage inserts the information into database. If any constraints are violated, application will inform to user. In this webpage the Volume should be already available for the given magazine in the database to insert an article. If the volume is not available, it will throw exception. In the "new_tables.sql", we have added a volume number '1' for the magazine id '1' with the year '2018' for testing purpose.


dev.cs.smu.ca/~s_punnoli/project/article.php

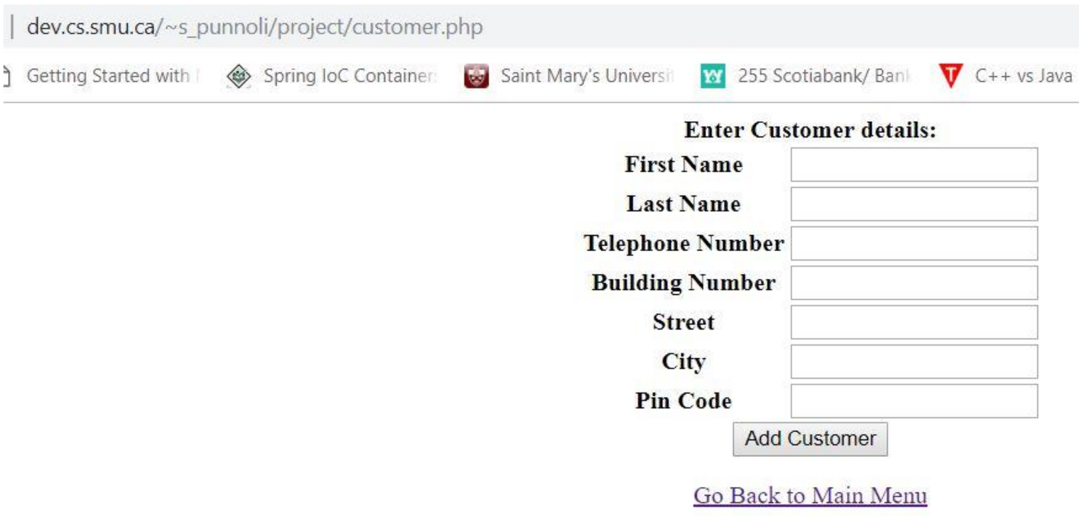
Add Article Page

please add article by filling up the below form

Title :	
Volume :	
Page Start :	
Page End :	
Magazine :	Theor. Comput. Sci.
Author :	Houda Hideo Mutlu Francine

- **Add New Customer:**

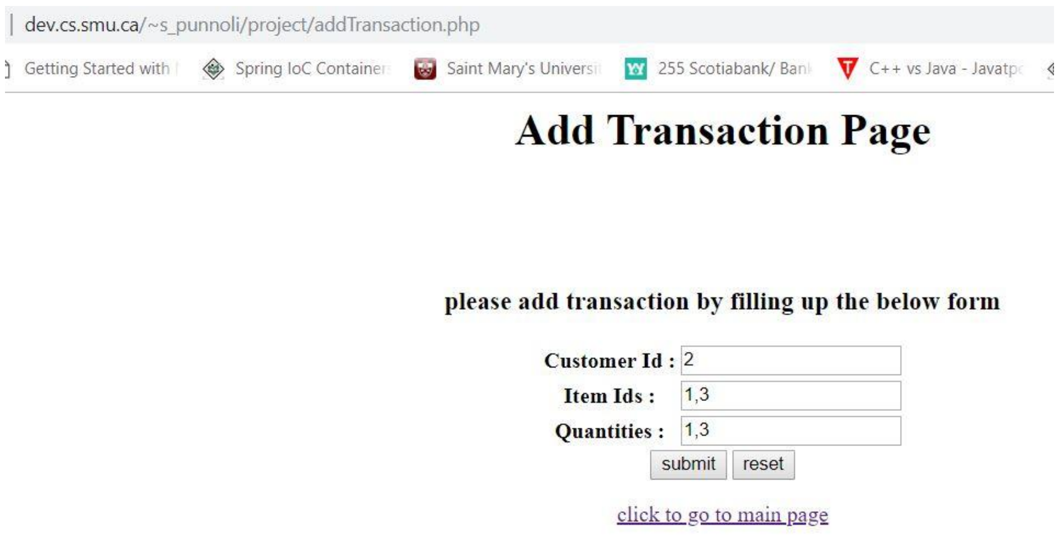
In this screen the user enters the details of the customer and PHP webpage inserts data into corresponding table in database. If any constraints areviolated the application informs to user. Telephone number should be a 10-digit number.


dev.cs.smu.ca/~s_punnoli/project/customer.php

Enter Customer details:

First Name	
Last Name	
Telephone Number	
Building Number	
Street	
City	
Pin Code	

- **Add New Transaction:**
  In this screen the user enters the customer id, item ids and their respective quantity separated by comma. If the item id count and the quantity count differ, it will not be accepted. When the user enters a valid input, data gets updated in database.

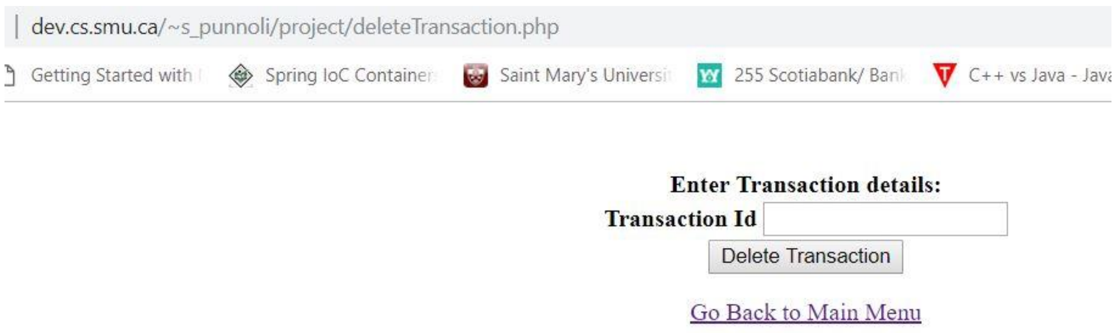

dev.cs.smu.ca/~s_punnoli/project/addTransaction.php

Add Transaction Page

please add transaction by filling up the below form

Customer Id :	2
Item Ids :	1,3
Quantities :	1,3

- **Cancel Transaction:**
  From this screen the user can delete the transaction details in database by entering the transaction id.


dev.cs.smu.ca/~s_punnoli/project/deleteTransaction.php

Enter Transaction details:

Transaction Id

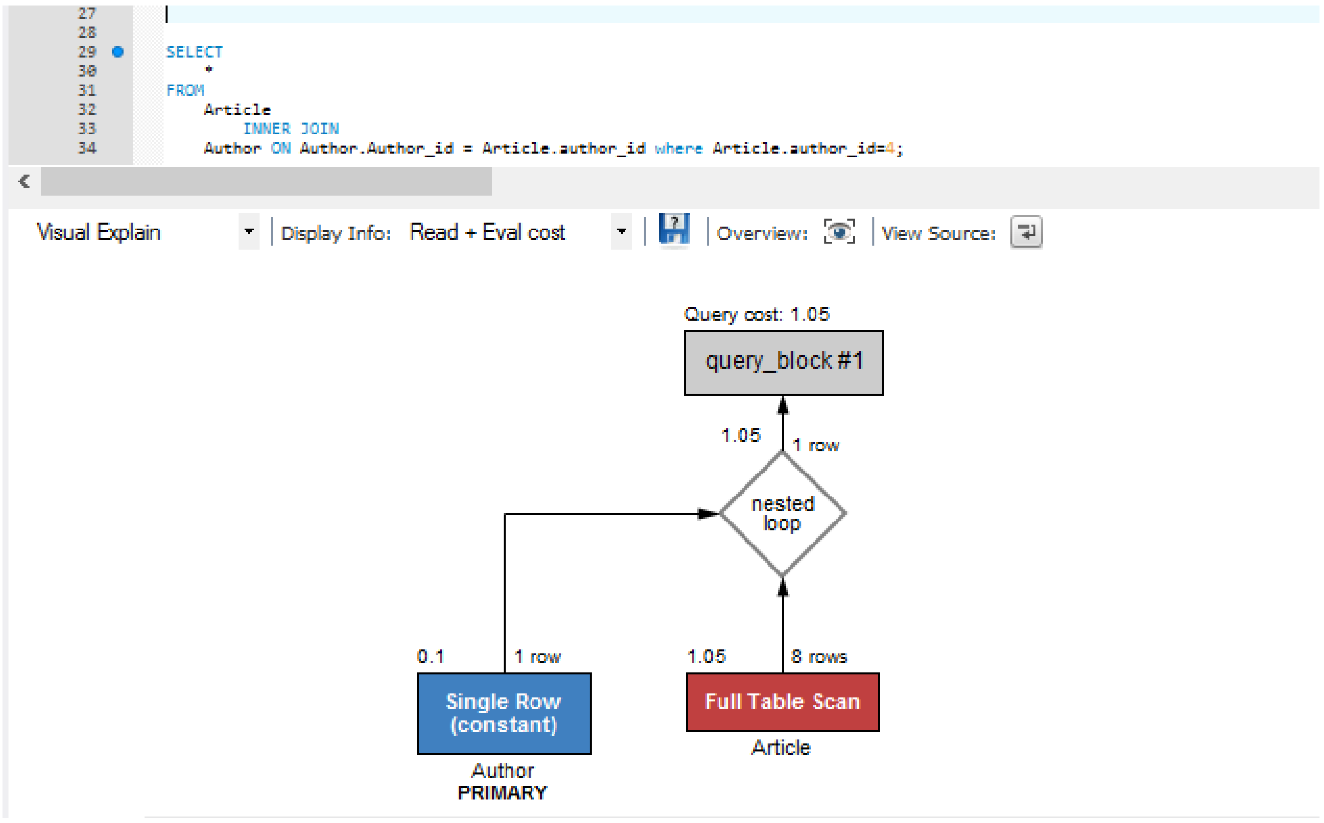
Optimization

The database and query optimization should determine the most efficient way to execute a given query by considering the possible query plans on the database. There are different ways to do the query optimization. Some of them are as follows.

a. Creating index

Indexing is powerful means to optimize the query cost. A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and storage space to maintain the index data structure. If there is no index, MySQL reads through the entire table to retrieve the data that we need and this increases the cost when the table size becomes more. If we create an index, MySQL directly retrieves the data without going through the entire table.
Basically an index is a map of all the keys that is sorted in order implemented using a B+ tree.
As an example to illustrate the effect of indexes in optimizing the cost, we have performed an inner join on Author and Article tables using the author_id.

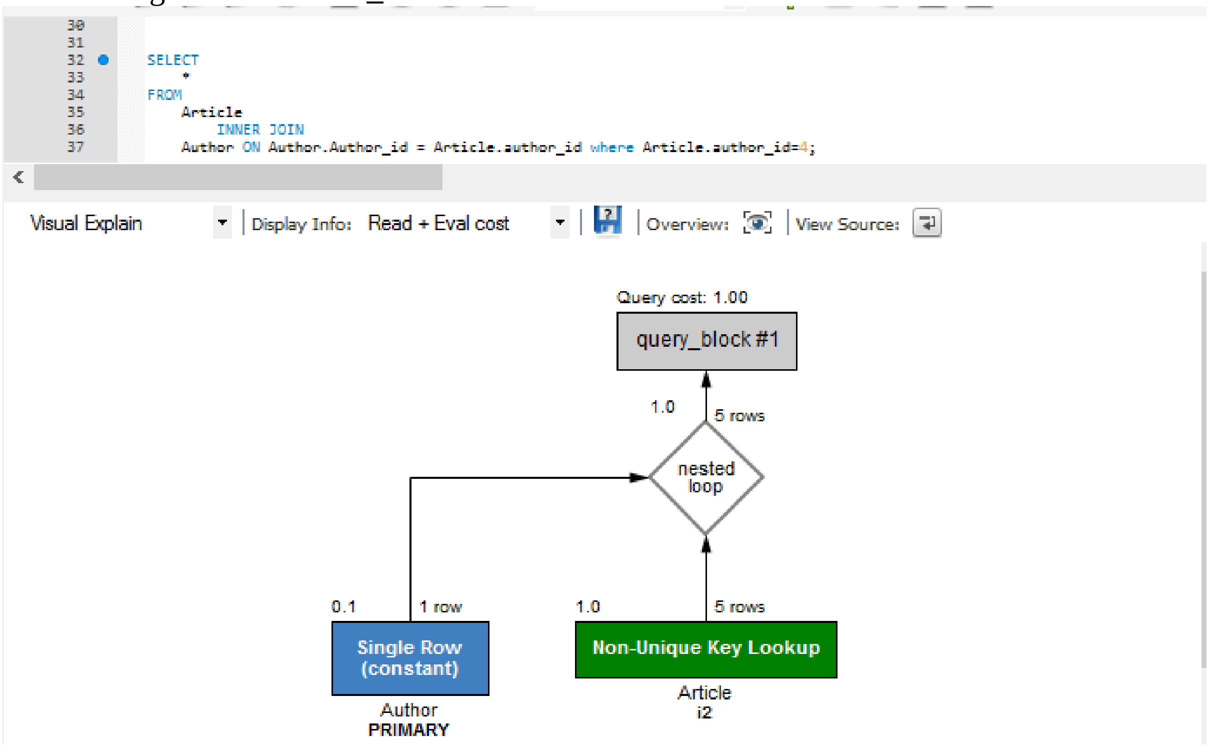
Before adding index to author_id column in both Author and Article table:



```

SELECT
    *
FROM
    Article
        INNER JOIN
    Author ON Author.Author_id = Article.author_id where Article.author_id=4;
```

Figure 3: Full-Table-Scan from Article table and joining with Author table

After adding index to author_id column in both Author and Article table:



```
SELECT
    *
FROM
    Article
        INNER JOIN
    Author ON Author.Author_id = Article.author_id where Article.author_id=4;
```

Visual Explain | Display Info: Read + Eval cost | Overview: | View Source:

Figure 4: Non-Unique Key Lookup from Article table and joining with Author table

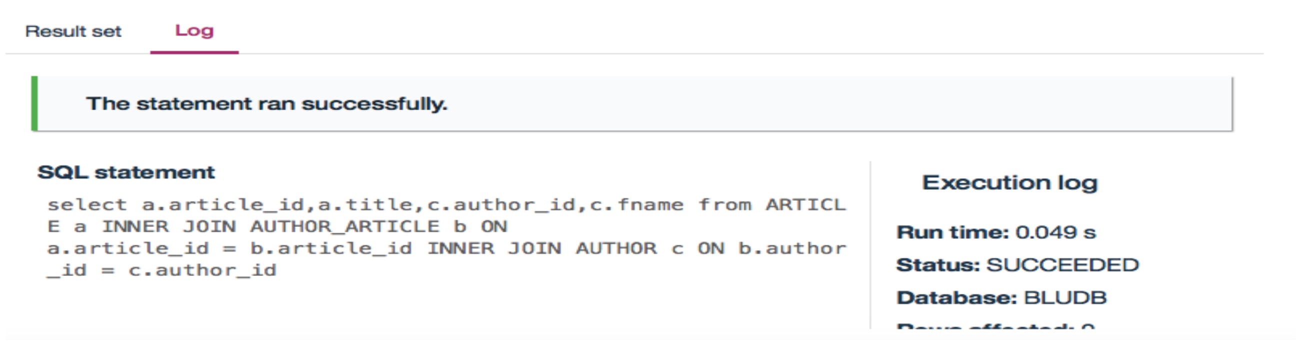
Therefore, after adding the index, the query cost got reduced since a full table scan has been avoided.

b. **Avoid OR conditions with index**
   Since the OR condition fetch the results of each part of the condition separately, the effect of index will be less.
c. **Avoid sorting with a mixed order**
   In order to the get best results from applying the index, we must not produce a mixed order by ordering ascending on one column and descending on another column.
d. **Non numeric column indexing**
   Adding index to a non-numeric column will not add much difference with query cost.
e. **Avoid LIKE searches with prefix wildcards**
   Having a wildcard '%' at the beginning of the pattern will prevent the database from using an index for this column's search. Such searches can take a while. In order to optimize the query, it is better to avoid them.
f. **Add caching when necessary**
   Adjust the size and properties of the memory areas that MySQL uses for caching. With efficient use of the InnoDB buffer pool, MyISAM key cache, and the MySQL query cache, repeated queries run faster because the results are retrieved from memory the second and subsequent times.

Comparing execution time of MySQL with IBMDB2:

IBM DB2 and MySQL execution time depends on lots of factors. For some queries, MySQL turns out to be faster than IBM DB2. In order to compare the execution time of DB2 and MySQL, we executed the same query on both the databases. For our query, DB2 took lesser time than MySQL.

IBMDB2:



The statement ran successfully.

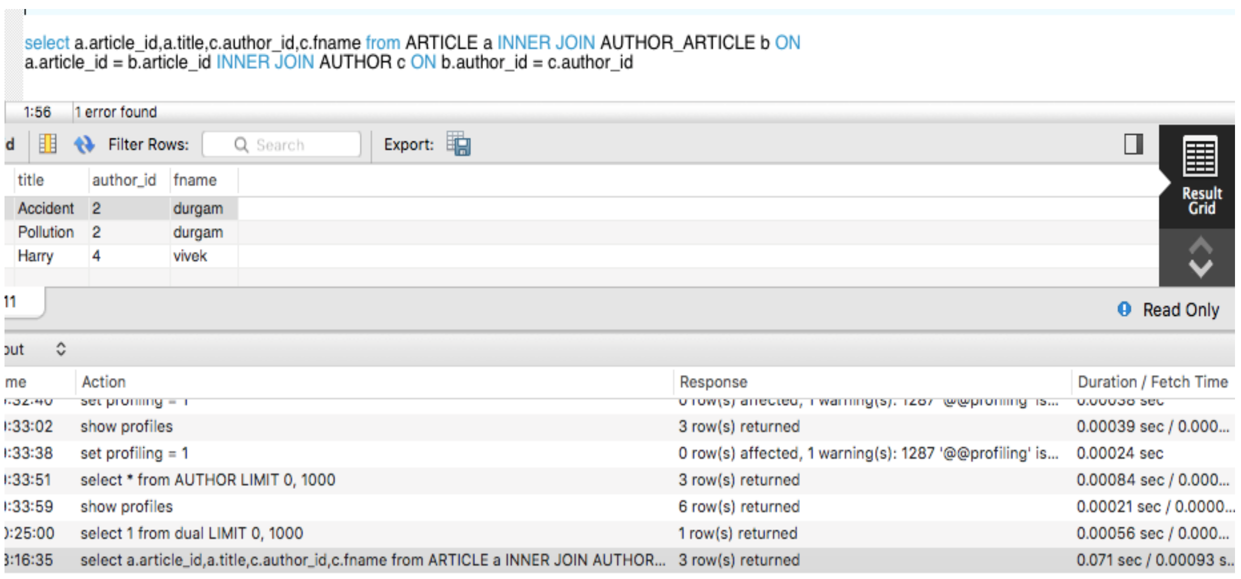
SQL statement

```
select a.article_id,a.title,c.author_id,c.fname from ARTICL
E a INNER JOIN AUTHOR_ARTICLE b ON
a.article_id = b.article_id INNER JOIN AUTHOR c ON b.author
_id = c.author_id
```

Execution log

Run time: 0.049 s
Status: SUCCEEDED
Database: BLUDB

MySQL



```
select a.article_id,a.title,c.author_id,c.fname from ARTICLE a INNER JOIN AUTHOR_ARTICLE b ON
a.article_id = b.article_id INNER JOIN AUTHOR c ON b.author_id = c.author_id
```

1:56 1 error found

title	author_id	fname
Accident	2	durgam
Pollution	2	durgam
Harry	4	vivek

```
me      Action                                                        Response                                                   Duration / Fetch Time
:32:40  set profiling = 1                                             0 row(s) affected, 1 warning(s): 1287 '@@profiling' is...  0.00038 sec
:33:02  show profiles                                                 3 row(s) returned                                          0.00039 sec / 0.000...
:33:38  set profiling = 1                                             0 row(s) affected, 1 warning(s): 1287 '@@profiling' is...  0.00024 sec
:33:51  select * from AUTHOR LIMIT 0, 1000                            3 row(s) returned                                          0.00084 sec / 0.000...
:33:59  show profiles                                                 6 row(s) returned                                          0.00021 sec / 0.0000...
:25:00  select 1 from dual LIMIT 0, 1000                              1 row(s) returned                                          0.00056 sec / 0.000...
:16:35  select a.article_id,a.title,c.author_id,c.fname from ARTICLE a INNER JOIN AUTHOR...  3 row(s) returned           0.071 sec / 0.00093 s...
```

References and tools

- Query Optimization**:** *https://www.eversql.com/sql-performance-tuning-tips-for-mysql-query-optimization/*
- MySQL documentation: https://dev.mysql.com/doc/refman/5.5/en/select-optimization.html
- Indexing: https://stackoverflow.com/questions/3049283/mysql-indexes-what-are-the-best-practices
- https://www.geeksforgeeks.org/sql-case-statement/
- https://stackoverflow.com/questions/3801188/sql-query-optimization
- https://docs.mongodb.com/manual/
- https://www.w3schools.com/php/
- https://www.youtube.com/watch?v=7fjJe5mgpVU&index=8&list=PLo4y_OQEqhGgwwhIJLxBrwFt8b3jYeg4v&t=138s
- https://www.youtube.com/watch?v=_mBTSBcT9Og&index=9&list=PLo4y_OQEqhGgwwhIJLxBrwFt8b3jYeg4v&t=0s