

## CSL7090: Software and Data Engineering

### SDE Assignment2

#### Task1:

For This task, we were required to migrate the data from MongoDB to Mysql automatically. This cannot be done using a simple command, thus first the database is exported in JSON format, and then the scripts are used to make the changes into the respective Mysql Database and for entering new records.

**MongoDB** and **Mysql** are both used **locally** using the localhost and port numbers **27017** and **3306** respectively. **Python** is used for scripting. First, the script **script\_db.py** is used to create a database into Mysql using the JSON file and then the file **input.py** can be used to enter a new document into the **Books** collection and it automatically calls **script.py** which is used to enter that new record to Mysql. Also, script.py can also be run independently to make the changes into the Mysql database.

**A.** To create a database, collection, and document named LibraryDB, Books in MongoDB use the commands.

- Use Databasename; -> use LibraryDB;
- db.createCollection( <name> ) -> db.createCollection("Books");
- db.Books.insert(<Document>) -> db.Books.insert( {"ISBN" : "000000000000000"} )

**B.** To create the database in MySQL, use **script\_db.py** script which exports the MongoDB into JSON format and then uses the keys as columns to create the Database in Mysql.

Here three tables are created because the attributes **Author** and **category** can be multivalued and thus they cannot be placed into a single table since it neglects **normalization** thus three tables are created and the **ids** which are created while inserting data in **MongoDB** is used as the **primary key** for books table and same is used for referential integrity.

**Note:** ISBN number or **Accession No.** can also be used as the primary key.

**C.** After creating the Database **script\_db.py** calls **script.py** which inserts the documents present in the MongoDB Books collection automatically into Mysql Database.

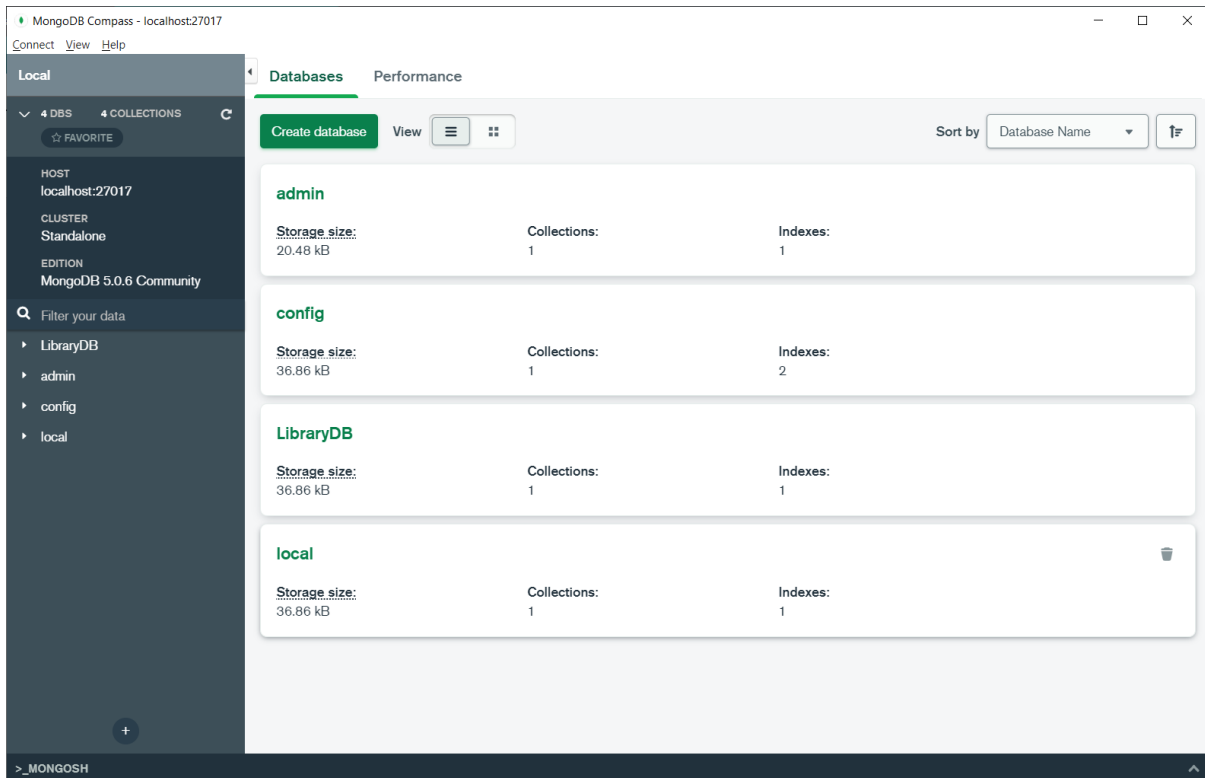
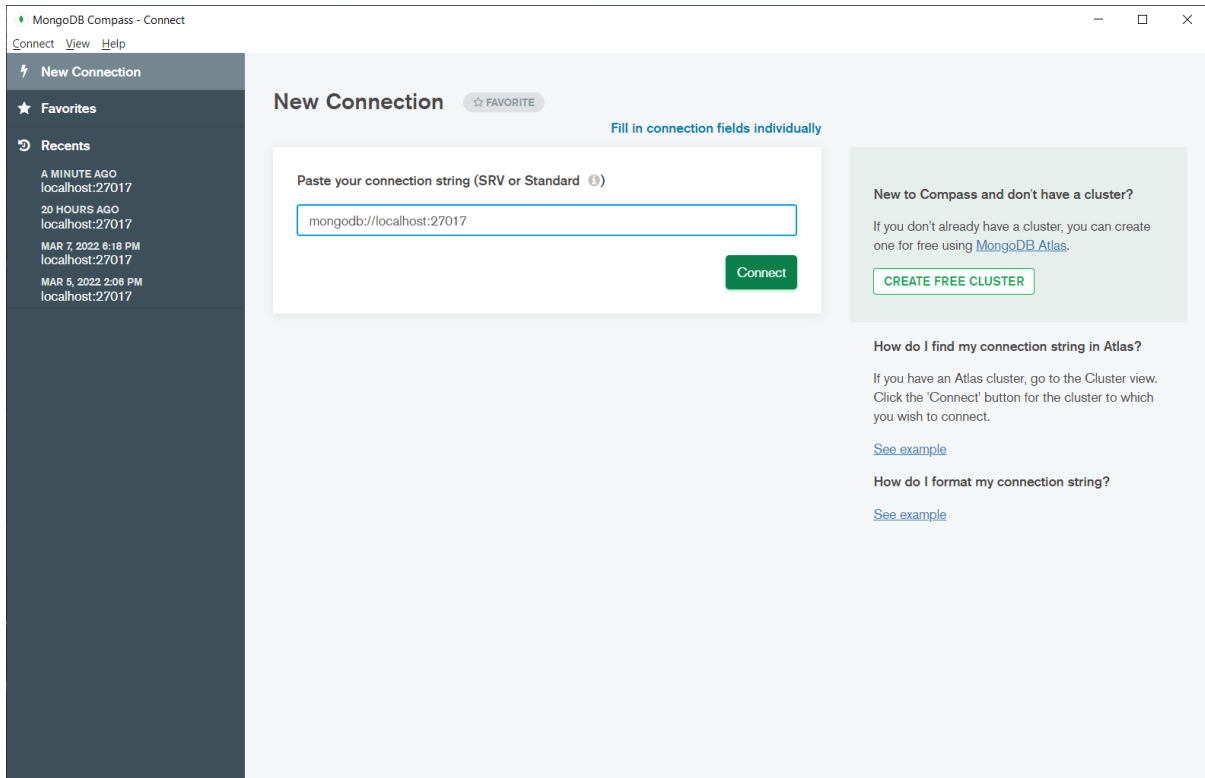
**D.** The file **insert.py** inserts a new document into the MongoDB server by taking input from the user and then call script.py which insert the new document into the Mysql Database.

**E.** The file **script.py** can also be run individually to make the changes of insertion or deletion into the Mysql Database.

**F.** Whenever a document is deleted from the MongoDB server then the same needs to be deleted from the Mysql server which can be done by using **script.py**.

**Outputs are shown below.**

# 1. Insert 10 documents into MongoDB manually showing output using Mongo Compass.



MongoDB Compass - localhost:27017/LibraryDB

Connect View Help

Local

4 DBS 4 COLLECTIONS

☆ FAVORITE

HOST  
localhost:27017

CLUSTER  
Standalone

EDITION  
MongoDB 5.0.6 Community

Filter your data

LibraryDB + -

Books

admin

config

local

+

>\_MONGOSH

Collections

Create collection View

Sort by Collection Name

Books

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	12	298.00 B	1	36.86 kB

```
Windows PowerShell
> db.Books.insert({"ISBN" : "9780619216351","Accession No." : "040620180010","Title" : "Java programming : from problem analysis to program design","Author" : "D. S. Malik",
,"Publisher" : "Boston : Course Technology/Thomson Learning","Edition" : 1,"Year of Publication" : 2003, "Category" : "java","Total Number of Pages" : 998,"Price" : 4897.53
})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"ISBN" : "059600088X","Accession No." : "230820211652","Title" : "Java programming with Oracle JDBC","Author" : "Donald Bales","Publisher" : "Sebastopol,
CA : O'Reilly","Edition" : 1,"Year of Publication" : 2002, "Category" : "java","Total Number of Pages" : 506,"Price" : 925})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"ISBN" : "9780136631705","Accession No." : "170220210045","Title" : "Advanced C programming","Author" : " Oualline,Steve","Publisher" : "New York : Brady
Pub.", "Edition" : 2,"Year of Publication" : 1992, "Category" : "C","Total Number of Pages" : 442,"Price" : 5726})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"ISBN" : "9780321356567","Accession No." : "191220190102","Title" : "C++ programming","Author" : " Ullman,Larry E.(Larry Edward)","Publisher" : "Berkele
y, Calif. : Peachpit Press","Edition" : 1,"Year of Publication" : 2006, "Category" : "C","Total Number of Pages" : 532,"Price" : 600})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"ISBN" : "9781887902991","Accession No." : "310120120523","Title" : "Python programming","Author" : "John M. Zelle","Publisher" : "Franklin, Beedle","Edi
tion" : 1,"Year of Publication" : 2004, "Category" : "python","Total Number of Pages" : 532,"Price" : 900})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"ISBN" : "9781435455009","Accession No." : "030120220023","Title" : "Python programming for the absolute beginner","Author" : "Dawson,Mike","Publisher" :
" Boston, MA : Course Technology Cengage Learning","Edition" : 3,"Year of Publication" : 2010, "Category" : "python","Total Number of Pages" : 484,"Price" : 1000})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"ISBN" : "9781617292453","Accession No." : "010120220022","Title" : "Hello Raspberry Pi! : Python programming for kids and other beginners","Author" : "H
eitz,Ryan","Publisher" : "Shelter Island, NY : Manning Publications Co.", "Edition" : 1,"Year of Publication" : 2016, "Category" : "python","Total Number of Pages" : 322,"Pr
ice" : 400})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"ISBN" : "9780314933096","Accession No." : "260820102014","Title" : "Introduction to data structures and algorithm analysis","Author" : "Naps,Thomas L,Si
ngh,Bhagat","Publisher" : " St. Paul, MN : West Pub. Co.", "Edition" : 2,"Year of Publication" : 1992, "Category" : "python,c","Total Number of Pages" : 714,"Price" : 800})
WriteResult({ "nInserted" : 1 })
> db.Books.insert({"ISBN" : "9780911537048","Accession No." : "270520100056","Title" : "Reliable data structures in C","Author" : "Plum,Thomas","Publisher" : "Cardiff, N.J.
: Plum Hall", "Edition" : 1,"Year of Publication" : 1985, "Category" : "c","Total Number of Pages" : 276,"Price" : 250})
WriteResult({ "nInserted" : 1 })
>
```

MongoDB Compass - localhost:27017/LibraryDB.Books

ConnectViewCollectionHelp

Local

7 DBS14 COLLECTIONS

☆ FAVORITE

HOST  
localhost:27017

CLUSTER  
Standalone

EDITION  
MongoDB 5.0.6 Community

Filter your data

Library

LibraryDB

Books

Rent

admin

config

local

test

+>\_MONGOSH

LibraryDB.Books

Documents

LibraryDB.Books

DOCUMENTS 9

STORAGE SIZE 20.5KB

AVG. SIZE 314B

INDEXES 1

TOTAL SIZE 36.9KB

AVG. SIZE 36.9KB

DocumentsAggregationsSchemaExplain PlanIndexesValidation

FILTER { field: 'value' }

OPTIONS

FIND

RESET

ADD DATA

VIEW

Displaying documents 1 - 9 of 9

REFRESH

\_id: ObjectId("622f392ac6ce4aa523a5575e")

ISBN: "9780619216351"

Accession No.: "040620180010"

Title: "Java programming : from problem analysis to program design"

Author: "D. S. Malik"

Publisher: "Boston : Course Technology/Thomson Learning"

Edition: 1

Year of Publication: 2003

Category: "java"

Total Number of Pag\_: 998

Price: 4897.53

>

\_id: ObjectId("622f3949c6ce4aa523a5575f")

ISBN: "059600888X"

Accession No.: "230820211652"

Title: "Java programming with Oracle JDBC"

Author: "Donald Bales"

Publisher: "Sebastopol, CA : O'Reilly"

Edition: 1

Year of Publication: 2002

Category: "java"

Total Number of Pag\_: 506


Price: 925

\_id: ObjectId("622f395ac6ce4aa523a55760")

ISBN: "9780136631705"

Accession No.: "170220210045"

2. Using **script\_db.py** to migrate the database in Mysql and insert records that are present in MongoDB.

 Windows PowerShell

```
PS D:\github\sde\assignment2\task1> python script_db.py
2022-03-14T18:22:20.466+0530    connected to: mongodb://localhost:27017/LibraryDB
2022-03-14T18:22:20.484+0530    exported 9 records
*****
Table created Sucessfully.
*****
Table created Sucessfully.
*****
Table created Sucessfully.
*****
Checking the Mysql Database for updates...
2022-03-14T18:22:22.004+0530    connected to: mongodb://localhost:27017/LibraryDB
2022-03-14T18:22:22.022+0530    exported 9 records
*****
Number of records present in MongoDB: 9

Number of records present in Mysql: 0

*****
Records need to be entered in Mysql

Entering the record:

ID                : 622f392ac6ce4aa523a5575e
ISBN              : 9780619216351
Accession No.     : 040620180010
Title             : Java programming : from problem analysis to program design
Author            : ['D. S. Malik']
Publisher         : Boston : Course Technology/Thomson Learning
Edition           : 1.0
Year of Publication : 2003.0
Category          : ['java']
Total Number of Pages : 998.0
Price             : 4897.53

1 Record Inserted sucessfully.
*****
Entering the record:

ID                : 622f3949c6ce4aa523a5575f
```

\*\*\*\*\*

Entering the record:

ID : 622f3949c6ce4aa523a5575f  
ISBN : 059600088X  
Accession No. : 230820211652  
Title : Java programming with Oracle JDBC  
Author : ['Donald Bales']  
Publisher : Sebastopol, CA : O'Reilly  
Edition : 1.0  
Year of Publication : 2002.0  
Category : ['java']  
Total Number of Pages : 506.0  
Price : 925.0

1 Record Inserted sucessfully.

\*\*\*\*\*

Entering the record:

ID : 622f395ac6ce4aa523a55760  
ISBN : 9780136631705  
Accession No. : 170220210045  
Title : Advanced C programming  
Author : ['Oualline', 'Steve']  
Publisher : New York : Brady Pub.  
Edition : 2.0  
Year of Publication : 1992.0  
Category : ['C']  
Total Number of Pages : 442.0  
Price : 5726.0

1 Record Inserted sucessfully.

\*\*\*\*\*

Entering the record:

ID : 622f396dc6ce4aa523a55761  
ISBN : 9780321356567  
Accession No. : 191220190102  
Title : C++ programming  
Author : ['Ullman', 'Larry E.(Larry Edward)']  
Publisher : Berkeley, Calif. : Peachpit Press

1 Record Inserted sucessfully.

\*\*\*\*\*

Entering the record:

ID : 622f39b5c6ce4aa523a55765  
ISBN : 9780314933096  
Accession No. : 260820102014  
Title : Introduction to data structures and algorithm analysis  
Author : ['Naps', 'Thomas L', 'Singh', 'Bhagat']  
Publisher : St. Paul, MN : West Pub. Co.  
Edition : 2.0  
Year of Publication : 1992.0  
Category : ['python', 'c']  
Total Number of Pages : 714.0  
Price : 800.0

1 Record Inserted sucessfully.

\*\*\*\*\*

Entering the record:

ID : 622f39c7c6ce4aa523a55766  
ISBN : 9780911537048  
Accession No. : 270520100056  
Title : Reliable data structures in C  
Author : ['Plum', 'Thomas']  
Publisher : Cardiff, N.J. : Plum Hall  
Edition : 1.0  
Year of Publication : 1985.0  
Category : ['c']  
Total Number of Pages : 276.0  
Price : 250.0

1 Record Inserted sucessfully.

\*\*\*\*\*

Do you want the complete status of all records...y/n

n

Thank you..

PS D:\github\sde\assignment2\task1>

### 3. Showing the data in MySQL database.

```
Select MySQL 8.0 Command Line Client
mysql> use librarydb;
Database changed
mysql> select * from books;
```

ID	ISBN	Accession	Title	Publisher	Edition	Year	Total	Pr
622f392ac6ce4aa523a5575e	9780619216351	040620180010	Java programming : from problem analysis to program design	Boston : Course Technology/Thomson Learning	1.0	2003.0	998.0	48
622f3949c6ce4aa523a5575f	059600088X	230820211652	Java programming with Oracle JDBC	Sebastopol, CA : O'Reilly	1.0	2002.0	506.0	92
622f395ac6ce4aa523a55760	9780136631705	170220210045	Advanced C programming	New York : Brady Pub.	2.0	1992.0	442.0	57
622f396dc6ce4aa523a55761	9780321356567	191220190102	C++ programming	Berkeley, Calif. : Peachpit Press	1.0	2006.0	532.0	60
622f3980c6ce4aa523a55762	9781887902991	310120120523	Python programming	Franklin, Beedle	1.0	2004.0	532.0	90
622f3991c6ce4aa523a55763	9781435455009	030120220023	Python programming for the absolute beginner	Boston, MA : Course Technology Cengage Learning	3.0	2010.0	484.0	10
622f39a4c6ce4aa523a55764	9781617292453	010120220022	Hello Raspberry Pi! : Python programming for kids and other beginners	Shelter Island, NY : Manning Publications Co.	1.0	2016.0	322.0	40
622f39b5c6ce4aa523a55765	9780314933096	260820102014	Introduction to data structures and algorithm analysis	St. Paul, MN : West Pub. Co.	2.0	1992.0	714.0	80
622f39c7c6ce4aa523a55766	9780911537048	270520100056	Reliable data structures in C	Cardiff, N.J. : Plum Hall	1.0	1985.0	276.0	25

```
9 rows in set (0.00 sec)
```

```
mysql> select * from bookauthor;
```

ID	Author_name
622f392ac6ce4aa523a5575e	D. S. Malik
622f3949c6ce4aa523a5575f	Donald Bales
622f395ac6ce4aa523a55760	Oualline
622f395ac6ce4aa523a55760	Steve
622f396dc6ce4aa523a55761	Ullman
622f396dc6ce4aa523a55761	Larry E.(Larry Edward)
622f3980c6ce4aa523a55762	John M. Zelle
622f3991c6ce4aa523a55763	Dawson
622f3991c6ce4aa523a55763	Mike
622f39a4c6ce4aa523a55764	Heitz
622f39a4c6ce4aa523a55764	Ryan
622f39b5c6ce4aa523a55765	Naps
622f39b5c6ce4aa523a55765	Thomas L
622f39b5c6ce4aa523a55765	Singh
622f39b5c6ce4aa523a55765	Bhagat
622f39c7c6ce4aa523a55766	Plum
622f39c7c6ce4aa523a55766	Thomas

```
17 rows in set (0.00 sec)
```

```
mysql> select * from category;
```

```
mysql> select * from category;
```

ID	Category
622f392ac6ce4aa523a5575e	java
622f3949c6ce4aa523a5575f	java
622f395ac6ce4aa523a55760	C
622f396dc6ce4aa523a55761	C
622f3980c6ce4aa523a55762	python
622f3991c6ce4aa523a55763	python
622f39a4c6ce4aa523a55764	python
622f39b5c6ce4aa523a55765	python
622f39b5c6ce4aa523a55765	c
622f39c7c6ce4aa523a55766	c

```
10 rows in set (0.00 sec)
```



4. Using insert.py to insert data into MongoDB which will automatically insert data into mysql.

Windows PowerShell

Thank you..

PS D:\github\sde\assignment2\task1> python insert.py

\*\*\*\*\*

Enter Document Details:

ISBN:

9780471807025

Accession No.:

071020100004

Title:

Database : step-by-step

Author:

Gillenson,Mark L

Publisher:

New York : Wiley

Edition:

2

Year of Publication:

1985

Category:

dbms

Total Numner of Pages:

418

Price:

500

622f3cec81c43a94e977e918 Inserted Document Sucessfully

\*\*\*\*\*

Checking the Mysql Database for updates...

2022-03-14T18:32:37.798+0530 connected to: mongodb://localhost:27017/LibraryDB

2022-03-14T18:32:37.821+0530 exported 10 records

\*\*\*\*\*

Number of records present in MongoDB: 10

Number of records present in Mysql: 9

\*\*\*\*\*

Records need to be entered in Mysql

Entering the record:

ID : 622f3cec81c43a94e977e918

ISBN : 9780471807025

```
Year of Publication:
1985
Category:
dbms
Total Numner of Pages:
418
Price:
500
622f3cec81c43a94e977e918 Inserted Document Sucessfully
*****
Checking the Mysql Database for updates...
2022-03-14T18:32:37.798+0530    connected to: mongodb://localhost:27017/LibraryDB
2022-03-14T18:32:37.821+0530    exported 10 records
*****
Number of records present in MongoDB: 10

Number of records present in Mysql: 9

*****
Records need to be entered in Mysql

Entering the record:

ID                : 622f3cec81c43a94e977e918
ISBN              : 9780471807025
Accession No.     : 071020100004
Title             : Database : step-by-step
Author            : ['Gillenson', 'Mark L']
Publisher         : New York : Wiley
Edition           : 2
Year of Publication : 1985
Category          : ['dbms']
Total Number of Pages : 418
Price             : 500.0

1 Record Inserted sucessfully.
*****
Do you want the complete status of all records....y/n
n
Thank you..

PS D:\github\sde\assignment2\task1> █
```

## 5. Running script.py after entering a document manually.

Windows PowerShell

```
PS D:\github\sde\assignment2\task1> python script.py
2022-03-14T18:39:25.274+0530    connected to: mongodb://localhost:27017/LibraryDB
2022-03-14T18:39:25.301+0530    exported 11 records
*****
Number of records present in MongoDB: 11

Number of records present in Mysql: 10

*****
Records need to be entered in Mysql

Entering the record:

ID                : 622f3e7c290d39eccd449218
ISBN              : 9780538453028
Accession No.     : 210320160010
Title             : C# Programming ( Barbara Doyle)
Author            : ['Barbara Doyle']
Publisher          : Course Techonology
Edition           : 3.0
Year of Publication : 2011.0
Category          : ['C']
Total Number of Pages : 1092.0
Price             : 44413.0

1 Record Inserted sucessfully.
*****
Do you want the complete status of all records...y/n
n
Thank you..

PS D:\github\sde\assignment2\task1>
```

6. Again use insert.py to enter another document.

Windows PowerShell

Thank you..

```
PS D:\github\sde\assignment2\task1> python insert.py
```

```
*****
```

Enter Document Details:

ISBN:

9780130351227

Accession No.:

061020101147

Title:

Database design and development : a visual approach

Author:

Frost,Raymond,Day,John C.,Van Slyke,Craig

Publisher:

Upper Saddle River, N.J. : Pearson Prentice Hall

Edition:

2

Year of Publication:

2006

Category:

dbms

Total Numner of Pages:

534

Price:

1200

622f3f7aca0d6cd23900fac1 Inserted Document Sucessfully

```
*****
```

Checking the Mysql Database for updates...

2022-03-14T18:43:31.726+0530 connected to: mongodb://localhost:27017/LibraryDB

2022-03-14T18:43:31.747+0530 exported 12 records

```
*****
```

Number of records present in MongoDB: 12

Number of records present in Mysql: 11

```
*****
```

Records need to be entered in Mysql

Entering the record:

ID : 622f3f7aca0d6cd23900fac1

ISBN : 9780130351227

Windows PowerShell

```
Year of Publication:
2006
Category:
dbms
Total Numner of Pages:
534
Price:
1200
622f3f7aca0d6cd23900fac1 Inserted Document Sucessfully
*****
Checking the Mysql Database for updates...
2022-03-14T18:43:31.726+0530    connected to: mongodb://localhost:27017/LibraryDB
2022-03-14T18:43:31.747+0530    exported 12 records
*****
Number of records present in MongoDB: 12

Number of records present in Mysql: 11

*****
Records need to be entered in Mysql

Entering the record:

ID                : 622f3f7aca0d6cd23900fac1
ISBN              : 9780130351227
Accession No.     : 061020101147
Title             : Database design and development : a visual approach
Author            : ['Frost', 'Raymond', 'Day', 'John C.', 'Van Slyke', 'Craig']
Publisher         : Upper Saddle River, N.J. : Pearson Prentice Hall
Edition           : 2
Year of Publication : 2006
Category          : ['dbms']
Total Number of Pages : 534
Price             : 1200.0

1 Record Inserted sucessfully.
*****
Do you want the complete status of all records....y/n
n
Thank you..

PS D:\github\sde\assignment2\task1>
```

## 7. Final status of Mysql Database.

```
mysql> select * from books;
```

ID	ISBN	Accession	Title	Publisher	Edition	Year	Total
622f392ac6ce4aa523a5575e	9780619216351	040620180010	Java programming : from problem analysis to program design	Boston : Course Technology/Thomson Learning	1.0	2003.0	998.0
622f3949c6ce4aa523a5575f	059600088X	230820211652	Java programming with Oracle JDBC	Sebastopol, CA : O'Reilly	1.0	2002.0	506.0
622f395ac6ce4aa523a55760	9780136631705	170220210045	Advanced C programming	New York : Brady Pub.	2.0	1992.0	442.0
622f396dc6ce4aa523a55761	9780321356567	191220190102	C++ programming	Berkeley, Calif. : Peachpit Press	1.0	2006.0	532.0
622f3980c6ce4aa523a55762	9781887902991	310120120523	Python programming	Franklin, Beedle	1.0	2004.0	532.0
622f3991c6ce4aa523a55763	9781435455009	030120220023	Python programming for the absolute beginner	Boston, MA : Course Technology Cengage Learning	3.0	2010.0	484.0
622f39a4c6ce4aa523a55764	9781617292453	010120220022	Hello Raspberry Pi! : Python programming for kids and other beginners	Shelter Island, NY : Manning Publications Co.	1.0	2016.0	322.0
622f39b5c6ce4aa523a55765	9780314933096	260820102014	Introduction to data structures and algorithm analysis	St. Paul, MN : West Pub. Co.	2.0	1992.0	714.0
622f39c7c6ce4aa523a55766	9780911537048	270520100056	Reliable data structures in C	Cardiff, N.J. : Plum Hall	1.0	1985.0	276.0
622f39ce81c43a94e977e918	9780471807025	071020100004	Database : step-by-step	New York : Wiley	2	1985	418
622f39e7c7c290d39eccd449218	9780538453028	210320160010	C# Programming ( Barbara Doyle)	Course Techonology	3.0	2011.0	1092.0
622f3f7aca0d6cd23900fac1	9780130351227	061020101147	Database design and development : a visual approach	Upper Saddle River, N.J. : Pearson Prentice Hall	2	2006	534

```
12 rows in set (0.00 sec)
```

```
mysql>
```

### Select MySQL 8.0 Command Line Client

```
mysql> select * from bookauthor;
```

ID	Author_name
622f392ac6ce4aa523a5575e	D. S. Malik
622f3949c6ce4aa523a5575f	Donald Bales
622f395ac6ce4aa523a55760	Oualline
622f395ac6ce4aa523a55760	Steve
622f396dc6ce4aa523a55761	Ullman
622f396dc6ce4aa523a55761	Larry E.(Larry Edward)
622f3980c6ce4aa523a55762	John M. Zelle
622f3991c6ce4aa523a55763	Dawson
622f3991c6ce4aa523a55763	Mike
622f39a4c6ce4aa523a55764	Heitz
622f39a4c6ce4aa523a55764	Ryan
622f39b5c6ce4aa523a55765	Naps
622f39b5c6ce4aa523a55765	Thomas L
622f39b5c6ce4aa523a55765	Singh
622f39b5c6ce4aa523a55765	Bhagat
622f39c7c6ce4aa523a55766	Plum
622f39c7c6ce4aa523a55766	Thomas
622f39ce81c43a94e977e918	Gillenson
622f39ce81c43a94e977e918	Mark L
622f39e7c7c290d39eccd449218	Barbara Doyle
622f3f7aca0d6cd23900fac1	Frost
622f3f7aca0d6cd23900fac1	Raymond
622f3f7aca0d6cd23900fac1	Day
622f3f7aca0d6cd23900fac1	John C.
622f3f7aca0d6cd23900fac1	Van Slyke
622f3f7aca0d6cd23900fac1	Craig

26 rows in set (0.00 sec)

### Select MySQL 8.0 Command Line Client

```
mysql> select * from category;
```

ID	Category
622f392ac6ce4aa523a5575e	java
622f3949c6ce4aa523a5575f	java
622f395ac6ce4aa523a55760	C
622f396dc6ce4aa523a55761	C
622f3980c6ce4aa523a55762	python
622f3991c6ce4aa523a55763	python
622f39a4c6ce4aa523a55764	python
622f39b5c6ce4aa523a55765	python
622f39b5c6ce4aa523a55765	c
622f39c7c6ce4aa523a55766	c
622f39ce81c43a94e977e918	dbms
622f39e7c7c290d39eccd449218	C
622f3f7aca0d6cd23900fac1	dbms

13 rows in set (0.00 sec)

8. If any record is deleted from MongoDB then after running script.py it deletes from Mysql also.

Windows PowerShell

Thank you..

```
PS D:\github\sde\assignment2\task1> python script.py
2022-03-14T18:47:42.722+0530    connected to: mongodb://localhost:27017/LibraryDB
2022-03-14T18:47:42.742+0530    exported 11 records
*****
Number of records present in MongoDB: 11

Number of records present in Mysql: 12

*****
Records need to be deleted from Mysql

Deleting the Record having ID as : 622f3cec81c43a94e977e918

1 Record Deleted sucessfully.
*****
Do you want the complete status of all records...y/n
n
Thank you..

PS D:\github\sde\assignment2\task1> _
```

### Task3:

For this task, we were required to create a **logo/face** detection. For this task, Google Cloud API which is **Cloud Vision API** is used. The complete task can be broken down into some steps which are.

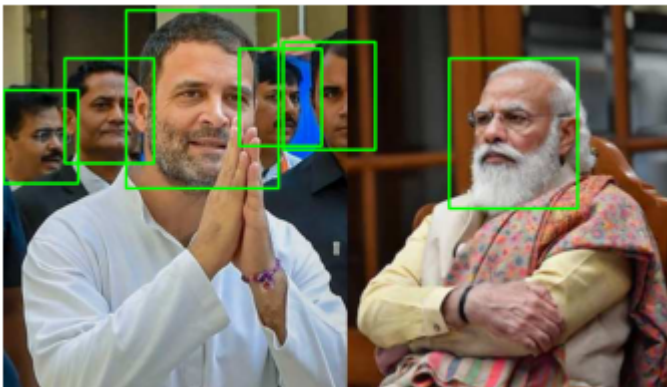
1. Create an Account on the Google Cloud Platform, then use the Cloud Vision API and create credentials in that API and then create a service account then download the key in JSON format which will give you a JSON file as **task3-343004-711575690c36.json** also provided in the zip. After you have the same file you can use it to access the API.
2. Create a python file in a local machine or use google colab. In that file update and import the google cloud libraries and other required libraries and provide the images locally or use a website containing those images.
3. For images, a **Github page** was created which contains the images. Which can be accessed from here [link](#). More images can also be inserted easily on this page.
4. Finally run the python file [link](#) (use IITJ account or use the file provided in compressed folder) and get the output (all the important details about the libraries and functions are provided in the python file).
5. The main functions used for the detection are **label\_detection**, **logo\_detection**, and **face\_detection**.

### Outputs:

1. Logo Detection:



2. Face detection:







## Task2:

For this task, a column store database was required to be implemented and a script was required to check whether the referential integrity was followed or not.

- As we know a column store database stores data column-wise means data of each column is stored in a separate file which helps in the faster execution of queries.
- Data storage is in form of two columns in a table means one for id and the other for value like ID, ISBN, or ISBN, Accession\_No.
- For the task, **Monet Database** was used in which two schemas **LibraryDB** and **Readers** were created and each contains a separate table for each column like one for **Reader ID, Issue date, and Return date** with explicit **Accession No.** in each of them as Id.
- As it was mentioned **Accession No.** was used as the **primary key** in LibraryDB and the same was used to check referential Integrity in the Readers schema.

The task was divided into steps.

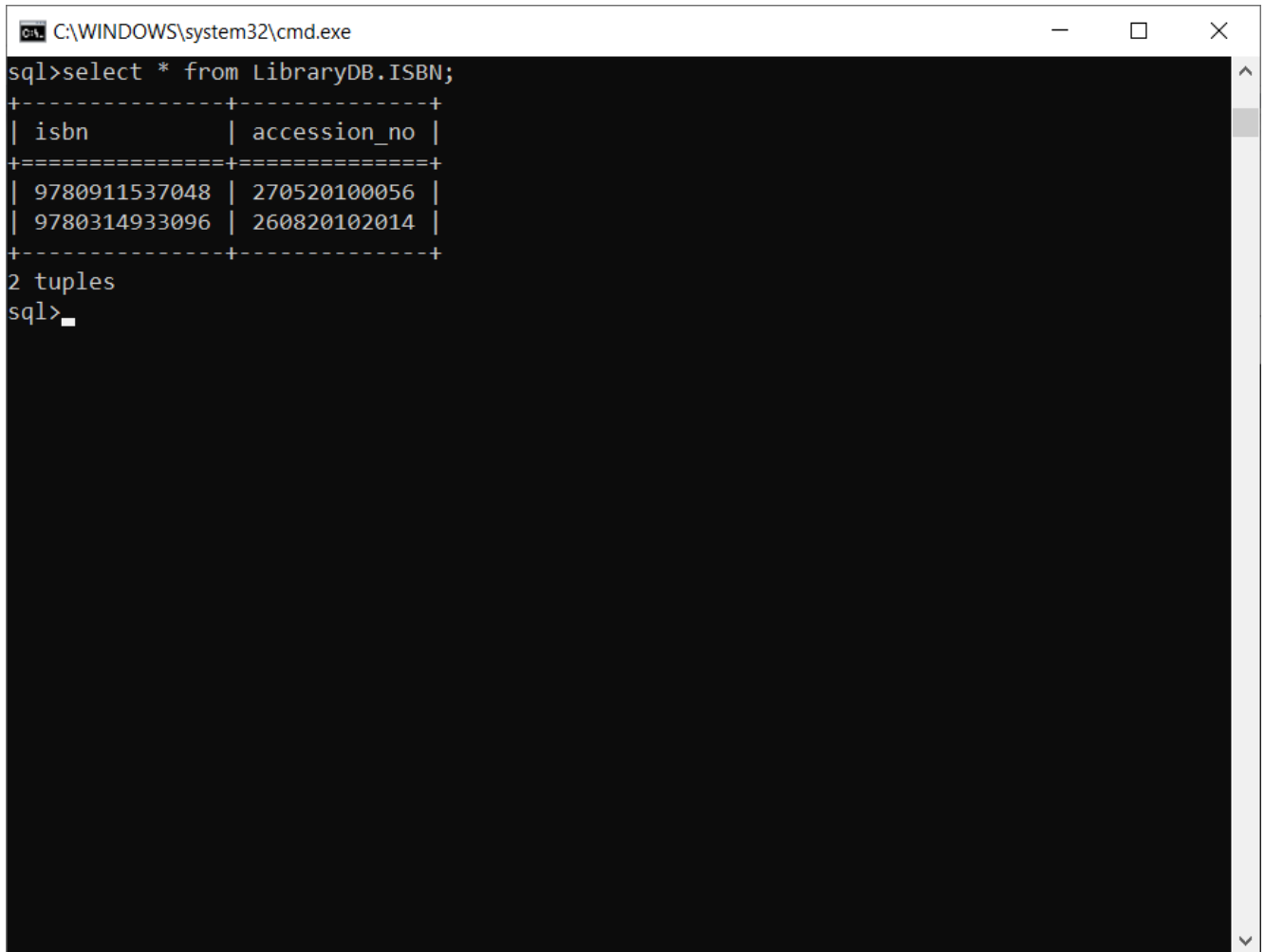
### Step 1: Creating the column store database and insert data.

- Use the [Link](#) to download and install the **Monet Database** server and client.
- First run the server using **Start -> MonetDB SQL Server** then run the client using **Start -> MonetDB SQL Client**.
- Create the schema LibraryDB using the command `CREATE SCHEMA <NAME>;`
- After creating the schema create a table using the command `CREATE TABLE <SCHEMA NAME>.<TABLE NAME> (<COLUMN NAME> <DATATYPE> <CONSTRAINT>,<COLUMN NAME> <DATATYPE> <CONSTRAINT>);`
- All the SQL commands manual for MonetDB can be found at [Link](#).
- For this task, three tables in Readers were created which contain Readers\_Id, Issue\_Date, and Return\_date with AccessionNo in each of them and ReaderID as a Primary Key.  
`CREATE TABLE Readers.Reader_Id( AccessionNo varchar(12), ReaderID int Primary Key);`  
`CREATE TABLE Readers.Issue_Date( AccessionNo varchar(12), IssueDate varchar(10) );`  
`CREATE TABLE Readers.Return_Date( AccessionNo varchar(12), ReturnDate varchar(10) );`
- This was because each record is like storing a value of a column and thus when referential integrity needs to be checked the Id in each can easily identify whether referential integrity is maintained or not.
- After creating the tables data was entered like.  
`insert into Readers.Reader_Id values (270520100056,0001);`  
`insert into Readers.Issue_date values (270520100056,'02/03/2021');`  
`insert into Readers.Return_date values (270520100056,'21/03/2021');`  
`insert into Readers.Reader_Id values (310120120523,0002);`  
`insert into Readers.Issue_date values (310120120523,'03/03/2021');`  
`insert into Readers.Return_date values (310120120523,'12/03/2021');`
- All these commands and information are also present in the Readme file.

As shown below the status of the values in different tables.

```
C:\WINDOWS\system32\cmd.exe
sql>select * from Readers.Reader_Id;
+-----+-----+
| accessionno | readerid |
+=====+=====+
| 270520100056 | 1 |
| 260820102014 | 3 |
| 310120120523 | 2 |
+-----+-----+
3 tuples
sql>select * from Readers.Issue_Date;
+-----+-----+
| accessionno | issuedate |
+=====+=====+
| 270520100056 | 02/03/2021 |
| 260820102014 | 14/04/2021 |
| 310120120523 | 03/03/2021 |
+-----+-----+
3 tuples
sql>select * from Readers.Return_Date;
+-----+-----+
| accessionno | returndate |
+=====+=====+
| 270520100056 | 21/03/2021 |
| 260820102014 | 16/04/2021 |
| 310120120523 | 12/03/2021 |
+-----+-----+
3 tuples
sql>
sql>
sql>
```

- Similarly, the LibraryDB schema was populated with tables and values like shown below.



The screenshot shows a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". Inside the window, a SQL query is executed: `sql>select * from LibraryDB.ISBN;`. The output is displayed in a table format with two columns: `isbn` and `accession_no`. The table contains two rows of data. Below the table, it says "2 tuples". The prompt `sql>` is shown at the bottom.

```
sql>select * from LibraryDB.ISBN;
+-----+-----+
| isbn      | accession_no |
+-----+-----+
| 9780911537048 | 270520100056 |
| 9780314933096 | 260820102014 |
+-----+-----+
2 tuples
sql>
```

- Now because Readers contain a record that contains Accession No. which is not present in LibraryDB it needs to be deleted for maintaining the referential integrity. For this, we use the wrapper script.

## Step 2: Creating the Wrapper script.

- The wrapper script was created in python that uses the connection with MonetDB to check whether referential integrity was maintained or not. If referential integrity is not maintained then records were deleted from Readers Schema that does not follow referential integrity.

Windows PowerShell

```
-----
PS D:\github\sde\assignment2\task2> python script.py
Connecting to MonetDB...
-----
Accession No. Present in Library are:

['270520100056', '260820102014']

Accession No. Present in Readers are:

['270520100056', '260820102014', '310120120523']
-----
Deleting the Record that are neglecting Referential Integrity

Deleting record with Accession No.: 310120120523
-----
After Deletion..
Connecting to MonetDB...
-----
Accession No. Present in Library are:

['270520100056', '260820102014']

Accession No. Present in Readers are:

['270520100056', '260820102014']
-----
Both Contains Same Accession Numbers Referential Integrity Maintained
Thanks
-----
Records Present in the Readers Table are:

Record Number: 1

Reader Id           : 1
Accession No.       : 270520100056
Issue Date          : 02/03/2021
Return Date         : 21/03/2021
-----
```

Accession No. Present in Readers are:

```
['270520100056', '260820102014', '310120120523']
```

-----  
Deleting the Record that are neglecting Referential Integrity

Deleting record with Accession No.: 310120120523

-----  
After Deletion..

Connecting to MonetDB...

-----  
Accession No. Present in Library are:

```
['270520100056', '260820102014']
```

Accession No. Present in Readers are:

```
['270520100056', '260820102014']
```

-----  
Both Contains Same Accession Numbers Referential Integrity Maintained  
Thanks

-----  
Records Present in the Readers Table are:

Record Number: 1

Reader Id	: 1
Accession No.	: 270520100056
Issue Date	: 02/03/2021
Return Date	: 21/03/2021

-----  
Record Number: 2

Reader Id	: 3
Accession No.	: 260820102014
Issue Date	: 14/04/2021
Return Date	: 16/04/2021

-----

- After running the script the values in Reader schema are.

```
C:\WINDOWS\system32\cmd.exe
+-----+-----+
2 tuples
sql>select * from Readers.Reader_Id;
+-----+-----+
| accessionno | readerid |
+=====+=====+
| 270520100056 |      1 |
| 260820102014 |      3 |
+-----+-----+
2 tuples
sql>select * from Readers.Issue_Date;
+-----+-----+
| accessionno | issuedate |
+=====+=====+
| 270520100056 | 02/03/2021 |
| 260820102014 | 14/04/2021 |
+-----+-----+
2 tuples
sql>select * from Readers.Return_Date;
+-----+-----+
| accessionno | returndate |
+=====+=====+
| 270520100056 | 21/03/2021 |
| 260820102014 | 16/04/2021 |
+-----+-----+
2 tuples
sql>
```

Thus referential Integrity is maintained.

- To Insert new records into readers table using the same script **script.py**.

```
pymonetdb.exceptions.OperationalError: 42000!unexpected end of input
```

```
PS D:\github\sde\assignment2\task2> python script.py
```

```
Connecting to MonetDB...
```

```
-----  
Accession No. Present in Library are:
```

```
['270520100056', '260820102014', '100120220022']
```

```
Accession No. Present in Readers are:
```

```
['270520100056', '260820102014']  
-----
```

```
Both Contains Same Accession Numbers Referential Integrity Maintained  
Thanks  
-----
```

```
Records Present in the Readers Table are:
```

```
Record Number: 1
```

```
Reader Id           : 1  
Accession No.       : 270520100056  
Issue Date          : 02/03/2021  
Return Date         : 21/03/2021  
-----
```

```
Record Number: 2
```

```
Reader Id           : 3  
Accession No.       : 260820102014  
Issue Date          : 14/04/2021  
Return Date         : 16/04/2021  
-----
```

```
Do you want to insert records in the table.... y/n
```

```
y  
-----
```

```
Enter Record Details
```

```
Enter Readers Id:
```

```
2
```

```
Enter Issue Date:
```

```
04/05/2021
```



```

-----
Do you want to insert records in the table.... y/n
y
-----
Enter Record Details
Enter Readers Id:
2
Enter Issue Date:
04/05/2021
Enter Return Date:
05/06/2021
Enter Accession No.
100120220022
Inserting Record in the Readers table.
Do you want to insert more records...y/n
n
After inserting the records the status of database is:-
Records Present in the Readers Table are:

Record Number: 1

Reader Id           : 1
Accession No.       : 270520100056
Issue Date          : 02/03/2021
Return Date         : 21/03/2021
-----

Record Number: 2

Reader Id           : 3
Accession No.       : 260820102014
Issue Date          : 14/04/2021
Return Date         : 16/04/2021
-----

Record Number: 3

Reader Id           : 2
Accession No.       : 100120220022
Issue Date          : 04/05/2021
Return Date         : 05/06/2021
-----

```

- This record was entered because the accession No. used in this record was present in the LibraryDB.
- In this way, if any record is manually inserted into MonetDB then using Script.py we can make sure that referential integrity is maintained and the same script is used to insert a new record ensuring the referential integrity.
- Data in LibraryDB needs to be entered manually.