

# Software Data Engineering

Ajit Kumar

M21CS017

## **Task 1: MongoDB and SQL**

**35 Marks**

**Using MongoDB, perform the following:**

- a) Create a database named ‘LibraryDB’ and a collection named ‘Books’, respectively.**
  
- b) Insert at least 10 documents in the newly created collection using the ‘insert()’ command. Each document should contain the following information:**
  - a. ISBN**
  - b. Accession No. (12-digit number, that denotes the date and sequence of acquisition, e.g., 88th book is acquired on 24-Feb-2022, its Accession No. would be 240220220088)**
  - c. Title**
  - d. Author**
  - e. Publisher**
  - f. Edition**
  - g. Year of Publication**
  - h. Category (Permissible values - Java, Python, DBMS, and C)**
  - i. Total Number of Pages**
  - j. Price**
  
- c) Write a script to export this MongoDB data and convert it to SQL. Also, your script should be capable of automatically inserting a record into the SQL table as soon as a new MongoDB document is created. Show the working of your script by inserting two new documents in ‘LibraryDB’.**

## Answer:1.(a)

I have created a database name as ‘LibraryDB’ and a also a collection as ‘Book’ in mongoDB. We can in given screenshot:

```
> show databases;
LibraryDB 0.000GB
LibraryDB 0.000GB
admin 0.000GB
config 0.000GB
local 0.000GB
> use LibraryDB;
switched to db LibraryDB
> show collections
Book
Books
> db.Books.find().pretty()
> db.Book.find().pretty()
{
    "_id" : ObjectId("622ee77166910f7883563794"),
    "ISBN" : "1234567891230",
    "Accession_No" : "240220220088",
    "Title" : "abc",
    "Author" : "xyz0",
    "Publisher" : "pqr0",
    "Edition" : "1",
    "Year_of_publication" : "2000",
    "Category" : "java",
    "Total_no_pages" : "200",
    "Price" : "500"
}
{
    "_id" : ObjectId("622eef866910f7883563796"),
    "ISBN" : "1234567891230",
    "Accession_No" : "240220220088",
    "Title" : "abc",
    "Author" : "xyz0",
    "Publisher" : "pqr0",
    "Edition" : "1",
    "Year_of_publication" : "2000",
    "Category" : "java",
    "Total_no_pages" : "200",
    "Price" : "500"
}
```

## Answer:1.(b)

After creating the database and collection in mongoDB , we have inserted the 10 entry for the document. You can see this in below screenshot:

```
db.createCollection("Book", {validator:{$jsonSchema:{bsonType:"object",required:['ISBN','Accession_No','Title','Author','Publisher','Edition','Year_of_publication','Category','Total_no_pages','Price'],properties:{Category:{enum:['java','python','dbms','c']},description:"Category is not permissible"}}}})
```

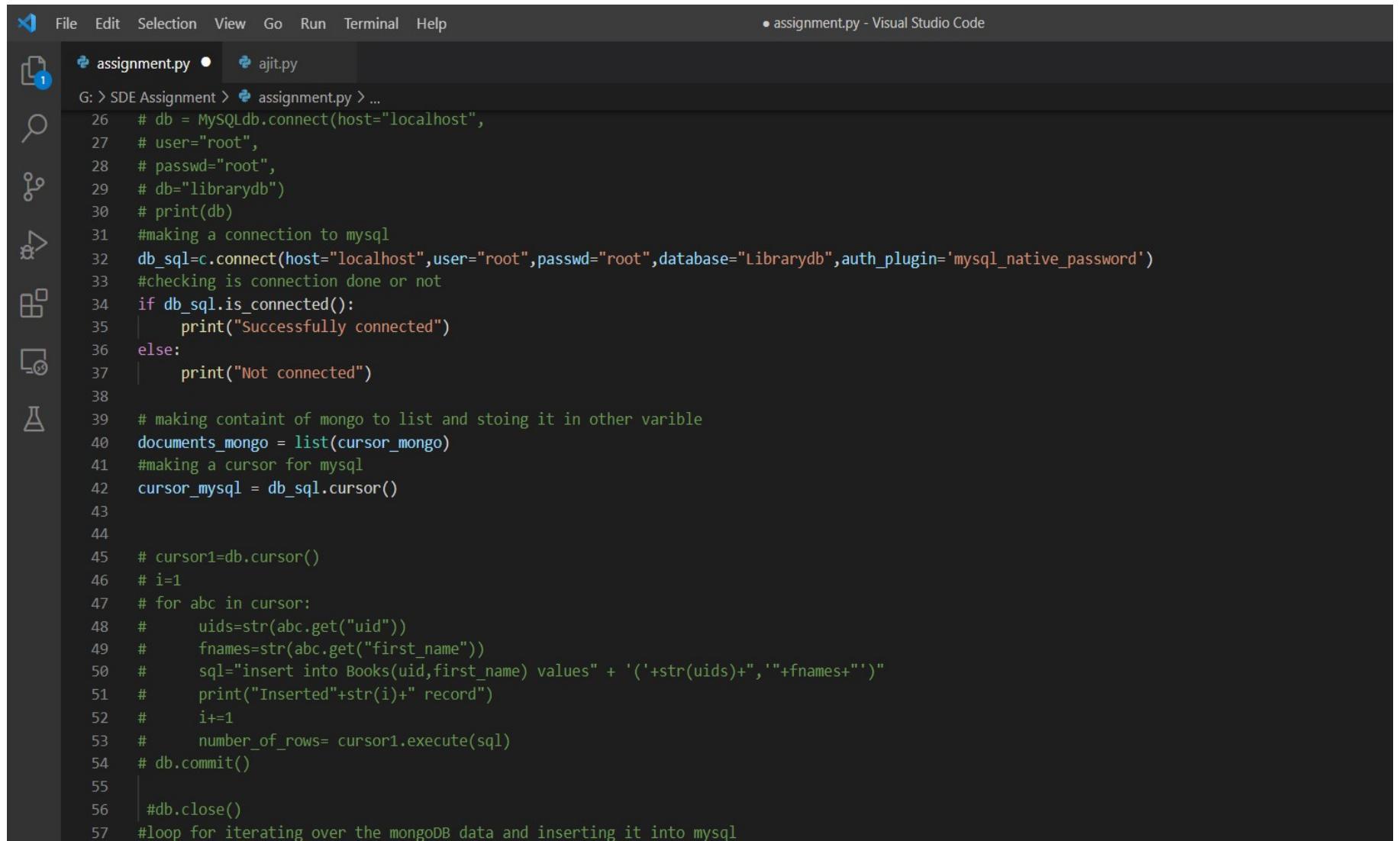
```
> db.Book.find().pretty()
{
  "_id" : ObjectId("622ee77166910f7883563794"),
  "ISBN" : "1234567891230",
  "Accession_No" : "240220220088",
  "Title" : "abc",
  "Author" : "xyz0",
  "Publisher" : "pqr0",
  "Edition" : "1",
  "Year_of_publication" : "2000",
  "Category" : "java",
  "Total_no_pages" : "200",
  "Price" : "500"
}
{
  "_id" : ObjectId("622eef866910f7883563796"),
  "ISBN" : "1234567891230",
  "Accession_No" : "240220220088",
  "Title" : "abc",
  "Author" : "xyz0",
  "Publisher" : "pqr0",
  "Edition" : "1",
  "Year_of_publication" : "2000",
  "Category" : "java",
  "Total_no_pages" : "200",
  "Price" : "500"
}
{
  "_id" : ObjectId("622eef866910f7883563797"),
  "ISBN" : "1234567891231",
```

- Here we are trying to insert not permissible value 'D'( in category field other than java, dbms, c, python ) then is showing some error, we can see in screenshot attached below:

```
EXCERPT / 1.5+
> db.Book.insert({'ISBN':'122222222222','Accession_No':'123','Title':'abc8','Author':'123','Publisher':'pqr1','Edition':'1','Year_of_publication':'2003','Category':'D',
WriteResult({
    "nInserted" : 0,
    "writeError" : {
        "code" : 121,
        "errmsg" : "Document failed validation",
        "errInfo" : {
            "failingDocumentId" : ObjectId("622f3f38448dc048afdd7012"),
            "details" : {
                "operatorName" : "$jsonSchema",
                "schemaRulesNotSatisfied" : [
                    {
                        "operatorName" : "properties",
                        "propertiesNotSatisfied" : [
                            {
                                "propertyName" : "Category",
                                "details" : [
                                    {
                                        "operatorName" : "enum",
                                        "specifiedAs" : {
                                            "enum" : [
                                                "java",
                                                "python",
                                                "dbms",
                                                "c"
                                            ]
                                        },
                                        "reason" : "value was not found in enum",
                                        "consideredValue" : "D"
                                    }
                                ]
                            ]
                        ]
                    }
                ]
            }
        }
    }
})
```

## Answer:1.(c)

To insert the data from mongoDB to mysql. First of all I have made the connection between python and mongoDB and python to mysql. After that I wrote the required script to insert the data .I used the following code that is attached in screenshot:



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help
- Editor Title:** assignment.py - Visual Studio Code
- Editor Content:** The code is written in Python and performs the following steps:
  - Connects to MySQL using `MySQLdb.connect` with host as "localhost", user as "root", password as "root", and database as "librarydb".
  - Prints the database object.
  - Makes a connection to MySQL using `c.connect` with host as "localhost", user as "root", password as "root", database as "Librarydb", and auth plugin as "mysql\_native\_password".
  - Checks if the MySQL connection is successful.
  - If successful, prints "Successfully connected"; otherwise, prints "Not connected".
  - Makes a cursor for MySQL using `db_sql.cursor`.
  - Creates a cursor for MongoDB using `db.cursor`.
  - Initializes a variable `i=1`.
  - Loops through each document in the MongoDB cursor.
  - Extracts the "uid" and "first\_name" fields from each document.
  - Constructs an SQL insert statement: `insert into Books(uid,first_name) values ('+str(uids)+',''+fnames+'')`.
  - Prints the SQL statement with the current value of `i`.
  - Increments `i` by 1.
  - Executes the SQL statement using `cursor1.execute(sql)`.
  - Commits the transaction using `db.commit()`.
  - Closes the MongoDB database using `#db.close()`.
  - Comments out the final line: `#loop for iterating over the mongoDB data and inserting it into mysql`.

- This is following 10 record I have inserted from MongoDB to Mysql using python:

```

mysql> show tables;
+-----+
| Tables_in_librarydb |
+-----+
| book
| books
+-----+
2 rows in set (0.05 sec)

mysql> desc book;
+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| ISBN        | varchar(20) | YES  |     | NULL    |          |
| Accession_No | varchar(20) | YES  |     | NULL    |          |
| Title       | varchar(20) | YES  |     | NULL    |          |
| Author      | varchar(20) | YES  |     | NULL    |          |
| Publisher   | varchar(20) | YES  |     | NULL    |          |
| Edition     | varchar(20) | YES  |     | NULL    |          |
| Year_of_publication | varchar(20) | YES  |     | NULL    |          |
| Category    | varchar(20) | YES  |     | NULL    |          |
| Total_no_pages | varchar(20) | YES  |     | NULL    |          |
| Price       | varchar(20) | YES  |     | NULL    |          |
+-----+-----+-----+-----+-----+
10 rows in set (0.28 sec)

mysql> select * from book;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ISBN        | Accession_No | Title       | Author     | Publisher   | Edition     | Year_of_publication | Category    | Total_no_pages | Price |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1234567891230 | 240220220088 | abc        | xyz0      | pqr0      | 1          | 2000        | java       | 200         | 500  |
| 1234567891230 | 240220220088 | abc        | xyz0      | pqr0      | 1          | 2000        | java       | 200         | 500  |
| 1234567891231 | 240220220089 | abc1      | xyz1      | pqr1      | 1          | 2001        | c          | 201         | 400  |
| 1234567891232 | 240220220090 | abc2      | xyz2      | pqr2      | 1          | 2002        | java       | 205         | 300  |
| 1234567891233 | 240220220091 | abc3      | xyz3      | pqr3      | 1          | 2005        | python    | 207         | 200  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

- To make automatic insertion from MongoDB to mysql I have done it in linux because in window while making chance in configuration file of mongoDB , I was getting some error. We have use the following code to make the automatic insertion from mongoDB to MYSQL.

- In the above code , first I make connection from python to mongoDB and python to MYSQL , and then make a cursor for mongoDB and one for Mysql , then i have use following script:

```
while True:  
    mongo_cursor = db_mongo.Book.watch()  
    mongo_document = next(mongo_cursor)  
    mongo_docs =  
[mongo_document['fullDocument']]  
  
    for data in mongo_docs:  
        cursor_mysql.execute("INSERT INTO Book  
(ISBN, Accession_No, Title, Author, Publisher,  
Edition, Yr, category, Total_no_pages, Price)  
VALUES (%s, %s, %s, %s, %s, %s, %s,  
%s,%s,%s)""",(data['ISBN'],  
data['Accession_No'], data['Title'],  
data['Author'], data['Publisher'],  
data['Edition'], data['Year_of_publication'],  
data['Category'], data['Total_no_pages'],  
data['Price']))  
  
    db_sql.commit()
```

- Then, We ran the python script , following is screenshot:

```
ajit@pop-os:~/Desktop/windowsshare$ python3 ajit.py
pymongo version: 4.0.2

host: Database(MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False, connect=True))
Successfully connected
```

- After the we have inserted data to mongoDB, and in background python script was running and Immediately insertion takes place into the Mysql. Following below is screenshot of result:

```
rs0:PRIMARY> db.Book.insert({'ISBN':'122222222225','Accession_No':'abc8','Author':'123','Publisher':'pqr1','Edition':'1','Yr':'2003','Total_no_pages':'200','Price':'200'});
WriteResult({ "nInserted" : 1 })
rs0:PRIMARY> db.Book.insert({'ISBN':'122222222222','Accession_No':'abc8','Author':'123','Publisher':'pqr1','Edition':'1','Yr':'2003','Total_no_pages':'200','Price':'200'});
```

```
mysql> select * from Book;
+-----+-----+-----+-----+-----+-----+-----+
| ISBN      | Accession_No | Title | Author | Publisher | Edition | Year_of_publication | Category |
+-----+-----+-----+-----+-----+-----+-----+
| 123456789123 | 22022025 | abc1 | pqr1 | xyz1 | 1 | 1998 | c
|
| 122222222225 | 123 | abc8 | 123 | pqr1 | 1 | 2003 | c
```

## Question2:

- a) Column column store database: MariaDB
- Created Database librarydb and created the Wrapper Script.

```
line 1
MariaDB [librarydb]> create database librararybd;
Query OK, 1 row affected (0.006 sec)

MariaDB [librarydb]> CREATE SERVER k FOREIGN DATA WRAPPER mysql OPTIONS(USER 'root',HOST'127.0.0.1',DATABASE 'librararydb');
Query OK, 0 rows affected (0.008 sec)
```

- In Librarydb Created a table name as Books with columns name as mention in question.  
(ISBN,Accession No,Title, Author, Publisher, Edition, Year\_of\_Publication, Category, Number of Pages, Price).

Primary key is ISBN number

```
B [librarydb]> CREATE TABLE `books` (
  `ISBN` VARCHAR(50) NOT NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
  `Accession No` VARCHAR(50) NOT NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
  `Title` VARCHAR(50) NOT NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
  `Author` VARCHAR(50) NOT NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
  `Publisher` VARCHAR(50) NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
  `Edition` VARCHAR(50) NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
  `Year_of_Publication` VARCHAR(50) NOT NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
  `Category` VARCHAR(50) NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
  `Total_Number_of_Pages` VARCHAR(50) NOT NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
  `Price` VARCHAR(50) NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
  PRIMARY KEY (`ISBN`) USING BTREE,
  INDEX `Accession No.`(`Accession No`) USING BTREE
)
COLLATE='utf8mb4_general_ci'
ENGINE=InnoDB
;
```

- In Librarydb created second table reader with columns names mention in question,  
(reader id,issue\_date, Accession No,Return\_date).

Primary key is reader id and foreign key is Accession No

```
+-----+-----+-----+-----+
10 rows in set (0.020 sec)

MariaDB [librarydb]> CREATE TABLE `reader` (
-> `reader id` VARCHAR(50) NOT NULL DEFAULT 'AUTO_INCREMENT' COLLATE 'utf8mb4_general_ci',
-> `issue_date` VARCHAR(50) NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
-> `Accession No` VARCHAR(50) NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
-> `Return_date` VARCHAR(50) NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
-> PRIMARY KEY (`reader id`) USING BTREE,
-> UNIQUE INDEX `Accession No.`(`Accession No`) USING BTREE,
-> CONSTRAINT `FK_reader_books` FOREIGN KEY (`Accession No`) REFERENCES `librarydb`.`books` (`Accession No`) ON UPDATE CASCADE
-> )
-> COLLATE='utf8mb4_general_ci'
-> ENGINE=InnoDB
-> ;
```

- For referential integrity chosen condition is **ON UPDATE CASCADE ON DELETE CASCADE**.

Book and reader tables are:

```
mariaDB [librarydb]> describe reader;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| reader_id | varchar(50) | NO | PRI | AUTO_INCREMENT | |
| issue_date | varchar(50) | NO | | | |
| Accession_No | varchar(50) | NO | UNI | | |
| Return_date | varchar(50) | NO | | | |
+-----+-----+-----+-----+-----+-----+
1 rows in set (0.022 sec)

mariaDB [librarydb]> describe books;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ISBN | varchar(50) | NO | PRI | | |
| Accession_No | varchar(50) | NO | MUL | | |
| Title | varchar(50) | NO | | | |
| Author | varchar(50) | NO | | | |
| Publisher | varchar(50) | NO | | | |
| Edition | varchar(50) | NO | | | |
| Year_of_Publication | varchar(50) | NO | | | |
| Category | varchar(50) | NO | | | |
| Total_Number_of_Pages | varchar(50) | NO | | | |
| Price | varchar(50) | NO | | | |
+-----+-----+-----+-----+-----+-----+
0 rows in set (0.020 sec)
```

- Data in Book table as:

ISBN	Accession No	Title	Author	Publisher	Edition	Year_of_Publication	Category	Total_Number_of_Pages	Price	
1234567891221	240220220076	abc10	xyz10	pqr10	1	2022	dbms	786	500	
1234567891223	240220220012	abc11	xyz0	pqr0	1	2022	java	370	400	
1234567891224	240220220021	abc12	xyz1	pqr0	1	2020	dbms	350	300	
1234567891226	240220220065	abc13	xyz2	pqr0	1	2000	java	270	100	
1234567891230	240220220088	abc	xyz0	pqr0	1	2000	java	100	200	
1234567891231	240220220089	abc1	xyz1	pqr1	1	2000	java	200	1400	
1234567891232	240220220090	abc2	xyz2	pqr2	1	2001	c	300	1300	
1234567891233	240220220091	abc3	xyz3	pqr3	1	2002	java	400	1200	
1234567891235	240220220093	abc5	xyz5	pqr5	1	1998	java	600	1000	
1234567891237	240220220095	abc7	xyz7	pqr7	1	2001	dbms	800	800	
1234567891238	240220220096	abc8	xyz8	pqr8	1	2002	dbms	900	700	
1234567891239	240220220097	abc9	xyz9	pqr9	2	2003	java	800	600	

- Data in reader table as:

reader id	issue_date	Accession No	Return_date
1	12/12/2012	240220220012	12/12/2013
3	14/12/2012	240220220095	14/12/2013
5	16/12/2012	240220220088	16/12/2013
6	17/12/2012	240220220097	17/12/2013
7	06/01/2019	240220220076	09/02/2019
8	08/01/2019	240220220021	09/02/2019
9	08/01/2019	240220220096	09/02/2019

- **ON UPDATE CASCADE ON DELETE CASCADE is satisfied as:**

If deleting the row in books table corresponding row is also deleted in reader row means satisfying the referential integrity.

ISBN	Accession No	Title	Author	Publisher	Edition	Year_of_Publication	Category	Total_Number_of_Pages	Price
1234567891223	240220220012	abc11	xyz0	pqr0	1	2022	java	370	400
1234567891224	240220220021	abc12	xyz1	pqr0	1	2020	dbms	350	300
1234567891226	240220220065	abc13	xyz2	pqr0	1	2000	java	270	100
1234567891230	240220220088	abc	xyz0	pqr0	1	2000	java	100	200
1234567891231	240220220089	abc1	xyz1	pqr1	1	2000	java	200	1400
1234567891232	240220220090	Testing: Confirm							
1234567891233	240220220091	Delete 1 row(s)?							
1234567891235	240220220093	<input type="button" value="OK"/> <input type="button" value="Cancel"/>							
1234567891237	240220220095								
1234567891238	240220220096								
1234567891239	240220220097	abc9	xyz9	pqr9	2	2003	java	800	600

- Corresponding row deleted in reader table:

reader id	issue_date	Accession No	Return_date
1	12/12/2012	240220220012	12/12/2013
3	14/12/2012	240220220095	14/12/2013
5	16/12/2012	240220220088	16/12/2013
6	17/12/2012	240220220097	17/12/2013
8	08/01/2019	240220220021	09/02/2019

- Java Connectivity is as :

```

public class Main {
    public static void main(String[] args) throws SQLException {
        Connection connection=null;
        String url="jdbc:mariadb://localhost:3306/librarydb";
        String user= "root";
        String pwd="root";
        Wrapper v=null;
        try {
            connection=DriverManager.getConnection(url, user, pwd);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        System.out.println("Successful Connected");
    }
}

```

- B) In a one-to-one mapping, an entity in the parent table is linked to only one entity in the child table, and an entity in the child table is linked to only one object in the parent table. So in this task for one to one mapping I chosen accession no in reader table (child table) as unique not null and foreign key as well. And Books table accession no is primary key of books table(Parent table).

```

MariaDB [librarydb]> CREATE TABLE `books` (
    -> `ISBN` VARCHAR(50) NOT NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
    -> `Accession No` VARCHAR(50) NOT NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
    -> `Title` VARCHAR(50) NOT NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
    -> `Author` VARCHAR(50) NOT NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
    -> `Publisher` VARCHAR(50) NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
    -> `Edition` VARCHAR(50) NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
    -> `Year_of_Publication` VARCHAR(50) NOT NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
    -> `Category` VARCHAR(50) NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
    -> `Total_Number_of_Pages` VARCHAR(50) NOT NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
    -> `Price` VARCHAR(50) NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
    -> PRIMARY KEY (`Accession No`) USING BTREE
    -> )_

```

```

MariaDB [librarydb]> CREATE TABLE `reader` (
-> `reader id` VARCHAR(50) NOT NULL DEFAULT 'AUTO_INCREMENT' COLLATE 'utf8mb4_general_ci',
-> `issue_date` VARCHAR(50) NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
-> `Accession No` VARCHAR(50) NOT NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
-> `Return_date` VARCHAR(50) NULL DEFAULT '0' COLLATE 'utf8mb4_general_ci',
-> PRIMARY KEY (`reader id`),
-> UNIQUE INDEX `Accession No`(`Accession No`),
-> CONSTRAINT `FK_reader_books` FOREIGN KEY (`Accession No`) REFERENCES `books`(`Accession No`) ON UPDATE CASCADE ON DELETE CASCADE
-> )
-> COLLATE='utf8mb4_general_ci'
-> ENGINE=InnoDB
-> ;

```

- One to one mapping tables are as:

ISBN	Accession No	Title	Author	Publisher	Edition	Year_of_Publication	Category	Total_Number_of_Pages	Price
1234567891223	240220220012	abc11	xyz0	pqr0	1	2022	java	370	400
1234567891224	240220220021	abc12	xyz1	pqr0	1	2020	dbms	350	300
1234567891226	240220220065	abc13	xyz2	pqr0	1	2000	java	270	100
1234567891230	240220220088	abc	xyz0	pqr0	1	2000	java	100	200
1234567891231	240220220089	abc1	xyz1	pqr1	1	2000	java	200	1400
1234567891232	240220220090	abc2	xyz2	pqr2	1	2001	c	300	1300
1234567891233	240220220091	abc3	xyz3	pqr3	1	2002	java	400	1200
1234567891235	240220220093	abc5	xyz5	pqr5	1	1998	java	600	1000
1234567891237	240220220095	abc7	xyz7	pqr7	1	2001	dbms	800	800
1234567891230	240220220097	abc0	xyz0	pqr0	0	2003	java	800	600

### Task 3: API

Marks 30

Write an API using any framework or programming language to perform face/logo detection.

Test your API against these images.



<https://console.cloud.google.com/home/dashboard?project=helpful-data-342913>

- From this link , we have enable the vision API then we created a service account there and generate the key for this and downloaded the Json file. Following is screenshot for the proof:

The screenshot shows the Google Cloud Platform interface for managing service accounts. The left sidebar is titled 'IAM & Admin' and includes options like IAM, Identity & Organization, Policy Troubleshooter, Policy Analyzer, Organization Policies, Service Accounts (which is selected), Workload Identity Federat..., Labels, Tags, Settings, Manage Resources, and Release Notes. The main content area is titled 'ProjectAPI' under 'Service Accounts'. It has tabs for DETAILS, PERMISSIONS, KEYS (which is selected), METRICS, and LOGS. The KEYS section contains a warning about service account keys being a security risk if compromised, suggesting Workload Identity Federation instead. Below the warning, it says 'Add a new key pair or upload a public key certificate from an existing key pair.' and provides links to organization policies and service account settings. A button labeled 'ADD KEY' is present. A table lists one key entry: Type (Service Account Key), Status (Active), Key (200a20202c91b0f28cf4899456c70f6fc336ff34), Key creation date (Mar 10, 2022), and Key expiration date (Jan 1, 10000). A trash can icon is next to the key entry.

Type	Status	Key	Key creation date	Key expiration date
Service Account Key	Active	200a20202c91b0f28cf4899456c70f6fc336ff34	Mar 10, 2022	Jan 1, 10000

- This is json file we have used:

```
{  
  "type": "service_account",  
  "project_id": "helpful-data-342913",  
  "private_key_id": "200a20202c91b0f28cf4899456c70f6fc336ff34",  
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwgSjAgEAAoIBAQD0paA9tHGMkT6M\\nq/sYFHnQXQAVZL+xMwIzRkZMhWvKwBYJshR  
  "client_email": "projectapi@helpful-data-342913.iam.gserviceaccount.com",  
  "client_id": "111324305828169911046",  
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",  
  "token_uri": "https://oauth2.googleapis.com/token",  
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",  
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/projectapi%40helpful-data-342913.iam.gserviceaccount.com"  
}
```

- After that downloaded all the python library for google vision API and for this we have used following command :

**pip install google-cloud-vision**

- Then we had written the python script as Facedetection.py to detect the logo and for detecting the logo we had written the one script as draw\_vertice.py for drawing line around the logo.
- This is screenshot of code for draw\_vertices.py:

```
#import io
import io
#import Image,ImageDraw,ImageFont
from PIL import Image, ImageDraw, ImageFont
#function for draw vertices
def drawVertices(image_source, vertices, display_text=''):
    # variable for read the object
    pillow_img = Image.open(io.BytesIO(image_source))
    #calling the draw function and storing it into draw variable
    draw = ImageDraw.Draw(pillow_img)
    #iterating over the vertices
    for i in range(len(vertices) - 1):
        #calling to draw line function
        draw.line(((vertices[i].x, vertices[i].y), (vertices[i + 1].x, vertices[i + 1].y)),
                  #drawing green line
                  fill='green',
                  #width of line is 8
                  width=8
        )
    # calling to draw line function
    draw.line(((vertices[len(vertices) - 1].x, vertices[len(vertices) - 1].y),
               #making a co-ordinate
               (vertices[0].x, vertices[0].y)),
              #filling the green line
              fill='green',
              #width 8
              width=8
    )
    #making a object front
    font = ImageFont.truetype('arial.ttf', 16)
    #calling to draw next function
    draw.text((vertices[0].x + 10, vertices[0].y),
              font=font, text=display_text,
              fill=(255, 255, 255))
    #calling to show function
    pillow_img.show()
```

- This is following code for LogoDetection.py for following image:



```
import pandas as pd
# importing draw vertices
from draw_vertices import drawVertices
# set the environment variable 'GOOGLE_APPLICATION_CREDENTIALS' to the file paths of the json file that contain service account key
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = r'helpful-data-342913-200a20202c91.json'
#creating the client instance for vision.ImageAnnotatorClient
client = vision.ImageAnnotatorClient()
# reading the file from file_name
file_name = 'img_01.png'
#reading the image
image_folder = './images/'
#variable to store image_path
image_path = os.path.join(image_folder, file_name)
#opening image from image path and rename as image_file
with io.open(image_path, 'rb') as image_file:
    #reading the image file and storing it into content variable
    content = image_file.read()
#creating a image object
image = types.Image(content=content)
#creating a object for logo_detection method
response = client.logo_detection(image=image)
#creating a object for logo_annotations
logos = response.logo_annotations
# iterating over the logos
for logo in logos:
    #printing the logo description
    print('Logo Description:', logo.description)
    #printing the confidence score
    print('Confidence Score:', logo.score)
    #printinf the *50
    print('*'*50)
    # print the boundary for logo with the help of draw vertices
    vertices = logo.bounding_poly.vertices
    #printing the vertices value
    print('Vertices Values {0}'.format(vertices))
    #calling to draw_vertices function
    drawVertices(content, vertices, logo.description)
```

- This is following result we get and we can see that Amazon logo is detected by green line :

```
[Running] python -u "g:\SDE Assignment\VisionAPI\api.py"
Logo Description: Amazon
Confidence Score: 0.8373565673828125
-----
Vertices Values [x: 15
y: 35
, x: 650
y: 35
, x: 650
y: 473
, x: 15
y: 473
]
'ping' is not recognized as an internal or external command,
operable program or batch file.

[Done] exited with code=0 in 3.514 seconds
```



- This is screenshot for python script for Facedetection.py for following image:



```

file_name = 'img_03.jpg'
#storing the image path in variable
image_path = f'.\Images\{file_name}'
#opening image from image path and rename as image_file
with io.open(image_path, 'rb') as image_file:
    #reading the image file and storing it into content variable
    content = image_file.read()
#createing a image object
image = types.Image(content=content)
#createing a object for logo_detection method
response = client.face_detection(image=image)
#calling to face_detection function and stroing the returning value into variable
faceAnnotations = response.face_annotations
# making a touple for result
likelihood = ('Unknown', 'Very Unlikely', 'Unlikely', 'Possibly', 'Likely', 'Very Likely')
#printing faces
print('Faces:')
#iterating over FaceAnnotations
for face in faceAnnotations:
    #printing the confidence
    print('Detection Confidence {0}'.format(face.detection_confidence))
    #printing the angryness
    print('Angry likelyhood: {0}'.format(likelihood[face.anger_likelihood]))
    # printing the joy value
    print('Joy likelyhood: {0}'.format(likelihood[face.joy_likelihood]))
    # printing the sorrow value of face Expression
    print('Sorrow likelyhood: {0}'.format(likelihood[face.sorrow_likelihood]))
    #printing the surprised value
    print('Surprised ikelihood: {0}'.format(likelihood[face.surprise_likelihood]))
    #printing the Headwear value
    print('Headwear likelyhood: {0}'.format(likelihood[face.headwear_likelihood]))
    #printing the face_vertices value
    face_vertices = ['({0},{1})'.format(vertex.x, vertex.y) for vertex in face.bounding_poly.vertices]
    #printing the face_bound value
    print('Face bound: {0}'.format(', '.join(face_vertices)))
    #printing the space
    print('')

```

- This is following result :

```
[Running] python -u "g:\SDE Assignment\VisionAPI\Facedetection.py"
Faces:
Detection Confidence 0.5008718967437744
Angry likelihood: Very Unlikely
Joy likelihood: Very Likely
Sorrow likelihood: Very Unlikely
Surprised likelihood: Very Unlikely
Headwear likelihood: Very Likely
Face bound: (173,12), (464,12), (464,351), (173,351)

Detection Confidence 0.9508340358734131
Angry likelihood: Very Unlikely
Joy likelihood: Likely
Sorrow likelihood: Very Unlikely
Surprised likelihood: Very Unlikely
Headwear likelihood: Likely
Face bound: (727,48), (996,48), (996,362), (727,362)

[Done] exited with code=0 in 4.484 seconds
```

- This is screenshot for python script for Facedetection.py for following image:

:



```
file_name = 'img_02.jpg'
#storing the image path in variable
image_path = f'.\Images\{file_name}'
#opening image from image path and rename as image_file
with io.open(image_path, 'rb') as image_file:
    #reading the image file and storing it into content variable
    content = image_file.read()
#createing a image object
image = types.Image(content=content)
#creating a object for logo_detection method
response = client.face_detection(image=image)
#calling to face_detection function and stroing the returning value into variable
faceAnnotations = response.face_annotations
# making a tuple for result
likehood = ('Unknown', 'Very Unlikely', 'Unlikely', 'Possibly', 'Likely', 'Very Likely')
#printing faces
print('Faces:')
#iterating over FaceAnnotations
for face in faceAnnotations:
    #printing the confidence
    print('Detection Confidence {0}'.format(face.detection_confidence))
    #printing the angryness
    print('Angry likelyhood: {0}'.format(likehood[face.anger_likelihood]))
    # printing the joy value
    print('Joy likelyhood: {0}'.format(likehood[face.joy_likelihood]))
    # printing the sorrow value of face Expression
    print('Sorrow likelyhood: {0}'.format(likehood[face.sorrow_likelihood]))
    #printing the surprised value
    print('Surprised ikelihood: {0}'.format(likehood[face.surprise_likelihood]))
    #printing the Headwear value
    print('Headwear likelyhood: {0}'.format(likehood[face.headwear_likelihood]))
    #printing the face_vertices value
    face_vertices = ['({0},{1})'.format(vertex.x, vertex.y) for vertex in face.bounding_poly.vertices]
    #printing the face_bound value
    print('Face bound: {0}'.format(', '.join(face_vertices)))
    #printing the space
    print('')
```

- This is following result :

```
[Running] python -u "g:\SDE Assignment\visionAPI\Facedetection.py"
Faces:
Detection Confidence 0.9451922178268433
Angry likelihood: Very Unlikely
Joy likelihood: Very Likely
Sorrow likelihood: Very Unlikely
Surprised likelihood: Very Unlikely
Headwear likelihood: Very Unlikely
Face bound: (128,5), (287,5), (287,190), (128,190)

Detection Confidence 0.5749183297157288
Angry likelihood: Very Unlikely
Joy likelihood: Unlikely
Sorrow likelihood: Very Unlikely
Surprised likelihood: Very Unlikely
Headwear likelihood: Very Unlikely
Face bound: (64,55), (157,55), (157,164), (64,164)

Detection Confidence 0.7792016267776489
Angry likelihood: Very Unlikely
Joy likelihood: Very Unlikely
Sorrow likelihood: Very Unlikely
Surprised likelihood: Very Unlikely
Headwear likelihood: Unlikely
Face bound: (246,45), (332,45), (332,146), (246,146)

Detection Confidence 0.8975594639778137
Angry likelihood: Very Unlikely
Joy likelihood: Very Unlikely
Sorrow likelihood: Very Unlikely
Surprised likelihood: Very Unlikely
Headwear likelihood: Very Unlikely
Face bound: (0,88), (78,88), (78,185), (0,185)
```

```
Detection Confidence 0.9361570477485657
Angry likelihood: Very Unlikely
Joy likelihood: Very Unlikely
Sorrow likelihood: Very Unlikely
Surprised likelihood: Very Unlikely
Headwear likelihood: Very Unlikely
Face bound: (465,55), (599,55), (599,211), (465,211)
```

```
Detection Confidence 0.6923196315765381
Angry likelihood: Very Unlikely
Joy likelihood: Very Unlikely
Sorrow likelihood: Very Unlikely
Surprised likelihood: Very Unlikely
Headwear likelihood: Very Unlikely
Face bound: (292,38), (388,38), (388,150), (292,150)
```