--TABLES CREATION

Create table Authors (

author\_id INT NOT NULL Primary Key ,

author\_name varchar(255) NOT NULL

);

Create table Books (

book\_id INT NOT NULL Primary Key,

title varchar(255) NOT NULL,

author\_id INT,

genre varchar(50),

publication\_year INT,

isbn varchar(13),

FOREIGN KEY(author\_id)REFERENCES Authors(author\_id)

);

--Alter isbn table to taking more values. - count as values too

ALTER TABLE books

ALTER COLUMN isbn VARCHAR(50);

Create table Borrowers (

borrower\_id INT Primary Key,

name varchar(255) NOT NULL,

email varchar(255) NOT NULL,

borrowed\_books TEXT

);

-------------------------------------Modifying-------------------------------------------------------------------------------

--adding column for foreign key

--Altering the table, from having borrowed books to having the book\_id rather

ALTER TABLE Borrowers

ADD book\_id INT;

-- Step 2: Add a foreign key constraint

ALTER TABLE Borrowers

ADD CONSTRAINT FK\_Borrowers\_Books FOREIGN KEY (book\_id) REFERENCES Books(book\_id);

--Removing the old column

Alter table Borrowers

Drop Column borrowed\_books;

---------------------------------------------------------------------------------------------------------------------------------

Create table Genres (

genre\_id INT Primary Key,

genre\_name varchar(50) NOT NULL,

);

Create table BookGenres(

book\_id INT,

genre\_id INT,

Primary key(book\_id,genre\_id),

Foreign key(book\_id)REFERENCES Books(book\_id),

Foreign key(genre\_id)REFERENCES Genres(genre\_id)

);

--Book Review System Table----

CREATE TABLE Reviews(

review\_id INT Primary key,

book\_id INT,

review\_name VARCHAR(255) NOT NULL,

rating INT,

review\_text TEXT,

FOREIGN KEY (book\_id)REFERENCES Books(book\_id)

);

--Insert data for authors

Insert into Authors(author\_id, author\_name) values

(1, 'F. Scott Fitzgerald'),

(2, 'Harper Lee'),

(3, 'George Orwell'),

(4, 'J.D. Salinger'),

(5, 'Gabriel Garcia Marquez'),

(6, 'Aldous Huxley'),

(7, 'J.R.R. Tolkien'),

(8, 'Jane Austen'),

(9, 'J.K. Rowling');

INSERT INTO Books (book\_id, title, author\_id, genre, publication\_year, isbn)

VALUES

(11, 'The Handmaid''s Tale', 1, 'Dystopian', 1985, '978-0-7710-0813-2');

--insert data for Books

Insert into Books(book\_id,title,author\_id,genre,publication\_year,isbn) values

(1, 'The Great Gatsby', 1, 'Fiction', 1925, '978-0-7432-7356-5'),

(2, 'To Kill a Mockingbird', 2, 'Fiction', 1960, '978-0-06-112008-4'),

(3, '1984', 3, 'Dystopian', 1949, '978-0-452-28423-4'),

(4, 'The Catcher in the Rye', 4, 'Fiction', 1951, '978-0-316-76948-0'),

(5, 'One Hundred Years of Solitude', 5, 'Magical Realism', 1967, '978-0-06-112008-4'),

(6, 'Brave New World', 6, 'Dystopian', 1932, '978-0-06-085052-4'),

(7, 'The Hobbit', 7, 'Fantasy', 1937, '978-0-618-24170-9'),

(8, 'The Lord of the Rings', 7, 'Fantasy', 1954, '978-0-395-08254-4'),

(9, 'Pride and Prejudice', 8, 'Romance', 1813, '978-1-85326-000-4'),

(10, 'Harry Potter and the Sorcerer\s Stone', 9, 'Fantasy', 1997, '978-0-590-35340-3');

--insert data fror borrowers

insert into Borrowers( borrower\_id,name,email,book\_id) values

(1, 'John Max', 'john.m@gmail.com', 1),

(2, 'Jane Smith', 'jane.smith@gmail.com', 4),

(3, 'Bob Johnson', 'bob.johnson@mweb.com', 9),

(4, 'Alice Brown', 'alice.brown@yahoo.com', 10),

(5, 'Charlie White', 'charlie.white@gmail.com', 5),

(6, 'Eva Green', 'eva.green@yahoo.com', 8),

(7, 'David Lee', 'david.lee@gmail.com', 1),

(8, 'Grace Taylor', 'grace.taylor@yahoo.com',3),

(9, 'Frank Miller', 'frank.miller@outlook.com', 7),

(10, 'Sophie Davis', 'sophie.davis@mweb.com',10);

--Insert into book review---

INSERT INTO Reviews(review\_id,book\_id,review\_name,rating,review\_text) VALUES

(01,1,'Charlie White',4,'Enjoyed the book.'),

(02,2,'John Max',5,'Highly recomemended'),

(03,3,'Charlie White',2,'Expected more'),

(04,4,'Sophie Davis',5,'Enjoyed the book so much, defs reccomend it'),

(05,5,'John Max',1,'Book could have been better'),

(06,6,'Grace Taylor',5,'Highly recommend the book!'),

(07,7,'Alice Brown',5,'What a book!Definetly takes you into a different world.Loved it'),

(08,10,'Bob Johnson',1,'Horrible book. Dont recommend'),

(09,9,'Bob Johnson',5,'Highly recommend this book!'),

(010,8,'John Max',4,'Expected a lot more from this book.');

--Retrieve name of a books along with their authors

Select distinct b.title as'Books', a.author\_name as 'Author'

from Books b

Join Authors a on b.author\_id =a.author\_id

;

--Retrive borrowers along with the books they borrowed

Select b.name, b.email,bk.title as 'Book'

from Borrowers b

join Books bk on bk.book\_id = b.book\_id

--Retrive the number of books written by authors.

Select a.author\_name, count(bk.book\_id) as 'Total Books'

from Authors a

left join Books bk on a.author\_id = bk.author\_id

Group by a.author\_name

Order by [Total Books] desc;

--Update borrowers email address

Update Borrowers

SET email = 'bob.johnson54@mweb.gov.za'

where borrower\_id=3

--Querying to retieve average rating for each book

Select b.title as 'Books', AVG(r.rating) as 'Average Rating'

from Books b

LEFT JOIN Reviews r On b.book\_id = r.book\_id

Group by b.title

--Here we will find books with the higest average rating---

WITH AvgRatings AS (

SELECT

b.book\_id,

b.title AS 'Books',

AVG(r.rating) AS 'Average Rating'

FROM

Books b

LEFT JOIN Reviews r ON b.book\_id = r.book\_id

GROUP BY

b.book\_id, b.title

)

SELECT

ar.\*,

r.rating,

r.review\_text

FROM

AvgRatings ar

JOIN Reviews r ON ar.book\_id = r.book\_id AND ar.[Average Rating] = r.rating;