Ahsanullah University of Science and Technology (AUST)
Department of Computer Science and Engineering

Offline 1

Course No.: CSE4126

Course Title: Distributed Database Systems Lab

Date of Submission-
21-05-2023

Submitted To-
Ms. Zarin Tasnim Shejuti & Sanzana Karim Lora

Submitted By-
Name: MD Shihabul Islam Shovo
ID: 190204075
Section: B1
Year- 4th
Semester-1st
Department-CSE

Question: Consider the following database schema that is maintained at a university library:
Student(studentID: integer, name: string, phone: string, age: integer)

Borrows(studentID: integer, bookID: integer, dateBorrowed: date)

Book(bookID: integer, authorID: integer, title: string, genre: string)

Author(authorID: integer, name: string, age: integer)

Tasks: 1. Create the four tables and insert necessary data in the tables using SQL commands (necessary data can be within 5 to 10 rows).

Tasks: 2. Write the following queries in SQL –

a. Show the name of the students who borrowed the book titled “Gulliver’s Travels”.

b. Show the age of the oldest author from among those who have published books belonging to the genre “Non-Fiction”.

c. Show the phone of the student who borrowed the book titled “Gitanjali” more than twice.

Solution:

```
clear screen;


drop table borrows;

drop table student;

drop table book;

drop table author;


–Task 1:

create table student(studentID number primary key, name varchar2(20), phone varchar2(20), age number);

insert into student values (1, 'Shihabul', '01701892749', 20);

insert into student values (2, 'Aranyak', '01809847210', 22);

insert into student values (3, 'Bulbul', '01968723937', 19);

insert into student values (4, 'Abir', '01712345345', 23);

insert into student values (5, 'Sajjad', '01943268557', 21);

insert into student values (6, 'Johir', '01385673862', 19);
```

```
create table author (authorID number primary key, name varchar2(20), age number);

insert into author values (101, 'Stieg Larsson', 45);

insert into author values (102, 'F. Scott Fitzgerald', 40);

insert into author values (103, 'J.K. Rowling', 55);

insert into author values (104, 'Jonathan Swift', 50);

insert into author values (105, 'Jane Austen', 41);

insert into author values (106, 'J.R.R. Tolkien', 60);

insert into author values (107, 'Rabindranath Tagore', 80);


create table book (bookID number primary key, authorID number, title varchar2(50), genre varchar2(20), foreign key (authorID) references author(authorID));

insert into book values (301, 102, 'The Great Gatsby', 'Fiction');

insert into book values (302, 101, 'The Girl with the Dragon Tattoo', 'Non-Ficton');

insert into book values (303, 103, 'Harry Potter and the Philosopher''s Stone', 'Fantasy');

insert into book values (304, 104, 'Gulliver''s Travels', 'Adventure');

insert into book values (305, 105, 'Pride and Prejudice', 'Non-Ficton');

insert into book values (306, 106, 'The Hobbit', 'Adventure');

insert into book values (307, 107, 'Gitanjali', 'Poetry');


create table borrows(studentID number, bookID number, dateBorrowed date, foreign key (studentID) references student(studentID), foreign key (bookID) references book(bookID));

insert into borrows values (6, 301, TO_DATE('02 March 2023','DD Month YYYY'));

insert into borrows values (1, 303, TO_DATE('04 April 2023','DD Month YYYY'));

insert into borrows values (2, 304, TO_DATE('05 May 2023','DD Month YYYY'));

insert into borrows values (3, 307, TO_DATE('23 March 2023','DD Month YYYY'));

insert into borrows values (1, 305, TO_DATE('12 May 023','DD Month YYYY'));

insert into borrows values (4, 304, TO_DATE('14 May 2023','DD Month YYYY'));

insert into borrows values (3, 307, TO_DATE('15 May 2023','DD Month YYYY'));
```

```
insert into borrows values (3, 307, TO_DATE('18 May 2023','DD Month YYYY'));
insert into borrows values (2, 305, TO_DATE('15 May 2023','DD Month YYYY'));
insert into borrows values (5, 304, TO_DATE('18 May 2023','DD Month YYYY'));

commit;
–Task 2: a
SELECT student.name FROM student
JOIN borrows ON student.studentID = borrows.studentID
JOIN book ON borrows.bookID = book.bookID
WHERE book.title = 'Gulliver''s Travels';

–Task 2: b
SELECT MAX(author.age) as oldestAge FROM author
JOIN book ON author.authorID = book.authorID
WHERE book.genre = 'Non-Ficton';

–Task 2: c
SELECT student.phone FROM student
JOIN borrows ON student.studentID = borrows.studentID
JOIN book ON borrows.bookID = book.bookID
WHERE book.title = 'Gitanjali'
GROUP BY phone
HAVING COUNT(student.studentID) > 2;
```

Output:

[Output that came from executing the query.]

Task 2:

a.

NAME

Aranyak

Abir

Sajjad

b.

OLDESTAGE

45

c.

PHONE

01968723937