Experiment 1

Student Name: MISHIKA **UID: 23BCS13811**
Branch: CSE **Section/Group: KRG-2A**
Semester: 5th **Date of Performance: 18 JUL**
Subject Name: ADBMS **Subject Code: 23CSP-333**

1. Problem Statement & SQL Code:

Q1) Author-Book Relationship Using Joins and Basic SQL operations

Design two tables — one for storing author details and the other for book details.
Ensure a foreign key relationship from the book to its respective author.
Insert at least three records in each table.
Perform an INNER JOIN to link each book with its author using the common author ID.
Select the book title, author name, and author's country.

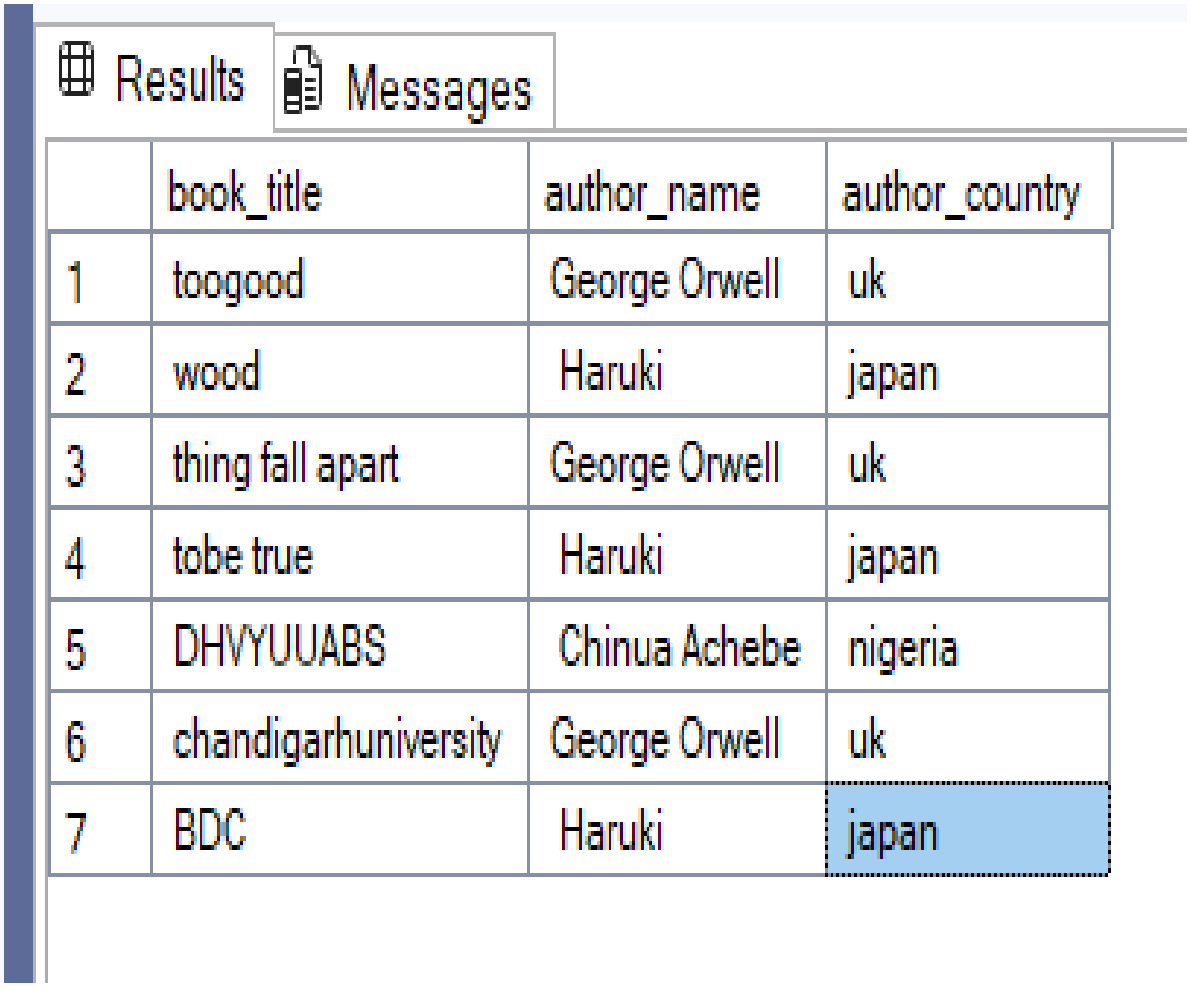
Solution:

```
create table author_tb (author_id int primary key, author_name varchar(20),
author_country varchar(10));
insert into author_tb (author_id, author_name, author_country)
values (101, 'George Orwell','uk'),
(102,' Haruki ', 'japan'),
(103,' Chinua Achebe', 'nigeria');

create table book_tb (book_id int primary key, book_name varchar(50), author_id int ,
publisheryear int);
insert into book_tb (book_id, book_name,author_id, publisheryear)
values (01, 'toogood', 101, 1978),
(02,'wood',102,2021),
(03,'thing fall apart',101,2021),
(04,'tobe true',102,2021),
(05,'DHVYUUABS ',103,2021),
(06,'chandigarhuniversity',101,2021),
(07,'BDC',102,2021);
```

```
select b1.book_name as [book_title] , a1.author_name, a1.author_country
from author_tb as a1
inner join
book_tb as b1
on a1.author_id=b1.author_id
```

OUTPUT:



Results | Messages

	book_title	author_name	author_country
1	toogood	George Orwell	uk
2	wood	Haruki	japan
3	thing fall apart	George Orwell	uk
4	tobe true	Haruki	japan
5	DHVYUUABS	Chinua Achebe	nigeria
6	chandigarhuniversity	George Orwell	uk
7	BDC	Haruki	japan

Q2) Department-Course Subquery and Access Control

Design normalized tables for departments and the courses they offer, maintaining a foreign key relationship.
Insert five departments and at least ten courses across those departments.
Use a subquery to count the number of courses under each department.
Filter and retrieve only those departments that offer more than two courses.

Grant SELECT-only access on the courses table to a specific user.

Solution:

```
CREATE TABLE Department (
    DeptID INT PRIMARY KEY,
    DeptName VARCHAR(100)
);


CREATE TABLE Course (
    CourseID INT PRIMARY KEY,
    CourseName VARCHAR(100),
    DeptID INT,
    FOREIGN KEY (DeptID) REFERENCES Department(DeptID)
);

INSERT INTO Department (DeptID, DeptName) VALUES
  (1, 'Computer Science'),
  (2, 'Physics'),
  (3, 'Mathematics'),
  (4, 'Chemistry'),
  (5, 'Biology');

INSERT INTO Course VALUES
(101, 'Data Structures', 1),
(102, 'Operating Systems', 1),
```

```
(103, 'Quantum Mechanics', 2),
(104, 'Electromagnetism', 2),
(105, 'Linear Algebra', 3),
(106, 'Calculus', 3),
(107, 'Organic Chemistry', 4),
(108, 'Physical Chemistry', 4),
(109, 'Genetics', 5),
(110, 'Molecular Biology', 5);


SELECT DeptName
  FROM Department
 WHERE DeptID IN (
    SELECT DeptID
    FROM Course
    GROUP BY DeptID
    HAVING COUNT(*) > 2
);


CREATE LOGIN Gautamcpp
WITH PASSWORD = 'Gautam1825'

CREATE USER Gautam
FOR LOGIN Gautamcpp

EXECUTE AS USER = ‘Gautam’
GRANT SELECT ON DEPARTMENT TO Gautam

REVOKE SELECT ON DEPARTMENT FROM Gautam
```

	DEPTNAME
1	Computer Science
2	Physics
3	Mathematics
4	Chemistry
5	Biology