NAME : YATEEN KUMAR

UID : 23BCS13730

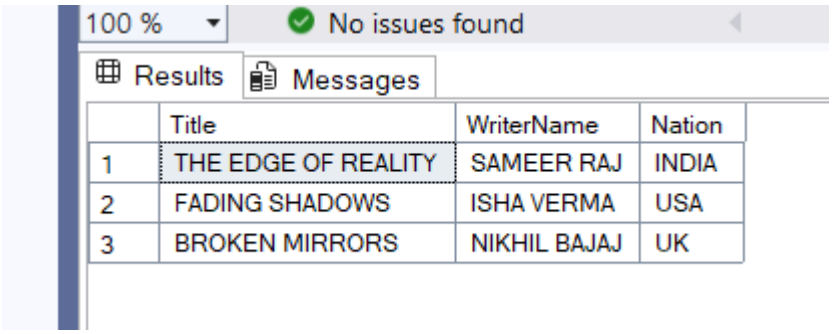
SECTION : 23_KRG_1(B)

QUESTION 1 : Write a query to retrieve book titles along with their author's name and country using INNER JOIN.

ANSWER :

```
CREATE TABLE Writers (
    WriterID INT PRIMARY KEY,
    WriterName VARCHAR(100),
    Nation VARCHAR(100)
);

CREATE TABLE Publications (
    PubID INT PRIMARY KEY,
    BookName VARCHAR(100),
    WriterID INT,
    FOREIGN KEY (WriterID) REFERENCES Writers(WriterID)
);

INSERT INTO Writers (WriterID, WriterName, Nation) VALUES (1, 'SAMEER RAJ', 'INDIA');
INSERT INTO Writers (WriterID, WriterName, Nation) VALUES (2, 'ISHA VERMA', 'USA');
INSERT INTO Writers (WriterID, WriterName, Nation) VALUES (3, 'NIKHIL BAJAJ', 'UK');

INSERT INTO Publications (PubID, BookName, WriterID) VALUES (201, 'THE EDGE OF REALITY', 1);
INSERT INTO Publications (PubID, BookName, WriterID) VALUES (202, 'FADING SHADOWS', 2);
INSERT INTO Publications (PubID, BookName, WriterID) VALUES (203, 'BROKEN MIRRORS', 3);

SELECT
    P.BookName AS Title,
```

```
    W.WriterName,
    W.Nation
FROM
    Publications P
INNER JOIN
Writers W ON P.WriterID = W.WriterID;
```



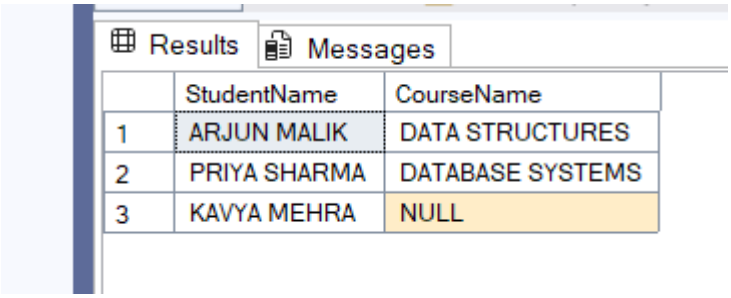
	Title	WriterName	Nation
1	THE EDGE OF REALITY	SAMEER RAJ	INDIA
2	FADING SHADOWS	ISHA VERMA	USA
3	BROKEN MIRRORS	NIKHIL BAJAJ	UK

QUESTION 2 : Write a query to display student names and their course names using LEFT OUTER JOIN.

ANSWER :

```
CREATE TABLE Students (
    StudentID INT PRIMARY KEY,
    StudentName VARCHAR(100)
);

CREATE TABLE Courses (
    CourseID INT PRIMARY KEY,
    CourseName VARCHAR(100),
    StudentID INT,
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID)
);

INSERT INTO Students (StudentID, StudentName) VALUES (1, 'ARJUN MALIK');
INSERT INTO Students (StudentID, StudentName) VALUES (2, 'PRIYA SHARMA');
INSERT INTO Students (StudentID, StudentName) VALUES (3, 'KAVYA MEHRA');
```

```
INSERT INTO Courses (CourseID, CourseName, StudentID) VALUES (101,
'DATA STRUCTURES', 1);

INSERT INTO Courses (CourseID, CourseName, StudentID) VALUES (102,
'DATABASE SYSTEMS', 2);


SELECT
    S.StudentName,
    C.CourseName
FROM
    Students S
LEFT OUTER JOIN
    Courses C ON S.StudentID = C.StudentID;
```



Results | Messages

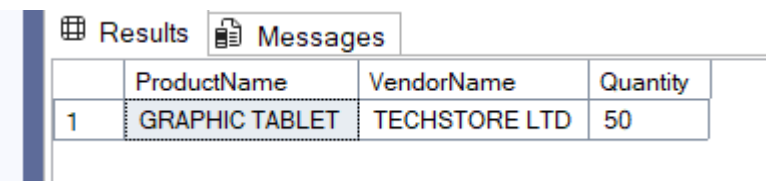
	StudentName	CourseName
1	ARJUN MALIK	DATA STRUCTURES
2	PRIYA SHARMA	DATABASE SYSTEMS
3	KAVYA MEHRA	NULL

QUESTION 3: Simulate a transaction where a product is inserted, a faulty product entry is added, a rollback to savepoint is performed, and the final product list is shown.

ANSWER :

```
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100)
);


CREATE TABLE Vendors (
    VendorID INT PRIMARY KEY,
    VendorName VARCHAR(100)
);
```

```
CREATE TABLE Supplies (
    SupplyID INT PRIMARY KEY,
    ProductID INT,
    VendorID INT,
    Quantity INT,
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID),
    FOREIGN KEY (VendorID) REFERENCES Vendors(VendorID)
);


INSERT INTO Products (ProductID, ProductName) VALUES (1, 'GRAPHIC TABLET');
INSERT INTO Vendors (VendorID, VendorName) VALUES (1, 'TECHSTORE LTD');
SET TRANSACTION;
INSERT INTO Supplies (SupplyID, ProductID, VendorID, Quantity) VALUES (1001, 1, 1, 50);
SAVEPOINT valid_supply;
INSERT INTO Supplies (SupplyID, ProductID, VendorID, Quantity) VALUES (1002, 99, 1, 30);
ROLLBACK TO valid_supply;
COMMIT;
SELECT
    P.ProductName,
    V.VendorName,
    S.Quantity
FROM
    Supplies S
JOIN
    Products P ON S.ProductID = P.ProductID
JOIN
    Vendors V ON S.VendorID = V.VendorID;
```



Results Messages

	ProductName	VendorName	Quantity
1	GRAPHIC TABLET	TECHSTORE LTD	50