Checking of Activities

**Activity 1 Conceptualized a Database:**

What are your entities?

* . Authors
* . Books
* . Customers
* . Orders
* . Payments

What are their attributes?

* . Authors (AuthorID, Name, Country)
* . Books (BookID, Title, Price,
* . Customers (CustomerID, Name, Email, Phone)
* . Orders (OrderID, CustomerID, OrderDate, TotalAmount)
* .Payments (PaymentID, OrderID, PaymentDate, AmountPaid, PaymentMethod)

**Activity 2: Applied Primary keys and foreign keys**

What are the primary keys for each table and what are its foreign keys?

* . Authors- AuthorID (Primary Key)
* . Books- BookID (Primary Key), AuthorID (Foreign Key)
* . Customers- CustomerID (Primary Key),
* . Orders- OrderID (Primary Key), CustomerID (Foreign Key)
* . Payments- PaymentID (Primary Key), OrderID (Foreign Key)

**Activity 3: Applied Joins and Scenarios for the purpose of the FOR EACH JOINS (Right, Left, Cross, Self).**

* LEFT JOIN

SELECT Orders.orderID, Payments.Amountpaid

FROM Orders

LEFT JOIN payments on Orders.OrderID = Payments.OrderID;

Scenario: This query allows you to see all orders, including those that may not have any payments associated with them.

* RIGHT JOIN

SELECT Authors.AuthorID, Books.Title

FROM Authors

RIGHT JOIN Books on Authors.AuthorID = Books.AuthorID;

Scenario: This query retrieves all books, including those that do not have any associated authors, ensuring visibility into books even if some lack authors.

* CROSS JOIN

SELECT Orders.OrderID, Customers.Name

FROM Orders

CROSS JOIN Customers;

Scenario: This query allows you to see every combination of orders and customers, which can be useful for analyses involving potential sales or customer interactions.

* SELF JOIN

SELECT A.Name AS AuthorName1, B.Name AS AuthorName2, A.Country

FROM Authors A, Authors B

WHERE A.AuthorID <> B.AuthorID

AND A.Country = B.Country

ORDER BY A.Country;

Scenario: This query pairs authors from the same country, letting you see which authors share the same geographical background.