**REPORT**

**1. Table Creation:**

The following tables were created:

* **Authors Table**: Contains information about authors, identified by an author\_id. The columns are:
  + author\_id (INT): Unique identifier for each author (Primary Key).
  + author\_name (VARCHAR(50)): The name of the author.
* **Books Table**: Contains information about books, identified by a book\_id. The columns are:
  + book\_id (INT): Unique identifier for each book (Primary Key).
  + title (VARCHAR(50)): The title of the book.
  + author\_id (INT): Foreign key referencing the author\_id from the Authors table.
* **Sales Table**: Contains sales information for each book, identified by book\_id. The columns are:
  + book\_id (INT): Foreign key referencing the book\_id from the Books table.
  + units\_sold (INT): The number of units sold for the book.
  + unit\_price (DECIMAL(10,2)): The price of each unit sold.

**2. Data Insertion:**

The following data was inserted into the tables:

* **Authors**:
  + 5 authors were added, including famous names like J.K. Rowling, George R. R. Martin, and Stephen King.
* **Books**:
  + 10 books were added, each associated with an author. For example, J.K. Rowling’s books are included such as "Harry Potter and the Sorcerer’s Stone" and "Harry Potter and the Chamber of Secrets".
* **Sales**:
  + 10 records of sales data for the books were added. Each record includes the number of units sold and the unit price for each book. For instance, "Harry Potter and the Sorcerer’s Stone" had 100,000 units sold at a price of $19.99 per unit.

**3. Queries Performed:**

* **Check for Missing Values in Authors Table**:

sql

Copy code

SELECT \* FROM Authors

WHERE author\_name IS NULL;

This query checks for missing values in the author\_name column in the Authors table. Since no authors have null values in their names, this query would return an empty result.

* **Join Books with Authors**:

sql

Copy code

SELECT Books.book\_id, Books.title, Authors.author\_name

FROM Books

INNER JOIN Authors ON Books.author\_id = Authors.author\_id;

This query joins the Books table with the Authors table based on the author\_id. It returns the book\_id, title, and author\_name for each book. This gives a detailed list of books with the corresponding author's name.

* **Sum of Sales Revenue Per Book**:

sql

Copy code

SELECT Books.title,

SUM(Sales.units\_sold \* Sales.unit\_price) AS total\_sales\_revenue

FROM Sales

INNER JOIN Books ON Sales.book\_id = Books.book\_id

GROUP BY Books.book\_id;

This query calculates the total sales revenue for each book by multiplying the units\_sold by the unit\_price for each book. The results show the total sales revenue per book. For example, "Harry Potter and the Sorcerer’s Stone" would have a revenue of 100,000 \* 19.99.

* **Join Books and Sales**:

sql

Copy code

SELECT Books.\*, Sales.\*

FROM Books

INNER JOIN Sales ON Sales.book\_id = Books.book\_id;

This query joins the Books table with the Sales table to display all columns from both tables, showing the full details of each book along with its sales data.