**Assignment Week 6 - Francky Ciceron**

**Public Library Management (Discussion Questions)**

1. In our library database, we track which branch a book was borrowed from, but books can exist at multiple branches. How would you modify the schema to track the actual inventory at each branch?

I would add a new table called “branch\_inventory” with 3 columns for branch\_id, book\_id, and copies\_owned. This would let the library track exactly how many copies of each book are available at each branch, instead of only knowing where a book was borrowed.

1. Based on the provided data model, what business questions could library administrators answer using SQL queries that we haven't covered in our exercise?

With the current data model, library administrators could answer questions such as:

* Which books or genres are the most borrowed?
* Which patrons borrow the most books?
* Which branches are the busiest?
* How many books are overdue at this moment?

1. How would you extend this schema to track additional patron interactions, such as reserved books, late fees, or participation in library programs?

To extend the schema, I would add 3 new tables:

* “Reservations” to track reserved books with 3 columns (patron\_id, book\_id, reservation\_date).
* “Fines” to track late fees with 3 columns (patron\_id, loan\_id, amount, paid/unpaid).
* “Program attendance” to track library programs and which patrons participate.

1. For tasks 1-3, how could you combine them into a single, more complex query that finds recent history books with multiple copies?

Using the extended schema, we could write a single query to find recent history books (genre = history, published in the last few years) that have multiple copies in the branch\_inventory table. The same query could also connect (by using JOIN) to reservations, late fees, or program attendance to give administrators a full picture of book demand and patron activity.

1. What performance considerations should be kept in mind when running complex joins and aggregations on large library datasets?

For large datasets, performance can slow down with complex joins and aggregations. To help with this, the library could:

* Add indexes on common search columns like book\_id, branch\_id, and patron\_id.
* Avoid joining unnecessary tables.
* Use efficient aggregations or pre-compute some summaries.