1. What is the purpose of using the DROP SCHEMA IF EXISTS and CREATE SCHEMA IF NOT EXISTS statements?

* Drop Schema- Drops one schema per command and related objects in the database. ‘Drop Scheme’ is the key word. Only using Drop Schema does not drop the dependent objects. ‘DROP SCHEMA [schema name] CASCADE’ is used instead.
* If Not Exists – is the condition which states that drop or create schema only if it is present in the database. This is to avoid any conflict and ensure a clean setup.
* It is possible to drop multiple schemas at ones.
* Only an owner or a superuser can drop the schema.
* Create Schema- key word is used to create new schema if one specified does not already exists in the database.

1. What data types are used for the user\_id and publisher\_id columns in the user\_login and publisher tables, respectively?

* User\_id, publisher\_id are of ‘text’ type – This store up to 1GB of text data. Basically, any length of text data within this range.

1. What is the role of the PRIMARY KEY constraint in this SQL code, and where is it applied?

* This is to ensure each record is identified uniquely. This column should not accept null values.

1. How are relationships between tables (e.g., books table and author or publisher table) established?

* Books table has a column called ‘Author\_id’ which references to author table
* Books table also has a column called Publisher\_id which references to publisher table.

1. What is the significance of the REFERENCES keyword in the books and books\_issue tables?

* REFERENCES is the key word used to define foreign key relationship in a table. Issue\_id in the books table references issue\_id(primary key) from the books\_issue table.

1. In the books\_issue table, what does the payment\_transaction\_id column represent?

* Payment\_transaction\_id is a text type column- which is the unique transaction id of the payment

1. How is the fine for overdue books calculated based on the structure of the books\_issue table?

Current\_fine – initial value set to 0

* For every day after return date add fine\_per\_day to current\_fine column in the books\_issue table
* Once the payment is made – this payment to populate in fine\_paid
* Current fine should come back to 0
* You can also add a column called late day in loans table to keep track of how many days are overdue on that book.

Ex : current fine = 0, fine\_per\_day= 0.19 cent , then if return date is Less than current day- calculate number of days from current day to return date , multiply by – fine amount and populate this in fine due.

* If return data is greater than current day, then fine due to be 0

1. What is the purpose of the settings table, and how does it influence the management of book loans?

* Settings table keeps a track of how many books are issued to the user
* Each book due date
* If returned – then return date
* And fine per day – which should be calculated field – Late date \* fine amount per day

1. What happens if a user’s current fine is not paid (fine\_paid is FALSE)? Does the system restrict further book issuance or impose additional penalties?

* Yes – there should be condition to say due date > 30 days per book and fine due is greater than x amount – then account should be blocked.

1. What is the is\_available field in the books table used for, and how can it be helpful in managing the library’s inventory

* is\_available column is a Boolean type – holds values ‘yes if book is not per-booked and present in the current library
* Filed set to ‘No’ if book is not available in the library or pre-booked by another user.

1. How do the start\_date and last\_date fields in the staff table impact the record of staff members in the system?

* Start and last date refer to how long the staff member is employed in the system. If last date is set, then their activity can be set to ‘inactive’ if the last date is less than current date.

1. What do you think would happen if the books\_issue table has multiple entries for the same reader\_id and book\_id combination?

* You cannot have same reader or same book multiple times in the system. It is one reader per book exhibiting one-one relationship.
* Database should still allow one reader – different book\_id. NOTE : that book id is unique to each book . library can have multiple copies of same book. But each book should have unique book\_id.

In case there are same readers are populated for same book in the book issue table then one to one relation is violated. Combination of Reader\_id and book\_id should be made unique in the system to ensure no duplicates created in the database

1. How is the email\_id column in the user\_login table significant in the context of user management?

* In the library management system – user\_id should represent one individual with one unique email address.
* There could be minors burrowing books in this case – parent email takes higher priority . this should be managed with in the DB.

1. What is the purpose of the books\_issued\_current column in the readers table, and how does it help track readers' borrowing activities?

* books\_issued\_current keeps track of how many books that they have currently borrowed.
* This should be (books\_issued\_current-returned\_books). Ex: if bookissued=3 , returned=3 then books\_issued\_current-returned=0
* If bookissued=7 and returned=5 then books\_issued\_current=2

1. What happens when a book\_id in the books\_issue table refers to a non-existing book in the books table (i.e., an invalid reference)?

* This will affect the total number of existing actual books in the library – by incorrectly inflating the number of books.
* If the if this book\_id is matched with burrower then it wrongly interpret the burrowing activity.

1. How would you modify the schema if you wanted to track the genre or category of books in the library?
2. If you wanted to add a table to store book reviews by readers, what columns would you consider including in the new table, and how would you link it to the books table?