

Lab 1 (TP1): Creating Tables, DML and DDL Statements

1) Database Schema (6 tables)

Consider the “**Library**” database consisting of the following six tables.

Table *Book*

Attribute	Type / Constraint	Description
book_id	integer (primary key)	Unique book identifier
title	string (100 chars), NOT NULL	Book title
author_id	integer (foreign key to <i>Author</i>)	Book author (reference)
publication_year	integer, must be > 1800	Year of publication
category	string (30 chars)	Book category
available	boolean (1=available, 0=not)	Availability indicator

Table *Author*

Attribute	Type / Constraint	Description
author_id	integer (primary key)	Unique author identifier
last_name	string (50 chars), NOT NULL	Author's last name
first_name	string (50 chars)	Author's first name
nationality	string (30 chars)	Author's nationality
birth_year	integer	Author's year of birth

Table *Member*

Attribute	Type / Constraint	Description
member_id	integer (primary key)	Unique member identifier
last_name	string (30 chars), NOT NULL	Member's last name
first_name	string (30 chars), NOT NULL	Member's first name
email	string, UNIQUE	Member's email
phone	string (10 chars)	Phone number
registration_date	date, default = today	Registration date

Table *Loan*

Attribute	Type / Constraint	Description
loan_id	integer (primary key)	Unique loan identifier
book_id	integer (foreign key to <i>Book</i>)	Borrowed book
member_id	integer (foreign key to <i>Member</i>)	Borrowing member
loan_date	date	Loan date
return_date	date	Return date

Table *Librarian*

Attribute	Type / Constraint	Description
librarian_id	integer (primary key)	Unique librarian identifier
last_name	string (30 chars), NOT NULL	Librarian's last name
first_name	string (30 chars), NOT NULL	Librarian's first name
username	string (15 chars), UNIQUE, NOT NULL	Login identifier
password	string	Password

Table *Reservation*

Attribute	Type / Constraint	Description
reservation_id	integer (primary key)	Unique reservation identifier
book_id	integer (foreign key to <i>Book</i>)	Reserved book
member_id	integer (foreign key to <i>Member</i>)	Reserving member
reservation_date	date	Reservation date

Initial Data (Seed Population)

Table *Book*

Book_ID	Title	Author	Year	Category	Avail.
1	Ouloum Al-Hadith	Mohammed Ibn Salih	2001	Religion	1
2	Moqaddima Ibn Khaldoun	Abdelrahman Ibn Khaldoun	1377	History	1
3	Notre-Dame de Paris	Victor Hugo	1831	Novel	1
4	Les Misérables	Victor Hugo	1862	Novel	1
5	Guerre et Paix	Léon Tolstoï	1869	Novel	1
6	Systèmes d'exploitation modernes	Andrew Tanenbaum	2014	Computer Science	1
7	Réseaux informatiques	Andrew Tanenbaum	2021	Computer Science	1
8	Réseaux : des bases à l'Internet	Guy Pujolle	2018	Computer Science	1

Book_ID	Title	Author	Year	Category	Avail.
9	Du Contrat Social	Jean-Jacques Rousseau	1762	Politics	1
10	Les Conditions de la Renaissance	Malek Bennabi	1948	Philosophy	1

Table Member

Member_ID	Last_Name	First_Name	Email	Phone	Reg. Date
1	Ben Youssef	Ibrahim	ibrahim.benyoussef@mail.com	0555123456	2024-01-10
2	Naimi	Khaled	khaled.naimi@mail.com	0666789012	2024-02-15
3	Sherif	Aicha	aicha.sherif@mail.com	0777345678	2024-03-05
4	Arabi	Youssef	youssef.arabi@mail.com	0544987654	2024-04-20
5	Zaoui	Fatima	fatima.zaoui@mail.com	0566098765	2024-05-30

Table Author

Author_ID	Last_Name	First_Name	Nationality	Birth_Year
1	Ibn Salih	Mohammed	Saudi	1925
2	Ibn Khaldoun	Abdelrahman	Tunisian	1332
3	Hugo	Victor	French	1802
4	Tolstoï	Léon	Russian	1828
5	Tanenbaum	Andrew	American	1944
6	Pujolle	Guy	French	1949
7	Rousseau	Jean-Jacques	French	1712
8	Bennabi	Malek	Algerian	1905

Table Librarian

Librarian_ID	Last_Name	First_Name	Username	Password
1	Eddine	Mohamed Amine	m.amine	hashed_password_1
2	Zeitouni	Hasna	h.zeitouni	hashed_password_2

Table Loan

Loan_ID	Book_ID	Member_ID	Loan_Date	Return_Date
1	3	1	01-06-2024	16-06-2024
2	1	3	05-06-2024	
3	5	2	10-06-2024	

Table *Reservation*

<code>Reservation_ID</code>	<code>Book_ID</code>	<code>Member_ID</code>	<code>Reservation_Date</code>
1	5	4	2024-06-12
2	1	5	2024-06-14

Tasks

Table creation and verification

- Q1) Create these six tables.
- Q2) Verify the creation of each table (`DESC table_name`).

Add and modify columns

- Q3) Add a column `page_count` (integer) to the *Book* table.
- Q4) Modify the `category` column of *Book* so that it accepts 50 characters.
- Q5) Remove the `page_count` column from *Book*.

Insert and display data

- Q6) Insert the seed population into the tables *Book*, *Author*, *Member*, *Loan*, *Librarian*, and *Reservation*.
- Q7) Confirm the insertion of rows (`SELECT * FROM table_name;`).

Update and delete data

- Q8) Update the `available` values in *Book* to set to 0 the books currently on loan.
- Q9) Update the return date of the book borrowed by member *Sherif Aicha* and set `available = 1`.
- Q10) Delete a member and observe the effect on loans (`ON DELETE CASCADE`).

Constraints and data integrity

- Q11) Add a `CHECK` constraint to ensure that `publication_year > 1800` in the *Book* table.
- Q12) Add a `UNIQUE` constraint on the `email` column of the *Member* table.
- Q13) Add a `NOT NULL` constraint on the `username` column of the *Librarian* table.
- Q14) Insert into *Member* a row with an email that already exists. What do you observe?
- Q15) Insert into *Book* a row with `publication_year = 1700`. What do you observe?
- Q16) Insert into *Loan* a row with a `return_date` earlier than `loan_date`. What do you observe?

Additional Proposed Tasks (DDL, DML, Constraints)

A) Schema refinement (DDL)

- Q17) Rename table `member` to `library_member`. Update dependent foreign keys accordingly.
- Q18) Rename column `book.title` to `book.book_title`.
- Q19) Change `member.phone` to accept up to 20 characters (to support country codes like +213...).
- Q20) Add column `created_at` `TIMESTAMP DEFAULT now()` to all tables.
- Q21) Add column `updated_at` `TIMESTAMP` to all tables.

B) Integrity constraints (CHECK, UNIQUE, NOT NULL, FK)

- Q22) Enforce that `book.available` cannot be `NULL`.
- Q23) Add a `CHECK` on `member.phone` to allow only digits (and optionally +) using a regex.
- Q24) Add a `CHECK` to `loan`: `return_date IS NULL OR return_date >= loan_date`.
- Q25) Add a `CHECK` to `reservation`: `reservation_date >= registration_date` of the corresponding member (you will discover this is **not** a simple `CHECK` and requires a trigger or a deferred constraint approach).
- Q26) Add a `UNIQUE` constraint so that two authors cannot share the same (`last_name`, `first_name`, `birth_year`).
- Q27) Add a `UNIQUE` constraint on `librarian.username` (if not already enforced in the schema).
- Q28) Add a foreign key from `loan(book_id)` to `book(book_id)` with `ON DELETE RESTRICT` and explain the rationale (avoid deleting borrowed books).

C) Data manipulation (DML) — inserts/updates/deletes with business rules

- Q33) Insert 3 new books and 2 new members of your choice; ensure constraints are respected.
- Q34) Update all books in category `Computer Science` to category `CS`.
- Q35) Update all `loan.return_date` values that are `NULL` to `CURRENT_DATE` **only if** `loan_date` is older than 30 days (use `CURRENT_DATE - loan_date > 30`).
- Q36) Delete all reservations older than 1 year.
- Q37) Delete a book that is referenced by a loan and observe the foreign key error.

D) Indexing (performance-oriented DDL)

- Q30) Create an index on `book(author_id)` and justify why it helps.
- Q31) Create a composite index on `loan(member_id, return_date)` to speed up “active loans per member” queries.
- Q32) Create a `UNIQUE` index on `member(email)` instead of a `UNIQUE` constraint.