1. **Створіть базу даних для керування бібліотекою книг згідно зі структурою, наведеною нижче. Використовуйте DDL-команди для створення необхідних таблиць та їх зв'язків:**

CREATE DATABASE LibraryManagement;

USE LibraryManagement;

CREATE TABLE authors (

author\_id INT AUTO\_INCREMENT PRIMARY KEY,

author\_name VARCHAR(100) NOT NULL

);

CREATE TABLE genres (

genre\_id INT AUTO\_INCREMENT PRIMARY KEY,

genre\_name VARCHAR(100) NOT NULL

);

CREATE TABLE books (

book\_id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(100) NOT NULL,

publication\_year YEAR NOT NULL,

author\_id INT,

genre\_id INT,

FOREIGN KEY (author\_id) REFERENCES authors(author\_id),

FOREIGN KEY (genre\_id) REFERENCES genres(genre\_id)

);

CREATE TABLE users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(100) NOT NULL,

email VARCHAR(100) NOT NULL

);

CREATE TABLE borrowed\_books (

borrow\_id INT AUTO\_INCREMENT PRIMARY KEY,

book\_id INT,

user\_id INT,

borrow\_date DATE NOT NULL,

return\_date DATE,

FOREIGN KEY (book\_id) REFERENCES books(book\_id),

FOREIGN KEY (user\_id) REFERENCES users(user\_id)

);

1. **Заповніть таблиці простими видуманими тестовими даними. Достатньо одного-двох рядків у кожну таблицю:**

INSERT INTO authors (author\_name) VALUES ('John Doe');

INSERT INTO authors (author\_name) VALUES ('Jane Smith');

INSERT INTO genres (genre\_name) VALUES ('Fiction');

INSERT INTO genres (genre\_name) VALUES ('Science');

INSERT INTO books (title, publication\_year, author\_id, genre\_id) VALUES ('The Great Adventure', 2020, 1, 1);

INSERT INTO books (title, publication\_year, author\_id, genre\_id) VALUES ('Science for Everyone', 2019, 2, 2);

INSERT INTO users (username, email) VALUES ('user1', 'user1@example.com');

INSERT INTO users (username, email) VALUES ('user2', 'user2@example.com');

INSERT INTO borrowed\_books (book\_id, user\_id, borrow\_date, return\_date) VALUES (1, 1, '2023-01-01', '2023-01-15');

INSERT INTO borrowed\_books (book\_id, user\_id, borrow\_date, return\_date) VALUES (2, 2, '2023-02-01', '2023-02-20');

**3. Напишіть запит за допомогою операторів FROM та INNER JOIN, що об’єднує всі таблиці даних, які ми завантажили з файлів: order\_details, orders, customers, products, categories, employees, shippers, suppliers:**

* 1. ***Для цього ви маєте знайти спільні ключі:***

use mydb;

ALTER TABLE products

ADD CONSTRAINT fk\_products\_suppliers

FOREIGN KEY (supplier\_id) REFERENCES suppliers(id);

ALTER TABLE products

ADD CONSTRAINT fk\_products\_categories

FOREIGN KEY (category\_id) REFERENCES categories(id);

ALTER TABLE orders

ADD CONSTRAINT fk\_orders\_customers

FOREIGN KEY (customer\_id) REFERENCES customers(id);

ALTER TABLE orders

ADD CONSTRAINT fk\_orders\_employees

FOREIGN KEY (employee\_id) REFERENCES employees(employee\_id);

ALTER TABLE orders

ADD CONSTRAINT fk\_orders\_shippers

FOREIGN KEY (shipper\_id) REFERENCES shippers(id);

ALTER TABLE order\_details

ADD CONSTRAINT fk\_order\_details\_orders

FOREIGN KEY (order\_id) REFERENCES orders(id);

ALTER TABLE order\_details

ADD CONSTRAINT fk\_order\_details\_products

FOREIGN KEY (product\_id) REFERENCES products(id);

* 1. **Напишіть запит за допомогою операторів FROM та INNER JOIN:**

use mydb;

CREATE TABLE all\_tables AS

SELECT

orders.id AS order\_id,

orders.customer\_id AS orders\_customer\_id,

orders.employee\_id AS orders\_employee\_id,

orders.date AS order\_date,

orders.shipper\_id AS orders\_shipper\_id,

customers.id AS customer\_id,

customers.name AS customer\_name,

customers.contact AS customer\_contact,

customers.address AS customer\_address,

customers.city AS customer\_city,

customers.postal\_code AS customer\_postal\_code,

customers.country AS customer\_country,

employees.employee\_id AS employee\_id,

employees.last\_name AS employee\_last\_name,

employees.first\_name AS employee\_first\_name,

employees.birthdate AS employee\_birthdate,

employees.photo AS employee\_photo,

employees.notes AS employee\_notes,

shippers.id AS shipper\_id,

shippers.name AS shipper\_name,

shippers.phone AS shipper\_phone,

order\_details.id AS order\_detail\_id,

order\_details.order\_id AS order\_detail\_order\_id,

order\_details.product\_id AS order\_detail\_product\_id,

order\_details.quantity AS order\_detail\_quantity,

products.id AS product\_id,

products.name AS product\_name,

products.supplier\_id AS product\_supplier\_id,

products.category\_id AS product\_category\_id,

products.unit AS product\_unit,

products.price AS product\_price,

categories.id AS category\_id,

categories.name AS category\_name,

categories.description AS category\_description,

suppliers.id AS supplier\_id,

suppliers.name AS supplier\_name,

suppliers.contact AS supplier\_contact,

suppliers.address AS supplier\_address,

suppliers.city AS supplier\_city,

suppliers.postal\_code AS supplier\_postal\_code,

suppliers.country AS supplier\_country,

suppliers.phone AS supplier\_phone

FROM

order\_details

INNER JOIN orders ON order\_details.order\_id = orders.id

INNER JOIN customers ON orders.customer\_id = customers.id

INNER JOIN employees ON orders.employee\_id = employees.employee\_id

INNER JOIN shippers ON orders.shipper\_id = shippers.id

INNER JOIN products ON order\_details.product\_id = products.id

INNER JOIN categories ON products.category\_id = categories.id

INNER JOIN suppliers ON products.supplier\_id = suppliers.id;

**4. Виконайте запити, перелічені нижче.**

* 1. ***Визначте, скільки рядків ви отримали (за допомогою оператора COUN*T):**

use mydb;

SELECT COUNT(\*) AS total\_rows

FROM all\_tables;

* 1. ***Змініть декілька операторів INNER на LEFT чи RIGHT. Визначте, що відбувається з кількістю рядків. Чому? Напишіть відповідь у текстовому файлі:***

use mydb;

SELECT COUNT(\*) AS total\_rows\_with\_left\_join

FROM order\_details

right JOIN orders ON order\_details.order\_id = orders.id

LEFT JOIN або RIGHT JOIN збільшує кількість рядків, оскільки ці оператори включають всі рядки з однієї таблиці, навіть якщо немає відповідних рядків у іншій таблиці.

* 1. ***Оберіть тільки ті рядки, де employee\_id > 3 та ≤ 10:***

use mydb;

SELECT \*

FROM all\_tables

WHERE employee\_id > 3 AND employee\_id <= 10;

* 1. ***Згрупуйте за іменем категорії, порахуйте кількість рядків у групі, середню кількість товару (кількість товару знаходиться в order\_details.quantity):***

use mydb;

SELECT category\_name, COUNT(\*) AS num\_rows, AVG(order\_detail\_quantity) AS avg\_quantity

FROM all\_tables

GROUP BY category\_name;

* 1. ***Відфільтруйте рядки, де середня кількість товару більша за 21***:

use mydb;

SELECT category\_name, num\_rows, avg\_quantity

FROM (

SELECT category\_name, COUNT(\*) AS num\_rows, AVG(order\_detail\_quantity) AS avg\_quantity

FROM all\_tables

GROUP BY category\_name

) AS grouped

WHERE avg\_quantity > 21;

* 1. ***Відсортуйте рядки за спаданням кількості рядків:***

use mydb;

SELECT category\_name, num\_rows, avg\_quantity

FROM (

SELECT category\_name, COUNT(\*) AS num\_rows, AVG(order\_detail\_quantity) AS avg\_quantity

FROM all\_tables

GROUP BY category\_name

) AS grouped

ORDER BY num\_rows DESC;

* 1. ***Виведіть на екран (оберіть) чотири рядки з пропущеним першим рядком:***

use mydb;

SELECT category\_name, num\_rows, avg\_quantity

FROM (

SELECT category\_name, COUNT(\*) AS num\_rows, AVG(order\_detail\_quantity) AS avg\_quantity

FROM all\_tables

GROUP BY category\_name

) AS grouped

ORDER BY num\_rows DESC

LIMIT 1, 4;