**Домашнє завдання до Теми 4. DML та DDL команди. Складні SQL вирази**

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

Структура БД

a) Назва схеми — “LibraryManagement”

b) Таблиця "authors":

author\_id (INT, автоматично зростаючий PRIMARY KEY)

author\_name (VARCHAR)

c) Таблиця "genres":

genre\_id (INT, автоматично зростаючий PRIMARY KEY)

genre\_name (VARCHAR)

d) Таблиця "books":

book\_id (INT, автоматично зростаючий PRIMARY KEY)

title (VARCHAR)

publication\_year (YEAR)

author\_id (INT, FOREIGN KEY зв'язок з "Authors")

genre\_id (INT, FOREIGN KEY зв'язок з "Genres")

e) Таблиця "users":

user\_id (INT, автоматично зростаючий PRIMARY KEY)

username (VARCHAR)

email (VARCHAR)

f) Таблиця "borrowed\_books":

borrow\_id (INT, автоматично зростаючий PRIMARY KEY)

book\_id (INT, FOREIGN KEY зв'язок з "Books")

user\_id (INT, FOREIGN KEY зв'язок з "Users")

borrow\_date (DATE)

return\_date (DATE)

--------------------------------------------------------------------------------------------------------------

use LibraryManagement;

DROP TABLE IF EXISTS borrowed\_books;

DROP TABLE IF EXISTS books;

DROP TABLE IF EXISTS authors;

DROP TABLE IF EXISTS genres;

DROP TABLE IF EXISTS users;

CREATE TABLE authors (

author\_id INT PRIMARY KEY AUTO\_INCREMENT,

author\_name VARCHAR(100)

);

CREATE TABLE genres (

genre\_id INT PRIMARY KEY AUTO\_INCREMENT,

genre\_name VARCHAR(100)

);

CREATE TABLE books (

book\_id INT PRIMARY KEY AUTO\_INCREMENT,

title VARCHAR(100),

publication\_year YEAR,

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 PRIMARY KEY AUTO\_INCREMENT,

username VARCHAR(100),

email VARCHAR(100)

);

CREATE TABLE borrowed\_books(

borrow\_id INT PRIMARY KEY AUTO\_INCREMENT,

book\_id INT,

user\_id INT,

borrow\_date DATE,

return\_date DATE,

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

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

);

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

INSERT INTO authors (author\_name) VALUES

('Vito'),

('Yulia'),

('Ania');

INSERT INTO genres (genre\_name) VALUES

('Science fiction'),

('Fiction'),

('Novel');

INSERT INTO books (title, publication\_year, author\_id, genre\_id) VALUES

('Book-1', '1921', 2, 3),

('Book-2', '1901', 1, 3),

('Book-3', '2021', 2, 1),

('Book-4', '2001', 3, 2),

('Book-5', '2013', 3, 2),

('Book-6', '1956', 2, 3);

INSERT INTO users (username, email) VALUES

('username-1', "email1@mail.com"),

('username-2', "email2@mail.com"),

('username-3', "email3@mail.com"),

('username-4', "email4@mail.com");

INSERT INTO borrowed\_books (book\_id, user\_id, borrow\_date, return\_date ) VALUES

(3, 2, '2024-03-31', NULL),

(2, 4, '2024-02-22', '2024-02-28'),

(5, 1, '2024-02-22', '2024-02-28');

**3. Перейдіть до бази даних, з якою працювали у темі 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. Для цього ви маєте знайти спільні ключі.**

Перевірте правильність виконання запиту.

SELECT \* FROM order\_details

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

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

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

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

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

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

LEFT JOIN suppliers ON products.supplier\_id = suppliers.id

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

**Визначте, скільки рядків ви отримали (за допомогою оператора COUNT).**

SELECT COUNT(\*) as total FROM order\_details

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

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

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

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

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

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

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

Кількість рядків - 518

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

SELECT COUNT(\*) total FROM order\_details

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

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

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

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

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

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

LEFT JOIN suppliers ON products.supplier\_id = suppliers.id

Кількість рядків -535

Деякі рядки лівих таблиць відповідають рядкам правих таблиць.

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

SELECT COUNT(\*) total FROM order\_details

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

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

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

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

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

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

LEFT JOIN suppliers ON products.supplier\_id = suppliers.id

WHERE employees.employee\_id > 3 AND employees.employee\_id <= 10

Кількість рядків -317

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

SELECT categories.name category, COUNT(categories.name) count, AVG(order\_details.quantity) average\_quantity FROM order\_details

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

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

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

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

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

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

LEFT JOIN suppliers ON products.supplier\_id = suppliers.id

GROUP BY categories.name

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

SELECT categories.name category, COUNT(categories.name) count, AVG(order\_details.quantity) average\_quantity FROM order\_details

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

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

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

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

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

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

LEFT JOIN suppliers ON products.supplier\_id = suppliers.id

GROUP BY categories.name

HAVING average\_quantity > 21

**Відсортуйте рядки за спаданням кількості рядків.**

SELECT categories.name category, COUNT(categories.name) count, AVG(order\_details.quantity) average\_quantity FROM order\_details

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

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

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

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

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

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

LEFT JOIN suppliers ON products.supplier\_id = suppliers.id

GROUP BY categories.name

HAVING average\_quantity > 21

ORDER BY count DESC

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

SELECT categories.name category, COUNT(categories.name) count, AVG(order\_details.quantity) average\_quantity FROM order\_details

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

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

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

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

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

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

LEFT JOIN suppliers ON products.supplier\_id = suppliers.id

GROUP BY categories.name

HAVING average\_quantity > 21

ORDER BY count DESC

LIMIT 4

OFFSET 1