**HW\_4**

1. Створіть базу даних.

CREATE SCHEMA LibraryManagement;

USE LibraryManagement;

CREATE TABLE authors (

author\_id INT AUTO\_INCREMENT PRIMARY KEY,

author\_name VARCHAR(45) NOT NULL

);

CREATE TABLE genres (

genre\_id INT AUTO\_INCREMENT PRIMARY KEY,

genre\_name VARCHAR(45) NOT NULL

);

CREATE TABLE books (

book\_id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(45) NOT NULL,

publication\_year YEAR,

author\_id INT,

genre\_id INT,

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

ON DELETE CASCADE

ON UPDATE CASCADE,

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

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(45) NOT NULL,

email VARCHAR(255)

);

CREATE TABLE borrowed\_books (

borrow\_id INT AUTO\_INCREMENT PRIMARY KEY,

book\_id INT,

user\_id INT,

borrow\_date DATE,

return\_date DATE,

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

ON DELETE CASCADE

ON UPDATE CASCADE,

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

ON DELETE CASCADE

ON UPDATE CASCADE

);

A screenshot of a computer

Description automatically generated

2. Заповніть таблиці

INSERT INTO authors (author\_name)

VALUES

('Stephenie Meyer'),

('J.K. Rowling');

INSERT INTO genres (genre\_name)

VALUES

('Fantasy'),

('Romance'),

('History');

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

VALUES

('Twilight', 2005, 1, 2),

('New Moon', 2006, 1, 2),

('Eclipse', 2007, 1, 2),

('Breaking Dawn', 2008, 1, 2),

('Life and Death', 2010, 1, 2),

('Midnight Sun', 2020, 1, 2),

("Harry Potter and the Philosopher's Stone", 1998, 2, 1),

('Harry Potter and the Chamber of Secrets', 1999, 2, 1),

('Harry Potter and the Prisoner of Azkaban', 1999, 2, 1),

('Harry Potter and the Goblet of Fire', 2000, 2, 1),

('Harry Potter and the Order of the Phoenix', 2003, 2, 1),

('Harry Potter and the Half-Blood Prince', 2005, 2, 1),

('Harry Potter and the Deathly Hallows', 2007, 2, 1);

INSERT INTO users (username, email)

VALUES

('Ivan Ivanov', 'ivan.ivanov@gmail.com'),

('Tom Toman', 'tom.toman@gmail.com'),

('Alen Rolan', 'alen.rolan@gmail.com');

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

VALUES

(1, 1, '2024-06-12', '2024-07-02'),

(7, 2, '2024-05-17', '2024-06-22'),

(6, 3, '2024-07-12', NULL);

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

3. Напишіть запит за допомогою операторів FROM та INNER JOIN

USE hw\_3;

SELECT

det.order\_id,

ord.date,

det.quantity,

det.product\_id,

prod.name as prod\_name,

cat.name as categories\_name,

cat.description,

prod.unit,

prod.price,

supp.name as suppliers\_name,

supp.contact,

supp.address,

supp.city,

supp.postal\_code,

supp.country,

supp.phone as phone\_suppliers,

ord.customer\_id,

cust.name as customer\_name,

cust.contact as customer\_contact,

cust.address as customer\_address,

cust.city as customer\_city,

cust.postal\_code as customer\_postal\_code,

cust.country as customer\_country,

emp.\*,

ord.shipper\_id,

ship.name as shipper\_name,

ship.phone as shipper\_phone

FROM orders as ord

INNER JOIN order\_details as det

ON ord.id = det.order\_id

INNER JOIN customers as cust

ON ord.customer\_id = cust.id

INNER JOIN employees as emp

ON ord.employee\_id = emp.employee\_id

INNER JOIN shippers as ship

ON ord.shipper\_id = ship.id

INNER JOIN products as prod

ON det.product\_id = prod.id

INNER JOIN suppliers as supp

ON prod.supplier\_id = supp.id

INNER JOIN categories as cat

ON prod.category\_id = cat.id;

A screenshot of a computer

Description automatically generated

4. Виконайте запити (**total\_table** – запит з п.3)

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

select count(tot.order\_id) as count\_rows from (total\_table) as tot

A screenshot of a computer

Description automatically generated

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

Якщо в декількох місцях змінити INNER на LEFT чи RIGHT, то в підсумку кількість рядків не зміниться (так само 518 записів):

A screenshot of a computer

Description automatically generated

Це пояснюється тим, що при кожному наступному з’єднані таблиць виникає відсутність відповідних значень.

Як приклад, на етапі приєднання

RIGHT JOIN customers AS cust ON ord.customer\_id = cust.id

маємо 535 записів,

A screenshot of a computer

Description automatically generated

але при наступному з’єднані

RIGHT JOIN employees AS emp ON ord.employee\_id = emp.employee\_id

Кількість записів зменшується через відсутність відповідних значень у таблиці employees.

A screenshot of a computer

Description automatically generated

Різниця:

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

4.6. Відсортуйте рядки за спаданням кількості рядків.

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

SELECT categories\_name, count(categories\_name) as count\_categ, avg(quantity) as avg\_quantity from (

SELECT

det.order\_id,

ord.date,

det.quantity,

det.product\_id,

prod.name as prod\_name,

cat.name as categories\_name,

cat.description,

prod.unit,

prod.price,

supp.name as suppliers\_name,

supp.contact,

supp.address,

supp.city,

supp.postal\_code,

supp.country,

supp.phone as phone\_suppliers,

ord.customer\_id,

cust.name as customer\_name,

cust.contact as customer\_contact,

cust.address as customer\_address,

cust.city as customer\_city,

cust.postal\_code as customer\_postal\_code,

cust.country as customer\_country,

emp.\*,

ord.shipper\_id,

ship.name as shipper\_name,

ship.phone as shipper\_phone

FROM orders as ord

INNER JOIN order\_details as det

ON ord.id = det.order\_id

INNER JOIN customers as cust

ON ord.customer\_id = cust.id

INNER JOIN employees as emp

ON ord.employee\_id = emp.employee\_id

INNER JOIN shippers as ship

ON ord.shipper\_id = ship.id

INNER JOIN products as prod

ON det.product\_id = prod.id

INNER JOIN suppliers as supp

ON prod.supplier\_id = supp.id

INNER JOIN categories as cat

ON prod.category\_id = cat.id) as tot

WHERE tot.employee\_id > 3 and tot.employee\_id <= 10

GROUP BY categories\_name

HAVING avg(quantity) > 21

ORDER BY avg\_quantity desc

LIMIT 4 OFFSET 1;