Завдання 1

create schema `LibraryManagement`;

create table `LibraryManagement`.`authors` (

`author\_id` int auto\_increment primary key,

`author\_name` varchar(255) not null

);

create table LibraryManagement.genres (

genre\_id int primary key auto\_increment,

genre\_name varchar(45) not null

);

create table LibraryManagement.books (

book\_id int primary key auto\_increment,

title varchar(45) not null,

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 LibraryManagement.users (

user\_id int auto\_increment primary key,

username varchar(45),

email varchar(45)

);

create table LibraryManagement.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),

foreign key (user\_id) references users(user\_id)

);

Завдання 2

insert into LibraryManagement.authors (author\_name) values

('Tolkien'),

('Zelazny'),

('Herbert');

insert into LibraryManagement.genres (genre\_name) values

('fantastic'), ('fantasy'), ('adventures');

insert into LibraryManagement.books (title, publication\_year, author\_id, genre\_id) values

('Lord of the Rings', 1954, 1, 1),

('The Chronicles of Amber', 1970, 2, 2),

('Dune', 1965, 3, 3);

insert into LibraryManagement.users (username, email) values

("Alex", "alex@alex"),

("Mike", "mike@mike"),

("Scott", "scott@scott");

insert into LibraryManagement.borrowed\_books (book\_id, user\_id, borrow\_date, return\_date) values

(1, 1, '2024-03-22', '2024-03-31'),

(2, 2, '2024-03-22', '2024-03-31'),

(3, 3, '2024-03-22', '2024-03-31');

Завдання 3

SELECT \*

FROM orders

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

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

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

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

Завдання 4.1.

select count(orders.id)

FROM orders

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

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

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

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

4.2.

Кількість рядків не зміниться, оскільки у колонок по яким встановлено зв’язки таблиць мають заповнені всі значення – inner join включає лише ті результати де встановлено співпадіння між таблицями, а left join всі дані лівої таблиці будуть включені у виборку незалежно від наявності співпадінь з правою.

SELECT count(orders.id)

FROM orders

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

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

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

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

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

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

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

4.3.

SELECT \*

FROM orders

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

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

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

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

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

4.4.

SELECT categories.name, COUNT(\*) AS row\_count, AVG(order\_details.quantity) AS average\_quantity

FROM orders

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

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

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

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

group by categories.name;

4.5.

SELECT categories.name, COUNT(\*) AS row\_count, AVG(order\_details.quantity) AS average\_quantity

FROM orders

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

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

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

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

group by categories.name

having average\_quantity > 21

4.6.

SELECT categories.name, COUNT(\*) AS row\_count, AVG(order\_details.quantity) AS average\_quantity

FROM orders

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

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

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

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

group by categories.name

having average\_quantity > 21

order by row\_count desc

4.7

SELECT categories.name, COUNT(\*) AS row\_count, AVG(order\_details.quantity) AS average\_quantity

FROM orders

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

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

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

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

group by categories.name

having average\_quantity > 21

order by row\_count desc

limit 4

offset 1