SQL Commands HW4

Task 1

1. create table authors   
   (  
    author\_id int auto\_increment primary key,   
    author\_name varchar(45)  
   )
2. create table geners

(

genre\_id int auto\_increment primary key,

genre\_name varchar(45)

)

1. create table books

(

book\_id int auto\_increment primary key,

title varchar(45),

publication\_year year,

author\_id int,

foreign key (author\_id) references authors(author\_id),

genre\_id int,

foreign key (genre\_id) references geners(genre\_id)

)

1. create table users

(

user\_id int auto\_increment primary key,

username varchar(45),

email varchar(128)

)

1. create table borrowed\_books

(

borrow\_id int auto\_increment primary key,

book\_id int,

foreign key (book\_id) references books(book\_id),

user\_id int,

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

borrow\_date date,

retrun\_date date

)

Task 3.

SELECT \* FROM orders

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

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

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

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

INNER JOIN products p ON p.id = o\_d.product\_id

INNER JOIN categories c ON c.id = p.category\_id

INNER JOIN suppliers su ON su.id = p.supplier\_id;

Task 4.

1.

SELECT COUNT(\*) AS all\_rows FROM orders

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

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

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

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

INNER JOIN products p ON p.id = o\_d.product\_id

INNER JOIN categories c ON c.id = p.category\_id

INNER JOIN suppliers su ON su.id = p.supplier\_id;

2.

SELECT COUNT(\*) AS all\_rows FROM orders

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

RIGHT JOIN employees em ON em.employee\_id = orders.employee\_id

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

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

INNER JOIN products p ON p.id = o\_d.product\_id

INNER JOIN categories c ON c.id = p.category\_id

INNER JOIN suppliers su ON su.id = p.supplier\_id;

При заміні полів на LEFT та RIGHT кількість полів не змінилася, тому, що у вибірці всі зв’язки мають значення. У даному запиті всі з’єднання базуються на ідентифікаторах, які відповідають значенням у таблицях. Але якби в деяких випадках зв’язок між таблицями був би відсутній, то заміна призвела б до різниці в кількості рядків у таблиці.

3.

SELECT COUNT(\*) AS all\_rows FROM orders

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

RIGHT JOIN employees em ON em.employee\_id = orders.employee\_id

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

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

INNER JOIN products p ON p.id = o\_d.product\_id

INNER JOIN categories c ON c.id = p.category\_id

INNER JOIN suppliers su ON su.id = p.supplier\_id

WHERE em.employee\_id > 3 AND em.employee\_id < 11;

4.

SELECT c.name category\_name, AVG(o\_d.quantity), COUNT(\*) AS all\_rows FROM orders

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

RIGHT JOIN employees em ON em.employee\_id = orders.employee\_id

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

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

INNER JOIN products p ON p.id = o\_d.product\_id

INNER JOIN categories c ON c.id = p.category\_id

INNER JOIN suppliers su ON su.id = p.supplier\_id

WHERE em.employee\_id > 3 AND em.employee\_id < 11

GROUP BY c.name

5.

SELECT c.name category\_name, AVG(o\_d.quantity) as total\_avg, COUNT(\*) AS all\_rows FROM orders

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

RIGHT JOIN employees em ON em.employee\_id = orders.employee\_id

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

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

INNER JOIN products p ON p.id = o\_d.product\_id

INNER JOIN categories c ON c.id = p.category\_id

INNER JOIN suppliers su ON su.id = p.supplier\_id

WHERE em.employee\_id > 3 AND em.employee\_id < 11

GROUP BY c.name

HAVING total\_avg > 21

6.

SELECT c.name category\_name, AVG(o\_d.quantity) as total\_avg, COUNT(\*) AS all\_rows FROM orders

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

RIGHT JOIN employees em ON em.employee\_id = orders.employee\_id

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

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

INNER JOIN products p ON p.id = o\_d.product\_id

INNER JOIN categories c ON c.id = p.category\_id

INNER JOIN suppliers su ON su.id = p.supplier\_id

WHERE em.employee\_id > 3 AND em.employee\_id < 11

GROUP BY c.name

HAVING total\_avg > 21

ORDER BY all\_rows DESC

7.

SELECT c.name category\_name, AVG(o\_d.quantity) as total\_avg, COUNT(\*) AS all\_rows FROM orders

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

RIGHT JOIN employees em ON em.employee\_id = orders.employee\_id

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

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

INNER JOIN products p ON p.id = o\_d.product\_id

INNER JOIN categories c ON c.id = p.category\_id

INNER JOIN suppliers su ON su.id = p.supplier\_id

WHERE em.employee\_id > 3 AND em.employee\_id < 11

GROUP BY c.name

HAVING total\_avg > 21

ORDER BY all\_rows DESC

LIMIT 4 OFFSET 1;