**4.1**

USE mydb;

SELECT COUNT (\*)

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;

**4.2**

USE mydb;

SELECT COUNT(\*) AS row\_count

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 не змінилася тому що всі рядки однієї таблиці мають відповідники в іншій таблиці.

**4.3**

USE mydb;

SELECT COUNT(\*) AS row\_count

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

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

**4.4**

USE mydb;

SELECT categories. name AS category\_name,

COUNT (\*) AS row\_count,

AVG (order\_details quantity) AS average\_quantity

FROM order\_details

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

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

GROUP BY categories.name;

**4.5**

USE mydb;

SELECT categories. name AS category\_name,

COUNT (\*) AS row\_count,

AVG (order\_details quantity) AS average\_quantity

FROM order\_details

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

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

GROUP BY categories.name

HAVING AVG(order\_details.quantity) > 21;

**4.6**

USE mydb;

SELECT categories.name AS category\_name,

COUNT (\*) AS row\_count,

AVG (order\_details.quantity) AS average\_quantity

FROM order\_details

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

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

GROUP BY categories.name

HAVING AVG(order\_details.quantity) > 21

ORDER BY row\_count DESC;

**4.7**

USE mydb;

SELECT categories.name AS category\_name,

COUNT (\*) AS row\_count,

AVG (order\_details quantity) AS average\_quantity

FROM order\_details

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

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

GROUP BY categories. name

HAVING AVG (order\_details.quantity) > 21

ORDER BY row\_count DESC

LIMIT 4 OFFSET 1;