```
4.1    SELECT COUNT(*) AS row_count FROM mydb.order_details
INNER JOIN orders ON orders.id = order_details.order_id
INNER JOIN products ON products.id = order_details.product_id
INNER JOIN customers ON customers.id = orders.customer_id
INNER JOIN employees ON employees.employee_id = orders.employee_id
INNER JOIN shippers ON shippers.id = orders.shipper_id
INNER JOIN suppliers ON suppliers.id = products.supplier_id
INNER JOIN categories ON categories.id = products.category_id

4.2    SELECT COUNT(*) AS row_count FROM mydb.order_details
LEFT JOIN orders ON orders.id = order_details.order_id
LEFT JOIN products ON products.id = order_details.product_id
INNER JOIN customers ON customers.id = orders.customer_id
INNER JOIN employees ON employees.employee_id = orders.employee_id
INNER JOIN shippers ON shippers.id = orders.shipper_id
LEFT JOIN suppliers ON suppliers.id = products.supplier_id
LEFT JOIN categories ON categories.id = products.category_id;

4.3    SELECT * FROM mydb.order_details
INNER JOIN orders ON orders.id = order_details.order_id
INNER JOIN products ON products.id = order_details.product_id
INNER JOIN customers ON customers.id = orders.customer_id
INNER JOIN employees ON employees.employee_id = orders.employee_id
INNER JOIN shippers ON shippers.id = orders.shipper_id
INNER JOIN suppliers ON suppliers.id = products.supplier_id
INNER JOIN categories ON categories.id = products.category_id
WHERE orders.employee_id > 3 AND orders.employee_id < 10

4.4    SELECT categories.name, COUNT(*) AS total_row, AVG(order_details.quantity) AS
avg_quantity
FROM mydb.order_details
INNER JOIN orders ON orders.id = order_details.order_id
INNER JOIN products ON products.id = order_details.product_id
INNER JOIN customers ON customers.id = orders.customer_id
INNER JOIN employees ON employees.employee_id = orders.employee_id
INNER JOIN shippers ON shippers.id = orders.shipper_id
INNER JOIN suppliers ON suppliers.id = products.supplier_id
INNER JOIN categories ON categories.id = products.category_id
WHERE orders.employee_id > 3 AND orders.employee_id < 10
GROUP BY categories.name;
4.5    HAVING avg_quantity > 21
4.6    ORDER BY total_row DESC
4.7    LIMIT 4
       OFFSET 1
```