1.a. CREATE SCHEMA LibraryManagement;

1.b. USE LibraryManagement;

```
CREATE TABLE authors (
author_id INT PRIMARY KEY AUTO_INCREMENT,
 author_name VARCHAR(45)
);
```

1.c. CREATE TABLE genres (

```
genre_id INT PRIMARY KEY AUTO_INCREMENT,
 genre_name VARCHAR(45)
);
```

1.d. CREATE TABLE books (

```
book_id INT AUTO_INCREMENT PRIMARY KEY,
title VARCHAR(45),
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)
);
```

1.e. CREATE TABLE users (

```
user_id INT PRIMARY KEY AUTO_INCREMENT,
username VARCHAR(45),
email VARCHAR(254)
);
```

1.f. CREATE TABLE borrowed_books (

```
borrow_id INT PRIMARY KEY AUTO_INCREMENT,
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)
```

);

2.a. INSERT INTO authors(author_name)

VALUES ("peter peterson"), ("oscar wilde");

2.b. INSERT INTO genres(genre_name)

VALUES ("horror"), ("romance");

2.c. INSERT INTO users(username, email)

VALUES

('john_doe', 'john.doe@example.com'),

('jane_smith', 'jane.smith@example.com'),

('alice_jones', 'alice.jones@example.com'),

('bob_brown', 'bob.brown@example.com'),

('charlie_black', 'charlie.black@example.com');

2.d. INSERT INTO books(title, publication_year, author_id, genre_id)

VALUES

("Rising from the dead", 1644, 1, 1),

("A lovely love", 1994, 2, 2);

2.e. INSERT INTO borrowed_books(book_id, user_id, borrow_date, return_date)

VALUES

(1, 3, "2024-04-21", "2024-05-31"),

(2, 5, "2024-01-11", "2024-04-27");

3. SELECT

order_details.id AS order_detail_id,

order_details.quantity,

orders.id AS order_id,

orders.date AS order_date,

customers.id AS customer_id,

customers.name AS customer_name,

customers.address AS customer_address,

customers.city AS customer_city,

customers.postal_code AS customer_postal_code,

customers.country AS customer_country,

products.id AS product_id,

```
    products.name AS product_name,
    products.unit AS product_unit,
    products.price AS product_price,
  categories.id AS category_id,
    categories.name AS category_name,
    categories.description AS category_description,
  employees.employee_id,
    employees.last_name AS employee_last_name,
    employees.first_name AS employee_first_name,
    employees.birthdate AS employee_birthdate,
    employees.photo AS employee_photo,
    employees.notes AS employee_notes,
  shippers.id AS shipper_id,
    shippers.name AS shipper_name,
    shippers.phone AS shipper_phone,
  suppliers.id AS supplier_id,
    suppliers.name AS supplier_name,
    suppliers.contact AS supplier_contact,
    suppliers.address AS supplier_address,
    suppliers.city AS supplier_city,
    suppliers.postal_code AS supplier_postal_code,
    suppliers.phone AS supplier_phone
FROM order_details
INNER JOIN orders ON order_details.order_id = orders.id
INNER JOIN customers ON customers.id = orders.customer_id
INNER JOIN products ON products.id = order_details.product_id
INNER JOIN categories ON categories.id = products.category_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;
```

4.a. SELECT COUNT(*) AS total_rows
FROM order_details

```
INNER JOIN orders ON order_details.order_id = orders.id
INNER JOIN customers ON customers.id = orders.customer_id
INNER JOIN products ON products.id = order_details.product_id
INNER JOIN categories ON categories.id = products.category_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;
```

4.b.

```
SELECT COUNT(*) AS total_rows
FROM order_details
LEFT JOIN orders ON order_details.order_id = orders.id
LEFT JOIN customers ON customers.id = orders.customer_id
LEFT JOIN products ON products.id = order_details.product_id
INNER JOIN categories ON categories.id = products.category_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;
```

Всі первинні ключі в таблицях зазначені як обов’язкові – Not Null, тому кількість рядків збігається

4.c.

```
SELECT COUNT(*) AS total_rows
FROM order_details
INNER JOIN orders ON order_details.order_id = orders.id
INNER JOIN customers ON customers.id = orders.customer_id
INNER JOIN products ON products.id = order_details.product_id
INNER JOIN categories ON categories.id = products.category_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
WHERE employees.employee_id > 3 AND employees.employee_id <= 10;
```

4.d.

```
SELECT
  categories.name AS category_name,
  COUNT(*) AS total,
  AVG(order_details.quantity) AS average_quantity
FROM order_details
```

```
INNER JOIN orders ON order_details.order_id = orders.id
INNER JOIN customers ON customers.id = orders.customer_id
INNER JOIN products ON products.id = order_details.product_id
INNER JOIN categories ON categories.id = products.category_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
GROUP BY category_name;
```

4.e.

```
SELECT *
FROM (
  SELECT
    order_details.*,
    AVG(order_details.quantity) OVER (PARTITION BY order_details.order_id) AS average_quantity
  FROM order_details
  INNER JOIN orders ON order_details.order_id = orders.id
  INNER JOIN customers ON customers.id = orders.customer_id
  INNER JOIN products ON products.id = order_details.product_id
  INNER JOIN categories ON categories.id = products.category_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
) subquery
WHERE average_quantity > 21;
```

4.f.

```
SELECT
 order_details.id AS order_detail_id,
  order_details.quantity,
 orders.id AS order_id,
  orders.date AS order_date,
 customers.id AS customer_id,
  customers.name AS customer_name,
  customers.address AS customer_address,
  customers.city AS customer_city,
```

```
    customers.postal_code AS customer_postal_code,
    customers.country AS customer_country,
  products.id AS product_id,
    products.name AS product_name,
    products.unit AS product_unit,
    products.price AS product_price,
  categories.id AS category_id,
    categories.name AS category_name,
    categories.description AS category_description,
  employees.employee_id,
    employees.last_name AS employee_last_name,
    employees.first_name AS employee_first_name,
    employees.birthdate AS employee_birthdate,
    employees.photo AS employee_photo,
    employees.notes AS employee_notes,
  shippers.id AS shipper_id,
    shippers.name AS shipper_name,
    shippers.phone AS shipper_phone,
  suppliers.id AS supplier_id,
    suppliers.name AS supplier_name,
    suppliers.contact AS supplier_contact,
    suppliers.address AS supplier_address,
    suppliers.city AS supplier_city,
    suppliers.postal_code AS supplier_postal_code,
    suppliers.phone AS supplier_phone
FROM order_details
INNER JOIN orders ON order_details.order_id = orders.id
INNER JOIN customers ON customers.id = orders.customer_id
INNER JOIN products ON products.id = order_details.product_id
INNER JOIN categories ON categories.id = products.category_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
ORDER BY order_details.quantity DESC;
```

4.g.

```
SELECT
  order_details.id AS order_detail_id,
    order_details.quantity,
  orders.id AS order_id,
    orders.date AS order_date,
  customers.id AS customer_id,
    customers.name AS customer_name,
    customers.address AS customer_address,
    customers.city AS customer_city,
    customers.postal_code AS customer_postal_code,
    customers.country AS customer_country,
  products.id AS product_id,
    products.name AS product_name,
    products.unit AS product_unit,
    products.price AS product_price,
  categories.id AS category_id,
    categories.name AS category_name,
    categories.description AS category_description,
  employees.employee_id,
    employees.last_name AS employee_last_name,
    employees.first_name AS employee_first_name,
    employees.birthdate AS employee_birthdate,
    employees.photo AS employee_photo,
    employees.notes AS employee_notes,
  shippers.id AS shipper_id,
    shippers.name AS shipper_name,
    shippers.phone AS shipper_phone,
  suppliers.id AS supplier_id,
    suppliers.name AS supplier_name,
    suppliers.contact AS supplier_contact,
```

```
    suppliers.address AS supplier_address,
    suppliers.city AS supplier_city,
    suppliers.postal_code AS supplier_postal_code,
    suppliers.phone AS supplier_phone
FROM order_details
INNER JOIN orders ON order_details.order_id = orders.id
INNER JOIN customers ON customers.id = orders.customer_id
INNER JOIN products ON products.id = order_details.product_id
INNER JOIN categories ON categories.id = products.category_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
LIMIT 4 OFFSET 1;
```