

1.

b

```
CREATE TABLE authors
(
author_id int primary key auto_increment, author_name varchar(45)
)
```

c.

```
CREATE TABLE genres
(
genre_id int primary key auto_increment, genre_name varchar(45)
)
```

d

```
CREATE TABLE books
(
book_id int primary key auto_increment, 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)
)
```

e.

```
CREATE TABLE users
(
user_id int primary key auto_increment, username varchar(45), email varchar(45)
)
```

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.

```
INSERT INTO users ( username, email)
VALUES ("romantic"), ("horror"), ("history");
```

```
INSERT INTO genres ( genre_name)
VALUES ("romantic"), ("horror"), ("history");
```

```
INSERT INTO books ( title, publication_year, author_id, genre_id)
```

```
VALUES (1, 'book1', 1998, 1, 1);
```

3.

```
USE mydb;
SELECT * From orders o
INNER JOIN customers c ON o.customer_id = c.id
INNER JOIN order_details od ON o.id = od.order_id
INNER JOIN products p ON od.product_id = p.id
INNER JOIN employees e ON o.employee_id = e.employee_id
INNER JOIN shippers sh ON o.shipper_id = sh.id
INNER JOIN suppliers su on p.supplier_id = su.id
INNER JOIN categories ca on p.category_id = ca.id
```

4. a.a

```
USE mydb;
SELECT count(*) From orders o
INNER JOIN customers c ON o.customer_id = c.id
INNER JOIN order_details od ON o.id = od.order_id
INNER JOIN products p ON od.product_id = p.id
INNER JOIN employees e ON o.employee_id = e.employee_id
INNER JOIN shippers sh ON o.shipper_id = sh.id
INNER JOIN suppliers su on p.supplier_id = su.id
INNER JOIN categories ca on p.category_id = ca.id
```

b.a

```
USE mydb;
SELECT COUNT(*) as rows_cont From orders o
LEFT JOIN customers c ON o.customer_id = c.id
INNER JOIN order_details od ON o.id = od.order_id
INNER JOIN products p ON od.product_id = p.id
LEFT JOIN employees e ON o.employee_id = e.employee_id
LEFT JOIN shippers sh ON o.shipper_id = sh.id
INNER JOIN suppliers su on p.supplier_id = su.id
LEFT JOIN categories ca on p.category_id = ca.id
Count 518
```

b.b (because was interesting)

```
USE mydb;
SELECT * From customers c
LEFT JOIN orders o ON o.customer_id = c.id
INNER JOIN order_details od ON o.id = od.order_id
```

```
INNER JOIN products p ON od.product_id = p.id
LEFT JOIN employees e ON o.employee_id = e.employee_id
LEFT JOIN shippers sh ON o.shipper_id = sh.id
INNER JOIN suppliers su on p.supplier_id = su.id
LEFT JOIN categories ca on p.category_id = ca.id
Also 518 rows
```

```
USE mydb;
SELECT * From customers c
LEFT JOIN orders o ON o.customer_id = c.id
INNER JOIN order_details od ON o.id = od.order_id
INNER JOIN products p ON od.product_id = p.id
LEFT JOIN employees e ON o.employee_id = e.employee_id
LEFT JOIN shippers sh ON o.shipper_id = sh.id
INNER JOIN suppliers su on p.supplier_id = su.id
LEFT JOIN categories ca on p.category_id = ca.id
Also 518 rows
```

Кількість рядків не змінюється тому що паа ключ є унікальною немає повторів які можна було б пропустити.

В загальному

Зміна операторів може призвести до зміни кількості рядків. Тому що INNER JOIN повертає тільки ті рядки, які мають відповідний запис у кожній з об'єднаних таблиць.

Якщо якісь з об'єднаних таблиць не мають відповідних записів, вони будуть відсутні.

LEFT JOIN повертає всі рядки з лівої таблиці і відповідні рядки з правої таблиці. Якщо в правій немає відповідного запису, буде NULL.

RIGHT JOIN повертає всі рядки з правої таблиці і відповідні рядки з лівої таблиці. Якщо в лівій таблиці немає відповідного запису, буде NULL.

```
USE mydb;
SELECT categories.name as category,
COUNT(*) AS row_amount,
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 average_quantity > 21
ORDER BY row_amount desc
LIMIT 4 OFFSET 1;
```