SQL Queries:

use book_store;

-- 1

select * from books;

select title,in_stock from books where title = 'Leaves of Grass' and in_stock > 0;

-- 2

SELECT customer.first_name, min(order_date) FROM book_store.books_transaction

inner join customer

on

books_transaction.customer_id1 = customer.id;

-- 3

select books.title,min(date_reseived) from books;

-- Querie 4

-- 4

select * from books_transaction where order_status<>"delivered" ORDER BY order_date desc ;

-- 5

select sum(quantity) from books_transaction

inner join books on books.id = books_transaction.books_id where books.title ='In Search of Lost Time';

-- 6

select writer.name_writer as books_amount from books

inner join books_transaction

on

        books.id = books_transaction.books_id

inner join writer_has_books

on

       writer_has_books.books_id = books_transaction.books_id

inner join writer

on

      writer.id = writer_has_books.writer_id

WHERE books_transaction.order_date >= '2019-01-01' and books_transaction.order_date < '2019-10-01'

GROUP BY writer.id

HAVING COUNT(writer.id) > 1

order by COUNT(writer.id) desc limit 1;

==-- Querie 7==

-- 7

select customer.first_name,books_transaction.quantity from books_transaction

inner join customer

on

      books_transaction.customer_id1 = customer.id

ORDER BY quantity desc

LIMIT 3;

==-- Querie 8==

-- 8

select title from books

inner join books_has_languages

 on

 books.id = books_has_languages.books_id

 inner join languages

 on

 books_has_languages.books_id = languages.id

 WHERE books.in_stock > 0

```
 group by books.id

 HAVING COUNT(books.id) > 0

 order by count( books.id) desc limit 1;
```

```
-- 9

select customer.first_name,customer.last_name,
books.title,books_transaction.order_date,books.price from books_transaction

inner join books

on

        books_transaction.books_id = books.id

inner join customer

on

        customer.id = books_transaction.customer_id1

WHERE  customer.first_name = 'fofo' and customer.last_name = 'sese'

order by order_date;
```

```
-- 10

select customer.first_name,customer.last_name, books_transaction.order_date,books.title,
books_transaction.quantity,books.price,books.in_stock,books_transaction.total from
books_transaction

inner join books

on

        books_transaction.books_id = books.id

inner join customer

on

        customer.id = books_transaction.books_id

WHERE  customer.first_name = 'fofo' and customer.last_name = 'sese'

ORDER BY order_date desc ;
```

-- 11

use book_store;

select books_transaction.order_number, books.price,books.weight,books_transaction.quantity,deliveries.name_1,deliveries_methods.name_2,deliveries_methods.price,((books.price + books.weight) * (books_transaction.quantity) + (deliveries_methods.price)) AS total_price from books_transaction

 inner join books

 on books_transaction.books_id = books.id

 inner join deliveries

 on deliveries.deliveries_id = books_transaction.books_id

 inner join deliveries_methods

 on deliveries_methods.delivery_method_id = books.id;

-- 12

select customer.first_name, customer.last_name, books.title, books_transaction.quantity, books_transaction.order_number, books_transaction.order_date from books

 inner join books_transaction

 on books.id = books_transaction.books_id

 inner join customer

 on books_transaction.customer_id1 = customer.id

 where customer.first_name = 'fofo' and customer.last_name = 'sese';

-- 13

select order_status from books_transaction where id = 12444;

-- 14

SELECT count(MONTH(order_date)) as num_delivery_by_Xpress_fer_month,MONTH(order_date) from books_transaction

inner join deliveries

on

       deliveries.deliveries_id = books_transaction.deliveries_deliveries_id1

inner join deliveries_methods

on

       deliveries_methods.delivery_method_id = books_transaction.deliveries_deliveries_id1

WHERE deliveries.name_1 = 'Xpress' and MONTH(order_date)=8

order by order_date desc;

-- 15 7,10 אפשר לבדוק בחודש

SELECT
books_transaction.method,books_transaction.total,MONTH(order_date),books_transaction.total,count(total), sum(total) from books_transaction

inner join books

on

       books.id = books_transaction.books_id

WHERE books_transaction.method ='bit_app' and MONTH(order_date)= 07;

-- 16 just books_transaction

select year(order_date),sum(total) from books_transaction

WHERE books_transaction.order_date >= '2019-01-01' and books_transaction.order_date <= '2020-01-01';

-- 17

SELECT deliveries.name_1 as DeliveriesName,count(deliveries.name_1) as amount from deliveries

inner join books_transaction

on

       deliveries.deliveries_id = books_transaction.deliveries_deliveries_id1

WHERE order_date between '2019-01-01' and now()

GROUP BY deliveries.name_1

HAVING COUNT(deliveries.name_1) >= 1

order by deliveries.name_1 desc;

-- 18

select
books_transaction.order_date,books_transaction.order_number,publication.year_of_publicatio
n,books_transaction.quantity,books_transaction.total,(books.id) as booksId(P booksId id)

from books

inner join books_has_publication

on books.id = books_has_publication.books_id

inner join publication

on books_has_publication.publication_id = publication.id

inner join books_transaction

on

books.id = books_transaction.books_id

where books_transaction.order_date = '2013-12-11';

-- 19

SELECT
customer.id,books_transaction.order_date,customer.first_name,customer.last_name,customer.
lines_phone from customer

 inner join books_transaction

on

        customer.id = books_transaction.customer_id1

WHERE books_transaction.order_date < '2018-07-28';

-- 20

select (shipped_date),customer.first_name, customer.last_name from books_transaction

inner join customer

on

  books_transaction.customer_id1 = customer.id

  where books_transaction.order_date < books_transaction.shipped_date;

-- 21

SELECT month(date_reseived) as month,(in_stock) as total_books_in_stock FROM books

WHERE year(date_reseived) = '2019'

group by month(date_reseived);

-- 22

SELECT sum(total) as total_paid, sum(quantity) as total_num_of_books FROM books_transaction

WHERE order_date >= '2019-12-01' and order_date < '2019-12-01';

-- 22B

SELECT MONTH(month_payment) as month,(price), (expenses_total), ((price) - (expenses_total)) AS store_profit FROM store_payments

inner join store_order

     WHERE MONTH(month_payment) = 04 and YEAR(month_payment) = 2020 and MONTH(order_date) = 04 and YEAR(order_date) = 2020

  GROUP BY store_payments.store_id

order by (store_payments.store_id) desc limit 1;

-- 23

SELECT month(order_date) as month,(total) as total FROM books_transaction

WHERE year(order_date) = '2019'

group by month(order_date);

-- 24

SELECT employee.first_name,(MONTH(month)) as salary_month,(salary) from salary_per_month

inner join employee

on

        salary_per_month.id = employee.employee_id

WHERE MONTH(month)=10 or employee.first_name= 'uouo'

order by month desc;

-- 25

select employee.first_name, employee.last_name,employee.employee_id from employee

inner join books_transaction

on

        employee.employee_id = books_transaction.employee_employee_id

WHERE MONTH(order_date)=10

GROUP BY employee.employee_id

order by COUNT(employee.employee_id) desc limit 1;