/*1. Show all available books in inventory*/
SELECT *
FROM books
LEFT JOIN inventory ON books.book_id = inventory.book_id
WHERE inventory.amount > 0;

/*2. Show all open orders*/
SELECT order_id, order_date, cust_id, first_name, suppliers.supp_id, supp_name, book_id, title, amount, tot_price, stat_type
FROM (
        SELECT order_id, order_date, customers.cust_id, first_name, supp_id, book_id, title, amount, tot_price, stat_type
        FROM (
                SELECT order_id, order_date, cust_id, supp_id, books.book_id, title, amount, tot_price, stat_type, stat_order.status_id
                FROM (
                        SELECT order_id, order_date, cust_id, supp_id, book_id, amount, tot_price, stat_type, orders.status_id
                        FROM orders
                        LEFT JOIN order_status ON orders.status_id = order_status.status_id
                        WHERE (cust_id AND orders.status_id != 5 AND orders.status_id != 6) OR
                                (cust_id IS NULL AND orders.status_id != 3 AND orders.status_id != 6)
) AS stat_order
                LEFT JOIN books ON books.book_id = stat_order.book_id
) AS book_info
        LEFT JOIN customers ON book_info.cust_id = customers.cust_id
) AS cust_info
LEFT JOIN suppliers ON cust_info.supp_id = suppliers.supp_id;

/*3. List of all customers who made a purchase*/
SELECT customers.cust_id, first_name, last_name, phone
FROM (
        SELECT cust_id
        FROM purchases
        WHERE canceled = false
        GROUP BY cust_id
) AS cust_who_bought
LEFT JOIN customers ON customers.cust_id = cust_who_bought.cust_id;

/*4. Show all suppliers*/
SELECT *
FROM suppliers;

/*5. Show all purchases between given dates: fromDate & tilDate*/
SELECT purch_id, book_id, title, seller_id, seller_name, customers.cust_id, customers.first_name, purch_date, canceled, cust_pay
FROM (
        SELECT purch_id, book_id, title, sellers.seller_id, sellers.first_name seller_name, cust_id, purch_date, canceled, cust_pay
        FROM (
                SELECT purch_id, purch.book_id, books.title, seller_id, cust_id, purch_date, canceled, cust_pay
                FROM (
                        SELECT purch_id, book_id, seller_id, cust_id, purch_date, IF(canceled = 0 , 'No', 'Yes') AS canceled, origin_price, cust_pay
                        FROM purchases
                        WHERE (purch_date BETWEEN 'fromDate' AND 'tilDate')
) AS purch
                LEFT JOIN books ON books.book_id = purch.book_id
) AS answer
        LEFT JOIN sellers ON answer.seller_id = sellers.seller_id
) AS ans_with_sellername
LEFT JOIN customers ON customers.cust_id =ans_with_sellername.cust_id
ORDER BY purch_id;

/*6. Show all books available for global sale*/
SELECT book_id, title, author_name, original_price, disc_price
FROM (
        SELECT books.book_id, title, author_name, price AS original_price, (price*(100-discount)/100) AS disc_price, discount
        FROM store_items
        LEFT JOIN books ON store_items.book_id=books.book_id
) AS books_info
WHERE discount > 0;

/*7. Check if a given book: bookTitle + bookAuthor is available in the inventory*/
SELECT inventory.book_id, title, author_name, amount
FROM (
        SELECT book_id, title, author_name
        FROM books
        WHERE title = 'bookTitle' AND author_name = 'bookAuthor'
) AS search_book
LEFT JOIN inventory ON search_book.book_id = inventory.book_id;

/*8. List of all suppliers of a given book: bookTitle + bookAuthor*/
SELECT suppliers.supp_id, supp_name, phone, bank_acc, book_id, price
FROM (
       SELECT book_prices.book_id, supp_id, price
       FROM (
              SELECT book_id
              FROM books
              WHERE title = 'bookTitle' AND author_name = 'bookAuthor'
) AS search_book
       LEFT JOIN book_prices ON search_book.book_id = book_prices.book_id
) AS supplied_books
LEFT JOIN suppliers ON suppliers.supp_id = supplied_books.supp_id;

/*9. How many books from type: bookTitle + bookAuthor where sold from date fromDate*/
SELECT COUNT(book_id) AS books_sold, title, author_name
FROM (
       SELECT purch_id, purchases.book_id, title, author_name, purch_date, canceled
       FROM (
              SELECT book_id, title, author_name
              FROM books
              WHERE title = 'bookTitle' AND author_name = 'bookAuthor'
) AS search_book
       LEFT JOIN purchases ON purchases.book_id = search_book.book_id
       WHERE purch_date >= 'fromDate' AND canceled = false
) AS books_purch;

/*10. How many books and different book types were purchased by customer: custID since given date fromDate*/
SELECT COUNT(book_id) AS tot_books, COUNT(DISTINCT book_id) AS diff_books
FROM (
       SELECT *
       FROM purchases
       WHERE cust_id = 'custID' AND canceled = false AND purch_date >= 'fromDate'
) AS cust_purch;

/*11. Show the customer details who bought the most since given date fromDate*/
```
SELECT top_cust.cust_id, book_amount, first_name, last_name, phone
FROM (
        SELECT *
        FROM (
                SELECT cust_id, COUNT(cust_id) book_amount
                FROM (
                        SELECT *
                        FROM purchases
                        WHERE canceled = false AND purch_date >= 'fromDate'
                ) AS purch
                GROUP BY cust_id
        ) AS custs_books
        ORDER BY book_amount DESC
        LIMIT 1
) AS top_cust
LEFT JOIN customers ON customers.cust_id = top_cust.cust_id;
```

/*12. show the supplier details who sold us the most books since given date fromDate*/
```
SELECT amount, max_supp.supp_id, supp_name, phone, bank_acc
FROM (
        SELECT *
        FROM (
                SELECT supp_id, SUM(amount) AS amount
                FROM orders
                WHERE order_date >= 'fromDate' AND  status_id != 1
                GROUP BY supp_id
        ) AS supps_amounts
        ORDER BY amount DESC
        LIMIT 1
) AS max_supp
LEFT JOIN suppliers ON suppliers.supp_id = max_supp.supp_id;
```

/*13. amount of orders made and how many books between given dates: fromDate & tilDate*/
```
SELECT COUNT(order_id) orders_amount, SUM(amount) books_amount_ordered
FROM (
        SELECT *
        FROM orders
        WHERE (order_date BETWEEN 'fromDate' AND 'tilDate') AND status_id != 1
) AS orders_range;
```

/*14. amount of orders made and how many books between given dates: fromDate & tilDate that were made by customers sold*/
SELECT COUNT(order_id) orders_amount, SUM(amount) books_amount_ordered
FROM (
       SELECT *
       FROM orders
       WHERE (order_date BETWEEN 'fromDate' AND 'tilDate') AND cust_id AND status_id = 5
) AS orders_range;

/*15. show the total discount a customer received since a certain date*/
SELECT SUM(discount) AS total_disc
FROM (
       SELECT origin_price - cust_pay AS discount
       FROM (
              SELECT *
              FROM purchases
              WHERE purch_date >= 'fromDate' AND cust_id = 'custID' AND canceled = false
) AS purch_disc
) AS disc_sum;

```
/*16. sum revenue in Q1, Q2, Q3, Q4 in a given year: y*/
SELECT *
FROM (
        SELECT *
        FROM (
                SELECT SUM(cust_pay) AS tot_Q1
                FROM purchases
                WHERE (purch_date BETWEEN 'y-01-01' AND 'y-03-31') AND canceled = false
        ) AS Q1
        LEFT JOIN (
                SELECT *
                FROM (
                        SELECT SUM(cust_pay) AS tot_Q2
                        FROM purchases
                        WHERE (purch_date BETWEEN 'y-04-01' AND 'y-06-30') AND canceled = false
                ) AS Q2
        ) Q1Q2 ON (tot_Q1 OR tot_Q1 IS NULL)
        LEFT JOIN (
                SELECT *
                FROM (
                        SELECT SUM(cust_pay) AS tot_Q3
                        FROM purchases
                        WHERE (purch_date BETWEEN 'y-07-01' AND 'y-09-30') AND canceled = false
                ) AS Q3
        ) Q1Q2Q3 ON (tot_Q1 OR tot_Q1 IS NULL)
        LEFT JOIN (
                SELECT *
                FROM (
                        SELECT SUM(cust_pay) AS tot_Q4
                        FROM purchases
                        WHERE (purch_date BETWEEN 'y-10-01' AND 'y-12-31') AND canceled = false
                ) AS Q4
        ) Q1Q2Q3Q4 ON (tot_Q1 OR tot_Q1 IS NULL)
) AS revenue;
```

/*17. Show how many customers were added from given date: fromDate*/
SELECT COUNT(cust_id) as new_custs
FROM (
      SELECT *
      FROM (
            SELECT *
            FROM (
                  SELECT cust_id, purch_date
                  FROM purchases
                  WHERE canceled = false
) AS cust_purch_date
            ORDER BY purch_date ASC
) AS ordered_date
      GROUP BY cust_id
) AS count_cust
WHERE purch_date >= 'fromDate';

/*18. total amount paid to a given supplier: suppID, between given dates: fromDate, tilDate*/
SELECT SUM(tot_price) AS tot_shekels
FROM (
      SELECT *
      FROM orders
      WHERE supp_id = 'suppID' AND (order_date BETWEEN 'fromDate' AND 'tilDate') AND status_id != 1
) AS supp_orders;

/*19. total amount a given seller: sellID earned between given dates: fromDate, tilDate*/
SELECT SUM(cust_pay) AS tot_shekels
FROM (
      SELECT *
      FROM purchases
      WHERE seller_id = 'sellID' AND (purch_date BETWEEN 'fromDate' AND 'tilDate') AND canceled = false
) AS seller_purch;

/*20. top 10 most sold books between given dates: fromDate, tilDate*/
SELECT *
FROM (
       SELECT books.book_id, title, author_name, amount
       FROM (
              SELECT book_id, COUNT(*) AS amount
              FROM purchases
              WHERE (purch_date BETWEEN 'fromDate' AND 'tilDate') AND canceled = false
              GROUP BY book_id
) AS count_amount
   LEFT JOIN books ON books.book_id = count_amount.book_id
) AS books_order
ORDER BY amount DESC
LIMIT 10;