1,SELECT customer\_id, first\_name, last\_name, email, street, city

FROM customers

WHERE city = 'San Angelo'

ORDER BY first\_name DESC, last\_name ASC;  
  
2, SELECT order\_id, order\_date, required\_date, shipped\_date, store\_id

FROM orders

WHERE order\_date BETWEEN '2016-03-01' AND '2016-03-31'

AND shipped\_date > required\_date

ORDER BY order\_id ASC;  
  
3, SELECT p.product\_id, p.product\_name, p.model\_year, b.brand\_name, s.store\_name, st.quantity

FROM stocks st

JOIN products p ON st.product\_id = p.product\_id

JOIN brands b ON p.brand\_id = b.brand\_id

JOIN stores s ON st.store\_id = s.store\_id

WHERE b.brand\_name = 'Surly'

AND s.store\_name = 'Baldwin Bikes'

AND st.quantity > 27;  
  
4, SELECT s.store\_id, s.store\_name, COUNT(o.order\_id) AS Number\_Of\_Orders\_2016

FROM stores s

JOIN orders o ON s.store\_id = o.store\_id

WHERE YEAR(o.order\_date) = 2016

GROUP BY s.store\_id, s.store\_name

ORDER BY Number\_Of\_Orders\_2016 ASC;  
  
5, SELECT c.customer\_id, c.first\_name, c.last\_name, COUNT(o.order\_id) AS Number\_Of\_Orders

FROM customers c

JOIN orders o ON c.customer\_id = o.customer\_id

GROUP BY c.customer\_id, c.first\_name, c.last\_name

HAVING COUNT(o.order\_id) = (

SELECT MAX(order\_count)

FROM (

SELECT COUNT(o2.order\_id) AS order\_count

FROM orders o2

GROUP BY o2.customer\_id

) AS subquery

);  
  
6, SELECT s.store\_name, st.staff\_id, st.first\_name, st.last\_name, COUNT(o.order\_id) AS Number\_Of\_Orders

FROM stores s

JOIN staff st ON s.store\_id = st.store\_id

JOIN orders o ON st.staff\_id = o.staff\_id

GROUP BY s.store\_name, st.staff\_id, st.first\_name, st.last\_name

HAVING COUNT(o.order\_id) = (

SELECT MAX(order\_count)

FROM (

SELECT COUNT(o2.order\_id) AS order\_count

FROM orders o2

WHERE o2.staff\_id = st.staff\_id

GROUP BY o2.staff\_id

) AS subquery

);  
  
7, SELECT i.item\_id, b.brand\_name, AVG(i.quantity \* i.list\_price) AS Average\_Of\_Price

FROM items i

JOIN brands b ON i.brand\_id = b.brand\_id

GROUP BY i.item\_id, b.brand\_name

HAVING AVG(i.quantity \* i.list\_price) > 1900

ORDER BY Average\_Of\_Price ASC;  
  
8, CREATE PROCEDURE sp\_Num\_Staff

@store\_id INT,

@numberOfStaffs INT OUTPUT

AS

BEGIN

SELECT @numberOfStaffs = COUNT(\*)

FROM staff

WHERE store\_id = @store\_id;

END;  
  
9, CREATE TRIGGER Tr\_Delete\_Stocks

ON stocks

FOR DELETE

AS

BEGIN

SELECT d.product\_id, p.product\_name, d.store\_id, s.store\_name, d.quantity

FROM deleted d

JOIN products p ON d.product\_id = p.product\_id

JOIN stores s ON d.store\_id = s.store\_id;

END;  
  
10, UPDATE stocks

SET quantity = 18

FROM stocks s

JOIN products p ON s.product\_id = p.product\_id

WHERE s.store\_id = 2 AND p.category\_name = 'Electric Bikes';