1. **Basic** – Build a SQL statement and run it in DataGrip connected to MySQL that will provide the customer name, order number, order date, and product name for all orders placed later than March 15th, 2015 and the customer’s name **starts** with ‘Mini’. Returns 15 rows.

select customers.customername, orders.ordernumber, orders.orderdate, products.productname from

customers inner join orders on customers.customernumber = orders.customernumber

inner join orderdetails on orderdetails.ordernumber = orders.ordernumber

inner join products on orderdetails.productCode = products.productCode

where customers.customername like 'Mini%' and orders.orderdate > date('2015-03-15')

τcustomername, orders.orderNumber, orderDate, productName πcustomername, orders.orderNumber, orderDate, productName σcustomername like 'Mini%' ∧ orderDate > date('2015-03-15') (customers ⨝ customers.customerNumber = orders.customerNumber orders ⨝ OrderDetails.orderNumber = orders.orderNumber OrderDetails ⨝ OrderDetails.productCode = products.productCode products)

A screenshot of a computer

Description automatically generated with medium confidence

1. **Inline Query** – Modify **Basic** to use two **inline** queries: one that produces only the Customers that start with ‘Mini%’, and another that only produces the orders placed after 3/15/2015. Join these inline query results in just as though they were tables.

select A.customerName,B.orderNumber,B.orderDate,B.productName from (select customername,customerNumber from customers where customername like 'Mini%') as A inner join

(select orders.customernumber, orders.ordernumber,products.productName, orders.orderDate from orders inner join orderdetails

on orders.ordernumber = orderdetails.ordernumber inner join products on orderdetails.productCode = products.productCode

where orders.orderDate > date('2015-03-15')) as B

on A.customernumber = B.customerNumber

π A.customername, B.orderNumber, B.orderDate, B.productName ( ρ A ( π customername, customerNumber σ customername like 'Mini%' customers ) ⨝ A.customerNumber = B.customerNumber ρ B ( π orders.customerNumber, orders.orderNumber, products.productName, orders.orderDate σ orders.orderDate > length('2015-03-15') ( ( orders ⨝ orders.orderNumber = OrderDetails.orderNumber OrderDetails ) ⨝ OrderDetails.productCode = products.productCode products ) ) )Diagram

Description automatically generated

3. **Parenthesis** – Modify the **Basic** statement to use parenthesis to change the join order. Join Products and OrderDetails together first, then join the rest.

4. **Date Digging** – Modify the **Basic** statement again, this time instead of using orderdate itself, use the following clause in your where:

select customers.customername, orders.ordernumber, orders.orderdate, products.productname from

customers inner join orders on customers.customernumber = orders.customernumber

inner join orderdetails on orderdetails.ordernumber = orders.ordernumber

inner join products on orderdetails.productCode = products.productCode

where customers.customername like 'Mini%' and (year(orderDate) > 2015 OR

(year(orderDate) = 2015 AND month(orderDate) > 3) or

(year(orderDate) = 2015 AND month(orderDate) = 3) and day(orderDate) > 15)

τcustomername, orders.orderNumber, orderDate, productName

πcustomername, orders.orderNumber, orderDate, productName

σcustomername like 'Mini%' ∧ (year(orderDate) > 2015 OR

(year(orderDate) = 2015 ∧ month(orderDate) > 3) ∨

(year(orderDate) = 2015 ∧ month(orderDate) = 3) ∧ day(orderDate) > 15)

(customers ⨝ customers.customerNumber = orders.customerNumber orders ⨝ OrderDetails.orderNumber = orders.orderNumber OrderDetails ⨝ OrderDetails.productCode = products.productCode products)

Diagram

Description automatically generated

5. **Index** – Add an index to the customer name, then run **Basic** again.

a. To add an index to a table in MySQL:  
create index <name> on <table name> (<list of columns>);

create index idx\_customername on customers (customername)

select customers.customername, orders.ordernumber, orders.orderdate, products.productname from

customers inner join orders on customers.customernumber = orders.customernumber

inner join orderdetails on orderdetails.ordernumber = orders.ordernumber

inner join products on orderdetails.productCode = products.productCode

where customers.customername like 'Mini%' and orders.orderdate > date('2015-03-15')

τcustomername, orders.orderNumber, orderDate, productName πcustomername, orders.orderNumber, orderDate, productName σcustomername like 'Mini%' ∧ orderDate > date('2015-03-15') (customers ⨝ customers.customerNumber = orders.customerNumber orders ⨝ OrderDetails.orderNumber = orders.orderNumber OrderDetails ⨝ OrderDetails.productCode = products.productCode products)

Diagram

Description automatically generated