```
-- question 1. list all orders and their corresponding person responsible for the region using an inner
join

SELECT
   a.`Order Id`, b.Person
FROM
   Order_details.orderdata AS a
      INNER JOIN
   Order_details.people AS b ON a.Region = b.Region;

-- question 2. retrieve the detail of orders that were returned along with customers names,using a
left join between orderdata
--            and returned.

SELECT
   b.`Product Name`, a.`Order Id`, a.Returned
FROM
   Order_details.returned AS a
      LEFT JOIN
   Order_details.orderdata AS b ON a.`Order Id` = b.`order Id`;

-- question 3. find all orders along with their product name includung those without matching entries
in the
--            returned table (hint:-left join)

SELECT
   a.`Order Id`, a.`Product Name`
FROM
   Order_details.orderdata AS a
      LEFT JOIN
   Order_details.returned AS b ON a.`Order Id` = b.`order Id`;
```

```
-- question 4. Identify returned orders and display the responsible person for the region
--          (hint:-orderdata+returned+people : inner join)

SELECT
   b.`Order Id`, c.Person
FROM
   Order_details.orderdata AS a
      INNER JOIN
   Order_details.returned AS b
      INNER JOIN
   Order_details.people AS c ON a.`Order Id` = b.`order Id`
      AND a.Region = c.Region;

-- question 5. find orders that were not returned along with their shipping mode and product name
--          (hint:-left join+where clause)

SELECT
   a.`Ship Mode`, a.`Product Name`
FROM
   Order_details.orderdata AS a
      LEFT JOIN
   Order_details.returned AS b ON a.`Order Id` = b.`Order Id`
WHERE
   b.`Returned` IS NULL;

-- question 6. Show all regions and the total sales for each region including regions where no orders
were placed
--          (hint:- right join between orderdata and people)

SELECT
```

```
   a.Region, SUM(Sales) AS total_sale
FROM
   Order_details.orderdata AS a
      RIGHT JOIN
   Order_details.people AS b ON a.Region = b.Region
GROUP BY Region;
```

-- question 7. identify orders that have no matching entry in the returned table (hint:-left join with where clause+null check)

```
SELECT
   *
FROM
   Order_details.orderdata AS a
      LEFT JOIN
   Order_details.returned AS b ON a.`Order ID` = b.`Order ID`
WHERE
   b.Returned IS NULL;
```

-- question 8. find the numbers of orders for each person responsible for a region (hint:- inner join + group by person)

```
SELECT
   b.Person, a.Region, COUNT(Quantity) AS total_quantity
FROM
   Order_details.orderdata AS a
      INNER JOIN
   Order_details.people AS b ON a.Region = b.Region
GROUP BY Person , Region;
```

-- question 9. display all order that were returned along with their respective region and responsible person

```
SELECT
   c.Person, a.Region, b.Returned
FROM
   Order_details.orderdata AS a
      INNER JOIN
   Order_details.returned AS b
      INNER JOIN
   Order_details.people AS c ON a.`Order ID` = b.`Order ID`
      OR a.Region = c.Region
WHERE
   Returned IS NOT NULL;


with cte as (
select c.Person,a.Region,b.Returned                 -- displays 2871743 rows
from Order_details.orderdata as a
inner join Order_details.returned as b
on a.`Order ID`=b.`Order ID`
inner join Order_details.people as c
on a.Region=c.Region
where Returned is not null


)


select count(*) from cte ;


-- question 10. retrieve orders with profit>100 along with their region and person responsible using
inner join


SELECT
   a.Region, b.Person, a.Profit
```

```
FROM
   Order_details.orderdata AS a
      INNER JOIN
   Order_details.people AS b ON a.Region = b.Region
WHERE
   Profit > 100
GROUP BY Region , Person , Profit;


--  question 11. Identify regions that have no associated orders using a right join between orderdata
and people


SELECT
   a.Region
FROM
   Order_details.orderdata AS a
      RIGHT JOIN
   Order_details.people AS b ON a.Region = b.Region
WHERE
   `Order ID` IS NULL;


-- question 12. Retrieve all orders and returns information ensuring even unmatched orders are
displayed
--            (hint:- full outer join)


SELECT
   a.`Order ID`, b.Returned
FROM
   Order_details.orderdata AS a
      LEFT JOIN
   Order_details.returned AS b ON a.`Order ID` = b.`Order ID`
UNION SELECT
   a.`Order ID`, b.Returned
```

```
FROM
   Order_details.orderdata AS a
      RIGHT JOIN
   Order_details.returned AS b ON a.`Order ID` = b.`Order ID`;

-- question 13. Show the total profit for each person responsible for a region even if some region have no orders
--            (hint:-left joins)

SELECT
   b.Person, a.Region, ROUND(SUM(Profit), 2) AS total_profit
FROM
   Order_details.orderdata AS a
      LEFT JOIN
   Order_details.people AS b ON a.Region = b.Region
GROUP BY Person , Region;

-- question 14. Retrieve orders with quantity>10 and their corresponding person using inner join across the three table

SELECT
   a.`Order ID`, b.Person, a.Quantity
FROM
   Order_details.orderdata AS a
      INNER JOIN
   Order_details.people AS b
      INNER JOIN
   Order_details.returned AS c ON a.Region = b.Region
      OR a.`Order ID` = c.`Order ID`
WHERE
   Quantity > 10;
```

```
-- question 15. Identify duplicate entries in orderdata based on order id using a self join

SELECT
    a.`Order ID`
FROM
    Order_details.orderdata AS a
        INNER JOIN
    Order_details.orderdata AS b ON a.`Order ID` = b.`Order ID`
WHERE
    a.`Order ID` = b.`Order ID`;
-- group by `Order ID`
-- having `Order ID`=`Order ID`;



SELECT
    a.`Customer Name`, COUNT(a.Quantity) AS total_order
FROM
    Order_details.orderdata AS a
        JOIN
    Order_details.orderdata AS b ON a.`Customer ID` = b.`Customer ID`
GROUP BY `Customer Name`
HAVING total_order > 1;

--  question 17. Retrieve all orders including their return status (use left join with returned)

SELECT
    a.`Order ID`, b.Returned
FROM
    Order_details.orderdata AS a
        LEFT JOIN
    Order_details.returned AS b ON a.`Order ID` = b.`Order ID`;
```

```sql
-- question 18. Find regions with the highest sale and show the person responsible for those regions(inner join +group by)

SELECT
    b.Person, a.Region, MAX(Sales) AS highest_sale
FROM
    Order_details.orderdata AS a
        INNER JOIN
    Order_details.people AS b ON a.Region = b.Region
GROUP BY Person , Region;

-- question 19. Identify orders that were not shipped by checking ship date is null along with their return status

SELECT
    a.`Order ID`, b.Returned
FROM
    Order_details.orderdata AS a
        LEFT JOIN
    Order_details.returned AS b ON a.`Order ID` = b.`Order ID`
WHERE
    `Ship Date` IS NULL;

-- question 20. list all product that were returned ,along with the person responsible for the region they were sold in

SELECT
    a.`Product Name`, b.Returned, c.Person, a.Region
FROM
    Order_details.orderdata AS a
        LEFT JOIN
    Order_details.returned AS b ON a.`Order ID` = b.`Order ID`
        LEFT JOIN
    Order_details.people AS c ON a.Region = c.Region
WHERE
    returned IS NOT NULL;
```

```
        -- Customers With Above-Average Sales Sub Querries.
SELECT Customer_Name, SUM(Sales) AS Total_Sales
FROM Orders
GROUP BY Customer_Name
HAVING Total_Sales > (
    SELECT AVG(TotalSales)
    FROM (
        SELECT SUM(Sales) AS TotalSales
        FROM Orders
        GROUP BY Customer_Name
    ) AS SubQuery
)
ORDER BY Total_Sales DESC;
```

- **Create Views for Reuse**
- **Create a View for Monthly Regional Sales**

```
CREATE VIEW MonthlyRegionalSales AS
SELECT Region, DATE_FORMAT(Order_Date, '%Y-%m') AS Month,
       SUM(Sales) AS Monthly_Sales
FROM Orders
```

```
GROUP BY Region, Month;
```

- Now you can query it easily:

```
SELECT * FROM MonthlyRegionalSales WHERE Region = 'West';
```