

# CMR Assign

---

```
create database crm;
use crm;
create table customers(
customer_id int primary key,
first_name varchar(50),
last_name varchar(50),
email varchar(100),
phone varchar(13),
address varchar(200),
status varchar(10),
create_date date
);
insert into customers(customer_id, first_name, last_name, email, phone, address, status, create_date) values (1, 'John',
'Doe', 'johndoe@example.com', '1234567890', '123 Main St, City A', 'Active', '2024-10-01'),
(2, 'Jane', 'Smith', 'janesmith@example.com', '1234567891', '456 Oak St, City B', 'Inactive', '2024-09-20'),
(3, 'Alice', 'Johnson', 'alicej@example.com', '1234567892', '789 Pine St, City C', 'Prospect', '2024-09-15'),
(4, 'Bob', 'Williams', 'bobw@example.com', '1234567893', '101 Maple St, City D', 'Active', '2024-08-30'),
(5, 'Emma', 'Brown', 'emmab@example.com', '1234567894', '202 Birch St, City E', 'Inactive', '2024-07-25'),
(6, 'Liam', 'Davis', 'liamd@example.com', '1234567895', '303 Cedar St, City F', 'Prospect', '2024-06-18'),
(7, 'Olivia', 'Miller', 'oliviam@example.com', '1234567896', '404 Elm St, City G', 'Active', '2024-05-20'),
(8, 'Noah', 'Garcia', 'noahg@example.com', '1234567897', '505 Aspen St, City H', 'Inactive', '2024-04-15'),
(9, 'Sophia', 'Martinez', 'sophiam@example.com', '1234567898', '606 Spruce St, City I', 'Prospect', '2024-03-10'),
(10, 'Mason', 'Lee', 'masonl@example.com', '1234567899', '707 Chestnut St, City J', 'Active', '2024-02-05');
select * from customers;
create table leads(lead_id int primary key,
customer_id int,
foreign key(customer_id) references customers(customer_id),
lead_source varchar(20) not null check (lead_source in('Website', 'Email Campaign', 'Referral', 'Cold Call')),
lead_status varchar(20) not null check (lead_status in('New', 'Contacted', 'Qualified', 'Disqualified')),
create_date date
);
insert into leads(lead_id, customer_id, lead_source, lead_status, create_date) values
(1, 1, 'Website', 'New', '2024-10-01'),
(2, 2, 'Email Campaign', 'Contacted', '2024-09-20'),
(3, 3, 'Referral', 'Qualified', '2024-09-15'),
(4, 4, 'Cold Call', 'Disqualified', '2024-08-30'),
(5, 5, 'Website', 'Contacted', '2024-07-25'),
(6, 6, 'Email Campaign', 'New', '2024-06-18'),
(7, 7, 'Referral', 'Qualified', '2024-05-20'),
(8, 8, 'Cold Call', 'Disqualified', '2024-04-15'),
(9, 9, 'Website', 'New', '2024-03-10'),
(10, 10, 'Email Campaign', 'Contacted', '2024-02-05');
select * from leads;
create table products(product_id int primary key,
product_name varchar(50),
product_description text,
price decimal(10,2));
insert into products(product_id, product_name, product_description, price) values
(1, 'Smartphone', 'A high-end smartphone with 128GB storage and a dual camera.', 799.99),
```

```

(2, 'Laptop', 'A lightweight laptop with a 15-inch display and 8GB RAM.', 1099.99),
(3, 'Wireless Earbuds', 'Compact wireless earbuds with noise-cancellation technology.', 199.99),
(4, 'Smartwatch', 'A fitness-tracking smartwatch with heart rate monitor.', 249.99),
(5, 'Tablet', 'A 10-inch tablet with a high-resolution screen and 64GB storage.', 499.99),
(6, 'Gaming Console', 'A next-gen gaming console with 4K resolution support.', 499.99),
(7, 'Bluetooth Speaker', 'Portable Bluetooth speaker with deep bass and 12-hour battery life.', 99.99),
(8, 'Camera', 'A DSLR camera with 24MP resolution and 4K video recording.', 1199.99),
(9, 'External Hard Drive', 'A 2TB external hard drive for data backup and storage.', 89.99),
(10, 'Monitor', 'A 27-inch 4K UHD monitor with HDR support.', 329.99);
select * from products;

create table sales(sales_id int primary key,
customer_id int,
foreign key(customer_id) references customers(customer_id),
product_id int,
foreign key(product_id) references products(product_id),
sale_amount decimal(10,2),
sale_date date);
insert into sales(sales_id,customer_id,product_id,sale_amount,sale_date) values
(1, 1, 1, 799.99, '2024-10-01'),
(2, 2, 2, 1099.99, '2024-09-20'),
(3, 3, 3, 199.99, '2024-09-15'),
(4, 4, 4, 249.99, '2024-08-30'),
(5, 5, 5, 499.99, '2024-07-25'),
(6, 6, 6, 499.99, '2024-06-18'),
(7, 7, 7, 99.99, '2024-05-20'),
(8, 8, 8, 1199.99, '2024-04-15'),
(9, 9, 9, 89.99, '2024-03-10'),
(10, 10, 10, 329.99, '2024-02-05');
select * from sales;

create table interactions(interaction_id int primary key,
customer_id int,
foreign key(customer_id) references customers(customer_id),
interaction_type varchar(50) not null check(interaction_type in ('Phone Call', 'Email', 'Meeting', 'Chat', 'Social Media')),
interaction_date date);
insert into interactions values
(2, 2, 'Email', '2024-09-20'),
(3, 3, 'Meeting', '2024-09-15'),
(4, 4, 'Chat', '2024-08-30'),
(5, 5, 'Social Media', '2024-07-25'),
(6, 6, 'Phone Call', '2024-06-18'),
(7, 7, 'Email', '2024-05-20'),
(8, 8, 'Meeting', '2024-04-15'),
(9, 9, 'Chat', '2024-03-10'),
(10, 10, 'Social Media', '2024-02-05');
select * from customers;
select * from leads;
select * from sales;
select * from products;
select * from interactions;

```

/

*Answers starts from here/*

--1

```
select * from customers;
```

--2

```
select first_name,email from customers;
```

--3

```
select * from customers where status='Active';
```

--4

```
select * from customers where create_date>'2024-01-01';
```

--5

```
select first_name,last_name from customers order by last_name;
```

--6

```
select * from leads;
```

--7

```
select * from leads where lead_status='Qualified';
```

--8

```
select count(*) as total_customers from customers;
```

--9

```
select count(lead_status) as total_new from leads where lead_status='New';
```

--10

```
select * from sales where sale_date>'2024-03-01';
```

--11

```
select * from products where price>500.00;
```

--12

```
select distinct  
customers.customer_id,customers.first_name,customers.last_name,customers.email,customers.phone,customers.address  
from customers join leads on customers.customer_id=leads.customer_id where leads.lead_source='Email campaign';
```

--13

```
select customer_id,count(interaction_id) as total_intraction_count from interactions group by customer_id;
```

--14

```
select top 1 customers.customer_id,customers.first_name,customers.last_name,sum(sales.sale_amount) as total_amt from  
customers join sales on customers.customer_id=sales.customer_id group by  
customers.customer_id,customers.first_name,  
customers.last_name order by total_amt desc;
```

--without join

```
select top 1 customer_id,sum(sale_amount) as total_sales from sales group by customer_id order by total_sales desc;
```

--15

```
select product_name,price from products;
```

--16

```
select * from interactions where interaction_type='Phone call';
```

--17

```
select * from sales where month(sale_date)=06;
```

--18

```
select * from leads where lead_source='Referral';
```

--19

```
select products.product_id,products.product_name,sum(products.price) as total_price
from products join sales on products.product_id=sales.product_id group by products.product_id,products.product_name
order by total_price desc;
```

--20

```
select customers.customer_id,customers.first_name,customers.phone from customers join sales on
customers.customer_id=sales.customer_id
where customers.customer_id is null;
```

--22

```
select customers.customer_id,customers.first_name,customers.last_name,sum(sales.sale_amount) as total_sales from
customers join sales on customers.customer_id=sales.customer_id
group by customers.customer_id,customers.first_name,customers.last_name order by total_sales desc;
```

--23

```
select customers.first_name,customers.last_name,count(interactions.interaction_id) as total_interactions from customers
left join
interactions on customers.customer_id=interactions.customer_id group by
customers.customer_id,customers.first_name,customers.last_name
order by total_interactions desc;
```

--24

```
select products.product_id,products.product_name,count(sales.sales_id) as total_sales from products products join sales
on
products.product_id=sales.product_id group by products.product_id,products.product_name;
```

--25

```
select customers.customer_id,customers.first_name,customers.last_name,count(sales.sales_id) as total_sales from
customers customers join
sales on customers.customer_id=sales.customer_id group by
customers.customer_id,customers.first_name,customers.last_name having count(sales.sales_id)>3;
```

--26. Retrieve the total number of leads grouped by lead source.

```
select lead_source,count(lead_id) as total_leads from leads group by lead_source;
```

--27. List all customers and include the total number of interactions they've had.

--repeated

--28. Retrieve all products and include the total quantity sold for each.

```
select products.product_id,products.product_name,count(sales.sales_id) as total_sales
from products join sales on products.product_id=sales.product_id group by products.product_id,products.product_name;
```

--29. Find the top 5 customers by total sales amount.

select top 5

```
customers.customer_id,customers.first_name,customers.last_name,sum(sales.sale_amount) as total_sales from
customers join
sales on customers.customer_id=sales.customer_id group by
customers.customer_id,customers.first_name,customers.last_name
order by total_sales desc;
```

--30. List all customers who have been contacted via 'Phone Call'.

select distinct

```
customers.customer_id,customers.first_name,customers.last_name,customers.email,customers.phone,customers.address
from customers join interactions on customers.customer_id=interactions.customer_id where
interactions.interaction_type='Phone call';
```

--31. Retrieve the details of the most recent interaction for each customer.

```
select customer_id,max(interaction_date) as max_date from interactions group by customer_id;
```

--32. Find the total sales amount for each customer, grouped by status.

select

```
customers.customer_id,customers.first_name,customers.last_name,sum(sales.sale_amount) as total_sales from
customers join sales on customers.customer_id=sales.customer_id group by
customers.customer_id,customers.first_name,customers.last_name,customers.status ;
```

--33. List all leads created after June 2023, sorted by lead source.

```
select * from leads where create_date>'2024-06-30' order by lead_source;
```

--34. Retrieve the details of all interactions that happened within the last 30 days.

```
select * from interactions where interaction_date>=DATEADD(DAY, -30, GETDATE());
```

--35. Find the customer who has had the most interactions.

```
select top 1 customers.customer_id,customers.first_name,customers.last_name,count(interactions.interaction_id) as
total_interaction from
customers join interactions on customers.customer_id=interactions.customer_id group by
customers.customer_id,customers.first_name,customers.last_name
order by total_interaction desc;
```

--36. Retrieve all products that have not been sold yet.

```
select products.product_id,products.product_name,products.price,products.product_description from products left join
sales
on products.product_id=sales.product_id where sales.sales_id is null;
```

--37. List all leads along with the customer's first and last names.

--repeated

--38. Retrieve the total number of interactions for each interaction type.

```
select * from interactions;
select interaction_type,count(interaction_id) as total_interactions from interactions group by interaction_type;
```

--39. Find all customers with more than one lead.

```
select * from customers;
select * from leads;
select customers.customer_id,customers.first_name,customers.last_name,count(leads.lead_id) as total_leads from
customers
join leads on customers.customer_id=leads.customer_id group by
customers.customer_id,customers.first_name,customers.last_name having count(leads.lead_id)>1;
```

--40

--40. Retrieve all leads with the status 'Disqualified' and their respective customer names.

```
select leads.lead_id,leads.lead_status,leads.lead_source,leads.create_date,customers.first_name from leads join
customers on leads.customer_id=customers.customer_id
where leads.lead_status='Disqualified';
```

--41. List all sales made by customers whose status is 'Prospect'.

```
select sales.sales_id,sales.sale_amount,sales.sale_date,customers.status from sales left join customers on
sales.customer_id=customers.customer_id where customers.status='Prospect';
```

--43 REPEATED

--44. List all leads and include the customer's first and last names from the Customers table.

```
select leads.lead_id,leads.lead_source,leads.create_date,leads.lead_status,customers.first_name,customers.last_name
from leads join customers on leads.customer_id=customers.customer_id;
```

--45. Find all sales along with the corresponding product names.

```
select sales.sales_id,sales.sale_amount,sales.sale_date,products.product_name from sales join products
on sales.product_id=products.product_id;
```

--46. Retrieve all interactions and include the customer's first and last names.

```
select interactions.interaction_id,interactions.interaction_date,interactions.interaction_type,customers.first_name,
customers.last_name from interactions join customers on interactions.customer_id=customers.customer_id;
```

--47. Retrieve all customers and include their total number of sales.

```
select
customers.customer_id,customers.first_name,customers.last_name,count(sales.sales_id) as total_sales
from customers join sales on customers.customer_id=sales.customer_id group by
customers.customer_id,customers.first_name,customers.last_name;
```

--48. List all customers and their total sales amounts, including those who have made no sales.

```
select customers.customer_id,customers.first_name,customers.last_name,sum(sales.sale_amount) as total_sales
from customers left join sales on customers.customer_id=sales.customer_id group by
customers.customer_id,customers.first_name,customers.last_name;
```

--49. Find all interactions and include the corresponding lead details.

```
select * from customers join leads on customers.customer_id=leads.lead_id;
```

--50. Retrieve all sales made by customers who have more than 5 interactions.

```
select sales.sales_id,sales.sale_amount,interactions.interaction_id,interactions.interaction_date from sales join
interactions on sales.customer_id=interactions.customer_id group by
sales.sales_id,sales.sale_amount,interactions.interaction_id,interactions.interaction_date
having count(interactions.interaction_id)>5;
```

--51. List all customers and their most recent lead status.

--doubt

```
select customers.first_name,customers.last_name,leads.lead_status from customers join leads on
customers.customer_id=leads.lead_id group by customers.first_name,customers.last_name,leads.lead_status having
leads.lead_status='New';
```

--52. Retrieve all interactions made via 'Email' along with the corresponding customer details.

select interactions.

```
,customers. from interactions join customers on customers.customer_id=interactions.customer_id
where interactions.interaction_type='Email';
```

--53. Retrieve all products that have been sold along with the total sales amount for each.

```
select products.product_id,products.product_name,sum(sales.sale_amount) as total_sales from products left join sales on
products.product_id=sales.product_id
group by products.product_id,products.product_name;
```

--54. Find all leads with their corresponding customer's sales amount.

```
select leads.,sales.sale_amount from leads join sales on leads.customer_id=sales.customer_id;
```

--55. List all interactions that happened after May 2023 and include the customer's details.

```
select interactions.,customers.* from interactions full join customers on interactions.customer_id=customers.customer_id
where interaction_date>'2024-05-31';
```

--56. Retrieve all leads created in 2024 and include their respective customer's total sales.

```
select leads.lead_id,leads.lead_source,sales.sale_amount,leads.create_date,sum(sales.sale_amount) as total_sales from
leads join sales on
leads.customer_id=sales.customer_id group by leads.lead_id,leads.lead_source,sales.sale_amount,leads.create_date
having leads.create_date>'2023';
```

--57. List all products with their total sales amount, including products with no sales.

--repeated

--58. Find all customers who have had at least 3 interactions and have made a sale.

```
select customers.customer_id,customers.first_name,customers.last_name,count(interactions.interaction_id) as
total_interaction,sales.sales_id
from customers join interactions on customers.customer_id=interactions.customer_id
join sales on customers.customer_id=sales.customer_id group by
customers.customer_id,customers.first_name,customers.last_name,sales.sales_id
having count(interactions.interaction_id)>=0;
```

--59. Retrieve the total sales amount and total number of interactions for each customer.

```
select sales.sales_id,sales.customer_id,interactions.customer_id,sum(sales.sale_amount) as
total_sales,count(interactions.interaction_id) as total_interactions
from sales join interactions on sales.customer_id=interactions.customer_id group by
sales.sales_id,sales.customer_id,interactions.customer_id ;
```

--60. List all customers and their corresponding lead details.

```
select customers.,leads. from customers full join leads on customers.customer_id=leads.lead_id;
```

--61. Retrieve all sales made in June 2023 along with the customer and product details.

```
select * from sales join customers on sales.customer_id=customers.customer_id join products on
sales.product_id=products.product_id where month(sales.sale_date)='06';
```

--62. Find all customers who have made at least one sale and interacted via 'Phone Call'.

```
select customers.customer_id,customers.first_name,customers.last_name,count(sales.sales_id) as
total_sales,interactions.interaction_type from customers join sales on
customers.customer_id=sales.sales_id join interactions on interactions.customer_id=customers.customer_id
group by customers.customer_id,customers.first_name,customers.last_name,interactions.interaction_type
having interactions.interaction_type='Phone call' and count(sales.sales_id)>=1;
```

--64.Find the customer with the largest total sales amount.

```
select customer_id,sale_amount from sales where sale_amount=(select max(sale_amount) as max_sales from sales);
```

--65. Retrieve customers who have had more interactions than the average number of interactions.

```
select customer_id from interactions group by customer_id having count(interaction_id)>(select avg(interaction_id)from
interactions);
```

--66. Find the total sales for each customer and sort the result in descending order.

```
select
customers.customer_id,customers.first_name,customers.last_name,sum(sales_id)
as total_sales from customers join sales on customers.customer_id=sales.customer_id
group by customers.customer_id,customers.first_name,customers.last_name order by total_sales desc;
```

--67. List the products that have been sold more than 0 times.

```
select products.product_id,products.product_name,count(sales_id) as total_sales from
products join sales on products.product_id=sales.product_id group by products.product_id,products.product_name
having count(sales_id)>0;
```

--68. Find customers who have more than 2 leads.

```
select customers.customer_id,customers.first_name,leads.lead_id,count(leads.lead_id) as total_leads from
customers join leads on customers.customer_id=leads.customer_id group by
customers.customer_id,customers.first_name,leads.lead_id
having count(leads.lead_id)>2;
```

--69. List all customers who have made more than one sale with a total amount over \$500.

```
select customers.customer_id,customers.first_name,count(sales.sales_id) as total_sales,
sum(sales.sale_amount) as total_sale_amount from customers join sales on customers.customer_id=sales.customer_id
group by customers.customer_id,customers.first_name having count(sales.sales_id)>1 and sum(sales.sale_amount)>100;-
```



-doubt

--70. Find customers who have sales but no interactions.

```
select customers.customer_id,customers.first_name from customers join sales on
customers.customer_id=sales.customer_id
left join interactions on interactions.customer_id=customers.customer_id where interactions.interaction_id is null;
```

--71. List the total number of sales for each product that has been sold in the last 6 months.

```
select products.product_id,products.product_name,count(sales.sale_id) as total_slas from products
join sales on products.product_id=sales.product_id where sales.sale_date >= DATEADD(MONTH, -6, GETDATE())
group by products.product_id,products.product_name;
```

--72. Find all customers with sales totaling more than \$1000.

```
select customers.customer_id,customers.first_name,sum(sales.sale_amount) from customers join sales
on customers.customer_id=sales.customer_id group by customers.customer_id,customers.first_name
having sum(sales.sale_amount)>1000;
```

--73. List all interactions that have occurred in the past 7 days.

```
select * from interactions where interaction_date>=dateadd(day,-30,getdate());
```

--74. Retrieve the average sale amount for each customer.

```
select customers.customer_id,customers.first_name,avg(interactions.interaction_id) as avg_interactions from
customers join interactions on customers.customer_id=interactions.interaction_id group by
customers.customer_id,customers.first_name;
```

--75. Find the customers who have interacted through more than two different interaction types.

```
select customers.customer_id,customers.first_name,count( distinct interactions.interaction_type) as unq_interactions from
customers join interactions on customers.customer_id=interactions.interaction_id group by
customers.customer_id,customers.first_name
having count(distinct interactions.interaction_type)>2;
```

--76. List all customers who have both a lead and a sale.

```
select customers.customer_id,customers.first_name from customers join leads
on customers.customer_id=leads.customer_id join sales on sales.customer_id=customers.customer_id
;
```

--77. Retrieve the customers who have interacted with 'Email' and 'Phone Call'.

```
select customers.customer_id,customers.first_name,interactions.interaction_type from customers join
interactions on customers.customer_id=interactions.customer_id where
interactions.interaction_type='Email' or interactions.interaction_type='Phone call'
group by customers.customer_id,customers.first_name,interactions.interaction_type;
```

--78. Find the total number of leads generated by each lead source.

```
select lead_source,count(lead_id) as total_leads from leads group by lead_source;
```

--79. List all customers who have had no interactions in the last 90 days.

```
select customers.customer_id,customers.first_name from customers left join interactions on
customers.customer_id=interactions.customer_id
and interactions.interaction_date>=DATEADD(DAY, -90, GETDATE()) where interactions.interaction_id is null;
```

--80. Find the average sale amount for each product.

```
select products.product_id,products.product_name,avg(sales.sale_amount) as avg_sales from
products join sales on products.product_id=sales.product_id group by products.product_id,products.product_name;
```

--81. List customers who have sales totaling more than \$5000 in the past year.

```
select customers.customer_id,customers.first_name,customers.last_name,sum(sales.sale_amount) as total_sales
from customers join sales on customers.customer_id=sales.customer_id group by
customers.customer_id,customers.first_name,
customers.last_name having sum(sales.sale_amount)>5000;
```

--82. Retrieve all customers who have not made any sales yet.

```
select customers.* ,sales.sales_id from customers join sales on customers.customer_id=sales.customer_id where
sales.sales_id is null;
```

--83. Find all leads where the source is 'Cold Call' and the status is 'Qualified'.

```
select * from leads where lead_source='Cold call'and lead_status='Qualified';
```

--84. List all products with the highest number of sales.

```
select products.product_id,products.product_name,count(sales.sales_id) as total_sales
from products join sales on products.product_id=sales.product_id group by products.product_id,products.product_name
order by total_sales desc;
```

--85. Retrieve customers who have interacted more than 5 times in the last 30 days.

--repeated

--86. Find the total number of leads for each customer, grouped by lead source.

```
select customers.customer_id,customers.first_name,leads.lead_source,count(leads.lead_id) as total_interactions
from customers join leads on customers.customer_id=leads.customer_id group by
customers.customer_id,customers.first_name,leads.lead_source;
```

--87. List the customers who have sales but no leads.

```
select customers.customer_id,customers.first_name from customers join
sales on customers.customer_id=sales.customer_id left join leads
on leads.customer_id=customers.customer_id where leads.lead_id is null
group by customers.customer_id,customers.first_name;
```

--88. Retrieve customers with the most interactions in a single day.

```
select top 1 customers.customer_id,
first_name,
last_name,
interaction_date,COUNT(interaction_id) AS interaction_count
from customers join interactions on customers.customer_id=interactions.customer_id
group by customers.customer_id,
```

```
first_name,
last_name,
interaction_date order by interaction_count desc;
```

--89. Find the top 3 customers by the number of interactions.

```
select top 3 customers.customer_id,first_name,last_name,count(interactions.interaction_id) as total_interactions
from customers join interactions on customers.customer_id=interactions.customer_id group by
customers.customer_id,first_name,last_name
order by total_interactions desc;
```

--90. Retrieve the total sales amount for each customer in the past 6 months.

```
select customers.customer_id,customers.first_name,customers.last_name,sales.sale_date,sum(sales.sale_amount) as
total_sales from
customers join sales on customers.customer_id=sales.customer_id where sales.sale_date>=DATEADD(month, -6,
GETDATE())
group by customers.customer_id,customers.first_name,customers.last_name,sales.sale_date;
```

--91. List the average number of interactions per customer.

```
select avg(interaction_count) as average_interactions from(
```

```

        select customers.customer_id,count(interactions.interaction_id) as
s interaction_count
        from customers leaft join interactions on
        customers.customer_id=interactions.customer_id
        group by customers.customer_id
    ) as subquery;
```

--92. Find the customers who have both 'Phone Call' and 'Email' interactions.

```
select customers.customer_id,customers.first_name,customers.last_name,interactions.interaction_type from
customers join interactions on customers.customer_id=interactions.interaction_id where
interactions.interaction_type='Call'
and interactions.interaction_type='Email';
```

--93. List all customers who have been contacted more than 3 times but have no sales.

```
select customers.customer_id,customers.first_name,customers.last_name from customers
left join interactions on customers.customer_id=interactions.customer_id left join
sales on sales.customer_id=customers.customer_id group by
customers.customer_id,customers.first_name,customers.last_name
having count(interactions.interaction_id) >3 and count(sales.sales_id)=0;
```

--94. Find the customers who have made a sale but have not been contacted via interactions.

```
select customers.customer_id,customers.first_name,customers.last_name from customers join sales
on customers.customer_id=sales.sales_id left join interactions on customers.customer_id=interactions.interaction_id
where interactions.interaction_id is null group by customers.customer_id,customers.first_name,customers.last_name;
```

--95. Retrieve the total sales amount for each lead source.

```
select leads.lead_source,sum(sales.sale_amount) as total_sales from leads join sales on
leads.customer_id=sales.customer_id group by leads.lead_source;
```

-96. List the total number of interactions made by each customer in the past year.

--repeated

--97. Find the customers who have interacted via 'Meeting' but have not made a sale.

```
select customers.customer_id,customers.first_name,customers.last_name,interactions.interaction_type from
customers join interactions on customers.customer_id=interactions.customer_id left join sales on customers.customer_id=
sales.customer_id where interactions.interaction_type='Meeting' and sales.sales_id is null;
```

--98. Retrieve the products with the highest total sales amounts.

```
select products.product_id,products.product_name,max(sales.sale_amount) as
total_sales from products join sales on products.product_id=sales.product_id group by
products.product_id,products.product_name
order by total_sales desc;
```

--99. List all customers who have made a sale and also interacted via 'Social Media'.

select

```
customers.customer_id,customers.first_name,customers.last_name,sales.sales_id,interactions.interaction_type from
customers join sales on customers.customer_id=sales.customer_id join interactions on
customers.customer_id=interactions.customer_id
where sales.sales_id is null and interactions.interaction_type='Social Media';
```

--100. Retrieve the total number of leads generated in the past 6 months.

--repeated

-101. List the products that have generated more than \$1000 in sales in the last year.

select

```
products.product_id,products.product_name,products.price,sales.sale_amount,sales.sale_date from products join
sales on products.product_id=sales.product_id where sales.sale_amount='100' group by
products.product_id,products.product_name,products.price,sales.sale_amount,sales.sale_date having
sales.sale_date>=DATEADD(year, -1, GETDATE());
```

-102. Find the total number of sales for each customer, including those with no sales.

```
select customers.customer_id,customers.first_name,count(sales.sales_id) as total_sales from
customers full join sales on customers.customer_id=sales.customer_id group by
customers.customer_id,customers.first_name;
```

-103. Retrieve all customers who have a lead source of 'Referral' and have made at least one sale.

```
select customers.customer_id,customers.first_name,leads.lead_source,sales.sales_id from customers join leads
on customers.customer_id=leads.customer_id join sales on customers.customer_id=sales.customer_id where
leads.lead_source='Referral';
```