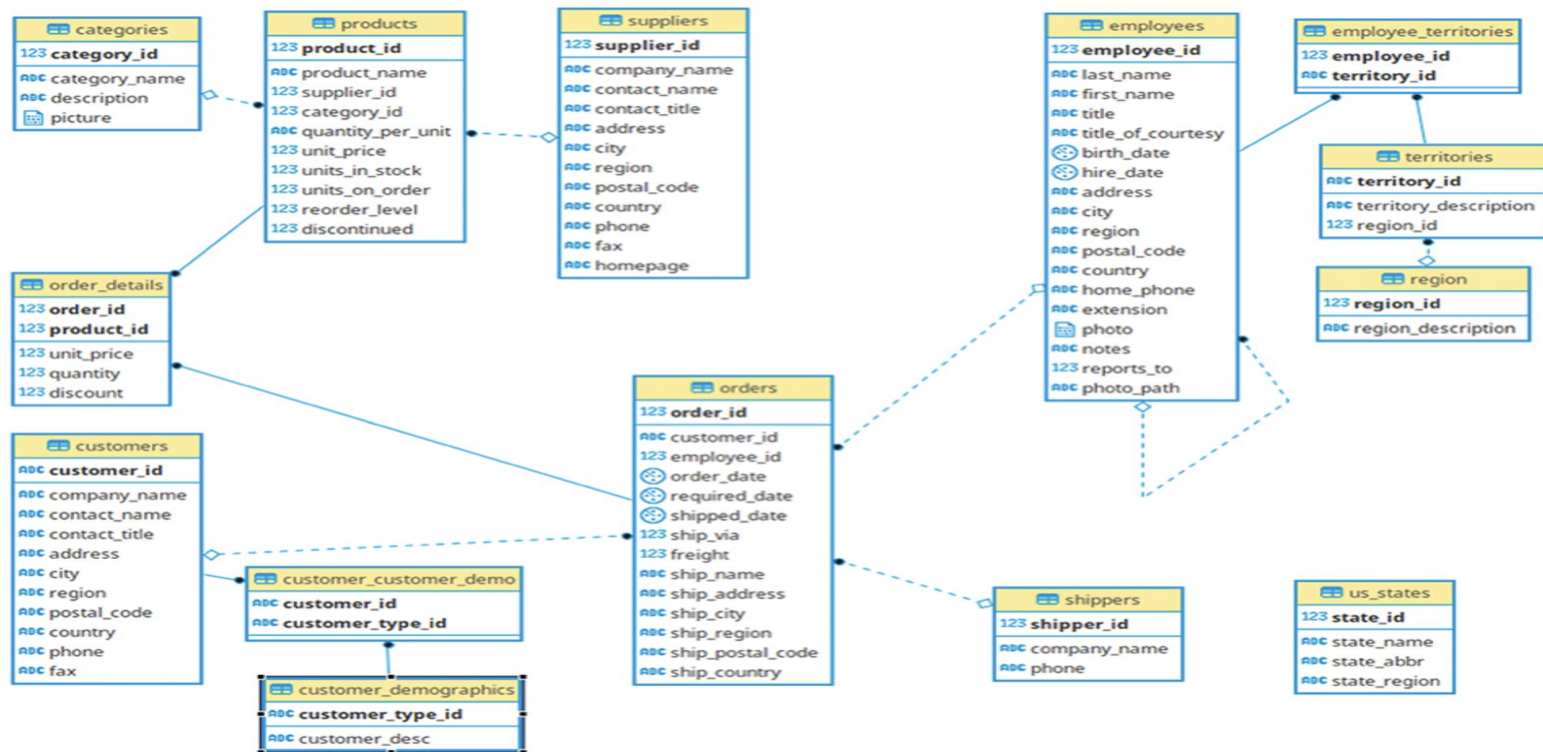


# Business Case Study

# Tables to be Downloaded

1. Customers
2. Suppliers
3. Employees
4. Products
5. Shippers
6. Orders
7. Order\_Details

# Schema



Ques1. Fetch the full name and hiring date of all Employees who work as Sales Representatives.

```
select  
concat(firstname, ' ', lastname) as full_name,  
hiredate  
from `business_case_study.employees`  
where title = 'Sales Representative';
```

## \*Ques2. Which of the products in our inventory need to be reordered?

**Note:** For now, just use the fields UnitsInStock and ReorderLevel, where UnitsInStock is less than or equal to the ReorderLevel, ignoring the fields UnitsOnOrder and Discontinued.

```
select
productid,productname,unitsinstock,reorderlevel from
`business_case_study.products`
where unitsinstock <= reorderlevel
order by productid;
```

\*Ques3. Find and display the details of customers who have placed more than 5 orders.

```
select * from `business_case_study.customers`  
where customerid IN  
(  
select customerid from `business_case_study.orders`  
group by customerid  
having count(*) > 5  
);
```

```
select c.customerid  
from `business_case_study.customers` c  
INNER JOIN `business_case_study.orders` o  
USING (customerid)  
group by c.customerid  
having count(*) > 5;
```

\*Ques4. An employee of ours (Margaret Peacock, EmployeeID 4) has the record of completing most orders. However, there are some customers who've never placed an order with her. Show all such customers.

```
select c.customerid,o.customerid
from `business_case_study.customers` c
LEFT JOIN `business_case_study.orders` o ON c.customerid = o.customerid
and o.employeeid = 4
where o.customerid is null;
```

```
select customerid from business_case_study.customers where customerid not in (
select customerid from business_case_study.employees join business_case_study.orders using(employeeid)
where employeeid=4
)
```

\*Ques5. The developers at Cochin Traders are testing an app that the customers will use to show orders. In order to make sure that even the largest orders will show up correctly on the app, they'd like some samples of orders that have lots of individual line items. Display the top 10 orders with the most line items.

```
select
o.orderid,
COUNT(*) as total_order_details
from `business_case_study.orders` o
INNER JOIN `business_case_study.orders_details` od
ON o.orderid = od.orderid
GROUP BY o.orderid
order by total_order_details desc
LIMIT 10;

select orderid, count(*) as number from `business_case_study.orders_details`
group by orderid
order by number desc
limit 10;
```



\*Ques6. Retrieve the top 5 best-selling products on the basis of the quantity ordered.

```
select
productname,
SUM(quantity) as total_qty
from `business_case_study.products` p
JOIN `business_case_study.orders_details` od
ON p.productid = od.productid
group by productname
order by total_qty desc
limit 5;
```

\*Ques7. Analyze the monthly order count for the year 1997.

```
select
EXTRACT(MONTH FROM orderdate) as month,
COUNT(*) as ordercount
from `business_case_study.orders`
where EXTRACT(YEAR FROM orderdate) = 1997
group by month
order by month;
```

\*Ques8. Calculate the difference in sales revenue for each month compared to the previous month.

```
WITH month_revenue AS
(
  select
  FORMAT_DATE('%Y-%m',orderdate) as monthdatayearwise,
  ROUND(SUM(quantity * unitprice),2) as revenue
  from `business_case_study.orders` o
  INNER join `business_case_study.orders_details` od
  ON o.orderid = od.orderid
  group by monthdatayearwise
)
SELECT monthdatayearwise,revenue,
LAG(revenue) OVER(order by monthdatayearwise) as previousMnthrevenue,
ROUND(revenue - LAG(revenue) OVER(order by monthdatayearwise),2) as revenue_diff,
from month_revenue
order by monthdatayearwise;
```

\*Ques9. Calculate the percentage of total sales revenue for each product.

#ans 9

SELECT

productname,

round(sum(od.unitprice \* quantity),2) as revenue,

round(sum(od.unitprice \* od.quantity)/ SUM(sum(od.unitprice \* quantity)) OVER(),2) \* 100 as  
revenuepercentage

from `business\_case\_study.products` p JOIN

`business\_case\_study.orders\_details` od ON p.productid = od.productid

group by productname

order by revenuepercentage DESC;

\*Ques10. Determine the cumulative percentage of total sales revenue for each month.

```
WITH monthly_revenue AS
(
  select
  FORMAT_DATE('%Y-%m',orderdate) as monthdatayearwise,
  ROUND(SUM(quantity * unitprice),2) as revenue
  from `business_case_study.orders` o
  JOIN `business_case_study.orders_details` od
  ON o.orderid = od.orderid
  group by monthdatayearwise
)
select monthdatayearwise,revenue,
ROUND(SUM(revenue) OVER(order by monthdatayearwise)/ SUM(revenue) OVER(),2) * 100 as cum_sales
from monthly_revenue
order by monthdatayearwise;
```