SQL PROJECT

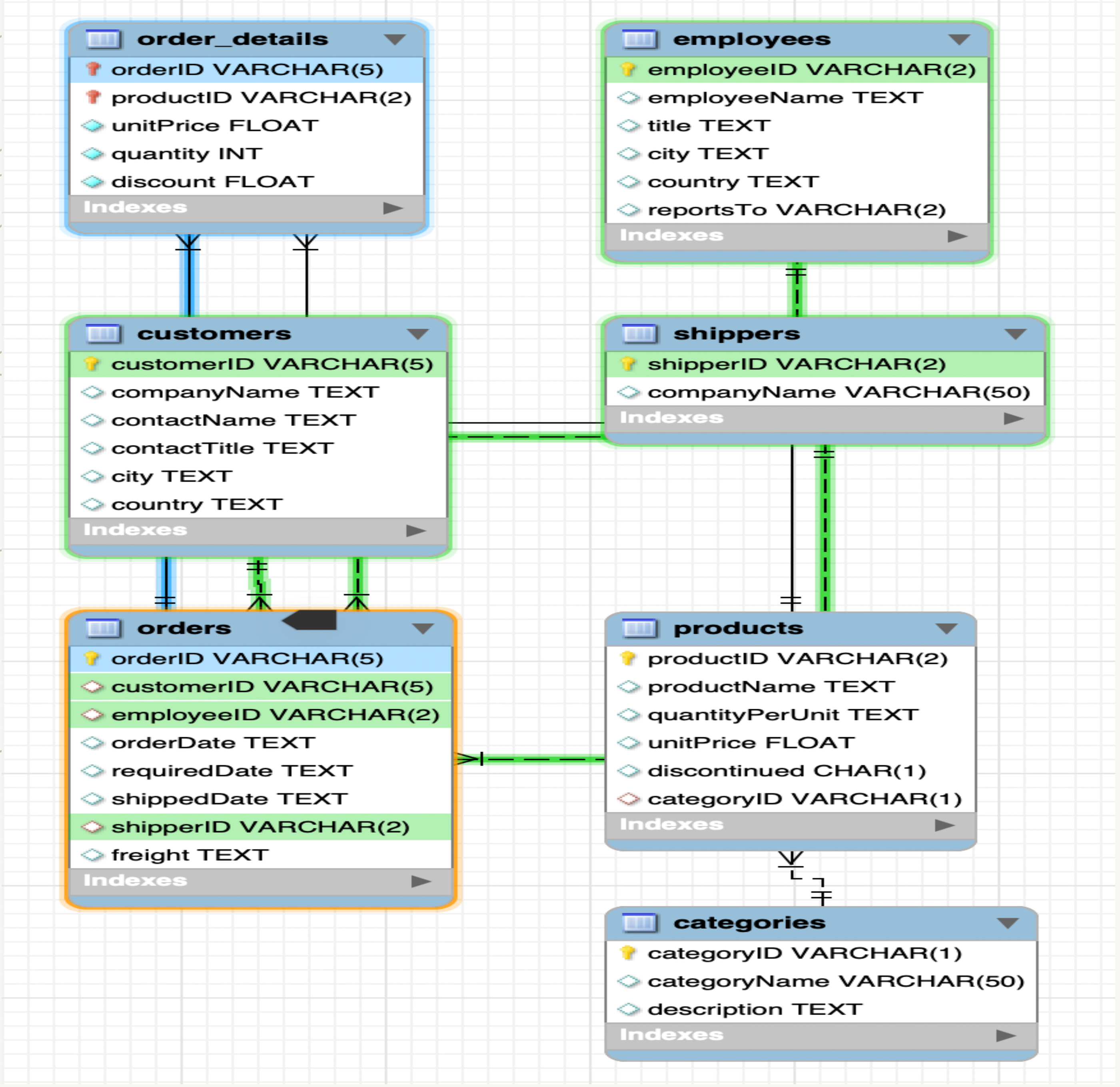
# NORTHWIND TRADERS

-Yash Khobragade

## ABSTRACT

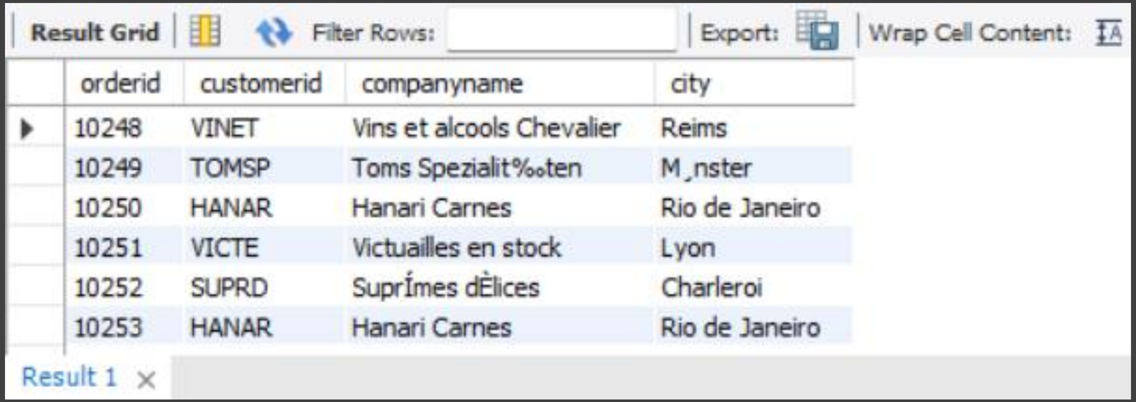
This project leverages SQL to uncover vital business insights from a structured retail database, focusing on sales performance, customer behavior, and operational efficiency. Through advanced SQL queries, the project explores essential metrics by joining tables, grouping data, and calculating key financial indicators. Initial queries reveal critical insights, such as geographic distribution of orders, high-demand products, and top-spending customers, helping to identify key market segments. The analysis also tracks monthly revenue by employee, assesses cost-effectiveness of shippers for high-value orders, and measures average discount rates by country to optimize regional pricing. Additionally, the project evaluates total revenue by product category, employee sales performance within these categories, and ensures data consistency by updating discounts on discontinued products. Together, these queries enable data-driven decision-making, transforming complex datasets into actionable intelligence for business growth and efficiency.

# E-R DIAGRAM


order_details
orderID VARCHAR(5)
productID VARCHAR(2)
unitPrice FLOAT
quantity INT
discount FLOAT
Indexes
employees
employeeID VARCHAR(2)
employeeName TEXT
title TEXT
city TEXT
country TEXT
reportsTo VARCHAR(2)
Indexes
customers
customerID VARCHAR(5)
companyName TEXT
contactName TEXT
contactTitle TEXT
city TEXT
country TEXT
Indexes
shippers
shipperID VARCHAR(2)
companyName VARCHAR(50)
Indexes
orders
orderID VARCHAR(5)
customerID VARCHAR(5)
employeeID VARCHAR(2)
orderDate TEXT
requiredDate TEXT
shippedDate TEXT
shipperID VARCHAR(2)
freight TEXT
Indexes
products
productID VARCHAR(2)
productName TEXT
quantityPerUnit TEXT
unitPrice FLOAT
discontinued CHAR(1)
categoryID VARCHAR(1)
Indexes
categories
categoryID VARCHAR(1)
categoryName VARCHAR(50)
description TEXT
Indexes

## LIST ALL ORDERS ALONG WITH THE CORRESPONDING CUSTOMER NAME AND CITY.

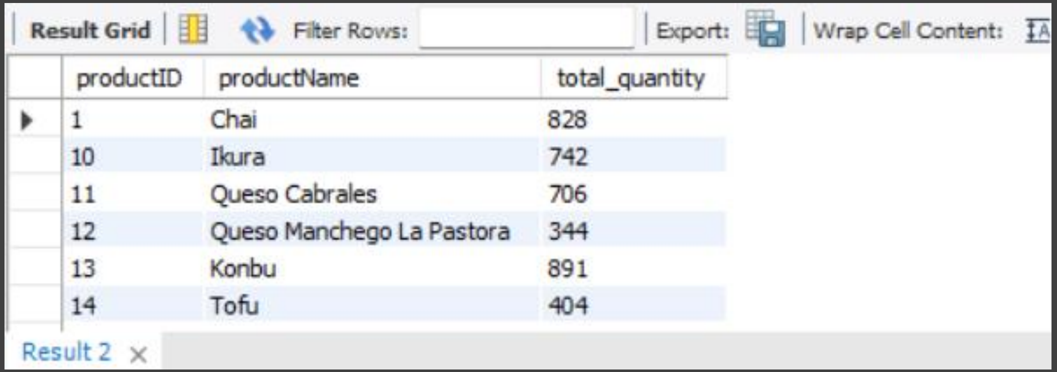
```sql
select  o.orderid , c.customerid , c.companyname , c.city
from orders o
join customers c
on o.customerID  = c.customerID;
```



| orderid | customerid | companyname | city |
|---|---|---|---|
| 10248 | VINET | Vins et alcools Chevalier | Reims |
| 10249 | TOMSP | Toms Spezialit‰ten | M¸nster |
| 10250 | HANAR | Hanari Carnes | Rio de Janeiro |
| 10251 | VICTE | Victuailles en stock | Lyon |
| 10252 | SUPRD | SuprÍmes dÈlices | Charleroi |
| 10253 | HANAR | Hanari Carnes | Rio de Janeiro |

## CALCULATE THE TOTAL QUANTITY ORDERED FOR EACH PRODUCT.

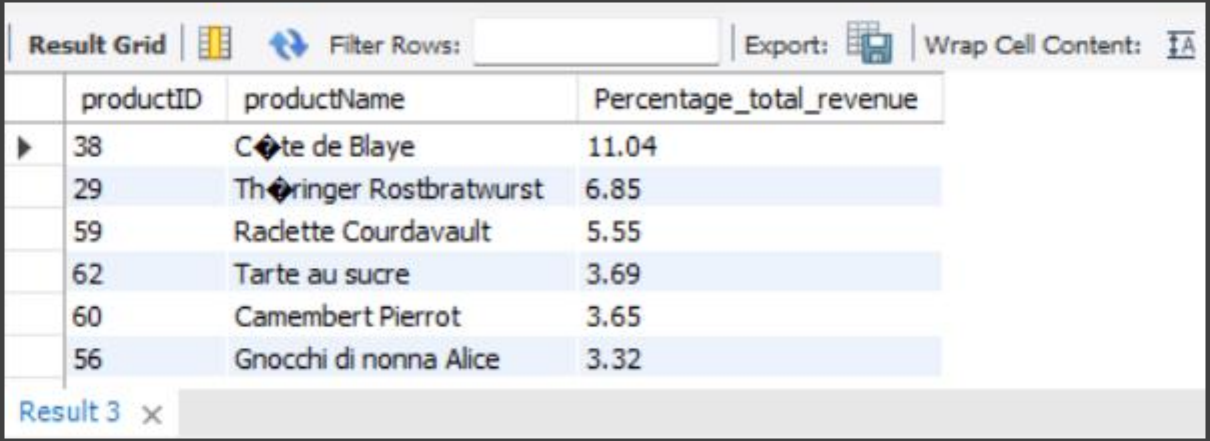
```
SELECT    p.productID,    p.productName,    SUM(od.quantity) AS total_quantity
FROM    order_details od
JOIN    products p
ON    od.productID = p.productid
GROUP BY    p.productID,  p.productName
ORDER BY    p.productID ASC;
```



| productID | productName | total_quantity |
|---|---|---|
| 1 | Chai | 828 |
| 10 | Ikura | 742 |
| 11 | Queso Cabrales | 706 |
| 12 | Queso Manchego La Pastora | 344 |
| 13 | Konbu | 891 |
| 14 | Tofu | 404 |

## PERCENTAGE OF TOTAL REVENUE BY PRODUCT

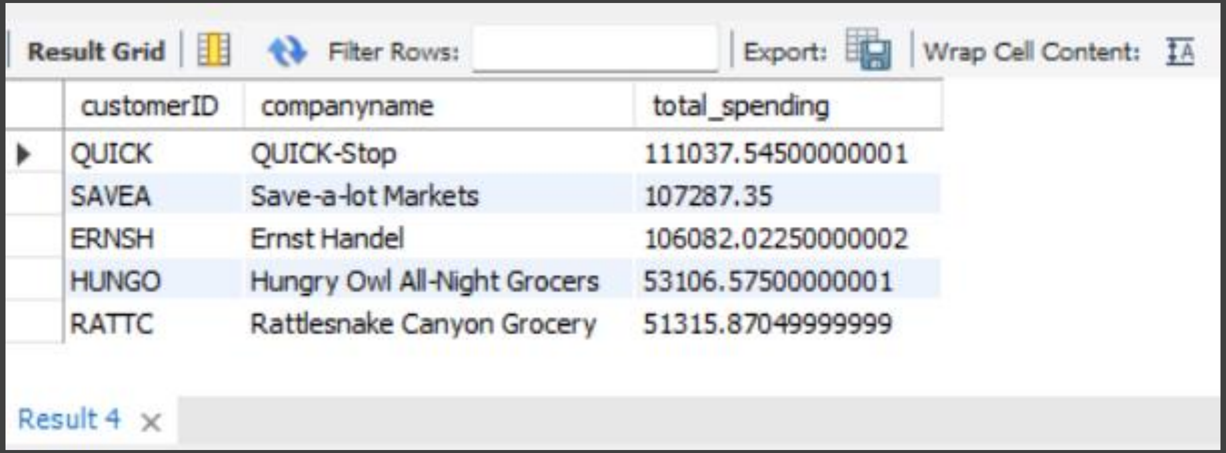
```
SELECT    p.productID,    p.productName,  SUM(od.quantity * od.unitPrice * (1-od.discount)) AS
          product_revenue,   round((SUM(od.quantity * od.unitPrice * (1-od.discount)) / (select
          SUM(od2.quantity * od2.unitPrice * (1-od2.discount)) from order_details od2 ))* 100 , 2)
as Percentage_total_revenue
FROM    products p
JOIN order_details od
ON    p.productID =od.productid
GROUP BY  p.productID , p.productName
ORDER BY    percentage_total_revenue desc;
```



| productID | productName | Percentage_total_revenue |
|---|---|---|
| 38 | C�te de Blaye | 11.04 |
| 29 | Th�ringer Rostbratwurst | 6.85 |
| 59 | Raclette Courdavault | 5.55 |
| 62 | Tarte au sucre | 3.69 |
| 60 | Camembert Pierrot | 3.65 |
| 56 | Gnocchi di nonna Alice | 3.32 |

## Find the top 5 customers with the highest total spending on orders.

```
select c.customerID ,c.companyname ,  sum(od.quantity * od.unitPrice * (1-od.discount)) as
            total_spending
from order_details od
join orders o
on od.orderid = o.orderID
join customers c
on o.customerID = c.customerID
group by customerID
order by total_spending
desc limit 5;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| customerID | companyname | total_spending |
|---|---|---|
| QUICK | QUICK-Stop | 111037.54500000001 |
| SAVEA | Save-a-lot Markets | 107287.35 |
| ERNSH | Ernst Handel | 106082.02250000002 |
| HUNGO | Hungry Owl All-Night Grocers | 53106.57500000001 |
| RATTC | Rattlesnake Canyon Grocery | 51315.87049999999 |

Result 4

## UPDATE THE DISCOUNT OF ORDERS FOR DISCONTINUED PRODUCTS TO ZERO.

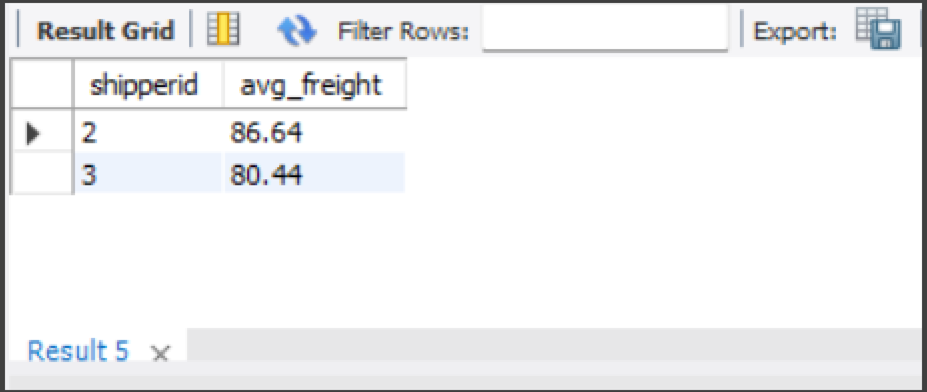
```
update order_details
set discount  = 0
where productid
in (select productid from products where discontinued = 1);
```

OUTPUT

```
21:46:57 update order_details set discount  = 0 where productid in   (select productid     from
products     where discontinued = 1)    0 row(s) affected Rows matched: 228  Changed: 0
Warnings: 0             0.016 sec
```

## CALCULATE THE AVERAGE FREIGHT COST FOR EACH SHIPPER AND SHOW ONLY THOSE WITH ABOVE-AVERAGE COSTS.

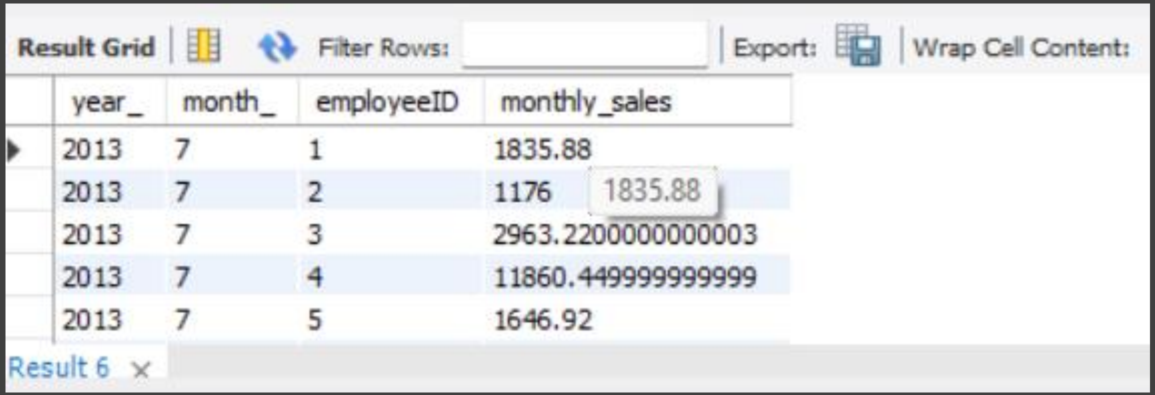
```
select shipperid , round(avg(freight),2) as avg_freight
from orders
group by shipperid
having avg_freight > (select avg(freight) from orders)
order by shipperid;
```



| shipperid | avg_freight |
|---|---|
| 2 | 86.64 |
| 3 | 80.44 |

## FIND THE MONTHLY SALES REVENUE GENERATED BY EACH EMPLOYEE.

```
select  year(o.orderdate) year_ , month(o.orderdate) month_ , o.employeeID , sum(od.quantity *
              od.unitPrice * (1 - od.discount)) as monthly_sales
from orders o
join order_details od
on o.orderid = od.ordered
group by year_ , month_ , employeeID
order by  year_ , month_ , employeeID;
```



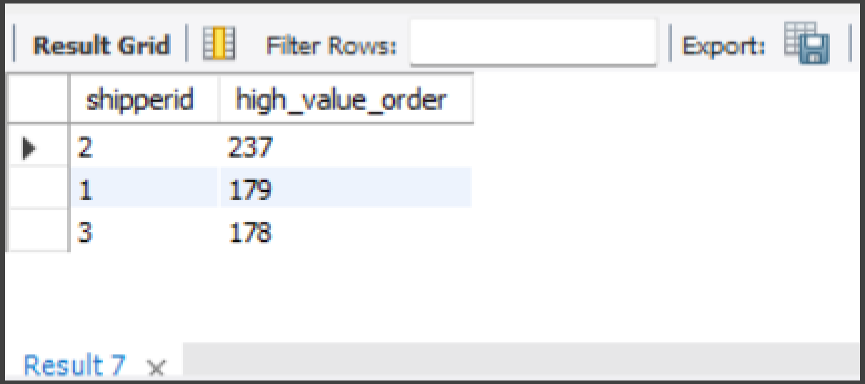
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| year_ | month_ | employeeID | monthly_sales |
|---|---|---|---|
| 2013 | 7 | 1 | 1835.88 |
| 2013 | 7 | 2 | 1176 |
| 2013 | 7 | 3 | 2963.2200000000003 |
| 2013 | 7 | 4 | 11860.449999999999 |
| 2013 | 7 | 5 | 1646.92 |

Result 6

## Identify which shipping company handles the most orders where the total order value exceeds $500.

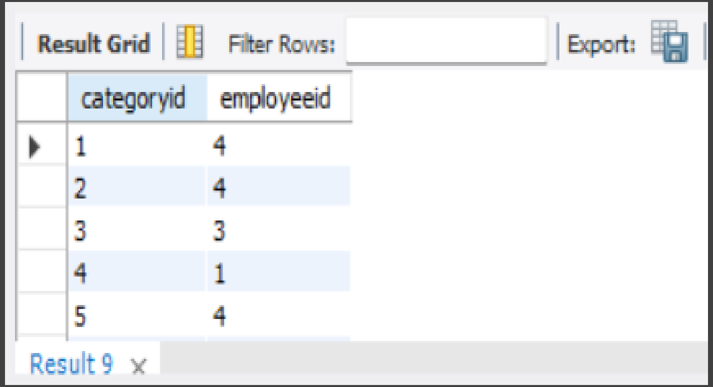
```
with above_500_orders as
        (select od.orderid  , sum(od.quantity * od.unitPrice * (1 - od.discount)) as
        total_order_value
        from order_details od
        group by od.orderID
        having total_order_value > 500)
select o.shipperid , count(*) as high_value_order
from above_500_orders a5o
join orders o
on a5o.orderid = o.ordered
group by o.shipperID
order by high_value_order desc;
```



```
shipperid  high_value_order
2          237
1          179
3          178
```

## Identify which employee has the highest sales (in terms of revenue) in each product category.

```
with cat_emp_sales as
            (select cat.categoryID , e.employeeID , sum(od.quantity * od.unitPrice * (1 -
            od.discount)) emp_cat_sales
             from categories cat
             join products p on cat.categoryID = p.categoryID
             join order_details od on od.productID = p.productID
             join orders o on o.orderID = od.orderID
            join employees e on e.employeeID = o.employeeID
            group by cat.categoryID , e.employeeID
            order by cat.categoryID , e.employeeID)
select ces.categoryid , ces.employeeid
from cat_emp_sales ces
join(select ces2.categoryid , max(emp_cat_sales) as max_sales
            from cat_emp_sales ces2
group by ces2.categoryid ) as cat_max_sales
on  ces.categoryid = cat_max_sales.categoryid and ces.emp_cat_sales =
cat_max_sales.max_sales;
```



| categoryid | employeeid |
|---|---|
| 1 | 4 |
| 2 | 4 |
| 3 | 3 |
| 4 | 1 |
| 5 | 4 |

## CONCLUSION

This project effectively demonstrates the use of SQL to extract actionable insights from a retail database. Key findings include identifying top-performing employees, high-value shippers, and top-spending customers, as well as tracking monthly revenue and analyzing freight costs. Updates to discontinued product discounts ensured data consistency, while revenue and demand analysis highlighted opportunities for inventory optimization and targeted marketing.

Overall, the project showcases how SQL can transform raw data into meaningful insights, enabling data-driven decisions to enhance business performance and efficiency.

THANK YOU!