Classicmodels

Q1. 1. Deep dive into product performance: Using window functions, calculate the percent of total yearly sales for each product within its product line for each year. This metric helps in understanding how each product is performing compared to others in the same product line.

Ans. select productlinesales.productline,products.productname,productlinesales.orderyear,round(percent\_rank()

over(partition by productlinesales.productline order by productlinesales.orderyear),2)percent

from products natural join productlinesales;

Q2. Understanding customer behavior: Compute the running total of sales for each customer for each year. This can give insights into the purchasing behavior of customers.

Ans. select customers.customerName,count(\*) as tot\_count

from customers

natural join payments

group by 1 order by 1;

Q3. Identifying top customers: For each year, identify the top 5 customers by sales

Ans. select customername,ordernumber,year(orderdate),sum(quantityordered\*priceeach)over(partition by orderdate) Sales from orders natural join orderdetails natural join customers group by orderdate;

Q2.1 Analyzing seasonal trends: Calculate the monthly sales for each year. Can you see any seasonal trends?

Ans. select customername,ordernumber,month(orderdate)as o\_month,year(orderdate)as o\_year,sum(quantityordered\*priceeach)over(partition by orderdate) Sales from orders natural join orderdetails natural join customers group by o\_month,o\_year;

Q2.2 Product line performance: Calculate the year-over-year growth rate in sales for each product line.

Ans.

Hr database.

Use a Window Function to calculate the average salary in each department, and rank the departments based on their average salaries in descending order.

Ans- select department\_name,round(avg(salary),2),rank() over(partition by department\_name) from departments inner join employees on employees.department\_id=departments.department\_id group by department\_name;