ASSIGNMENT ON SQL(Analyzing a Pizza outlet Dataset)

1. Retrieve the top 3 pizzas (Pizza\_Sold) based on the total revenue generated, Include the pizza name, total revenue, and the sales target.

select sales\_table.Pizza\_Sold,sales\_target.Sales\_Target,sum(Revenue) as Total\_Revenue

from sales\_table join sales\_target

on sales\_table.Pizza\_Sold = sales\_target.Pizza

group by sales\_table.Pizza\_Sold,sales\_target.Sales\_Target

order by Total\_Revenue desc

limit 3;

1. Compare the total revenue generated for each type of pizza (Pizza\_Sold) across all branches. Display the pizza name, total revenue, and the average revenue for each pizza.

SELECT sales\_table.Pizza\_Sold,sales\_table.Branch,sum(Revenue) as Total\_Revenue,avg(Revenue) as Average\_Revenue

from sales\_table

group by sales\_table.Pizza\_Sold,sales\_table.Branch;

1. Identify pizzas (Pizza\_Sold) that are below the average sales target and have generated revenue less than the overall average revenue.

select sales\_table.Pizza\_Sold, sum(sales\_table.Revenue) as Total\_Revenue, avg(sales\_table.Revenue) as Average\_Revenue,

avg(sales\_target.Sales\_Target) as Average\_Sales\_Target

from sales\_table

join sales\_target

on sales\_table.Pizza\_Sold = sales\_target.Pizza

where sales\_target.Pizza < (select avg(sales\_target.Sales\_Target) from sales\_target)

group by sales\_table.pizza\_sold

having Total\_Revenue < (select avg(sales\_table.Revenue) from sales\_table);

1. Find pizzas (Pizza\_Sold) that have a price higher than the overall average price and a sales target greater than the overall average target. Exclude the 'Ikoyi' branch.

SELECT sales\_table.Pizza\_Sold,sales\_target.Sales\_Target,sales\_table.Branch,avg(Price) as Average\_Price, avg(Sales\_Target) as Average\_Sales\_Target

from sales\_table join sales\_target

on sales\_table.Pizza\_Sold = sales\_target.Pizza

where sales\_table.Branch <> 'Ikoyi' and sales\_table.Price > (select avg(Price) from sales\_table)

and sales\_target.Sales\_Target > (select avg(Sales\_Target) from sales\_target)

group by sales\_table.Pizza\_Sold,sales\_target.Sales\_Target,sales\_table.Branch;

1. Determine the top-performing pizza (Pizza\_Sold) in each branch based on total revenue. Display the branch, pizza name, and total revenue for each top performer.

SELECT sales\_table.Pizza\_Sold,sales\_table.Branch,sum(Revenue) as Total\_Revenue

from sales\_table

group by Pizza\_Sold,Branch

order by Total\_Revenue desc;

1. Retrieve the daily revenue, the daily sales target, and the variance between the revenue and target for each day. Include the date and the calculated variance.

select sales\_table.Revenue, sales\_table.Date, daily\_sales\_target.Target, sales\_table.Revenue - daily\_sales\_target.Target as Variance

from sales\_table

join daily\_sales\_target

on sales\_table.Date = daily\_sales\_target.Day;

1. Find the days where the total revenue exceeds the daily sales target. Include the date, total revenue, and daily sales target for each exceeding day.

Select sales\_table.`date`,sum(Revenue) as Total\_Revenue,daily\_sales\_target.Target

from sales\_table join daily\_sales\_target on sales\_table.`Date`=daily\_sales\_target.`day`

group by sales\_table.`Date`,daily\_sales\_target.Target

having Total\_Revenue > daily\_sales\_target.Target;

1. Calculate the average daily revenue and identify days where the revenue is above the average. Display the date, total revenue, and daily sales target for each above-average day.

SELECT sales\_table.`Date`,sum(Revenue) as Total\_Revenue,avg(Revenue) as Average\_Revenue,sum(Target) as Daily\_Sales\_Target

from sales\_table join daily\_sales\_target

on sales\_table.`Date`= daily\_sales\_target.`Day`

group by sales\_table.`date`

having avg(sales\_table.Revenue) >(select avg(sales\_table.Revenue) from sales\_table)

order by Total\_Revenue desc;

1. List the days where the total revenue falls below the daily sales target. Include the date, total revenue, and daily sales target for each day below the target.

Select sales\_table.`date`,sum(Revenue) as Total\_Revenue,daily\_sales\_target.Target

from sales\_table join daily\_sales\_target on sales\_table.`Date`=daily\_sales\_target.`day`

group by sales\_table.`Date`,daily\_sales\_target.Target

having Total\_Revenue < daily\_sales\_target.Target;