Question	SQL_Query
KPI achievements Line items per bill	Sample SQL Query:----- SELECT sf.RepId, sf.InvoiceNo, COUNT(DISTINCT sf.ProductCode) AS LineItemsPerBill FROM sales_flat sf WHERE sf.Date >= DATE_FORMAT(CURRENT_DATE(), '%Y-%m-01') AND sf.Date < LAST_DAY(CURRENT_DATE()) + INTERVAL 1 DAY GROUP BY sf.InvoiceNo, sf.RepId; Key factors:----- - SKU refers to the number of unique products (SKUs) engaged or sold during that interaction. - When generating SQL queries, if you refer to columns using a table alias (e.g., sf.column_name), always define that alias by including the full table name followed by a meaningful short alias (e.g., sales_flat AS sf). The alias should be a logical abbreviation of the table name for clarity and consistency. - If the user doesn't mention any time period, do not use a time filter in the SQL query. Tables: - sales_flat(Date, InvoiceNo, ProductCode, ProductName, Qty, Discount, NetValue, Brand, CustomerCategory, ProductCategory, Route, RepId, RepCode, RepName, CustomerCode, CustomerName,ASM, RSM, Distributor, Type(sales,return)) - sales_hierarchy_nodes(Id, Code, Name) - (This is Rep table) - planned_routes(PlannedDate, RouteId, RepId) - sales_targets(Type, Value, Qty, ProductId, StartDate, EndDate) - route_customer_assignments(RouteId, CustomerID) - Products(Id, Code, Name, PackSize) - external_parties(Id, Code, Name,CreatedDate) is this customer table. CreatedDate is added when new customer is created. Table Relationships: - sales_targets.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.ProductCode = products.Code - sales_Flat.CustomerCode = external_parties.Code
KPI achievements Total Bills	Sample SQL Query:----- SELECT sf.Date, sf.RepId, COUNT(DISTINCT sf.InvoiceNo) AS TotalBills FROM sales_flat sf WHERE sf.Date >= DATE_FORMAT(CURRENT_DATE(), '%Y-%m-01') AND sf.Date < LAST_DAY(CURRENT_DATE()) + INTERVAL 1 DAY GROUP BY sf.Date, sf.RepId ORDER BY TotalBills DESC LIMIT 10; Key factors:----- - SKU refers to the number of unique products (SKUs) engaged or sold during that interaction. - When generating SQL queries, if you refer to columns using a table alias (e.g., sf.column_name), always define that alias by including the full table name followed by a meaningful short alias (e.g., sales_flat AS sf). The alias should be a logical abbreviation of the table name for clarity and consistency. - If the user doesn't mention any time period, do not use a time filter in the SQL query. Tables: - sales_flat(Date, InvoiceNo, ProductCode, ProductName, Qty, Discount, NetValue, Brand, CustomerCategory, ProductCategory, Route, RepId, RepCode, RepName, CustomerCode, CustomerName,ASM, RSM, Distributor, Type(sales,return)) - sales_hierarchy_nodes(Id, Code, Name) - (This is Rep table) - planned_routes(PlannedDate, RouteId, RepId) - sales_targets(Type, Value, Qty, ProductId, StartDate, EndDate) - route_customer_assignments(RouteId, CustomerID) - Products(Id, Code, Name, PackSize) - external_parties(Id, Code, Name,CreatedDate) is this customer table. CreatedDate is added when new customer is created. Table Relationships: - sales_targets.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.ProductCode = products.Code - sales_Flat.CustomerCode = external_parties.Code
KPI achievements per bill value / volume	Sample SQL Query:----- SELECT rh.Name AS RepName, IFNULL(S.TotalSales, 0) / NULLIF(T.TotalTarget, 0) AS KPI_Per_Bill_Value, IFNULL(S.TotalVolume, 0) AS KPI_Volume (IFNULL(S.TotalSales, 0) / NULLIF(T.TotalTarget, 0)) / NULLIF(IFNULL(S.TotalVolume, 0), 0) AS KPI FROM sales_hierarchy_nodes rh LEFT JOIN (SELECT sf.RepId, SUM(CASE WHEN sf.Type = 'sales' THEN sf.NetValue ELSE 0 END) AS TotalSales, SUM(sf.Qty) AS TotalVolume FROM sales_flat sf WHERE DATE(sf.Date) >= DATE_FORMAT(CURRENT_DATE(), '%Y-%m-01') GROUP BY sf.RepId) S ON rh.Id = S.RepId LEFT JOIN (SELECT st.RepId, SUM(st.Value) AS TotalTarget FROM sales_targets st WHERE st.StartDate <= CURRENT_DATE() AND st.EndDate >= CURRENT_DATE() AND st.Type = 1 GROUP BY st.RepId) T ON rh.Id = T.RepId ORDER BY KPI_Per_Bill_Value DESC LIMIT 10; Key factors:----- - When generating SQL queries, if you refer to columns using a table alias (e.g., sf.column_name), always define that alias by including the full table name followed by a meaningful short alias (e.g., sales_flat AS sf). The alias should be a logical abbreviation of the table name for clarity and consistency. - If the user doesn't mention any time period, do not use a time filter in the SQL query. - In the sales_hierarchy_nodes table, the Type field indicates the level of the sales target. When Type = 0, it refers to a primary target, which is the sales target assigned to the distributor. When Type = 1, it refers to a secondary target, which is the target allocated to the sales representative (rep). Aggregated metrics must be separately grouped by Rep before josining with Rep-level information. Joins between differently grouped data must happen only after summarizing to the same granularity level (Rep level). Sales and Targets must be calculated separately and only joined after aggregation is complete. Avoid causing row duplication due to joins between differently grouped datasets. Validate that no direct joins happen between daily grouped data (Productive Calls) and sales_flat invoice data.) Tables: - sales_flat(Date, InvoiceNo, ProductCode, ProductName, Qty, Discount, NetValue, Brand, CustomerCategory, ProductCategory, Route, RepId, RepCode, RepName, CustomerCode, CustomerName,ASM, RSM, Distributor, Type(sales,return)) - sales_hierarchy_nodes(Id, Code, Name) - (This is Rep table) - planned_routes(PlannedDate, RouteId, RepId) - sales_targets(Type, Value, Qty, ProductId, StartDate, EndDate) - route_customer_assignments(RouteId, CustomerID) - Products(Id, Code, Name, PackSize) - external_parties(Id, Code, Name,CreatedDate) is this customer table. CreatedDate is added when new customer is created. Table Relationships: - sales_targets.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.ProductCode = products.Code - sales_Flat.CustomerCode = external_parties.Code
Zero sales outlets	Sample SQL Query:----- SELECT ep.Name AS CustomerName FROM external_parties ep where code not in (select CustomerCode from sales_flat); Key factors:----- - When generating SQL queries, if you refer to columns using a table alias (e.g., sf.column_name), always define that alias by including the full table name followed by a meaningful short alias (e.g., sales_flat AS sf). The alias should be a logical abbreviation of the table name for clarity and consistency. - If the user doesn't mention any time period, do not use a time filter in the SQL query. Tables: - sales_flat(Date, InvoiceNo, ProductCode, ProductName, Qty, Discount, NetValue, Brand, CustomerCategory, ProductCategory, Route, RepId, RepCode, RepName, CustomerCode, CustomerName,ASM, RSM, Distributor, Type(sales,return)) - sales_hierarchy_nodes(Id, Code, Name) - (This is Rep table) - planned_routes(PlannedDate, RouteId, RepId) - sales_targets(Type, Value, Qty, ProductId, StartDate, EndDate) - route_customer_assignments(RouteId, CustomerID) - Products(Id, Code, Name, PackSize) - external_parties(Id, Code, Name,CreatedDate) is this customer table. CreatedDate is added when new customer is created. Table Relationships: - sales_targets.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.ProductCode = products.Code - sales_Flat.CustomerCode = external_parties.Code

Rep wise top 10 best performance outlet	Sample SQL Query:-----
	
	WITH OutletSales AS (
	    SELECT
	        sf.CustomerCode,
	        SUM(CASE WHEN sf.Type = 'sales' THEN sf.NetValue ELSE 0 END) AS TotalSales,
	        shn.Id AS RepId,
	        shn.Name AS RepName
	    FROM
	        sales_flat sf
	    JOIN
	        sales_hierarchy_nodes shn ON sf.RepId = shn.Id
	    GROUP BY
	        shn.Id, sf.CustomerCode),
	RankedOutlets AS (
	    SELECT
	        os.CustomerCode,
	        os.TotalSales,
	        os.RepId,
	        os.RepName,
	        ROW_NUMBER() OVER (PARTITION BY os.RepId ORDER BY os.TotalSales DESC) AS SalesRank
	    FROM
	        OutletSales os
	)
	SELECT
	    RepName,
	    CustomerCode,
	    TotalSales,
	    SalesRank
	FROM
	    RankedOutlets
	WHERE
	    SalesRank <= 10
	ORDER BY
	    RepId, TotalSales DESC;
	
	Key factors:-----
	
	- Sales = Net Sales calculated as select sum(case when fv.Type = 'sales' then (fv.netvalue) else -fv.netvalue end) from Sales_Flat fv;
	- When generating SQL queries, if you refer to columns using a table alias (e.g., sf.column_name), always define that alias by including the full table name followed by a meaningful short alias (e.g., sales_flat AS sf). The alias should be a logical abbreviation of the table name for clarity and consistency.
	- In the sales_hierarchy_nodes table, the Type field indicates the level of the sales target. When Type = 0, it refers to a primary target, which is the sales target assigned to the distributor. When Type = 1, it refers to a secondary target, which is the target allocated to the sales representative (rep).
	- If the user doesn't mention any time period, do not use a time filter in the SQL query.
	- In Sales_Flat table, CustomerName = Outlet Name
	
	Aggregated metrics must be separately grouped by Rep before josining with Rep-level information.
	Joins between differently grouped data must happen only after summarizing to the same granularity level (Rep level).
	Sales and Targets must be calculated separately and only joined after aggregation is complete.
	Avoid causing row duplication due to joins between differently grouped datasets.
	
	Tables:
	- sales_flat(Date, InvoiceNo, ProductCode, ProductName, Qty, Discount, NetValue, Brand, CustomerCategory, ProductCategory, Route, RepId, RepCode, RepName, CustomerCode, CustomerName,ASM, RSM, Distributor, Type(sales,return))
	- sales_hierarchy_nodes(Id, Code, Name) - (This is Rep table)
	- planned_routes(PlannedDate, RouteId, RepId)
	- sales_targets(Type, Value, Qty, ProductId, StartDate, EndDate)
	- route_customer_assignments(RouteId, CustomerID)
	- Products(Id, Code, Name, PackSize)
	- external_parties(Id, Code, Name,CreatedDate) is this customer table. CreatedDate is added when new customer is created.
	
	Table Relationships:
	- sales_targets.RepId = sales_Hierarchy_Nodes.Id
	- sales_Flat.RepId = sales_Hierarchy_Nodes.Id
	- sales_Flat.ProductCode = products.Code
	- sales_Flat.CustomerCode = external_parties.Code

What are the product wise discounts promotions that can be offered to	Sample SQL Query:----- SELECT sf.ProductName AS ProductName, sf.CustomerName AS OutletName, AVG(sf.Discount) AS AverageDiscount FROM sales_flat AS sf WHERE sf.Type = 'sales' GROUP BY sf.CustomerName,sf.ProductName; Key factors:----- - When generating SQL queries, if you refer to columns using a table alias (e.g., sf.column_name), always define that alias by including the full table name followed by a meaningful short alias (e.g., sales_flat AS sf). The alias should be a logical abbreviation of the table name for clarity and consistency. - If the user doesn't mention any time period, do not use a time filter in the SQL query. - In Sales_Flat table, CustomerName = Outlet Name Tables: - sales_flat(Date, InvoiceNo, ProductCode, ProductName, Qty, Discount, NetValue, Brand, CustomerCategory, ProductCategory, Route, RepId, RepCode, RepName, CustomerCode, CustomerName,ASM, RSM, Distributor, Type(sales,return)) - sales_hierarchy_nodes(Id, Code, Name) - (This is Rep table) - planned_routes(PlannedDate, RouteId, RepId) - sales_targets(Type, Value, Qty, ProductId, StartDate, EndDate) - route_customer_assignments(RouteId, CustomerID) - Products(Id, Code, Name, PackSize) - external_parties(Id, Code, Name,CreatedDate) is this customer table. CreatedDate is added when new customer is created. Table Relationships: - sales_targets.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.ProductCode = products.Code - sales_Flat.CustomerCode = external_parties.Code
Minimum sales volume to be made based on history sales	Sample SQL Query:----- SELECT sf.CustomerName AS OutletName, SUM(CASE WHEN sf.Type = 'sales' THEN sf.NetValue ELSE -sf.NetValue END) AS TotalSales FROM sales_flat AS sf WHERE MONTH(sf.Date) = MONTH(CURRENT_DATE()) AND YEAR(sf.Date) = YEAR(CURRENT_DATE()) GROUP BY sf.CustomerName ORDER BY TotalSales ASC LIMIT 10; Key factors:----- - Sales = Net Sales calculated as select sum(case when fv.Type = 'sales' then (fv.netvalue) else -fv.netvalue end) from Sales_Flat fv; - When generating SQL queries, if you refer to columns using a table alias (e.g., sf.column_name), always define that alias by including the full table name followed by a meaningful short alias (e.g., sales_flat AS sf). The alias should be a logical abbreviation of the table name for clarity and consistency. - If the user asks anything like ""based on historical"" or ""past data"", return results using currently available data only. Example: User input: ""Based on historical data, what is the most suitable outlet to visit?"" LLM output: (Return the top-performing outlets from current data only — no explanation.) - If the user doesn't mention any time period, do not use a time filter in the SQL query. - In Sales_Flat table, CustomerName = Outlet Name Tables: - sales_flat(Date, InvoiceNo, ProductCode, ProductName, Qty, Discount, NetValue, Brand, CustomerCategory, ProductCategory, Route, RepId, RepCode, RepName, CustomerCode, CustomerName,ASM, RSM, Distributor, Type(sales,return)) - sales_hierarchy_nodes(Id, Code, Name) - (This is Rep table) - planned_routes(PlannedDate, RouteId, RepId) - sales_targets(Type, Value, Qty, ProductId, StartDate, EndDate) - route_customer_assignments(RouteId, CustomerID) - Products(Id, Code, Name, PackSize) - external_parties(Id, Code, Name,CreatedDate) is this customer table. CreatedDate is added when new customer is created. Table Relationships: - sales_targets.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.ProductCode = products.Code - sales_Flat.CustomerCode = external_parties.Code
What are the best-selling SKUs at the Thashvin St outlet	Sample SQL Query:----- SELECT fv.ProductCode AS SKU, SUM(CASE WHEN fv.Type = 'sales' THEN fv.NetValue ELSE -fv.NetValue END) AS TotalSales FROM sales_flat AS fv WHERE fv.CustomerName = 'Thashvin St' GROUP BY fv.ProductCode ORDER BY TotalSales DESC; Key factors:----- - Sales = Net Sales calculated as select sum(case when fv.Type = 'sales' then (fv.netvalue) else -fv.netvalue end) from Sales_Flat fv; - When generating SQL queries, if you refer to columns using a table alias (e.g., sf.column_name), always define that alias by including the full table name followed by a meaningful short alias (e.g., sales_flat AS sf). The alias should be a logical abbreviation of the table name for clarity and consistency. - SKU refers to the number of unique products (SKUs) engaged or sold during that interaction. - If the user doesn't mention any time period, do not use a time filter in the SQL query. Tables: - sales_flat(Date, InvoiceNo, ProductCode, ProductName, Qty, Discount, NetValue, Brand, CustomerCategory, ProductCategory, Route, RepId, RepCode, RepName, CustomerCode, CustomerName,ASM, RSM, Distributor, Type(sales,return)) - sales_hierarchy_nodes(Id, Code, Name) - (This is Rep table) - planned_routes(PlannedDate, RouteId, RepId) - sales_targets(Type, Value, Qty, ProductId, StartDate, EndDate) - route_customer_assignments(RouteId, CustomerID) - Products(Id, Code, Name, PackSize) - external_parties(Id, Code, Name,CreatedDate) is this customer table. CreatedDate is added when new customer is created. Table Relationships: - sales_targets.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.ProductCode = products.Code - sales_Flat.CustomerCode = external_parties.Code

What are the potential outlets for rep Baticalo Rep 17 to gain a better	Sample SQL Query:----- SELECT sf.CustomerName AS OutletName, SUM(CASE WHEN sf.Type = 'sales' THEN sf.NetValue ELSE -sf.NetValue END) AS TotalSales FROM sales_flat AS sf WHERE sf.RepName = 'Baticalo Rep 17' GROUP BY sf.CustomerName ORDER BY TotalSales DESC LIMIT 5; Key factors:----- - Sales = Net Sales calculated as select sum(case when fv.Type = 'sales' then (fv.netvalue) else -fv.netvalue end) from Sales_Flat fv; - When generating SQL queries, if you refer to columns using a table alias (e.g., sf.column_name), always define that alias by including the full table name followed by a meaningful short alias (e.g., sales_flat AS sf). The alias should be a logical abbreviation of the table name for clarity and consistency. - If the user asks anything like ""based on historical"" or ""past data"", return results using currently available data only. Example: User input: ""Based on historical data, what is the most suitable outlet to visit?"" LLM output: (Return the top-performing outlets from current data only — no explanation.) - If the user doesn't mention any time period, do not use a time filter in the SQL query. - In Sales_Flat table, CustomerName = Outlet Name Tables: - sales_flat(Date, InvoiceNo, ProductCode, ProductName, Qty, Discount, NetValue, Brand, CustomerCategory, ProductCategory, Route, RepId, RepCode, RepName, CustomerCode, CustomerName,ASM, RSM, Distributor, Type(sales,return)) - sales_hierarchy_nodes(Id, Code, Name) - (This is Rep table) - planned_routes(PlannedDate, RouteId, RepId) - sales_targets(Type, Value, Qty, ProductId, StartDate, EndDate) - route_customer_assignments(RouteId, CustomerID) - Products(Id, Code, Name, PackSize) - external_parties(Id, Code, Name,CreatedDate) is this customer table. CreatedDate is added when new customer is created. Table Relationships: - sales_targets.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.ProductCode = products.Code - sales_Flat.CustomerCode = external_parties.Code
Market return goods trend at the outlet Baticalo Rep 17, is this outlet gi	Sample SQL Query:----- SELECT sf.ProductName AS PackSize, SUM(CASE WHEN sf.Type = 'return' THEN sf.Qty ELSE 0 END) AS TotalReturns FROM sales_flat AS sf WHERE sf.RepName = 'Baticalo Rep 17' GROUP BY sf.ProductName HAVING TotalReturns > 0 ORDER BY TotalReturns DESC; Key factors:----- - If the user doesn't mention any time period, do not use a time filter in the SQL query. - When generating SQL queries, if you refer to columns using a table alias (e.g., sf.column_name), always define that alias by including the full table name followed by a meaningful short alias (e.g., sales_flat AS sf). The alias should be a logical abbreviation of the table name for clarity and consistency. Tables: - sales_flat(Date, InvoiceNo, ProductCode, ProductName, Qty, Discount, NetValue, Brand, CustomerCategory, ProductCategory, Route, RepId, RepCode, RepName, CustomerCode, CustomerName,ASM, RSM, Distributor, Type(sales,return)) - sales_hierarchy_nodes(Id, Code, Name) - (This is Rep table) - planned_routes(PlannedDate, RouteId, RepId) - sales_targets(Type, Value, Qty, ProductId, StartDate, EndDate) - route_customer_assignments(RouteId, CustomerID) - Products(Id, Code, Name, PackSize) - external_parties(Id, Code, Name,CreatedDate) is this customer table. CreatedDate is added when new customer is created. Table Relationships: - sales_targets.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.ProductCode = products.Code - sales_Flat.CustomerCode = external_parties.Code
What are the best performing outlets for rep 'Baticalo Rep 17' for last 8	Sample SQL Query:----- SELECT sf.CustomerName AS OutletName, SUM(CASE WHEN sf.Type = 'sales' THEN sf.NetValue ELSE -sf.NetValue END) AS TotalSales FROM sales_flat AS sf WHERE sf.RepName = 'Baticalo Rep 17' AND sf.Date >= DATE_SUB(CURDATE(), INTERVAL 8 MONTH) GROUP BY sf.CustomerName ORDER BY TotalSales DESC LIMIT 5; Key factors:----- - If the user doesn't mention any time period, do not use a time filter in the SQL query. - Sales = Net Sales calculated as select sum(case when fv.Type = 'sales' then (fv.netvalue) else -fv.netvalue end) from Sales_Flat fv; - When generating SQL queries, if you refer to columns using a table alias (e.g., sf.column_name), always define that alias by including the full table name followed by a meaningful short alias (e.g., sales_flat AS sf). The alias should be a logical abbreviation of the table name for clarity and consistency. - In Sales_Flat table, CustomerName = Outlet Name Tables: - sales_flat(Date, InvoiceNo, ProductCode, ProductName, Qty, Discount, NetValue, Brand, CustomerCategory, ProductCategory, Route, RepId, RepCode, RepName, CustomerCode, CustomerName,ASM, RSM, Distributor, Type(sales,return)) - sales_hierarchy_nodes(Id, Code, Name) - (This is Rep table) - planned_routes(PlannedDate, RouteId, RepId) - sales_targets(Type, Value, Qty, ProductId, StartDate, EndDate) - route_customer_assignments(RouteId, CustomerID) - Products(Id, Code, Name, PackSize) - external_parties(Id, Code, Name,CreatedDate) is this customer table. CreatedDate is added when new customer is created. Table Relationships: - sales_targets.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.ProductCode = products.Code - sales_Flat.CustomerCode = external_parties.Code
What are the best sales routes for 'Baticalo Rep 17' and give top 10 rou	Sample SQL Query:----- SELECT sf.Route AS Route, SUM(CASE WHEN sf.Type = 'sales' THEN sf.NetValue ELSE -sf.NetValue END) AS TotalSales FROM sales_flat AS sf WHERE sf.RepName = 'Baticalo Rep 17' AND sf.Date >= DATE_SUB(CURDATE(), INTERVAL 10 MONTH) GROUP BY sf.Route ORDER BY TotalSales DESC LIMIT 10; Key factors:----- - If the user doesn't mention any time period, do not use a time filter in the SQL query. - Sales = Net Sales calculated as select sum(case when fv.Type = 'sales' then (fv.netvalue) else -fv.netvalue end) from Sales_Flat fv; - When generating SQL queries, if you refer to columns using a table alias (e.g., sf.column_name), always define that alias by including the full table name followed by a meaningful short alias (e.g., sales_flat AS sf). The alias should be a logical abbreviation of the table name for clarity and consistency. Tables: - sales_flat(Date, InvoiceNo, ProductCode, ProductName, Qty, Discount, NetValue, Brand, CustomerCategory, ProductCategory, Route, RepId, RepCode, RepName, CustomerCode, CustomerName,ASM, RSM, Distributor, Type(sales,return)) - sales_hierarchy_nodes(Id, Code, Name) - (This is Rep table) - planned_routes(PlannedDate, RouteId, RepId) - sales_targets(Type, Value, Qty, ProductId, StartDate, EndDate) - route_customer_assignments(RouteId, CustomerID) - Products(Id, Code, Name, PackSize) - external_parties(Id, Code, Name,CreatedDate) is this customer table. CreatedDate is added when new customer is created. Table Relationships: - sales_targets.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.ProductCode = products.Code - sales_Flat.CustomerCode = external_parties.Code
Give the Average sales in Rs and Average Volume in the top 10 outlets	Sample SQL Query:----- SELECT sf.CustomerName AS Outlet, AVG(CASE WHEN sf.Type = 'sales' THEN sf.NetValue ELSE -sf.NetValue END) AS AverageSales, AVG(sf.Qty) AS AverageVolume FROM sales_flat AS sf WHERE YEAR(sf.Date) = YEAR(CURRENT_DATE) - 0 GROUP BY sf.CustomerName ORDER BY AverageSales DESC LIMIT 10; Key factors:----- - If the user doesn't mention any time period, do not use a time filter in the SQL query. - When generating SQL queries, if you refer to columns using a table alias (e.g., sf.column_name), always define that alias by including the full table name followed by a meaningful short alias (e.g., sales_flat AS sf). The alias should be a logical abbreviation of the table name for clarity and consistency. - In Sales_Flat table, CustomerName = Outlet Name - Sales = Net Sales calculated as select sum(case when fv.Type = 'sales' then (fv.netvalue) else -fv.netvalue end) from Sales_Flat fv; Tables: - sales_flat(Date, InvoiceNo, ProductCode, ProductName, Qty, Discount, NetValue, Brand, CustomerCategory, ProductCategory, Route, RepId, RepCode, RepName, CustomerCode, CustomerName,ASM, RSM, Distributor, Type(sales,return)) - sales_hierarchy_nodes(Id, Code, Name) - (This is Rep table) - planned_routes(PlannedDate, RouteId, RepId) - sales_targets(Type, Value, Qty, ProductId, StartDate, EndDate) - route_customer_assignments(RouteId, CustomerID) - Products(Id, Code, Name, PackSize) - external_parties(Id, Code, Name,CreatedDate) is this customer table. CreatedDate is added when new customer is created. Table Relationships: - sales_targets.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.RepId = sales_Hierarchy_Nodes.Id - sales_Flat.ProductCode = products.Code - sales_Flat.CustomerCode = external_parties.Code