use dataspark;

SELECT \* FROM customers;

SELECT \* FROM products;

SELECT \* FROM sales;

SELECT \* FROM stores;

SELECT \* FROM exchange\_rates;

describe customers;

describe sales;

#Customer Distribution by Continent, Gender\_1

WITH AgeCalculation AS (SELECT CustomerKey,Gender,FLOOR(DATEDIFF(CURDATE(), STR\_TO\_DATE(Birthday, '%Y-%m-%d')) / 365) AS Age,Continent,

CASE

WHEN FLOOR(DATEDIFF(CURDATE(), STR\_TO\_DATE(Birthday, '%Y-%m-%d')) / 365) < 20 THEN 'Below 20'

WHEN FLOOR(DATEDIFF(CURDATE(), STR\_TO\_DATE(Birthday, '%Y-%m-%d')) / 365) BETWEEN 20 AND 40 THEN '20-40'

ELSE 'Above 40'

END AS AgeGroup FROM customers)

SELECT Continent,Gender,AgeGroup,COUNT(CustomerKey) AS CustomerCount,

(SELECT COUNT(\*) FROM customers) AS TotalCustomers

FROM AgeCalculation

GROUP BY Continent, Gender, AgeGroup

ORDER BY CustomerCount DESC;

Continent Gender AgeGroup CustomerCount TotalCustomers

North America Male Above 40 3068 15266

North America Female Above 40 2976 15266

Europe Male Above 40 1998 15266

#purchasing pattern = frequency of purchases seasonwise, preferred products\_2

WITH OrderDetails AS (SELECT s.order\_number,s.order\_date,c.Country,s.ProductKey,p.product\_name,p.Brand,p.Color,s.Quantity,(s.Quantity \* p.unit\_price) AS TotalSpending

FROM sales s JOIN products p ON s.ProductKey = p.ProductKey JOIN customers c ON s.CustomerKey = c.CustomerKey),

SeasonalOrderDetails AS (SELECT order\_number,order\_date,Country,ProductKey,product\_name,Brand,Color,Quantity,TotalSpending,

CASE

WHEN MONTH(order\_date) IN (11, 12, 1, 2, 3) THEN 'Nov-Mar'

WHEN MONTH(order\_date) IN (4, 5, 6, 7) THEN 'Apr-Jul'

ELSE 'Aug-Oct'

END AS Season

FROM OrderDetails)

SELECT Country,Season,product\_name,Brand,Color,

COUNT(order\_number) AS TotalOrders,

SUM(Quantity) AS TotalQuantitySold,

SUM(TotalSpending) AS TotalSpending

FROM SeasonalOrderDetails GROUP BY Country, Season, product\_name, Brand, Color ORDER BY TotalOrders DESC, TotalQuantitySold DESC, TotalSpending DESC;

Country Season Product Name Brand Colour Total Order Total quantity sold Total spending

United States Nov-Mar Adventure Works Desktop PC1.60 ED160 Silver Adventure Works Silver 52 158 42652.10192871094

United States Nov-Mar Adventure Works Desktop PC2.30 MD230 White Adventure Works White 51 168 100632

United States Nov-Mar Adventure Works Desktop PC1.60 ED160 Black Adventure Works Black 51 161 43461.95196533203

# Country-wise Product Sales, Ordered by Month\_3

SELECT

c.Country,

DATE\_FORMAT(s.order\_date, '%Y-%m') AS OrderMonth,

COUNT(s.order\_number) AS TotalOrders,

SUM(s.Quantity) AS TotalQuantitySold,

SUM(s.Quantity \* p.unit\_price) AS TotalRevenueGenerated

FROM sales s

JOIN products p ON s.ProductKey = p.ProductKey

JOIN customers c ON s.CustomerKey = c.CustomerKey

GROUP BY c.Country, DATE\_FORMAT(s.order\_date, '%Y-%m')

ORDER BY DATE\_FORMAT(s.order\_date, '%Y-%m'), SUM(s.Quantity) DESC;

Country Order Month Total Order Total quantity sold TotalRevenueGenerated

United States 2016-01 292 909 249165.40946912766

United Kingdom 2016-01 104 333 96850.90934801102

Italy 2016-01 68 245 93283.09992712736

Canada 2016-01 76 239 65730.12978219986

#Sales by Currency, Considering Exchange Rates and how products are affected\_4

WITH SalesWithExchange AS (

SELECT s.order\_number,s.order\_date,s.ProductKey,p.product\_name,p.Brand,p.Color,s.Quantity,s.currency\_code,e.Exchange AS ExchangeRate,

(s.Quantity \* p.unit\_price \* e.Exchange) AS TotalSpendingInUSD

FROM sales s

JOIN products p ON s.ProductKey = p.ProductKey

JOIN exchange\_rates e ON s.currency\_code = e.Currency AND DATE(s.order\_date) = DATE(e.Date))

SELECT currency\_code,product\_name,Brand,Color,

COUNT(DISTINCT order\_number) AS TotalOrders,

SUM(Quantity) AS TotalQuantitySold,

SUM(TotalSpendingInUSD) AS TotalRevenueInUSD

FROM SalesWithExchange

GROUP BY currency\_code, product\_name, Brand, Color

ORDER BY TotalRevenueInUSD DESC, TotalQuantitySold DESC;

currency\_code product\_name Brand Color Total Orders Totalquantitysold TotalrevenueUSD

USD WWI Desktop PC2.33 X2330 Black Wide World Importers Black 92 329 302351

USD Adventure Works Desktop PC2.33 XD233 White Adventure Works White 83 272 263568

USD Adventure Works Desktop PC2.33 XD233 Black Adventure Works Black 81 267 258723

USD WWI Desktop PC2.33 X2330 White Wide World Importers White 72 259 238021

#Continent-wise and Age-wise Product Brand\_5

WITH CustomerDetails AS (

SELECT

c.CustomerKey,

c.Continent,

c.Birthday,

FLOOR(DATEDIFF(CURDATE(), STR\_TO\_DATE(c.Birthday, '%Y-%m-%d')) / 365) AS Age

FROM customers c

),

AgeGroups AS (

SELECT

CustomerKey,

Continent,

CASE

WHEN Age BETWEEN 20 AND 40 THEN '20-40'

WHEN Age > 40 THEN 'Above 40'

END AS AgeGroup

FROM CustomerDetails

WHERE Age BETWEEN 20 AND 40 OR Age > 40

),

OrderDetails AS (

SELECT

s.order\_number,

s.CustomerKey,

s.ProductKey,

p.product\_name,

p.Brand,

p.Color,

s.Quantity

FROM sales s

JOIN products p ON s.ProductKey = p.ProductKey

),

AggregatedData AS (

SELECT

ag.Continent,

ag.AgeGroup,

od.product\_name,

od.Brand,

od.Color,

COUNT(DISTINCT od.order\_number) AS TotalOrders,

SUM(od.Quantity) AS TotalQuantitySold

FROM AgeGroups ag

JOIN OrderDetails od ON ag.CustomerKey = od.CustomerKey

GROUP BY ag.Continent, ag.AgeGroup, od.product\_name, od.Brand, od.Color

)

SELECT

Continent,

AgeGroup,

product\_name,

Brand,

Color,

TotalOrders,

TotalQuantitySold

FROM AggregatedData

ORDER BY TotalOrders DESC, TotalQuantitySold DESC;

country agegroup product\_name Brand Color totalorder TotalQuantitySold

North America Above 40 Adventure Works Desktop PC2.30 MD230 Black Adventure Works Black 78 239

North America Above 40 WWI Desktop PC2.33 X2330 Brown Wide World Importers Brown 78 218

North America Above 40 WWI Desktop PC1.60 E1600 Red Wide World Importers Red 76 261

North America Above 40 Adventure Works Desktop PC2.30 MD230 White Adventure Works White 74 235

North America Above 40 WWI Desktop PC1.80 E1802 White Wide World Importers White 74 232

North America Above 40 Adventure Works Desktop PC1.60 ED160 Silver Adventure Works Silver 74 226

#Most and Least Preferred Products\_6

WITH ProductSales AS (

SELECT

p.ProductKey,

p.product\_name,

p.Brand,

p.Color,

SUM(s.Quantity) AS TotalQuantitySold,

SUM(s.Quantity \* p.unit\_price) AS TotalRevenueGenerated

FROM sales s

JOIN products p ON s.ProductKey = p.ProductKey

GROUP BY p.ProductKey, p.product\_name, p.Brand, p.Color

)

SELECT

product\_name,

Brand,

Color,

TotalQuantitySold,

TotalRevenueGenerated

FROM ProductSales

ORDER BY TotalQuantitySold DESC, TotalRevenueGenerated DESC;

product\_name Brand Color TotalQuantitySold TotalRevenueGenerated

WWI Desktop PC2.33 X2330 Black Wide World Importers Black 550 505450

WWI Desktop PC1.80 E1800 White Wide World Importers White 538 123686.1967163086

Adventure Works Desktop PC2.30 MD230 White Adventure Works White 521 312079

Adventure Works Desktop PC1.60 ED160 Black Adventure Works Black 521 140643.95635986328

Adventure Works Desktop PC1.80 ED180 Black Adventure Works Black 520 191880

Adventure Works Desktop PC2.30 MD230 Black Adventure Works Black 514 307886

#profit margins\_7

WITH ProductSales AS (

SELECT

p.ProductKey,

p.product\_name,

p.Brand,

p.Color,

p.unit\_cost,

p.unit\_price,

SUM(s.Quantity) AS TotalQuantitySold,

SUM(s.Quantity \* p.unit\_price) AS TotalRevenue,

SUM(s.Quantity \* p.unit\_cost) AS TotalCost,

SUM(s.Quantity \* (p.unit\_price - p.unit\_cost)) AS TotalProfit

FROM sales s

JOIN products p ON s.ProductKey = p.ProductKey

GROUP BY p.ProductKey, p.product\_name, p.Brand, p.Color, p.unit\_cost, p.unit\_price

)

SELECT

product\_name,

Brand,

Color,

TotalQuantitySold,

TotalRevenue,

TotalCost,

TotalProfit,

(TotalProfit / TotalRevenue) \* 100 AS ProfitMarginPercentage

FROM ProductSales

ORDER BY ProfitMarginPercentage DESC;

product\_name Brand Color TotalQuantitySold TotalRevenue TotalCost TotalProfit ProfitMarginPercentage

SV DVD 60 DVD Storage Binder L20 Black Southridge Video Black 305 6981.449813842773 2311.8999767303467 4669.549837112427 66.88510211523219

SV DVD 60 DVD Storage Binder L20 Silver Southridge Video Silver 310 7095.899810791016 2349.799976348877 4746.099834442139 66.88510211523219

Contoso DVD 60 DVD Storage Binder L20 Black Contoso Black 254 5814.059844970703 1925.319980621338 3888.7398643493652 66.88510211523219

Contoso DVD 60 DVD Storage Binder L20 Silver Contoso Silver 311 7118.789810180664 2357.379976272583 4761.409833908081 66.88510211523219

#Category Analysis\_8

WITH SalesData AS (

SELECT

p.CategoryKey,

p.SubtextKey,

p.Category,

p.Subtext,

p.product\_name,

p.Brand,

p.unit\_cost,

p.unit\_price,

SUM(s.Quantity) AS TotalQuantitySold,

SUM(s.Quantity \* p.unit\_price) AS TotalRevenue,

SUM(s.Quantity \* p.unit\_cost) AS TotalCost,

SUM(s.Quantity \* (p.unit\_price - p.unit\_cost)) AS TotalProfit

FROM sales s

JOIN products p ON s.ProductKey = p.ProductKey

GROUP BY p.CategoryKey, p.SubtextKey, p.Category, p.Subtext, p.product\_name, p.Brand, p.unit\_cost, p.unit\_price

)

SELECT

Category,

Subtext,

SUM(TotalQuantitySold) AS TotalQuantitySold,

SUM(TotalRevenue) AS TotalRevenue,

SUM(TotalCost) AS TotalCost,

SUM(TotalProfit) AS TotalProfit,

(SUM(TotalProfit) / SUM(TotalRevenue)) \* 100 AS ProfitMarginPercentage

FROM SalesData

GROUP BY Category, Subtext

ORDER BY TotalRevenue DESC, TotalProfit DESC;

Category Subtext TotalQuantitySold TotalRevenue TotalCost TotalProfit ProfitMarginPercentage

Computers Desktops 20626 9906356.492660522 4277201.17099762 5629155.321662903 56.82366999242823

TV and Video Televisions 5625 4308719.140701294 1676810.8229064941 2631908.3177948 61.08331111520131

Computers Projectors & Screens 4757 3767522 1410179.6179542542 2357342.382045746 62.57010263100642

Home Appliances Water Heaters 4563 3547822.5 1507840.1993103027 2039982.3006896973 57.499559256126744

Cameras and camcorders Camcorders 4482 3357990 1339870.2944030762 2018119.7055969238 60.09903857953489

#Store Performance\_9

WITH StoreSales AS (

SELECT

st.StoreKey,

st.Country,

st.State,

st.square\_meters,

st.open\_date,

SUM(sl.Quantity) AS TotalQuantitySold,

SUM(sl.Quantity \* p.unit\_price) AS TotalRevenue,

DATEDIFF(CURDATE(), st.open\_date) / 365 AS StoreAgeYears

FROM sales sl

JOIN stores st ON sl.StoreKey = st.StoreKey

JOIN products p ON sl.ProductKey = p.ProductKey

GROUP BY st.StoreKey, st.Country, st.State, st.square\_meters, st.open\_date

),

PerformanceMetrics AS (

SELECT

StoreKey,

Country,

State,

square\_meters,

open\_date,

TotalQuantitySold,

TotalRevenue,

StoreAgeYears,

TotalRevenue / square\_meters AS RevenuePerSquareMeter

FROM StoreSales

)

SELECT

StoreKey,

Country,

State,

square\_meters,

open\_date,

TotalQuantitySold,

TotalRevenue,

StoreAgeYears,

RevenuePerSquareMeter

FROM PerformanceMetrics

ORDER BY TotalRevenue DESC, RevenuePerSquareMeter DESC, StoreAgeYears DESC;

StoreKey Country State square\_meters open\_date TotalQuantitySold TotalRevenue StoreAgeYears RevenuePerSquareMeter

55 United States Nevada 2000 2009-12-15 4826 1417885.4081779718 14.6521 708.9427040889859

50 United States Kansas 2000 2008-03-06 4773 1394738.0586917996 16.4301 697.3690293458998

54 United States Nebraska 2000 2013-06-07 4686 1384396.2391705513 11.1726 692.1981195852757

9 Canada Northwest Territories 1500 2005-03-04 4894 1336150.0573043227 19.4384 890.7667048695484

57 United States New Mexico 1645 2010-06-03 4407 1325611.8881688714 14.1863 805.8430931117759

#Analyze sales by store location to identify high-performing regions\_10

WITH RegionalSales AS (

SELECT

st.Country,

st.State,

SUM(sl.Quantity) AS TotalQuantitySold,

SUM(sl.Quantity \* p.unit\_price) AS TotalRevenue,

COUNT(DISTINCT sl.StoreKey) AS NumberOfStores

FROM sales sl

JOIN stores st ON sl.StoreKey = st.StoreKey

JOIN products p ON sl.ProductKey = p.ProductKey

GROUP BY st.Country, st.State

),

RegionalPerformance AS (

SELECT

Country,

State,

TotalQuantitySold,

TotalRevenue,

NumberOfStores,

CASE

WHEN NumberOfStores > 0 THEN TotalRevenue / NumberOfStores

ELSE 0

END AS RevenuePerStore

FROM RegionalSales

)

SELECT

Country,

State,

TotalQuantitySold,

TotalRevenue,

NumberOfStores,

RevenuePerStore

FROM RegionalPerformance

ORDER BY TotalRevenue DESC, TotalQuantitySold DESC, RevenuePerStore DESC;

Country State TotalQuantitySold TotalRevenueNumberOfStores RevenuePerStore

United States Nevada 4826 1417885.4081779718 1 1417885.4081779718

United States Kansas 4773 1394738.0586917996 1 1394738.0586917996

United States Nebraska 4686 1384396.2391705513 1 1384396.2391705513