BlinkIT Analysis

Data set:

A screenshot of a computer

AI-generated content may be incorrect.

Data Cleaning:

1. SELECT DISTINCT Item\_Fat\_Content FROM blinkit\_data

A screenshot of a computer

AI-generated content may be incorrect.

1. SELECT \* FROM blinkit\_data

UPDATE blinkit\_data

SET Item\_Fat\_Content =

CASE

WHEN Item\_Fat\_Content IN ('Lf' , 'Low Fat') THEN 'Low Fat'

WHEN Item\_Fat\_Content = 'reg' THEN 'Regular'

ELSE Item\_Fat\_Content

END;

A screenshot of a computer

AI-generated content may be incorrect.

KPI:

1. All Metrics by Outlet Type item type.

WITH item\_summery AS(

SELECT

Item\_Type,

COUNT(\*) AS item\_count,

AVG(Rating) AS avg\_rating\_per\_item,

AVG(Total\_Sales) AS avg\_sales\_per\_item,

SUM(Total\_Sales) as total\_sale\_per\_item

FROM blinkit\_data

GROUP BY Item\_Type

),

overall AS(

SELECT

COUNT(\*) AS total\_item,

AVG(Rating) AS avg\_overall\_rating,

AVG(Total\_Sales) AS avg\_total\_sale,

SUM(Total\_Sales) AS total\_sale

FROM blinkit\_data

)

SELECT

s.Item\_Type,

s.item\_count,

o.total\_item,

CAST(S.item\_count \* 100.0 / o.total\_item AS DECIMAL(10,2)) AS contribution\_on\_total\_items,

CAST(S.avg\_rating\_per\_item AS decimal(10,1)) AS avg\_rating\_per\_item,

CAST(O.avg\_overall\_rating AS decimal(10,1)) AS avg\_overall\_rating,

CAST(S.avg\_sales\_per\_item AS decimal(10,2)) AS avg\_sales\_per\_item,

CAST(O.avg\_total\_sale AS decimal(10,2)) AS avg\_total\_sale,

CAST(S.total\_sale\_per\_item/1000000 AS decimal(10,2)) AS total\_sale\_per\_item\_MILLION,

CAST(O.total\_sale/1000000 AS decimal(10,2)) AS total\_sale\_MILLION,

CAST(S.total\_sale\_per\_item \* 100.0 / O.total\_sale AS DECIMAL(10,2)) AS ontribution\_on\_total\_sales

FROM item\_summery AS S

CROSS JOIN overall AS O

ORDER BY S.total\_sale\_per\_item DESC

A white grid with many small black and red dots

AI-generated content may be incorrect.

1. All Metrics by Outlet Type for ‘Low Fat’

WITH item\_summary AS (

SELECT

Item\_Type,

COUNT(\*) AS item\_count,

AVG(Rating) AS avg\_rating\_per\_item,

AVG(Total\_Sales) AS avg\_sales\_per\_item,

SUM(Total\_Sales) AS total\_sale\_per\_item

FROM blinkit\_data

WHERE Item\_Fat\_Content = 'Low Fat'

GROUP BY Item\_Type

),

overall AS (

SELECT

COUNT(\*) AS total\_item,

AVG(Rating) AS avg\_overall\_rating,

AVG(Total\_Sales) AS avg\_total\_sale,

SUM(Total\_Sales) AS total\_sale

FROM blinkit\_data

WHERE Item\_Fat\_Content = 'Low Fat'

)

SELECT

S.Item\_Type,

S.item\_count,

O.total\_item,

CAST(S.item\_count \* 100.0 / O.total\_item AS DECIMAL(10,2)) AS contribution\_on\_total\_items,

CAST(S.avg\_rating\_per\_item AS DECIMAL(10,1)) AS avg\_rating\_per\_item,

CAST(O.avg\_overall\_rating AS DECIMAL(10,1)) AS avg\_overall\_rating,

CAST(S.avg\_sales\_per\_item AS DECIMAL(10,2)) AS avg\_sales\_per\_item,

CAST(O.avg\_total\_sale AS DECIMAL(10,2)) AS avg\_total\_sale,

CAST(S.total\_sale\_per\_item AS DECIMAL(10,2)) AS total\_sale\_per\_item,

CAST(O.total\_sale AS DECIMAL(10,2)) AS total\_sale,

CAST(S.total\_sale\_per\_item \* 100.0 / O.total\_sale AS DECIMAL(10,2)) AS contribution\_on\_total\_sales,

'Low Fat' AS Item\_Fat\_Content

FROM item\_summary AS S

CROSS JOIN overall AS O

ORDER BY S.total\_sale\_per\_item DESC;

A screenshot of a computer

AI-generated content may be incorrect.

1. All Metrics by Outlet Type for ‘Regular Fat’

A screenshot of a computer

AI-generated content may be incorrect.

1. All Metrics by Outlet Type by Fat Content

SELECT

Item\_Fat\_Content,

COUNT(\*) AS item\_count,

SUM(COUNT(\*)) OVER () AS total\_item\_count,

CAST(COUNT(\*) \* 100.0 / SUM(COUNT(\*)) OVER () AS DECIMAL(10,2)) AS contribution\_in\_total\_items,

CAST(AVG(Rating) AS DECIMAL(10,1)) AS avg\_ratings,

SUM(Total\_Sales)/1000000 AS sales\_in\_million,

SUM(SUM(Total\_Sales)/1000000) OVER () AS total\_sales\_in\_million,

CAST(SUM(Total\_Sales) \* 100.0 / SUM(SUM(Total\_Sales)) OVER () AS DECIMAL(10,2)) AS contribution\_in\_total\_sales

FROM blinkit\_data

GROUP BY Item\_Fat\_Content;

A screenshot of a computer

AI-generated content may be incorrect.

1. Performance by Outlet Type for Total Sales:

SELECT Outlet\_Type,

COUNT(\*) AS item\_sold,

SUM(COUNT(\*)) OVER () AS total\_item\_count,

CAST(COUNT(\*) \* 100.0 / SUM(COUNT(\*)) OVER () AS DECIMAL(10,2)) AS contribution\_in\_total\_items,

CAST(AVG(Rating) AS DECIMAL(10,1)) AS avg\_ratings,

SUM(Total\_Sales)/1000000 AS sales\_in\_million,

SUM(SUM(Total\_Sales)/1000000) OVER () AS total\_sales\_in\_million,

CAST(SUM(Total\_Sales) \* 100.0 / SUM(SUM(Total\_Sales)) OVER () AS DECIMAL(10,2)) AS contribution\_in\_total\_sales

FROM blinkit\_data

GROUP BY Outlet\_Type

A screenshot of a computer

AI-generated content may be incorrect.

1. Fat Content by Outlet Location Type for Total Sales

SELECT

Outlet\_Location\_Type,

ISNULL([Low Fat],0) AS Low\_fat\_Sales,

ISNULL([Regular], 0) AS Regular\_Sales

FROM (

SELECT

Outlet\_Location\_Type,

Item\_Fat\_Content,

CAST(SUM(Total\_Sales) AS DECIMAL(10,2)) AS Total\_Sales

FROM blinkit\_data

GROUP BY Outlet\_Location\_Type, Item\_Fat\_Content) AS Source\_table

PIVOT

(

SUM(Total\_Sales)

FOR Item\_Fat\_Content IN ([Low fat],[Regular])

) AS Pivot\_table

ORDER BY Outlet\_Location\_Type

A screenshot of a computer

AI-generated content may be incorrect.

1. Fat Content by Outlet Type for Total Sales

SELECT

Outlet\_Type,

ISNULL([Low Fat], 0) AS Low\_Fat\_Sales,

ISNULL([Regular], 0) AS Regular\_Sales

FROM

(

SELECT

Outlet\_Type,

Item\_Fat\_Content,

SUM(Total\_Sales) AS Total\_Sales

FROM blinkit\_data

GROUP BY Outlet\_Type, Item\_Fat\_Content

) AS SourceTable

PIVOT

(

SUM(Total\_Sales)

FOR Item\_Fat\_Content IN ([Low Fat], [Regular])

) AS PivotTable

ORDER BY Outlet\_Type;

A screenshot of a computer

AI-generated content may be incorrect.

1. Identifying the top 5 outlets based on their total sales using a ranking window function.

WITH OutletSales AS (

SELECT

Outlet\_Identifier,

SUM(Total\_Sales) AS Total\_Outlet\_Sales

FROM

blinkit\_data

GROUP BY

Outlet\_Identifier

),

RankedOutlets AS (

SELECT

Outlet\_Identifier,

Total\_Outlet\_Sales,

RANK() OVER (ORDER BY Total\_Outlet\_Sales DESC) AS Sales\_Rank

FROM

OutletSales

)

SELECT

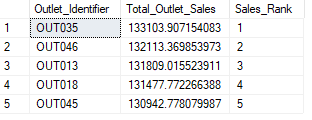
\*

FROM

RankedOutlets

WHERE

Sales\_Rank <= 5;



1. Identifying the lowest 5 outlets based on their total sales using a ranking window function.

A screenshot of a computer

AI-generated content may be incorrect.

1. Sales Performance by Outlet Size

SELECT

Outlet\_Size,

COUNT(\*) AS total\_items\_sold,

CAST(SUM(Total\_Sales) AS DECIMAL(10, 2)) AS total\_sales,

CAST(AVG(Rating) AS DECIMAL(10, 1)) AS average\_rating,

CAST(AVG(Item\_Weight) AS DECIMAL(10, 2)) AS average\_item\_weight

FROM

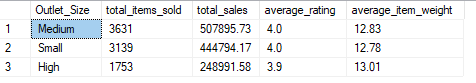
blinkit\_data

GROUP BY

Outlet\_Size

ORDER BY

total\_sales DESC;



1. SALES performance on Outlet Establishment Year

SELECT

Outlet\_Establishment\_Year,

COUNT(\*) AS Iitem\_sold,

SUM(Total\_Sales) as total\_sales

FROM blinkit\_data

GROUP BY Outlet\_Establishment\_Year

ORDER BY Outlet\_Establishment\_Year DESC

A screenshot of a data

AI-generated content may be incorrect.

1. Identifying the Top-Selling Item by Outlet Type

WITH ItemSalesRank AS (

SELECT

Outlet\_Type,

Item\_Type,

CAST(SUM(Total\_Sales) AS DECIMAL(10, 2)) AS Total\_Sales,

ROW\_NUMBER() OVER(PARTITION BY Outlet\_Type ORDER BY SUM(Total\_Sales) DESC) AS Rank\_Number

FROM

blinkit\_data

GROUP BY

Outlet\_Type,

Item\_Type

)

SELECT

Outlet\_Type,

Item\_Type,

Total\_Sales

FROM

ItemSalesRank

WHERE

Rank\_Number = 1

ORDER BY

Total\_Sales DESC;

A screenshot of a computer

AI-generated content may be incorrect.

1. Sales Trend by Outlet Type and Location Type

SELECT

Outlet\_Type,

Outlet\_Location\_Type,

CAST(SUM(Total\_Sales) AS DECIMAL(10, 2)) AS Total\_Sales,

CAST(AVG(Rating) AS DECIMAL(10, 1)) AS Average\_Rating,

COUNT(\*) AS Total\_Transactions

FROM

blinkit\_data

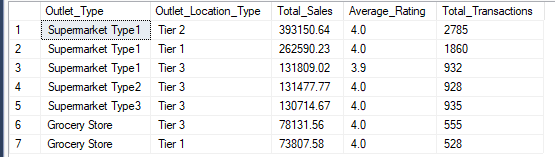
GROUP BY

Outlet\_Type,

Outlet\_Location\_Type

ORDER BY

Total\_Sales DESC;



1. Identify the highest-selling outlet for **each** item category.

WITH RankedSales AS (

SELECT

Outlet\_Identifier,

Item\_Type,

CAST(SUM(Total\_Sales) AS DECIMAL(10, 2)) AS Total\_Sales,

ROW\_NUMBER() OVER(PARTITION BY Item\_Type ORDER BY SUM(Total\_Sales) DESC) as rn

FROM

blinkit\_data

GROUP BY

Outlet\_Identifier,

Item\_Type

)

SELECT

Outlet\_Identifier,

Item\_Type,

Total\_Sales

FROM

RankedSales

WHERE rn = 1

ORDER BY

Item\_Type,

Total\_Sales DESC;

A screenshot of a data

AI-generated content may be incorrect.

1. the highest-selling **item** for each outlet.

A screenshot of a computer

AI-generated content may be incorrect.