**Metrics**

1. Average Score per Metric
2. Market Score per Metric
3. Rank questions within each Metric
4. Top 3 questions for each Metric
5. Bottom 3 questions for each Metric

/\*

Metric with custom order, via Mapping Table using INNER JOIN

\*/

SELECT

sqms.Metric

FROM

SurveyQuestion\_MarketScore\_2023 AS sqms

INNER JOIN

MetricMapping AS mm

ON

sqms.Metric = mm.Metric

GROUP BY

sqms.Metric,

mm.MetricOrder -- any column used in ORDER BY must also be included in the GROUP BY

ORDER BY

mm.MetricOrder;

/\*

Average Score per Metric

\*/

SELECT

sqms.Metric,

ROUND(AVG(CAST(sr.LikertScore AS FLOAT)), 2) AS AverageScore

FROM

SurveyQuestion\_MarketScore\_2023 AS sqms

INNER JOIN

MetricMapping AS mm

ON

sqms.Metric = mm.Metric

INNER JOIN

SurveyResponse\_2023 AS sr

ON

sqms.QuestionID = sr.QuestionID

GROUP BY

sqms.Metric,

mm.MetricOrder

ORDER BY

mm.MetricOrder;

/\*

Market Score per Metric

\*/

SELECT

sqms.Metric,

ROUND(AVG(CAST(sqms.MarketScore AS FLOAT)), 2) AS MarketScore

FROM

SurveyQuestion\_MarketScore\_2023 AS sqms

INNER JOIN

MetricMapping AS mm

ON

sqms.Metric = mm.Metric

INNER JOIN

SurveyResponse\_2023 AS sr

ON

sqms.QuestionID = sr.QuestionID

GROUP BY

sqms.Metric,

mm.MetricOrder

ORDER BY

mm.MetricOrder;

/\*

Use ROW\_NUMBER() with PARTITION BY Metric to rank questions within each Metric group, based on Average Score in DESC

PARTITION BY is to restart the row number within each Metric group

\*/

WITH CombinedData AS (

SELECT

sqms.Metric,

sqms.QuestionText,

ROUND(AVG(CAST(sr.LikertScore AS FLOAT)), 2) AS AverageScore,

mm.MetricOrder

FROM

SurveyQuestion\_MarketScore\_2023 AS sqms

INNER JOIN

MetricMapping AS mm

ON

sqms.Metric = mm.Metric

INNER JOIN

SurveyResponse\_2023 AS sr

ON

sqms.QuestionID = sr.QuestionID

GROUP BY

sqms.Metric,

sqms.QuestionText,

mm.MetricOrder)

SELECT

Metric,

QuestionText,

AverageScore,

ROW\_NUMBER() OVER(PARTITION BY Metric ORDER BY AverageScore DESC) AS RowNumber

FROM

CombinedData

ORDER BY

MetricOrder;

/\*

Top 3 questions for each Metric, multiple CTEs

\*/

WITH CombinedData AS (

SELECT

sqms.Metric,

sqms.QuestionText,

ROUND(AVG(CAST(sr.LikertScore AS FLOAT)), 2) AS AverageScore,

mm.MetricOrder

FROM

SurveyQuestion\_MarketScore\_2023 AS sqms

INNER JOIN

MetricMapping AS mm

ON

sqms.Metric = mm.Metric

INNER JOIN

SurveyResponse\_2023 AS sr

ON

sqms.QuestionID = sr.QuestionID

GROUP BY

sqms.Metric,

sqms.QuestionText,

mm.MetricOrder),

RankedQuestion AS (

SELECT

Metric,

QuestionText,

AverageScore,

ROW\_NUMBER() OVER(PARTITION BY Metric ORDER BY AverageScore DESC) AS RowNumber,

MetricOrder

FROM

CombinedData)

SELECT

Metric,

QuestionText,

AverageScore

FROM

RankedQuestion

WHERE

RowNumber <= 3

ORDER BY

MetricOrder;

/\*

Bottom 3 questions for each Metric, multiple CTEs

\*/

WITH CombinedData AS (

SELECT

sqms.Metric,

sqms.QuestionText,

ROUND(AVG(CAST(sr.LikertScore AS FLOAT)), 2) AS AverageScore,

mm.MetricOrder

FROM

SurveyQuestion\_MarketScore\_2023 AS sqms

INNER JOIN

MetricMapping AS mm

ON

sqms.Metric = mm.Metric

INNER JOIN

SurveyResponse\_2023 AS sr

ON

sqms.QuestionID = sr.QuestionID

GROUP BY

sqms.Metric,

sqms.QuestionText,

mm.MetricOrder),

RankedQuestion AS (

SELECT

Metric,

QuestionText,

AverageScore,

ROW\_NUMBER() OVER(PARTITION BY Metric ORDER BY AverageScore) AS RowNumber,

MetricOrder

FROM

CombinedData)

SELECT

Metric,

QuestionText,

AverageScore

FROM

RankedQuestion

WHERE

RowNumber <= 3

ORDER BY

MetricOrder;