

Project Title

Easy Analytics Corporation - Employee Engagement Survey 2023

Project Overview

This project analyzes employee engagement survey data using SQL. The analysis focuses on division-department engagement, key performance themes, gender-based insights and benchmarking against market data

Purpose

SQL is essential for real-world data analytics. This project shows how to extract, transform and analyze business survey data using SQL scripts, answering key questions relevant to HR and leadership

Tools used

SQL (SQL Server in SSMS)

Dataset

1. EmployeeData_2023 (DataYear, EmployeeID, DivisionName, DepartmentName, Gender, Year, Month)
2. SurveyResponse_2023 (DataYear, EmployeeID, QuestionID, LikertScore, AverageScore)
3. SurveyQuestion_MarketScore_2023 (DataYear, QuestionID, Metric, Category, QuestionText, Theme, MarketScore)
4. MetricMapping (Metric, MetricOrder)
5. CategoryMapping (Metric, Category, CategoryOrder)
6. QuestionMapping (QuestionID, QuestionText, QuestionOrder)

Business questions

1. What is the overall employee engagement score?
2. Which divisions or departments are most/least engaged?
3. Do male and female employees report different engagement levels?
4. What are the top 3 and bottom 3 questions company-wide?
5. What are the top and bottom themes?
6. How do we compare to the market?

Survey is structured into Core, Self and Group metric:

1. Core: Organizational support, systems and culture
2. Self: Individual motivation, alignment and career outlook
3. Group: Team collaboration, trust and shared responsibility

I have imported the data into SQL Server & I am using SSMS to query it

```
/*  
tell SQL Server to create a new database named EAC_EES2023, short & meaningful  
*/  
CREATE DATABASE EAC_EES2023;
```

```
/*  
tell SQL Server to use the EAC_EES2023 database for all subsequent commands, this is to ensure that we  
choose the correct database to CREATE TABLE + BULK INSERT later  
*/  
USE EAC_EES2023;
```

A **default** schema named **dbo** (database owner) is **automatically created** after a database is created
CREATE TABLE EmployeeData_2023 - - SQL Server interpret this as dbo.EmployeeData

To store data permanently in SQL Server

1. CREATE DATABASE
2. CREATE TABLE
3. BULK INSERT + FROM + WITH (for CSV file)

I set the PKs & FKs, to establish logical relationships between Tables

This is to ensure data integrity by preventing orphan records & enforcing referential integrity

1st Table: EmployeeData_2023

CREATE TABLE EmployeeData_2023

```
(
    DataYear INT NOT NULL,
    EmployeeID INT NOT NULL,
    Gender CHAR(1) NOT NULL,
    DivisionName NVARCHAR(30) NOT NULL,
    DepartmentName NVARCHAR(32) NOT NULL,
    YearsEmployed INT NOT NULL,
    MonthsEmployed INT NOT NULL,
    CONSTRAINT PK_EmployeeData_2023 PRIMARY KEY (EmployeeID)
);
```

BULK INSERT EmployeeData_2023

FROM 'C:\Users\izziiw\Downloads\EmployeeData_2023.csv'

WITH

```
(
    FORMAT = 'CSV',           -- file type
    FIRSTROW = 2,             -- skip the 1st row (header)
    FIELDTERMINATOR = ',',    -- CSV column separator
    ROWTERMINATOR = '\n',     -- each new line is new row of data
    TABLOCK                   -- lock the Table while inserting for better speed
);
```

-- data will be inserted in the Table here

-- file location

2nd Table: SurveyQuestion_MarketScore_2023

CREATE TABLE SurveyQuestion_MarketScore_2023

```
(
    DataYear INT NOT NULL,
    QuestionID NVARCHAR(3) NOT NULL,
    Metric NVARCHAR(5) NOT NULL,
    Category NVARCHAR(10) NOT NULL,
    QuestionText NVARCHAR(104) NOT NULL,
    Theme NVARCHAR(25) NOT NULL,
    MarketScore DECIMAL(3,2) NOT NULL, -- Total 3 digits, 2 after decimal with range: -9.99 to 9.99
    CONSTRAINT PK_SurveyQuestion_MarketScore_2023 PRIMARY KEY (QuestionID)
);
```

```

BULK INSERT SurveyQuestion_MarketScore_2023
FROM 'C:\Users\izziiw\Downloads\SurveyQuestion_MarketScore_2023.csv'
WITH
(
    FORMAT = 'CSV',
    FIRSTROW = 2,
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    TABLOCK
);

```

3rd Table: SurveyResponse_2023

```

CREATE TABLE SurveyResponse_2023
(
    DataYear INT NOT NULL,
    EmployeeID INT NOT NULL,
    QuestionID NVARCHAR(3) NOT NULL,
    LikertScore TINYINT NOT NULL,
    AverageScore DECIMAL (10,9) NOT NULL,
    CONSTRAINT PK_SurveyResponse_2023 PRIMARY KEY (EmployeeID, QuestionID),
    CONSTRAINT FK_SurveyResponse_2023_EmployeeID FOREIGN KEY (EmployeeID)
        REFERENCES EmployeeData_2023(EmployeeID) ON DELETE CASCADE,
    CONSTRAINT FK_SurveyResponse_2023_QuestionID FOREIGN KEY (QuestionID)
        REFERENCES SurveyQuestion_MarketScore_2023(QuestionID) ON DELETE CASCADE
);
/*
    TINYINT (1 byte, 0–255), INT (4 bytes), save space & boost speed
    add CPK to prevent duplicate responses & keep data structured, the BEST practice
    add ON DELETE CASCADE for EmployeeID & QuestionID, prevent orphaned records */

```

```

BULK INSERT SurveyResponse_2023
FROM 'C:\Users\izziiw\Downloads\SurveyResponse_2023.csv'
WITH
(
    FORMAT = 'CSV',
    FIRSTROW = 2,
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    TABLOCK
);

```

Tables for SurveyMapping

1st Table: MetricMapping

```
CREATE TABLE MetricMapping
(
    Metric NVARCHAR(5) NOT NULL,
    MetricOrder INT NOT NULL
);
```

```
INSERT INTO
    MetricMapping (Metric, MetricOrder)
VALUES
    ('Core', 1),
    ('Self', 2),
    ('Group', 3);
```

2nd Table: CategoryMapping

```
CREATE TABLE CategoryMapping
(
    Metric NVARCHAR(5) NOT NULL,
    Category NVARCHAR(10) NOT NULL,
    Metric_Category NVARCHAR(15) NOT NULL,
    CategoryOrder INT NOT NULL
);
```

```
INSERT INTO
    CategoryMapping (Metric, Category, Metric_Category, CategoryOrder)
VALUES
    ('Core', 'Leadership', 'Core-Leadership', 1),
    ('Core', 'Culture', 'Core-Culture', 2),
    ('Core', 'Initiative', 'Core-Initiative', 3),
    ('Self', 'Heart', 'Self-Heart', 1),
    ('Self', 'Mind', 'Self-Mind', 2),
    ('Self', 'Soul', 'Self-Soul', 3),
    ('Group', 'Think', 'Group-Think', 1),
    ('Group', 'Feel', 'Group-Feel', 2),
    ('Group', 'Do', 'Group-Do', 3);
```

3rd Table: QuestionMapping

```
CREATE TABLE QuestionMapping
(
    QuestionID NVARCHAR(3) NOT NULL,
    QuestionText NVARCHAR(68) NOT NULL,
    QuestionID_QuestionText NVARCHAR(72) NOT NULL,
    QuestionOrder INT NOT NULL
);
```

```
BULK INSERT QuestionMapping
FROM 'C:\Users\izziw\Downloads\QuestionMapping.csv'
WITH
(
    FORMAT = 'CSV',
    FIRSTROW = 2,
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    TABLOCK
);
```

/*

What is the overall employee engagement score?

SQL concepts used: FROM, SELECT, CAST(), AVG()

*/

SELECT

CAST(AVG(CAST(LikertScore AS DECIMAL(10,2))) / 5 * 100 AS DECIMAL (3,0)) AS
OverallEngagementScore -- convert LikertScore to decimal, then display in decimal

FROM

SurveyResponse_2023;

77

Which divisions or departments are most/least engaged?

/*

Engagement by Division, sorted by Engagement Score in DESC order including Engagement Level based on score thresholds

SQL concepts used: FROM, INNER JOIN, GROUP BY, SELECT, AVG(), CAST(), CASE, ORDER BY

*/

SELECT

ed.DivisionName,
CAST(AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 AS DECIMAL (3,0)) AS
EngagementScore,

CASE

WHEN AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 >= 80 THEN 'Highly Engaged'
WHEN AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 >= 65 THEN 'Engaged'
WHEN AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 >= 50 THEN 'Neutral'
WHEN AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 >= 25 THEN 'Disengaged'
ELSE 'Actively Disengaged'

END AS 'EngagementLevel'

FROM

EmployeeData_2023 **AS** ed

INNER JOIN

SurveyResponse_2023 **AS** sr

ON

ed.EmployeeID = sr.EmployeeID

GROUP BY

ed.DivisionName

ORDER BY

EngagementScore **DESC**;

People & Culture	82	Highly Engaged
Strategy & Planning	79	Engaged
Finance & Legal	78	Engaged
Product & Technology	78	Engaged
Marketing & Communications	77	Engaged
Business Operations & Services	77	Engaged

All divisions scored above 65%, meaning every division is at least Engaged

- People & Culture leads with 82% (Highly Engaged), showing exceptionally strong engagement
- The rest range from 77%–79%, showing healthy overall engagement

❖ Not just identifying extremes, but also recognizing overall trends

/*

Engagement by Division, sorted by Engagement Score in DESC order including Engagement Level based on score thresholds

using CTE

*/

```

WITH CombinedData AS (
    SELECT
        ed.DivisionName,
        CAST(AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 AS DECIMAL (3,0)) AS
        EngagementScore,
    CASE
        WHEN AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 >= 80 THEN 'Highly Engaged'
        WHEN AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 >= 65 THEN 'Engaged'
        WHEN AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 >= 50 THEN 'Neutral'
        WHEN AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 >= 25 THEN 'Disengaged'
        ELSE 'Actively Disengaged'
    END AS 'EngagementLevel'
    FROM
        EmployeeData_2023 AS ed
    INNER JOIN
        SurveyResponse_2023 AS sr
    ON
        ed.EmployeeID = sr.EmployeeID
    GROUP BY
        ed.DivisionName)

SELECT
    DivisionName,
    EngagementScore,
    EngagementLevel
FROM
    CombinedData
ORDER BY
    EngagementScore DESC;

```


/*

Engagement by Department, sorted by Engagement Score in DESC order including Engagement Level based on score thresholds
using CTE

*/

```
WITH CombinedData AS (
    SELECT
        ed.DepartmentName,
        CAST(AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 AS DECIMAL (3,0)) AS
        EngagementScore,
    CASE
        WHEN AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 >= 80 THEN 'Highly Engaged'
        WHEN AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 >= 65 THEN 'Engaged'
        WHEN AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 >= 50 THEN 'Neutral'
        WHEN AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 >= 25 THEN 'Disengaged'
        ELSE 'Actively Disengaged'
    END AS 'EngagementLevel'
    FROM
        EmployeeData_2023 AS ed
    INNER JOIN
        SurveyResponse_2023 AS sr
    ON
        ed.EmployeeID = sr.EmployeeID
    GROUP BY
        ed.DivisionName)

SELECT
    DepartmentName,
    EngagementScore,
    EngagementLevel
FROM
    CombinedData
ORDER BY
    EngagementScore DESC;
```

Innovation & Emerging Tech 100

Technical Support Engineering 87

- These departments fall within the Highly Engaged category ($\geq 80\%$), indicating strong alignment, motivation and satisfaction among their teams

Channel & Retail Sales 66

Client Onboarding & Activation 63

- These departments are in the 60% range, which is below the Highly Engaged threshold (80%)
- Client Onboarding & Activation (63%) is the only department still in the Neutral zone
- Channel & Retail Sales (66%) is just barely above the Engaged threshold (65%), indicating that support and motivation strategies may still be needed

/*

Do male and female employees report different engagement levels?

SQL concepts used - FROM, INNER JOIN, GROUP BY, SELECT, AVG(), CAST(), CASE, ORDER BY using CTE

*/

```
WITH CombinedData AS (  
    SELECT  
        ed.Gender,  
        CAST(AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 AS DECIMAL (3,0)) AS  
        EngagementScore,  
    CASE  
        WHEN AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 >= 80 THEN 'Highly Engaged'  
        WHEN AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 >= 65 THEN 'Engaged'  
        WHEN AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 >= 50 THEN 'Neutral'  
        WHEN AVG(CAST(sr.LikertScore AS DECIMAL(10,2))) / 5 * 100 >= 25 THEN 'Disengaged'  
        ELSE 'Actively Disengaged'  
    END AS 'EngagementLevel'  
    FROM  
        EmployeeData_2023 AS ed  
    INNER JOIN  
        SurveyResponse_2023 AS sr  
    ON  
        ed.EmployeeID = sr.EmployeeID  
    GROUP BY  
        ed.Gender)
```

```
SELECT  
    Gender,  
    EngagementScore,  
    EngagementLevel  
FROM  
    CombinedData;
```

F	77	Engaged
M	78	Engaged

The engagement level is consistent across genders, with males reporting only 1% higher engagement than females. This suggests a relatively balanced experience between groups

What are the top 3 and bottom 3 questions company-wide?

Tracked based on average scores across Core, Self and Group engagement metrics

/*

Top 3 questions for each Metric

SQL concepts used: FROM, INNER JOIN, SELECT, CAST(), AVG(), ROUND(), GROUP BY, ROW_NUMBER() with PARTITION BY, ORDER BY, 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;
```

Core	Our organization invests in continuous learning	3.96
Core	Our organization supports flexible ways of working	3.94
Core	Our organization funds clubs that build engagement	3.85
Self	I look for ways to improve my contribution	4.13
Self	I believe my role contributes to the organization's goals	4.13
Self	I see myself staying in this organization long-term	4.12
Group	We are happy to help one another	4.21
Group	We share ownership of our work	4.16
Group	We respect each other's expertise	4.15

/*

Bottom 3 questions for each Metric

SQL concepts used: FROM, INNER JOIN, SELECT, CAST(), AVG(), ROUND(), GROUP BY, ROW_NUMBER() with PARTITION BY, ORDER BY, 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;
```

Core	Our organization stays connected with former employees	3.48
Core	Our organization values people more than profit	3.49
Core	Our organization favors flexibility over rigid processes	3.6
Self	I trust that my efforts are recognized	3.75
Self	I feel my workspace is comfortable	3.76
Self	I feel energized at work	3.79
Group	We welcome diverse opinions	3.96
Group	We trust each other to deliver	4.1
Group	We respect each other's expertise	4.15

What are the top and bottom themes?

/*

Top 3 theme for each Metric

SQL concepts used: FROM, INNER JOIN, SELECT, CAST(), AVG(), ROUND(), GROUP BY, ROW_NUMBER() with PARTITION BY, ORDER BY, multiple CTEs

*/

```
WITH CombinedData AS (
    SELECT
        sqms.Metric,
        sqms.Theme,
        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.Theme,
        mm.MetricOrder),

RankedQuestion AS (
    SELECT
        Metric,
        Theme,
        AverageScore,
        ROW_NUMBER() OVER(PARTITION BY Metric ORDER BY AverageScore DESC) AS
        RowNumber,
        MetricOrder
    FROM
        CombinedData)

SELECT
    Metric,
    Theme,
    AverageScore
FROM
    RankedQuestion
WHERE
    RowNumber <= 3
ORDER BY
    MetricOrder;
```

Core	Learning & Development	3.96
Core	Work-Life Balance	3.94
Core	Empowerment	3.78
Self	Continuous Improvement	4.13
Self	Organizational Alignment	4.13
Self	Organizational Commitment	4.12
Group	Accountability	4.13
Group	Collaboration	4.11

Core:

Results reflect strong organizational support for growth and flexibility. However, Empowerment shows slightly lower sentiment compared to other Core themes, suggesting room to improve decision-making autonomy

Self:

High scores here indicate that employees feel motivated, aligned with company goals and committed to staying which is a healthy indicator of internal engagement and personal ownership

Group:

Results show a strong team culture where individuals take responsibility and collaborate effectively. This can drive high performance and knowledge sharing across functions

/*

Bottom 3 theme for each Metric

SQL concepts used: FROM, INNER JOIN, SELECT, CAST(), AVG(), ROUND(), GROUP BY, ROW_NUMBER() with PARTITION BY, ORDER BY, multiple CTEs

*/

```
WITH CombinedData AS (  
    SELECT  
        sqms.Metric,  
        sqms.Theme,  
        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.Theme,  
        mm.MetricOrder),
```

```
RankedQuestion AS (  
    SELECT  
        Metric,  
        Theme,  
        AverageScore,  
        ROW_NUMBER() OVER(PARTITION BY Metric ORDER BY AverageScore) AS  
        RowNumber,  
        MetricOrder  
    FROM  
        CombinedData)
```

```
SELECT  
    Metric,  
    Theme,  
    AverageScore  
FROM  
    RankedQuestion  
WHERE  
    RowNumber <= 3  
ORDER BY  
    MetricOrder;
```

Core	Employer Branding	3.48
Core	People-First Culture	3.49
Core	Agile Work Culture	3.6
Self	Recognition	3.75
Self	Work Environment	3.76
Self	Employee Well-Being	3.8
Group	Collaboration	4.11
Group	Accountability	4.13

Core:

Themes around culture and external reputation received the lowest sentiment in the Core category. These areas represent opportunities for brand and cultural transformation

Self:

Recognition and well-being being among the lowest may signal that employees feel underappreciated or strained. It is crucial for HR and team leads to address to prevent disengagement or burnout

Group:

Collaboration and accountability, while the lowest-ranked in this category, still reflect highly positive sentiment. This suggests that employees generally feel they work well together and take ownership, though there may still be subtle opportunities to enhance collaboration across teams or reinforce shared responsibility

/*

How do we compare to the market?

SQL concepts used: FROM, INNER JOIN, SELECT, CAST(), AVG(), ROUND(), GROUP BY, ORDER BY, multiple CTEs, CASE logic

*/

```
WITH CombinedData AS (
    SELECT
        sqms.Metric,
        ROUND(AVG(CAST(sr.LikertScore AS FLOAT)), 2) AS AverageScore,
        ROUND(AVG(CAST(sqms.MarketScore AS FLOAT)), 2) AS MarketScore,
        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,
        mm.MetricOrder)

SELECT
    Metric,
    AverageScore,
    MarketScore,
    AverageScore - MarketScore AS ScoreDifference,
    CASE
        WHEN AverageScore > MarketScore THEN 'Above market'
        WHEN AverageScore < MarketScore THEN 'Below market'
        ELSE 'Same as market'
    END AS MarketComparison
FROM
    CombinedData
ORDER BY
    MetricOrder;
```

Core	3.71	3.47	0.24	Above market
Self	3.95	3.6	0.35	Above market
Group	4.12	3.81	0.31	Above market

Across all 3 engagement metrics, the organization performs above market benchmarks