Aggregating Data

INTRODUCTION TO SQL SERVER

John MacKintosh
Instructor

DataCamp

SUM - single column

Calculate the total amount of a column value with `SUM()`

```
SELECT
  SUM(affected_customers) AS total_affected
FROM grid;
```

```
+----------------+
| total_affected |
|----------------|
| 70143996       |
+----------------+
```

SUM - two or more columns

```
SELECT
  SUM(affected_customers) AS total_affected
FROM grid;
```

```
SELECT
  SUM (affected_customers) AS total_affected,
  SUM (demand_loss_mw) AS total_loss
FROM grid;
```

```
+----------------+------------+
| total_affected | total_loss |
|----------------+------------|
| 70143996       | 177888     |
+----------------+------------+
```

The wrong way...

```
SELECT
  SUM (affected_customers) AS total_affected,
  (demand_loss_mw) AS total_loss
FROM grid;
```

```
Msg 8120, Level 16, State 1, Line 6
Column 'grid_demand_loss_mw' is invalid in the select list because
it is not contained in either an aggregate function or the GROUP BY clause.
```

Use aliases

```
SELECT
  SUM (affected_customers),
  SUM (demand_loss_mw)
FROM grid;
```

```
+------------------+------------------+
| (No column name) | (No column name) |
|------------------+------------------|
| 70143996         | 177888           |
+------------------+------------------+
```

```
SELECT
  SUM (affected_customers) AS total_affected,
  SUM (demand_loss_mw) AS total_loss
FROM grid;
```

```
+----------------+------------+
| total_affected | total_loss |
|----------------+------------|
| 70143996       | 177888     |
+----------------+------------+
```

COUNT

```
SELECT
  COUNT(affected_customers) AS count_affected
FROM grid;
```

```
+----------------+
| count_affected |
|----------------|
| 807            |
+----------------+
```

COUNT Distinct

```
SELECT
  COUNT(DISTINCT affected_customers) AS unique_count_affected
FROM grid;
```

```
+-----------------------+
| unique_count_affected |
|-----------------------|
| 280                   |
+-----------------------+
```

MIN

```
SELECT
  MIN(affected_customers) AS min_affected_custome
FROM grid;
```

```
+------------------------+
| min_affected_customers |
|------------------------|
| 0                      |
+------------------------+
```

```
SELECT
  MIN(affected_customers) AS min_affected_custome
FROM grid
WHERE affected_customers > 0;
```

```
+------------------------+
| min_affected_customers |
|------------------------|
| 1                      |
+------------------------+
```

MAX

```
SELECT
  MAX(affected_customers) AS max_affected_customers
FROM grid;
```

```
+-------------------------+
| max_affected_customers |
|-------------------------|
| 4645572                 |
+-------------------------+
```

Average

```
SELECT
  AVG(affected_customers) AS avg_affected_customers
FROM grid;
```

```
+-------------------------+
| avg_affected_customers |
|-------------------------|
| 86919                   |
+-------------------------+
```

Let's practice!

INTRODUCTION TO SQL SERVER

DataCamp

Strings

INTRODUCTION TO SQL SERVER

John MacKintosh
Instructor

DataCamp

```
SELECT
  description,
  LEN(description) AS description_length
FROM grid;
```

```
+---------------+-----------------------------------+
| description                    | description_length |
|--------------------------------+--------------------|
| Severe Weather  Thunderstorms | 29                 |
| Severe Weather  Thunderstorms | 29                 |
| Severe Weather  Thunderstorms | 29                 |
| Fuel Supply Emergency  Coal   | 27                 |
| Physical Attack  Vandalism    | 26                 |
+---------------+-----------------------------------+
```

```
SELECT
  description,
  LEFT(description, 20) AS first_20_left
FROM grid;
```

```
+------------------------------+----------------------+
| description                  | first_20_left        |
|------------------------------+----------------------|
| Severe Weather  Thunderstorms | Severe Weather  Thun  |
| Severe Weather  Thunderstorms | Severe Weather  Thun  |
| Severe Weather  Thunderstorms | Severe Weather  Thun  |
| Fuel Supply Emergency  Coal   | Fuel Supply Emergenc  |
| Physical Attack  Vandalism    | Physical Attack  Van  |
+------------------------------+----------------------+
```

```
SELECT
  description,
  RIGHT(description, 20) AS last_20
FROM grid;
```

```
+------------------------------+----------------------+
| description                  | last_20              |
|------------------------------+----------------------|
| Severe Weather  Thunderstorms | ather  Thunderstorms |
| Severe Weather  Thunderstorms | ather  Thunderstorms |
| Severe Weather  Thunderstorms | ather  Thunderstorms |
| Fuel Supply Emergency  Coal   | pply Emergency  Coal |
| Physical Attack  Vandalism    | al Attack  Vandalism |
+------------------------------+----------------------+
```

```
SELECT
  CHARINDEX ('_', url) AS char_location,
  url
FROM courses;
```

```
+---------------+-----------------------------------+
| char_location | url                               |
|---------------+-----------------------------------|
| 34            | datacamp.com/courses/introduction_  |
| 34            | datacamp.com/courses/intermediate_  |
| 29            | datacamp.com/courses/writing_       |
| 29            | datacamp.com/courses/joining_       |
| 27            | datacamp.com/courses/intro_         |
+---------------+-----------------------------------+
```

```
SELECT
  SUBSTRING(url, 12, 12) AS target_section,
  url
FROM courses;
```

```
+-----------------+-----------------------------------+
| target_section  | url                               |
+-----------------+-----------------------------------+
| www.datacamp.com |https//www.datacamp.com/courses   |
+-----------------+-----------------------------------+
```

REPLACE

```
SELECT
  TOP(5) REPLACE(url, '_', '-') AS replace_with_hyphen
FROM courses;
```

```
+-----------------------------------+
| replace_with_hyphen               |
+-----------------------------------|
| datacamp.com/courses/introduction-  |
| datacamp.com/courses/intermediate-  |
| datacamp.com/courses/writing-       |
| datacamp.com/courses/joining-       |
| datacamp.com/courses/intro-         |
+-----------------------------------+
```

Let's practice!

INTRODUCTION TO SQL SERVER

DataCamp

Grouping and Having

INTRODUCTION TO SQL SERVER

John MacKintosh
Instructor

DataCamp

A simple SELECT

```
SELECT
  SUM(demand_loss_mw) AS lost_demand
FROM grid;
```

```
+-------------+
| lost_demand |
+-------------+
| 177888      |
+-------------+
```

Grouping error

Can we break this down by adding an additional column?

```
SELECT
  SUM(demand_loss_mw) AS lost_demand,
  description
FROM grid;
```

```
Msg 8120, Level 16, State 1, Line 1
Column 'grid.description' is invalid in the select list because it is not contained in
either an aggregate function or the GROUP BY clause.
```

```
SELECT
  SUM(demand_loss_mw) AS lost_demand,
  description
FROM grid
GROUP BY description;
```

```
+----------- -+-----------------------------------------------------+
| lost_demand | description                                         |
+-------------+-----------------------------------------------------+
| NULL        | Actual Physical Attack                              |
| NULL        | Cold Weather Event                                  |
| NULL        | Cyber Event with Potential to Cause Impact          |
| 40          | Distribution Interruption                           |
| 2           | Distribution System Interruption                    |
| NULL        | Earthquake                                          |
| NULL        | Electrical Fault at Generator                       |
| 338         | Electrical System Islanding                         |
| 24514       | Electrical System Separation Islanding              |
| 15          | Electrical System Separation Islanding Severe Weather |
+-------------+-----------------------------------------------------+
```

```
SELECT
  SUM(demand_loss_mw) AS lost_demand,
  description
FROM grid
WHERE
  description LIKE '%storm'
  AND demand_loss_mw IS NOT NULL
GROUP BY description;
```

```
+----------- -+-------------------------------------------+
| lost_demand | description                                 |
|-------------+-------------------------------------------|
| 996         | Ice Storm                                   |
| 420         | Load Shed Severe Weather   Lightning Storm |
| 332         | Major Storm                                 |
| 3           | Severe Weather  Thunderstorm                |
| 413         | Severe Weather  Wind Storm                  |
| 4171        | Severe Weather  Winter Storm                |
| 1352        | Winter Storm                                |
+-------------+-------------------------------------------+
```

HAVING

- Can use aggregate functions in `SELECT`
- Filter data using `WHERE`
- Split data into groups using `GROUP BY`
- What if we want to sum values based on groups?
- ... and then filter on those sums?

```
SELECT
  SUM(demand_loss_mw) AS lost_demand,
  description
FROM grid
WHERE
  description LIKE '%storm'
  AND demand_loss_mw IS NOT NULL
GROUP BY description;
```

```
+----------- -+-------------------------------------------+
| lost_demand | description                                 |
+-------------+-------------------------------------------+
| 996         | Ice Storm                                   |
| 420         | Load Shed Severe Weather   Lightning Storm |
| 332         | Major Storm                                 |
| 3           | Severe Weather  Thunderstorm                |
| 413         | Severe Weather  Wind Storm                  |
| 4171        | Severe Weather  Winter Storm                |
| 1352        | Winter Storm                                |
+-------------+-------------------------------------------+
```

```
SELECT
  SUM(demand_loss_mw) AS lost_demand,
  description
FROM grid
WHERE
  description LIKE '%storm'
  AND demand_loss_mw IS NOT NULL
GROUP BY description
HAVING SUM(demand_loss_mw) > 1000;
```

```
+----------- -+-----------------------------------------+
| lost_demand | description                               |
|-------------+-------------------------------------------|
| 4171        | Severe Weather  Winter Storm              |
| 1352        | Winter Storm                              |
+-------------+-------------------------------------------+
```

Summary

- `GROUP BY` splits the data up into combinations of one or more values
- `WHERE` filters on row values
- `HAVING` appears after the `GROUP BY` clause and filters on groups or aggregates

Let's put our skills to the test!

INTRODUCTION TO SQL SERVER

DataCamp