Welcome

INTERMEDIATE SQL SERVER

SQL

Ginger Grant
Instructor

Course overview

- Chapter 1: Summarizing data
- Chapter 2: Date and math functions
- Chapter 3: Processing data with T-SQL
- Chapter 4: Window functions

Exploring Data with Aggregation

- Reviewing summarized values for each column is a common first step in analyzing data
- If the data exists in a database, fastest way to aggregate is to use SQL

Data Exploration with EconomicIndicators

```sql
SELECT Country, Year, InternetUse, GDP,
       ExportGoodsPercent, CellPhonesper100
FROM EconomicIndicators
```

```
+-----------+-----+------------+------------+------------------+-----------------+
|Country    |Year |InternetUse | GDP        |ExportGoodsPercent|CellPhonesper100 |
+-----------+-----+------------+------------+------------------+-----------------+
|Swaziland  |2011 |20.43165813 |7335004354  |56.30476059       |63.7015615       |
|Sweden     |2011 |90.88204559 |394271163688|49.93022195       |118.5711258      |
|Switzerland|2011 |82.98773087 |395111518596|51.20242546       |130.0623629      |
...
+-----------+-----+------------+------------+------------------+-----------------+
```

Common summary statistics

- `MIN()` for the minimum value of a column
- `MAX()` for the maximum value of a column
- `AVG()` for the mean or average value of a column

Common summary statistics in T-SQL

This T-SQL query returns the aggregated values of column InternetUse

```
SELECT AVG(InternetUse) AS MeanInternetUse,
MIN(InternetUse) AS MINInternet,
MAX(InternetUse) AS MAXInternet
FROM EconomicIndicators
```

```
+-------------------+------------+-------------+
|MeanInternetUse    |MINInternet |  MAXInternet|
|-------------------+------------+-------------|
|   18.9854496196171|           0 |  375.5970064|
+-------------------+------------+-------------+
```

Filtering Summary Data with WHERE

This T-SQL query filters the aggregated values using a WHERE clause

Notice the text value is in

```
SELECT AVG(InternetUse) AS MeanInternetUse,
MIN(InternetUse) AS MINInternet,
MAX(InternetUse) AS MAXInternet
FROM EconomicIndicators
WHERE Country = 'Solomon Islands'
```

```
+-------------------+------------+-------------+
|MeanInternetUse    |MINInternet |  MAXInternet|
|-------------------+------------+-------------|
|            1.79621|          0 |         6.00|
+-------------------+------------+-------------+
```

Subtotaling Aggregations into Groups with GROUP BY

```
SELECT Country, AVG(InternetUse) AS MeanInternetUse,
MIN(InternetUse) AS MINInternet,
MAX(InternetUse) AS MAXInternet
FROM EconomicIndicators
GROUP BY Country
```

```
+--------------------+------------------+------------+-------------+
|  Country           |MeanInternetUse   |MINInternet |  MAXInternet|
|--------------------+------------------+------------+-------------|
|Solomon Islands     |           1.79621|           0|         6.00|
|Hong Kong           |          245.1067|           0|       375.00|
|Liechtenstein       |           63.8821|     36.5152|        85.00|
...
+--------------------+------------------+------------+-------------+
```

HAVING is the WHERE for Aggregations

Cannot use `WHERE` with `GROUP BY` as it will give you an error

```
-- This throws an error
...
GROUP BY
WHERE Max(InternetUse) > 100
```

Instead, use `HAVING`

```
-- This is how you filter with a GROUP BY
...
GROUP BY
HAVING Max(InternetUse) > 100
```

HAVING is the WHERE for Aggregations

```sql
SELECT Country, AVG(InternetUse) AS MeanInternetUse,
MIN(GDP) AS SmallestGDP,
MAX(InternetUse) AS MAXInternetUse
FROM EconomicIndicators
GROUP BY Country
HAVING MAX(InternetUse) > 100
```

```
+--------------+-------------------+-------------+---------------+
|Country       |MeanInternetUse    |SmallestGDP  | MAXInternetUse|
|--------------+-------------------+-------------+---------------|
|Macedonia     |   71.3060150792857| -0.465059948|    110.5679538|
|Hong Kong     |   245.106718614286|            0|    375.5970064|
|Congo         |   60.8972476010714| -9.492757847|    104.6455529|
...
+--------------+-------------------+-------------+---------------+
```

Examining UFO Data in the Incidents Table

- The exercise will explore data gathered from Mutual UFO Network
- UFO spotted all over the world are contained in the Incidents Table

Let's practice!

INTERMEDIATE SQL SERVER

Finding and Resolving Missing Data

INTERMEDIATE SQL SERVER

SQL

Ginger Grant
Instructor

datacamp

Detecting missing values

- When you have no data, the empty database field contains the word `NULL`
- Because `NULL` is not a number, it is not possible to use `=`, `<`, or `>` to find or compare missing values
- To determine if a column contains a `NULL` value, use `IS NULL` and `IS NOT NULL`

Returning No NULL Values in T-SQL

```sql
SELECT Country, InternetUse, Year
FROM EconomicIndicators
WHERE InternetUse IS NOT NULL
```

```
+-------------------+-------------------+------------+
|Country            |InternetUse        |Year        |
|-------------------+-------------------+------------+
|Afghanistan        |4.58066992         |2011        |
|Albania            |49                 |2011        |
|Algeria            |14                 |2011        |
....
+-------------------+-------------------+------------+
```

Detecting NULLs in T-SQL

```sql
SELECT  Country, InternetUse, Year
FROM EconomicIndicators
WHERE InternetUse IS NULL
```

```
+-------------------+-------------------+------------+
|Country            |InternetUse        |Year        |
|-------------------+-------------------+------------+
|Angola             |NULL               |2013        |
|Argentina          |NULL               |2013        |
|Armenia            |NULL               |2013        |
....
+-------------------+-------------------+------------+
```

Blank is not NULL

- A blank is not the same as a NULL value
- May show up in columns containing text
- An empty string `''` can be used to find blank values
- The best way is to look for a column where the Length or LEN > 0

Blank is not NULL

```sql
SELECT Country, GDP, Year
FROM EconomicIndicators
WHERE LEN(GDP) > 0
```

```
+--------------------+--------------------+-------------+
|Country             |GDP                 |Year         |
|--------------------+--------------------+-------------+
|Afghanistan         |54852215624         |2011         |
|Albania             |29334492905         |2011         |
|Algeria             |453558093404        |2011         |
....
+--------------------+--------------------+-------------+
```

Substituting missing data with a specific value using ISNULL

```sql
SELECT GDP, Country,
ISNULL(Country, 'Unknown') AS NewCountry
FROM EconomicIndicators
```

```
+--------------------+----------------+-----------------+
|GDP                 |Country         |NewCountry       |
|--------------------+----------------+-----------------+
|5867920022          |NULL            |Unknown          |
|597873038497        |South Africa    |South Africa     |
|1474091271101       |NULL            |Unknown          |
...
+--------------------+----------------+-----------------+
```

Substituting missing data with a column using ISNULL

```sql
/*Substituting values from one column for another with ISNULL*/
SELECT TradeGDPPercent, ImportGoodPercent,
ISNULL(TradeGDPPercent, ImportGoodPercent) AS NewPercent
FROM EconomicIndicators
```

```
+-------------------+-------------------+-----------------+
|TradeGDPPercent    |ImportGoodPercent  |NewPercent       |
|-------------------+-------------------+-----------------+
|NULL               |56.7               |56.7             |
|52.18720739        |51.75273421        |52.18720739      |
|NULL               |NULL               |NULL             |
...
+-------------------+-------------------+-----------------+
```

Substituting NULL values using COALESCE

`COALESCE` returns the first non-missing value

```
COALESCE( value_1, value_2, value_3, ... value_n )
```

- If `value_1` is `NULL` and `value_2` is not `NULL`, return `value_2`
- If `value_1` and `value_2` are `NULL` and `value_3` is not `NULL`, return `value_3`
- ...

SQL Statement using COALESCE

```sql
SELECT TradeGDPPercent, ImportGoodPercent,
COALESCE(TradeGDPPercent, ImportGoodPercent, 'N/A') AS NewPercent
FROM EconomicIndicators
```

```
+-------------------+---------------------+----------------+
|TradeGDPPercent    |ImportGoodPercent    |NewPercent      |
|-------------------+---------------------+----------------+
|NULL               |56.7                 |56.7            |
|NULL               |NULL                 |N/A             |
|52.18720739        |51.75273421          |52.18720739     |
+-------------------+---------------------+----------------+
```

Let's practice!

INTERMEDIATE SQL SERVER

Binning Data with Case

INTERMEDIATE SQL SERVER

SQL

Ginger Grant
Instructor

datacamp

Changing column values with CASE

```
CASE
     WHEN Boolean_expression THEN result_expression [ ...n ]
     [ ELSE else_result_expression ]
END
```

Changing column values with CASE in T-SQL

```sql
SELECT Continent,
CASE WHEN Continent = 'Europe' or Continent = 'Asia' THEN 'Eurasia'
     ELSE 'Other'
     END AS NewContinent
FROM EconomicIndicators
```

```
+-----------+--------------+
|Continent  |NewContinent  |
+-----------+--------------+
|Europe     |Eurasia       |
|Asia       |Eurasia       |
|Americas   |Other         |
...
+-----------+--------------+
```

Changing column values with CASE in T-SQL

```
SELECT Continent,
CASE WHEN Continent = 'Europe' or Continent = 'Asia' THEN 'Eurasia'
     ELSE Continent
     END AS NewContinent
FROM EconomicIndicators
```

```
+-----------+--------------+
|Continent  |NewContinent  |
+-----------+--------------+
|Europe     |Eurasia       |
|Asia       |Eurasia       |
|Americas   |Americas      |
...
+-----------+--------------+
```

Using CASE statements to create value groups

```sql
-- We are binning the data here into discrete groups
SELECT Country, LifeExp,
CASE WHEN LifeExp < 30 THEN 1
     WHEN LifeExp > 29 AND LifeExp < 40 THEN 2
     WHEN LifeExp > 39 AND LifeExp < 50 THEN 3
     WHEN LifeExp > 49 AND LifeExp < 60 THEN 4
     ELSE 5
     END AS LifeExpGroup
FROM EconomicIndicators
WHERE Year = 2007
```

```
+-----------+--------------+
|LifeExp    |LifeExpGroup  |
+-----------+--------------+
|25         |1             |
|30         |2             |
|65         |5             |
...
+-----------+--------------+
```

Let's practice!

INTERMEDIATE SQL SERVER