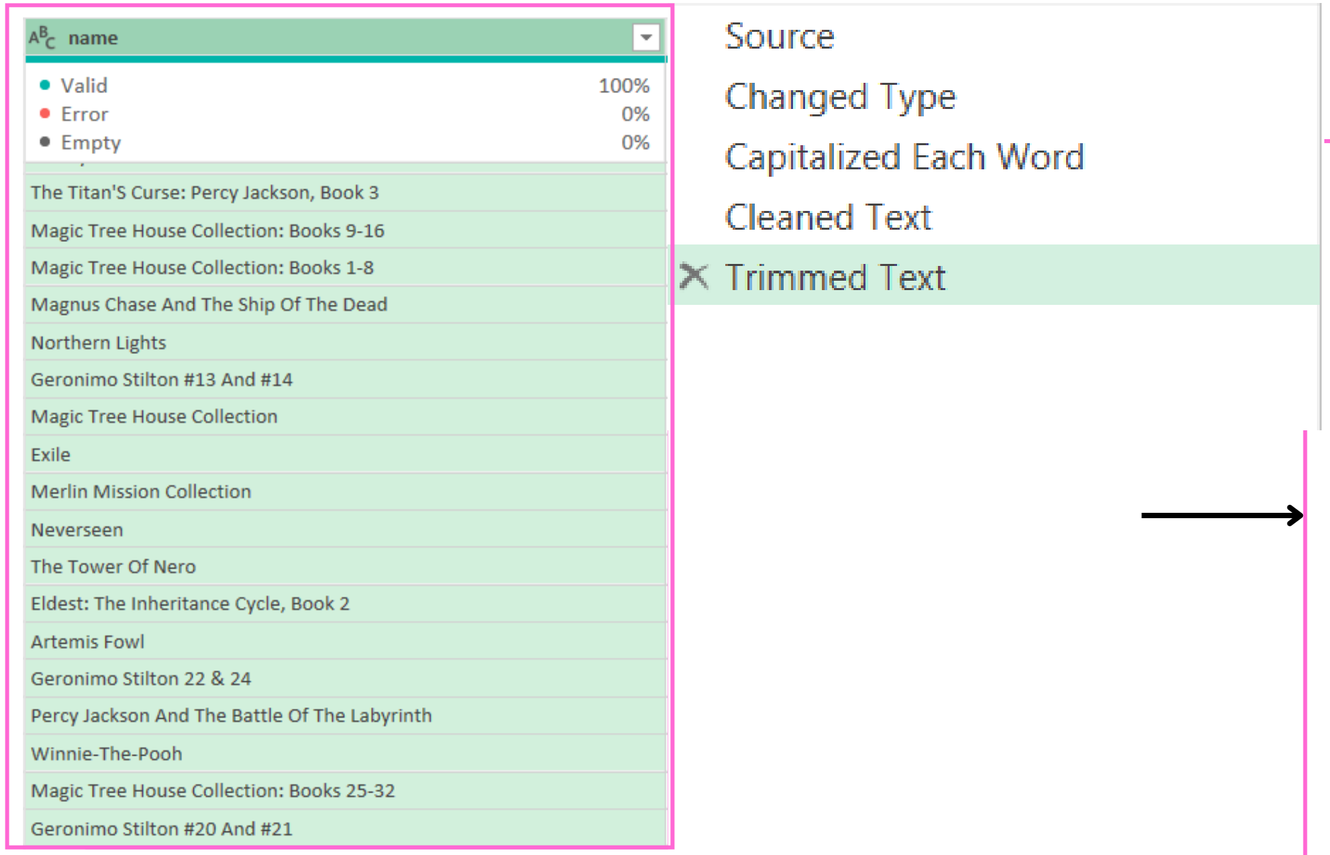
audible
an amazon company

Audible Case Study

Presented by: Nikhil Take.

1. Standardize the name column to ensure consistent title casing.



name
Valid 100%
Error 0%
Empty 0%
The Titan'S Curse: Percy Jackson, Book 3
Magic Tree House Collection: Books 9-16
Magic Tree House Collection: Books 1-8
Magnus Chase And The Ship Of The Dead
Northern Lights
Geronimo Stilton #13 And #14
Magic Tree House Collection
Exile
Merlin Mission Collection
Neverseen
The Tower Of Nero
Eldest: The Inheritance Cycle, Book 2
Artemis Fowl
Geronimo Stilton 22 & 24
Percy Jackson And The Battle Of The Labyrinth
Winnie-The-Pooh
Magic Tree House Collection: Books 25-32
Geronimo Stilton #20 And #21

✅ Standardizing Name Column in Power Query

1️⃣ Open Power Query Editor → Select the name column.

2️⃣ Go to the Transform tab → Click on Format → Choose Capitalize Each Word.

3️⃣ Power Query automatically converts all names to Title Case (e.g., "john doe" → "John Doe").

4️⃣ Click Close & Load → The cleaned data updates in Excel.

🔍 Result: All names are now consistently formatted in Title Case! ✅

✨ This improves data consistency and readability across the dataset.

audible
an amazon company

2.Separate combined first and last names in the author column if they are currently combined.

First Name		Last Name	
Valid	100%	Valid	91%
Error	0%	Error	0%
Empty	0%	Empty	9%
Geronimo		Stilton	
Rick		Riordan	
Jeff		Kinney	
Rick		Riordan	
Rick		Riordan	
Suzanne		Collins	
Winter		Morgan	
Rick		Riordan	
Mary		Pope	
Rick		Riordan	
Rick		Riordan	
Mary		Pope	
Mary		Pope	
Rick		Riordan	
Philip		Pullman	
Geronimo		Stilton	
Mary		Pope	
Shannon		Messenger	

✅ Separating First and Last Names in the Author Column

1️⃣ Removed "Writtenby:" from the author column to keep only names.

2️⃣ Selected the cleaned author column and used Split Column → By Delimiter in Power Query.

3️⃣ Chose Lowercase to Uppercase transition as the delimiter to split names correctly.

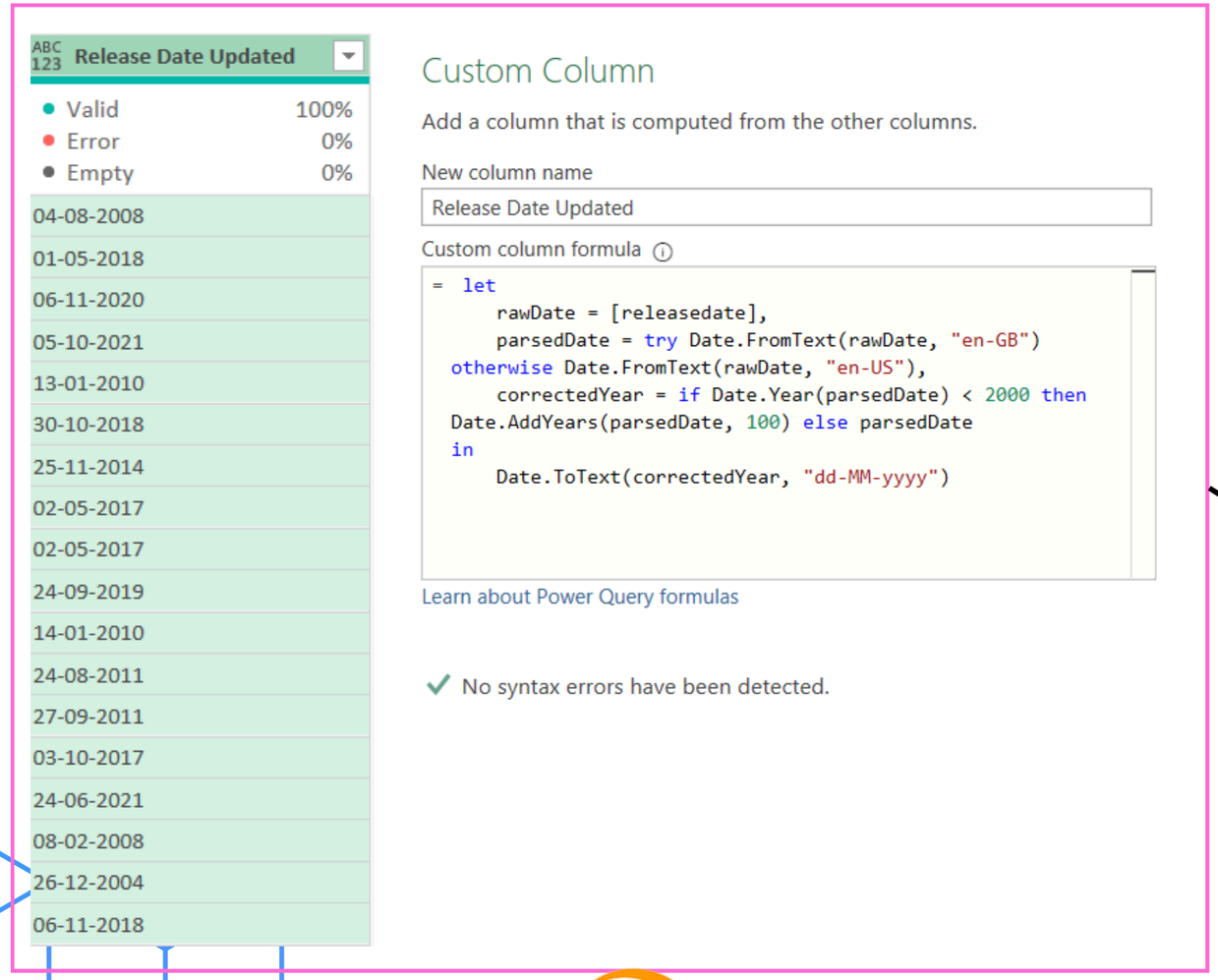
4️⃣ Clicked OK → The column split into two:

- The first part contained the first name.
- The second part contained the last name.
- 5️⃣ Ensured proper formatting and verified the separation of names.

🔍 Result: First and last names are now correctly separated! ✅

✨ The dataset is now structured and ready for analysis.

3.Ensure all entries in the releasedate column follow a consistent date format (DD-MM-YYYY).



✅ **Ensuring Consistent Date Format in the releasedate Column**

1️⃣ **Used a Power Query formula to standardize all dates in the releasedate column.**

2️⃣ **Applied the following formula to handle different formats and correct century errors:**

Formula:-

```
= let
    rawDate = [releasedate],
    parsedDate = try Date.FromText(rawDate, "en-GB")
otherwise Date.FromText(rawDate, "en-US"),
    correctedYear = if Date.Year(parsedDate) < 2000 then
Date.AddYears(parsedDate, 100) else parsedDate
in
    Date.ToText(correctedYear, "dd-MM-yyyy")
```

3️⃣ **This formula:**

- **Parses dates in both "en-GB" (DD/MM/YYYY) and "en-US" (MM/DD/YYYY) formats.**
- **Fixes century errors by adding 100 years to dates with years below 2000.**
- **Converts the final output into DD-MM-YYYY format.**

🔍 **Result: All dates are now uniformly formatted as DD-MM-YYYY, ensuring accuracy in analysis!** ✅

audible
an amazon company

4.Convert the time column from text format to a duration format that Excel recognizes.

time	Updated Time Duration
Valid 100%	Valid 100%
Error 0%	Error 0%
Empty 0%	Empty 0%
2 hrs and 20 mins	0.02:20:00
13 hrs and 8 mins	0.13:08:00
2 hrs and 3 mins	0.02:03:00
11 hrs and 16 mins	0.11:16:00
10 hrs	0.10:00:00
10 hrs and 35 mins	0.10:35:00
2 hrs and 23 mins	0.02:23:00
12 hrs and 32 mins	0.12:32:00
10 hrs and 56 mins	0.10:56:00
13 hrs and 22 mins	0.13:22:00
8 hrs and 48 mins	0.08:48:00
5 hrs and 23 mins	0.05:23:00
6 hrs and 1 min	0.06:01:00
12 hrs and 58 mins	0.12:58:00
11 hrs and 55 mins	0.11:55:00
2 hrs and 25 mins	0.02:25:00
5 hrs and 4 mins	0.05:04:00
14 hrs and 41 mins	0.14:41:00

✅ Converting the time Column to Duration Format

1️⃣ The time column was initially in text format with various formats such as:

- "2 hrs and 3 mins"
- "11 hrs and 16 mins"
- "10 hrs"
- "37 mins"
- "Less than 1 minute"

2️⃣ Used Power Query transformations to extract hours and minutes separately.

3️⃣ Applied the following logic to convert text into Excel-recognized duration format:

Formula:-

```
= let
   rawTime = [time],
   hours = try Number.FromText(Text.BeforeDelimiter(rawTime, " hrs")) otherwise
         try Number.FromText(Text.BeforeDelimiter(rawTime, " hr")) otherwise 0,
   minutes = if Text.Contains(rawTime, " and ") then
            try Number.FromText(Text.BeforeDelimiter(Text.AfterDelimiter(rawTime, " and "), " min"))
otherwise 0
           else if Text.Contains(rawTime, " mins") then
             try Number.FromText(Text.BeforeDelimiter(rawTime, " mins")) otherwise 0
           else if Text.Contains(rawTime, " min") then
             try Number.FromText(Text.BeforeDelimiter(rawTime, " min")) otherwise 0
           else 0,
   finalMinutes = if Text.Contains(rawTime, "Less than 1 minute") then 1 else minutes,
   duration = #duration(0, hours, finalMinutes, 0)
in
   duration
```

4️⃣ This formula:

- Extracts hours (if present).
- Extracts minutes (if present).
- Converts the extracted values into an Excel duration format (hh:mm:ss).
- Handles cases where only minutes or only hours are present.

🔍 Result: The time column is now in a proper duration format (e.g., 02:03:00 for "2 hrs and 3 mins") and is ready for calculations! ✅

audible
an amazon company

5.Ensure the price column is in a numeric format, and identify any non-numeric values.

price	Price Numeric Format
Valid - %	Valid - %
Error < 1%	Error < 1%
Empty - %	Empty - %
468	Numeric
820	Numeric
410	Numeric
615	Numeric
820	Numeric
656	Numeric
233	Numeric
820	Numeric
1256	Numeric
820	Numeric
820	Numeric
1206	Numeric
1206	Numeric
820	Numeric
1093	Numeric

✅ Ensuring the price Column is Numeric & Identifying Non-Numeric Values

1️⃣ Checked whether each value in the price column is numeric using the formula:

Formula:-

= if Value.Is([price], type number) then "Numeric" else "Non-Numeric"

2️⃣ This formula:

- Labels numeric values as "Numeric".
- Flags non-numeric values as "Non-Numeric" for easy identification.

3️⃣ Any "Non-Numeric" values were reviewed and corrected to maintain consistency in the dataset.

🔍 Result: The price column now contains only numeric values, and all non-numeric entries have been identified for correction! ✅

6.Convert text ratings in the stars column to numeric values.

stars	Ratings in Numeric
Valid 100%	Valid 100%
Error 0%	Error 0%
Empty 0%	Empty 0%
5 out of 5 stars34 ratings	5
4.5 out of 5 stars41 ratings	4.5
4.5 out of 5 stars38 ratings	4.5
4.5 out of 5 stars12 ratings	4.5
4.5 out of 5 stars181 ratings	4.5
5 out of 5 stars72 ratings	5
5 out of 5 stars11 ratings	5
5 out of 5 stars50 ratings	5
5 out of 5 stars5 ratings	5
5 out of 5 stars58 ratings	5
4.5 out of 5 stars130 ratings	4.5
5 out of 5 stars6 ratings	5
5 out of 5 stars7 ratings	5
5 out of 5 stars41 ratings	5
4 out of 5 stars2 ratings	4

✅ Converting Text Ratings in the stars Column to Numeric Values

1️⃣ Converted the stars column from text format to numeric using the formula:

```
= if [stars] = "Not rated yet" then 0
  else try
Number.FromText(Text.BeforeDelimiter([stars], " out of")) otherwise 0
```

2️⃣ This formula:

- Assigns a value of 0 for "Not rated yet".
- Extracts the numeric rating before "out of".
- Uses try...otherwise to handle errors and ensure smooth conversion.

🔍 Result: The stars column is now cleaned and contains only numeric values, making it ready for analysis! ✅

audible
an amazon company

7.Split the narratedby column into multiple columns if multiple narrators are listed.

narrator.1	narrator.2	narrator.3
Valid 100%	Valid 7%	Valid 3%
Error 0%	Error 0%	Error 0%
Empty 0%	Empty 93%	Empty 97%
PeterDennis	null	null
MaryPopeOsborne	null	null
BillLobley	null	null
JaneCollingwood	JosephMay	null
LukeDaniels	null	null
BillLobley	null	null
PeterDennis	null	null
MichaelGoldstrom	null	null
ShannonMcManus	null	null
MonicaRachelle	JimD.Johnston	KatieOtten
NehaGargava	null	null
BillLobley	null	null
ShannonMcManus	null	null
CaitlinKelly	null	null
LukeDaniels	null	null
CaitlinKelly	null	null
JorjeanaMarie	null	null
JorjeanaMarie	null	null

✅ Splitting the narratedby Column into Multiple Columns

1️⃣ The narratedby column contained values in formats like:

- "Narratedby:BillLobely"
- "Narratedby:PhilipPullman,fullcast,RuthWilson"

2️⃣ Used Power Query to split the column:

- Removed "Narratedby:" to keep only the names.
- Split the names using a comma , as the delimiter.
- Created separate columns for each narrator.

3️⃣ If an entry had only one narrator, it remained in the first column, while multiple narrators were split into separate columns.

🔍 Result: The narratedby column is now properly structured, making it easier to analyze narrator details! ✅

audible
an amazon company

8.Merge the releasedate and language columns into a single new column named releaseinfo with the format "DD-MM-YYYY, Language."

Release Date Updated	releaseinfo
Valid 100%	Valid 100%
Error 0%	Error 0%
Empty 0%	Empty 0%
04-08-2008	4/8/2008, English
01-05-2018	1/5/2018, English
06-11-2020	6/11/2020, English
05-10-2021	5/10/2021, English
13-01-2010	13/01/10, English
30-10-2018	30/10/18, English
25-11-2014	25/11/14, English
02-05-2017	2/5/2017, English
02-05-2017	2/5/2017, English
24-09-2019	24/09/19, English
14-01-2010	14/01/10, English
24-08-2011	24/08/11, English
27-09-2011	27/09/11, English
03-10-2017	3/10/2017, English
24-06-2021	24/06/21, English

✅ Merging releasedate and language Columns into releaseinfo

1️⃣ Created a new column releaseinfo by combining releasedate and language.

2️⃣ Used the formula:

powerquery

CopyEdit

= [releasedate] & ", " & [language]

3️⃣ This ensures the new column follows the format:

- DD-MM-YYYY, Language
- Example: "15-08-2020, English"

🔍 Result: The releaseinfo column now contains both the release date and language in a consistent format, improving dataset clarity! ✅

audible
an amazon company

9.Ensure all currency values in the price column are formatted consistently with two decimal places.

price	Rounded Price
Valid - %	Valid - %
Error < 1%	Error < 1%
Empty - %	Empty - %
468	468.00
820	820.00
410	410.00
615	615.00
820	820.00
656	656.00
233	233.00
820	820.00
1256	1256.00
820	820.00
820	820.00
1206	1206.00
1206	1206.00
820	820.00
1093	1093.00
467	467.00
1206	1206.00
836	836.00
1256	1256.00
1003	1003.00
615	615.00

- ✅ Ensuring Consistent Currency Formatting in the price Column
- 1️⃣ Converted the price column to a numeric format to ensure proper calculations.
- 2️⃣ Applied formatting to display all values with two decimal places using:
- powerquery
- CopyEdit
- = Number.ToText([price], "0.00")
- 3️⃣ This ensures that all prices, whether whole numbers or decimals, appear in the format:
- 99.00 instead of 99
- 49.50 remains 49.50
- 🔍 Result: The price column now has a consistent currency format, making it ready for financial analysis! ✅

Thank You for Viewing My Presentation!