

Assessment 1

Understanding the Data

Assuming your data is structured in a table called "IDUpdates" with columns "Current ID" and "Old ID", we'll create the necessary DAX measures and calculated columns.

Change Count =

VAR CurrentID = [Current ID]

VAR OldID = [Old ID]

RETURN

```
IF (
    ISBLANK ( OldID ),
    0,
    COUNTROWS (
        FILTER (
            ALL ( 'YourTable' ),
            [Current ID] = OldID
        )
    ) + 1
)
```

Use code with caution.

Explanation:

- Checks if the "Old ID" is blank. If so, it returns 0 (initial ID).
- Otherwise, it counts the number of rows where the current ID is the old ID in other rows and adds 1 to account for the current row itself.

Step 2: DAX Calculated Column for Descriptions

Change Description =

```
SWITCH (
    [Change Count],
    0, "Initial ID",
    1, "1st Change",
    2, "2nd Change",
    3, "3rd Change",
```

```
4, "4th Change",  
"Other Change"  
)
```

Use code with caution.

Explanation:

- Uses the SWITCH function to assign descriptions based on the calculated change count.

Step 3: Measure for Total Changes

```
Total Changes =  
CALCULATE (  
    MAX ( [Change Count] ),  
    ALLEXCEPT ( 'YourTable', [Current ID] )  
)
```

Use code with caution.

Explanation:

- Calculates the maximum change count for each unique "Current ID" by removing all filters except for the "Current ID" column.

Final Result

Once these formulas are applied to your table, you should see the desired output:

Current ID	Old ID	Change Description	Change Count	Total Changes
6716634	0	Initial ID	0	4
6716809	6716634	1st Change	1	4
6720667	6716809	2nd Change	2	4
6720776	6720667	3rd Change	3	4
6721291	6720776	4th Change	4	4

Assessment 2

Creating a Power BI Report to Show Active/Inactive Employee Status

1. Data Preparation

Ensure your data source includes the necessary columns:

- Employee ID
- Name
- Status (Active/Inactive)
- Hire Date
- Termination Date (if applicable)

2. Create a Calculated Column for Status

If your data doesn't have a pre-defined "Status" column, you can create one using a calculated column:

```
Status = VAR TerminationDate = RELATED('YourTable'[Termination Date])
```

```
RETURN
```

```
IF(ISBLANK(TerminationDate), "Active", "Inactive")
```

This formula checks if there's a termination date. If not, the employee is considered "Active"; otherwise, they are "Inactive."

3. Create a Date Table

If you don't have a dedicated date table, create one using the DAX function `CALENDARAUTO()`. This will provide a continuous range of dates.

4. Build the Report

- Add visuals: Drag the following fields to your report:

- Employee ID
- Name
- Status
- Hire Date
- Termination Date

5. Filter Data Using the Slicer

- Use the date slicer to filter the data based on the desired time period. The "Status" column will automatically update with the corresponding color based on the filter.

Assessment 3

Step 1: DAX Calculated Column for Date Handling

1. **Create a date dimension table:** Use the CALENDARAUTO() function to create a date dimension table with columns for Year, Month, Quarter, and Day.
2. **Add a date column to the tenant score table:** Create a calculated column in the tenant score table that uses the DATE() function to combine the year, month, and day from the existing columns into a single date value.

Step 2: Handling Missing Data with DAX

1. **Create a calculated column:** Use the EARLIER() function to calculate the last non-blank score for a tenant.
2. **Apply conditional logic:** Use an if() statement to check if the current month's score is blank. If it is, return the last non-blank score; otherwise, return the current month's score.

Step 3: Resolving Duplicates

1. **Create a calculated column:** Use the MAXX() function with a FILTER() expression to find the maximum score for each tenant and date combination.

Example DAX Formulas:

Date Dimension Table

LastNonBlankScore =

VAR LastScore =

CALCULATE(

MAX([Score]),

FILTER(

ALL('Tenant Score'),

'Tenant Score'[Tenant] = EARLIER('Tenant Score'[Tenant])

&& 'Tenant Score'[Date] <= EARLIER('Tenant Score'[Date])

)

)

RETURN

IF(ISBLANK([Score]), LastScore, [Score])

Calculated Column for Maximum Score for Each Tenant and Date

Code snippet

MaxScore =

CALCULATE(

MAXX('Tenant Score', 'Tenant Score'[Score]),

FILTER(

ALL('Tenant Score'),

'Tenant Score'[Tenant] = EARLIER('Tenant Score'[Tenant])

&& 'Tenant Score'[Date] = EARLIER('Tenant Score'[Date]))

)

)

After applying these steps, you should have a monthly snapshot fact table that includes the following columns:

- Tenant
- Month
- Year
- MaxScore

This table will contain only the maximum score for each tenant and month, with missing months filled in using the last available score.

