Basic (1–10) — Calculated columns

1. **Concatenate First + Last**

DAX

КопироватьРедактировать

```
FullName = TRIM(Customers[FirstName] & " " & Customers[LastName])
```

2. **Email → UPPER**

DAX

КопироватьРедактировать

```
Email_Upper = UPPER(Customers[EmailAddress])
```

3. **First 3 chars of FirstName**

DAX

КопироватьРедактировать

```
First3 = LEFT(Customers[FirstName], 3)
```

4. **Count chars in LastName**

DAX

КопироватьРедактировать

```
LastName_Length = LEN(Customers[LastName])
```

5. **FirstName → lower**

DAX

КопироватьРедактировать

```
FirstName_Lower = LOWER(Customers[FirstName])
```

6. **Trim spaces in EnglishEducation**

DAX

КопироватьРедактировать

```
EnglishEducation_Trim = TRIM(Customers[EnglishEducation])
```

7. **Repeat "*" = length of FirstName**

DAX

КопироватьРедактировать

```
FirstName_Mask = REPT("*", LEN(Customers[FirstName]))
```

8. **Last 4 of Phone**

DAX

КопироватьРедактировать

```
Phone_Last4 = RIGHT(Customers[Phone], 4)
```

9. **Format YearlyIncome as currency (2 decimals)**

DAX

КопироватьРедактировать

```
Income_Format = FORMAT(Customers[YearlyIncome], "[$-en-US]$#,##0.00")
```

10. **Check FirstName = LastName (exact)**

DAX

КопироватьРедактировать

```
Is_First_Equals_Last = Customers[FirstName] = Customers[LastName]
```

Intermediate (11–20) — Calculated columns (unless noted)

11. **Find “Manager” in Occupation (case-sensitive)**

DAX

КопироватьРедактировать

```
Has_Manager_CS = IF( ISERROR( FIND("Manager", Customers[EnglishOccupation]) ), FALSE(), TRUE() )
```

12. **Search “graduate” in Education (case-insensitive)**

DAX

КопироватьРедактировать

```
Has_graduate_CI = IF( ISERROR( SEARCH("graduate", Customers[EnglishEducation]) ), FALSE(), TRUE() )
```

13. **Chars 3–7 of FirstName**

DAX

КопироватьРедактировать

```
FirstName_3_to_7 = MID(Customers[FirstName], 3, 5)
```

14. **Replace area code in Phone with “XXX” (first 3 chars)**

DAX

КопироватьРедактировать

```
Phone_Mask_Area = REPLACE(Customers[Phone], 1, 3, "XXX")
```

15. **BirthDate as “DD-MM-YYYY”**

DAX

КопироватьРедактировать

```
BirthDate_Text = FORMAT(Customers[BirthDate], "dd-MM-yyyy")
```

16. **Initial + Last (e.g., J.Smith)**

DAX

КопироватьРедактировать

```
Initial_Last = LEFT(Customers[FirstName],1) & "." & Customers[LastName]
```

17. **Capitalize first letter, rest lower (FirstName)**

DAX

КопироватьРедактировать

```
FirstName_Proper =
UPPER(LEFT(Customers[FirstName],1)) &
LOWER(MID(Customers[FirstName],2, LEN(Customers[FirstName])-1))
```

18. **Substitute dashes with spaces (Phone)**

DAX

КопироватьРедактировать

```
Phone_Spaces = SUBSTITUTE(Customers[Phone], "-", " ")
```

19. **BirthDate → numeric (serial)**

DAX

КопироватьРедактировать

```
BirthDate_Numeric = INT(Customers[BirthDate])
```

20. **YearlyIncome rounded to 1 decimal, no commas (text)**

DAX

КопироватьРедактировать

```
Income_1dp_NoCommas = FORMAT( ROUND(Customers[YearlyIncome], 1), "0.0" )
```

Advanced (21–30)

21. **Customer Code: first 2 of LastName + last 2 of CustomerKey** (calculated column)

DAX

КопироватьРедактировать

```
CustomerCode =
UPPER( LEFT(Customers[LastName],2) ) &
RIGHT( FORMAT(Customers[CustomerKey], "00"), 2 )
```

22. **Validate email ends with ".com" and contains "@"** (calculated column)

DAX

КопироватьРедактировать

```
Email_IsValid =
CONTAINSSTRING(Customers[EmailAddress], "@") &&
RIGHT(Customers[EmailAddress], 4) = ".com"
```

23. **Extract domain name from EmailAddress** (everything after @) (calculated column)

DAX

КопироватьРедактировать

```
Email_Domain =
RIGHT( Customers[EmailAddress],
    LEN(Customers[EmailAddress]) - FIND("@", Customers[EmailAddress]) )
```

24. **Mask phone except last 4 digits** (calculated column)

DAX

КопироватьРедактировать

```
Phone_Masked =
REPT("X", LEN(Customers[Phone]) - 4) & RIGHT(Customers[Phone], 4)
```

25. **Proper casing of LastName (simulate manually)** (calculated column)

DAX

КопироватьРедактировать

```
LastName_Proper =
UPPER(LEFT(Customers[LastName],1)) &
LOWER(MID(Customers[LastName],2, LEN(Customers[LastName])-1))
```

26. **Replace multiple spaces in EnglishOccupation with single space** (calculated column)

DAX

КопироватьРедактировать

```
Occupation_SingleSpace = TRIM(Customers[EnglishOccupation])
```

27. **Custom ID: initials + birth year (e.g., JD_1985)** (calculated column)

DAX

КопироватьРедактировать

```
CustomID =
UPPER( LEFT(Customers[FirstName],1) & LEFT(Customers[LastName],1) )
& "_" & YEAR(Customers[BirthDate])
```

28. **Remove hyphens in Phone and convert to number** (calculated column)

DAX

КопироватьРедактировать

```
Phone_Number = VALUE( SUBSTITUTE(Customers[Phone], "-", "") )
```

29. **Segment by Education + YearlyIncome** (calculated column)

DAX

КопироватьРедактировать

```
Customer_Segment =
SWITCH(
   TRUE(),
   Customers[EnglishEducation] = "Graduate Degree" && Customers[YearlyIncome] > 90000, "Elite",
   Customers[EnglishEducation] = "Bachelors"       && Customers[YearlyIncome] >= 60000 && Customers[YearlyIncome] <= 90000, "Professional",
   Customers[EnglishEducation] = "High School", "Basic",
   "Other"
)
```

30. **Measure:** return

- total customers if **no Gender selection**;
- count for the **selected Gender**;
- "Multiple Values Selected" if >1 gender selected.

DAX

КопироватьРедактировать

```
Customers by Gender =
VAR TotalCust = DISTINCTCOUNT( Customers[CustomerKey] )
RETURN
IF(
    NOT ISFILTERED(Customers[Gender]),
    TotalCust,   -- no gender selected
    IF(
        HASONEVALUE(Customers[Gender]),
        DISTINCTCOUNT( Customers[CustomerKey] ),
        "Multiple Values Selected"
    )
)
```

If you prefer the measure to always return **text**, wrap the first two branches with FORMAT(… , "0").