# Power BI DAX Course Notes

### 00:00 - Intro to DAX

DAX (Data Analysis Expressions) is a formula language for creating custom calculations and aggregations in Power BI, Analysis Services, and Power Pivot. It allows for powerful data modeling and analytics.

---

### 10:00 - Navigation Function

- Navigation functions in DAX, such as `NEXTDAY`, `PREVIOUSDAY`, and `DATESYTD`, are used for navigating across time-based tables and datasets.

---

### 14:00 - Row Context

- \*\*Row context\*\* is created automatically for each row in a table when a calculation operates on a row-by-row basis (e.g., in calculated columns).

- Example:

```DAX

Total Cost = Sales[Price] \* Sales[Quantity]

```

---

### 17:00 - Conditional Logic

- Conditional logic in DAX is implemented using the `IF` function, enabling decision-making.

- Example:

```DAX

Sales Category = IF(Sales[Amount] > 1000, "High", "Low")

```

---

### 20:00 - First Pass Rule in Conditional Logic

- The \*\*First Pass Rule\*\* ensures that DAX evaluates conditional statements in a top-down order until it finds the first true condition.

---

### 24:00 - Building Relationship

- Relationships between tables in Power BI allow for data to interact and are critical for creating a unified model.

- Types:

- One-to-One

- One-to-Many

- Many-to-Many (requires a bridge table in many cases).

---

### 26:26 - Active Filters Don't Work in Row Context Even If There Is an Active Relationship

- Filters in DAX do not propagate automatically within row context, even with an active relationship. Use `RELATED` or `CALCULATE` to bring filters into the calculation.

---

### 30:11 - Many-to-Many Relationships: Use Bridge Table

- Many-to-Many relationships can create ambiguity. A \*\*bridge table\*\* resolves this by introducing a linking table that ensures clarity.

- Example:

```plaintext

Sales Table -> Product Table -> Bridge Table <- Customer Table

```

---

### 33:00 - RELATED or RELATEDTABLE in Row Context Scenarios

- Use `RELATED` to fetch values from related tables when working in row context.

- Example:

```DAX

Sales Amount = Sales[Quantity] \* RELATED(Product[Price])

```

- `RELATEDTABLE` is used for aggregations across related tables.

---

### 36:20 - Use of SWITCH Function

- The `SWITCH` function simplifies multiple `IF` conditions.

- Example:

```DAX

Category = SWITCH(TRUE(),

Sales[Amount] > 1000, "High",

Sales[Amount] > 500, "Medium",

"Low")

```

---

### 39:35 - X Functions

- Iterative functions like `SUMX`, `AVERAGEX`, `MAXX`, etc., perform row-by-row calculations.

- Example:

```DAX

Total Cost = SUMX(Sales, Sales[Quantity] \* Sales[Price])

```

---

### 46:15 - Calculated Measures

- Measures are dynamic aggregations evaluated at query time and are ideal for efficient performance.

- Example:

```DAX

Total Sales = SUM(Sales[Amount])

```

---

### 48:00 - Filter Context

- \*\*Filter context\*\* determines which rows are visible to a calculation, driven by slicers, filters, or rows in a visualization.

---

### 49:25 - Override Filter Context

- Use `CALCULATE` to modify or override the filter context.

- Example:

```DAX

Total Sales (USA) = CALCULATE(SUM(Sales[Amount]), Sales[Country] = "USA")

```

---

### 51:30 - Semi-Additive Measures

- Semi-additive measures aggregate partially, often requiring specific rules for summing data.

- Example:

- `Opening Balance` or `Closing Balance` over time.

---

### 53:14 - Time Intelligence and Importance of Time Table

- Time intelligence functions require a proper date table.

- Example Functions:

- `TOTALYTD`

- `SAMEPERIODLASTYEAR`

---

### 56:27 - Calculated Measures Done in Report View

- Creating measures in the report view allows you to directly visualize changes in real time.

---

### 1:33:40 - Why Calculated Measures vs. Calculated Columns

- \*\*Calculated Measures:\*\* Efficient, evaluated only when used in visuals.

- \*\*Calculated Columns:\*\* Stored in the model, consuming more memory.

---

### 1:40:56 - Modify Filter Context: Use CALCULATE

- `CALCULATE` modifies existing filter contexts to include or exclude specific filters.

- Example:

```DAX

Total Sales (Specific Product) = CALCULATE(SUM(Sales[Amount]), Sales[Product] = "Laptop")

```

---

### 1:45:37 - Remove Filters

- Use `REMOVEFILTERS` to clear specific filters in calculations.

- Example:

```DAX

Total Sales (Ignoring Region) = CALCULATE(SUM(Sales[Amount]), REMOVEFILTERS(Sales[Region]))

```

---

### 1:58:14 - Using OR

- The `OR` operator evaluates multiple conditions.

- Example:

```DAX

Is High Value = IF(Sales[Amount] > 1000 || Sales[Quantity] > 10, "Yes", "No")

```

---

### 2:01:00 - Using IN for OR

- Use `IN` as a shorthand for multiple OR conditions.

- Example:

```DAX

Is Target Product = IF(Sales[Product] IN {"Laptop", "Tablet"}, "Yes", "No")

```

---

### 2:09:19 - Total Year-to-Date Sale

- Calculate cumulative totals up to the current date.

- Example:

```DAX

YTD Sales = TOTALYTD(SUM(Sales[Amount]), 'Date'[Date])

```

---

### 2:10:00 - Comparing Current and Previous Year

- Use `SAMEPERIODLASTYEAR` for year-over-year comparisons.

- Example:

```DAX

Sales YoY = SUM(Sales[Amount]) - CALCULATE(SUM(Sales[Amount]), SAMEPERIODLASTYEAR('Date'[Date]))

```

---

### 2:26:00 - Semi-Additive Measures

- Reiteration of measures like `Opening Balance` and `Closing Balance` for specific time periods.

---

### 2:38:00 - Opening Balance Month

- Calculate opening balances for a month.

- Example:

```DAX

Opening Balance = FIRSTNONBLANK('Date'[Date], SUM(Sales[Amount]))

```

---

### 2:45:00 - Context Transition

- \*\*Context transition\*\* happens when row context is transformed into filter context, typically by `CALCULATE`.

---

This document summarizes the key concepts and examples from the Power BI DAX course.