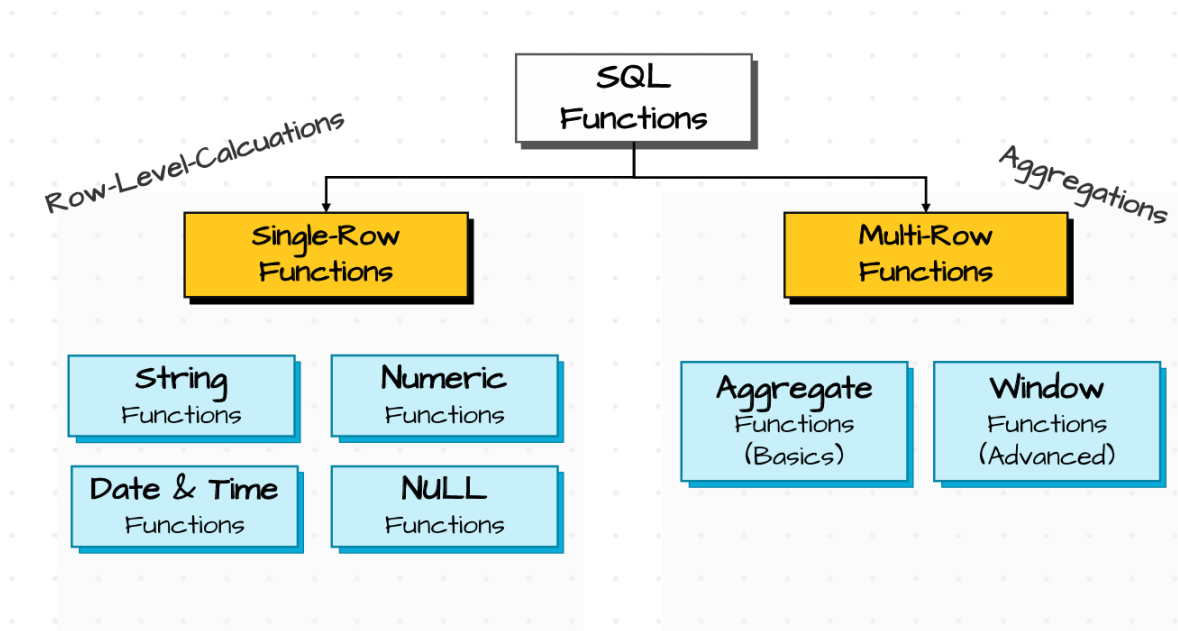
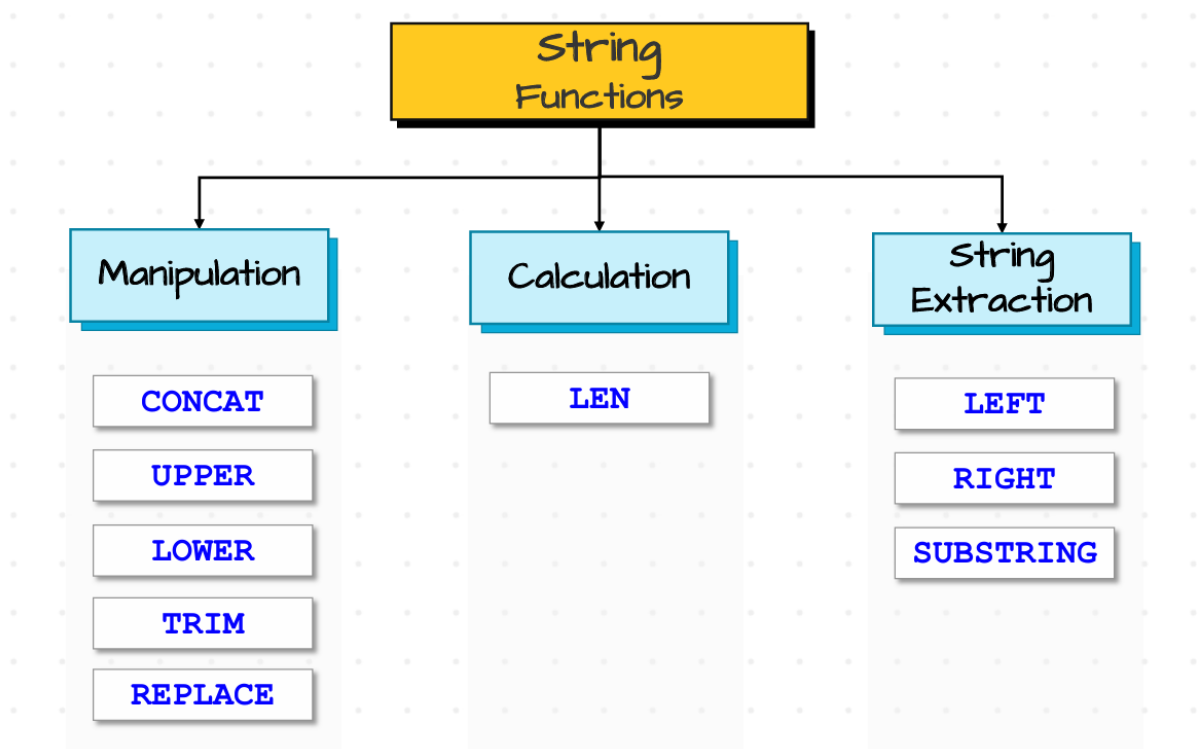


ROW-LEVEL FUNCTIONS



✓ STRING FUNCTIONS



- **CONCAT:**

```
SELECT
CONCAT(FirstName, ' ', LastName) AS Customer_FullName
FROM Sales.Customers
```

	Customer_FullName
1	Jossef Goldberg
2	Kevin Brown
3	Mary
4	Mark Schwarz
5	Anna Adams

- **UPPER AND LOWER:**

```
SELECT
UPPER(FirstName) AS Capitals,
LOWER(LastName) AS NON_CAPITALS
FROM Sales.Customers
```

	Capitals	NON_CAPITALS
1	JOSSEF	goldberg
2	KEVIN	brown
3	MARY	NULL
4	MARK	schwarz
5	ANNA	adams

- **TRIM:**

REMOVE LEADING AND TRAILING SPACES.

QUERY 1: FIND THE CUSTOMERS WHOSE FIRST NAME HAS TRAILING OR LEADING SPACES.

```
SELECT
first_name
FROM customers
WHERE first_name != TRIM(first_name)
```

	first_name
1	John

ANOTHER METHOD:

```
SELECT
    first_name,
    LEN(first_name) len_name,
    LEN(TRIM(first_name)) len_trim_name,
    LEN(first_name) - LEN(TRIM(first_name)) flag
FROM customers
WHERE LEN(first_name) != LEN(TRIM(first_name))
-- WHERE first_name != TRIM(first_name)
```

	first_name	len_name	len_trim_name	flag
1	John	5	4	1

- REPLACE:

Replaces a specific character in to a new character.

```
SELECT  
'123-456-7890' AS Phn_No,  
REPLACE('123-456-7890', '-', '') AS Corrected_Phn_No
```


	Phn_No	Corrected_Phn_No
1	123-456-7890	1234567890

Another example,

```
-- Replace File Extence from txt to csv  
  
SELECT  
'report.txt' AS old_filename,  
REPLACE('report.txt', '.txt', '.csv') AS new_filename
```

I

Results		Messages
	old_filename	new_filename
1	report.txt	report.csv



- **LEN:**

Counts how many characters in a value.

```
SELECT
first_name,
LEN(first_name) AS len_name
FROM customers
```

	first_name	len_name
1	Maria	5
2	John	5
3	Georg	5
4	Martin	6
5	Peter	5

- **LEFT AND RIGHT:**

LEFT extracts specific number of characters from the start.

RIGHT extracts specific number of characters from the end.

-- Retrieve the first two characters of each first name.

```
SELECT
first_name,
LEFT(TRIM(first_name), 2) first_2_char
FROM customers
```

	first_name	first_2_char
1	Maria	Ma
2	John	Jo
3	Georg	Ge
4	Martin	Ma
5	Peter	Pe



- **SUBSTRING:**

Extracts a part of string at a specified position.

```
-- Retrieve a list of customers' first names after removing the first character.
```

```
SELECT
    first_name,
    SUBSTRING(TRIM(first_name), 2, LEN(first_name)) AS sub_name
FROM customers
```

	first_name	sub_name
1	Maria	aria
2	John	ohn
3	Georg	eorg
4	Martin	artin
5	Peter	eter



✓ **NUMBER FUNCTIONS**

- **ROUND**

```
SELECT
    3.516,
    ROUND(3.516, 2) AS round_2,
    ROUND(3.516, 1) AS round_1,
    ROUND(3.516, 0) AS round_0
```



	(No column name)	round_2	round_1	round_0
1	3.516	3.520	3.500	4.000



- ABS

SELECT

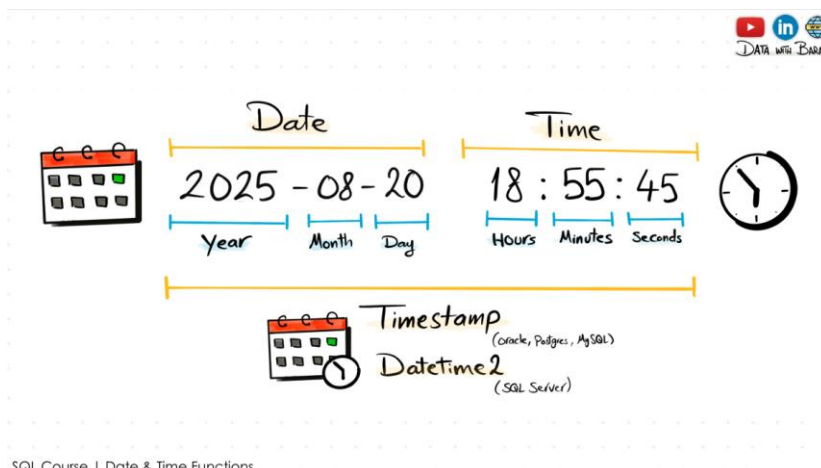
-10,

ABS(-10)



Results	Messages
	(No column name) (No column name)
1	-10 10

✓ DATA AND TIME FUNCTIONS



SQL Course | Date & Time Functions

Object Explorer

- DESKTOP-848QB\SQLSERVER (SQL Server 16.0.1000)
 - Databases
 - AdventureWorks2012
 - AdventureWorks2017
 - SalesDB
 - Tables
 - System Tables
 - External Tables
 - Graph Tables
 - Sales Customers
 - Sales Employees
 - Sales Orders
 - Columns
 - OrderID (int, not null)
 - ProductID (int, null)
 - CustomerID (int, null)
 - SalesPersonID (int, null)
 - OrderDate (datetime2, null)
 - ShipDate (datetime2, null)
 - OrderStatus (varchar(50), null)
 - ShipAddress (varchar(255), null)
 - BillAddress (varchar(255), null)
 - Quantity (int, null)
 - Sales (int, null)
 - CreationTime (datetime2(7), null)
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
 - Sales Orders Archive
 - Sales Products

SQL Query Editor - DESKTOP-848QB\SQLSERVER (SSP)

```

SELECT
OrderID,
OrderDate,
ShipDate,
CreationTime
FROM Sales.Orders

```

Results

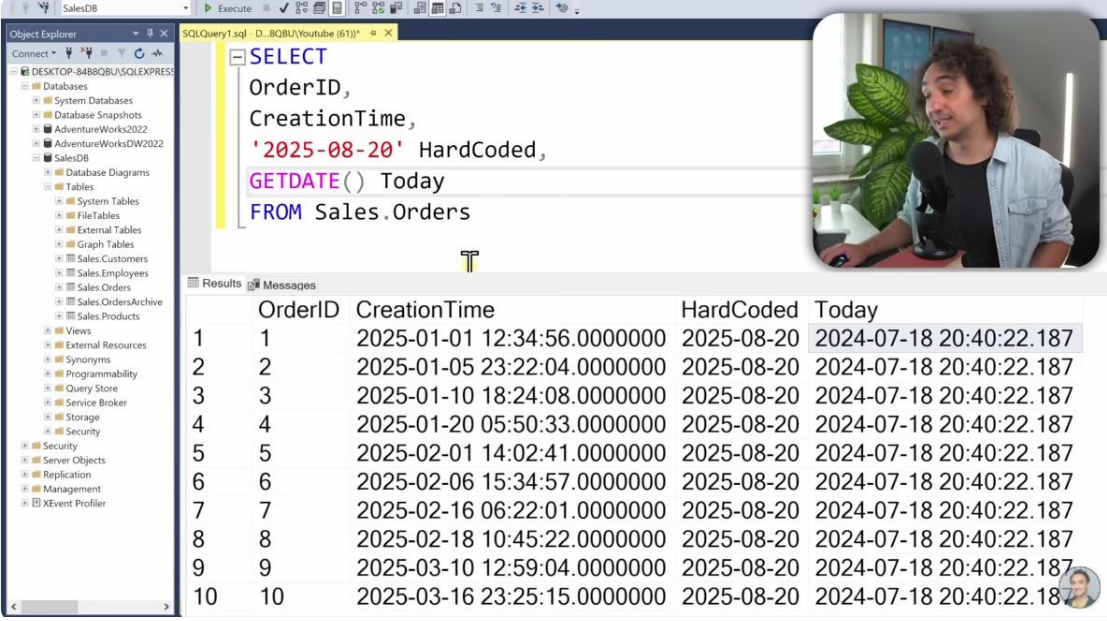
OrderID	OrderDate	ShipDate	CreationTime
1	2025-01-01	2025-01-05	2025-01-01 12:34:56.0000000
2	2025-01-05	2025-01-10	2025-01-05 23:22:11.0000000
3	2025-01-10	2025-01-25	2025-01-10 18:24:08.0000000
4	2025-01-20	2025-01-25	2025-01-20 05:50:33.0000000
5	2025-02-01	2025-02-05	2025-02-01 14:02:41.0000000
6	2025-02-05	2025-02-10	2025-02-06 15:34:57.0000000
7	2025-02-15	2025-02-27	2025-02-16 06:22:01.0000000
8	2025-02-18	2025-02-27	2025-02-18 10:45:22.0000000
9	2025-03-10	2025-03-15	2025-03-10 12:59:04.0000000
10	2025-03-15	2025-03-20	2025-03-16 23:25:15.0000000

Query executed successfully.

DATA AND TIME VALUES:

GETDATE ()

Returns the current date and time at the moment when the query is executed.



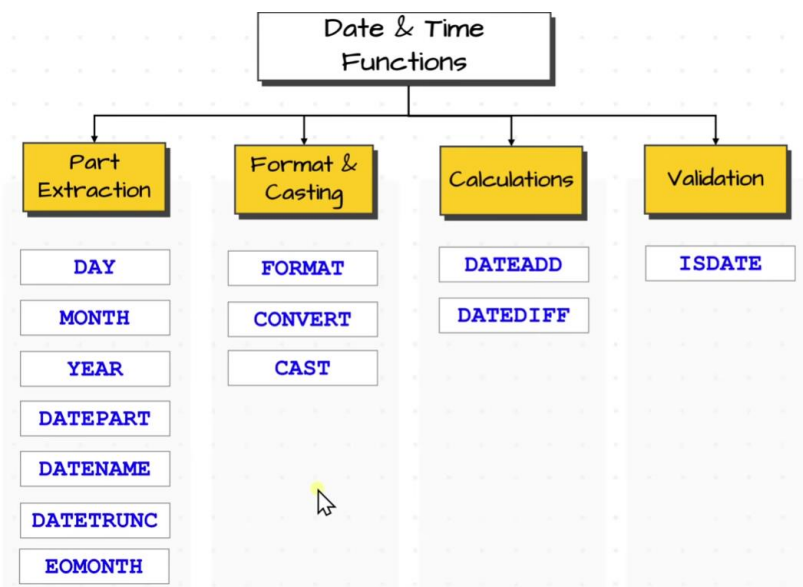
The screenshot shows a SQL Server Enterprise Manager window with a query executed in the 'SalesDB' database. The query is as follows:

```
SELECT  
OrderID,  
CreationTime,  
'2025-08-20' HardCoded,  
GETDATE() Today  
FROM Sales.Orders
```

The results are displayed in a table with the following columns: OrderID, CreationTime, HardCoded, and Today. The table contains 10 rows of data.

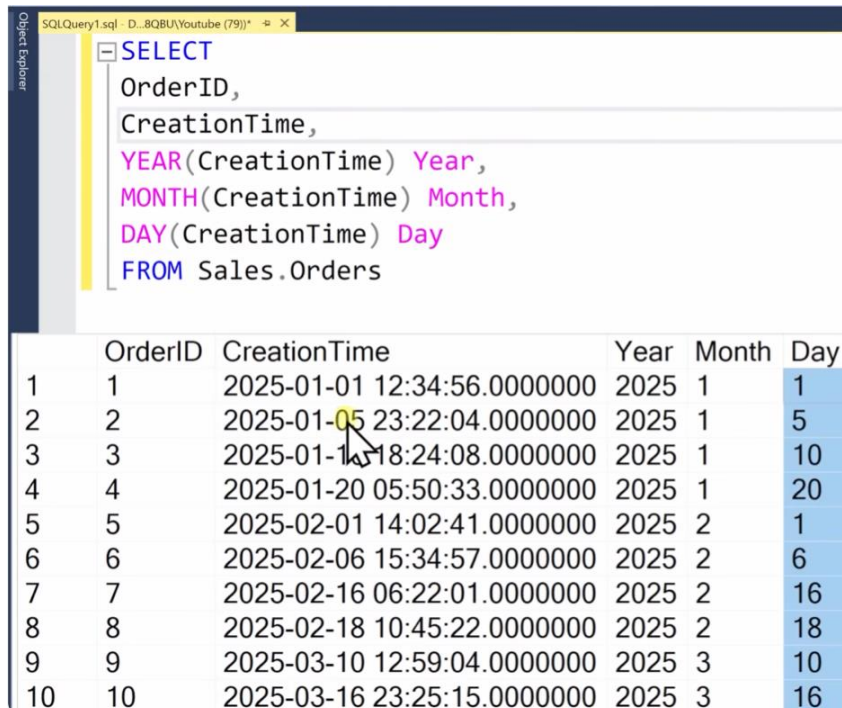
	OrderID	CreationTime	HardCoded	Today
1	1	2025-01-01 12:34:56.0000000	2025-08-20	2024-07-18 20:40:22.187
2	2	2025-01-05 23:22:04.0000000	2025-08-20	2024-07-18 20:40:22.187
3	3	2025-01-10 18:24:08.0000000	2025-08-20	2024-07-18 20:40:22.187
4	4	2025-01-20 05:50:33.0000000	2025-08-20	2024-07-18 20:40:22.187
5	5	2025-02-01 14:02:41.0000000	2025-08-20	2024-07-18 20:40:22.187
6	6	2025-02-06 15:34:57.0000000	2025-08-20	2024-07-18 20:40:22.187
7	7	2025-02-16 06:22:01.0000000	2025-08-20	2024-07-18 20:40:22.187
8	8	2025-02-18 10:45:22.0000000	2025-08-20	2024-07-18 20:40:22.187
9	9	2025-03-10 12:59:04.0000000	2025-08-20	2024-07-18 20:40:22.187
10	10	2025-03-16 23:25:15.0000000	2025-08-20	2024-07-18 20:40:22.187

DATA AND TIME FUNCTIONS:



PART EXTRACTION :

DAY, MONTH, YEAR:

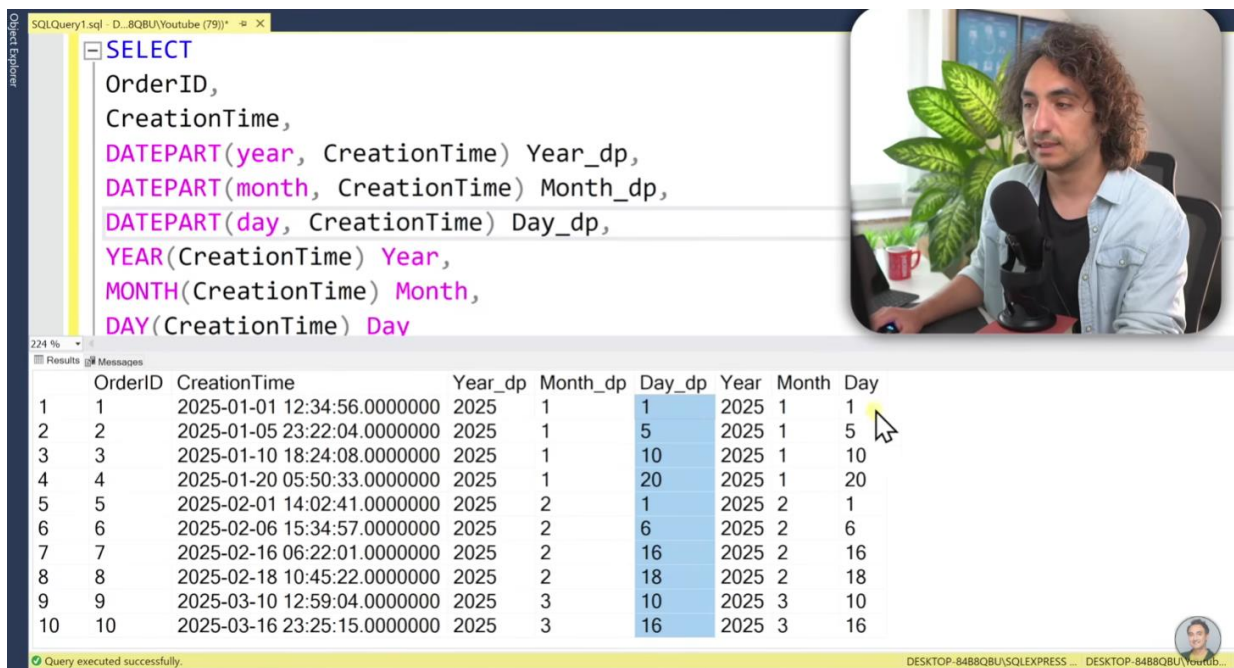


The screenshot shows a SQL query in the 'SQLQuery1.sql' file. The query selects OrderID, CreationTime, and extracts the Year, Month, and Day from CreationTime using YEAR, MONTH, and DAY functions. The results are displayed in a table with 10 rows.

```
SELECT
OrderID,
CreationTime,
YEAR(CreationTime) Year,
MONTH(CreationTime) Month,
DAY(CreationTime) Day
FROM Sales.Orders
```

	OrderID	CreationTime	Year	Month	Day
1	1	2025-01-01 12:34:56.0000000	2025	1	1
2	2	2025-01-05 23:22:04.0000000	2025	1	5
3	3	2025-01-10 18:24:08.0000000	2025	1	10
4	4	2025-01-20 05:50:33.0000000	2025	1	20
5	5	2025-02-01 14:02:41.0000000	2025	2	1
6	6	2025-02-06 15:34:57.0000000	2025	2	6
7	7	2025-02-16 06:22:01.0000000	2025	2	16
8	8	2025-02-18 10:45:22.0000000	2025	2	18
9	9	2025-03-10 12:59:04.0000000	2025	3	10
10	10	2025-03-16 23:25:15.0000000	2025	3	16

DATEPART:



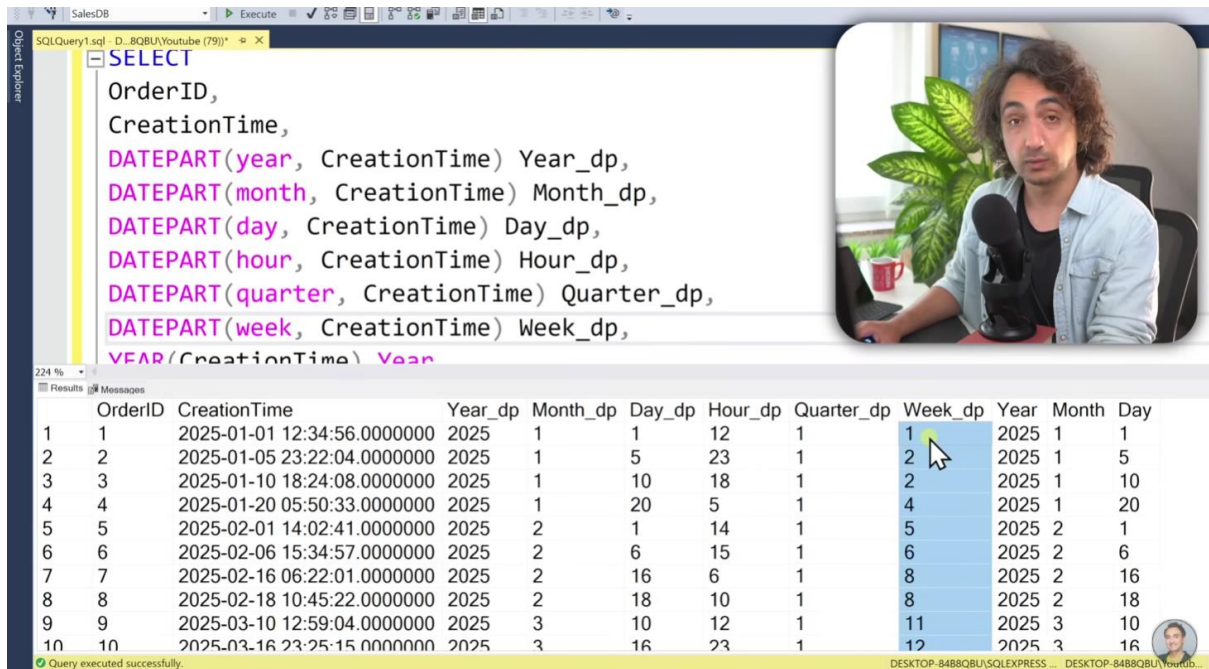
The screenshot shows a SQL query in the 'SQLQuery1.sql' file. The query selects OrderID, CreationTime, and extracts the Year, Month, and Day from CreationTime using DATEPART, YEAR, MONTH, and DAY functions. The results are displayed in a table with 10 rows. An inset video shows a person with curly hair sitting at a desk with a microphone and a potted plant.

```
SELECT
OrderID,
CreationTime,
DATEPART(year, CreationTime) Year_dp,
DATEPART(month, CreationTime) Month_dp,
DATEPART(day, CreationTime) Day_dp,
YEAR(CreationTime) Year,
MONTH(CreationTime) Month,
DAY(CreationTime) Day
```

	OrderID	CreationTime	Year_dp	Month_dp	Day_dp	Year	Month	Day
1	1	2025-01-01 12:34:56.0000000	2025	1	1	2025	1	1
2	2	2025-01-05 23:22:04.0000000	2025	1	5	2025	1	5
3	3	2025-01-10 18:24:08.0000000	2025	1	10	2025	1	10
4	4	2025-01-20 05:50:33.0000000	2025	1	20	2025	1	20
5	5	2025-02-01 14:02:41.0000000	2025	2	1	2025	2	1
6	6	2025-02-06 15:34:57.0000000	2025	2	6	2025	2	6
7	7	2025-02-16 06:22:01.0000000	2025	2	16	2025	2	16
8	8	2025-02-18 10:45:22.0000000	2025	2	18	2025	2	18
9	9	2025-03-10 12:59:04.0000000	2025	3	10	2025	3	10
10	10	2025-03-16 23:25:15.0000000	2025	3	16	2025	3	16

Query executed successfully.

DESKTOP-84B8QBU\SQLEXPRESS ... DESKTOP-84B8QBU\Youtube ...



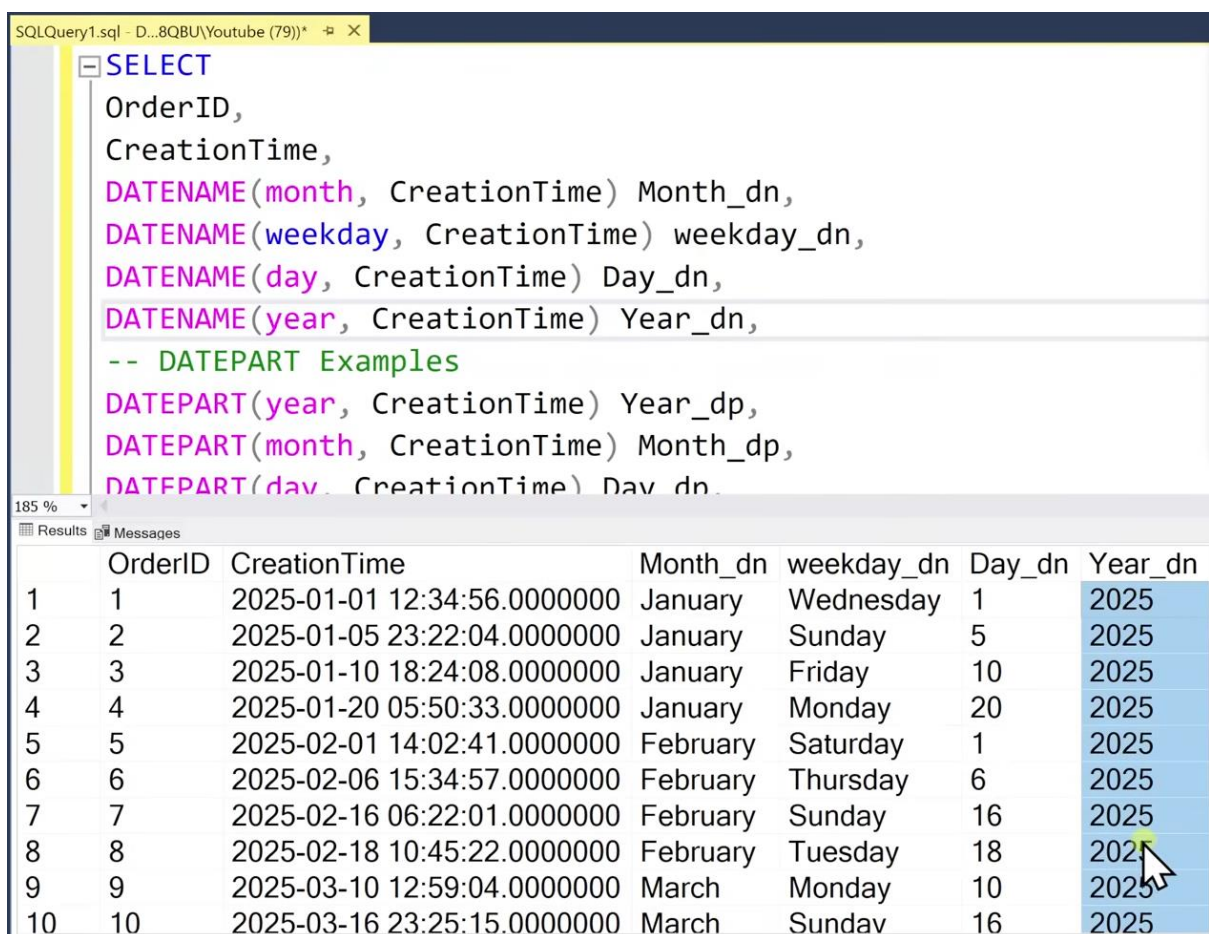
```

SELECT
  OrderID,
  CreationTime,
  DATEPART(year, CreationTime) Year_dp,
  DATEPART(month, CreationTime) Month_dp,
  DATEPART(day, CreationTime) Day_dp,
  DATEPART(hour, CreationTime) Hour_dp,
  DATEPART(quarter, CreationTime) Quarter_dp,
  DATEPART(week, CreationTime) Week_dp,
  YEAR(CreationTime) Year

```

OrderID	CreationTime	Year_dp	Month_dp	Day_dp	Hour_dp	Quarter_dp	Week_dp	Year	Month	Day
1	2025-01-01 12:34:56.0000000	2025	1	1	12	1	1	2025	1	1
2	2025-01-05 23:22:04.0000000	2025	1	5	23	1	2	2025	1	5
3	2025-01-10 18:24:08.0000000	2025	1	10	18	1	2	2025	1	10
4	2025-01-20 05:50:33.0000000	2025	1	20	5	1	4	2025	1	20
5	2025-02-01 14:02:41.0000000	2025	2	1	14	1	5	2025	2	1
6	2025-02-06 15:34:57.0000000	2025	2	6	15	1	6	2025	2	6
7	2025-02-16 06:22:01.0000000	2025	2	16	6	1	8	2025	2	16
8	2025-02-18 10:45:22.0000000	2025	2	18	10	1	8	2025	2	18
9	2025-03-10 12:59:04.0000000	2025	3	10	12	1	11	2025	3	10
10	2025-03-16 23:25:15.0000000	2025	3	16	23	1	12	2025	3	16

DATENAME:



```

SELECT
  OrderID,
  CreationTime,
  DATENAME(month, CreationTime) Month_dn,
  DATENAME(weekday, CreationTime) weekday_dn,
  DATENAME(day, CreationTime) Day_dn,
  DATENAME(year, CreationTime) Year_dn,
  -- DATEPART Examples
  DATEPART(year, CreationTime) Year_dp,
  DATEPART(month, CreationTime) Month_dp,
  DATEPART(day, CreationTime) Day_dp

```

OrderID	CreationTime	Month_dn	weekday_dn	Day_dn	Year_dn
1	2025-01-01 12:34:56.0000000	January	Wednesday	1	2025
2	2025-01-05 23:22:04.0000000	January	Sunday	5	2025
3	2025-01-10 18:24:08.0000000	January	Friday	10	2025
4	2025-01-20 05:50:33.0000000	January	Monday	20	2025
5	2025-02-01 14:02:41.0000000	February	Saturday	1	2025
6	2025-02-06 15:34:57.0000000	February	Thursday	6	2025
7	2025-02-16 06:22:01.0000000	February	Sunday	16	2025
8	2025-02-18 10:45:22.0000000	February	Tuesday	18	2025
9	2025-03-10 12:59:04.0000000	March	Monday	10	2025
10	2025-03-16 23:25:15.0000000	March	Sunday	16	2025

Here the output will be string, eventhough the day_dn, year_dn are looking like integers they are strings.

DATETRUNC:

Syntax

DATEPART(part, date)

DATENAME(part, date)

DATETRUNC(part, date)

DATETRUNC



DATETRUNC

day

Year Month Day Hours Minutes Seconds
2025 - 08 - 20 00 : 00 : 00

DATETRUNC

month

Keep Reset
Year Month Day Hours Minutes Seconds
2025 - 08 - 01 00 : 00 : 00

DATETRUNC

year

Keep Reset
Year Month Day Hours Minutes Seconds
2025 - 01 - 01 00 : 00 : 00

Datepart resets to 01 Time part resets to 00

SQL Course | Date & Time Functions

SQLQuery1.sql - DESKTOP-8488QBU\SQLEXPRESS\SalesDB (DESKTOP-8488QBU\Youtube (79)) - Microsoft SQL Server Management Studio

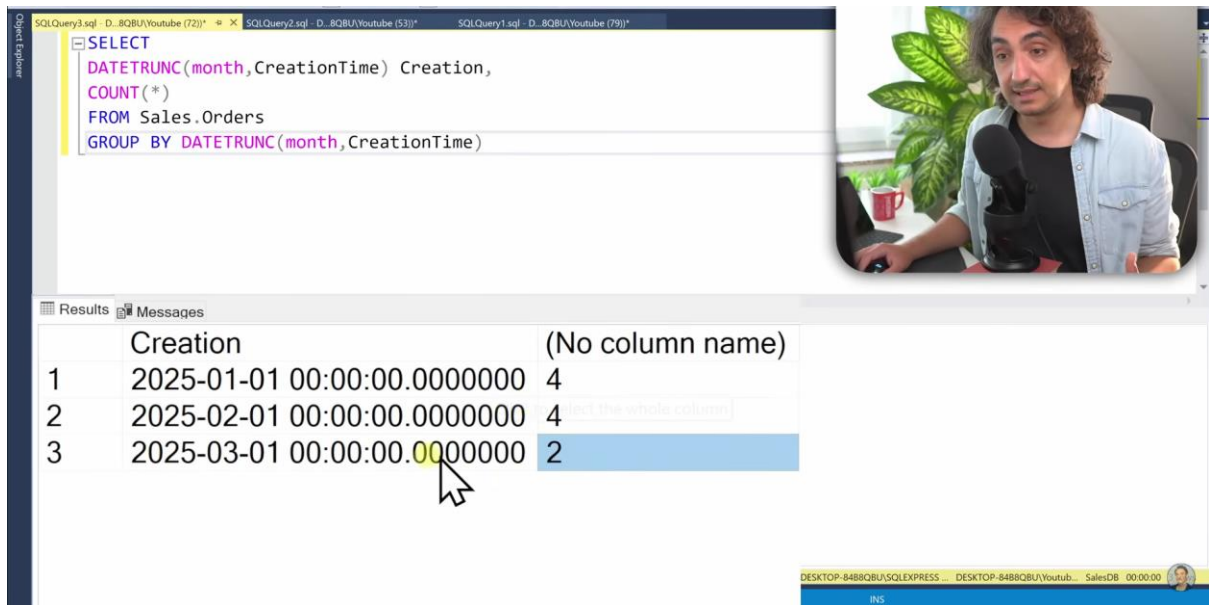
```

SELECT
  OrderID,
  CreationTime,
  -- DATETRUNC Examples
  DATETRUNC(year, CreationTime) Year_dt,
  DATETRUNC(day, CreationTime) Day_dt,
  DATETRUNC(minute, CreationTime) Minute_dt,
  -- DATENAME Examples
  DATENAME(month, CreationTime) Month_dn,
  DATENAME(weekday, CreationTime) Weekday_dn,
  DATENAME(day, CreationTime) Day_dn
  
```

OrderID	CreationTime	Year_dt	Day_dt	Minute_dt
1	2025-01-01 12:34:56.0000000	2025-01-01 00:00:00.0000000	2025-01-01 00:00:00.0000000	2025-01-01 12:34:00.0000000
2	2025-01-05 23:22:04.0000000	2025-01-01 00:00:00.0000000	2025-01-05 00:00:00.0000000	2025-01-05 23:22:00.0000000
3	2025-01-10 18:24:08.0000000	2025-01-01 00:00:00.0000000	2025-01-10 00:00:00.0000000	2025-01-10 18:24:00.0000000
4	2025-01-20 05:50:33.0000000	2025-01-01 00:00:00.0000000	2025-01-20 00:00:00.0000000	2025-01-20 05:50:00.0000000
5	2025-02-01 14:02:41.0000000	2025-01-01 00:00:00.0000000	2025-02-01 00:00:00.0000000	2025-02-01 14:02:00.0000000
6	2025-02-06 15:34:57.0000000	2025-01-01 00:00:00.0000000	2025-02-06 00:00:00.0000000	2025-02-06 15:34:00.0000000
7	2025-02-16 06:22:01.0000000	2025-01-01 00:00:00.0000000	2025-02-16 00:00:00.0000000	2025-02-16 06:22:00.0000000
8	2025-02-18 10:45:22.0000000	2025-01-01 00:00:00.0000000	2025-02-18 00:00:00.0000000	2025-02-18 10:45:00.0000000
9	2025-03-10 12:59:04.0000000	2025-01-01 00:00:00.0000000	2025-03-10 00:00:00.0000000	2025-03-10 12:59:00.0000000
10	2025-03-16 23:25:15.0000000	2025-01-01 00:00:00.0000000	2025-03-16 00:00:00.0000000	2025-03-16 23:25:00.0000000

Query executed successfully.

WHY IS IT USEFULL:



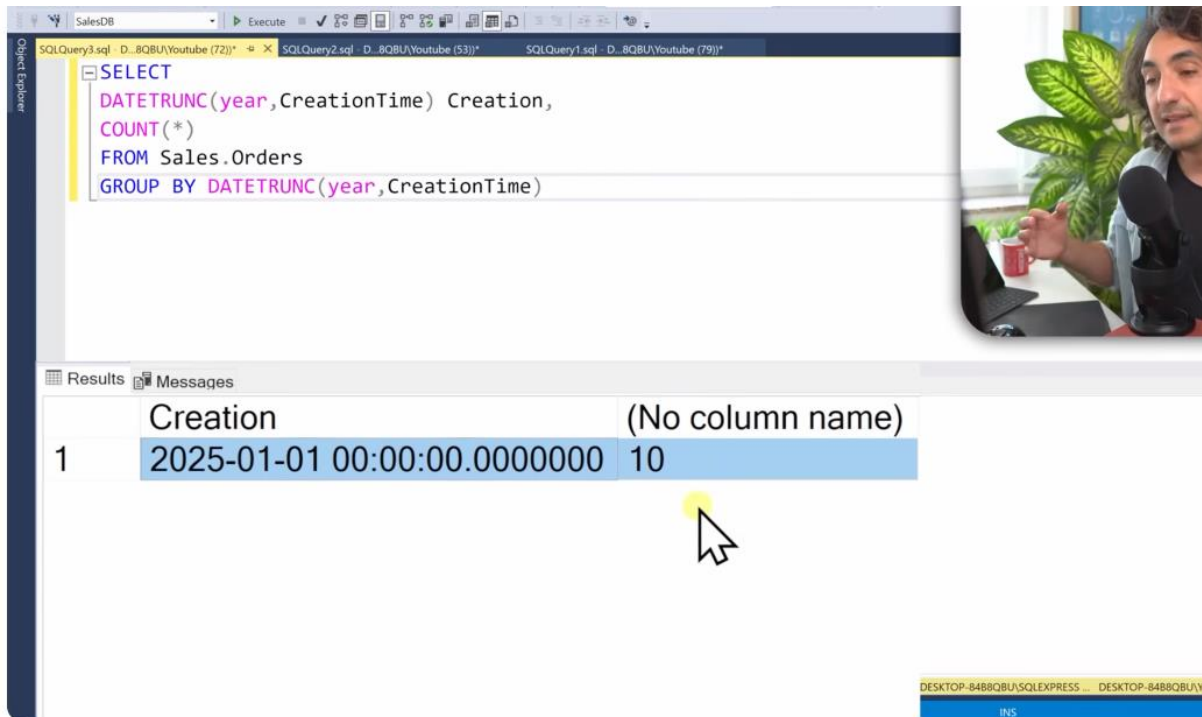
The screenshot shows a SQL query in SQL Server Enterprise Manager. The query is:

```
SELECT  
    DATETRUNC(month,CreationTime) Creation,  
    COUNT(*)  
FROM Sales.Orders  
GROUP BY DATETRUNC(month,CreationTime)
```

The results pane shows the following data:

	Creation	(No column name)
1	2025-01-01 00:00:00.0000000	4
2	2025-02-01 00:00:00.0000000	4
3	2025-03-01 00:00:00.0000000	2

A mouse cursor is pointing at the third row. A tooltip "select the whole column" is visible over the second column header. A video inset shows a man speaking into a microphone.



The screenshot shows a SQL query in SQL Server Enterprise Manager. The query is:

```
SELECT  
    DATETRUNC(year,CreationTime) Creation,  
    COUNT(*)  
FROM Sales.Orders  
GROUP BY DATETRUNC(year,CreationTime)
```

The results pane shows the following data:

	Creation	(No column name)
1	2025-01-01 00:00:00.0000000	10

A mouse cursor is pointing at the second column header. A video inset shows a man speaking into a microphone.

EOMONTH:

SQLQuery3.sql - D:\8QBU\Youtube (72))*

SQLQuery2.sql - D:\8QBU\Youtube (53))*

SQLQuery1.sql - D:\8QBU\Youtube (79))*

Object Explorer

SELECT

OrderID,

CreationTime,

EOMONTH(CreationTime) EndOfMonth

FROM Sales.Orders

Results

Messages

	OrderID	CreationTime	EndOfMonth
1	1	2025-01-01 12:34:56.0000000	2025-01-31
2	2	2025-01-05 23:22:04.0000000	2025-01-31
3	3	2025-01-10 18:24:08.0000000	2025-01-31
4	4	2025-01-20 05:50:33.0000000	2025-01-31
5	5	2025-02-01 14:02:41.0000000	2025-02-28
6	6	2025-02-06 15:34:57.0000000	2025-02-28
7	7	2025-02-16 06:22:01.0000000	2025-02-28
8	8	2025-02-18 10:45:22.0000000	2025-02-28
9	9	2025-03-10 12:59:04.0000000	2025-03-31
10	10	2025-03-16 23:25:15.0000000	2025-03-31

SQLQuery3.sql - D:\8QBU\Youtube (72))*

SQLQuery2.sql - D:\8QBU\Youtube (53))*

SQLQuery1.sql - D:\8QBU\Youtube (79))*

Object Explorer

SELECT

OrderID,

CreationTime,

EOMONTH(CreationTime) EndOfMonth,

CAST(DATETRUNC(month, CreationTime) AS DATE) StartOfMonth

FROM Sales.Orders

Results

Messages

	OrderID	CreationTime	EndOfMonth	StartOfMonth
1	1	2025-01-01 12:34:56.0000000	2025-01-31	2025-01-01
2	2	2025-01-05 23:22:04.0000000	2025-01-31	2025-01-01
3	3	2025-01-10 18:24:08.0000000	2025-01-31	2025-01-01
4	4	2025-01-20 05:50:33.0000000	2025-01-31	2025-01-01
5	5	2025-02-01 14:02:41.0000000	2025-02-28	2025-02-01
6	6	2025-02-06 15:34:57.0000000	2025-02-28	2025-02-01
7	7	2025-02-16 06:22:01.0000000	2025-02-28	2025-02-01
8	8	2025-02-18 10:45:22.0000000	2025-02-28	2025-02-01
9	9	2025-03-10 12:59:04.0000000	2025-03-31	2025-03-01
10	10	2025-03-16 23:25:15.0000000	2025-03-31	2025-03-01

Part Extraction USE CASE:

QUERY 1: How many orders were placed every year.

```
SELECT  
YEAR(OrderDate) AS Year,  
COUNT(*) AS No_of_Orders  
FROM Sales.Orders  
GROUP BY YEAR(OrderDate)
```

	Year	No_of_Orders
1	2025	10

QUERY 2: How many orders placed each month.

```
SELECT  
MONTH(OrderDate) AS Month,  
COUNT(*) AS No_of_Orders  
FROM Sales.Orders  
GROUP BY MONTH(OrderDate)
```

	Month	No_of_Orders
1	1	4
2	2	4
3	3	2

If we want the month name instead of numbers we use DATENAME

```
SELECT  
DATENAME(MONTH,OrderDate) AS Month,  
COUNT(*) AS No_of_Orders  
FROM Sales.Orders  
GROUP BY DATENAME(MONTH,OrderDate)
```

	Month	No_of_Orders
1	February	4
2	January	4
3	March	2

Used for Data Filtering:

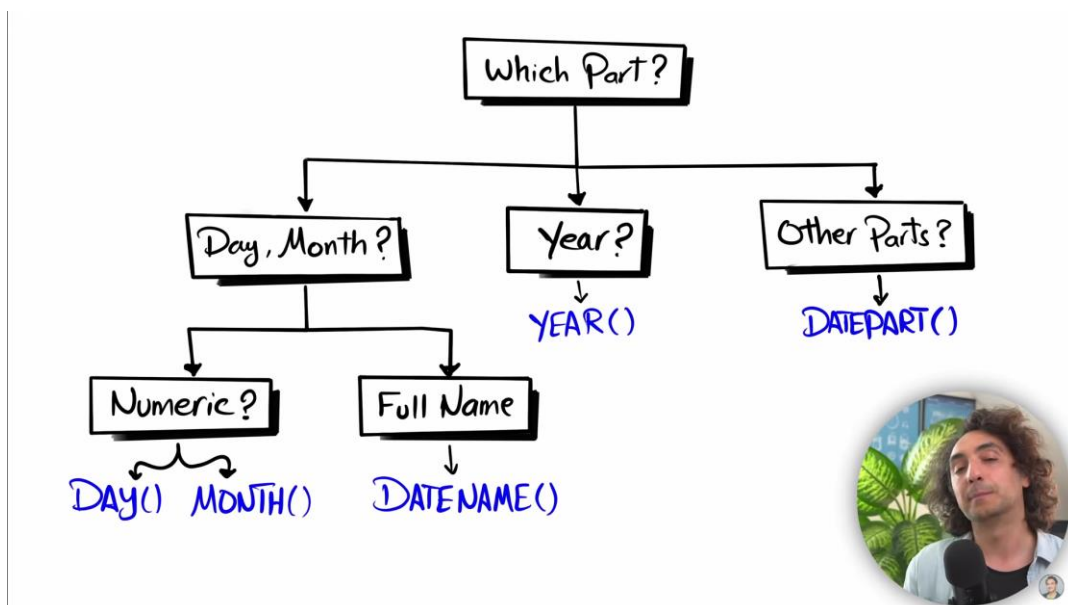
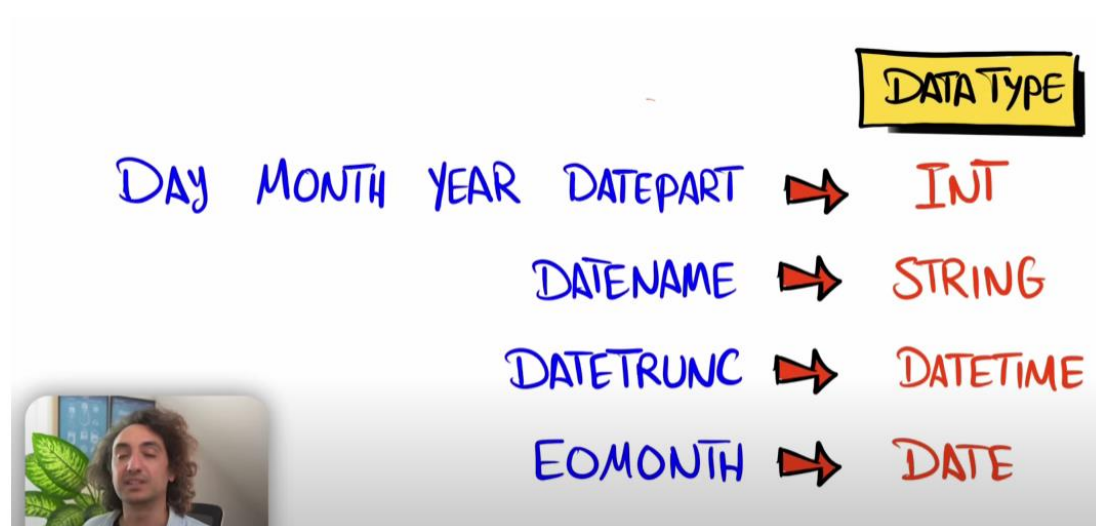
QUERY 1: Show all the orders placed during the month of February.

```
SELECT *  
FROM Sales.Orders  
WHERE MONTH(OrderDate) = 2
```

	OrderID	ProductID	CustomerID	SalesPersonID	OrderDate	ShipDate	OrderStatus	ShipAddress	BillAddress	Quantity	Sales	CreationTime
1	5	104	2	5	2025-02-01	2025-02-05	Delivered	NULL	NULL	1	25	2025-02-01 14:02:41.0000000
2	6	104	3	5	2025-02-05	2025-02-10	Delivered	1792 Belmont Rd.	NULL	2	50	2025-02-06 15:34:57.0000000
3	7	102	1	1	2025-02-15	2025-02-27	Delivered	136 Balboa Court		2	30	2025-02-16 06:22:01.0000000
4	8	101	4	3	2025-02-18	2025-02-27	Shipped	2947 Vine Lane	4311 Clay Rd	3	90	2025-02-18 10:45:22.0000000

BEST PRACTICE : FILTERING DATA USING AN INTEGER IS FASTER THAN USING A STRING.

AVOID USING DATENAME, INSTEAD USE DATEPART.



2025-08-20
09:38:54.840

Date Parts

DATEPART

DATENAME

DATETRUNC

Part	Abbre.	INT	String	Datetime2
		DATEPART	DATENAME	DATETRUNC
year	yy, yyyy	2025	2025	2025-01-01 00:00:00
quarter	qq, q	3	3	2025-07-01 00:00:00
month	mm, m	8	August	2025-08-01 00:00:00
dayofyear	dy, y	232	232	2025-08-20 00:00:00
day	dd, d	20	20	2025-08-20 00:00:00
weekday	dw	4	Wednesday	Not supported
week	wk, ww	34	34	2025-08-17 00:00:00
iso_week	ns	34	34	2025-08-18 00:00:00
hour	hh	9	9	2025-08-20 09:00:00
minute	mi, n	45	45	2025-08-20 09:45:00
second	ss, s	21	21	2025-08-20 09:45:21
millisecond	ms	0	0	2025-08-20 09:45:21
microsecond	msc	0	0	2025-08-20 09:45:21
nanosecond	ns	0	0	Not supported
iso_week	isowk, isoww	0	+00:00	Not supported

