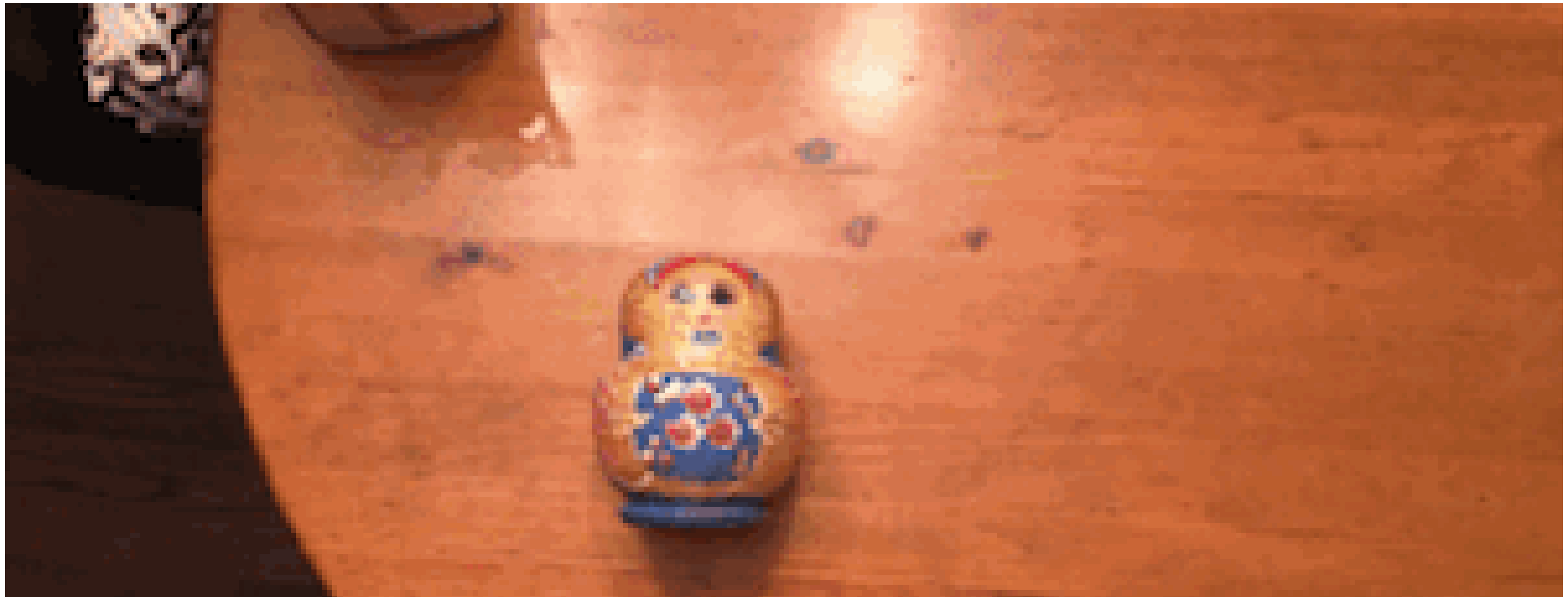
Recap of Common Table Expressions (CTE)

HIERARCHICAL AND RECURSIVE QUERIES IN SQL SERVER

Dominik Egarter
Data Engineering Enthusiast

Outline of the course

- What is recursion and how to us it?
- How could you use Common Expression Tables (CTE) for hierarchical and recursive queries?
- How could you represent hierarchy with SQL and how to query it?
- Real-World examples (e.g. company organizations, bill of materials, flight plan)



What is a CTE?

- Definition of a CTE

> Specifies a temporary named result set, known as a common table expression (CTE). This is derived from a simple query and defined within the execution scope of a single statement

- Structure of a CTE

```
WITH CTEtable as
(
    -- a query --
)
SELECT *
    FROM CTEtable
```

Use of CTEs

- Manage complicated queries
- Can be used within `SELECT` , `INSERT` , `UPDATE` , or `DELETE` statements
- More than one CTE can be defined in one `WITH` statement
- Combine several CTEs with `UNION` or `JOIN`
- Substitute for a view
- Self-reference a table
- Recursion query

Define a CTE for an IT-organization

Select managers using a CTE:

```
WITH JOBS (id, SurName, JobTitle) AS
(
    SELECT
          ID,
          Name,
          Position
    FROM employee
    WHERE Position like '%Manager%'
),
```

- `WITH` AS to define the CTE
- `SELECT` on the `Employee` table

Define a CTE for an IT-organization

Select employees with salary over 10 000

```
SALARIES (ID,Salary) AS (
   SELECT
        ID,
        Salary
   FROM ITSalary
   WHERE Salary > 10000)
```

Define a CTE for an IT-organization

The whole query:

```
WITH JOBS (id, SurName, JobTitle) AS
(    SELECT ID, Name, Position
        FROM employee
        WHERE Position like '%Manager%'),
SALARIES (ID,Salary) AS
(    SELECT ID, Salary
        FROM ITSalary
        WHERE Salary > 10000)
SELECT JOBS.Name, JOBS.Position, SALARIES.Salary
    FROM JOBS
    INNER JOIN SALARIES
    on JOBS.ID = SALARIES.ID;
```

```
SurNmames | JobTitle | Salary
Paul Smith | IT Manager | 15,000
Adam Peterson | Sourcing Manager | 12,500
Anna Nilson | Portfolio Manager | 10,500
```

Let's practice!

HIERARCHICAL AND RECURSIVE QUERIES IN SQL SERVER

Introduction to recursion

HIERARCHICAL AND RECURSIVE QUERIES IN SQL SERVER

SQL

Dominik Egarter
Data Engineering Enthusiast

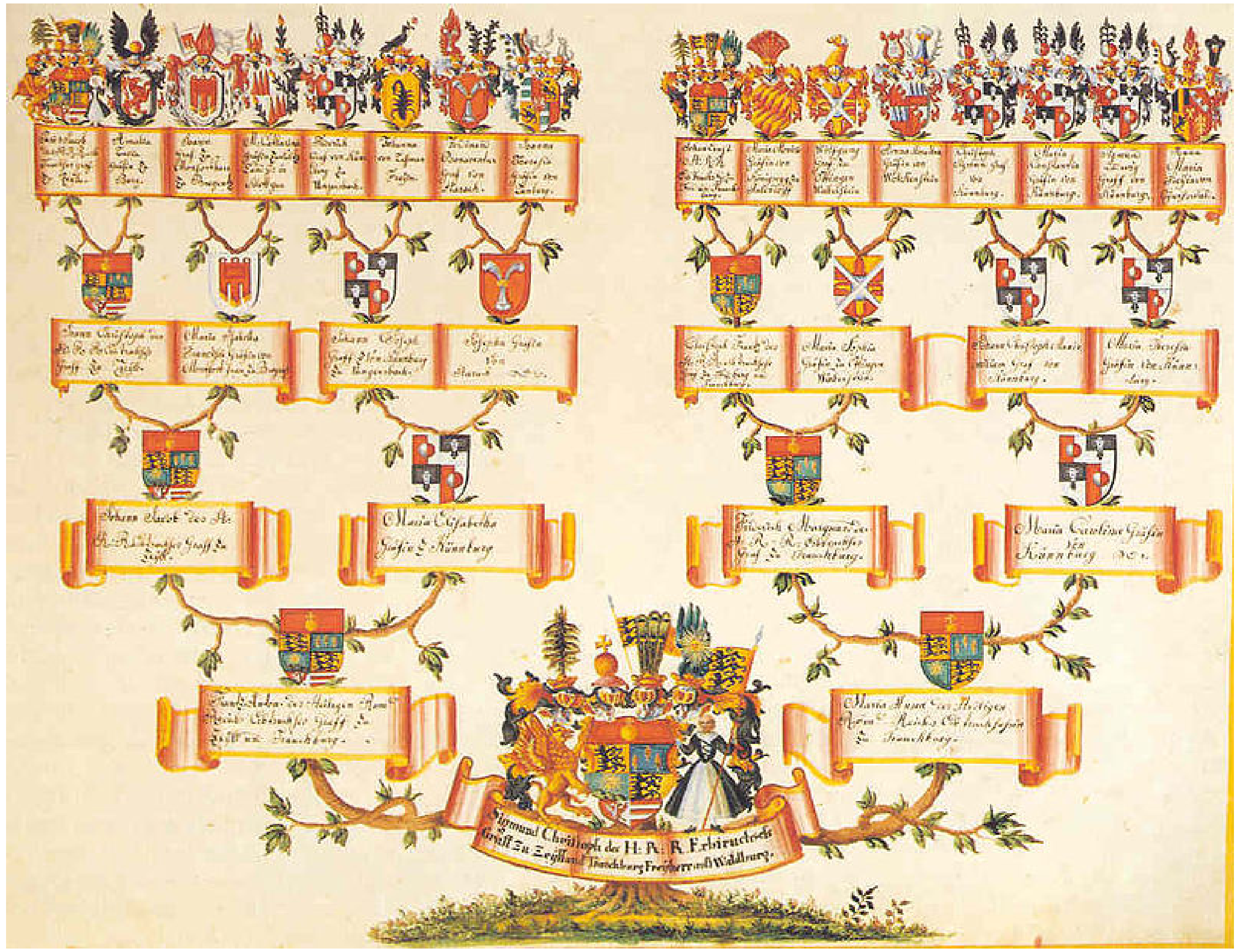
What is recursion?

> Recursion is the use of a procedure, subroutine, function, or algorithm that calls itself one or more times until a specified condition is met

Real-world example for recursion

Family Tree - Find all fathers until the 5th generations

- Reduce the problem to a smaller problem of the same type
  1. Whole problem: Find all generations
  2. Small problem: Find the father, find the father of the father, ...
- Limit the number of steps



Facts about recursion

Advantage:

- Solve problems in a recursive way
- Easy to read and follow
- Recursion could be limited by the termination condition

Disadvantage:

- Slow execution time

Simple recursion example - Sum of numbers

Mathematical definition

The sum of consecutive numbers is defined recursively as follows:

```
number = 1
    for iteration = 1
number = number + (iteration - 1)
    for iteration > 1
```

The sum of numbers to 5 is:

```
1+2+3+4+5 = 15
```

Simple recursion example - Sum of numbers

- Recursion with SQL: **Common Table Expression - CTE**

```
WITH calculate_SumOfNumbers AS
     ( -- Initial Query
    SELECT 1 AS iteration, 1 AS SumOfNumbers
     UNION ALL
     -- Recursion Part
     SELECT iteration + 1, SumOfNumbers + (iteration + 1)
         FROM calculate_SumOfNumbers
         WHERE iteration < 6
     )
SELECT SumOfNumbers
    FROM calculate_SumOfNumbers
```

Let's practice!

HIERARCHICAL AND RECURSIVE QUERIES IN SQL SERVER

Solve recursive maths problems

HIERARCHICAL AND RECURSIVE QUERIES IN SQL SERVER

SQL

Dominik Egarter
Data Engineering Enthusiast

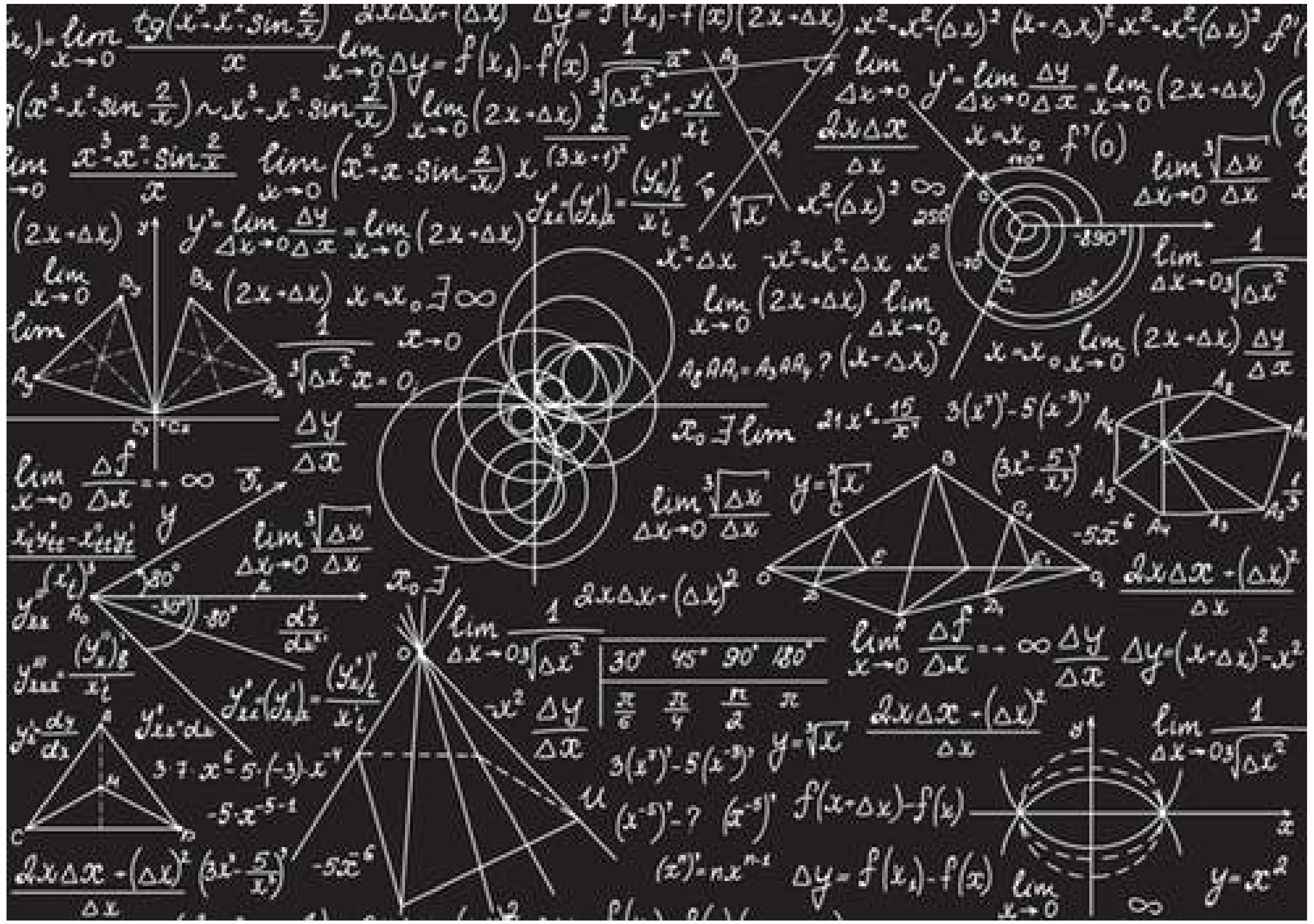
datacamp

The 2 maths problems

- Count numbers recursively
- Calculate the sum of potencies

Pseudo-Code:

> Informal high-level description of the operation principle. It is intended for human reading rather than machine reading.



Counting numbers

Recursive definition:

```
number = 1 for iteration = 1,
number = number +1 for iteration > 1
```

Pseudo-code:

```
WITH recursion AS (
    number = 1 -- Initial query
    UNION ALL
    number = number + 1 -- Recursive query)
-- Statement on CTE
SELECT * FROM recursion;
```

```
1+1+1+1 = 4
```

The sum of potencies

Recursive definition:

```
number = 1 for iteration = 1
number = number + iteration^iteration for iteration > 1
```

Pseudo-code

```
WITH recursion AS (
      number = 1 -- Initial query
      UNION ALL
    number = number + iteration^iteration -- Recursive query
SELECT * FROM recursion;
```

```
1 + 2^2 + 3^3 = 32
```

Let's practice!

HIERARCHICAL AND RECURSIVE QUERIES IN SQL SERVER