Introduction to Data Science

Lecture 6; November 9th, 2016

Ernst Henle
ErnstHe@UW.edu
Skype: ernst-henle

Copyright 2016 by Ernst Henle

Agenda

- Announcements
  - The social component is a course requirement: Contribute on LinkedIn and collaborate on homework!
  - Guest Lecture in November: Business Side of Data Science by Marius Marcu on November 16th 2016
- Review Relational Algebra (Homework)
- Quiz 06a Relational Algebra
- Sparse 2D Matrix Format
- Sparse Matrix Manipulation in SQL (related Homework Assignment)
- Break
- Sparse Matrices and EAV
- Quiz 06b EAV
- Sparse Matrix Exercises (related to Homework Assignment)
- Break
- Predictive Analytics Iteration Trap
- Predictive Faux Pas (Time Permitting)
- NoSQL Scale Out (Time Permitting)
- Assignment. See assignment slides at the end of the deck. Complete all assignments items from all assignment slides. Submit by Saturday 11:57 PM

Review

- Homework 1-5: RelationalAlgebraSQLHomework.sql
- Homework 6: TestOuterJoin.R
- Last week's quiz: PregnancyExercise_complete.R
- References:
    - RelationalAlgebraAndSQL.pdf
    - RelationalAlgebraAndSQL.sql

Quiz 06a (Relational Algebra)

- The questions are presented during the quiz

Sparse 2D Matrices

Copyright 2016 by Ernst Henle

Sparse 2D Matrices

2D Matrix

- A 2D Matrix is a rectangular data structure with rows and columns.
- Each matrix element is uniquely identified by a row and column.
- All matrix elements have the same type. (Not so in a table!)
- Typically, rows and columns are numbered.

Sparse 2D Matrices

2D Matrix

- A 2D Matrix is a rectangular data structure with rows and columns.
- Each matrix element is uniquely identified by a row and column.
- All matrix elements have the same type. (Not so in a table!)
- Typically, rows and columns are numbered.
- Below is a random 2D matrix generated in R.

```
> matrix(trunc(10*runif(35)), ncol=7, nrow=5)


         [,1] [,2] [,3] [,4] [,5] [,6] [,7]
    [1,]   7    4    8    8    2    2    5
    [2,]   6    8    0    3    5    0    1
    [3,]   4    9    7    9    1    8    4
    [4,]   9    3    3    7    3    2    0
    [5,]   3    1    9    2    2    0    7
```

Sparse 2D Matrices

2D Matrix

- A 2D Matrix is a rectangular data structure with rows and columns.
- Each matrix element is uniquely identified by a row and column.
- All matrix elements have the same type. (Not so in a table!)
- Typically, rows and columns are numbered.
- Below is a random 2D matrix generated in R.
- This 2D matrix is dense because most of the elements are non-null

Dense 2D Matrix

	C1	C2	C3	C4	C5	C6	C7
R1	7	4	8	8	2	2	5
R2	6	8	0	3	5	0	1
R3	4	9	7	9	1	8	4
R4	9	3	3	7	3	2	0
R5	3	1	9	2	2	0	7

Sparse 2D Matrices

Sparse 2D Matrix

- In a Sparse 2D Matrix most of the matrix elements are null.
- Left below is a sparse 5 X 7 matrix with 4 entries and 31 nulls.

Most matrix elements have null values

Sparse 2D Matrix

	C1	C2	C3	C4	C5	C6	C7
R1				8			
R2						0	1
R3							
R4							
R5	3						

Most matrix elements have non-null values

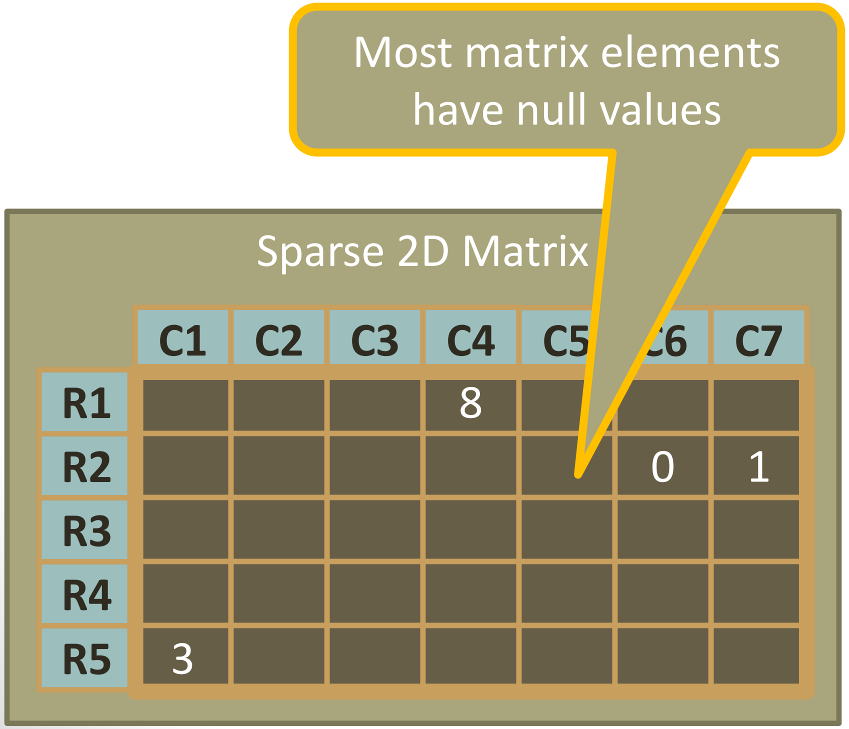
Dense 2D Matrix

	C1	C2	C3	C4	C5	C6	C7
R1	7	4	8	8	2	2	5
R2	6	8	0	3	5	0	1
R3	4	9	7	9	1	8	4
R4	9	3	3	7	3	2	0
R5	3	1	9	2	2	0	7

Sparse 2D Matrices

Sparse 2D Matrix

- In a Sparse 2D Matrix most of the matrix elements are null.
- Below is a sparse 5 X 7 matrix with 4 entries and 31 nulls.
- The traditional matrix layout is wasteful for large sparse matrices



Most matrix elements have null values

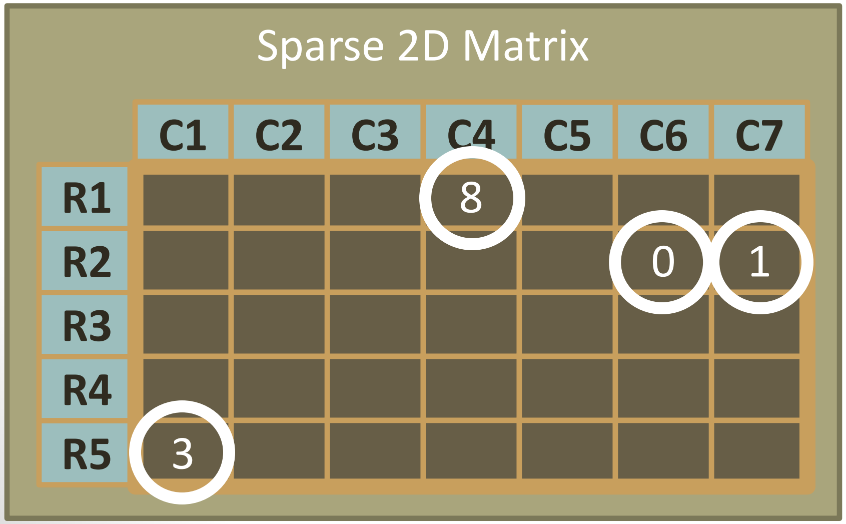
Sparse 2D Matrix

	C1	C2	C3	C4	C5	C6	C7
R1				8			
R2						0	1
R3							
R4							
R5	3						

Sparse 2D Matrices

Sparse 2D Matrix

- In a Sparse 2D Matrix most of the matrix elements are null.
- Below is a sparse 5 X 7 matrix with 4 entries and 31 nulls.
- The traditional matrix layout is wasteful for large sparse matrices
- An efficient representation would only reference non-null values.



Sparse 2D Matrix

	C1	C2	C3	C4	C5	C6	C7
R1				8			
R2						0	1
R3							
R4							
R5	3						

Sparse 2D Matrices

The Sparse 2D Matrix format has three columns: Row (R), Column (C), and Value (V)

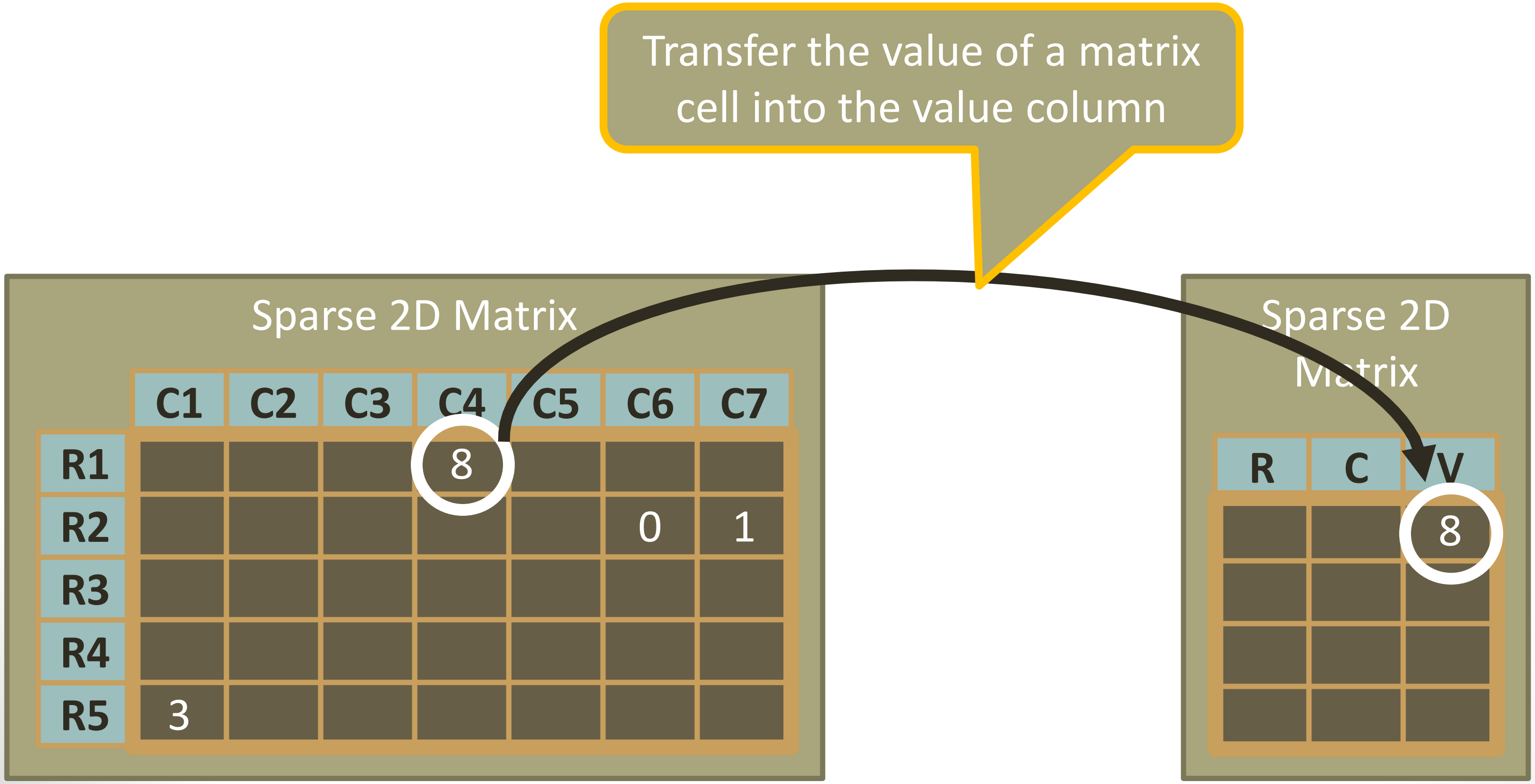
Sparse 2D Matrix

	C1	C2	C3	C4	C5	C6	C7
R1				8			
R2						0	1
R3							
R4							
R5	3						

Sparse 2D Matrix

R	C	V

Sparse 2D Matrices



Transfer the value of a matrix cell into the value column

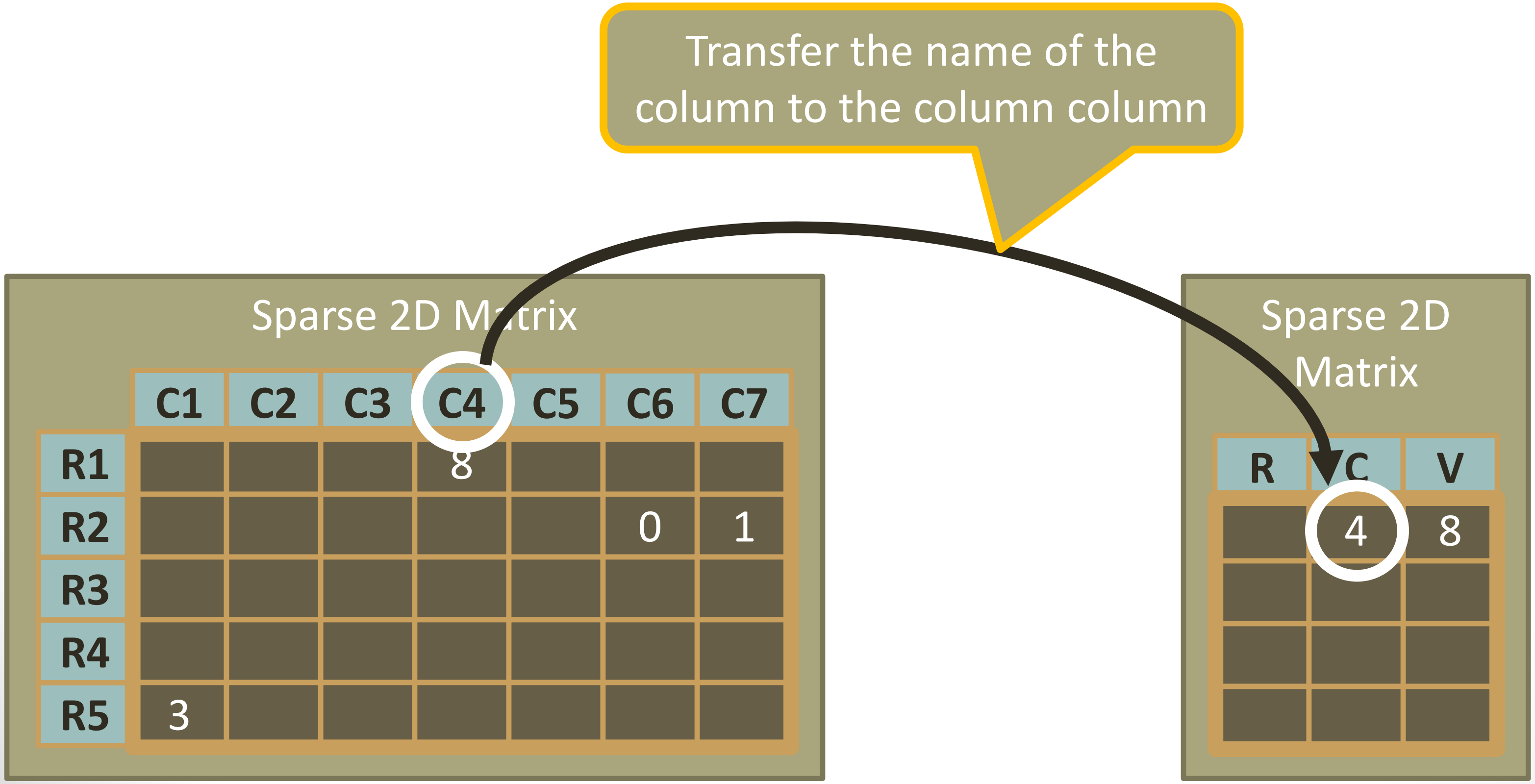
Sparse 2D Matrix

	C1	C2	C3	C4	C5	C6	C7
R1				8			
R2						0	1
R3							
R4							
R5	3						

Sparse 2D Matrix

R	C	V
		8

Sparse 2D Matrices



Transfer the name of the column to the column column

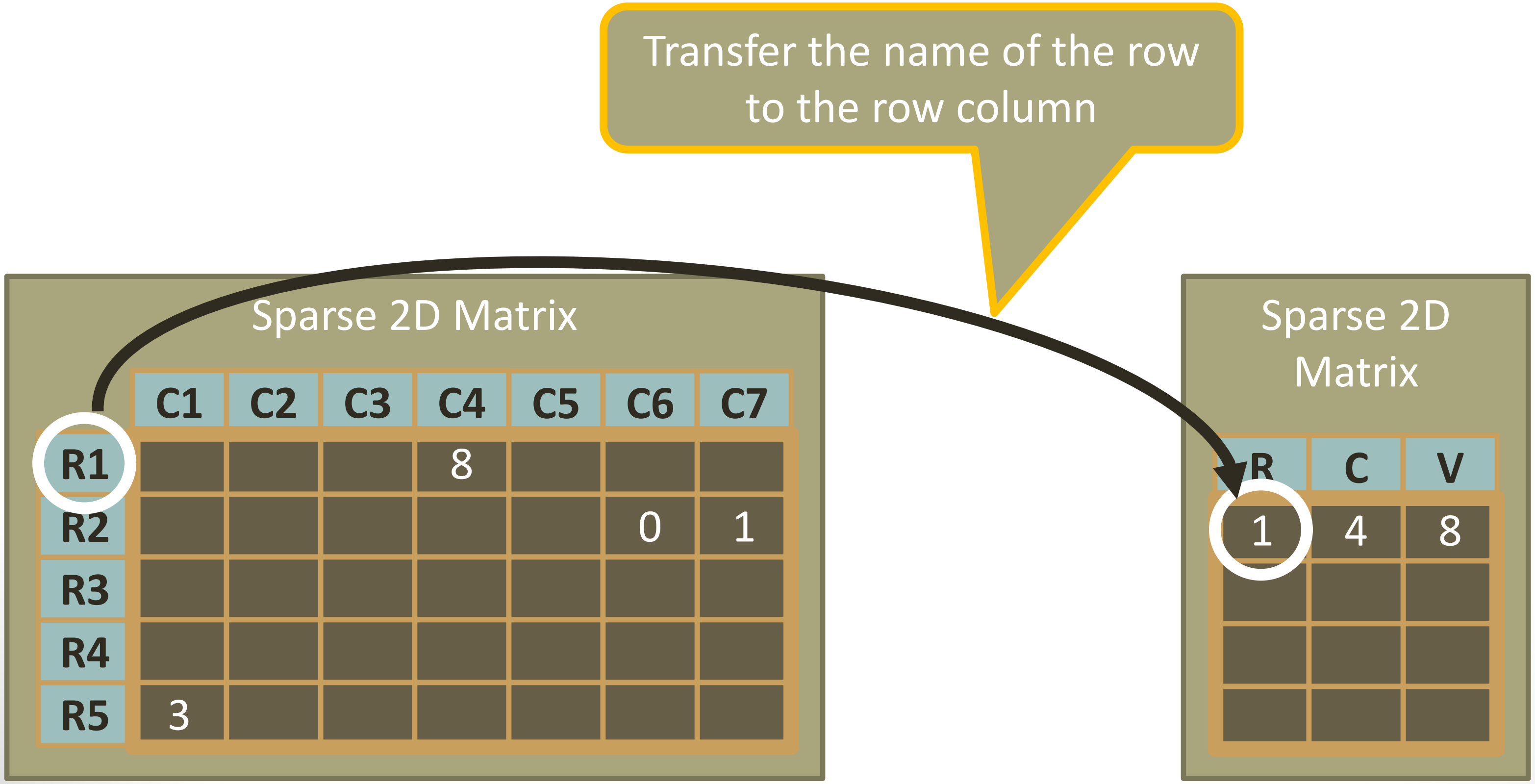
Sparse 2D Matrix

	C1	C2	C3	C4	C5	C6	C7
R1				8			
R2						0	1
R3							
R4							
R5	3						

Sparse 2D Matrix

R	C	V
	4	8

Sparse 2D Matrices



Transfer the name of the row to the row column

Sparse 2D Matrix

	C1	C2	C3	C4	C5	C6	C7
R1				8			
R2						0	1
R3							
R4							
R5	3						

Sparse 2D Matrix

R	C	V
1	4	8

Sparse 2D Matrices

Sparse 2D Matrix Format.
Does not contain null values

Sparse 2D Matrix

	C1	C2	C3	C4	C5	C6	C7
R1				8			
R2						0	1
R3							
R4							
R5	3						

Sparse 2D Matrix

R	C	V
1	4	8
2	6	0
2	7	1
5	1	3

Sparse 2D Matrices

Examples of Sparse 2D-Matrix Manipulation in a relational database. Open MatrixAlgebra.sql

- Matrix Addition
- Scalar Multiplication
- Matrix Multiplication
  - Inner Product (Dot Product, Scalar Product)
  - Outer Product (Cartesian Product)
- Matrix Transposition

Sparse 2D Matrices

Copyright 2016 by Ernst Henle

Break

Data as Sparse Matrices

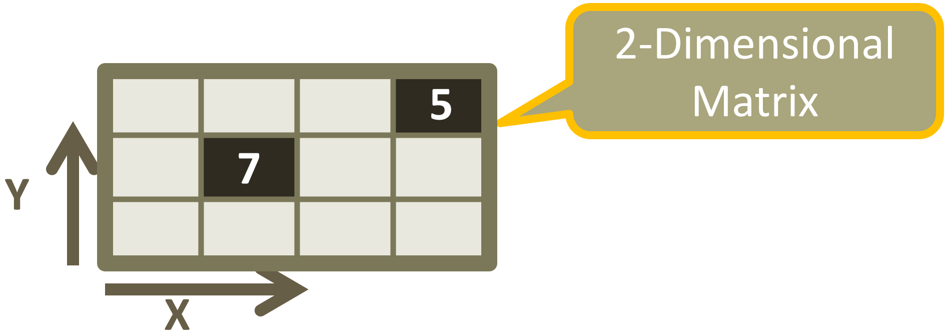
Multidimensional Sparse Matrices

Copyright 2016 by Ernst Henle

Sparse Matrices

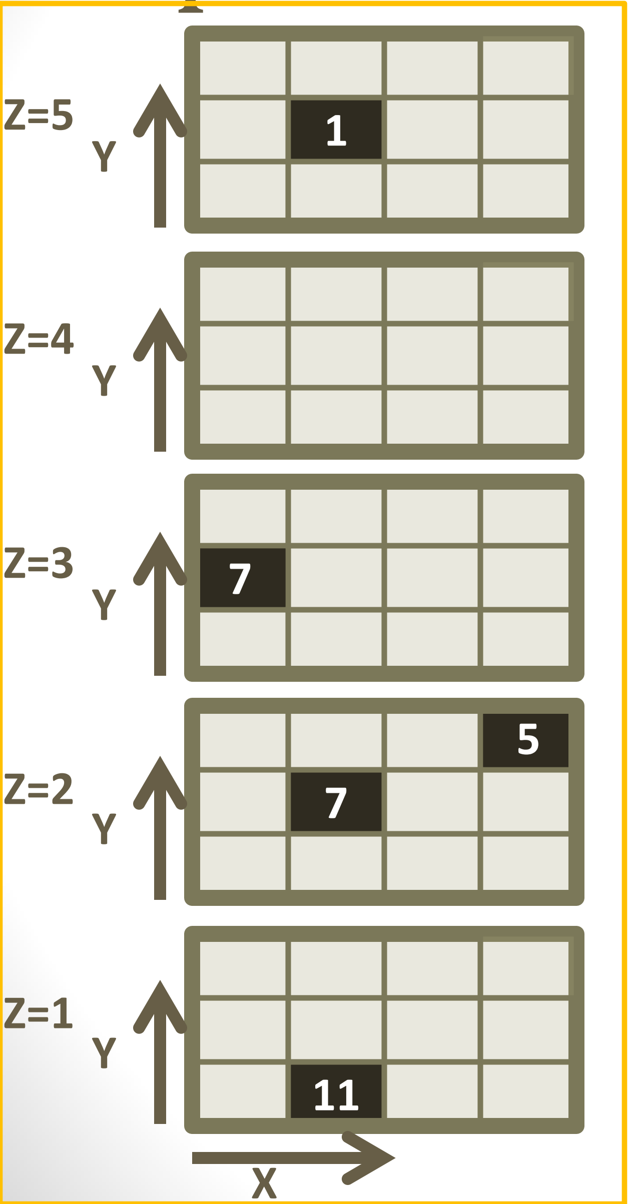
- Cartesian product
  - http://en.wikipedia.org/wiki/Cartesian_product
  - The Cartesian product of two sets A and B is the set of all ordered pairs ab, where a is element of A and b is element of B.
- Relational Algebra
  - http://en.wikipedia.org/wiki/Relational_algebra
  - In Relational Algebra we need the Cartesian product to combine tuples into a single tuple. The Cartesian product creates a new schema (relation) from other relations.
- Hyperrectangle (Sparse Multi-Dimensional Matrix)
  - http://en.wikipedia.org/wiki/Hyperrectangle
  - Hyperrectangle is the generalization of a rectangle for higher dimensions and is defined as the Cartesian product of intervals

Sparse Matrices



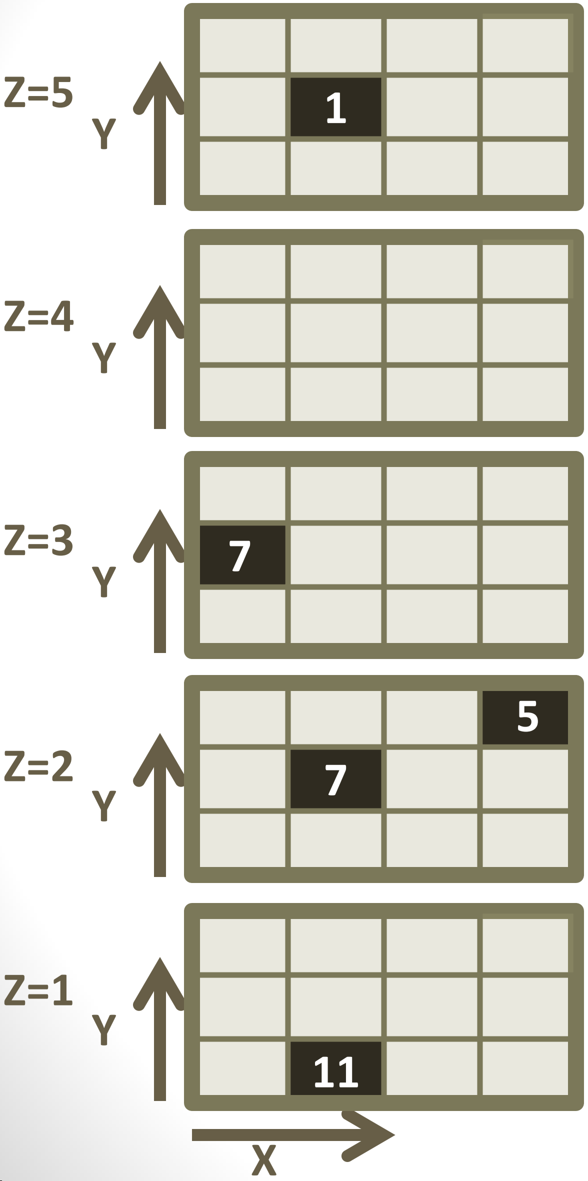
2-Dimensional Matrix

Sparse Matrices



A series of equal-sized 2-dimensional matrices is a 3-dimensional matrix

Sparse Matrices



Z=5

	1		

Z=4

Z=3

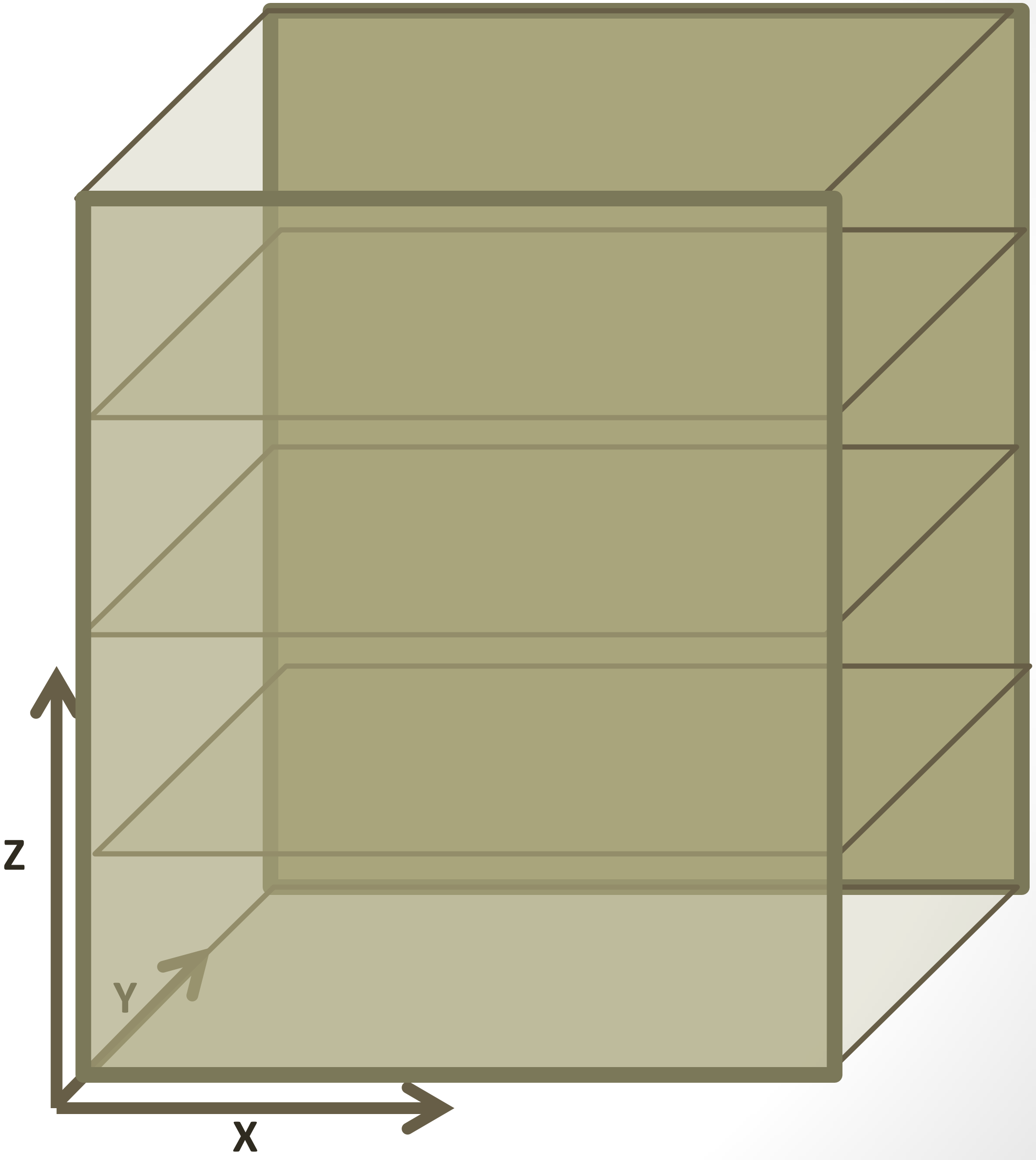
7			

Z=2

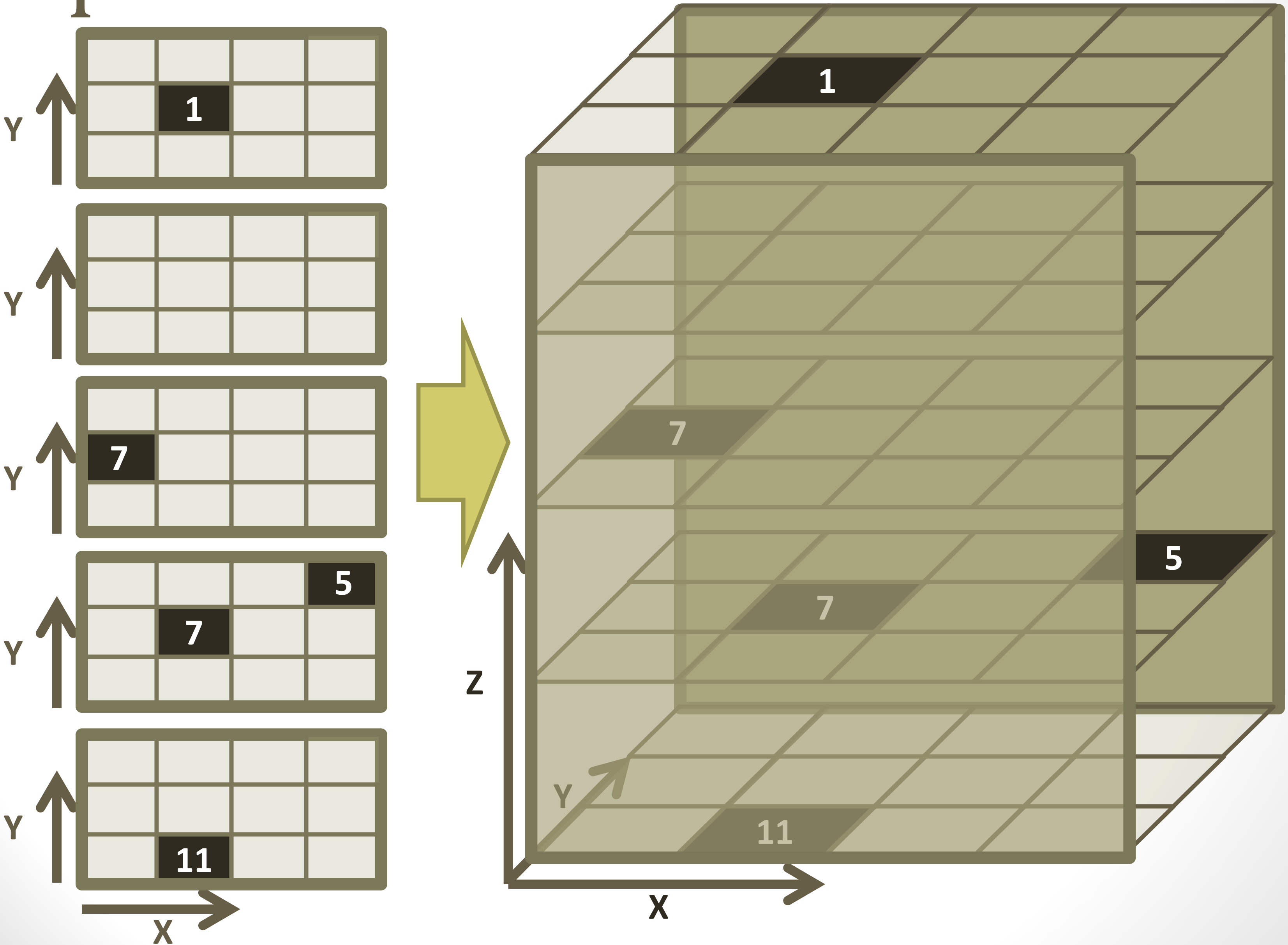
			5
	7		

Z=1

	11		



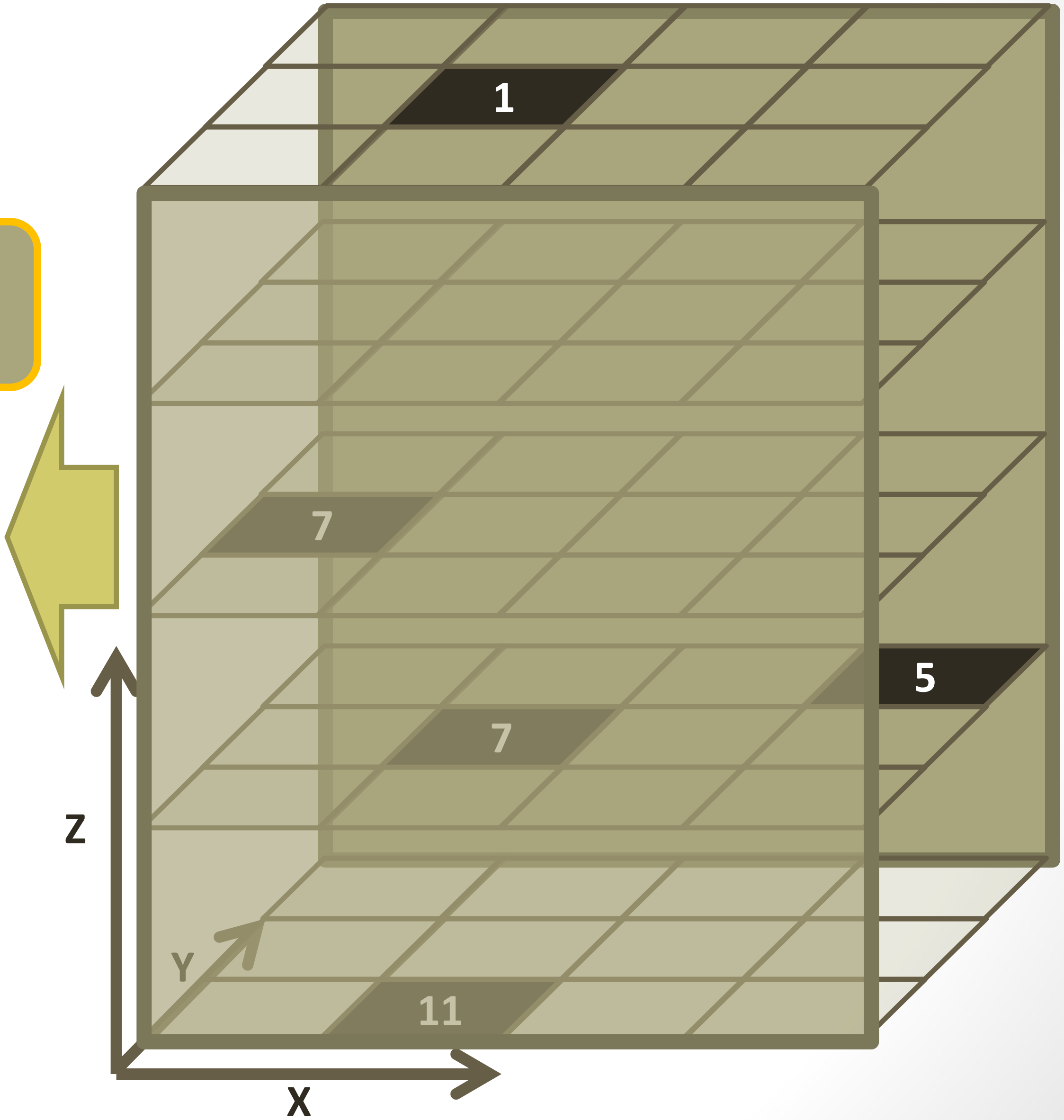
Sparse Matrices



Sparse Matrices

A table with n columns represents values in an n-1 dimensional matrix

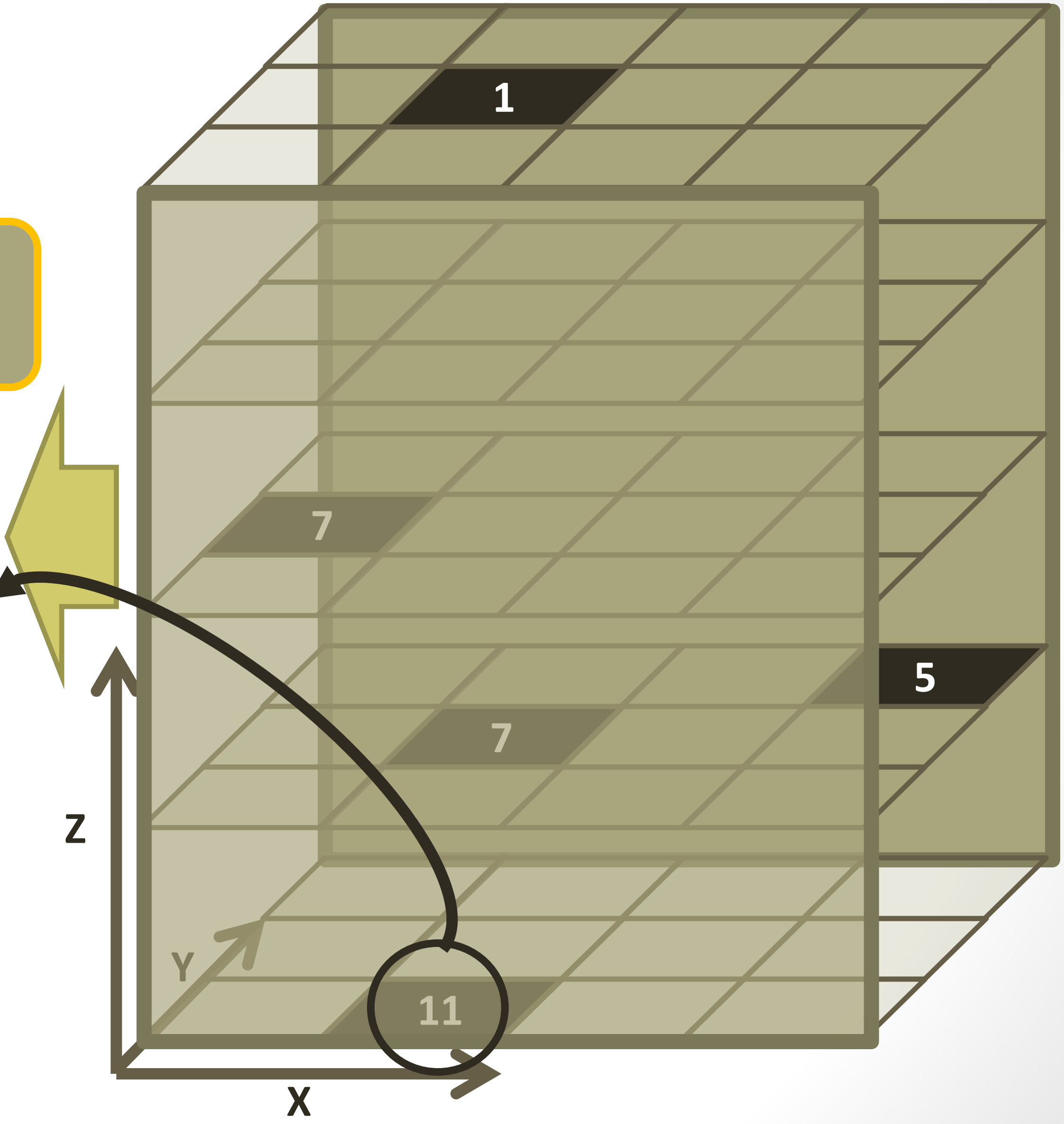
<u>X</u>	<u>Y</u>	<u>Z</u>	<u>V</u>



Sparse Matrices

A table with n columns represents values in an n-1 dimensional matrix

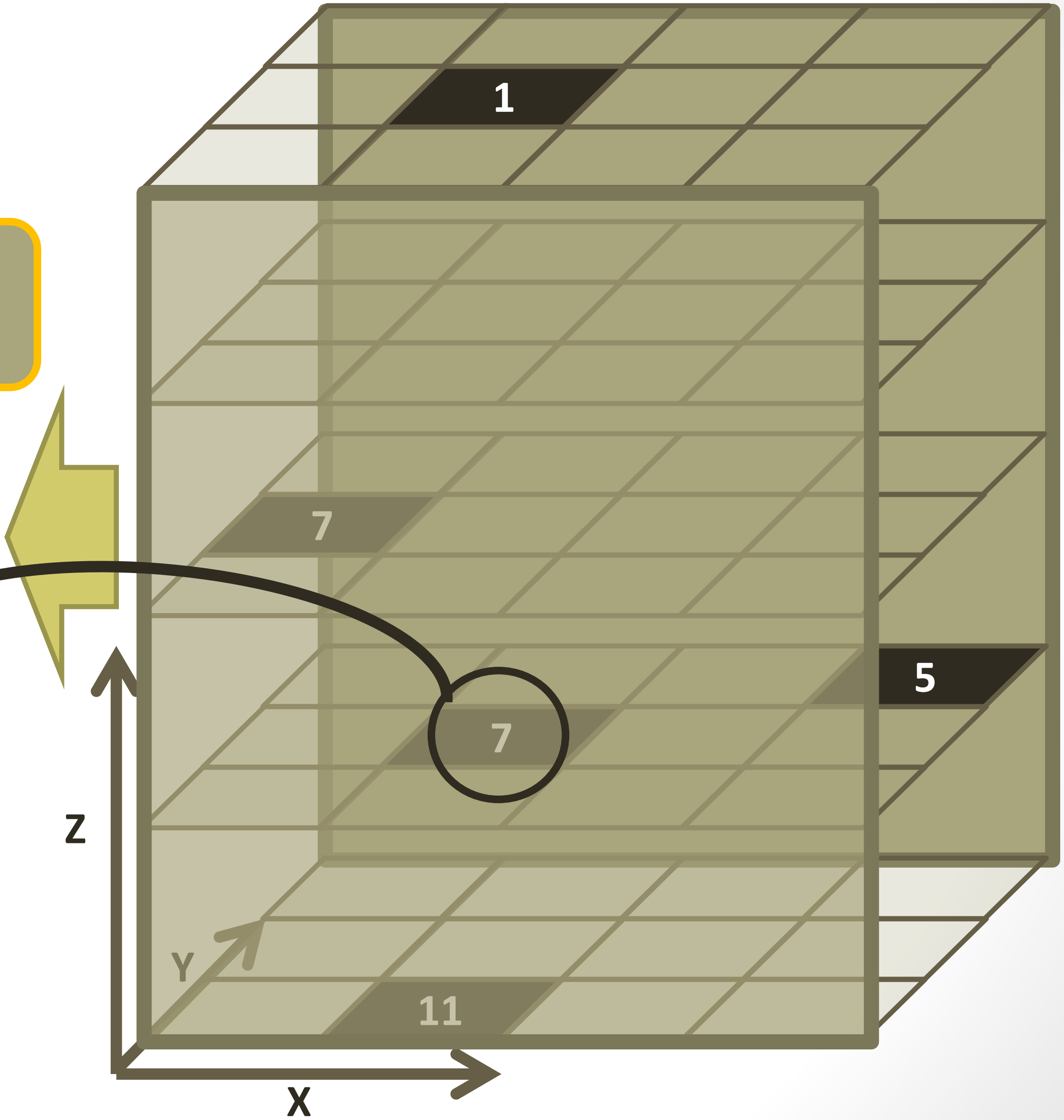
X	Y	Z	V
2	1	1	11



Sparse Matrices

A table with n columns represents values in an n-1 dimensional matrix

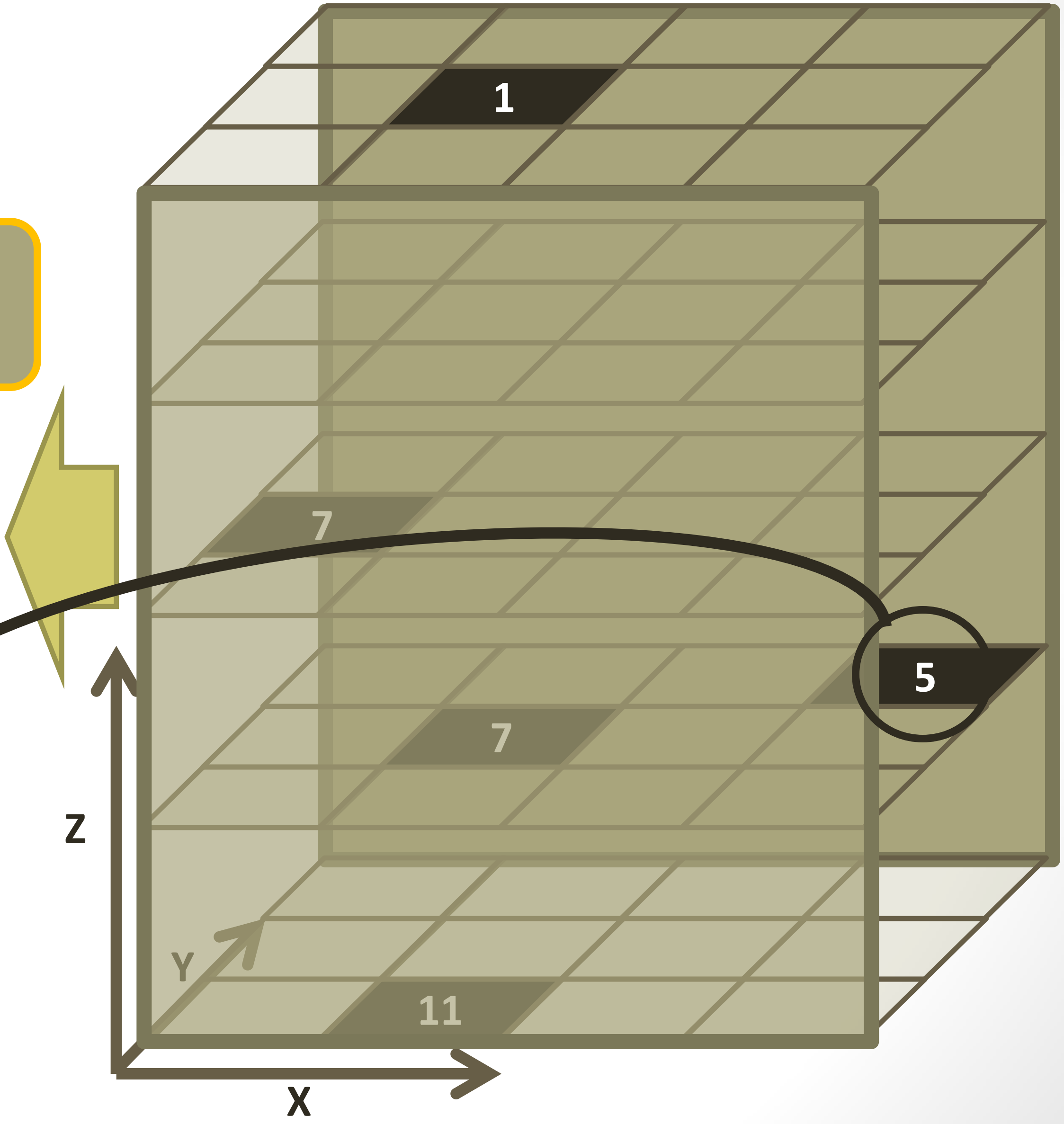
X	Y	Z	V
2	1	1	11
2	2	2	7



Sparse Matrices

A table with n columns represents values in an n-1 dimensional matrix

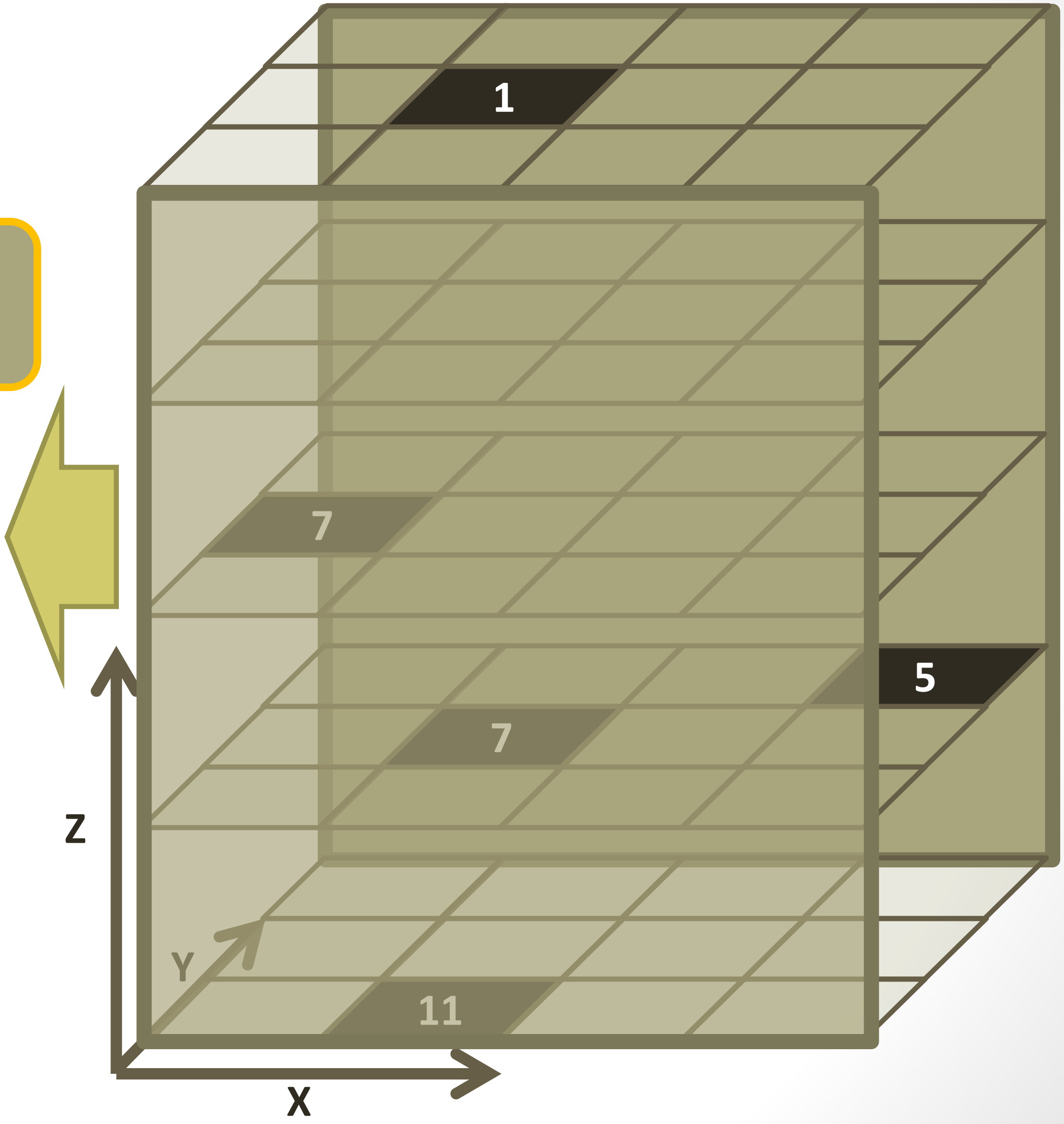
X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5



Sparse Matrices

A table with n columns represents values in an n-1 dimensional matrix

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1



Sparse Matrices

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1

Sparse Matrices

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1

Think of V as just another dimension

Sparse Matrices

A table with n columns represents points in an n-dimensional matrix

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1

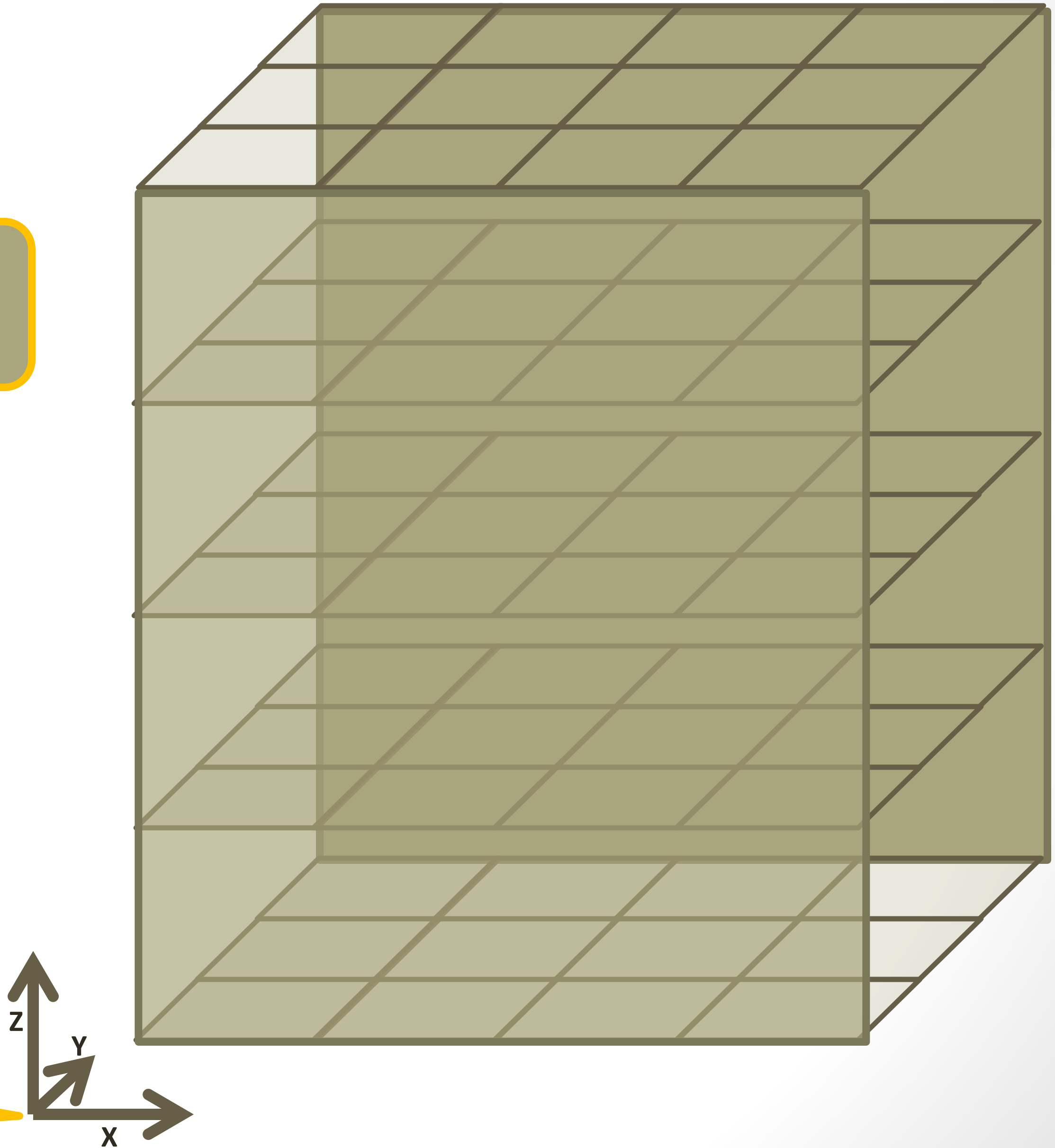
Think of V as just another dimension

Sparse Matrices

This table represents points in 4-Dimensional Space.

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1

3 Dimensions

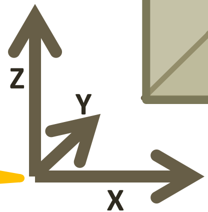


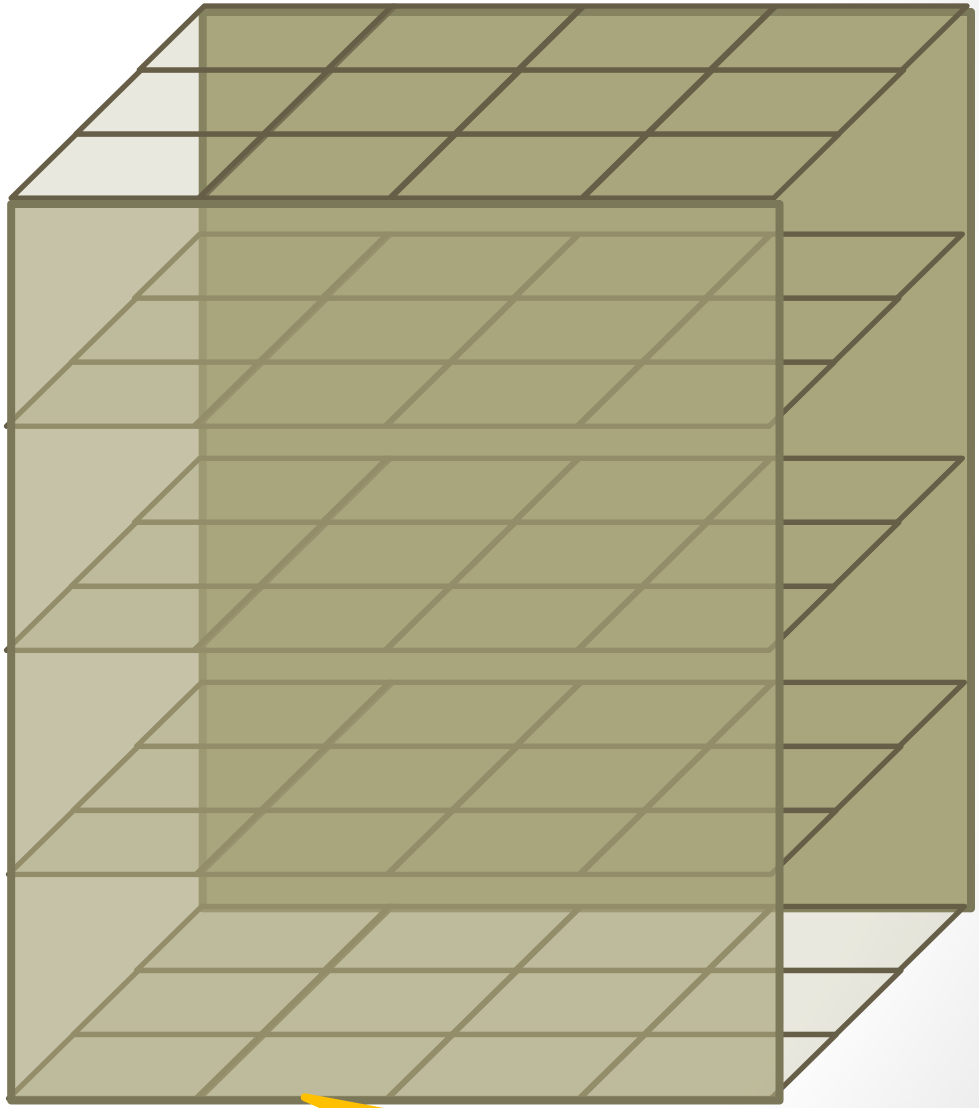
Sparse Matrices

This table represents points in 4-Dimensional Space.

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1

3 Dimensions





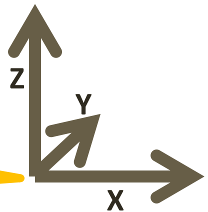
3-Dimensional Space

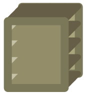
Sparse Matrices

This table represents points in 4-Dimensional Space.

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1

3 Dimensions



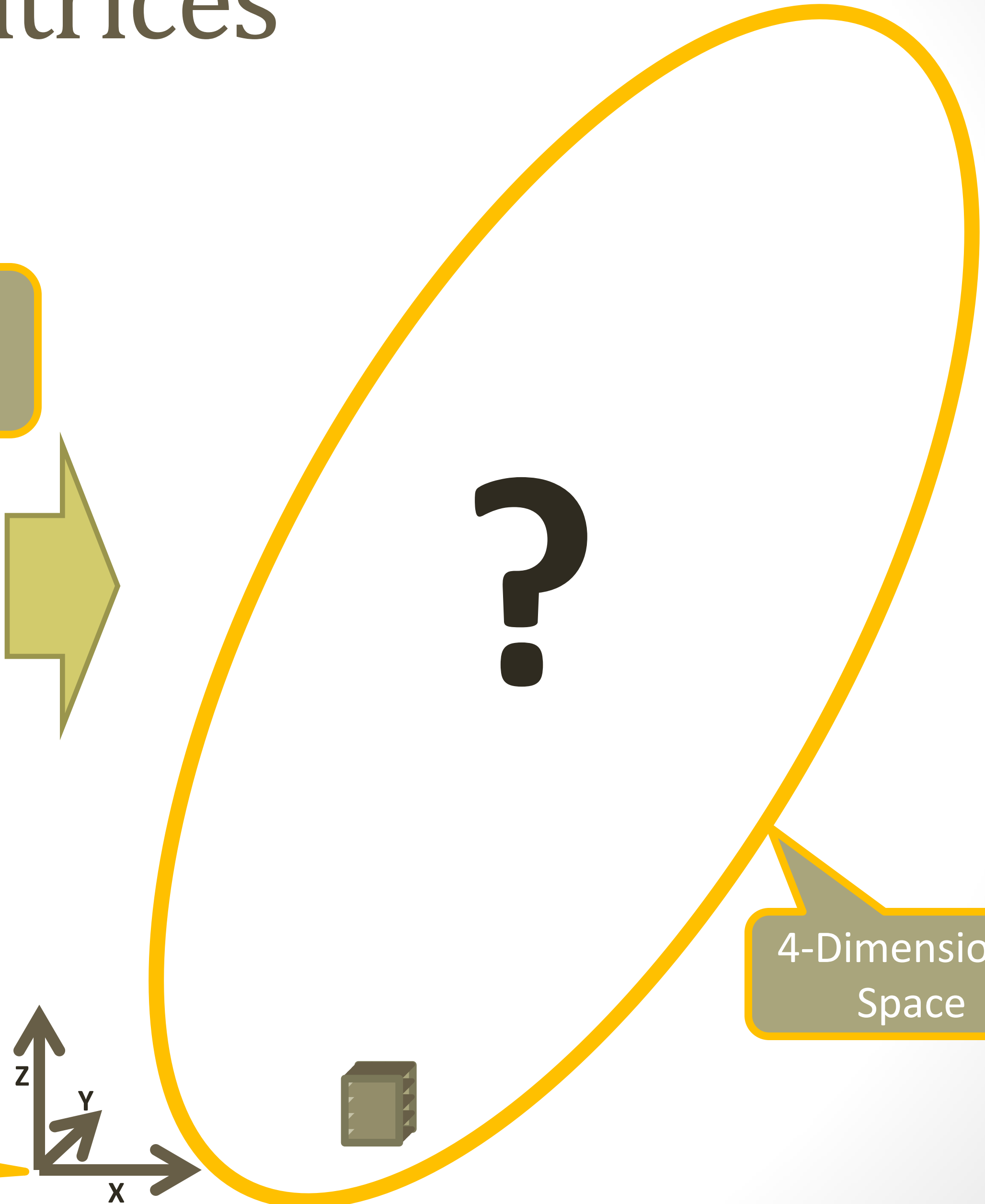


3-Dimensional Space

Sparse Matrices

This table represents points in 4-Dimensional Space.

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1



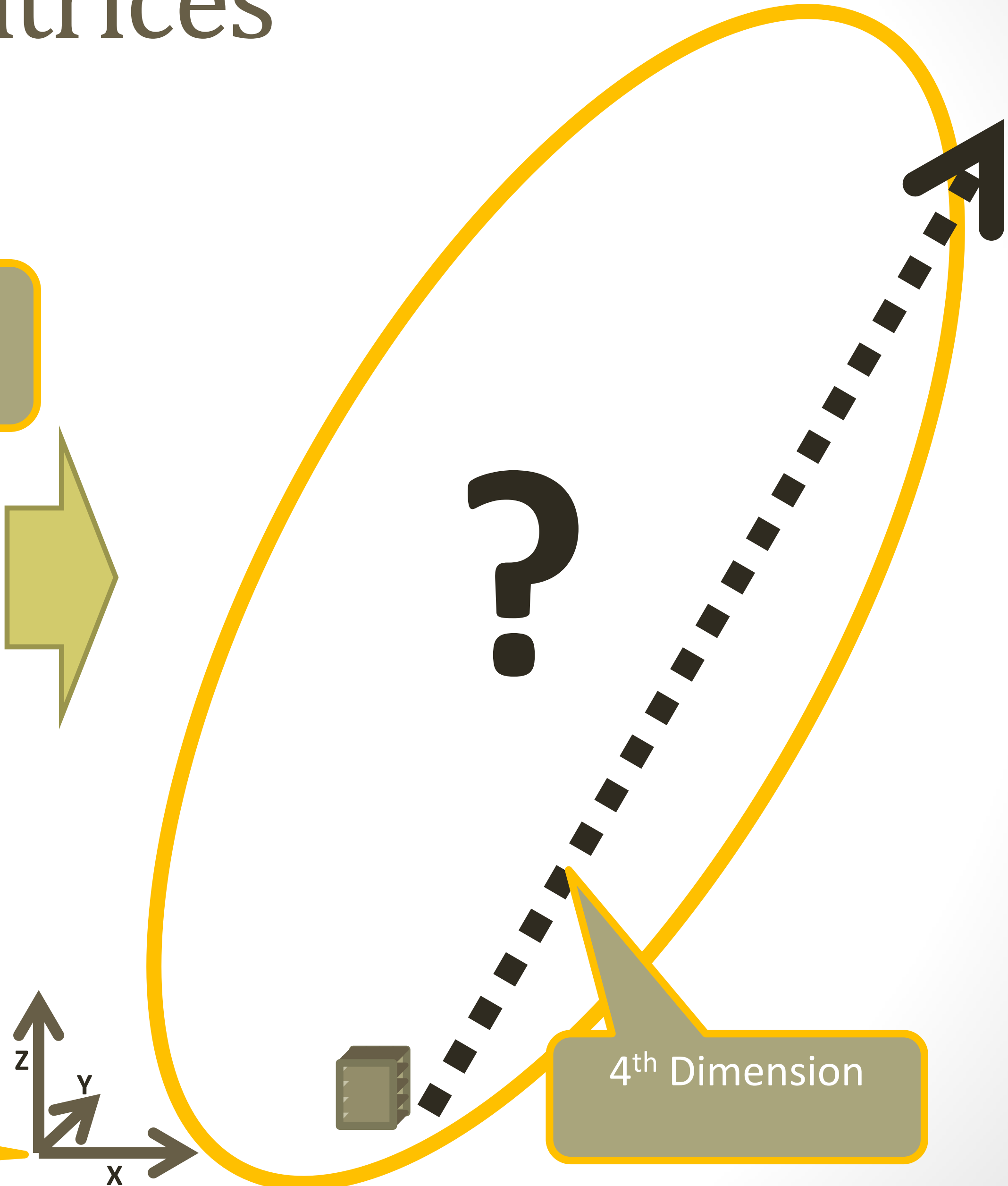
3 Dimensions

4-Dimensional Space

Sparse Matrices

This table represents points in 4-Dimensional Space.

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1



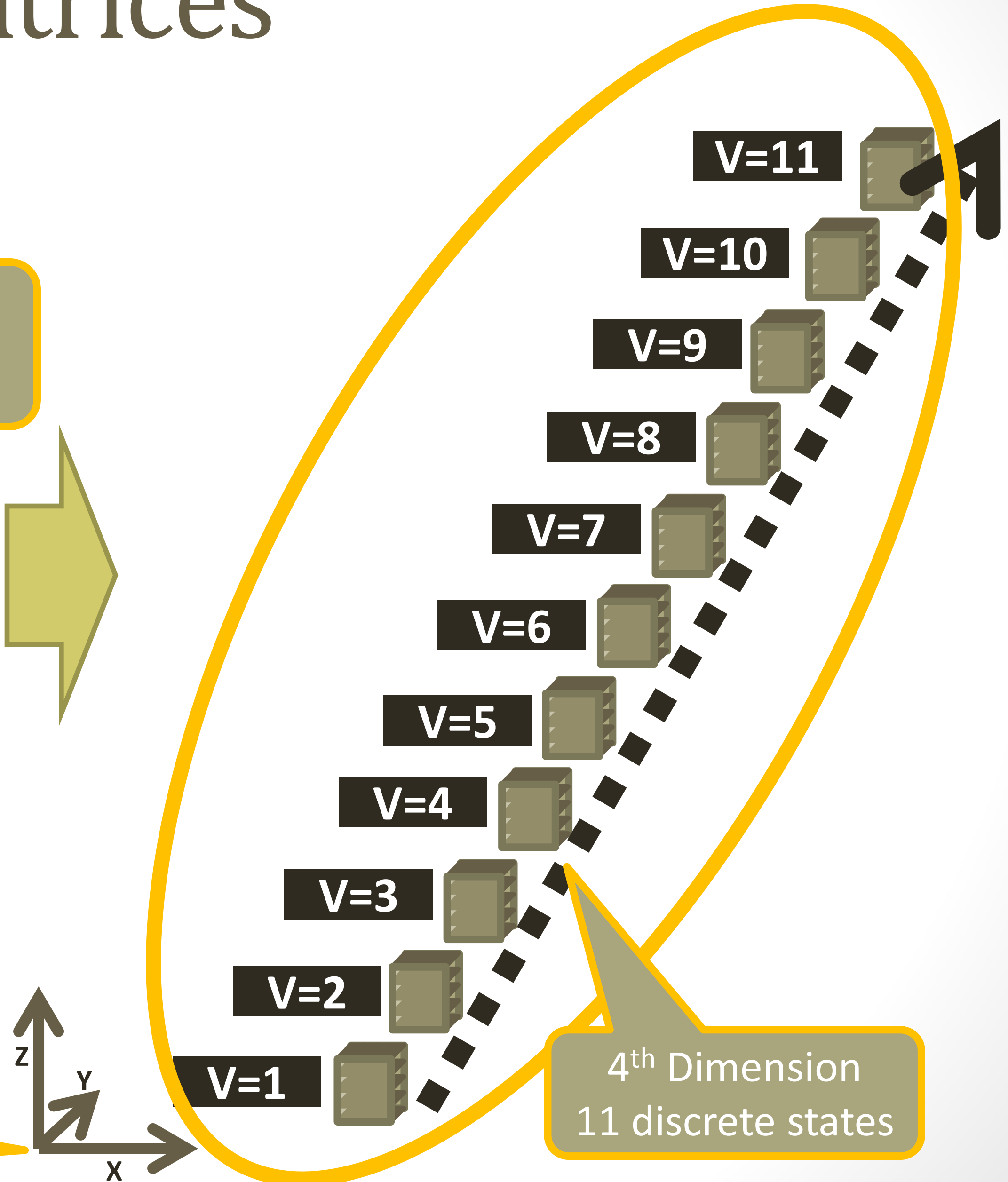
3 Dimensions

4th Dimension

Sparse Matrices

This table represents points in 4-Dimensional Space.

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1



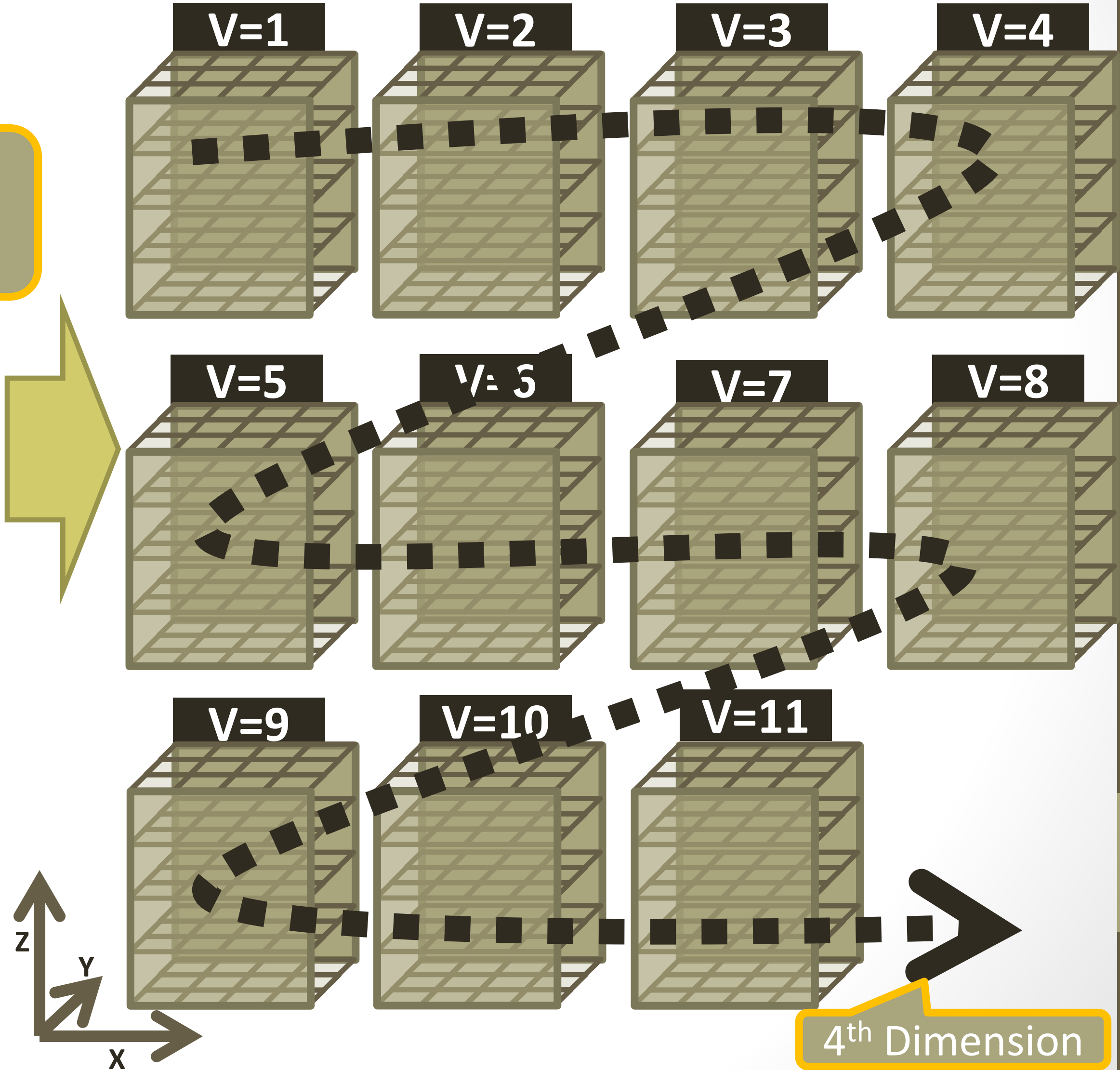
4th Dimension
11 discrete states

3 Dimensions

Sparse Matrices

This table represents points in 4-Dimensional Space.

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1

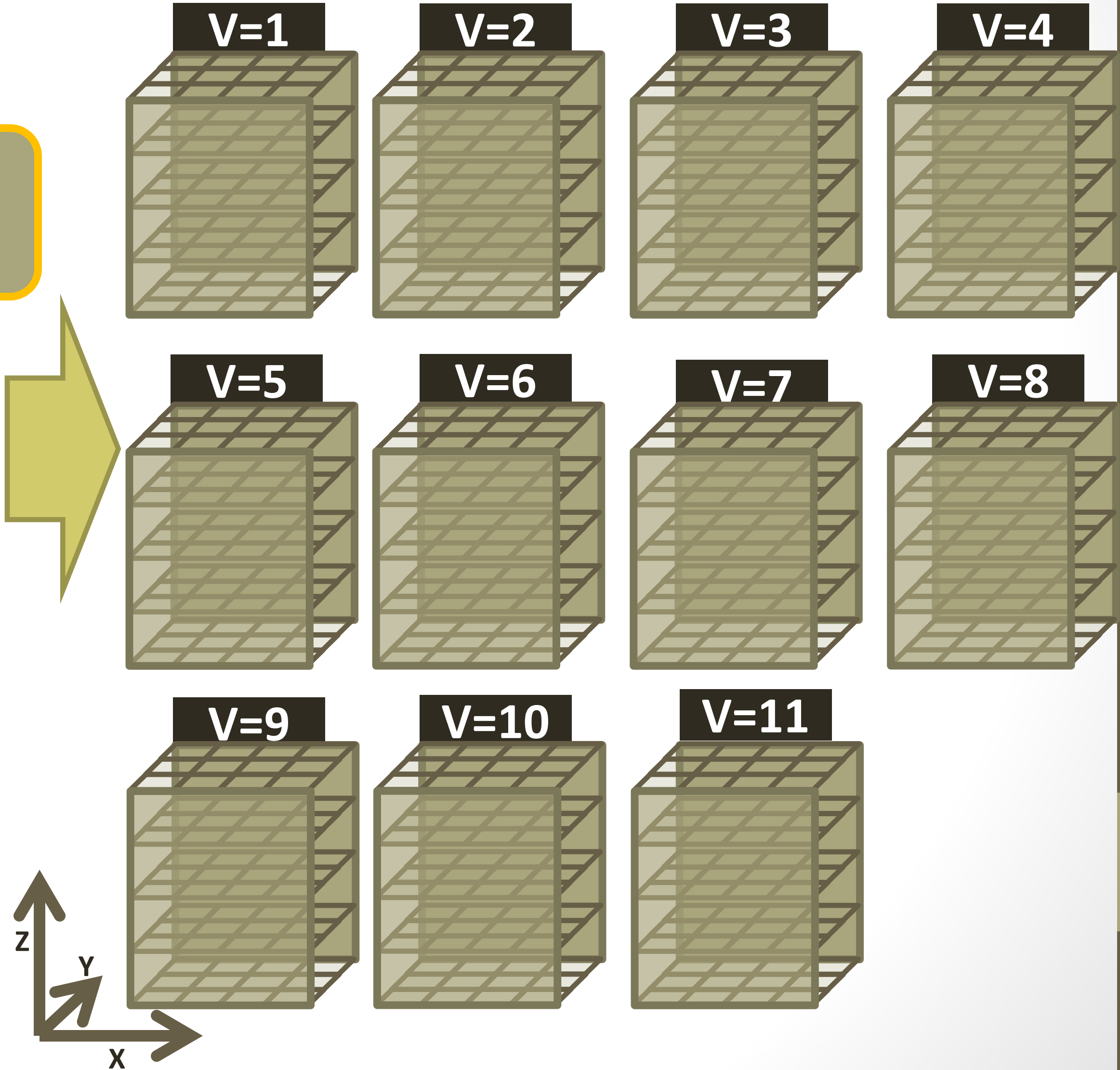


4th Dimension

Sparse Matrices

This table represents points in 4-Dimensional Space.

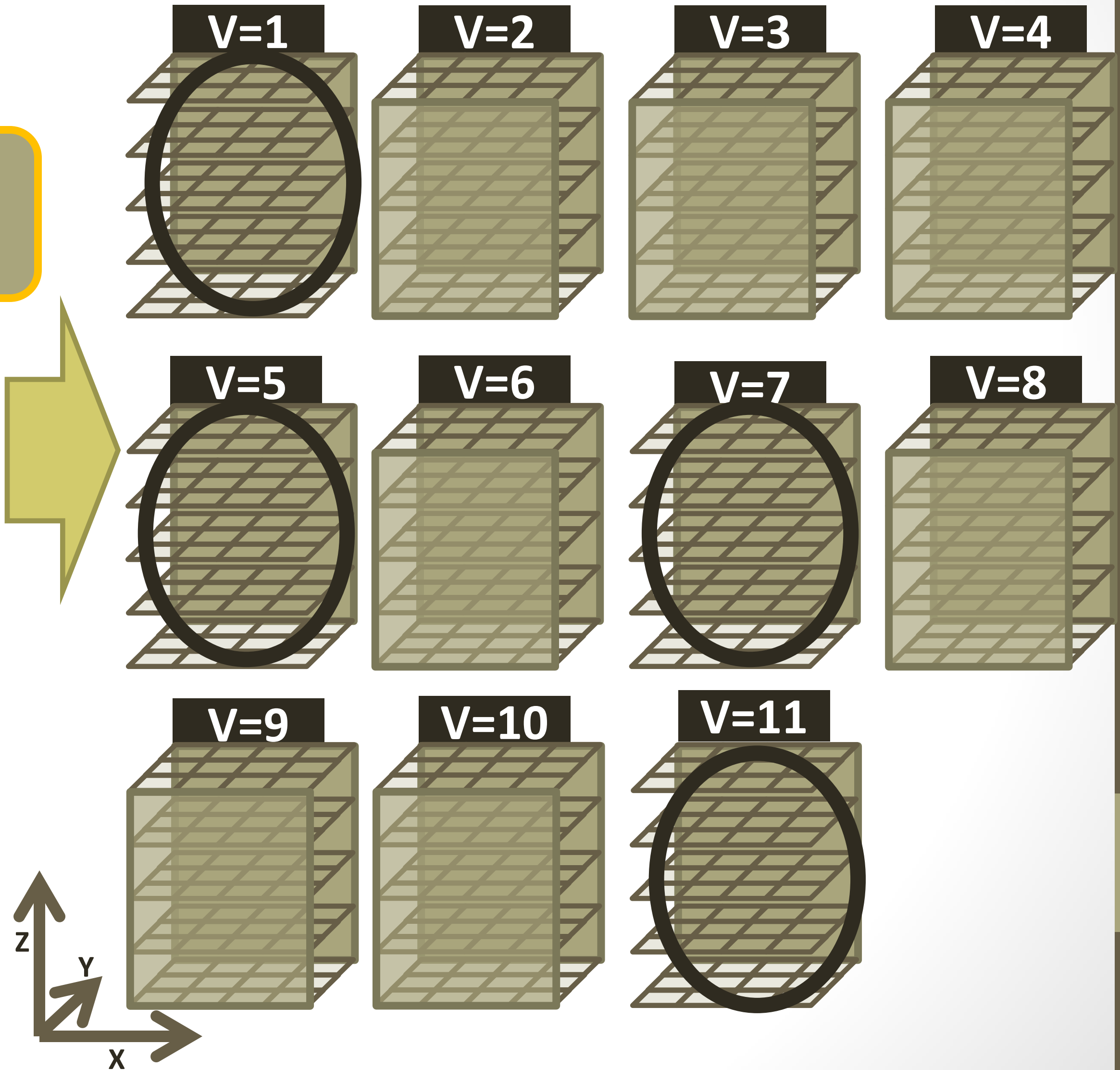
X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1



Sparse Matrices

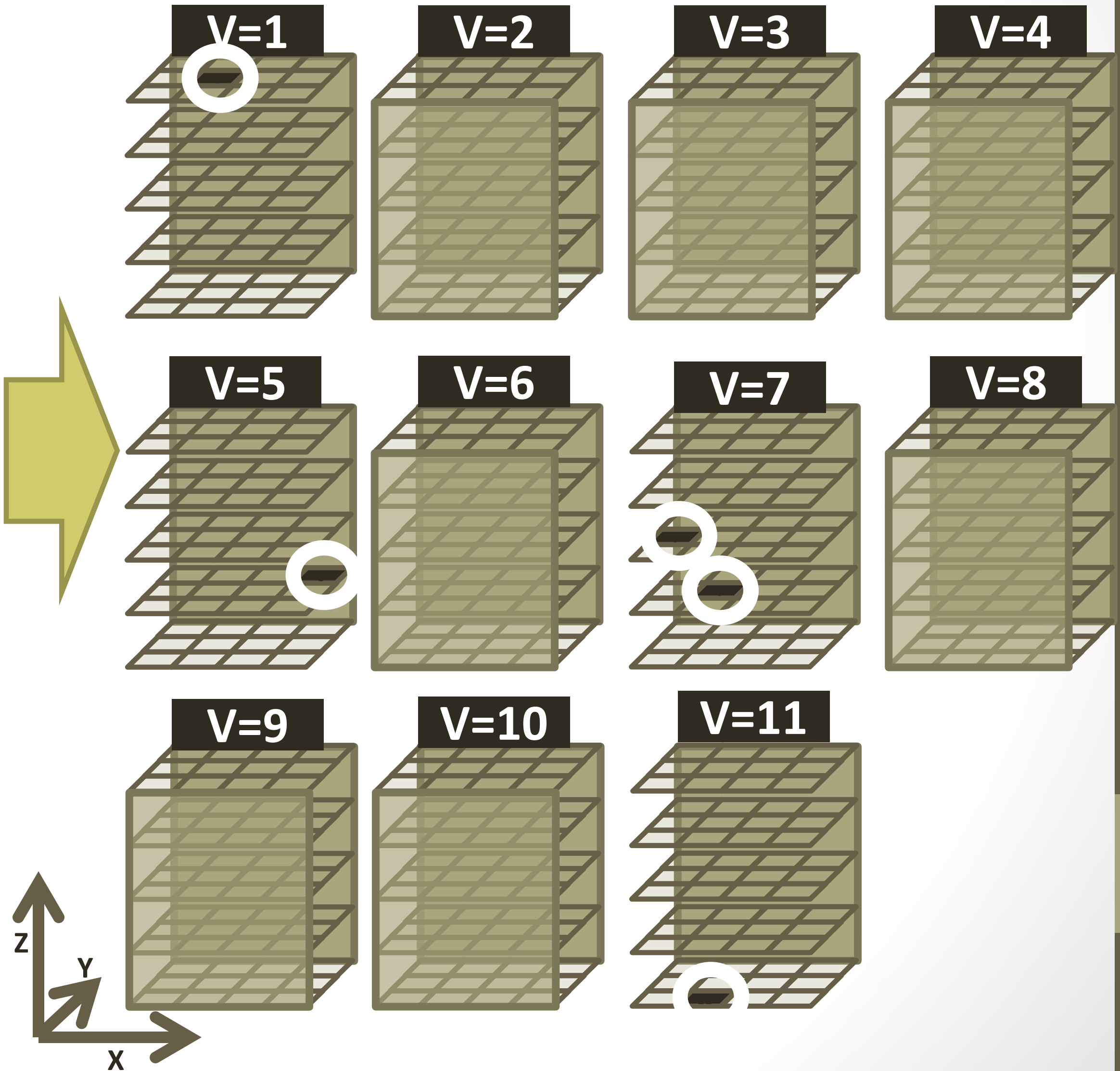
This table represents points in 4-Dimensional Space.

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1



Sparse Matrices

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1



Sparse Matrices: EAV

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1

Sparse Matrices: EAV

A table represents points in n-Dimensional Space.

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1

Sparse Matrices: EAV

A table represents points in n-Dimensional Space.

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1

- Can we represent all tables in a single schema?
- Any table or matrix cell can be described by row, column and value.
- Represent each cell of a table in its own row.
- **Entity–attribute–value model**

Sparse Matrices: EAV

Column Name

Row ID. Needs to be unique for a given row in the original table. Does not need to be a number or sequential

Cell Values

R	C	M

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1

- Can we represent all tables in a single schema?
- Any table or matrix cell can be described by row, column and value.
- Represent each cell of a table in its own row.
- **Entity–attribute–value model**

Sparse Matrices: EAV

Row ID. Needs to be unique for a given row in the original table. Does not need to be a number or sequential

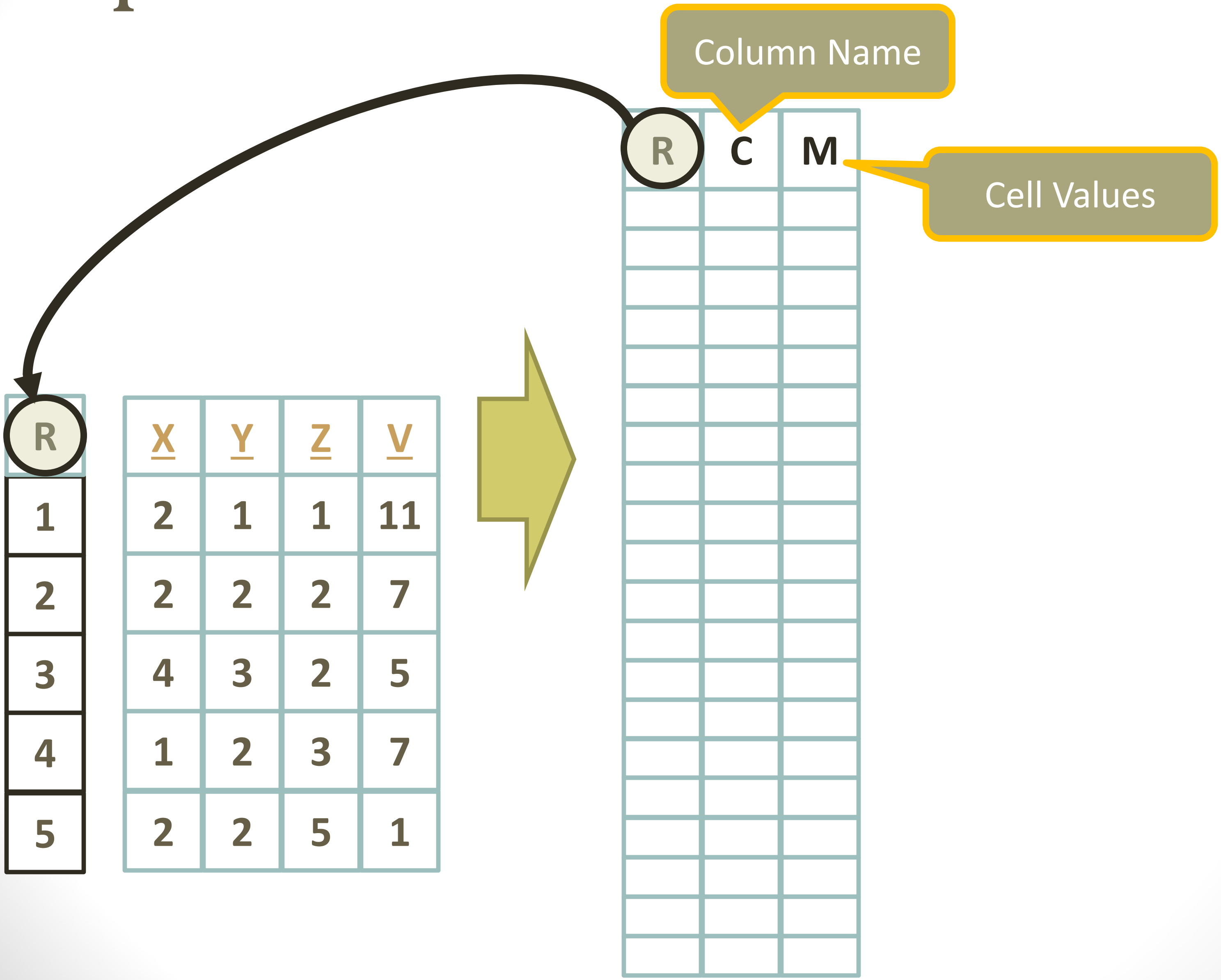
Column Name

Cell Values

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1

R	C	M

Sparse Matrices: EAV



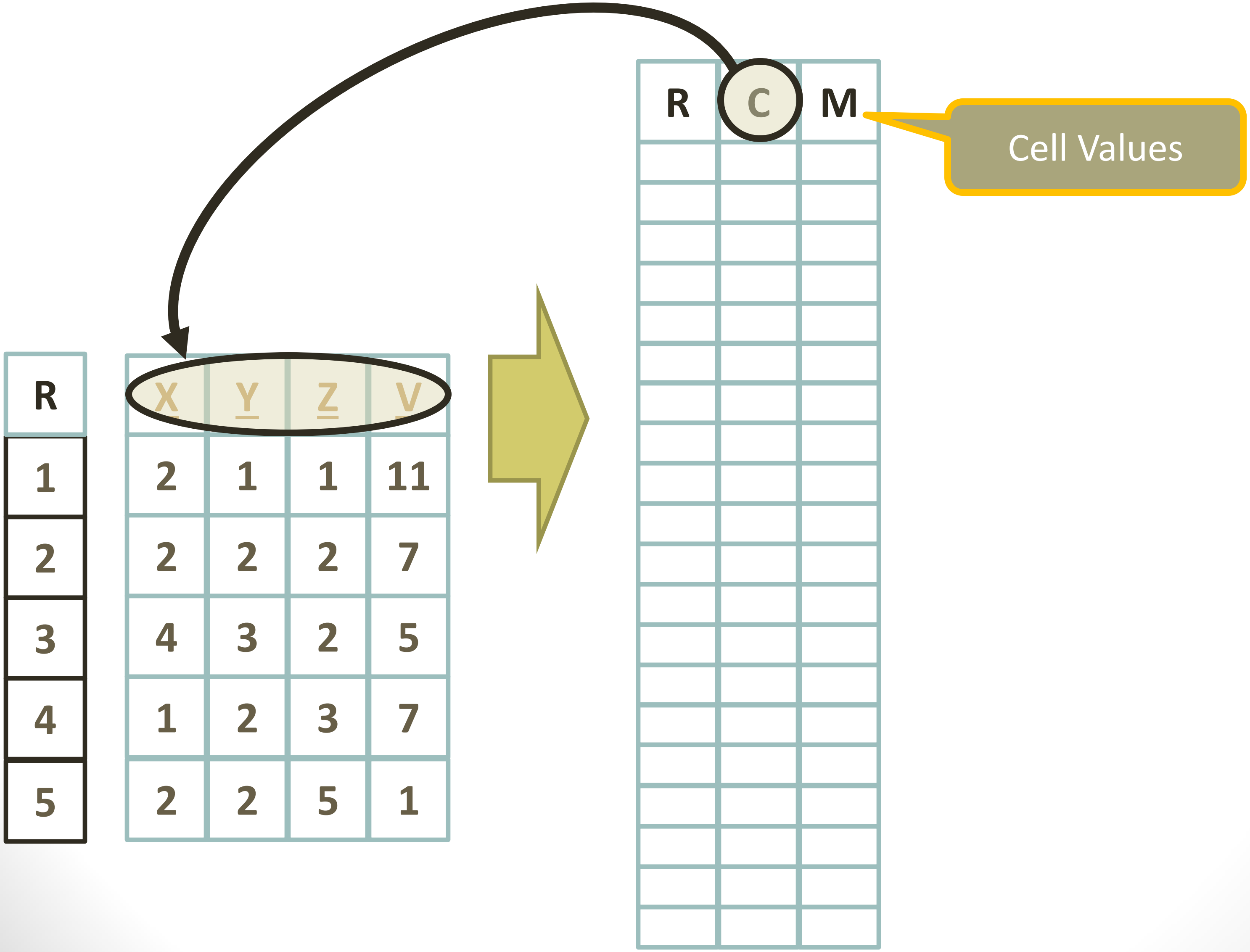
R	X	Y	Z	V
1	2	1	1	11
2	2	2	2	7
3	4	3	2	5
4	1	2	3	7
5	2	2	5	1

Column Name

Cell Values

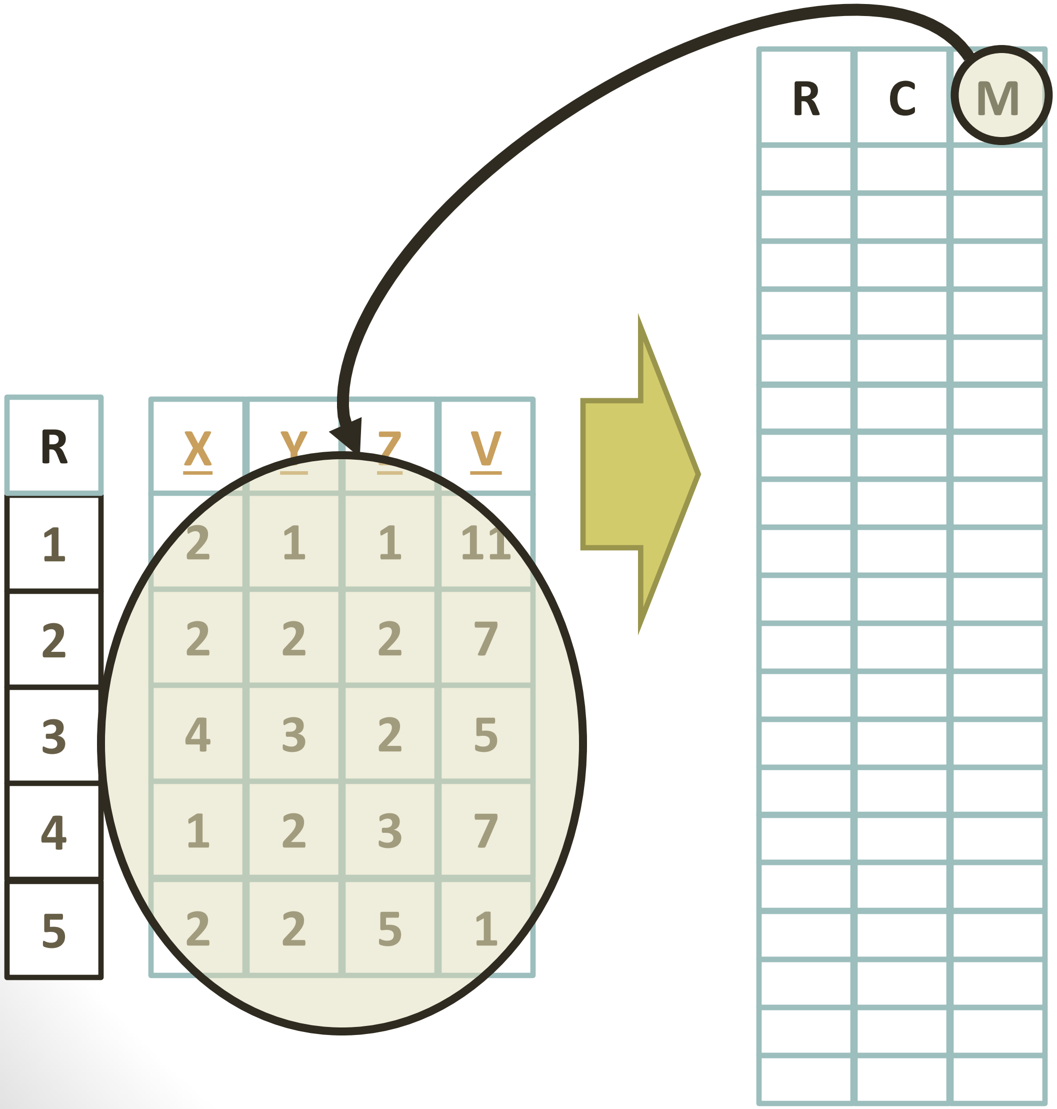
R	C	M

Sparse Matrices: EAV



R	X	Y	Z	V
1	2	1	1	11
2	2	2	2	7
3	4	3	2	5
4	1	2	3	7
5	2	2	5	1

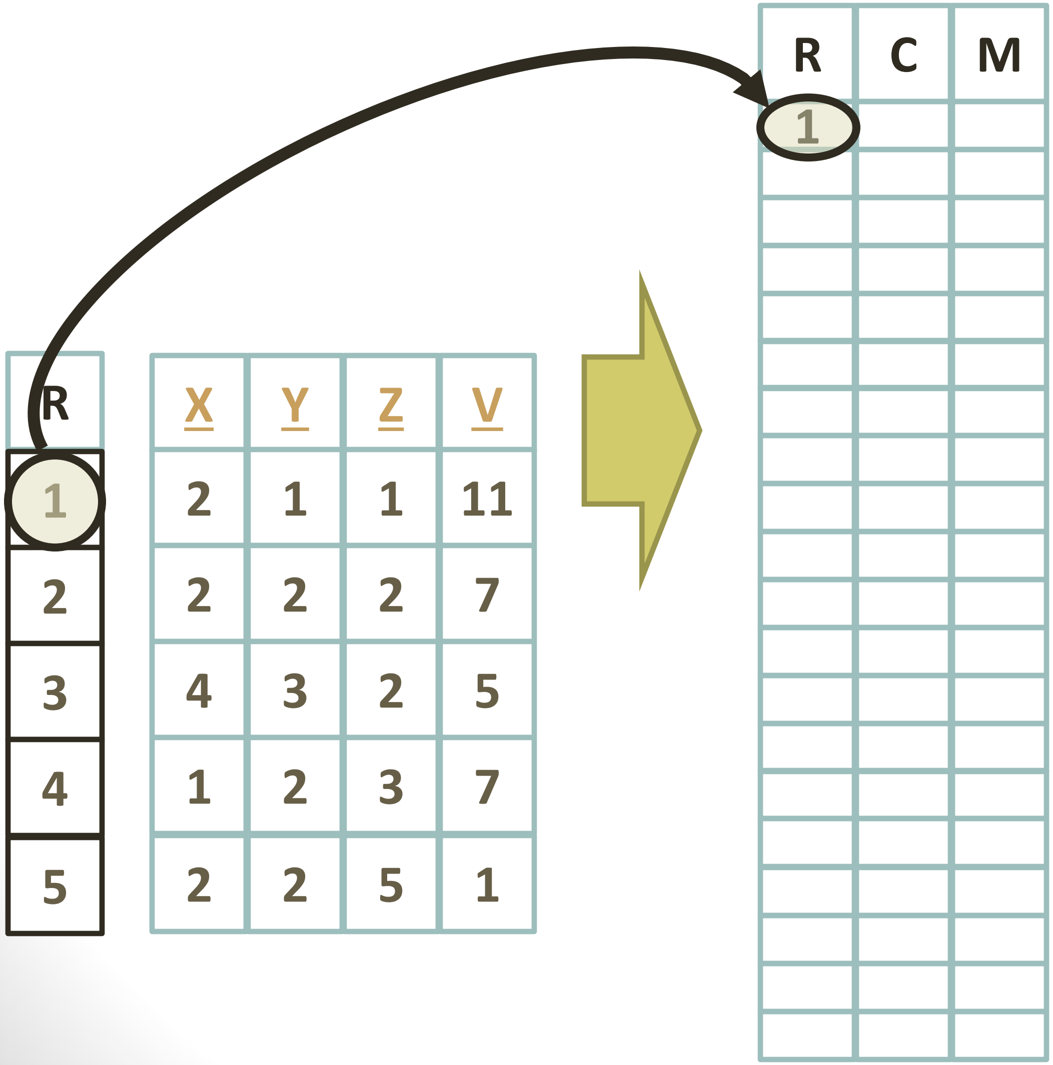
Sparse Matrices: EAV



R	X	Y	Z	V
1	2	1	1	11
2	2	2	2	7
3	4	3	2	5
4	1	2	3	7
5	2	2	5	1

R	C	M

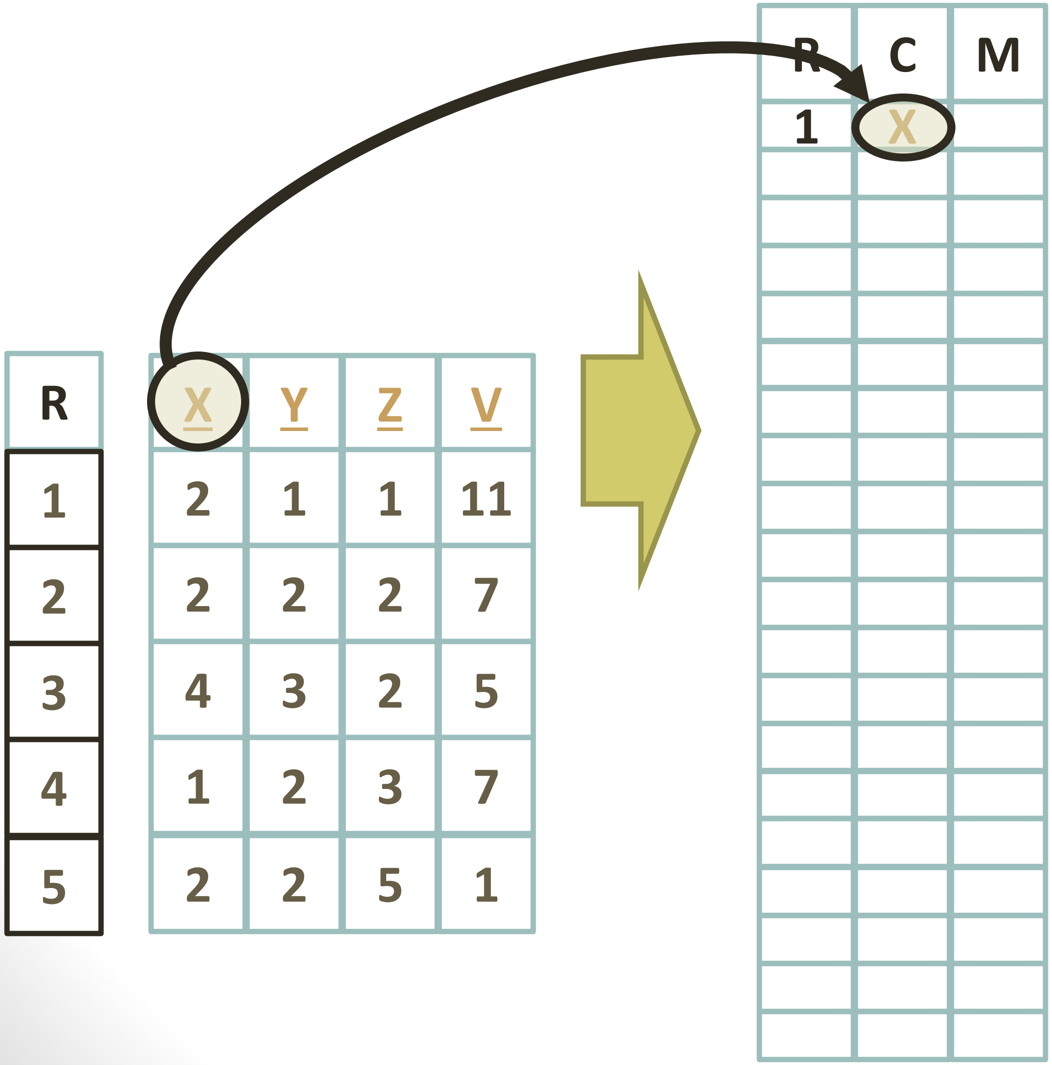
Sparse Matrices: EAV



R	X	Y	Z	V
1	2	1	1	11
2	2	2	2	7
3	4	3	2	5
4	1	2	3	7
5	2	2	5	1

R	C	M
1		

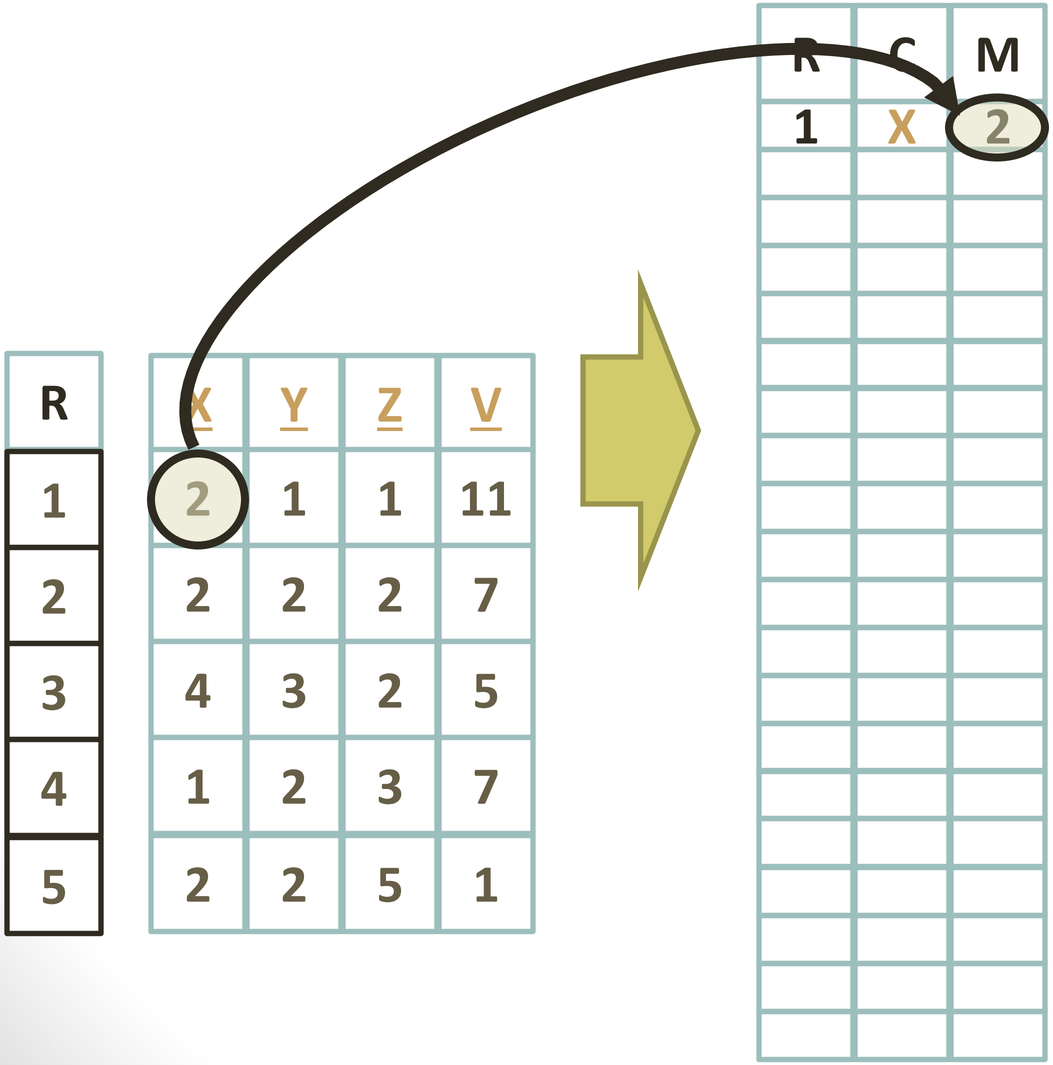
Sparse Matrices: EAV



R	X	Y	Z	V
1	2	1	1	11
2	2	2	2	7
3	4	3	2	5
4	1	2	3	7
5	2	2	5	1

R	C	M
1	X	

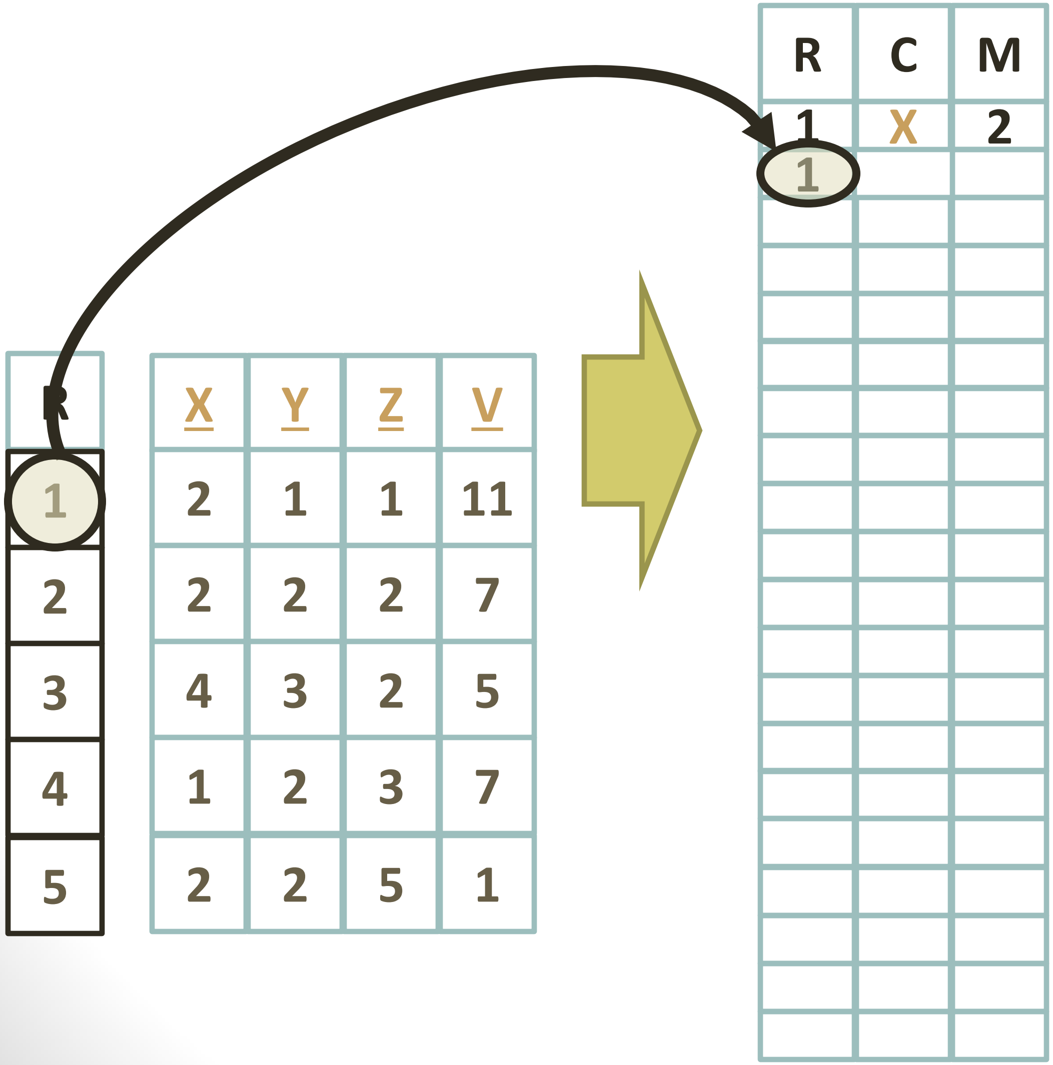
Sparse Matrices: EAV



R	X	Y	Z	V
1	2	1	1	11
2	2	2	2	7
3	4	3	2	5
4	1	2	3	7
5	2	2	5	1

R	C	M
1	X	2

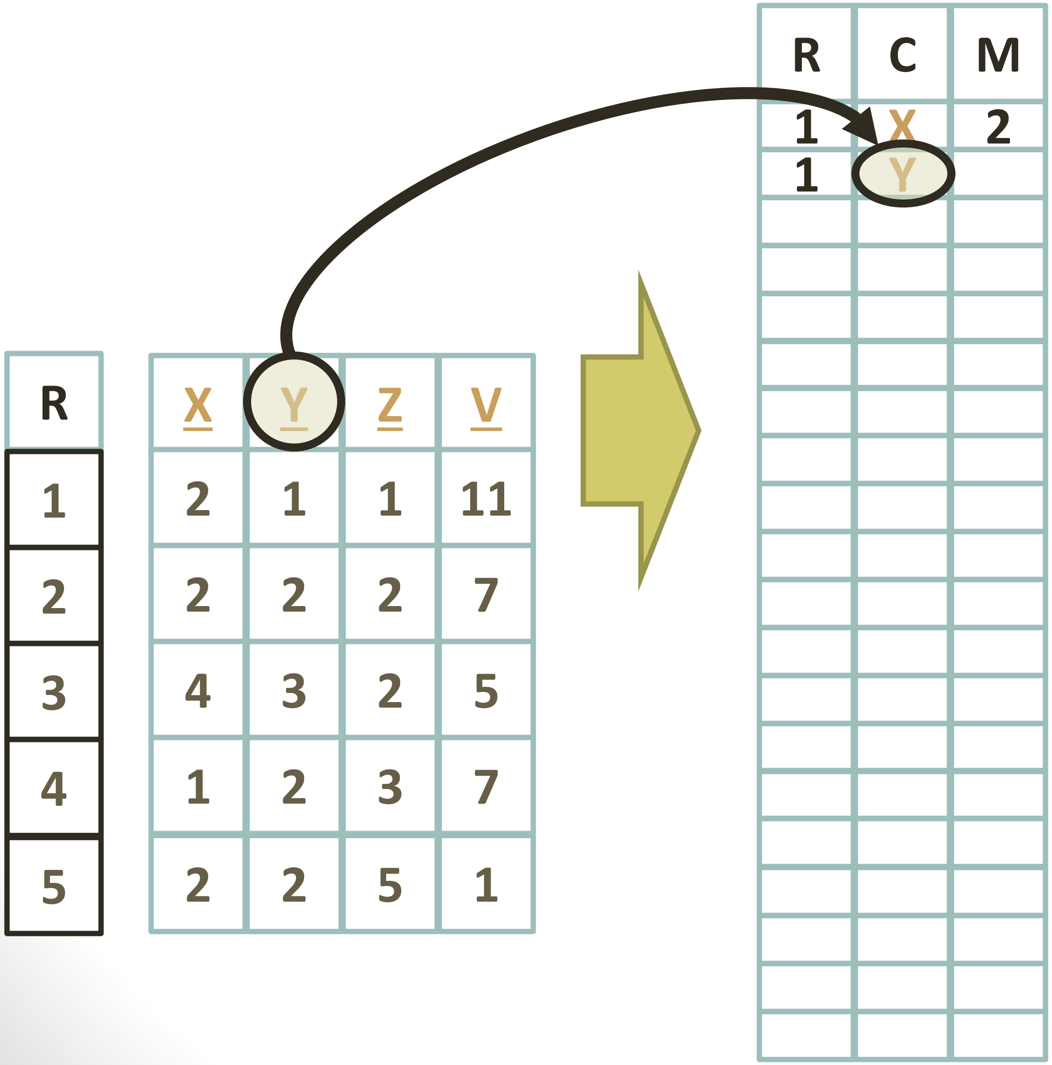
Sparse Matrices: EAV



R	X	Y	Z	V
1	2	1	1	11
2	2	2	2	7
3	4	3	2	5
4	1	2	3	7
5	2	2	5	1

R	C	M
1	X	2
1		

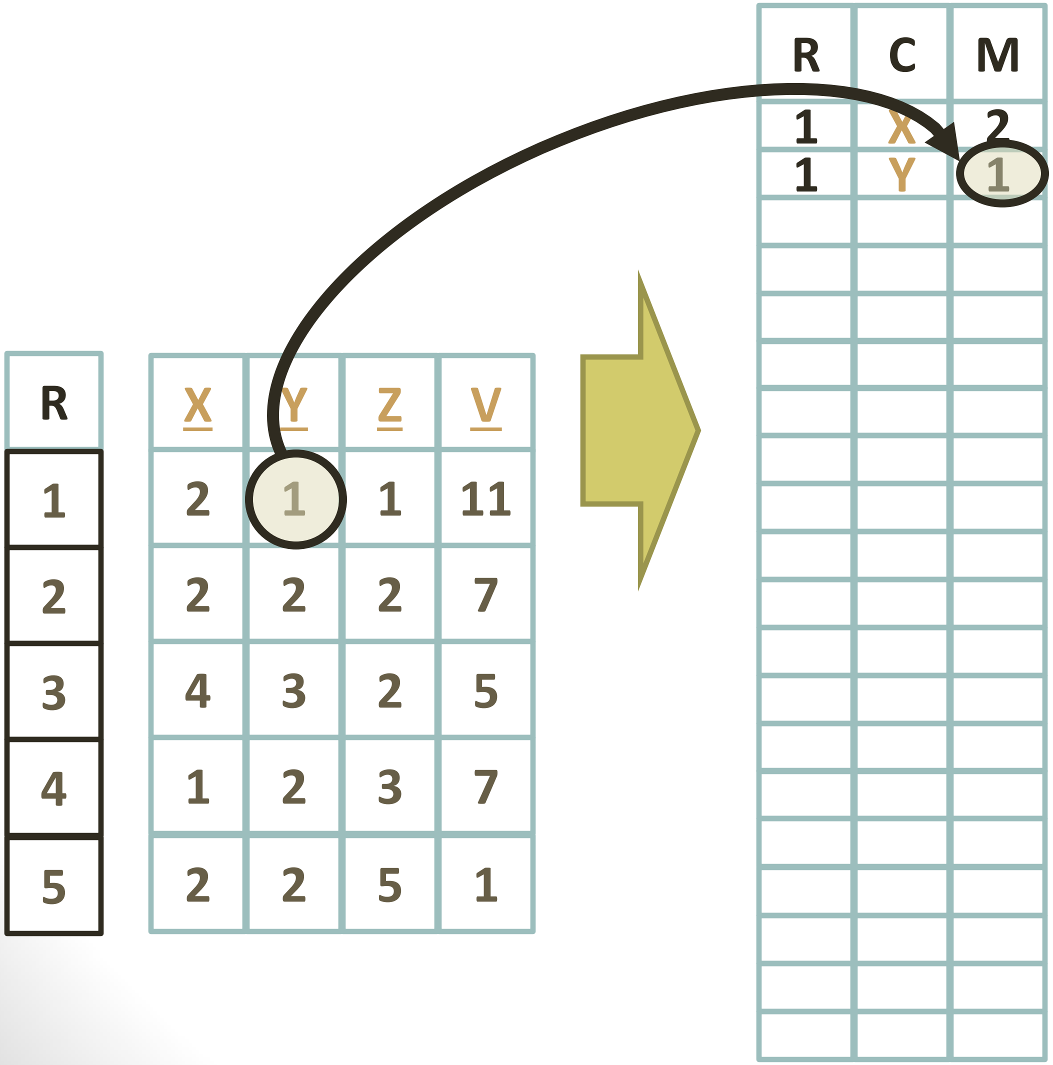
Sparse Matrices: EAV



R	X	Y	Z	V
1	2	1	1	11
2	2	2	2	7
3	4	3	2	5
4	1	2	3	7
5	2	2	5	1

R	C	M
1	X	2
1	Y	

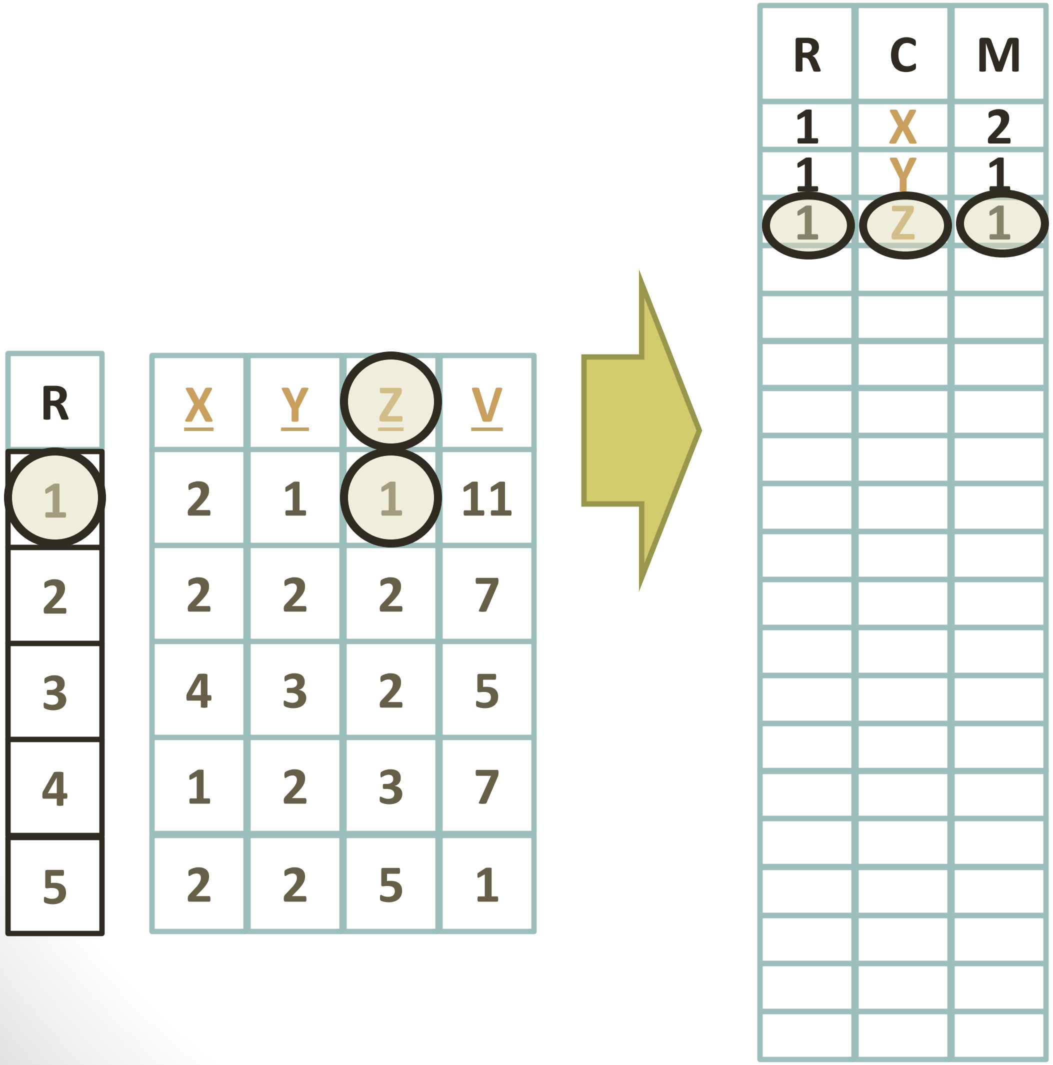
Sparse Matrices: EAV



R	X	Y	Z	V
1	2	1	1	11
2	2	2	2	7
3	4	3	2	5
4	1	2	3	7
5	2	2	5	1

R	C	M
1	X	2
1	Y	1

Sparse Matrices: EAV



R	X	Y	Z	V
1	2	1	1	11
2	2	2	2	7
3	4	3	2	5
4	1	2	3	7
5	2	2	5	1

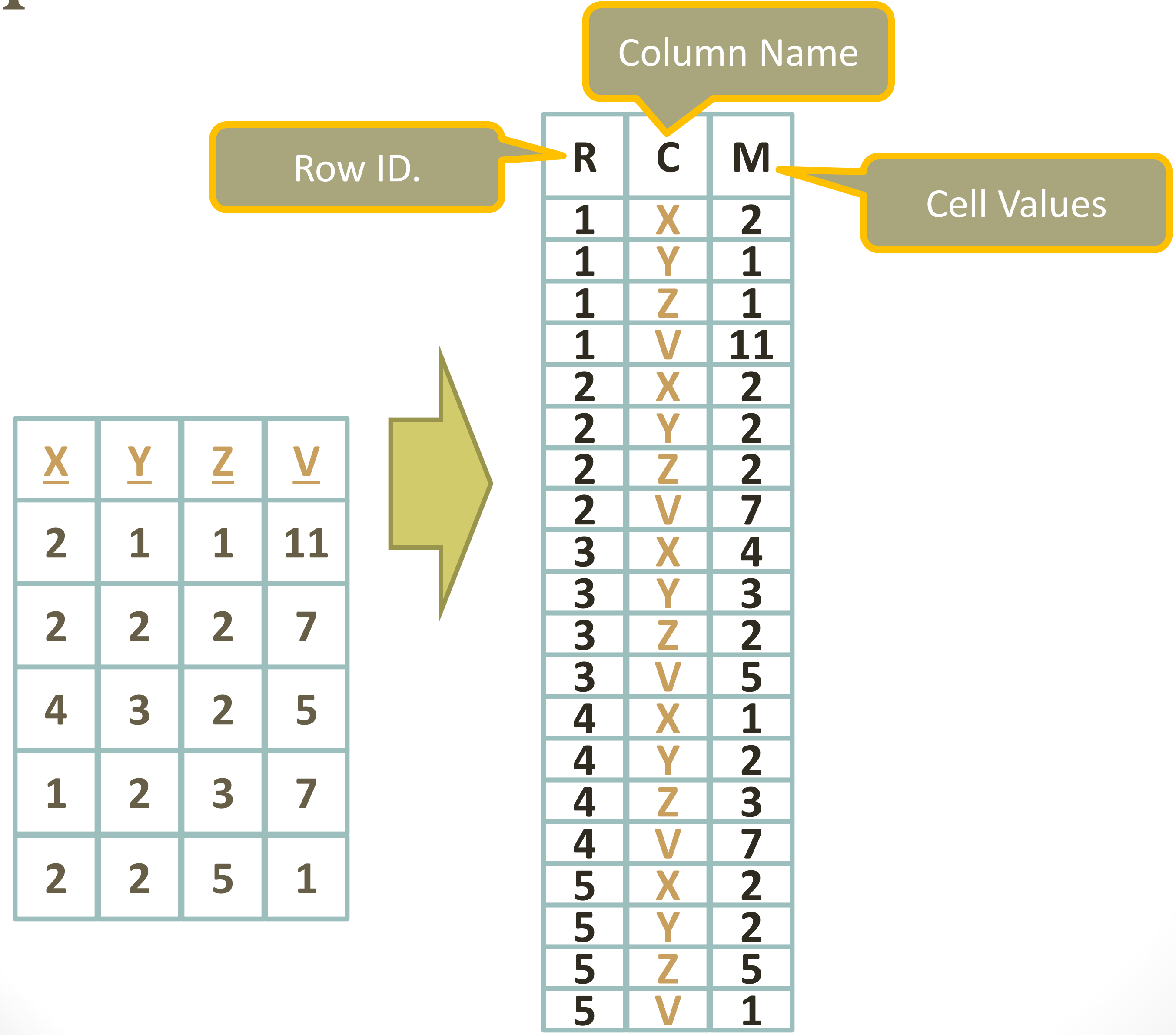
R	C	M
1	X	2
1	Y	1
1	Z	1

Sparse Matrices: EAV

X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1

R	C	M
1	X	2
1	Y	1
1	Z	1
1	V	11
2	X	2
2	Y	2
2	Z	2
2	V	7
3	X	4
3	Y	3
3	Z	2
3	V	5
4	X	1
4	Y	2
4	Z	3
4	V	7
5	X	2
5	Y	2
5	Z	5
5	V	1

Sparse Matrices: EAV



X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1

Row ID.

Column Name

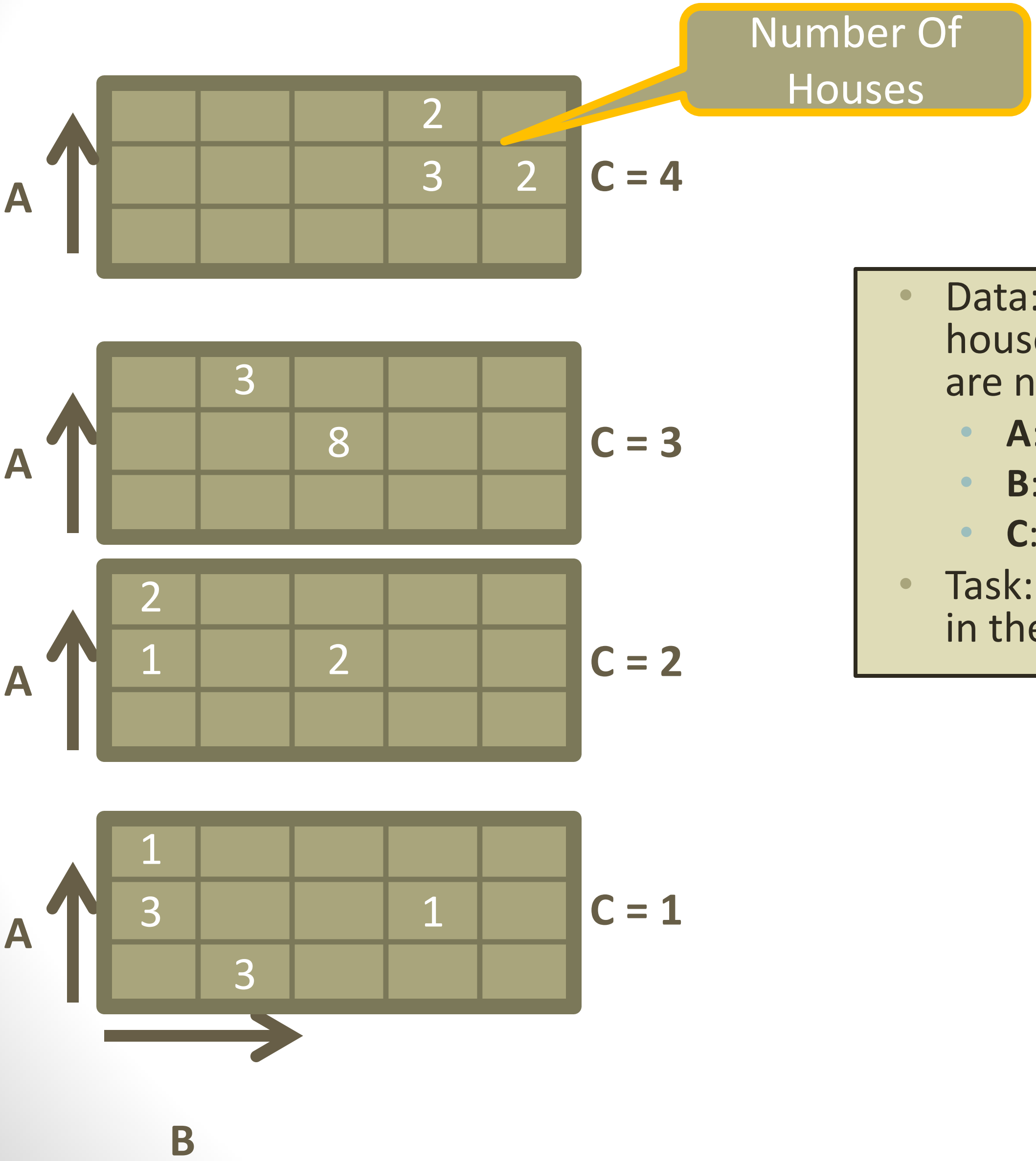
Cell Values

R	C	M
1	X	2
1	Y	1
1	Z	1
1	V	11
2	X	2
2	Y	2
2	Z	2
2	V	7
3	X	4
3	Y	3
3	Z	2
3	V	5
4	X	1
4	Y	2
4	Z	3
4	V	7
5	X	2
5	Y	2
5	Z	5
5	V	1

Quiz on EAV

- The questions are presented during the quiz

Sparse Matrices: Exercise (1)



Number Of Houses

C = 4

			2	
			3	2

C = 3

	3			
		8		

C = 2

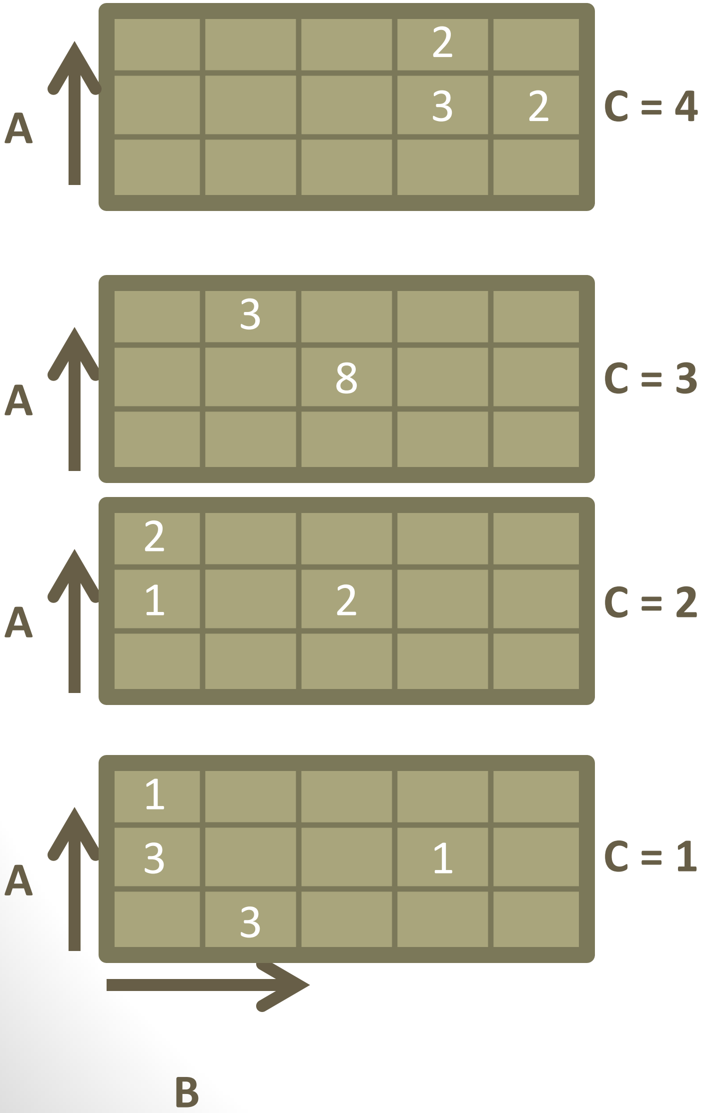
2				
1		2		

C = 1

1				
3			1	
	3			

- Data: Real estate survey of single-family houses in downtown Seattle. Cell values are number (**N**) of houses found for sale.
  - **A**: Area in 1000’s of square feet
  - **B**: Number of Bathrooms
  - **C**: Cost in $100,000.-
- Task: Create sparse matrices of the type in the previous slide.

Sparse Matrices: Exercise (2)



C = 4

			2	
			3	2

C = 3

	3			
		8		

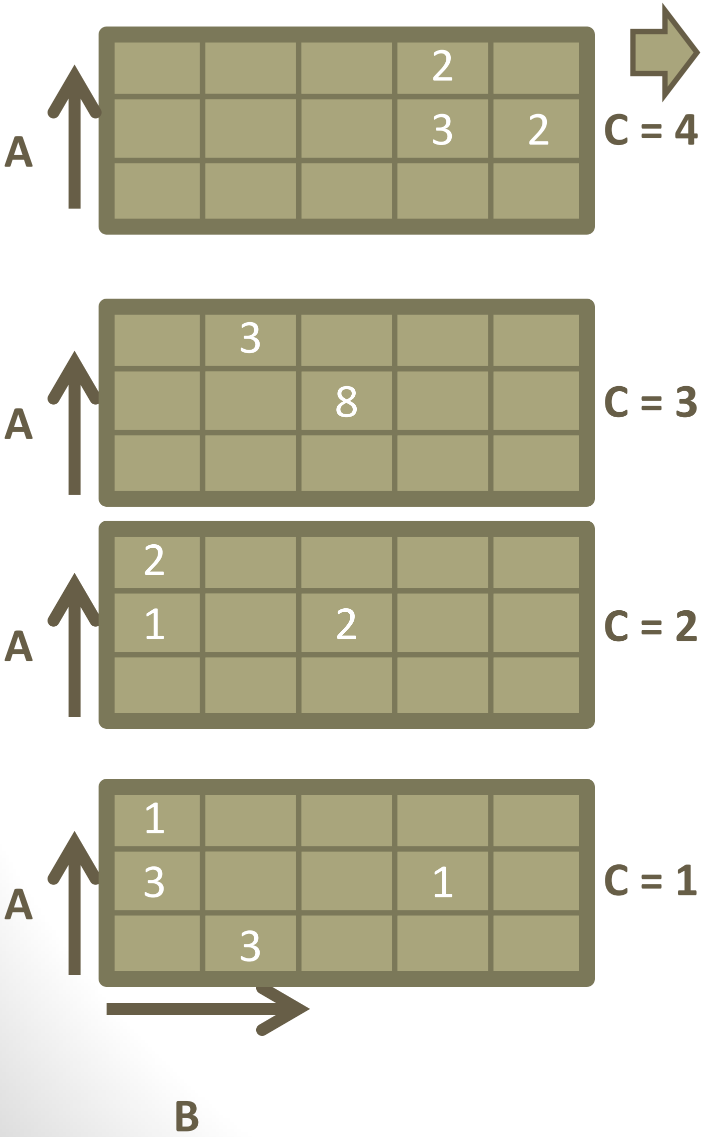
C = 2

2				
1		2		

C = 1

1				
3			1	
	3			

Sparse Matrices: Exercise (3)



C = 4

			2	
			3	2

C = 3

	3			
		8		

C = 2

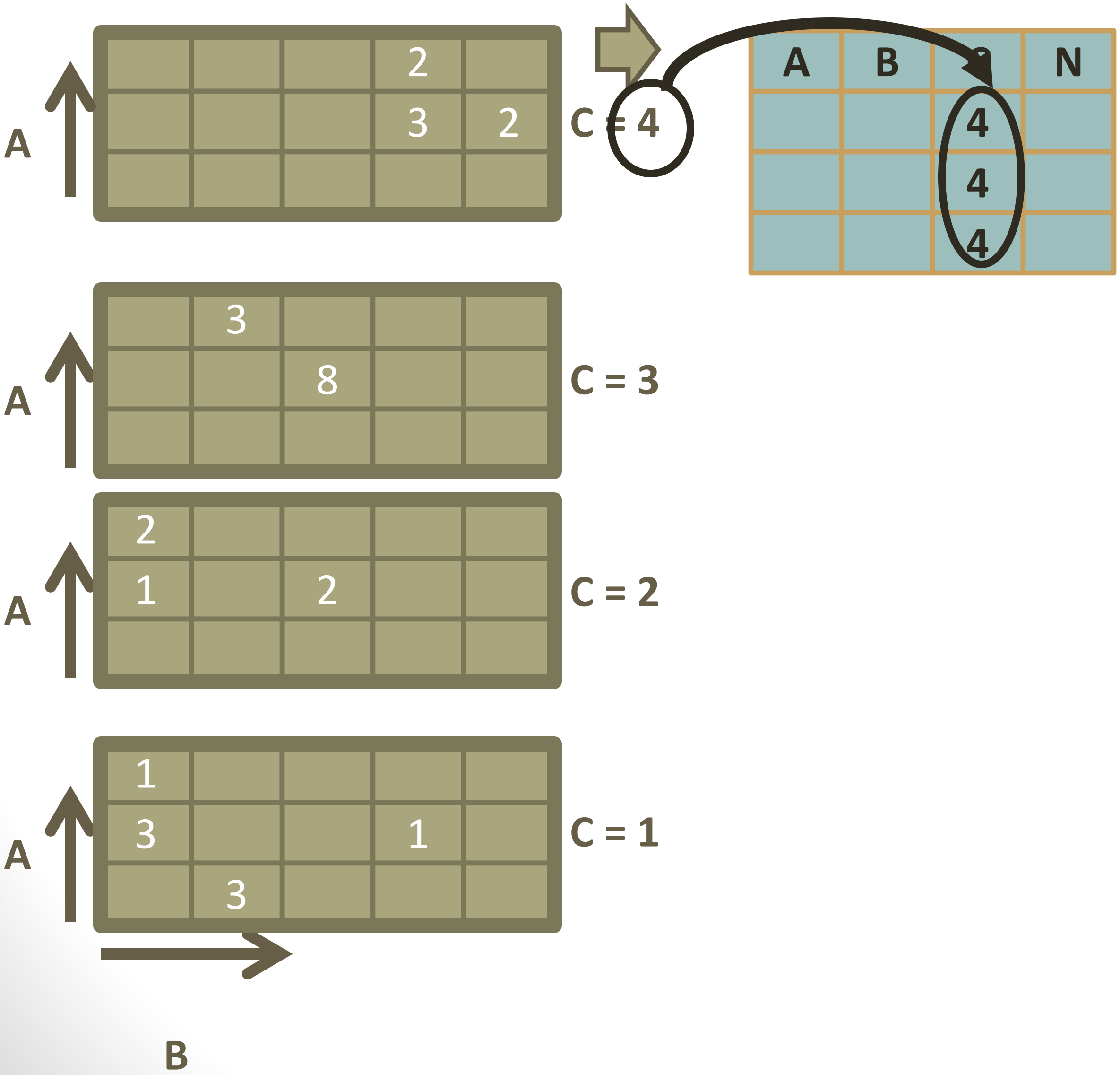
2				
1		2		

C = 1

1				
3			1	
	3			

A	B	C	N

Sparse Matrices: Exercise (4)



C = 4

			2	
			3	2

C = 3

	3			
		8		

C = 2

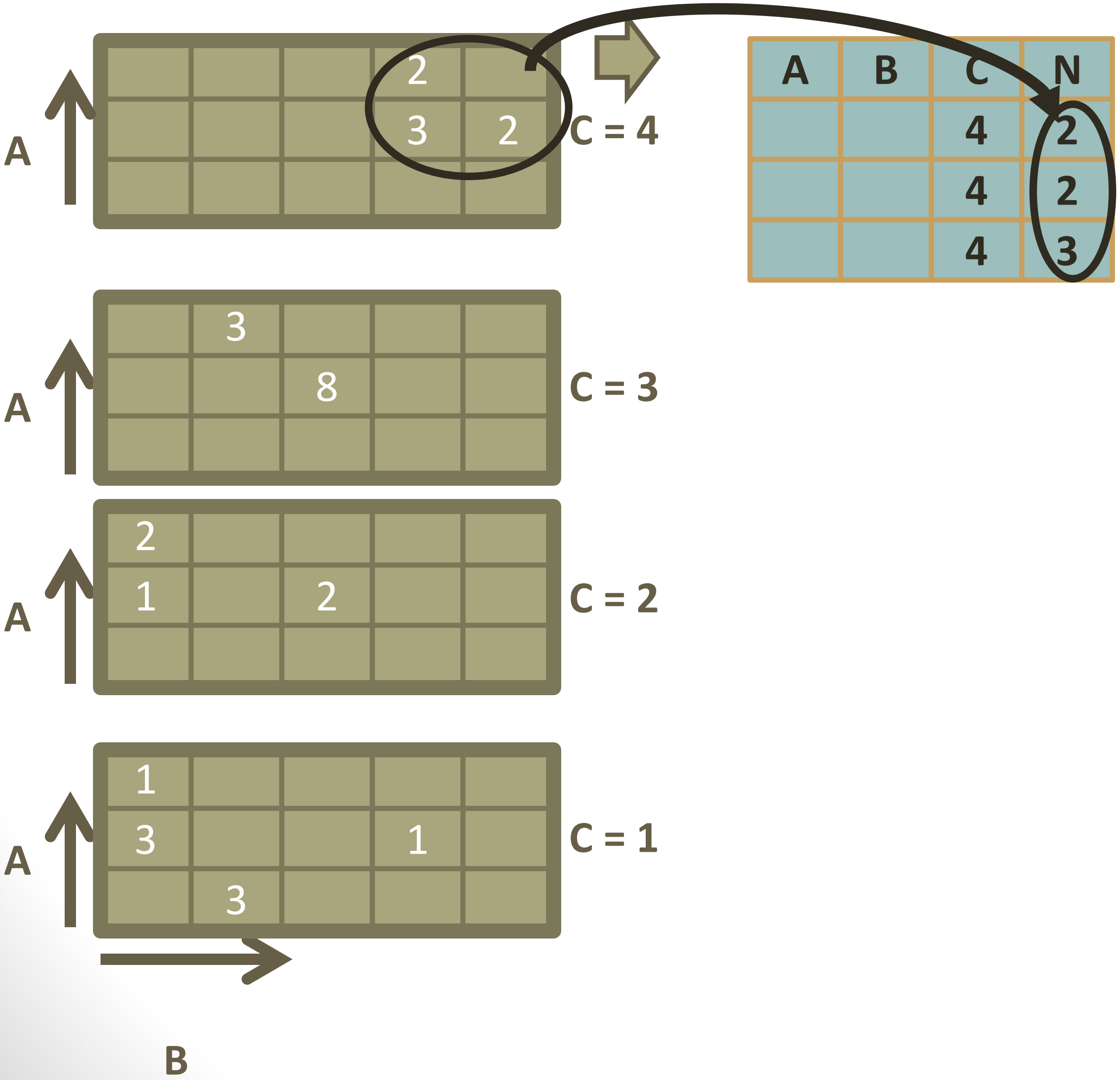
2				
1		2		

C = 1

1				
3			1	
	3			

A	B	C	N
		4	
		4	
		4	

Sparse Matrices: Exercise (5)



C = 4

			2	
			3	2

C = 3

	3			
		8		

C = 2

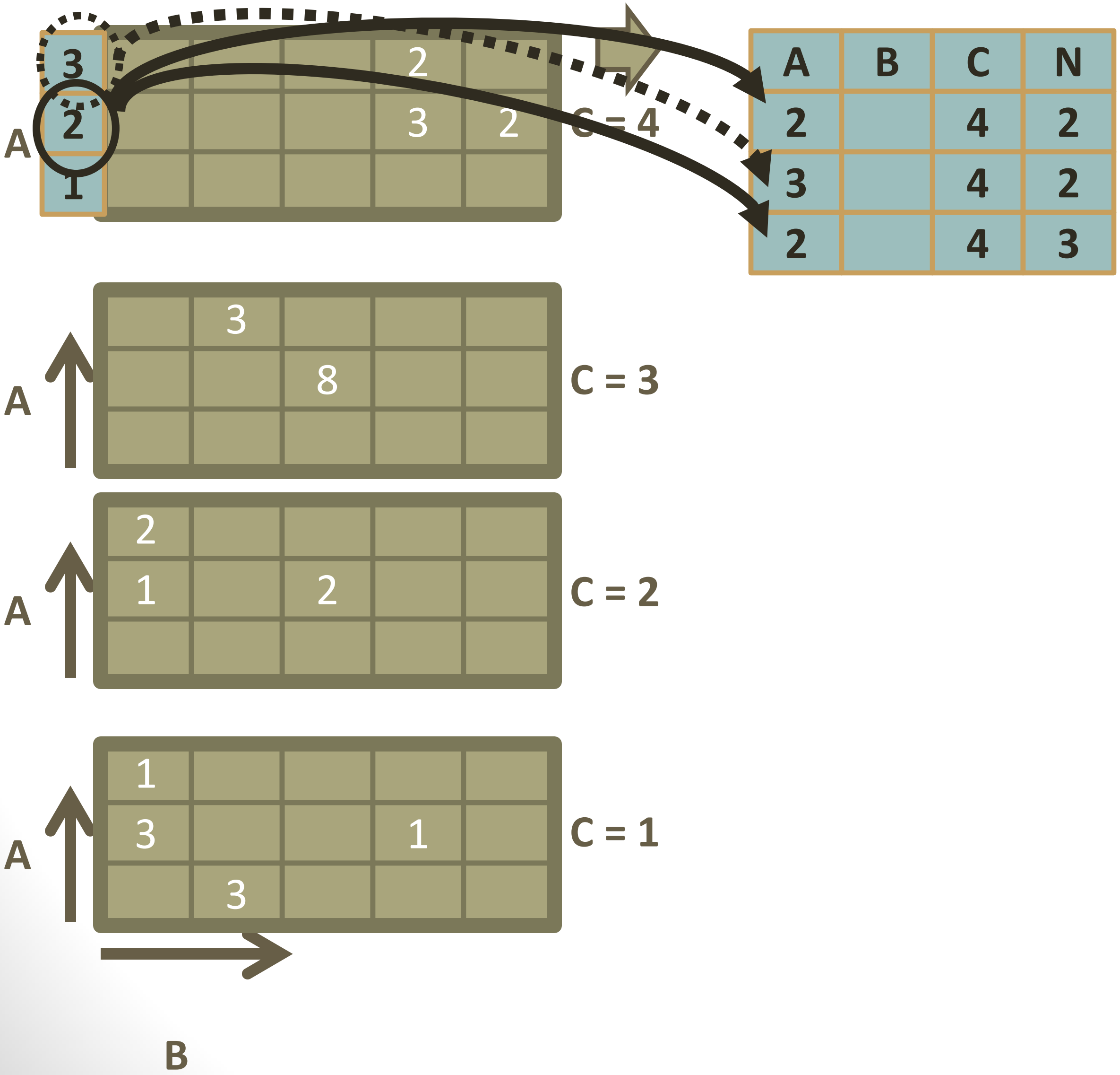
2				
1		2		

C = 1

1				
3			1	
	3			

A	B	C	N
		4	2
		4	2
		4	3

Sparse Matrices: Exercise (6)



3				2	
2				3	2
1					

C = 4

A	B	C	N
2		4	2
3		4	2
2		4	3

	3			
		8		

C = 3

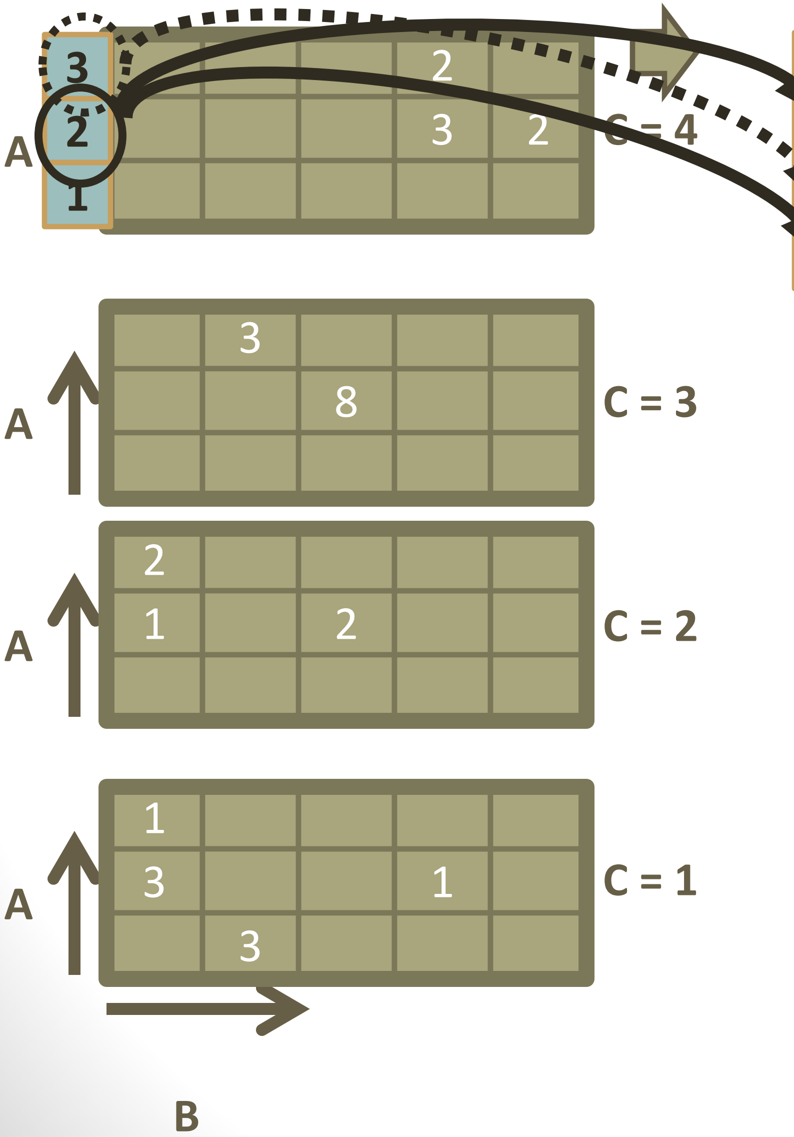
2				
1		2		

C = 2

1				
3			1	
	3			

C = 1

Sparse Matrices: Exercise (7)



A					
3				2	
2				3	2
1					

C = 4

	3			
		8		

C = 3

2				
1		2		

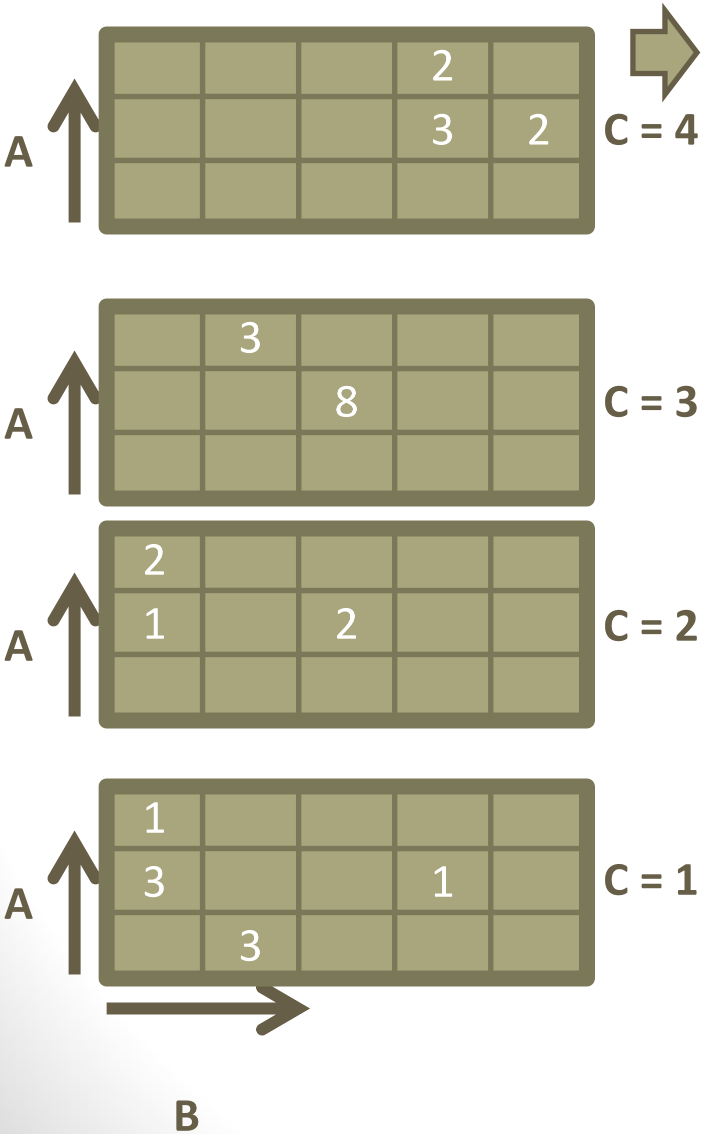
C = 2

1				
3			1	
	3			

C = 1

A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3

Sparse Matrices: Exercise (8)



C = 4

			2	
			3	2

C = 3

	3			
		8		

C = 2

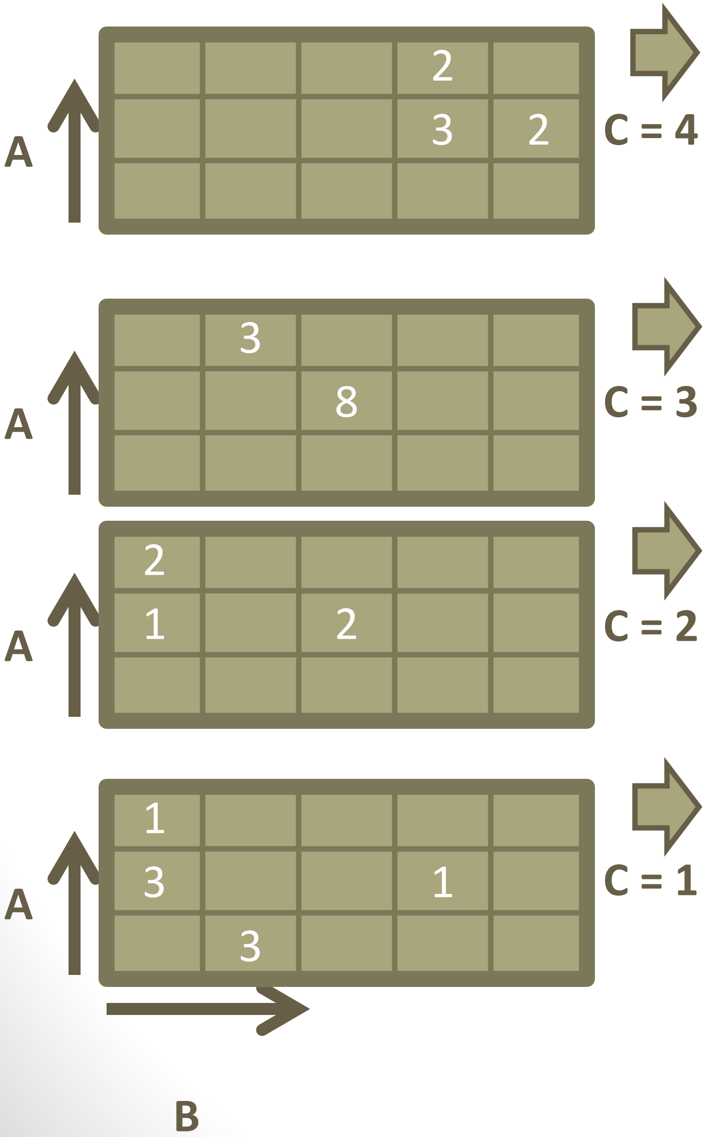
2				
1		2		

C = 1

1				
3			1	
	3			

A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3

Sparse Matrices: Exercise (9)



C = 4

			2	
			3	2

C = 3

	3			
		8		

C = 2

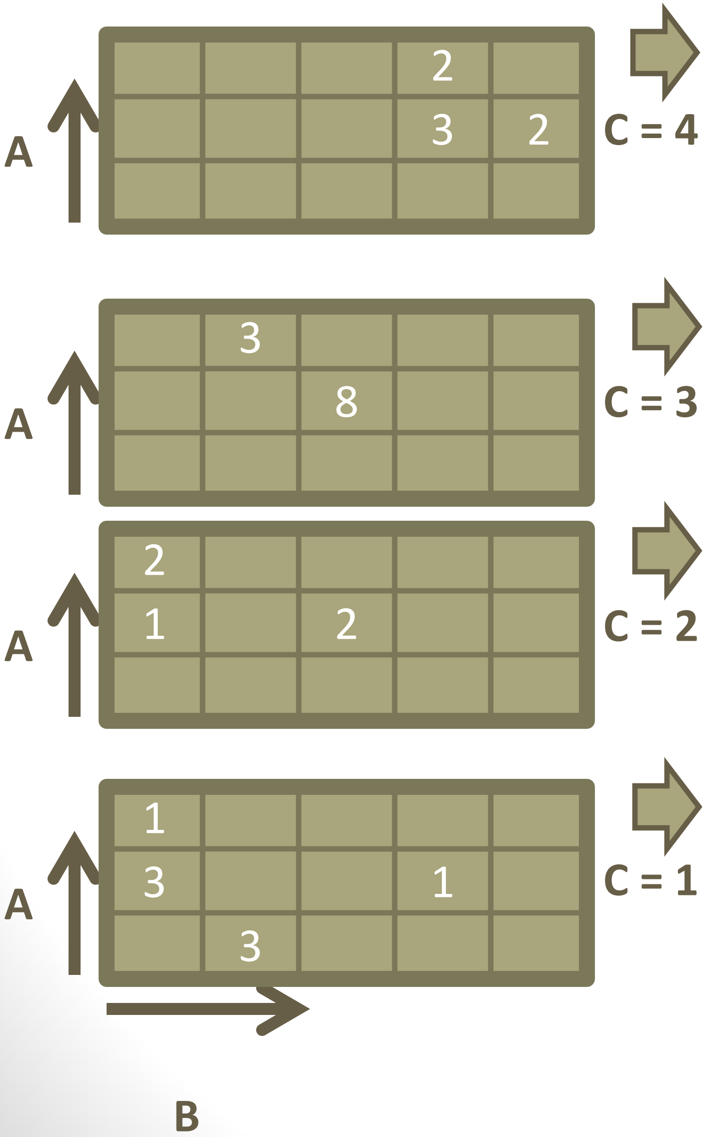
2				
1		2		

C = 1

1				
3			1	
	3			

A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3

Sparse Matrices: Exercise (10)



C = 4

			2	
			3	2

C = 3

	3			
		8		

C = 2

2				
1		2		

C = 1

1				
3			1	
	3			

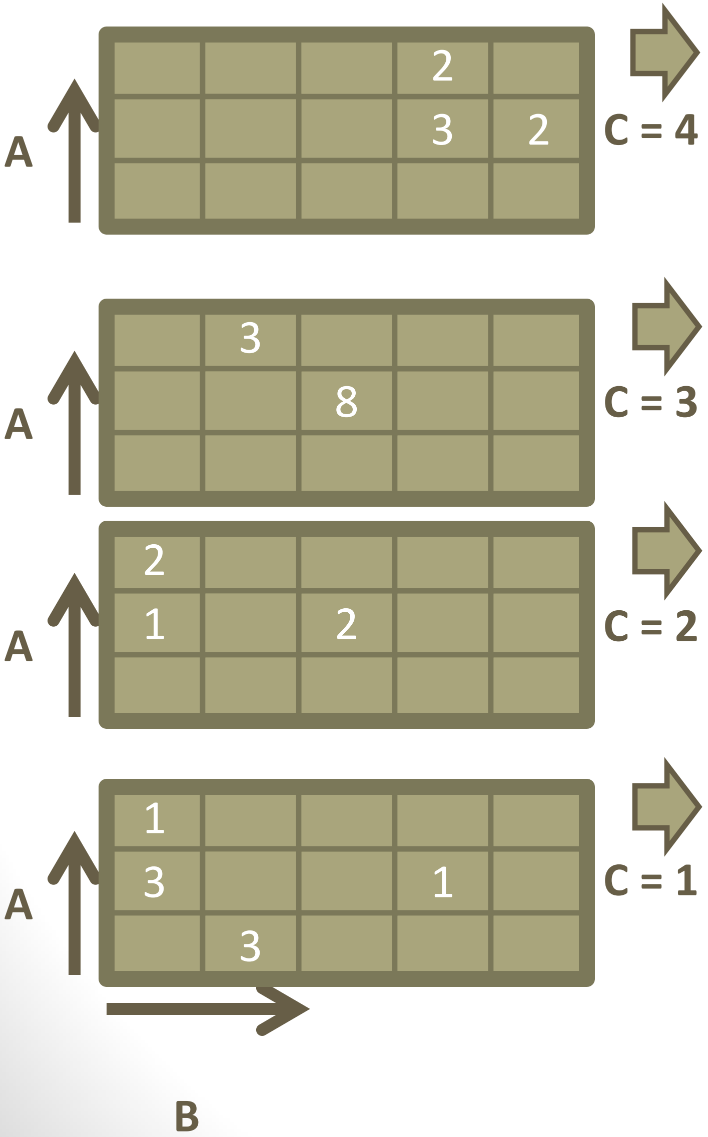
A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3

A	B	C	N
2	3	3	8
3	2	3	3

A	B	C	N
2	3	2	2
3	1	2	2
2	1	2	1

A	B	C	N
2	4	1	1
1	2	1	3
3	1	1	1
2	1	1	3

Sparse Matrices: Exercise (11)



			2	
			3	2

	3			
		8		

2				
1		2		

1				
3			1	
	3			

A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3

A	B	C	N
2	3	3	8
3	2	3	3

A	B	C	N
2	3	2	2
3	1	2	2
2	1	2	1

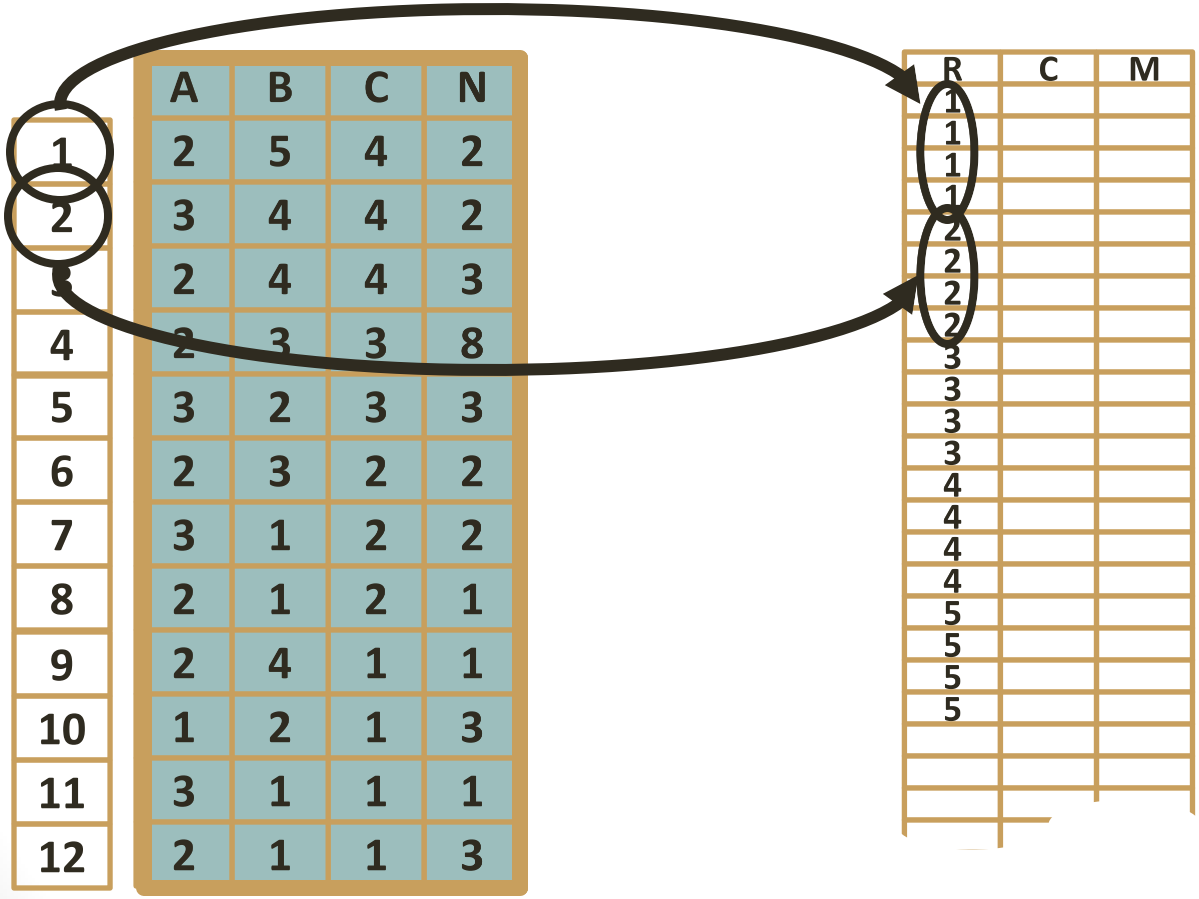
A	B	C	N
2	4	1	1
1	2	1	3
3	1	1	1
2	1	1	3

A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3
2	3	3	8
3	2	3	3
2	3	2	2
3	1	2	2
2	1	2	1
2	4	1	1
1	2	1	3
3	1	1	1
2	1	1	3

Sparse Matrices: Exercise (12)

A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3
2	3	3	8
3	2	3	3
2	3	2	2
3	1	2	2
2	1	2	1
2	4	1	1
1	2	1	3
3	1	1	1
2	1	1	3

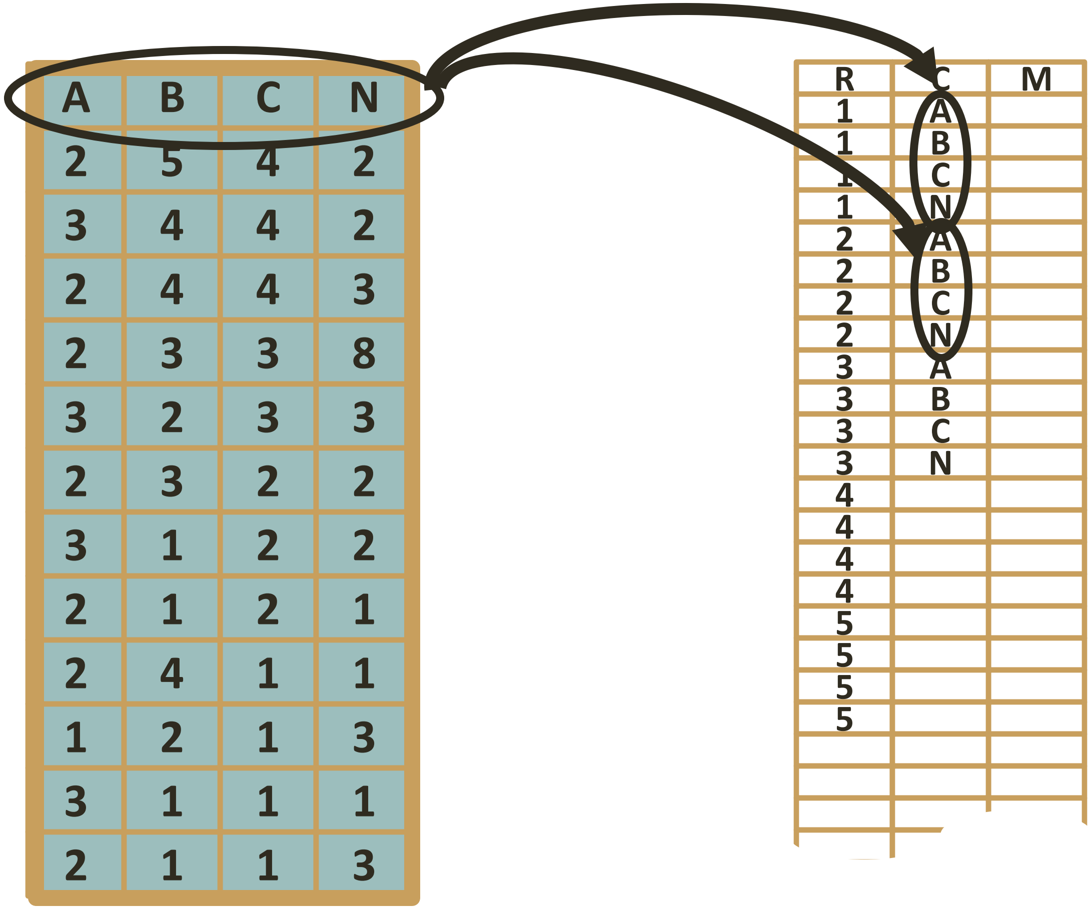
Sparse Matrices: Exercise (13)



	A	B	C	N
1	2	5	4	2
2	3	4	4	2
3	2	4	4	3
4	2	3	3	8
5	3	2	3	3
6	2	3	2	2
7	3	1	2	2
8	2	1	2	1
9	2	4	1	1
10	1	2	1	3
11	3	1	1	1
12	2	1	1	3

R	C	M
1		
1		
1		
1		
2		
2		
2		
2		
3		
3		
3		
3		
4		
4		
4		
4		
5		
5		
5		
5		

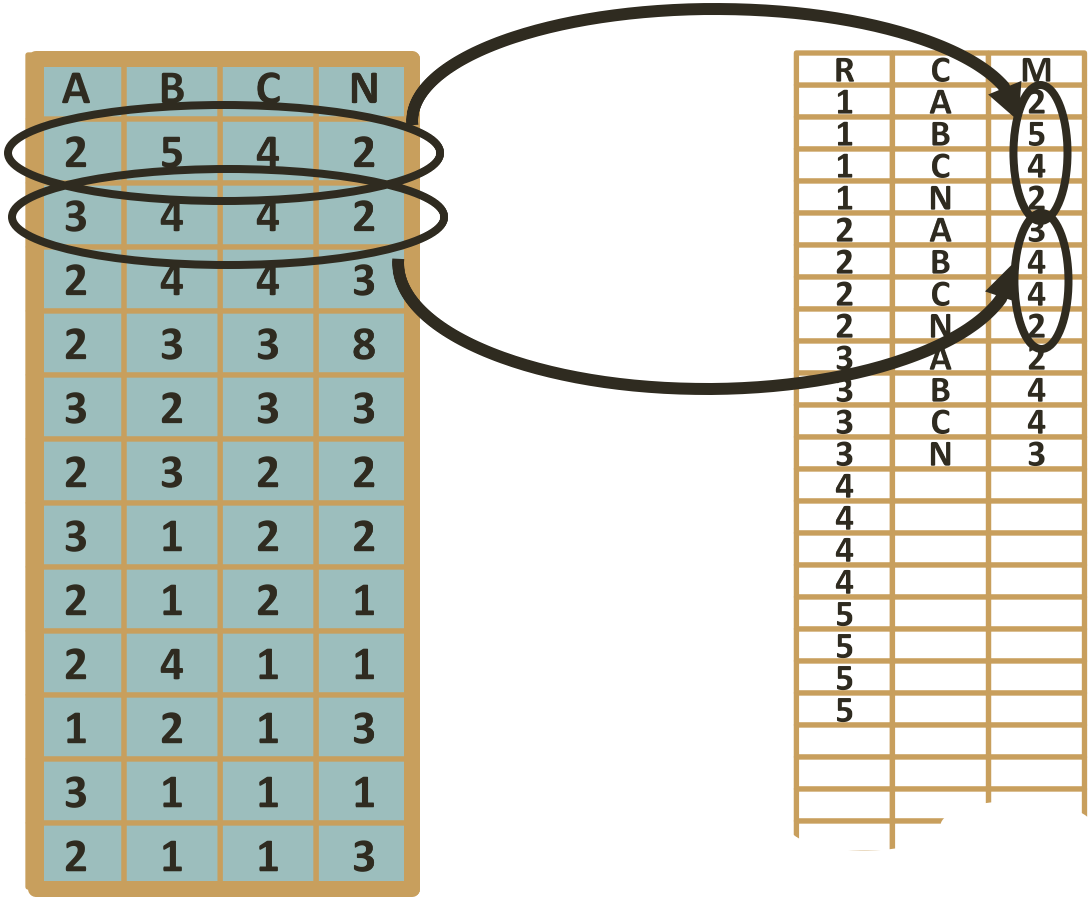
Sparse Matrices: Exercise (14)



A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3
2	3	3	8
3	2	3	3
2	3	2	2
3	1	2	2
2	1	2	1
2	4	1	1
1	2	1	3
3	1	1	1
2	1	1	3

R	C	M
1	A	
1	B	
1	C	
1	N	
2	A	
2	B	
2	C	
2	N	
3	A	
3	B	
3	C	
3	N	
4		
4		
4		
4		
5		
5		
5		
5		

Sparse Matrices: Exercise (15)



A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3
2	3	3	8
3	2	3	3
2	3	2	2
3	1	2	2
2	1	2	1
2	4	1	1
1	2	1	3
3	1	1	1
2	1	1	3

R	C	M
1	A	2
1	B	5
1	C	4
1	N	2
2	A	3
2	B	4
2	C	4
2	N	2
3	A	2
3	B	4
3	C	4
3	N	3
4		
4		
4		
4		
5		
5		
5		
5		

Sparse Matrices: Exercise (16)

A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3
2	3	3	8
3	2	3	3
2	3	2	2
3	1	2	2
2	1	2	1
2	4	1	1
1	2	1	3
3	1	1	1
2	1	1	3

R	C	M
1	A	2
1	B	5
1	C	4
1	N	2
2	A	3
2	B	4
2	C	4
2	N	2
3	A	2
3	B	4
3	C	4
3	N	3
4		
4		
4		
4		
5		
5		
5		
5		

Sparse Matrices: Exercise (17)

- Main Point:
  - Condensing information from multi-dimensional entity is good but not the main point.
  - The main point is to convince you that a relation and an EAV represent multi-dimensional matrices (Hyper-rectangles, or Cartesian products of their intervals)
- Further Lessons:
  - These tables abide by the rules of relational algebra
    - Rows are unique
    - Columns have headers
    - Row order is irrelevant
  - Relaxed Layout / Schema
  - Extensible: New tables can be added without disrupting the schema

Schema Change: add a column

- Schema change can happen by adding rows (tuples) to a table that indexes another table

Schema Change: add a column

A	B	C
2	5	4
3	4	4
2	4	4
2	3	3
3	2	3
2	3	2

Schema Change: add a column

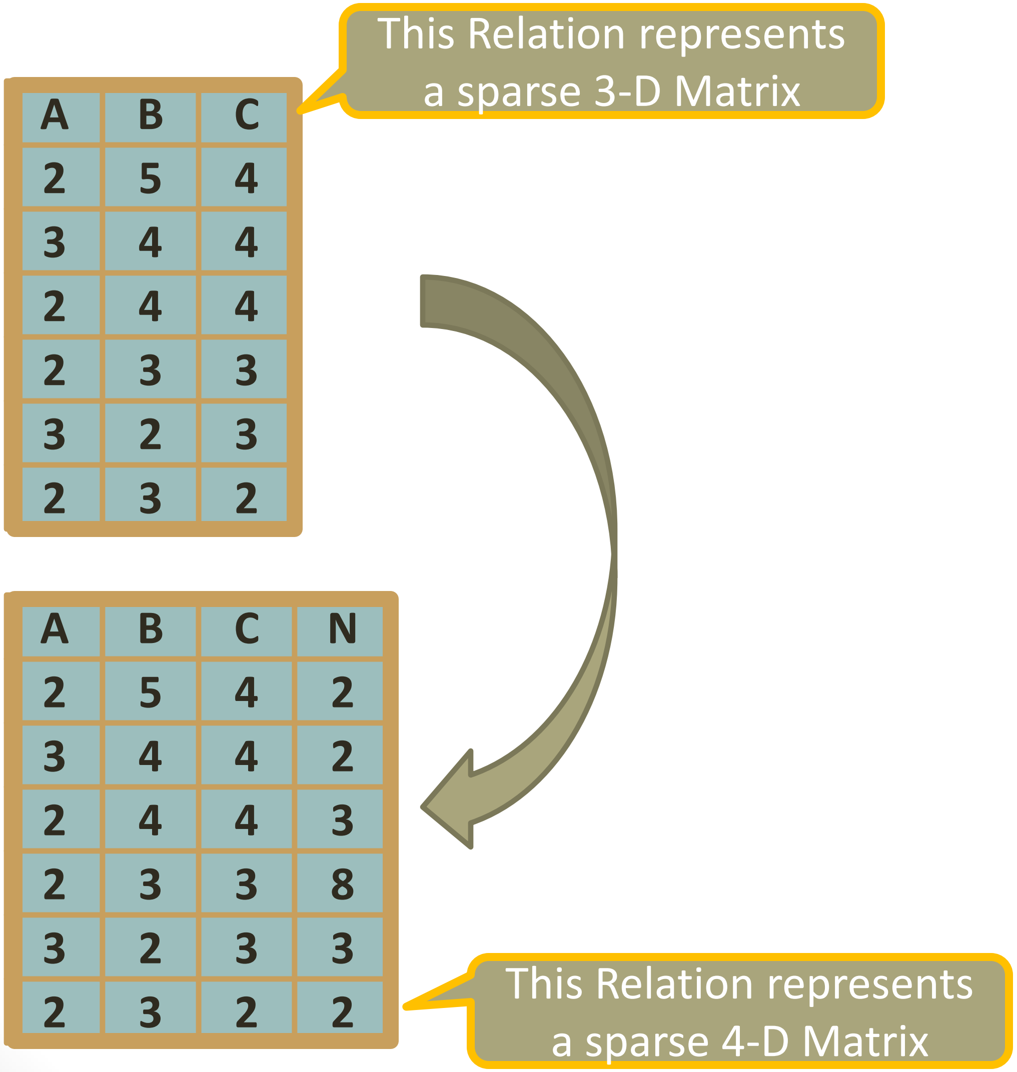
A	B	C
2	5	4
3	4	4
2	4	4
2	3	3
3	2	3
2	3	2

This Relation represents a sparse 3-D Matrix

A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3
2	3	3	8
3	2	3	3
2	3	2	2

This Relation represents a sparse 4-D Matrix

Schema Change: add a column



A	B	C
2	5	4
3	4	4
2	4	4
2	3	3
3	2	3
2	3	2

This Relation represents a sparse 3-D Matrix

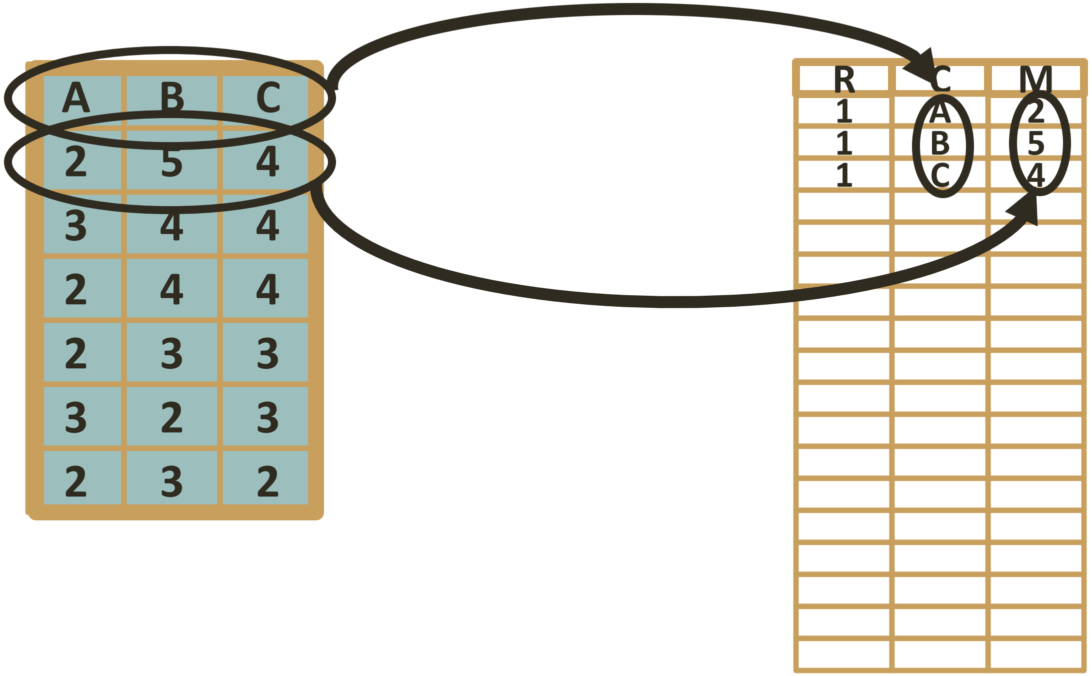
A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3
2	3	3	8
3	2	3	3
2	3	2	2

This Relation represents a sparse 4-D Matrix

Schema Change: add a column

A	B	C
2	5	4
3	4	4
2	4	4
2	3	3
3	2	3
2	3	2

Represent Relation by indexing Row, Column, and Value



A	B	C
2	5	4
3	4	4
2	4	4
2	3	3
3	2	3
2	3	2

R	C	M
1	A	2
1	B	5
1	C	4

Represent Relation by indexing Row, Column, and Value

A	B	C
2	5	4
3	4	4
2	4	4
2	3	3
3	2	3
2	3	2

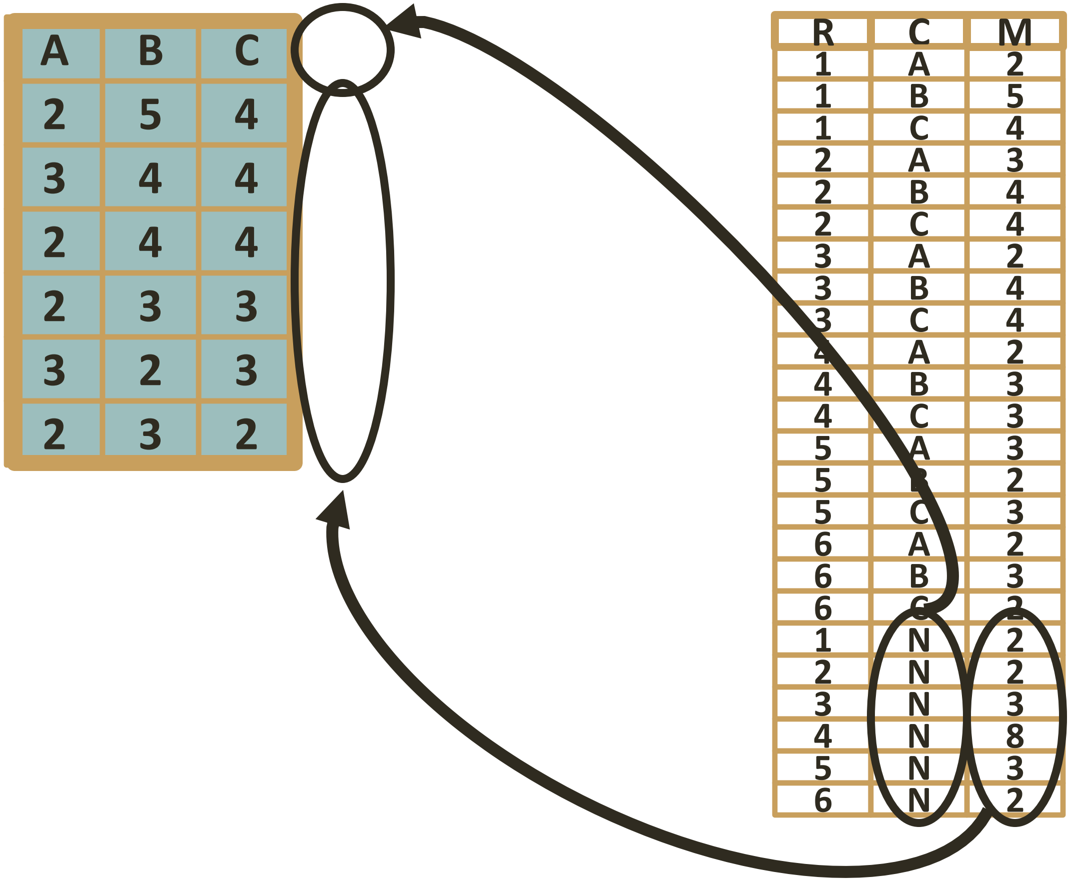
R	C	M
1	A	2
1	B	5
1	C	4
2	A	3
2	B	4
2	C	4
3	A	2
3	B	4
3	C	4
4	A	2
4	B	3
4	C	3
5	A	3
5	B	2
5	C	3
6	A	2
6	B	3
6	C	2

Adding new rows to second table with a new index

A	B	C
2	5	4
3	4	4
2	4	4
2	3	3
3	2	3
2	3	2

R	C	M
1	A	2
1	B	5
1	C	4
2	A	3
2	B	4
2	C	4
3	A	2
3	B	4
3	C	4
4	A	2
4	B	3
4	C	3
5	A	3
5	B	2
5	C	3
6	A	2
6	B	3
6	C	2
1	N	2
2	N	2
3	N	3
4	N	8
5	N	3
6	N	2

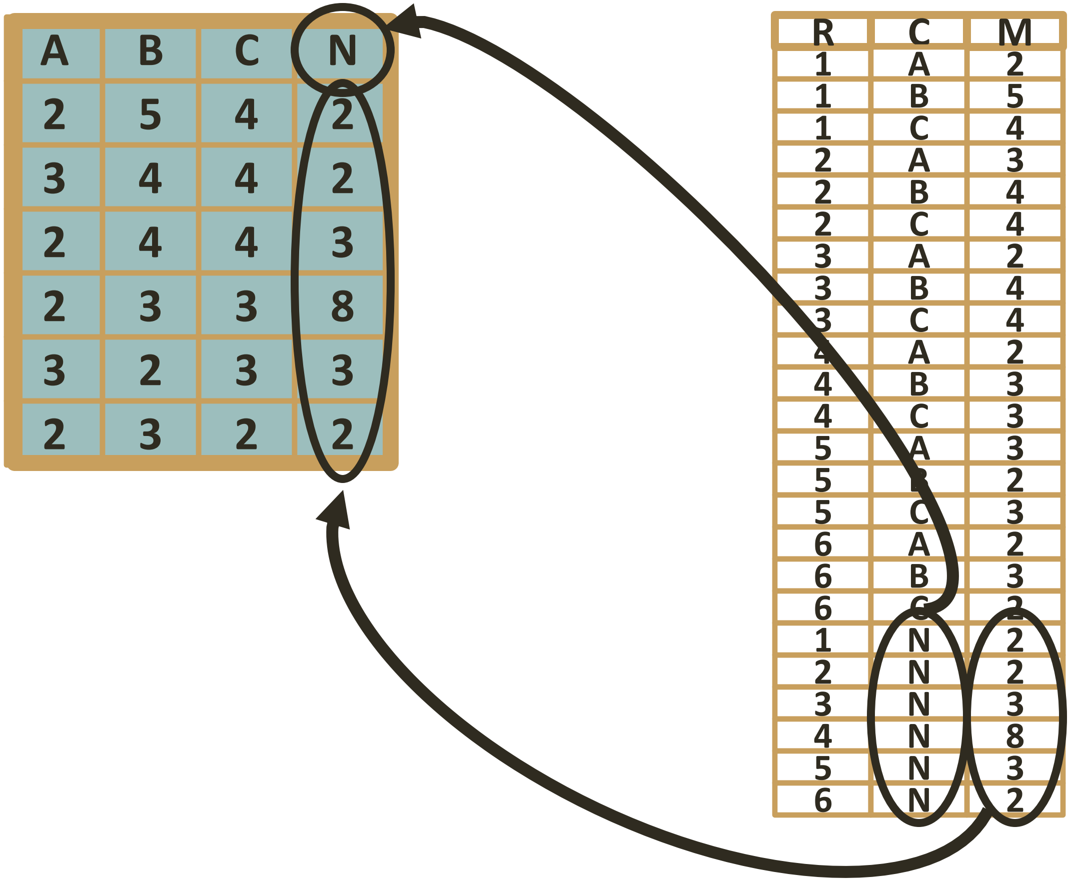
Adding new rows to second table with a new index



A	B	C
2	5	4
3	4	4
2	4	4
2	3	3
3	2	3
2	3	2

R	C	M
1	A	2
1	B	5
1	C	4
2	A	3
2	B	4
2	C	4
3	A	2
3	B	4
3	C	4
4	A	2
4	B	3
4	C	3
5	A	3
5	B	2
5	C	3
6	A	2
6	B	3
6	C	2
1	N	2
2	N	2
3	N	3
4	N	8
5	N	3
6	N	2

Adding new rows to second table with a new index



A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3
2	3	3	8
3	2	3	3
2	3	2	2

R	C	M
1	A	2
1	B	5
1	C	4
2	A	3
2	B	4
2	C	4
3	A	2
3	B	4
3	C	4
4	A	2
4	B	3
4	C	3
5	A	3
5	B	2
5	C	3
6	A	2
6	B	3
6	C	2
1	N	2
2	N	2
3	N	3
4	N	8
5	N	3
6	N	2

Adding new rows to second table with a new index

A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3
2	3	3	8
3	2	3	3
2	3	2	2

R	C	M
1	A	2
1	B	5
1	C	4
2	A	3
2	B	4
2	C	4
3	A	2
3	B	4
3	C	4
4	A	2
4	B	3
4	C	3
5	A	3
5	B	2
5	C	3
6	A	2
6	B	3
6	C	2
1	N	2
2	N	2
3	N	3
4	N	8
5	N	3
6	N	2

Rows may be resorted without changing the relation

A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3
2	3	3	8
3	2	3	3
2	3	2	2

R	C	M
1	A	2
1	B	5
1	C	4
2	A	3
2	B	4
2	C	4
3	A	2
3	B	4
3	C	4
4	A	2
4	B	3
4	C	3
5	A	3
5	B	2
5	C	3
6	A	2
6	B	3
6	C	2

1	N	2
2	N	2
3	N	3
4	N	8
5	N	3
6	N	2

Rows may be resorted without changing the relation

A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3
2	3	3	8
3	2	3	3
2	3	2	2

R	C	M
1	A	2
1	B	5
1	C	4
2	A	3
2	B	4
2	C	4
3	A	2
3	B	4
3	C	4
4	A	2
4	B	3
4	C	3
5	A	3
5	B	2
5	C	3
6	A	2
6	B	3
6	C	2

1	N	2
2	N	2
3	N	3
4	N	8
5	N	3
6	N	2

Rows may be resorted without changing the relation

A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3
2	3	3	8
3	2	3	3
2	3	2	2

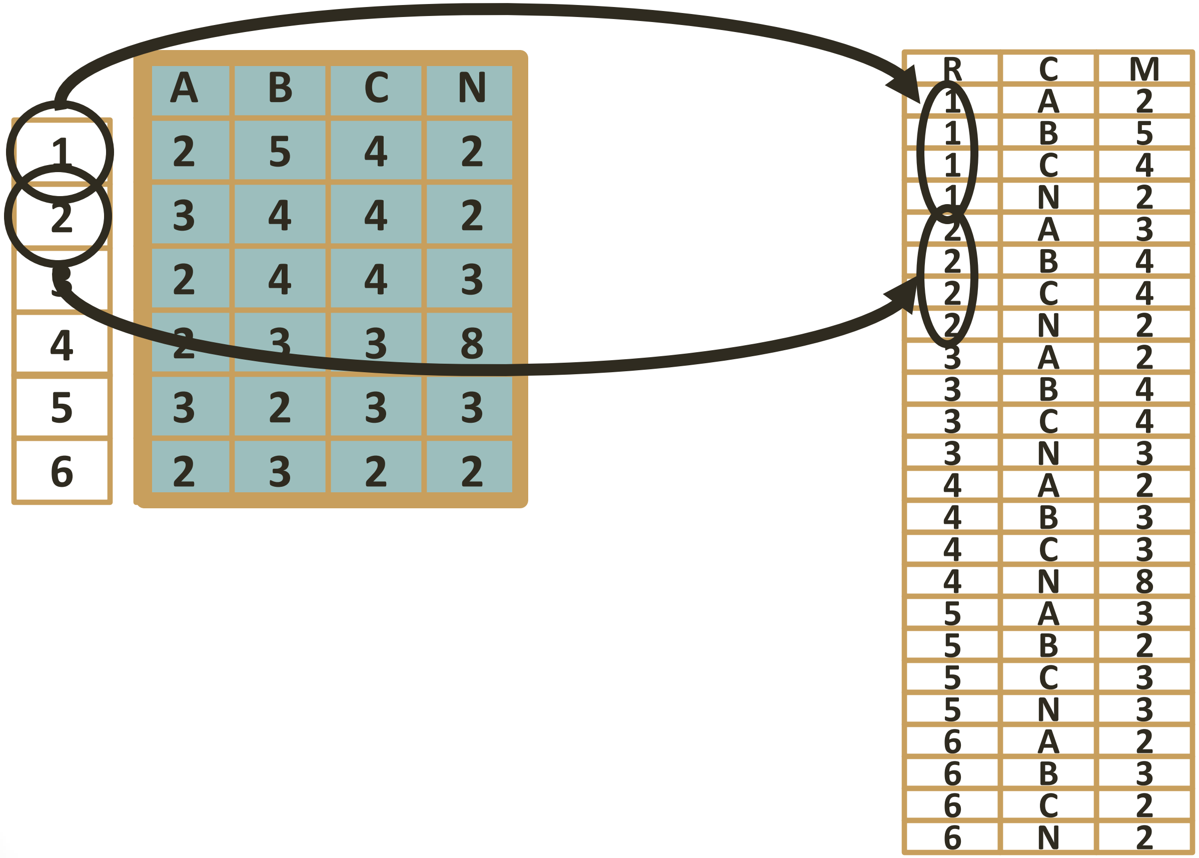
R	C	M
1	A	2
1	B	5
1	C	4
1	N	2
2	A	3
2	B	4
2	C	4
2	N	2
3	A	2
3	B	4
3	C	4
3	N	3
4	A	2
4	B	3
4	C	3
4	N	8
5	A	3
5	B	2
5	C	3
5	N	3
6	A	2
6	B	3
6	C	2
6	N	2

Rows may be resorted without changing the relation

A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3
2	3	3	8
3	2	3	3
2	3	2	2

R	C	M
1	A	2
1	B	5
1	C	4
1	N	2
2	A	3
2	B	4
2	C	4
2	N	2
3	A	2
3	B	4
3	C	4
3	N	3
4	A	2
4	B	3
4	C	3
4	N	8
5	A	3
5	B	2
5	C	3
5	N	3
6	A	2
6	B	3
6	C	2
6	N	2

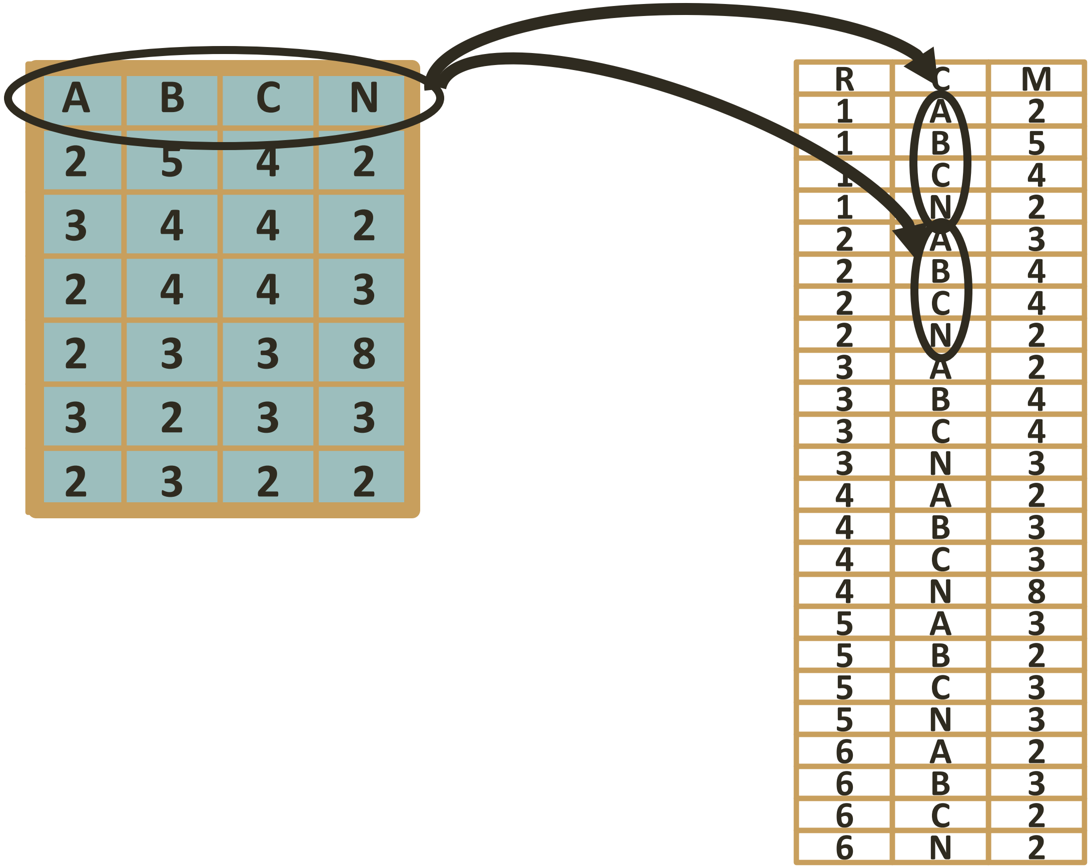
Schema Change Proved



	A	B	C	N
1	2	5	4	2
2	3	4	4	2
3	2	4	4	3
4	2	3	3	8
5	3	2	3	3
6	2	3	2	2

R	C	M
1	A	2
1	B	5
1	C	4
1	N	2
2	A	3
2	B	4
2	C	4
2	N	2
3	A	2
3	B	4
3	C	4
3	N	3
4	A	2
4	B	3
4	C	3
4	N	8
5	A	3
5	B	2
5	C	3
5	N	3
6	A	2
6	B	3
6	C	2
6	N	2

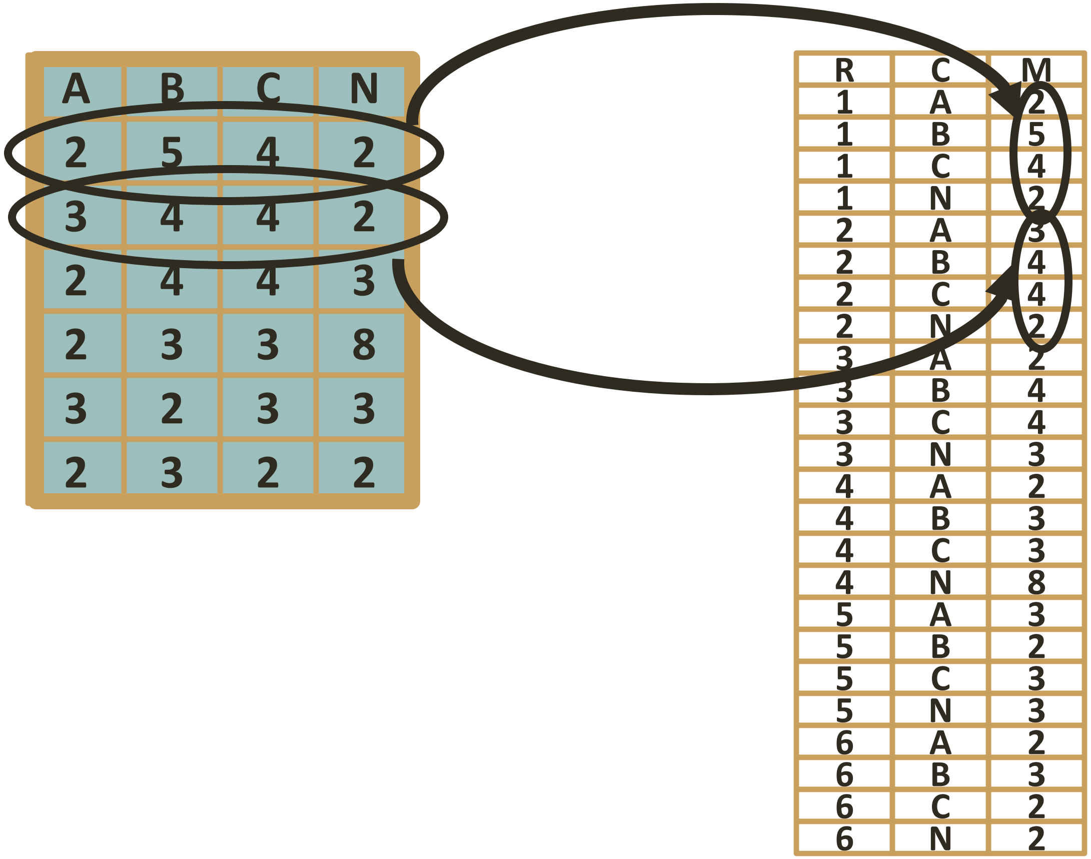
Schema Change Proved



A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3
2	3	3	8
3	2	3	3
2	3	2	2

R	C	M
1	A	2
1	B	5
1	C	4
1	N	2
2	A	3
2	B	4
2	C	4
2	N	2
3	A	2
3	B	4
3	C	4
3	N	3
4	A	2
4	B	3
4	C	3
4	N	8
5	A	3
5	B	2
5	C	3
5	N	3
6	A	2
6	B	3
6	C	2
6	N	2

Schema Change Proved



A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3
2	3	3	8
3	2	3	3
2	3	2	2

R	C	M
1	A	2
1	B	5
1	C	4
1	N	2
2	A	3
2	B	4
2	C	4
2	N	2
3	A	2
3	B	4
3	C	4
3	N	3
4	A	2
4	B	3
4	C	3
4	N	8
5	A	3
5	B	2
5	C	3
5	N	3
6	A	2
6	B	3
6	C	2
6	N	2

Data as Sparse Matrices

Multidimensional Sparse Matrices

Copyright 2016 by Ernst Henle

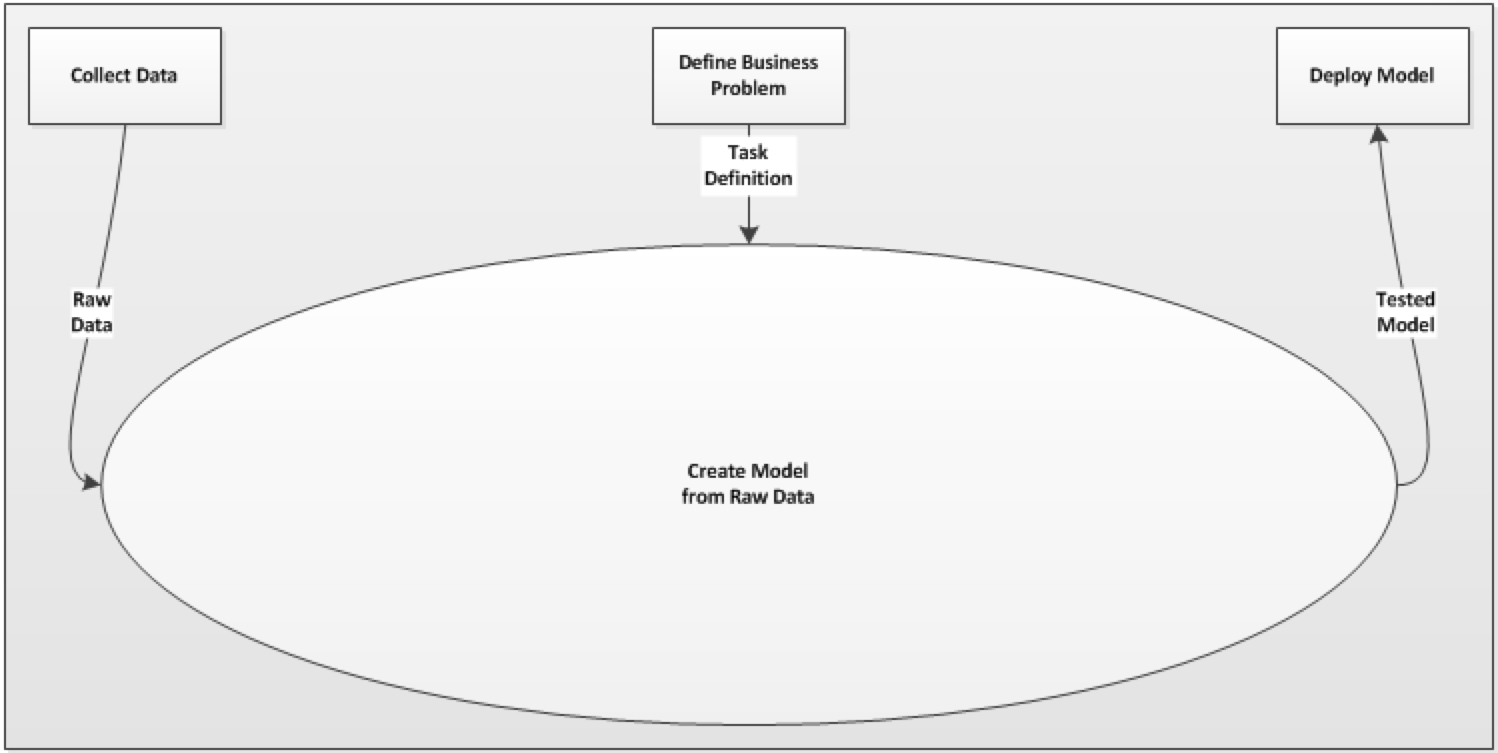
Break

Modeling Iteration Trap

The problem with iterative data preparation and training refinements

Copyright 2016 by Ernst Henle

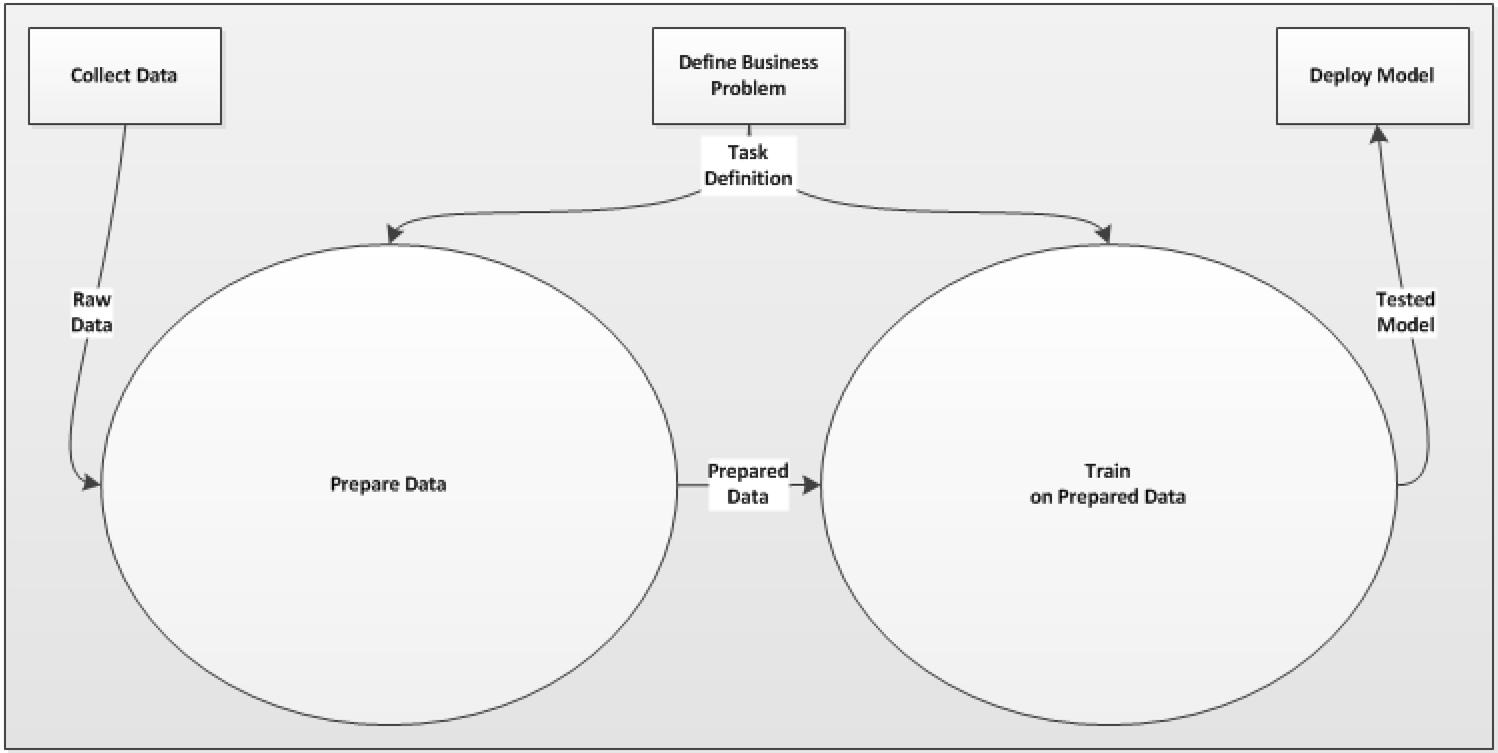
Modeling Iteration Trap (0)



Basic Predictive Analytics

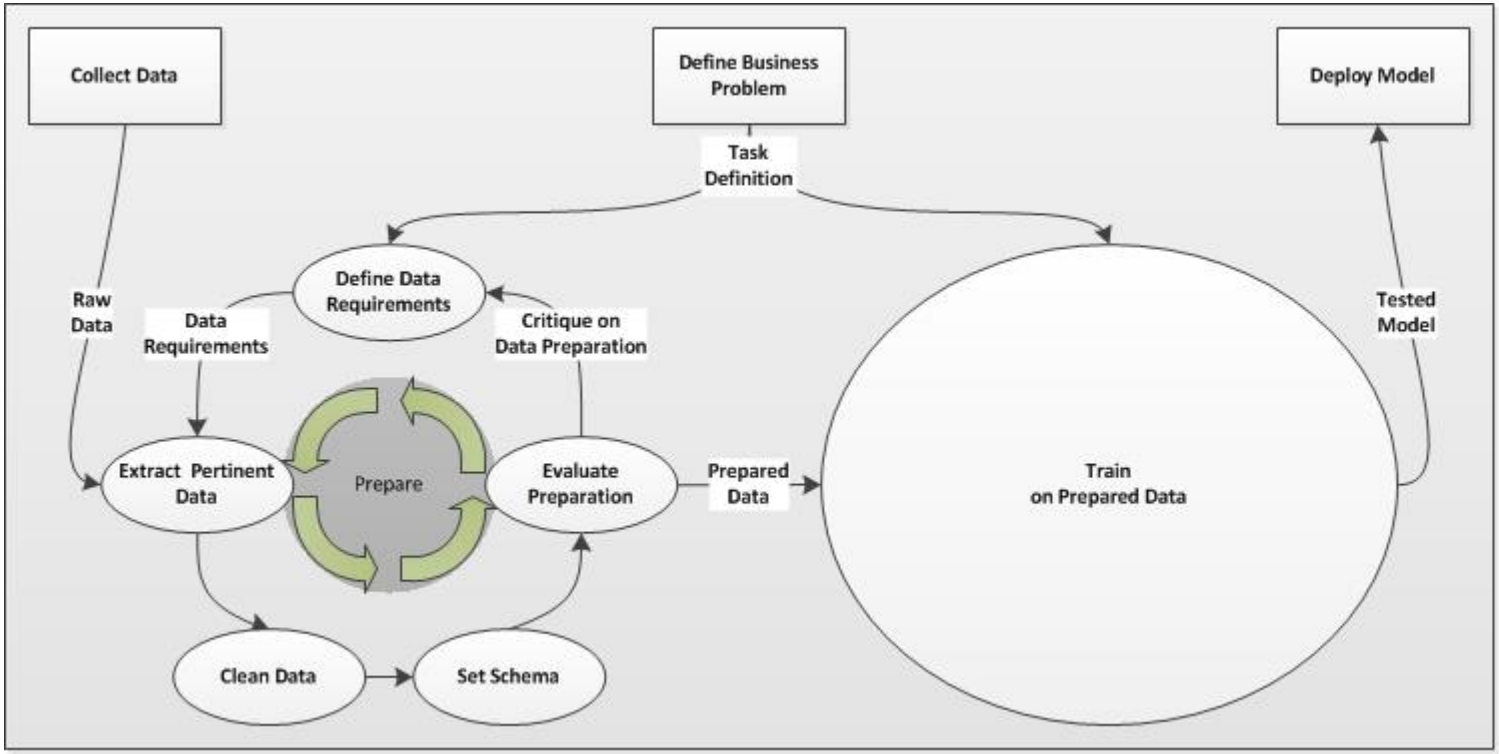
L0-DFD

Modeling Iteration Trap (1)



Predictive Analytics consists of Data Preparation and Modeling
L1-DFD

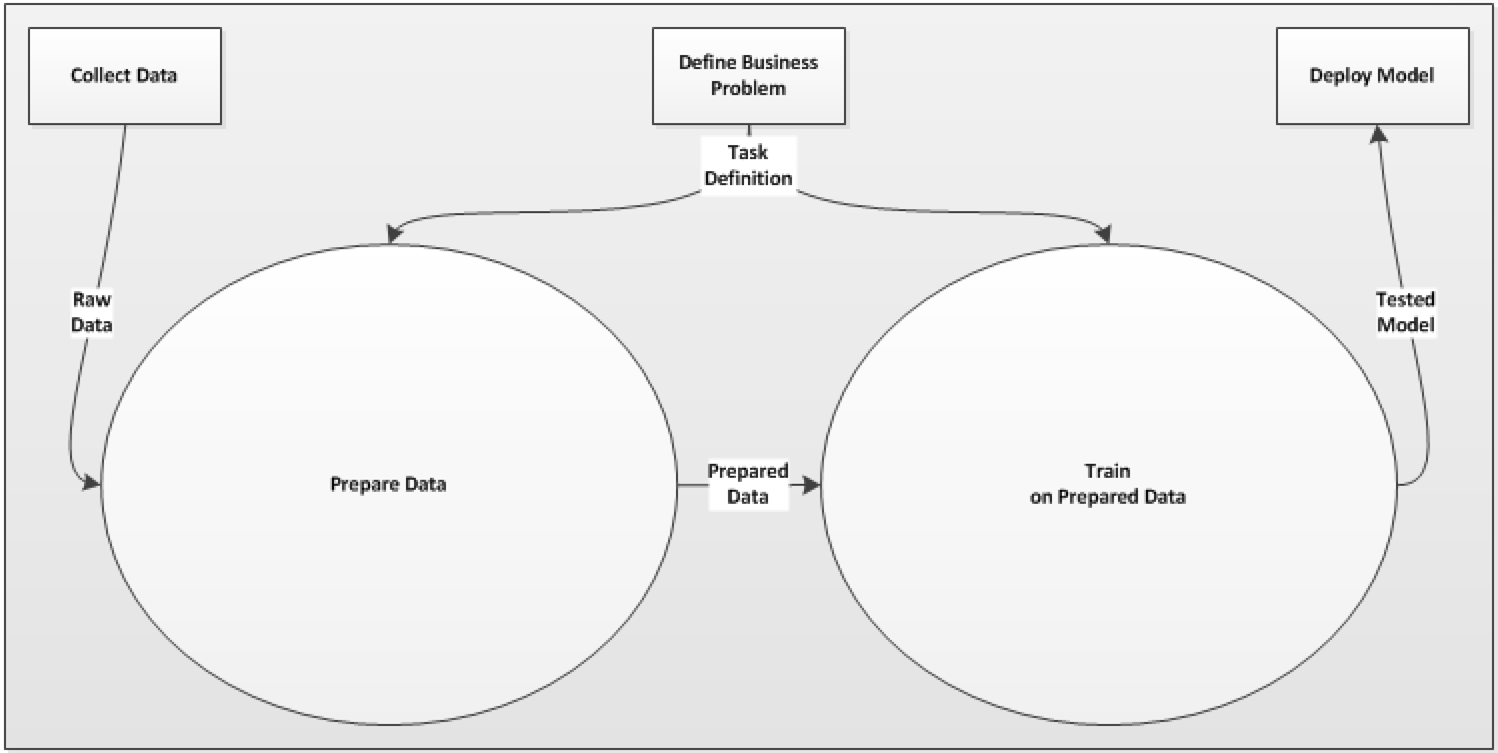
Modeling Iteration Trap (2)



Data Preparation is Iterative

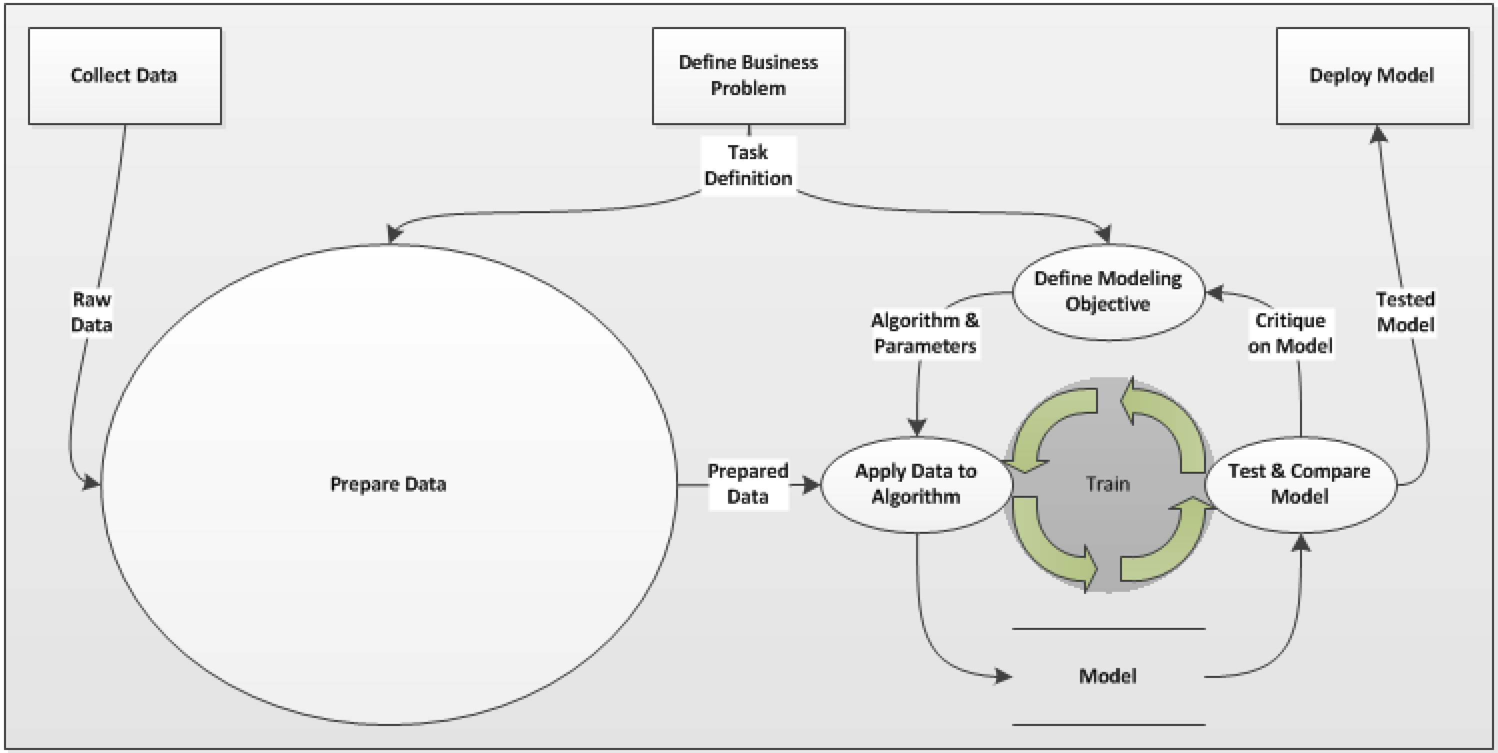
L2-DFD

Modeling Iteration Trap (3)



Predictive Analytics consists of Data Preparation and Modeling
L1-DFD

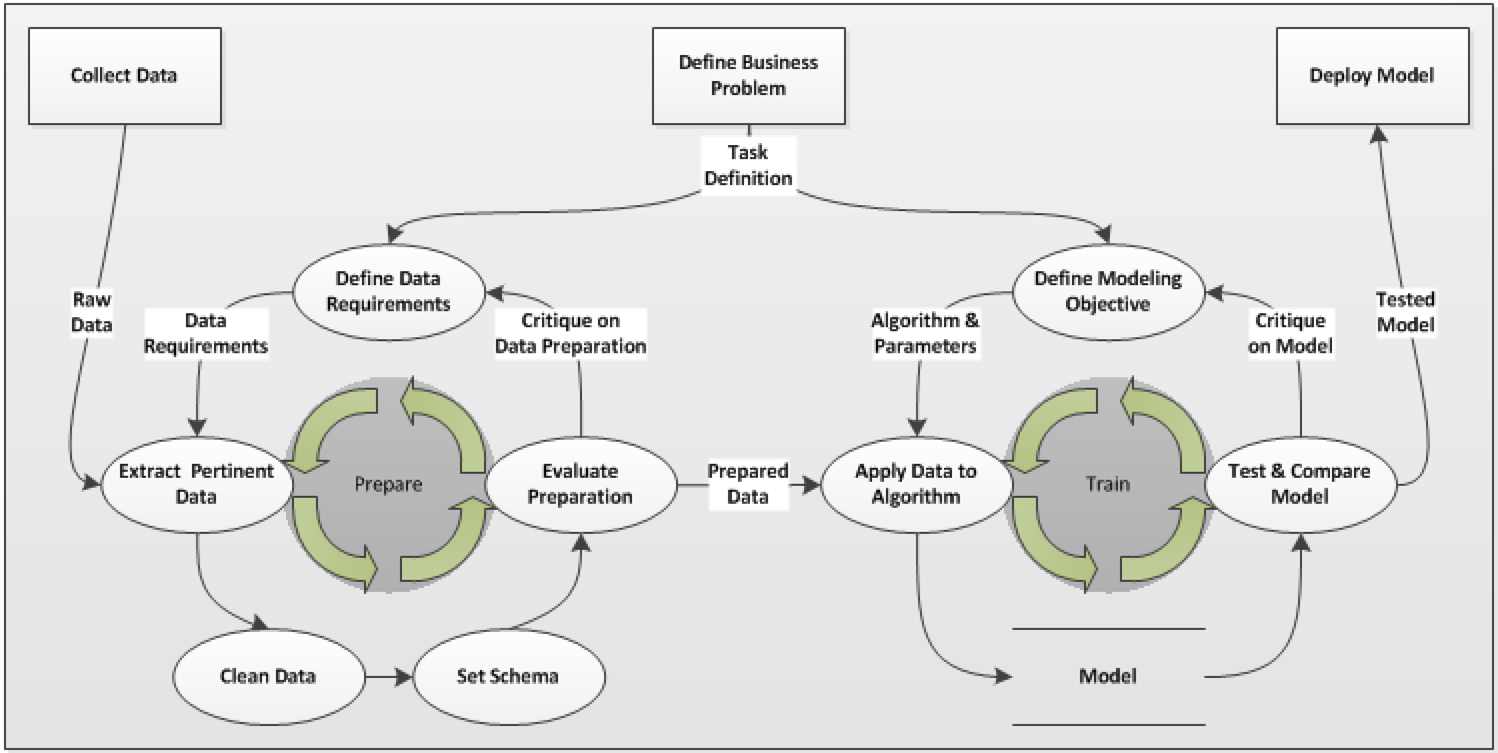
Modeling Iteration Trap (4)



Training is Iterative

L2-DFD

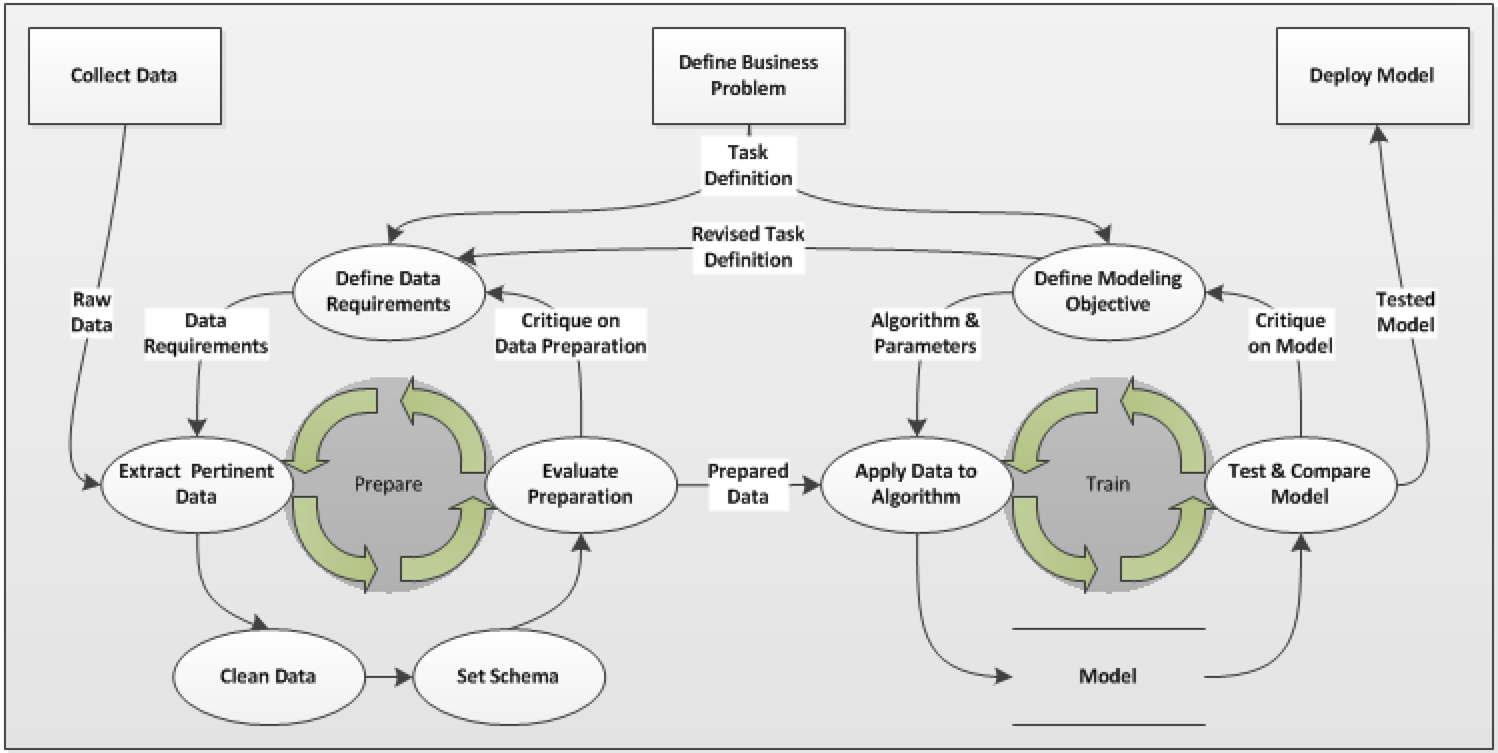
Modeling Iteration Trap (5)



Data Preparation and Training

L2-DFD

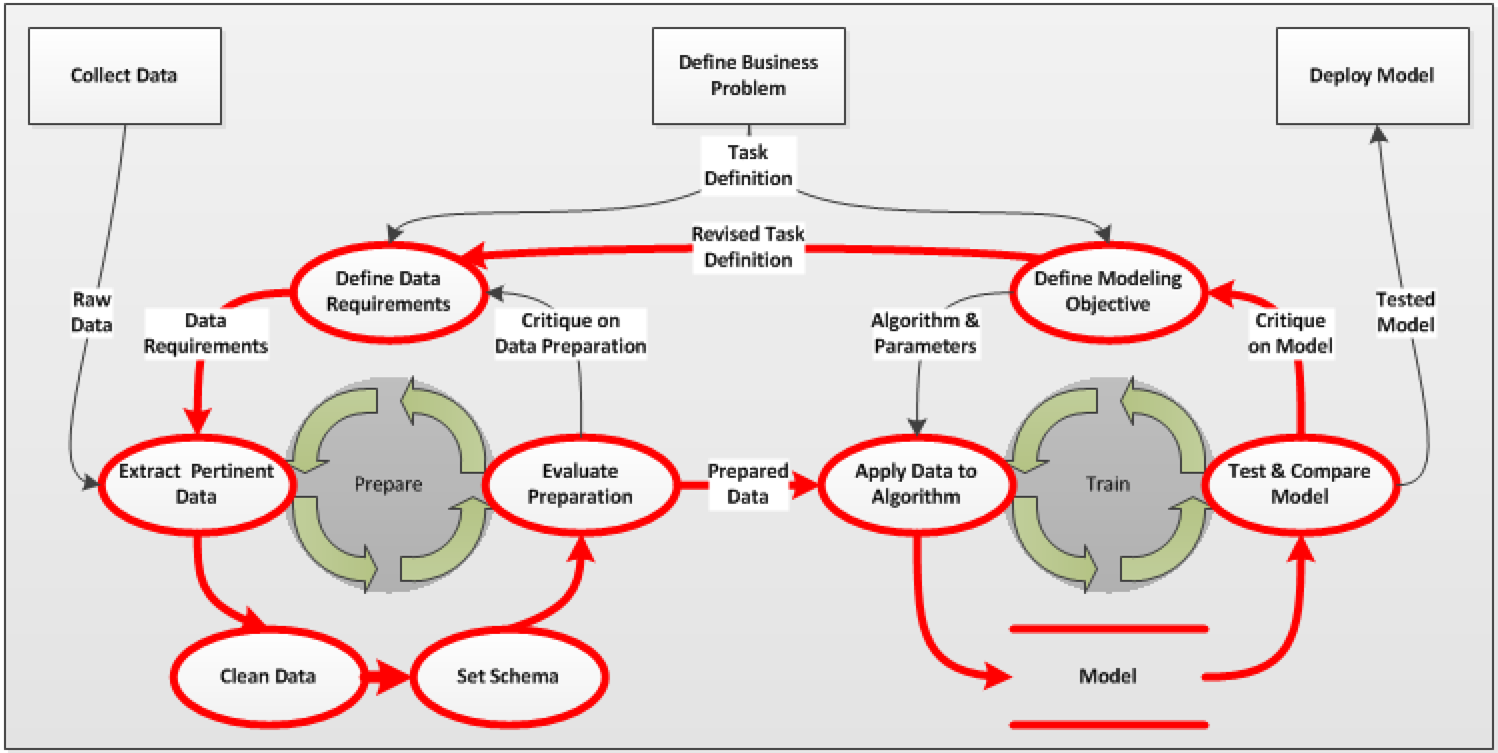
Modeling Iteration Trap (6)



Modeling Revises the Task Definition

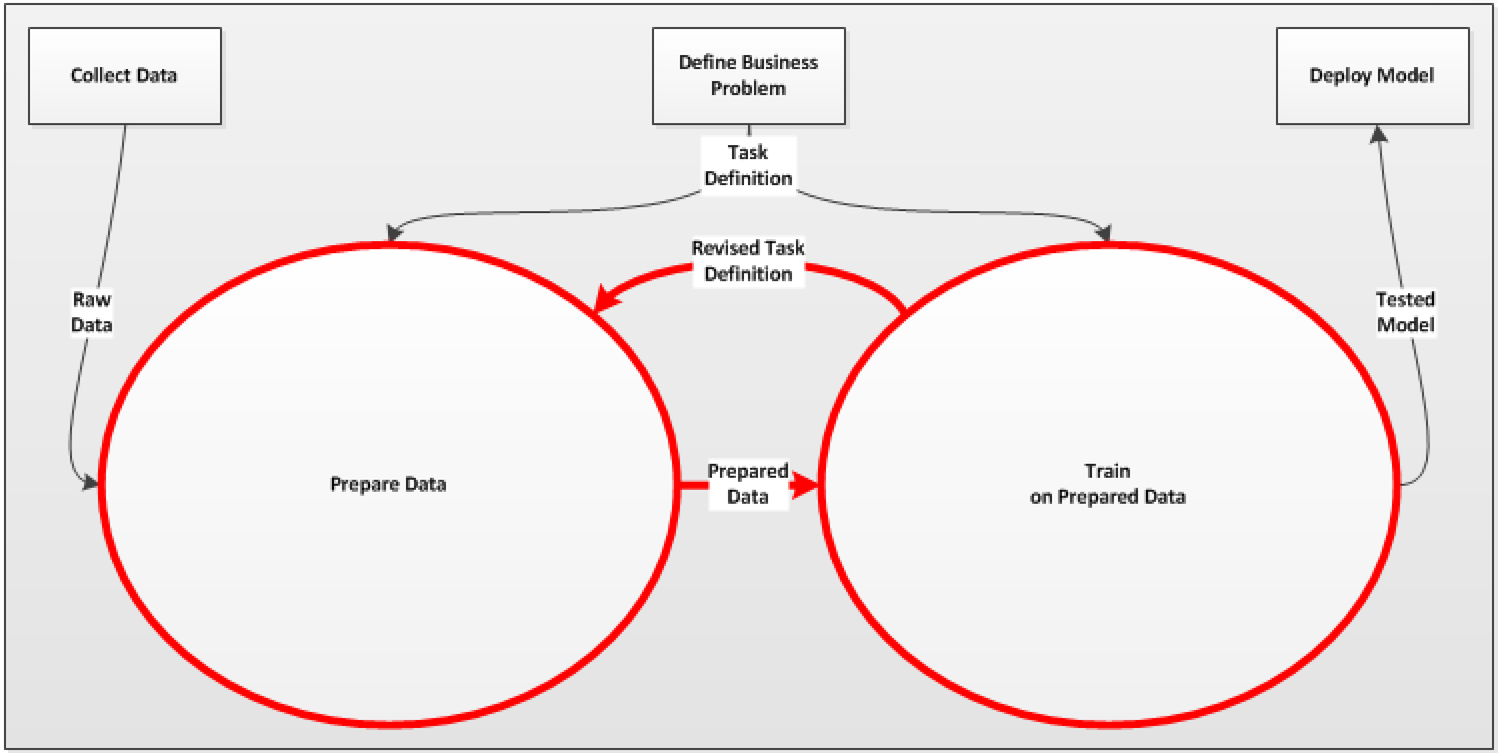
L2-DFD

Modeling Iteration Trap (7)



Revised Task Definition Completes a Cycle that is hard to optimize. Engineers get caught in a loop.

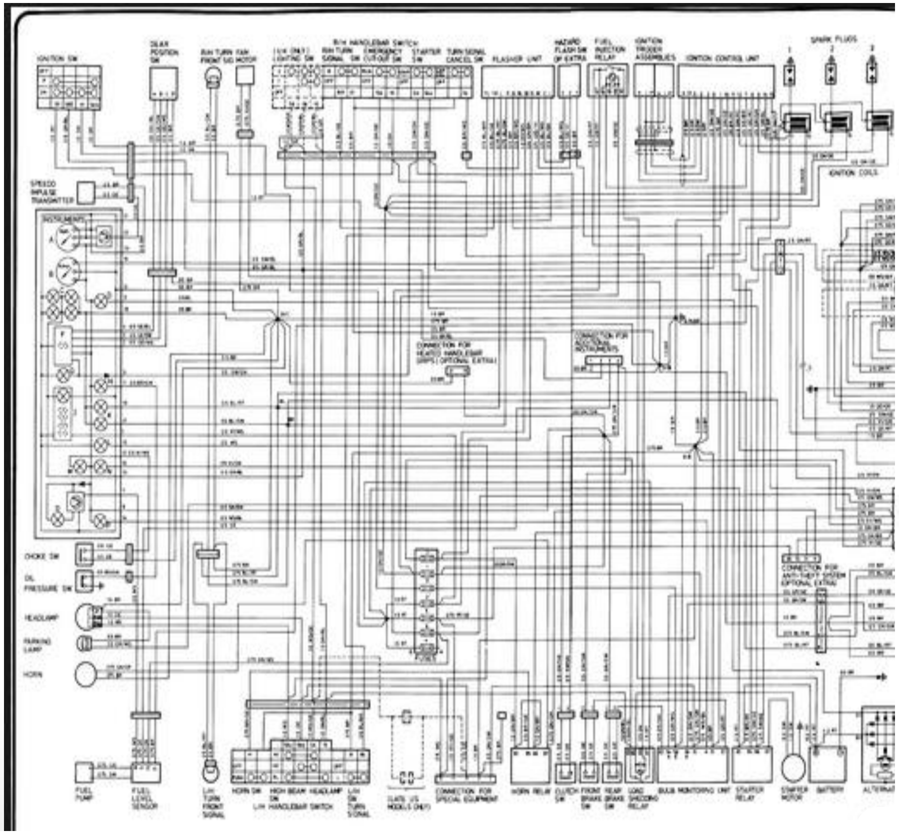
Modeling Iteration Trap (8)

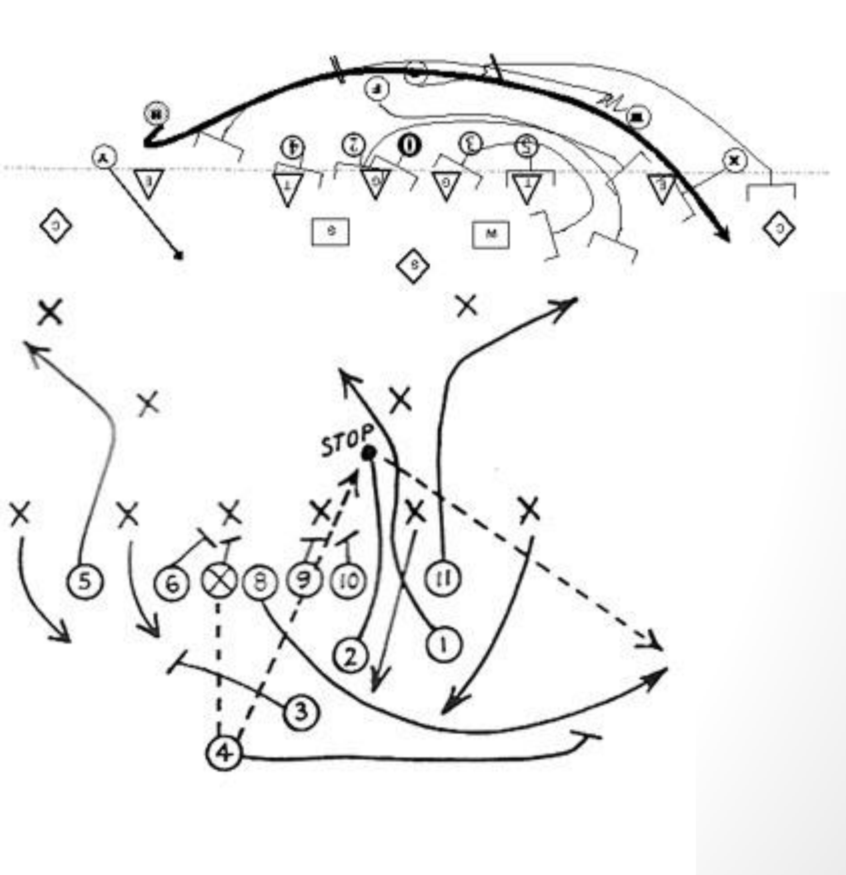


Revised Task Definition Completes a Cycle that is hard to optimize. Engineers get caught in a loop.

Modeling Iteration Trap (9)

More confusing diagrams





Modeling Iteration Trap

The problem with iterative data preparation and training refinements

Copyright 2016 by Ernst Henle

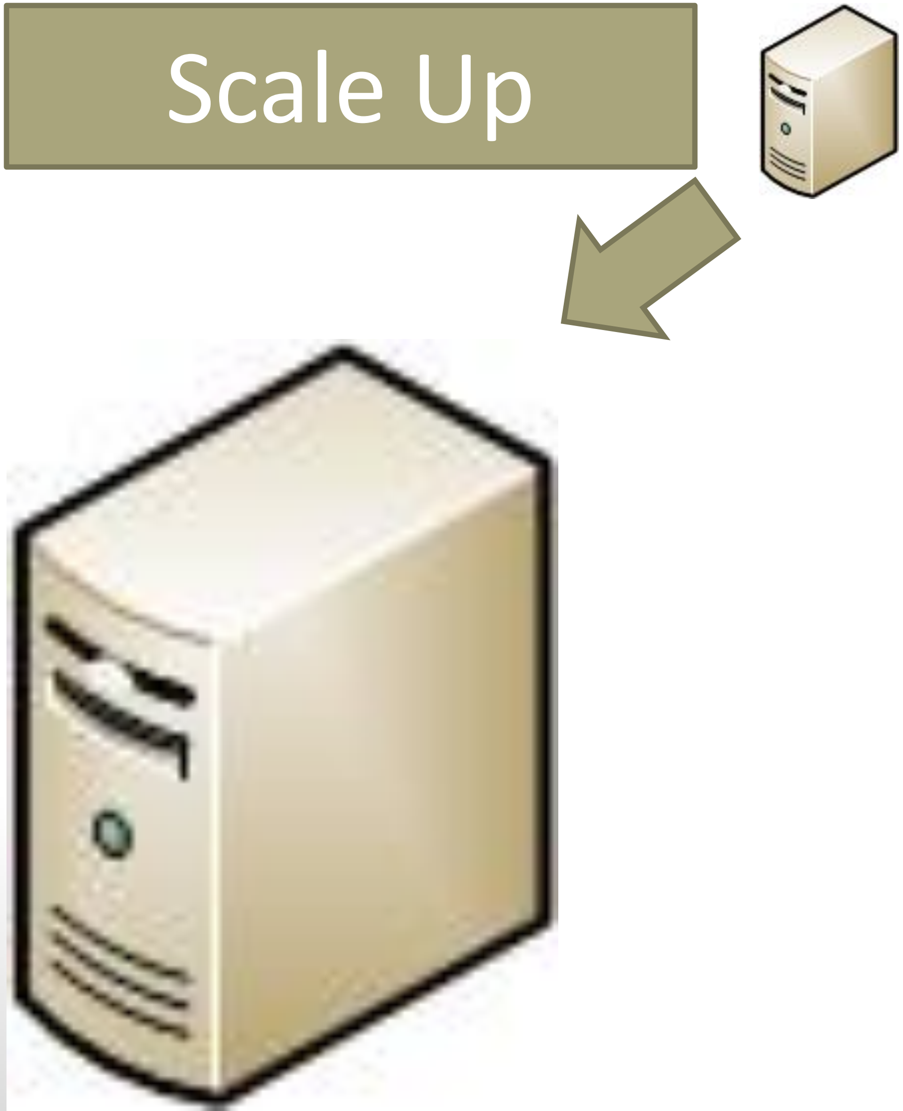
NOSQL:
Scale-out

Copyright 2016 by Ernst Henle

Scale-up vs. Scale-out

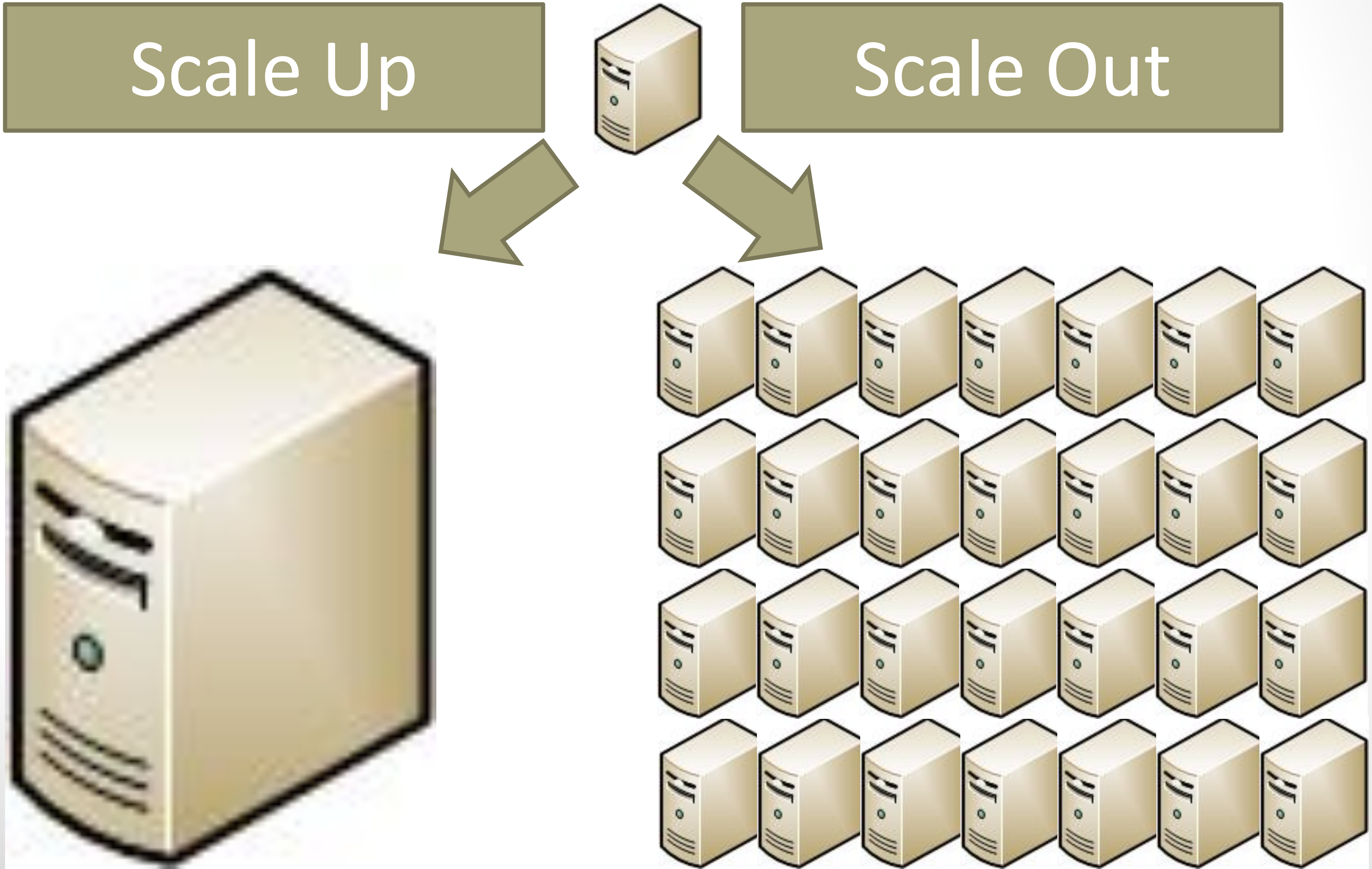
Before we discuss the nature of NOSQL, we should discuss the reasons for NOSQL.

Scale Up vs. Scale Out



Scale Up

Scale Up vs. Scale Out



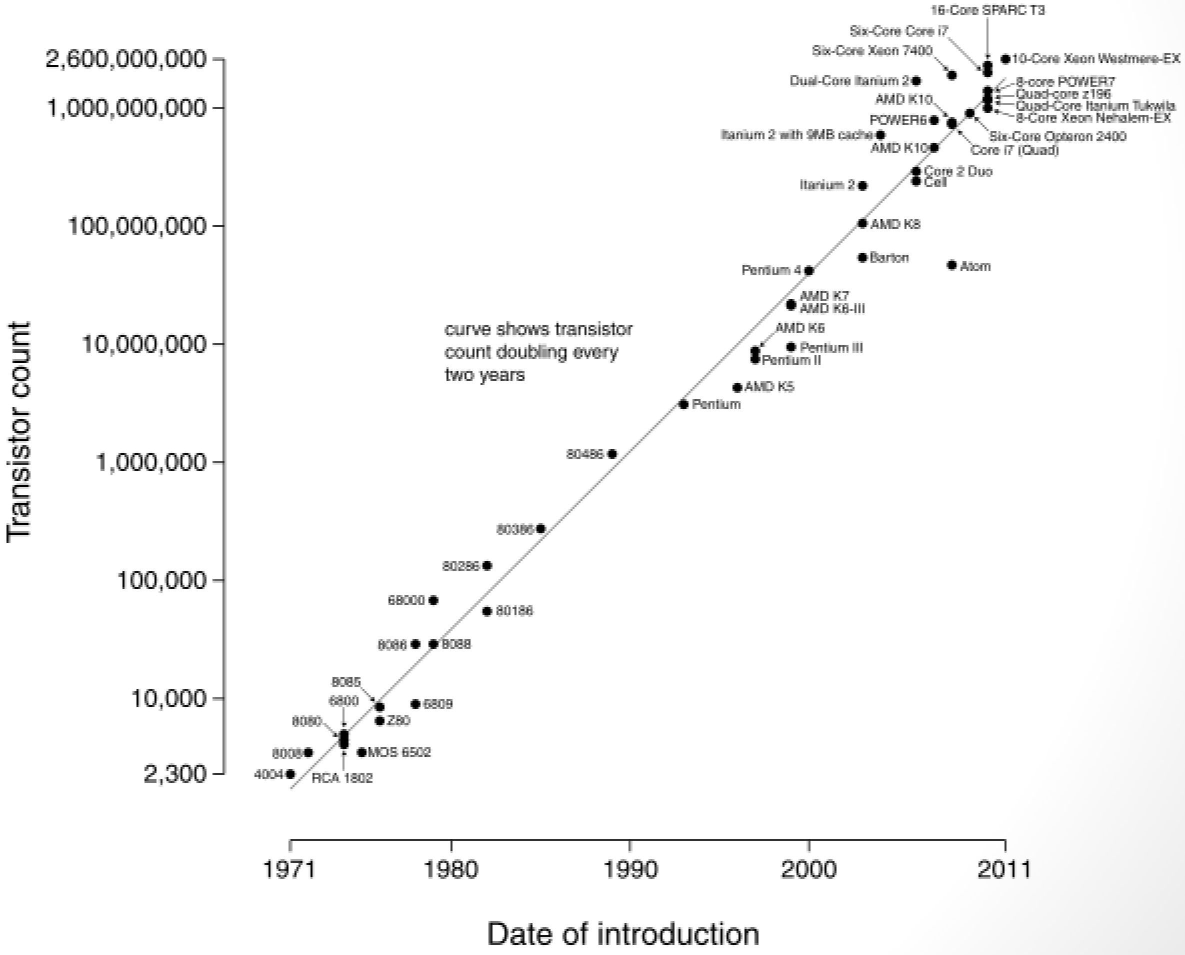
Scale Up

Scale Out

Scale-up vs. Scale-out

- Scale-up
- Moore’s Law

Microprocessor Transistor Counts 1971-2011 & Moore’s Law



Scale-up vs. Scale-out

Grace Hopper

Scale-up vs. Scale-out

Grace Hopper

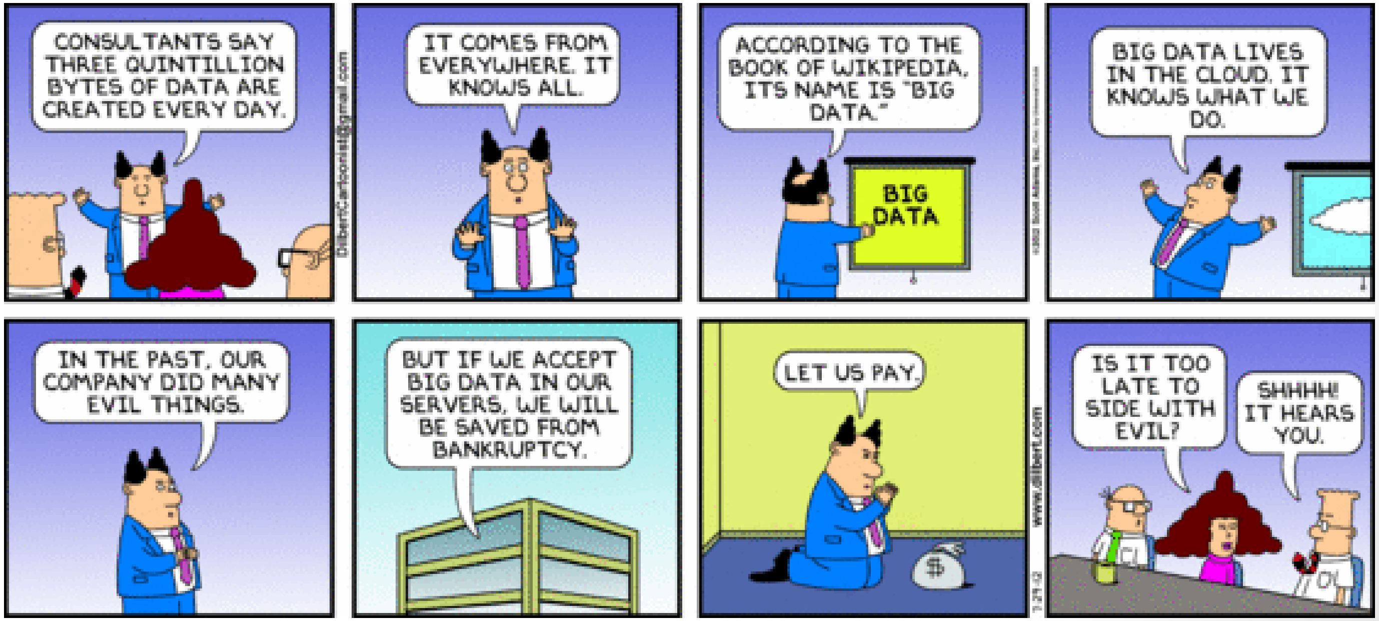
"In pioneer days they used oxen for heavy pulling, and when one ox couldn't budge a log, they didn't try to grow a larger ox. We shouldn't be trying for bigger computers, but for more systems of computers."

Cloud: Scale-out

- The primary characteristic of NOSQL is scale out.
- From a practical level, scale out requires an adjustable number of commodity computers.
- Cluster Elasticity: http://en.wikipedia.org/wiki/Elasticity_%28data_store%29
- Virtual Machine
  - One computer “mimics” another computer. (A system platform supports execution of an operating system)
  - Allows hardware standardization.
  - Allows one server to “host” many computers.
  - Virtual machines in the cloud can be set up and taken down (dehydrated, reduced to an image).
- Cloud: What is the “cloud”? Remote access to a single point provides many online services like servers and storage. (http://en.wikipedia.org/wiki/Cloud_computing).

Cloud: Services

- Amazon Web Services
- GoGrid
- Google Compute Engine
- Microsoft Azure
- Rackspace
- SoftLayer


CONSULTANTS SAY THREE QUINTILLION BYTES OF DATA ARE CREATED EVERY DAY.
DilbertCartoonist@gmail.com
IT COMES FROM EVERYWHERE. IT KNOWS ALL.
ACCORDING TO THE BOOK OF WIKIPEDIA, ITS NAME IS "BIG DATA."
BIG DATA
BIG DATA LIVES IN THE CLOUD. IT KNOWS WHAT WE DO.
IN THE PAST, OUR COMPANY DID MANY EVIL THINGS.
BUT IF WE ACCEPT BIG DATA IN OUR SERVERS, WE WILL BE SAVED FROM BANKRUPTCY.
LET US PAY.
$
www.dilbert.com
IS IT TOO LATE TO SIDE WITH EVIL?
SHHHH! IT HEARS YOU.

Scale-out and the "Cloud"

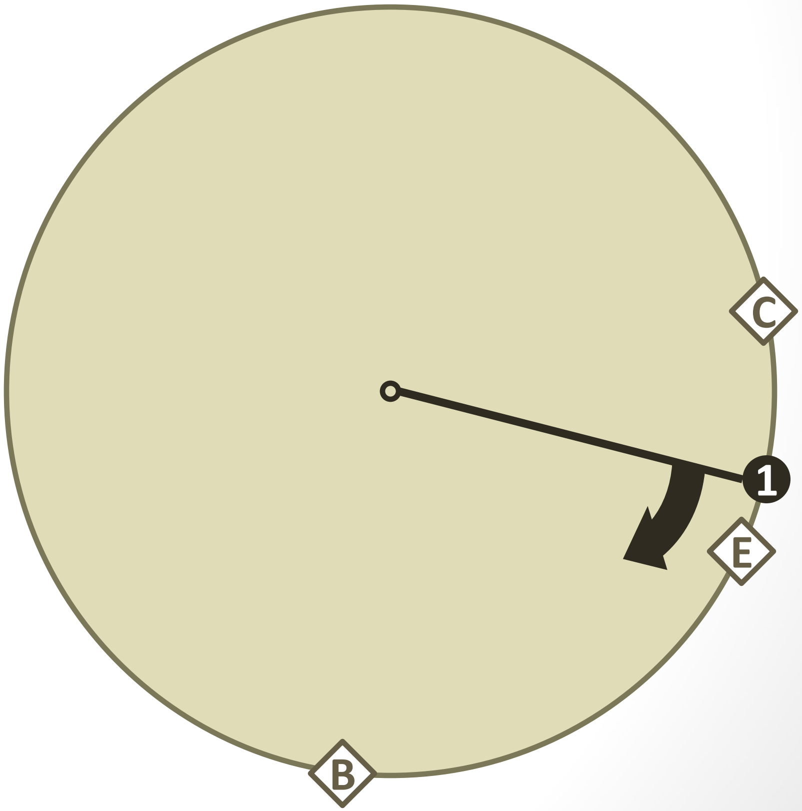
- **Elasticity** has made cloud computing feasible
- Clouds generally employ **virtual machines** that can be created at a moments notice, reduced to an image (dehydrated), re-started from an image, and deleted (recycled).
- How do we partition storage or usage among an unknown number of machines? Often we do not know ahead of time if new machines will become available or which machines will be recycled.
- Storage and usage are mapped to machines by a hash table. In traditional hash tables a change in the number of slots requires most keys to be remapped.
- We need a strategy to minimize remapping of storage and usage among the available computers: Consistent Hashing: http://en.wikipedia.org/wiki/Consistent_hashing

Consistent Hashing

- Consider a hash map where each object is mapped to a point on the circumference of a circle. For instance an object is mapped to the number of minutes on a clock.
- Computers, Files, Processes, etc., are mapped in this manner on the same circle.
- A computer "claims" all files and processes who have a hash that is clock wise to that computer.

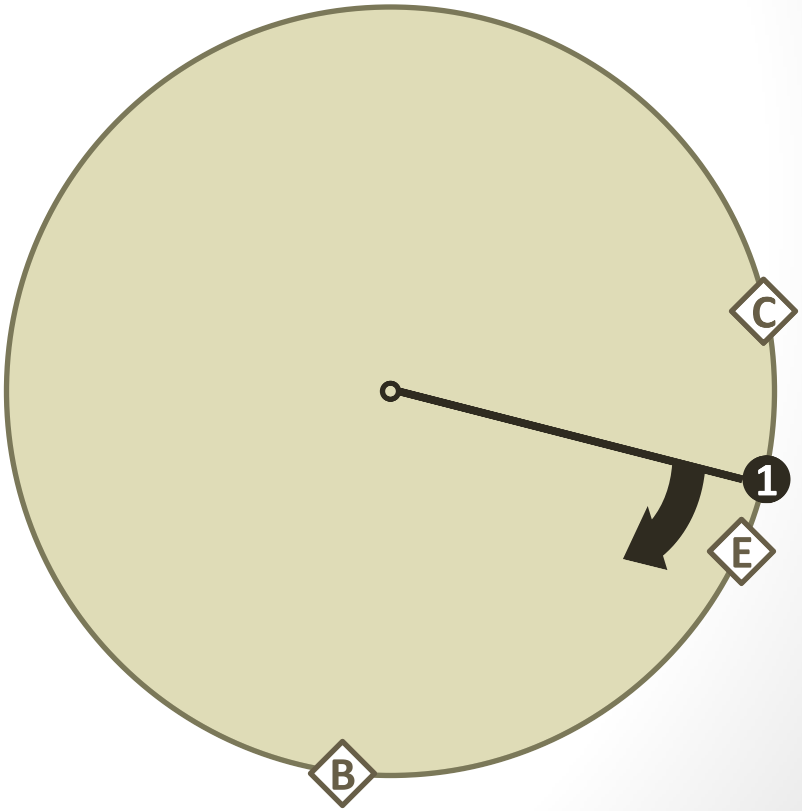
Consistent Hashing

Symbol	Object Type	Hash	Relation
B	Data Object	32	1
C	Data Object	14	1
E	Data Object	18	1
1	Machine 1	17	E C B



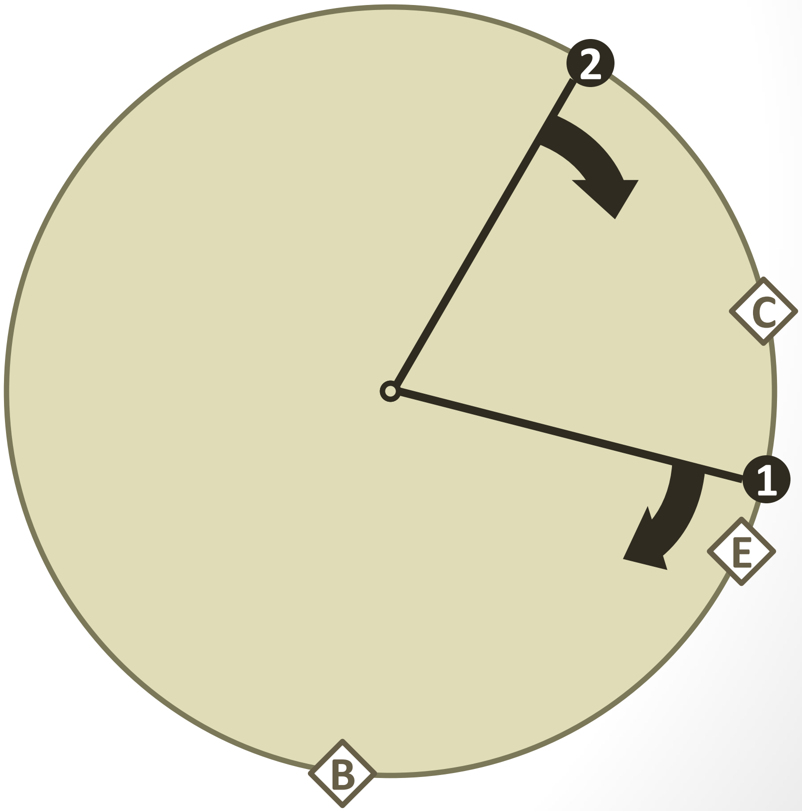
Consistent Hashing

Symbol	Object Type	Hash	Relation
B	Data Object	32	1
C	Data Object	14	1
E	Data Object	18	1
1	Machine 1	17	E C B



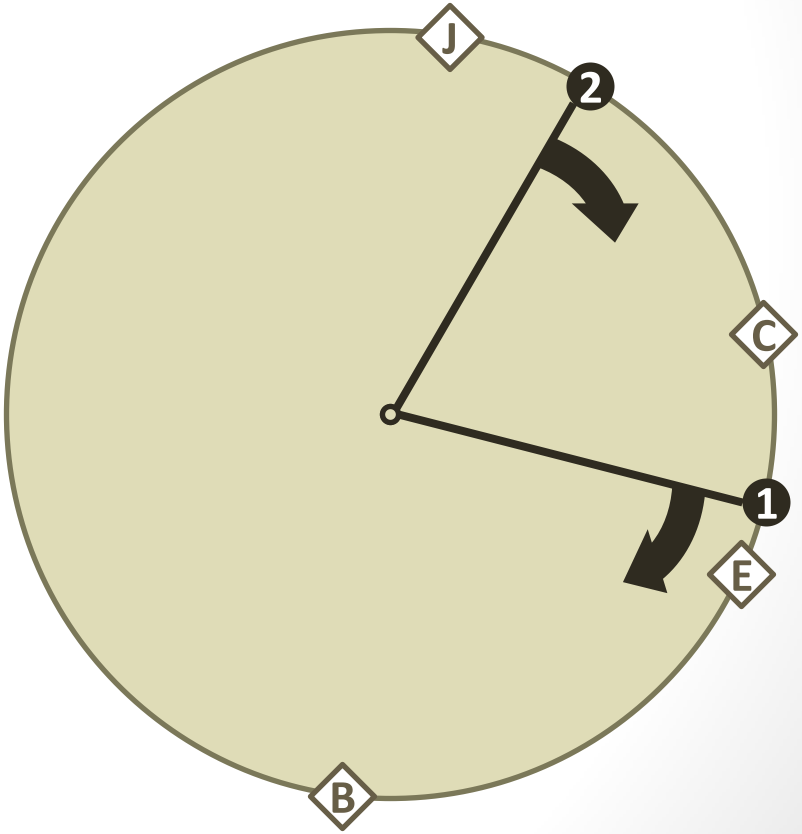
Consistent Hashing

Symbol	Object Type	Hash	Relation
B	Data Object	32	1
C	Data Object	14	2
E	Data Object	18	1
1	Machine 1	17	E B
2	Machine 2	5	C



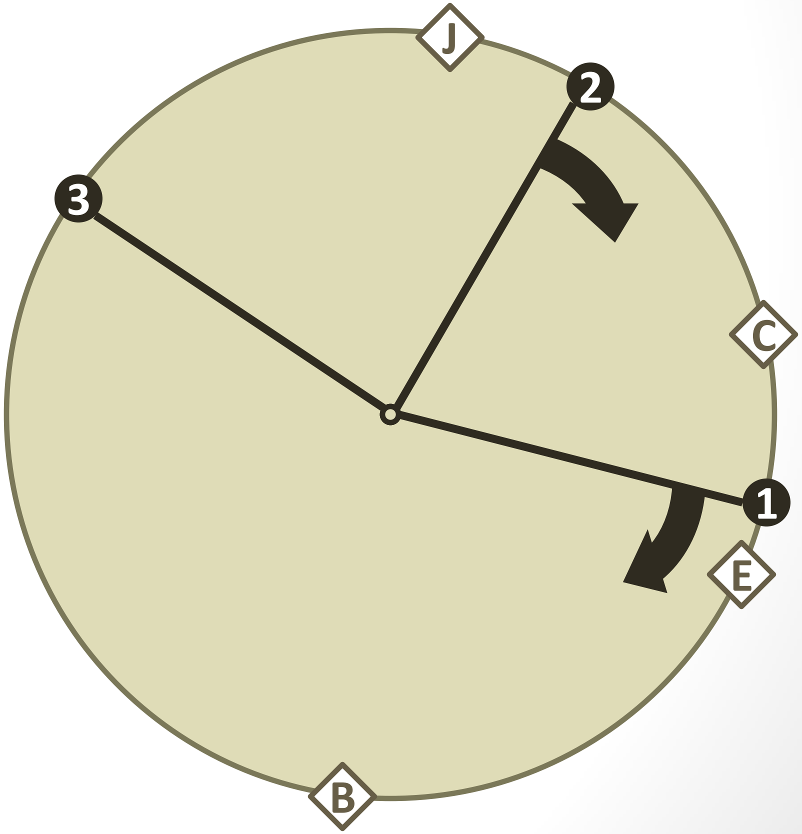
Consistent Hashing

Symbol	Object Type	Hash	Relation
B	Data Object	32	1
C	Data Object	14	2
E	Data Object	18	1
J	Data Object	2	1
1	Machine 1	17	E J B
2	Machine 2	5	C



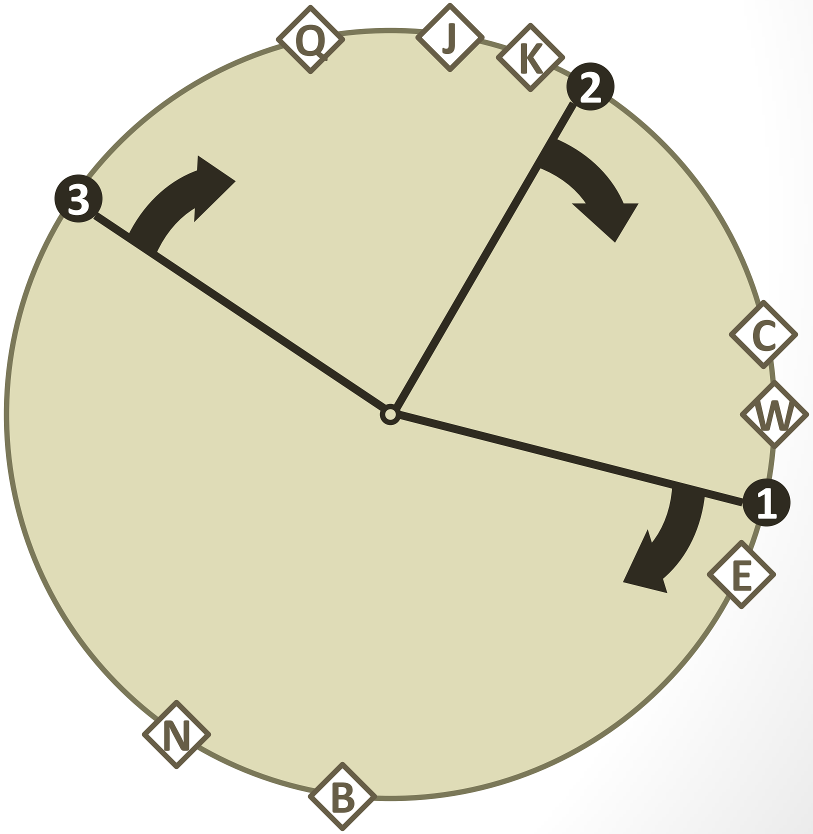
Consistent Hashing

Symbol	Object Type	Hash	Relation
B	Data Object	32	❶
C	Data Object	14	❷
E	Data Object	18	❶
J	Data Object	2	❸
❶	Machine 1	17	E B
❷	Machine 2	5	C
❸	Machine 3	51	J



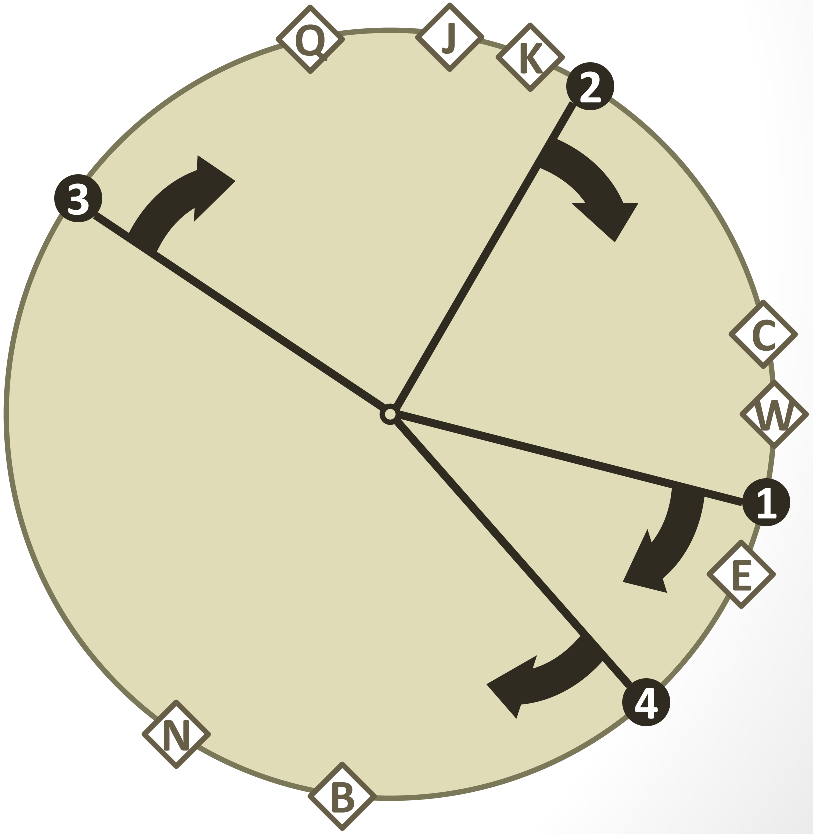
Consistent Hashing

Symbol	Object Type	Hash	Relation
B	Data Object	32	❶
C	Data Object	14	❷
E	Data Object	18	❶
J	Data Object	2	❸
K	Data Object	4	❸
N	Data Object	35	❶
Q	Data Object	57	❸
W	Data Object	15	❷
❶	Machine 1	17	E B N
❷	Machine 2	5	C W
❸	Machine 3	51	J K Q



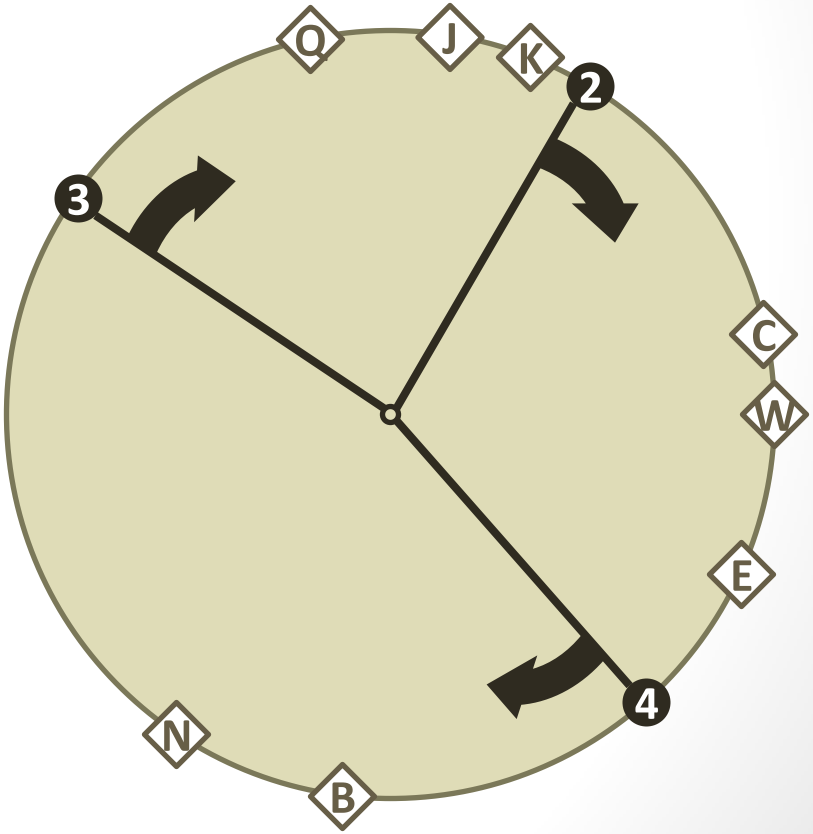
Consistent Hashing

Symbol	Object Type	Hash	Relation
B	Data Object	32	❹
C	Data Object	14	❷
E	Data Object	18	❶
J	Data Object	2	❸
K	Data Object	4	❸
N	Data Object	35	❹
Q	Data Object	57	❸
W	Data Object	15	❷
❶	Machine 1	17	E
❷	Machine 2	5	C W
❸	Machine 3	51	J K Q
❹	Machine 4	23	B N



Consistent Hashing

Symbol	Object Type	Hash	Relation
B	Data Object	32	❹
C	Data Object	14	❷
E	Data Object	18	❷
J	Data Object	2	❸
K	Data Object	4	❸
N	Data Object	35	❹
Q	Data Object	57	❸
W	Data Object	15	❷
❷	Machine 2	5	C W E
❸	Machine 3	51	J K Q
❹	Machine 4	23	B N



What does Scale-Out have to do with NOSQL?

- Traditional Relational Database Management Systems (RDBMS) have problems with scale-out.
- Therefore, new data base management schemes were desired.

NOSQL: Scale-out

Copyright 2016 by Ernst Henle

New Terminology

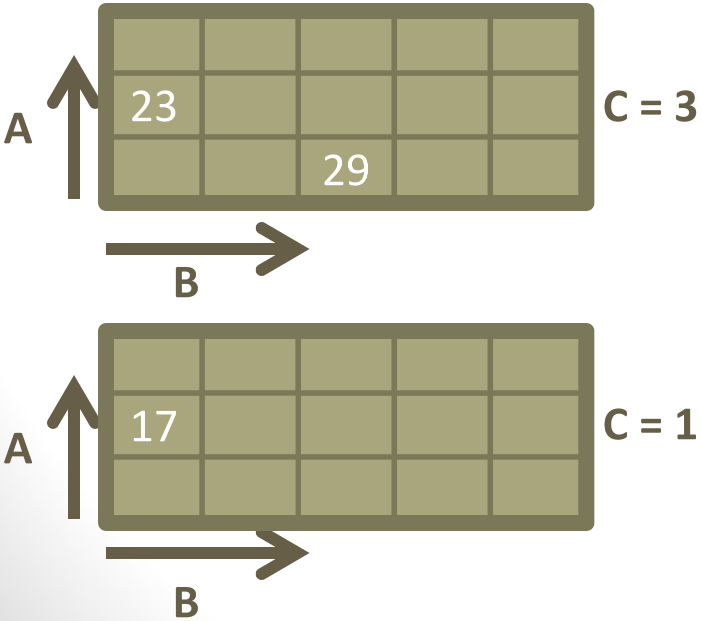
- Hadoop
- Master Node
- Data Node
- Cluster
- Hive
- Impala
- MapReduce
- HDFS
- Doug Cutting
- Scalability
- AWS
- Elastic Cloud
- NoSQL
- CAP Theorem
- Consistency (CAP)
- Availability (CAP)
- Partition Tolerance (CAP)
- Eric Brewer
- RDBMS
- ACID
- Atomic (ACID)
- Consistent (ACID)
- Isolation (ACID)
- Durability (ACID)
- BASE
- Eventual Consistency
- Paxos
- Sqoop
- CouchDB
- Shared Data
- Stale Data
- Scale-out
- Scale-up
- Grace Hopper
- Data Replication
- Horizontal Partitioning
- Vertical Partitioning
- Heartbeats
- Multi-Version Concurrency Control
- EAV
- Relational Algebra
- Relational Calculus
- Relational Model
- Ted Codd
- Codd's Theorem
- Transaction Shell
- Column-oriented DBMS
- Row-oriented
- SPARQL

Assignment

Copyright 2016 by Ernst Henle

Assignment (1)

1. Below is a sparse multi-dimensional matrix. Create a standard table, called Table 1, and an EAV representation, called Table 2, of this sparse matrix. Create these tables by “hand”. You do not need to write code. You may want to consult the lecture slides. Show the actual tables:
   a) Table 1 will have as headers: A, B, C, & N.
   b) Table 2 will have as its headers: R, C, & M. The “C” in Table 2 has a different meaning than the “C” in Table 1.



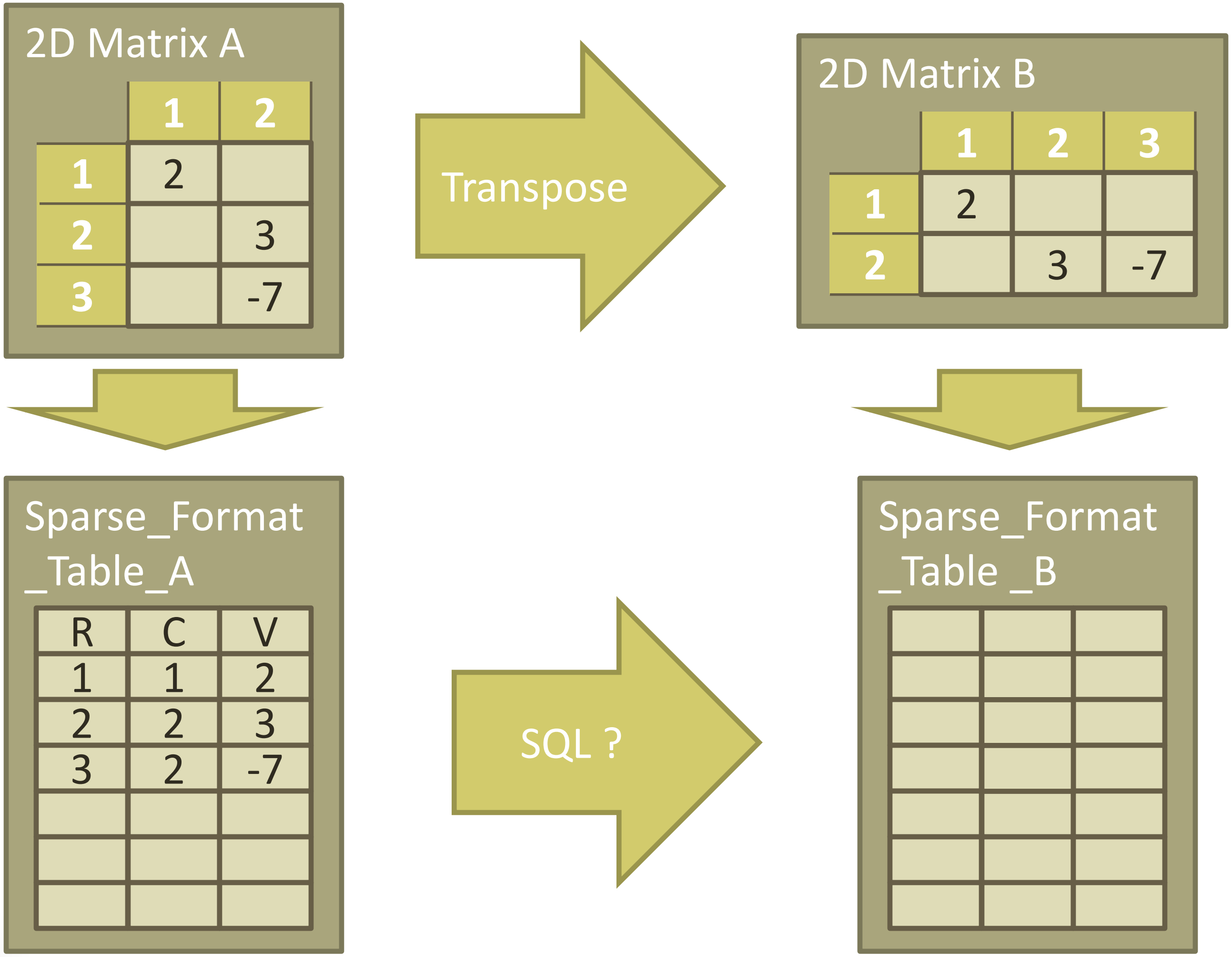
Data from a real estate survey of single-family houses in downtown Seattle. Cell values are number (**N**) of houses found for sale:

- **A**: Area in 1000’s of square feet
- **B**: Number of Bathrooms
- **C**: Cost in $100,000.-

Assignment (2)

2. Change the schema of the data in item 1 above by changing the EAV table, called Table 2 in the following way: New values will represent Cost per Area (CPA). You can calculate CPA from the existing information. Modify this table by “hand”. You do not need to write code.
3. Use SQL to manipulate sparse 2D matrices in the sparse 2D matrix format. Use select statements to transform the relations. Do not use create, update, or insert to modify the database. The SQL code is simple like in the Exercises 1 through 4 of MatrixAlgebra.sql. Given that 2D matrices are encoded in the sparse 2D matrix format, do the following:
   a) Write SQL for scalar multiplication of a 2D matrix stored in the sparse 2D matrix format. See Exercise 5 in MatrixAlgebra.sql
   b) Write SQL for transposition of a 2D Matrix stored in the sparse 2D matrix format. See Exercise 6 in MatrixAlgebra.sql and the next slide.
   c) Optional: Write SQL for addition of two 2D matrices in the sparse 2D matrix format. See Exercise 7 in MatrixAlgebra.sql.

Assignment (3)



2D Matrix A

	1	2
1	2	
2		3
3		-7

2D Matrix B

	1	2	3
1	2		
2		3	-7

Sparse_Format_Table_A

R	C	V
1	1	2
2	2	3
3	2	-7

Sparse_Format_Table _B

Assume that Sparse_Format_Table_A is the only table in the database. What SQL statement will present Sparse_Format_Table_B?

Assignment (4)

4. Complete Assignment items 1, 2, and 3 in a *.txt or *.doc file. I will copy and paste the sql statements into a database engine. Submit by Saturday 11:57 PM.
5. Readings
   a. Look through the preview section
   b. Re-review the new terminology slide
   c. Read **Graph structure in the web** by Broder et al.:
      - http://www.cis.upenn.edu/~mkearns/teaching/NetworkedLife/broder.pdf

Assignment

Copyright 2016 by Ernst Henle

Introduction to Data Science

Copyright 2016 by Ernst Henle