Introduction to Data Science

Lecture 7; May 9th, 2016

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

Agenda

- Announcements
  - Encourage Group Homework and ask questions on LinkedIn
  - If you need to submit late, you can get 50% of the points, if you inform me prior to the deadline. I can give you a 12-hour extension. Your work must be at least 8/10 points to get the necessary 4/10 points.
- Data Science – The Business Point of View
- Break
- Review Relational Algebra (Homework)
- Quiz 07a Relational Algebra
- Sparse Matrices and EAV
- Quiz 07b EAV
- Sparse Matrix Exercises (related to Homework Assignment)
- Break
- Sparse Matrix Manipulation in SQL (related Homework Assignment)
- Break
- NoSQL Scale Out (Time Permitting)
- Assignment (Complete all assignments items from all assignment slides. Must be completed by Saturday 11:56 PM)

Data Science – The Business Point of View

Marius Marcu is a strategic innovator with a great passion for high-tech. His career path includes companies like Intel, where he managed nine microprocessor product lines from planning to end-of-life, and Microsoft, where he led Competitive Business strategy efforts for SQL Server 2012 and helped launch great products like Microsoft Surface 3 Pro. Last Year, Marius led marketing for Data Protection solutions at Quantum Corporation. Currently, Marius is the Senior Product Marketing Manager at Smartsheet.

While at Microsoft, Marius fell in love with big data, cloud technologies and data science, which he thinks will make a lot of people rich in the next 10-20 years, including himself. In his personal life, he cares a lot about privacy, and even more about how he can help his 2 sons grow up with a healthy sense of data. Marius is an alumnus of this certificate program. Data Science professionals, like Marius, who come from a business background are few. We are fortunate to get his analysis of the Data Science business landscape.

Marius' lecture slides are posted on Canvas in: Data science - the business point of view May 2016 v8.pdf

- https://www.linkedin.com/in/mariusmarcu
- mariusmarcu@global.t-bird.edu

Break

Relational Algebra: Review

- Homework: RelationalAlgebraSQLHomework.sql
- References:
  - RelationalAlgebraAndSQL.pdf
  - RelationalAlgebraAndSQL.sql

Quiz 07a (Relational Algebra)

- The questions are presented during the quiz

Data as Sparse Matrices

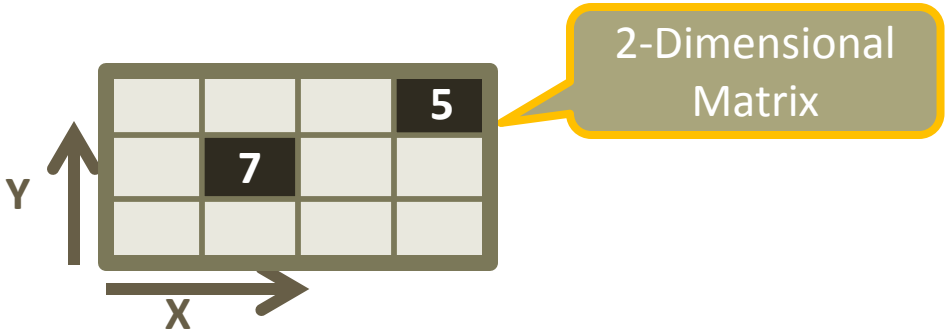
Cartesian Product

- 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.

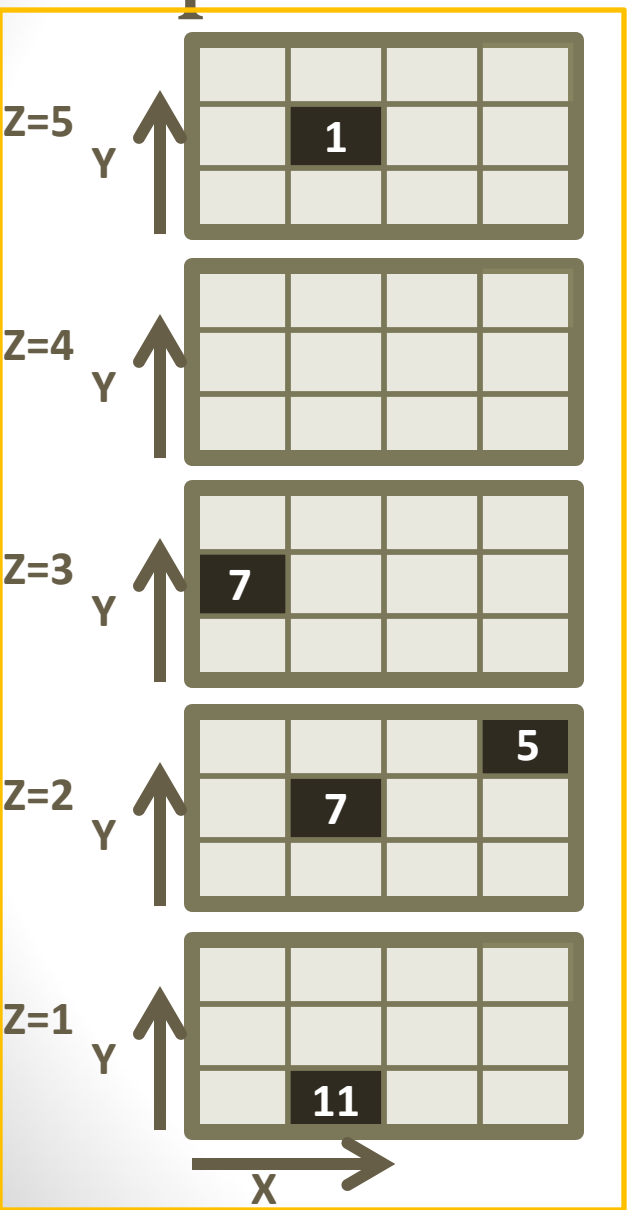
- Relational Algebra
- http://en.wikipedia.org/wiki/Relational_algebra
- In Relational Algebra we need the Cartesian product to combine tuples into a single tuple. The Cartesian product creates a new schema (relation) from other relations.

- 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

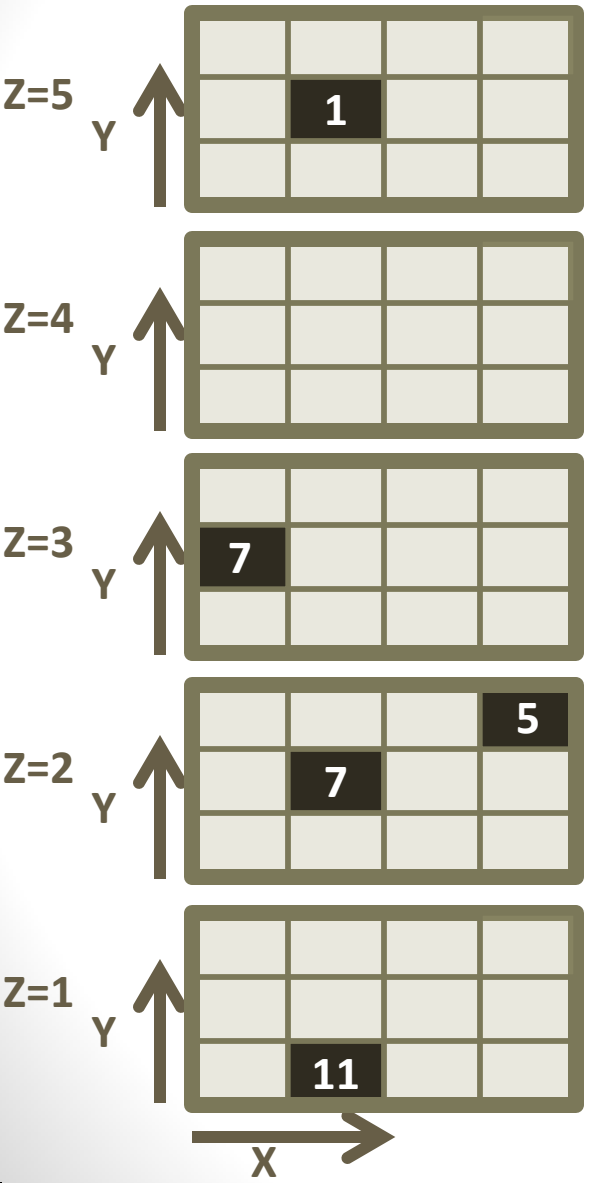


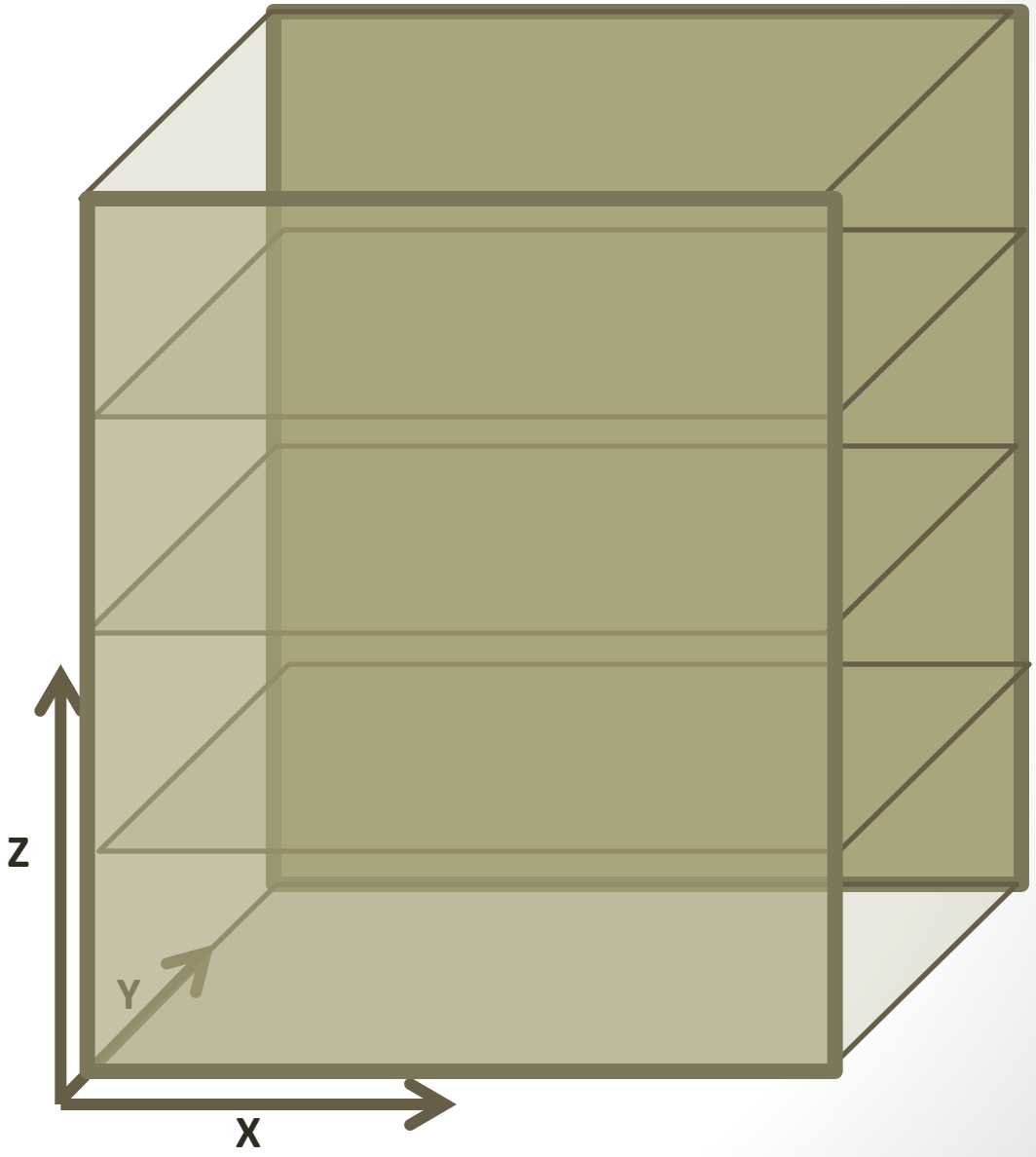
Sparse Matrices



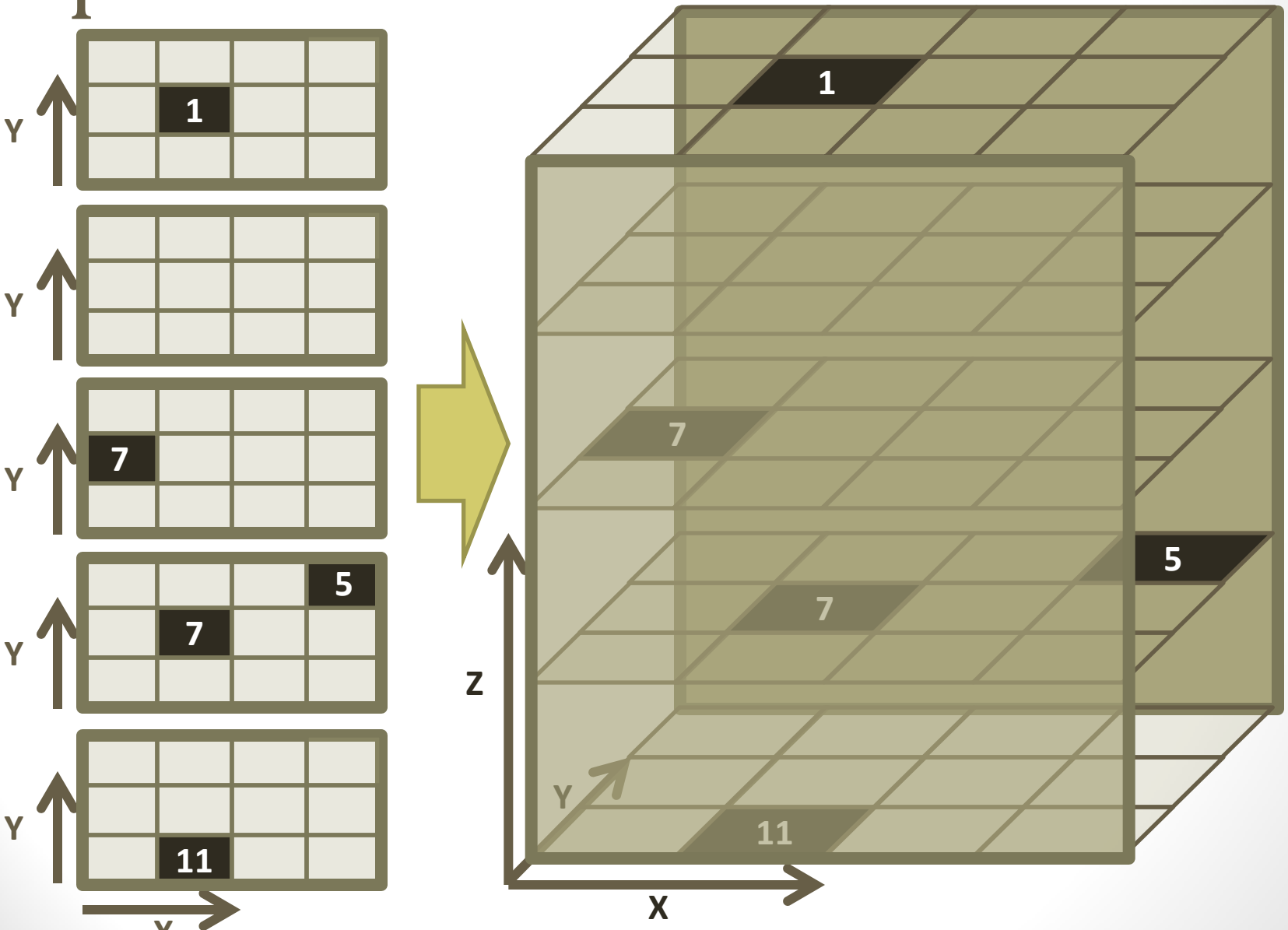
A series of equal-sized 2-dimensional matrices is a 3-dimensional matrix

Sparse Matrices





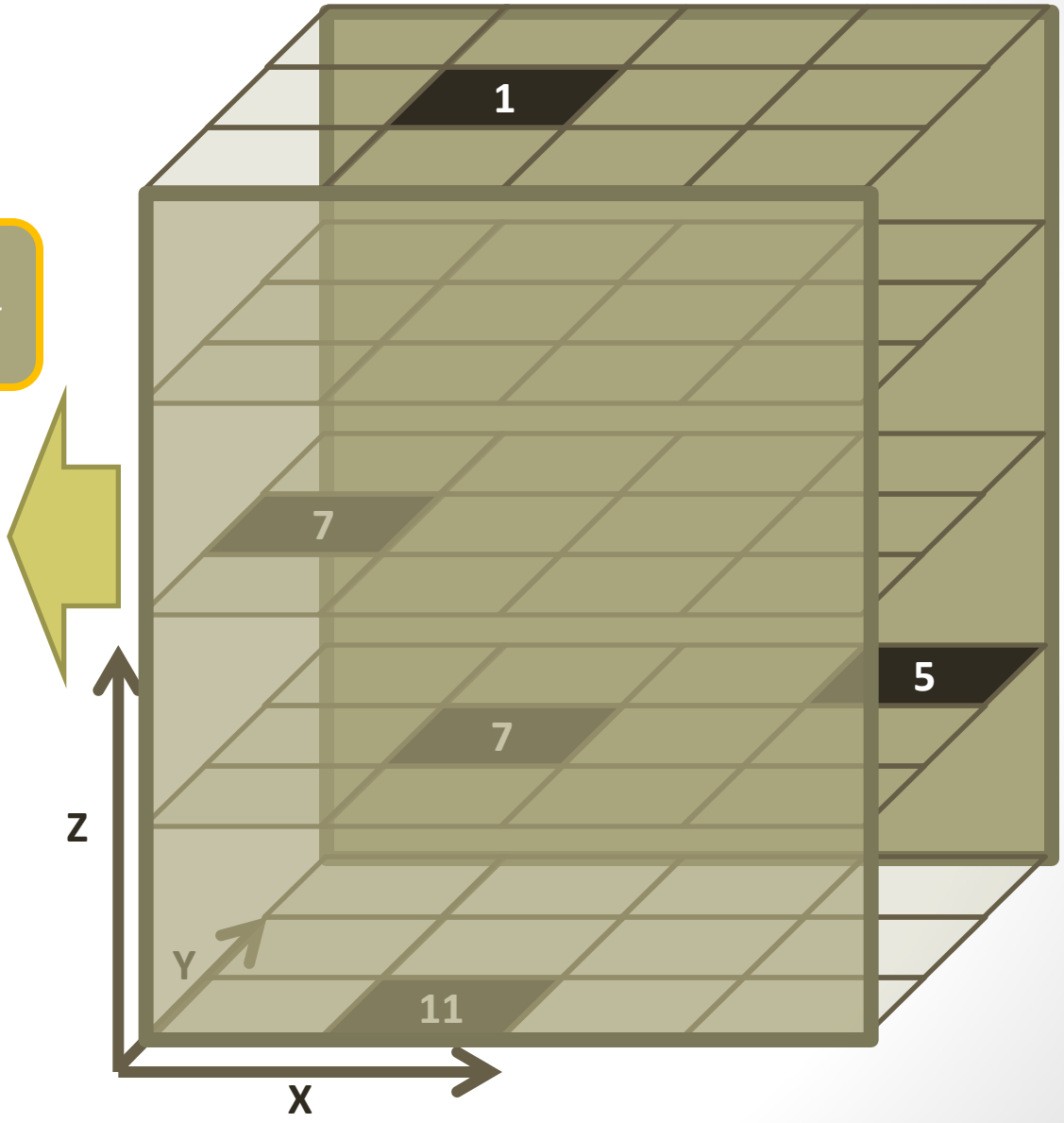
Sparse Matrices



Sparse Matrices

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

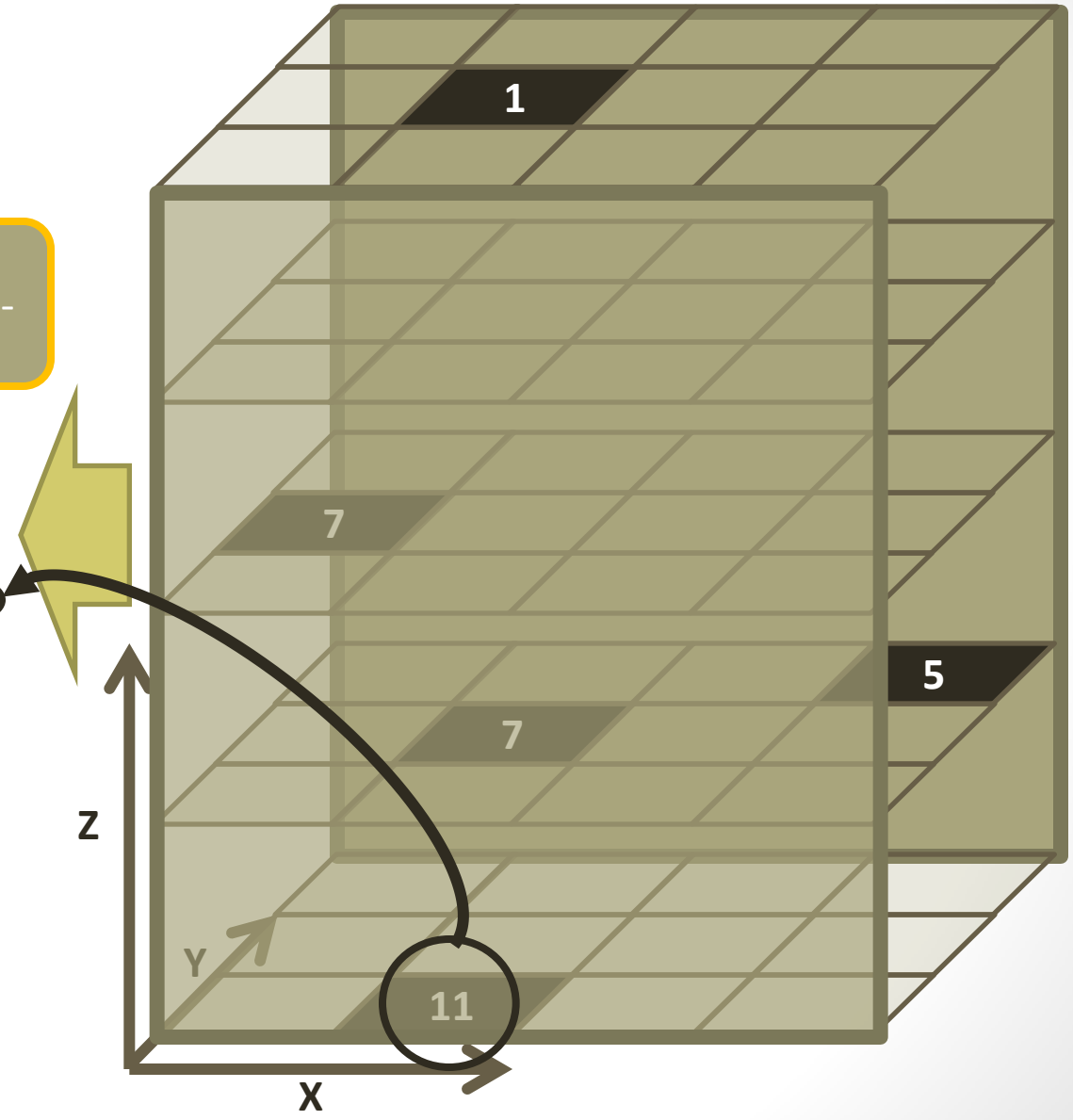
X	Y	Z	V



Sparse Matrices

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

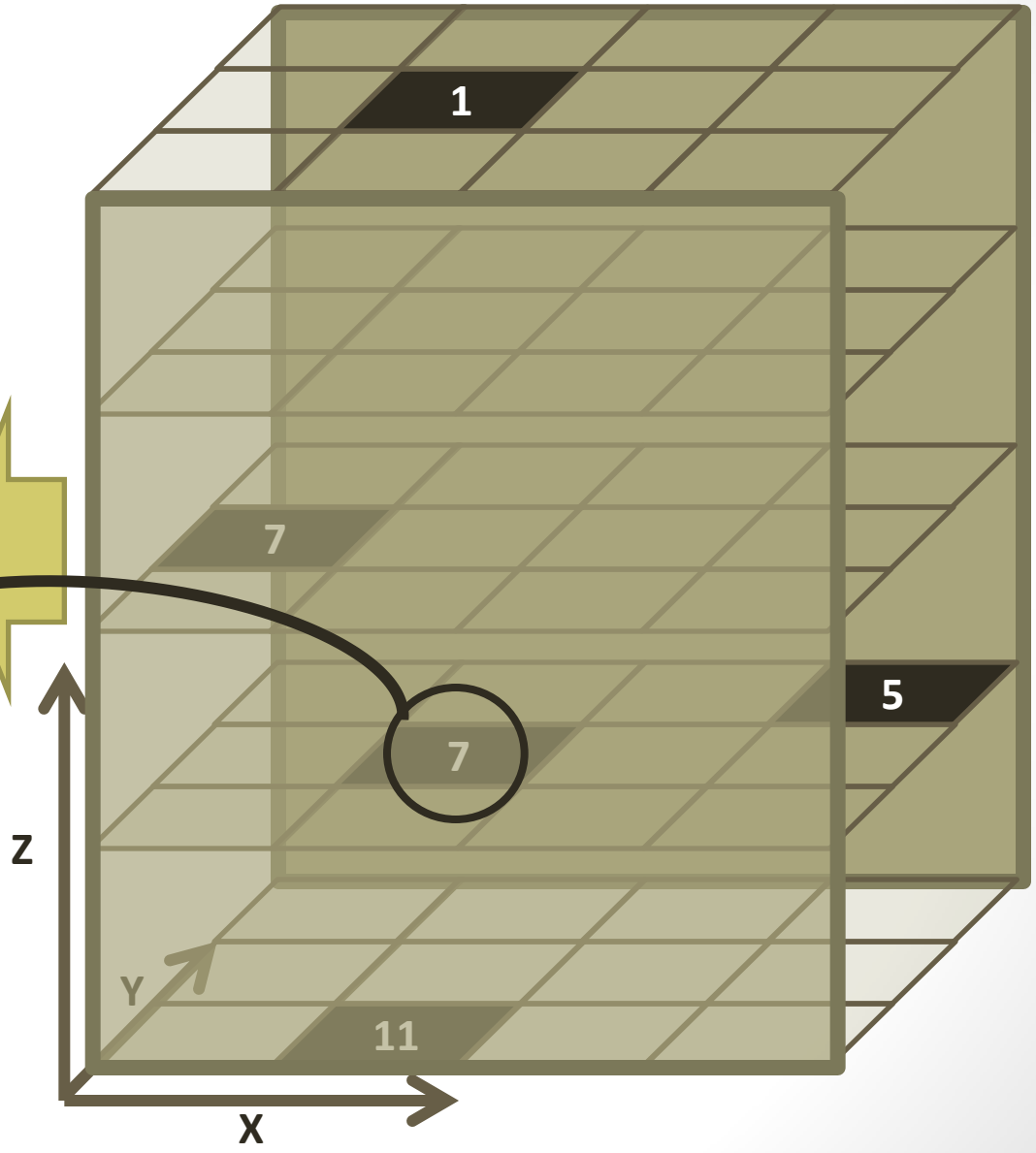
X	Y	Z	V
2	1	1	11



Sparse Matrices

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

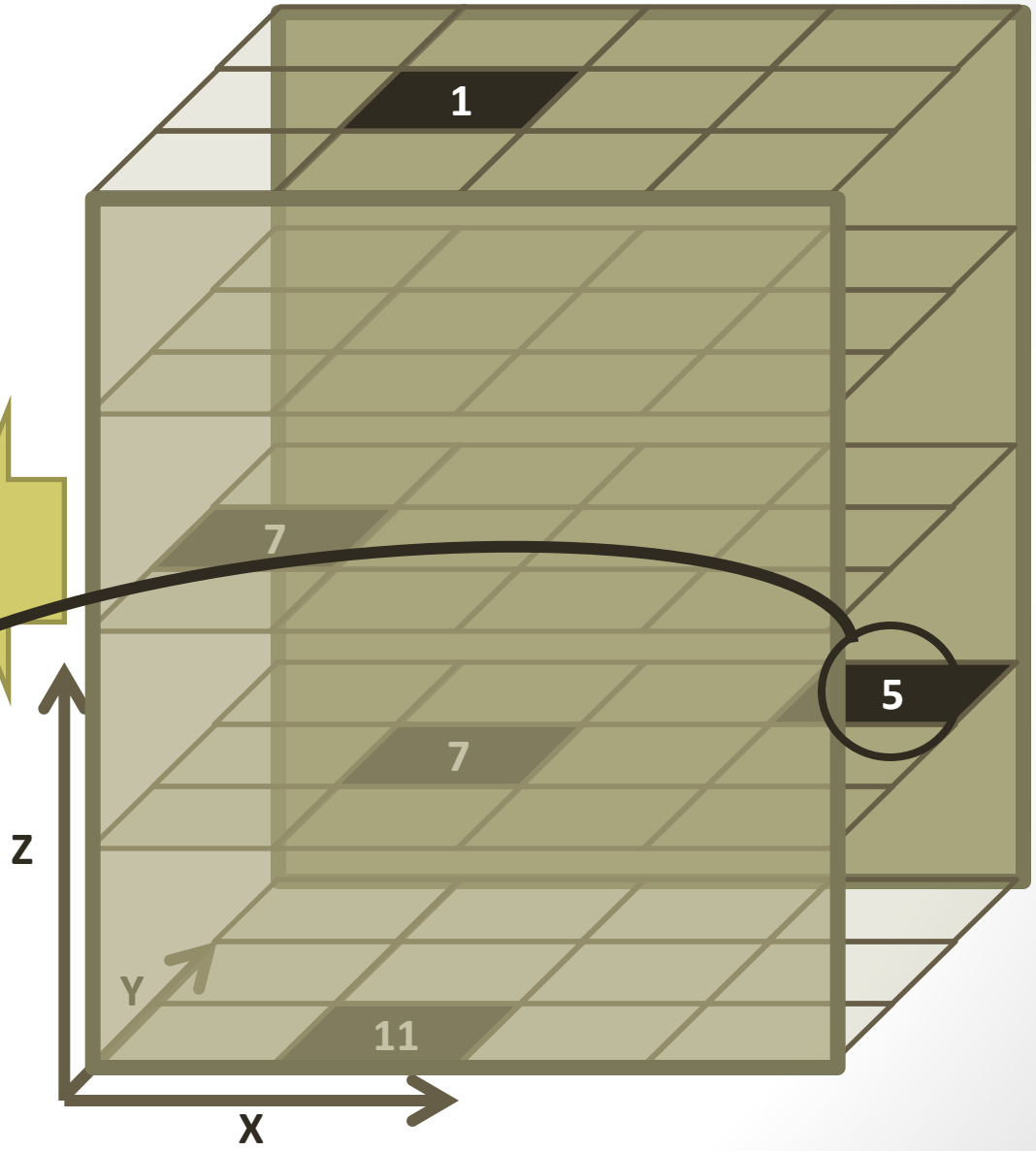
X	Y	Z	V
2	1	1	11
2	2	2	7



Sparse Matrices

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

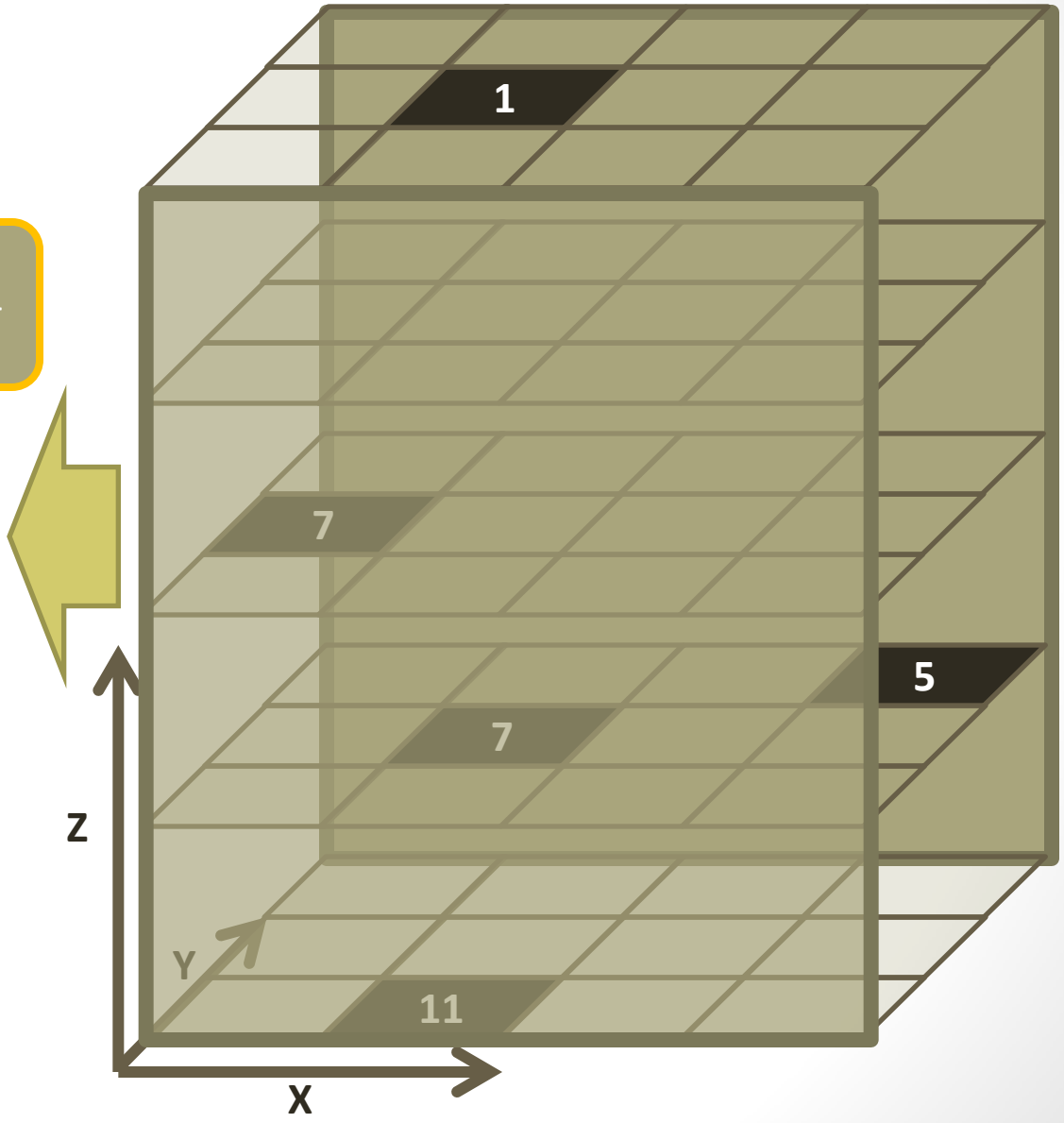
X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5



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

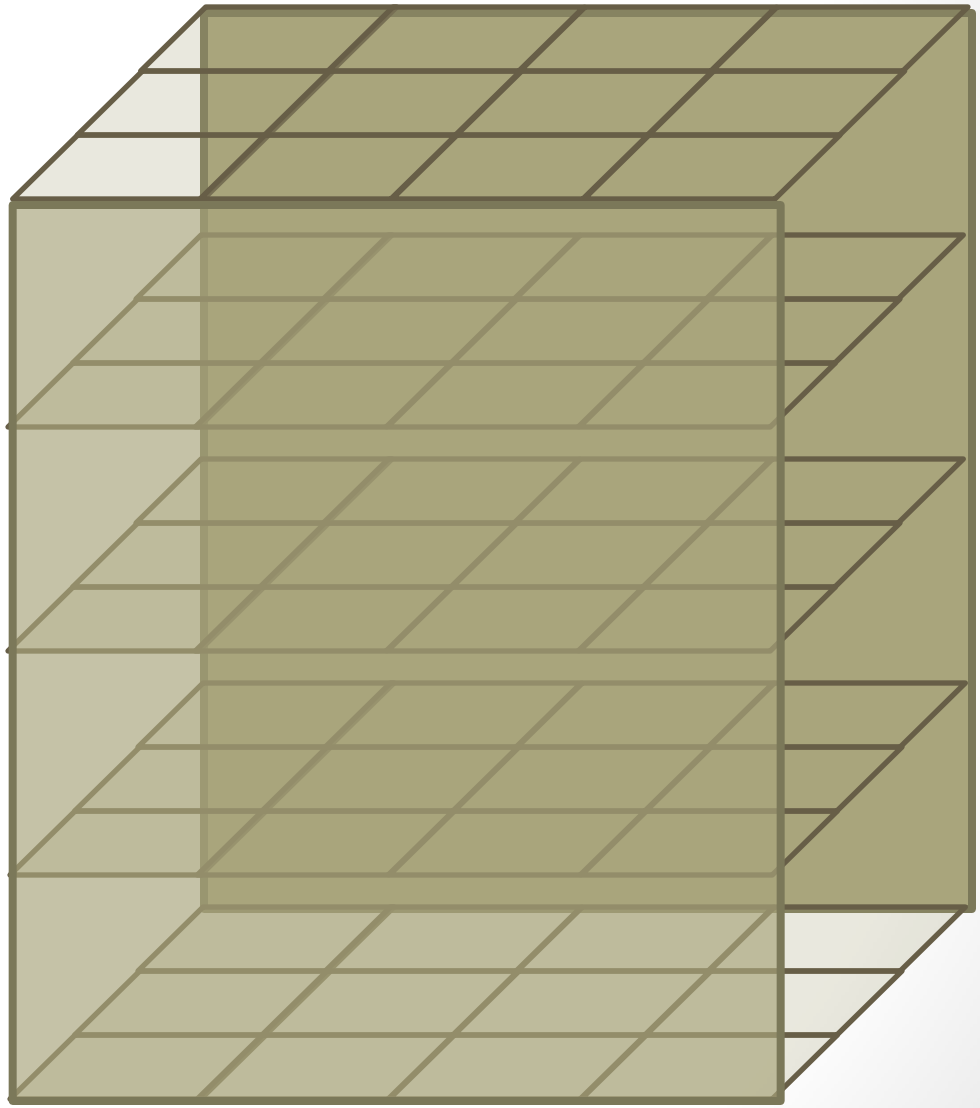
X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1

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

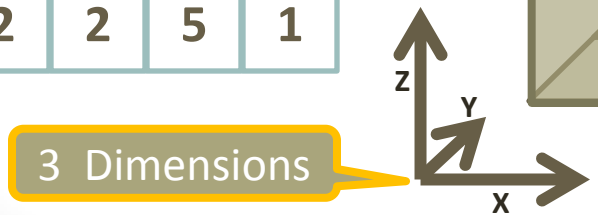


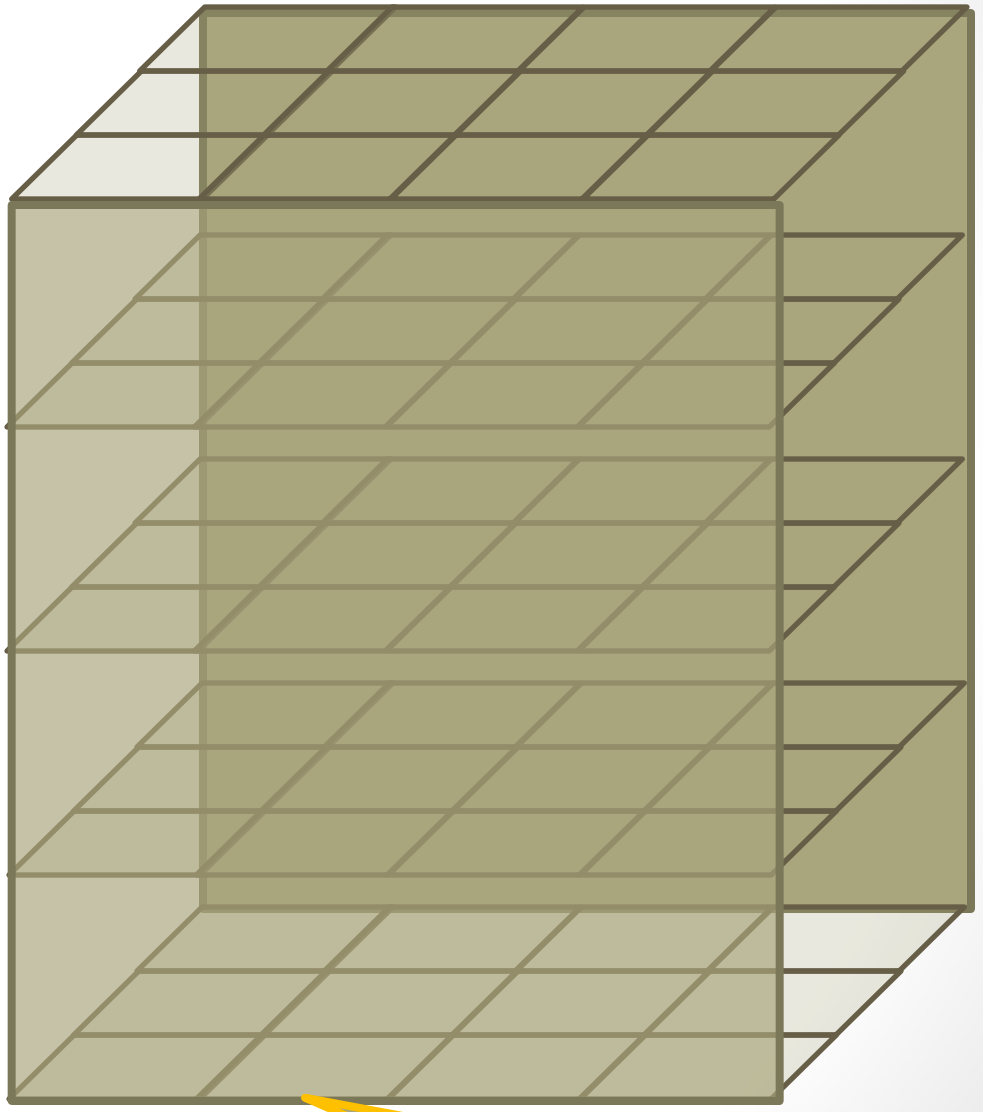
3 Dimensions

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

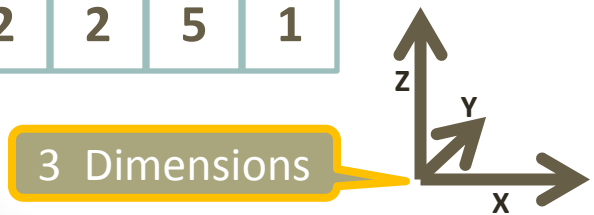




Sparse Matrices

This table represents points in 4-Dimensional Space.

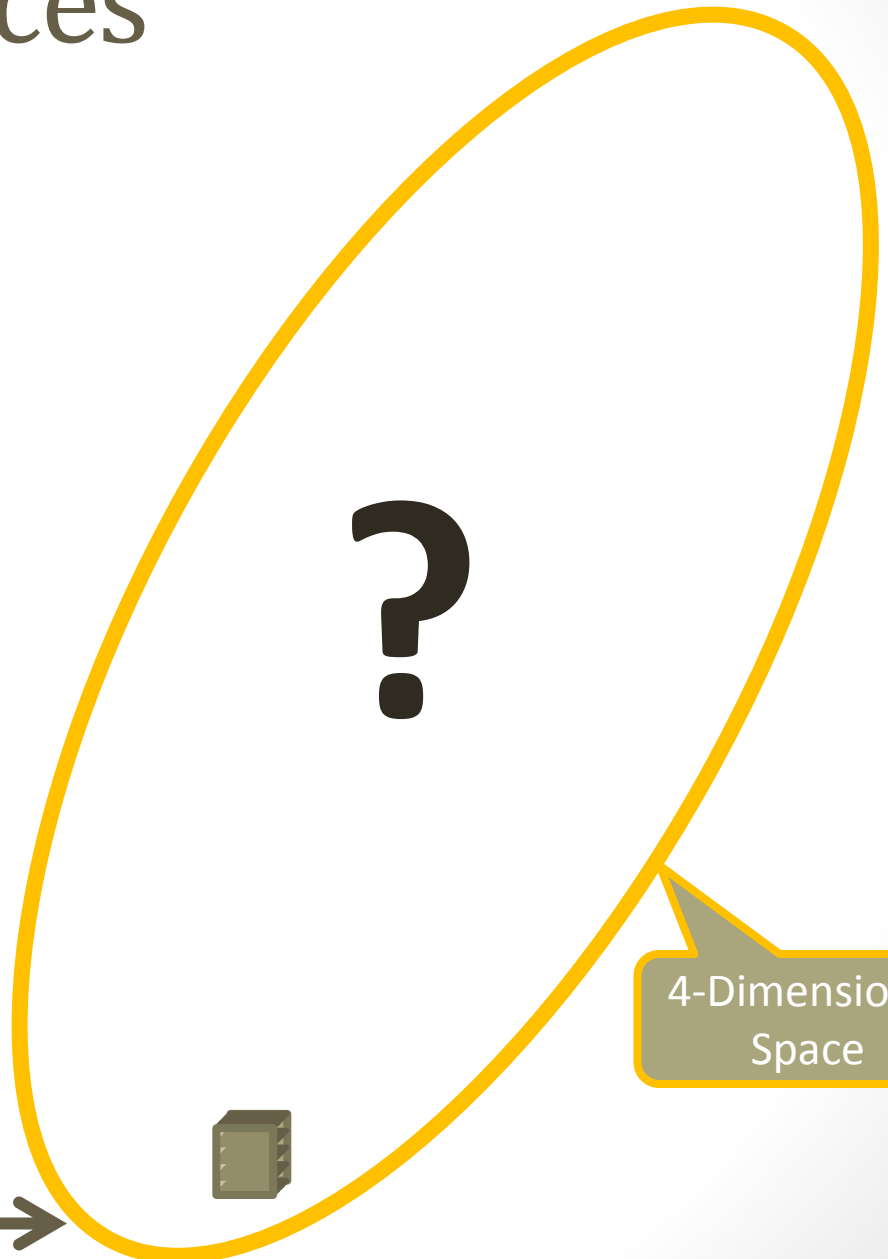
X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1



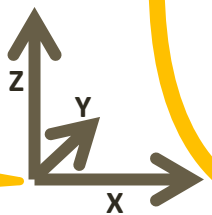
Sparse Matrices

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



4-Dimensional Space

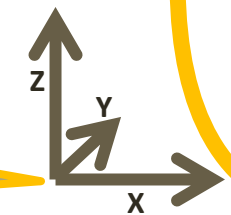


3 Dimensions

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



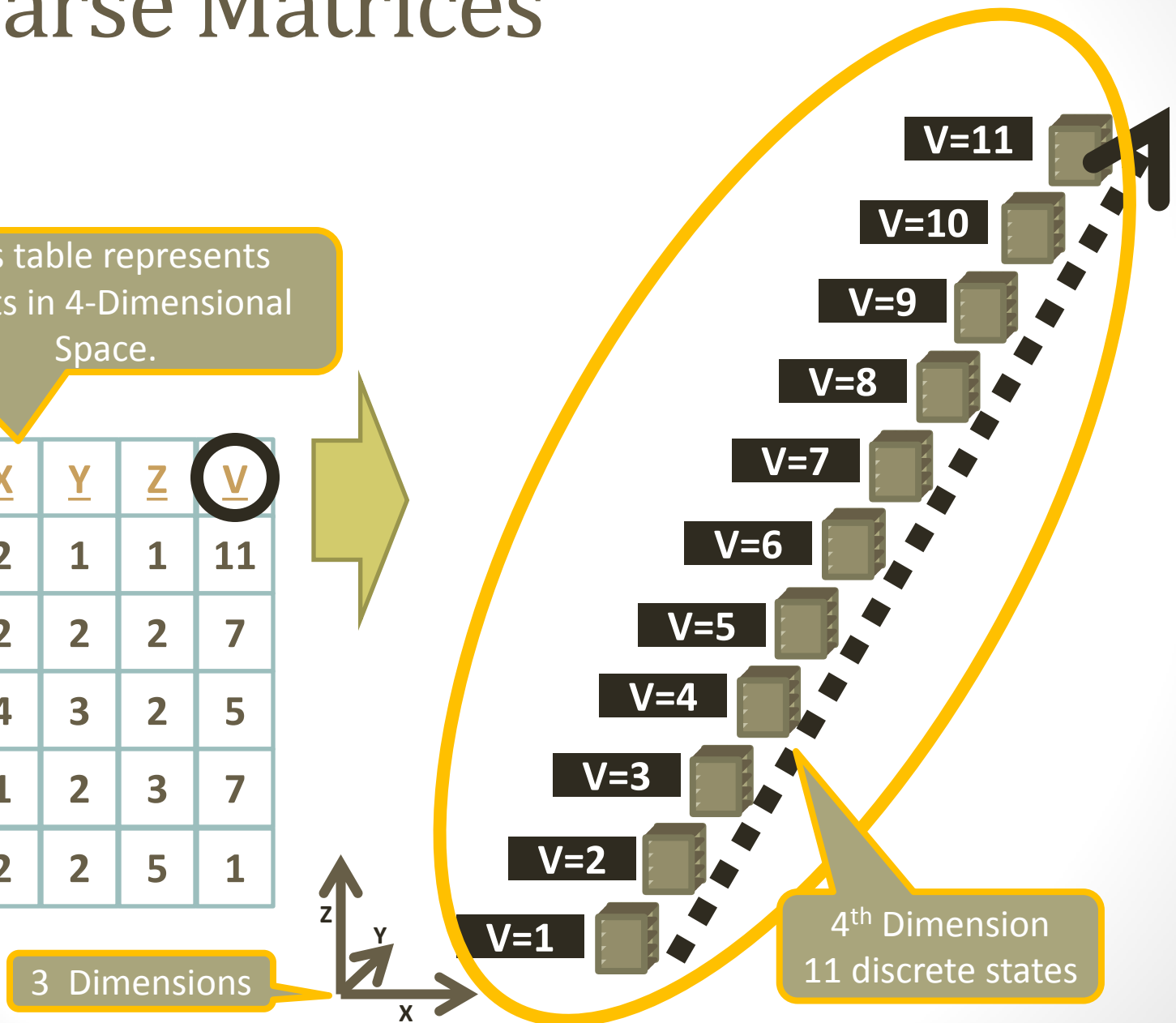
3 Dimensions

4th Dimension

Sparse Matrices

This table represents points in 4-Dimensional Space.

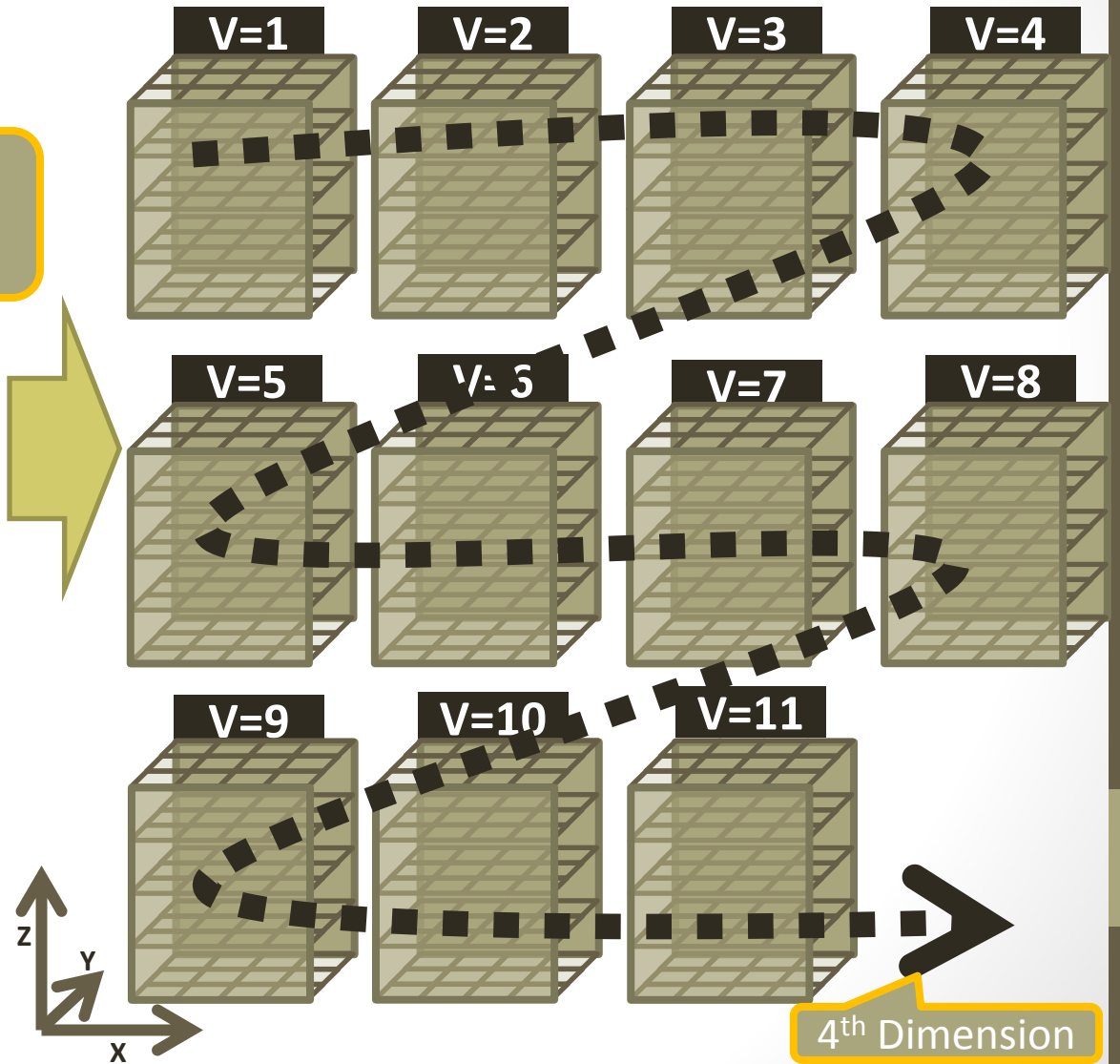
X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1



Sparse Matrices

This table represents points in 4-Dimensional Space.

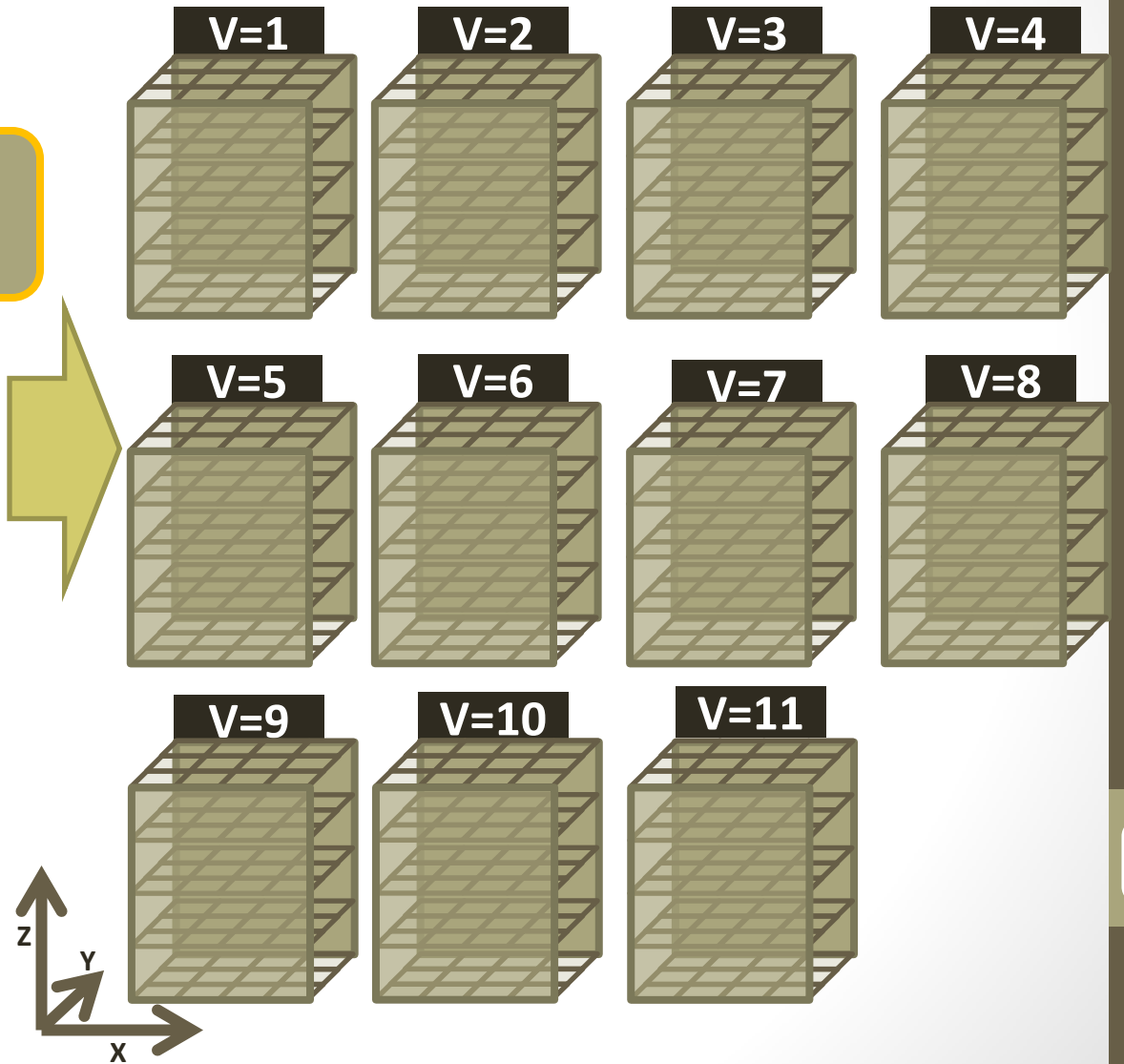
X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1



Sparse Matrices

This table represents points in 4-Dimensional Space.

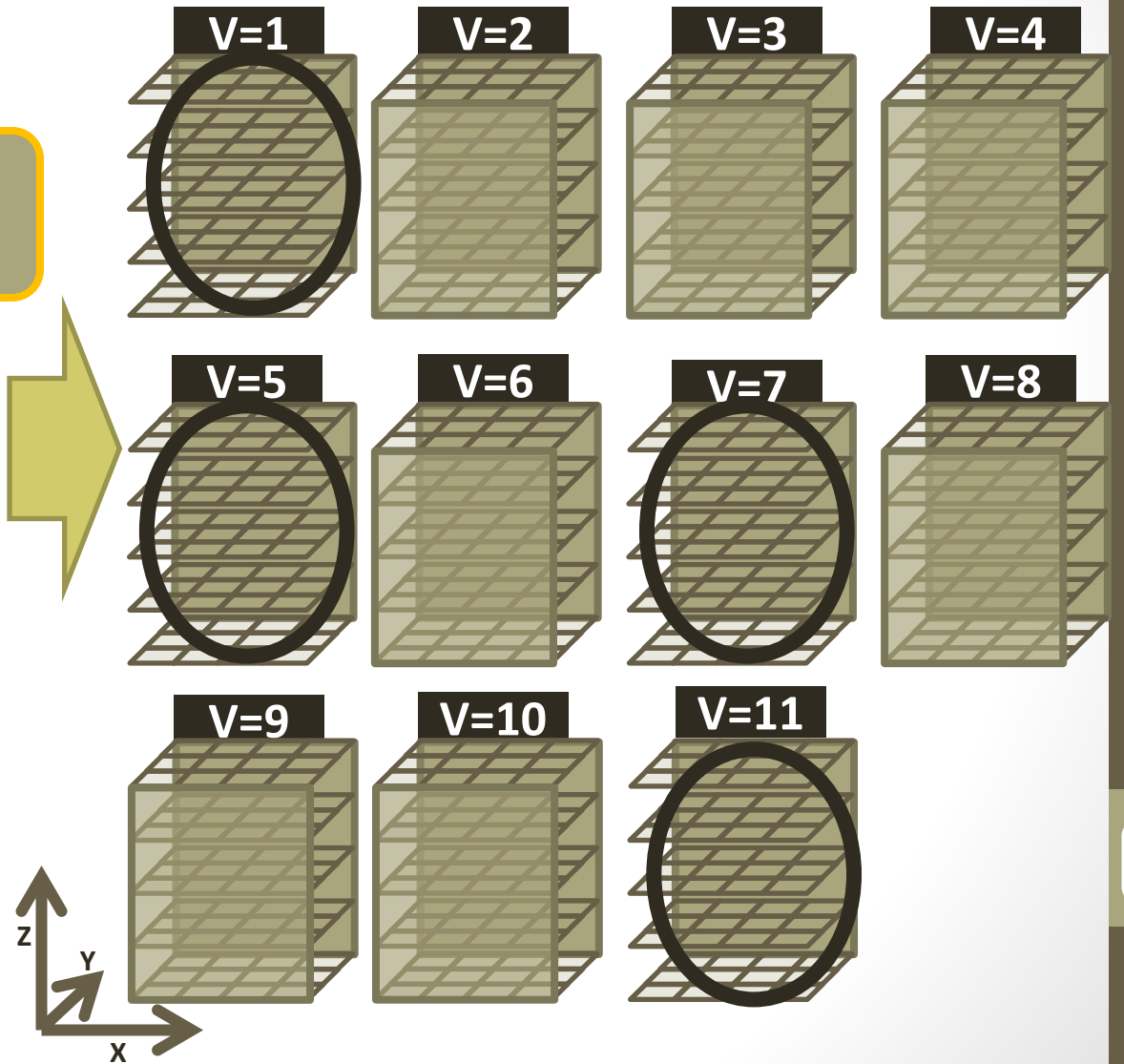
X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1



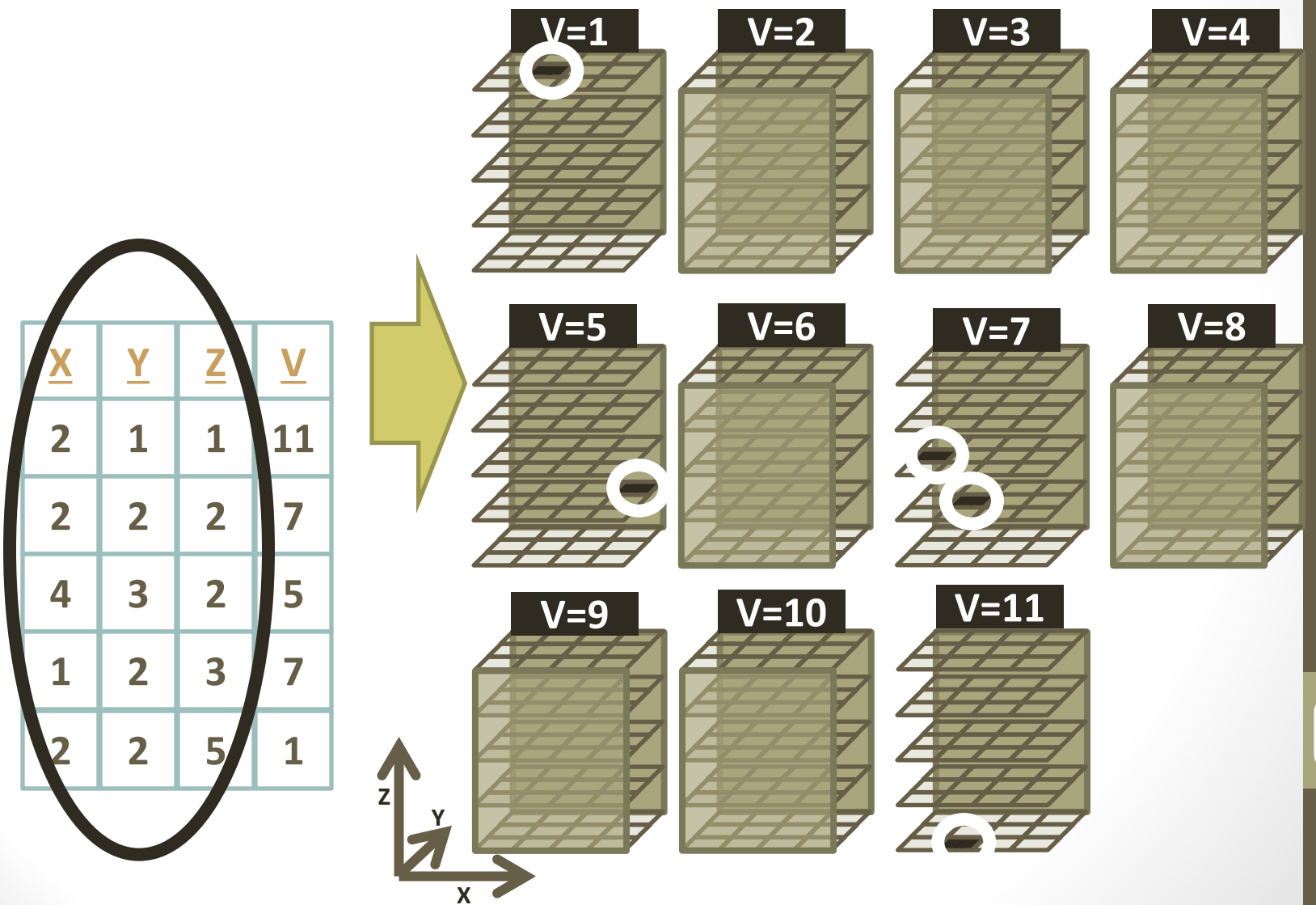
Sparse Matrices

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



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

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

- 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

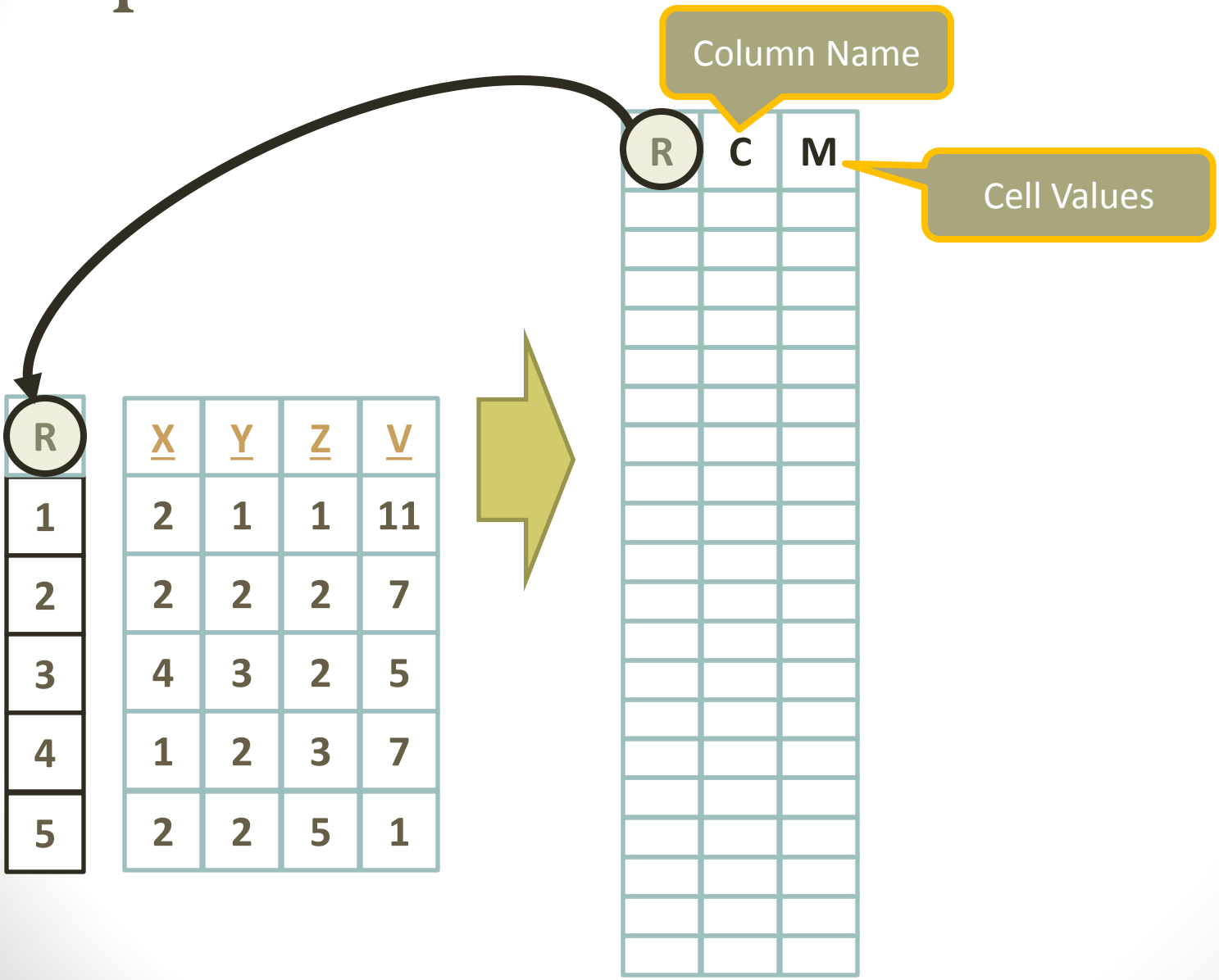
Column Name

Cell Values

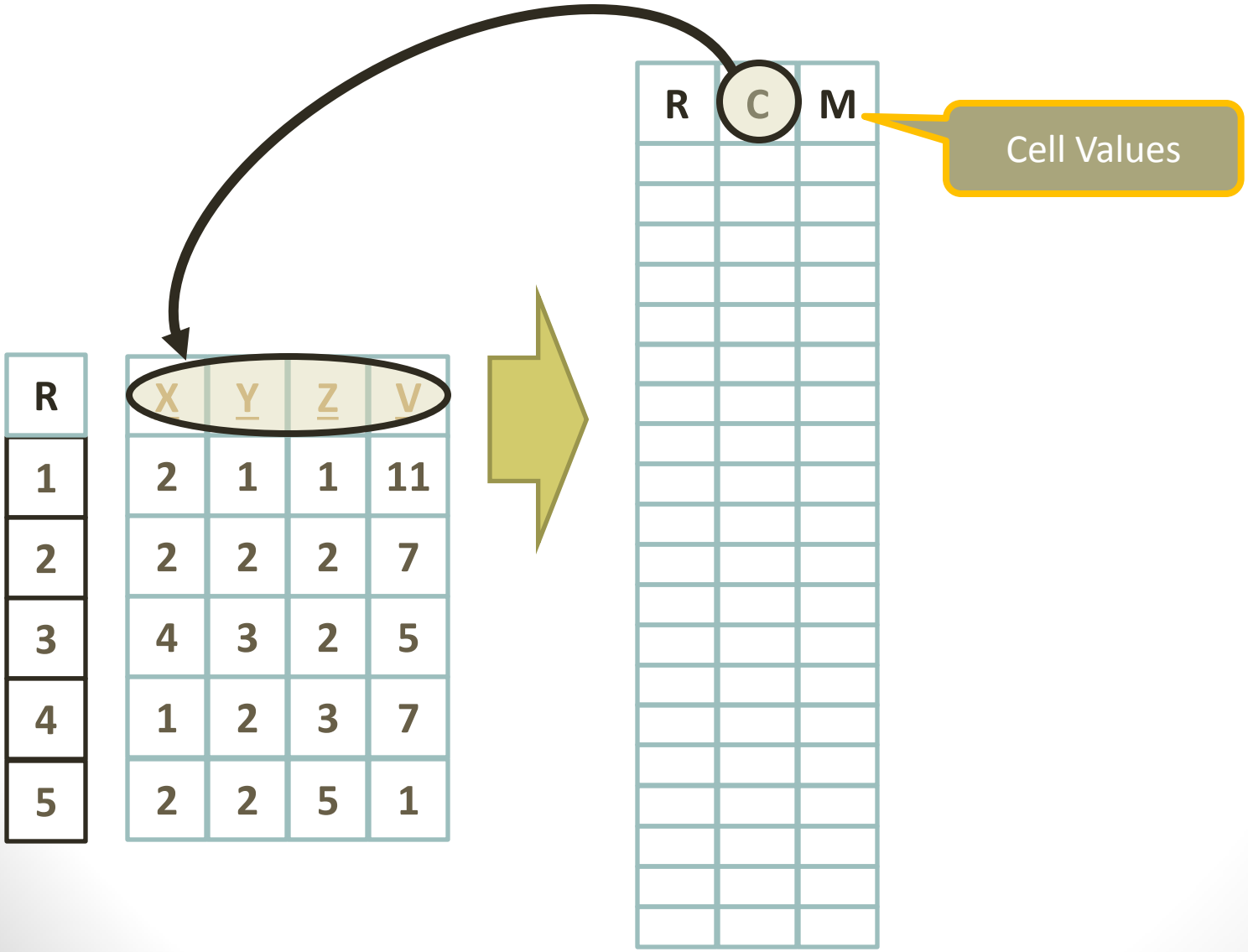
X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1

R	C	M

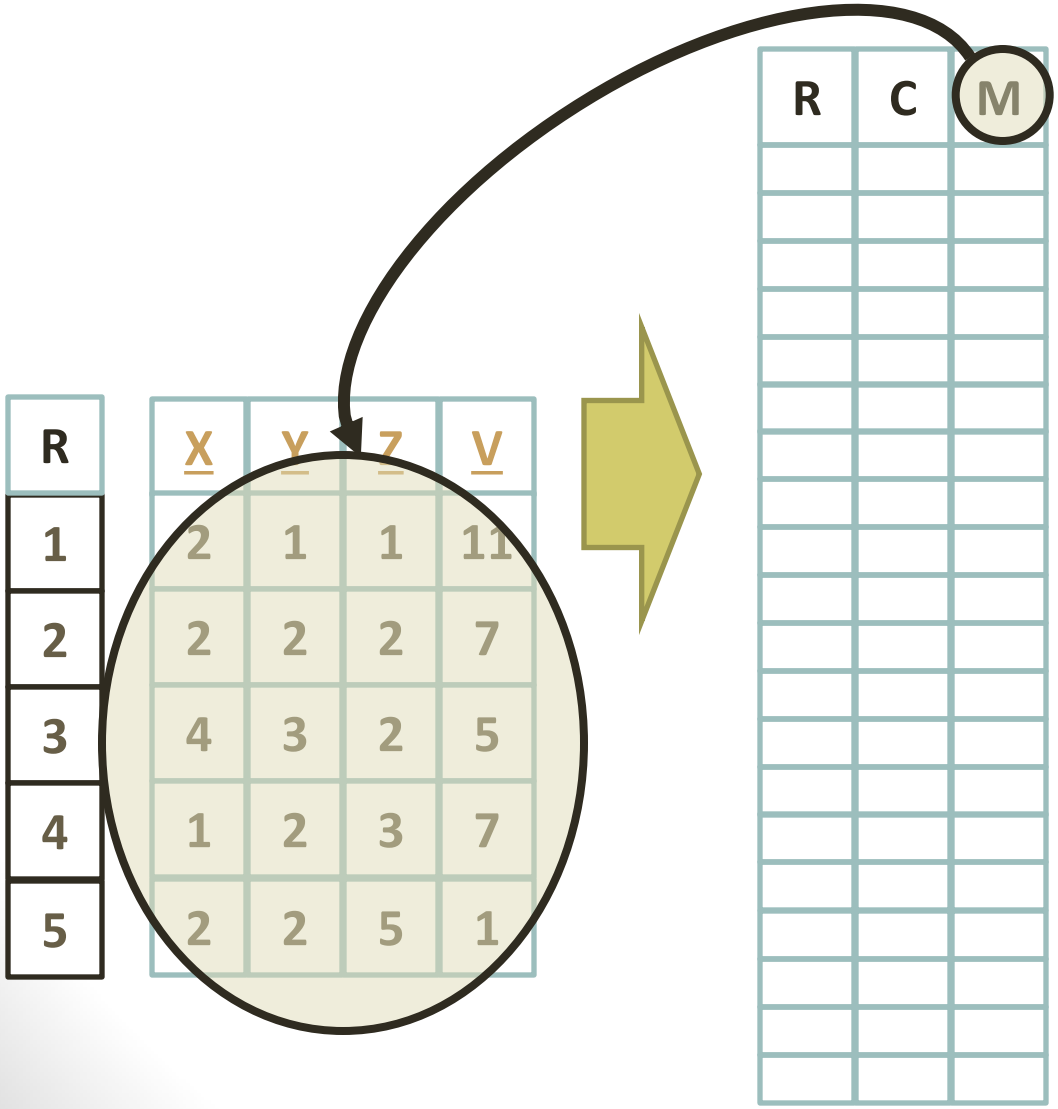
Sparse Matrices: EAV



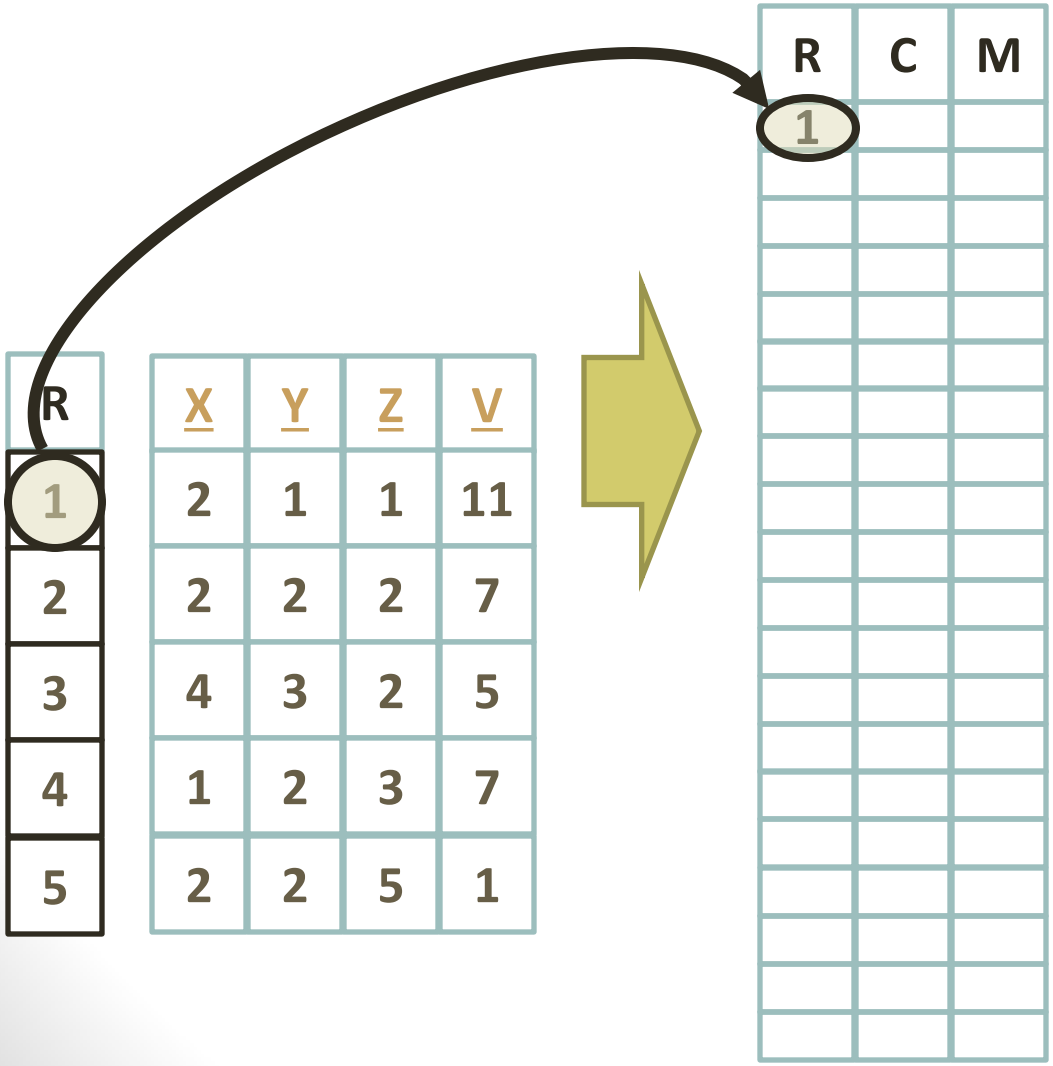
Sparse Matrices: EAV



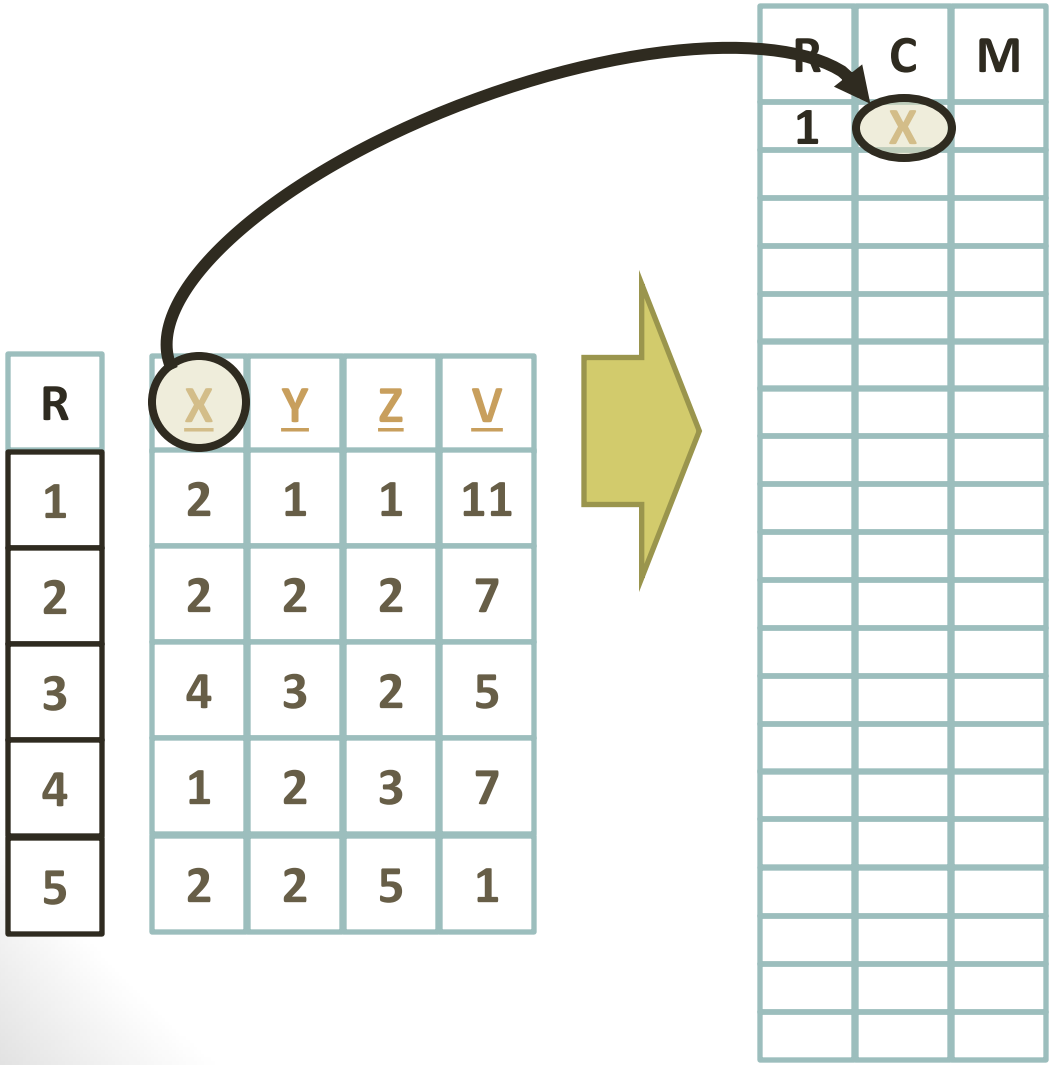
Sparse Matrices: EAV



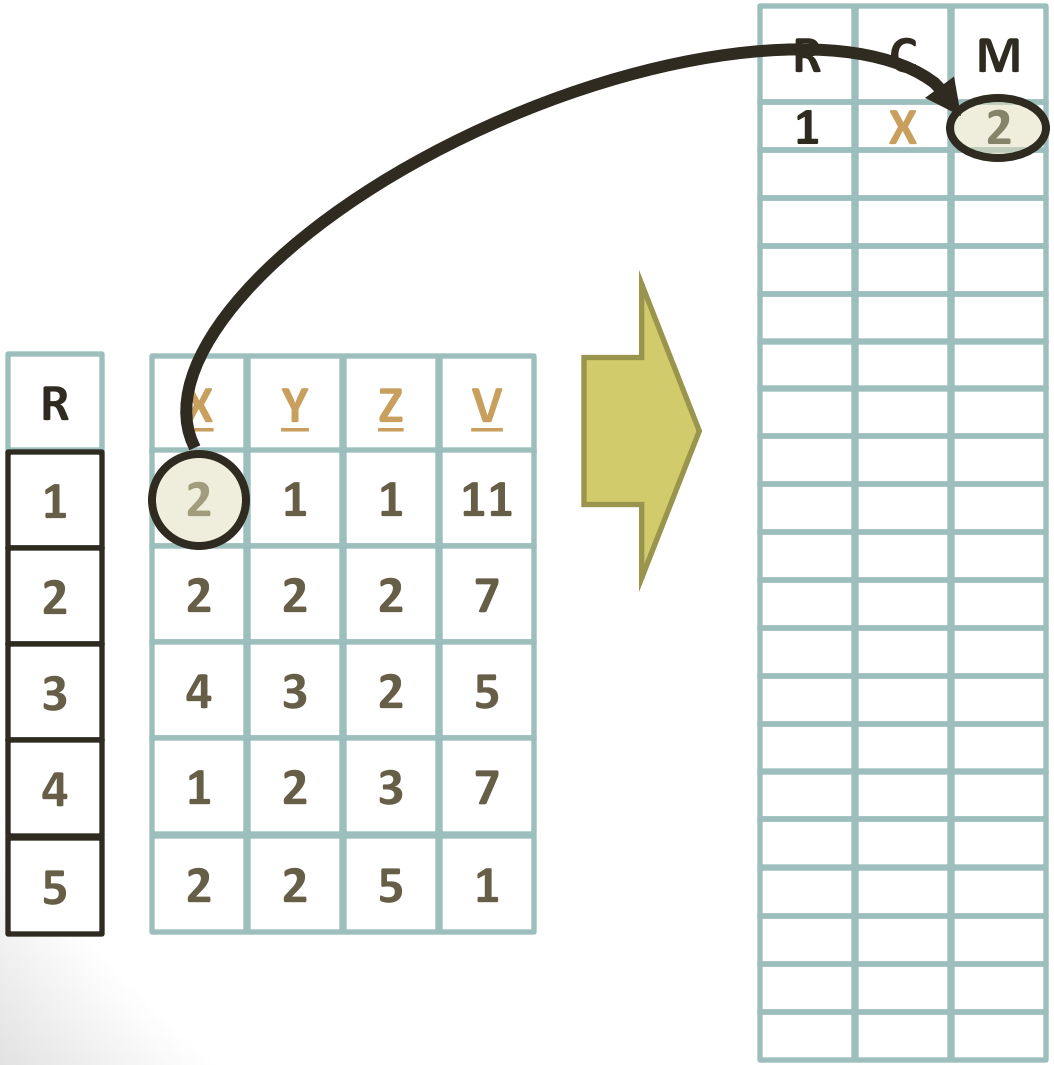
Sparse Matrices: EAV



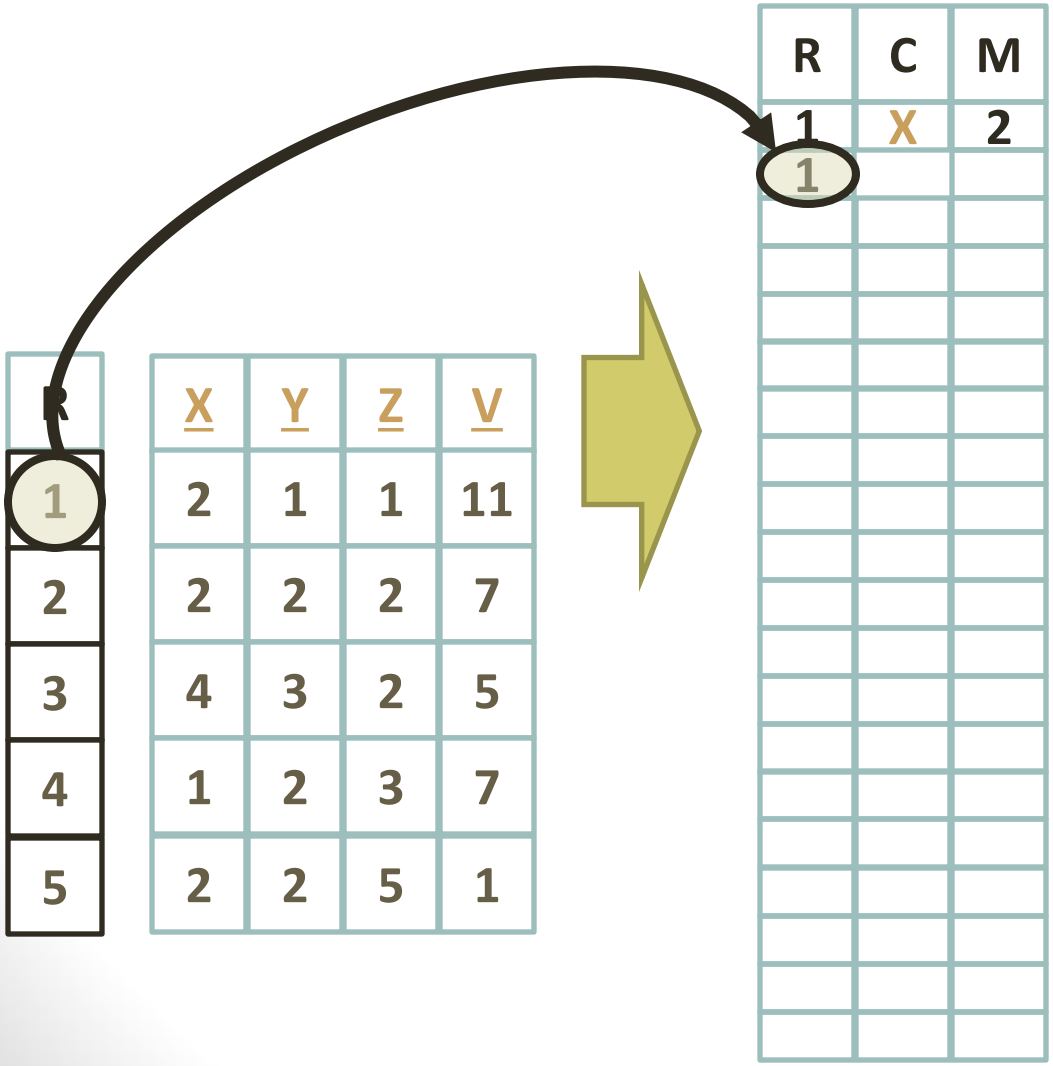
Sparse Matrices: EAV



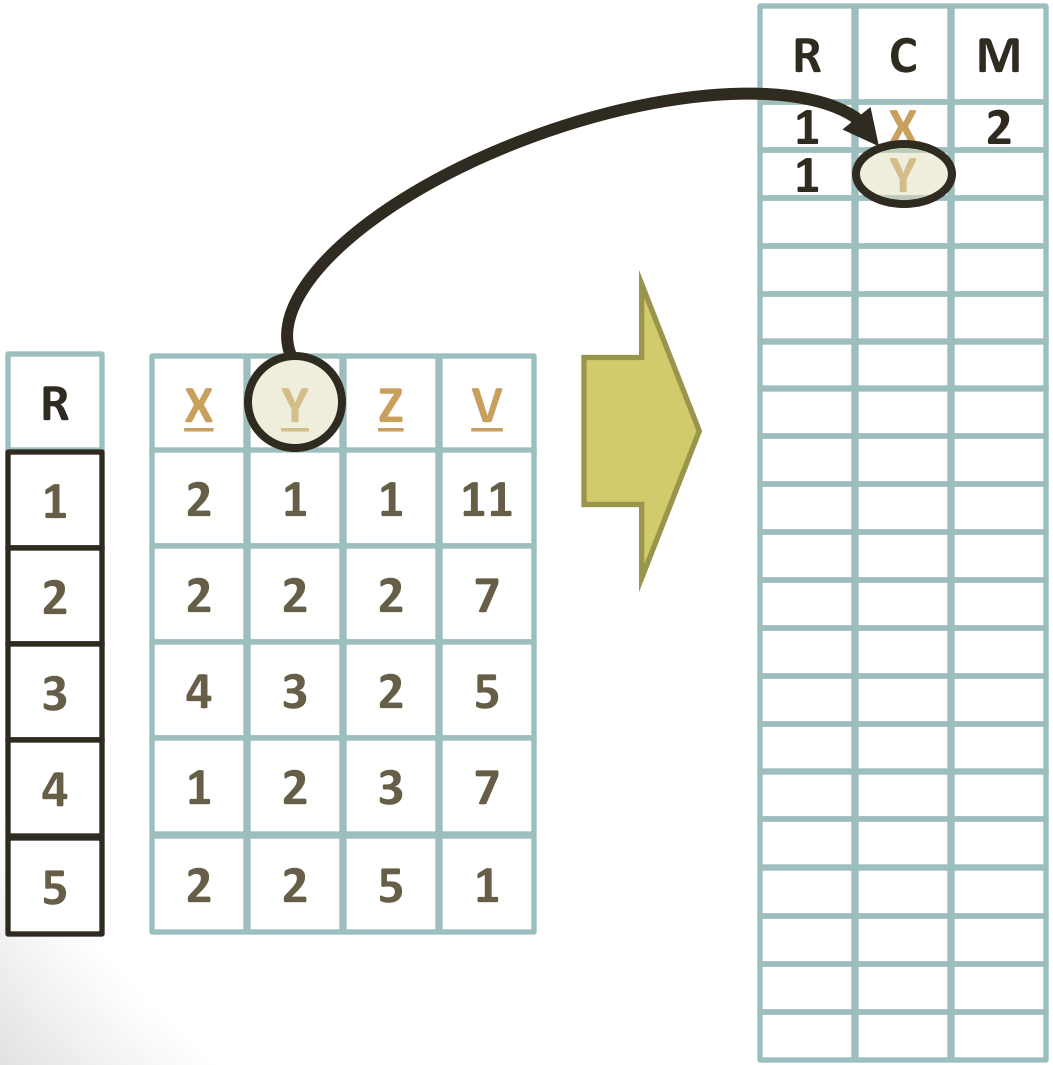
Sparse Matrices: EAV



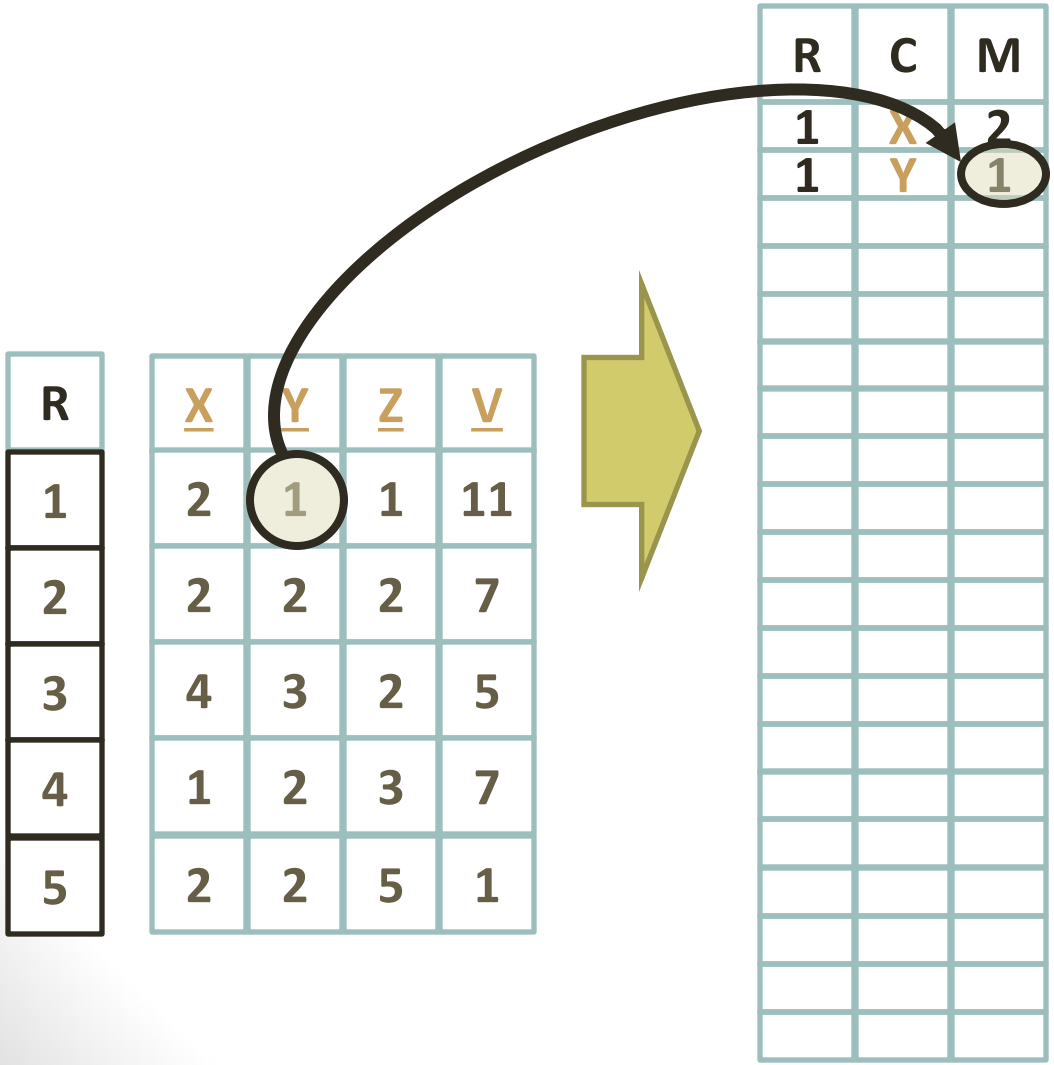
Sparse Matrices: EAV



Sparse Matrices: EAV



Sparse Matrices: EAV



Sparse Matrices: EAV

R
1
2
3
4
5

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

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

Column Name

Row ID.

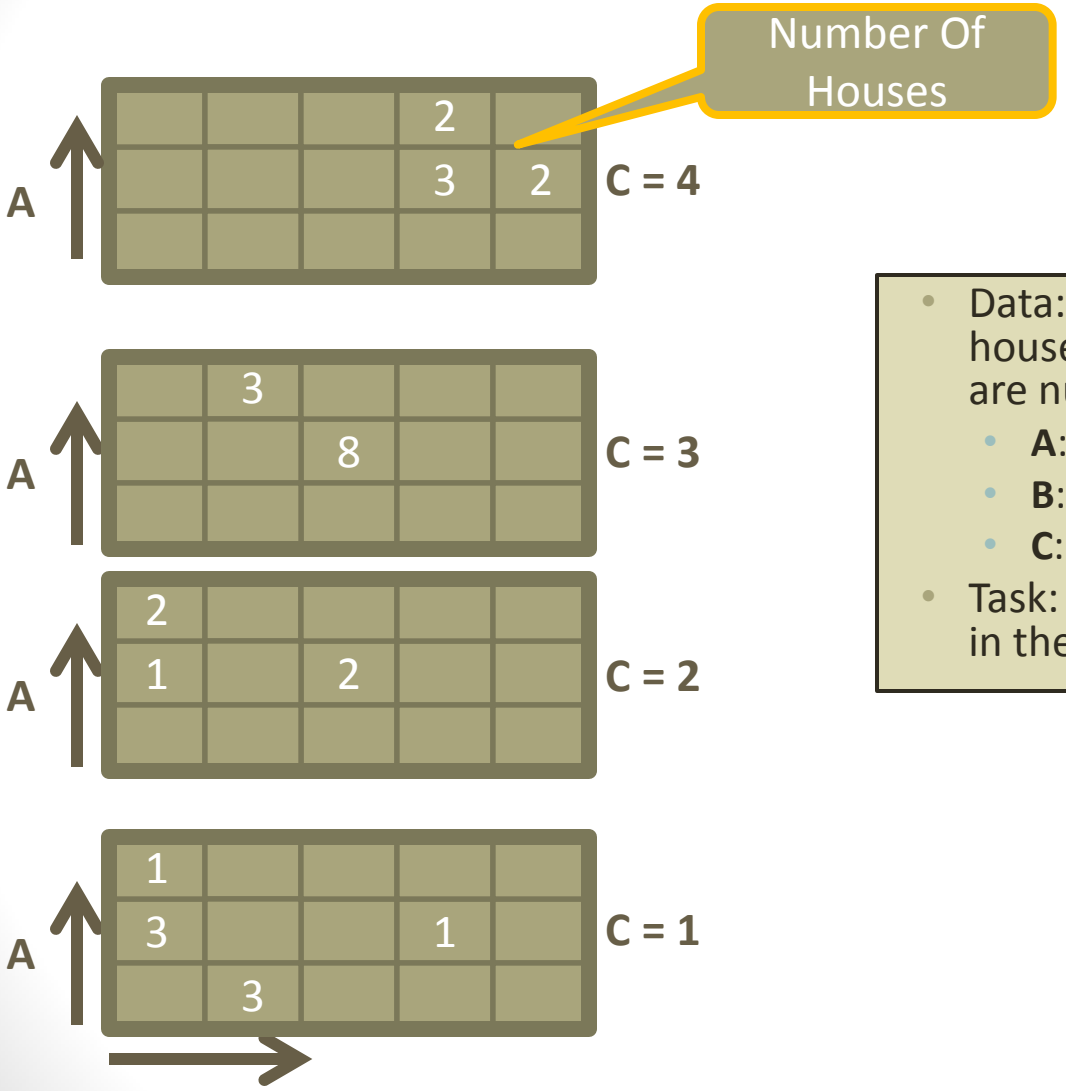
Cell Values

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

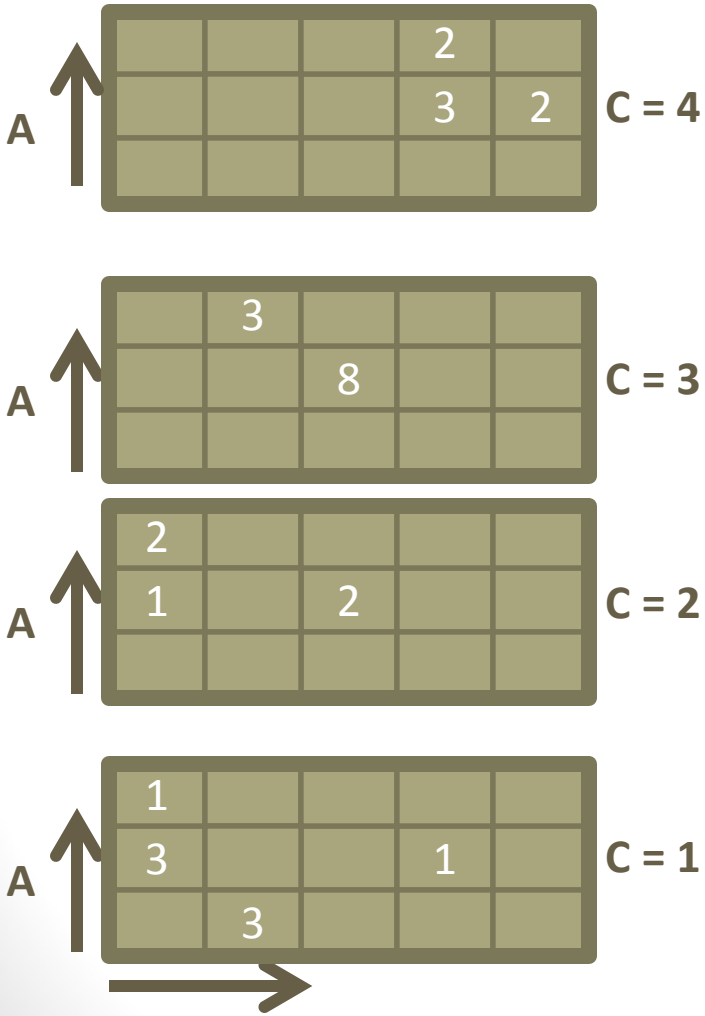
- The questions are presented during the quiz

Sparse Matrices: Exercise (1)

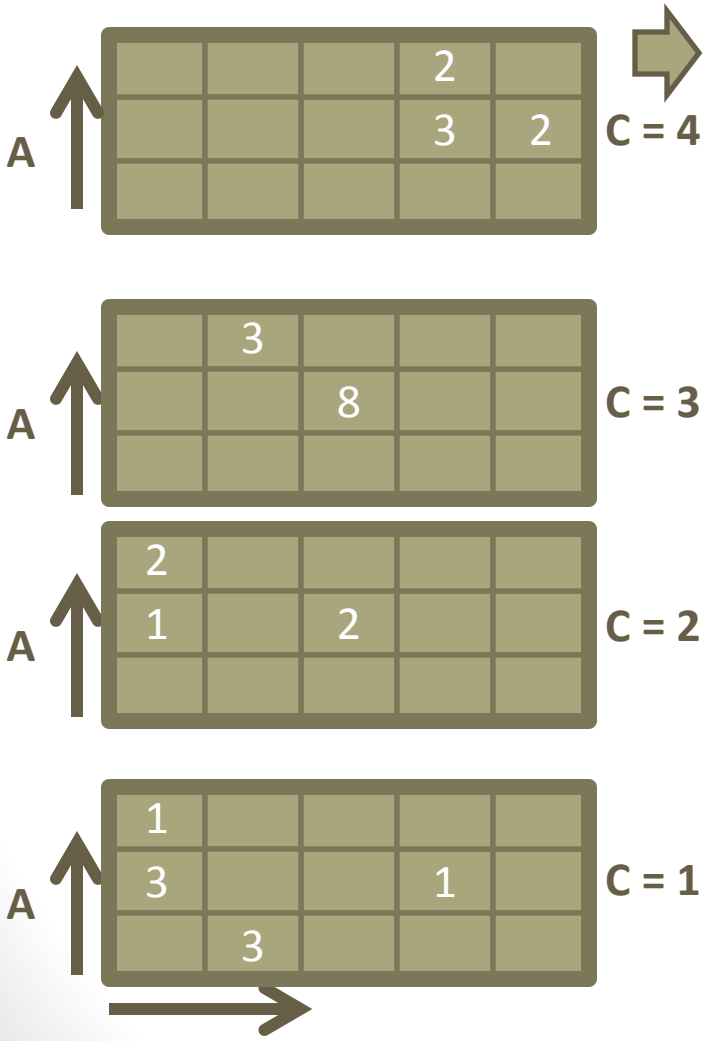


- Data: Real estate survey of single-family houses in downtown Seattle. Cell values are number (**N**) of houses found for sale.
  - **A**: Area in 1000's of square feet
  - **B**: Number of Bathrooms
  - **C**: Cost in $100,000.-
- Task: Create sparse matrices of the type in the previous slide.

Sparse Matrices: Exercise (2)

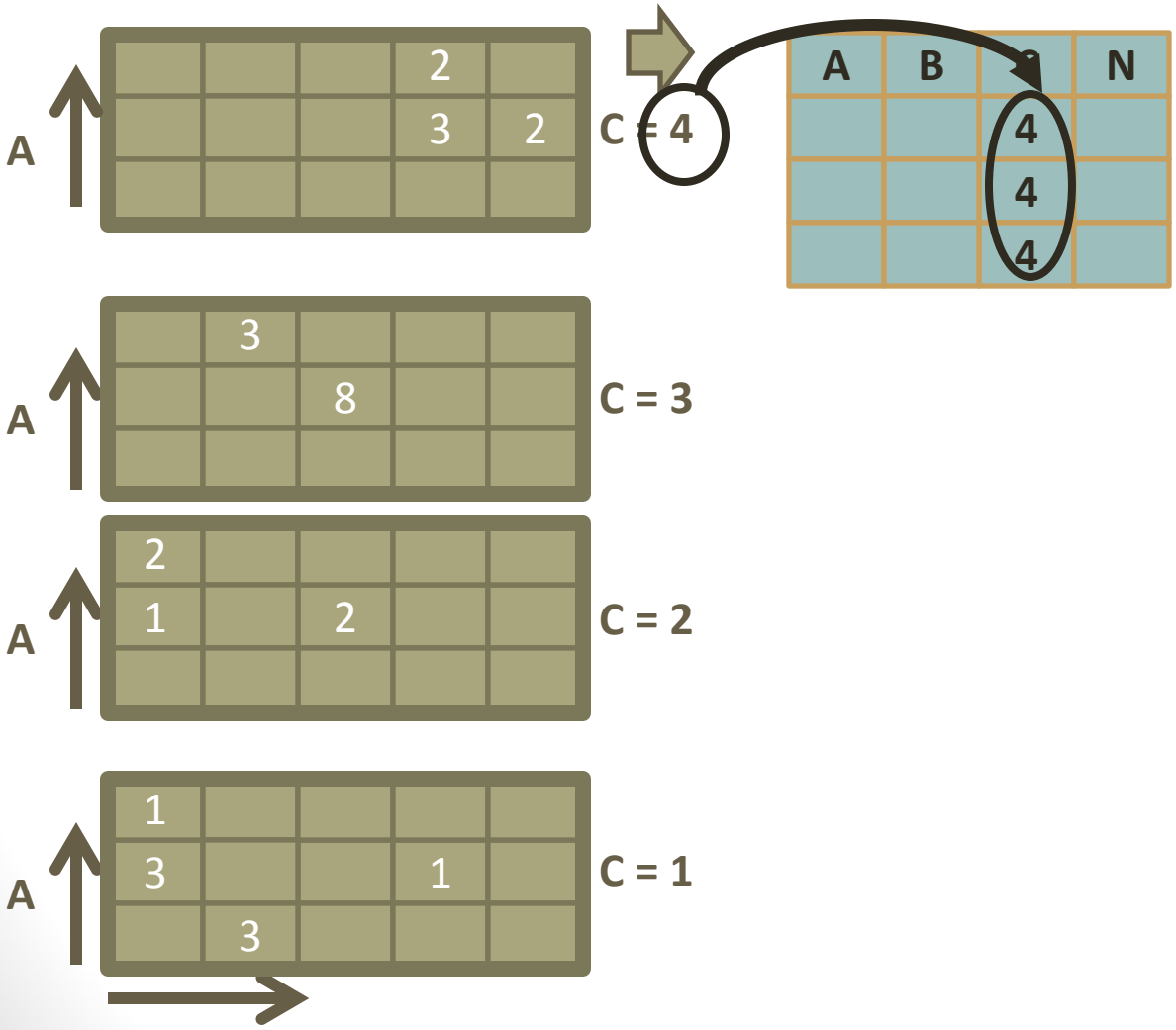


Sparse Matrices: Exercise (3)

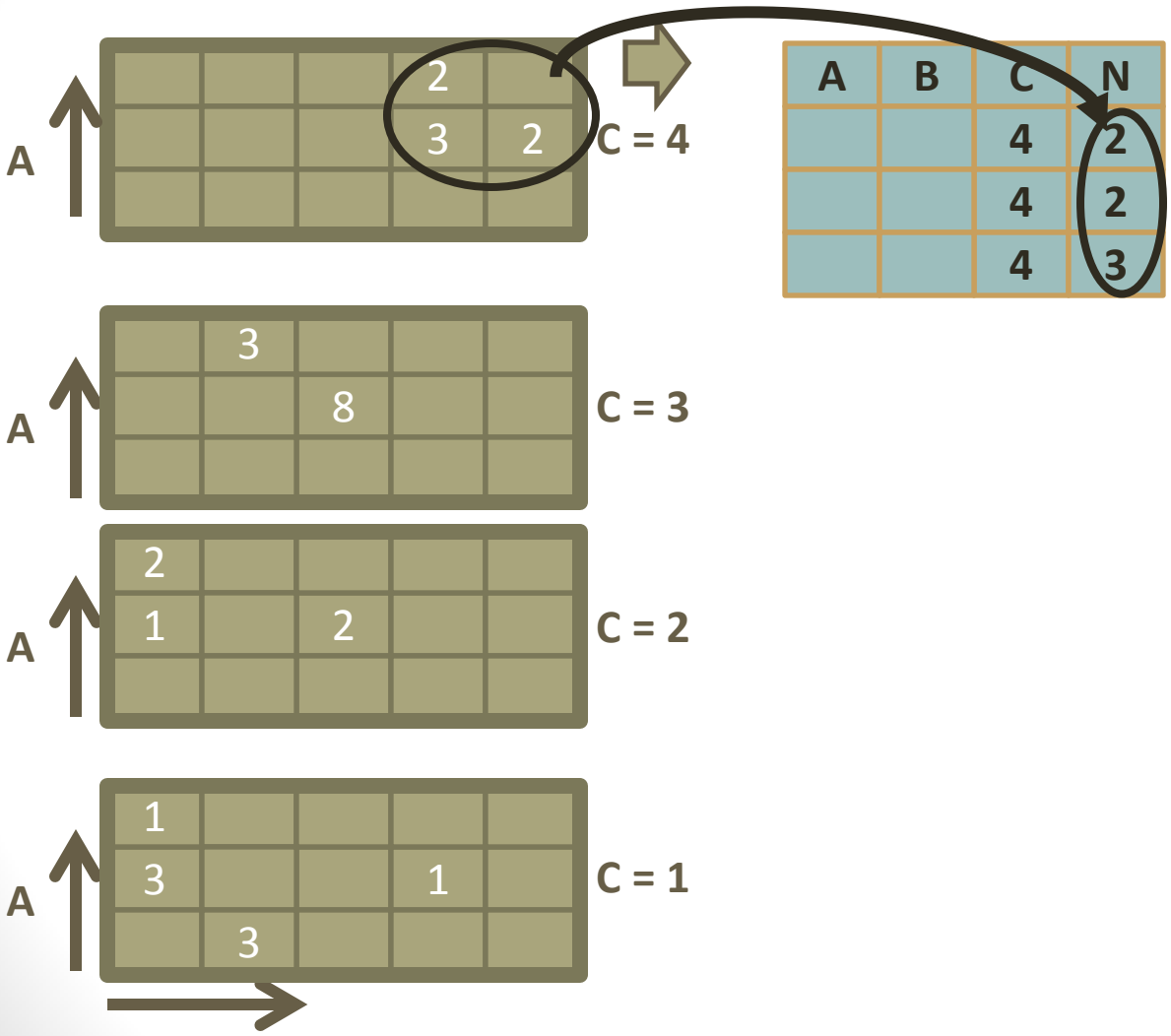


A	B	C	N

Sparse Matrices: Exercise (4)

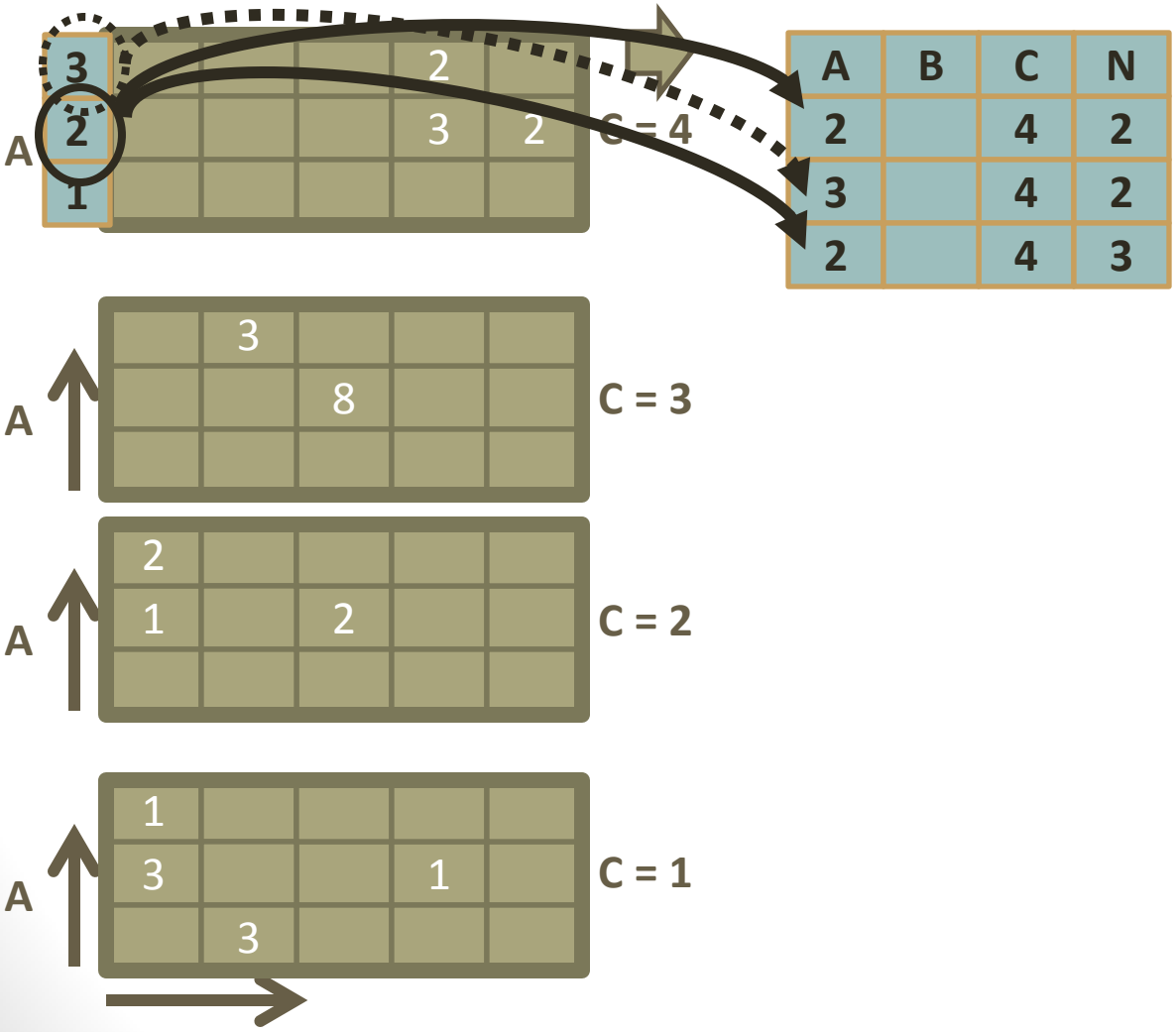


Sparse Matrices: Exercise (5)



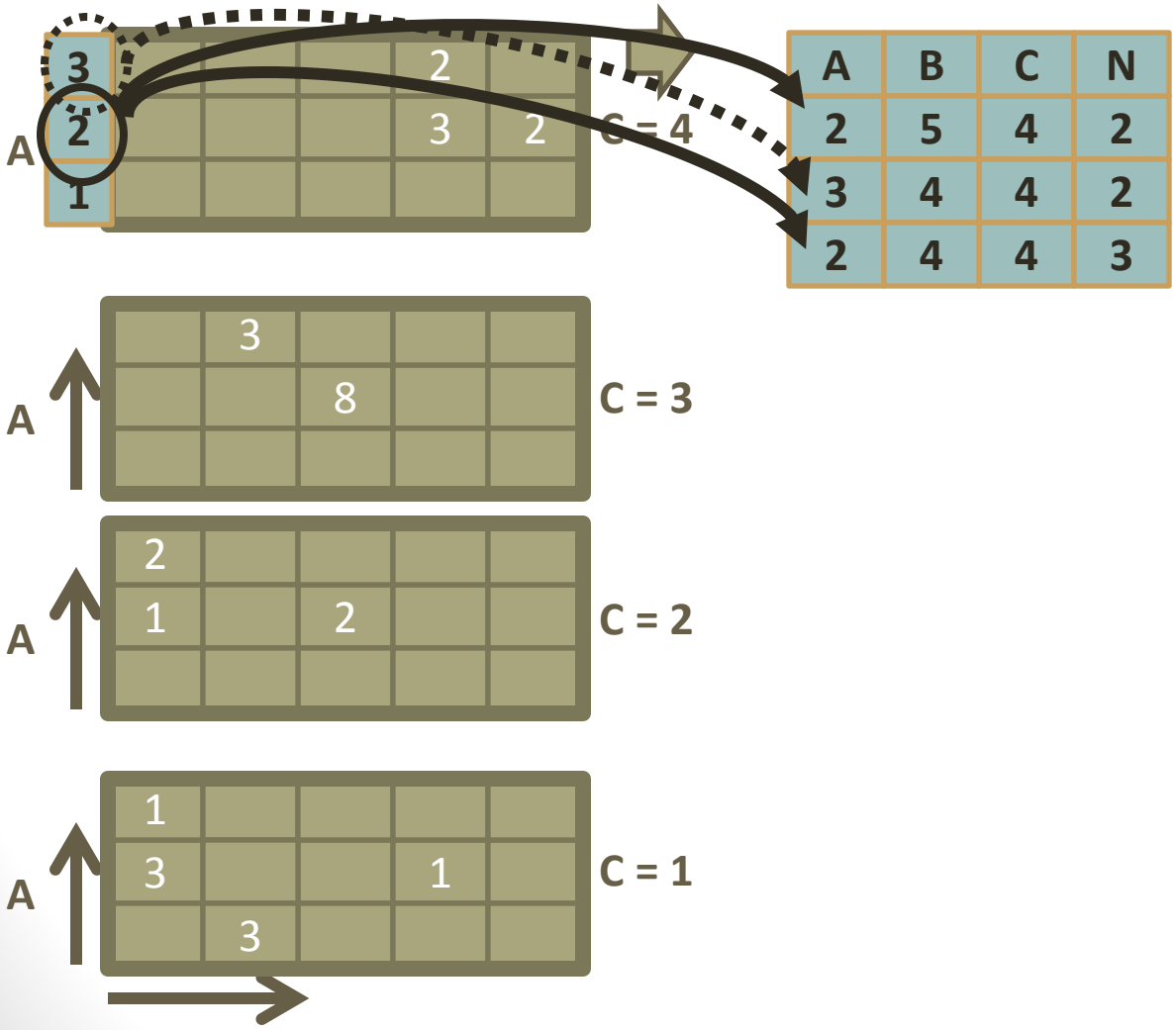
A	B	C	N
		4	2
		4	2
		4	3

Sparse Matrices: Exercise (6)



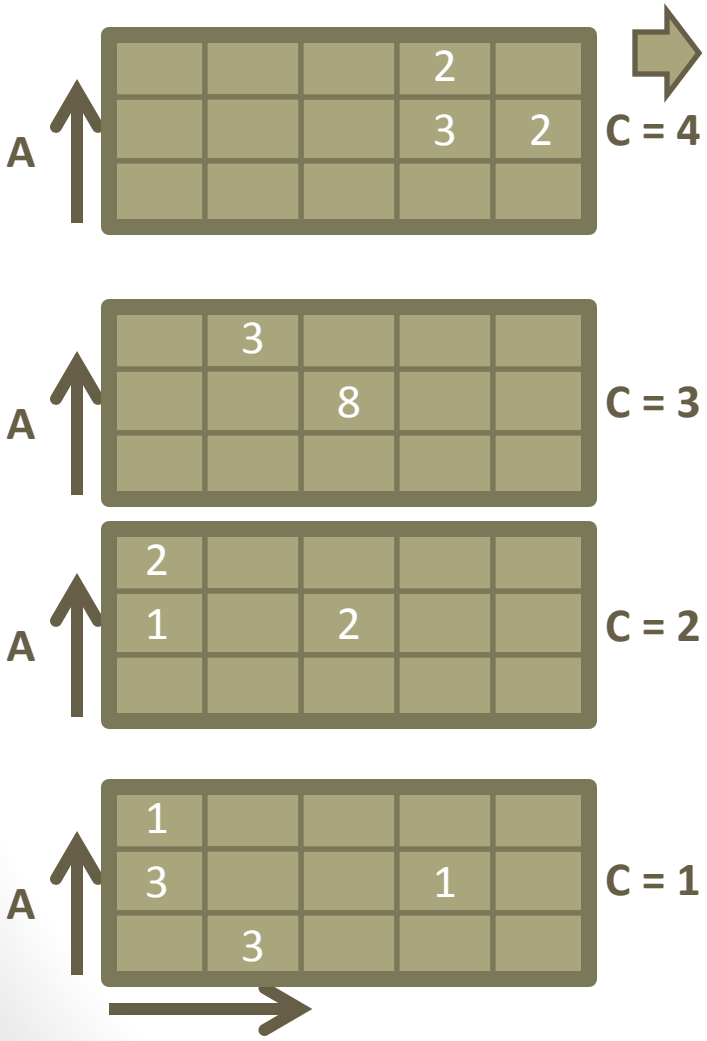
A	B	C	N
2		4	2
3		4	2
2		4	3

Sparse Matrices: Exercise (7)



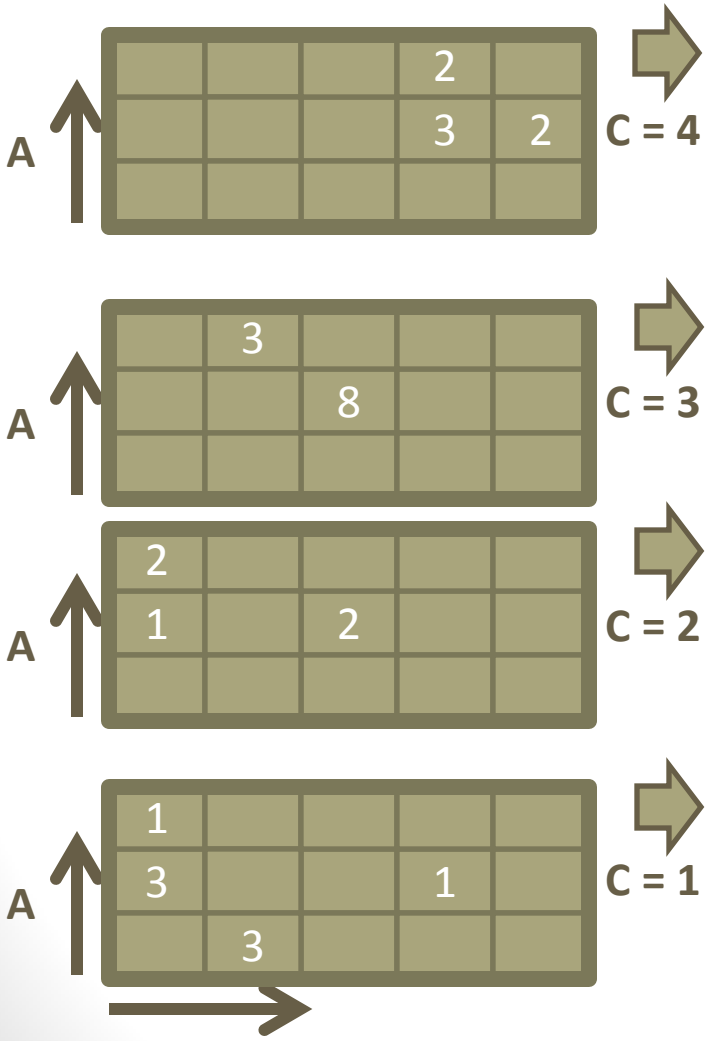
A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3

Sparse Matrices: Exercise (8)



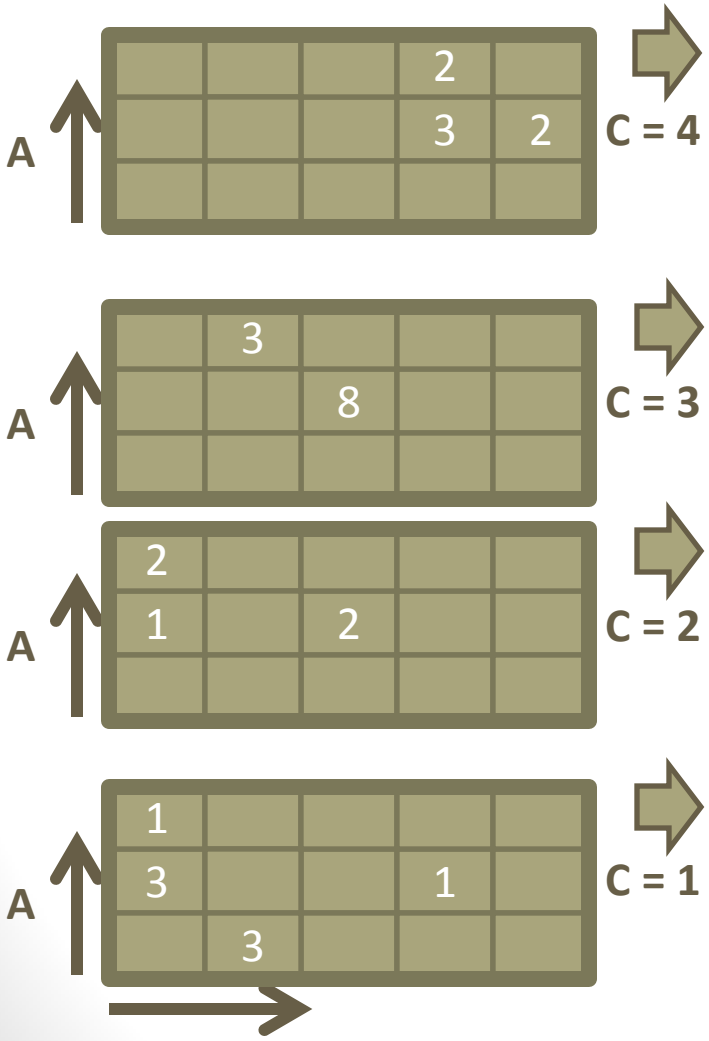
A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3

Sparse Matrices: Exercise (9)



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

Sparse Matrices: Exercise (10)



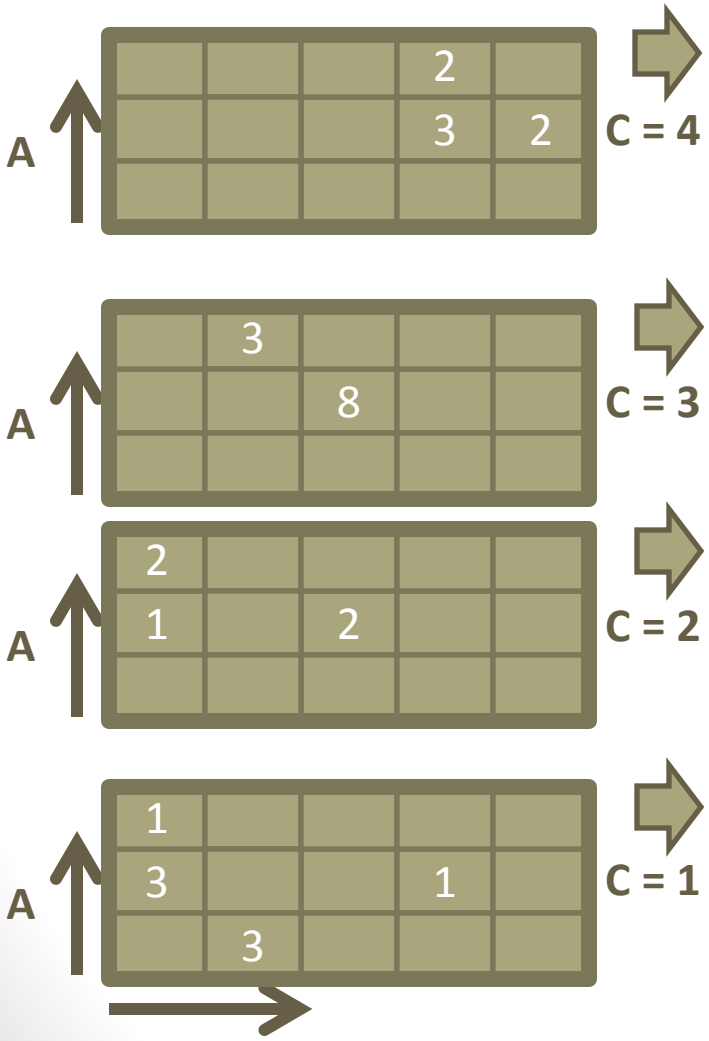
A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3

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)



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

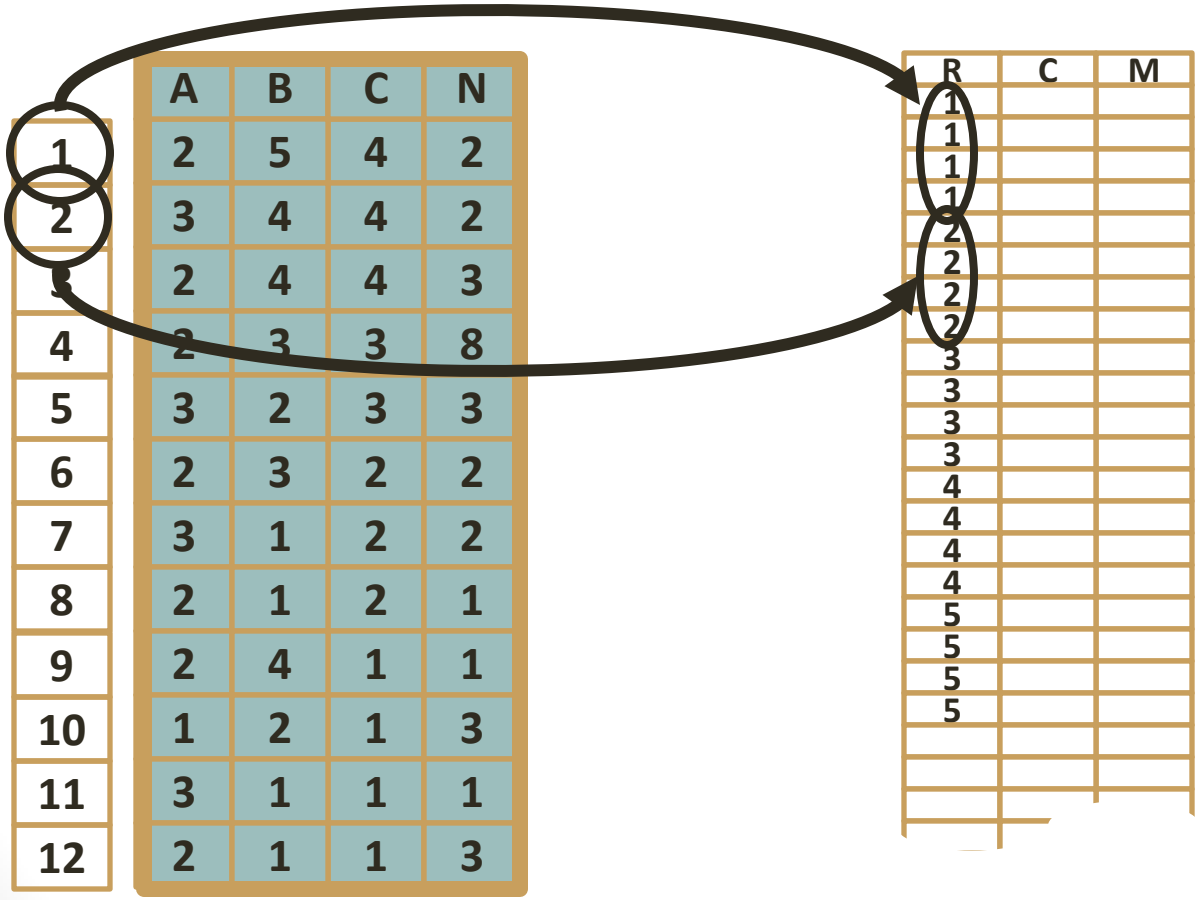
A	B	C	N
2	4	1	1
1	2	1	3
3	1	1	1
2	1	1	3

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

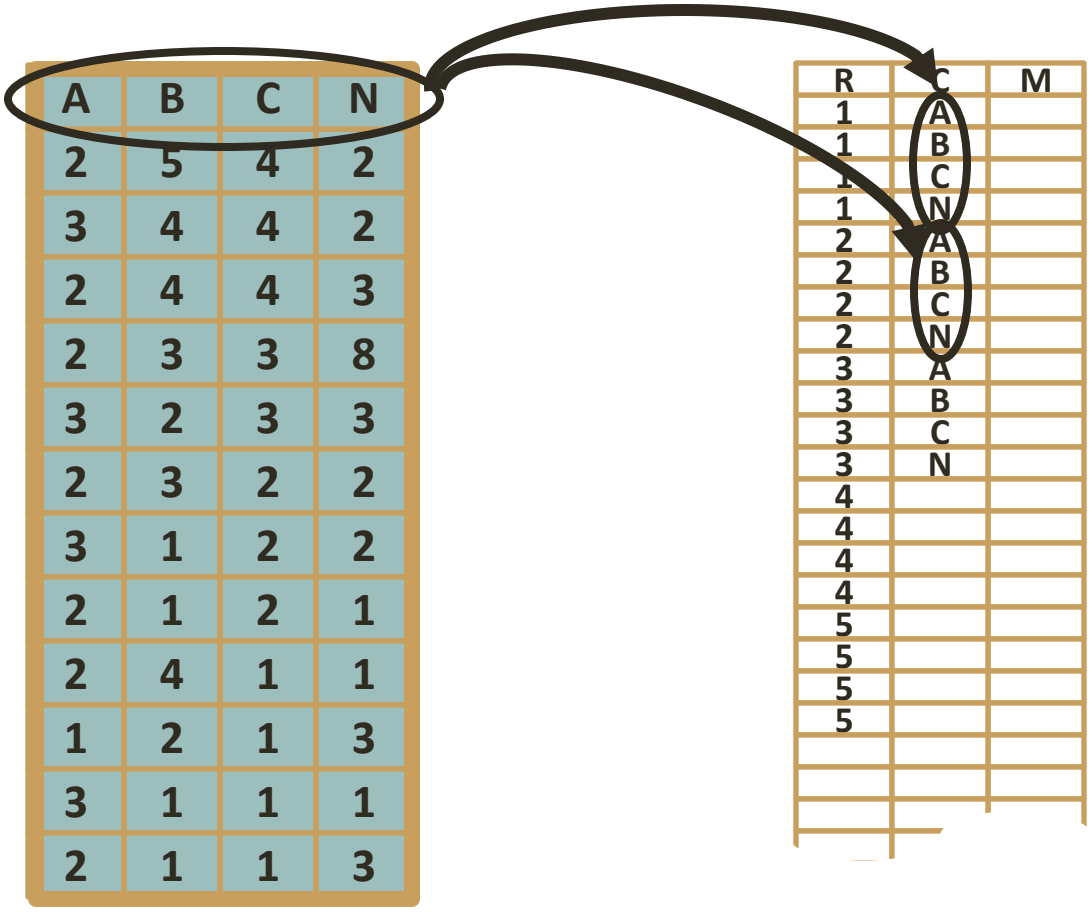
Sparse Matrices: Exercise (12)

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)



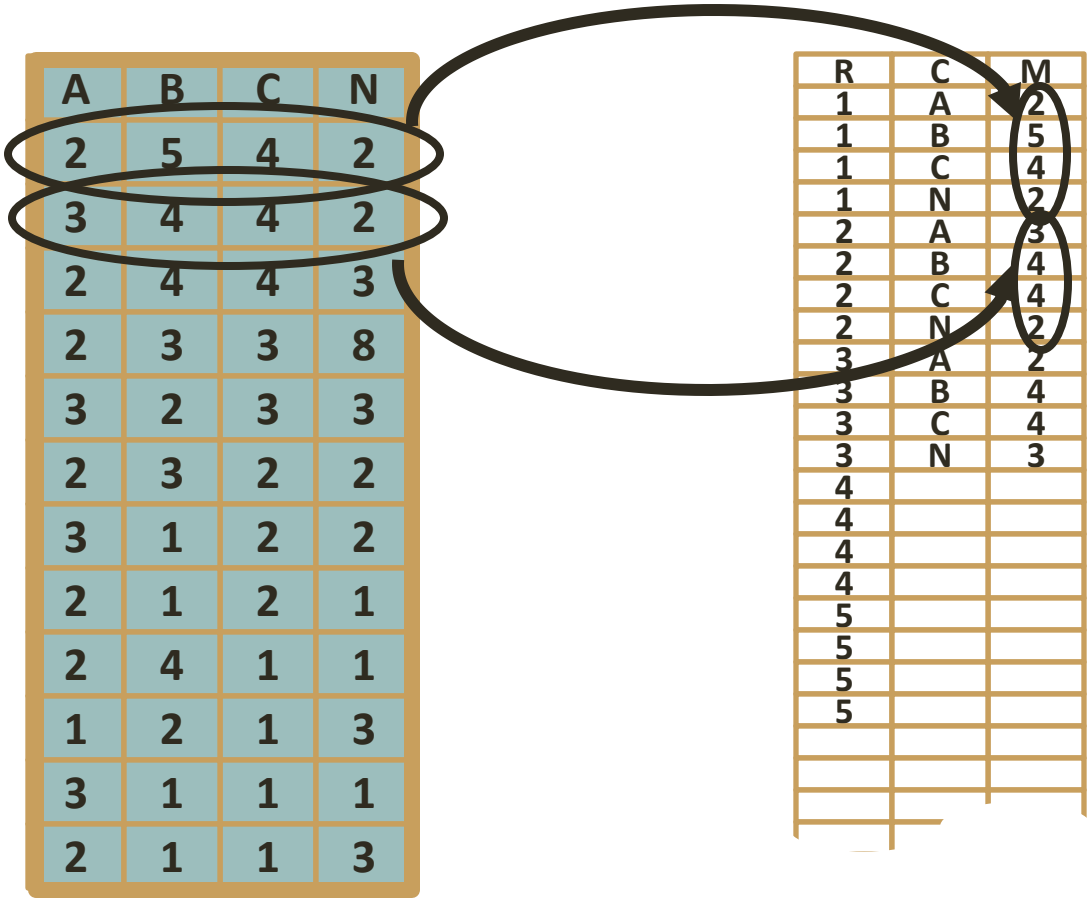
Sparse Matrices: Exercise (14)



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)



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 the last two tables 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

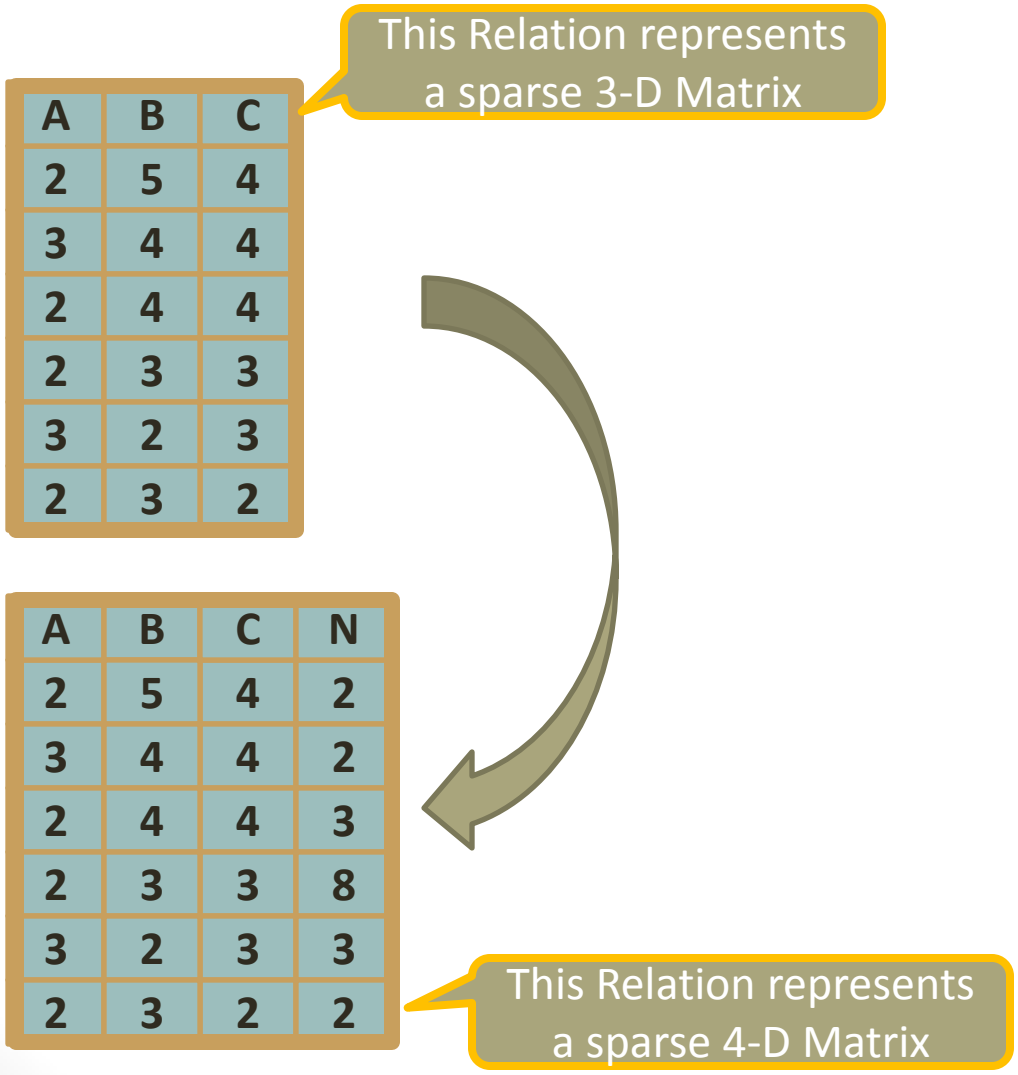
This Relation represents a sparse 3-D Matrix

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

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

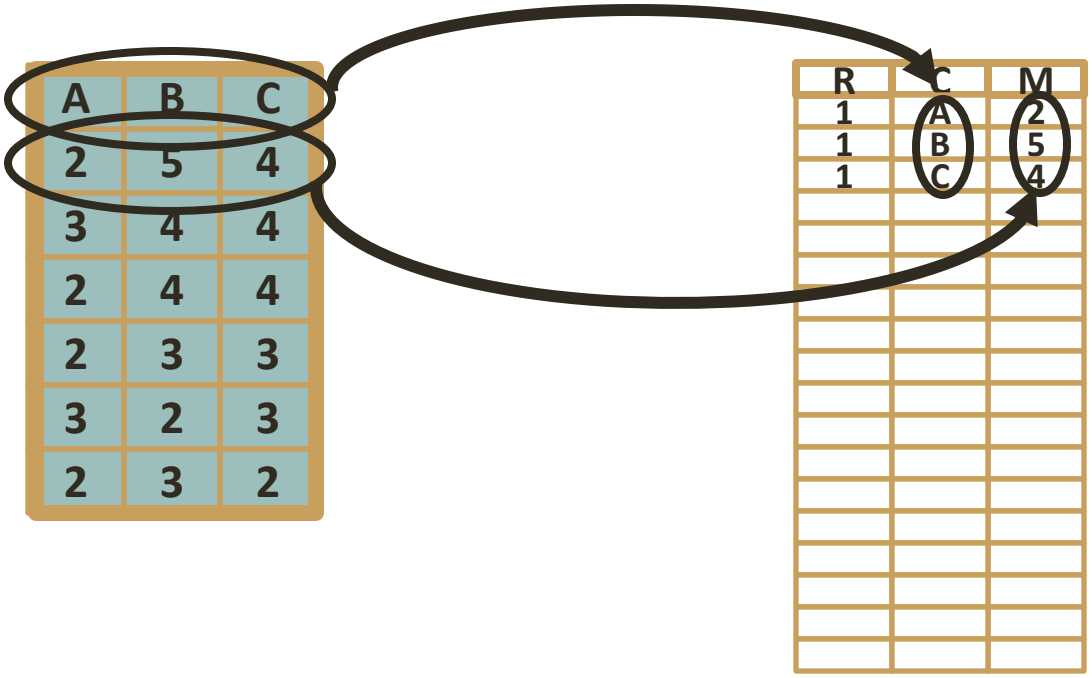
Schema Change: add a column



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



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

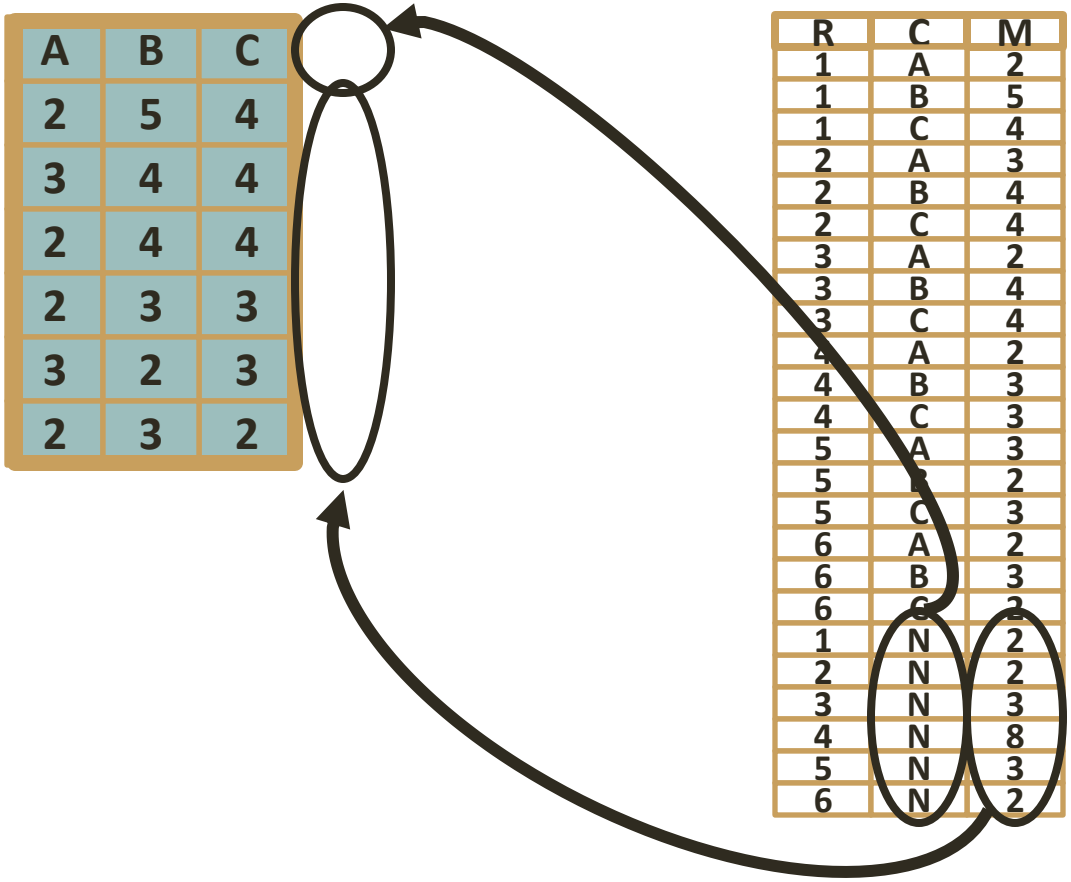
R	C	M
1	A	2
1	B	5
1	C	4
2	A	3
2	B	4
2	C	4
3	A	2
3	B	4
3	C	4
4	A	2
4	B	3
4	C	3
5	A	3
5	B	2
5	C	3
6	A	2
6	B	3
6	C	2

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

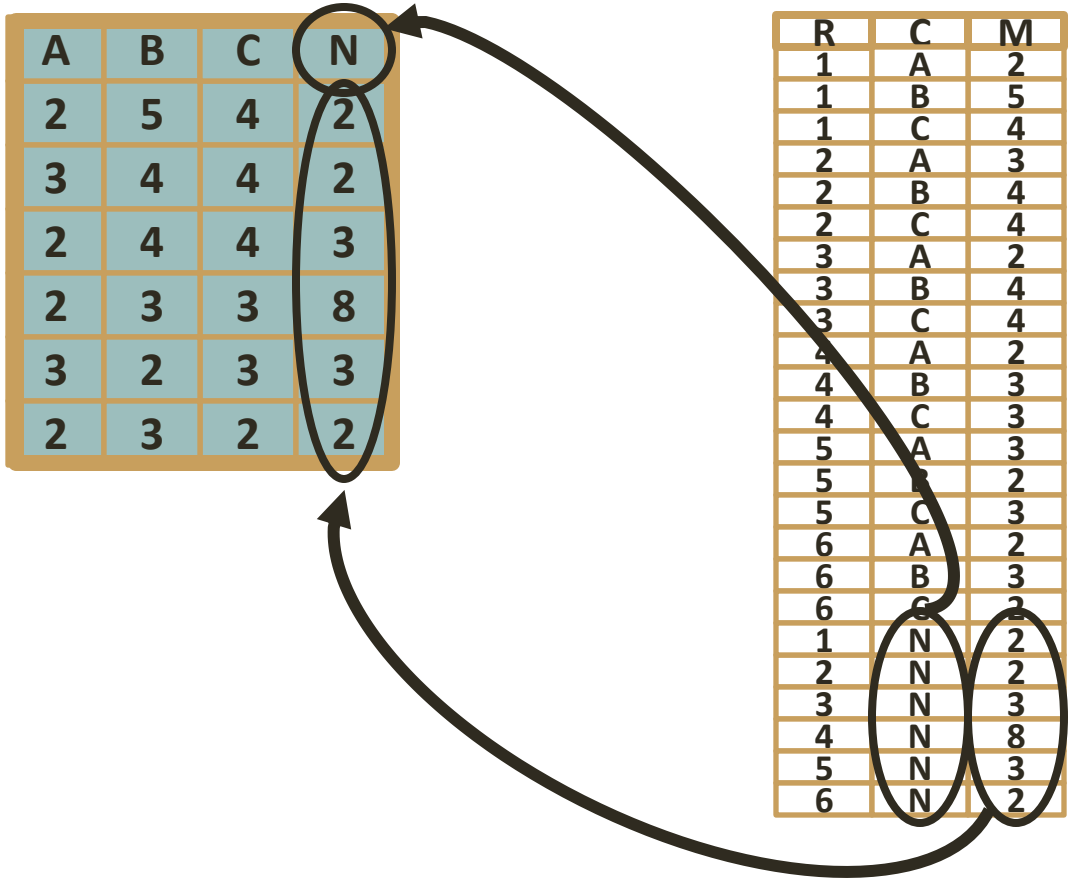
Adding new rows to second table with a new index



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

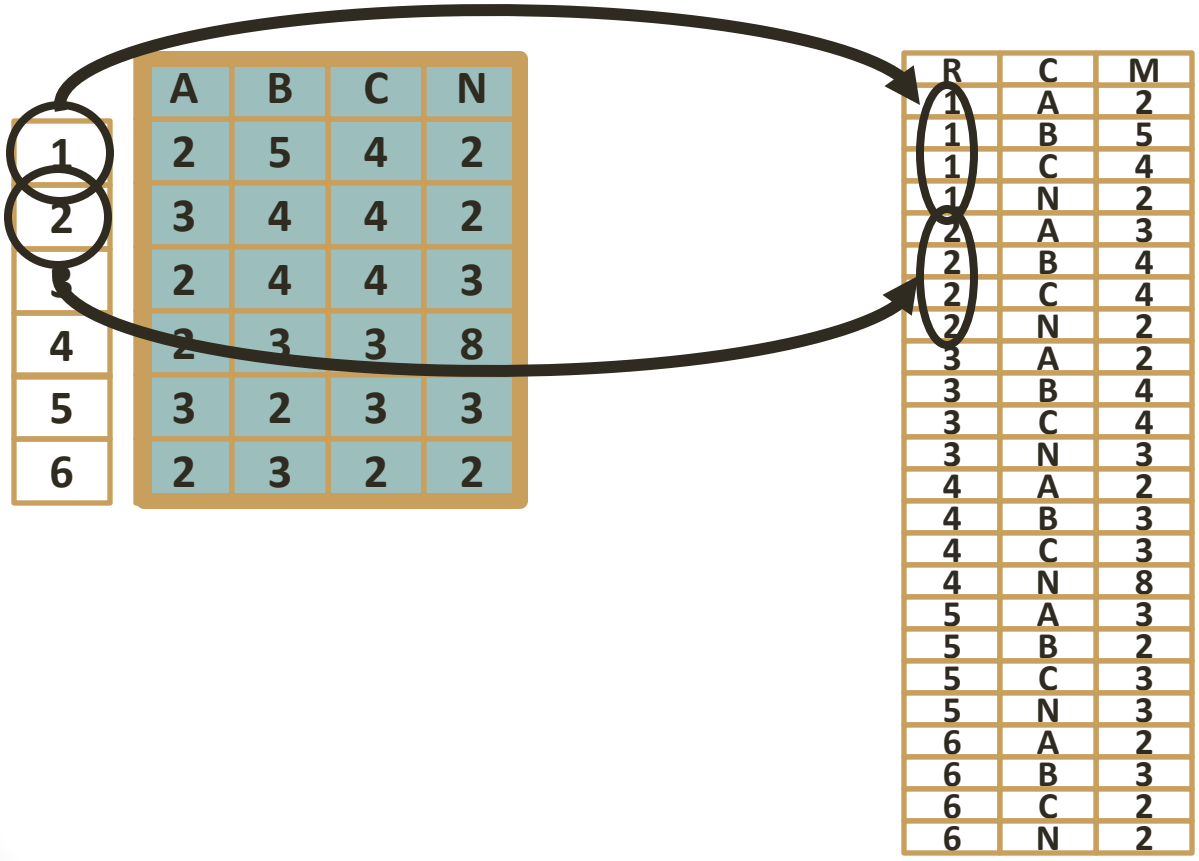
R	C	M
1	A	2
1	B	5
1	C	4
1	N	2
2	A	3
2	B	4
2	C	4
2	N	2
3	A	2
3	B	4
3	C	4
3	N	3
4	A	2
4	B	3
4	C	3
4	N	8
5	A	3
5	B	2
5	C	3
5	N	3
6	A	2
6	B	3
6	C	2
6	N	2

Rows may be resorted without changing the relation

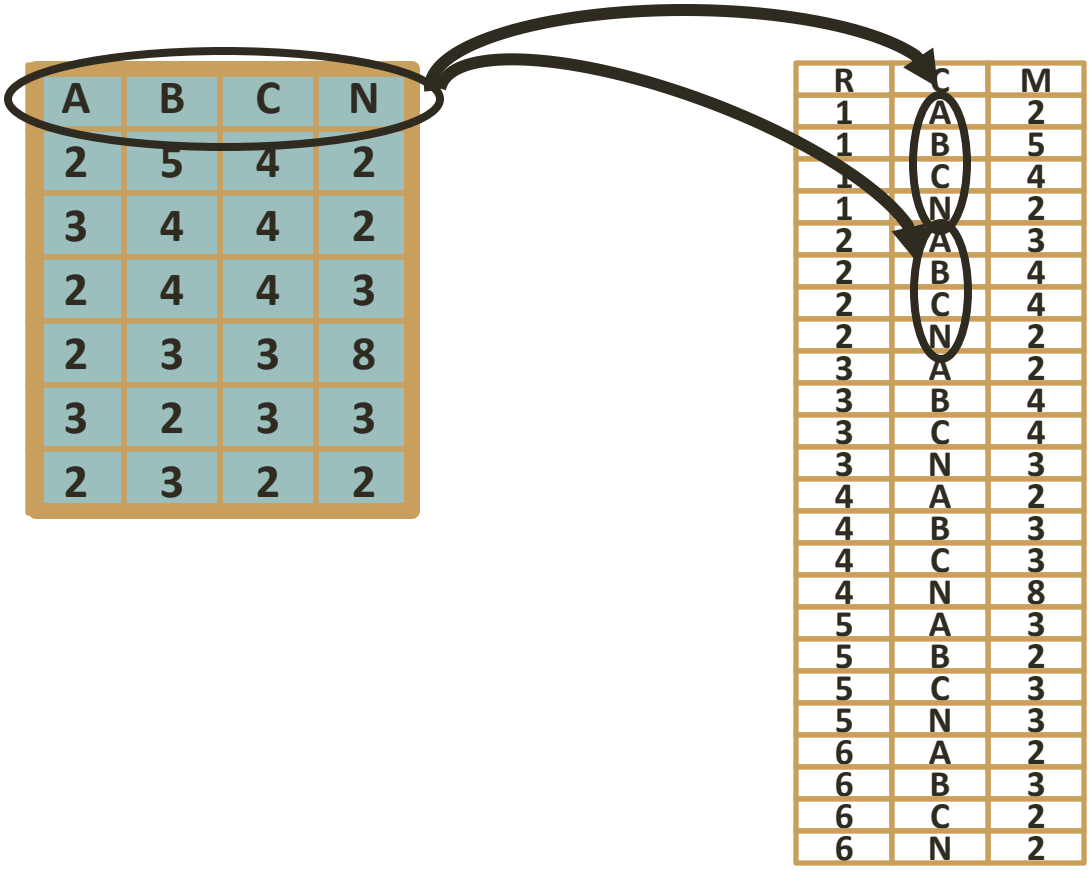
A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3
2	3	3	8
3	2	3	3
2	3	2	2

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



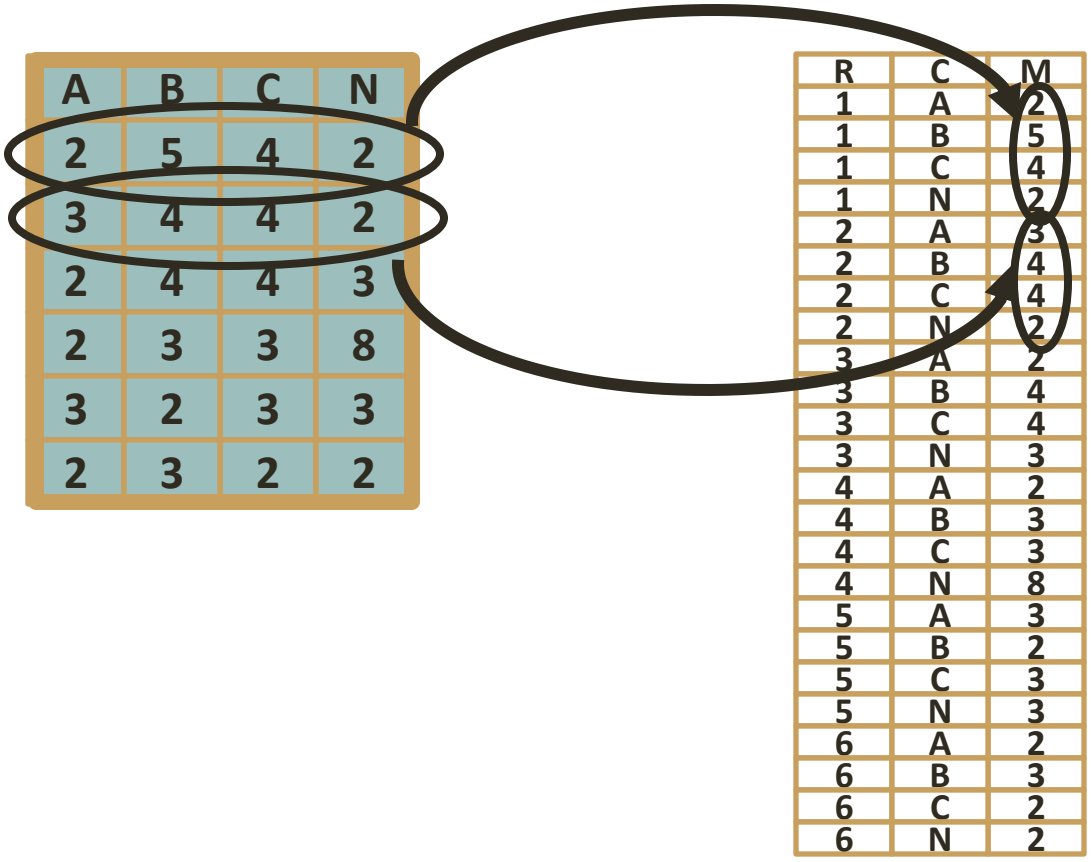
Schema Change Proved



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

Break

Sparse Matrices Manipulation

Examples of Sparse Matrix Manipulation in a database

(see MatrixAlgebra.sql)

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

Data as Sparse Matrices

Break

NOSQL: Scale-out

Scale-up vs. Scale-out

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

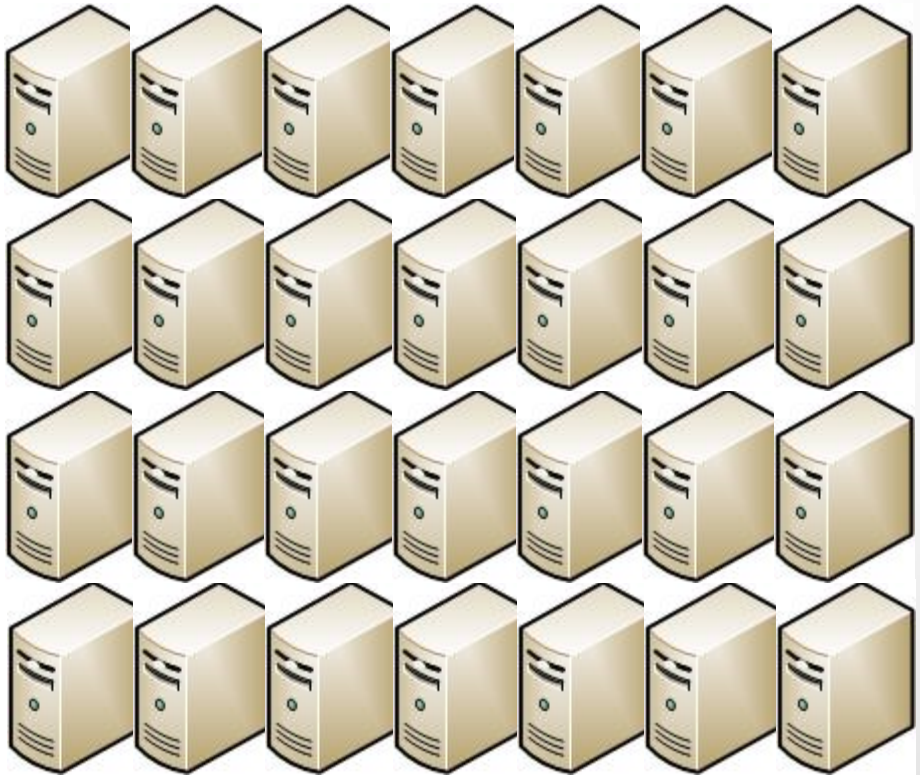
Scale Up vs. Scale Out

Scale Up

Scale Up vs. Scale Out

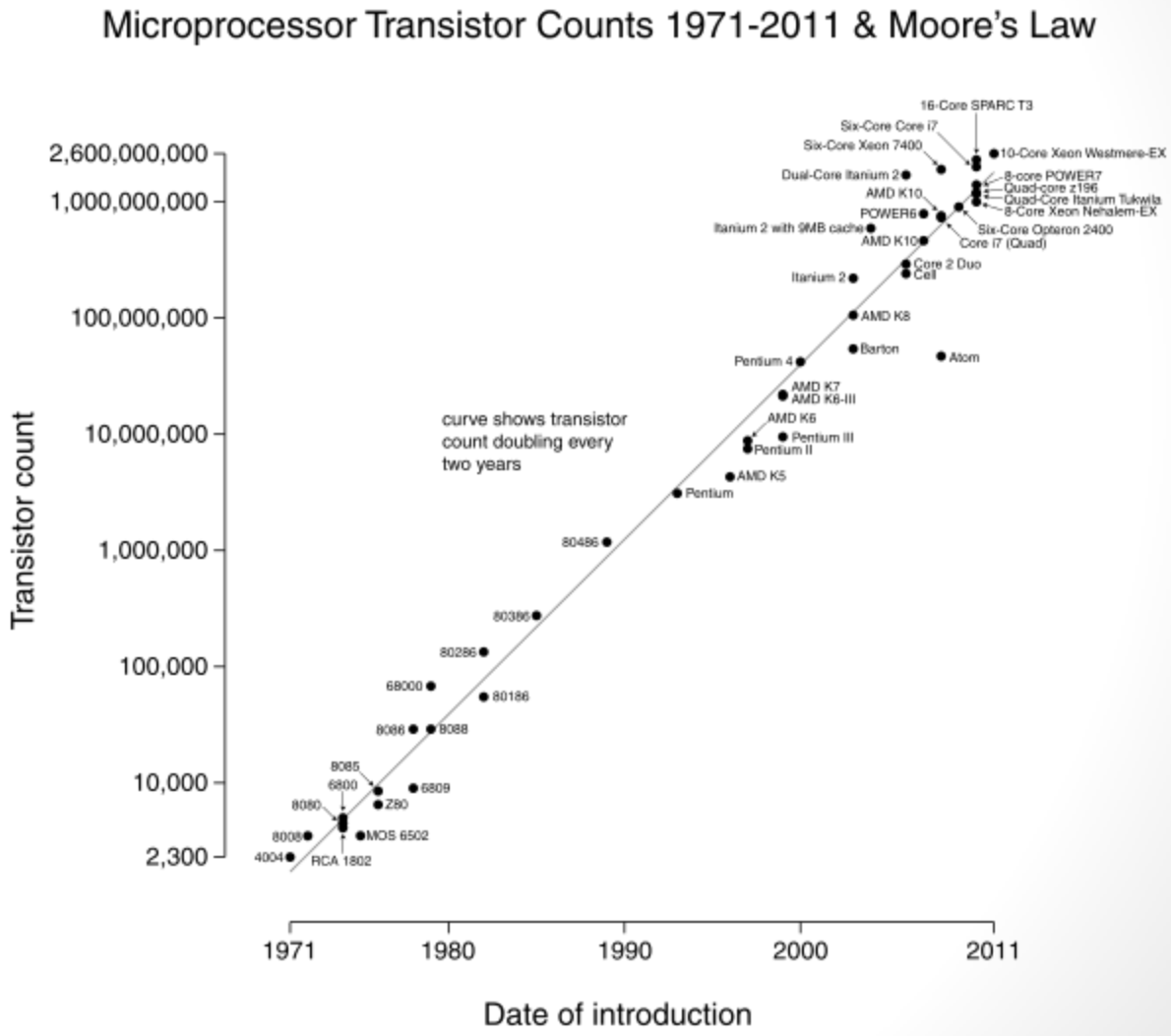
Scale Up

Scale Out



Scale-up vs. Scale-out

- Scale-up
- Moore's Law



Scale-up vs. Scale-out

Grace Hopper

Scale-up vs. Scale-out

Grace Hopper

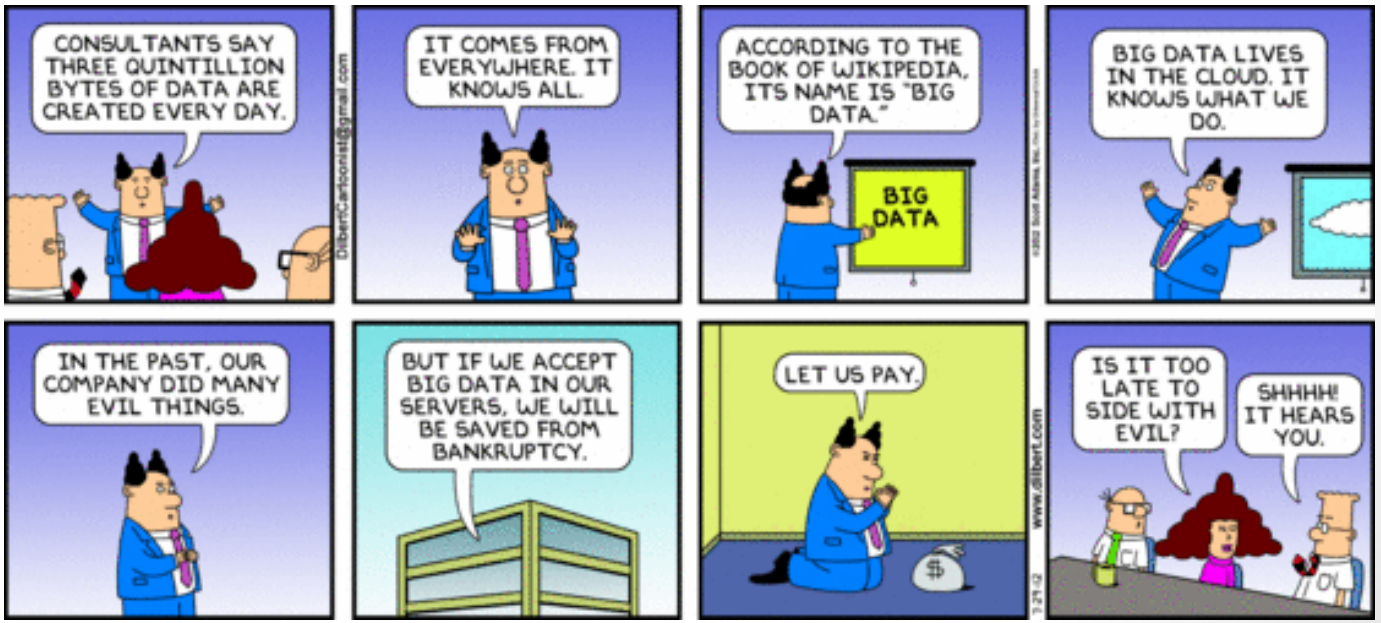
"In pioneer days they used oxen for heavy pulling, and when one ox couldn't budge a log, they didn't try to grow a larger ox. We shouldn't be trying for bigger computers, but for more systems of computers."

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



Scale-out and the "Cloud"

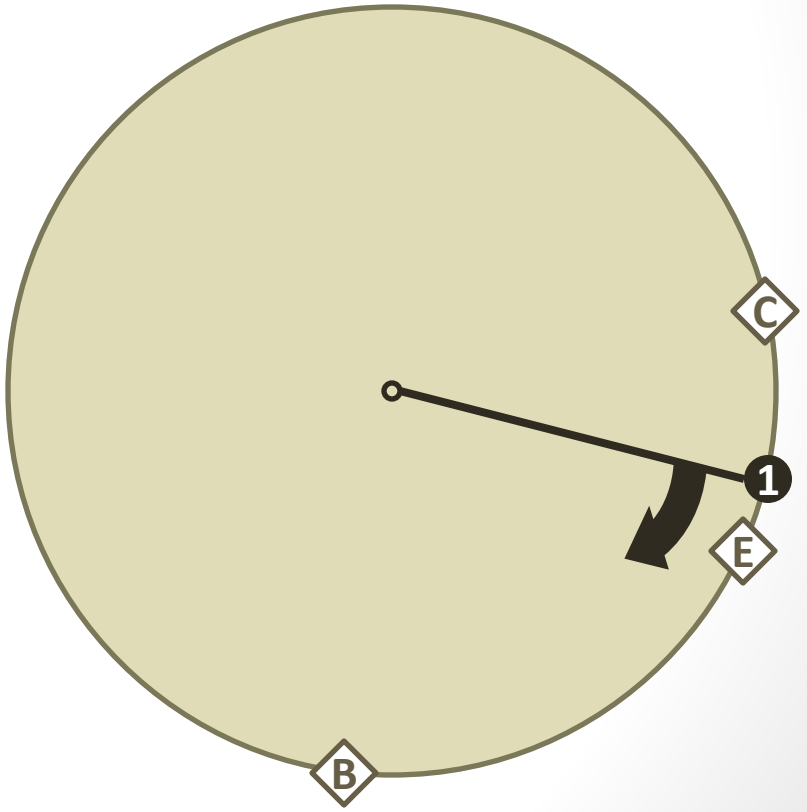
- **Elasticity** has made cloud computing feasible
- Clouds generally employ **virtual machines** that can be created at a moments notice, reduced to an image (dehydrated), re-started from an image, and deleted (recycled).
- How do we partition storage or usage among an unknown number of machines? Often 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.

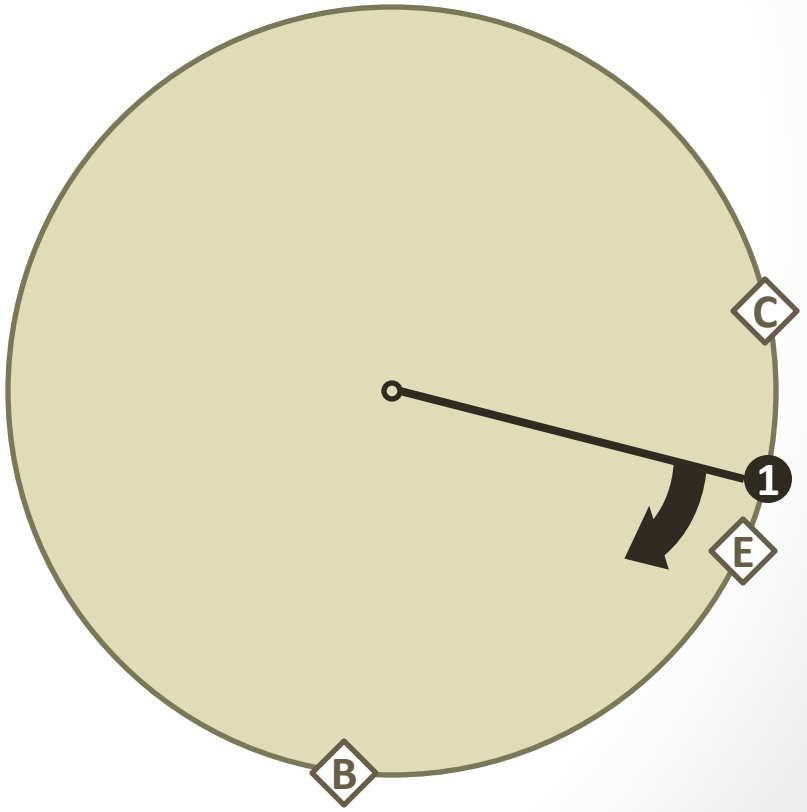
Consistent Hashing

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



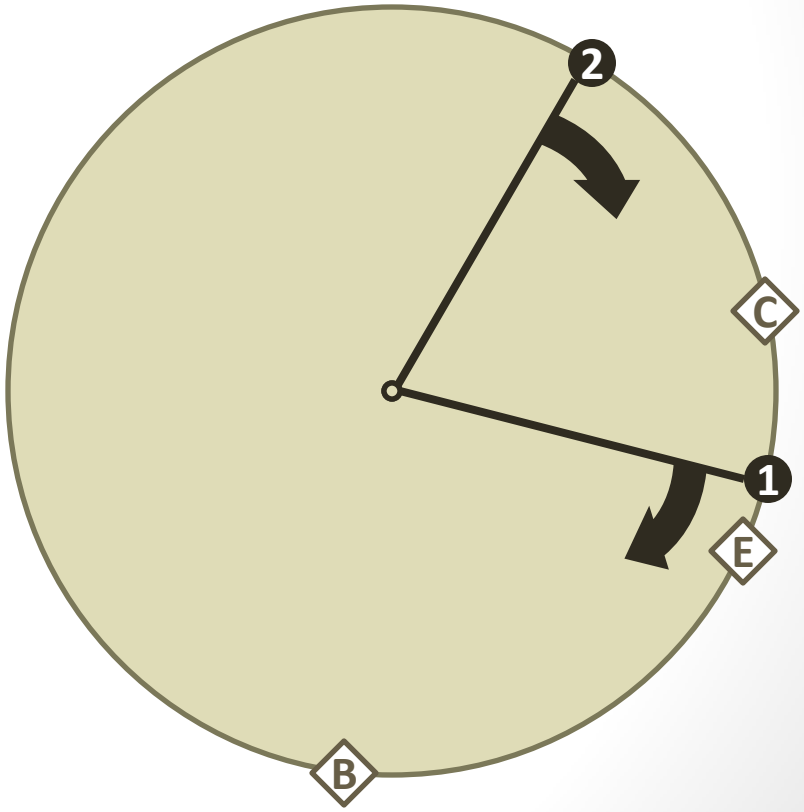
Consistent Hashing

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



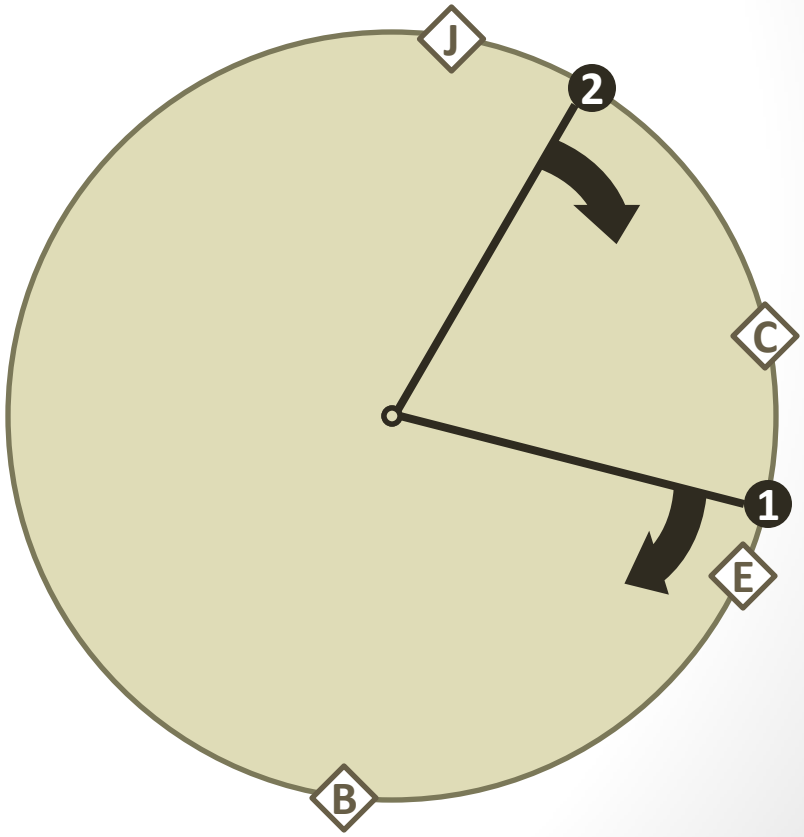
Consistent Hashing

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



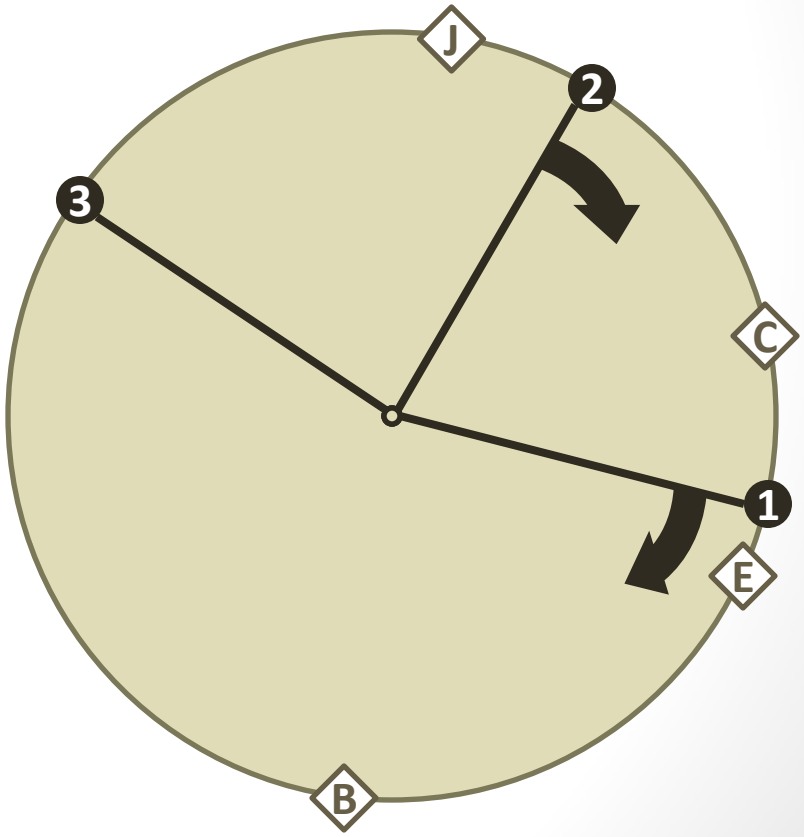
Consistent Hashing

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



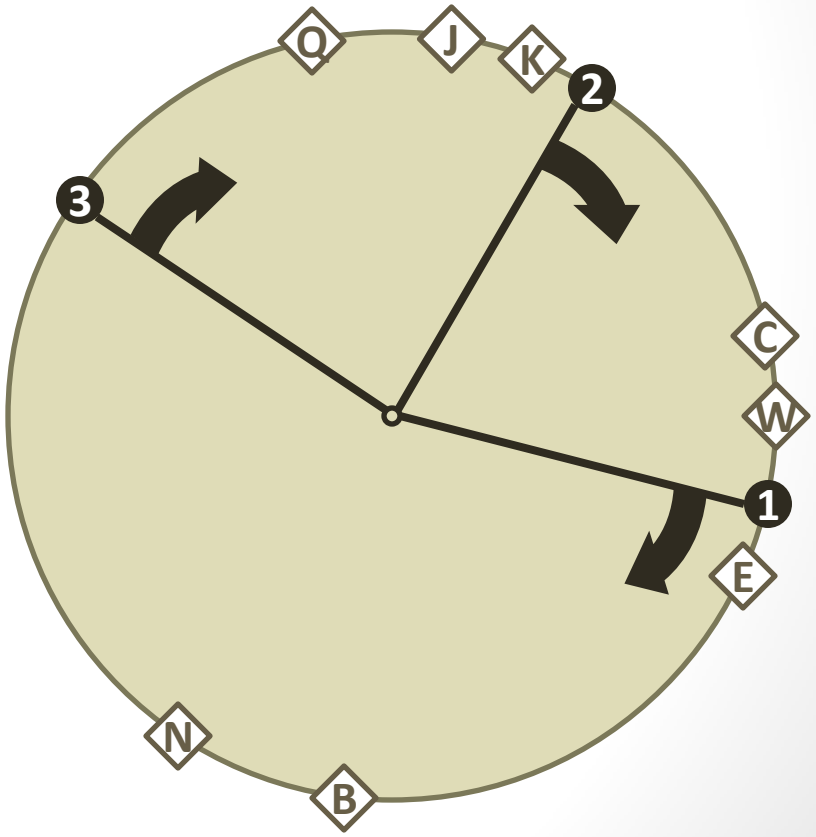
Consistent Hashing

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



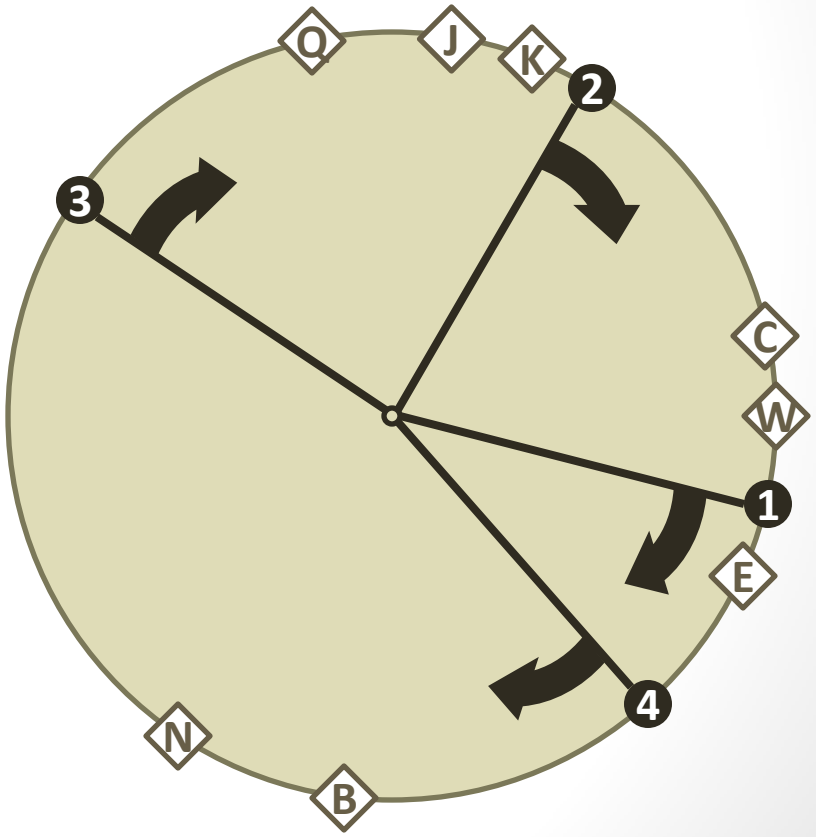
Consistent Hashing

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



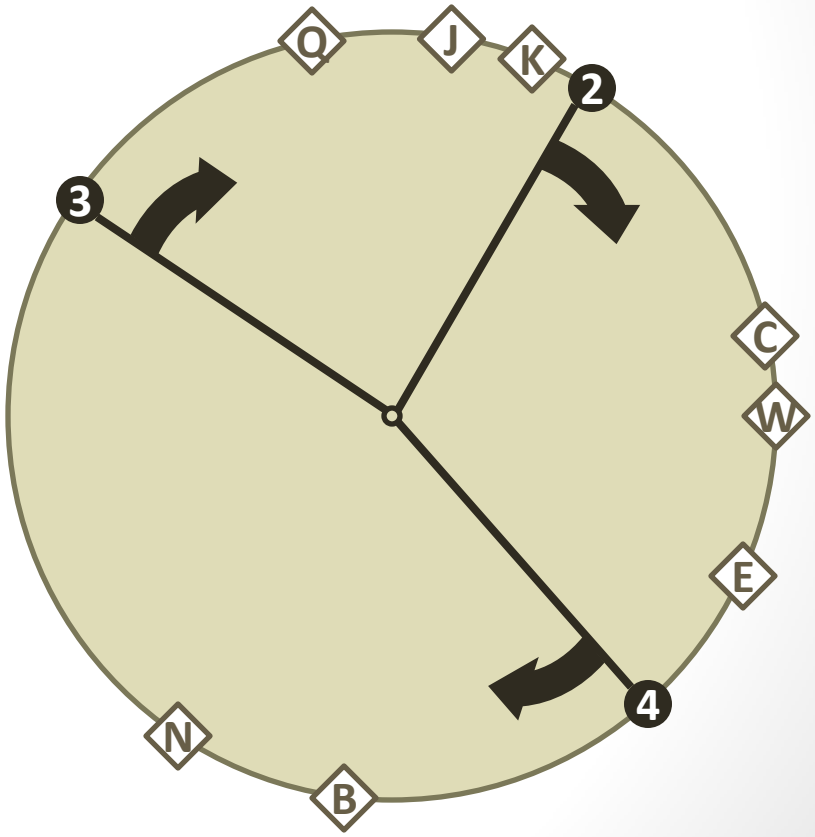
Consistent Hashing

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



Consistent Hashing

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

New Terminology

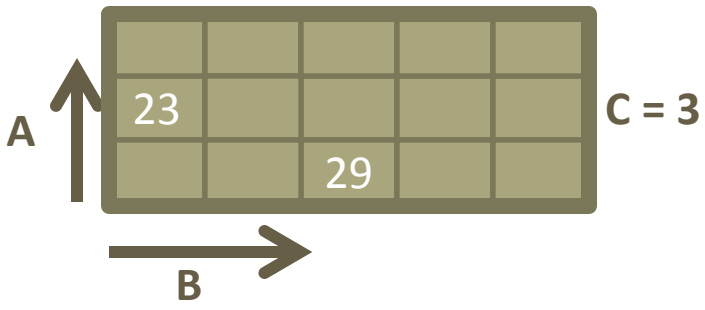
- Hadoop
- Master Node
- Data Node
- Cluster
- Hive
- Impala
- MapReduce
- HDFS
- Doug Cutting
- Scalability
- AWS
- Elastic Cloud
- NoSQL
- CAP Theorem
- Consistency (CAP)
- Availability (CAP)
- Partition Tolerance (CAP)
- Eric Brewer
- RDBMS
- ACID
- Atomic (ACID)
- Consistent (ACID)
- Isolation (ACID)
- 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

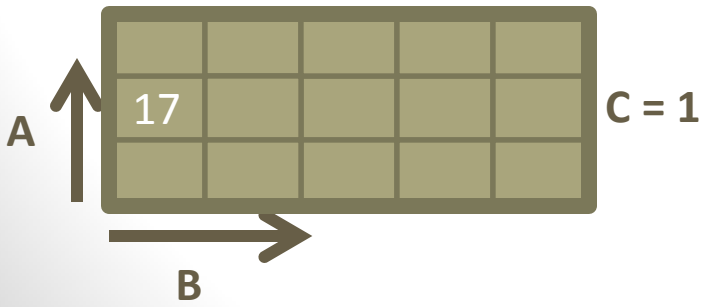
Assignment 07

Assignment (1)

1. On the next slide is a sparse multi-dimensional matrix. Create a standard table, called Table 1, and an EAV representation, called Table 2, of this sparse matrix. The slides titled “Sparse Matrices: Exercise” present an example of this kind of task. Create these tables by “hand”. Show the actual tables. You do not need to write code.
   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.

Assignment (2)





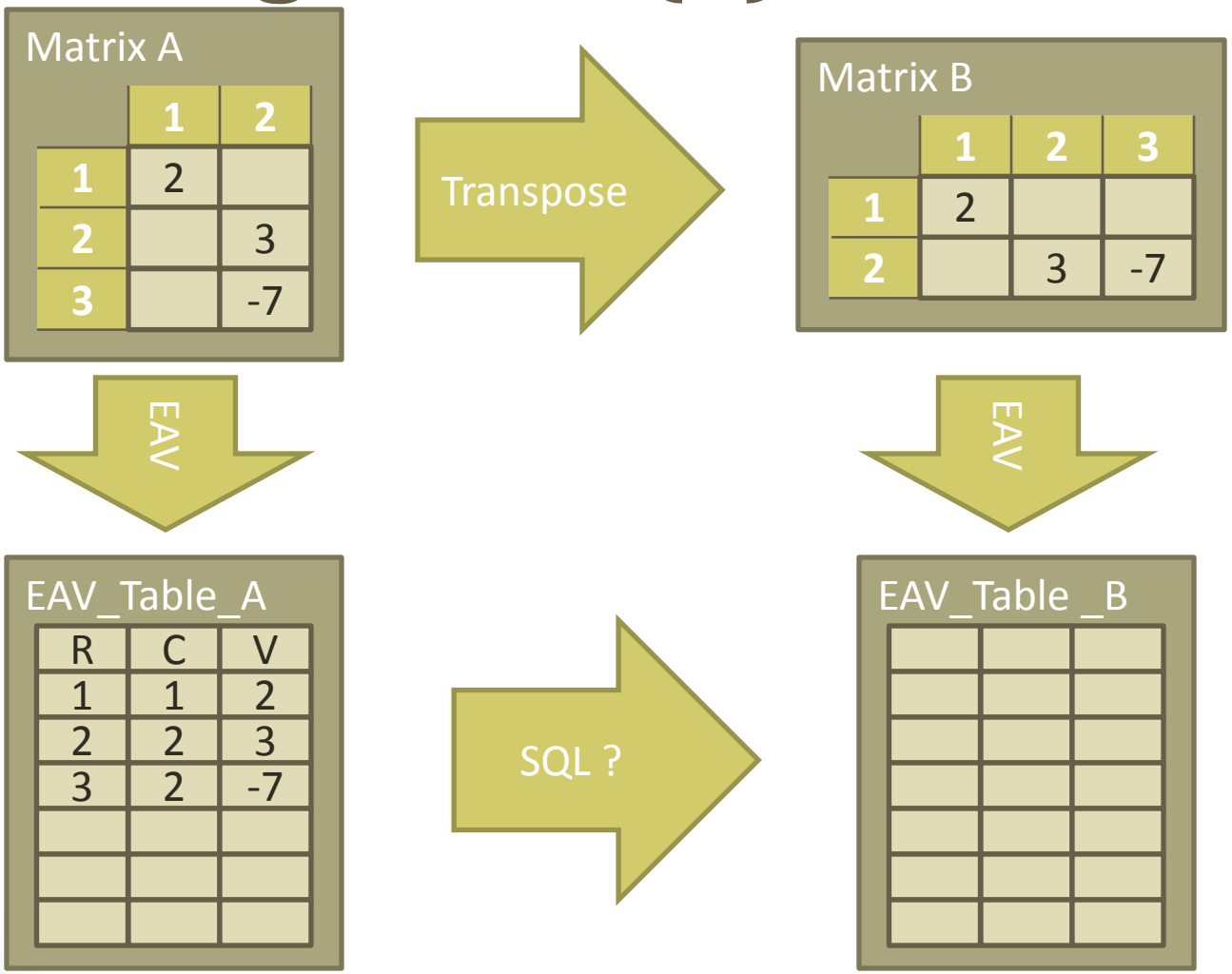
Data from a real estate survey of single-family houses in downtown Seattle. Cell values are number (**N**) of houses found for sale:

- **A**: **A**rea in 1000's of square feet
- **B**: Number of **B**athrooms
- **C**: **C**ost in $100,000.-

Assignment (3)

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 Matrices in the EAV 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 directly encoded with the EAV schema do the following:
   a) Write SQL for scalar multiplication of a 2D Matrix stored in the EAV schema. See Exercise 5 in MatrixAlgebra.sql
   b) Write SQL for transposition of a 2D Matrix stored in the EAV schema. See Exercise 6 in MatrixAlgebra.sql and the next slide.
   c) Optional: Write SQL for addition of two 2D matrices in the EAV schema. See Exercise 7 in MatrixAlgebra.sql.

Assignment (4)



Assume that EAV_Table_A is the only table in the database. What SQL statement will present EAV_Table_B?

Assignment (5)

4. Complete Assignment items 1, 2, and 3 in a *.txt or *.doc file so that I can copy and paste the sql statements into a database engine. Submit by Saturday 11:56 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 07

Introduction to Data Science