Introduction to Data Science

Lecture 07; May 18th, 2015

Ernst Henle
ErnstHe@UW.edu
Skype: ernst.predixion

Agenda

- Social Interactions
  - Get and provide help through the LinkedIn group
  - Encourage Group Homework
- Announcements
- Data science - the business point of view by Marius Marcu
- Break
- Review Relational Algebra (Homework)
- Continue with Relational Algebra
- Quiz 08a Relational Algebra
- Sparse Matrices and EAV
- Quiz 08b Sparse Matrices and EAV
- Sparse Matrix Exercises
- Break
- Sparse Matrix Manipulation
- Time Permitting: Hadoop Intro and HDFS
- Assignment

Announcements

- May 25th No Class. Memorial Day
- 1-hour guest lecture on June 1st by Matt Danielson "A (brief) introduction to Python for Data Science"

Data science
the business point of view

Marius Marcu
2015
mariusmarcu@global.t-bird.edu

Break

Relational Algebra: Review

- Homework: HomeWork07.sql
- RelationalAlgebraAndSQL.pdf
- RelationalAlgebraAndSQL.sql

Relational Algebra

The Theory behind Relational Databases

Continued from Last Week

Relational Algebra: Product

Combine Rows

Name	**Age**
Blackburn	5
Kobayashi	21

Name	**Home**
Menchú	Rent
Alvarez	Rent
Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

SQL Statement:
SELECT * FROM TableR, TableS

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
		Alvarez	Rent
		Yamana	Own
Kobayashi	21	Menchú	Rent
		Alvarez	Rent
		Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Kobayashi	21		
Blackburn	5	Alvarez	Rent
Kobayashi	21		
Blackburn	5	Yamana	Own
Kobayashi	21		

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Kobayashi	21	Menchú	Rent
Blackburn	5	Alvarez	Rent
Kobayashi	21	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Kobayashi	21	Menchú	Rent
Blackburn	5	Alvarez	Rent
Kobayashi	21	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

The result of a product is a relation with n*m tuples where n and m are the number of tuples in the operands. The arity of the result is i + j where i and j are the arities of the operands

Name	**Age**
Blackburn	5
Kobayashi	21

Name	**Home**
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	**Age**	**Name 2**	**Home**
Blackburn	5	Menchú	Rent
Kobayashi	21	Menchú	Rent
Blackburn	5	Alvarez	Rent
Kobayashi	21	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Product

Combine Rows

The result of a product is a relation with n*m tuples where n and m are the number of tuples in the operands. The arity of the result is i + j where i and j are the arities of the operands

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own

Relational Algebra Product:
R X S

Relational Algebra: Join

Combine Rows

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Kobayashi	21	Menchú	Rent
Blackburn	5	Alvarez	Rent
Kobayashi	21	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Yamana	Own

Relational Algebra Product with Select:

$\sigma_{\varphi}(R \times S)$ where φ: Home = “Rent”

Relational Algebra Join:

$R \bowtie_{\varphi} S$ where φ: Home = “Rent”

Relational Algebra: Join

Combine Rows

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Kobayashi	21	Menchú	Rent
Blackburn	5	Alvarez	Rent
Kobayashi	21	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Yamana	Own

Relational Algebra Product with Select:

$\sigma_{\varphi}(R \times S)$ where φ: Home = "Rent"

Relational Algebra Join:

$R \bowtie_{\varphi} S$ where φ: Home = "Rent"

Relational Algebra: Join

Combine Rows

Name	**Age**
Blackburn	5
Kobayashi	21

Name	**Home**
Menchú	Rent
Alvarez	Rent
Yamana	Own

→

Name 1	**Age**	**Name 2**	**Home**
Blackburn	5	Menchú	Rent
Kobayashi	21	Menchú	Rent
Blackburn	5	Alvarez	Rent
Kobayashi	21	Alvarez	Rent

Relational Algebra Product with Select:

$\sigma_{\varphi}(R \times S)$ where φ: Home = "Rent"

Relational Algebra Join:

$R \bowtie_{\varphi} S$ where φ: Home = "Rent"

Relational Algebra: Join

- A Join is a Product with a select statement

- Product followed by Select
  - SELECT * FROM TableR, TableS WHERE Home = "Rent"
  - $\sigma_{\varphi}(R \times S)$ where φ: Home = "Rent"

- JOIN
  - SELECT * FROM TableR JOIN TableS ON Home = "Rent"
  - $R \bowtie_{\varphi} S$ where φ: Home = "Rent"

Relational Algebra: Division

A Division is sort of like the reverse of a Product

This was a Product Operand

Name	Age
Blackburn	5
Kobayashi	21

This was the result of a Product

Name 1	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

This was a Product Operand

Relational Algebra Division:
$R \div S$

Relational Algebra: Division

Exclude columns

Project

Name	Age
Blackburn	5
Kobayashi	21

Name	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own
Segrè	54	Yamana	Own

Result of Projection

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

Relational Algebra Division:
R ÷ S

Relational Algebra: Division

Select

Name	Age
Blackburn	5
Kobayashi	21

Name	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own
Segrè	54	Yamana	Own

Result of Selection

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

Relational Algebra Division:
R ÷ S

Relational Algebra: Division

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

Name	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own
Segrè	54	Yamana	Own

Relational Algebra Division:
$R \div S$

Relational Algebra: Division

The result of a division is a relation with n tuples of arity I where the divisor operand has exactly m tuples of arity j that are a subset of the of the dividend tuples.

Name	Age
Blackburn	5
Kobayashi	21

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

Name	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own
Segrè	54	Yamana	Own

Relational Algebra Division:

$R \div S$

Relational Algebra: Division

The result of a division is a relation with n tuples of arity i where the dividend operand contains n*m tuples of arity i + j that are a superset of the result tuples.

Name	**Age**
Blackburn	5
Kobayashi	21

Name	**Home**
Menchú	Rent
Alvarez	Rent
Yamana	Own

Name	**Age**	**Name 2**	**Home**
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own
Segrè	54	Yamana	Own

Relational Algebra Division:
$R \div S$

Relational Algebra: Division

The result of a division is a relation with n tuples of arity I where the dividend operand has n*m tuples of arity i + j and the divisor operand has exactly m tuples of arity j that are a subset of the of the dividend tuples.

Name	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own
Segrè	54	Yamana	Own

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

Name	Age
Blackburn	5
Kobayashi	21

Relational Algebra Division:
R ÷ S

Relational Algebra: Division

The result of a division is a relation with n tuples of arity I where the dividend operand has n*m tuples of arity i + j and the divisor operand has exactly m tuples of arity j that are a subset of the of the dividend tuples.

Name	Age
Blackburn	5
Kobayashi	21

Name	Age	Name 2	Home
Blackburn	5	Menchú	Rent
Blackburn	5	Alvarez	Rent
Blackburn	5	Yamana	Own
Kobayashi	21	Menchú	Rent
Kobayashi	21	Alvarez	Rent
Kobayashi	21	Yamana	Own
Segrè	54	Yamana	Own

Name	Home
Menchú	Rent
Alvarez	Rent
Yamana	Own

Relational Algebra Division:
R ÷ S

Relational Algebra: Resources

- Relational Algebra and SQL
  - RelationalAlgebraAndSQL.pdf
  - RelationalAlgebraAndSQL.sql

- http://en.wikipedia.org/wiki/Cartesian_product
- http://en.wikipedia.org/wiki/Commutative_property
- http://en.wikipedia.org/wiki/Associative_property
- http://en.wikipedia.org/wiki/Closure_(mathematics)

Relational Algebra

Quiz 08a

- https://catalyst.uw.edu/webq/survey/ernsthe/271575
- The questions are presented during the quiz

Data as Sparse Matrices

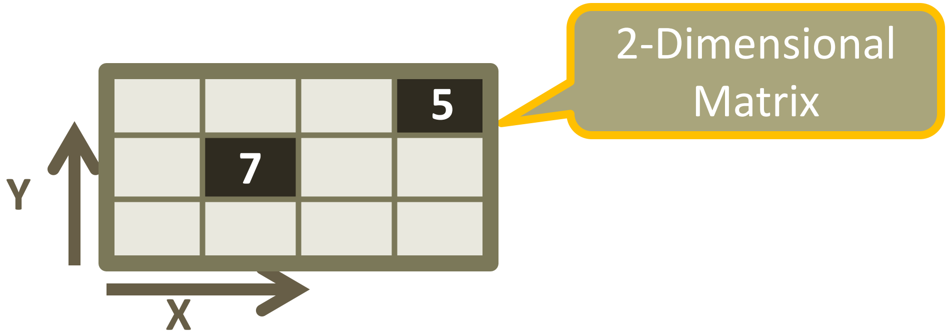
Cartesian Product

- 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.

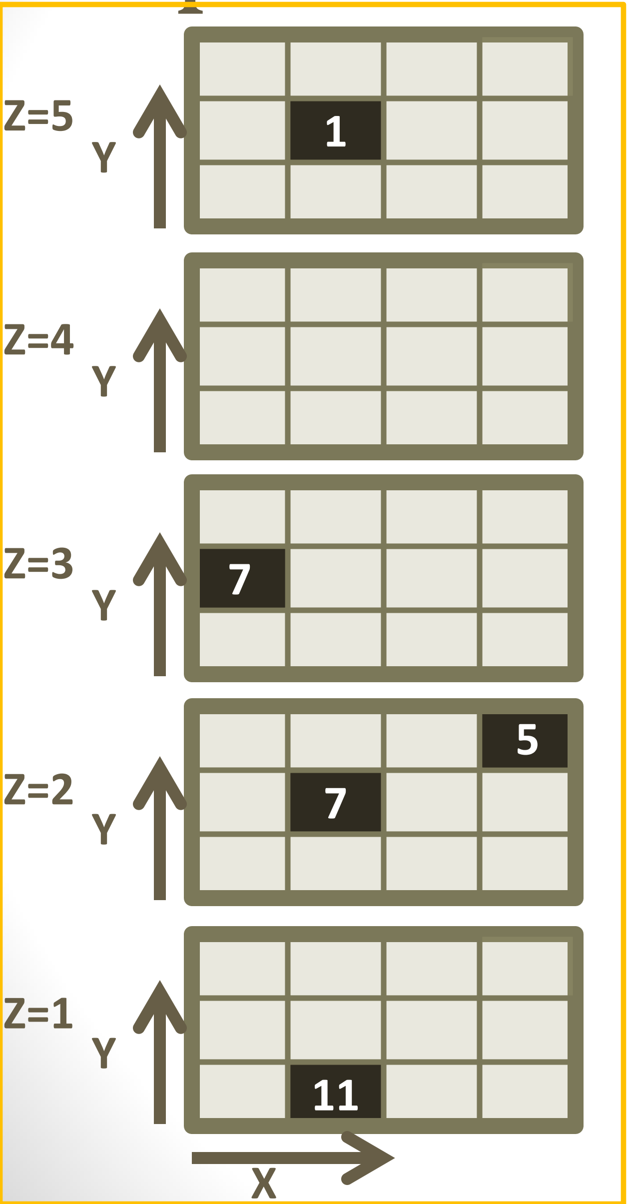
- Relational Algebra
- http://en.wikipedia.org/wiki/Relational_algebra
- In Relational Algebra we need the Cartesian product to combine tuples into a single tuple. The Cartesian product creates a new schema (relation) from other relations.

- 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

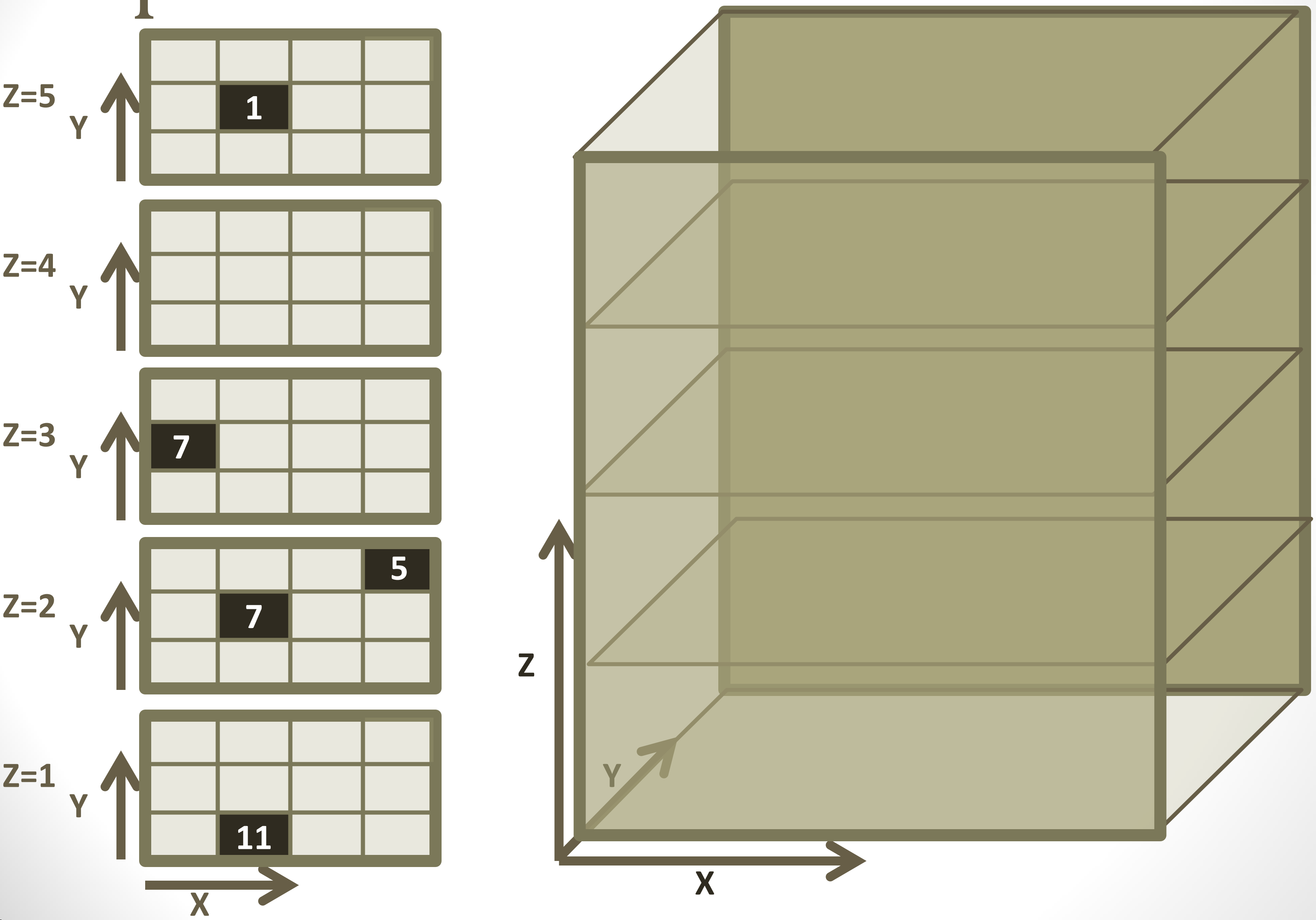


Sparse Matrices

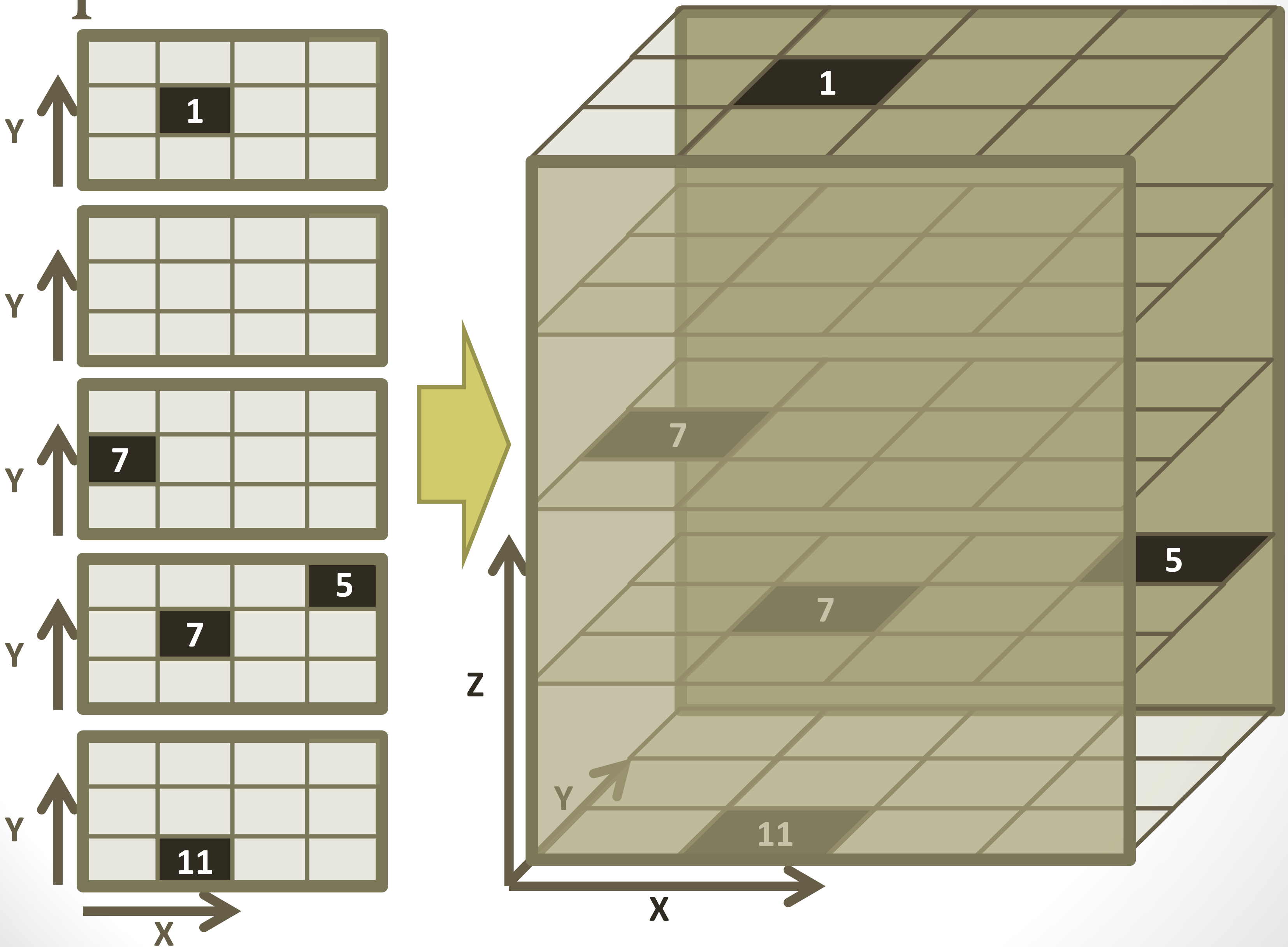


A series of equal-sized 2-dimensional matrices is a 3-dimensional matrix

Sparse Matrices



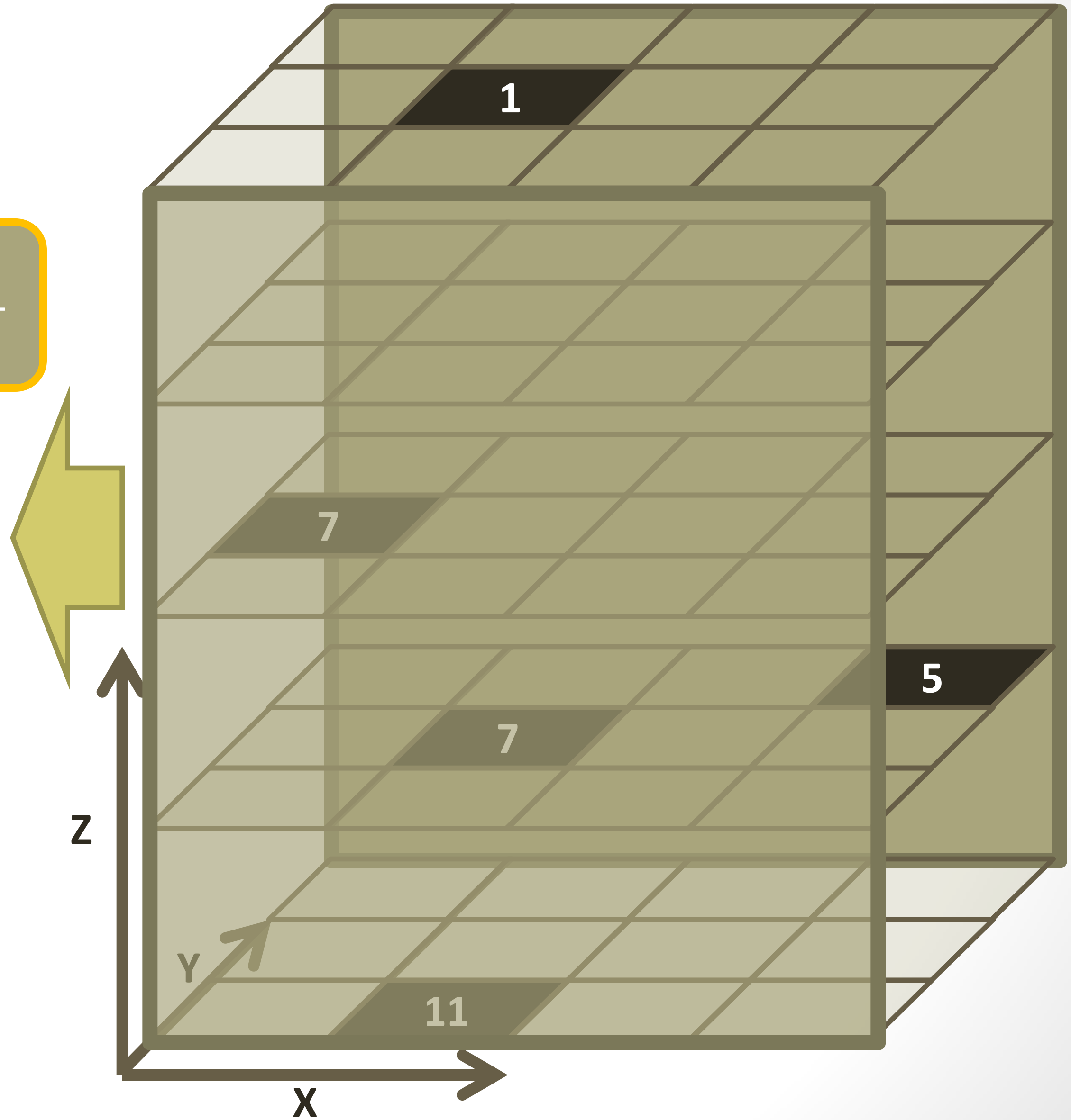
Sparse Matrices



Sparse Matrices

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

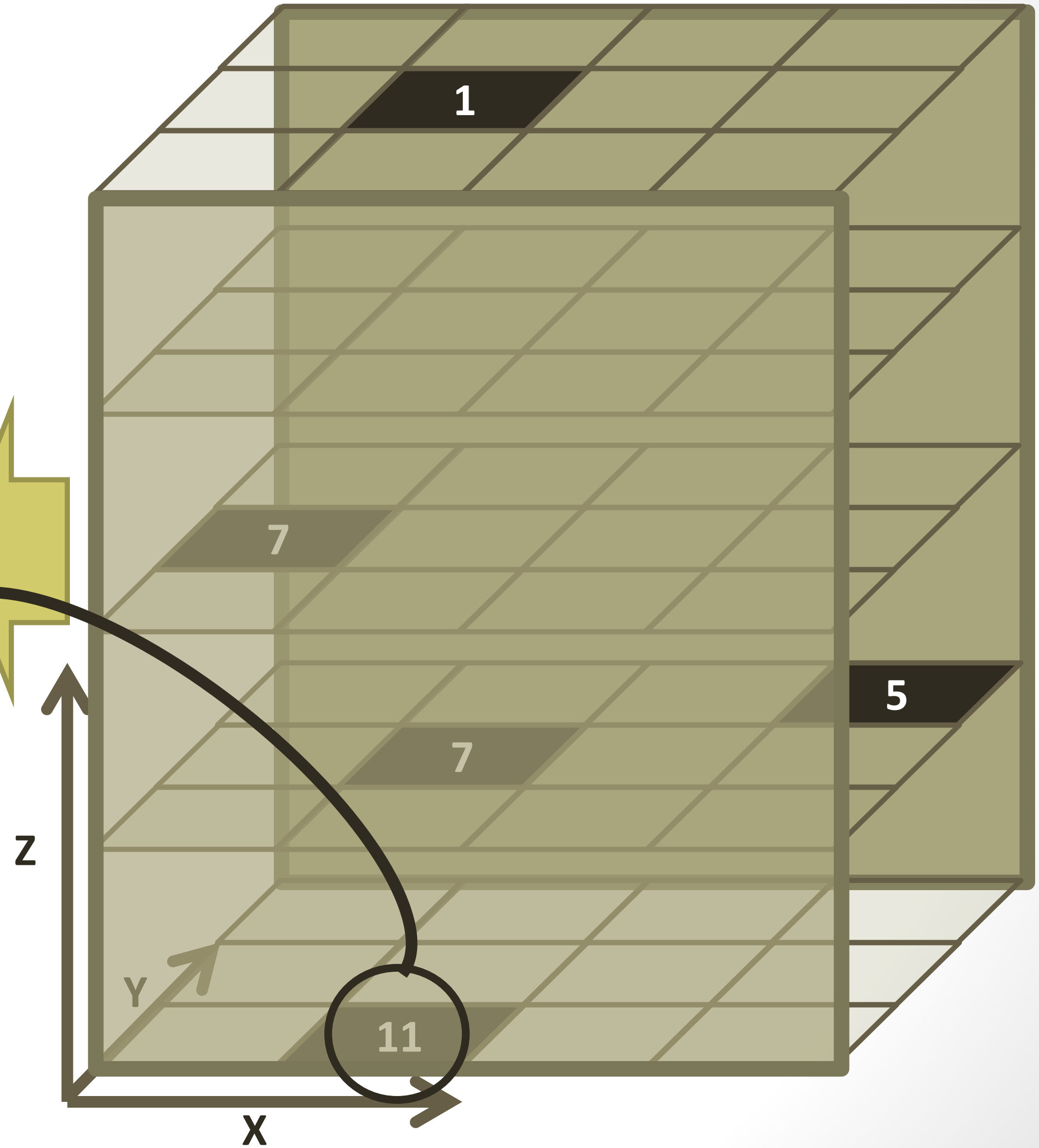
X	Y	Z	V



Sparse Matrices

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

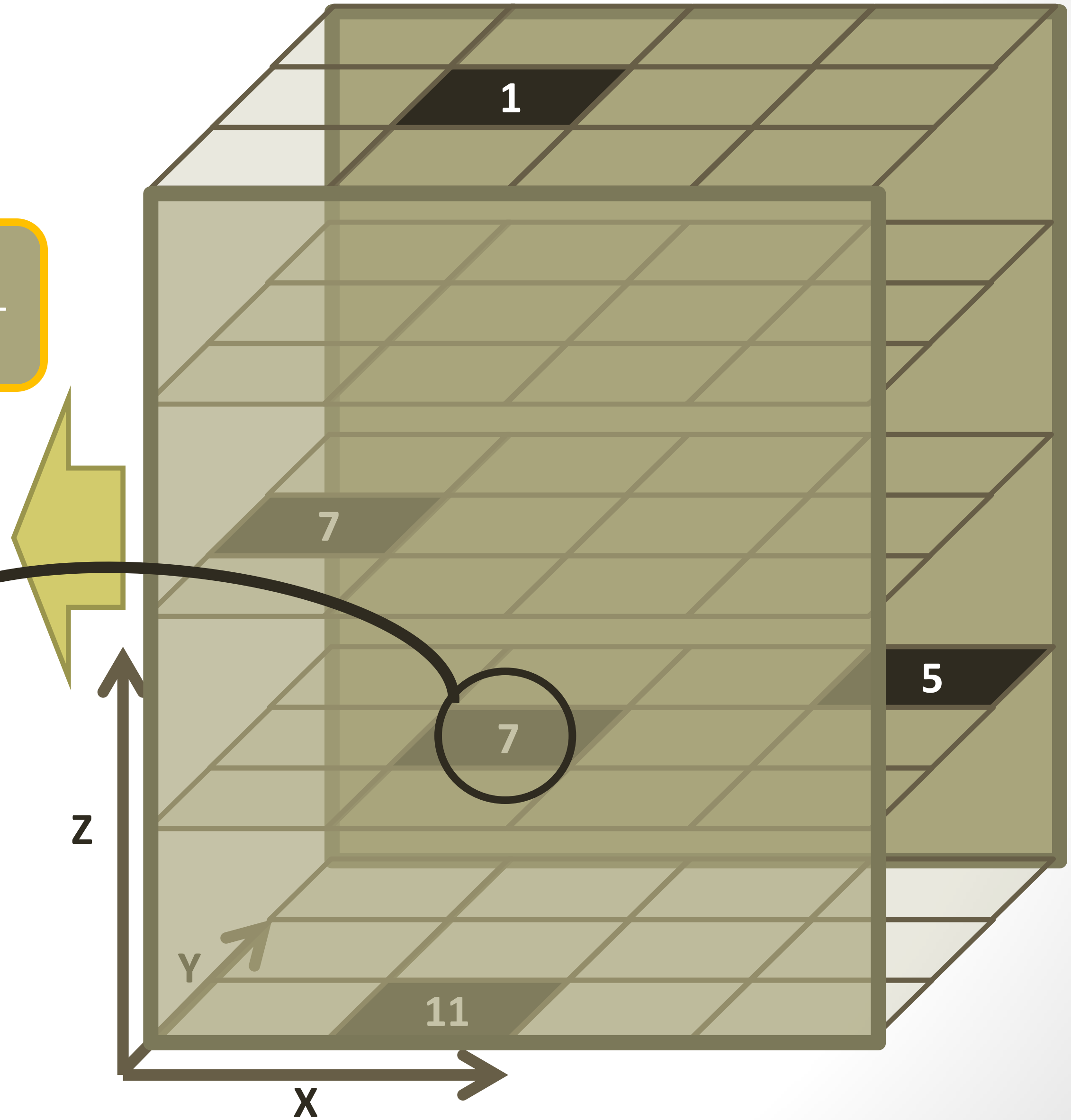
X	Y	Z	V
2	1	1	11



Sparse Matrices

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

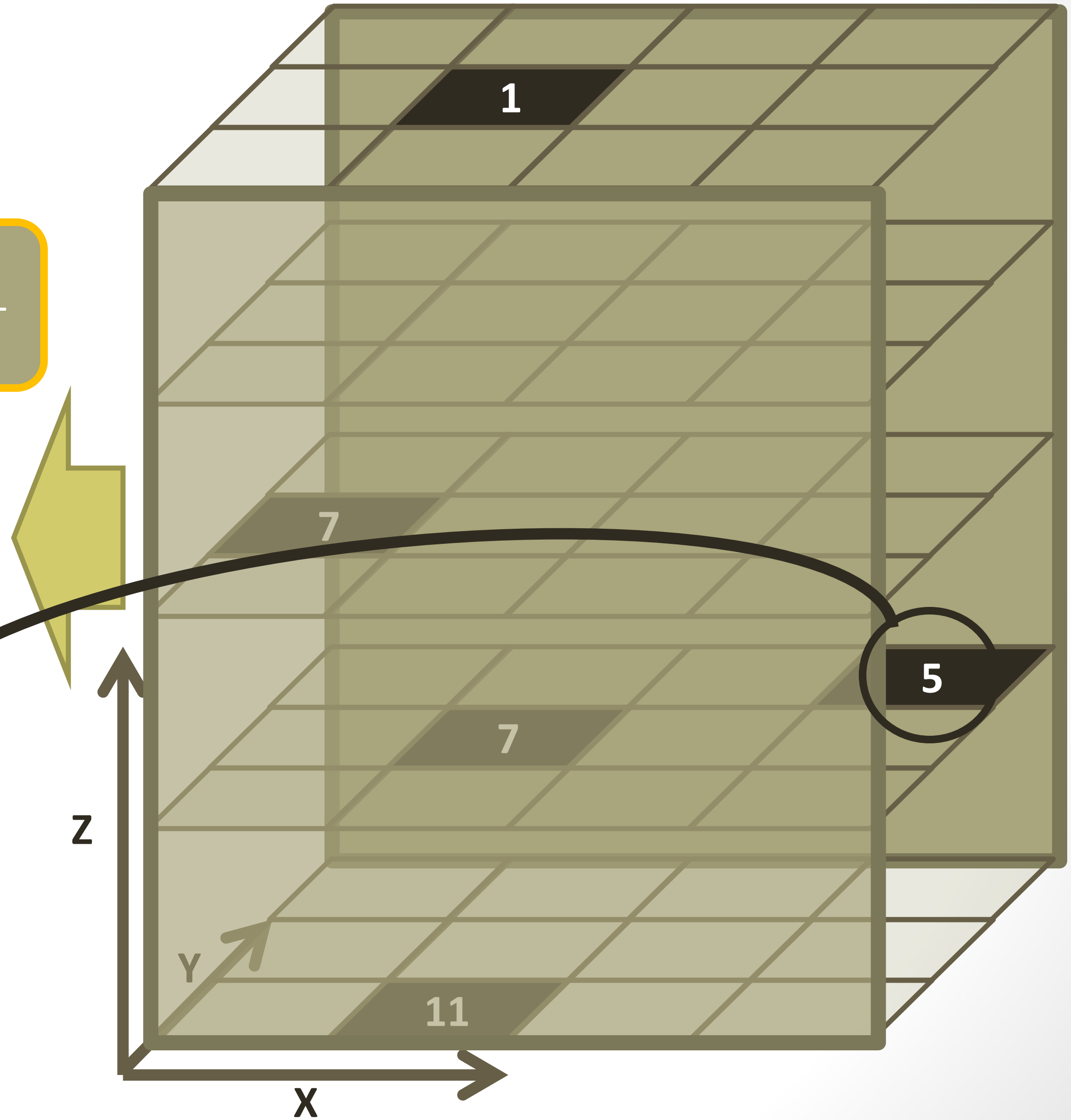
X	Y	Z	V
2	1	1	11
2	2	2	7



Sparse Matrices

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

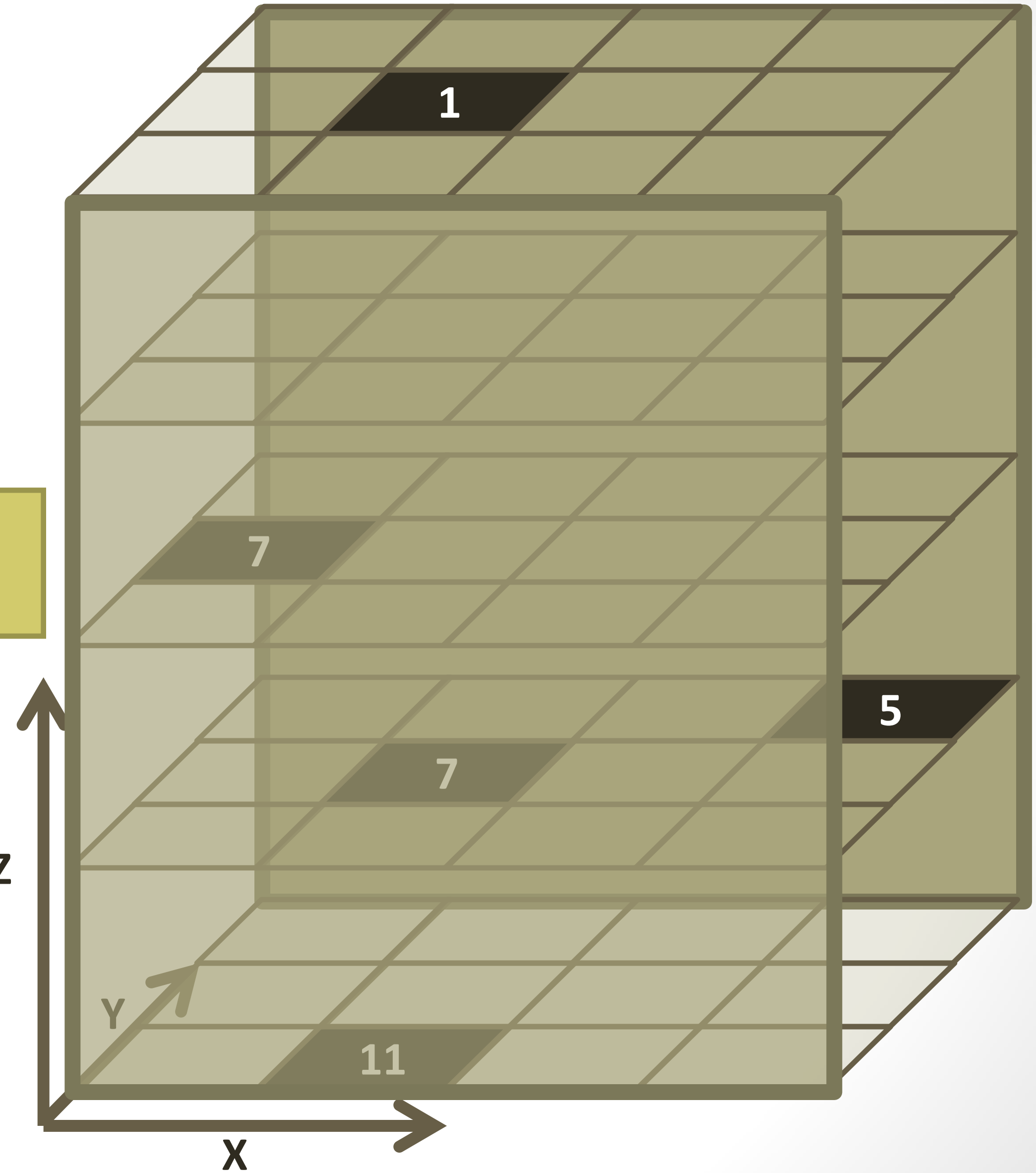
X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5



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

I can 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

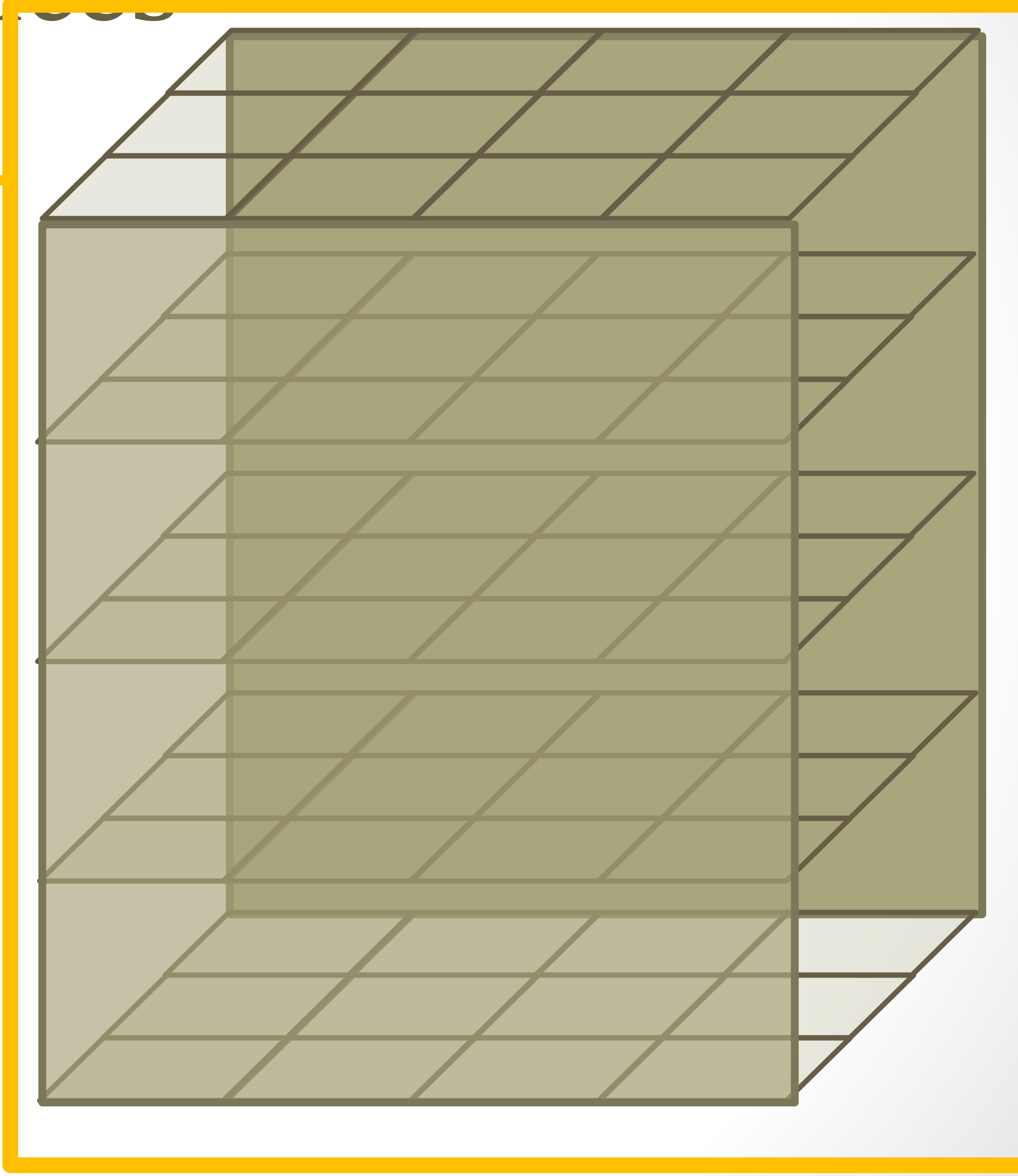
I can think of V as just another dimension

Sparse Matrices

3-Dimensional Space.

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

4-Dimensional Space.

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

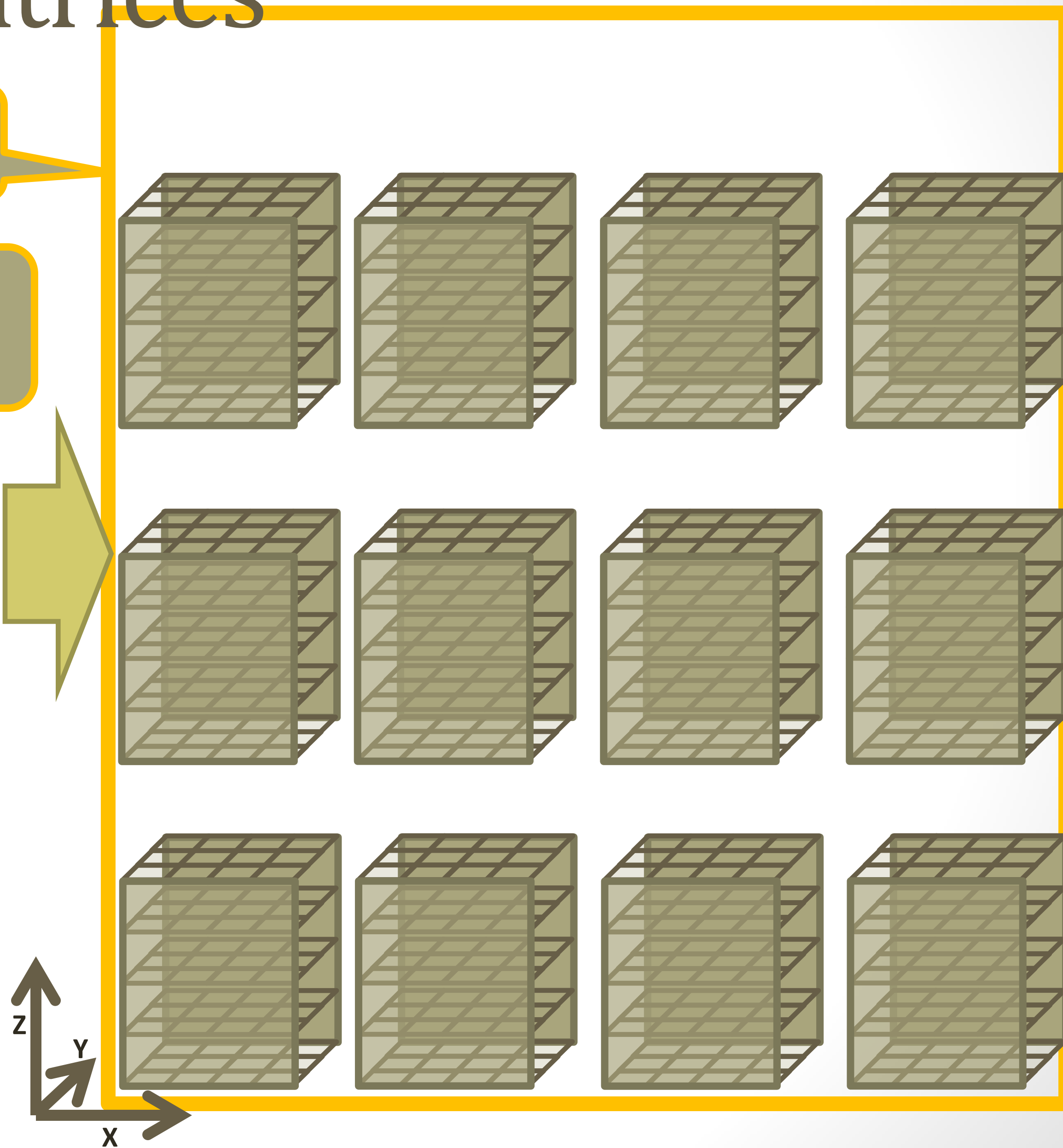
?

Sparse Matrices

4-Dimensional Space with 12 discrete states.

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

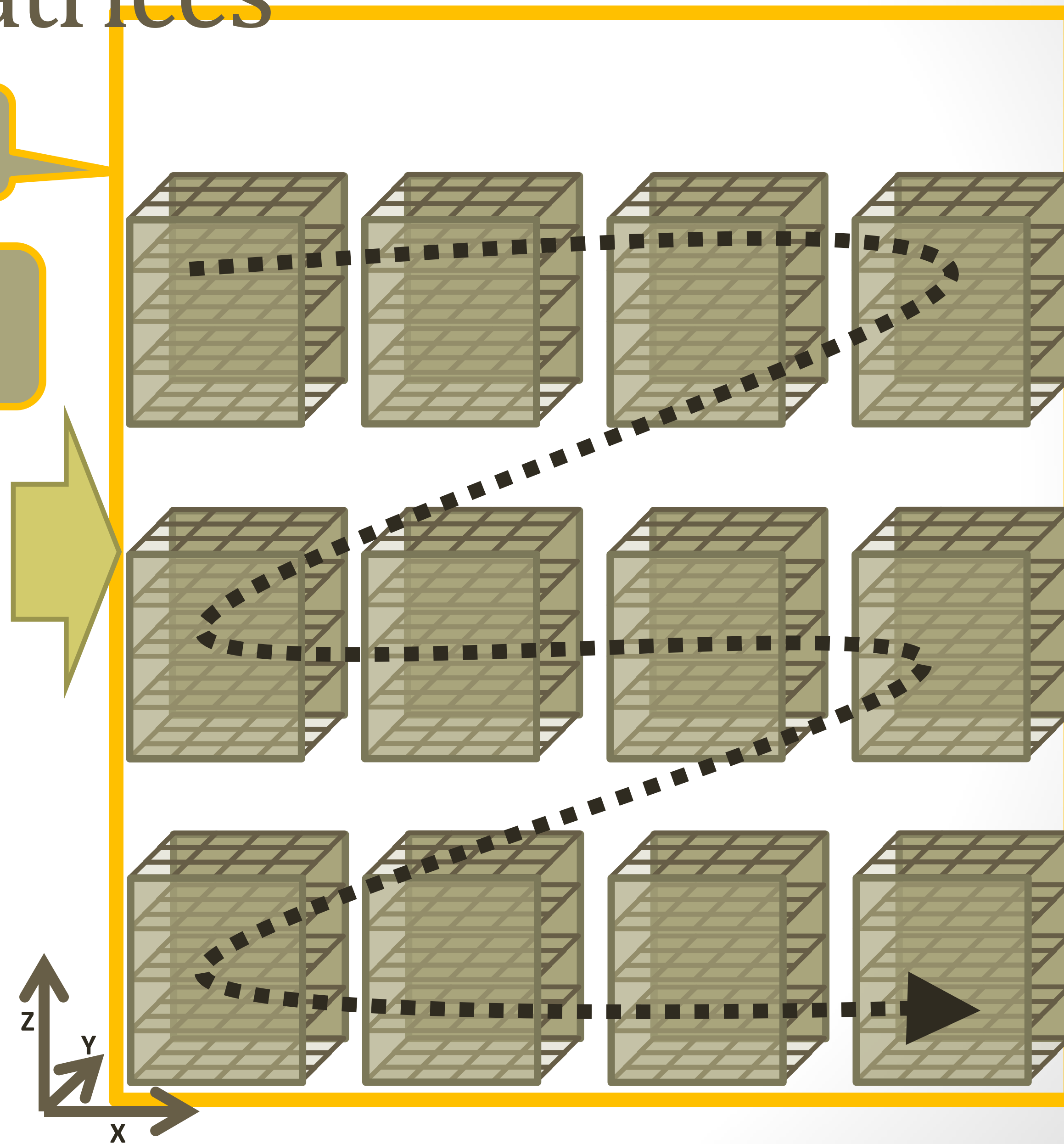


Sparse Matrices

4-Dimensional Space with 12 discrete states.

This table represents points in 4-Dimensional Space.

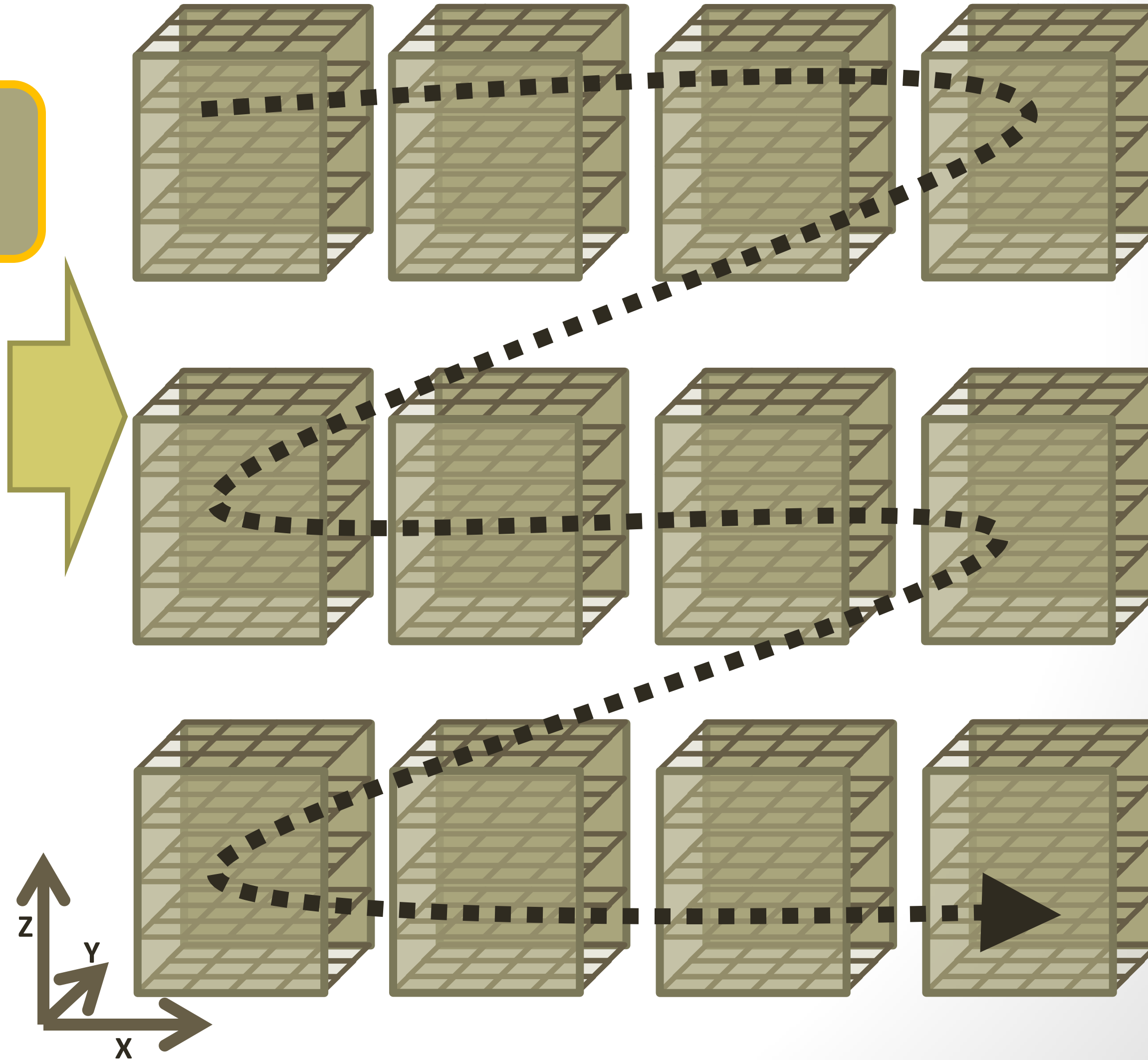
X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1



Sparse Matrices

This table represents points in 4-Dimensional Space.

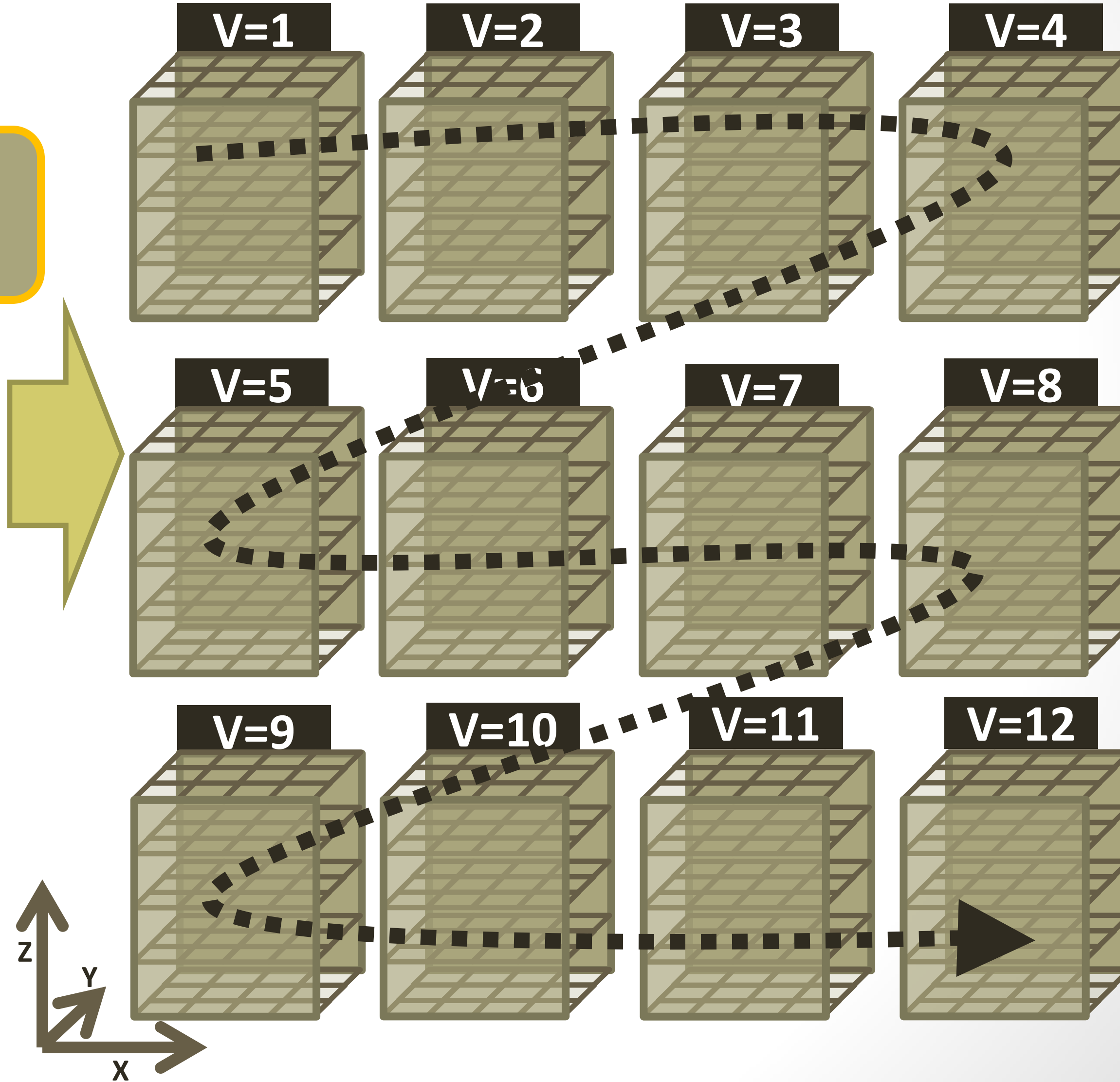
X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1



Sparse Matrices

This table represents points in 4-Dimensional Space.

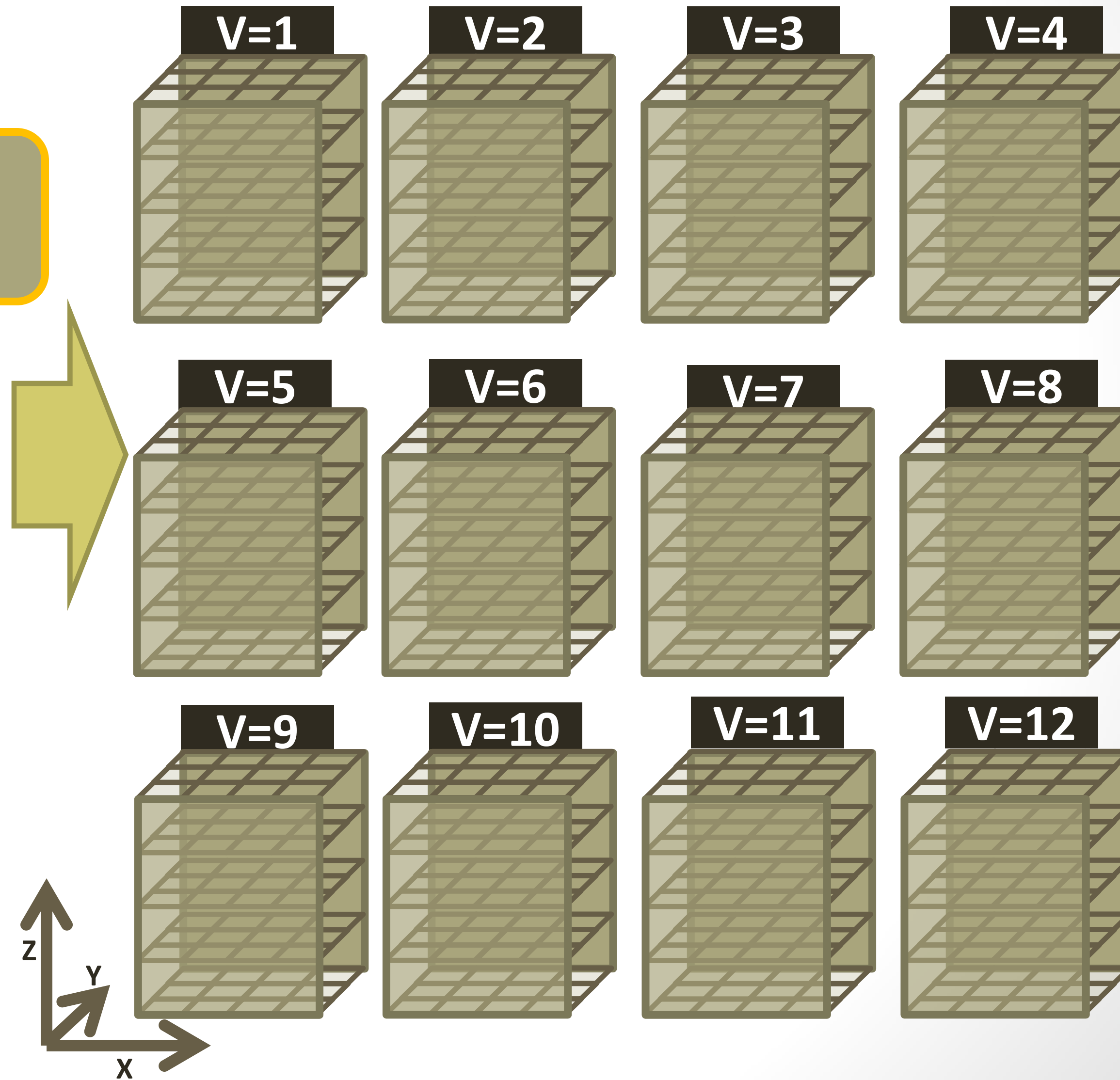
X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1



Sparse Matrices

This table represents points in 4-Dimensional Space.

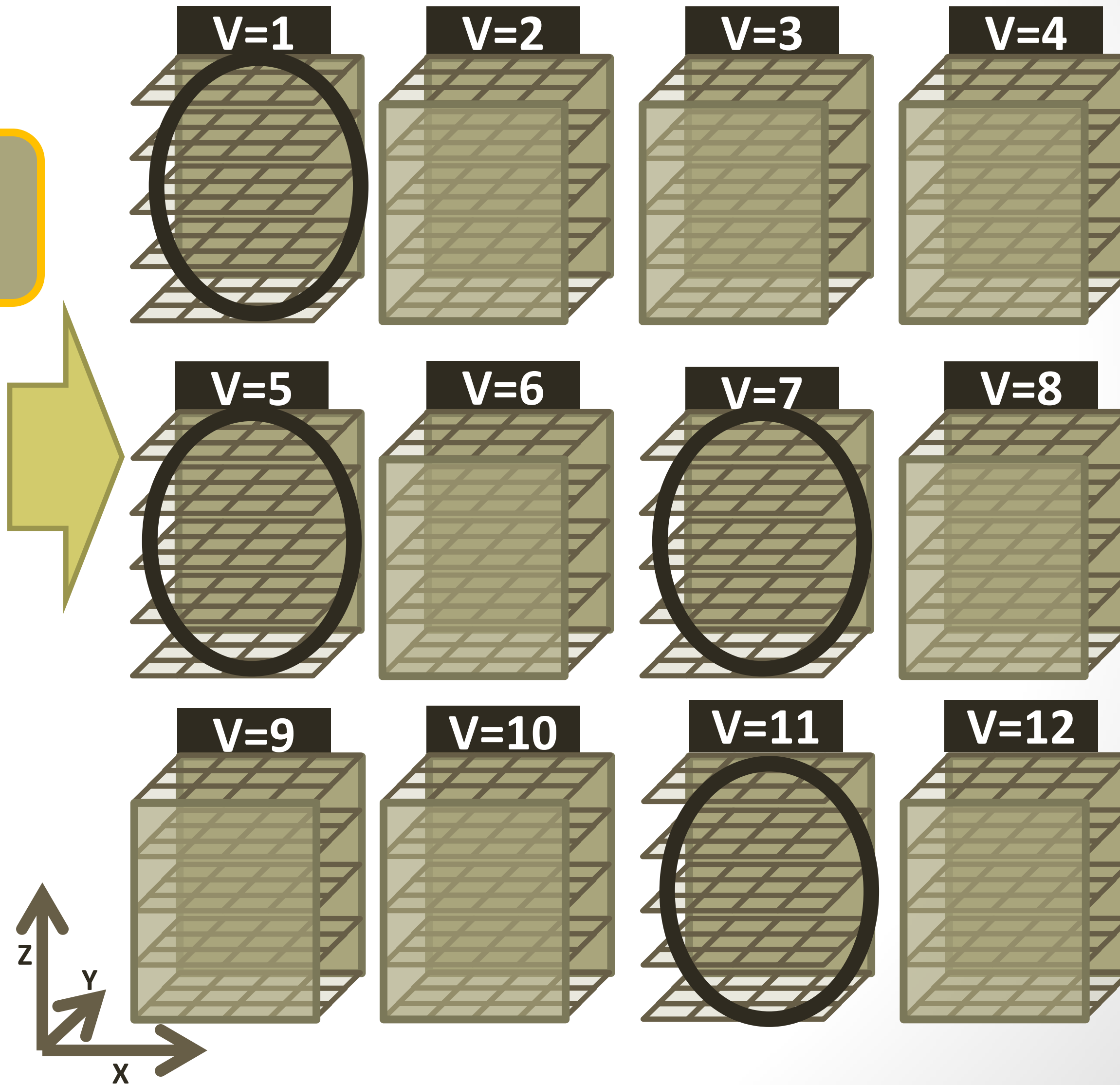
X	Y	Z	V
2	1	1	11
2	2	2	7
4	3	2	5
1	2	3	7
2	2	5	1



Sparse Matrices

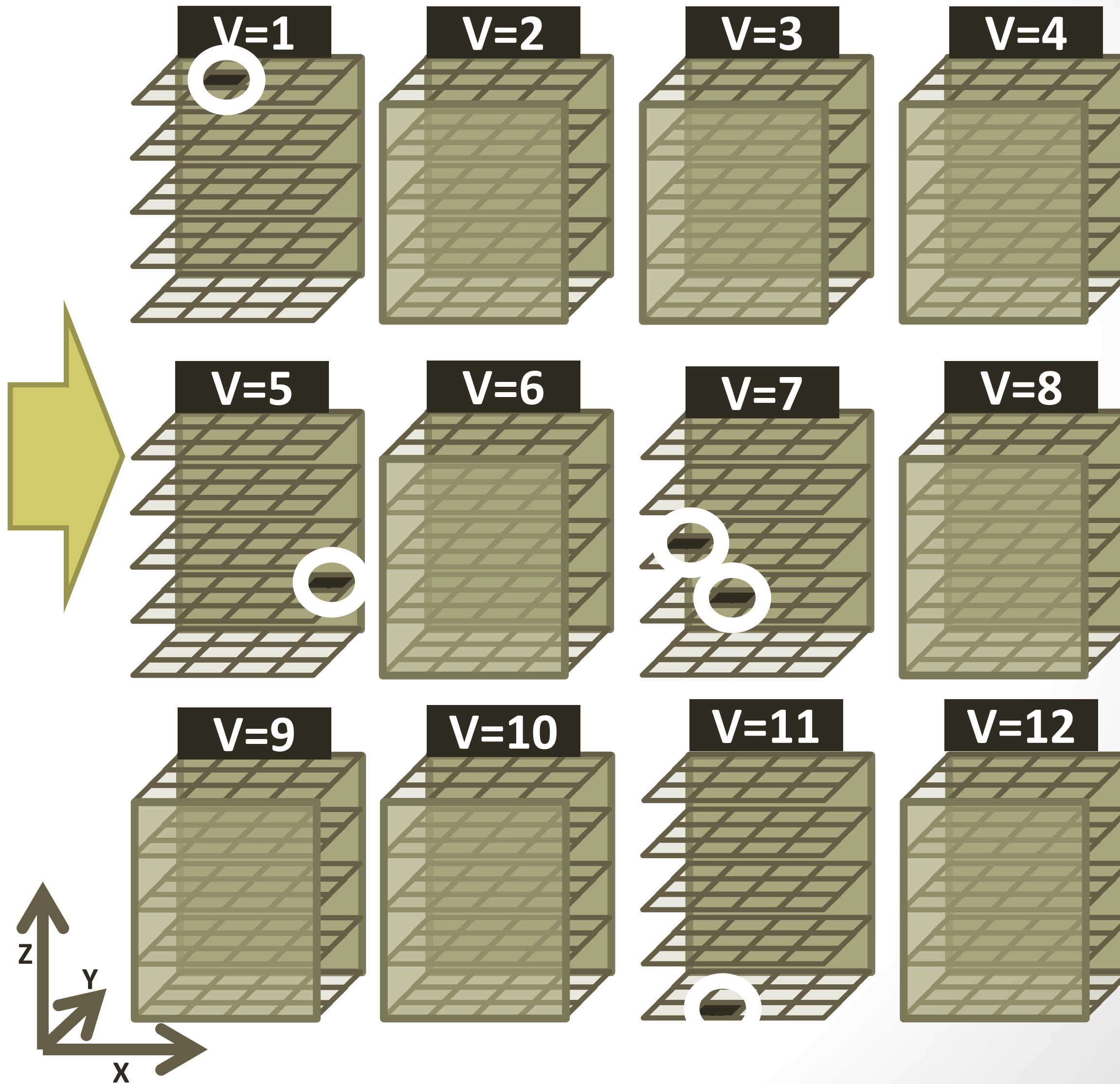
This table represents points in 4-Dimensional Space.

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

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

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

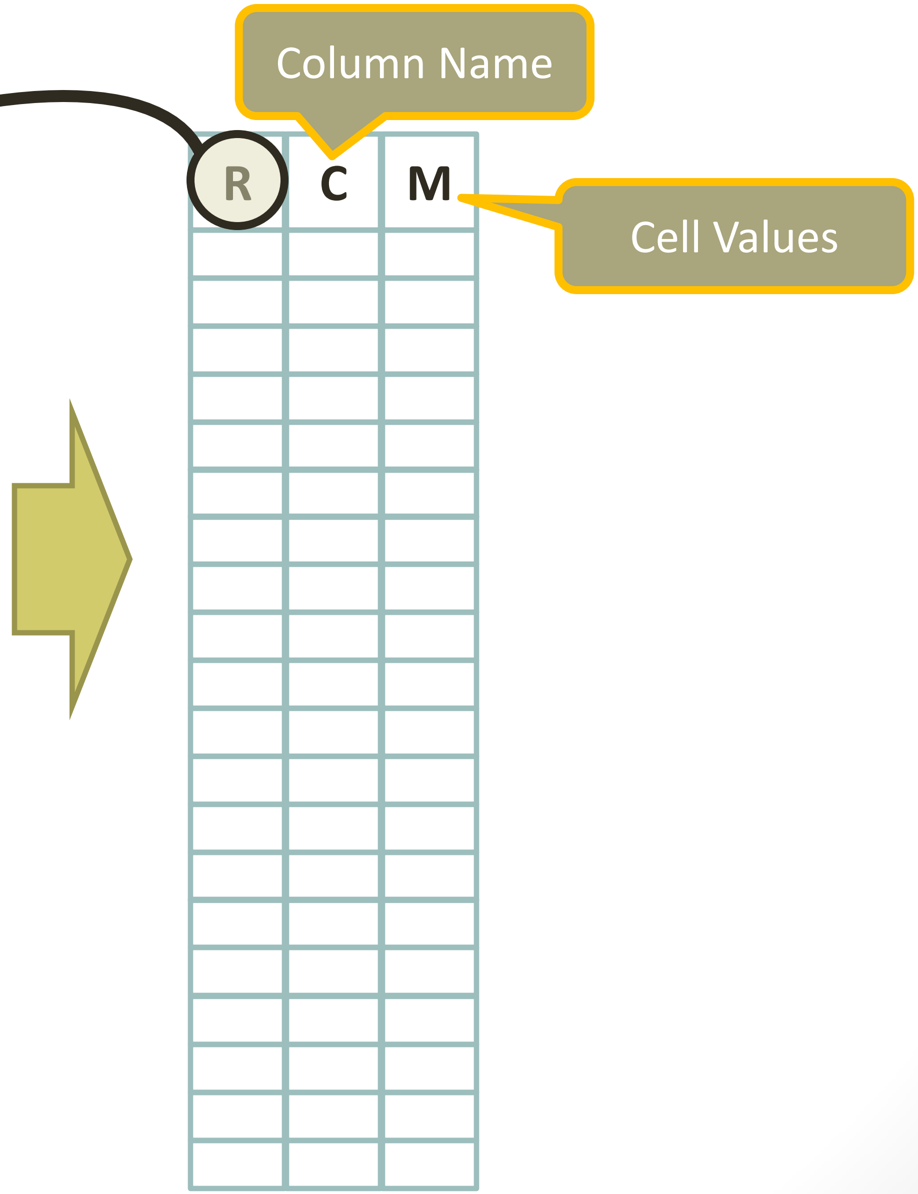
Cell Values

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



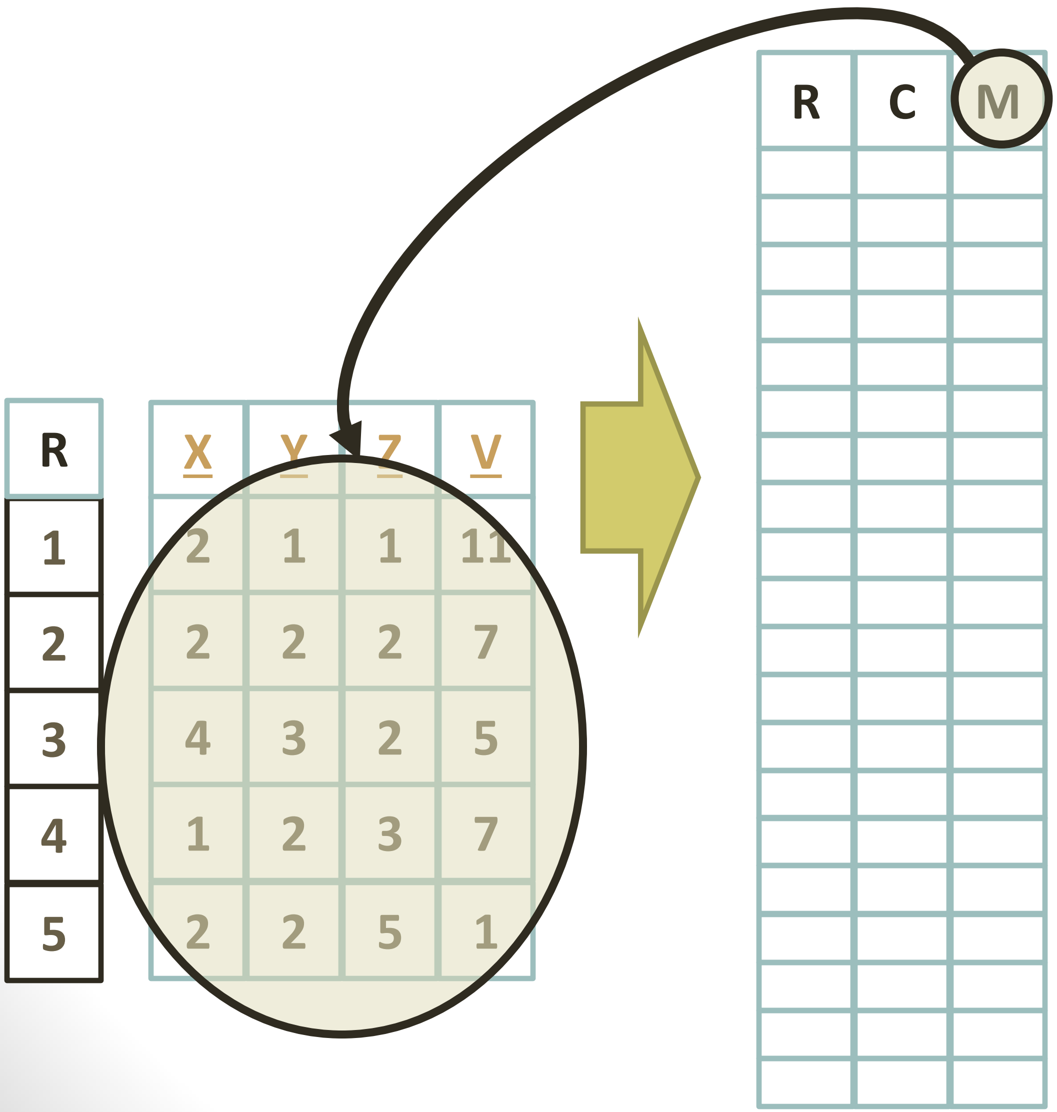
Sparse Matrices: EAV

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

Cell Values

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		

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

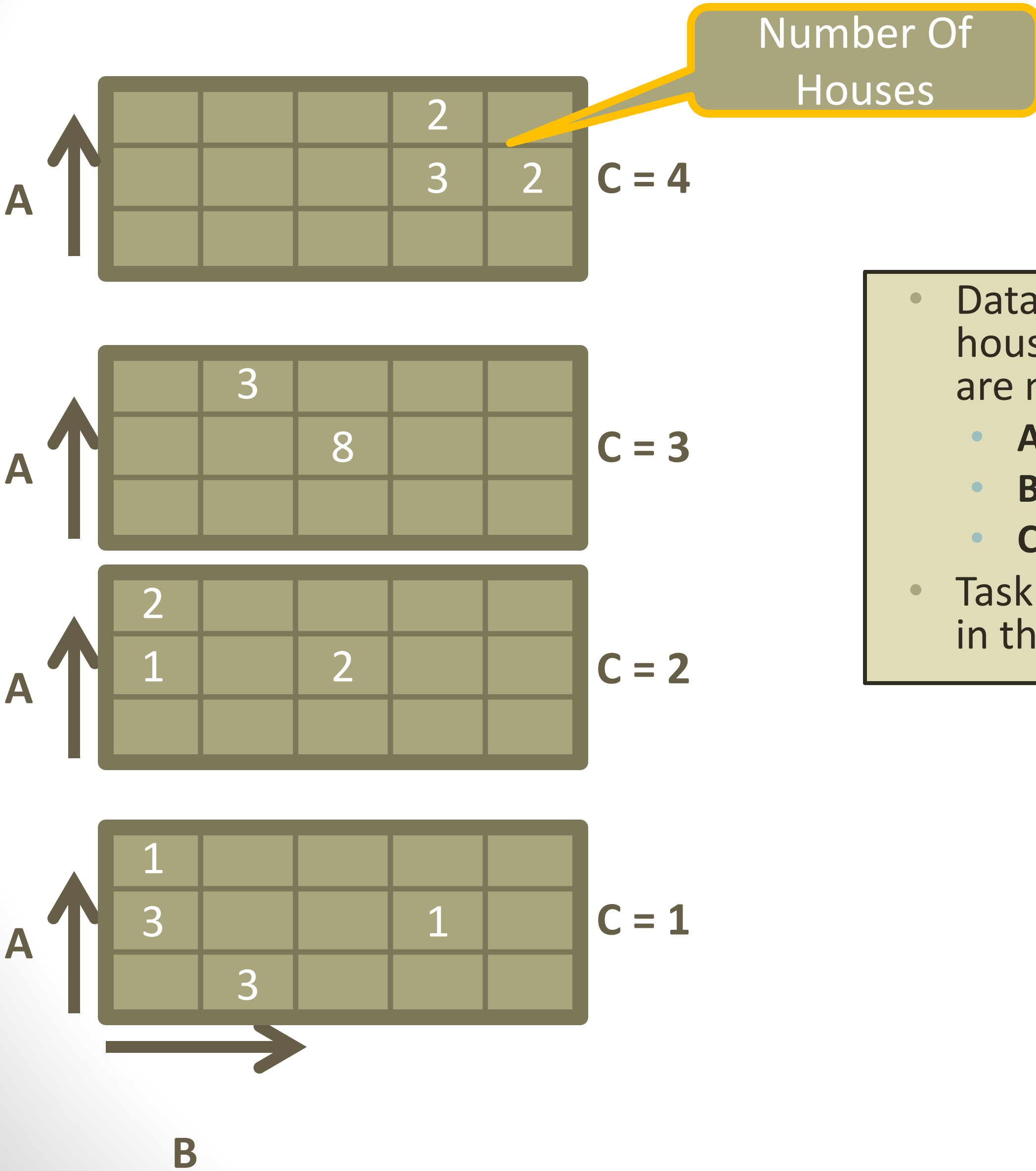
Cell Values

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 08b

- https://catalyst.uw.edu/webq/survey/ernsthe/271576
- 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)

A

			2	
			3	2

C = 4

A

	3			
		8		

C = 3

A

2				
1		2		

C = 2

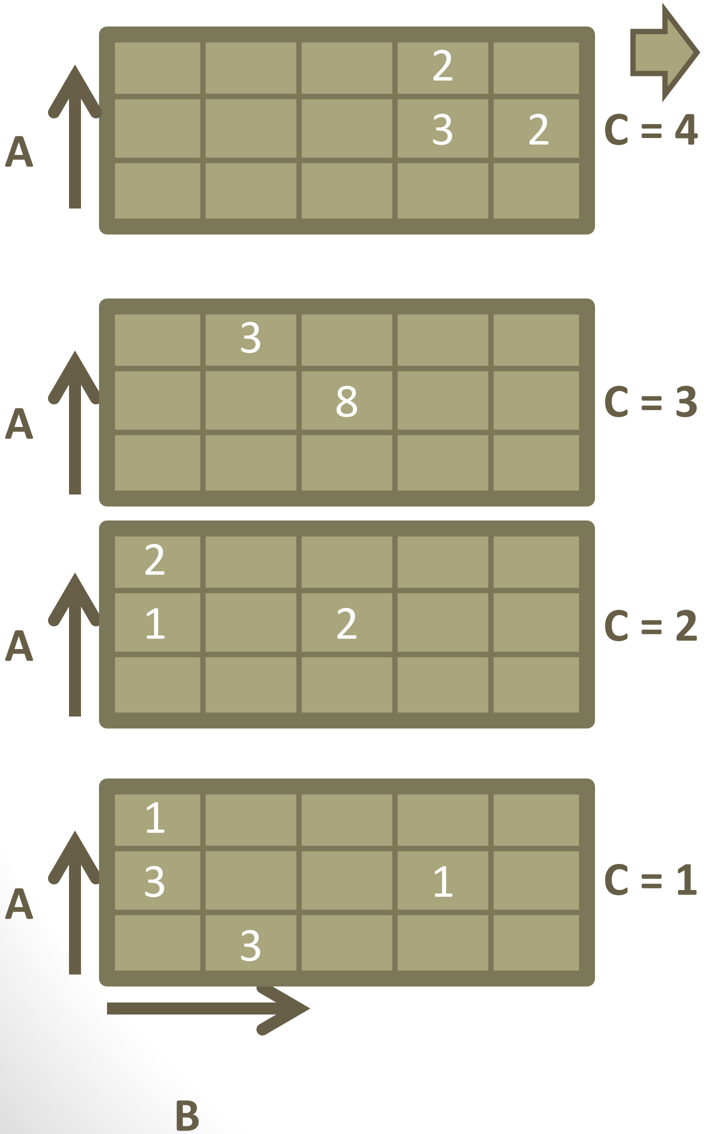
A

1				
3			1	
	3			

C = 1

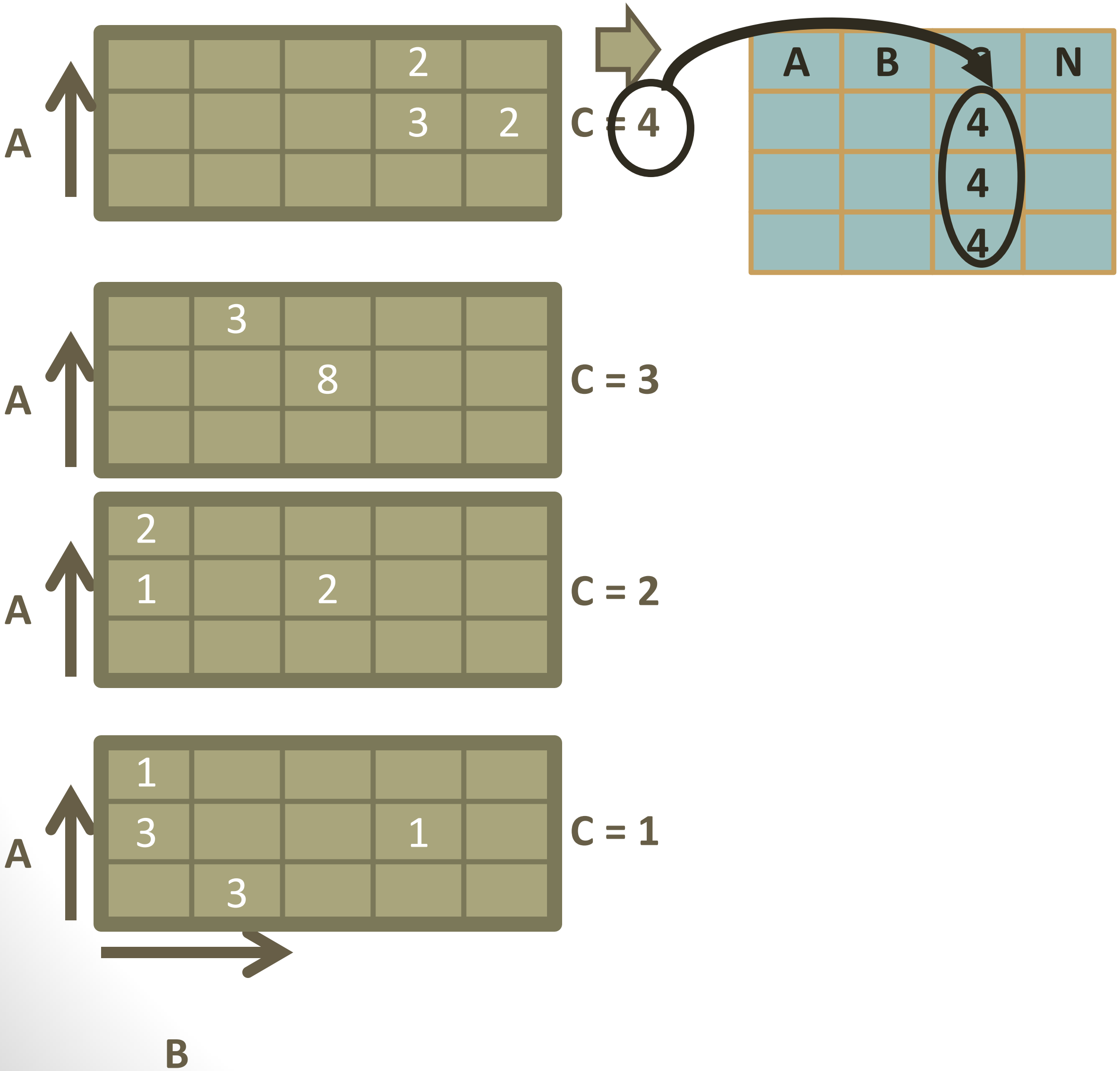
B

Sparse Matrices: Exercise (3)

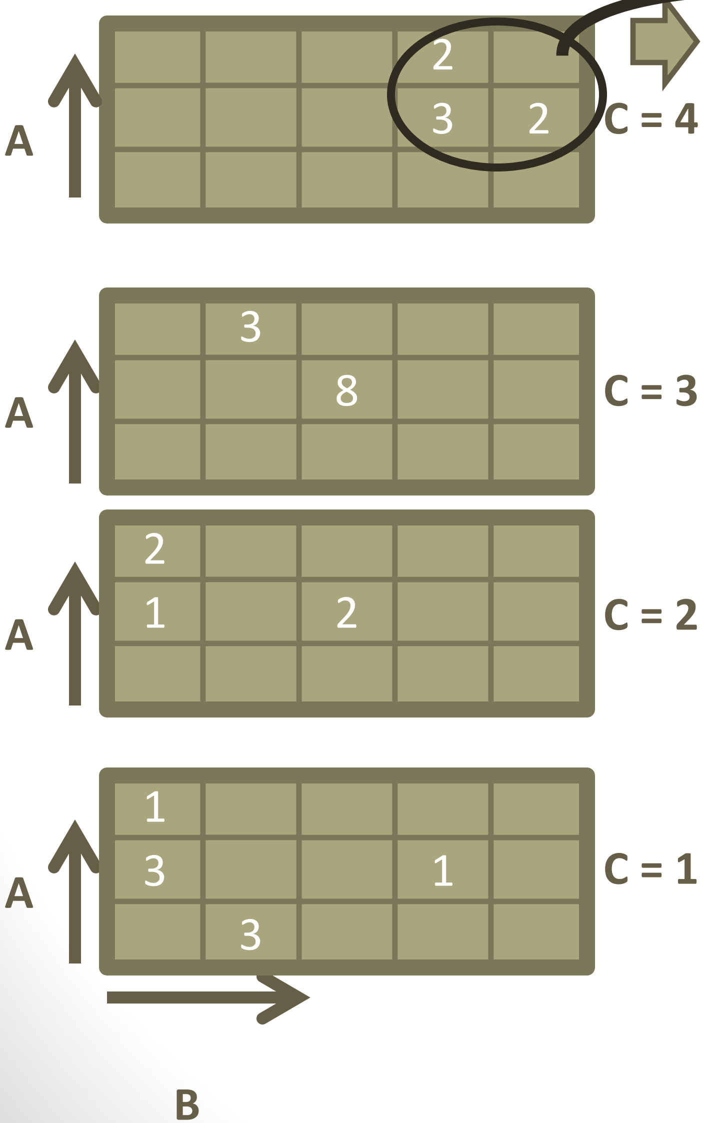


A	B	C	N

Sparse Matrices: Exercise (4)

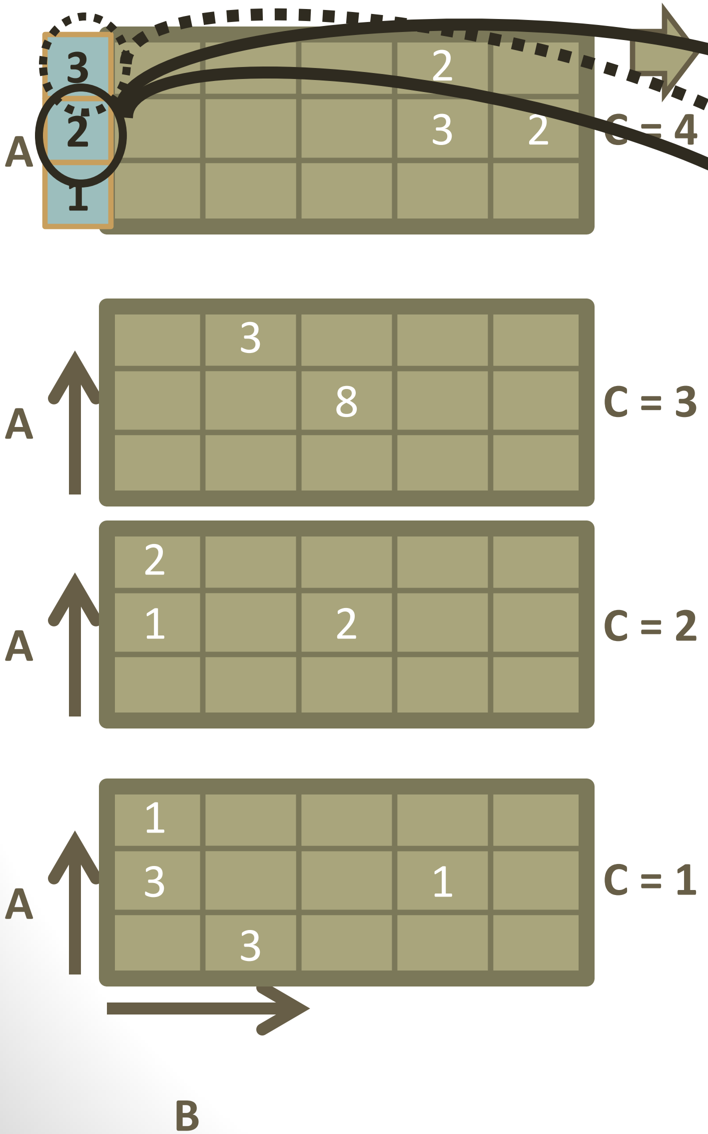


Sparse Matrices: Exercise (5)



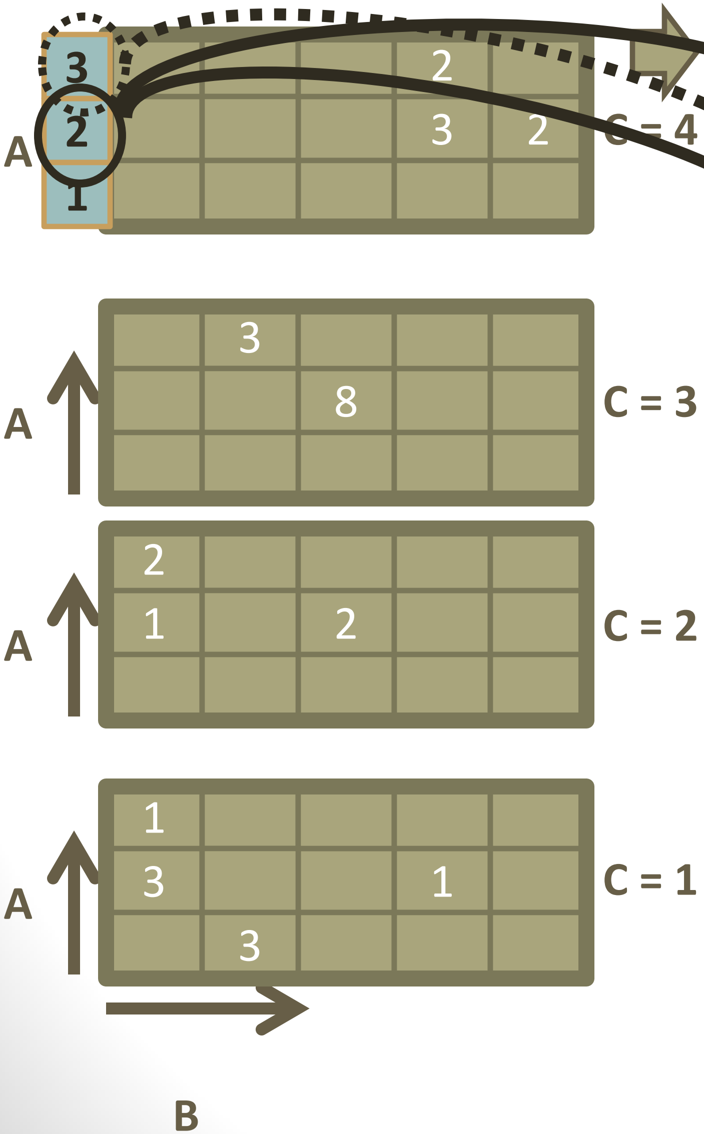
A	B	C	N
		4	2
		4	2
		4	3

Sparse Matrices: Exercise (6)



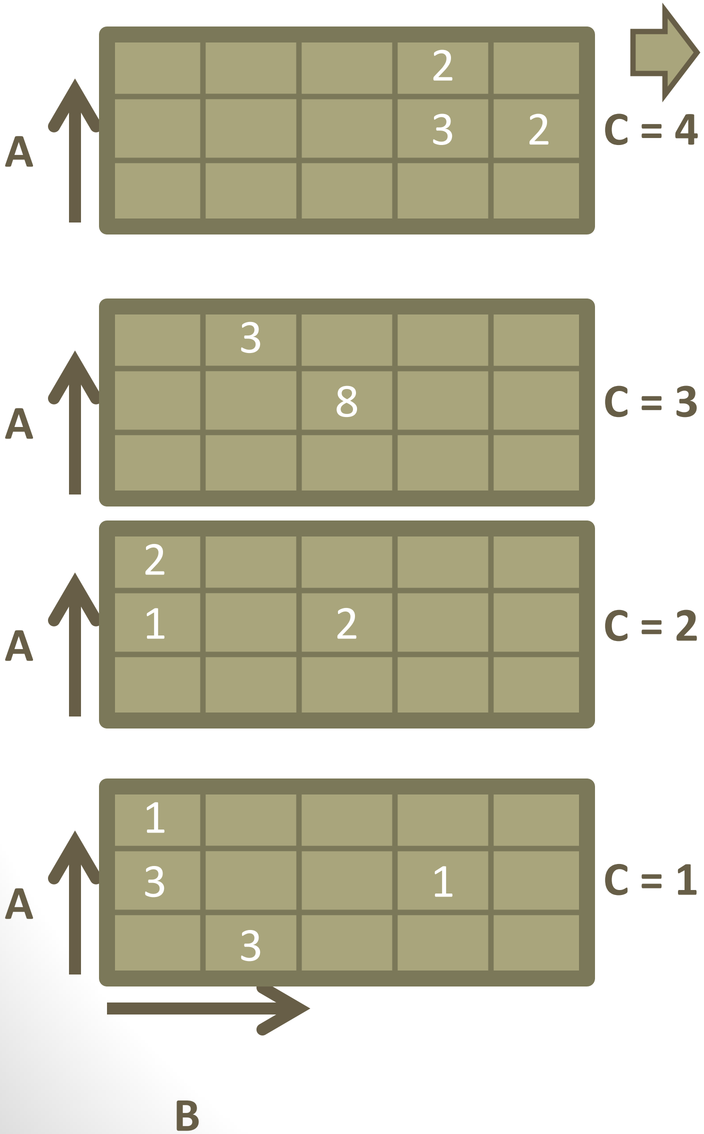
A	B	C	N
2		4	2
3		4	2
2		4	3

Sparse Matrices: Exercise (7)



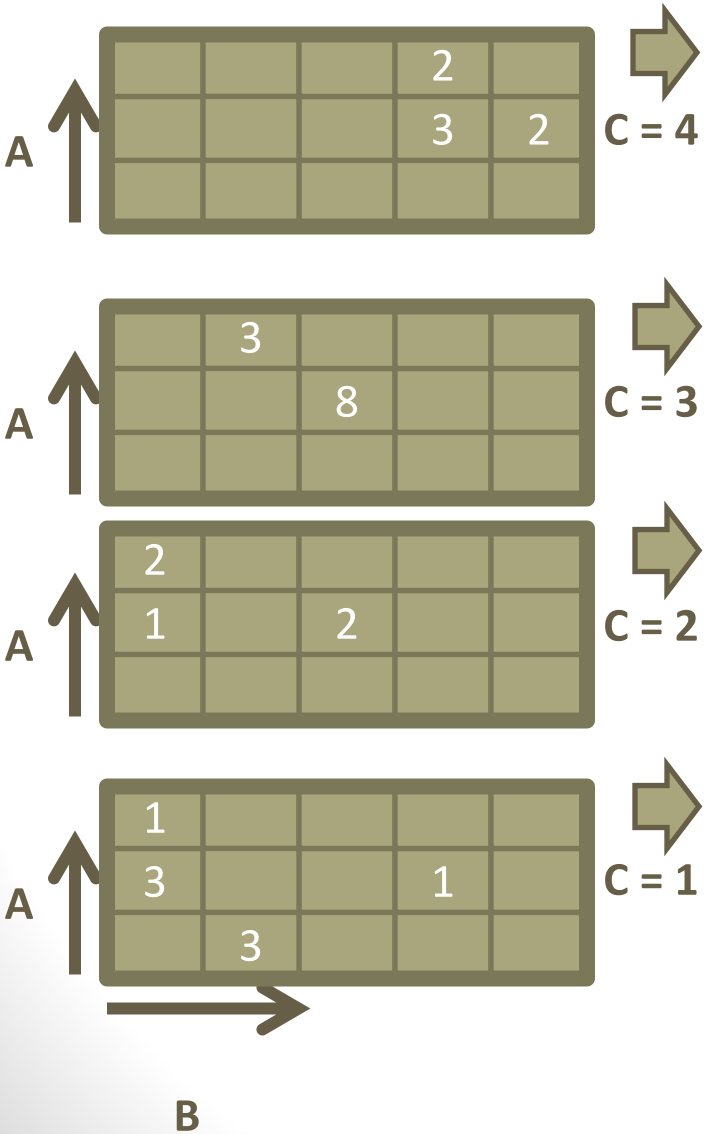
A	B	C	N
2	5	4	2
3	4	4	2
2	4	4	3

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

			2	
			3	2

C = 4

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

A

	3			
		8		

C = 3

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

A

2				
1		2		

C = 2

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

A

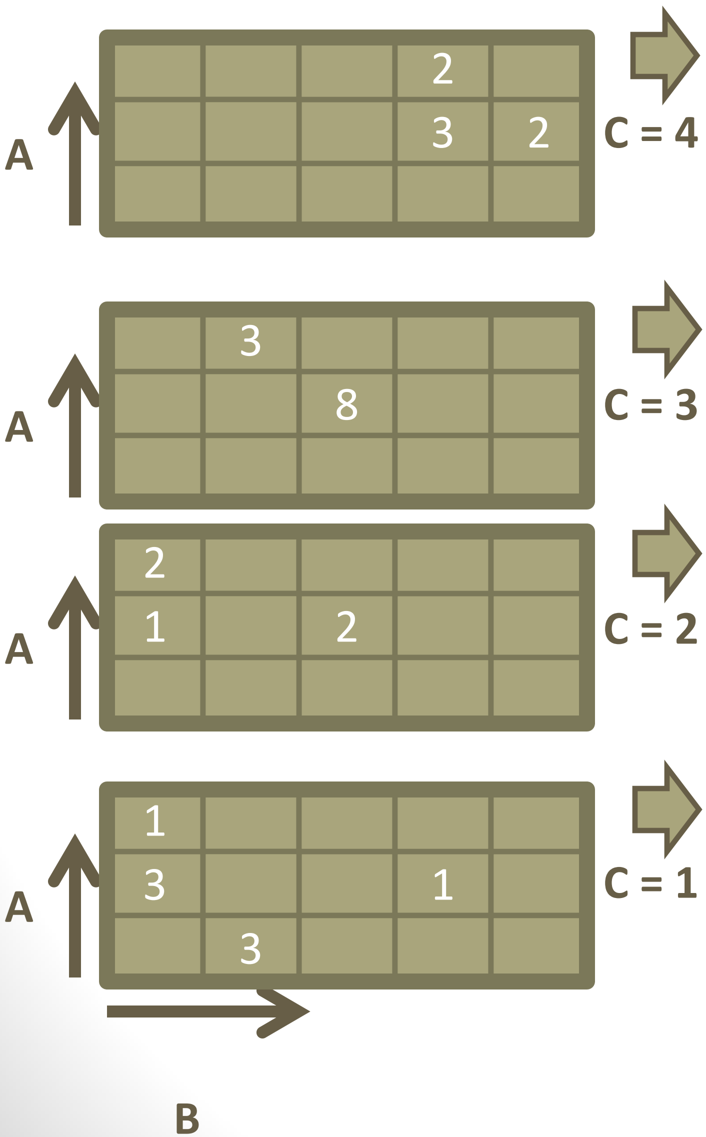
1				
3			1	
	3			

B

C = 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)

	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 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

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

Break

Sparse Matrices Manipulation

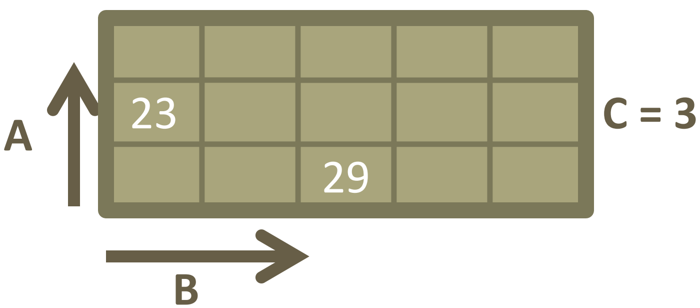
Examples of Sparse Matrix Manipulation in a database

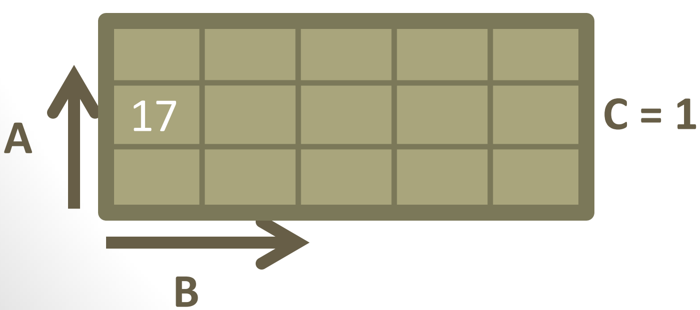
(see MatrixAlgebra.sql)

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

Data as Sparse Matrices

Assignment (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 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.

Assignment (2)

1. Create the two tables that result from the task specified in “Assignment (1)”. Create these tables by “hand”. 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.
2. Change the schema of the data in item 1 above by changing the EAV table, called Table 2. 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 matrices are encoded with the EAV schema do the following:
   a) Write SQL for scalar multiplication of a Matrix in the EAV schema. See Exercise 5 in MatrixAlgebra.sql
   b) Write SQL for transposition of a Matrix in the EAV schema. See Exercise 6 in MatrixAlgebra.sql
   c) Optional: Write SQL for addition of two matrices in the EAV schema. See Exercise 7 in MatrixAlgebra.sql.

Assignment (3)

4. Complete Assignment items 1, 2, and 3. Submit by Sunday 11:00 PM.
5. Read **Graph structure in the web** by Broder et al.:

   http://www.cis.upenn.edu/~mkearns/teaching/NetworkedLife/broder.pdf
6. Read **MapReduce: Simplied Data Processing on Large Clusters**

   http://static.googleusercontent.com/media/research.google.com/es/us/archive/mapreduce-osdi04.pdf
7. Look through the preview section on Catalyst.

Introduction to Data Science