Introduction to Data Science

Lecture 07; May 11th, 2015

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

Agenda

- Social Interactions
  - Get and provide help through the LinkedIn group
  - Encourage Group Homework
- Announcements
- The Science of Data Visualization by Ben Olsen
- Break
- Review Accuracy Measures
  - Homework
  - In-Class Exercise
- Quiz (Accuracy Measures)
- NoSQL: CAP Theorem
- Break
- Relational Algebra (Intro)
- Quiz (Persistence)
- Relational Algebra (continued)

Announcements

- 1-hour guest lecture on May 18th by Marius Marcu “Business Aspects of Data Science” (Changed back to original date)
- May 25th No Class. Memorial Day
- 1-hour guest lecture on June 1st by Matt Danielson “A (brief) introduction to Python for Data Science”

The Science of Data Visualization

Ben Olsen
ben.olsen@matisia.com

Break

Accuracy Measures Exercise

Homework Review

- Question: Why are performance metrics better on training data than on test data?
  - Answer: Because model is optimized for (trained on) training data
- Question: How do you determine which data are training data and which data are test data?
  - Answer a: Prior to training the determination is random.
  - Answer b: After training you can identify the training data in that the model is optimized for those data.

Homework Review

- The Confusion Matrix
  - Calculate the accuracy measures including the F-measure for the Homework. Positive and negative are just points-of-view:
    - Illness is positive (as in a test to determine if one is ill)
    - Health is positive (as in: its positive to be healthy)

Homework Review

- A model was trained on 300 individuals where 149 had the cold and 151 were healthy.
  - These numbers are irrelevant.
  - The accuracy measures are assessed by predictions and the test data.
  - Accuracy is not assessed with the training data.
- The model was tested on 100 individuals where 10 were ill.
  - Total population: 100
  - Support for ill: 10
  - Therefore, support for healthy: 90
- The model correctly predicted that 85 of the healthy individuals were indeed healthy
  - Correct predictions of healthy: 85
  - Therfore, incorrect prediction of ill (they were actually healthy): 5
  - (90 healthy - 85 correct predictions of healthy -> 5 healthy that were not predicted as healthy)
- and correctly predicted that 7 of the ill individuals were indeed ill.
  - Correct predictions of ill: 7
  - Therefore, incorrect prediction of healthy (they were actually ill): 3
  - (10 ill - 7 correct predictions of ill -> 3 ill that were not predicted as ill)

Homework Review

85 predicted healthy	**and were healthy**
3 predicted healthy	**but were ill**
5 predicted ill	**but were healthy**
7 predicted ill	**and were ill**

- A model was trained on 300 individuals where 149 had the cold and 151 were healthy.
  - These numbers are irrelevant.
  - The accuracy measures are assessed by predictions and the test data.
  - Accuracy is not assessed with the training data.
- The model was tested on 100 individuals where 10 were ill.
  - Total population: 100
  - Support for ill: 10
  - Therefore, support for healthy: 90
- The model correctly predicted that 85 of the healthy individuals were indeed healthy
  - Correct predictions of healthy: 85
  - Therfore, incorrect prediction of ill (they were actually healthy): 5
  - (90 healthy - 85 correct predictions of healthy -> 5 healthy that were not predicted as healthy)
- and correctly predicted that 7 of the ill individuals were indeed ill.
  - Correct predictions of ill: 7
  - Therefore, incorrect prediction of healthy (they were actually ill): 3
  - (10 ill - 7 correct predictions of ill -> 3 ill that were not predicted as ill)

Homework Review

85 predicted healthy and were healthy
3 predicted healthy but were ill
5 predicted ill but were healthy
7 predicted ill and were ill

	Actual	Predicted
Healthy	90	88
Ill	10	12

- A model was trained on 300 individuals where 149 had the cold and 151 were healthy.
  - These numbers are irrelevant.
  - The accuracy measures are assessed by predictions and the test data.
  - Accuracy is not assessed with the training data.
- The model was tested on 100 individuals where 10 were ill.
  - Total population: 100
  - Support for ill: 10
  - Therefore, support for healthy: 90
- The model correctly predicted that 85 of the healthy individuals were indeed healthy
  - Correct predictions of healthy: 85
  - Therfore, incorrect prediction of ill (they were actually healthy): 5
  - (90 healthy - 85 correct predictions of healthy -> 5 healthy that were not predicted as healthy)
- and correctly predicted that 7 of the ill individuals were indeed ill.
  - Correct predictions of ill: 7
  - Therefore, incorrect prediction of healthy (they were actually ill): 3
  - (10 ill - 7 correct predictions of ill -> 3 ill that were not predicted as ill)

Homework: Confusion Matrix

85 predicted healthy and were healthy
3 predicted healthy but were ill
5 predicted ill but were healthy
7 predicted ill and were ill

	Actual	Predicted
Healthy	90	88
Ill	10	12

Homework: Confusion Matrix

85 predicted healthy and were healthy
3 predicted healthy but were ill
5 predicted ill but were healthy
7 predicted ill and were ill

Homework: Confusion Matrix

85 predicted healthy and were healthy
3 predicted healthy but were ill
5 predicted ill but were healthy
7 predicted ill and were ill

Positive and negative are just points-of-view:

- Illness could be positive (as in a test to determine if one is ill)
- Health could be positive (as in: it's a positive thing to be healthy)

Homework: Confusion Matrix

85 predicted healthy and were healthy
3 predicted healthy but were ill
5 predicted ill but were healthy
7 predicted ill and were ill

Health is Positive

Predicted		Actual P	Actual N
	P’	TP	FP
	N’	FN	TN

Homework: Confusion Matrix

85 predicted healthy and were healthy
3 predicted healthy but were ill
5 predicted ill but were healthy
7 predicted ill and were ill

Predicted \ Actual	P	N
P’	85	3
N’	5	7

Health is Positive

Predicted \ Actual	P	N
P’	TP	FP
N’	FN	TN

Homework: Confusion Matrix

85 predicted healthy and were healthy
3 predicted healthy but were ill
5 predicted ill but were healthy
7 predicted ill and were ill

Actual

Predicted	P	N
P’	85	3
N’	5	7

Health is Positive

Illness is Positive

Actual

Predicted	P	N
P’	TP	FP
N’	FN	TN

Predicted	P	N
P’	TP	FP
N’	FN	TN

Homework: Confusion Matrix

85 predicted healthy and were healthy
3 predicted healthy but were ill
5 predicted ill but were healthy
7 predicted ill and were ill

Predicted \ Actual	P	N
P’	85	3
N’	5	7

Health is Positive

Predicted \ Actual	P	N
P’	7	5
N’	3	85

Illness is Positive

Predicted \ Actual	P	N
P’	TP	FP
N’	FN	TN

Predicted \ Actual	P	N
P’	TP	FP
N’	FN	TN

Homework: Confusion Matrix

85 predicted healthy and were healthy
3 predicted healthy but were ill
5 predicted ill but were healthy
7 predicted ill and were ill

Predicted \ Actual	P	N
P'	85	3
N'	5	7

Health is Positive

Predicted \ Actual	P	N
P'	7	5
N'	3	85

Illness is Positive

Homework: Confusion Matrix

85 predicted healthy and were healthy
3 predicted healthy but were ill
5 predicted ill but were healthy
7 predicted ill and were ill

		Actual	
		P	N
Predicted	P’	85	3
	N’	5	7

Health is Positive

- True Positive: 85
- True Negative: 7
- False Positive: 3
- False Negative: 5

		Actual	
		P	N
Predicted	P’	7	5
	N’	3	85

Homework: Confusion Matrix

85 predicted healthy and were healthy
3 predicted healthy but were ill
5 predicted ill but were healthy
7 predicted ill and were ill

Predicted / Actual	P	N
P’	85	3
N’	5	7

Health is Positive

- True Positive: 85
- True Negative: 7
- False Positive: 3
- False Negative: 5

Predicted / Actual	P	N
P’	7	5
N’	3	85

Illness is Positive

- True Positive: 7
- True Negative: 85
- False Positive: 5
- False Negative: 3

Homework: Confusion Matrix

85 predicted healthy and were healthy
3 predicted healthy but were ill
5 predicted ill but were healthy
7 predicted ill and were ill

Predicted \ Actual	P	N
P’	85	3
N’	5	7

Health is Positive

- True Positive: 85
- True Negative: 7
- False Positive: 3
- False Negative: 5

Predicted \ Actual	P	N
P’	7	5
N’	3	85

Illness is Positive

- True Positive: 7
- True Negative: 85
- False Positive: 5
- False Negative: 3

- Sensitivity*: tp / (tp + fn)
- Specificity: tn/(tn + fp)
- Accuracy: (tp + tn) /(tp + fp + tn + fn)
- Precision : tp/(tp + fp)
- Recall*: tp/(tp + fn)
- F-measure: 2tp/(2tp + fn + fp)

Homework: Confusion Matrix

85 predicted healthy and were healthy
3 predicted healthy but were ill
5 predicted ill but were healthy
7 predicted ill and were ill

	Actual P	Actual N
Predicted P’	85	3
Predicted N’	5	7

Health is Positive
- True Positive: 85
- True Negative: 7
- False Positive: 3
- False Negative: 5

	Actual P	Actual N
Predicted P’	7	5
Predicted N’	3	85

Illness is Positive
- True Positive: 7
- True Negative: 85
- False Positive: 5
- False Negative: 3

- Sensitivity*: tp / (tp + fn)
- Specificity: tn/(tn + fp)
- Accuracy: (tp + tn) /(tp + fp + tn + fn)
- Precision : tp/(tp + fp)
- Recall*: tp/(tp + fn)
- F-measure: 2tp/(2tp + fn + fp)

- Sensitivity*: 0.94
- Specificity: 0.7
- Accuracy: 0.92
- Precision: 0.97
- Recall*: 0.94
- F-measure: 0.95

Homework: Confusion Matrix

85 predicted healthy and were healthy
3 predicted healthy but were ill
5 predicted ill but were healthy
7 predicted ill and were ill

Predicted \ Actual	P	N
P’	85	3
N’	5	7

Health is Positive

- True Positive: 85
- True Negative: 7
- False Positive: 3
- False Negative: 5

Predicted \ Actual	P	N
P’	7	5
N’	3	85

Illness is Positive

- True Positive: 7
- True Negative: 85
- False Positive: 5
- False Negative: 3

- Sensitivity*: tp / (tp + fn)
- Specificity: tn/(tn + fp)
- Accuracy: (tp + tn) /(tp + fp + tn + fn)
- Precision : tp/(tp + fp)
- Recall*: tp/(tp + fn)
- F-measure: 2tp/(2tp + fn + fp)

Health is Positive:

- Sensitivity*: 0.94
- Specificity: 0.7
- Accuracy: 0.92
- Precision: 0.97
- Recall*: 0.94
- F-measure: 0.95

Illness is Positive:

- Sensitivity*: 0.7
- Specificity: 0.94
- Accuracy: 0.92
- Precision: 0.58
- Recall*: 0.7
- F-measure: 0.63

Homework 06 Problem 5 (0)

- ClassificationAccuracy.R

Homework 06 Problem 5 (1)

- # Problem statement
- # I   A Classification is tested on 1000 cases.
- # II The false positive rate is 0.4
- # III The true positive rate is 0.8.
- # IV The accuracy is 0.7.

- # Problem statement expressed using TP, FP, FN, TN
- # I   N = TP + FP + FN + TN   = 1000
- # II FPR = FP/(FP + TN) = 0.4
- # III TPR = TP/(TP + FN) = 0.8
- # IV (TP + TN)/(TP + FP + FN + TN) = 0.7

- # Problem statement expressed as linear equations

```
# I       1*TP + 1*FP + 1*FN + 1*TN  = 1000
# II      0    + 3*FP + 0    - 2*TN  = 0
# III     1*TP + 0    - 4*FN + 0     = 0
# IV     -3*TP + 7*FP + 7*FN - 3*TN  = 0
```

Homework 06 Problem 5 (2)

- # Problem statement expressed as linear equations
- # I   1*TP + 1*FP + 1*FN + 1*TN = 1000
- # II   0   + 3*FP + 0   - 2*TN = 0
- # III   1*TP + 0   - 4*FN + 0   = 0
- # IV   -3*TP + 7*FP + 7*FN - 3*TN = 0

- # Problem statement expressed in terms of linear algebra:
- # We want to solve the linear equation: Ax = b
- # Where:
- # A is the matrix
- # x is a vector of TP, FP, FN, TN
- # b is the right-hand side of the linear equation

```
# --------------     --------
#   matrix A         vector b
# --------------     --------
# TP  FP  FN  TN   | b
# --------------     --------
#  1   1   1   1   | 1000
#  0   3   0  -2   | 0
#  1   0  -4   0   | 0
# -3   7   7  -3   | 0
# --------------     --------
```

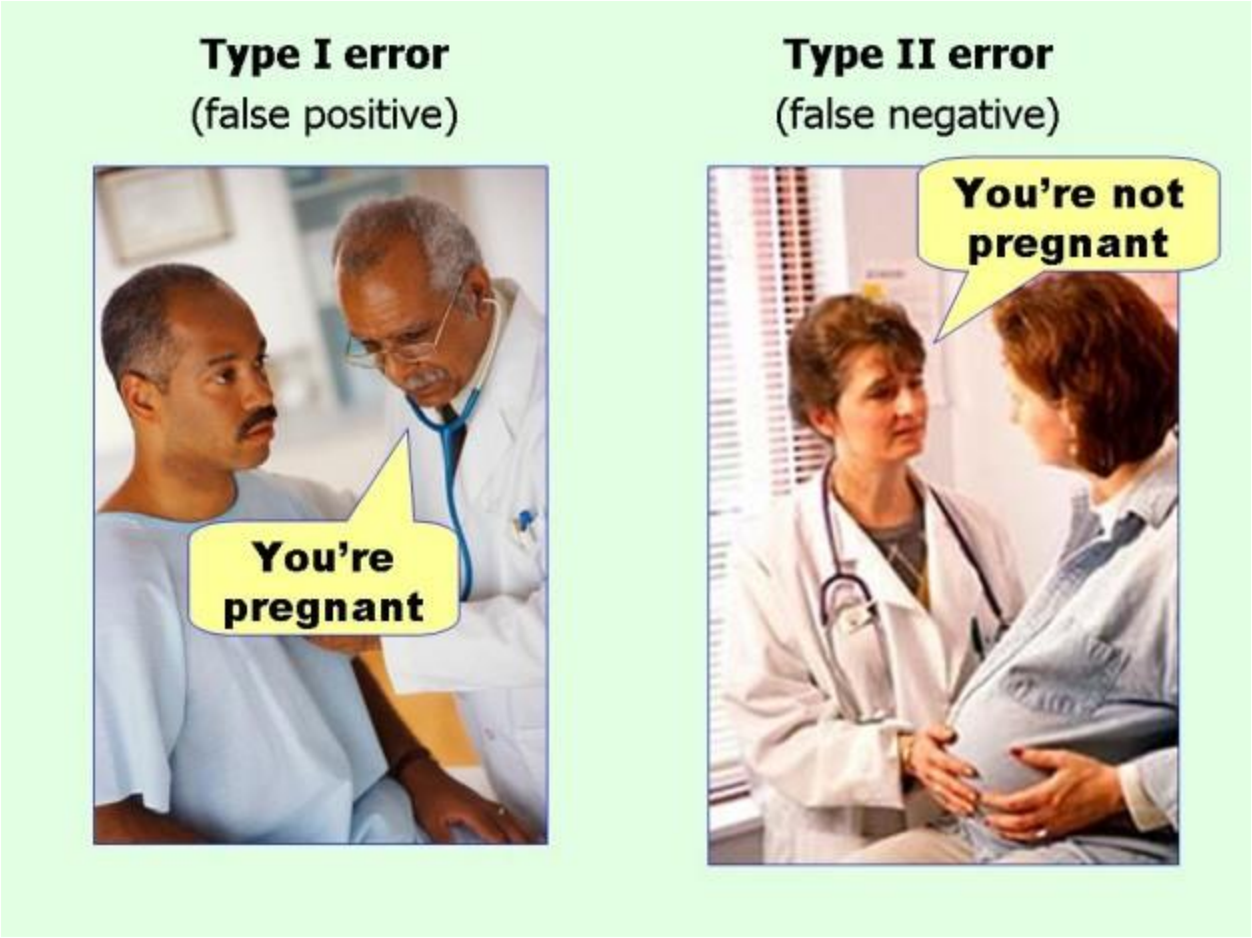
Homework 06 Problem 6

- HowToMakeAnROC_Results.xls

Accuracy

- Links
  - http://en.wikipedia.org/wiki/Accuracy_and_precision
  - http://en.wikipedia.org/wiki/F1_score
  - http://en.wikipedia.org/wiki/Precision_and_recall
- Exercise
  - Question 1: Why is the following statement both correct and useless? "My pregnancy test has a 95% accuracy".
  - Question 2: What is the precision of the pregnancy test with the following measures?
    - A pregnancy test correctly predicted pregnancy 80% of the time among pregnant women.
    - 10% of all the women were predicted pregnant but were actually not pregnant.
    - The accuracy of the test was 89%.

Pregnancy Test Exercise



Pregnancy Test Exercise

- Question 1: Accuracy does not address Recall or Precision. For instance, 95% Accuracy could mean 95% TN and 0% TP. Both Recall and Precision would be 0%
- Question 2: Use ClassificationAccuracy.R (homework) as a template to complete PregnancyExercise.R
- **PregnancyExercise.R**
- Problem Statement
  - I TP + FN + FP + TN = 1
  - II TP / (TP + FN) = Recall = 0.80
  - III (TP + TN)/(TP + FP + TN + FN) = 0.89
  - IV FP = 0.1
- Algebra on statements II and III
  - II FN = TP*0.20/0.80
  - III*I TN = 0.89 - TP
- Substitute FN, TN, and FP:
  - I,II,III,
  - TP*0.
  - TP =
- Results:
  - TP =
  - Precis

Accuracy Measures Exercise

Quiz 07a

- Confusion Matrix and Accuracy Measures
- https://catalyst.uw.edu/webq/survey/ernsthe/270452
- (Last question is like last question of homework review. Complete PregnancyExercise.R by using ClassificationAccuracy.R as an example)

NOSQL:
CAP Theorem

CAP Theorem

- Continue at 8:43 PM

CAP Theorem

Distributed system with Shared Data: Vasanti Bhat-Nayak and Grace Hopper need a package from R to do a naïve Bayes classification. If there were only one server that contained this package, then consistency would be easy. But, availability would be restricted. When multiple R users want to download a package, the server gets clogged. Therefore, the cran packages are replicated on multiple servers around the world. When a package needs to be updated, then the master node asks all servers to update simultaneously. So when Vasanti and Grace download a package from different servers they will get the same version of the Naive Bayes package.

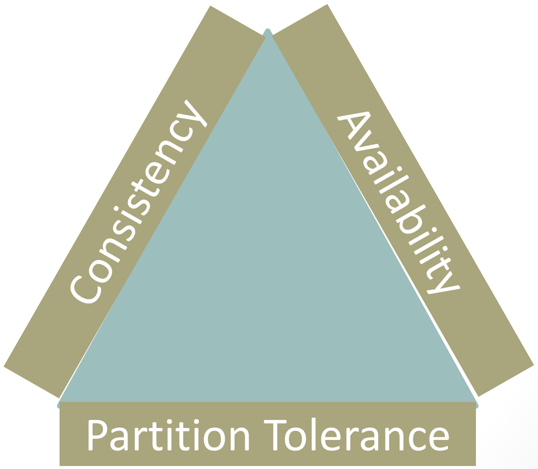
CAP Theorem

Distributed system with Shared Data: Vasanti Bhat-Nayak and Grace Hopper need a package from R to do a naïve Bayes classification. If there were only one server that contained this package, then consistency would be easy. But, availability would be restricted. When multiple R users want to download a package, the server gets clogged. Therefore, the cran packages are replicated on multiple servers around the world. When a package needs to be updated, then the master node asks all servers to update simultaneously. So when Vasanti and Grace download a package from different servers they will get the same version of the Naive Bayes package.

Partition of the Distributed System: But, what happens if on that day the Andorran server that Vasanti uses, can't be updated because of a communication error. The database has two choices: (1) It can wait until the Andorran server is fixed and then do the update. (2) Or, it updates all the other servers that allow the update. In the first case we forgo availability and nobody has access to the most recent Naive Bayes package. In the second case Vasanti and Grace will have different results because the packages are different.

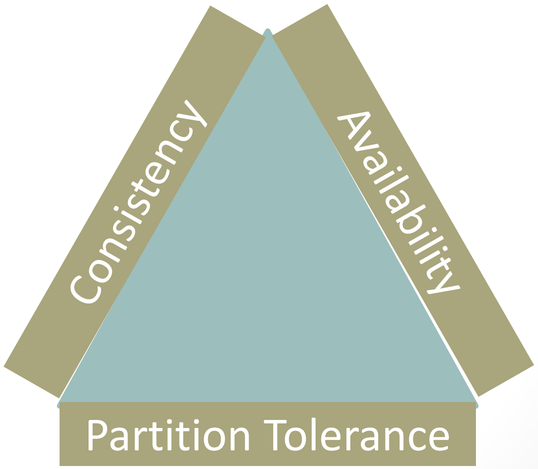
CAP Theorem

- CAP stands for:
  - **C**onsistency
  - **A**vailability
  - **P**artition Tolerance



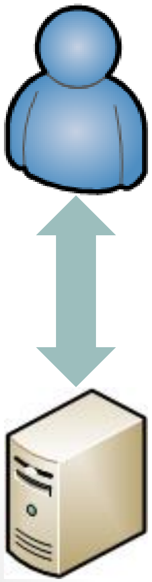
CAP Theorem

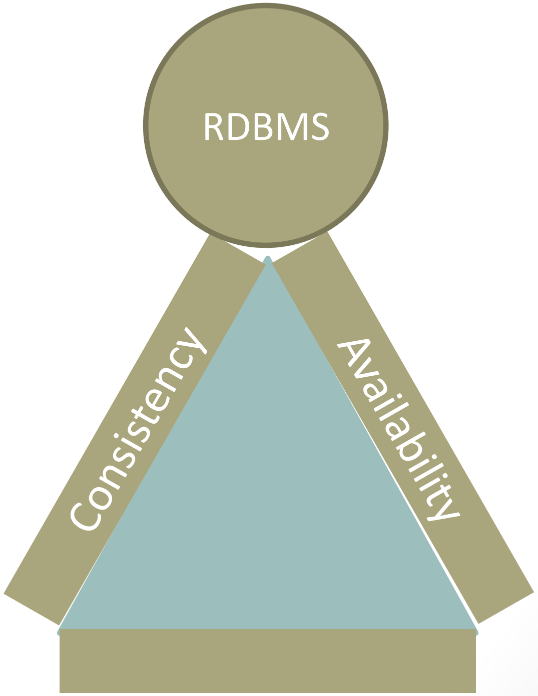
- CAP stands for:
  - **C**onsistency: All nodes see the same data at the same time
  - **A**vailability: Nodes are available for updates and reads
  - **P**artition Tolerance: Arbitrary message loss or partial failure does not bring down the system



CAP Theorem

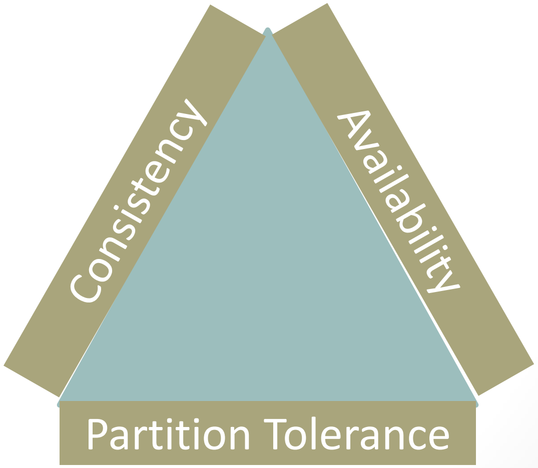
- Assume a single node with one set of data.
- This simple system resembles a typical RDBMS.
- Partition tolerance is irrelevant, because we only have one node.





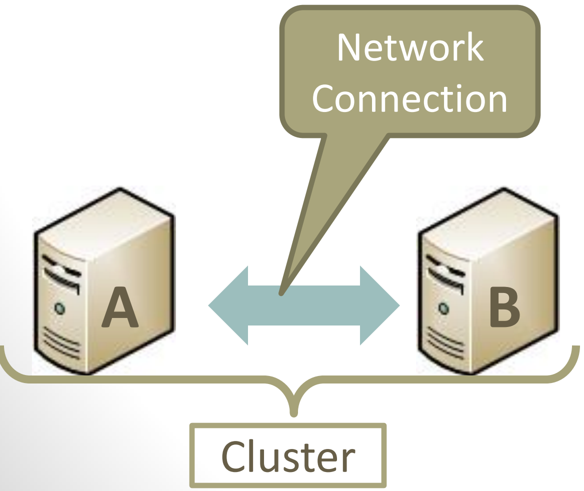
CAP Theorem

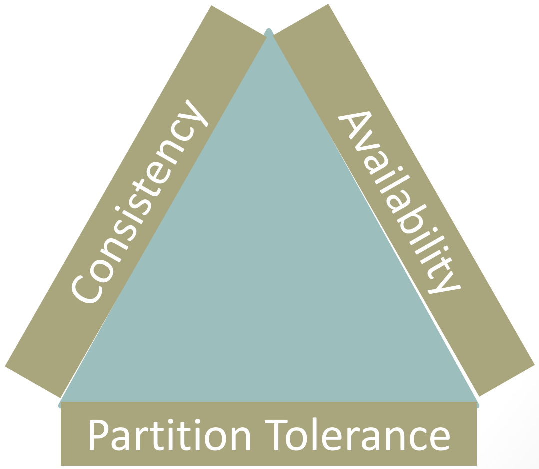
- The CAP theorem was formulated by Eric Brewer http://en.wikipedia/wiki/CAP_theorem
- Two formulations of the CAP theorem:
  - You can have at most two of the CAP properties for any shared data system.
  - During a network partition, a distributed system must choose either Consistency or Availability.



CAP Theorem

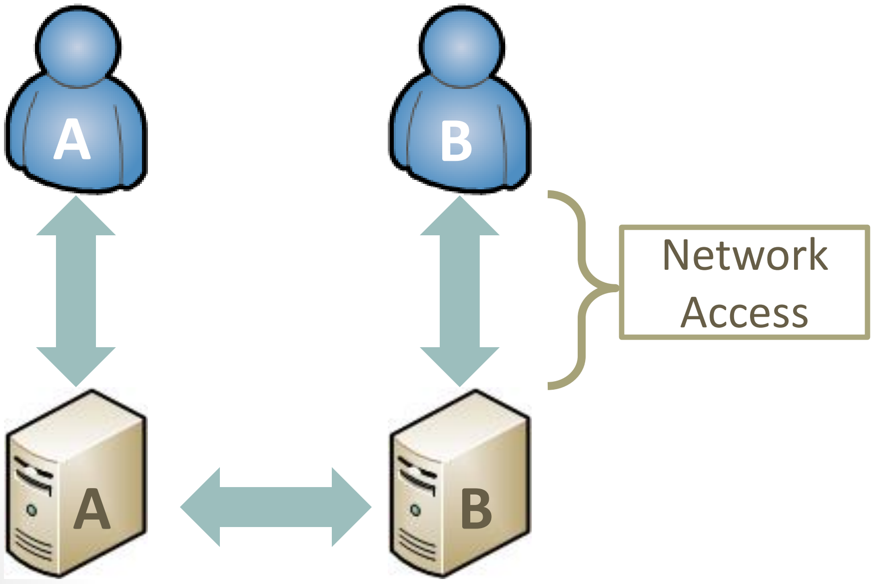
- Assume a cluster with shared and replicated data.
- The cluster consists of two connected nodes called A and B.

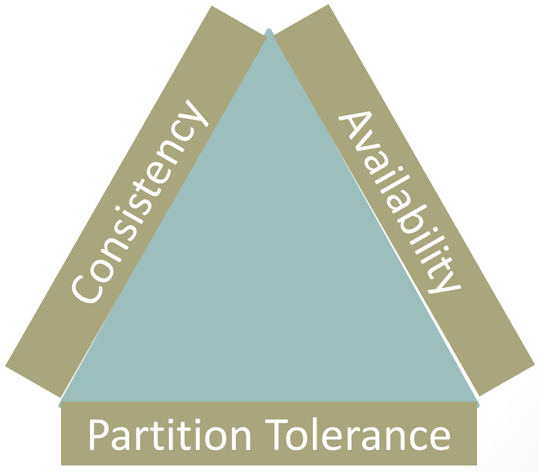




CAP Theorem

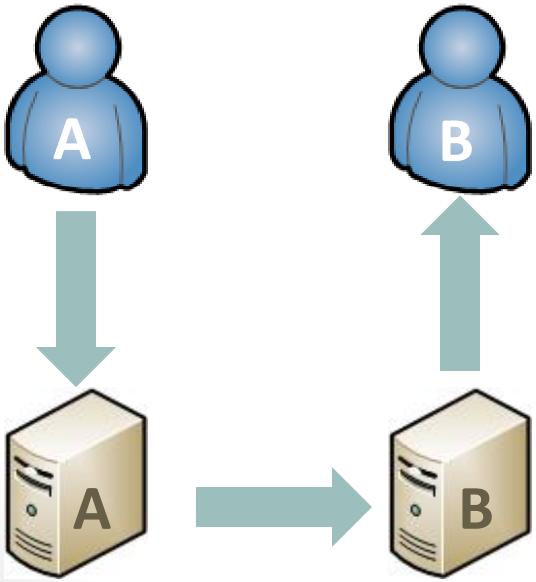
- Assume a cluster with shared and replicated data.
- The cluster consists of two connected nodes called A and B.
- The cluster is used by two users, called A and B. Each user has network access to a separate node

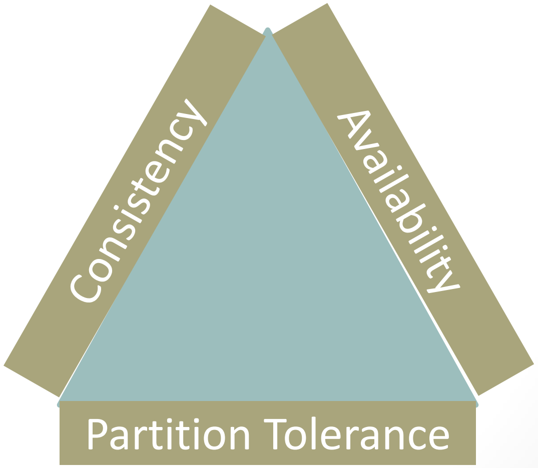




CAP Theorem

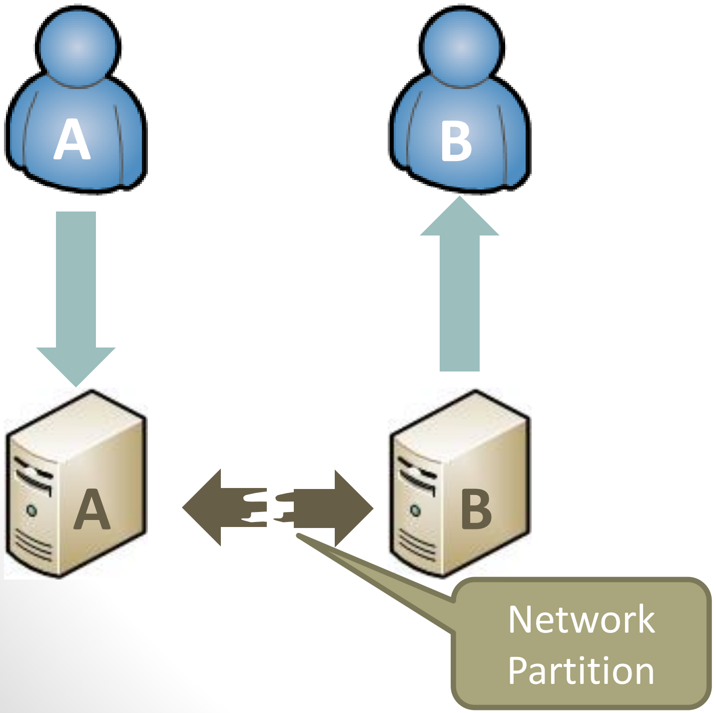
- Scenario 1: Network is available and Data are Consistent
  1. User A updates node A
  2. Update is communicated to node B
  3. User B reads the update from node B

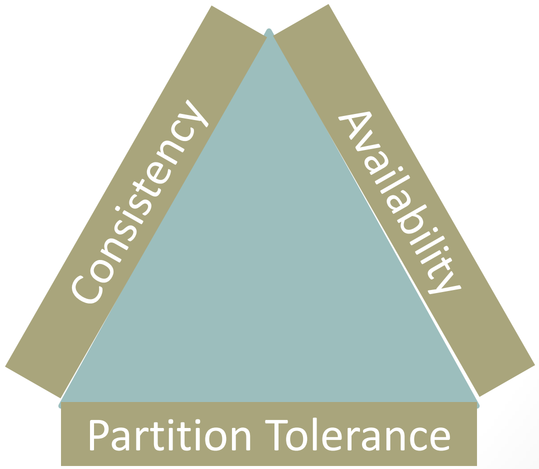




CAP Theorem

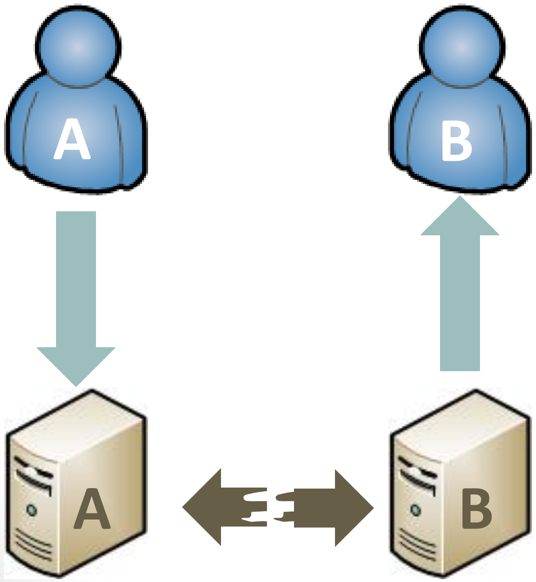
- Scenario 2: A network failure occurred.
  1. User A attempts to update node A
  2. Any Update cannot be communicated
  3. User B attempts to read the update

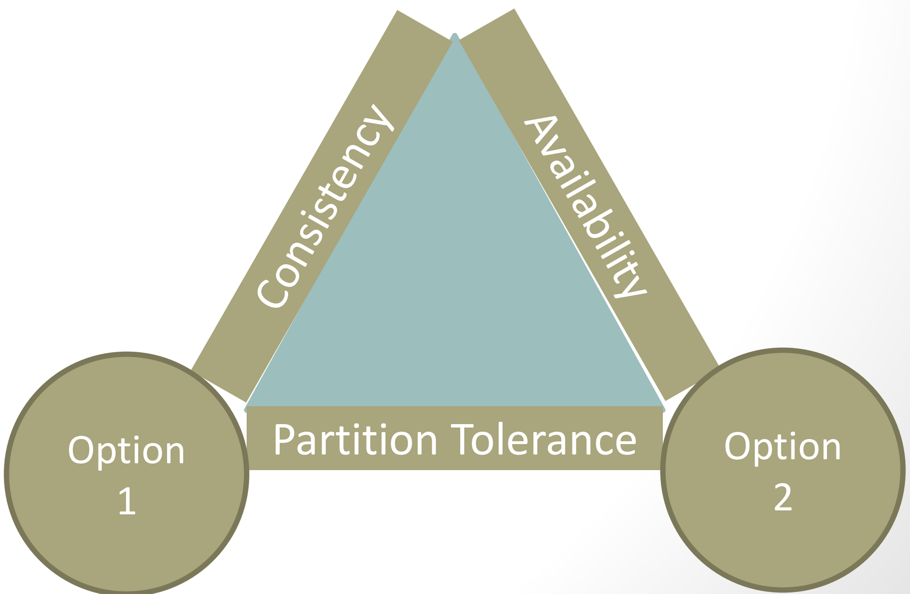




CAP Theorem

- Scenario 2: A network failure occurred. Two options:
  1. Make the database unavailable to avoid inconsistency
  2. Keep the database available and tolerate inconsistency





NOSQL:
CAP Theorem

Break

Relational Algebra

The Theory behind Relational Databases

Relational Algebra: What and Why

- Ted Codd introduced relational algebra to databases and created the relational model.

- Relational algebra provides a theoretical foundation for relational databases, and particularly for query languages like SQL.

- Why do you want a theoretical foundation?
  - If you want to optimize a query or a database
  - If you are thinking about using NOSQL, then you should be aware of the limitations and advantages of NOSQL data management. In other words, relational algebra assists in comparing SQL with NOSQL (**NO**T-SQL, **N**ot-**O**nly-SQL, K**NO**W-SQL, http://www.youtube.com/watch?v=sh1YACOK_bo)

New Terminology (1)

Term	Comments
Table	Part of a database
Relation	A table where rows are unique. Operand in Relational Algebra/Calculus
Tuple	single, double, triple, qudruple, quintuple, sextuple; Like a row in a table
Arity	unary, binary, ternary, quaternary
Closure	Operation on a type produces a value of that same type. Natural Numbers have closure under + and * (3 * 5 = 15) Natural Numbers do not have closure under – or /; 5 – 3 = -2

New Terminology (2)

Term	Comments
Procedural	Step-by-step solution to solving problem or achieving goal. I will drive to Bellevue, enter the class room and listen to the lecture. (Relational Algebra is procedural or imperative)
Declarative	Stating what one wants in non-ambiguous terms without describing how one is to achieve ones goal. Example: I want to know what was said in class last week. I don't care if you use the slide deck, your memory, or the recording to get me that information. (SQL is declarative)
Relational Algebra	The algebra that describes relations as operands and results
Relational Calculus	The calculus that uses relations as operands and results (SQL)

New Terminology (3)

Operation	Symbols	Comments
Selection	σ (sigma); $\sigma_{\varphi}(R)$;	SELECT * FROM <table name> WHERE Column1 = 1
Projection	π (pi); $\pi_{c1, c2, \dots, cn}(R)$	SELECT Column1, Column 2 FROM <table name>
Rename	P (rho)	
Union	∪	A∪B; A={1,2,3, 5}; B={0,2}; {1,2,3, 5}∪{0,2}={0,1,2,3,5}
Intersection	∩	A∩B; A={1,2,3, 5}; B={0,2}; {1,2,3, 5} ∩{0,2}= {2}
Difference	\, -,	B\A = B-A; {0,2} - {1,2,3,5} = {0}

New Terminology (4)

Operation	Symbols	Comments
Product	X	AXB A={1,2,3,5}; B={0,2}; {1,2,3, 5} X{0,2}= {{1,0}, {2,0}, {3,0}, {5,0}, {1,2}, {2,2}, {3,2}, {5,2}}
Join	$\bowtie_{\varphi}$	B$\bowtie_{\varphi}$A; φ: A > B; A={1,2,3,5}; B={0,2}; {1,2,3,5} $\bowtie_{\varphi}${0,2} = {{1,0},{2,0},{3,0},{3,2},{5,0},{5,2}}
Division	÷	A÷B = C; Project to show me the columns in A that are not in B; Select to show me the tuples in A that are a superset of the a tuple in B.

Quiz 07b

- https://catalyst.uw.edu/webq/survey/ernsthe/270453

Relational Algebra

Name	**Age**	**Home**
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra: Relation

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relation

Relational Algebra: Relation

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relation is like a table except that each row must be unique like in a set

Relation

Relational Algebra: Attribute

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Attribute

Relational Algebra: Attribute

Attribute:

Must be of the same data type.

Have a name

Name	**Age**	**Home**
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Attribute

Relational Algebra: Tuple

<u>Name</u>	<u>Age</u>	<u>Home</u>
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

tuple

Relational Algebra: Tuple

tuple from: sing**le**, doub**le**, tri**ple**, quadr**uple**, quin**tuple**
arity from: un**ary**, bin**ary**, tern**ary**

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

tuple with arity of 3

Relational Algebra: Operands and Simple Operations

- Operand
  - Relation (Table)
- Operations
  - UNION
  - INTERSECT
  - PROJECT
  - SELECT
  - PRODUCT
  - DIVISION

Relational Algebra: Union

Combine Relations

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent

Name	Age	Home
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra: Union

Combine Relations

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent

Name	Age	Home
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra Union:
R ∪ S

Relational Algebra: Union

Combine Relations

SQL Statement:
SELECT * FROM MyTableR UNION
SELECT * FROM MyTableS

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent

Name	Age	Home
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra Union:
R ∪ S

Relational Algebra: Union

Combine Relations

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent

Name	Age	Home
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

→

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra Union:
R ∪ S

Relational Algebra: Intersect

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Yamana	50	Own

Same Rows

Name	Age	Home
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra: Intersect

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Yamana	50	Own

Same Rows

Name	Age	Home
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra: Intersect

Name	**Age**	**Home**
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Yamana	50	Own

Same Rows

Name	**Age**	**Home**
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Relational Algebra Intersection:
$R \cap S$

Relational Algebra: Intersect

Name	**Age**	**Home**
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Yamana	50	Own

Same Rows

Name	**Age**	**Home**
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

SQL Statement:

```
SELECT * FROM MyTableR
INTERSECT
SELECT * FROM MyTableS
```

Relational Algebra Intersection:

$R \cap S$

Relational Algebra: Intersect

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Yamana	50	Own

Same Rows

Name	Age	Home
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

→

Name	Age	Home
Menchú	31	Rent
Yamana	50	Own

Relational Algebra Intersection:
$R \cap S$

Relational Algebra: Examples

- $R \cup S$
  - SELECT * FROM MyTableR UNION SELECT * FROM MyTableS

- SELECT * FROM MyTableR UNION SELECT * FROM MyTableS
  - $R \cup S$ or $S \cup R$

- $R \cap S$
  - SELECT * FROM MyTableR INTERSECT SELECT * FROM MyTableS

- SELECT * FROM MyTableR INTERSECT SELECT * FROM MyTableS
  - $R \cap S$ or $S \cap R$

- In General:
  - An operation with $\cup$ or $\cap$ produces a relation
  - $R \cup S = S \cup R$
  - $R \cap S = S \cap R$
  - $(R \cup S) \cap T = (R \cap T) \cup (S \cap T)$
  - $(R \cap S) \cup T = (R \cup T) \cap (S \cup T)$

Relational Algebra: Project

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Vertical partition

Relational Algebra: Project

Name	**Age**	**Home**
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Vertical partition

Relational Algebra Project:

$\pi_{c1,\ c2,\ \ldots,\ cn}(R)$

where

c1, c2, …, cn: Age, Home

R: MyTable

Relational Algebra: Project

SQL Statement:
SELECT Age, Home FROM MyTable

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Vertical partition

Relational Algebra Project:
$\pi_{c1,\ c2,\ \ldots,\ cn}(R)$
where
c1, c2, …, cn: Age, Home
R: MyTable

Relational Algebra: Project

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

→

Age	Home
5	None
21	Rent
31	Rent
42	Rent
50	Own

Relational Algebra Project:

$\pi_{c1,\ c2,\ \ldots,\ cn}(R)$

where

c1, c2, …, cn: Age, Home

R: MyTable

Relational Algebra: Project

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

→

Age	Home
5	None
21	Rent
31	Rent
42	Rent
50	Own

The result of a projection is a relation with 0 to n attributes where n is the number of attributes in the operand

Relational Algebra Project:

$\pi_{c1,\ c2,\ \ldots,\ cn}(R)$

where

c1, c2, …, cn: Age, Home

R: MyTable

Relational Algebra: Select

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Horizontal partition

Relational Algebra: Select

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Horizontal partition

Relational Algebra Select:

$\sigma_{\varphi}(R)$

where

φ: Home = "Rent"

R: MyTable

Relational Algebra: Select

SQL Statement:

SELECT * FROM MyTable WHERE Home = “Rent”

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

Horizontal partition

Relational Algebra Select:

$\sigma_{\varphi}(R)$

where

φ: Home = “Rent”

R: MyTable

Relational Algebra: Select

Name	Age	Home
Blackburn	5	None
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent
Yamana	50	Own

→

Name	Age	Home
Kobayashi	21	Rent
Menchú	31	Rent
Alvarez	42	Rent

The result of a selection is a relation with 0 to n tuples where n is the number of tuples in the operand

Relational Algebra Select:

$\sigma_{\varphi}(R)$

where

φ: Home = “Rent”

R: MyTable

Relational Algebra: Examples

- $\pi_{Age,Home}(R)$
  - SELECT Age, Home FROM MyTable
- $\sigma_{Home=\text{“Rent”}}(R)$
  - SELECT * FROM MyTable WHERE Home = “Rent”
- SELECT Age, Home FROM MyTable WHERE Home = “Rent”
  - $\pi_{Age,Home}(\sigma_{Home=\text{“Rent”}}(R))$ or $\sigma_{Home=\text{“Rent”}}(\pi_{Age,Home}(R))$
- In General:
  - An operation with σ produces a relation
  - An operation with π produces a relation
  - $\sigma_{\varphi 1}(\sigma_{\varphi 2}(R)) = \sigma_{\varphi 2}(\sigma_{\varphi 1}(R))$
  - $\pi_{[c1]}(\pi_{[c2]}(R)) = \pi_{[c2]}(\pi_{[c1]}(R))$
  - $\pi_{[c]}(\sigma_{\varphi}(R)) = \sigma_{\varphi}(\pi_{[c]}(R))$ (<u>**only if**</u> φ is not dependent on [c])

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

Name	**Age**
Blackburn	5
Kobayashi	21

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

SQL Statement:
SELECT * FROM TableR, TableS

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

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

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: 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 ÷ 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 ÷ 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	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
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

Assignment (1)

1. {a, b, c} is a relation that contains the tuples a, b, and c. In the following cases the tuples have arity of 1. Calculate the following:
   a. $(\{1, 2, 3\} \cup \{5, 7, 11\}) \cap \{2, 4, 6, 8, 10\}$
   b. $(\{1, 2, 3\} \cap \{2, 4, 6, 8, 10\}) \cup (\{5, 7, 11\} \cap \{2, 4, 6, 8, 10\})$
2. Use formal notation to write an algebraic example of the following SQL:
   a. SELECT Column1, Column3 FROM MyTable WHERE Column2 = Column3
   b. Reverse the order of projection and selection in your algebraic formulation. What happened?
3. $\pi_{c1,\ c2}(\sigma_{\varphi 1}(\sigma_{\varphi 2}(\pi_{c1,\ c2,\ c3,\ c5}(R))))$

   Where
   - φ1: C1 = C5;
   - φ2: C5 = "Test";
   - R: MyTable;

   a. Write a SQL statement that declares the intent of the algebraic notation
   b. Simplify the algebraic statement. Simplification means minimize the number of parentheses and terms.

Assignment (2)

4. SELECT * FROM T1 JOIN T2 ON T1.C1 = T2.C1
   a. Write out an equivalent in relational algebra using the join operator
   b. Write out an equivalent in relational algebra without using the join operator
5. $\pi_{S.C1,\ R.C2}(\sigma_{\varphi 1}(R) \bowtie_{\varphi 2} S)$
   where
   - $\varphi 1 = (R.C2 = \text{'A'})$
   - $\varphi 2 = (R.C1 = S.C2)$
   - Write out equivalent SQL and test this SQL using relations R and S that you create for this example. The relations R and S in RelationalAlgebraAndSQL.pdf and RelationalAlgebraAndSQL.sql don't quite work because their column types do not match for this assignment.
6. Submit answers to items 1 through 5 in a file by Saturday 11:00 PM. The SQL statements from 3a and 5 must be in a txt, doc, or sql file. I will need to copy and paste those statements.
7. If you did not complete Quiz 07b during class, then complete the quiz before Saturday 11:00 PM.

Introduction to Data Science