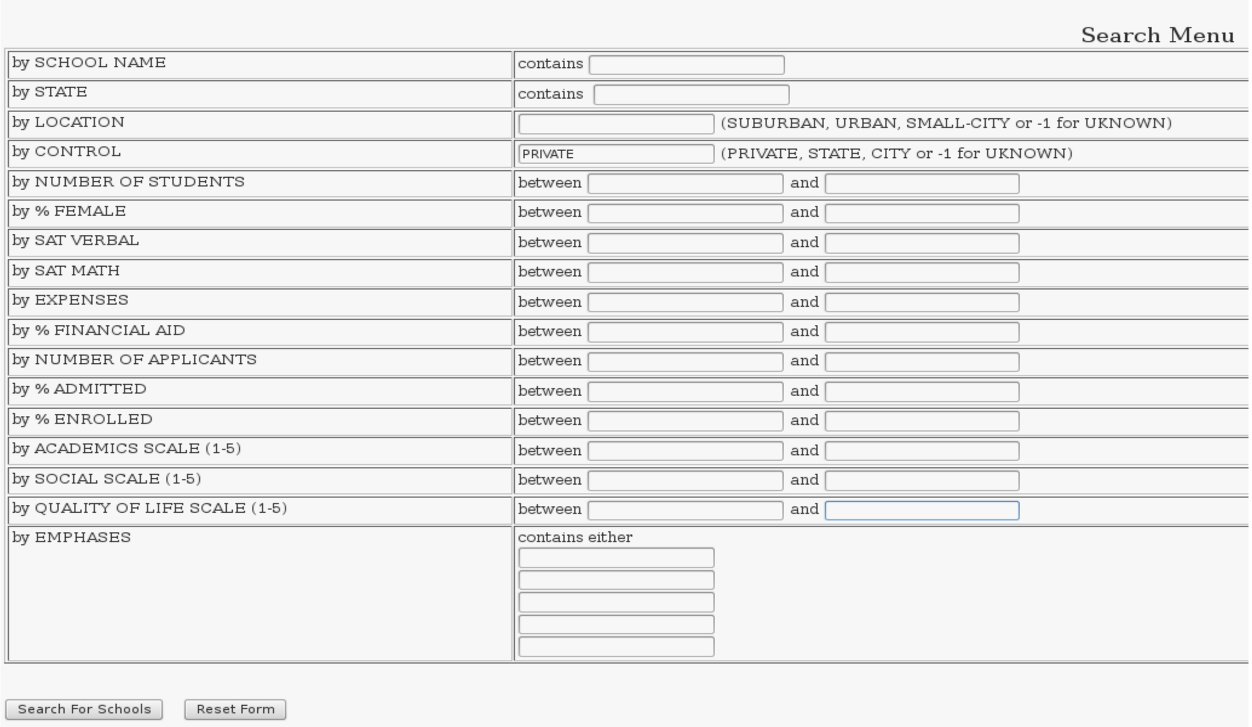


Search Menu

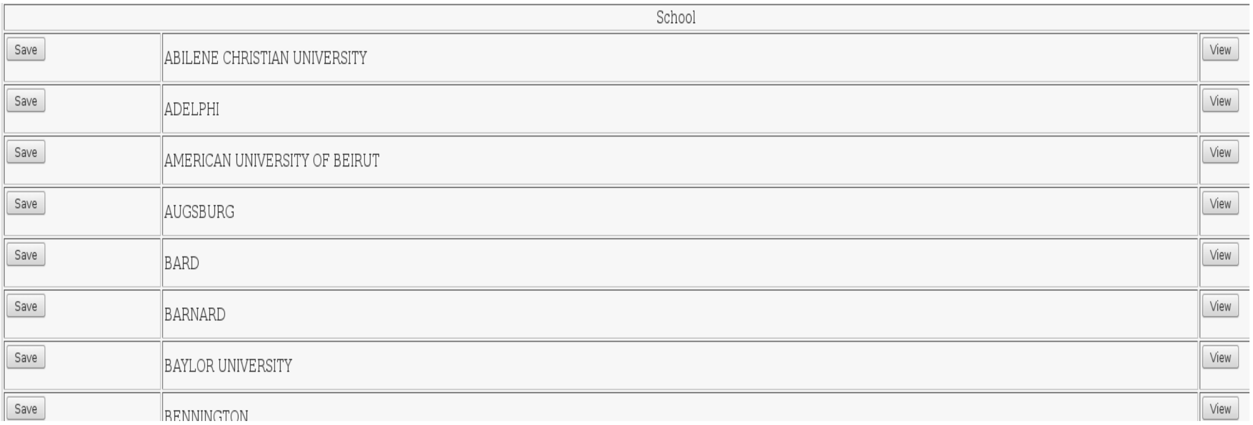
by SCHOOL NAME	contains
by STATE	contains
by LOCATION	(SUBURBAN, URBAN, SMALL-CITY or -1 for UKNOWN)
by CONTROL	PRIVATE (PRIVATE, STATE, CITY or -1 for UKNOWN)
by NUMBER OF STUDENTS	between and
by % FEMALE	between and
by SAT VERBAL	between and
by SAT MATH	between and
by EXPENSES	between and
by % FINANCIAL AID	between and
by NUMBER OF APPLICANTS	between and
by % ADMITTED	between and
by % ENROLLED	between and
by ACADEMICS SCALE (1-5)	between and
by SOCIAL SCALE (1-5)	between and
by QUALITY OF LIFE SCALE (1-5)	between and
by EMPHASES	contains either

The user will be utilizing an interface like the one above to search for schools. For our purposes, we will need to return ALL schools that match ALL search criteria. For example, if the user searches for

STATE: contains "ta"
CONTROL: contains "CITY"
EXPENSES: between 10000 and ??? (i.e., upper value not provided by user)
EMPHASES: contains either
"SCIENCE"
"MATH"

The system should return ALL schools which contain the string "ta" in their STATE field **and** the string "CITY" in their CONTROL field, **and** charge >= 10000 USD in expenses **and** have at least one area of emphasis containing either the string "SCIENCE" or "MATH".

The figure above displays the search results matching some user-provided search criteria. Clicking on the view button of one of the matches will display info on the selected school along with 5 recommended schools.



	School	
Save	ABILENE CHRISTIAN UNIVERSITY	View
Save	ADELPHI	View
Save	AMERICAN UNIVERSITY OF BEIRUT	View
Save	AUGSBURG	View
Save	BARD	View
Save	BARNARD	View
Save	BAYLOR UNIVERSITY	View
Save	BENNINGTON	View

The trick is to find the 5 schools MOST similar to the selected one. One way to do this is to think of every school as a vector and compute distance measures between the selected school and ALL other schools in the databases. For e.g., suppose we have a vector V (1, 1000, 200, “A”) – it really does not matter what the values mean – and our database contains the following 8 vectors:

	X1	X2	X3	X4
V1	0	1000	500	“B”
V2	1	1033	300	“B”
V3	2	11000	400	“A”
V4	1	1000	200	“F”
V5	1	1200	220	“F”
V6	0	1000	443	“C”
V7	0	1500	333	“A”
V8	5	1000	1200	“B”

To compute the distance between any two vectors V1 and V2 we can simply add up the ***absolute differences*** along all columns X1 thru X4:

dist(V1,V2) = $\sum_{i=1}^{4}|V1.Xi - V2.Xi|$ = |V1.X1−V2.X1| + |V1.X2−V2.X2| +
|V1.X3−V2.X3| + |V1.X4−V2.X4|

So distance dist(V, V1) = |1-0| + |1000-1000| + |200-500| + |“A” - “B”| = 1+0+300+?

This has two apparent problems

1- It is obvious that fields or columns with large ranges (like X2) will overshadow columns with small ranges (like X1). An easy fix would be to replace the $V1.Xi - V2.Xi$ in the formula with $\frac{V1.Xi-V2.Xi}{\max(Xi)-\min(Xi)}$ which will essentially map all individual differences to range [0,1]

2- How to deal with non-numeric columns such as X4? We can simply record a distance of 0 if the values are equal or 1 otherwise.

We these fixes, dist(V, V1) now becomes = |1-0|/|5-0| + |1000-1000|/|11000-1000| + |200-500|/|1200-200| + 1 = 0.2 + 0 + 0.3 + 1 = 1.50

Similarly, we end up with the following distances table:

dist(V,V1)	= 1.5
dist(V,V2)	= 1.1033
dist(V,V3)	= 2.4
dist(V,V4)	= 1
dist(V,V5)	= 1.04
dist(V,V6)	= 1.443
dist(V,V7)	= 1.383
dist(V,V8)	= 2.8

Based on our computations, the 5 most similar vectors to V are (and in this order): V4, V5, V2, V7 and V6. Of course, we will need to sort the vectors based on the distances in order to get the top 5.