Student Searching using Field / Value Searches

PowerSchool stores student information in fields. For example, the "first_name" field stores the students' first names. The "DOB" filed stores the students' date-of-birth. Search the various student fields based on the information you are looking for.

The following is an example of how to search student records.

Suppose you would like to find all of the students in your school whose first name is Jenny. Type the following into the search box:

first_name=jenny

PowerSchool displays the search results for all students named Jenny. This is a search command line; a line of text that is commanding PowerSchool to perform a search.

Every search command line is broken into three parts: a field name (such as first_name or grade_level or DOB); a comparator (such as =); and the search argument itself (in the example, jenny is the search argument).

The general format is:

[field name] [comparator] [search argument]

On the PowerSchool start page, select the View Field List link below the Search Students box to view a list of available field names.

The following table lists the comparators you can use:

Comparator	Meaning
=	equals
<	is less than
>	is greater than
<=	is less than or equal to
>=	is greater than or equal to
#	does not equal
in	is [field] present in the search argument?
contains	is the search argument contained in the [field]?
!contain	is the search argument not contained in the [field]?
@	wildcard

The comparator "in" is a powerful tool that allows you to verify if the value of the field matches any search argument in a list you provide. For example, to obtain a single list of all 9th, 10th, and 11th graders in your school, enter the following search command line:

```
grade_level in 9,10,11
```

PowerSchool displays all students whose grade level is contained in the list 9, 10, and 11. In this example, the field is "grade_level", the comparator is "in", and the search argument is "9,10,11". The search function looks at the specified field for every student and checks to see if that field matches anything in the search argument specified. When entering a search command line, separate a list of items in a search argument with commas.

The comparator "contains" is another powerful search tool. For example, you want to find all of the students in your school who live on Cherry Lane. Enter the following search command line:

```
mailing_street contains Cherry Lane
```

PowerSchool displays all students with Cherry Lane appearing in the mailing_street field. It does not matter if the mailing_street field is 194 Cherry Lane, or 24230 Cherry Lane Parkway; anything that contains Cherry Lane is considered a match.

The comparator "!contain" provides another search function. For example, you wanted to find all of the students in your school who do *not* live on Cherry Lane. Enter the following search command line:

```
mailing_street !contain Cherry Lane
```

PowerSchool displays all students with Cherry Lane not appearing anywhere in the mailing_street field. It does not matter if the mailing_street field is 194 Cherry Lane, or 24230 Cherry Lane Parkway; anything that contains Cherry Lane is not considered a match.

The wildcard "@" provides additional search flexibility. For example, you want to find all of the students whose first names started with jen. Enter the following search command line:

```
first_name = jen@
```

PowerSchool displays all students with the first name that begins with jen.