sentinel

Report Generator – Query Operators Explained
Report generator Query Option

To refine the detail produced on a report, the option exists to search and extract data within individual fields and report if required. The query option allows you to choose a question within the desired form (these questions are termed ‘dictionary items’) and add an operator and a value forcing the system to only produce the data if the condition is satisfied.

To create a query, the following procedure is used.

- Select the **dictionary item** that will be interrogated from the list.
- Select the **Operator** to be applied to the condition you wish to build e.g. ‘equal to’ or ‘less than’.
- Enter the **Value** to apply to the condition. The format depends on the column selected.
- Click the **Add** button to add this to the query details.

Any dictionary item can be utilised in a query statement even if it does not appear on the eventual report. The columns on the report are not affected, the query drives which records are displayed.

The query is then displayed in the Query Details area. If you wish to remove a query, click once on the entry within the Query Details box and click on **Remove**.

To build more complex queries, more conditions can be added using combinations of any dictionary items and any operators. Sentinel will combine the queries with an **AND** link however this can be changed to an **OR** by clicking once on the appropriate statement and clicking on the toggle **And / Or** button.

The use of brackets will be needed when building queries especially when using the **OR** link to group sets of statements together. If the desired results are not obtained then it is possible that brackets need adding or moving within your statements.

Once you have built your query then clicking **OK** accepts the constructed query statement(s) and returns you to the appropriate Report Generator screen from which the report can be produced.
Query Operators explained

Below is a list of available operators, an explanation of their use and how then can be used for the above purpose.

**Equal To ( = )** - This operator will extract data that is equal to the information entered in the Value field. For example, if the dictionary item is Date of Incident and the Value is 02/11/2005, the print will only report on records that have the Date of Incident equal to this date. **(Caution – the field value must match exactly to the data stored within the dictionary item to produce a successful result)**

<table>
<thead>
<tr>
<th>Dictionary Item</th>
<th>Operator</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incident Type</td>
<td>Equal to ( = )</td>
<td>Clinical Incident – Adverse Outcome</td>
</tr>
</tbody>
</table>

*This will report on only the records where the incident type is *Clinical Incident – Adverse Outcome* *

**Not Equal To ( NE )** – Using this operator will allow the system to report records that show any data other than the information entered in the Value field.

<table>
<thead>
<tr>
<th>Dictionary Item</th>
<th>Operator</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incident Type</td>
<td>Not Equal to (NE)</td>
<td>Clinical Incident – Adverse Outcome</td>
</tr>
</tbody>
</table>

*This will report on any records where the incident type is not equal to *Clinical Incident – Adverse Outcome* *

**Greater Than ( > )** - This operator will allow you to display on the report any data that has a value greater than the one entered in the Value field. If the field being queried contains text, then it is dealt with alphabetically.

*Example 1 –*

<table>
<thead>
<tr>
<th>Dictionary Item</th>
<th>Operator</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date of Incident</td>
<td>Greater Than ( &gt; )</td>
<td>01/04/2006</td>
</tr>
</tbody>
</table>

*This will report on any records where the Date of incident is 02/04/2006 or after.*

*Example 2 –*

<table>
<thead>
<tr>
<th>Dictionary Item</th>
<th>Operator</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk Score</td>
<td>Greater Than ( &gt; )</td>
<td>10</td>
</tr>
</tbody>
</table>

*This will report on any records where the Risk Score is 11 or higher.*
Greater Than Or Equal To (>=) – This is similar to the above, however the report will include any data that is equal to the information entered in the Value field.

Example 1 –
Dictionary item  Operator  Value
'Date of Incident'  Greater Than or Equal to (>=)  '01/04/2006'

This will report on any records where the Date of incident is 01/04/2006 or after, therefore including the start date.

Example 2 –
Dictionary item  Operator  Value
'Risk Score'  Greater Than or Equal to (>=)  '10'

This will report on any records where the Risk Score is 10 or higher.

Less Than (<) - This operator will allow you to report on any data that has a value less than the one entered in the Value field. If the field being queried contains text, then it is dealt with alphabetically.

Example 1 –
Dictionary item  Operator  Value
'Date of Incident'  Less Than (<)  '01/04/2006'

This will report on any records where the Date of incident is 31/03/2006 or before.

Example 2 –
Dictionary item  Operator  Value
'Risk Score'  Less Than (<)  '10'

This will report on any records where the Risk Score is 9 or less.

Less Than Or Equal To (<=) – This is similar to the above, however the report will include any data that is equal to the information entered in the Value field.

Example 1 –
Dictionary item  Operator  Value
'Date of Incident'  Less Than Or Equal To (<=)  '01/04/2006'

This will report on any records where the Date of incident is 01/04/2006 or before, therefore including the date in the value field.

Example 2 –
Dictionary item  Operator  Value
'Risk Score'  Less Than Or Equal To (<=)  '10'

This will report on any records where the Risk Score is 10 or less.
**Containing** – This is an extremely useful operator which can be used to investigate the selected field for a specific piece of information. The system will display any record that contains the data entered in the value field. For example, if the user wishes to see any incident that contains the word **Fire**, the operator **containing** and enter the word **Fire** in the **Value** field. This would extract the following data as it all contains the word Fire.

<table>
<thead>
<tr>
<th>Dictionary item</th>
<th>Operator</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Incident Type'</td>
<td>Containing</td>
<td>'Fire'</td>
</tr>
</tbody>
</table>

**Results**
- Fire incident - False alarm
- Fire incident - Accidental (Injury)
- Fire incident - Accidental (No injury)
- Fire incident - Wilful (Injury)
- Fire incident - Wilful (No injury)

**Example 2** –

<table>
<thead>
<tr>
<th>Dictionary item</th>
<th>Operator</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Cause of Injury(Injuries)'</td>
<td>Containing</td>
<td>'Fall'</td>
</tr>
</tbody>
</table>

This will report on any records where the Cause of Injury contains the word 'fall', i.e.

- Fall from Height
- Slip, trip or fall at the same level
- Other type of fall

**Not Containing** – This used in a similar way to the above, however the query will eliminate any record that contains the information entered in the value field and display every other record. So with the example above, the system will extract any record where the Cause of Injury does not contain the work ‘fall’.
**Starting With** – This can be used to extract records where the beginning of the sentence within the field is known. If you are extracting all data within a particular site and you know how the data starts.

Example - If your data is laid out like this

<table>
<thead>
<tr>
<th>Site Name</th>
<th>Building Name</th>
<th>Room Name</th>
</tr>
</thead>
</table>

Example -

<table>
<thead>
<tr>
<th>Dictionary item</th>
<th>Operator</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Location of Incident'</td>
<td>Starting With</td>
<td>'Site Name'</td>
</tr>
</tbody>
</table>

*This will report on any records where the Location of incident starts with 'Site Name'*. 

Results

- Site Name – Building 1 – Room 1
- Site Name – Building 1 – Room 2
- Site Name – Building 2 – Room 1
- Etc...

**Not Starting with** – Using this operator will mean that the records reported will be where the field in question does not start with the details entered in the value field.

**Ending With** – This operator follows the same principals as ‘Starting With’ but interrogates the end of the data string rather than the beginning.

**Not Ending With** – Again this follows similar rules to the ‘Not Starting With’ operator.