Hi Suhanisa,

Thank you for arising the issue.

The ScriptRunner version of that documentation is 6.20 which is outdated. Therefore, you may use the jira script below for executing the JQL to create a custom scripted field. You may change the clauseName, operator and functionName in the addFunctionCondition() to create your own query. Also, you can add filters as many as you like starting from the fourth parameter for this function.

| import com.atlassian.jira.bc.issue.search.SearchService import com.atlassian.jira.component.ComponentAccessor import com.atlassian.jira.jql.builder.JqlQueryBuilder import com.atlassian.jira.web.bean.PagerFilter import com.atlassian.query.operator.Operator  def searchService = ComponentAccessor.getComponent(SearchService) def loggedInUser = ComponentAccessor.jiraAuthenticationContext.loggedInUser  // Modify the name of your issue picker field here def issuePickerFieldName = 'Root Cause'  // This creates a query like: "issueFunction in issuePickerField('Root Cause', 'key = ${issue.key}')" // You could also use JqlQueryParser. def query = JqlQueryBuilder.newClauseBuilder().addFunctionCondition(  'issueFunction',  Operator.IN,  'issuePickerField',  issuePickerFieldName,  "key = ${issue.key}" ).buildQuery()  def searchResults = searchService.search(loggedInUser, query, PagerFilter.getUnlimitedFilter())  searchResults.results.each { documentIssue ->  log.debug(documentIssue.key)   // if you need a mutable issue you can do:  def issue = issueManager.getIssueObject(documentIssue.id)   // do something to the issue...  log.debug(issue.summary) } |
| --- |

I hope this helps, but please let me know if you need anything else or if you have any other questions.

Regards,

Lewis