STONEWARE

Name :   Yahya Cherkaoui El Moqadem
Date :   01.05.2023

Summary

GetIssues

Queries covered

- Get all outstanding issues (all products)
- Get all outstanding issues for a product (all versions)
- Get all outstanding issues for a product (single version)
- Get all outstanding issues within date range for a product (all versions)
- Get all outstanding issues within date range for a product (single version)
- Get all outstanding issues containing list of keywords (all products)
- Get all outstanding issues for a product containing list of keywords (all versions)
- Get all outstanding issues for a product containing list of keywords (single version)
- Get all outstanding issues within date range for a product containing list of keywords (all versions)
- Get all outstanding issues within date range for a product containing list of keywords (single version)
- Get all resolved issues (all products)
- Get all resolved issues for a product (all versions)
- Get all resolved issues for a product (single version)
- Get all resolved issues within date range for a product (all versions)
- Get all resolved issues within date range for a product (single version)
- Get all resolved issues containing list of keywords (all products)
- Get all resolved issues for a product containing list of keywords (all versions)
- Get all resolved issues for a product containing list of keywords (single version)
- Get all resolved issues within date range for a product containing list of keywords (all versions)
- Get all resolved issues within date range for a product containing list of keywords (single version)

Variables

The variables below can be defined or just ignored. If ignored, a null (or empty) value will be passed as a parameter and that specific filter will not be applied.

```
@Product int = null,
@Version int = null,
@Keywords nvarchar(MAX) = '',
@Status int = null,
@CreationDate datetime = null,
@ResolutionDate datetime = null
```

Query

```sql
SELECT Issues.ID,
       pv_products.Name,
       pv_versions.Version,
       pos_operatingsystems.OperatingSystem,
       Status.Status,
       Issues.Problem,
       Issues.Solution,
       Issues.CreationDate,
       Issues.ResolutionDate

FROM dbo.Issues
       JOIN ProductsVersions ON Issues.ProductVersionID = ProductsVersions.ID
       JOIN ProductsOperatingSystems ON Issues.ProductOperationSystemID =
       ProductsOperatingSystems.ID
       JOIN Products pv_products ON ProductsVersions.ProductID = pv_products.ID
       JOIN Versions pv_versions ON ProductsVersions.VersionID = pv_versions.ID
       JOIN Products pos_products ON ProductsOperatingSystems.ProductID =
       pos_products.ID
       JOIN OperatingSystems pos_operatingsystems ON ProductsOperatingSystems.ProductID
       = pos_operatingsystems.ID
       JOIN Status ON Issues.StatusID = Status.ID
       JOIN string_split(@Keywords, ' ') tKeyword ON Issues.Solution LIKE '%' +
       tKeyword.value + '%'
```

*Select columns from table Issues joining on the other necessary tables and the function
"string_split" with the list of keywords as a parameter.*

Conditions

```sql
WHERE
       pv_products.ID LIKE CASE
              WHEN @Product IS null THEN pv_products.ID
              WHEN @Product IS NOT null THEN @Product
              END
       AND pv_versions.ID LIKE CASE
              WHEN @Version IS null THEN pv_versions.ID
              WHEN @Version IS NOT null THEN @Version
              END
       AND UPPER(Issues.Solution) LIKE '%' + UPPER(tKeyword.value) + '%'
       AND Status.ID LIKE CASE
              WHEN @Status IS null THEN Status.ID
              WHEN @Status IS NOT null THEN @Status
              END
       AND (Issues.CreationDate >= CASE
              WHEN @CreationDate IS null THEN Issues.CreationDate
              WHEN @CreationDate IS NOT null THEN @CreationDate
              END
       OR Issues.CreationDate IS null AND @CreationDate IS null)
       AND (Issues.ResolutionDate <= CASE
              WHEN @ResolutionDate IS null THEN Issues.ResolutionDate
              WHEN @ResolutionDate IS NOT null THEN @ResolutionDate
              END
       OR Issues.ResolutionDate IS null AND @ResolutionDate IS null)
```

*Use "CASE WHEN" to check if a parameter has been passed for each variable. When variable is
null or empty, we simply ignore it by returning all the results.*

Group By

```
GROUP BY
    Issues.ID,
    pv_products.Name,
    pv_versions.Version,
    pos_operatingsystems.OperatingSystem,
    Status.Status,
    Issues.Problem,
    Issues.Solution,
    Issues.CreationDate,
    Issues.ResolutionDate;
```

Use "GROUP BY" on every column to remove duplicates created by the keywords filter.

```
GROUP BY
    Issues.ID,
    pv_products.Name,
    pv_versions.Version,
    pos_operatingsystems.OperatingSystem,
    Status.Status,
```