REFERENCE DOCUMENTATION

STONEWARE

Name : Yahya Cherkaoui El Moqadem
Date : 19.04.2023

Table des matières

GetIssuesForProduct

Queries covered

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

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,
	@Keywords nvarchar(MAX) = '',
	@Resolved bit = null,
	@CreationDate datetime = null,
	@ResolutionDate datetime = null
```

Query

```
SELECT Issues.ID,
	Products.Name,
	Versions.Version,
	OperatingSystems.Name,
	Status.Name,
	Issues.Problem,
	Issues.Solution,
	Issues.CreationDate,
	Issues.ResolutionDate

FROM dbo.Issues
	JOIN Versions ON Issues.VersionID = Versions.ID
	JOIN Products ON Versions.ProductID = Products.ID
	JOIN OperatingSystems ON Issues.OperatingSystemID = OperatingSystems.ID
	JOIN Status ON Issues.StatusID = Status.ID
	LEFT 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

```
WHERE
	Products.ID LIKE CASE
	WHEN @Product IS null THEN Products.ID
	WHEN @Product IS NOT null THEN @Product
	END
	AND UPPER(Issues.Solution) LIKE '%' + UPPER(tKeyword.value) + '%'
	AND Status.ID LIKE CASE
	WHEN @Resolved IS null THEN Status.ID
	WHEN @Resolved = 'True' AND Status.ID = 3 THEN Status.ID
	WHEN @Resolved = 'False' AND Status.ID != 3 THEN Status.ID
	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,
	Products.Name,
	Versions.Version,
	OperatingSystems.Name,
	Status.Name,
	Issues.Problem,
	Issues.Solution,
	Issues.CreationDate,
	Issues.ResolutionDate;
```

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

GetIssuesForVersion

Queries covered

- Get all outstanding issues (all products)
- Get all outstanding issues for a product (single version)
- 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 (single version)
- 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 (single version)
- 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 (single version)
- 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.

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

Query

```
SELECT Issues.ID,
	Products.Name,
	Versions.Version,
	OperatingSystems.Name,
	Status.Name,
	Issues.Problem,
	Issues.Solution,
	Issues.CreationDate,
	Issues.ResolutionDate

FROM dbo.Issues
	JOIN Versions ON Issues.VersionID = Versions.ID
	JOIN Products ON Versions.ProductID = Products.ID
	JOIN OperatingSystems ON Issues.OperatingSystemID = OperatingSystems.ID
	JOIN Status ON Issues.StatusID = Status.ID
	LEFT 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

```
WHERE
	Versions.ID LIKE CASE
	WHEN @Version IS null THEN Versions.ID
	WHEN @Version IS NOT null THEN @Version
	END
	AND UPPER(Issues.Solution) LIKE '%' + UPPER(tKeyword.value) + '%'
	AND Status.ID LIKE CASE
	WHEN @Resolved IS null THEN Status.ID
	WHEN @Resolved = 'True' AND Status.ID = 3 THEN Status.ID
	WHEN @Resolved = 'False' AND Status.ID != 3 THEN Status.ID
	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,
	Products.Name,
	Versions.Version,
	OperatingSystems.Name,
	Status.Name,
	Issues.Problem,
	Issues.Solution,
	Issues.CreationDate,
	Issues.ResolutionDate;
```

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