StoneWare

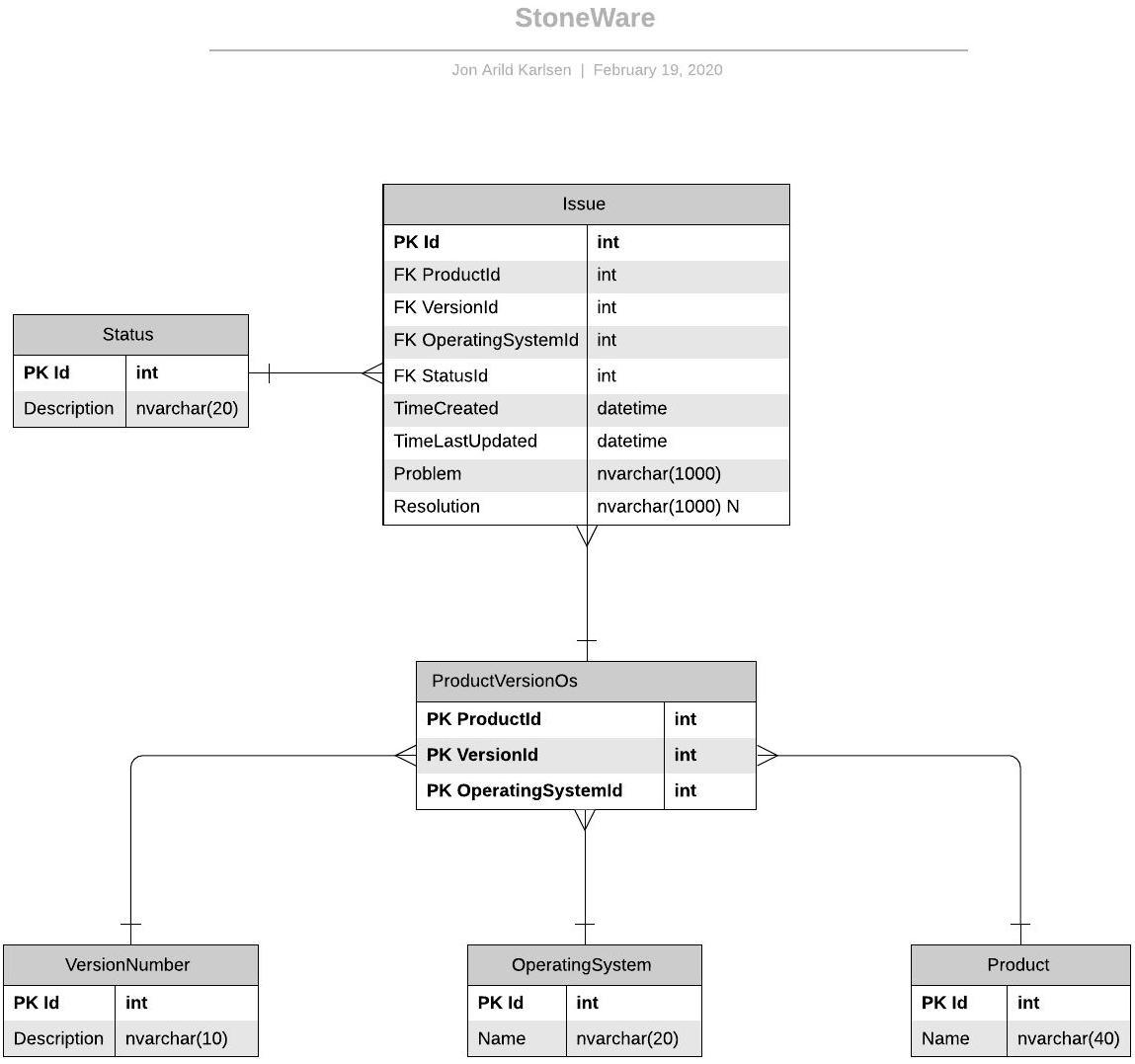
Jon Karlsen

February 19, 2020

Overview

This document contains the specifications for the StoneWare issue tracker database, as well as reference documentation for each stored procedure.

Entity Relationship Diagram



In order to reduce data redundancy and improve data integrity, the database is normalised to the third normal form (3NF). Fourth normal form, also called Boyce Codd Normal Form (BCNF), and

fifth normal form do exist but are rarely considered in practical design. Disregarding these rules may result in less than perfect database design, but should not affect functionality.

A brief overview of the relevant normal forms follows:

1NF

- Eliminate repeating groups in individual tables
- Create a separate table for each related set of data
- Identify each set of related data with a primary key

2NF

- Create separate tables for sets of values that apply to multiple records
- Relate these tables with a foreign key

3NF

- Eliminate fields which do not depend on the key

Stored Procedures

Stored procedures are executable subroutines stored within a database. Because they are precompiled, they offer an increase in performance (at the cost of memory overhead). They also improve security by allowing for granular access with security based on the stored procedure rather than the underlying tables, and by preventing SQL injection attacks through parameterisation of input.

Naming

Each stored procedure is named according to what it does, eg. `dbo.spIssuesOutstanding` retrieves all outstanding issues. `dbo` refers to the database owner schema, and the `sp` prefix indicates a stored procedure object (not to be confused with system stored procedures, which are indicated by the prefix `_sp`).

Usage

Using SQL, a stored procedure can be invoked with parameters thus:

```
EXECUTE dbo.IssuesOutstanding 'XSS, SSL, HDD'
```

In this case, querying all outstanding issues with the keywords XSS, SSL, and HDD (supplied as a string containing a comma-separated list of values).

In C# (using Entity Framework Core and extension methods[1]), the same could be called as follows:

```
var result = await _context
    .LoadStoredProc("spIssuesOutstandingKeywords")
    .WithSqlParam("@Keywords", "XSS, SSL, HDD")
    .ExecuteStoredProc<IssueDto>();
```

[1] https://gist.github.com/jakarlse88/682ddf3d6719d6527898131e557298e9

01 IssuesOutstanding

Description

This stored procedure returns all issues with status set to “outstanding”.

Parameters

This stored procedure takes no parameters.

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesOutstanding")
    .ExecuteStoredProc<IssueStoredProcedureResult>();
```

02 IssuesOutstandingSingleProdAllVer

Description

This stored procedure returns all issues with status set to “outstanding” for a single product of all versions.

Parameters

Name	SQL Datatype	C# Datatype
ProductId	int	int

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesOutstandingSingleProdAllVer")
    .WithSqlParam("@ProductId", 1)
    .ExecuteStoredProc<IssueStoredProcedureResult>();
```

03 IssuesOutstandingSingleProdSingleVer

Description

This stored procedure returns all issues with status set to “outstanding” for a single product of a single version.

Parameters

Name	SQL Datatype	C# Datatype
ProductId	int	int
VersionNumberId	int	int
OperatingSystemId	int	int

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesOutstandingSingleProdSingleVer")
    .WithSqlParam("@ProductId", 1)
    .WithSqlParam("@VersionNumberId", 1)
    .WithSqlParam("@OperatingSystemId", 1)
    .ExecuteStoredProc<IssueStoredProcedureResult>();
```

04 IssuesOutstandingSingleProdAllVerDateRange

Description

This stored procedure returns all issues with status set to “outstanding” for a single product of all versions within a date range.

Parameters

Name	SQL Datatype	C# Datatype
ProductId	int	int
StartDate	datetime	DateTime
EndDate	datetime	DateTime

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesOutstandingSingleProdAllVerDateRange")
        .WithSqlParam("@ProductId", 1)
        .WithSqlParam("@StartDate", new DateTime(2020, 02, 09))
        .WithSqlParam("@EndDate", new DateTime(2020, 02, 11))
        .ExecuteStoredProc<IssueStoredProcedureResult>();
```

05

IssuesOutstandingSingleProdSingleVerDateRange

Description

This stored procedure returns all issues with status set to “outstanding” for a single product of a single version within a date range.

Parameters

Name	SQL Datatype	C# Datatype
ProductId	int	int
VersionNumberId	int	int
OperatingSystemId	int	int
StartDate	datetime	DateTime
EndDate	datetime	DateTime

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesOutstandingSingleProdSingleVerDateRange")
        .WithSqlParam("@ProductId", 1)
        .WithSqlParam("@VersionNumberId", 1)
        .WithSqlParam("@OperatingSystemId", 1)
        .WithSqlParam("@StartDate", new DateTime(2020, 02, 09))
        .WithSqlParam("@EndDate", new DateTime(2020, 02, 11))
        .ExecuteStoredProc<IssueStoredProcedureResult>();
```

06 IssuesOutstandingKeywords

Description

This stored procedure returns all issues with status set to “outstanding” for all products of all versions containing a list of keywords (passed to the procedure as a comma-separated string of values).

Parameters

Name	SQL Datatype	C# Datatype
Keywords	nvarchar(500)	string

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesOutstandingKeywords")
    .WithSqlParam("@Keywords", "SSL, XSS, HDD")
    .ExecuteStoredProc<IssueStoredProcedureResult>();
```

07 IssuesOutstandingSingleProdAllVerKeywords

Description

This stored procedure returns all issues with status set to “outstanding” for a product of all versions containing a list of keywords (passed to the procedure as a string of comma-separated values).

Parameters

Name	SQL Datatype	C# Datatype
ProductId	int	int
Keywords	nvarchar(500)	string

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesOutstandingSingleProdAllVerKeywords")
        .WithSqlParam("@ProductId", 1)
        .WithSqlParam("@Keywords", "SSL, XSS, HDD")
        .ExecuteStoredProc<IssueStoredProcedureResult>();
```

08

IssuesOutstandingSingleProdSingleVerKeywords

Description

This stored procedure returns all issues with status set to “outstanding” for a product of a single version containing a list of keywords (passed to the procedure as a string of comma-separated values).

Parameters

Name	SQL Datatype	C# Datatype
ProductId	int	int
VersionNumberId	int	int
OperatingSystemId	int	int
Keywords	nvarchar(500)	string

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesOutstandingSingleProdSingleVerKeywords")
        .WithSqlParam("@ProductId", 1)
        .WithSqlParam("@VersionNumberId", 1)
        .WithSqlParam("@OperatingSystemId", 1)
        .WithSqlParam("@Keywords", "SSL, XSS, HDD")
        .ExecuteStoredProc<IssueStoredProcedureResult>();
```

09

IssuesOutstandingSingleProdAllVerDateRangeKeywords

Description

This stored procedure returns all issues with status set to “outstanding” for a product of all versions within a date range containing a list of keywords (passed to the procedure as a string of comma-separated values).

Parameters

Name	SQL Datatype	C# Datatype
ProductId	int	int
StartDate	datetime	DateTime
EndDate	datetime	DateTime
Keywords	nvarchar(500)	string

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesOutstandingSingleProdAllVerDateRangeKeywords")
         .WithSqlParam("@ProductId", 2)
         .WithSqlParam("@StartDate", new DateTime(2020, 02, 06))
         .WithSqlParam("@EndDate", new DateTime(2020, 02, 07))
         .WithSqlParam("@Keywords", "SSL, XSS, HDD")
         .ExecuteStoredProc<IssueStoredProcedureResult>();
```

10

IssuesOutstandingSingleProdSingleVerDateRangeKeywords

Description

This stored procedure returns all issues with status set to “outstanding” for a product of a single version within a date range containing a list of keywords (passed to the procedure as a string of comma-separated values).

Parameters

Name	SQL Datatype	C# Datatype
ProductId	int	int
VersionNumberId	int	int
OperatingSystemid	int	in
StartDate	datetime	DateTime
EndDate	datetime	DateTime
Keywords	nvarchar(500)	string

Example Call

```
var result = await _context
.LoadStoredProc("spIssuesOutstandingSingleProdSingleVerDateRangeKeywords")
    .WithSqlParam("@ProductId", 1)
    .WithSqlParam("@VersionNumberId", 1)
    .WithSqlParam("@OperatingSystemId", 1)
    .WithSqlParam("@StartDate", new DateTime(2020, 02, 09))
    .WithSqlParam("@EndDate", new DateTime(2020, 02, 11))
    .WithSqlParam("@Keywords", "USB")
    .ExecuteStoredProc<IssueStoredProcedureResult>();
```

11 IssuesResolved

Description

This stored procedure returns all issues with status set to “resolved”.

Parameters

This stored procedure takes no parameters.

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesResolved")
    .ExecuteStoredProc<IssueStoredProcedureResult>();
```

12 IssuesResolvedSingleProdAllVer

Description

This stored procedure returns all issues with status set to “resolved” for a single product of all versions.

Parameters

Name	SQL Datatype	C# Datatype
ProductId	int	int

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesResolvedSingleProdAllVer")
    .WithSqlParam("@ProductId", 1)
    .ExecuteStoredProc<IssueStoredProcedureResult>();
```

13 IssuesResolvedSingleProdSingleVer

Description

This stored procedure returns all issues with status set to “resolved” for a single product of a single version.

Parameters

Name	SQL Datatype	C# Datatype
ProductId	int	int
VersionNumberId	int	int
OperatingSystemId	int	int

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesResolvedSingleProdSingleVer")
    .WithSqlParam("@ProductId", 1)
    .WithSqlParam("@VersionNumberId", 2)
    .WithSqlParam("@OperatingSystemId", 1)
    .ExecuteStoredProc<IssueStoredProcedureResult>();
```

14 IssuesResolvedSingleProdAllVerDateRange

Description

This stored procedure returns all issues with status set to “resolved” for a single product of all versions within a date range.

Parameters

Name	SQL Datatype	C# Datatype
ProductId	int	int
StartDate	datetime	DateTime
EndDate	datetime	DateTime

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesResolvedSingleProdAllVerDateRange")
        .WithSqlParam("@ProductId", 1)
        .WithSqlParam("@StartDate", new DateTime(2020, 02, 09))
        .WithSqlParam("@EndDate", new DateTime(2020, 02, 11))
        .ExecuteStoredProc<IssueStoredProcedureResult>();
```

15 IssuesResolvedSingleProdSingleVerDateRange

Description

This stored procedure returns all issues with status set to “resolved” for a single product of a single version within a date range.

Parameters

Name	SQL Datatype	C# Datatype
ProductId	int	int
VersionNumberId	int	int
OperatingSystemId	int	int
StartDate	datetime	DateTime
EndDate	datetime	DateTime

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesResolvedSingleProdSingleVerDateRange")
        .WithSqlParam("@ProductId", 1)
        .WithSqlParam("@VersionNumberId", 3)
        .WithSqlParam("@OperatingSystemId", 2)
        .WithSqlParam("@StartDate", new DateTime(2020, 02, 09))
        .WithSqlParam("@EndDate", new DateTime(2020, 02, 12))
        .ExecuteStoredProc<IssueStoredProcedureResult>();
```

16 IssuesResolvedKeywords

Description

This stored procedure returns all issues with status set to “resolved” and containing a list of keywords (passed to the procedure as a string of comma-separated values).

Parameters

Name	SQL Datatype	C# Datatype
Keywords	nvarchar(500)	string

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesResolvedKeywords")
    .WithSqlParam("@Keywords", "ssl")
    .ExecuteStoredProc<IssueStoredProcedureResult>();
```

17 IssuesResolvedSingleProdAllVerKeywords

Description

This stored procedure returns all issues with status set to “resolved” for a single product of all versions and containing a list of keywords (passed to the procedure as a string of comma-separated values).

Parameters

Name	SQL Datatype	C# Datatype
ProductId	int	int
Keywords	nvarchar(500)	string

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesResolvedSingleProdAllVerKeywords")
    .WithSqlParam("@ProductId", 1)
    .WithSqlParam("@Keywords", "SSL, XSS, HDD")
    .ExecuteStoredProc<IssueStoredProcedureResult>();
```

18 IssuesResolvedSingleProdSingleVerKeywords

Description

This stored procedure returns all issues with status set to “resolved” for a single product of a single version and containing a list of keywords (passed to the procedure as a string of comma-separated values).

Parameters

Name	SQL Datatype	C# Datatype
ProductId	int	int
VersionNumberId	int	int
OperatingSystemId	int	int
Keywords	nvarchar(500)	string

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesResolvedSingleProdSingleVerKeywords")
        .WithSqlParam("@ProductId", 1)
        .WithSqlParam("@VersionNumberId", 1)
        .WithSqlParam("@OperatingSystemId", 1)
        .WithSqlParam("@Keywords", "SSL, XSS, HDD")
        .ExecuteStoredProc<IssueStoredProcedureResult>();
```

19 IssuesResolvedSingleProdAllVerDateRangeKeywords

Description

This stored procedure returns all issues with status set to “resolved” for a single product of all versions within a date range and containing a list of keywords (passed to the procedure as a string of comma-separated values).

Parameters

Name	SQL Datatype	C# Datatype
ProductId	int	int
StartDate	datetime	DateTime
EndDate	datetime	DateTime
Keywords	nvarchar(500)	string

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesResolvedSingleProdAllVerDateRangeKeywords")
        .WithSqlParam("@ProductId", 4)
        .WithSqlParam("@StartDate", new DateTime(2020, 02, 06))
        .WithSqlParam("@EndDate", new DateTime(2020, 02, 09))
        .WithSqlParam("@Keywords", "SSL, XSS, HDD")
        .ExecuteStoredProc<IssueStoredProcedureResult>();
```

20 IssuesResolvedSingleProdSingleVerDateRangeKeywords

Description

This stored procedure returns all issues with status set to “resolved” for a single product of a single version within a date range and containing a list of keywords (passed to the procedure as a string of comma-separated values).

Parameters

Name	SQL Datatype	C# Datatype
ProductId	int	int
VersionNumberId	int	int
OperatingSystemid	int	int
StartDate	datetime	DateTime
EndDate	datetime	DateTime
Keywords	nvarchar(500)	string

Example Call

```
var result = await _context
    .LoadStoredProc("spIssuesResolvedSingleProdSingleVerDateRangeKeywords")
        .WithSqlParam("@ProductId", 4)
        .WithSqlParam("@VersionNumberId", 2)
        .WithSqlParam("@OperatingSystemId", 3)
        .WithSqlParam("@StartDate", new DateTime(2020, 02, 06))
        .WithSqlParam("@EndDate", new DateTime(2020, 02, 12))
        .WithSqlParam("@Keywords", "SSL, XSS, HDD")
        .ExecuteStoredProc<IssueStoredProcedureResult>();
```