Weed Traits Database (WTDB) Design

This document covers the database design and refers to the *Weed Traits Database (WTDB) Requirements* document (www.agrsci.dk/herbtol/ra45/database/requirements.pdf) and covers the database design. How to access the database e.g. the design of the user interface etc. will be covered in a later document.

Abstract

The Weed Traits Database (WTDB) is a relational relational database designed in a generic way to facilitate easy implementation of additional weed traits and parameters.

1. Goals

The design enables these usage scenarios to be supported:

- Carry out statistical analyses to explore the relationships between weed traits.
- Identify functional groups and typical species of each group.
- Identify indicator traits, i.e. traits that are good indicators to which functional group a species belongs.
- Access WTDB from a simulation models to look up values for traits and parameter.
- Standardise experimental design by rational selection of weed species for experimentation.
- Identify gaps in knowledge about weed traits and model parameters to aid rational planning of research activities.
- Inspire to new scientific perspectives on weed ecology.
- Serve as a basis for scientific publications.
- Used by researchers outside the ENDURE RA4.5 Working Group.

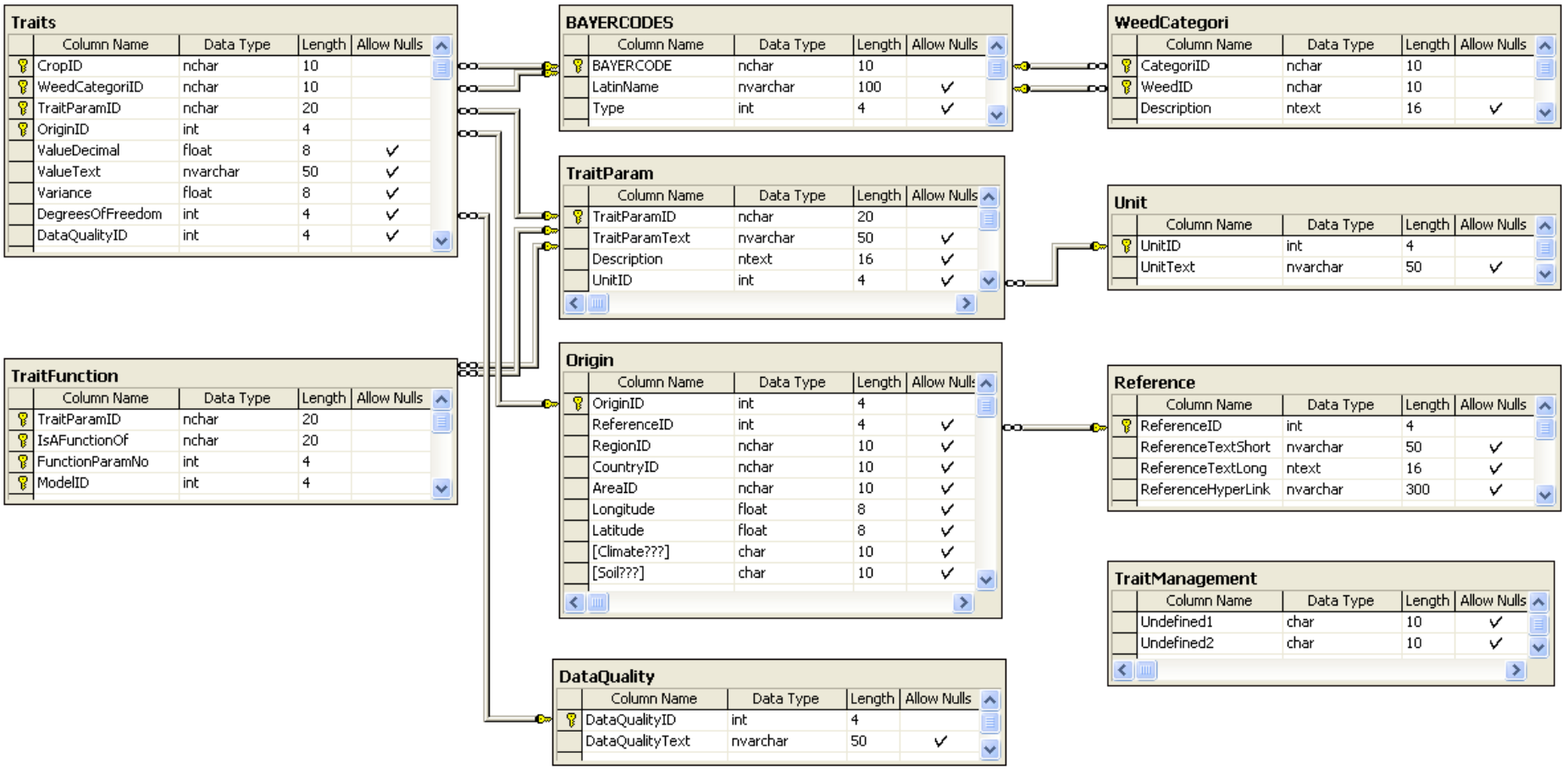
The following requirements are also taken into account:

- WTDB must include traits to describe both annual and perennial weed species.
- WTDB must be implemented in standard, widely-used database software.
- WTDB should not contain the raw data but statistical summaries thereof.
- WTDB must reside on the Internet, possibly on an AU server, or maybe INRA.
- WTDB may include how management procedures affect traits.

2. Tables and relationships

Tables and relationships are presented in figure 1.

Weed Traits Database (WTDB)



Traits

Column Name	Data Type	Length	Allow Nulls
CropID	nchar	10	
WeedCategoriID	nchar	10	
TraitParamID	nchar	20	
OriginID	int	4	
ValueDecimal	float	8	✓
ValueText	nvarchar	50	✓
Variance	float	8	✓
DegreesOfFreedom	int	4	✓
DataQualityID	int	4	✓

BAYERCODES

Column Name	Data Type	Length	Allow Nulls
BAYERCODE	nchar	10	
LatinName	nvarchar	100	✓
Type	int	4	✓

WeedCategori

Column Name	Data Type	Length	Allow Nulls
CategoriID	nchar	10	
WeedID	nchar	10	
Description	ntext	16	✓

TraitParam

Column Name	Data Type	Length	Allow Nulls
TraitParamID	nchar	20	
TraitParamText	nvarchar	50	✓
Description	ntext	16	✓
UnitID	int	4	✓

Unit

Column Name	Data Type	Length	Allow Nulls
UnitID	int	4	
UnitText	nvarchar	50	✓

TraitFunction

Column Name	Data Type	Length	Allow Nulls
TraitParamID	nchar	20	
IsAFunctionOf	nchar	20	
FunctionParamNo	int	4	
ModelID	int	4	

Origin

Column Name	Data Type	Length	Allow Nulls
OriginID	int	4	
ReferenceID	int	4	✓
RegionID	nchar	10	✓
CountryID	nchar	10	✓
AreaID	nchar	10	✓
Longitude	float	8	✓
Latitude	float	8	✓
[Climate???]	char	10	✓
[Soil???]	char	10	✓

Reference

Column Name	Data Type	Length	Allow Nulls
ReferenceID	int	4	
ReferenceTextShort	nvarchar	50	✓
ReferenceTextLong	ntext	16	✓
ReferenceHyperLink	nvarchar	300	✓

TraitManagement

Column Name	Data Type	Length	Allow Nulls
Undefined1	char	10	✓
Undefined2	char	10	✓

DataQuality

Column Name	Data Type	Length	Allow Nulls
DataQualityID	int	4	
DataQualityText	nvarchar	50	✓

Table: Traits

This table holds trait data and model parameters. Basically, one record in this table corresponds to a single trait or parameter value for a given crop and weed species.

**CropID* uniquely identifies a crop by an ID (up to 10 characters). Five characters BAYER-CODES are universal codes well suited for this purpose (Table: BAYERCODES). In this way, traits or parameters can be different for different crops. If a trait or parameter is common to all crops a special CropID designating all crops can be used.

**WeedCategoriID* uniquely identifies a weed or a weed functional group by an ID (up to 10 characters). Five characters BAYER-CODES are well suited for this purpose. DB-administrators are allowed to create additional codes (e.g. 10 characters) for functional weed species if a specific weed species does not serve the purpose (Table: BAYERCODES).

**TraitParamID* uniquely identifies a trait or a parameter by an ID (up to 20 characters). DB-administrators are allowed to create as many ID's as needed (Table: TraitParam). In this way, the database is generally open to future expansion.

**OriginID* uniquely identifies the source or the origin of a trait or parameter value by an ID (integer). In this way, traits or parameters can be different for different sources.

ValueDecimal holds the value of a trait or parameter as a floating point. Integer values, therefore also will be represented as floating points.

ValueText (alternatively or complimentary) can hold a value of a trait or parameter in a textual representation if a decimal value does not make sense. (ValueText possibly might not be needed and could be considered left out of the DB-design)

Variance is a floating point value denoting the variance of ValueDecimal based on a prior statistical analysis.

DegreesOfFreedom is an integer value denoting the degrees of freedom in the prior statistical analysis.

DataQualityID is an integer value denoting the quality of Valuedecimal. (foreign key, table: DataQuality)

Table: BAYERCODES

This table holds codes for weeds, crops and functional groups. Weeds and crops are represented by genuine five character BAYER-CODES while functional groups are represented by made-up codes in ten characters.

**BAYERCODE* uniquely identifies a weed, a crop or a weed functional group by an ID (up to 10 characters). Five characters BAYER-CODES are well suited for this purpose. DB-administrators are allowed to create additional codes (e.g. 10 characters) for functional weed species if a specific weed species does not serve the purpose.

LatinName is a full-length name of the weed species, crop or functional group (up to 100 characters). If possible, a Latin name is used.

Type is an integer denoting whether BAYERCODE is a crop, a weed or a functional group.

Table: TraitParam

This table contains definitions of traits and model parameters

**TraitParamID* uniquely identifies a trait or a parameter by an ID (up to 20 characters). DB-administrators are allowed to create as many ID's as needed. In this way, the database is generally open to future expansion.

TraitParamText is a text string containing the full name of the trait or the parameter

Description is a text string describing the trait or parameter in words

UnitID assigns a unit to a trait or parameter (integer) (foreign key, table: Unit)

Table: Origin

This table defines the origin of traits or parameters according to geography and the producer of data.

**OriginID* uniquely identifies the source or the origin of a trait or parameter value by an ID (integer).

ReferenceID uniquely identifies the producer of the data (foreign key, table: Reference).

RegionID identifies the region of the data (e.g. Southern Europe)

CountryID identifies the country of the data (e.g. Italy).

AreaID identifies the area in the country of the data (e.g. Northern Italy)

Longitude identifies the geographic longitude coordinate

Latitude identifies the geographic latitude coordinate

Climate ??? conditions

Soil ??? conditions

Table: DataQuality

This table defines classes to give trait and parameter values a score.

**DataQualityID* uniquely identifies a quality class of data values for traits and parameters

DataQualityText describes the quality class in words

Table: WeedCategori
This table is used to categorize weeds in functional groups

**CategoriID* identifies a weed or functional group (BAYERCODE etc.)

**WeedID* identifies a weed belonging to a specific CategoriID (BAYERCODE etc)

Description describes the relation between CategoriID and WeedID in words (unlimited text)

Table: Unit
This table defines units for traits or parameters

**UnitID* uniquely defines a Unit (integer)

UnitText defines the unit in text (50 characters)

Table: Reference
This table defines a reference to the documentation for trait and parameter values

**ReferenceID* uniquely defines a reference (integer)

ReferenceTextShort is a short description of the reference (50 characters)

ReferenceTextLong is a long description of the reference (unlimited text)

ReferenceHyperLink is a hyperlink to the reference (300 characters)

Table: TraitFunction
This table defines how traits relates to other traits or parameters (design yet not complete)

**TraitParamID* identifies a trait or parameter (20 characters)

**IsAFunctionOf* identifies a trait or parameter that TraitParamID depends on (20 characters)

**FunctionParamNo* identifies IsAFunctionOf as a number in a list of parameters in the model below (integer)

**ModelID* identifies method no. x to model the trait or parameter (integer). The model is defined elsewhere.

Table: TraitManagement
A framework for describing the influence of management should be designed at a later point.
