
**Institute of Mathematics and Computer Science
European Bioinformatics Institute**

**SIMS
Configuration Guide**

Version 2.10

Table of Contents

1 Database 3
 Level tables 4
 Vocabularies 5
 Meta-data tables 5
 Example of new configuration 5
2 Hibernate Mappings 7
3 Meta-data 9
4 Java classes 9
5 Level names in application 12
6 Application forms for record editing/inserting - JSPs 13

1 Database

SIMS DB is based on three-level architecture: Bottom level (*a3_samples*) → Middle level (*a2_visits*) → Top level (*a1_persons*). In the following text these tables are referenced as “level tables”. There is a meta-data table associated with each level table: Top meta-data table (*a1_person_meta_data*), Middle meta-data table (*a2_visit_meta_data*), Bottom meta-data table (*a3_sample_meta_data*). In the following text these tables are referenced as “meta-data tables”.

See Figure 1 Default SIMS DB configuration.

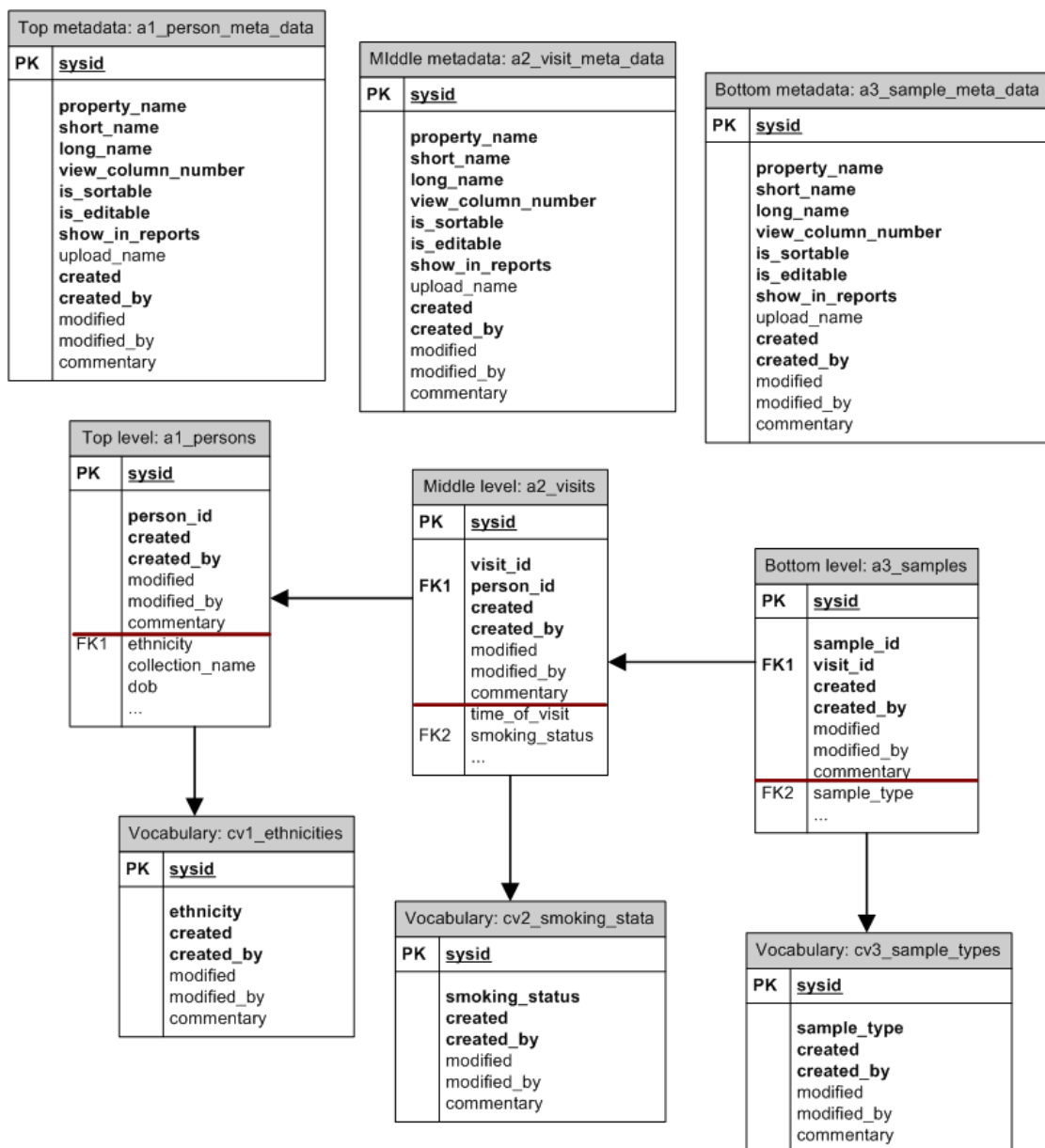


Figure 1 Default SIMS DB configuration

All changes in SIMS configuration start at database. It is possible to change level table name and attributes (to delete, to add new, to rename attributes).

Level tables

There are following mandatory attributes for each of the three mentioned level tables:

Attribute	Type	Description
sysid	bigint	Internal record identifier. Not null.
a<level><name>_id	character varying(400)	External record identifier. Not null.
<previous_level_name>	character varying(400)	If not Top level. Reference to the previous level. Not null.
created_by	bigint	SIMS user who created the record. Not null.
modified_by	bigint	SIMS user who last modified the record
created	timestamp without time zone	Record creation date and time. Not null.
modified	timestamp without time zone	Last record modification date and time.
commentary	text	Comment in free text format

Table 1 Level table

Other level table attributes are optional and can be changed if needed. The following types of optional attributes are supported in SIMS:

SIMS notation	Title	Description	Maximum number ¹
IT	Integer	Integer is a <i>bigint</i> datatype in PostgreSQL	19
ST	String	String is <i>character varying (400)</i> datatype in PostgreSQL	22
DT	Date	Date is “ <i>timestamp without time zone</i> ” datatype in PostgreSQL	4
FT	Float	Float is <i>numeric(20,5)</i> datatype in PostgreSQL	26
VT	Vocabulary	Vocabulary is not the real type of column, but “label” for the case when table's column references to vocabulary (small table in DB with particular structure). This type will be presented as list in SIMS forms with values from appropriate vocabulary table.	29

Table 2 Types of attributes

¹ This number means that in each level table can be used not more than 19 integer attributes, 22 string attributes etc. If more attributes of particular type are needed appropriate Java classes have to be changed (see Section “Java classes”).

Vocabularies

1. In the default configuration there are a lot of vocabulary tables all of them are named **cv**<level>_<name> (example: “*cv3_sample_types*”). Level indicates that this table is referenced from main table of appropriate level (example: “*cv3_sample_transport_conditions*” is referenced from table “*a3_samples*”).
2. In vocabularies that are named **cv0**_<name> are listed values for common promptings in SIMS forms.
3. Vocabularies with names **cv**<level>**m**_<name> indicate that this table is referenced from other vocabulary table (example: “*cv3_sample_types*” contains references to table “*cv3m_sample_types_sample_subtypes*”).

N.B. Described here naming principles of tables in SIMS database are not required condition for system functioning. User can rename tables following other rules.

Described types of optional attributes are associated with columns of level tables using hibernate mappings and meta-data tables.

Meta-data tables

Structure of meta-data tables can't be changed for normal SIMS functioning. However, context of meta-data tables has to be changed together with changes in level tables.

Example of new configuration

Let assumed that in new SIMS configuration Middle level table is renamed – *a2_characteristics*, attribute *time_of_visit* is renamed into *measurement_time* and number of attributes are added: *string_characteristic* is **ST** attribute (see Table 2 Types of attributes), *integer_characteristic* is **IT** attribute, *float_characteristic* is **FT** attribute and *list_characteristic* is **VT** attribute. Since new VT attribute has been added new vocabulary table is created (*cv2_list_values*) and new foreign key is created for Middle level table *a2_characteristics*.

See Figure 2 Example of new SIMS DB configuration.

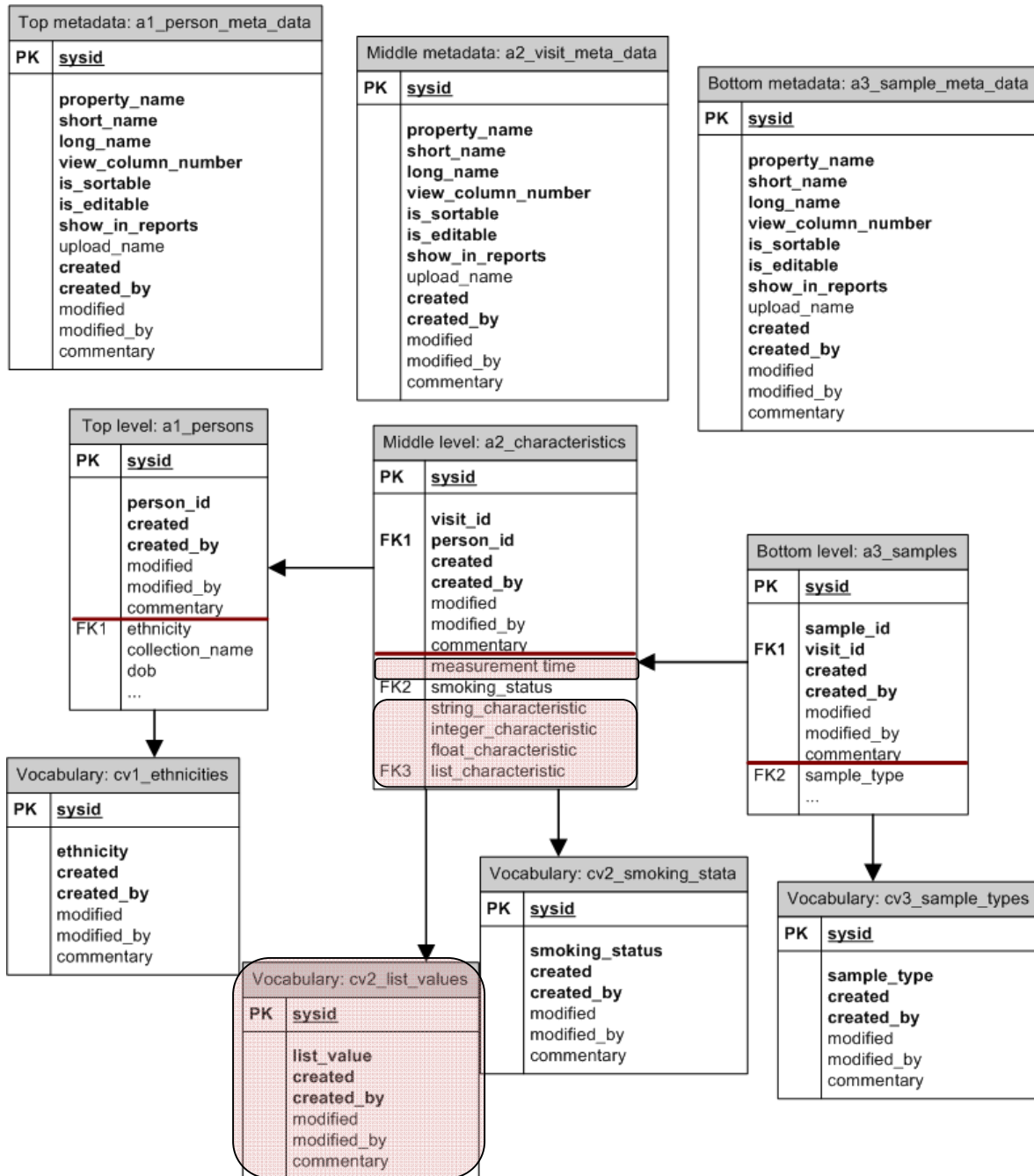


Figure 2 Example of new SIMS DB configuration

After changes in DB context of meta-data tables and hibernate mappings have to be changed in appropriate way. See details in the following sections.

2 Hibernate Mappings

Hibernate mapping has to be changed in appropriate way together with changes in structure of level tables. Besides, new Hibernate mapping files have to be created together with new vocabulary table.

1. Top level appropriate mapping file is: `../sims/hibernate/Top.hbm.xml`
2. Middle level appropriate mapping file is: `../sims/hibernate/Middle.hbm.xml`
3. Bottom level appropriate mapping file is: `../sims/hibernate/Bottom.hbm.xml`
4. Vocabulary's tables: `../sims/hibernate/VT<number>.hbm.xml`

The changes can be divided into following parts:

1. Table name. If level or vocabulary table name has been changed then in the mapping file string `<class name="Top/Middle/Bottom" table="a1/2/3_old name">` has to be changed: `<class name="Top/Middle/Bottom" table="a1/2/3_new name">`

In example of new DB configuration (Figure 2) Middle level table name `a2_visits` has been changed to `a2_characteristics`. In Hibernate mapping file `../sims/hibernate/Middle.hbm.xml` the same change has to be done:

```
...  
<class name="Middle" table="a2_characteristics">  
...
```

2. Attributes.

Attributes of level tables are mapped to appropriate types (ST, IT, FT, DT or VT) in Hibernate mapping files.

In example (Figure 2) attribute `time_of_visit` of table `a2_characteristic` has been changed to `measurement_time`.

In Hibernate mapping file `../sims/hibernate/Middle.hbm.xml` the same change has to be done:

```
...  
<property name="dt01" column="measurement_time"/>  
...
```

Here `dt01` means that this attribute has type DT and is first attribute (01) with DT type in Middle level table.

Mappings have to be added into Hibernate mapping file `../sims/hibernate/Middle.hbm.xml` for all new attributes (changes are marked with red color):

```
<?xml version="1.0"?>  
<!DOCTYPE hibernate-mapping PUBLIC  
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"  
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">  
  
<hibernate-mapping  
    package="sims.hibernate">  
    <class name="Middle" table="a2_characteristics">  
        <id name="id" column="sysid">  
            <generator class="native"/>
```



```

</id>
<property name="visibleName" column="visit_id" not-null="true"/>
<many-to-one name="parent" column="person" not-null="true"/>
<many-to-one name="creator" column="created_by" not-null="true"/>
<many-to-one name="modifier" column="modified_by"/>
<property name="createDate" column="created" not-null="true"/>
<property name="modifDate" column="modified"/>
<property name="comment" column="commentary"/>
<!-- configuration of integer types -->
<property name="it01" column="age_first_myocardial_infarction"/>
...
<property name="it13" column="integer_characteristic"/>
<!-- configuration of float types -->

<property name="ft01" column="weight"/>
...
<property name="ft27" column="float_characteristic"/>
<!-- configuration of string types -->
<property name="st02" column="glucose_timing"/>
...
<property name="st23" column="string_characteristic"/>
<!-- configuration of timestamp types -->
<property name="dt01" column="measurement_time"/>
<!-- configuration of cv types -->
<many-to-one name="vt03" column="alcohol_status"/>
...
<many-to-one name="vt30" column="list_characteristic"/>
<!-- the configurable part ends here -->
</class>
</hibernate-mapping>

```

Changes that have been made in SIMS DB and Hibernate mapping files require also changes in java classes if the maximum number of attributes with appropriate type has been exceeded (see Table 2 Types of attributes”). There are two such cases in example of new DB configuration: **ft27** (maximum is 26) and **vt30** (maximum is 29). See in Section “Java classes”.

2. New vocabulary table.

New vocabulary table `cv2_list_values` is created in example (Figure 2). This vocabulary is mapped as **vt30** in `./.../sims/hibernate/Middle.hbm.xml` hibernate mapping file. In such a case new hibernate mapping file `./.../sims/hibernate/VT30hbm.xml` is needed:

```

<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="sims.hibernate">
    <class name="VT30" table="cv2_list_values">
        <id name="id" column="sysid">
            <generator class="native"/>

```



```

        </id>
        <property name="visibleName" column="list_value" not-null="true"/>
        <many-to-one name="creator" column="created_by" not-null="true"/>
        <many-to-one name="modifier" column="modified_by"/>
        <property name="createDate" column="created" not-null="true"/>
        <property name="modifDate" column="modified"/>
        <property name="comment" column="commentary"/>
    </class>
</hibernate-mapping>

```

3 Meta-data

Meta-data can be changed from SIMS interface (Admin Tables) or by using DB tables: *a1_person_meta_data*, *a2_visit_meta_data*, *a3_sample_meta_data* in default configuration.

The structure of meta-data tables have to remain as it is for normal SIMS functioning. However, values have to correspond to desirable SIMS configuration.

In example (Figure 2) attribute *time_of_visit* of table *a2_characteristic* has been changed to *measurement_time* and number of new attributes has been added. In such a case context of meta-data table for Middle level has to be changed (new records are colored with red and bold, changed parts are colored red):

Property name	Short name	Long name	...	Upload name
...				
dt01	Measurement time	Measurement time		Time of measurement
...				
it13	Integer Value	Integer Characteristic		Integer Characteristic
ft27	Float Value	Float Characteristic		Float Characteristic
st23	String Value	String Characteristic		String Characteristic
vt30	List Value	List		List

Table 3 Context of meta-data table

Values of attributes **short name**, **long name** and **upload name** will be used in SIMS forms.

4 Java classes

Java classes have to be changed when there is more than maximum number of attributes of some particular type in new configuration (see Table 2 Types of attributes”).

In such case following classes have to be changed:

1. *sims/hibernate/Top.java* (if first level table is changed: *a1_persons* in default configuration),
2. *sims/hibernate/Middle.java* (if second level table is changed: *a2_vists* in default configuration),
3. *sims/hibernate/Bottom.java* (if third level table is changed: *a3_samples* in default configuration),
4. *sims/hibernate/VocabulName.java*.

In example (Figure 2) float type is exceeded default maximum **ft27** (maximum is 26) and vocabulary type is exceeded default maximum **vt30** (maximum is 29).

1. VocabulName.java

Java class *sims/hibernate/VocabulName.java* has to be changed (changes/new code is emphasized with red) and recompiled:

```
package sims.hibernate;
...
//----- data fields -----
private static final int vocabularyFirst = 20;
private static final int vocabularyCount = 30;

private static final int stringFirst = 60;
private static final int stringCount = 22;

private static final int integerFirst = 90;
private static final int integerCount = 19;

private static final int floatFirst = 110;
private static final int floatCount = 27;

private static final int dateFirst = 150;
private static final int dateCount = 4;

private static final int totalCount = 161; //+1!!!
...
//used by admintable
private static final String[][] vocabularyName = {{,
```

```
    {"Sample timings", "Sample timing"},  
    ...  
    {"Concentration units", "Concentration unit"},  
    {"Sample types", "Sample type"},  
    {"List values", "List values"} // 30  
};  
...
```

2. Middle.java

Also java class *sims/hibernate/Middle.java* has to be changed (changes/new code is colored with red) and recompiled:

```
package sims.hibernate;  
...  
// the number of parameters should be consistent with configuration in "vocalname.java"  
// for this configuration: st=22, it=9, ft=27, dt=3, vt=30  
...  
//----- Float parameters  
    private Double ft01;  
    public Double getFt01() {  
        return ft01;  
    }  
    public void setFt01(Double val) {  
        ft01 = val;  
    }  
    ...  
    private Double ft27;  
    public Double getFt27() {  
        return ft27;  
    }  
    public void setFt27(Double val) {  
        ft27 = val;  
    }  
    ...  
//----- vocabularies  
    private VT01 vt01;
```

```
public VT01 getVt01() {  
    return vt01;  
}  
public void setVt01(VT01 val) {  
    vt01 = val;  
}  
...  
private VT30 vt30;  
public VT30 getVt30() {  
    return vt30;  
}  
public void setVt30(VT29 val) {  
    vt30= val;  
}  
...
```

3. VT30.java

New class *sims/hibernate/VT30.java* has to be created and compiled for new vocabulary:

```
package sims.hibernate;  
public final class VT30 extends Persistent {  
}
```

5 Level names in application

To change default level names in SIMS application class *sims/hibernate/VocabulName.java* has to be modified:

```
...  
private static final String[][] mainName = {  
  
    {"Persons", "Person", "Person."},  
  
    {"Characteristics", "Characteristic", "Characteristic."},  
  
    {"Samples", "Sample", "Sample."}  
  
};  
...
```


6 Application forms for record editing/inserting - JSPs

There are three JSPs that have to be changed to complete new SIMS configuration:

1. *AddTopB.jspf* (if first level table is changed: *a1_persons* in default configuration),
2. *AddMiddleB.jspf* (if second level table is changed: *a2_vists* in default configuration),
3. *AddBottomB.jspf* (if third level table is changed: *a3_samples* in default configuration).

In example (Figure 2) Middle level table was changed, so *AddMiddleB.jspf* code has to be modified – new fields has to be added:

```

...
<!--Time and date of measurement (yyyy/mm/dd):--%>
<tr>
<td><%=sims.changeMark("dt01")%></td>
<td class="dialh"><%=sims.getLongName("dt01")%></td>
<td><%=sims.getYear("dt01", true)%><b>-</b><%=sims.getMonth("dt01", true)%><b>-</b><%=sims.getDay("dt01", true)%></td>
</tr>
...
<!--New Float characteristic--%>
<tr>
<td><%=sims.changeMark("ft27")%></td>
<td class="dialh"><%=sims.getLongName("ft27")%></td>
<td><%=sims.getInput("ft27", 4) %></td>
</tr>
...
<!--New Vocabulary – Listed Characteristic--%>
<tr>
<td><%=sims.changeMark("vt30")%></td>
<td class="dialh"><%=sims.getLongName("vt30")%></td>
<td><%=sims.getPrompterVt(VT30.class, "vt30", false) %></td>
</tr>
...
<!--New String Characteristic--%>
<tr>
<td><%=sims.changeMark("st23")%></td>
<td class="dialh"><%=sims.getLongName("st28")%></td>
<td><%=sims.getInput("st23", 10) %></td>
</tr>
...
<!--New Integer Characteristic--%>
<tr>
<td><%=sims.changeMark("it13")%></td>
<td class="dialh"><%=sims.getLongName("it13")%></td>
<td><%=sims.getCheckBox("it13") %></td>
</tr>
...
<!--end of included part-->

```

Following methods are used for SIMS edit/insert form creation:

Method	Description
Sims.changeMark	
Sims.getCheckBox	Create check box HTML element, checked if value of attribute is 1 and unchecked if value is 0. Only for IT type.
Sims.getLongName	Print LongName from meta-data table. Is used with all types of attributes.
Sims.getPromterVt	Create ...HTML element with values from appropriate vocabulary table. Only for VT type.
Sims.getInput	Create input HTML element with value of attribute. For ST, DT, IT and FT types.
Sims.getYear	Create input HTML element with year from
Sims.getMonth	
Sims.getDay	

Table 4 Methods for attributes processing

Form for Middle level record edit/insert is changed: