Making field changes to the test catalog

[History 2](#_Toc389220053)

[Current Advantages: 2](#_Toc389220054)

[Current Disadvantages: 3](#_Toc389220055)

[The way forward 3](#_Toc389220056)

[Steps for maintaining the test catalog 3](#_Toc389220057)

[Best practices 3](#_Toc389220058)

[Making changes to OpenELIS test catalog 4](#_Toc389220059)

[Overview 4](#_Toc389220060)

[Understanding the online test catalog 4](#_Toc389220061)

[Writing sql to update the database from the test catalog 7](#_Toc389220062)

[General notes 7](#_Toc389220063)

[Update Unit 8](#_Toc389220064)

[Add Unit 8](#_Toc389220065)

[Deactivate Unit 8](#_Toc389220066)

[Update a select list member 8](#_Toc389220067)

[Insert a new select list member 9](#_Toc389220068)

[Deactivate a select list member 9](#_Toc389220069)

[Update a sample type 9](#_Toc389220070)

[Insert a sample type 10](#_Toc389220071)

[Deactivate a sample type 10](#_Toc389220072)

[Update panel name 10](#_Toc389220073)

[Insert new panel 11](#_Toc389220074)

[Deactivation a panel 11](#_Toc389220075)

[Test Operations 11](#_Toc389220076)

[Updated test name 12](#_Toc389220077)

[Changing the sample type for a test 12](#_Toc389220078)

[Changing the panel for a test 12](#_Toc389220079)

[Removing a test from a panel 13](#_Toc389220080)

[Adding a test to a panel 13](#_Toc389220081)

[Changing UOM for a test 13](#_Toc389220082)

[Changing the test section/unit for a test 14](#_Toc389220083)

[Adding tests 14](#_Toc389220084)

[Tables 14](#_Toc389220085)

[Result types 14](#_Toc389220086)

[Result limits 15](#_Toc389220087)

[Limits 15](#_Toc389220088)

[Add test without uom 15](#_Toc389220089)

[Add test with uom 16](#_Toc389220090)

[Binding test to sample type 16](#_Toc389220091)

[Test result for numeric, free text or alpha numeric results 17](#_Toc389220092)

[Test result for select list, multi-select or cascading multi-select results 17](#_Toc389220093)

[Result limits for numeric values with no age or sex qualification 18](#_Toc389220094)

[Result limits for numeric values with only sex qualification 18](#_Toc389220095)

[Result limits for numeric values with only age qualification 19](#_Toc389220096)

[Result limits with both age and sex qualifiers 20](#_Toc389220097)

[Applying the changes to the database 21](#_Toc389220098)

[Where to put your SQL code 22](#_Toc389220099)

[Applying Liquibase changes 23](#_Toc389220100)

[Code: 23](#_Toc389220101)

[Example: 23](#_Toc389220102)

[Checking for errors 24](#_Toc389220103)

[Correcting errors 25](#_Toc389220104)

# History

The historic way of updating the test catalog for OpenELIS global has been for the country office to request a change to the test catalog and the headquarters office to make the changes which would be available in the next release.

## Current Advantages:

1. Headquarters has the skill set to make the changes
2. Headquarters provided a second level of QA to insure that the requested changes were correct.
3. The changes could have end to end testing done to insure that there were no behavioral bugs
4. Any bugs reported from the field having to do with changes to the test catalog could be tested at headquarters
5. Test catalogs being used at multiple sites would be uniform

## Current Disadvantages:

1. It could take up to 6 months to get a new test in the test catalog or to have an error corrected
2. We were not developing the country expertise to maintain their test catalogs.

The disadvantages of this system outweigh the advantages so we are moving responsibility for maintaining the test catalog away from headquarters and towards the country offices.

# The way forward

The move to country office ability to modify the test catalog will be in three steps

1. Near term. Guidance on writing sql scripts to make the necessary changes to the database (this document).
2. Medium term. Providing a UI within OpenELIS to make the changes.
3. Far term. Creating an application from which lab managers will be able to import preconfigured tests from a list into OpenELIS.

Steps for maintaining the test catalog

1. Update the online test catalog. This may be done by personal with expertise in defining tests. They should make notes on what they have changed or added to the online test catalog otherwise it will be impossible to know what needs to be added to the database. Your responsibility will be to double check their work, to note obvious misspellings, to make sure the result types match the reference values and that all required information has been added. Expect errors.
2. Write the sql to modify or add tests.
3. Apply the changes to the database using Liquibase.

# Best practices

Directly updating a database always has some risk, best practices are a way to reduce that risk.

1. Review the changes requested. Make sure they are complete and consistent.
2. Always work with a copy of the database on an instance of OpenELIS that is separate from the production
3. Before starting work make a backup copy of the database
4. Always set the encoding of files to utf-8, without BOM (byte order mark). A free editor which supports utf-8 is Notepad++ <http://notepad-plus-plus.org/fr/>
5. If there are errors when applying the updates with Liquibase run it with the error flag, ‑‑logLevel=fine
6. If you are trying to debug the SQL statements it is ok to run them directly from a Postgres client tool but before you run Liquibase again restore the database with the copy you made when you started.
7. Validate your work that the changes are what you expected
8. If possible have the person who requested the change validate the work
9. Before you make the change in production make a backup of the production database
10. Send a copy of all new and modified files to the Seattle office. We will add them to version control OR if you have forked openelis-global at github you can make a pull request

# Making changes to OpenELIS test catalog

## Overview

1. Make changes to the online test catalog. The online test catalog is where non-developers will look to see what tests are available and how they are configured. They serve as the reference document for tests in OpenELIS. We currently maintain 5 catalogs, in addition Cote d’Ivoire RetroCI maintains its own test catalog.
   1. Haiti clinical -- <https://docs.google.com/spreadsheet/ccc?key=0Aq9_KNCzCMZ-dHBrNnlxUXZHZ2dISnBWdW5GV0p4LUE&usp=drive_web#gid=6>
   2. Haiti LNSP -- <https://docs.google.com/spreadsheet/ccc?key=0AsypeeVl0ZbtdExyRlpBWkxZdFJ5WnZNa2xFZGtkU1E&usp=drive_web#gid=6>
   3. Cote d’Ivoire LNSP -- <https://docs.google.com/spreadsheet/ccc?key=0AsypeeVl0ZbtdGNQV00xcWNNT2VERWwxOE5DWnJ0S0E&usp=drive_web#gid=0>
   4. Cote d’Ivoire Regional labs -- <https://docs.google.com/spreadsheet/ccc?key=0AmHBu5WgZ956dDBYSFpUYmJaRDZWSXllTnh2alNwNmc&usp=drive_web#gid=0>
   5. Cote d’Ivoire IPCI -- <https://docs.google.com/spreadsheet/ccc?key=0AsypeeVl0ZbtdFRlT3dDTFlBamFVZmpxa3RmQUlHYmc&usp=drive_web#gid=0>
2. Write the needed sql scripts to make the needed changes in the database, that is the bulk of this document.
3. Apply the changes to the database with liquibase
4. Verify that the changes match the test catalog and that they are what the lab manager wanted.

# Understanding the online test catalog

The test catalogs for all configurations have the same basic format but do differ in some details. We will use Haiti LNSP as an example and explain each field. The order of the tests does matter. The order in the test catalog will be the same as the order in OpenELIS. If the same test is entered more than once then what matters is the first time it is found in the catalog.

|  |  |  |  |
| --- | --- | --- | --- |
| Header name | Notes | Mandatory | Ignored |
| Item |  |  | X |
| Test name French (<=60 ch) | The name of the test as it will appear on sample entry. Must be less than 60 characters. | X |  |
| Test Name in English |  |  | X |
| Lab Section (French) | Where in the lab this test will be done. It should be an existing lab section. If a new lab section is being added then this work cannot currently be done in the country office. There is one special case. If “user” is entered that means that the user selects the lab section after they select the test | X |  |
| Panel association (<=30ch) | The name of the panel for this test. The header is labeled as < 30 characters but it can be up to 64 charaters. If the test is part of more than one panel then a new row is needed for each panel. |  |  |
| ch count |  |  | X |
| Test Name Abbreviation (<= 16ch OE) | This is often used on reports where space is at a premium. It depends on the lab policy. If the value is not given then the test name is used and truncated at 16 characters |  |  |
| Blank |  |  | X |
| (Current) OE 2.3 test name (needed for OE update) |  |  | X |
| New tests vs old tests |  |  | X |
| SAMPLE TYPE (FRENCH) | The type of sample this test is done on. There is one special case. If “Variable” is entered that means that the user will select the actual sample type after they select the test | X |  |
| SAMPLE TYPE (ENGLISH) | The English translation of the sample type. Will be used if the user changes the language to English | X |  |
| LOINC | The LOINC code for this test done on this sample |  |  |
| RESULT TYPE | Determines how the user will be able to enter the results. Must be one of "Free Text", "Select List",  "Numeric", "Alpha numeric", "Multiselect" or "Cascading Multiselect".   1. Free Text results in a text area input 2. Select List results in a select list input 3. Numeric results in text input which only accepts numbers 4. Alpha numeric results in a text input which accepts both numbers and characters 5. Multiselectresults in a multiselect list 6. Cascading multiselect results in a select list plus buttons to make more copies of the multiselect list. Used to group multiselect choices | X |  |
| NORMAL RANGE MIN | The lower end of the normal range. This can vary by sex and age. If sex is specified then both sexes must be specified. If age is specified then it must be possible to know the value for any age. | Only for numeric result types. Defaults to 0 if not given |  |
| NORMAL RANGE MAX | The upper end of the normal range. Must be greater than the NORMAL RANGE MIN value. This can vary by sex and age. If sex is specified then both sexes must be specified. If age is specified then it must be possible to know the value for any age. | Only for numeric result types. Defaults to positive infinity if not given |  |
| VALID RANGE MIN | The lower end of the valid range. Must be less than or equal to NORMAL RANGE MIN This can vary by sex and age. If sex is specified then both sexes must be specified. If age is specified then it must be possible to know the value for any age. | Only for numeric result types. Defaults to negative infinity if not given |  |
| VALID RANGE MAX | The upper end of the valid range. Must be greater than or equal to NORMAL RANGE MAX This can vary by sex and age. If sex is specified then both sexes must be specified. If age is specified then it must be possible to know the value for any age. | Only for numeric result types. Defaults to positive infinity if not given. |  |
| REFERENCE RANGE | The value for a normal healthy person for this test. Must be one of the values in the select list | Only for select lists, multiselect Lists or cascading multiselect list |  |
| UNITS | The units of measure for numeric results although it can also apply to select lists |  |  |
|  |  | Only for select lists, multiselect Lists or cascading multiselect list |  |
| SELECT LIST OPTIONS | The list of options for select lists, multiselect lists and cascading multiselect lists. The order in this list is the order in which they will be presented to the user. | Only for select lists, multiselect Lists or cascading multiselect list |  |
| QUALIFIIED OPTIONS | An entry here will result in a text box being presented to the user if they select that option. Only applicable to select lists, multiselect list or cascading multiselect lists |  |  |
| validation date | Used by the test catalog maintainers to track when validation is done |  | X |
| validated by | Used by the test catalog maintainers to track who did the validation |  | X |
| Notes | If a change is made to the test catalog it should be noted here. As soon as the change is in the database the notes should be deleted. This is to assist the people doing the work of updating the DB |  |  |

# Writing sql to update the database from the test catalog

## General notes

1. All sql should be in files with UTF-8 no BOM (bit order marker) encoding. If BOM is not mentioned then that is the correct UTF-8
2. Be careful about cutting and pasting from this document. Word changes single quotes ' to something else, ‘
3. There are several places where you will be asked to give the sort order. This is a number which determines the order in which the item will be displayed when it is in a list. To determine the number select the item before and after it and select a number which is equal distance between their numbers. When the database was built items were added with space between the sort numbers but as time goes on that space may be filled in. If this happens then the numbers above where you want to add may need to be moved up. An example of how to move all of the panel sort\_order numbers up above 60 is: Update clinlims.panel set sort\_order=sort\_order + 10 where sort\_order > 60;
4. The order of applying test catalog updates does matter. We want to do it in the order of increasing dependencies. The suggested order is test section/unit, dictionary (select list values), sample type, panel and then tests.
5. When identifying tests in the database always use the description, not the name. The description includes the sample type and is unique. The test name may be in several rows in the database.
6. If a name has an ' in it then it has to be treated specially. As an example if we want to set the name in the test table to Oeuf d'oxyure we could not use name='Oeuf d'oxyure' The parse would see name='Oeuf d' and would not know the meaning of oxyure' Instead we use name=$$Oeuf d'oxyure$$

## Update Unit

### Context:

The only time a unit is ever updated is when it is incorrect as is; i.e. it is currently “Ppm” but it should be “ppm”. It should NEVER be updated if it in any way changes the meaning of the entry. Any change to units should be cleared by a lab manager. If the issue is an incorrect unit for a particular test then a new unit should be entered and used for that test.

### Code:

update clinlims.unit\_of\_measure set name='<new unit>', description='<new unit>', lastupdated=now() where name='<old unit>';

### Example:

update clinlims.unit\_of\_measure set name='ppm', description='ppm', lastupdated=now() where name='Ppm';

## Add Unit

### Context:

A new unit of measure is needed

### Code:

INSERT INTO clinlims.unit\_of\_measure( id, name , description, lastupdated)

VALUES ( nextval( 'unit\_of\_measure\_seq' ) , '<unit of measure>' , '<unit of measure>' , now());

### Example:

INSERT INTO clinlims.unit\_of\_measure( id, name , description, lastupdated)

VALUES ( nextval( 'unit\_of\_measure\_seq' ) , 'trophzoïtes/µl de sang' , 'trophzoïtes/µl de sang' , now());

## Deactivate Unit

### Context:

There is no reason to deactivate a unit of measure

## Update a select list member

### Context:

The only time a select list member is ever updated is when it is misspelled or capitalization needs to be made uniform; i.e. it is currently “negative” but it should be “Negative” because all of the other items on the list begin with an upper case letter. It should NEVER be updated if it in any way changes the meaning of the entry.

### Code:

update clinlims.dictionary set dict\_entry='<new name>' , lastupdated=now() where dict\_entry = '<old name>';

### Example:

update clinlims.dictionary set dict\_entry='Negative' , lastupdated=now() where dict\_entry = 'negative';

## Insert a new select list member

### Context:

A select list entry is added to the test catalog when the entry currently does not exist. To check if it does exist use

Select \* from clinlims.dictionary where dict\_entry ilke '<dictionary name>';

### Code:

INSERT INTO clinlims.dictionary ( id, is\_active, dict\_entry, lastupdated, dictionary\_category\_id )

VALUES ( nextval( 'dictionary\_seq' ) , 'Y' , '<dictionary name>' , now(), ( select id from clinlims.dictionary\_category where description = 'Haiti Lab' ));

### Example:

INSERT INTO clinlims.dictionary ( id, is\_active, dict\_entry, lastupdated, dictionary\_category\_id )

VALUES ( nextval( 'dictionary\_seq' ) , 'Y' , 'ADN VIH-1 Indeterminé' , now(), ( select id from clinlims.dictionary\_category where description = 'Haiti Lab' ));

## Deactivate a select list member

### Context:

The only time a select list member would be deactivated is if it was part of dictionary category, such as ‘Possible nationalities’ which was listed for the user but should no longer be part of that list. The reason dictionary entries are not deleted is because it may already be referenced as part of an existing record.

### Code:

update clinlims.dictionary set is\_active='N' , lastupdated=now() where dict\_entry = '<dictionary entry>';

### Example:

update clinlims.dictionary set is\_active='N' , lastupdated=now() where dict\_entry = 'Negative';

## Update a sample type

### Context:

The sample type is misspelled or capitalization needs to be changed. The meaning of the sample type must not be changed. The maximum length for the description is 40 characters, the maximum length for the local\_abbrev is 10 characters. If the local\_abbrev has to be wider it is safe to change the length in the correct schema.xml. The name will also need to be updated in MessageResources.properties and MessageResources\_fr.properties if there is a value in display\_key.

### Code:

update clinlims.type\_of\_sample set description='<description of sample type>', local\_abbrev='<name of sample type>', lastupdated=now() where local\_abbrev='<current name>';

### Example:

update clinlims.type\_of\_sample set description='Selles', local\_abbrev='Selles', lastupdated=now() where local\_abbrev='Selle';

## Insert a sample type

### Context:

A new sample type is needed. The maximum length for the description is 40 characters, the maximum length for the local\_abbrev is 10 characters. If the local\_abbrev has to be wider it is safe to change the length in the correct schema.xml. The name will also need to be added to MessageResources.properties and MessageResources\_fr.properties for the display key value.

### Code:

INSERT INTO clinlims.type\_of\_sample( id, description, domain, lastupdated, local\_abbrev, display\_key, is\_active )

VALUES ( nextval( 'type\_of\_sample\_seq' ) , '<sample type>','H', now() , '<sample type>', '<localization key value>', 'Y');

### Example:

INSERT INTO clinlims.type\_of\_sample( id, description, domain, lastupdated, local\_abbrev, display\_key, is\_active )

VALUES ( nextval( 'type\_of\_sample\_seq' ) , 'Selles','H', now() , 'Selles', 'sample.type.Selles', 'Y');

## Deactivate a sample type

### Context:

The sample type is no longer being handled in the lab.

### Code:

update clinlims.type\_of\_sample set is\_active='N', lastupdated=now() where local\_abbrev='<sample type>';

### Example:

update clinlims.type\_of\_sample set is\_active='N', lastupdated=now() where local\_abbrev='Selles';

## Update panel name

### Context:

Somebody has noticed that a panel is misspelled or has the incorrect name. The name will also need to be updated in MessageResources.properties and MessageResources\_fr.properties if there is a value in display\_key.

### Code:

update clinlims.panel set name='<new name>', lastupdated=now() where name='<old name>';

### Example:

update clinlims.panel set name='Hemogramme', lastupdated=now() where name='Hemogramme-Auto';

## Insert new panel

### Context:

A new panel is needed. The panel will need to be inserted into both the panel table and into the sampletype\_panel table so that it will appear when that sample type is selected. The name will also need to be added to MessageResources.properties and MessageResources\_fr.properties for the value in display\_key.

### Code:

INSERT INTO clinlims.panel( id, name, description, lastupdated, display\_key, sort\_order) VALUES

(nextval( 'panel\_seq' ) , ‘<panel name>' , '<panel name>' , now() ,'<localization key>' ,<display order>);

INSERT INTO clinlims.sampletype\_panel (id, sample\_type\_id, panel\_id ) VALUES

(nextval( 'sample\_type\_panel\_seq') , (select id from clinlims.type\_of\_sample where description = '<type of sample description>' ) , (select id from clinlims.panel where name = '<panel name>' ) );

### Example:

INSERT INTO clinlims.panel( id, name, description, lastupdated, display\_key, sort\_order) VALUES

(nextval( 'panel\_seq' ) , 'Méningite-PCR' , 'Méningite-PCR' , now() ,'panel.name.meningitis' ,25);

INSERT INTO clinlims.sampletype\_panel (id, sample\_type\_id, panel\_id ) VALUES

(nextval( 'sample\_type\_panel\_seq') , (select id from clinlims.type\_of\_sample where description = 'LCR' ) , (select id from clinlims.panel where name = 'Méningite-PCR' ) );

## Deactivation a panel

### Context:

The lab no longer supports a previous panel. We cannot just delete panels because there may be existing results which depended on those panels so instead we deactivate them.

### Code:

update clinlims.panel set is\_active='N', lastupdated=now() where name = '<panel name>';

### Example:

update clinlims.panel set is\_active='N', lastupdated=now() where name = ' Méningite-PCR'';

## Test Operations

Updating tests is more complicated than the above items and will be broken into smaller steps. For a successful update or addition more than one of the sql statements will be needed.

## Updated test name

### Context:

The test is either spelled wrong or does not conform to naming standards for the test. The rename must not change the processes in the lab.

### Code:

UPDATE clinlims.test

SET description=replace(description,'<old name>','<new name>'),

reporting\_description=replace(reporting\_description,'<old name>','new name'),

name=replace(name,'<old name>','<new name>'), lastupdated = now()

WHERE name='<old name>';

### Example:

UPDATE clinlims.test

SET description=replace(description,'Mycobacterium tuberculosis Drug Resistant','Mycobacterium tuberculosis résistant'),

reporting\_description=replace(reporting\_description,'Mycobacterium tuberculosis Drug Resistant','Mycobacterium tuberculosis résistant'),

name=replace(name,'Mycobacterium tuberculosis Drug Resistant','Mycobacterium tuberculosis résistant'), lastupdated = now()

WHERE name='Mycobacterium tuberculosis Drug Resistant';

## Changing the sample type for a test

### Context:

An error has been made and the test is listed under the wrong sample type. The old test description will include the old sample type name. The test description name will also need to be updated. The order of these updates is important.

### Code:

update clinlims.sampletype\_test set sample\_type\_id=(select id from type\_of\_sample where description = '<new sample type>'), lastupdated= now() where test\_id= (select id from clinlims.test where description='<old description>');

update clinlims.test set description=replace(description,'<old sample type name>','<new sample type name>'), lastupdated = now() where description = '<old test description>'

### Example:

update clinlims.sampletype\_test set sample\_type\_id=(select id from type\_of\_sample where description = 'Selles 1'), lastupdated= now() where test\_id= (select id from clinlims.test where description=' Test Rapide Rotavirus(Selles)');

update clinlims.test set description=replace(description,'Selles','Selles 1'), lastupdated = now() where description = ' Test Rapide Rotavirus(Selles)'

## Changing the panel for a test

### Context:

A test is in the wrong panel and needs to be moved to a new one

### Code:

update clinlims.panel\_item set panel\_id = ( select id from clinlims.panel where name = '<new panel name>'), lastupdated = now()

where test\_id = (select id from clinlims.test where description = '<test description>');

Example:

update clinlims.panel\_item set panel\_id = ( select id from clinlims.panel where name = 'CD4'), lastupdated = now()

where test\_id = (select id from clinlims.test where description = 'CD4 Compte Absolu(Sang Total)');

## Removing a test from a panel

### Context:

A test has been incorrectly assigned to a panel

### Code:

delete from clinlims.panel\_item where panel\_id = ( select id from clinlims.panel where name = '<panel name>') and

test\_id = (select id from clinlims.test where description = '<test description>');

### Example:

delete from clinlims.panel\_item where panel\_id = ( select id from clinlims.panel where name = 'CD4') and

test\_id = (select id from clinlims.test where description = 'CD4 Compte Absolu(Sang Total)');

## Adding a test to a panel

### Context:

A test needs to be added to a panel

### Code:

INSERT INTO panel\_item( id, panel\_id, lastupdated , test\_id)

VALUES ( nextval( 'panel\_item\_seq' ) , (select id from panel where name = '<panel name>') , now(), (select id from test where description = '<test description>' and is\_active = 'Y' ) );

### Example:

INSERT INTO panel\_item( id, panel\_id, lastupdated , test\_id)

VALUES ( nextval( 'panel\_item\_seq' ) , (select id from panel where name = 'Hemogramme') , now(), (select id from test where description = 'Basophiles(Sang Total)' and is\_active = 'Y' ) );

## Changing UOM for a test

### Context:

The unit of measure (UOM) for a test needs to be corrected

### Code:

update clinlims.test set uom\_id = (select id from clinlims.unit\_of\_measure where name = '<uom name>'), lastupdated = now()

where description = 'test description';

Example:

update clinlims.test set uom\_id = (select id from clinlims.unit\_of\_measure where name = 'ppl'), lastupdated = now()

where description = 'CD4 Compte Absolu(Sang Total)';

## Changing the test section/unit for a test

### Context:

The test section (or unit) has to be changed for a test. You will never want to delete a test section.

### Code:

update clinlims.test set test\_section\_id = (select id from clinlims.test\_section where name = '<test section name>' ), lastupdated = now()

where description = '<test description>';

### Example:

update clinlims.test set test\_section\_id = (select id from clinlims.test\_section where name = 'Biologie Moleculaire' ), lastupdated = now()

where description = 'VIH-1 PCR 2 Qualitatif(DBS)';

# Adding tests

## Tables

Adding tests is its own section because it is also more complex. There are five tables which need to be modified to add a test:

1. test – The main table which defines the basic elements of a test
2. sampletype\_test –Binds a test to a sample type
3. test\_result –Binds a test to the type of result
4. result\_limits – Defines the reference ranges or reference values for a test
5. panel\_item – Binds a test to a panel

## Result types

All tests are entered the same for the test and sampletype\_test tables. What is entered in the test\_result and result\_limits table depends on the type of results. We currently support six types

1. R Remark – In test catalog this is marked as “free text and will present the user with a text area
2. D Dictionary – In text catalog this is marked as select l
3. T Titer
4. N Numeric
5. A Alpha (accepts any character)
6. M Multiselect
7. C Cascading Multiselect

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Code | Name | Test catalog | UI | Note |
| R | Remark | Free Text | Text area | Used for descriptive results |
| D | Dictionary | Select List | Drop down select list |  |
| N | Numeric | Numeric | Text box – allows only numeric characters |  |
| A | Alpha numeric | Alpha numeric | Text box – allows mix of numeric and alphabetic characters |  |
| M | Multiselect | Multi select | Multi select dropdown list |  |
| C | Cascading multiselect | Cascading multiselect | Multi select dropdown list with capacity to add duplicate selections for single test | An example of usage if for cell culture in which more than one organism may be found and multiple instances of organism name, stage of development and density may be reported |

## Result limits

The reference value for a numeric result will be a normal range and a valid range. There may be multiple ranges defined by age and/or sex. When defining by age there must not be any gaps; i.e. there must not exist an age for which there is no normal/valid values. The default is for all ages. Ages are defined by months. If ranges are defined for one sex it must also be defined for the other sex.

The reference range for a dictionary or multiselect result is a single value.

## Limits

This document does not handle tests that are not orderable. Some tests are only done as a reflex to the results of other tests and the process for adding reflexes is beyond the scope of this document.

## Add test without uom

### Context:

A new test has been requested. The test does not have a unit of measure associated with it. Most of the time only tests with numeric results have uom. The sort order will be determined by looking at the value of the sort order for the tests before and following the new test and picking a value in the middle. If the loinc code is not known it can be left blank.

### Code:

INSERT INTO clinlims.test( id, uom\_id, description, reporting\_description, is\_active, is\_reportable, lastupdated, test\_section\_id, local\_abbrev, sort\_order, name, loinc )

VALUES ( nextval( 'test\_seq' ) , null , '<test name(sample type)>' , '<reporting description or test name truncated to 16 char>' , 'Y' , 'N' , now() , (select id from clinlims.test\_section where name = '<test section name>' ) ,'<test name>' ,<sort order> , '<test name>' , '<loinc code if known>');

### Example:

INSERT INTO clinlims.test( id, uom\_id, description, reporting\_description, is\_active, is\_reportable, lastupdated, test\_section\_id, local\_abbrev, sort\_order, name, loinc )

VALUES ( nextval( 'test\_seq' ) , null , 'VIH-1 PCR 2 Qualitatif(DBS)' , 'VIH-1 PCR 2 Qualitatif' , 'Y' , 'N' , now() , (select id from clinlims.test\_section where name = 'Biologie Moleculaire' ) ,'VIH-1 PCR 2 Qualitatif' ,562 , 'VIH-1 PCR 2 Qualitatif' , '');

## Add test with uom

### Context:

A new test has been requested. The test has a unit of measure associated with it. Most of the time only tests with numeric results have uom. The sort order will be determined by looking at the value of the sort order for the tests before and following the new test and picking a value in the middle. If the loinc code is not known it can be left blank.

### Code:

INSERT INTO clinlims.test( id, uom\_id, description, reporting\_description, is\_active, is\_reportable, lastupdated, test\_section\_id, local\_abbrev, sort\_order, name, loinc )

VALUES ( nextval( 'test\_seq' ) , ( select id from clinlims.unit\_of\_measure where name=’<UOM NAME>') , '<test name(sample type)>' , '<reporting description or test name truncated to 16 char>' , 'Y' , 'N' , now() , (select id from clinlims.test\_section where name = '<test section name>' ) ,'<test name>' ,<sort order> , '<test name>' , '<loinc code if known>');

### Example:

INSERT INTO test( id, uom\_id, description, reporting\_description, is\_active, is\_reportable, lastupdated, test\_section\_id, local\_abbrev, sort\_order, name, loinc )

VALUES ( nextval( 'test\_seq' ) , ( select id from clinlims.unit\_of\_measure where name='%') , 'CD4 Compte en %(Sang Total)' , 'CD4%' , 'Y' , 'N' , now() , (select id from clinlims.test\_section where name = 'Hematology' ) ,'CD4 Compte en %' ,660 , 'CD4 Compte en %' , '26464-8');

## Binding test to sample type

### Context:

Every test has to be bound to a sample type. Without this step the test will not be presented on the order form.

### Code:

INSERT INTO clinlims.sampletype\_test (id, test\_id , sample\_type\_id) VALUES

(nextval( 'sample\_type\_test\_seq' ) , (select id from test where description = '<test description>' ) , (select id from type\_of\_sample where description = '<sample type description>') );

### Example:

INSERT INTO clinlims.sampletype\_test (id, test\_id , sample\_type\_id) VALUES

(nextval( 'sample\_type\_test\_seq' ) , (select id from test where description = 'VIH-1 PCR 2 Qualitatif(DBS)' ) , (select id from type\_of\_sample where description = 'DBS') );

## Test result for numeric, free text or alpha numeric results

### Context:

Only a single entry for test\_result needs to be made for these types. Sort order does not really matter for results of this type. Test result types of numeric also have an additional value of significant digits. The significant digits are the number of significant digits given in the normal and valid ranges. In the example given below the normal and valid values are given in the table so the significant digits is 2. This can be left out for free text or alpha numeric results. The default is 0.

|  |  |  |  |
| --- | --- | --- | --- |
| Homme: 4.50 / femme: 4.00 / nouveau-ne: 5.00 /enfant : 3.50 | Homme: 6.20 / femme: 5.40 / nouveau-ne: 6.00 /enfant : 6.00 | 0.30 | 7.00 |

### Code:

INSERT INTO test\_result( id, test\_id, tst\_rslt\_type, value , lastupdated, sort\_order, significant\_digits)

VALUES ( nextval( 'test\_result\_seq' ) , ( select id from clinlims.test where description = '<test description>' ) , '<one of N, R or A>' , null , now() , <sort order>, <significant digits>);

### Example:

INSERT INTO test\_result( id, test\_id, tst\_rslt\_type, value , lastupdated, sort\_order)

VALUES ( nextval( 'test\_result\_seq' ) , ( select id from clinlims.test where description = 'Compte des Globules Rouges(Sang)' ) , 'N' , null , now() , 2250, 2);

## Test result for select list, multi-select or cascading multi-select results

### Context:

A result needs to be made for each one of the possible results. Sort order determines how they will be ordered in the select list so they only have to be ordered correctly for each test. There typically several entries for each test that differ only by dictionary value and sort order. The dictionary value may have to be checked against what is in the database, it may differ by capitalization or accents from what is in the test catalog. The ‘is\_quantifiable’ flag will only be true if when that result is selected a numeric result also needs to be given. An example are antibiotic sensitivity tests. If the result is sensitive or intermediate then the size of the affective bacteria area also has to be given. The default is false so most of the time the flag does not have to be listed. It is listed below for clarity even though the value is false

### Code:

INSERT INTO test\_result( id, test\_id, tst\_rslt\_type, value , lastupdated, sort\_order, is\_quantifiable)

VALUES ( nextval( 'test\_result\_seq' ) , ( select id from clinlims.test where description = <test name(sample type)>' ) , '<one of D,M or C>' , ( select max(id) from clinlims.dictionary where dict\_entry ='<dictionary value>' ) , now() , <sort order>, <true or false>);

INSERT INTO test\_result( id, test\_id, tst\_rslt\_type, value , lastupdated, sort\_order, is\_quantifiable)

VALUES ( nextval( 'test\_result\_seq' ) , ( select id from clinlims.test where description = <test name(sample type)>' ) , '<one of D,M or C>' , ( select max(id) from clinlims.dictionary where dict\_entry ='<dictionary value>' ) , now() , <sort order>, , <true or false>);

Example:

INSERT INTO test\_result( id, test\_id, tst\_rslt\_type, value , lastupdated, sort\_order, is\_quantifiable)

VALUES ( nextval( 'test\_result\_seq' ) , ( select id from clinlims.test where description = 'Polio Selles 1(Selles 1)' ) , 'D' , ( select max(id) from clinlims.dictionary where dict\_entry ='Negatif' ) , now() , 1350, false);

INSERT INTO test\_result( id, test\_id, tst\_rslt\_type, value , lastupdated, sort\_order, is\_quantifiable)

VALUES ( nextval( 'test\_result\_seq' ) , ( select id from clinlims.test where description = 'Polio Selles 1(Selles 1)' ) , 'D' , ( select max(id) from clinlims.dictionary where dict\_entry ='Positif' ) , now() , 1360, false);

## Result limits for numeric values with no age or sex qualification

### Context:

This is the simplest case for numeric limits. The following is the source for the example

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test name | Result type | Normal range min | Normal range max | Valid range min | Valid range max |
| 'CD4 Compte en % | Numeric | 30 | 60 | 5 | 65 |

### Code:

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = '<test name(sample type)>' ) , (select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , 0, 'Infinity' , '' ,<normal min>,<normal max> ,<valid min>,<valid max> , now() );

### Example:

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = 'CD4 Compte en %(Sang Total)' ) , (select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , 0, 'Infinity' , '' ,30,60,5,65, now() );

## Result limits for numeric values with only sex qualification

### Context:

Where the only qualifier is sex. The following is the source for the example

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test name | Result type | Normal range min | Normal range max | Valid range min | Valid range max |
| Vitesse de Sedimentation | Numeric | Homme0 ; Femme 0 | Homme15 ; Femme 20 | 5 | 65 |

### Code:

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = '<test name(sample type)>' ) , (select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , 0, 'Infinity' , '<M or F>' , <normal min>,<normal max ,<valid min>,<valid max> , now() );

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = '<test name(sample type)>' ) , (select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , 0, 'Infinity' , '<M or F>' , <normal min>,<normal max ,<valid min>,<valid max> , now() );

### Example:

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = 'Vitesse de Sedimentation(Sang)' ) , (select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , 0, 'Infinity' , 'M' , 0,15,0,100, now() );

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = 'Vitesse de Sedimentation(Sang)' ) ,

(select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , 0, 'Infinity' , 'F' ,0,20,0,100, now() );

## Result limits for numeric values with only age qualification

### Context:

Where the only qualifier is age. The following is the source for the example

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test name | Result type | Normal range min | Normal range max | Valid range min | Valid range max |
| Compte des Globules Blancs | Numeric | 3.5 / nouveau-ne: 10.0 /enfant : 6.0 | 11.0 / nouveau-ne: 25.0 /enfant : 10.0 | 1.0 | 99.9 |

The online test catalog uses labels for ages, in the database these are represented in months with the following conversions:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | French name |  | English name | min age | max age |
|  | neauveau ne |  | newborn | 0 months | 1 month |
|  | nourrisson |  | infant | 1 month | 12 months |
|  | enfant jeune |  | young child | 12 months | 60 months |
|  | enfant |  | child | 60 months | 168 months |
|  | adult (homme, femme) |  | adult (homme, femme) | 169 months | infinity |

### Code:

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = '<test name(sample type)>' ) , (select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , <min age>, <max age>, '' , <normal min>, <normal max>,<valid min> ,<valid max> , now() );

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = '<test name(sample type)>' ) , (select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , <min age>, <max age>, '' , <normal min>, <normal max>,<valid min> ,<valid max> , now() );

Etc.

### Example:

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = 'Compte des Globules Blancs(Sang)' ) , (select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , 0, 1 , '' , 10.0, 25.0,1.0,99.9, now() );

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = 'Compte des Globules Blancs(Sang)' ) , (select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , 1, 168 , '' ,6.0,10.0,1.0,99.9, now() );

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = 'Compte des Globules Rouges(Sang)' ) , (select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , 168, 'Infinity' , '' ,3.5,11.0,1.0,99.9, now() );

## Result limits with both age and sex qualifiers

### Context:

Where the only qualifier is age. The following is the source for the example

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test name | Result type | Normal range min | Normal range max | Valid range min | Valid range max |
| Compte des Globules Rouges | Numeric | Homme: 4.50 / femme: 4.00 / nouveau-ne: 5.00 /enfant : 3.50 | Homme: 6.20 / femme: 5.40 / nouveau-ne: 6.00 /enfant : 6.00 | 0.30 | 7.00 |

### Code:

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = '<test name(sample type)>' ) , (select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , <min age>, <max age>, '<F or M>' , <normal min>, <normal max>,<valid min> ,<valid max> , now() );

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = '<test name(sample type)>' ) , (select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , <min age>, <max age>, '<F or M>' , <normal min>, <normal max>,<valid min> ,<valid max> , now() );

Etc.

### Example:

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = 'Compte des Globules Rouges(Sang)' ) , (select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , 168, 'Infinity' , 'M' ,4.50,6.20,0.30,7.00, now() );

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = 'Compte des Globules Rouges(Sang)' ) , (select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , 168, 'Infinity' , 'F' ,4.00,5.40,0.30,7.00, now() );

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = 'Compte des Globules Rouges(Sang)' ) , (select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , 0, 1 , '' ,5.00,6.00,0.30,7.00, now() );

INSERT INTO result\_limits( id, test\_id, test\_result\_type\_id, min\_age, max\_age, gender, low\_normal, high\_normal, low\_valid, high\_valid, lastupdated)

VALUES ( nextval( 'result\_limits\_seq' ) , ( select id from clinlims.test where description = 'Compte des Globules Rouges(Sang)' ) , (select id from clinlims.type\_of\_test\_result where test\_result\_type = 'N' ) , 1, 168 , '' ,3.50,6.00,0.30,7.00, now() );

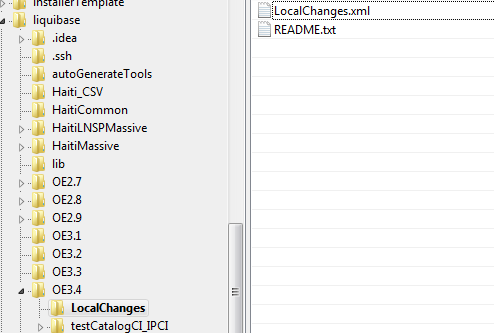
# Applying the changes to the database

We always use Liquibase <http://www.liquibase.org/> to apply any changes to the database. It will be tempting to write the SQL statements, open a database client and apply the SQL statements to the database. If nothing ever goes wrong and you have an excellent system for knowing what statements were successfully applied and in which order and can communicate that information to others then you may not need Liquibase, but very few of us have the discipline to do so. And Liquibase will do all of that for us.

For more information about Liquibase visit both our documentation <https://code.google.com/p/openelisglobal/wiki/Liquibase> and the documentation at the Liquibase site. We will give you enough pointers to get the job done but you may need other sources if there are errors.

## Where to put your SQL code

All of the Liquibase are added to the Liquibase source tree. The organization of the source tree has a directory for each release and in that directory will a directory named LocalChanges for changes made locally. Within that directory is a file called LocalChanges.xml which is where changes are to be made.



To understand how to add the SQL code we are going to pretend we have to change a test name and add a panel. We could do it with one file but to make the example more interesting we will do it with two files, one for each task. The file for changing the test name will be TestNameChange.sql and the one for adding a panel will be PanelUpdate.sql

Both of the files will be added to the LocalChanges directory for the latest release. To have Liquibase know about the files we will have to tell it where they are. To do that we will edit LocalChanges.xml

<databaseChangeLog xmlns="http://www.liquibase.org/xml/ns/dbchangelog/1.9"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog/1.9 http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-1.9.xsd">

<changeSet author="jimmy" id="1" context="haitiLNSP">

<comment>Updates test name changes</comment>

<sqlFile path="OE3.4/LocalChanges/TestNameChange.sql" />

</changeSet>

<changeSet author="jimmy" id="2" context="haitiLNSP">

<comment>adds new panels</comment>

<sqlFile path="OE3.4/LocalChanges/PanelUpdate.sql" />

</changeSet>

</databaseChangeLog>

## Applying Liquibase changes

Before you apply the changes to the database you will need to make sure that the liquibase.properties file is correct. The password can be gotten from the xml file in ../tomcat6/conf/Catalina/localhost/ Remember that the password on production machine will be different from the password on the developer machine. Also remember that if you move the liquibase.properties file off of the host machines then you will be exposing the password.

## Code:

#liquibase.properties

driver: org.postgresql.Driver

classpath: ./lib/postgresql.jar

url: jdbc:postgresql://localhost:5432/clinlims

username: clinlims

password: <password for installed database>

defaultSchemaName: clinlims

changeLogFile: ./changeLogs.xml

contexts: '<liquibase context>'

## Example:

#liquibase.properties

driver: org.postgresql.Driver

classpath: ./lib/postgresql.jar

url: jdbc:postgresql://localhost:5432/clinlims

username: clinlims

password: kiwc3TEqqT

defaultSchemaName: clinlims

changeLogFile: ./changeLogs.xml

contexts: 'haitiLNSP'

To apply the change just run the following in the base directory of Liquibase. Note that the directory separators in the examples ‘/’ are for a Linux system. If you are running on Windows use ‘\’ as the separator.

java -jar ./lib/liquibase-1.9.5.jar update

If you get a response similar to this

Validation Failed:

6 change sets check sum

HaitiLNSPMassive/MassiveUpdate.xml::8::paulsc::(MD5Sum: 921aa77dc4afcb2ff03d4074c3f8ab)

HaitiLNSPMassive/MassiveUpdate.xml::9::paulsc::(MD5Sum: f1a5fc689219fc92ad976932d431af0)

HaitiLNSPMassive/MassiveUpdate.xml::5::paulsc::(MD5Sum: 87ff92904879dda6d266a35b4e6404e)

HaitiLNSPMassive/MassiveUpdate.xml::7::paulsc::(MD5Sum: 53621e41e638781c61bbc524f0423e87)

HaitiLNSPMassive/MassiveUpdate.xml::10::paulsc::(MD5Sum: e6813d9842f25fc1eecf816d3bb5ed7)

HaitiLNSPMassive/MassiveUpdate.xml::12::paulsc::(MD5Sum: a0d5c4ad44ab516426aa523d58f51cf)

That means that the encoding for the console you are using is different than then encoding used when the changes were originally made. The fix for that is to run the command again but include “‑Dfile.encoding=utf-8” as follows:

java -jar -Dfile.encoding=utf-8 ./lib/liquibase-1.9.5.jar update

## Checking for errors

For an update of any size it is not unexpected for there to be errors when trying to make the changes or in the results of the changes. This is not a complete list but does cover some of the more common ones:

### Liquibase errors:

1. Statement is not valid SQL. This is mostly due to not having matching ‘(‘ and ‘)’, leaving out a string closing ’, leaving off the statement terminator ; or using a name with a ‘ and not using $$ as the string opening and closing symbols.
2. Table X does not have column Y. This is most likely due to using “ instead of ‘ to define string values. i.e. “positive” instead of 'positive’.
3. Id cannot be null. Throughout we use constructs such as “*update clinlims.panel\_item set panel\_id = ( select id from clinlims.panel where name = 'CD4'), lastupdated = now() where test\_id = (select id from clinlims.test where description = 'CD4 %(Sang total)')*;” If there is no panel with the name CD4 then the id will be null and panel\_id cannot be null. If there is no test with the description CD4 %(Sang total) then no error will be thrown by Liquibase because it is in the where clause and they are always allowed to return null values.
4. Duplicate primary key. This is a rare error but can be perplexing. Sometimes just running Liqubase again will solve the problem but it often means looking at the details of the error message to understand the problem. Some tables have compound primary keys so that there can only be a single record with the same patient\_id and sample\_id. If you try and add another record with the same patient\_id and sample\_id that exists then it will fail. Find out what table it is and what the duplicate values are and work from there to understand why another record is being added.

### Implementation errors

Once Liquibase has made the changes to the database it has to be validated that there are no errors, that what is in the database represents what is in the online test catalog. This should always be done from the users point of view. If a spelling correction was made, check that the spelling is now correct. If a new test was added then make an order with that test and follow the complete workflow through the system to make sure that it behaves as expected. If it has normal min and max values that vary by sex and age then you will have to create an order for each condition with a different patient who fits the criteria for each variation.

## Correcting errors

Once a changeSet has been applied to a database any SQL files or ChangeSets cannot be altered and reapplied to the same database. If an error has been discovered there are two ways to correct it.

1. The error is discovered during development. The backup copy of the database can be restored, corrections made to the code and it can be reapplied.
2. The error is discovered after deployment to production. Code will need to be written to correct the error and a new ChangeSet created and, after it is validated, applied to the production database.