# Do slots need versions?

Versions are to

* mark the changes of data in chronological order.
* Prevent data incoherence caused by merging incompatible data of different times.

If slots don’t have versions, they must not change (modify examinees, rooms, etc) when the examination starts, typically at the time when data are being transferred from server to subservers. Following this way, we have to set up a scheme that enables the test of data transfer between server and subserver (after the test, we can continue to modify slot data)

The question is when the slots are opened for modification?

Bottom lines are

* Versioning approach seems to be unprofessional. Potential change during the examination is worrying.
* No-version approach is more compact and certain.

# When are slots opened for modification? (or When is a slot marked to start?)

To be simple, certain and professional, a slot starts when an administrator clicks the button “Start examination.” After that, board can’t be modified.

# Then how can we test the data transfer?

We can create another date to test, since MySql supports date range from 1000-01-01 to 9999-12-31. For example, if we want to test for the exam on 2017-07-09, we can create a date 1000-07-09 to test.

To be consistent, we don’t implement an area for testing. When everything looks serious and the same as an official test, students will have chance to get real experience during practicing.

We propose to use the code year “x000 – x400” for testing, and “x500 – x900” for backup, with “x” is different from 2. And client always replaces “x” by 2 for the good experience look.

# Three stages of modification

Data in exam slots have 3 stages, each of which has different limitation of modification.

1. Preparation: Administrator can delete all, and import data. That means everything can be changed freely.
2. Opearation: Administrator can turn off connection, and go to preparation to import data. That means existing data are intact. This is a method to fix something wrong in the examinee list.
3. Archieve: No modification can be done with this stage. View is the only permitted.

# How to handle qpack and reserved qpack in a slot?

The first scenario we think about is to have vQPack and vQPackR, along with using an indicating boolean variable bQPackR.

* Do not let the 2 packs swapping freely. QPack is always the main pack. We only use QPackR in case incident happens. Even when being used, the meaning reserved qpack is unchanged. That means the names “active pack” and “passive pack” don’t convey full meaning of them and should be avoided.
* When we activate the QpackR, we cannot undo the action.
* When QpackR is activated, BOTH packs must be available in server1 because examinees may still need Qpack to submit their qsheets which have been finished before the incident happened.
* Modification rules
  + In server0, bQPackR can only change from “false” to “true”, the value must be stored in database.
  + In server1, bQPackR mustn’t change.
* Using rule
  + In server0, when generating qpack, qpackR is also generated. They yeild twin.  
    If bQPackR is false, only Qpack is sent to server1.  
    If bQPackR is true, both are sent to server1.
  + In server1, no modification is allowed to make. They remain intact after being retrieved from server0.  
    If bQPackR is false, only qpack is available.  
    If bQPackR is true, both pack is