

## ***Jak se co chová?***

### **INSERT**

#### **K insertu dojde v následujících případech:**

- ***Vložen nový nález (ADD)*** – v této chvíli dojde k insertu do tabulek tOccurrences, tHabitats (vazba mezi těmito tabulkami pro tento konkrétní nález bude 1:1 a tomu tak bude navěky). Data v ostatních tabulkách (tAuthor, tPlants, tPublications, tTerritories, tPhytochoria, tVillages) jsou již uloženy a jen se použijí. Pokud nějaký údaj chybí je o tom uživatel informován a daný insert se neprovede. Uživatel si musí aktualizovat chybějící informace a pak může insert opakovat.
- ***Vložen nový nález (IMPORT)*** – postupně jsou do databáze vkládány jednotlivé záznamy (jednotlivá vložení jsou stejná jako při použití ADD). Pokud po vložení X záznamů z N dojde k vkládání záznamu X+1, pro který chybí nějaký údaj v databázi, tak aplikace měla rozumět tuto situaci ošetřit – např. import neskončí, do nějakého souboru se uloží data o nálezu, který nemůže být vložen, ukládají se další nálezy. Po skončení importu bude uživatel informován, že z důvodu chybějících dat (vyjmenovat jakých) nebylo do databáze vloženo Q záznamů. Tyto záznamy jsou připraveny pro import ve vygenerovaném souboru a uživatel je může na importovat po aktualizaci databáze.
- ***Dávkové vložení nálezu (škrtnutí, ADD)*** – v této chvíli dojde k N insertům do tabulky tOccurrences a jednomu insertu do tabulky tHabitats (vazba je v této chvíli N:1. Tato vazba se pro některé nálezy může v budoucnu změnit na 1: 1 viz. Editace). Dál stejné jako u Add.
- ***Dávkové vložení nálezu (dávkový, IMPORT)*** – otázka zda nějaký import bude umět vložit dávkově údaje o kytkách
- ***Vložení nového údaje do tPlants, tTerritories, tVillages, tPhytochoria*** – řešeno importem dat z XML souboru
- ***Vložení nového údaje do tPublication, tMetadata, tAuthor, tUser (tRight)*** - řešeno managemem
- ***K importu může dojít při editaci nálezu*** – jedná se pak o import do tabulky tHabitats. Nastane v případě, když editujeme nález, který byl vložen dávkově, a nechceme změnu promítnout do ostatních nálezu (viz. Editace).

#### **Co zaznamenávat do tabulek historie při insertu?**

- ***Vložen nový nález*** – při vložení nového nálezu se do historie zaznamená jeden záznam, kde se uvede
  - tHistoryChange.cOccurrenceId = „id vloženého nálezu v tOccurrences“
  - tHistoryChange.cRecordId = 0 (defaultní hodnota v db – znamená, že RecordId se pro insert nenastavuje. Indexování řádků v db je od jedné.)
  - tHistoryChange.cOldRecordId = 0
  - tHistoryChange.cOperation = 1 (odpovídá insertu)
  - tHistoryChange.cWhen ... čas provedení insertu
  - tHistoryChange.cWho ... kdo provedl insert

- tHistory.cChangeId, tHistory.cColumnId (v tHistoryColumn bude speciální řádek, kde bude jen název tabulky tOccurrences a na ten se bude odkazovat). Není potřeba ukládat informace o insertu do tHabitats (insert do tHabitats se provede vždy při přidání nového záznamu), protože id nového řádku v tHabitats, které nás bude pro konkrétní záznam zajímat, zjistíme z tOccurrences.cHabitatId.
- **Vložení nového údaje do tPublication, tTerritories, tVillages, tPhytochoria** – vkládání těchto údajů není v rámci vložení nálezu, ale děje se samostatně tzn.
  - tHistoryChange.cOccurrenceId = 0 (bude mít nastavenou defaultní hodnotu, v db budou id záznamu od 1)
  - tHistoryChange.cRecordId bude mít hodnotu id pro vložený řádek
  - tHistoryChange.cOldRecordId = 0
  - tHistoryChange.cOperation = 1 (odpovídá insertu)
  - tHistoryChange.cWhen, tHistoryChange.cWho
  - tHistory.cChange, tHistory.cColumn (v tHistoryColumn bude speciální řádek, kde bude jen název tabulky podle příslušné tabulky, které se insert týkal).

## EDIT

1. Editace nálezu, který **nebyl** vložen dávkově (vazba tOccurrences a tHabitats je 1:1) – můžeme bez problémů dělat update
2. Editace nálezu, který **byl** vložen dávkově (vazba tOccurrences a tHabitats je N:1) - systém se zeptá, zda se daný údaj má změnit pro všechny nebo jen pro ten jeden nález. Je několik případů:
  - A) Údaj je z tabulky tOccurrences – v této chvíli by se systém neměl ani ptát a rovnou daný údaj zeditovat
  - B) Údaj je z tabulky tHabitats + uživatel to chce změnit všem – ok, tak uděláme update do dané tabulky
  - C) Údaj je z tabulky tHabitats + uživatel to chce změnit jen tomu jednomu záznamu – tak do tabulky tHabitats se udělá insert a tOccurrence.cHabitatId se nastaví nová hodnota. V této chvíli se pro tento záznam vytvoří vazba 1:1 mezi tOccurrences a tHabitats, tato vazba se pro tento nález už nikdy nezmění.
3. Editace tabulky tPublications
4. Editace tabulek tTerritories, tVillages a tPhytochoria

## Co zaznamenávat do tabulek historie při editu?

- Příklad 1) ....
  - tHistoryChange.cOccurrenceId = „id editovaného nálezu v tOccurrences“
  - tHistoryChange.cRecordId = „id záznamu obsahujícího editovanou položku“ (může to být tHabitat.cId či cOccurrenceId)
  - tHistoryChange.cOldRecordId .... je to pomocná položka, kterou využijí ve chvíli, kdy došlo ke změně informace o kytce (tOccurrences.cPlantId), o publikaci (tOccurrences.cPublicationId), o territory (tHabitats.cTerritoryId), o phytochoria (tHabitats.cPhytochoriaId) a o village (tHabitat.cNearestVillageId). Ke změně těchto informací dojde tak, že si uživatel vybere z nabídnutého seznamu jinou hodnotu .... tato hodnota je

uložena pod jiným id a tak si v oldRecordId zapamatují to id před změnou. Pokud se edituje jiná položka než „FK“, tak je hodnota 0.

- tHistoryChange.cOperation = 2 (odpovídá editu)
- tHistoryChange.cWhen ... čas provedení editu
- tHistoryChange.cWho ... kdo provedl edit
- tHistory.cChangeId, tHistory.cColumnId
- tHistoryColumn – zde cTableName a cColumnName určí tabulku a sloupek, kterého se daná změna týká
- tHistory – do cOldValue a cNewValue se uloží stará a nová hodnota
- Příklad 2.A) .... bude se chovat jako v případě 1)
- Příklad 2.B) .... bude se chovat jako v případě 1) ... až na to, že je třeba do historie nějak poznačit, že tato změna ovlivnila více záznamů, abychom o tom při „undo“ mohli uživatele informovat. Nebo to lze řešit dotazem do DB a zjistit, jaká je vazba mezi tOccurrences a tHabitats pro dané id. Myslím, že toto řešení bude zbytečně moc časově náročné a přikláním se k uložení informace do tabulek s historií – přidat typ operace EDITGROUP = 4, která nás bude informovat o tom, že šlo o změnu pro více položek.
- Příklad 2.C) ..... do tabulek historie se bude v podstatě zaznamenávat to samé co v případě 1) . Do tHistoryChange.cRecordId se uloží aktuální ID nově vloženého záznamu do tHabitats. Bylo by dobré mít možnost jednoduše zjistit, že zde došlo k přechodu z vazby mezi tabulkami tOccurrences a tHabitats z 1:N na vazbu 1:1. To by se dalo zařídit opět přidáním typu operace EDITONE = 5.
- Příklad 3a4) ....
  - tHistoryChange.cOccurrenceId = 0 ... needituje se to v rámci žádného nálezu
  - tHistoryChange.cRecordId = „id záznamu obsahujícího editovanou položku“ ... tPublication.cId, tTerritories.cId, tVillages.cId, tPhytochoria.cId
  - tHistoryChange.cOldRecordId = 0
  - tHistoryChange.cOperation = 2 (odpovídá editu nebo použít EDITGROUP, protože je pravděpodobné, že změna ovlivní více nálezů. Tento edit proběhne mimo konkrétní záznam, takže se nebudou zobrazovat v historii pro daný záznam, ale až v celkové historii.)
  - tHistoryChange.cWhen ... čas provedení editu
  - tHistoryChange.cWho ... kdo provedl edit
  - tHistory.cChangeId, tHistory.cColumnId
  - tHistoryColumn – zde cTableName a cColumnName určí tabulku a sloupek, kterého se daná změna týká
  - tHistory – do cOldValue a cNewValue se uloží stará a nová hodnota

### **Seznam sloupků, pro které se bude sledovat historie.....**

- tOccurrences
  - cYearCollected, cMontCollected, cDayCollected, cTimeCollected, cDataSource, cHerbarium, cNote ... v podstatě i cPlantId, cPublicationId, cMetadataId, kde se v případě změny id uloží původní id do tHistoryChange.cOldRecordId
  - **nesleduje** se pro cUnitIdDb, cUnitValue, cCreateWho, cCreateWhen (tyto čtyři hodnoty by se neměly měnit), cUpdateWho, cUpdateWhen (jsou v podstatě zaznamenávány v tHistoryChange.cWhen, tHistoryChange.cWho, po „UNDO“ se do těchto položek zapíše aktuální čas a uživatel, který „undo“ pro daný záznam vyvolal.), cIsoDateTimeBegin (tato položka je složena

z ostatních časových položek tabulky, takže ji nemusíme zaznamenávat do historie, jen se bude muset při „undo“ ohlídat její aktualizace při změně některé z časových položek)

- tHabitats
  - cQuadrant, cDescription, cCountry, cAltitude, cLatitude, cLongitude, cNote ..... v podstatě i cTerritoryId, cPhytochoriaId, cNearestVillageId, kde se v případě změny id uloží původní id do tHistoryChange.cOldRecordId
- tPhytochoria, tVillages, tTerritories, tPublications
  - všechny sloupky uvedených tabulek až na tPublications.cReferenceCitation (tento údaj je složen z cCollectionName, cCollectionYearPublication, cJournalName a cJournalAuthorName – nemusíme zaznamenávat do historie, jen se bude muset při „undo“ ohlídat její aktualizace při změně některé z uvedených položek)

## DELETE

1. delete celého nálezu - záznam nebude fyzicky smazán, bude nastavena hodnota položky tOccurrences.cDelete na 1 ( v tHabitats se již položka cDelete nastavovat nebude, protože natavením příznaku delete pro celý nález už máme dostatek informací. K tHabitats přistupujeme jen přes tOccurrencesId ... nikde se z této tabulky nezobrazují seznamy položek). ....**jak či kde řeknu BioCASE, že je daný záznam smazán??**
2. delete z tabulky tPublications – záznam nebude fyzicky smazán, bude nastavena hodnota položky tPublications.cDelete na 1.
3. V tabulkách tTerritories, tVillages, tPhytochoria se **nedělá** operace DELETE ( tyto tabulky obsahují seznamy dat, u kterých se předpokládá minimální množství změn )

### Co zaznamenávat do tabulek historie při editu?

- Příklad 1) ....
  - tHistoryChange.cOccurrenceId = „id editovaného nálezu v tOccurrences“
  - tHistoryChange.cRecordId = 0
  - tHistoryChange.cOldRecordId = 0
  - tHistoryChange.cOperation = 3 (odpovídá delete)
  - tHistoryChange.cWhen ... čas provedení editu
  - tHistoryChange.cWho ... kdo provedl edit
  - tHistory.cChangeId, tHistory.cColumnId
  - tHistoryColumn – použije se speciální řádek (např. řádek, kde je cTableName = tOccurrences a cColumnName = null)
  - tHistory – do cOldValue a cNewValue bude NULL
- Příklad 2) ...
  - tHistoryChange.cOccurrenceId = 0
  - tHistoryChange.cRecordId = „id editovaného nálezu v tPublications“
  - tHistoryChange.cOldRecordId = 0
  - tHistoryChange.cOperation = 3 (odpovídá delete)
  - tHistoryChange.cWhen ... čas provedení editu
  - tHistoryChange.cWho ... kdo provedl edit

- tHistory.cChangeId, tHistory.cColumnId
  - tHistoryColumn – použije se speciální řádek (např. řádek, kde je cTableName = tPublications a cColumnName = null)
  - tHistory – do cOldValue a cNewValue bude NULL
- Příklad 3) .... nic :-)