

Механизм актуализации записи

Материал из Wikipedia

Содержание

Механизм актуализации записи до версии ИРБИС 2011.1 включительно

Механизм актуализации записи до версии ИРБИС 2011.1 включительно:

- рассматриваются две копии актуализируемой записи: последняя копия (т.е. актуальное состояние записи после корректировки) и предыдущая копия (т.е. состояние записи до выполнения корректировки, или точнее - то состояние записи, для которого выполнялась предыдущая актуализация);
- обе копии записи подвергаются обработке (в оперативной памяти) в соответствии с таблицей инвертирования данной БД (<имя_БД>.FST). Обработка состоит в том, что копии записи форматируются по КАЖДОЙ строке таблицы инвертирования (см. структуру FST) и в соответствии с методами инвертирования создаются списки терминов. В результате - формируются два списка терминов: один - связанный с предыдущей копией записи, другой - с последней копией;
- два списка терминов (после сортировки) сравниваются. В результате: а) термины, являющиеся общими для обоих списков отбрасываются (не рассматриваются), б) термины, оставшиеся в списке последней копии (т.е. те термины, которые присутствовали в списке последней копии и отсутствовали в списке предыдущей копии) ДОБАВЛЯЮТСЯ в словарь БД, в) термины, оставшиеся в списке предыдущей копии (т.е. те термины, которые присутствовали в списке предыдущей копии и отсутствовали в списке последней копии) УДАЛЯЮТСЯ из словаря.

Нетрудно понять, что данный механизм имеет существенный недостаток, связанный с нечувствительностью к тому, КАКИЕ и СКОЛЬКО полей записи подвергались изменению (корректировке). Т.е. если изменялись все или большинство полей записи (или точнее – большинство тех полей, которые участвуют в инвертировании), этот механизм эффективен; если же изменялось одно или несколько полей (а именно это имеет место на практике при работе с библиографическими БД) - механизм малоэффективен. И уж совсем "впустую" он работает в тех случаях, когда изменялись только те поля, которые не участвуют в инвертировании (т.е. те, по которым не создаются словари). Проиллюстрировать это можно на следующем примере. При корректировке записи изменялось только поле 910 (экземпляры). В этом случае форматирование последней и предыдущей копий записи по строкам таблицы инвертирования, не имеющим отношения к полю 910, заведомо бессмысленно, поскольку сформированные в результате этого термины окажутся общими для обоих списков и будут отброшены. Имеет смысл использовать только те строки таблицы инвертирования, которые имеют отношение к полю 910. Но этого не происходит - старый механизм "тупо" обрабатывает ВСЕ строки таблицы инвертирования и тратит время на ненужное форматирование. Таким образом, идея нового механизма актуализации лежит как будто на поверхности: определять, какие поля в записи изменялись (сравнивая последнюю и предыдущую копии записи) и отбирать из таблицы инвертирования только те строки, которые имеют отношение к изменившимся полям. (Все именно так - только надо сразу видеть и "обратную сторону", т.е. недостатки нового механизма: в случае изменения всех или большинства полей записи впустую уже будет тратиться время на сравнение копий и отбор строк из таблицы инвертирования.) Процесс сравнения копий (с целью выявления изменившихся полей) - очевиден и его реализация не вызывает никаких проблем. А вот задача отбора строк таблицы инвертирования, имеющих отношение к изменившимся полям, более "хитрая". (И даже более того - в силу изощренности языка форматирования ИРБИС эта задача алгоритмически строго говоря вообще неразрешима. Вот пример - разумеется, чисто теоретический - строки таблицы FST, по которой нельзя определить, к какому полю она относится:

```
100 0 ....&uf('AV',f(val(v100),0,0),'#1').....
```

понятно, что это не поле 100 - а то поле, чья метка указана в поле 100 текущей записи.)

Таблица актуализации

И тем не менее, для решения "хитрой" задачи предлагается следующее. Вводится новая структура - дополнительная по отношению к таблице выбора полей для инвертированного файла. Будем называть ее *таблица актуализации*. Таблица актуализации как отдельная структура сохраняется в текстовом файле с расширением .ifs.

Таблица актуализации создается ИСКЛЮЧИТЕЛЬНО на основе таблицы инвертирования (каким образом - об этом ниже) и является ее ОДНОЗНАЧНЫМ следствием ("придатком"). Отличие таблицы актуализации от таблицы инвертирования только лишь в структуре первого элемента (если смотреть через Редактор РЛ - это первая колонка). В таблице инвертирования это т.н. "точка входа". В таблице актуализации - в первом элементе дополнительно к точке входа указываются (через запятую) МЕТКИ полей, имеющих отношение к данной строке (т.е. метки тех полей, которые используются в соответствующем формате - третьем элементе таблицы).

Схематичный пример строки таблицы инвертирования:

```
100 0 .....v100.....v200.....v300....
```

Ей будет соответствовать следующая строка таблицы актуализации:

```
100,100,200,300 0 .....v100.....v200.....v300....
```

Понятно, что такая структура таблицы актуализации позволяет легко решать задачу отбора строк инвертирования, имеющих отношение к изменившимся полям.

Механизм актуализации записи, действующий начиная с версии ИРБИС 2012.1

Механизм актуализации записи, действующий начиная с версии ИРБИС 2012.1:

- сравниваются последняя и предыдущая копии записи - в результате определяются метки изменившихся полей;
- из таблицы актуализации отбираются строки, имеющие отношение к изменившимся полям и на их основе формируется временная (создаваемая налету для конкретной записи) таблица инвертирования;
- обрабатывает "старый" механизм актуализации на основе временной таблицы инвертирования.

Необходимо отметить, что по понятным причинам при актуализации (т.е. сохранении) НОВОЙ записи (у которой нет никаких предыдущих копий), а также при УДАЛЕНИИ и РАЗУДАЛЕНИИ записи будет применяться старый механизм актуализации. Логичен вопрос: зачем поддерживать две структуры - таблицу инвертирования и таблицу актуализации, почему нельзя отказаться от таблицы инвертирования и применять только новую структуру - таблицу актуализации. Ответ очевиден: исключительно для соблюдения принципа наследования. Т.е. если в структуре БД присутствует таблица актуализации (<имя_БД>.IFS), будет применяться новый механизм актуализации, в противном случае - старый механизм.

Файл <имя_БД>.fst **остаётся обязательным** в структуре базы данных, в отличие от таблицы актуализации <имя_БД>.ifs. (См. также информацию на форуме (<http://irbis.gpntb.ru/read.php?3,76647>) .)

Ссылки

См. также:

- Индекс базы данных ИРБИС
- Язык форматирования системы ИРБИС
- Язык запросов ИРБИС

Источники информации:

- Файл описания релиза 2012.1 (ftp://ftp.gpntb.ru/pub/irbis/RELEASE_12_1.doc)

Источник —

«http://wiki.elnit.org/index.php/%D0%9C%D0%B5%D1%85%D0%B0%D0%BD%D0%B8%D0%B7%D0%BC_%D0%B0%D0%BA%D1%82%D1%83%D0%B0%I

Категории: [Индекс базы данных ИРБИС](#) | [Анонсированные статьи](#)

- Последнее изменение этой страницы: 21:17, 4 сентября 2015.
- Содержимое доступно в соответствии с GNU Free Documentation License 1.3.