1. 1 В схеме oltp\_src\_system создана новая таблица new\_db.
   1. Нужно определить, что будет являться ключом, однозначно идентифицирующем каждую запись.
   2. Нужно создать триггер и подключить его к таблице, который бы на каждое событие над записью, в соответствии с механизмом cdc передавал бы запись, временную метку и тип операции над записью в следующую схему.
2. В схеме oltp\_cdc\_src\_system создадим таблицу new\_db\_changes.
   1. В таблице создаем все поля как в исходной, new\_db, плюс поля с временной меткой (stamp) и типом операции (operation)
3. В следующем слое dwh\_stage создадим таблицу new\_db\_changes\_dwh
   1. В таблице создаем все поля как в new\_db\_changes, плюс поля с хэш ключем (hash\_key) и флагом обработки записи.
   2. Хэш ключ создается для удобства работы с записью, так как идентификатор каждой сущности, как правило, состоит из нескольких полей, а удобней работать с одним полем.
   3. Модифицируем процедуру dwh\_stage.load\_dwh\_stage(), что бы она, по аналогии, копировала из oltp\_cdc\_src\_system.request\_data\_changes в dwh\_stage.request\_data\_changes\_dwh отсутствующие записи.
4. В схеме dwh\_ods создадим таблицу new\_db\_hist. Данная таблица и будет основной детальной таблицей, хранящей версии записей исходной таблицы. Таблица строится на основании таблицы из предыдущего слоя (dwh\_stage. new\_db\_changes\_dwh)
   1. В новой таблице добавляем следующий поля, по сравнению таблицей с предыдущего слоя: valid\_from\_dttm, valid\_to\_dttm, deleted\_flg, deleted\_dttm (начало действия версии, конец действия версии, флаг удалении записи, дата удаления)
   2. В новой таблице исключаем поля, которые можно рассчитать из версионности, например дата обновления записи.
   3. В новой таблице исключаем вспомогательные поля, которые нам будут не нужны далее: временная метка (stump), флаг обработки записи и тип операции (operation)
   4. Модифицируем dwh\_ods.load\_from\_stage\_hist\_request(), что бы он по аналогии вносил записи в dwh\_ods. new\_db\_hist из dwh\_stage.request\_data\_changes\_dwh. При этом:
5. Для работы создадим временную таблицу и скопируем туда необработанные записи из dwh\_stage.request\_data\_changes\_dwh.
6. Вставляем в new\_db\_hist новые записи, одновременно обрабатываем удаленные записи ( заполняем поля deleted\_flg, deleted\_dttm) и в поле valid\_to\_dttm вносим значение '2999/12/31 23:59:59'
7. Корректируем поле valid\_to\_dttm, для этого группируем по ключу hash\_key, сортируем по valid\_from\_dttm. И используя lead, для каждой записи находим следующее значение для valid\_from\_dttm и заносим его в valid\_to\_dttm, если следующее отсутствует, то вносим '2999/12/31 23:59:59'.
8. Если возможно, по тех заданию, то объединяем касающиеся интервалы (подробнее в «5\_Требования к Информационной Системе1\_1»)
9. Помечаем записи, которые брали для работы в dwh\_stage.request\_data\_changes\_dwh как отработанные
   1. Данные в new\_db\_hist получены!!!
10. Теперь создадим вспомогательную таблицу в этом же слое, на основании new\_db\_hist, назовем её agg\_new\_db\_hist
    1. Все поля типа timestamp изменим на data
    2. Модифицируем dwh\_ods.load\_agg\_request\_data\_hist(), что бы он переносил данные из new\_db\_hist в agg\_new\_db\_hist, изменяя тип записей с timestamp на data
11. Модифицируем таблицу dwh\_ods.dim\_work.
    1. Добавляем новые поля для агрегированных показателей, новых сущностей из agg\_new\_db\_hist.
    2. Модифицируем dwh\_ods.load\_dim\_work() , что бы в dim\_work в новые поля вносились агрегированные данные из agg\_new\_db\_hist.
    3. Данные в dim\_work получены!!!
12. Создадим в слое report новое представление v\_recover\_new\_db, по которому можно получить крайнее актуальное состояние данных в системе источнике («5\_Требования к Информационной Системе1\_3»)
13. Создадим в слое report новое представление v\_dif\_source\_target\_new\_db, по которому можно увидеть отличие между восстановленными данными, из хранилища и исходными.
14. Создадим в слое report новое представление v\_dif\_source\_target\_new\_db, по которому можно увидеть частичную разницу между восстановленными данными, из хранилища и исходными. (Исключаются восстанавливаемые поля (пункт 4.б), например поле – дата обновления записи)
15. Модифицируем представление report .v\_agg\_per\_week, добавив новые поля (пункт 6.a )
16. Модифицируем представление report.v\_corr.
    1. Добавляем новые поля (пункт 6.a )
    2. Добавляем новые строчки, в поле agg\_name – имя новой переменной, а в остальных - корреляция переменной с именем поля к переменной с именем в agg\_name.

Не забываем логировать все операции по новой таблице и проверки данных.