# Пояснительная записка

**Шаг 0 Создать скрипт переноса данных из источника в хранилище**
создание таблицы `tmp_sources` с данными из всех источников `source1`, `source2`, `source3`, `external_source`.
обновление существующих записей и добавление новых в `dwh.d_craftsmans`, `dwh.d_products`, `dwh.d_customer`, `dwh.f_order`

**Шаг 1 Создать дополнительную таблицу***

`load_dates_customer_report_datamart` - две основные колонки: id записи, дата и время загрузки новых данных, определяться как максимальное время из всего, что было загружено в хранилище. На основе даты принимать решение о том, какие данные были изменены или добавлены.

**Шаг 2 Создать таблицу с данными, которые нужно обновить для витрины или добавить новые из хранилища - `dwh_delta`.

Объединить данные из таблиц: `f_order`, `d_craftsman`, `d_customer`, `d_product` с данными из витрины `customer_report_datamart` и `external_sourse` . Выберите только те данные, дата загрузки которых не превышает дату из таблицы загрузок `load_dates_customer_report_datamart`. Чтобы определить, какие данные были изменены или добавлены, нужно добавить в запрос следующее условие: **дата загрузки данных в DWH должна быть позже (больше) даты из дополнительной таблицы.**
Первый раз инкрементальная загрузка сработает как полное перестроение витрины: изначально витрина пустая, поэтому нужно будет загрузить в неё все данные. Чтобы в витрину данные попали в первый раз, надо сначала в таблицу загрузок (`load_dates_customer_report_datamart`) вставить значение с максимально старой датой, чтобы началась первая загрузка `'1900-01-01'`, т.е. все данные в витрине условно старые (при повторной загрузке уже не будем писать 1900).
если в "старой" витрине не было нового `craftsman_id` то при соединении витрины ( `left join` ) с таблицей `order` `craftsman_id` с алиасой `exist_craftsman_id` примет значение Null.

**Шаг 3 Создать таблицу с данными, которые нужно только обновить - `dwh_update_delta`

`dwh_update_delta` - запрос в таблицу `dwh_delta`, который оставит `customer_id` только изменённых данных ( `WHERE exist_customer_id IS NOT NULL`) т.е. тех `customer_id`, которые уже есть в витрине, но более поздними временами загрузки в базу. Запрос возвращает только новые `customer_id`.
*Текущая реализация будет немного отличаться от классической инкрементальной загрузки с учётом обновления исторических данных. Такой подход можно назвать `MERGE REFRESH`.

**Шаг 4 Выполнить расчёт витрины только для данных, которые нужно вставить - `dwh_delta_insert_result`

В `dwh_update_delta` находятся `customer_id`, которые нужно обновить. А те данные, что нужно добавить, — в блоке `dwh_delta` с `exist_customer_id is NULL`.
Новые данные для витрины, которые можно просто вставить (insert) в витрину без обновления предварительно сохранить в `dwh_delta_insert_result`.

**Шаг 5 Выполнить расчёт витрины для данных, которые нужно обновить - `dwh_delta_update_result`.

Т.е. все данные, которые не вошли по условию `WHERE exist_customer_id IS NULL` при создании таблицы с данными на шаге №2.
Расчёт витрины для данных обновления похож на расчёт витрины для данных вставки, есть лишь один нюанс: для начала нужно получить конкретные данные по колонкам, которые нужно пересчитать вместе с новыми данными. Для существующих в хранилище мастеров ручной работы появились новые данные по их заработку, продаже и прочим атрибутам, и теперь нужно обновить для них итоговый отчёт.

***Шаг 6 Вставить новые рассчитанные данные в витрину.

Вставить новые данные `dwh_delta_insert_result` в витрину `customer_report_datamart`

***Шаг 7 Выполнить обновление изменённых данных в витрине**

Выполнить обновление изменённых данных `dwh_delta_update_result` в витрине `customer_report_datamart`

**Шаг 8 Вставить крайние даты загрузок из таблицы (новые данные и данные для обновления) в дополнительную таблицу.**

Вставить в дополнительную таблицу `load_dates_customer_report_datamart` данные с самыми последними временами из `dwh_delta`.

Нужно использовать максимальное значение даты из соответствующих столбцов для `customer`, `craftmans`, `products`.
Если же `*_load_dttm` для соответствующих значений — `NULL`, то использовать текущее время на момент вставки.

При вставке максимальной даты загрузки из дельты, во время следующей загрузки данных в витрину запрос на получение дельты будет учитывать только вновь добавленные или изменённые данные с более поздней датой загрузки.
В предыдущих заданиях вы обновили витрину. Чтобы в следующий раз снова посчитать именно дельту, а не все значения из хранилища, вам нужно добавить информацию о дате последнего обновления в таблицу загрузок.