

Алгоритм инкрементального обновления

1. Создание временной таблицы temp_client_txn. Туда сохраняются только те данные, которые были добавлены только первые и последние транзакции (MIN(txn_dt) AS min_txn_dt, MAX(txn_dt) AS max_txn_dt:) в последний день WHERE txn_dt = CURRENT_DATE
2. Далее обновляется таблица dm_client_txn. Для уже существующих клиентов обновляется последняя транзакция (FROM temp_client_txn t WHERE dm_client_txn.client_id = t.client_id AND (t.min_txn_dt < dm_client_txn.first_txn_dt OR t.max_txn_dt > dm_client_txn.last_txn_dt);) и для новых клиентов добавляется первая и последняя транзакции (first_txn_dt = CASE WHEN t.min_txn_dt < first_txn_dt THEN t.min_txn_dt ELSE first_txn_dt END, last_txn_dt = CASE WHEN t.max_txn_dt > last_txn_dt THEN t.max_txn_dt ELSE last_txn_dt)
3. После обновления таблицы временная таблица удаляется

*существуют и другие варианты инкрементального обновления таблицы, например, создание таблицы PUBLICATION ft_card_txn_publication, которая сохраняет все добавления за 1 день. Похожим образом будет работать Change Data Capture (CDC). Также, возможно, успешен был бы алгоритм не min/max за 1 день, а первое и последнее появление client_id.

Но протестирован и реализован мной был алгоритм, описанный выше.