

## Требования к Информационной Системе

### Бизнес-описание

В объекте DFCT\_PHONE хранится список клиентов физлиц (по глобальному ключу MDM) с историей изменения их контактов (телефонов).

### Требования к выходным данным

Схема	Объект	Описание	Карта отображения
dfct_phone@student_71_serina_ms	Dfct_phone	История телефонов клиентов ФЛ	Приложение 1

### Источники

Схема	Объект	Описание	Цель использования
dfct_phone@student_71_serina_ms	counterparty	Клиенты	Получение списка клиентов
dfct_phone@student_71_serina_ms	counterparty_x_uniq_counterparty	Связь глобального и локального клиентов	Перекодировка локальных ключей в глобальные
dfct_phone@student_71_serina_ms	counterparty_contact	Телефоны клиента	Основные атрибуты контактов клиента
dfct_phone@student_71_serina_ms	dict_counterparty_type_cd	Справочник типов клиентов	Фильтрация по списку и отбор клиентов ФЛ

### Регламент и механизм загрузки

Тип загрузки: полный срез

Регламент загрузки: ежедневный

### Входящие параметры

Параметр	Описание
-	

### Условия отбора записей из таблиц-источников

Схема	Объект	Условия отбора записей	Описание
dfct_phone	counterparty	src_cd = 'MDMP'  counterparty_type_cd соответствует значению	отбираются клиенты из системы MDM

		counterparty_type_desc = 'физическое лицо' в справочнике dict_counterparty_type_cd	Отбираются клиенты ФЛ
dfct_phone	counterparty_contact	Для контактных данных  src_cd = 'MDMP', 'RTLL' 'RTLS' 'CFTB' 'WAYN'  Для клиентов - только имеющие соответствие с глобальным ключом MDM	Отбирается история контактов клиентов
dfct_phone	counterparty_x_uniq_counterparty	Для одного уникального ключа клиента uniq_counterparty_rk (соответствующего глобальному ключу MDM) ставится в соответствие несколько локальных ключей counterparty в различные временные периоды	Необходим для перекодировки локальных ключей в глобальные
dfct_phone	dict_counterparty_type_cd	counterparty_type_desc = 'физическое лицо'	Отбираются типы для определения клиента ФЛ

Предварительные преобразования таблиц  
Отсутствуют

Описание алгоритмов заполнения сущности

Название шага	Описание алгоритма
p_dfct_phone_0010	Отбираем <b>всю</b> историю клиентов ФЛ из <b>counterparty</b> , когда клиент являлся ФЛ.  Для этого проводим сравнение типа клиента со справочником <b>dict_counterparty_type_cd</b> . Осуществляется проверка, чтобы системой-источником была MDMP.  <i>Поскольку в ТЗ нет указания, что необходимо отслеживать историю с какого -то периода - принято решение использовать всю историчность существования.</i>
p_dfct_phone_0020_1	отбираем из <b>counterparty_contact</b> историю клиентов НЕ из источников <b>MDMP</b>  присоединяем по локальному клиенту все версии из <b>counterparty_x_uniq_conterparty</b>

	<p>Проверяем и корректируем даты начала <b>effective_from_date</b> и завершения версии <b>effective_to_date</b></p> <p>Алгоритм корректировки дат:</p> <p>Если версия из <b>counterparty_contact</b> является вложенной, то оставляем ее даты.</p> <p>Обрабатываем пересечение версий слева и справа.</p> <p>Если версия <b>counterparty_contact</b> не имеет пересечений с <b>counterparty_x_uniq_conterparty</b>, то даты устанавливаются в <b>NULL</b> и отсекаются на следующем шаге.</p>
p_dfct_phone_0020_2	отбираем из <b>counterparty_contact</b> историю клиентов из источника MDMP
p_dfct_phone_0020	Объединяем (UNION) обе собранные истории <b>p_cdh_demography_0020_1</b> и <b>p_cdh_demography_0020_2</b> , при этом отсекаем данные клиентов не из <b>MDMP</b> версии, выходящие за пределы дат в <b>counterparty_x_uniq_conterparty</b> (отбираем даты NOT NULL)
p_dfct_phone_0030	<p>Отбираем историю о контактах клиентов с глобальным ключом, когда клиенты были ФЛ.</p> <p>Проверяем и корректируем даты начала <b>effective_from_date</b> и завершения версии <b>effective_to_date</b></p>
p_dfct_phone_0040	<p>Для соблюдения историчности "пересечения" телефонов проводим соединение таблицы собранных контактов клиентов физлиц <b>p_dfct_phone_0030</b> с собой. INNER JOIN проводится по номеру телефона. При этом происходит отсечение NULL дат.</p> <p>в качестве даты завершения берём дату начала следующей версии конструкцией - rows between 1 following and 1 following</p>
p_dfct_phone_0050_1	<p>На основании значения атрибута <b>phone_type_cd</b> устанавливаем флаги <b>notification_flg</b> и <b>atm_flg</b></p> <p>Флаг <b>duplication_flg</b> устанавливается если для отдельного номера телефона находится более 1 клиента в одну дату начала версии.</p> <p>Производится кодирование параметров из источников для того чтобы выполнить ранжирование при построении флагов <b>main_dup_flg</b> и <b>main_phone_flg</b>.</p>

p_dfct_phone_0050_2	<p>Создаем таблицу приоритетов для установки флага лучшего телефона для клиента (<b>main_phone_flg</b>) в период времени.</p> <p>Используем конструкцию row_number() over (partition by ..order by)</p> <p>Порядок приоритизации для расчёта <b>main_phone_flg</b>:</p> <ol style="list-style-type: none"> <li>1. trust_system_flg: значение true "лучше" false</li> <li>2. contact_qualify_code: любые коды, содержащие GOOD, лучше прочих, но все вариации GOOD равнозначны</li> <li>3. система источник: MDMP &gt; WAYN &gt; RTLL &gt; RTLS &gt; CFTB</li> <li>4. тип контакта: NotificPhone &gt; ATMPPhone &gt; MobilePersonalPhone &gt; MobileWorkNumber &gt; HomePhone</li> <li>5. дата появления версии: более свежая запись лучше</li> </ol>
p_dfct_phone_0050_3	<p>Для телефонов, являющихся дублями (<b>duplication_flg</b> = true) устанавливаем флаг основного клиента для телефона(<b>main_dup_flg</b>).</p> <p>Используем конструкцию row_number() over (partition by ..order by)</p> <p>Порядок приоритизации для расчёта <b>main_dup_flg</b>:</p> <ol style="list-style-type: none"> <li>1. наличие телефона у других клиентов</li> <li>2. trust_system_flg: значение true "лучше" false</li> <li>3. contact_qualify_code: любые коды, содержащие GOOD, лучше прочих, но все вариации GOOD равнозначны</li> <li>4. система источник: MDMP &gt; WAYN &gt; RTLL &gt; RTLS &gt; CFTB</li> <li>5. тип контакта: NotificPhone &gt; ATMPPhone &gt; MobilePersonalPhone &gt; MobileWorkNumber &gt; HomePhone</li> <li>6. дата появления версии: более свежая запись лучше</li> </ol>
p_dfct_phone_0050	<p>Создаем окончательную историю данных, объединяя данные для дублированных телефонов с данными без дубликатов.</p> <p>При этом устанавливаем в <b>true</b> флаг основного клиента для телефона <b>main_dup_flg</b>, если у данного</p>

	<p>номера телефона нет дублей.</p> <p>Формирование данных окончено.</p>
p_dfct_phone	<p>Отбираем из <b>p_dfct_phone_0050</b> необходимые атрибуты</p> <p>При создании сущности проводим явное преобразование типов для даты начала и конца версий к указанному в ТЗ (:: timestamp(0)).</p>