

Краткое название проверки	SQL-код проверки	Ожидаемый результат	Комментарий
NULL PK	<b>select</b> count(1) <b>from</b> p_dfct_phone <b>where</b> mdm_customer_rk <b>is null</b> ; <b>select</b> count(1) <b>from</b> p_dfct_phone <b>where</b> phone_type_cd <b>is null</b> ; <b>select</b> count(1) <b>from</b> p_dfct_phone <b>where</b> phone_num <b>is null</b> ; <b>select</b> count(1) <b>from</b> p_dfct_phone <b>where</b> business_start_dt <b>is null</b> ;	0	не должно быть незаполненных ключей клиента
-1 в PK	<b>select</b> count(1) <b>from</b> p_dfct_phone <b>where</b> mdm_customer_rk = -1; <b>select</b> count(1) <b>from</b> p_dfct_phone <b>where</b> phone_type_cd = -1; <b>select</b> count(1) <b>from</b> p_dfct_phone <b>where</b> phone_num = -1; <b>select</b> count(1) <b>from</b> p_dfct_phone <b>where</b> business_start_dt = -1;	0	-1 это т.н. потеряшки, когда внешний ключ есть, а соответствующего первичного нет
дубли по PK	<b>select</b> mdm_customer_rk, business_start_dt, phone_type_cd, phone_num <b>from</b> p_dfct_phone <b>group by</b> 1, 2, 3, 4 <b>having</b> count(1) > 1	пусто	не должно быть дублей по ключу (если составной, то по всем атрибутам)
заполненность атрибута	<p>в рамках задачи все атрибуты либо ключи (кроме business_end_dt, которая рассчитывается), либо флаги (значит null не можем проверить)</p> <p>Шаблон для проверок:</p> <b>select</b> count(1) <b>from</b> p_dfct_phone <b>where</b> &attr_name <b>is null</b> где &attr_name - название атрибута, например, full_nm	согласовывается с заказчиком	<p>некоторые атрибуты могут не быть заполнены. если на момент построения</p> <p>в источнике ещё нет данных или атрибут осознанно не заполняется в скрипте, т.к. ещё не проработан алгоритм</p>
пересечения по истории		допускается	проверяем 4 типа пересечения версий в разрезе бизнес-ключа

			<p>(counterparty_rk):</p> <p>версия включает в себя другую версию (outside_flg)</p> <p>версия пересекается с другой версией слева (left_flg)</p> <p>версия пересекается с другой версией справа (right_flg)</p> <p>версия содержится внутри другой версии (inside_flg)</p>
дырки в истории		допускается	<p>проверяем, что все версии заканчиваются:</p> <p>либо на 2999-12-31</p> <p>либо датой начала следующей версии</p>
неправильна я дата завершения	<b>select</b> count(1) <b>from</b> p_dfct_phone <b>where</b> business_start_dt > now() <b>and</b> business_end_dt <> date'2999-12-31'	0	версий с датой завершения в будущем, отличной от 2999-12-31, не должно быть
версии из будущего	<b>select</b> count(1) <b>from</b> p_dfct_phone <b>where</b> business_start_dt > now()	0	<p>версий из будущего не должно быть (в нашем прототипе)</p> <p>в целом иногда они допустимы по согласованию с заказчиком</p>

Бизнес-проверки			
Некорректно установлены даты версий	<b>select count(1) from p_dfct_phone where business_start_dt &gt; business_end_dt</b>	0	Могли неверно установить даты при при пересчете дат при проведении JOIN (здесь проверяем, есть ли дата конца версии больше даты начала версии)
Мертвые души	<b>select count(*) from p_dfct_phone where mdm_customer_rk not in (select distinct counterparty_rk from counterparty)</b>	0	В прототипе не должны присутствовать клиенты, которых нет в таблице источника counterparty
Клиенты ФЛ	<b>select count(*) from p_dfct_phone where mdm_customer_rk not in (select distinct c_.counterparty_rk from counterparty c_ inner join dict_counterparty_type_cd dctl_ on c_.counterparty_type_cd = dctl_.counterparty_type_cd where dctl_.counterparty_type_desc = 'физическое лицо' )</b>	0	Проверить по справочнику и таблице источника counterparty, что отобранные контакты принадлежат клиентам, которые являются физлицами
Флаг основного клиента	<b>with phone_number_count as( select count(mdm_customer_rk) as cnt, duplication_flg, business_start_dt from p_dfct_phone group by phone_num, 2,3 having duplication_flg = true )</b>  <b>select count(1) from phone_number_count where cnt = 1</b>	0	Если телефон принадлежит только одному клиенту, то должен быть установлен флаг основного клиента для этого телефона

			duplication_flg.
--	--	--	------------------