**Требования к выгрузке остатков из BARS GL в STAMT**

# Общее

Для реализации требования необходимо:

1. Разработать процедуру выгрузки остатков по заданным счетам из BARS GL в STAMT
2. Добавить описание запуска процедуры в таблицу заданий DWH.GL\_SCHED
3. Доработать экранную форму по ведению настроечной таблицы DWH.GL\_STMPARM

# Основные требования для реализации задачи

Для выполнения задачи разработки процедуры выгрузки остатков необходимо:

1. Реализовать возможность выгрузки остатков за текущий операционный день (ДАТА = CURDATE) и на дату, отличную от CURDATE (ДАТА < CURDATE)
2. Включить в процедуру входные параметры:

* DateUnload – дата операционного дня, за которую необходимо выгрузить остатки (если DateUnload = NULL, то считать DateUnload = CURDATE)

1. Использовать настроечную таблицу DWH.GL\_STMPARM для отбора счетов, добавив в нее поле IncludeBln для определения по каким счетам необходимо выгружать или не выгружать остатки (структура описана ниже).
2. Включить в экранную форму, отвечающую за заполнение и обновление настроечной таблицы DWH.GL\_STMPARM, поле IncludeBln. При первичном заполнении поля IncludeBln и Include равны.
3. Создать две таблицы с одинаковой структурой (структура и правила заполнения описаны в [Приложении 1](#Приложение1)):

* DWH.GL\_BALSTM – для выгрузки остатков за текущий операционный день или за заданную во входных параметрах дату.
* DWH.GL\_BALSTMD – для выгрузки остатков по счетам, по которым были созданы проводки в текущей или заданной дате и проведены в прошлую дату (по операциям BackValue).

1. Включить в процедуру выгрузки остатков последовательное заполнение обеих таблиц (сначала DWH.GL\_BALSTM, затем DWH.GL\_BALSTMD) данными об остатках по выбранным счетам и за требуемую дату.
2. При каждом запуске процедуры добавлять в таблицу заголовков DWH.GL\_ETLSTMS записи с описанием действия по каждой таблице (со временем начала и окончания).
3. Написать функцию определения даты предыдущего движения по счету относительно заданной даты.

# Описание структуры настроечной таблицы DWH.GL\_STMPARM

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Размерность** | **Nullable** | **Комментарий** |
| Account | Varchar(20) | No | Баланс второго порядка или номер счета |
| AccType | Char(1) | No | Тип счета : B – баланс, A - счет ( буквы латинские) |
| Include | Char(1) | No | Признак включения/исключения (для процедуры выгрузки **проводок**): 0 – не включать, 1 – включать. |
| IncludeBln | Char(1) | No | Признак включения/исключения счета (для процедуры выгрузки **остатков**): 0 – не включать, 1 – включать. |

# Условия отбора записей из таблицы остатков DWH.BALTUR

## **Заполнение таблицы DWH.GL\_BALSTM (***за текущую или заданную дату***)**

Если запрашиваемая дата выгрузки остатков

1. **ДАТА** = CURDATE, отбираем все записи, для которых

BALTUR.DATTO = '2029-01-01'

1. **ДАТА** < CURDATE, отбираем все записи, для которых

**ДАТА** between BALTUR.DAT and BALTUR.DATTO

Данное условие можно использовать и для первого случая, если разница во времени незначительная.

Дополнительно необходимо выбрать только те записи из таблицы DWH.BALTUR, по которым найдены соответствующие записи в таблице DWH.GL\_ACC

BALTUR.BSAACID = GL\_ACC.BSAACID

при условии, что счета открыты на заданную дату

GL\_ACC.DTO <= **ДАТА** and

GL\_ACC.DTC is NULL or GL\_ACC.DTC >= **ДАТА,**

и которые удовлетворяют условиям настроечной таблицы DWH.GL\_STMPARM:

счет включается в выборку (существует запись, удовлетворяющая условию)

* по бал/счету 2-го порядка - GL\_STMPARM.ACCTYPE = ‘B’

(GL\_STMPARM.ACCOUNT = GL\_ACC.ACC2 and GL\_STMPARM.INCLUDEBLN = 1)

или

* по признаку лицевого счета - GL\_STMPARM.ACCTYPE = ‘A’

(GL\_STMPARM.ACCOUNT = GL\_ACC.BSAACID and GL\_STMPARM.INCLUDEBLN = 1)

при этом

* счет не исключен из выгрузки (не существует запись, удовлетворяющая условию) по признаку лицевого счета - GL\_STMPARM.ACCTYPE = ‘A’

(GL\_STMPARM.ACCOUNT = GL\_ACC.BSAACID and GL\_STMPARM.INCLUDEBLN = 0)

## **Заполнение таблицы DWH.GL\_BALSTMD (***за прошлые даты относительно текущей или заданной даты***)**

Для выгрузки остатков по счетам за прошлую дату необходимо сначала сделать выборку из таблицы DWH.GL\_OPER записей со статусом проводки STATE = ‘POST’, с датой операции (PROCDATE) равной заданной и датой проводки (POSTDATE) меньшей даты операции, для которых в таблице DWH.PD найдены соответствующие видимые проводки (INVISIBLE = 0):

GL\_OPER.POSTDATE < GL\_OPER.PROCDATE and GL\_OPER.PROCDATE = **ДАТА** and GL\_OPER.STATE = ‘POST’

В случае наличия операций за прошлые даты (выборка непустая), необходимо выбрать только те записи, которые удовлетворяют условиям настроечной таблицы DWH.GL\_STMPARM (по полю GL\_STMPARM.INCLUDE):

счет по дебету или по кредиту включается в выборку (существует запись, удовлетворяющая условию)

* по бал/счету 2-го порядка - GL\_STMPARM.ACCTYPE = ‘B’

((GL\_STMPARM.ACCOUNT = left(GL\_OPER.AC\_DR,5) or GL\_STMPARM.ACCOUNT = left(GL\_OPER.AC\_CR,5)) and GL\_STMPARM.INCLUDE = 1)

или

* по признаку лицевого счета - GL\_STMPARM.ACCTYPE = ‘A’

((GL\_STMPARM.ACCOUNT = GL\_OPER.AC\_DR or GL\_STMPARM.ACCOUNT = GL\_OPER.AC\_CR) and GL\_STMPARM.INCLUDE = 1)

при этом

* счет по дебету или по кредиту не исключен из выгрузки (не существует запись, удовлетворяющая условию) по признаку лицевого счета - GL\_STMPARM.ACCTYPE = ‘A’

((GL\_STMPARM.ACCOUNT = GL\_OPER.AC\_DR or GL\_STMPARM.ACCOUNT = GL\_OPER.AC\_CR) and GL\_STMPARM.INCLUDE = 0)

Из полученной выборки, если она непустая, необходимо:

* выбрать уникальные счета (СЧЕТ) как по дебету, так и по кредиту операций

В полученный список уникальных значений номеров счетов могут попасть счета, не удовлетворяющие условиям выгрузки проводок по ним, но попавшие в этот список в качестве счетов-корреспондентов.

* отфильтровать данный список по настроечной таблице DWH.GL\_STMPARM по полю GL\_STMPARM.INCLUDEBLN

Условия, по которым необходимо определить включать или исключать счет из выборки, аналогичны, описанным выше, но вместо поля GL\_STMPARM.INCLUDE используем поле GL\_STMPARM.INCLUDEBLN.

* определить дату первой проводки (FirstOperDate) по каждому выбранному счету

Каждая следующая дата (ДАТА = CAL.DAT) определяется по календарю – таблица DWH.CAL для CAL.CCY=’RUR’ and CAL.HOL=’’ and CAL.DAT between FirstOperDate and CURDATE

* для каждого выбранного счета и каждой даты (ДАТА), начиная с даты первой проводки (FirstOperDate) по текущую (CURDATE), необходимо вставить запись в таблицу DWH.GL\_BALSTMD

Для этого необходимо выбрать из таблицы DWH.BALTUR запись, для которой выполняется условие

ДАТА between BALTUR.DAT and BALTUR.DATTO

Поля GL\_BALSTMD.STATDATE и GL\_BALSTMD.ED заполняются соответствующим значением ДАТА.

Для заполнения таблицы DWH.GL\_BALSTMD в основу следует брать данные из таблицы DWH.ACCRLN, а не из таблицы DWH.GL\_ACC, поскольку в выборку могут попасть счета, отсутствующие в таблице DWH.GL\_ACC.

## **Особенности расчета входящих и исходящих остатков**

Для каждой рассматриваемой даты операционного дня (ДАТА), за которую выгружаются остатки, необходимо учитывать следующее:

1. если в BALTUR найдена запись **BALTUR.DAT = ДАТА**, то все данные берутся из записи согласно описанию соответствующих полей в [Приложении 1](#Приложение1),
2. если в BALTUR найдена запись **BALTUR.DAT < ДАТА**, то данные об остатках и оборотах за заданную дату (ДАТА) должны рассчитываться следующим образом:

|  |  |  |
| --- | --- | --- |
| 15 | OpenBlncA | DEC ((baltur.OBAC + baltur.DTAC + baltur.CTAC) / POWER(10,currency.NBDP), 22, 2) |
| 16 | CloseBlncA | DEC ((baltur.OBAC + baltur.DTAC + baltur.CTAC) / POWER(10,currency.NBDP), 22, 2) |
| 17 | DbDocNt | 0 |
| 18 | DbTurnovrA | 0.00 |
| 19 | CrDocNt | 0 |
| 20 | CrTurnovrA | 0.00 |

## **Использование данных других таблиц для заполнения полей таблиц DWH.GL\_BALSTM и DWH.GL\_BALSTMD**

| **Название поля в таблице GL\_BALSTM(D)** | **Таблица GL\_ACC** | **Таблица ACCRLN** | **Размерность** | **Описание** |
| --- | --- | --- | --- | --- |
| FCCCustNum | CUSTNO | CNUM | 8 | Код клиента FLEX |
| EXT\_CustID | Right(CUSTNO,6) | Right(CNUM,6) | 6 | Код клиента Midas (Flex без 2-x лидирующих нулей) |
| accname | SDCUSTPD.BBCRNM  при условии | | 35 | Сокращенное наименование клиента |
| SDCUSTPD.BBCUST = CUSTNO | SDCUSTPD. BBCUST = CNUM |
| RcustName | SDCUSTPD.BXRUNM | | 195 | Русское наименование клиента |
| CustINN | SDCUSTPD.BXTPID | | 12 | ИНН клиента |
| EcustName | trim(SDCUSTPD.BBCNA1) || ' ' || trim(SDCUSTPD.BBCNA2) || ' ' || trim(SDCUSTPD.BBCNA3) || ' ' || trim(SDCUSTPD.BBCNA4) | | 195 | Английское наименование клиента |
| CstAddrKl | SDCUSTPD.BXADDR | | 255 | Юридический адрес |
| CstAddrKF | SDCUSTPD.BXPSAD | | 255 | Почтовый адрес |
| AccBRN | BRANCH | для ACID <> ’’  Отделение = Right(ACID,3)  для ACID = ’’  Отделение = IMBCBBRP.A8BRCD,  при условии  IMBCBBRP.A8BICN = CNUM | 3 | Код отделения Midas (цифровой) – используется также для поиска в таблице отделений по полю imbcbbrp.A8BRCD:  английское наименование отделения –  **imbcbbrp.A8BRNM**  символьный код отделения - **imbcbbrp.A8CMCD**  (можно брать из таблицы филиалов) |
| FCCBranch | DH\_BR\_MAP.FCC\_BRANCH  при условии | | 3 | Код отделения FLEX |
| DH\_BR\_MAP.MIDAS\_BRANCH = BRANCH | DH\_BR\_MAP.MIDAS\_BRANCH = Отделение |
| EBankName | IMBCBBRP.A8BRNM  where | | 35 | Aнглийское наименование отделения |
| IMBCBBRP.A8BRCD = BRANCH | IMBCBBRP.A8BRCD = Отделение |
|  | CBCCN | CCODE | 4 | Код филиала (цифровой) – для поиска в таблице филиалов по полю imbcbcmp.CCBBR:  русское наименование филиала – **imbcbcmp.CCPNR**  символьный код отделения - **imbcbcmp.CCPCD** |
| RBankName | IMBCBCMP.CCPNR  при условии | | 160 | Русское наименование филиала |
| IMBCBCMP.CCBBR = CBCCN | IMBCBCMP.CCBBR = CCODE |
| Branch\_id | CBCC | IMBCBCMP.CCPCD  where  IMBCBCMP.CCBBR = CCODE | 3 | Код отделения (символьный) |
| CBAccount | BSAACID | BSAACID | 20 | 20-значный счет ЦБ |
|  | Substr(BSAACID,6,3) | CBCCY | 3 | Код валюты (цифровой) – для поиска валюты по полю **currency.CBCCY** |
| CurrCode | CCY | для ACID <> ’’  Валюта=Substr(ACID,9,3)  для любого ACID (в т.ч. ACID = ’’)  Валюта=CURRENCY.GLCCY,  при условии  CURRENCY.CBCCY = CBCCY | 3 | Код валюты (символьный) – для поиска валюты по полю **currency.GLCCY**, используется также для определения курса валюты счета |
| Rate | CURRATES.RATE  при условии | | 13,8 | Курс ЦБ за запрашиваемую дату или дату Back Value (ДАТА) |
| CURRATES.DAT = ДАТА and CURRATES.CCY = CCY | CURRATES.DAT = ДАТА and CURRATES.CCY = Валюта |
| currname | currency.GLCCY || ' ' || currency.CYNM  при условии | | 40 | Наименование валюты |
| CURRENCY.GLCCY=CCY | CURRENCY.CBCCY=CBCCY |
| ALT\_AC\_NO | ACID | ACID | 20 | Счет Midas |
| acodname | substr(DESCRIPTION,1,255) | NULL | 255 | Acod name |

# Функция определения даты последней операции по счету

Входные параметры:

* **СЧЕТ**
* **ДАТА**, относительно которой необходимо определить последнее движение по счету - дата, заданная во входных параметрах процедуры выгрузки остатков в STAMT или дата, определяемая для счетов по операциям BackValue

Условие выборки:

BALTUR.BSAACID = **СЧЕТ**

and

**DateLast** between BALTUR.DAT and BALTUR.DATTO, где DateLast = **ДАТА** - 1

Результат выполнения функции:

* Дата предыдущей операции (BALTUR.DAT), если найдена запись, удовлетворяющая условию выборки
* NULL, если не найдена запись, удовлетворяющая условию выборки

# Вставка записи о запуске процедуры выгрузки остатков в таблицу заголовков GL\_ETLSTMS.

При каждом запуске процедуры выгрузки остатков в STAMTв таблицу GL\_ETLSTMS необходимо добавлять запись со следующими значениями полей:

* при заполнении таблицы DWH.GL\_BALSTM

PARNAME = 'BARS\_GL\_STAMT\_BALANCE\_FULL’

PARDESC = 'GL\_BALSTM LOAD’

* при заполнении таблицы DWH.GL\_BALSTMD

PARNAME = 'BARS\_GL\_STAMT\_BALANCE\_DELTA’

PARDESC = ‘GL\_BALSTMD LOAD’

* START\_LOAD = дата и время начала заполнения соответствующей таблицы
* END\_LOAD = дата и время окончания заполнения соответствующей таблицы

Приложение 1

# Структура таблицы и правила ее заполнения для выгрузки остатков в STAMT

| **№ п/п** | **Наименование поля буферной таблицы BARS GL** | **Тип данных** | **данные из BARS GL** | **условия при gl\_acc.BSAACID = СЧЕТ или**  **accrln.BSAACID = СЧЕТ** | **комментарий** | **Бизнес значение поля** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | StatDate | DATE | **ДАТА** = gl\_od.CURDATE  или  **ДАТА** = заданной дате | baltur.BSAACID = **СЧЕТ** and **ДАТА** between baltur.dat and baltur.datto | загрузка текущих остатков по всем выбранным по настройке счетам, независимо от наличия оборотов | business date |
| **ДАТА** = gl\_oper.POSTDATE | baltur.BSAACID = **СЧЕТ** and  baltur.DAT >= ДАТА  при условии: **СЧЕТ** = (gl\_oper.AC\_DR or gl\_oper.AC\_CR) and (gl\_oper.POSTDATE < gl\_oper.PROCDATE and gl\_oper.PROCDATE = gl\_od.CURDATE and gl\_oper.STATE = 'POST') | загрузка остатков при поводках BackValue по счетам из настроечной таблицы: переносим все остатки по счету, начиная с даты BackValue по текущую |
| 2 | StatType | Varchar(1) | ='F' |  |  | If branch status is normal, intermediate otherwise final. |
| ‘I’ – intermediate |
| ‘F’ – final |
| 3 | HostSystem | Varchar(5) | ='BARS' |  |  | constant= 'BARS' |
| 4 | FCCCustNum | Varchar(8) | gl\_acc.CUSTNO или accrln.CNUM |  |  | Customer number |
| 5 | FCCAccount | Varchar(35) | NULL |  |  | Customer account number |
| 6 | FCCBranch | Varchar(3) | dh\_br\_map.FCC\_BRANCH | dh\_br\_map.MIDAS\_BRANCH = gl\_acc.BRANCH или  ((accrln.ACID<>'' and imbcbbrp.A8BRCD = Right(accrln.ACID,3) or (accrln.ACID='' and imbcbbrp.A8BRCD = Отделение)) | если accrln.ACID='', то Отделение = imbcbbrp.A8BRCD при условии imbcbbrp.A8BICN = accrln.CNUM | Branch code of the account |
| 7 | EXT\_CustID | Varchar(8) | Right(gl\_acc.CUSTNO,6) или Right(accrln.CNUM,6) |  |  | Customer No. in MIDAS. Leading Zeros in Customer No. will be deleted |
|
| 8 | AccBRN | Varchar(3) | gl\_acc.BRANCH или  - Right(accrln.ACID,3) для accrln.ACID<>'’  - imbcbbrp.A8BRCD = Отделение для accrln.ACID='' |  | цифровой код  если accrln.ACID='', то Отделение = imbcbbrp.A8BRCD при условии imbcbbrp.A8BICN = accrln.CNUM | External branch code of flexcube padded with leading zeros. MIDAS Branch Code |
| 9 | CBAccount | Varchar(20) | gl\_acc.BSAACID или accrln.BSAACID |  |  | System account number (20-digit).Central bank account number |
| 10 | acctype | Varchar(35) | NULL |  |  | Account name description Field in Customer Account Screen- Basic account details tab. If this is NULL, then with Customer name in Basic Details tab |
| 11 | CurrCode | Varchar(3) | gl\_acc.CCY или currency.GLCCY для | currency.CBCCY=accrln.CBCCY | буквенный код | Currency of the customer account |
| 12 | SQNumber | Number(8) | NULL |  |  | For final statement fill with sequence number for the account within year, ever year starts from 1. |
| 13 | CreateDate | Date | = CURDATE() |  |  | YYYY-MM-DD. Statement creation date |
| 14 | TimeCreate | Timestamp | = NOW() |  |  | DATE HH24:MI:SS Statement creation time |
| 15 | OpenBlncA | Number(22,2) | DEC (baltur.OBAC / POWER(10,currency.NBDP), 22, 2) | currency.GLCCY=gl\_acc.CCY  или  currency.CBCCY=accrln.CBCCY |  | For Final Statement Closing balance from previous statement. |
| 16 | CloseBlncA | Number(22,2) | DEC ((baltur.OBAC + baltur.DTAC + baltur.CTAC) / POWER(10,currency.NBDP), 22, 2) |  | Closing balance as of current statement in account currency . Final statement – closing balance |
| 17 | DbDocNt | Number(4,0) | COUNT (gl\_oper.AC\_DR) | gl\_oper.AC\_DR = **СЧЕТ** and gl\_oper.STATE = 'POST' and gl\_oper.POSTDATE = **ДАТА** | кол-во записей по счету Дебета в таб. gl\_oper | No. of debit entries in current movements file. |
| 18 | DbTurnovrA | Number(22,2) | DEC (baltur.DTAC / POWER(10,currency.NBDP), 22, 2) | currency.GLCCY=gl\_acc.CCY  или  currency.CBCCY=accrln.CBCCY |  | Sum of debit entries in the current movements file in account currency. |
| 19 | CrDocNt | Number(4,0) | COUNT (gl\_oper.AC\_CR) | gl\_oper.AC\_CR = **СЧЕТ** and gl\_oper.STATE = 'POST' and gl\_oper.POSTDATE = **ДАТА** | кол-во записей по счету Кредита в таб. gl\_oper | No. of credit entries in current movements file. |
| 20 | CrTurnovrA | Number(22,2) | DEC (baltur.CTAC / POWER(10,currency.NBDP), 22, 2) | currency.GLCCY=gl\_acc.CCY  или  currency.CBCCY=accrln.CBCCY |  | Sum of credit entries in the current movements file in account currency |
|
| 21 | OpenBlncN | Number(22,2) | DEC (baltur.OBBC \* 0,01, 22, 2) |  | Equivalent of opening balance in Local currency. This will be as is balance stored in the system and no revaluation or any other processing will be involved while showing the balances. |
|
| 22 | CloseBlncN | Number(22,2) | DEC (baltur.OBBC + baltur.DTBC + baltur.CTBC \* 0,01, 22, 2) |  | Equivalent of closing balance in local currency. This will be as Is balance stored in the system and no revaluation or any other processing will be involved while showing balances. |
|
| 23 | DbTurnovrN | Number(22,2) | DEC (baltur.DTBC \* 0,01, 22, 2) |  | Sum of debit entries equivalent in local currency in current movements file |
|
| 24 | CrTurnovrN | Number(22,2) | DEC (baltur.CTBC \* 0,01, 22, 2) |  | Sum of credit entries equivalent in local currency in current movements file. |
| 25 | Rate | Number(13,8) | currates.RATE | currates.DAT = ДАТА and (currates.CCY = gl\_acc.CCY  или  currency.CBCCY=accrln.CBCCY) |  | Exchange rate between account currency and local currency. This could be only what is the exchange rate maintained in the system and need not be in sync with what is the value of account currency figures and local currency equivalent |
| 26 | InBDate | DATE | NULL |  |  | Current FLEXCUBE date |
| 27 | OutBDate | DATE | NULL |  |  | Current FLEXCUBE date |
| 28 | RcustName | Varchar(195) | sdcustpd.BXRUNM | sdcustpd.BBCUST = gl\_acc.CUSTNO  или  sdcustpd.BBCUST = accrln.CNUM |  | Russian customer name |
| 29 | CustINN | Varchar(12) | sdcustpd.BXTPID |  | Customer’s INN |
| 30 | ECustName | Varchar(195) | trim(sdcustpd.BBCNA1) || ' ' || trim(BBCNA2) || ' ' || trim(BBCNA3) || ' ' || trim(BBCNA4) |  | English customer name |
| 31 | CstAddrSW1 | Varchar(35) | NULL |  |  | Customer address SWIFT1 |
|
| 32 | CstAddrSW2 | Varchar(35) | NULL |  |  | Customer address SWIFT2 |
| 33 | CstAddrSW3 | Varchar(35) | NULL |  |  | Customer address SWIFT3 |
| 34 | CstAddrSW4 | Varchar(35) | NULL |  |  | Customer address SWIFT4 |
| 35 | CstAddrKl | Varchar(255) | sdcustpd.BXADDR | sdcustpd.BBCUST = gl\_acc.CUSTNO  или  sdcustpd.BBCUST = accrln.CNUM |  | Customer address KLADR law |
| 36 | CstAddrKF | Varchar(255) | sdcustpd.BXPSAD | почтовый | Customer address KLADR fact |
| 37 | BCustBIC | Varchar(9) | sdcustpd.BXBICC |  | BIC of the account branch |
| 38 | RBankName | Varchar(160) | imbcbcmp.CCPNR | imbcbcmp.CCBBR = gl\_acc.CBCCN или  imbcbcmp.CCBBR = accrln.CCODE | русское наименование филиала | Russian IMB branch name |
|
| 39 | EBankName | Varchar(35) | imbcbbrp.A8BRNM | imbcbbrp.A8BRCD = gl\_acc.BRANCH или  ((accrln.ACID<>'' and imbcbbrp.A8BRCD = Right(accrln.ACID,3) or (accrln.ACID='' and imbcbbrp.A8BRCD = Отделение)) | английское наименование отд.  если accrln.ACID='', то Отделение = imbcbbrp.A8BRCD при условии imbcbbrp.A8BICN = accrln.CNUM | foreign IMB branch name, where customer account is opened. English name of the customer’s branch |
| 40 | BCstCorrAc | Varchar(35) | NULL |  |  | IMB branch correspondence account, where customer account is opened |
| take the "Clearing Bank Code" and get the appropriate correspondence account. Correspondent account of the customer’s branch |
| 41 | BCustSWIFT | Varchar(12) | NULL |  |  | SWIFT-BIC of the customer’s branch. |
| If there is no swift bic for particular branch, take it from the parent branch |
| 42 | BCustINN | Varchar(14) | NULL |  |  | INN of branch of account |
| (do not fill if there is no such field in FCC) |
| 43 | LstOperDat | Date | Дата предыдущей проводки  или  NULL . | Функция определения даты предыдущей проводки для StatDate (business date) | если предыдущей проводки не было, то поле оставляем = NULL | Date of last movement happens before FLEXCUBE business date – this information is not available in FLEXCUBE. System will give only latest date available in Previous date. |
| 44 | ed | DATE | **= StatDate** |  |  | business date |
| 45 | branch\_id | Varchar(3) | gl\_acc.CBCC или imbcbcmp.CCPCD | imbcbcmp.CCBBR=accrln.CCODE | буквенный код | MIDAS branch code of the account |
| 46 | ALT\_AC\_NO | Varchar(20) | gl\_acc.ACID или accrln.ACID |  |  | Alternate account number.(MIDAS Account Number) |
| 47 | accname | Varchar(35) | sdcustpd.BBCRNM | sdcustpd.BBCUST = gl\_acc.CUSTNO  или  sdcustpd.BBCUST = accrln.CNUM |  | Short customer name |
| 48 | acodname | Varchar(255) | substr(gl\_acc.DESCRIPTION,1,255)  или NULL для табл. ACCRLN | **Нет данных**, если они берутся из табл. Accrln | описание счета | Acod name |
| 49 | currname | Varchar(40) | currency.GLCCY || ' ' || currency.CYNM | currency.GLCCY=gl\_acc.CCY  или  currency.CBCCY=accrln.CBCCY |  | Currency name |
| 50 | DateUnload | Date |  |  | Дата операционного дня, за которую запрашиваются остатки  (может не совпадать с StatDate по счетам с операциями в прошлую дату) | Not used in STAMT |