**Полезные ссылки:**

[**https://confluence.nexign.com/pages/viewpage.action?pageId=171189260**](https://confluence.nexign.com/pages/viewpage.action?pageId=171189260)

[**https://confluence.nexign.com/pages/viewpage.action?pageId=320586180**](https://confluence.nexign.com/pages/viewpage.action?pageId=320586180)

**Выбрать любого тестового клиента**

Создадим нового тестового клиента:

BEGIN

lite\_upd\_clients.insert\_new\_client(pclnt\_clnt\_id => clnt\_seq.nextval,

pclis\_clis\_id => 2,

plang\_lang\_id => 1,

pctyp\_ctyp\_id => 99,

pccat\_ccat\_id => 1,

pclpt\_clpt\_id => 1,

pclcl\_clcl\_id => 1,

paccount => '89990406443',

pname => 'YaroslavNX',

pjrtp\_id => 1);

END;

Проверим:

SELECT \* FROM CLIENT\_HISTORIES WHERE ACCOUNT = '89990406443' AND NAME = 'YaroslavNX';

Создадим абонента:

begin

lite\_upd\_clients.create\_subscriber(pclnt\_id => 29342996,

pscrd\_id => 15445,

pimsi => 434081402996794,

pmsisdn => '956042194',

pnset\_id => 10013,

prtpl\_id => 1010);

end;

Проверим:

SELECT \* FROM SUBSCRIBERS WHERE CLNT\_CLNT\_ID = 29342996;

Проверим застрявшие в реквестах запросы на активацию абонента:

SELECT \* FROM REQUESTS R

JOIN REQ\_DETAILS RD ON R.REQS\_ID = RD.REQS\_REQS\_ID WHERE SUBS\_SUBS\_ID = 21274112;

Активируем рандомный сервис, чтобы проявился стандарт и привяжем симку:

BEGIN LITE\_COM\_WORK\_PG.ACTIVATE\_SERVICE(21274112, 114, '956042194', 1, 3,'BIS'); END;

BEGIN LITE\_COM\_WORK\_PG.PERFORM\_LINK\_SUBS\_SIM(29342996,15445,'434081402996794','956042194',10013,1010,21274112, 'BIS',''); END;

Также, добавим клиенту баланс, чтобы можно было вводить начисления (т.к. клиент предоплатны):

SELECT \* FROM BALANCES WHERE CLNT\_CLNT\_ID = 29342996;

UPDATE BALANCES SET BALANCE\_$ = -800000 WHERE BLNC\_ID = 29493967;

После чего проверяем:

SELECT \* FROM BALANCES WHERE CLNT\_CLNT\_ID = 29342996;

**Ввод начислений – разовых, абонентская плата. Ввести платежи**

Создаем начисление:

begin

lite\_upd\_clients.make\_charge(psubs\_id => 21274112,

pprcl\_id => 80);

end;

Проверим:

SELECT \* FROM CHARGES WHERE CLNT\_CLNT\_ID = 29342996;

SELECT \* FROM CHARGES WHERE SUBS\_SUBS\_ID = 21274112;

Произведем расчет АП:

begin

dbms\_output.enable(buffer\_size => 1000000);

bis\_month\_charge.gv\_debug\_flag := true;

bis\_month\_charge.bis\_subs\_charges(subsid => 21274112);

end;

Введем платеж, для этого сначала создадим реестр чеков на сегодняшнюю дату:

Выберем случайную кассу:

SELECT \* FROM KASSA\_HISTORIES WHERE END\_DATE > SYSDATE;

Создание реестра:

declare

reestrId NUMBER(10);

begin

reestrId := iik\_kassa.new\_reestr(p\_kassaid => 444,

p\_date => SYSDATE,

p\_myname => 'YaroslavNX',

p\_auto\_commit\_yn => 'N',

p\_lock\_data\_yn => 'N');

DBMS\_OUTPUT.put\_line(reestrId);

end;

Создался реестр с идентификатором – 145669 (судя по оутпуту)

Проверим:

SELECT \* FROM REESTRES WHERE RSTR\_ID = 145669;

Введем чек:

declare

v\_result VARCHAR2(100);

begin

iik\_kassa.checks(fio => 'Test client',

mycontract => 'test contract series',

numberphone => '956042194',

clientid => 29342996,

myaccount => '89990406443',

payid => 1,

take => 300,

kassaid => 444,

result => v\_result);

DBMS\_OUTPUT.put\_line(v\_result);

end;

Нужная информация получена так:

SELECT \* FROM CLIENT\_HISTORIES WHERE CLNT\_CLNT\_ID = 29342996;

SELECT \* FROM PHONE\_HISTORIES WHERE Subs\_Subs\_Id = 21274112;

SELECT \* FROM NUMBER\_SETS WHERE NSET\_ID = 10013;

Выполним разнесение:

begin

job\_move\_pay.move\_pay();

end;

Проверим:

SELECT \* FROM RECEIPTS WHERE RCPT\_ID = 279624207;

**Вставить звонки в звонковые таблицы calls\_00\_XXXXXX**

CREATE TABLE CALLS\_00\_072021 AS SELECT \* FROM CALLS\_00\_011000;

INSERT INTO CALLS\_00\_072021 (CALL\_ID, IN\_BALANCE\_$, MAIN\_CLNT\_ID) VALUES (1, 350, 29342996);

**Вести корректировки начислений, отрицательную и положительную**

Добавим отрицательную корректировку начисления:

begin

lite\_upd\_clients.add\_adjustment(pclnt\_clnt\_id => 29342996,

pajtp\_ajtp\_id => 3,

pamount\_$ => -30,

pvat\_$ => 10,

pline\_num => 1,

pdetg\_detg\_id => 7, psubs\_id => 21274112);

end;

Проверяем:

SELECT \* FROM ADJUSTMENTS WHERE CLNT\_CLNT\_ID = 29342996;

Добавим положительную корректировку начисления:

begin

lite\_upd\_clients.add\_adjustment(pclnt\_clnt\_id => 29342996,

pajtp\_ajtp\_id => 3,

pamount\_$ => 30,

pvat\_$ => 10,

pline\_num => 1,

pdetg\_detg\_id => 7, psubs\_id => 21274112);

end;

Проверяем:

SELECT \* FROM ADJUSTMENTS WHERE CLNT\_CLNT\_ID = 29342996;

**Ввести корректировки платежей отрицательную и положительную.**

Чтобы сменить корректировку начислений нужно изменить ajtp\_ajtp\_id на другой из:

SELECT \* FROM ADJUST\_TYPES;

Введем положительную корректировку платежей:

begin

lite\_upd\_clients.add\_adjustment(pclnt\_clnt\_id => 29342996,

pajtp\_ajtp\_id => 2,

pamount\_$ => 30,

pvat\_$ => 10,

pline\_num => 1,

pdetg\_detg\_id => 7, psubs\_id => 21274112);

end;

Проверим:

SELECT \* FROM ADJUSTMENTS WHERE CLNT\_CLNT\_ID = 29342996;

Введем отрицательную корректировку платежей:

begin

lite\_upd\_clients.add\_adjustment(pclnt\_clnt\_id => 29342996,

pajtp\_ajtp\_id => 2,

pamount\_$ => -30,

pvat\_$ => 10,

pline\_num => 1,

pdetg\_detg\_id => 7, psubs\_id => 21274112);

end;

Проверяем:

SELECT \* FROM ADJUSTMENTS WHERE CLNT\_CLNT\_ID = 29342996;

**Провести внеочередной биллинг. Процедура pack\_billing.bill\_spec**

Для проведения внеочередного биллинга создадим сначала биллинговый цикл с типом «внеочередной» (id = 2 из таблицы BILL\_CYCLE\_TYPES).

INSERT INTO BILL\_CYCLES (BLCL\_ID, CLST\_CLST\_ID, BCT\_BCT\_ID, CYCLE\_NUM, CYCLE\_DATE)

VALUES (BLCL\_seq.Nextval, 1, 2, 1, DATE '2021-08-20'); *--810*

Свяжем клиента с биллинговым циклом:

INSERT INTO BILL\_CYCLE\_DETAILS (BCD\_ID, BLCL\_BLCL\_ID, CLNT\_CLNT\_ID)

VALUES (BCD\_SEQ.NEXTVAL, 810, 10463625);

Чтобы провести внеочередной биллинг, нужно, чтобы последний ежемесячный биллинговый цикл имел статус «постинг» (id=4 из таблицы CYCLE\_STATUSES), для этого сначала проведем обычный ежемесячный биллинг:

begin

pack\_billing.bill\_month(v\_bill\_date => DATE '2021-08-31', v\_start\_clnt => 10463625, v\_end\_clnt =>10463625);

end;

В результате создастся новая запись в BILLS (не запощенная). Посмотреть ее можно так:

SELECT \* FROM BILLS WHERE BILL\_ID = 210800000300000;

Для проведения внеочередного этого мало, далее нам нужно проверить, чтобы все биллинговые циклы имели статус «Постинг». Найти их можно так (Столбец BLCL\_BLCL\_ID):

SELECT \* FROM BILLS WHERE BILL\_ID = 210800000300000;

Чтобы запостить нужные биллинговые циклы выполним:

begin

bill\_posting.start\_bill\_posting (what => 0, cycleid => 810);

end;

В теории, все ежемесячные биллинговые циклы переведутся в статус 9 (Постинг начат).

Вероятно, должен быть job, выполняющий постинг, но я нашел статью на конфле:

<https://confluence.nexign.com/pages/viewpage.action?pageId=320586180>

Где человек просто апдейтит все нужные нам циклы на 4. Поэтому это сделал и я:

UPDATE BILL\_CYCLES SET CLST\_CLST\_ID = 4 WHERE BLCL\_ID = 210731000000000;

В моем случае был один ежемесячный.

Далее, проведем уже внеочередной биллинг:

begin

PACK\_BILLING.BILL\_SPEC(V\_REQUIRED\_CALC => false, V\_BLCL\_ID => 810);

end;

В результате:

SELECT \* FROM BILL\_CYCLES WHERE BLCL\_ID = 798;

У нас будет статус 3, который означает, что расчет завершен. Но финальный статус все же 4, добиться его мне не удалось, процедура для постинга выше не обрабатывает его :С

**Проверить учтенность объектов в биллинге, проверить формирование деталей.**

Проверить учтенность не получится в связи с отсутствием статуса «Постинг», я полагаю:

SELECT \* FROM CHARGES WHERE BILL\_BILL\_ID = 210800000300000;

SELECT \* FROM ADJUSTMENTS WHERE BILL\_BILL\_ID = 210800000300000;

SELECT \* FROM PAYMENTS WHERE BILL\_BILL\_ID = 210800000300000;

SELECT \* FROM BILL\_DETAILS WHERE BILL\_BILL\_ID = 210800000300000;

**Построить список настроечных параметров, которые определяют глубину поиска неучтенных объектов.**

<https://confluence.nexign.com/pages/viewpage.action?pageId=44107349>