

Outpatient Billing Interface

1. PL/SQL package Header

create or replace package wxgh.ps_appreg_wx is

-- Author: Zhigang Zhao

-- Created: 2019/12/4 18:17:10

-- Purpose: Outpatient Billing Interface

-- version:11.2.3

/*functional description: outpatients can handle registration, payment, and refunds directly through their mobile devices . It works by calling a backend PL/SQL package. This package has around 40 functions and procedures, making up over 20,000 lines of code. */

-- 3.1 Patient Information Binding 003

procedure patInfoBind003(as_xmlParm in varchar2,

aCur_temp out sys_refcursor,

on_return out number,

on_message out varchar2);

-- 3.2 Patient Information Unbinding 001

procedure patInfoUnbind001(as_xmlParm in varchar2,

aCur_temp out sys_refcursor,

on_return out number,

on_message out varchar2);

-- 3.3 Patient Information Update 001

procedure patInfoUpd001(as_xmlParm in varchar2,

aCur_temp out sys_refcursor,

on_return out number,

on_message out varchar2);

-- 4.1.1 Department Registration Information Query Interface 001

```
procedure getDeptRegInfo001(hospitalId in varchar2, -- (Mandatory) Hospital ID

    deptId in varchar2, -- Department code, if empty, information of all
departments is obtained

    beginDate in varchar2, -- (Mandatory) Registration start date, format: YYYY-
MM-DD

    endDate in varchar2, -- (Mandatory) Registration end date, format: YYYY-MM-
DD

    aCur_temp out sys_refcursor);
```

-- 4.1.2 Doctor Registration Shift Information Query Interface 001

```
procedure getDrRegScheduleInfo001(hospitalId in varchar2, -- (Mandatory) Hospital ID

    deptId in varchar2, -- (Mandatory) Department code (primary)

    doctorId in varchar2, -- Doctor code

    beginDate in varchar2, -- (Mandatory) Source start date, format: YYYY-MM-
DD

    endDate in varchar2, -- (Mandatory) Source end date, format: YYYY-MM-
DD

    aCur_temp out sys_refcursor);
```

-- 4.1.3 Doctor Registration Time-slot Information Query Interface 001

```
procedure getDrRegTimeInfo001(hospitalId in varchar2, -- (Mandatory) Hospital ID

    deptId in varchar2, -- (Mandatory) Department code (primary)

    doctorName in varchar2, -- Doctor

    scheduleType in varchar2, -- Period

    regDate in varchar2, -- (Mandatory) Scheduling date, format: YYYY-MM-DD

    aCur_temp out sys_refcursor);
```

-- 4.1.4 Suspension of Medical Service Notification Query Interface 001

```
procedure getStopShift001(hospitalId in varchar2, -- (Mandatory) Hospital ID
```

queryBeginTime in varchar2, -- (Mandatory) Query start time, format: YYYY-MM-DD hh24:mi:ss

queryEndTime in varchar2, -- (Mandatory) Query end time, format: YYYY-MM-DD hh24:mi:ss

aCur_temp out sys_refcursor);

-- 4.2.1 Appointment Registration Order Generation Interface 001

procedure addPreRegOrder001(axmlparm in varchar2,

aCur_temp out sys_refcursor,

on_return out number, -- Result code

on_message out varchar2 -- Result description

);

-- 4.2.2 Appointment Registration Order Cancellation Interface 001

procedure cancelPreRegOrder001(axmlparm in varchar2,

aCur_temp out sys_refcursor,

on_return out number, -- Result code

on_message out varchar2 -- Result description

);

-- 4.2.3 Appointment Registration Order Payment Interface 001

procedure payPreRegOrder001(axmlparm in varchar2,

aCur_temp out sys_refcursor,

on_return out number, -- Result code

on_message out varchar2 -- Result description

);

-- 4.2.4 Appointment Registration Order Refund Interface 001

procedure refundPreReg001(axmlparm in varchar2,

aCur_temp out sys_refcursor,

```

on_return out number, -- Result code
on_message out varchar2 -- Result description
);

```

--4.2.5 Refund Rewriting

```

procedure returnRefundNo001(axmlparm in varchar2,
    aCur_temp out sys_refcursor,
    on_return out number, -- Result code
    on_message out varchar2 -- Result description
);

```

-- 4.3.1 Same-day Registration Source Locking Interface 001

```

procedure lockRegSeqToday001(axmlparm in varchar2,
    aCur_temp out sys_refcursor,
    on_return out number, -- Result code
    on_message out varchar2 -- Result description
);

```

-- 4.3.2 Same-day Registration Source Unlocking Interface 001

```

procedure unlockRegSeqToday001(hospitalId in varchar2, -- (Mandatory) Hospital ID
    isZJ in varchar2, -- Whether it is an additional number in the clinic (Y-Yes,
N-No, default is N)
    lockId in varchar2, -- (Mandatory) Source lock ID (generated by the channel)
    on_return out number, -- Result code
    on_message out varchar2 -- Result description
);

```

-- 4.3.3 Same-day Registration Interface 001

```

procedure addRegOrder001(axmlparm in varchar2,
    aCur_temp out sys_refcursor,

```

```

on_return out number, -- Result code
on_message out varchar2 -- Result description
);

```

-- 5.1.1 Unpaid Medical Visit Record Query Interface 001

```

procedure getNoPayVisitInfo001(hospitalId in varchar2, -- Hospital ID

    patId    in varchar2, -- Patient main index id
    opPatId  in varchar2, -- Outpatient patient id
    patMedType in varchar2, -- Patient medical card type
    patMedNo  in varchar2, -- Patient medical card number
    beginDate in varchar2, -- Query start date, format: YYYY-MM-DD
    endDate   in varchar2, -- Query end date, format: YYYY-MM-DD
    aCur_temp out sys_refcursor);

```

-- 5.1.2 Unpaid Detail Record Query Interface 001

```

procedure getNoPayDetailInfo001(hospitalId in varchar2, -- (Mandatory) Hospital ID

    patId    in varchar2, -- Patient main index id
    opPatId  in varchar2, -- Outpatient patient id
    visitId  in varchar2, -- (Mandatory) HIS visit serial number
    deptId   in varchar2,
    doctorId in varchar2,
    aCur_temp out sys_refcursor);

```

-- 5.1.3 Unpaid Record Payment Interface 001

```

procedure payChgOrder001(as_xmlParm in varchar2,

    aCur_temp out sys_refcursor,

    on_return out number, -- Result code
    on_message out varchar2 -- Result description
);

```

-- 5.1.4 Paid Record Query Interface 001

```
procedure getPayReceiptInfo001(patMedNo in varchar2, -- Patient medical card number  
beginDate in varchar2, -- Query start date, format: YYYY-MM-DD  
endDate in varchar2, -- Query end date, format: YYYY-MM-DD  
aCur_temp out sys_refcursor);
```

-- 5.1.5 Paid Record Detail Query Interface 001

```
procedure getPayDetailInfo001(visitId in varchar2, -- (Mandatory) HIS visit serial number  
aCur_temp out sys_refcursor);
```

-- 5.2.1 Exceptional Order Refund Application 001

```
procedure applyRefund001(hospitalId in varchar2, -- (Mandatory) Hospital ID  
busType in varchar2, -- (Mandatory) Business type (YYGH-Appointment  
registration payment, DBGH-On-duty registration payment, MZJF-Outpatient payment, ZYYJ-  
Inpatient deposit payment)  
orderPayId in varchar2, -- WeChat, Alipay merchant generated transaction serial  
number  
orderId in varchar2, -- Third-party platform generated order ID  
payTime in varchar2, -- Transaction time  
refundAmount in varchar2, -- Refund amount  
aCur_temp out sys_refcursor);
```

-- 6.1.1 Laboratory Report List Query 001

```
procedure getLisReportList001(hospitalId in varchar2, -- (Mandatory) Hospital ID  
patMedNo in varchar2, -- Patient medical card number  
beginDate in varchar2, -- Query start date, format: YYYY-MM-DD  
endDate in varchar2, -- Query end date, format: YYYY-MM-DD  
aCur_temp out sys_refcursor);
```

-- 6.1.2 Laboratory Report Detail Query 002

```
procedure getLisReportDetail002(hospitalId in varchar2, -- (Mandatory) Hospital ID
                                reportId in varchar2, -- Laboratory report id
                                aCur_temp out sys_refcursor);
```

-- 6.2.3 Examination Report List Query 002

```
procedure getPacsReportList002(hospitalId in varchar2, -- (Mandatory) Hospital ID
                                patMedNo in varchar2, -- Patient medical card number
                                beginDate in varchar2, -- Query start date, format: YYYY-MM-DD
                                endDate in varchar2, -- Query end date, format: YYYY-MM-DD
                                aCur_temp out sys_refcursor);
```

-- Patient Record Creation 002

```
procedure newCard002(as_xmlParm in varchar2,
                    aCur_temp out sys_refcursor,
                    on_return out number,
                    on_message out varchar2);
```

-- Outpatient Queuing Information Query Interface 001

```
procedure getOpQueuingInfo001(patMedNo in varchar2,
                               aCur_temp out sys_refcursor);
```

-- Inpatient Basic Information Query Interface 001

```
procedure getIpPatientInfo002(as_xmlParm in varchar2,
                               aCur_temp out sys_refcursor,
                               on_return out number,
                               on_message out varchar2);
```

-- Inpatient Deposit Detail Query Interface 001

```
procedure getIpDepositDetail002(as_xmlParm in varchar2,  
                                aCur_temp out sys_refcursor,  
                                on_return out number,  
                                on_message out varchar2);
```

-- Current Inpatient Expense Query Interface 002

```
procedure getIpChgInfo001(as_xmlParm in varchar2,  
                           aCur_temp out sys_refcursor,  
                           on_return out number,  
                           on_message out varchar2);
```

-- Inpatient Daily Bill Expense Category Summary Query Interface 001

```
procedure getIpDayExpenseItemTotal002(as_xmlParm in varchar2,  
                                       aCur_temp out sys_refcursor,  
                                       on_return out number,  
                                       on_message out varchar2);
```

-- Inpatient Deposit Payment Interface 001

```
procedure payIpDeposit002(as_xmlParm in varchar2,  
                           aCur_temp out sys_refcursor,  
                           on_return out number,  
                           on_message out varchar2);
```

-- Hospital Insurance Drug Comparison Interface

```
procedure getPriceVsInsurDict(pageSize in number, -- Number of rows per page  
                               currentPage in number, -- Current page number  
                               aCur_temp out sys_refcursor);
```


-- Query for registration records without a set scan URL

```
procedure getNoSetQrCode(beginDate in varchar2,  
                        endDate   in varchar2,  
                        aCur_temp out sys_refcursor);
```

-- Set scan URL for registration records by Visit ID

```
procedure setQrCodeByVisitId(as_xmlParm in varchar2,  
                             aCur_temp  out sys_refcursor,  
                             on_return  out number,  
                             on_message out varchar2);
```

-- National Electronic Health Card Binding

```
procedure countryEleHCNoBind001(as_xmlParm in varchar2,  
                                aCur_temp  out sys_refcursor,  
                                on_return  out number,  
                                on_message out varchar2);
```

-- National Electronic Health Card Query

```
procedure countryEleHCNoInfo001(patIdCard in varchar2,  
                                patMedNo  in varchar2,  
                                eleHCNo   in varchar2,  
                                aCur_temp out sys_refcursor);
```

-- Nucleic Acid Testing Item Query

```
procedure getHsjcItem001(as_xmlParm in varchar2,  
                         aCur_temp  out sys_refcursor,  
                         on_return  out number,  
                         on_message out varchar2);
```

-- Nucleic Acid Testing Self-Registration

```
procedure selfRegHsjc001(as_xmlParm in varchar2,  
    aCur_temp out sys_refcursor,  
    on_return out number,  
    on_message out varchar2);
```

-- Retrieve Tag Value from XML String

```
function of_get_xmlval(as_xml_string in varchar2,  
    as_xml_mark in varchar2) return varchar2;
```

-- Patient Registration Rules

- 1. Males are not allowed to register for gynecology and obstetrics.
- 2. Patients over 14 years old (inclusive) are not allowed to register for pediatrics.

```
clinicLabel in varchar2,  
timeDesc in varchar2,  
regDate in date) return varchar2;
```

-- 5.1.8 Query for Unpaid Medical Insurance Prescription Information

```
procedure getWaitPayMedInsInfo001(hospitalId in varchar2, -- Hospital ID  
    visitId in varchar2, -- Visit serial number  
    aCur_temp out sys_refcursor);
```

-- xml Parsing Functions

```
procedure PRO_OUTPBILL_TJ_INSERT(v_xml_string in varchar2,  
    aCur_temp out sys_refcursor,  
    on_return OUT number,  
    on_message OUT varchar2);
```

```
procedure PRO_OUTPBILL_INSUR_upload(v_xml_string in clob,  
    aCur_temp out sys_refcursor,  
    on_return OUT number,
```

```
on_message OUT varchar2);
```

-- Physical Examination Refund Interface, applicable for nucleic acid testing
appointment refund

```
procedure pro_outpbill_Tj_Ref(v_xml_string in varchar2,  
    aCur_temp out sys_refcursor,  
    on_return out number, -- Result code  
    on_message out varchar2 -- Result description  
);
```

-- New Test 1

```
procedure PRO_OUTPBILL_TJ_INSERT1(v_xml_string in varchar2,  
    aCur_temp out sys_refcursor,  
    on_return OUT number,  
    on_message OUT varchar2);
```

```
end ps_appreg_wx;
```

create or replace package body wxgh.ps_appreg_wx is

2. PL/SQL package body

```
-- 3.1 Patient information binding 003
procedure patInfoBind003(as_xmlParm in varchar2,
                        aCur_temp out sys_refcursor,
                        on_return out number,
                        on_message out varchar2) is
/* 输出结果
* 字段      描述
* -----
patIdCard   身份证号
patId       患者主索引 id
opPatNo     门诊病人号
ipPatNo     住院病人号
patMedNo    就诊卡号
homeAddress 患者住址
*/
patMedNo  varchar2(20);
patName   varchar2(20);
patIdCard varchar2(18);
patMobile varchar2(19);
guardMobile varchar2(16);
n_record  number;

v_patId varchar2(20);
v_idNo  varchar2(18);
v_name  varchar2(20);
begin
```

```

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patIdCard') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-1',
    'patInfoBind003',
    as_xmlParm,
    '进入 patInfoBind003-01' ||
    of_get_xmlval(as_xmlParm, 'patName'),
    null,
    null);

```

```

on_return := 0;
on_message := '绑定成功';
patMedNo := of_get_xmlval(as_xmlParm, 'patMedNo');
patName := of_get_xmlval(as_xmlParm, 'patName');
patIdCard := of_get_xmlval(as_xmlParm, 'patIdCard');
patMobile := of_get_xmlval(as_xmlParm, 'patMobile');
guardMobile := of_get_xmlval(as_xmlParm, 'guardMobile');

```

```

if patMedNo is null and patIdCard is null then
    on_return := -1;
    on_message := '参数 patMedNo 和 patIdCard 不能同时为空';
    goto end_program;
end if;

```

```

if patMedNo is not null then
    if length(patMedNo) <> 9 then
        on_return := -1;
        on_message := '卡号' || patMedNo || '长度不正确';
        goto end_program;
    end if;
end if;

```

```

    end if;
end if;

/*if patIdCard is not null then
if length(patIdCard) <> 18 or length(patIdCard) <> 15 then
    on_return := -1;
    on_message := '身份证' || patIdCard || '长度不正确';
    goto end_program;
end if;
end if;*/

if patName is null then
    on_return := -1;
    on_message := '参数 patName 为空';
    goto end_program;
end if;

/*if patMobile is null and guardMobile is null then
    on_return := -1;
    on_message := '参数 patMobile 和 guardMobile 不能同时为空';
    goto end_program;
end if;*/

if patMedNo is not null then
    select count(1)
        into n_record
        from MEDREC.Pat_Master_Index b
        where 1 = 1
            and b.patient_id = patMedNo
            AND B.PATIENT_ID NOT LIKE 'T%'

```

```

        and b.name = patName;
        -- and b.id_no is not null;
else
    select count(1)
        into n_record
    from MEDREC.Pat_Master_Index b
    where 1 = 1
        and b.name = patName
        AND B.PATIENT_ID NOT LIKE 'T%'
        and b.id_no = patIdCard;
end if;

if n_record < 1 then
    on_return := -1;
    on_message := '患者信息匹配失败，请至挂号窗口进行确认';
    goto end_program;
end if;

if patMedNo is not null then
    select b.id_no, b.Patient_ID, b.name
        into v_idNo, v_patId, v_name
    from MEDREC.Pat_Master_Index b
    where 1 = 1
        and b.patient_id = patMedNo
        AND B.PATIENT_ID NOT LIKE 'T%'
        and b.name = patName;
        --and b.id_no is not null;
else
    select b.id_no, b.Patient_ID, b.name

```

```

into v_idNo, v_patId, v_name
from (select case
        when t.create_date is null then
            to_date('1900-01-01', 'yyyy-mm-dd')
        else
            t.create_date
        end,
        t.name,
        t.id_no,
        t.patient_id,
        t.inp_no,
        t.mailing_address
    from pat_master_index t
    where id_no = patIdCard
        and name = patName
        and length(t.patient_id) >= 9
        AND T.PATIENT_ID NOT LIKE 'T%'
    order by t.Patient_ID desc) b
where rownum = 1;
end if;

begin
update MEDREC.Pat_Master_Index t
    set t.issu_flag = '2' --建档绑定-1， 绑定历史记录-2， 解绑-3
where t.patient_id = v_patId
    and t.name = v_name
    and t.id_no = v_idNo;
exception
when others then

```



```

ROLLBACK;

on_return := -1;

on_message := '更新患者主索引记录[' || v_patId || ']出错' || SQLERRM;

goto end_program;

end;

<<end_program>>

if on_return = 0 then

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patIdCard') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
    'patInfoBind003',
    as_xmlParm,
    '结束 patInfoBind003-02' ||
    of_get_xmlval(as_xmlParm, 'patName'),
    on_return,
    null);

open aCur_temp for
select to_char(on_return) "resultCode",
    on_message "resultDesc",
    b.id_no as "patIdCard", -- 身份证号
    b.Patient_ID as "patId", -- 患者主索引 id
    b.Patient_ID as "opPatNo", -- 门诊病人号
    b.inp_no as "ipPatNo", -- 住院病人号
    b.Patient_ID as "patMedNo", -- 就诊卡号
    b.mailing_address as "homeAddress" -- 患者住址

```

```

from MEDREC.Pat_Master_Index b

where 1 = 1

and b.patient_id = v_patId
and b.name = v_name
and b.id_no = v_idNo;

else

insert_wx_webservice_err_logs('patInfoBind003' ||

to_char(systimestamp,

'-yyyymmddhh24missxff3'),

'patInfoBind003',

as_xmlParm,

",

on_message);

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patIdCard') || '-' ||

to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',

'patInfoBind003',

as_xmlParm,

'[错]结束 patInfoBind003-02' ||

of_get_xmlval(as_xmlParm, 'patName'),

on_return,

on_message);

open aCur_temp for

select to_char(on_return) as "resultCode",

on_message as "resultDesc",

" as "patIdCard", -- 身份证号

```

```

        " as "patId", -- 患者主索引 id
        " as "opPatNo", -- 门诊病人号
        " as "ipPatNo", -- 住院病人号
        " as "patMedNo", -- 就诊卡号
        " as "homeAddress" -- 患者住址
    from dual;
end if;
end;

-- 3.2 Patient Information Unbinding 001 患者信息解绑 001
procedure patInfoUnbind001(as_xmlParm in varchar2,
    aCur_temp out sys_refcursor,
    on_return out number,
    on_message out varchar2) is

/* 输出结果
* 字段      描述
* -----
patIdCard    身份证号
patId        患者主索引 id
opPatNo      门诊病人号
ipPatNo      住院病人号
patMedNo     就诊卡号
homeAddress  患者住址
*/
patMedNo varchar2(20);
patName varchar2(20);
n_record number;

```

begin

```
insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patMedNo') || '-' ||  
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-1',  
    'patInfoUnbind001',  
    as_xmlParm,  
    '进入 patInfoUnbind001-01' ||  
    of_get_xmlval(as_xmlParm, 'patMedNo'),  
    null,  
    null);
```

on_return := 0;

on_message := '解绑成功';

patMedNo := of_get_xmlval(as_xmlParm, 'patMedNo');

patName := of_get_xmlval(as_xmlParm, 'patName');

if patMedNo is null then

on_return := -1;

on_message := '参数 patMedNo 不允许为空';

goto end_program;

end if;

select count(1)

into n_record

from MEDREC.Pat_Master_Index b

where 1 = 1

and b.patient_id = patMedNo;

```

if n_record < 1 then
    on_return := -1;
    on_message := '未查询到诊疗卡号记录';
    goto end_program;
end if;

begin
    update MEDREC.Pat_Master_Index t
        set t.issu_flag = '3'
        where t.patient_id = patMedNo;
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '更新患者主索引记录[' || patMedNo || ']出错' || SQLERRM;
        goto end_program;
end;

<<end_program>>

if on_return = 0 then

    insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patMedNo') || '-' ||
        to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
        'patInfoUnbind001',
        as_xmlParm,
        '结束 patInfoUnbind001-02' ||
        of_get_xmlval(as_xmlParm, 'patMedNo'),
        on_return,

```

```

null);

open aCur_temp for
select to_char(on_return) "resultCode", on_message "resultDesc"
from dual;

else

insert_wx_webservice_err_logs('patInfoUnbind001' ||

to_char(systimestamp,

'-yyyymmddhh24missxff3'),

'patInfoUnbind001',

as_xmlParm,

",

on_message);

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patMedNo') || '-' ||

to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',

'patInfoUnbind001',

as_xmlParm,

'[错]结束 patInfoUnbind001-02' ||

of_get_xmlval(as_xmlParm, 'patMedNo'),

on_return,

on_message);

open aCur_temp for
select to_char(on_return) as "resultCode",
on_message as "resultDesc"
from dual;

```

```

end if;

end;

-- 3.3 Patient Information Update 001 患者信息修改 001
procedure patInfoUpd001(as_xmlParm in varchar2,
                        aCur_temp out sys_refcursor,
                        on_return out number,
                        on_message out varchar2) is
/*
* 字段      描述
* -----
patName      in varchar2, -- (必传) 患者姓名
patIdCard    in varchar2, -- (必传) 病人身份证 (就诊卡号和身份证两个不能同时为
空)
patMedType   in varchar2, -- 病人就诊卡类型
patMedNo     in varchar2, -- 病人就诊卡号
opPatId      in varchar2, -- 门诊病人 id
ipPatId      in varchar2, -- 住院病人 id
oldPatMobile in varchar2, -- (必传) 原电话号码
patMobile    in varchar2, -- (必传) 新电话号码
on_return    out number, -- 结果编码
on_message   out varchar2 -- 结果描述
*/

patMobile varchar2(19);
patMedNo  varchar2(20);
begin

```

```

on_return := 0;
on_message := '成功';
patMedNo := of_get_xmlval(as_xmlParm, 'patMedNo');
patMobile := of_get_xmlval(as_xmlParm, 'patMobile');
begin
    update pat_master_index
        set next_of_kin_phone = patMobile ----修改联系人电话
        where patient_id = patMedNo;
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '修改患者信息出现错误' || SQLERRM;
        goto end_program;
end;

<<end_program>>

open aCur_temp for
    select to_char(on_return) as "resultCode", on_message as "resultDesc"
    from dual;

end;

```

-- 4.1.1 Department Registration Information Query Interface 001 科室挂号信息查询接口 001

procedure getDeptRegInfo001(hospitalId in varchar2, -- (必传) 医院 ID

deptId in varchar2, -- 科室代码, 如果为空则获取所有科室的信息

beginDate in varchar2, -- (必传) 号源开始日期, 格式: YYYY-MM-DD


```

        endDate in varchar2, -- (必传) 号源结束日期, 格式: YYYY-MM-DD
        aCur_temp out sys_refcursor) is
/* 输出结果
* 字段      描述
* -----
branchId  分院代码
branchName 分院名称
deptClassId 科室分类 id
deptClassName 科室分类名称
deptId    科室代码
deptName  科室名称
deptDesc  科室简介
regMode   挂号模式 (DEPT-直接挂号到科室, DOCTOR-挂号到医生)
*/
v_beginDate varchar2(20);
v_endDate   varchar2(20);
begin

v_beginDate := beginDate;
v_endDate   := endDate;

if deptId = '' or deptId is null then
    open aCur_temp for
        select "branchId",
               "branchName",
               "deptClassId",
               "deptClassName",
               "deptId",

```

```

"deptName",
"deptDesc",
"regMode",
"clinicposition"
from (select distinct (select mr_hospital_code
                        from COMM.Hospital_Config) as "branchId", -- 分院代码
      (select hospital from COMM.Hospital_Config) as "branchName", -- 分院名称
      b.clinic_dept as "deptClassId", -- 科室分类 id
      d.dept_name as "deptClassName", -- 科室分类名称
      b.clinic_dept as "deptId", -- 科室代码
      d.dept_name || '(' ||
      substr(b.clinic_type,
              1,
              decode(instr(b.clinic_type, '号', 1),
                      0,
                      length(b.clinic_type),
                      instr(b.clinic_type, '号', 1) - 1)) || ')' as "deptName", -- 科室名称
      " as "deptDesc", -- 科室简介
      'DEPT' as "regMode", -- 挂号模式（DEPT-直接挂号到科室，DOCTOR-挂号
到医生）
      d.input_code as "pinyinCode",
      b.clinic_position as "clinicposition"
from OUTPADM.CLINIC_FOR_REGIST a,
      OUTPADM.CLINIC_INDEX    b,
      COMM.dept_dict          d
where 1 = 1
      and a.stop_flag = 2
      and a.clinic_label = b.clinic_label
      and d.dept_code = b.clinic_dept

```

```

and a.CLINIC_DATE >= TO_DATE(v_beginDate, 'YYYY-MM-DD')
and a.CLINIC_DATE <= TO_DATE(v_endDate, 'YYYY-MM-DD')
and a.clinic_label not like '%急诊%'
and a.clinic_label not like '%体检%'
-- and a.clinic_label not like '%新院%'
and d.dept_name not in ('重症医学一区普通门诊')
and (case
    when beginDate = endDate and
        to_date(beginDate, 'yyyy-mm-dd') =
            trunc(sysdate) then
        b.sh_registration_flag
    when to_date(beginDate, 'yyyy-mm-dd') <=
        to_date(endDate, 'yyyy-mm-dd') and
        to_date(beginDate, 'yyyy-mm-dd') >
            trunc(sysdate) then
        b.sh_appointment_flag
    end) = 1

order by d.input_code);

elsif deptId <> '' or deptId is not null then
    open aCur_temp for
        select distinct (select mr_hospital_code from COMM.Hospital_Config) as "branchId", -- 分
院代码
        (select hospital from COMM.Hospital_Config) as "branchName", -- 分院名称
        b.clinic_dept as "deptClassId", -- 科室分类 id
        d.dept_name as "deptClassName", -- 科室分类名称
        b.clinic_dept as "deptId", -- 科室代码
        d.dept_name || '(' ||

```

```

        substr(b.clinic_type,
            1,
            decode(instr(b.clinic_type, '号', 1),
                0,
                length(b.clinic_type),
                instr(b.clinic_type, '号', 1) - 1)) || ' ' as "deptName", -- 科室名称
        " as "deptDesc", -- 科室简介
        'DEPT' as "regMode", -- 挂号模式（DEPT-直接挂号到科室，DOCTOR-挂号到医
生）

        b.clinic_position as "clinicposition"
from OUTPADM.CLINIC_FOR_REGIST a,
    OUTPADM.CLINIC_INDEX    b,
    COMM.dept_dict          d
where 1 = 1
and a.stop_flag = 2
and a.clinic_label = b.clinic_label
and d.dept_code = b.clinic_dept
and a.CLINIC_DATE >= TO_DATE(v_beginDate, 'YYYY-MM-DD')
and a.CLINIC_DATE <= TO_DATE(v_endDate, 'YYYY-MM-DD')
and b.clinic_dept = deptId
and a.clinic_label not like '%急诊%'
and a.clinic_label not like '%体检%'
    -- and a.clinic_label not like '%新院%'
and d.dept_name not in ('重症医学一区普通门诊')
and (case
    when beginDate = endDate and
        to_date(beginDate, 'yyyy-mm-dd') = trunc(sysdate) then
        b.sh_registration_flag
    when to_date(beginDate, 'yyyy-mm-dd') <=

```

```

        to_date(endDate, 'yyyy-mm-dd') and
        to_date(beginDate, 'yyyy-mm-dd') > trunc(sysdate) then
        b.sh_appointment_flag
    end) = 1;
end if;

end;

-- 4.1.2. Doctor Registration Shift Information Query Interface 001 procedure
getDrRegScheduleInfo001(hospitalId in varchar2, -- (必传) 医院 ID
    deptId in varchar2, -- (必传) 科室代码 (一级)
    doctorId in varchar2, -- 医生代码
    beginDate in varchar2, -- (必传) 号源开始日期, 格式: YYYY-MM-DD
    endDate in varchar2, -- (必传) 号源结束日期, 格式: YYYY-MM-DD
    aCur_temp out sys_refcursor) is
/* 输出结果
* 字段      描述
* -----
hospitalId  医院 ID
doctorId    医生 ID
doctorName  医生名称
branchName  分院名称
deptId      科室 id (二级)
deptName    科室名称 (二级)
doctorTitle 医生职称
doctorRemark 医生备注, 如名医、退休
doctorGender 医生性别, 1: 男, 2 女, 3:未填
betterFor   擅长

```

urlPic 头像地址

scheduleNo 排班记录号，表示当前排班记录的唯一标识

regDate 出诊日期，格式：YYYY-MM-DD

regWeekDay 出诊日期对应的中文星期几，如星期五

scheduleType 班别 A:上午 P:下午 N:晚上

regStatus 出诊状态（STOP-停诊，OPEN-出诊，CLOSE-暂未开放）

regTotalCount 该时段可预约的总号源数

scheduleNo 排班标志，标识该排班的标志，代表了该排班

regleaveCount 该时段剩余号源数

regFee 挂号费(单位“分”)

treatFee 诊疗费(单位“分”)

isTimeReg 是否挂号到具体时段（Y - 是，N - 否）

*/

d_now date;

v_now varchar2(20);

v_sch_flag varchar2(20);

v_sch_flag2 varchar2(20);

begin

select sysdate into d_now from dual;

v_now := to_char(d_now, 'yyyy-mm-dd');

if beginDate = v_now then

--当天

/*if d_now < to_date(v_now || ' 12:00:00', 'yyyy-mm-dd hh24:mi:ss') then

v_sch_flag := '上午';

elsif d_now >= to_date(v_now || ' 14:00:00', 'yyyy-mm-dd hh24:mi:ss') and

```

        d_now < to_date(v_now || ' 18:00:00', 'yyyy-mm-dd hh24:mi:ss') then
        v_sch_flag := '下午';
    end if;*/

    v_sch_flag := '上午';
    v_sch_flag2 := '下午';

    elsif to_date(beginDate, 'yyyy-mm-dd') > trunc(sysdate) then
        --非当天

        v_sch_flag := '上午';
        v_sch_flag2 := '下午';
    end if;

    -- TODO SQL 逻辑

    if deptId <> '' or deptId is not null then
        open aCur_temp for
            select (select mr_hospital_code from COMM.Hospital_Config) as "hospitalId", -- 医院 ID
                   (select hospital from COMM.Hospital_Config) as "branchName", -- 分院名称
                   b.clinic_dept as "deptId", -- 科室 id (二级)
                   d.dept_name || '(' ||
                   substr(b.clinic_type,
                        1,
                        decode(instr(b.clinic_type, '号', 1),
                                0,
                                length(b.clinic_type),
                                instr(b.clinic_type, '号', 1) - 1)) || ')' as "deptName", -- 科室名称 (二级)
                   " as "doctorId", -- 医生 ID
                   b.clinic_label as "doctorName", -- 医生名称
                   case b.doctor_title

```

```

when '1' then
    '主任医师'

when '2' then
    '副主任医师'

when '3' then
    '主治医师'

end as "doctorTitle", -- 医生职称

" as "doctorRemark", -- 医生备注，如名医、退休

" as "doctorGender", -- 医生性别，1: 男，2 女，3:未填

(select f.better_for from staff_dict f where f.user_name=b.doctor and rownum=1) as
"betterFor", -- 擅长

" as "urlPic", -- 头像地址

" as "scheduleNo", -- 排班记录号，表示当前排班记录的唯一标识

to_char(a.clinic_date, 'yyyy-mm-dd') as "regDate", -- 出诊日期，格式: YYYY-MM-DD

TO_CHAR(CLINIC_DATE, 'DAY') as "regWeekDay", -- 出诊日期对应的中文星期几，如
星期五

e.regtime as "scheduleType", -- 班别 A:上午 P:下午 N:晚上

'OPEN' as "regStatus", -- 出诊状态（STOP-停诊，OPEN-出诊，CLOSE-暂未开放）

sum(case

    when a.clinic_date = trunc(sysdate) then

        a.registration_limits

    when a.clinic_date > trunc(sysdate) then

        a.appointment_limits

end) as "regTotalCount", -- 该时段可预约的总号源数

sum(case

    when a.clinic_date = trunc(sysdate) then

        case

            when (a.registration_limits - a.current_no + 1) < 0 then

                0

```



```

        else
            a.registration_limits - a.current_no + 1
        end
    when a.clinic_date > trunc(sysdate) then
        case
            when (a.appointment_limits - a.current_no + 1) < 0 then
                0
            else
                a.appointment_limits - a.current_no + 1
            end
        end
    end) as "regleaveCount", -- 该时段剩余号源数

case
when a.clinic_date = trunc(sysdate) then
    (select sum(price_list.price) * 100
     from COMM.price_list, COMM.CLINIC_TYPE_SETTING
     where price_list.item_code =
            CLINIC_TYPE_SETTING.Price_Item
            and CLINIC_TYPE_SETTING.clinic_type = b.clinic_type
            and CLINIC_TYPE_SETTING.Charge_Item = '挂号费'
            and price_list.stop_date is null)
when a.clinic_date > trunc(sysdate) then
    (select sum(price_list.price) * 100
     from COMM.price_list
     where price_list.item_name = '计算机预约挂号'
            and price_list.stop_date is null)
end as "regFee", -- 挂号费(单位 “分” )

(select sum(price_list.price) * 100

```

```

from COMM.price_list, COMM.CLINIC_TYPE_SETTING
where price_list.item_code = CLINIC_TYPE_SETTING.Price_Item
and CLINIC_TYPE_SETTING.clinic_type = b.clinic_type
and CLINIC_TYPE_SETTING.Charge_Item = '诊疗费'
and price_list.stop_date is null) as "treatFee", -- 诊疗费(单位“分”)
'Y' as "isTimeReg" --, -- 是否挂号到具体时段（Y - 是，N - 否）
-- to_char(a.time_start, 'hh24:mi') star_time, --开始时间
-- to_char(a.time_end, 'hh24:mi') end_time --结束时间
from OUTPADM.CLINIC_FOR_REGIST a,
    OUTPADM.CLINIC_INDEX    b,
    COMM.dept_dict          d,
    outpbill.period_vs_regtime e
where 1 = 1
and a.clinic_label = b.clinic_label
and b.clinic_dept = d.dept_code
and a.time_desc = e.period_time
and a.stop_flag = '2'
and a.CLINIC_DATE >= TO_DATE(beginDate, 'YYYY-MM-DD')
and a.CLINIC_DATE <= TO_DATE(endDate, 'YYYY-MM-DD')
and b.clinic_dept = deptId
and a.clinic_label not like '%急诊%'
and a.clinic_label not like '%体检%'
--and (case
--    when a.clinic_date = trunc(sysdate) then
--        a.registration_limits - a.registration_num
--    when a.clinic_date > trunc(sysdate) then
--        a.appointment_limits - a.appointment_num
--    end) > 0

```

```

and (case
    when a.CLINIC_DATE = trunc(sysdate) then
        b.sh_registration_flag
    when a.CLINIC_DATE > trunc(sysdate) then
        b.sh_appointment_flag
    end) = 1
and e.regtime in (v_sch_flag, v_sch_flag2)
and a.time_end >= sysdate
group by b.clinic_dept,
        d.dept_name,
        b.clinic_type,
        b.clinic_label,
        b.doctor_title,
        a.clinic_date,
        e.regtime,
        b.doctor
order by a.clinic_date, d.dept_name, b.doctor_title, e.regtime;

```

elsif deptId = '' or deptId is null then

open aCur_temp for

```

select (select mr_hospital_code from COMM.Hospital_Config) as "hospitalId", -- 医院 ID
       (select hospital from COMM.Hospital_Config) as "branchName", -- 分院名称
       b.clinic_dept as "deptId", -- 科室 id (二级)
       d.dept_name || '(' ||
       substr(b.clinic_type,
            1,
            decode(instr(b.clinic_type, '号', 1),
                0,
                length(b.clinic_type),

```

```

instr(b.clinic_type, '号', 1) - 1)) || ' ' as "deptName", -- 科室名称（二级）
" as "doctorId", -- 医生 ID
b.clinic_label as "doctorName", -- 医生名称
case b.doctor_title
  when '1' then
    '主任医师'
  when '2' then
    '副主任医师'
  when '3' then
    '主治医师'
end as "doctorTitle", -- 医生职称
" as "doctorRemark", -- 医生备注，如名医、退休
" as "doctorGender", -- 医生性别，1：男，2 女，3:未填
(select f.better_for from staff_dict f where f.user_name=b.doctor and rownum=1) as
"betterFor", -- 擅长
" as "urlPic", -- 头像地址
" as "scheduleNo", -- 排班记录号，表示当前排班记录的唯一标识
to_char(a.clinic_date, 'yyyy-mm-dd') as "regDate", -- 出诊日期，格式：YYYY-MM-DD
TO_CHAR(CLINIC_DATE, 'DAY') as "regWeekDay", -- 出诊日期对应的中文星期几，如
星期五
e.regtime as "scheduleType", -- 班别 A:上午 P:下午 N:晚上
'OPEN' as "regStatus", -- 出诊状态（STOP-停诊，OPEN-出诊，CLOSE-暂未开放）
sum(case
  when a.clinic_date = trunc(sysdate) then
    a.registration_limits
  when a.clinic_date > trunc(sysdate) then
    a.appointment_limits
end) as "regTotalCount", -- 该时段可预约的总号源数
sum(case

```

```

when a.clinic_date = trunc(sysdate) then
case
    when (a.registration_limits - a.current_no + 1) < 0 then
        0
    else
        a.registration_limits - a.current_no + 1
    end
when a.clinic_date > trunc(sysdate) then
case
    when (a.appointment_limits - a.current_no + 1) < 0 then
        0
    else
        a.appointment_limits - a.current_no + 1
    end
end) as "regleaveCount", -- 该时段剩余号源数

case
when a.clinic_date = trunc(sysdate) then
(select sum(price_list.price) * 100
 from COMM.price_list, COMM.CLINIC_TYPE_SETTING
 where price_list.item_code =
        CLINIC_TYPE_SETTING.Price_Item
 and CLINIC_TYPE_SETTING.clinic_type = b.clinic_type
 and CLINIC_TYPE_SETTING.Charge_Item = '挂号费'
 and price_list.stop_date is null)
when a.clinic_date > trunc(sysdate) then
(select sum(price_list.price) * 100
 from COMM.price_list
 where price_list.item_name = '计算机预约挂号'

```

```

        and price_list.stop_date is null)
end as "regFee", -- 挂号费(单位 “分” )

(select sum(price_list.price) * 100
 from COMM.price_list, COMM.CLINIC_TYPE_SETTING
 where price_list.item_code = CLINIC_TYPE_SETTING.Price_Item
 and CLINIC_TYPE_SETTING.clinic_type = b.clinic_type
 and CLINIC_TYPE_SETTING.Charge_Item = '诊疗费'
 and price_list.stop_date is null) as "treatFee", -- 诊疗费(单位 “分” )
'Y' as "isTimeReg" --, -- 是否挂号到具体时段 (Y - 是, N - 否)
-- to_char(a.time_start, 'hh24:mi') star_time, --开始时间
-- to_char(a.time_end, 'hh24:mi') end_time --结束时间
from OUTPADM.CLINIC_FOR_REGIST a,
    OUTPADM.CLINIC_INDEX    b,
    COMM.dept_dict          d,
    outpbill.period_vs_regtime e
where 1 = 1
and a.clinic_label = b.clinic_label
and b.clinic_dept = d.dept_code
and a.time_desc = e.period_time
and a.stop_flag = '2'
and a.CLINIC_DATE >= TO_DATE(beginDate, 'YYYY-MM-DD')
and a.CLINIC_DATE <= TO_DATE(endDate, 'YYYY-MM-DD')
and a.clinic_label not like '%急诊%'
and a.clinic_label not like '%体检%'
--and (case
--    when a.clinic_date = trunc(sysdate) then
--        a.registration_limits - a.registration_num

```

```

--    when a.clinic_date > trunc(sysdate) then
--    a.appointment_limits - a.appointment_num
--    end) > 0
and (case
    when a.CLINIC_DATE = trunc(sysdate) then
        b.sh_registration_flag
    when a.CLINIC_DATE > trunc(sysdate) then
        b.sh_appointment_flag
    end) = 1
and e.regtime in (v_sch_flag, v_sch_flag2)
and a.time_end >= sysdate
group by b.clinic_dept,
        d.dept_name,
        b.clinic_type,
        b.clinic_label,
        b.doctor_title,
        a.clinic_date,
        e.regtime,
        b.doctor
order by a.clinic_date, d.dept_name, b.doctor_title, e.regtime;

end if;
end;

-- 4.1.3. Doctor Registration Time-slot Information Query Interface 001 procedure
getDrRegTimeInfo001(hospitalId in varchar2, -- (必传) 医院 ID
                    deptId in varchar2, -- (必传) 科室代码 (一级)
                    doctorName in varchar2, -- 医生
                    scheduleType in varchar2, -- 时段

```

```

        regDate    in varchar2, -- (必传) 排班日期, 格式: YYYY-MM-DD
        aCur_temp out sys_refcursor) is
begin
    open aCur_temp for
    select b.clinic_dept as "deptId", -- 科室 id (二级)
           d.dept_name || '(' ||
           substr(b.clinic_type,
                 1,
                 decode(instr(b.clinic_type, '号', 1),
                        0,
                        length(b.clinic_type),
                        instr(b.clinic_type, '号', 1) - 1)) || ')' as "deptName", -- 科室名称 (二级)
           b.clinic_label as "doctorName", -- 医生名称
           to_char(a.clinic_date, 'yyyy-mm-dd') as "regDate", -- 出诊日期, 格式: YYYY-MM-DD

    case
        when a.clinic_date = trunc(sysdate) then
            a.registration_limits
        when a.clinic_date > trunc(sysdate) then
            a.appointment_limits
    end as "regTotalCount", -- 该时段可预约的总号源数

    case
        when a.clinic_date = trunc(sysdate) then
            case
                when (a.registration_limits - a.current_no + 1) < 0 then
                    0
                else
                    a.registration_limits - a.current_no + 1
            end
        else
            a.registration_limits
    end
end;

```



```

end
when a.clinic_date > trunc(sysdate) then
case
  when (a.appointment_limits - a.current_no + 1) < 0 then
    0
  else
    a.appointment_limits - a.current_no + 1
  end
end as "regleaveCount", -- 该时段剩余号源数

case
  when a.clinic_date = trunc(sysdate) then
    (select sum(price_list.price) * 100
     from COMM.price_list, COMM.CLINIC_TYPE_SETTING
     where price_list.item_code = CLINIC_TYPE_SETTING.Price_Item
     and CLINIC_TYPE_SETTING.clinic_type = b.clinic_type
     and CLINIC_TYPE_SETTING.Charge_Item = '挂号费'
     and price_list.stop_date is null)
  when a.clinic_date > trunc(sysdate) then
    (select sum(price_list.price) * 100
     from COMM.price_list
     where price_list.item_name = '计算机预约挂号'
     and price_list.stop_date is null)
end as "regFee", -- 挂号费(单位 “分” )

(select sum(price_list.price) * 100
 from COMM.price_list, COMM.CLINIC_TYPE_SETTING
 where price_list.item_code = CLINIC_TYPE_SETTING.Price_Item
 and CLINIC_TYPE_SETTING.clinic_type = b.clinic_type

```

```

        and CLINIC_TYPE_SETTING.Charge_Item = '诊疗费'

        and price_list.stop_date is null) as "treatFee", -- 诊疗费(单位“分”)
-- to_char(a.time_start, 'hh24:mi') as "beginTime", --开始时间
-- to_char(a.time_end, 'hh24:mi') as "endTime", --结束时间
a.time_desc as "scheduleFlag",
to_char(a.time_start, 'yyyy-mm-dd hh24:mi:ss') as "beginTime",
to_char(a.time_end, 'yyyy-mm-dd hh24:mi:ss') as "endTime"
from OUTPADM.CLINIC_FOR_REGIST a,
    OUTPADM.CLINIC_INDEX    b,
    COMM.dept_dict          d,
    outpbill.period_vs_regtime e
where 1 = 1

and a.clinic_label = b.clinic_label
and b.clinic_dept = d.dept_code
and a.time_desc = e.period_time
and a.stop_flag = '2'
and a.CLINIC_DATE = TO_DATE(regDate, 'YYYY-MM-DD')
and b.clinic_dept = deptId
and a.clinic_label = doctorName
and e.regtime = scheduleType

--and (case
--    when a.clinic_date = trunc(sysdate) then
--        a.registration_limits - a.registration_num
--    when a.clinic_date > trunc(sysdate) then
--        a.appointment_limits - a.appointment_num
--    end) > 0
and (case
    when a.CLINIC_DATE = trunc(sysdate) then
        b.sh_registration_flag

```

```

        when a.CLINIC_DATE > trunc(sysdate) then
            b.sh_appointment_flag
        end) = 1

    and a.time_end >= sysdate

    order by a.clinic_date, d.dept_name, a.time_start;
end;

-- 4.1.4. Suspension of Medical Service Notification Query Interface 001
procedure getStopShift001(hospitalId in varchar2, -- (必传) 医院 ID
    queryBeginTime in varchar2, -- (必传) 查询开始时间, 格式: YYYY-MM-DD
    hh24:mi:ss
    queryEndTime in varchar2, -- (必传) 查询结束时间, 格式: YYYY-MM-DD
    hh24:mi:ss
    aCur_temp out sys_refcursor) is
/* 输出结果
* 字段      描述
* -----
hospitalId  医院 ID
hospitalName  医院名称
deptClassId  科室分类 id
deptClassName  科室分类名称
deptId      科室 id
deptName     科室名称
doctorId     医生 ID
doctorName   医生名称
regDate      出诊日期, 格式: YYYY-MM-DD
scheduleTypeId  出诊班别 id

```

```

scheduleTypeName 出诊班别名称
optTime    操作时间
*/
sSql varchar2(2000);
begin
-- TODO SQL 逻辑
sSql := '
    select
    " as "hospitalId", -- 医院 ID
    " as "hospitalName", -- 医院名称
    " as "deptClassId", -- 科室分类 id
    " as "deptClassName", -- 科室分类名称
    " as "deptId", -- 科室 id
    " as "deptName", -- 科室名称
    " as "doctorId", -- 医生 ID
    " as "doctorName", -- 医生名称
    " as "regDate", -- 出诊日期, 格式: YYYY-MM-DD
    " as "scheduleTypeId", -- 出诊班别 id
    " as "scheduleTypeName", -- 出诊班别名称
    " as "optTime", -- 操作时间
    from [数据表]
    where 1 = 1
    and [条件]
';

open aCur_temp for sSql
    using hospitalId, queryBeginTime, queryEndTime;
end;

```

-- 4.2.1. Appointment Registration Order Generation Interface 001

procedure addPreRegOrder001(axmlparm in varchar2,

 aCur_temp out sys_refcursor,

 on_return out number, -- 结果编码

 on_message out varchar2 -- 结果描述

) is

/* 输出结果

* 字段 描述

* -----

on_return 结果代码

on_message 结果描述

patMedNo

regDate

doctorName

scheduleFlag

appointSeq

regFee

treatFee

*/

regDate varchar2(10); --预约挂号日期(格式: yyyy-MM-dd)

scheduleFlag varchar2(20); --班别（上午 下午 白天 昼夜）

doctorName varchar2(20); --医生名称(号别)

patMedNo varchar2(20); --患者诊疗卡号码

payAgentId varchar2(20); --虚拟操作员工号

deptId varchar2(20); --科室代码

mchId varchar2(30); --商户号。微信商户号、支付宝商户号等

```

payAppId  varchar2(30); --公众账号 ID
patMobile varchar2(30); --患者手机号码
operMobile varchar2(30); -- 挂号人手机号码
tsBeginTime varchar2(20);
tsEndTime  varchar2(20);

d_reg_date date; --挂号日期
d_now      date; --当前系统时间
d_pBirth   date; --出生日期

v_pid      VARCHAR2(20); --患者 ID
v_pIdNo    VARCHAR2(18); --身份证号
v_pInHosNo VARCHAR2(10); --住院号
v_pName    VARCHAR2(20); --患者姓名
v_pSex     VARCHAR2(4); --患者性别
v_pNamePhonetic varchar2(16);
v_pIdentity varchar2(40);
v_pChargeType varchar2(40);
v_pUnitInContract varchar2(40);
v_pMailingaddr varchar2(300);
v_pPhonenum varchar2(16);

v_clinic_type varchar2(8); --号别类型
v_reg_doctor  varchar2(20); --挂号医生
v_opera_no    varchar2(20); --操作工号
v_opera_name  varchar2(20); --操作人

v_reg_mode_code varchar2(1); --挂号类型编码

```

v_reg_mode_name varchar2(20); --挂号类型名称

n_pAge number(3); --患者年龄

n_registcount_ck number; --号别校验结果标识

n_regnum number; --剩余号源数

n_regcount number; --号别挂号费校验标识

n_operasum number; --操作员记录校验标识

n_reg_mode_count number; --挂号方式记录

n_regFee number; --HIS 挂号金额

n_reg number(5, 2); --挂号费（单位：元）

n_treat number(5, 2); --诊查费（单位：元）

n_appoint_count number;

n_appoint_seq number;

n_register_count number;

begin

```
insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'patMedNo') || '-' ||  
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-1',  
    'addPreRegOrder001',  
    axmlParm,  
    '进入 addPreRegOrder001-01' ||  
    of_get_xmlval(axmlParm, 'patMedNo'),  
    null,  
    null);
```

on_return := 0;

on_message := '成功';

```

regDate    := of_get_xmlval(axmlparm, 'regDate');
scheduleFlag := of_get_xmlval(axmlparm, 'scheduleFlag');
doctorName  := of_get_xmlval(axmlparm, 'doctorName');
patMedNo    := of_get_xmlval(axmlparm, 'patMedNo');
payAgentId  := of_get_xmlval(axmlparm, 'payAgentId');
deptId      := of_get_xmlval(axmlparm, 'deptId');
mchId       := of_get_xmlval(axmlparm, 'mchId');
payAppId    := of_get_xmlval(axmlparm, 'payAppId');
patMobile   := of_get_xmlval(axmlparm, 'patMobile');
operMobile  := of_get_xmlval(axmlparm, 'operMobile');
tsBeginTime := of_get_xmlval(axmlparm, 'beginTime');
tsEndTime   := of_get_xmlval(axmlparm, 'endTime');

```

```

d_reg_date := to_date(regDate, 'yyyy-mm-dd');

```

-- 获取并校验患者信息

```

select b.patient_id, --病人标识号
       b.id_no, --身份证
       b.inp_no, --住院号
       b.date_of_birth, --出生日期
       b.name, --姓名
       b.sex, --性别
       b.name_phonetic, --姓名拼音
       b.identity, --身份
       b.charge_type, --费别
       b.unit_in_contract, --合同单位
       b.mailing_address,

```



```

        b.phone_number_home
into v_pid,
        v_pIdNo,
        v_pInHosNo,
        d_pBirth,
        v_pName,
        v_pSex,
        v_pNamePhonetic,
        v_pIdentity,
        v_pChargeType,
        v_pUnitInContract,
        v_pMailingaddr,
        v_pPhonenum
from MEDREC.pat_master_index b
where b.patient_id = patMedNo;
if v_pid = '' or v_pid is null then
    on_return := -1;
    on_message := '不存在患者信息';
    goto end_program;
end if;

-- 校验挂号规则
if check_reg_rule(patMedNo, doctorName, scheduleFlag, d_reg_date) = '1' then
    on_return := -1;
    on_message := '男性患者不能挂妇科或产科';
    goto end_program;
elsif check_reg_rule(patMedNo, doctorName, scheduleFlag, d_reg_date) = '2' then
    on_return := -1;
    on_message := '14 岁(含)以下挂儿科';

```

```

    goto end_program;
end if;

-- 校验号别 clinic_label 是否存在
SELECT count(1)
into n_registcount_ck
FROM OUTPADM.clinic_for_regist
WHERE CLINIC_DATE = d_reg_date
and clinic_label = doctorName
and time_desc = scheduleFlag
and time_start = to_date(tsBeginTime, 'yyyy-mm-dd hh24:mi:ss')
and time_end = to_date(tsEndTime, 'yyyy-mm-dd hh24:mi:ss');
if n_registcount_ck < 1 then
on_return := -1;
on_message := '挂号类别不存在';
goto end_program;
end if;

--查询剩余号源数
SELECT nvl(t.appointment_limits, 0) - nvl(t.current_no, 0) + 1
into n_regnum
FROM OUTPADM.clinic_for_regist t
WHERE CLINIC_DATE = d_reg_date
and clinic_label = doctorName
and time_desc = scheduleFlag
and time_start = to_date(tsBeginTime, 'yyyy-mm-dd hh24:mi:ss')
and time_end = to_date(tsEndTime, 'yyyy-mm-dd hh24:mi:ss');
if n_regnum <= 0 then
on_return := -1;

```

```

on_message := '预约号源数已满';

goto end_program;

end if;

--校验号别挂号费是否维护

SElect count(1)

into n_regcount

FROM OUTPADM.CLINIC_INDEX, COMM.CLINIC_TYPE_DICT

WHERE (CLINIC_INDEX.CLINIC_TYPE = CLINIC_TYPE_DICT.CLINIC_TYPE)

and clinic_index.clinic_label = doctorName;

if n_regcount < 1 then

on_return := -1;

on_message := '挂号费未维护';

goto end_program;

end if;

--获取挂号费

SElect distinct a.CLINIC_TYPE,

((select price

from price_list x

where x.item_name = '计算机预约挂号'

and x.stop_date is null) +

(select price

from (select t.clinic_type, x.price, t.charge_item

from COMM.CLINIC_TYPE_SETTING t

left join price_list x

on t.price_item = x.item_code

and x.stop_date is null) c

where c.clinic_type = b.clinic_type

```

```

        and c.charge_item = '诊疗费'
        and rownum = 1)) * 100 as price,
(select price
 from price_list x
 where x.item_name = '计算机预约挂号'
        and x.stop_date is null) as regamount,
(select price
 from (select t.clinic_type, x.price, t.charge_item
        from COMM.CLINIC_TYPE_SETTING t
        left join price_list x
            on t.price_item = x.item_code
        and x.stop_date is null) c
 where c.clinic_type = b.clinic_type
        and c.charge_item = '诊疗费'
        and rownum = 1) as treatamount
into v_clinic_type, n_regFee, n_reg, n_treat
FROM OUTPADM.CLINIC_INDEX a, COMM.CLINIC_TYPE_SETTING b
WHERE a.clinic_type = b.clinic_type
        and a.clinic_label = doctorName;

if n_regFee is null then
    on_return := -1;
    on_message := '挂号费金额获取出错';
    goto end_program;
end if;

-- 校验同一天同一个科室是否已存在预约记录
select count(1)
into n_appoint_count

```

```

from outpadm.clinic_appoints t
where t.visit_date_appted = d_reg_date
    -- and t.clinic_label = doctorName
and t.visit_dept = deptId
    -- and t.visit_time_appted = scheduleFlag
and t.patient_id = patMedNo;

if n_appoint_count > 0 then
    on_return := -1;
    on_message := '同一天存在相同科室的未确认取号的预约记录，不允许重复预约';
    goto end_program;
end if;

-- 校验同一天同一个科室是否已存在挂号记录
select count(1)
into n_register_count
from outpadm.clinic_master t
where t.visit_date = d_reg_date
    -- and t.clinic_label = doctorName
and t.visit_dept = deptId
and t.patient_id = patMedNo
and t.returned_date is null
and t.registration_status <> 2;

if n_register_count > 0 then
    on_return := -1;
    on_message := '同一天存在相同科室的已挂号未就诊记录，不允许重复预约';
    goto end_program;
end if;

```

```

-- 取操作工号

select count(1)

into n_opera_sum

from COMM.staff_dict a

where 1 = 1 --job = '自助收款'

and user_name = upper(payAgentId);

if n_opera_sum > 0 then

select a.id, a.name

into v_opera_no, v_opera_name

from COMM.staff_dict a

where 1 = 1 --job = '自助收款'

and user_name = upper(payAgentId)

and rownum = 1;

else

v_opera_no := upper(payAgentId);

v_opera_name := upper(payAgentId);

end if;


select count(1)

into n_reg_mode_count

from comm.regist_mode_dict t

where t.mode_name = '微信预约';

if n_reg_mode_count > 0 then

select t.mode_code, t.mode_name

into v_reg_mode_code, v_reg_mode_name

from comm.regist_mode_dict t

where t.mode_name = '微信预约';

else

```

```

v_reg_mode_code := 'K';
v_reg_mode_name := '微信预约';
end if;

--系统时间
select sysdate into d_now from dual;

--计算患者年龄
n_pAge := floor((d_now - d_pBirth) / 365);

-- 取出号表当前排队号
select nvl(current_no, 0)
into n_appoint_seq
from OUTPADM.clinic_for_regist
Where clinic_date = d_reg_date
And clinic_label = doctorName
And time_desc = scheduleFlag;

-- 更新门诊号表
begin
update OUTPADM.clinic_for_regist
Set current_no    = nvl(current_no, 0) + 1,
    appointment_num = nvl(appointment_num, 0) + 1
Where clinic_date = d_reg_date
And clinic_label = doctorName
And time_desc = scheduleFlag
and current_no = n_appoint_seq
and time_start = to_date(tsBeginTime, 'yyyy-mm-dd hh24:mi:ss')
and time_end = to_date(tsEndTime, 'yyyy-mm-dd hh24:mi:ss');

```

```

if sql%rowcount <> 1 then

    ROLLBACK;

    on_return := -1;

    on_message := '更新门诊号表当前值失败，[' || sql%rowcount || ']条记录被更新，取出当前值' ||

        n_appoint_seq;

    goto end_program;

end if;

insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'patMedNo') || '-' ||

    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-3',

    'addPreRegOrder001',

    axmlParm,

    'addPreRegOrder001 更新门诊号表当前值-03' ||

    of_get_xmlval(axmlParm, 'patMedNo') || '-' ||

    sql%rowcount || ']条记录被更新，取出当前值' ||

    n_appoint_seq,

    on_return,

    on_message);

exception

when others then

    ROLLBACK;

    on_return := -1;

    on_message := '更新门诊号表失败:' || SQLERRM;

    goto end_program;

end;

```


--插入预约记录表

begin

INSERT INTO outpadm.clinic_appoints

(VISIT_DATE_APPTED,

CLINIC_LABEL,

VISIT_TIME_APPTED,

PATIENT_ID,

APPT_MADE_DATE,

serial_no,

card_name,

card_no,

mode_code,

name,

name_phonetic,

sex,

age,

identity,

charge_type,

MAILING_ADDRESS,

id_no,

DATE_OF_BIRTH,

operator,

APPOINT_TYPE,

phone_number_home,

time_start,

time_end)

VALUES

(d_reg_date,

doctorName,

```

scheduleFlag,
v_pid,
SYSDATE,
n_appoint_seq,
'微信',
null,
v_reg_mode_code,
v_pName,
v_pNamePhonetic,
v_pSex,
n_pAge,
v_pIdentity,
'自费',
-- v_pChargeType,
v_pMailingaddr,
v_pIdNo,
d_pBirth,
v_opera_no,
v_reg_mode_name,
v_pPhonenum,
to_date(tsBeginTime, 'yyyy-mm-dd hh24:mi:ss'),
to_date(tsEndTime, 'yyyy-mm-dd hh24:mi:ss'));
exception
when others then
    ROLLBACK;
on_return := -1;
on_message := '写入预约表失败:' || SQLERRM || '[' || d_reg_date || ']' ||
    doctorName || ',' || scheduleFlag || ',' ||
    n_appoint_seq || ']';

```

```

        goto end_program;
end;

<<end_program>>

if on_return = 0 then

insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'patMedNo') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
    'addPreRegOrder001',
    axmlParm,
    '结束 addPreRegOrder001' ||
    of_get_xmlval(axmlParm, 'patMedNo'),
    on_return,
    on_message);

open aCur_temp for
select to_char(on_return) as "returnCode",
    on_message as "resultDesc",
    t.patient_id as "patMedNo",
    t.visit_date_appted as "regDate",
    t.clinic_label as "doctorName",
    t.visit_time_appted as "scheduleFlag",
    t.serial_no as "appointSeq",
    (n_reg * 100) as "regFee",
    (n_treat * 100) as "treatFee",
    to_char(t.time_start, 'yyyy-mm-dd hh24:mi:ss') as "beginTime",
    to_char(t.time_end, 'yyyy-mm-dd hh24:mi:ss') as "endTime"
from outpadm.clinic_appoints t

```

```

where t.visit_date_appted = d_reg_date

and t.clinic_label = doctorName

and t.visit_time_appted = scheduleFlag

and t.patient_id = patMedNo

and t.serial_no = (select max(serial_no)
                    from outpadm.clinic_appoints s
                    where s.visit_date_appted = d_reg_date
                    and s.clinic_label = doctorName
                    and s.visit_time_appted = scheduleFlag
                    and s.patient_id = patMedNo);

else

insert_wx_webservice_err_logs('addPreRegOrder001' ||
                               to_char(systimestamp,
                                         '-yyyymmddhh24missxff3'),
                               'addPreRegOrder001',
                               axmlParm,
                               ",
                               on_message);

insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'patMedNo') || '-' ||
                      to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
                      'addPreRegOrder001',
                      axmlParm,
                      '[错]结束 addPreRegOrder001' ||
                      of_get_xmlval(axmlParm, 'patMedNo'),
                      on_return,
                      on_message);

```

```

open aCur_temp for
select to_char(on_return) as "returnCode",
       on_message as "resultDesc",
       " as "patMedNo",
       " as "regDate",
       " as "doctorName",
       " as "scheduleFlag",
       " as "appointitSeq",
       " as "regFee",
       " as "treatFee",
       " as "tsBeginTime",
       " as "tsEndTime"
from dual;
end if;

end;

-- 4.2.2. Appointment Registration Order Cancellation Interface 001
procedure cancelPreRegOrder001(axmlparm in varchar2,
                                aCur_temp out sys_refcursor,
                                on_return out number, -- 结果编码
                                on_message out varchar2 -- 结果描述
                                ) is

/* 输出结果
* 字段      描述
* -----
on_return    结果代码

```

```

    on_message    结果描述
*/

regDate    varchar2(10); --预约挂号日期(格式: yyyy-MM-dd)
scheduleFlag varchar2(20); --班别 (上午 下午 白天 昼夜)
doctorName  varchar2(20); --医生名称(号别)
patMedNo    varchar2(20); --患者诊疗卡号码
appointSeq  varchar2(3); --预约序号
isTimer     varchar2(1); --是否定时任务取消
tsBeginTime varchar2(20);
tsEndTime   varchar2(20);

d_reg_date date; --挂号日期
d_now    date; --当前系统时间

n_appoint_count number;

v_visit_date_appted date;
v_clinic_label    varchar2(16);
v_patient_id      VARCHAR2(20);
v_visit_time_appted VARCHAR2(8);
v_appt_made_date  DATE;
v_mode_code      CHAR(1);
v_card_name      VARCHAR2(16);
v_card_no        VARCHAR2(20);
v_serial_no      NUMBER(3);
v_name           VARCHAR2(20);
v_sex            VARCHAR2(4);
v_age            NUMBER(16);

```

```

v_identity      VARCHAR2(20);
v_charge_type   VARCHAR2(8);
v_name_phonetic VARCHAR2(16);
v_appoint_type  VARCHAR2(20);
v_operator      VARCHAR2(20);
v_phone_number_home VARCHAR2(12);
v_time_start    date;
v_time_end      date;

```

```

d_cancel_time date; --允许取消时间

```

```

begin

```

```

insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'patMedNo') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-1',
    'cancelPreRegOrder001',
    axmlParm,
    '进入 cancelPreRegOrder001-01' ||
    of_get_xmlval(axmlParm, 'patMedNo'),
    null,
    null);

```

```

on_return := 0;

```

```

on_message := '成功';

```

```

regDate      := of_get_xmlval(axmlparm, 'regDate');
scheduleFlag := of_get_xmlval(axmlparm, 'scheduleFlag');
doctorName   := of_get_xmlval(axmlparm, 'doctorName');
patMedNo     := of_get_xmlval(axmlparm, 'patMedNo');

```

```

appointitSeq := of_get_xmlval(axmlparm, 'appointitSeq');
isTimer      := of_get_xmlval(axmlparm, 'isTimer');

tsBeginTime := of_get_xmlval(axmlparm, 'beginTime');
tsEndTime   := of_get_xmlval(axmlparm, 'endTime');

d_reg_date := to_date(regDate, 'yyyy-mm-dd');

d_cancel_time := to_date(to_char((to_date(regDate, 'yyyy-mm-dd') - 1),
                                'yyyy-mm-dd') || ' 22:00:00',
                        'yyyy-mm-dd hh24:mi:ss');

--系统时间
select sysdate into d_now from dual;

if isTimer <> 'Y' then
    if sysdate > d_cancel_time then
        on_return := -1;
        on_message := '仅支持在预约日期前一天 22:00 前取消';
        goto end_program;
    end if;
end if;

--校验是否有预约记录
select count(1)
into n_appoint_count
from outpadm.clinic_appoints t
where t.visit_date_appted = d_reg_date
and t.clinic_label = doctorName

```



```

and t.visit_time_appted = scheduleFlag
and t.patient_id = patMedNo
and t.serial_no = appointSeq
and t.time_start = to_date(tsBeginTime, 'yyyy-mm-dd hh24:mi:ss')
and t.time_end = to_date(tsEndTime, 'yyyy-mm-dd hh24:mi:ss')
and t.card_name = '微信';

if n_appoint_count = 0 then
on_return := -1;
on_message := '未找到对应的预约记录';
goto end_program;
end if;

select t.visit_date_appted,
       t.clinic_label,
       t.patient_id,
       t.visit_time_appted,
       t.appt_made_date,
       t.mode_code,
       t.serial_no,
       t.card_name,
       t.name,
       t.name_phonetic,
       t.sex,
       t.age,
       t.identity,
       t.charge_type,
       t.phone_number_home,
       t.time_start,

```

```

        t.time_end
into v_visit_date_appted,
        v_clinic_label,
        v_patient_id,
        v_visit_time_appted,
        v_appt_made_date,
        v_mode_code,
        v_serial_no,
        v_card_name,
        v_name,
        v_name_phonetic,
        v_sex,
        v_age,
        v_identity,
        v_charge_type,
        v_phone_number_home,
        v_time_start,
        v_time_end
from outpadm.clinic_appoints t
where t.visit_date_appted = d_reg_date
and t.clinic_label = doctorName
and t.visit_time_appted = scheduleFlag
and t.patient_id = patMedNo
and t.serial_no = appointSeq
and t.time_start = to_date(tsBeginTime, 'yyyy-mm-dd hh24:mi:ss')
and t.time_end = to_date(tsEndTime, 'yyyy-mm-dd hh24:mi:ss')
and t.card_name = '微信';

```

--更新门诊号表，把 clinic_for_regist 当前号和已经预约号不变，限号字段加 1

```

begin
update clinic_for_regist t
    set --current_no    = current_no - 1,
        --appointment_num = appointment_num - 1
        registration_limits = registration_limits + 1,
        appointment_limits = appointment_limits + 1
where clinic_date = v_visit_date_appted
    and clinic_label = v_clinic_label
    and time_desc = v_visit_time_appted
    and t.time_start = v_time_start
    and t.time_end = v_time_end;
exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '更新门诊号表失败:' || SQLERRM;
    goto end_program;
end;

```

--删除预约记录

```

begin
DELETE FROM OUTPADM.CLINIC_APPOINTS
WHERE VISIT_DATE_APPTED = v_visit_date_appted
    AND CLINIC_LABEL = v_clinic_label
    AND PATIENT_ID = v_patient_id
    AND VISIT_TIME_APPTED = v_visit_time_appted
    AND APPT_MADE_DATE = v_appt_made_date
    -- AND MODE_CODE = v_mode_code
    -- AND CARD_NO IS NULL

```

```

-- AND SERIAL_NO = v_serial_no
AND CARD_NAME = v_card_name

-- AND NAME = v_name

-- AND NAME_PHONETIC = v_name_phonetic

-- AND SEX = v_sex

-- AND AGE = v_age

-- AND IDENTITY = v_identity

-- AND CHARGE_TYPE = v_charge_type

-- AND UNIT_IN_CONTRACT IS NULL

-- AND INSURANCE_TYPE IS NULL

-- AND INSURANCE_NO IS NULL

-- AND PHONE_NUMBER_HOME = v_phone_number_home

and time_start = v_time_start

and time_end = v_time_end;

if sql%rowcount <> 1 then

    ROLLBACK;

    on_return := -1;

    on_message := '取消预约删除记录失败, [' || sql%rowcount || ']条记录被删除';

    goto end_program;

end if;

exception

when others then

    ROLLBACK;

    on_return := -1;

    on_message := '删除预约记录失败:' || SQLERRM;

    goto end_program;

end;
```

```

--更新同一排班号预约记录排序号

/*begin

update OUTPADM.CLINIC_APPOINTS t

    set SERIAL_NO = SERIAL_NO - 1

where t.visit_date_appted = v_visit_date_appted

    and t.clinic_label = v_clinic_label

    and t.visit_time_appted = v_visit_time_appted

    and t.serial_no > v_serial_no;

exception

when others then

    ROLLBACK;

    on_return := -1;

    on_message := '更新其他预约记录排序号失败:' || SQLERRM;

    goto end_program;

end;

```

```

--更新同一排班号预约门诊就诊记录排序号

begin

UPDATE OUTPADM.clinic_master t

    set t.serial_no = t.serial_no - 1

where t.VISIT_DATE = v_visit_date_appted

    and t.clinic_label = v_clinic_label

    and t.visit_time_desc = v_visit_time_appted

    and t.serial_no > v_serial_no;

exception

when others then

    ROLLBACK;

    on_return := -1;

    on_message := '更新门诊就诊记录排序号失败:' || SQLERRM;

```

```

        goto end_program;
end;*/

<<end_program>>

if on_return = 0 then

insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'patMedNo') || '-' ||
        to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
        'cancelPreRegOrder001',
        axmlParm,
        '结束 cancelPreRegOrder001-02' ||
        of_get_xmlval(axmlParm, 'patMedNo'),
        null,
        null);

on_message := '取消预约成功';
open aCur_temp for
select to_char(on_return) as "returnCode",
        '取消预约成功' as "resultDesc"
from dual;
else

insert_wx_webservice_err_logs('cancelPreRegOrder001' ||
        to_char(systimestamp,
        '-yyyymmddhh24missxff3'),
        'cancelPreRegOrder001',
        axmlParm,
        ",

```

```

        on_message);

insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'patMedNo') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
    'cancelPreRegOrder001',
    axmlParm,
    '[错]结束 cancelPreRegOrder001-02' ||
    of_get_xmlval(axmlParm, 'patMedNo'),
    null,
    null);

open aCur_temp for
    select to_char(on_return) as "returnCode",
        on_message as "resultDesc"
    from dual;
end if;

end;

-- 4.2.3 Appointment Registration Order Payment Interface 001 预约挂号订单支付接口 001
procedure payPreRegOrder001(axmlparm in varchar2,
    aCur_temp out sys_refcursor,
    on_return out number, -- 结果编码
    on_message out varchar2 -- 结果描述
) is
/* 输出结果
* 字段      描述
* -----

```

on_return 结果代码
 on_message 结果描述
 bindChannels 就诊卡已绑定渠道，多个渠道用英文分号分割
 patMedNo 就诊卡号
 hisOrderPayId HIS 缴费订单号（缴费成功时返回 HIS 系统生成的缴费订单，要求唯一）
 orderDesc 订单相关信息描述
 printMsg 待打印信息
 consultingRoomId 诊察室代码
 receiptId 收据 ID
 */
 regDate varchar2(10); --预约挂号日期(格式: yyyy-MM-dd)
 fee varchar2(10); --挂号费(单位“分”)
 treatfee varchar2(10); --诊疗费(单位“分”)
 payAmount varchar2(10); --支付金额(单位“分”)
 patMedNo varchar2(20); --患者诊疗卡号码
 orderPayId varchar2(50); --第三方平台的流水号
 orderId varchar2(50); --渠道订单号
 scheduleFlag varchar2(20); --班别（上午 下午 白天 昼夜）
 doctorName varchar2(20); --医生名称(号别)
 payAgentId varchar2(20); --虚拟操作员工号
 deptId varchar2(20); --科室代码
 payTime varchar2(20); --交易时间，格式: YYYY-MM-DD HI24:MI:SS
 mchId varchar2(30); --商户号。微信商户号、支付宝商户号等
 payAppId varchar2(30); --公众账号 ID
 patMobile varchar2(30); --患者手机号码
 operMobile varchar2(30); --挂号人手机号码
 appointSeq varchar2(3); --预约序号


```

tsBeginTime varchar2(20);
tsEndTime  varchar2(20);

payMethod  varchar2(10);
v_payMethod varchar2(20);

insurPersonalId  VARCHAR2(50); --医保个人编号/帐号
insurCardNo      VARCHAR2(50); --医保卡号
insurCardClass   VARCHAR2(1); --职工居民标志
insurMedicalClassName VARCHAR2(50); --医保病人医疗类别
insurClassType   VARCHAR2(10); --医保病人类型
insurReturnid    VARCHAR2(50); --医保病人登记流水号
insurSettleId    VARCHAR2(50); --医保病人结算流水号
insurOvAreaCode  VARCHAR2(10); --统筹区号
insurAreaCode    VARCHAR2(10); --区域编号
samno           VARCHAR2(30); --小程序对账订单号
insurNo          VARCHAR2(30); --医保对账订单号

insurBegfee      VARCHAR2(30); --医保病人起付线
insurCardAmount  VARCHAR2(30); --医保病人消费前卡余额
insurMxbTclj     VARCHAR2(30); --医保病人统筹年度累计
insurMxbNIlj     VARCHAR2(30); --医保病人年度累计
insurBegfeeRemain VARCHAR2(30); --医保病人起付线剩余

n_insurBegfee    NUMBER(12, 4); --医保病人起付线
n_insurCardAmount NUMBER(12, 4); --医保病人消费前卡余额
n_insurMxbTclj   NUMBER(12, 4); --医保病人统筹年度累计
n_insurMxbNIlj   NUMBER(12, 4); --医保病人年度累计

```

n_insurBegfeeRemain NUMBER(12, 4); --医保病人起付线剩余

d_reg_date date; --挂号日期

d_now date; --当前系统时间

--d_pBirth date; --出生日期

v_visit_date_appted date;

v_clinic_label varchar2(16);

v_patient_id VARCHAR2(20);

v_visit_time_appted VARCHAR2(8);

v_appt_made_date DATE;

v_mode_code CHAR(1);

v_card_name VARCHAR2(16);

v_card_no VARCHAR2(20);

v_serial_no NUMBER(3);

v_name VARCHAR2(20);

v_sex VARCHAR2(4);

v_age NUMBER(16);

v_identity VARCHAR2(20);

v_charge_type VARCHAR2(8);

v_name_phonetic VARCHAR2(16);

v_appoint_type VARCHAR2(20);

v_operator VARCHAR2(20);

v_phone_number_home VARCHAR2(12);

v_unit_in_contract varchar2(11);

v_time_start date;

v_time_end date;

v_clinic_type varchar2(8); --号别类型

```

v_reg_doctor   varchar2(20); --挂号医生
v_opera_no     varchar2(20); --操作工号
v_opera_name   varchar2(20); --操作人
v_visit_prefix varchar2(20); --门诊流水号前缀
v_visit_suffix varchar2(20); --门诊流水号后缀
v_visit_no     varchar2(20); --门诊号
v_invoice_no   varchar2(20); --发票号
n_translog_count number; --交易记录校验结果标识
--n_registcount_ck number; --号别校验结果标识
n_regcount     number; --号别挂号费校验标识
n_opera_sum     number; --操作员记录校验标识
n_no_admit_visit number; --就诊记录校验标识
--n_regnum      number; --剩余号源数
n_regFee       number; --HIS 挂号金额
n_totalFee     number; --挂号费+诊查费（单位：分）
n_payMoney     number; --支付金额（单位：分）
n_reg          number(5, 2); --挂号费（单位：元）
n_treat       number(5, 2); --诊查费（单位：元）
--n_serial_no   number; --排队号
n_visit_seq    number(5); --就诊序号

v_simcode varchar2(20);

v_daytime   varchar2(14);
v_rcpt_seq  varchar2(10);
v_tmp_num   varchar2(5);
v_rcpt_nos  varchar2(20); --收据号
n_rcpt_costs number(8, 2);

```

```

n_rcpt_charges number(8, 2);
n_new_visit_no number(5);
v_n          number(5);

v_dr_name varchar2(20);

v_class_on_outp_rcpt VARCHAR2(20);
v_seriano      VARCHAR2(20); --流水号

v_class_on_reckoning varchar2(10);
v_subj_code     varchar2(4);
n_item_price    number(14, 4);

n_appoint_count number;
v_printMsg      clinic_index.clinic_position%type;
r_master        outp_rcpt_master%ROWTYPE; -- 结算主表
r_order         outp_order_desc%ROWTYPE;
r_items         outp_bill_items%ROWTYPE;
r_clinic_master OUTPADM.clinic_master%ROWTYPE; --就诊记录对象
r_payments      outp_payments_money%ROWTYPE;

cursor c_reg_price is
  SELECT price_list.item_class,
         price_list.item_code,
         price_list.item_name,
         price_list.item_spec,
         price_list.units,
         price_list.price,
         price_list.class_on_outp_rcpt,

```

```

        price_list.class_on_mr
FROM PRICE_LIST
left join COMM.CLINIC_TYPE_SETTING b
    on b.price_item = PRICE_LIST.item_code
and b.charge_item <> '挂号费'
AND (SYSDATE > PRICE_LIST.START_DATE)
AND (SYSDATE < PRICE_LIST.STOP_DATE OR
    PRICE_LIST.STOP_DATE IS NULL)
join OUTPADM.CLINIC_INDEX a
    on a.clinic_type = b.clinic_type
and a.clinic_label = doctorName
union all
SELECT price_list.item_class,
        price_list.item_code,
        price_list.item_name,
        price_list.item_spec,
        price_list.units,
        price_list.price,
        price_list.class_on_outp_rcpt,
        price_list.class_on_mr
FROM PRICE_LIST
where item_name = '计算机预约挂号'
and stop_date is null;

begin

insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'orderId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-1',
    'payPreRegOrder001',

```

```

        axmlParm,
        '进入 payPreRegOrder001-01' ||
        of_get_xmlval(axmlParm, 'orderId'),
        null,
        null);

on_return := 0;
on_message := '成功';

regDate    := of_get_xmlval(axmlparm, 'regDate');
fee        := of_get_xmlval(axmlparm, 'fee');
treatfee   := of_get_xmlval(axmlparm, 'treatfee');
payAmount  := of_get_xmlval(axmlparm, 'payAmount');
patMedNo   := of_get_xmlval(axmlparm, 'patMedNo');
orderPayId := of_get_xmlval(axmlparm, 'orderPayId');
orderId    := of_get_xmlval(axmlparm, 'orderId');
scheduleFlag := of_get_xmlval(axmlparm, 'scheduleFlag');
doctorName := of_get_xmlval(axmlparm, 'doctorName');
payAgentId := of_get_xmlval(axmlparm, 'payAgentId');
deptId     := of_get_xmlval(axmlparm, 'deptId');
payTime    := of_get_xmlval(axmlparm, 'payTime');
mchId      := of_get_xmlval(axmlparm, 'mchId');
payAppId   := of_get_xmlval(axmlparm, 'payAppId');
patMobile  := of_get_xmlval(axmlparm, 'patMobile');
operMobile := of_get_xmlval(axmlparm, 'operMobile');
appointSeq := of_get_xmlval(axmlparm, 'appointSeq');

tsBeginTime := of_get_xmlval(axmlparm, 'beginTime');
tsEndTime   := of_get_xmlval(axmlparm, 'endTime');

```

```

payMethod := of_get_xmlval(axmlparm, 'payMode');

insurPersonalId := of_get_xmlval(axmlparm, 'insurPersonalId');
insurCardNo := of_get_xmlval(axmlparm, 'insurCardNo');
insurCardClass := of_get_xmlval(axmlparm, 'insurCardClass');
insurMedicalClassName := of_get_xmlval(axmlparm,
                                         'insurMedicalClassName');
insurClassType := of_get_xmlval(axmlparm, 'insurClassType');
insurReturnid := of_get_xmlval(axmlparm, 'insurReturnid');
insurSettleid := of_get_xmlval(axmlparm, 'insurSettleid');
insurBegfee := of_get_xmlval(axmlparm, 'insurBegfee');
insurCardAmount := of_get_xmlval(axmlparm, 'insurCardAmount');
insurMxbTclj := of_get_xmlval(axmlparm, 'insurMxbTclj');
insurMxbNllj := of_get_xmlval(axmlparm, 'insurMxbNllj');
insurOvAreaCode := of_get_xmlval(axmlparm, 'insurOvAreaCode');
insurAreaCode := of_get_xmlval(axmlparm, 'insurAreaCode');
insurBegfeeRemain := of_get_xmlval(axmlparm, 'insurBegfeeRemain');
samno := of_get_xmlval(axmlparm, 'samno');
insurNo := of_get_xmlval(axmlparm, 'insurNo');

n_insurBegfee := to_number(insurBegfee);
n_insurCardAmount := to_number(insurCardAmount);
n_insurMxbTclj := to_number(insurMxbTclj);
n_insurMxbNllj := to_number(insurMxbNllj);
n_insurBegfeeRemain := to_number(insurBegfeeRemain);

d_reg_date := to_date(regDate, 'yyyy-mm-dd');
-- 挂号费+诊查费

```

```

n_totalFee := to_number(fee) + to_number(treatfee);

-- 支付金额

n_payMoney := to_number(payAmount);


-- 校验挂号规则

if check_reg_rule(patMedNo, doctorName, scheduleFlag, d_reg_date) = '1' then

    on_return := -1;

    on_message := '男性患者不能挂妇科或产科';

    goto end_program;

elsif check_reg_rule(patMedNo, doctorName, scheduleFlag, d_reg_date) = '2' then

    on_return := -1;

    on_message := '14 岁(含)以下请挂儿科';

    goto end_program;

end if;


--是否存在相同一笔交易

select count(1)

into n_translog_count

from wxgh.WX_TRANS_HIS_LOG t

where t.order_id = orderId

    and t.order_id_pay = orderPayId;

if n_translog_count > 0 then

    on_return := -1;

    on_message := '订单号' || orderId || '的交易已经存在';

    goto end_program;

end if;


--获取预约记录

```



```

select count(1)
into n_appoint_count
from outpadm.clinic_appoints t
where t.visit_date_appted = d_reg_date
and t.clinic_label = doctorName
and t.visit_time_appted = scheduleFlag
and t.patient_id = patMedNo
and t.serial_no = appointSeq
and t.time_start = to_date(tsBeginTime, 'yyyy-mm-dd hh24:mi:ss')
and t.time_end = to_date(tsEndTime, 'yyyy-mm-dd hh24:mi:ss')
and t.card_name = '微信';
if n_appoint_count = 0 then
on_return := -1;
on_message := '未找到对应的预约记录';
goto end_program;
end if;

begin
select t.visit_date_appted,
       t.clinic_label,
       t.patient_id,
       t.visit_time_appted,
       t.appt_made_date,
       t.mode_code,
       t.serial_no,
       t.card_name,
       t.name,
       t.name_phonetic,
       t.sex,

```

```

t.age,
t.identity,
t.charge_type,
t.phone_number_home,
t.unit_in_contract,
t.time_start,
t.time_end
into v_visit_date_appted,
v_clinic_label,
v_patient_id,
v_visit_time_appted,
v_appt_made_date,
v_mode_code,
v_serial_no,
v_card_name,
v_name,
v_name_phonetic,
v_sex,
v_age,
v_identity,
v_charge_type,
v_phone_number_home,
v_unit_in_contract,
v_time_start,
v_time_end
from outpadm.clinic_appoints t
where t.visit_date_appted = d_reg_date
and t.clinic_label = doctorName
and t.visit_time_appted = scheduleFlag

```

```

and t.patient_id = patMedNo
and t.serial_no = appointSeq
and t.time_start = to_date(tsBeginTime, 'yyyy-mm-dd hh24:mi:ss')
and t.time_end = to_date(tsEndTime, 'yyyy-mm-dd hh24:mi:ss')
and t.card_name = '微信';
exception
when others then
    ROLLBACK;
on_return := -1;
on_message := '获取预约记录出现异常:' || SQLERRM;
goto end_program;
end;

```

--获取挂号费

```

SElect distinct a.CLINIC_TYPE,
    ((select price
        from price_list x
        where x.item_name = '计算机预约挂号'
        and x.stop_date is null) +
    (select price
        from (select t.clinic_type, x.price, t.charge_item
            from COMM.CLINIC_TYPE_SETTING t
            left join price_list x
            on t.price_item = x.item_code
            and x.stop_date is null) c
        where c.clinic_type = b.clinic_type
        and c.charge_item = '诊疗费'
        and rownum = 1)) * 100 as price,
    (select price

```

```

        from price_list x
        where x.item_name = '计算机预约挂号'
        and x.stop_date is null) as regamount,
(select price
  from (select t.clinic_type, x.price, t.charge_item
        from COMM.CLINIC_TYPE_SETTING t
        left join price_list x
          on t.price_item = x.item_code
        and x.stop_date is null) c
  where c.clinic_type = b.clinic_type
  and c.charge_item = '诊疗费'
  and rownum = 1) as treatamount
into v_clinic_type, n_regFee, n_reg, n_treat
FROM OUTPADM.CLINIC_INDEX a, COMM.CLINIC_TYPE_SETTING b
WHERE a.clinic_type = b.clinic_type
  and a.clinic_label = v_clinic_label;

if n_regFee is null then
  on_return := -1;
  on_message := '挂号费金额获取出错';
  goto end_program;
end if;

-- 校验 HIS 挂号金额与传入挂号金额
if n_regFee <> n_totalFee then
  on_return := -1;
  on_message := 'HIS 挂号费金额与挂号费合计金额不一致';
  goto end_program;
end if;

```

```

-- 校验 HIS 挂号金额与支付金额
if n_regFee <> n_payMoney then
    on_return := -1;
    on_message := '支付金额与挂号费金额不一致';
    goto end_program;
end if;

```

```

-- 取操作工号
select count(1)
into n_opera_sum
from COMM.staff_dict a
where 1 = 1 --job = '自助收款'
    and user_name = upper(payAgentId);
if n_opera_sum > 0 then
    select a.id, a.name
    into v_opera_no, v_opera_name
    from COMM.staff_dict a
    where 1 = 1 --job = '自助收款'
        and user_name = upper(payAgentId)
        and rownum = 1;
else
    v_opera_no := upper(payAgentId);
    v_opera_name := upper(payAgentId);
end if;

```

```

-- 取排队号
/*select nvl(t.current_no, 1)

```

```

into n_serial_no
from OUTPADM.clinic_for_regist t
where t.clinic_date = v_visit_date_appted
and t.clinic_label = v_clinic_label
and t.time_desc = scheduleFlag;*/

-- 取就诊序号
SELECT OUTPADM.visit_no.NEXTVAL INTO n_visit_seq FROM dual;

-- 构造门诊流水号
v_visit_prefix := to_char(v_visit_date_appted, 'yyyymmdd');
v_visit_suffix := to_char(n_visit_seq);
v_visit_no := v_visit_prefix || v_visit_suffix;

--系统时间
select sysdate into d_now from dual;

--取挂号医生
select d.doctor,d.clinic_position
into v_reg_doctor,v_printMsg
FROM OUTPADM.CLINIC_INDEX d
where d.clinic_label = doctorName
and d.clinic_dept = deptId
and d.clinic_type = v_clinic_type
and rownum = 1;

--设置挂号就诊记录对象
r_clinic_master.visit_date := d_reg_date;
r_clinic_master.visit_no := n_visit_seq;

```

```

r_clinic_master.clinic_label      := doctorName;
r_clinic_master.visit_time_desc   := scheduleFlag;
r_clinic_master.serial_no        := v_serial_no;
r_clinic_master.patient_id       := v_patient_id;
r_clinic_master.name             := v_name;
r_clinic_master.name_phonetic    := v_name_phonetic;
r_clinic_master.sex              := v_sex;
r_clinic_master.age              := v_age;
r_clinic_master.identity         := v_identity;
r_clinic_master.charge_type      := '自费';
r_clinic_master.unit_in_contract := v_unit_in_contract;
r_clinic_master.visit_dept       := deptId;
r_clinic_master.doctor           := v_reg_doctor;
r_clinic_master.mr_provide_indicator := 0;
r_clinic_master.registering_date := d_now;
r_clinic_master.regist_fee       := 0; --n_reg;
r_clinic_master.clinic_fee       := 0; --n_treat;
r_clinic_master.other_fee        := 0;
r_clinic_master.operator         := v_opera_no;
r_clinic_master.clinic_charge    := 0; --(n_payMoney / 100);
r_clinic_master.pay_way          := '微信预约';
r_clinic_master.mode_code        := 'K';
r_clinic_master.registration_status := 1;
r_clinic_master.clinic_no        := v_visit_no;
r_clinic_master.CLINIC_TYPE      := v_clinic_type;
r_clinic_master.card_name        := '5';

r_clinic_master.REGISTRATION_STATUS := '1';

```

```

SELECT F_TRANS_PINYIN_CAPITAL(v_name) into v_simcode FROM DUAL;

r_clinic_master.simcode      := v_simcode;

r_clinic_master.phone_number_home := nvl(v_phone_number_home,
                                         nvl(patMobile, operMobile));


r_clinic_master.time_start := v_time_start;
r_clinic_master.time_end  := v_time_end;


-- 更新门诊号表
/*begin
update OUTPADM.clinic_for_regist
  Set current_no    = nvl(current_no, 0) + 1,
      registration_num = nvl(registration_num, 0) + 1
Where clinic_date = d_reg_date
  And clinic_label = doctorName
  And time_desc = scheduleFlag;
exception
when others then
  ROLLBACK;

  on_return := -1;

  on_message := '更新门诊号表失败:' || SQLERRM;

  goto end_program;
end;*/


-- 更新请求提供病案队列
begin
update OUTPADM.mr_request_queue
  set destination_dept = deptId,
      registration_date = sysdate,

```



```

        mr_location    = '4007', --写死
        clinic_label   = v_clinic_label,
        serial         = 1, --写死
        time_desc      = v_visit_time_appted
    where patient_id = v_patient_id;
exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '更新请求提供病案队列失败:' || SQLERRM;
    goto end_program;
end;

--收据号
v_daytime := to_char(d_now, 'yyyymmdd');

select nvl(max(substr(t.rcpt_no, 10)), 0)
into v_rcpt_seq
from outp_rcpt_master t
where t.rcpt_no like to_char(sysdate, 'yyyymmdd') || 'w%'
and t.rcpt_no not like to_char(sysdate, 'yyyymmdd') || 'wx%';
if v_rcpt_seq = 0 then
    select lpad('1', 5, '0') into v_tmp_num from dual;
    v_rcpt_nos := v_daytime || 'w' || v_tmp_num;
else
    v_rcpt_seq := to_number(v_rcpt_seq) + 1;
    select lpad(v_rcpt_seq, 5, '0') into v_tmp_num from dual;
    v_rcpt_nos := v_daytime || 'w' || v_tmp_num;
end if;

```

```

r_master.name      := v_name;
r_master.charge_type := '自费';
r_master.patient_id := v_patient_id;
r_master.name_phonetic := v_name_phonetic;
r_master.identity   := v_identity;

n_rcpt_costs := n_reg + n_treat;
n_rcpt_charges := (n_payMoney / 100);

r_master.rcpt_no      := v_rcpt_nos;
r_master.Visit_Date   := d_now; --预约写哪个日期
r_master.operator_no   := v_opera_no;
r_master.total_costs   := n_rcpt_costs; --(n_payamount / 100);
r_master.total_charges := n_rcpt_charges; --(n_payamount / 100);
r_master.charge_indicator := 0;
r_master.order_id_pay   := orderPayId;

if payMethod = 'XSYB' then
    r_master.insur_personal_id := insurPersonalId;
    r_master.insur_card_no     := insurCardNo;
    r_master.insur_card_class  := insurCardClass;
    r_master.insur_medical_class_name := insurMedicalClassName;
    r_master.insur_class_type   := insurClassType;
    r_master.insur_returnid     := insurReturnid;
    r_master.insur_settle_id    := insurSettleId;
    r_master.insur_ov_area_code := insurOvAreaCode;
    r_master.insur_area_code    := insurAreaCode;
    r_master.samno              := samno;

```

```

r_master.insur_no      := insurNo;
r_master.auto_flag     := '2';
r_master.charge_type   := '两定医保';

r_master.insur_begfee  := n_insurBegfee;
r_master.insur_card_amount := n_insurCardAmount;
r_master.insur_mxb_tclj := n_insurMxbTclj;
r_master.insur_mxb_nllj := n_insurMxbNllj;
r_master.insur_begfee_remain := n_insurBegfeeRemain;
else
    r_master.samno := orderId;
end if;

begin
    -- 插入门诊医疗收据记录
    INSERT INTO OUTPBILL.outp_rcpt_master VALUES r_master;
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '插入门诊医疗收据记录出错' || '['当前收据号:' || v_rcpt_nos || ']' ||
            SQLERRM;
        goto end_program;
end;

SELECT VISIT_NO.NEXTVAL into n_new_visit_no FROM DUAL; --注意权限问题
r_order.visit_date := d_now; --预约写哪个日期
r_order.visit_no := n_new_visit_no;
r_order.patient_id := v_patient_id;

```

```

r_order.ordered_by_dept := r_clinic_master.visit_dept;

if v_reg_doctor <> null and v_reg_doctor <> '' then
    select s.user_name
    into v_dr_name
    from comm.staff_dict s
    where s.user_name = r_clinic_master.doctor;
    r_order.ordered_by_doctor := v_dr_name;
else
    r_order.ordered_by_doctor := v_reg_doctor;
end if;

r_order.rcpt_no      := v_rcpt_nos;
r_order.visit_id    := r_clinic_master.visit_no;
r_order.clinic_no   := r_clinic_master.clinic_no;
r_order.presc_indicator := 0;

--插入开单记录
begin
    INSERT INTO OUTPBILL.outp_order_desc VALUES r_order;
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '插入开单记录出错' || SQLERRM;
        goto end_program;
end;

```

v_n := 0;

for v_reg_price in c_reg_price loop

SELECT PRICE_LIST.CLASS_ON_OUTP_RCPT,

price_list.class_on_reckoning,

price_list.subj_code,

price_list.price

into v_class_on_outp_rcpt,

v_class_on_reckoning,

v_subj_code,

n_item_price

FROM PRICE_LIST

WHERE PRICE_LIST.ITEM_CLASS = v_reg_price.item_class

AND PRICE_LIST.ITEM_CODE = v_reg_price.item_code

AND PRICE_LIST.ITEM_SPEC = v_reg_price.item_spec

AND PRICE_LIST.UNITS = v_reg_price.units

AND (SYSDATE > PRICE_LIST.START_DATE)

AND (SYSDATE < PRICE_LIST.STOP_DATE OR

PRICE_LIST.STOP_DATE IS NULL)

and rownum = 1;

r_items.visit_date := d_now;

r_items.visit_no := n_new_visit_no;

r_items.rcpt_no := v_rcpt_nos;

r_items.ITEM_no := v_n + 1;

r_items.item_class := v_reg_price.item_class;

r_items.class_on_rcpt := v_CLASS_ON_OUTP_RCPT;

```

r_items.item_code    := v_reg_price.item_code;
r_items.item_name    := v_reg_price.item_name;
r_items.ITEM_SPEC    := v_reg_price.item_spec;
r_items.amount       := 1;
r_items.units        := v_reg_price.units;
r_items.performed_by := deptId;
r_items.costs        := v_reg_price.price;
r_items.charges      := v_reg_price.price;

--r_items.flag        := 0;
r_items.repetition    := 1;
r_items.class_on_reckoning := v_class_on_reckoning;
r_items.subj_code     := v_subj_code;
--r_items.price_quotiety := v_orders_cost.price_quotiety;
r_items.item_price := n_item_price;
--r_items.order_no     := v_orders_cost.order_no;
r_items.class_on_outp_rcpt := v_reg_price.class_on_outp_rcpt;
r_items.class_on_mr    := v_reg_price.class_on_mr;

--插入门诊病人诊疗费用项目
begin
    INSERT INTO OUTPBILL.OUTPUT_BILL_ITEMS VALUES r_items;
    -- INSERT INTO OUTPUT_BILL_ITEMS_bak VALUES r_items;
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '插入门诊病人诊疗费用项目出错' || SQLERRM;
        goto end_program;

```

```

end;

v_n := v_n + 1;

end loop;

if payMethod = 'XCXJF' or payMethod = '' or payMethod is null then
    v_payMethod := '小程序';
elsif payMethod = 'XSYB' then
    v_payMethod := '线上医保';
end if;

-- 写入挂号支付方式记录表
begin
    INSERT INTO OUTPADM.CLINIC_PAYMENTS_MONEY
        (visit_date,
         visit_no,
         PAYMENT_NO,
         MONEY_TYPE,
         PAYMENT_AMOUNT,
         refunded_amount)
    VALUES
        (d_reg_date, n_visit_seq, 1, v_payMethod, 0, 0);
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '写入挂号支付方式记录:' || SQLERRM;
        goto end_program;
end;

```

```

r_payments.rcpt_no      := v_rcpt_nos;
r_payments.payment_no   := 1;
r_payments.money_type   := v_payMethod;
r_payments.payment_amount := (n_payMoney / 100);
r_payments.refunded_amount := 0;

--插入门诊病人支付方式记录
begin
    INSERT INTO OUTPBILL.outp_payments_money VALUES r_payments;
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '插入门诊病人支付方式记录出错' || SQLERRM;
        goto end_program;
end;

-- 写入门诊就诊记录
begin
    INSERT INTO OUTPADM.clinic_master VALUES r_clinic_master;
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '写入门诊就诊记录失败:' || SQLERRM;
        goto end_program;
end;

```


--预约确认支付后，删除预约记录

begin

DELETE FROM OUTPADM.CLINIC_APPOINTS

WHERE VISIT_DATE_APPTED = v_visit_date_appted

AND CLINIC_LABEL = v_clinic_label

AND PATIENT_ID = v_patient_id

AND VISIT_TIME_APPTED = v_visit_time_appted

AND SERIAL_NO = v_serial_no

-- and mode_code = v_mode_code

AND CARD_NAME = v_card_name

-- AND NAME = v_name

-- AND NAME_PHONETIC = v_name_phonetic

-- AND SEX = v_sex

-- AND AGE = v_age

-- AND IDENTITY = v_identity

-- AND CHARGE_TYPE = v_charge_type

-- AND PHONE_NUMBER_HOME = v_phone_number_home

and time_start = v_time_start

and time_end = v_time_end;

if sql%rowcount <> 1 then

ROLLBACK;

on_return := -1;

on_message := '预约确认删除记录失败，[' || sql%rowcount || ']条记录被删除';

goto end_program;

end if;

exception

when others then

ROLLBACK;

```

on_return := -1;

on_message := '预约确认，删除预约记录失败:' || SQLERRM;

goto end_program;

end;

-- 写入微信交易日志表

begin

insert into wxgh.wx_trans_his_log

(PATIENT_ID,

PATIENT_MOBILE,

ORDER_TYPE,

ORDER_ID,

ORDER_ID_PAY,

DEAL_MONEY,

PAY_APP_ID,

MCH_ID,

ORDER_PAY_TIME,

ORDER_DESC,

RCPT_NO,

SYS_DATETIME,

REG_RCPT_NO)

values

(v_patient_id,

nvl(patMobile, operMobile),

'REG',

orderId,

orderPayId,

payAmount,

payAppId,

```

```

        mchId,
        payTime,
        v_payMethod,
        v_visit_no,
        sysdate,
        v_rcpt_nos);
exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '写入微信交易日志表失败:' || SQLERRM;
    goto end_program;
end;

<<end_program>>

if on_return = 0 then

    insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'orderId') || '-' ||
        to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
        'payPreRegOrder001',
        axmlParm,
        '结束 payPreRegOrder001-02' ||
        of_get_xmlval(axmlParm, 'orderId'),
        on_return,
        on_message);

    open aCur_temp for
        select to_char(on_return) as "returnCode",

```

```

on_message as "resultDesc",
t.clinic_no as "hisOrderId",
t.visit_dept as "consultingRoomId",
--nvl(t.invoice_no, t.clinic_no || '-' || t.serial_no) as "receiptId",
v_rcpt_nos as "receiptId",
v_printMsg printMsg
from OUTPADM.clinic_master t
where t.clinic_no = v_visit_no;
else

```

```

insert_wx_webservice_err_logs('payPreRegOrder001' ||
to_char(systimestamp,
'-yyyymmddhh24missxff3'),
'payPreRegOrder001',
axmlParm,
",
on_message);

```

```

insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'orderId') || '-' ||
to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
'payPreRegOrder001',
axmlParm,
'['错]结束 payPreRegOrder001-02' ||
of_get_xmlval(axmlParm, 'orderId'),
on_return,
on_message);

```

```

open aCur_temp for
select to_char(on_return) as "returnCode",

```

```

        on_message as "resultDesc",
        " as "hisOrderId",
        " as "consultingRoomId",
        " as "receiptId",
        " printMsg
    from dual;
end if;

end;

-- 4.2.4 Appointment Registration Order Refund Interface 001 预约挂号订单退费接口 001
procedure refundPreReg001(axmlparm in varchar2,
    aCur_temp out sys_refcursor,
    on_return out number, -- 结果编码
    on_message out varchar2 -- 结果描述
) is
/* 输出结果
* 字段      描述
* -----
on_return    结果代码
on_message   结果描述
orderDesc    订单相关信息描述
printMsg     待打印信息
consultingRoomId 诊察室代码
receiptId    收据 ID
*/

patMedNo     varchar2(20); --患者诊疗卡号码

```

hisOrderId varchar2(20); --his 预约索引号
orderId varchar2(50); --订单号
orderPayId varchar2(50); --第三方平台支付流水号
orderRefundId varchar2(50); --第三方平台退费流水号
refundFee varchar2(10); --退款金额(单位“分”)
refundAgentId varchar2(20); --虚拟操作员工号

n_refundFee number;
n_translog_count number;
n_dealMoney number;
n_clinicmaster number;
n_no_admit_visit number;

v_visit_date date;
v_visit_no number;
v_regist_fee number(5, 2);
v_clinic_fee number(5, 2);
v_other_fee number(5, 2);
v_clinic_charge number(5, 2);
v_operator varchar2(20);
v_clinic_label varchar2(16);
v_time_desc varchar2(8);
v_patient_id varchar2(20);
v_patient_name varchar2(20);
v_serial_no number;
v_pay_way varchar2(8);
v_invoice_no varchar2(20);
v_clinic_no varchar2(13);

```

v_visit_dept varchar2(8);
v_doctor     varchar2(20);
v_time_start date;
v_time_end   date;

n_opera_sum number;
v_opera_no  varchar2(20); --操作工号
v_opera_name varchar2(20); --操作人

v_sysdate date;

v_pay_visit_no  number(5);
v_pay_rcpt_no   varchar2(20); --原挂号收据号
v_pay_rcpt_no_temp varchar2(20); --原挂号收据号(逻辑传输用)

n_rcpt_costs  number(8, 2);
n_rcpt_charges number(8, 2);
v_charge_type VARCHAR2(8);
v_name_phonetic VARCHAR2(16);
v_identity    VARCHAR2(20);

v_money_type varchar2(20); -- 原支付方式

v_refund_rcpt_no varchar2(20); --退费收据号
v_daytime        varchar2(14);
v_rcpt_seq       varchar2(10);
v_tmp_num        varchar2(5);

```

```
n_refund_visit_no number(5);
```

```
r_master outp_rcpt_master%ROWTYPE; -- 结算主表
```

```
r_order outp_order_desc%ROWTYPE;
```

```
r_items outp_bill_items%ROWTYPE;
```

```
r_payments outp_payments_money%ROWTYPE;
```

```
cursor c_OUTP_BILL_ITEMS is
```

```
select t.VISIT_DATE,  
       t.VISIT_NO,  
       t.RCPT_NO,  
       t.ITEM_NO,  
       t.ITEM_CLASS,  
       t.CLASS_ON_RCPT,  
       t.ITEM_CODE,  
       t.ITEM_NAME,  
       t.ITEM_SPEC,  
       t.AMOUNT,  
       t.UNITS,  
       t.PERFORMED_BY,  
       t.COSTS,  
       t.CHARGES,  
       t.INVOICE_NO,  
       t.FLAG,  
       t.REPETITION,  
       t.CLASS_ON_RECKONING,  
       t.SUBJ_CODE,  
       t.PRICE_QUOTIETY,  
       t.ITEM_PRICE,
```



```

        t.ORDER_NO,

        t.SUB_ORDER_NO,

        t.CLASS_ON_MR,

        t.CLASS_ON_OUTP_RCPT
    from OUTP_BILL_ITEMS t
    where t.visit_no = v_pay_visit_no
        and t.rcpt_no = v_pay_rcpt_no;

begin

insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'orderId') || '-' ||
                    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-1',
                    'refundPreReg001',
                    axmlParm,
                    '进入 refundPreReg001-01' ||
                    of_get_xmlval(axmlParm, 'orderId'),
                    null,
                    null);

hisOrderId := of_get_xmlval(axmlparm, 'hisOrderId');
patMedNo   := of_get_xmlval(axmlparm, 'patMedNo');
orderPayId := of_get_xmlval(axmlparm, 'orderPayId');
orderId    := of_get_xmlval(axmlparm, 'orderId');
refundFee  := of_get_xmlval(axmlparm, 'refundFee');
refundAgentId := of_get_xmlval(axmlparm, 'refundAgentId');

on_return := 0;
on_message := '成功';

```

```

n_refundFee := to_number(refundFee);

v_sysdate := sysdate;

--是否存在要退的订单交易
select count(1)
into n_translog_count
from wxgh.WX_TRANS_HIS_LOG t
where t.order_id = orderId
and t.order_id_pay = orderPayId
and t.order_id_refund is null
and t.order_type = 'REG';
if n_translog_count = 0 then
on_return := -1;
on_message := '订单号' || orderId || '的交易不存在或已退款';
goto end_program;
end if;

-- 校验原交易金额与传入退费金额
select t.deal_money, t.reg_rcpt_no
into n_dealMoney, v_pay_rcpt_no_temp
from wxgh.WX_TRANS_HIS_LOG t
where t.order_id = orderId
and t.order_id_pay = orderPayId
and t.order_id_refund is null
and t.order_type = 'REG';
if n_dealMoney <> n_refundFee then
on_return := -1;
on_message := '原交易金额与传入退费金额不一致';

```

```

        goto end_program;
    end if;

--校验预约挂号记录是否存在
select count(1)
    into n_clinicmaster
    from OUTPADM.clinic_master t
    where t.patient_id = patMedNo
        and t.clinic_no = hisOrderId;
if n_clinicmaster = 0 then
    on_return := -1;
    on_message := '预约号' || hisOrderId || '的 HIS 预约挂号记录不存在';
    goto end_program;
end if;

--校验预约是否已接诊
select t.registration_status
    into n_no_admit_visit
    from OUTPADM.clinic_master t
    where t.patient_id = patMedNo
        and t.clinic_no = hisOrderId;
if n_no_admit_visit = 2 then
    on_return := -1;
    on_message := '预约号' || hisOrderId || '已接诊，不允许退号';
    goto end_program;
end if;

--获取挂号主记录
select VISIT_DATE,

```

```
VISIT_NO,  
CLINIC_LABEL,  
VISIT_TIME_DESC,  
SERIAL_NO,  
PATIENT_ID,  
NAME,  
REGIST_FEE,  
CLINIC_FEE,  
OTHER_FEE,  
CLINIC_CHARGE,  
PAY_WAY,  
INVOICE_NO,  
clinic_no,  
operator,  
t.visit_dept,  
t.doctor,  
t.time_start,  
t.time_end  
into v_visit_date,  
v_visit_no,  
v_clinic_label,  
v_time_desc,  
v_serial_no,  
v_patient_id,  
v_patient_name,  
v_regist_fee,  
v_clinic_fee,  
v_other_fee,  
v_clinic_charge,
```

```

v_pay_way,
v_invoice_no,
v_clinic_no,
v_operator,
v_visit_dept,
v_doctor,
v_time_start,
v_time_end

from OUTPADM.clinic_master t
where t.patient_id = patMedNo
and t.clinic_no = hisOrderId;

if v_sysdate >=
to_date(to_char((v_visit_date - 1), 'yyyy-mm-dd') || ' 22:00:00',
'yyyy-mm-dd hh24:mi:ss') then
on_return := -1;
on_message := '仅支持' || to_char(v_visit_date - 1, 'yyyy-mm-dd') ||
' 22:00:00 前进行预约退号，逾期请与医院工作人员联系。';
goto end_program;
end if;

--查询预约挂号开单记录，获取挂号支付收据号
select t.visit_no, t.rcpt_no
into v_pay_visit_no, v_pay_rcpt_no
from OUTPBILL.outp_order_desc t
where t.patient_id = patMedNo
and t.clinic_no = hisOrderId
and t.rcpt_no = v_pay_rcpt_no_temp;
if v_pay_rcpt_no is null or v_pay_rcpt_no = '' then

```

```
on_return := -1;
on_message := '获取预约挂号支付收据号异常';
goto end_program;
end if;
```

--获取原收费主记录

```
select t.total_costs,
       t.total_charges,
       t.charge_type,
       t.name_phonetic,
       t.identity
into n_rcpt_costs,
     n_rcpt_charges,
     v_charge_type,
     v_name_phonetic,
     v_identity
from OUTPBILL.outp_rcpt_master t
where t.rcpt_no = v_pay_rcpt_no
and t.patient_id = v_patient_id;
```

--获取原支付方式

```
select t.money_type
into v_money_type
from OUTPBILL.OUTP_PAYMENTS_MONEY t
where t.rcpt_no = v_pay_rcpt_no;
```

-- 取操作工号

```
select count(1)
into n_opera_sum
```

```

from COMM.staff_dict a
where 1 = 1 --job = '自助收款'
      and user_name = upper(refundAgentId);
if n_operate_sum > 0 then
select a.id, a.name
      into v_operate_no, v_operate_name
      from COMM.staff_dict a
      where 1 = 1 --job = '自助收款'
            and user_name = upper(refundAgentId)
            and rownum = 1;
else
v_operate_no := upper(refundAgentId);
v_operate_name := upper(refundAgentId);
end if;

-----退费数据流逻辑 start-----

--设置退费收据号
v_daytime := to_char(v_sysdate, 'yyyymmdd');
select nvl(max(substr(t.rcpt_no, 10)), 0)
      into v_rcpt_seq
      from outp_rcpt_master t
      where t.rcpt_no like to_char(sysdate, 'yyyymmdd') || 'w%'
            and t.rcpt_no not like to_char(sysdate, 'yyyymmdd') || 'wx%';
if v_rcpt_seq = 0 then
select lpad('1', 5, '0') into v_tmp_num from dual;
v_refund_rcpt_no := v_daytime || 'w' || v_tmp_num;
else
v_rcpt_seq := to_number(v_rcpt_seq) + 1;

```

```

select lpad(v_rcpt_seq, 5, '0') into v_tmp_num from dual;
v_refund_rcpt_no := v_daytime || 'w' || v_tmp_num;
end if;

--设置退费收据主记录
r_master.name      := v_patient_name;
r_master.charge_type := v_charge_type;
r_master.patient_id := v_patient_id;
r_master.name_phonetic := v_name_phonetic;
r_master.identity   := v_identity;
r_master.rcpt_no     := v_refund_rcpt_no;
r_master.Visit_Date  := v_sysdate; --预约写哪个日期
r_master.operator_no := v_opera_no;
r_master.total_costs := n_rcpt_costs * -1;
r_master.total_charges := n_rcpt_charges * -1;
r_master.charge_indicator := 2;

begin
-- 插入门诊医疗收据记录

INSERT INTO OUTPBILL.outp_rcpt_master VALUES r_master;
exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '插入门诊医疗收据记录出错' || SQLERRM;
    goto end_program;
end;

--设置开单记录

```


SELECT VISIT_NO.NEXTVAL into n_refund_visit_no FROM DUAL; --注意权限问题

r_order.visit_date := v_sysdate; --预约写哪个日期

r_order.visit_no := n_refund_visit_no;

r_order.patient_id := v_patient_id;

r_order.ordered_by_dept := v_visit_dept;

r_order.ordered_by_doctor := v_doctor;

r_order.rcpt_no := v_refund_rcpt_no;

r_order.clinic_no := v_clinic_no;

r_order.presc_indicator := 0;

--插入开单记录

begin

INSERT INTO OUTPBILL.outp_order_desc VALUES r_order;

exception

when others then

ROLLBACK;

on_return := -1;

on_message := '插入开单记录出错' || SQLERRM;

goto end_program;

end;

--设置费用明细负的记录

for v_outp_bill_items in c_OUTP_BILL_ITEMS loop

r_items.visit_date := v_sysdate;

r_items.visit_no := n_refund_visit_no;

r_items.rcpt_no := v_refund_rcpt_no;

r_items.ITEM_no := v_outp_bill_items.item_no;

r_items.item_class := v_outp_bill_items.item_class;

r_items.class_on_rcpt := v_outp_bill_items.class_on_rcpt;

```

r_items.item_code      := v_outp_bill_items.item_code;
r_items.item_name      := v_outp_bill_items.item_name;
r_items.ITEM_SPEC      := v_outp_bill_items.item_spec;
r_items.amount         := v_outp_bill_items.amount * -1;
r_items.units          := v_outp_bill_items.units;
r_items.performed_by   := v_outp_bill_items.performed_by;
r_items.costs          := v_outp_bill_items.costs * -1;
r_items.charges        := v_outp_bill_items.charges * -1;
r_items.flag           := v_outp_bill_items.flag;
r_items.repetition     := v_outp_bill_items.repetition;
r_items.class_on_reckoning := v_outp_bill_items.class_on_reckoning;
r_items.subj_code      := v_outp_bill_items.subj_code;
r_items.price_quotiety := v_outp_bill_items.price_quotiety;
r_items.item_price     := v_outp_bill_items.item_price;
r_items.order_no       := v_outp_bill_items.order_no;
r_items.sub_order_no   := v_outp_bill_items.sub_order_no;
r_items.class_on_outp_rcpt := v_outp_bill_items.class_on_outp_rcpt;
r_items.class_on_mr     := v_outp_bill_items.class_on_mr;

```

--插入门诊病人诊疗费用项目

begin

```
    INSERT INTO OUTPBILL.OUTPUT_BILL_ITEMS VALUES r_items;
```

exception

when others then

```
    ROLLBACK;
```

```
    on_return := -1;
```

```
    on_message := '插入门诊病人诊疗费用项目出错' || SQLERRM;
```

```
    goto end_program;
```

end;

```

end loop;

--设置支付方式

r_payments.rcpt_no      := v_refund_rcpt_no;
r_payments.payment_no   := 1;
r_payments.money_type   := v_money_type; --'小程序';
r_payments.payment_amount := (n_refundFee / 100) * -1;
r_payments.refunded_amount := 0;
--插入门诊病人支付方式记录
begin
    INSERT INTO OUTPBILL.outp_payments_money VALUES r_payments;
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '插入门诊病人支付方式记录出错' || SQLERRM;
        goto end_program;
end;

--更新原收费主记录
begin
    update OUTPBILL.outp_rcpt_master t
        set t.charge_indicator = 2, t.refunded_rcpt_no = v_refund_rcpt_no
    where t.rcpt_no = v_pay_rcpt_no
        and t.patient_id = v_patient_id;
exception
    when others then
        ROLLBACK;
        on_return := -1;

```

```

on_message := '插入门诊病人支付方式记录出错' || SQLERRM;

goto end_program;

end;

```

--更新门诊号表

```

begin

update OUTPADM.clinic_for_regist t

set --current_no    = current_no - 1,

    --appointment_num = appointment_num - 1

    registration_limits = registration_limits + 1,

    appointment_limits = appointment_limits + 1

where clinic_label = v_clinic_label

and time_desc = v_time_desc

and clinic_date = v_visit_date

and t.time_start = v_time_start

and t.time_end = v_time_end;

exception

when others then

    ROLLBACK;

on_return := -1;

on_message := '更新门诊号表失败:' || SQLERRM;

goto end_program;

end;

```

--更新门诊就诊记录

```

begin

UPDATE OUTPADM.clinic_master t

set RETURNED_DATE = v_sysdate, RETURNED_OPERATOR = v_opera_no

where VISIT_NO = v_visit_no

```

```

        and t.patient_id = v_patient_id
        and t.clinic_no = v_clinic_no;
exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '更新门诊就诊记录失败:' || SQLERRM;
    goto end_program;
end;

--更新门诊就诊记录排序号
/*begin
UPDATE OUTPADM.clinic_master t
    set t.serial_no = t.serial_no - 1
where t.VISIT_DATE = v_visit_date
    and t.clinic_label = v_clinic_label
    and t.visit_time_desc = v_time_desc
    and t.serial_no > v_serial_no;
exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '更新门诊就诊记录排序号失败:' || SQLERRM;
    goto end_program;
end;*/
-----退费数据流逻辑 end-----

<<end_program>>

```

if on_return = 0 then

```
insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'orderId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
    'refundPreReg001',
    axmlParm,
    '结束 refundPreReg001-02' ||
    of_get_xmlval(axmlParm, 'orderId'),
    on_return,
    null);
```

open aCur_temp for

```
select to_char(on_return) as "returnCode",
    on_message as "resultDesc",
    t.clinic_no as "hisOrderId",
    t.returned_date as "returnDate",
    t.name || ', 就诊流水号 【' || nvl(t.invoice_no, t.clinic_no) || '】', ' ||
    to_char(t.visit_date, 'yyyy-mm-dd') || ' ' || t.clinic_label ||
    ', 排队号 (' || t.serial_no || ') 的预约已取消。' as "printMsg"
from OUTPADM.CLINIC_MASTER t
where t.patient_id = patMedNo
    and t.clinic_no = hisOrderId;
```

else

```
insert_wx_webservice_err_logs('refundPreReg001' ||
    to_char(systimestamp,
        '-yyyymmddhh24missxff3'),
    'refundPreReg001',
    axmlParm,
```

```

        ",
        on_message);

insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'orderId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
    'refundPreReg001',
    axmlParm,
    '[错]结束 refundPreReg001-02' ||
    of_get_xmlval(axmlParm, 'orderId'),
    null,
    null);

open aCur_temp for
select to_char(on_return) as "returnCode",
    on_message as "resultDesc",
    " as "hisOrderId",
    " as "returnDate",
    " as "printMsg"
from dual;

end if;

end;

-- 4.2.5 Refund Rewriting 退费回传退费单号接口 001
procedure returnRefundNo001(axmlparm in varchar2,
    aCur_temp out sys_refcursor,
    on_return out number, -- 结果编码
    on_message out varchar2 -- 结果描述
) is

```

```

patMedNo  varchar2(20);

orderPayId varchar2(50); --第三方平台的支付流水号

orderId   varchar2(50); --渠道订单号

refundPayId varchar2(50); --第三方平台的退费流水号

refundTime varchar2(30);

refundDesc varchar2(50);


n_count number;

begin

insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'orderId') || '-' ||
                      to_char(systimestamp, 'yyyymmddhh24missxff3') || '-1',
                      'returnRefundNo001',
                      axmlParm,
                      '进入 returnRefundNo001-01' ||
                      of_get_xmlval(axmlParm, 'orderId'),
                      null,
                      null);


patMedNo  := of_get_xmlval(axmlparm, 'patMedNo');
orderPayId := of_get_xmlval(axmlparm, 'orderPayId');
orderId    := of_get_xmlval(axmlparm, 'orderId');
refundPayId := of_get_xmlval(axmlparm, 'refundPayId');
refundTime := of_get_xmlval(axmlparm, 'refundTime');
refundDesc := of_get_xmlval(axmlparm, 'refundDesc');


on_return := 0;

on_message := '成功';

```


--查询订单交易日志记录

```
select count(1)
into n_count
from wxgh.wx_trans_his_log t
where t.patient_id = patMedNo
and t.order_id = orderId
and t.order_id_pay = orderPayId;

if n_count = 0 then
on_return := -1;
on_message := '订单号' || orderId || '交易记录不存在';
goto end_program;
end if;
```

--更新微信交易日志

```
begin
update wxgh.wx_trans_his_log t
set t.order_id_refund = refundPayId,
t.order_refund_time = refundTime,
t.refund_desc = refundDesc
where t.patient_id = patMedNo
and t.order_id = orderId
and t.order_id_pay = orderPayId;
exception
when others then
ROLLBACK;
on_return := -1;
on_message := '更新微信交易日志表失败:' || SQLERRM;
goto end_program;
```

```

end;

<<end_program>>

if on_return = 0 then

insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'orderId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
    'returnRefundNo001',
    axmlParm,
    '结束 returnRefundNo001-02' ||
    of_get_xmlval(axmlParm, 'orderId'),
    null,
    null);

open aCur_temp for
select to_char(on_return) as "returnCode",
    on_message as "resultDesc",
    t.order_id as "orderId",
    t.order_id_pay as "orderPayId",
    t.order_id_refund as "refundPayId",
    t.order_refund_time as "refundTime",
    t.refund_desc as "refundDesc"
from wxgh.wx_trans_his_log t
where t.order_id = orderId;

else

insert_wx_webservice_err_logs('returnRefundNo001' ||
    to_char(systimestamp,

```

```

        '-yyyymmddhh24missxff3'),
        'returnRefundNo001',
        axmlParm,
        ",
        on_message);

insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'orderId') || '-' ||
        to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
        'returnRefundNo001',
        axmlParm,
        '[错]结束 returnRefundNo001-02' ||
        of_get_xmlval(axmlParm, 'orderId'),
        null,
        null);

open aCur_temp for
select to_char(on_return) as "returnCode",
        on_message as "resultDesc",
        null as "orderId",
        null as "orderPayId",
        null as "refundPayId",
        null as "refundTime",
        null as "refundDesc"
        from dual;
end if;
end;

-- 4.3.1 Same-day Registration Source Locking Interface 001 当天挂号号源锁定接口 001
procedure lockRegSeqToday001(axmlparm in varchar2,

```

```

        aCur_temp out sys_refcursor,
        on_return out number, -- 结果编码
        on_message out varchar2 -- 结果描述
    ) is
/* 输出结果
* 字段      描述
* -----
on_return    结果代码
on_message   结果描述
isFullAddr   是否是四级地址（Y -- 是，N -- 不是）
*/
regDate      varchar2(10); --预约挂号日期(格式： yyyy-MM-dd)
fee          varchar2(10); --挂号费(单位“分” )
treatfee     varchar2(10); --诊疗费(单位“分” )
payAmount    varchar2(10); --支付金额(单位“分” )
patMedNo     varchar2(20); --患者诊疗卡号码
orderPayId   varchar2(50); --第三方平台的流水号
orderId      varchar2(50); --渠道订单号
scheduleFlag varchar2(20); --班别（上午 下午 白天 昼夜）
doctorName   varchar2(20); --医生名称(号别)
payAgentId   varchar2(20); --虚拟操作员工号
deptId       varchar2(20); --科室代码
payTime      varchar2(20); --交易时间， 格式： YYYY-MM-DD HI24:MI:SS
mchId        varchar2(30); --商户号。微信商户号、支付宝商户号等
payAppId     varchar2(30); --公众账号 ID
patMobile    varchar2(30); --患者手机号码
operMobile   varchar2(30); -- 挂号人手机号码
tsBeginTime  varchar2(20);

```

```

tsEndTime varchar2(20);

d_reg_date date; --挂号日期
d_now date; --当前系统时间
d_pBirth date; --出生日期

v_pid VARCHAR2(20); --患者 ID
v_pIdNo VARCHAR2(18); --身份证号
v_pInHosNo VARCHAR2(10); --住院号
v_pName VARCHAR2(20); --患者姓名
v_pSex VARCHAR2(4); --患者性别
v_pNamePhonetic varchar2(40);
v_pIdentity varchar2(40);
v_pChargeType varchar2(40);
v_pUnitInContract varchar2(40);
v_pMobie varchar2(20);

v_clinic_type varchar2(8); --号别类型
n_translog_count number; --交易记录校验结果标识
n_registcount_ck number; --号别校验结果标识
n_regcount number; --号别挂号费校验标识
n_no_admit_visit number; --就诊记录校验标识
n_regnum number; --剩余号源数
n_regFee number; --HIS 挂号金额
n_totalFee number; --挂号费+诊查费（单位：分）
n_payMoney number; --支付金额（单位：分）
n_reg number(5, 2); --挂号费（单位：元）
n_treat number(5, 2); --诊查费（单位：元）

```

```

begin
insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'visitId') || '-' ||
                    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-1',
                    'lockRegSeqToday001',
                    axmlParm,
                    '进入 lockRegSeqToday001-01' ||
                    of_get_xmlval(axmlParm, 'orderId'),
                    null,
                    null);

on_return := 0;
on_message := '成功';

regDate    := of_get_xmlval(axmlparm, 'regDate');
fee        := of_get_xmlval(axmlparm, 'fee');
treatfee   := of_get_xmlval(axmlparm, 'treatfee');
payAmount  := of_get_xmlval(axmlparm, 'payAmount');
patMedNo   := of_get_xmlval(axmlparm, 'patMedNo');
orderPayId := of_get_xmlval(axmlparm, 'orderPayId');
orderId    := of_get_xmlval(axmlparm, 'orderId');
scheduleFlag := of_get_xmlval(axmlparm, 'scheduleFlag');
doctorName := of_get_xmlval(axmlparm, 'doctorName');
payAgentId := of_get_xmlval(axmlparm, 'payAgentId');
deptId     := of_get_xmlval(axmlparm, 'deptId');
payTime    := of_get_xmlval(axmlparm, 'payTime');
mchId      := of_get_xmlval(axmlparm, 'mchId');
payAppId   := of_get_xmlval(axmlparm, 'payAppId');
patMobile  := of_get_xmlval(axmlparm, 'patMobile');
operMobile := of_get_xmlval(axmlparm, 'operMobile');

```

```
tsBeginTime := of_get_xmlval(axmlparm, 'beginTime');
tsEndTime   := of_get_xmlval(axmlparm, 'endTime');
```

```
d_reg_date := to_date(regDate, 'yyyy-mm-dd');
-- 挂号费+诊查费
n_totalFee := to_number(fee) + to_number(treatfee);
-- 支付金额
n_payMoney := to_number(payAmount);
```

```
-- 获取并校验患者信息
```

```
select b.patient_id, --病人标识号
       b.id_no, --身份证
       b.inp_no, --住院号
       b.date_of_birth, --出生日期
       b.name, --姓名
       b.sex, --性别
       b.name_phonetic, --姓名拼音
       b.identity, --身份
       b.charge_type, --费别
       b.unit_in_contract, --合同单位
       b.phone_number_home
into v_pid,
    v_pIdNo,
    v_pInHosNo,
    d_pBirth,
    v_pName,
    v_pSex,
    v_pNamePhonetic,
```

```

    v_pIdentity,
    v_pChargeType,
    v_pUnitInContract,
    v_pMobie
from MEDREC.pat_master_index b
where b.patient_id = patMedNo;
if v_pid = '' or v_pid is null then
    on_return := -1;
    on_message := '不存在患者信息';
    goto end_program;
end if;

-- 校验挂号规则
if check_reg_rule(patMedNo, doctorName, scheduleFlag, d_reg_date) = '1' then
    on_return := -1;
    on_message := '男性患者不能挂妇科或产科';
    goto end_program;
elsif check_reg_rule(patMedNo, doctorName, scheduleFlag, d_reg_date) = '2' then
    on_return := -1;
    on_message := '14 岁(含)以下请挂儿科';
    goto end_program;
end if;

--是否存在相同一笔交易
select count(1)
into n_translog_count
from wxgh.WX_TRANS_HIS_LOG t
where t.order_id = orderId
and t.order_id_pay = orderPayId;

```



```

if n_translog_count > 0 then
    on_return := -1;
    on_message := '订单号' || orderId || '的交易已经存在';
    goto end_program;
end if;

-- 校验号别 clinic_label 是否存在
SELECT count(1)
into n_registcount_ck
FROM OUTPADM.clinic_for_regist
WHERE CLINIC_DATE = d_reg_date
    and clinic_label = doctorName
    and time_desc = scheduleFlag
    and time_start = to_date(tsBeginTime, 'yyyy-mm-dd hh24:mi:ss')
    and time_end = to_date(tsEndTime, 'yyyy-mm-dd hh24:mi:ss');
if n_registcount_ck < 1 then
    on_return := -1;
    on_message := '挂号类别不存在';
    goto end_program;
end if;

--查询剩余号源数
SELECT nvl(t.registration_limits, 0) - nvl(t.current_no, 0) + 1
into n_regnum
FROM OUTPADM.clinic_for_regist t
WHERE CLINIC_DATE = d_reg_date
    and clinic_label = doctorName
    and time_desc = scheduleFlag
    and time_start = to_date(tsBeginTime, 'yyyy-mm-dd hh24:mi:ss')

```

```

        and time_end = to_date(tsEndTime, 'yyyy-mm-dd hh24:mi:ss');
if n_regnum <= 0 then
    on_return := -1;
    on_message := '挂号号源数已满';
    goto end_program;
end if;

--校验号别挂号费是否维护
SElect count(1)
into n_regcount
FROM OUTPADM.CLINIC_INDEX, COMM.CLINIC_TYPE_DICT
WHERE (CLINIC_INDEX.CLINIC_TYPE = CLINIC_TYPE_DICT.CLINIC_TYPE)
    and clinic_index.clinic_label = doctorName;
if n_regcount < 1 then
    on_return := -1;
    on_message := '挂号费未维护';
    goto end_program;
end if;

--获取挂号费
SElect distinct a.CLINIC_TYPE,
    ((select price
        from (select t.clinic_type, x.price, t.charge_item
            from COMM.CLINIC_TYPE_SETTING t
            left join price_list x
            on t.price_item = x.item_code
            and x.stop_date is null) c
        where c.clinic_type = b.clinic_type
        and c.charge_item = '挂号费'
    )

```

```

        and rownum = 1) +
(select price
  from (select t.clinic_type, x.price, t.charge_item
        from COMM.CLINIC_TYPE_SETTING t
        left join price_list x
          on t.price_item = x.item_code
        and x.stop_date is null) c
 where c.clinic_type = b.clinic_type
       and c.charge_item = '诊疗费'
       and rownum = 1)) * 100 as price,
(select price
  from (select t.clinic_type, x.price, t.charge_item
        from COMM.CLINIC_TYPE_SETTING t
        left join price_list x
          on t.price_item = x.item_code
        and x.stop_date is null) c
 where c.clinic_type = b.clinic_type
       and c.charge_item = '挂号费'
       and rownum = 1) as regamount,
(select price
  from (select t.clinic_type, x.price, t.charge_item
        from COMM.CLINIC_TYPE_SETTING t
        left join price_list x
          on t.price_item = x.item_code
        and x.stop_date is null) c
 where c.clinic_type = b.clinic_type
       and c.charge_item = '诊疗费'
       and rownum = 1) as treatamount
into v_clinic_type, n_regFee, n_reg, n_treat

```

```

FROM OUTPADM.CLINIC_INDEX a, COMM.CLINIC_TYPE_SETTING b

WHERE a.clinic_type = b.clinic_type

and a.clinic_label = doctorName;

if n_regFee is null then
    on_return := -1;
    on_message := '挂号费金额获取出错';
    goto end_program;
end if;

-- 校验 HIS 挂号金额与传入挂号金额
if n_regFee <> n_totalFee then
    on_return := -1;
    on_message := 'HIS 挂号费金额与挂号费合计金额不一致';
    goto end_program;
end if;

-- 校验 HIS 挂号金额与支付金额
if n_regFee <> n_payMoney then
    on_return := -1;
    on_message := '支付金额与挂号费金额不一致';
    goto end_program;
end if;

--校验当天是否存在相同号别的未就诊记录
select count(1)
into n_no_admit_visit
from OUTPADM.clinic_master v

```

```

where v.patient_id = v_pid
      -- and v.clinic_label = doctorName
and v.visit_dept = deptId
and v.visit_date = d_reg_date
      -- and v.visit_time_desc = scheduleFlag
and v.returned_date is null
and v.registration_status <> 2;
if n_no_admit_visit > 0 then
  on_return := -1;
  on_message := '当天存在相同科室的未就诊记录，不允许重复挂号';
  goto end_program;
end if;

<<end_program>>

if on_return = 0 then
  insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'visitId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
    'lockRegSeqToday001',
    axmlParm,
    '结束 lockRegSeqToday001-02' ||
    of_get_xmlval(axmlParm, 'orderId'),
    null,
    null);
  open aCur_temp for
    select to_char(on_return) as "returnCode",
           on_message as "resultDesc",
           " as "hisOrderId",
           " as "consultingRoomId",
           " as "receiptId"

```

```

        from dual;
else
insert_wx_webservice_err_logs('lockRegSeqToday001' ||

        to_char(systimestamp,

                '-yyyymmddhh24missxff3'),

        'lockRegSeqToday001',

        axmlparm,

        ",

        on_message);

insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'visitId') || '-' ||

        to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',

        'lockRegSeqToday001',

        axmlParm,

        '[错误]结束 lockRegSeqToday001-02' ||

        of_get_xmlval(axmlParm, 'orderId'),

        null,

        null);

open aCur_temp for
select to_char(on_return) as "returnCode",
        on_message as "resultDesc",
        " as "hisOrderId",
        " as "consultingRoomId",
        " as "receiptId"
        from dual;
end if;
end;
```

-- 4.3.2 Same-day Registration Source Unlocking Interface 001 当天挂号解除锁定接口 001

```
procedure unlockRegSeqToday001(hospitalId in varchar2, -- (必传) 医院 ID
                                isZJ      in varchar2, -- 是否是诊间加号 (Y-是, N-否, 默认值是 N)
                                lockId     in varchar2, -- (必传) 号源锁定 ID(渠道产生)
                                on_return  out number, -- 结果编码
                                on_message out varchar2 -- 结果描述
                                ) is
/* 输出结果
* 字段      描述
* -----
on_return   结果代码
on_message  结果描述
*/
v_result integer;
begin
begin
-- TODO SQL 逻辑

--例：执行 当天挂号解锁 逻辑

if v_result > 0 then
on_return := 0;
on_message := '成功';
goto end_program;
else
on_return := -1;
on_message := '失败';
end if;
```

```

exception
when no_data_found then
    ROLLBACK;
    on_return := -1;
    on_message := '错误信息' || SQLERRM;
    goto end_program;
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '错误信息' || SQLERRM;
    goto end_program;
end;

```

```

<<end_program>>
null;
end;

```

-- 4.3.3 Same-day Registration Interface 001 当天挂号接口 001

```

procedure addRegOrder001(axmlparm in varchar2,
    aCur_temp out sys_refcursor,
    on_return out number, -- 结果编码
    on_message out varchar2 -- 结果描述
) is
/* 输出结果
* 字段      描述
* -----
on_return   结果代码

```


on_message 结果描述
 bindChannels 就诊卡已绑定渠道，多个渠道用英文分号分割
 patMedNo 就诊卡号
 hisOrderPayId HIS 缴费订单号（缴费成功时返回 HIS 系统生成的缴费订单，要求唯一）
 orderDesc 订单相关信息描述
 printMsg 待打印信息
 consultingRoomId 诊察室代码
 receiptId 收据 ID
 */
 regDate varchar2(10); --预约挂号日期(格式: yyyy-MM-dd)
 fee varchar2(10); --挂号费(单位“分”)
 treatfee varchar2(10); --诊疗费(单位“分”)
 payAmount varchar2(10); --支付金额(单位“分”)
 patMedNo varchar2(20); --患者诊疗卡号码
 orderPayId varchar2(50); --第三方平台的流水号
 orderId varchar2(50); --渠道订单号
 scheduleFlag varchar2(20); --班别（上午 下午 白天 昼夜）
 doctorName varchar2(20); --医生名称(号别)
 payAgentId varchar2(20); --虚拟操作员工号
 deptId varchar2(20); --科室代码
 payTime varchar2(20); --交易时间，格式: YYYY-MM-DD HI24:MI:SS
 mchId varchar2(30); --商户号。微信商户号、支付宝商户号等
 payAppId varchar2(30); --公众账号 ID
 patMobile varchar2(30); --患者手机号码
 operMobile varchar2(30); --挂号人手机号码
 tsBeginTime varchar2(20);
 tsEndTime varchar2(20);

```

payMethod varchar2(10);
v_payMethod varchar2(20);

insurPersonalId VARCHAR2(50); --医保个人编号/帐号
insurCardNo VARCHAR2(50); --医保卡号
insurCardClass VARCHAR2(1); --职工居民标志
insurMedicalClassName VARCHAR2(50); --医保病人医疗类别
insurClassType VARCHAR2(10); --医保病人类型
insurReturnId VARCHAR2(50); --医保病人登记流水号
insurSettleId VARCHAR2(50); --医保病人结算流水号
insurOvAreaCode VARCHAR2(10); --统筹区号
insurAreaCode VARCHAR2(10); --区域编号
samno VARCHAR2(30); --小程序对账订单号
insurNo VARCHAR2(30); --医保对账订单号

insurBegfee VARCHAR2(30); --医保病人起付线
insurCardAmount VARCHAR2(30); --医保病人消费前卡余额
insurMxbTclj VARCHAR2(30); --医保病人统筹年度累计
insurMxbNllj VARCHAR2(30); --医保病人年度累计
insurBegfeeRemain VARCHAR2(30); --医保病人起付线剩余

n_insorBegfee NUMBER(12, 4); --医保病人起付线
n_insorCardAmount NUMBER(12, 4); --医保病人消费前卡余额
n_insorMxbTclj NUMBER(12, 4); --医保病人统筹年度累计
n_insorMxbNllj NUMBER(12, 4); --医保病人年度累计
n_insorBegfeeRemain NUMBER(12, 4); --医保病人起付线剩余

d_reg_date date; --挂号日期

```

```

d_now    date; --当前系统时间
d_pBirth date; --出生日期

v_pid      VARCHAR2(20); --患者 ID
v_pIdNo    VARCHAR2(18); --身份证号
v_pInHosNo VARCHAR2(10); --住院号
v_pName     VARCHAR2(20); --患者姓名
v_pSex      VARCHAR2(4); --患者性别
v_pNamePhonetic varchar2(40);
v_pIdentity varchar2(40);
v_pChargeType varchar2(40);
v_pUnitInContract varchar2(40);
v_pMobie    varchar2(20);

v_clinic_type varchar2(8); --号别类型
v_reg_doctor  varchar2(20); --挂号医生
v_opera_no    varchar2(20); --操作工号
v_opera_name  varchar2(20); --操作人
v_visit_prefix varchar2(20); --门诊流水号前缀
v_visit_suffix varchar2(20); --门诊流水号后缀
v_visit_no    varchar2(20); --门诊号
v_invoice_no   varchar2(20); --发票号
n_translog_count number; --交易记录校验结果标识
n_registcount_ck number; --号别校验结果标识
n_regcount    number; --号别挂号费校验标识
n_opera_sum    number; --操作员记录校验标识
n_no_admit_visit number; --就诊记录校验标识
n_regnum      number; --剩余号源数

```

n_regFee number; --HIS 挂号金额
n_totalFee number; --挂号费+诊查费（单位：分）
n_payMoney number; --支付金额（单位：分）
n_reg number(5, 2); --挂号费（单位：元）
n_treat number(5, 2); --诊查费（单位：元）
n_serial_no number; --排队号
n_visit_seq number(5); --就诊序号
n_pAge number(3); --患者年龄

v_simcode varchar2(20);

v_daytime varchar2(14);
v_rcpt_seq varchar2(10);
v_tmp_num varchar2(5);
v_rcpt_nos varchar2(20); --收据号
n_rcpt_costs number(8, 2);
n_rcpt_charges number(8, 2);
n_new_visit_no number(5);
v_n number(5);

v_dr_name varchar2(20);

v_class_on_outp_rcpt VARCHAR2(20);
v_seriano VARCHAR2(20); --流水号

v_class_on_reckoning varchar2(10);
v_subj_code varchar2(4);
n_item_price number(14, 4);

```

v_printMsg    clinic_index.clinic_position%type;

r_master      outp_rcpt_master%ROWTYPE; -- 结算主表

r_order       outp_order_desc%ROWTYPE;

r_items       outp_bill_items%ROWTYPE;

r_clinic_master OUTPADM.clinic_master%ROWTYPE; --就诊记录对象

r_payments    outp_payments_money%ROWTYPE;

```

```

cursor c_reg_price is

```

```

    SELECT price_list.item_class,
           price_list.item_code,
           price_list.item_name,
           price_list.item_spec,
           price_list.units,
           price_list.price,
           price_list.prefer_price,
           price_list.foreigner_price,
           price_list.performed_by,
           price_list.fee_type_mask,
           price_list.class_on_inp_rcpt,
           price_list.class_on_outp_rcpt,
           price_list.class_on_reckoning,
           price_list.subj_code,
           price_list.class_on_mr,
           price_list.memo,
           price_list.start_date,
           price_list.stop_date,
           price_list.operator,
           price_list.enter_date,
           price_list.high_price,

```

price_list.tib_rowguid,
price_list.tib_flowguid,
price_list.note1,
price_list.material_code,
price_list.item_content,
price_list.item_exclusion,
price_list.input_code,
price_list.input_code_wb,
price_list.ybz1,
price_list.lgz1,
price_list.ybscdm,
price_list.mzfb,
price_list.bffb,
price_list.ybbzmc,
price_list.score_1,
price_list.score_2,
price_list.price_name_code,
price_list.control_flag,
price_list.std_code_1,
price_list.changed_memo,
price_list.operation_level,
price_list.license_id,
price_list.class_on_insur_mr,
price_list.group_flag,
price_list.tj_flag,
price_list.bbyc,
price_list.package_spec,
price_list.firm_id,
price_list.charge_according,

```

        price_list.mr_bill_class
FROM OUTPADM.CLINIC_INDEX a, COMM.CLINIC_TYPE_SETTING b
left join PRICE_LIST
    on b.price_item = PRICE_LIST.item_code
AND (SYSDATE > PRICE_LIST.START_DATE)
AND (SYSDATE < PRICE_LIST.STOP_DATE OR
    PRICE_LIST.STOP_DATE IS NULL)
where a.clinic_type = b.clinic_type
    and a.clinic_label = doctorName;

```

begin

```

insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'orderId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-1',
    'addRegOrder001',
    axmlParm,
    '进入 addRegOrder001-01' ||
    of_get_xmlval(axmlParm, 'orderId'),
    null,
    null);

```

on_return := 0;

on_message := '成功';

regDate := of_get_xmlval(axmlparm, 'regDate');

fee := of_get_xmlval(axmlparm, 'fee');

treatfee := of_get_xmlval(axmlparm, 'treatfee');

payAmount := of_get_xmlval(axmlparm, 'payAmount');

patMedNo := of_get_xmlval(axmlparm, 'patMedNo');

```

orderPayId := of_get_xmlval(axmlparm, 'orderPayId');
orderId    := of_get_xmlval(axmlparm, 'orderId');
scheduleFlag := of_get_xmlval(axmlparm, 'scheduleFlag');
doctorName  := of_get_xmlval(axmlparm, 'doctorName');
payAgentId  := of_get_xmlval(axmlparm, 'payAgentId');
deptId      := of_get_xmlval(axmlparm, 'deptId');
payTime     := of_get_xmlval(axmlparm, 'payTime');
mchId       := of_get_xmlval(axmlparm, 'mchId');
payAppId    := of_get_xmlval(axmlparm, 'payAppId');
patMobile   := of_get_xmlval(axmlparm, 'patMobile');
operMobile  := of_get_xmlval(axmlparm, 'operMobile');
tsBeginTime := of_get_xmlval(axmlparm, 'beginTime');
tsEndTime   := of_get_xmlval(axmlparm, 'endTime');
payMethod   := of_get_xmlval(axmlparm, 'payMode');

insurPersonalId := of_get_xmlval(axmlparm, 'insurPersonalId');
insurCardNo      := of_get_xmlval(axmlparm, 'insurCardNo');
insurCardClass   := of_get_xmlval(axmlparm, 'insurCardClass');
insurMedicalClassName := of_get_xmlval(axmlparm,
                                         'insurMedicalClassName');
insurClassType   := of_get_xmlval(axmlparm, 'insurClassType');
insurReturnid    := of_get_xmlval(axmlparm, 'insurReturnid');
insurSettleId    := of_get_xmlval(axmlparm, 'insurSettleId');
insurBegfee      := of_get_xmlval(axmlparm, 'insurBegfee');
insurCardAmount  := of_get_xmlval(axmlparm, 'insurCardAmount');
insurMxbTclj     := of_get_xmlval(axmlparm, 'insurMxbTclj');
insurMxbNllj     := of_get_xmlval(axmlparm, 'insurMxbNllj');
insurOvAreaCode  := of_get_xmlval(axmlparm, 'insurOvAreaCode');
insurAreaCode    := of_get_xmlval(axmlparm, 'insurAreaCode');

```



```

insurBegfeeRemain := of_get_xmlval(axmlparm, 'insurBegfeeRemain');
samno             := of_get_xmlval(axmlparm, 'samno');
insurNo           := of_get_xmlval(axmlparm, 'insurNo');

```

```

n_insurBegfee     := to_number(insurBegfee);
n_insurCardAmount := to_number(insurCardAmount);
n_insurMxbTclj    := to_number(insurMxbTclj);
n_insurMxbNllj    := to_number(insurMxbNllj);
n_insurBegfeeRemain := to_number(insurBegfeeRemain);

```

```

d_reg_date := to_date(regDate, 'yyyy-mm-dd');
-- 挂号费+诊查费
n_totalFee := to_number(fee) + to_number(treatfee);
-- 支付金额
n_payMoney := to_number(payAmount);

```

-- 获取并校验患者信息

```

select b.patient_id, --病人标识号
       b.id_no, --身份证
       b.inp_no, --住院号
       b.date_of_birth, --出生日期
       b.name, --姓名
       b.sex, --性别
       b.name_phonetic, --姓名拼音
       b.identity, --身份
       b.charge_type, --费别
       b.unit_in_contract, --合同单位
       b.phone_number_home

```

```

into v_pid,
    v_pIdNo,
    v_pInHosNo,
    d_pBirth,
    v_pName,
    v_pSex,
    v_pNamePhonetic,
    v_pIdentity,

    v_pChargeType,
    v_pUnitInContract,
    v_pMobie
from MEDREC.pat_master_index b
where b.patient_id = patMedNo;
if v_pid = '' or v_pid is null then
    on_return := -1;
    on_message := '不存在患者信息';
    goto end_program;
end if;

-- 校验挂号规则
if check_reg_rule(patMedNo, doctorName, scheduleFlag, d_reg_date) = '1' then
    on_return := -1;
    on_message := '男性患者不能挂妇科或产科';
    goto end_program;
elsif check_reg_rule(patMedNo, doctorName, scheduleFlag, d_reg_date) = '2' then
    on_return := -1;
    on_message := '14 岁(含)以下请挂儿科';
    goto end_program;

```

```

end if;

--是否存在相同一笔交易
select count(1)
into n_translog_count
from wxgh.WX_TRANS_HIS_LOG t
where t.order_id = orderId
and t.order_id_pay = orderPayId;
if n_translog_count > 0 then
on_return := -1;
on_message := '订单号' || orderId || '的交易已经存在';
goto end_program;
end if;

-- 校验号别 clinic_label 是否存在
SELECT count(1)
into n_registcount_ck
FROM OUTPADM.clinic_for_regist
WHERE CLINIC_DATE = d_reg_date
and clinic_label = doctorName
and time_desc = scheduleFlag
and time_start = to_date(tsBeginTime, 'yyyy-mm-dd hh24:mi:ss')
and time_end = to_date(tsEndTime, 'yyyy-mm-dd hh24:mi:ss');
if n_registcount_ck < 1 then
on_return := -1;
on_message := '挂号类别不存在';
goto end_program;
end if;

```

--查询剩余号源数

```
SELECT nvl(t.registration_limits, 0) - nvl(t.current_no, 0) + 1
into n_regnum
FROM OUTPADM.clinic_for_regist t
WHERE CLINIC_DATE = d_reg_date
and clinic_label = doctorName
and time_desc = scheduleFlag
and time_start = to_date(tsBeginTime, 'yyyy-mm-dd hh24:mi:ss')
and time_end = to_date(tsEndTime, 'yyyy-mm-dd hh24:mi:ss');
if n_regnum <= 0 then
on_return := -1;
on_message := '挂号号源数已满';
goto end_program;
end if;
```

--校验号别挂号费是否维护

```
SElect count(1)
into n_regcount
FROM OUTPADM.CLINIC_INDEX, COMM.CLINIC_TYPE_DICT
WHERE (CLINIC_INDEX.CLINIC_TYPE = CLINIC_TYPE_DICT.CLINIC_TYPE)
and clinic_index.clinic_label = doctorName;
if n_regcount < 1 then
on_return := -1;
on_message := '挂号费未维护';
goto end_program;
end if;
```

--获取挂号费

```
SElect distinct a.CLINIC_TYPE,
```

```
((select price
  from (select t.clinic_type, x.price, t.charge_item
        from COMM.CLINIC_TYPE_SETTING t
        left join price_list x
        on t.price_item = x.item_code
        and x.stop_date is null) c
 where c.clinic_type = b.clinic_type
 and c.charge_item = '挂号费'
 and rownum = 1) +
```

```
(select price
  from (select t.clinic_type, x.price, t.charge_item
        from COMM.CLINIC_TYPE_SETTING t
        left join price_list x
        on t.price_item = x.item_code
        and x.stop_date is null) c
 where c.clinic_type = b.clinic_type
 and c.charge_item = '诊疗费'
 and rownum = 1)) * 100 as price,
```

```
(select price
  from (select t.clinic_type, x.price, t.charge_item
        from COMM.CLINIC_TYPE_SETTING t
        left join price_list x
        on t.price_item = x.item_code
        and x.stop_date is null) c
 where c.clinic_type = b.clinic_type
 and c.charge_item = '挂号费'
 and rownum = 1) as regamount,
```

```
(select price
  from (select t.clinic_type, x.price, t.charge_item
```

```

        from COMM.CLINIC_TYPE_SETTING t
        left join price_list x
            on t.price_item = x.item_code
        and x.stop_date is null) c
    where c.clinic_type = b.clinic_type
    and c.charge_item = '诊疗费'
    and rownum = 1) as treatamount
into v_clinic_type, n_regFee, n_reg, n_treat
FROM OUTPADM.CLINIC_INDEX a, COMM.CLINIC_TYPE_SETTING b
WHERE a.clinic_type = b.clinic_type
    and a.clinic_label = doctorName;

if n_regFee is null then
    on_return := -1;
    on_message := '挂号费金额获取出错';
    goto end_program;
end if;

-- 校验 HIS 挂号金额与传入挂号金额
if n_regFee <> n_totalFee then
    on_return := -1;
    on_message := 'HIS 挂号费金额与挂号费合计金额不一致';
    goto end_program;
end if;

-- 校验 HIS 挂号金额与支付金额
if n_regFee <> n_payMoney then
    on_return := -1;

```

```
on_message := '支付金额与挂号费金额不一致';  
goto end_program;  
end if;
```

```
-- 取操作工号
```

```
select count(1)  
into n_opera_sum  
from COMM.staff_dict a  
where 1 = 1 --job = '自助收款'  
and user_name = upper(payAgentId);  
if n_opera_sum > 0 then  
select a.id, a.name  
into v_opera_no, v_opera_name  
from COMM.staff_dict a  
where 1 = 1 --job = '自助收款'  
and user_name = upper(payAgentId)  
and rownum = 1;  
else  
v_opera_no := upper(payAgentId);  
v_opera_name := upper(payAgentId);  
end if;
```

```
select count(1)  
into n_no_admit_visit  
from OUTPADM.clinic_master v  
where v.patient_id = v_pid  
-- and v.clinic_label = doctorName  
and v.visit_dept = deptId  
and v.visit_date = d_reg_date
```

```

        -- and v.visit_time_desc = scheduleFlag
        and v.returned_date is null
        and v.registration_status <> 2;
if n_no_admit_visit > 0 then
    on_return := -1;
    on_message := '当天存在相同科室的未就诊记录，不允许重复挂号';
    goto end_program;
end if;

-- 取排队号
select nvl(t.current_no, 1)
into n_serial_no
from OUTPADM.clinic_for_regist t
where t.clinic_date = d_reg_date
    and t.clinic_label = doctorName
    and t.time_desc = scheduleFlag
    and time_start = to_date(tsBeginTime, 'yyyy-mm-dd hh24:mi:ss')
    and time_end = to_date(tsEndTime, 'yyyy-mm-dd hh24:mi:ss');

-- 取就诊序号
SELECT OUTPADM.visit_no.NEXTVAL INTO n_visit_seq FROM dual;

-- 构造门诊流水号
v_visit_prefix := to_char(d_reg_date, 'yyyymmdd');
v_visit_suffix := to_char(n_visit_seq);
v_visit_no := v_visit_prefix || v_visit_suffix;

--系统时间
select sysdate into d_now from dual;

```


--计算患者年龄

n_pAge := floor((d_now - d_pBirth) / 365);

--取挂号医生

```
select d.doctor,d.clinic_position
into v_reg_doctor,v_printMsg
FROM OUTPADM.CLINIC_INDEX d
where d.clinic_label = doctorName
and d.clinic_dept = deptId
and d.clinic_type = v_clinic_type
and rownum = 1;
```

--设置挂号就诊记录对象

```
r_clinic_master.visit_date      := d_reg_date;
r_clinic_master.visit_no       := n_visit_seq;
r_clinic_master.clinic_label    := doctorName;
r_clinic_master.visit_time_desc := scheduleFlag;
r_clinic_master.serial_no       := n_serial_no;
r_clinic_master.patient_id      := v_pid;
r_clinic_master.name            := v_pName;
r_clinic_master.name_phonetic   := v_pNamePhonetic;
r_clinic_master.sex             := v_pSex;
r_clinic_master.age             := n_pAge;
r_clinic_master.identity        := v_pidentity;
r_clinic_master.charge_type     := '自费';--v_pChargeType;
r_clinic_master.unit_in_contract := v_pUnitInContract;
r_clinic_master.visit_dept      := deptId;
r_clinic_master.doctor          := v_reg_doctor;
```

```

r_clinic_master.mr_provide_indicator := 0;
r_clinic_master.registering_date    := d_now;
r_clinic_master.regist_fee          := 0; --n_reg;
r_clinic_master.clinic_fee          := 0; --n_treat;
r_clinic_master.other_fee           := 0;
r_clinic_master.operator             := v_opera_no;
r_clinic_master.clinic_charge        := 0; --(n_payMoney / 100);
r_clinic_master.pay_way              := '微信预约';
r_clinic_master.mode_code            := 'J';
r_clinic_master.registration_status := 1;
r_clinic_master.clinic_no            := v_visit_no;
r_clinic_master.CLINIC_TYPE          := v_clinic_type;
r_clinic_master.card_name            := '5';

r_clinic_master.REGISTRATION_STATUS := '1';

SELECT F_TRANS_PINYIN_CAPITAL(v_pName) into v_simcode FROM DUAL;
r_clinic_master.simcode              := v_simcode;
r_clinic_master.phone_number_home := nvl(v_pMobie,
                                         nvl(patMobile, operMobile));

r_clinic_master.time_start := to_date(tsBeginTime,
                                       'yyyy-mm-dd hh24:mi:ss');
r_clinic_master.time_end   := to_date(tsEndTime,
                                       'yyyy-mm-dd hh24:mi:ss');

-- 更新门诊号表
begin
    update OUTPADM.clinic_for_regist

```

```

    Set current_no      = nvl(current_no, 0) + 1,
        registration_num = nvl(registration_num, 0) + 1
Where clinic_date = d_reg_date
And clinic_label = doctorName
And time_desc = scheduleFlag
and time_start = to_date(tsBeginTime, 'yyyy-mm-dd hh24:mi:ss')
and time_end = to_date(tsEndTime, 'yyyy-mm-dd hh24:mi:ss');
exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '更新门诊号表失败:' || SQLERRM;
    goto end_program;
end;

-- 更新请求提供病案队列
begin
    update OUTPADM.mr_request_queue
        set destination_dept = deptId,
            registration_date = sysdate,
            mr_location      = '4007', --写死
            clinic_label     = doctorName,
            serial           = 1, --写死
            time_desc        = scheduleFlag
        where patient_id = v_pid;
exception
when others then
    ROLLBACK;
    on_return := -1;

```

```

on_message := '更新请求提供病案队列失败:' || SQLERRM;
goto end_program;
end;

--收据号
v_daytime := to_char(d_now, 'yyyymmdd');

select nvl(max(substr(t.rcpt_no, 10)), 0)
into v_rcpt_seq
from outp_rcpt_master t
where t.rcpt_no like to_char(sysdate, 'yyyymmdd') || 'w%'
and t.rcpt_no not like to_char(sysdate, 'yyyymmdd') || 'wx%';
if v_rcpt_seq = 0 then
select lpad('1', 5, '0') into v_tmp_num from dual;
v_rcpt_nos := v_daytime || 'w' || v_tmp_num;
else
v_rcpt_seq := to_number(v_rcpt_seq) + 1;
select lpad(v_rcpt_seq, 5, '0') into v_tmp_num from dual;
v_rcpt_nos := v_daytime || 'w' || v_tmp_num;
end if;

r_master.name      := v_pName;
r_master.charge_type := v_pChargeType;
r_master.patient_id := v_pid;
r_master.name_phonetic := v_pNamePhonetic;
r_master.identity   := v_plidentity;

n_rcpt_costs := n_reg + n_treat;
n_rcpt_charges := (n_payMoney / 100);

```

```

r_master.rcpt_no      := v_rcpt_nos;
r_master.Visit_Date   := sysdate;
r_master.operator_no  := v_opera_no;
r_master.total_costs  := n_rcpt_costs; --(n_payamount / 100);
r_master.total_charges := n_rcpt_charges; --(n_payamount / 100);
r_master.charge_indicator := 0;
r_master.order_id_pay  := orderPayId;

```

```

if payMethod = 'XSYB' then

```

```

    r_master.insur_personal_id      := insurPersonalId;
    r_master.insur_card_no          := insurCardNo;
    r_master.insur_card_class       := insurCardClass;
    r_master.insur_medical_class_name := insurMedicalClassName;
    r_master.insur_class_type       := insurClassType;
    r_master.insur_returnid         := insurReturnid;
    r_master.insur_settle_id        := insurSettleId;
    r_master.insur_ov_area_code     := insurOvAreaCode;
    r_master.insur_area_code        := insurAreaCode;
    r_master.samno                  := samno;
    r_master.insur_no               := insurNo;
    r_master.auto_flag              := '2';
    r_master.charge_type            := '两定医保';

```

```

    r_master.insur_begfee          := n_insurBegfee;
    r_master.insur_card_amount     := n_insurCardAmount;
    r_master.insur_mxb_tclj       := n_insurMxbTclj;
    r_master.insur_mxb_nllj       := n_insurMxbNllj;
    r_master.insur_begfee_remain := n_insurBegfeeRemain;

```

```

else
    r_master.samno := orderId;
end if;

begin
    -- 插入门诊医疗收据记录
    INSERT INTO OUTPBILL.outp_rcpt_master VALUES r_master;
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '插入门诊医疗收据记录出错' || '['当前收据号:' || v_rcpt_nos || ']' ||
            SQLERRM;
        goto end_program;
end;

SELECT VISIT_NO.NEXTVAL into n_new_visit_no FROM DUAL; --注意权限问题
r_order.visit_date := d_now;
r_order.visit_no := n_new_visit_no;
r_order.patient_id := v_pid;

r_order.ordered_by_dept := r_clinic_master.visit_dept;

if v_reg_doctor <> null and v_reg_doctor <> '' then
    select s.user_name
        into v_dr_name
        from comm.staff_dict s
        where s.user_name = r_clinic_master.doctor;
    r_order.ordered_by_doctor := v_dr_name;

```

```

else

    r_order.ordered_by_doctor := v_reg_doctor;
end if;

r_order.rcpt_no      := v_rcpt_nos;
r_order.visit_id     := r_clinic_master.visit_no;
r_order.clinic_no     := r_clinic_master.clinic_no;
r_order.presc_indicator := 0;

--插入开单记录
begin
    INSERT INTO OUTPBILL.outp_order_desc VALUES r_order;
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '插入开单记录出错' || SQLERRM;
        goto end_program;
end;

v_n := 0;
for v_reg_price in c_reg_price loop

    SELECT PRICE_LIST.CLASS_ON_OUTP_RCPT,
           price_list.class_on_reckoning,
           price_list.subj_code,
           price_list.price
    into v_class_on_outp_rcpt,
        v_class_on_reckoning,

```

```

        v_subj_code,
        n_item_price
FROM PRICE_LIST
WHERE PRICE_LIST.ITEM_CLASS = v_reg_price.item_class
      AND PRICE_LIST.ITEM_CODE = v_reg_price.item_code
      AND PRICE_LIST.ITEM_SPEC = v_reg_price.item_spec
      AND PRICE_LIST.UNITS = v_reg_price.units
      AND (SYSDATE > PRICE_LIST.START_DATE)
      AND (SYSDATE < PRICE_LIST.STOP_DATE OR
           PRICE_LIST.STOP_DATE IS NULL)
      and rownum = 1;

r_items.visit_date := d_now;
r_items.visit_no  := n_new_visit_no;
r_items.rcpt_no   := v_rcpt_nos;

r_items.ITEM_no := v_n + 1;

r_items.item_class := v_reg_price.item_class;
r_items.class_on_rcpt := v_CLASS_ON_OUTP_RCPT;
r_items.item_code   := v_reg_price.item_code;
r_items.item_name    := v_reg_price.item_name;
r_items.ITEM_SPEC    := v_reg_price.item_spec;
r_items.amount       := 1;
r_items.units        := v_reg_price.units;
r_items.performed_by := deptId;
r_items.costs        := v_reg_price.price;
r_items.charges       := v_reg_price.price;

```



```

--r_items.flag          := 0;
r_items.repetition      := 1;
r_items.class_on_reckoning := v_class_on_reckoning;
r_items.subj_code       := v_subj_code;
--r_items.price_quotiety := v_orders_cost.price_quotiety;
r_items.item_price := n_item_price;
--r_items.order_no      := v_orders_cost.order_no;
r_items.class_on_outp_rcpt := v_reg_price.class_on_outp_rcpt;
r_items.class_on_mr      := v_reg_price.class_on_mr;

--插入门诊病人诊疗费用项目
begin
    INSERT INTO OUTPBILL.OUTPUT_BILL_ITEMS VALUES r_items;
    -- INSERT INTO OUTPUT_BILL_ITEMS_bak VALUES r_items;
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '插入门诊病人诊疗费用项目出错' || SQLERRM;
        goto end_program;
end;
v_n := v_n + 1;

end loop;

if payMethod = 'XCXJF' or payMethod = '' or payMethod is null then
    v_payMethod := '小程序';
elseif payMethod = 'XSYB' then
    v_payMethod := '线上医保';

```

```

end if;

-- 写入挂号支付方式记录表

begin
    INSERT INTO OUTPADM.CLINIC_PAYMENTS_MONEY
        (visit_date,
         visit_no,
         PAYMENT_NO,
         MONEY_TYPE,
         PAYMENT_AMOUNT,
         refunded_amount)
    VALUES
        (d_reg_date, n_visit_seq, 1, v_payMethod, 0, 0);
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '写入挂号支付方式记录:' || SQLERRM;
        goto end_program;
end;

```

```

r_payments.rcpt_no      := v_rcpt_nos;
r_payments.payment_no   := 1;
r_payments.money_type   := v_payMethod;
r_payments.payment_amount := (n_payMoney / 100);
r_payments.refunded_amount := 0;

```

--插入门诊病人支付方式记录

```

begin

```

```

INSERT INTO OUTPBILL.outp_payments_money VALUES r_payments;

exception

when others then

    ROLLBACK;

    on_return := -1;

    on_message := '插入门诊病人支付方式记录出错' || SQLERRM;

    goto end_program;

end;

```

-- 写入门诊就诊记录

```

begin

    INSERT INTO OUTPADM.clinic_master VALUES r_clinic_master;

exception

when others then

    ROLLBACK;

    on_return := -1;

    on_message := '写入门诊就诊记录失败:' || SQLERRM;

    goto end_program;

end;

```

-- 写入微信交易日志表

```

begin

insert into wxgh.wx_trans_his_log

(PATIENT_ID,

PATIENT_MOBILE,

ORDER_TYPE,

ORDER_ID,

ORDER_ID_PAY,

DEAL_MONEY,

```

```

    PAY_APP_ID,
    MCH_ID,
    ORDER_PAY_TIME,
    ORDER_DESC,
    RCPT_NO,
    SYS_DATETIME,
    REG_RCPT_NO)
values
(v_pid,
nvl(patMobile, operMobile),
'REG',
orderId,
orderPayId,
payAmount,
payAppId,
mchId,
payTime,
v_payMethod,
v_visit_no,
sysdate,
v_rcpt_nos);
exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '写入微信交易日志表失败:' || SQLERRM;
    goto end_program;
end;
```

<<end_program>>

if on_return = 0 then

```
insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'orderId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
    'addRegOrder001',
    axmlParm,
    '结束 addRegOrder001-02' ||
    of_get_xmlval(axmlParm, 'orderId'),
    null,
    null);
```

open aCur_temp for

```
select to_char(on_return) as "returnCode",
    on_message as "resultDesc",
    t.clinic_no as "hisOrderId",
    t.visit_dept as "consultingRoomId",
    nvl(t.invoice_no, t.clinic_no || '-' || t.serial_no) as "receiptId",
    v_printMsg printMsg
from OUTPADM.clinic_master t
where t.clinic_no = v_visit_no;
```

else

```
insert_wx_webservice_err_logs('addRegOrder001' ||
    to_char(systimestamp,
        '-yyyymmddhh24missxff3'),
    'addRegOrder001',
    axmlParm,
```

```

        ",
        on_message);

insert_wx_ps_step_log(of_get_xmlval(axmlParm, 'orderId') || '-' ||
        to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
        'addRegOrder001',
        axmlParm,
        '[错误]结束 addRegOrder001-02' ||
        of_get_xmlval(axmlParm, 'orderId'),
        null,
        null);

open aCur_temp for
select to_char(on_return) as "returnCode",
        on_message as "resultDesc",
        " as "hisOrderId",
        " as "consultingRoomId",
        v_rcpt_nos as "receiptId",
        " printMsg
from dual;
end if;

end;

-- 5.1.1 Unpaid Medical Visit Record Query Interface 001 待缴费就诊记录查询接口 001
procedure getNoPayVisitInfo001(hospitalId in varchar2, -- 医院 ID
        patId in varchar2, -- 患者主索引 id
        opPatId in varchar2, -- 门诊病人 id

```

```

        patMedType in varchar2, -- 患者诊疗卡类型
        patMedNo in varchar2, -- 患者诊疗卡号码
        beginDate in varchar2, -- 查询开始日期，格式：YYYY-MM-DD
        endDate in varchar2, -- 查询结束日期，格式：YYYY-MM-DD
        aCur_temp out sys_refcursor) is

/* 输出结果
* 字段      描述
* -----
patName      患者姓名
unconfirmedPayAmount  未审核处方总金额
hospitalId    院区 Id
visitId       HIS 就诊流水号（HIS 系统中就诊流水号（内部 ID），要求唯一。如：
12345）
visitNo       HIS 流水号备注说明（HIS 系统中就诊流水号的显式信息，如：2012-3-
1/123 代表挂号日期为 2012-3-1，流水号为 123）
deptId        接诊科室 ID（二级）
deptName      接诊科室名称（二级）
doctorId      接诊医生 ID
doctorName    接诊医生姓名
payAmount     未缴费总金额，单位：分
checkPresAmount  未审方金额
settleMethodId  结算方式 ID
settleMethodName  结算方式名称
medicareCardPayFlag  医保卡个账支付标识（0-为可用个账支付，1-不可用个账支
付）
*/

begin

```

```

open aCur_temp for

SELECT c.name as "patName", -- 患者姓名
      " as "unconfirmedPayAmount",
      (select mr_hospital_code from COMM.Hospital_Config) as "hospitalId", -- 院区 Id
      a.clinic_no as "visitId", -- HIS 就诊流水号
      --a.visit_no as "visitNo", -- HIS 流水号备注说明 ----modify 20230914
      --a.visit_dept
      b.ordered_by as "deptId", -- 接诊科室 ID
      f.dept_name || '(' ||
      substr(e.clinic_type, 1, instr(e.clinic_type, '号', 1) - 1) || ')' as "deptName", -- 接诊科室名称
      /* b.doctor_no as "doctorId", -- 接诊医生 ID ----modify 20230914
      (select s.name
      from comm.staff_dict s
      where s.user_name = b.doctor_no) as "doctorName", -- 接诊医生姓名 */
      sum(round(d.costs, 2)) * 100 as "payAmount", -- 未缴费总金额，单位：分
      " as "checkPresAmount", -- 未审方金额
      " as "settleMethodId", -- 结算方式 ID
      " as "settleMethodName", -- 结算方式名称
      " as "medicareCardPayFlag" -- 医保卡个账支付标识（0-为可用个账支付，1-不可用个账支付）
      -- " as "medicareCardPayFlag"--审核通过未通过标志

FROM OUTPADM.CLINIC_MASTER    a,
     OUTPDOCT.OUTP_ORDERS    b,
     OUTPDOCT.OUTP_ORDERS_COSTS d,
     MEDREC.Pat_Master_Index  c,
     OUTPADM.CLINIC_INDEX     e,
     COMM.dept_dict           f
WHERE (b.VISIT_DATE = a.VISIT_DATE)

```



```

and (b.VISIT_NO = a.VISIT_NO)
and (b.patient_id = c.patient_id)
and (a.clinic_label = e.clinic_label)
    --and e.clinic_dept
and b.ordered_by = f.dept_code
and a.Visit_Date >= to_date(beginDate, 'yyyy-mm-dd')
    --and a.Visit_Date <= to_date(endDate, 'yyyy-mm-dd')
and a.Visit_Date <=
    to_date(to_char(sysdate + 7, 'yyyy-mm-dd'), 'yyyy-mm-dd')
and a.visit_date > sysdate - 3
and d.order_class in
    ('A', 'B', 'C', 'Z', 'E', 'F', 'G', 'H', 'J', 'D', 'I')
and d.serial_no = b.SERIAL_NO
and a.patient_id = patMedNo
and d.charge_indicator = '0'
and b.rcpt_no is null
group by b.PATIENT_ID,
    c.name,
    a.clinic_no,
    -- a.visit_no, ----modify 20230914
    --a.visit_dept,
    b.ordered_by,
    e.clinic_type,
    -- b.doctor_no,
    -- b.DOCTOR,
    f.dept_name--,D.HOSP_APPR_FLAG
order by a.clinic_no;-- desc, a.visit_no desc; ----modify 20230914

end;

```

-- 5.1.2 Unpaid Detail Record Query Interface 001 待缴费明细记录查询接口 001

procedure getNoPayDetailInfo001(hospitalId in varchar2, -- (必传) 医院 ID

 patId in varchar2, -- 患者主索引 id

 opPatId in varchar2, -- 门诊病人 id

 visitId in varchar2, -- (必传) HIS 就诊流水号

 deptId in varchar2,

 doctorId in varchar2,

 aCur_temp out sys_refcursor) is

/* 输出结果

* 字段 描述

* -----

patName 患者姓名

deptName 接诊科室名称（二级）

doctorName 接诊医生姓名

payAmount 未缴费总金额，单位：分

detailTypeId 费用类别 Id

detailFee 缴费费别

detailId 缴费细目流水号，要求唯一

detailName 缴费细目名称

detailCount 缴费细目数量

detailUnit 缴费细目单位

detailAmount 缴费细目金额，单位：分

detailSpec 缴费规格

detailPrice 缴费细目单价，单位：分

*/

begin

```

open aCur_temp for

SELECT c.name as "patName", -- 患者姓名
      f.dept_name || '(' ||
      substr(e.clinic_type, 1, instr(e.clinic_type, '号', 1) - 1) || ')' as "deptName", -- 接诊科室名
称
/* (select s.name
   from comm.staff_dict s
  where s.user_name = b.doctor_no) as "doctorName", -- 接诊医生姓名*/ ----modify
20230914

```

```

(select payAmount
 from (SELECT a1.clinic_no, -- HIS 就诊流水号
            sum(round(d1.costs, 2)) * 100 as payAmount -- 未缴费总金额，单位：分
      FROM OUTPADM.CLINIC_MASTER    a1,
            OUTPDOCT.OUTP_ORDERS    b1,
            OUTPDOCT.OUTP_ORDERS_COSTS d1
     WHERE (b1.VISIT_DATE = a1.VISIT_DATE)
           and (b1.VISIT_NO = a1.VISIT_NO)
           and (b1.VISIT_NO = d1.VISIT_NO)
           and (b1.serial_no = d1.serial_no)
           and d1.order_class in ('A',
                                   'B',
                                   'C',
                                   'Z',
                                   'E',
                                   'F',
                                   'G',
                                   'H',
                                   'J',

```

```

        'D',
        'I')
    and a1.clinic_no = visitId
    --and a1.visit_dept
    and b1.ordered_by = deptId
    -- and b1.doctor_no = doctorId ----modify 20230914
    and d1.charge_indicator = '0'
    and b1.rcpt_no is null
group by a1.clinic_no)) as "payAmount", -- 未缴费总金额，单位：分

d.item_class as "detailTypeId", -- 费用类别 Id
(case
    when d.item_class = 'A' then
        '西药费'
    when d.item_class = 'B' then
        '中药费'
    when d.item_class = 'C' then
        '检验费'
    when d.item_class = 'D' then
        '检查费'
    else
        '处置费'
end) as "detailFee", -- 缴费费别
d.serial_no as "detailId", -- 缴费细目流水号，要求唯一
d.item_name as "detailName", -- 缴费细目名称
d.amount as "detailCount", -- 缴费细目数量
d.units as "detailUnit", -- 缴费细目单位
round(d.costs, 2) * 100 as "detailAmount", -- 缴费细目金额，单位：分

```

```

d.item_spec as "detailSpec", -- 缴费规格
round(d.costs / ((case
    when d.item_class = 'B' then
        --decode(nvl(d.repetition, 1), 0, 1, nvl(d.repetition, 1))
        1
    else
        1
    end) * d.amount),
4) * 100 as "detailPrice", -- 缴费细目单价，单位：分
d.item_code as "detailCode", -- 缴费细目编码
-- (select T.price from CURRENT_PRICE_LIST t where T.item_class='A' AND
T.item_code=d.item_code and T.item_spec=d.item_spec and t.units=d.units )
" actPurcPric,
D.HOSP_APPR_FLAG as "hospApprFlag"
FROM OUTPADM.CLINIC_MASTER    a,
    OUTPDOCT.OUTP_ORDERS    b,
    OUTPDOCT.OUTP_ORDERS_COSTS d,
    MEDREC.Pat_Master_Index  c,
    OUTPADM.CLINIC_INDEX    e,
    COMM.dept_dict          f
WHERE (b.VISIT_DATE = a.VISIT_DATE)
and (b.VISIT_NO = a.VISIT_NO)
and (b.patient_id = c.patient_id)
and (a.clinic_label = e.clinic_label)
    --and e.clinic_dept
and b.ordered_by = f.dept_code
and d.order_class in
    ('A', 'B', 'C', 'Z', 'E', 'F', 'G', 'H', 'J', 'D', 'I')
and d.serial_no = b.SERIAL_NO

```

```

and a.visit_date > sysdate - 3
and a.clinic_no = visitId
    --and a.visit_dept
and b.ordered_by = deptId
-- and b.doctor_no = doctorId ----modify 20230914
and d.charge_indicator = '0'
and b.rcpt_no is null;
end;

-- 5.1.3 Unpaid Record Payment Interface 001 待缴费记录支付接口 001
procedure payChgOrder001(as_xmlParm in varchar2,
    aCur_temp out sys_refcursor,
    on_return out number, -- 结果编码
    on_message out varchar2 -- 结果描述
) is

/* 输出结果
* 字段      描述
* -----
on_return    结果代码

```

on_message 结果描述
 bindChannels 就诊卡已绑定渠道，多个渠道用英文分号分割
 patMedNo 就诊卡号
 hisOrderId HIS 缴费订单号（缴费成功时返回 HIS 系统生成的缴费订单，要求唯一）
 orderDesc 订单相关信息描述
 printMsg 待打印信息
 executePlace 诊察室代码
 receiptId 收据 ID
 */
 -- on_return number; -- 结果编码
 -- on_message varchar2(100); --结果描述

 patMedNo varchar2(20); --患者诊疗卡号码
 visitId varchar2(20); --HIS 就诊流水号
 orderPayId varchar2(50); --第三方平台的流水号
 orderId varchar2(50); --渠道订单号
 payAgentId varchar2(10); --虚拟操作员工号
 payAmount varchar2(10); --支付金额(单位“分”)
 payTime varchar2(20); --交易时间，格式：YYYY-MM-DD HI24:MI:SS
 payAppId varchar2(20); -- 公众账号 ID
 mchId varchar2(20); -- 商户号。
 deptId varchar2(20);
 doctorId varchar2(20);

 payMethod varchar2(10);
 v_payMethod varchar2(20);
 v_enteredBy varchar2(20);

```

insurPersonalId  VARCHAR2(50); --医保个人编号/帐号
insurCardNo      VARCHAR2(50); --医保卡号
insurCardClass   VARCHAR2(10); --职工居民标志--新医保接口是人员类别 从 1 增加到 10
rcpt 表只有 1
v_INSUR_CARD_CLASS VARCHAR2(1);
insurMedicalClassName VARCHAR2(50); --医保病人医疗类别
insurClassType    VARCHAR2(10); --医保病人类型
insurReturnid     VARCHAR2(50); --医保病人登记流水号
insurSettleId     VARCHAR2(50); --医保病人结算流水号
insurOvAreaCode   VARCHAR2(10); --统筹区号
insurAreaCode     VARCHAR2(10); --区域编号
samno            VARCHAR2(30); --小程序对账订单号
insurNo          VARCHAR2(30); --医保对账订单号

insurBegfee  VARCHAR2(30); --医保病人起付线
insurCardAmount VARCHAR2(30); --医保病人消费前卡余额
-- insurMxbTclj  VARCHAR2(30); --医保病人统筹年度累计
-- insurMxbNllj  VARCHAR2(30); --医保病人年度累计
-- insurBegfeeRemain VARCHAR2(30); --医保病人起付线剩余

subsidyMethod  varchar2(10); --公务补贴方式
selfAccountMoney varchar2(10); --医保个账金额
subsidyMoney   varchar2(10); --公务补贴金额

n_selfAccountMoney number; --医保个账金额
n_subsidyMoney     number; --公务补贴金额

-- n_insurBegfee    NUMBER(12, 4); --医保病人起付线

```


n_insurCardAmount NUMBER(12, 4); --医保病人消费前卡余额
 -- n_insurMxbTclj NUMBER(12, 4); --医保病人统筹年度累计
 -- n_insurMxbNllj NUMBER(12, 4); --医保病人年度累计
 -- n_insurBegfeeRemain NUMBER(12, 4); --医保病人起付线剩余

d_now date; --当前系统时间

d_pBirth date; --出生日期

v_pid VARCHAR2(20); --患者 ID
 v_pIdNo VARCHAR2(18); --身份证号
 v_pInHosNo VARCHAR2(10); --住院号
 v_pName VARCHAR2(20); --患者姓名
 v_pSex VARCHAR2(4); --患者性别
 v_pNamePhonetic varchar2(40);
 v_pIdentity varchar2(40);
 v_pChargeType varchar2(40);
 v_pUnitInContract varchar2(40);
 v_opera_no varchar2(20); --操作工号
 v_opera_name varchar2(20); --操作人
 v_class_on_outp_rcpt VARCHAR2(20);
 v_seriano VARCHAR2(20); --流水号

n_translog_count number; --交易记录校验结果标识

v_rcpt_nos varchar2(20); --收据号

n_total_amount number; --his 待收费金额

n_payamount number; --支付金额

n_opera_sum number; --操作员记录校验标识

v_n number(5);

v_app_no number(5);

v_lab_no number(5);

v_day varchar2(8);

v_daytime varchar2(14);

n_new_visit_no number(5);

n_presc_count number(5);

v_rcpt_seq varchar2(10);

v_tmp_num varchar2(5);

v_status varchar2(1);

n_presc_master_count number;

n_presc_master_disp_count number;

n_exam_costs number(9, 2);

n_exam_charges number(9, 2);

n_lab_costs number(9, 2);

n_lab_charges number(9, 2);

v_class_on_reckoning varchar2(10);

v_subj_code varchar2(4);

n_item_price number(14, 4);

n_rcpt_costs number(8, 2);

n_rcpt_charges number(8, 2);

```
v_rcpt_str varchar2(2000);  
v_drug_rcpt_no varchar2(20);
```

```
v_drup_presc_tmp varchar2(20);  
n_presc_detail_count number;  
n_presc_detail_disp_count number;  
v_tmp_presc_no varchar2(10);
```

```
v_presc_visit_no varchar2(10);
```

```
v_administration varchar2(100);
```

```
-----  
  
/*v_mdtrt_id varchar2(30);--就诊 ID 字符型 30 Y  
v_setl_id varchar2(30);--结算 id  
v_psn_no varchar2(30);--人员编号 字符型 30 Y  
v_psn_name varchar2(50);--人员姓名 字符型 50 Y  
v_psn_cert_type varchar2(6);--人员证件类型 字符型 6 Y Y  
v_certno varchar2(50);--证件号码 字符型 50 Y  
v_gend varchar2(6);--性别 字符型 6 Y  
v_naty varchar2(3);--民族 字符型 3 Y  
v_brdy date;--出生日期 日期型 yyyy-MM-dd  
v_age number(4,1);--年龄 数值型 4,1  
v_insutype varchar2(6);--险种类型 字符型 6 Y  
v_psn_type varchar2(6);--人员类别 字符型 6 Y Y  
v_cvlserv_flag varchar2(3);--公务员标志 字符型 3 Y Y  
v_setl_time date;--结算时间
```

v_mdtrt_cert_type varchar2(3);--就诊凭证类型 字符型 3 Y
 v_med_type varchar2(6);--医疗类别 字符型 6 Y Y
 v_medfee_sumamt number(8,2);--医疗费总额 数值型 16,2 Y
 v_fulamt_ownpay_amt number(8, 2);-- 全自费金额 数值型 16,2 Y
 v_overlmt_selfpay number(8, 2);--超限价自费费用 数值型 16,2 Y
 v_preselfpay_amt number(8, 2);--先行自付金额 数值型 16,2 Y
 v_inscp_scp_amt number(8, 2);--符合政策范围金额 数值型 16,2 Y
 v_act_pay_dedc number(8, 2) ;--实际支付起付线 数值型 16,2
 v_hifp_pay number(8, 2);--基本医疗保险统筹基金支出数值型 16,2 Y
 v_pool_prop_selfpay number(8, 2);-- 基本医疗保险统筹基金 支付比例 数值型 5,4
 Y
 v_cvlserv_pay number(8, 2); --公务员医疗补助资金支出 数值型 16,2 Y
 v_hifes_pay number(8, 2);--企业补充医疗保险基金支出数值型 16,2 Y
 v_hifmi_pay number(8, 2);--居民大病保险资金支出 数值型 16,2 Y
 v_hifob_pay number(8, 2);--职工大额医疗费用补助基金支出 数值型 16,2 Y
 v_maf_pay number(8, 2);--医疗救助基金支出 数值型 16,2 Y
 v_oth_pay number(8, 2);--其他支出 数值型 16,2 Y
 v_fund_pay_sumamt number(8, 2);-- 基金支付总额 数值型 16,2 Y
 v_psn_part_amt number(8, 2);--个人负担总金额 数值型 16,2 Y
 v_acct_pay number(8, 2);--个人账户支出 数值型 16,2 Y
 v_psn_cash_pay number(8, 2);--个人现金支出 数值型 16,2 Y
 v_hosp_part_amt number(8, 2);-- 医院负担金额 数值型 16,2 Y
 v_balc number(16, 2);--余额 数值型 16,2 YXJ-K01-2020
 v_acct_mulaid_pay number(8, 2);--个人账户共济支付金额 数值型 16,2 Y
 v_medins_setl_id number(8, 2);--医药机构结算 ID 字符型 30 Y 存放发送方报文 ID
 v_clr_optins number(8, 2);--清算经办机构 字符型 6
 v_clr_way number(8, 2);--清算方式 字符型 6 Y
 v_clr_type number(8, 2);--清算类别 字符型 6 Y Y

*/

-----新增医保结算返回字段

```
r_master      outp_rcpt_master%ROWTYPE; -- 结算主表
r_order       outp_order_desc%ROWTYPE;
r_items       outp_bill_items%ROWTYPE;
r_payments    outp_payments_money%ROWTYPE;
r_drug_presc_master PHARMACY.DRUG_PRESC_MASTER_TEMP%ROWTYPE;
r_drug_presc_detail PHARMACY.Drug_Presc_Detail_Temp%ROWTYPE;
```

t_ld_outp_insur_settle ldinsur.ld_outp_insur_settlement%ROWTYPE; --新增医保表 赵志刚
20211021

```
v_cp          number;
v_later_paysumamt outp_payments_money.payment_amount%type;
v_later_ACCTPAY outp_payments_money.payment_amount%type;
n_rcpt_costs_one number(8, 2);
n_rcpt_costs_two number(8, 2);
n_rcpt_costs_three number(8, 2);
v_MoneyTtype_one outp_payments_money.Money_Type%TYPE;
v_MoneyTtype_two outp_payments_money.Money_Type%TYPE;
v_MoneyTtype_three outp_payments_money.Money_Type%TYPE;
```

cursor c_order_desc is
select distinct a.patient_id,
 a.visit_no,
 a.serial_no,

```

        b.ordered_by,
        b.doctor_no,
        a.visit_date
from OUTPDOCT.outp_orders_costs a,
        OUTPDOCT.OUTP_ORDERS    b,
        OUTPADM.CLINIC_MASTER    c
where c.visit_date = b.visit_date
and c.visit_no = b.visit_no
and b.serial_no = a.serial_no
and c.clinic_no = visitId
        --and c.visit_dept
and b.ordered_by = deptId
-- and b.doctor_no = doctorId ----modify 20230914
and a.charge_indicator = '0'
and b.rcpt_no is null;

```

cursor c_order_cost is

```

select e.visit_no,
        e.item_class,
        e.item_no,
        e.item_name,
        e.item_code,
        e.item_spec,
        ((case
            when e.item_class = 'B' then
                nvl(e.repetition, 1)
            else
                1
            end) * e.amount) as amount,

```

```

        e.units,
        e.performed_by,
        round(e.costs, 2) as costs,
        round(e.charges, 2) as charges,
        e.order_no,
        e.repetition,
        e.price_quotient,
        e.serial_no,
        e.visit_date
from OUTPDOCT.outp_orders_costs e,
     OUTPDOCT.OUTP_ORDERS      b,
     OUTPADM.CLINIC_MASTER     c
where c.visit_date = b.visit_date
     and b.visit_no = c.visit_no
     and b.serial_no = e.serial_no
     and c.clinic_no = visitId
     --and c.visit_dept
     and b.ordered_by = deptId
     --and b.doctor_no = doctorId ----modify 20230914
     and e.charge_indicator = '0'
     and b.rcpt_no is null;

cursor c_drug_presc is
select c.visit_date,
       c.visit_no,
       c.serial_no,
       c.presc_no,
       c.item_no,
       c.drug_code,

```

```

        c.drug_name,
        c.firm_id,
        c.drug_spec,
        c.units,
        c.amount,
        c.costs,
        c.charges,
        c.administration,
        c.order_no,
        c.sub_order_no
    from clinic_master a, outp_orders b, outp_presc c
    where a.patient_id = b.patient_id
        and a.visit_date = b.visit_date
        and a.visit_no = b.visit_no
        and b.visit_date = c.visit_date
        and b.visit_no = c.visit_no
        and b.serial_no = c.serial_no
        and a.clinic_no = visitId
        and c.presc_no = v_tmp_presc_no
        and c.charge_indicator = '0';

begin
    --暂时注释
    insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
        to_char(systimestamp, 'yyyymmddhh24missxff3') || '-1',
        'payChgOrder001',
        as_xmlParm,
        '进入 payChgOrder001-01' ||
        of_get_xmlval(as_xmlParm, 'orderId'),

```



```

        on_return,
        on_message);

on_return := 0;
on_message := '成功';

patMedNo := of_get_xmlval(as_xmlParm, 'patMedNo');
visitId  := of_get_xmlval(as_xmlParm, 'visitId');
orderPayId := of_get_xmlval(as_xmlParm, 'orderPayId');
orderId  := of_get_xmlval(as_xmlParm, 'orderId');
payAgentId := of_get_xmlval(as_xmlParm, 'payAgentId');
payAmount := of_get_xmlval(as_xmlParm, 'payAmount');
payTime   := of_get_xmlval(as_xmlParm, 'payTime');
payAppId  := of_get_xmlval(as_xmlParm, 'payAppId');
mchId     := of_get_xmlval(as_xmlParm, 'mchId');
deptId    := of_get_xmlval(as_xmlParm, 'deptId');
doctorId  := of_get_xmlval(as_xmlParm, 'doctorId');
payMethod := of_get_xmlval(as_xmlParm, 'payMethod');

----注释的新医保接口里面有了
-- insurPersonalId := of_get_xmlval(as_xmlParm, 'insurPersonalId');
-- insurCardNo     := of_get_xmlval(as_xmlParm, 'insurCardNo');
insurCardClass     := of_get_xmlval(as_xmlParm, 'insurCardClass');
insurMedicalClassName := of_get_xmlval(as_xmlParm,
                                         'insurMedicalClassName');
insurClassType     := of_get_xmlval(as_xmlParm, 'insurClassType');
-- insurReturnid   := of_get_xmlval(as_xmlParm, 'insurReturnid');
-- insurSettleId    := of_get_xmlval(as_xmlParm, 'insurSettleId');----
-- insurBegfee      := of_get_xmlval(as_xmlParm, 'insurBegfee');

```

```

-- insurCardAmount    := of_get_xmlval(as_xmlParm, 'insurCardAmount');
-- insurMxbTclj       := of_get_xmlval(as_xmlParm, 'insurMxbTclj');
-- insurMxbNllj       := of_get_xmlval(as_xmlParm, 'insurMxbNllj');
-- insurOvAreaCode    := of_get_xmlval(as_xmlParm, 'insurOvAreaCode');
-- insurAreaCode      := of_get_xmlval(as_xmlParm, 'insurAreaCode');
-- insurBegfeeRemain   := of_get_xmlval(as_xmlParm, 'insurBegfeeRemain');
samno := of_get_xmlval(as_xmlParm, 'samno');
insurNo := of_get_xmlval(as_xmlParm, 'insurNo');

subsidyMethod := of_get_xmlval(as_xmlParm, 'subsidyMethod');

--selfAccountMoney := of_get_xmlval(as_xmlParm, 'selfAccountMoney'); --个人卡支付
--subsidyMoney      := of_get_xmlval(as_xmlParm, 'subsidyMoney'); --线上公务员补助

```

-----新增医保结算返回字段

```

-- n_insurBegfee      := to_number(insurBegfee);
-- n_insurCardAmount  := to_number(insurCardAmount);
--n_insurMxbTclj      := to_number(insurMxbTclj);
-- n_insurMxbNllj     := to_number(insurMxbNllj);
-- n_insurBegfeeRemain := to_number(insurBegfeeRemain);

```

```

n_payamount := to_number(payAmount);

```

```

n_selfAccountMoney := to_number(selfAccountMoney);
n_subsidyMoney      := to_number(subsidyMoney);

```

```

select t.patient_id
into patMedNo

```

```

from OUTPADM.CLINIC_MASTER t
where t.clinic_no = visitId;
if patMedNo = '' or patMedNo is null then
    on_return := -1;
    on_message := '就诊记录不存在';
    goto end_program;
end if;

```

-- 获取并校验患者信息

```

select b.patient_id, --病人标识号
       b.id_no, --身份证
       b.inp_no, --住院号
       b.date_of_birth, --出生日期
       b.name, --姓名
       b.sex, --性别
       b.name_phonetic, --姓名拼音
       b.identity, --身份
       b.charge_type, --费别
       b.unit_in_contract --合同单位
into v_pid,
     v_pIdNo,
     v_pInHosNo,
     d_pBirth,
     v_pName,
     v_pSex,
     v_pNamePhonetic,
     v_pIdentity,
     v_pChargeType,

```

```

        v_pUnitInContract
    from MEDREC.pat_master_index b
    where b.patient_id = patMedNo;
    if v_pid = '' or v_pid is null then
        on_return := -1;
        on_message := '不存在患者信息';
        goto end_program;
    end if;

    insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
        to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
        'payChgOrder001',
        as_xmlParm,
        '完成就诊记录校验及获取患者信息-02' ||
        of_get_xmlval(as_xmlParm, 'orderId'),
        on_return,
        on_message);

--是否存在相同一笔交易
select count(1)
into n_translog_count
from wxgh.WX_TRANS_HIS_LOG t
where t.order_id = orderId
    and t.order_id_pay = orderPayId;
if n_translog_count > 0 then
    on_return := -1;
    on_message := '订单号' || orderId || '的交易已经存在';
    goto end_program;
end if;

```

```

--获取系统时间

select sysdate into d_now from dual;


--获取未收费金额金额

select sum(round(nvl(d.costs, 0), 2)) * 100
into n_total_amount
from OUTPADM.CLINIC_MASTER    a,
     OUTPDOCT.OUTP_ORDERS    b,
     OUTPDOCT.OUTP_ORDERS_COSTS d
where b.VISIT_DATE = a.VISIT_DATE
and b.VISIT_NO = a.VISIT_NO
and b.VISIT_NO = d.VISIT_NO
and b.serial_no = d.serial_no
and a.patient_id = v_pid
and a.clinic_no = visitId
--and a.visit_dept
and b.ordered_by = deptId
-- and b.doctor_no = doctorId ----modify 20230914
and d.charge_indicator = '0'
and b.rcpt_no is null;

if n_total_amount is null then
on_return := -1;
on_message := '门诊费用获取出错';
goto end_program;
end if;

if n_total_amount <> n_payamount then

```

```

on_return := -1;

on_message := '支付金额与门诊处方费用不一致';

goto end_program;

end if;


insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-3',
    'payChgOrder001',
    as_xmlParm,
    '完成订单校验及获取金额信息和费用校验-03' ||
    of_get_xmlval(as_xmlParm, 'orderId'),
    on_return,
    on_message);

```

```

-- 取操作工号

select count(1)
into n_opera_sum
from COMM.staff_dict a
where 1 = 1 --job = '自助收款'
and user_name = upper(payAgentId);

if n_opera_sum > 0 then
    select a.id, a.name
    into v_opera_no, v_opera_name
    from COMM.staff_dict a
    where 1 = 1 --job = '自助收款'
    and user_name = upper(payAgentId)
    and rownum = 1;
else
    v_opera_no := upper(payAgentId);

```

```

v_opera_name := upper(payAgentId);
end if;

if payMethod = 'XCXJF' or payMethod = '' or payMethod is null then
    v_payMethod := '小程序';
    v_enteredBy := '小程序';
elseif payMethod = 'XSYB' then

    v_payMethod := '线上医保';
    v_enteredBy := '线上医保';

elseif payMethod = 'SCANPAY' then
    v_payMethod := '扫码支付';
    v_enteredBy := '扫码支付';
end if;

v_rcpt_str := '';

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-4',
    'payChgOrder001',
    as_xmlParm,
    '准备循环处方信息-04' ||
    of_get_xmlval(as_xmlParm, 'orderId'),
    on_return,
    on_message);

for v_orders_desc in c_order_desc loop

```

```

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-5',
    'payChgOrder001',
    as_xmlParm,
    '循环处方信息-05-' || v_orders_desc.serial_no || '-' ||
    of_get_xmlval(as_xmlParm, 'orderId'),
    on_return,
    on_message);

```

--收据号

```

/*v_daytime := to_char(d_now, 'yyyymmdd');

```

```

select nvl(max(substr(t.rcpt_no, 10)), 0)
into v_rcpt_seq
from outp_rcpt_master t
where t.rcpt_no like to_char(sysdate, 'yyyymmdd') || 'w%'
and t.rcpt_no not like to_char(sysdate, 'yyyymmdd') || 'wx%';
if v_rcpt_seq = 0 then
    select lpad('1', 5, '0') into v_tmp_num from dual;
    v_rcpt_nos := v_daytime || 'w' || v_tmp_num;
else
    v_rcpt_seq := to_number(v_rcpt_seq) + 1;
    select lpad(v_rcpt_seq, 5, '0') into v_tmp_num from dual;
    v_rcpt_nos := v_daytime || 'w' || v_tmp_num;
end if;*/
select to_char(sysdate, 'yymmddhh24miss') ||
    trunc(dbms_random.value(1000, 9999))
into v_rcpt_nos

```



```
from dual;
```

```
-- 获取就诊记录信息并设置到门诊收据记录参数对象
```

```
begin
```

```
select a.name, '自费', a.patient_id, a.name_phonetic, a.identity
into r_master.name,
      r_master.charge_type,
      r_master.patient_id,
      r_master.name_phonetic,
      r_master.identity
from clinic_master a
where a.patient_id = v_pid
      and a.clinic_no = visitId
      and a.visit_date > sysdate - 3
      and rownum = 1;
```

```
-- 设置到门诊收据记录参数对象
```

```
begin
```

```
select sum(round(e.costs, 2)), sum(round(e.charges, 2))
into n_rcpt_costs, n_rcpt_charges
from OUTPDOCT.outp_orders_costs e,
      OUTPDOCT.OUTP_ORDERS      b,
      OUTPADM.CLINIC_MASTER      c
where c.visit_date = b.visit_date
      and b.visit_no = c.visit_no
      and b.serial_no = e.serial_no
      and c.clinic_no = visitId
```

```

        and b.ordered_by = deptId
-- and b.doctor_no = doctorId ----modify 20230914

        and e.serial_no = v_orders_desc.serial_no
        and e.charge_indicator = '0'
        and b.rcpt_no is null
    group by e.visit_date, e.clinic_no;

-- group by e.visit_date, e.visit_no, e.serial_no; ----modify 20230914

exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '开单查询数据失败异常' || SQLERRM;
    goto end_program;
end;

r_master.rcpt_no      := v_rcpt_nos;
r_master.Visit_Date   := sysdate;
r_master.operator_no  := v_opera_no;
r_master.total_costs  := n_rcpt_costs; --(n_payamount / 100);
r_master.total_charges := n_rcpt_charges; --(n_payamount / 100);
r_master.charge_indicator := 0;
r_master.order_id_pay := orderPayId;

if payMethod = 'XSYB' then

    -----

```

```

-- r_master.insur_personal_id      := insurPersonalId;
r_master.insur_personal_id := of_get_xmlval(as_xmlParm, 'PSN_NO');
r_master.insur_card_no      := insurCardNo;
-----赵志刚-职工居民转换
select nvl(t.his_code, 0)
into v_INSUR_CARD_CLASS
from ldinsur.dict_value_gather t
where t.dict_type = '人员类别'
and t.data_value = nvl(t_ld_outp_insur_settle.psn_type, '11');

r_master.insur_card_class := v_INSUR_CARD_CLASS;

r_master.insur_medical_class_name := insurMedicalClassName;

r_master.insur_class_type := insurClassType;
--r_master.insur_returnid      := insurReturnid;
r_master.insur_returnid := of_get_xmlval(as_xmlParm, 'MDTRT_ID');
--r_master.insur_settle_id      := insurSettleId;
r_master.insur_settle_id := of_get_xmlval(as_xmlParm, 'SETL_ID');
--r_master.insur_ov_area_code    := insurOvAreaCode;
--r_master.insur_area_code       := insurAreaCode;
r_master.samno      := samno;
r_master.insur_no    := insurNo;
r_master.auto_flag   := '2';
r_master.charge_type := '两定医保';

-- r_master.insur_begfee      := n_insurBegfee;

-- r_master.insur_card_amount := n_insurCardAmount;

```

```
r_master.insur_card_amount := of_get_xmlval(as_xmlParm, 'BALC');
```

```
-- r_master.insur_mxb_tclj := n_insurMxbTclj;
```

```
-- r_master.insur_mxb_nllj := n_insurMxbNllj;
```

```
-- r_master.insur_begfee_remain := n_insurBegfeeRemain;
```

```
----- 赵志刚 20211021 -----新增医保表赋值 赵志刚 20211021
```

```
r_master.remarks      := of_get_xmlval(as_xmlParm,  
                                         'emp_name');
```

```
r_master.insur_ov_area_code := of_get_xmlval(as_xmlParm,  
                                              'insuplc_admdvs');
```

```
r_master.insur_area_code  := of_get_xmlval(as_xmlParm,  
                                            'insuplc_admdvs');
```

```
r_master.insur_card_amount := of_get_xmlval(as_xmlParm,  
                                              'BALC');
```

```
t_ld_outp_insur_settle.setl_id := r_master.insur_settle_id;
```

```
t_ld_outp_insur_settle.acct_pay := of_get_xmlval(as_xmlParm,  
                                                  'ACCT_PAY'); -- 32    acct_pay, --个人账户支出 数值型
```

16,2 Y

```
t_ld_outp_insur_settle.psn_cash_pay := of_get_xmlval(as_xmlParm,  
                                                      'PSN_CASH_PAY'); -- 33    psn_cash_pay, --个人现金支出
```

数值型 16,2 Y

```
t_ld_outp_insur_settle.fund_pay_sumamt := of_get_xmlval(as_xmlParm,  
                                                         'FUND_PAY_SUMAMT'); -- 30    fund_pay_sumamt, -- 基
```

金支付总额 数值型 16,2 Y

```
select nvl(sum(decode(a.money_type,  
                      '线上基金',  
                      a.payment_amount,  
                      0)),
```

```

    0),
    nvl(sum(decode(a.money_type,
        '线上医保',
        a.payment_amount,
        0)),
    0),
    nvl(count(*), 0)
into v_later_paysumamt, v_later_ACCTPAY, v_cp

from outp_rcpt_master t, outp_payments_money a
where t.rcpt_no = a.rcpt_no
    and t.patient_id = v_pid
    and t.samno = samno
    and t.insur_settle_id = t_ld_outp_insur_settle.setl_id;

-----循环调用开单缴费-----
--1、基金支付有剩余，足够基金支付本单
IF n_rcpt_costs <=
    t_ld_outp_insur_settle.fund_pay_sumamt - v_later_paysumamt THEN
    v_MoneyTtype_one := '线上基金';
    n_rcpt_costs_one := n_rcpt_costs;
else
    --2、报销基金支付有剩余，不够支付本单，两种支付方式 基金支付或者卡支付
    n_rcpt_costs_one := t_ld_outp_insur_settle.fund_pay_sumamt -
        v_later_paysumamt; -- 基金支付

if 0 <= n_rcpt_costs_one and n_rcpt_costs_one < n_rcpt_costs then
    v_MoneyTtype_one := '线上基金';

```

```

n_rcpt_costs_two := n_rcpt_costs - n_rcpt_costs_one; --卡支付

--2.1 基金支付之后，线上医保卡支付够支付本单

if n_rcpt_costs_two <=
    t_ld_outp_insur_settle.acct_pay - v_later_ACCTPAY then
    v_MoneyTptype_two := '线上医保';

else
    ---2.2 基金支付之后，线上医保卡支付不够支付本单，需要现金支付
    if 0 < n_rcpt_costs_two and
        n_rcpt_costs_two >
            t_ld_outp_insur_settle.acct_pay - v_later_ACCTPAY then
        v_MoneyTptype_two := '线上医保';
        v_MoneyTptype_three := '小程序';
        n_rcpt_costs_two := t_ld_outp_insur_settle.acct_pay -
            v_later_ACCTPAY;
        n_rcpt_costs_three := n_rcpt_costs - n_rcpt_costs_one -
            n_rcpt_costs_two;
    end if;
end if;
end if;
END IF;

if n_rcpt_costs_one + n_rcpt_costs_two + n_rcpt_costs_three <>
    n_rcpt_costs then
    ROLLBACK;
    on_return := -1;

```

```
on_message := '支付方式不等于本单金额,支付失败: 收据号' || v_rcpt_nos;
```

```
goto end_program;
```

```
end if;
```

```
begin
```

```
if n_rcpt_costs_one <> 0 then
```

```
insert into OUTPBILL.OUTPUT_PAYMENTS_MONEY
```

```
(RCPT_NO,
```

```
PAYMENT_NO,
```

```
MONEY_TYPE,
```

```
PAYMENT_AMOUNT,
```

```
REFUNDED_AMOUNT)
```

```
values
```

```
(v_rcpt_nos, 1, '线上基金', n_rcpt_costs_one, 0.00);
```

```
end if;
```

```
if n_rcpt_costs_two <> 0 then
```

```
insert into OUTPBILL.OUTPUT_PAYMENTS_MONEY
```

```
(RCPT_NO,
```

```
PAYMENT_NO,
```

```
MONEY_TYPE,
```

```
PAYMENT_AMOUNT,
```

```
REFUNDED_AMOUNT)
```

```

values
    (v_rcpt_nos, 2, '线上医保', n_rcpt_costs_two, 0.00);
end if;

if n_rcpt_costs_three <> 0 then

insert into OUTPBILL.OUTPUT_PAYMENTS_MONEY
    (RCPT_NO,
    PAYMENT_NO,
    MONEY_TYPE,
    PAYMENT_AMOUNT,
    REFUNDED_AMOUNT)
values
    (v_rcpt_nos, 3, '小程序', n_rcpt_costs_three, 0.00);
end if;

exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '插入支付方式错误' || SQLERRM;
    goto end_program;

end;

if v_cp = 0 then
begin
    -- 插入门诊医保结算数据 Id_outp_insur_settlement (赵志刚)

    -- if v_cp > 0 then

```


t_ld_outp_insor_settle.mdtrt_id := r_master.insur_returnid; --1 mdtrt_id, --就诊 ID
字符型 30 Y

--setl_id,--结算 id

t_ld_outp_insor_settle.psn_no := r_master.insur_personal_id; -- 2 psn_no, --
人员编号 字符型 30 Y

t_ld_outp_insor_settle.psn_name := of_get_xmlval(as_xmlParm,
'PSN_NAME'); -- 3 psn_name, --人员姓名 字符型 50 Y

t_ld_outp_insor_settle.psn_cert_type := of_get_xmlval(as_xmlParm,
'PSN_CERT_TYPE'); -- 4 psn_cert_type, --人员证件类型
字符型 6 Y Y

t_ld_outp_insor_settle.certno := of_get_xmlval(as_xmlParm,
'CERTNO'); -- 5 certno, --证件号码 字符型 50 Y

t_ld_outp_insor_settle.gend := of_get_xmlval(as_xmlParm,
'GEND'); -- 6 gend, --性别 字符型 6 Y

t_ld_outp_insor_settle.naty := of_get_xmlval(as_xmlParm,
'NATY'); -- 7 naty, --民族 字符型 3 Y

t_ld_outp_insor_settle.brdy := of_get_xmlval(as_xmlParm,
'BRDY'); -- 8 brdy, --出生日期 日期型 yyyy-MM-dd

t_ld_outp_insor_settle.age := of_get_xmlval(as_xmlParm,
'AGE'); -- 9 age, --年龄 数值型 4,1

t_ld_outp_insor_settle.insutype := of_get_xmlval(as_xmlParm,
'INSUTYPE'); -- 10 insutype, --险种类型 字符型 6 Y

t_ld_outp_insor_settle.psn_type := of_get_xmlval(as_xmlParm,
'PSN_TYPE'); -- 11 psn_type, --人员类别 字符型 6 Y
Y

t_ld_outp_insor_settle.cvlserv_flag := of_get_xmlval(as_xmlParm,
'CVLSERV_FLAG'); -- 12 cvlserv_flag, --公务员标志 字
符型 3 Y Y

t_ld_outp_insor_settle.setl_time := of_get_xmlval(as_xmlParm,'SETL_TIME'); -- 13
setl_time, --结算时间

t_ld_outp_insur_settle.mdtrt_cert_type := of_get_xmlval(as_xmlParm,
'MDTRT_CERT_TYPE'); -- 14 mdtrt_cert_type, --就诊
凭证类型 字符型 3 Y

t_ld_outp_insur_settle.med_type := of_get_xmlval(as_xmlParm,
'MED_TYPE'); -- 15 med_type, --医疗类别 字符型 6
Y Y

t_ld_outp_insur_settle.medfee_sumamt := of_get_xmlval(as_xmlParm,
'MEDFEE_SUMAMT'); -- 16 medfee_sumamt, --医疗
费总额 数值型 16,2 Y

t_ld_outp_insur_settle.fulamt_ownpay_amt := of_get_xmlval(as_xmlParm,
'FULAMT_OWNPAY_AMT'); -- 17
fulamt_ownpay_amt, --全自费金额 数值型 16,2 Y

t_ld_outp_insur_settle.overlmt_selfpay := of_get_xmlval(as_xmlParm,
'OVERLMT_SELFPAY'); --18 overlmt_selfpay, --超限
价自费费用 数值型 16,2 Y

t_ld_outp_insur_settle.preselfpay_amt := of_get_xmlval(as_xmlParm,
'PRESELFPAY_AMT'); -- 19 preselfpay_amt, --先行自
付金额 数值型 16,2 Y

t_ld_outp_insur_settle.inscp_scp_amt := of_get_xmlval(as_xmlParm,
'INSCP_SCP_AMT'); -- 20 inscp_scp_amt, --符合政策
范围金额 数值型 16,2 Y

t_ld_outp_insur_settle.act_pay_dedc := of_get_xmlval(as_xmlParm,
'ACT_PAY_DEDC'); -- 21 act_pay_dedc, --实际支付起
付线 数值型 16,2

t_ld_outp_insur_settle.hifp_pay := of_get_xmlval(as_xmlParm,
'HIFP_PAY'); -- 22 hifp_pay, --基本医疗保险统筹基
金支出 数值型 16,2 Y

t_ld_outp_insur_settle.pool_prop_selfpay := of_get_xmlval(as_xmlParm,
'POOL_PROP_SELFPAY'); -- 23 pool_prop_selfpay, --
基本医疗保险统筹基金 支付比例 数值型 5,4 Y

t_ld_outp_insur_settle.cvlserv_pay := of_get_xmlval(as_xmlParm,

'CVLSERV_PAY'); -- 24 cvlserv_pay, --公务员医疗补
 助资金支出 数值型 16,2 Y
 t_ld_outp_insur_settle.hifes_pay := of_get_xmlval(as_xmlParm,
 'HIFES_PAY'); -- 25 hifes_pay, --企业补充医疗保险
 基金支出数值型 16,2 Y
 t_ld_outp_insur_settle.hifmi_pay := of_get_xmlval(as_xmlParm,
 'HIFMI_PAY'); -- 26 hifmi_pay, --居民大病保险资金
 支出 数值型 16,2 Y
 t_ld_outp_insur_settle.hifob_pay := of_get_xmlval(as_xmlParm,
 'HIFOB_PAY'); -- 27 hifob_pay, --职工大额医疗费用
 补助基金支出 数值型 16,2 Y
 t_ld_outp_insur_settle.maf_pay := of_get_xmlval(as_xmlParm,
 'MAF_PAY'); -- 28 maf_pay, --医疗救助基金支出 数
 值型 16,2 Y
 t_ld_outp_insur_settle.oth_pay := of_get_xmlval(as_xmlParm,
 'OTH_PAY'); -- 29 oth_pay, --其他支出 数值型 16,2
 Y
 t_ld_outp_insur_settle.psn_part_amt := of_get_xmlval(as_xmlParm,
 'PSN_PART_AMT'); -- 31 psn_part_amt, --个人负担总
 金额 数值型 16,2 Y
 t_ld_outp_insur_settle.acct_pay := of_get_xmlval(as_xmlParm,
 'ACCT_PAY'); -- 32 acct_pay, --个人账户支出 数值型
 16,2 Y
 t_ld_outp_insur_settle.psn_cash_pay := of_get_xmlval(as_xmlParm,
 'PSN_CASH_PAY'); -- 33 psn_cash_pay, --个人现金支
 出 数值型 16,2 Y
 t_ld_outp_insur_settle.hosp_part_amt := of_get_xmlval(as_xmlParm,
 'hosp_part_amt'); -- 34 hosp_part_amt, -- 医院负担
 金额 数值型 16,2 Y
 t_ld_outp_insur_settle.balc := of_get_xmlval(as_xmlParm,
 'BALC'); -- 35 balc, --余额 数值型 16,2 YXJ-K01-2020

```
t_ld_outp_insor_settle.acct_mulaid_pay := of_get_xmlval(as_xmlParm,
'ACCT_MULAID_PAY'); -- 36 acct_mulaid_pay, --个人
账户共济支付金额 数值型 16,2 Y
```

```
t_ld_outp_insor_settle.medins_setl_id := of_get_xmlval(as_xmlParm,
'MEDINS_SETL_ID'); -- 37 medins_setl_id, --医药机构
结算 ID 字符型 30 Y 存放发送方报文 ID
```

```
t_ld_outp_insor_settle.clr_optins := of_get_xmlval(as_xmlParm,
'CLR_OPTINS'); -- 38 clr_optins, --清算经办机构 字符
型 6
```

```
-- t_ld_outp_insor_settle.clr_way := of_get_xmlval(as_xmlParm, 'CLR_WAY'); -- 39
clr_way, --清算方式 字符型 6 Y
```

```
t_ld_outp_insor_settle.clr_way := '1';
```

```
t_ld_outp_insor_settle.clr_type := of_get_xmlval(as_xmlParm,
'CLR_TYPE'); -- 40 clr_type --清算类别 字符型 6 Y Y
```

```
t_ld_outp_insor_settle.insuplc_admdvs := of_get_xmlval(as_xmlParm,
'insuplc_admdvs');
```

```
t_ld_outp_insor_settle.rcpt_no := samno;
```

```
INSERT INTO ldinsur.ld_outp_insor_settlement
```

```
VALUES t_ld_outp_insor_settle;
```

```
exception
```

```
when others then
```

```
ROLLBACK;
```

```
on_return := -1;
```

```
on_message := '插入医保结算表录出错' || '[当前收据号:' || v_rcpt_nos || ']' ||
```

```
SQLERRM;
```

```
goto end_program;
```

```

        end;
    end if;

    -----

else
    r_master.samno := orderId;
end if;

exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '设置门诊收据记录参数对象异常' || SQLERRM;
    goto end_program;
end;

begin
    -- 插入门诊医疗收据记录
    INSERT INTO OUTPBILL.outp_rcpt_master VALUES r_master;

exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '插入门诊医疗收据记录出错' || '['当前收据号:' || v_rcpt_nos || ']' ||
        SQLERRM;
    goto end_program;

```

end;

-- INSERT INTO t_ld_outp_insur_settle VALUES ldinsur.ld_outp_insur_settlement;----赵志刚
20211021-----医保表

begin

SELECT VISIT_NO.NEXTVAL into n_new_visit_no FROM DUAL; --注意权限问题

r_order.visit_date := d_now;

--r_order.visit_no := n_new_visit_no;

r_order.patient_id := v_pid;

r_order.ordered_by_dept := v_orders_desc.ordered_by;

r_order.ordered_by_doctor := v_orders_desc.doctor_no;

r_order.rcpt_no := v_rcpt_nos;

r_order.visit_id := v_orders_desc.visit_no;

r_order.clinic_no := visitId;

select count(1)

into n_presc_count

from outp_presc

where serial_no = v_orders_desc.serial_no

and visit_date = v_orders_desc.visit_date

and visit_no = v_orders_desc.visit_no;

if n_presc_count > 0 then

r_order.presc_indicator := 1;

-- 有药品处方的 r_order.visit_no=outp_presc。 presc_no

select distinct presc_no

into v_presc_visit_no

```

        from outp_presc
        where serial_no = v_orders_desc.serial_no
              and visit_date = v_orders_desc.visit_date
              and visit_no = v_orders_desc.visit_no;

        r_order.visit_no := v_presc_visit_no;
        r_order.presc_no := v_presc_visit_no;

    else
        r_order.presc_indicator := 0;
        r_order.visit_no      := n_new_visit_no;
    end if;
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '设置开单记录参数对象异常' || SQLERRM;
        goto end_program;
end;

--插入开单记录
begin
    INSERT INTO OUTPBILL.outp_order_desc VALUES r_order;
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '插入开单记录出错' || SQLERRM;
        goto end_program;

```

end;

```
insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||  
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-6',  
    'payChgOrder001',  
    as_xmlParm,  
    '准备循环处方明细-06' ||  
    of_get_xmlval(as_xmlParm, 'orderId'),  
    on_return,  
    on_message);
```

for v_orders_cost in c_order_cost loop

```
insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||  
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-7',  
    'payChgOrder001',  
    as_xmlParm,  
    '循环处方明细-07-' || v_orders_desc.serial_no || '-' ||  
    v_orders_cost.item_no || '-' ||  
    of_get_xmlval(as_xmlParm, 'orderId'),  
    on_return,  
    on_message);
```

if v_orders_cost.serial_no = v_orders_desc.serial_no then

begin

```
SELECT PRICE_LIST.CLASS_ON_OUTP_RCPT,  
    price_list.class_on_reckoning,  
    price_list.subj_code,
```



```

        price_list.price
into v_class_on_outp_rcpt,
        v_class_on_reckoning,
        v_subj_code,
        n_item_price
FROM PRICE_LIST
WHERE PRICE_LIST.ITEM_CLASS = v_orders_cost.item_class
      AND PRICE_LIST.ITEM_CODE = v_orders_cost.item_code
      AND PRICE_LIST.ITEM_SPEC = v_orders_cost.item_spec
      AND PRICE_LIST.UNITS = v_orders_cost.units
      AND (SYSDATE > PRICE_LIST.START_DATE)
      AND (SYSDATE < PRICE_LIST.STOP_DATE OR
           PRICE_LIST.STOP_DATE IS NULL)
and rownum = 1;

r_items.visit_date := d_now;
r_items.visit_no  := r_order.visit_no; --n_new_visit_no;
r_items.rcpt_no   := v_rcpt_nos;

if v_orders_desc.serial_no = v_seriano then
    r_items.ITEM_no := v_n;
else
    v_n             := 1;
    r_items.ITEM_no := 1;
end if;

r_items.item_class := v_orders_cost.item_class;
r_items.class_on_rcpt := v_CLASS_ON_OUTP_RCPT;
r_items.item_code   := v_orders_cost.item_code;
r_items.item_name    := v_orders_cost.item_name;
r_items.ITEM_SPEC    := v_orders_cost.item_spec;

```

```

if v_orders_cost.item_class = 'B' then
    r_items.amount := v_orders_cost.amount /
        v_orders_cost.repetition;
else
    r_items.amount := v_orders_cost.amount;
end if;

r_items.units      := v_orders_cost.units;
r_items.performed_by := v_orders_cost.performed_by;
r_items.costs      := v_orders_cost.costs;
r_items.charges     := v_orders_cost.charges;

r_items.flag       := 0;
r_items.repetition  := v_orders_cost.repetition;
r_items.class_on_reckoning := v_class_on_reckoning;
r_items.subj_code   := v_subj_code;
r_items.price_quotient := v_orders_cost.price_quotient;
r_items.item_price  := n_item_price;
r_items.order_no    := v_orders_cost.order_no;

if v_orders_cost.item_class = 'A' or
    v_orders_cost.item_class = 'B' then
    select t.dosage || t.dosage_units || ' ' || t.administration || ' ' ||
        (select prn_name
         from comm.PERFORM_FREQ_DICT
         where freq_desc = t.frequency)
    into v_administration
    from OUTPDOCT.outp_presc t
    where t.visit_date = v_orders_cost.visit_date

```

```

        and t.visit_no = v_orders_cost.visit_no
        and t.serial_no = v_orders_cost.serial_no
        and t.item_no = v_orders_cost.item_no;

        r_items.administration := v_administration;
    end if;
exception
    when others then
        ROLLBACK;

        on_return := -1;

        on_message := '设置门诊诊疗费用项目参数对象异常' || SQLERRM;

        goto end_program;
end;

--插入门诊病人诊疗费用项目
begin
    INSERT INTO OUTPBILL.OUTPUT_BILL_ITEMS VALUES r_items;

    -- INSERT INTO OUTPUT_BILL_ITEMS_bak VALUES r_items;
exception
    when others then
        ROLLBACK;

        on_return := -1;

        on_message := '插入门诊病人诊疗费用项目出错' || SQLERRM;

        goto end_program;
end;

--更新门诊医生收费明细
begin
    update OUTPDOCT.outp_orders_costs

```

```

set class_on_rcpt = v_class_on_outp_rcpt,
rcpt_no          = v_rcpt_nos,
charge_indicator = 1,
bill_date        = sysdate
where patient_id = v_pid
and serial_no = v_orders_desc.serial_no
and item_class = v_orders_cost.item_class
and order_no = v_orders_cost.order_no
and item_no = v_orders_cost.item_no;
exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '更新门诊医生收费明细出错' || SQLERRM;
    goto end_program;
end;

v_n := v_n + 1;
v_seriano := v_orders_desc.serial_no;
end if;
end loop;

```

----- start 插入待发记录 -----

```

if n_presc_count > 0 then
begin
select distinct t.presc_no,
               decode(t.item_class, 'A', 0, 'B', 1),
               t.presc_attr,
               t.repetition,

```

```

        t.dispensary
into v_tmp_presc_no, --r_drug_presc_master.presc_no,

    r_drug_presc_master.presc_type,

    r_drug_presc_master.presc_attr,

    r_drug_presc_master.repetition,

    r_drug_presc_master.dispensary

from outp_presc t

where t.visit_date = v_orders_desc.visit_date

    and t.visit_no = v_orders_desc.visit_no

    and t.serial_no = v_orders_desc.serial_no

    and t.charge_indicator = '0';

--r_drug_presc_master.presc_no := n_new_visit_no;
r_drug_presc_master.presc_no := v_tmp_presc_no;

r_drug_presc_master.ordered_by    := v_orders_desc.ordered_by;
r_drug_presc_master.prescribed_by := v_orders_desc.doctor_no;
r_drug_presc_master.presc_date    := d_now;
r_drug_presc_master.patient_id    := v_pid;
r_drug_presc_master.name          := v_pName;
r_drug_presc_master.name_phonetic := v_pNamePhonetic;
r_drug_presc_master.identity      := v_plidentity;
r_drug_presc_master.charge_type   := '自费';
r_drug_presc_master.unit_in_contract := v_pUnitInContract;
r_drug_presc_master.presc_source  := 0;
r_drug_presc_master.entered_by    := v_enteredBy; --'小程序缴费'; --
r_drug_presc_master.prescribed_by;

r_drug_presc_master.doctor_user   := r_drug_presc_master.prescribed_by;
r_drug_presc_master.rcpt_no       := v_rcpt_nos;

```

```

r_drug_presc_master.clinic_no    := visitId;
exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '设置门诊待发药主记录参数对象异常' || SQLERRM;
    goto end_program;
end;

```

```

select count(1)
into n_presc_master_count
from PHARMACY.DRUG_PRESC_MASTER_TEMP m
where m.presc_no = r_drug_presc_master.presc_no
and m.patient_id = v_pid
and m.clinic_no = visitId
and to_char(m.presc_date, 'yyyy-mm-dd') =
    to_char(sysdate, 'yyyy-mm-dd');

```

```

select count(1)
into n_presc_master_disp_count
from pharmacy.drug_presc_master p
where p.presc_no = r_drug_presc_master.presc_no
and p.patient_id = v_pid
and p.clinic_no = visitId
and to_char(p.presc_date, 'yyyy-mm-dd') =
    to_char(sysdate, 'yyyy-mm-dd');

```

```

if n_presc_master_count = 0 and n_presc_master_disp_count = 0 then

```

```

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
    to_char(systimestamp,
        'yyyymmddhh24missxff3') || '-8-' ||
    r_drug_presc_master.presc_no || '-1',
    'payChgOrder001',
    as_xmlParm,
    '插入待发药处方主记录-08-' ||
    r_drug_presc_master.presc_no || '-1-' ||
    of_get_xmlval(as_xmlParm, 'orderId'),
    null,
    null);

```

--插入待发药主记录

```

begin
    insert into PHARMACY.DRUG_PRESC_MASTER_TEMP
        values r_drug_presc_master;
exception
    when others then
        ROLLBACK;
    on_return := -1;
    on_message := '插入待发药处方主记录出错' || SQLERRM;
    goto end_program;
end;

```

```

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
    to_char(systimestamp,
        'yyyymmddhh24missxff3') || '-8-' ||
    r_drug_presc_master.presc_no || '-2',
    'payChgOrder001',

```

```

        as_xmlParm,
        '插入待发药处方主记录完成-08-' ||
        r_drug_presc_master.presc_no || '-2-' ||
        of_get_xmlval(as_xmlParm, 'orderId'),
        null,
        null);
end if;

--v_tmp_presc_no := r_drug_presc_master.presc_no;

for v_drug_presc in c_drug_presc loop
    v_drug_presc_tmp := v_drug_presc.presc_no;

    select count(1)
        into n_presc_detail_count
        from PHARMACY.DRUG_PRESC_detail_TEMP m
        where m.presc_no = r_drug_presc_master.presc_no
            and m.item_no = v_drug_presc.item_no
            and m.presc_date = d_now;

    select count(1)
        into n_presc_detail_disp_count
        from pharmacy.drug_presc_detail m
        where m.presc_no = r_drug_presc_master.presc_no
            and m.item_no = v_drug_presc.item_no
            and m.presc_date = d_now;

    if v_drug_presc.presc_no = v_tmp_presc_no and
        n_presc_detail_count = 0 and n_presc_detail_disp_count = 0 then

```



```

begin
    r_drug_presc_detail.presc_date := d_now;
    r_drug_presc_detail.presc_no  := r_drug_presc_master.presc_no;
    r_drug_presc_detail.item_no   := v_drug_presc.item_no;
    r_drug_presc_detail.dosage_each := r_drug_presc_detail.quantity;

    r_drug_presc_detail.drug_code   := v_drug_presc.drug_code;
    r_drug_presc_detail.drug_name   := v_drug_presc.drug_name;
    r_drug_presc_detail.firm_id     := v_drug_presc.firm_id;
    r_drug_presc_detail.package_spec := v_drug_presc.drug_spec;
    r_drug_presc_detail.package_units := v_drug_presc.units;
    r_drug_presc_detail.quantity    := v_drug_presc.amount;
    r_drug_presc_detail.costs       := v_drug_presc.costs;
    r_drug_presc_detail.payments    := v_drug_presc.charges;
    r_drug_presc_detail.administration := v_drug_presc.administration;
    r_drug_presc_detail.order_no     := v_drug_presc.order_no;
    r_drug_presc_detail.order_sub_no := v_drug_presc.sub_order_no;

    SELECT min_spec
    into r_drug_presc_detail.drug_spec
    FROM drug_price_list t
    WHERE stop_date IS NULL
        AND drug_code = r_drug_presc_detail.drug_code
        AND drug_spec || firm_id = (r_drug_presc_detail.package_spec ||
        r_drug_presc_detail.firm_id);
exception
    when others then
        ROLLBACK;
    on_return := -1;

```

```

on_message := '设置门诊待发药明细记录参数对象异常' || SQLERRM;
goto end_program;
end;

```

```

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
    to_char(systimestamp,
        'yyyymmddhh24missxff3') || '-9-' ||
    r_drug_presc_detail.presc_no || '-' ||
    r_drug_presc_detail.item_no || '-1',
    'payChgOrder001',
    as_xmlParm,
    '插入待发药处方明细记录-09-' ||
    r_drug_presc_detail.presc_no || '-' ||
    r_drug_presc_detail.item_no || '-1' || '-' ||
    of_get_xmlval(as_xmlParm, 'orderId'),
    null,
    null);

```

--插入待发药处方明细记录

```

begin
    insert into PHARMACY.Drug_Presc_Detail_Temp
    values r_drug_presc_detail;
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '插入待发药处方明细记录出错' || SQLERRM;
        goto end_program;
end;

```

```

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
    to_char(sysimestamp,
        'yyyymmddhh24missxff3') || '-9-' ||
    r_drug_presc_detail.presc_no || '-' ||
    r_drug_presc_detail.item_no || '-2',
    'payChgOrder001',
    as_xmlParm,
    '插入待发药处方明细记录完成-09-' ||
    r_drug_presc_detail.presc_no || '-' ||
    r_drug_presc_detail.item_no || '-2' || '-' ||
    of_get_xmlval(as_xmlParm, 'orderId'),
    null,
    null);

end if;

end loop;

end if;

----- end 插入待发记录 -----

if n_presc_count > 0 then
    -- 更新处方医嘱明细记录
    begin
        update OUTPDOCT.outp_presc t
            set charge_indicator = 1
        where serial_no = v_orders_desc.serial_no
            and visit_date = v_orders_desc.visit_date
            and visit_no = v_orders_desc.visit_no;
    end;
end if;

```

```

exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '更新处方医嘱明细记录出错' || SQLERRM;
    goto end_program;
end;

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') ||
    '-191-' || v_orders_desc.visit_date || '-' ||
    v_orders_desc.visit_no || '-' ||
    v_orders_desc.serial_no,
    'payChgOrder001',
    as_xmlParm,
    '更新处方医嘱明细记录-' || v_orders_desc.serial_no || '-' ||
    v_orders_desc.visit_date || '-' ||
    v_orders_desc.visit_no || '-' ||
    of_get_xmlval(as_xmlParm, 'orderId'),
    null,
    '更新[' || sql%rowcount || ']条记录');
end if;

-- 更新检查治疗医嘱主记录
begin
    update OUTPDOCT.outp_treat_rec_t
        set charge_indicator = 1
    where serial_no = v_orders_desc.serial_no
        and visit_date = v_orders_desc.visit_date

```

```

        and visit_no = v_orders_desc.visit_no;
exception
when others then
    ROLLBACK;
on_return := -1;
on_message := '更新检查治疗医嘱主记录出错' || SQLERRM;
goto end_program;
end;

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') ||
    '-192-' || v_orders_desc.visit_date || '-' ||
    v_orders_desc.visit_no || '-' ||
    v_orders_desc.serial_no,
    'payChgOrder001',
    as_xmlParm,
    '更新检查治疗医嘱主记录-' || v_orders_desc.serial_no || '-' ||
    v_orders_desc.visit_date || '-' ||
    v_orders_desc.visit_no || '-' ||
    of_get_xmlval(as_xmlParm, 'orderId'),
    null,
    '更新[' || sql%rowcount || ']条记录');

begin
update OUTPDOCT.outp_treat_rec
    set charge_indicator = 1
where serial_no = v_orders_desc.serial_no
    and visit_date = v_orders_desc.visit_date
    and visit_no = v_orders_desc.visit_no;

```

```

exception
when others then

    ROLLBACK;

    on_return := -1;

    on_message := '更新检查治疗医嘱明细记录出错' || SQLERRM;

    goto end_program;

end;

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') ||
    '-193-' || v_orders_desc.visit_date || '-' ||
    v_orders_desc.visit_no || '-' ||
    v_orders_desc.serial_no,
    'payChgOrder001',
    as_xmlParm,
    '更新检查治疗医嘱明细记录-' || v_orders_desc.serial_no || '-' ||
    v_orders_desc.visit_date || '-' ||
    v_orders_desc.visit_no || '-' ||
    of_get_xmlval(as_xmlParm, 'orderId'),
    null,
    '更新[' || sql%rowcount || ']条记录');

select count(*)
into v_app_no
from outp_treat_rec a
where a.item_class = 'D'
and a.serial_no = v_orders_desc.serial_no
and a.visit_date = v_orders_desc.visit_date
and a.visit_no = v_orders_desc.visit_no;

```

```

if v_app_no > 0 then
begin
    select sum(round(a.costs, 2)), sum(round(a.charges, 2))
        into n_exam_costs, n_exam_charges
    from outp_treat_rec a
    where a.item_class = 'D'
        and a.serial_no = v_orders_desc.serial_no
        and a.visit_date = v_orders_desc.visit_date
        and a.visit_no = v_orders_desc.visit_no
    group by a.serial_no;
exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '获取 outp_treat_rec 的 item_class=D 费用异常' || SQLERRM;
    goto end_program;
end;

```

--更新检查预约记录

```

begin
    update EXAM.exam_appoints c
        set c.billing_indicator = '1',
            c.costs          = n_exam_costs,
            c.charges        = n_exam_charges
    where c.exam_no in
        (select appoint_no
         from OUTPDOCT.outp_treat_rec
         where item_class = 'D'
            and serial_no = v_orders_desc.serial_no

```

```

        and visit_date = v_orders_desc.visit_date
        and visit_no = v_orders_desc.visit_no);
exception
when others then
    ROLLBACK;
on_return := -1;
on_message := '更新检查预约记录出错' || SQLERRM;
goto end_program;
end;

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') ||
    '-194-' || v_orders_desc.visit_date || '-' ||
    v_orders_desc.visit_no || '-' ||
    v_orders_desc.serial_no,
    'payChgOrder001',
    as_xmlParm,
    '更新检查预约记录-' || v_orders_desc.serial_no || '-' ||
    v_orders_desc.visit_date || '-' ||
    v_orders_desc.visit_no || '-' ||
    of_get_xmlval(as_xmlParm, 'orderId'),
    null,
    '更新[' || sql%rowcount || ']条记录');

```

--更新检查主记录

```

begin
update EXAM.Exam_Master c
    set c.costs      = n_exam_costs,
        c.charges    = n_exam_charges,

```



```

        c.charge_indicator = '1'
where c.exam_no in
    (select appoint_no
     from OUTPDOCT.outp_treat_rec
     where item_class = 'D'
     and serial_no = v_orders_desc.serial_no
     and visit_date = v_orders_desc.visit_date
     and visit_no = v_orders_desc.visit_no);
exception
when others then
    ROLLBACK;
on_return := -1;
on_message := '更新检查主记录出错' || SQLERRM;
goto end_program;
end;

insert wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') ||
    '-195-' || v_orders_desc.visit_date || '-' ||
    v_orders_desc.visit_no || '-' ||
    v_orders_desc.serial_no,
    'payChgOrder001',
    as_xmlParm,
    '更新检查主记录-' || v_orders_desc.serial_no || '-' ||
    v_orders_desc.visit_date || '-' ||
    v_orders_desc.visit_no || '-' ||
    of_get_xmlval(as_xmlParm, 'orderId'),
    null,
    '更新[' || sql%rowcount || ']条记录');

```

--更新检查项目记录

```
begin
update EXAM.Exam_Items c
  set c.billing_indicator = '1'
where c.exam_no in
  (select appoint_no
   from OUTPDOCT.outp_treat_rec
   where item_class = 'D'
   and serial_no = v_orders_desc.serial_no
   and visit_date = v_orders_desc.visit_date
   and visit_no = v_orders_desc.visit_no);
exception
when others then
  ROLLBACK;
on_return := -1;
on_message := '更新检查项目记录出错' || SQLERRM;
goto end_program;
end;

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
  to_char(systimestamp, 'yyyymmddhh24missxff3') ||
  '-196-' || v_orders_desc.visit_date || '-' ||
  v_orders_desc.visit_no || '-' ||
  v_orders_desc.serial_no,
  'payChgOrder001',
  as_xmlParm,
  '更新检查项目记录-' || v_orders_desc.serial_no || '-' ||
  v_orders_desc.visit_date || '-' ||
```

```

        v_orders_desc.visit_no || '-' ||
        of_get_xmlval(as_xmlParm, 'orderId'),
        null,
        '更新[' || sql%rowcount || ']条记录');
end if;

select count(*)
into v_lab_no
from outp_treat_rec a
where a.item_class = 'C'
and a.serial_no = v_orders_desc.serial_no
and visit_date = v_orders_desc.visit_date
and visit_no = v_orders_desc.visit_no;
if v_lab_no > 0 then
begin
select sum(round(a.costs, 2)), sum(round(a.charges, 2))
into n_lab_costs, n_lab_charges
from outp_treat_rec a
where a.item_class = 'C'
and a.serial_no = v_orders_desc.serial_no
and a.visit_date = v_orders_desc.visit_date
and a.visit_no = v_orders_desc.visit_no
group by a.serial_no;
exception
when others then
rollback;
on_return := -1;
on_message := '获取 outp_treat_rec 的 item_class=C 费用异常' || SQLERRM;
goto end_program;

```

```

end;

--更新检验主记录

begin
  update LAB.Lab_Test_Master c
    set c.costs      = n_lab_costs,
        c.charges    = n_lab_charges,
        c.billing_indicator = '1'
  where c.test_no in
    (select appoint_no
      from OUTPDOCT.outp_treat_rec
     where item_class = 'C'
        and serial_no = v_orders_desc.serial_no
        and visit_date = v_orders_desc.visit_date
        and visit_no = v_orders_desc.visit_no);
exception
  when others then
    ROLLBACK;
    on_return := -1;
    on_message := '更新检验主记录出错' || SQLERRM;
    goto end_program;
end;

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
  to_char(systimestamp, 'yyyymmddhh24missxff3') ||
  '-197-' || v_orders_desc.visit_date || '-' ||
  v_orders_desc.visit_no || '-' ||
  v_orders_desc.serial_no,
  'payChgOrder001',

```

```

as_xmlParm,
'更新检验主记录-' || v_orders_desc.serial_no || '-' ||
v_orders_desc.visit_date || '-' ||
v_orders_desc.visit_no || '-' ||
of_get_xmlval(as_xmlParm, 'orderId'),
null,
'更新[' || sql%rowcount || ']条记录');

```

--更新检验项目记录

```

begin
update LAB.Lab_Test_Items c
set c.billing_indicator = '1'
where c.test_no in
(select appoint_no
from OUTPDOCT.outp_treat_rec
where item_class = 'C'
and serial_no = v_orders_desc.serial_no
and visit_date = v_orders_desc.visit_date
and visit_no = v_orders_desc.visit_no);
exception
when others then
ROLLBACK;
on_return := -1;
on_message := '更新检验项目记录出错' || SQLERRM;
goto end_program;
end;

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
to_char(systimestamp, 'yyyymmddhh24missxff3') ||

```

```

        '-198-' || v_orders_desc.visit_date || '-' ||
        v_orders_desc.visit_no || '-' ||
        v_orders_desc.serial_no,
        'payChgOrder001',
        as_xmlParm,
        '更新检验项目记录-' || v_orders_desc.serial_no || '-' ||
        v_orders_desc.visit_date || '-' ||
        v_orders_desc.visit_no || '-' ||
        of_get_xmlval(as_xmlParm, 'orderId'),
        null,
        '更新[' || sql%rowcount || ']条记录');
end if;
--end if;

-- 更新医嘱表
begin
    update OUTPDOCT.outp_orders t
        set t.rcpt_no = v_rcpt_nos
    where t.clinic_no = visitId
        and t.serial_no = v_orders_desc.serial_no
        and t.patient_id = v_pid;
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '更新医嘱表出错' || SQLERRM;
        goto end_program;
end;
```

```

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') ||
    '-200-' || v_orders_desc.visit_date || '-' ||
    v_orders_desc.visit_no || '-' ||
    v_orders_desc.serial_no,
    'payChgOrder001',
    as_xmlParm,
    '更新医嘱表-' || v_orders_desc.serial_no || '-' ||
    v_orders_desc.visit_date || '-' ||
    v_orders_desc.visit_no || '-' ||
    of_get_xmlval(as_xmlParm, 'orderId'),
    null,
    '更新[' || sql%rowcount || ']条记录');

```

--有公补逻辑

```
if payMethod <> 'XSYB' then
```

--原逻辑（没有公补）

```
--if subsidyMethod <> 'XSGB' then
```

```
begin
```

```
    r_payments.rcpt_no      := v_rcpt_nos;
```

```
    r_payments.payment_no   := 1;
```

```
    r_payments.money_type   := v_payMethod;
```

```
    r_payments.payment_amount := n_rcpt_costs; --(n_payamount / 100);
```

```
    r_payments.refunded_amount := 0;
```

```
exception
```

```
when others then
```

```
    ROLLBACK;
```

```
    on_return := -1;
```

```
    on_message := '设置门诊支付方式参数对象异常' || SQLERRM;
```

```

        goto end_program;
end;
--插入门诊病人支付方式记录
begin
    INSERT INTO OUTPBILL.outp_payments_money VALUES r_payments;
exception
    when others then
        ROLLBACK;
        on_return := -1;
        on_message := '插入门诊病人支付方式记录出错' || SQLERRM;
        goto end_program;
end;
end if;

v_rcpt_str := v_rcpt_str || '|' || v_rcpt_nos;
end loop;

-- 写入微信交易日志表
begin
insert into wxgh.wx_trans_his_log
(PATIENT_ID,
PATIENT_MOBILE,
ORDER_TYPE,
ORDER_ID,
ORDER_ID_PAY,
DEAL_MONEY,
PAY_APP_ID,
MCH_ID,
ORDER_PAY_TIME,

```



```

ORDER_DESC,
RCPT_NO,
SYS_DATETIME)
values
(v_pid,
null,
'CHG',
orderId,
orderPayId,
payAmount,
payAppId,
mchId,
payTime,
v_payMethod,
substr(v_rcpt_str, 2),
sysdate);
exception
when others then
ROLLBACK;
on_return := -1;
on_message := '写入微信交易日志表失败:' || SQLERRM;
goto end_program;
end;

/* begin

v_invo_content := '{
"rcpt_no":"' || v_rcpt_nos || '", ' ||

```

```
"alipayCode":"","  
"weChatOrderNo":"","||orderId||","||  
"weChatMedTransNo":"","||orderPayId||","||  
"openID":"","}";
```

```
system.get_einvoice_res(v_invo_content);
```

```
exception
```

```
when others then
```

```
on_return := -1;
```

```
on_message := '调用 evioce 接口失败:' || SQLERRM;
```

```
goto end_program;
```

```
end;*/
```

```
<<end_program>>
```

```
if on_return = 0 then
```

```
    insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||  
        to_char(systimestamp, 'yyyymmddhh24missxff3') ||  
        '-900',  
        'payChgOrder001',  
        as_xmlParm,  
        '结束' || of_get_xmlval(as_xmlParm, 'orderId'),  
        on_return,  
        on_message);
```

```
open aCur_temp for
```

```
    select to_char(on_return) as "resultCode",  
        on_message as "resultDesc",  
        visitId as "hisOrderId",  
        substr(v_rcpt_str, 2) as "receiptId"  
    from dual;
```

```
else
```

```
    insert_wx_webservice_err_logs('payChgOrder001' ||  
        to_char(systimestamp,  
            '-yyyymmddhh24missxff3'),  
        'payChgOrder001',  
        as_xmlParm,  
        ",  
        on_message);
```

```
insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
```

```

to_char(systimestamp, 'yyyymmddhh24missxff3') ||
'-900',
'payChgOrder001',
as_xmlParm,
'['错]结束' || of_get_xmlval(as_xmlParm, 'orderId'),
on_return,
on_message);

```

```

open aCur_temp for
select to_char(on_return) as "resultCode",
       on_message as "resultDesc",
       null "hisOrderId",
       null as "receiptId"
from dual;
end if;

```

```

exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '门诊缴费出错' || SQLERRM;
    insert_wx_webservice_err_logs('payChgOrder001' ||
        to_char(systimestamp,
            '-yyyymmddhh24missxff3'),
        'payChgOrder001',
        as_xmlParm,
        "",
        on_message);

```

```

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') ||
    '-11',
    'payChgOrder001',
    as_xmlParm,
    '['错]结束' || of_get_xmlval(as_xmlParm, 'orderId'),
    on_return,
    on_message);

open aCur_temp for
select to_char(on_return) as "resultCode",
    on_message as "resultDesc",
    null "hisOrderId",
    null as "receiptId"
from dual;

end;

```

-- 5.1.4 Paid Record Query Interface 001 已缴费记录查询接口 001

```

procedure getPayReceiptInfo001(patMedNo in varchar2, -- 患者诊疗卡号码

    beginDate in varchar2, -- 查询开始日期, 格式: YYYY-MM-DD

    endDate in varchar2, -- 查询结束日期, 格式: YYYY-MM-DD

    aCur_temp out sys_refcursor) is

/* 输出结果

* 字段      描述

* -----

patName      患者姓名

visitId      HIS 就诊流水号 (HIS 系统中就诊流水号 (内部 ID), 要求唯一。如:
12345)

visitNo      HIS 流水号备注说明 (HIS 系统中就诊流水号的显式信息, 如: 2012-3-
1/123 代表挂号日期为 2012-3-1, 流水号为 123)

deptId       接诊科室 ID (二级)

deptName     接诊科室名称 (二级)

doctorId     接诊医生 ID

doctorName   接诊医生姓名

payAmount    已缴费总金额, 单位: 分

chargeDate   缴费日期

payMode       支付方式 (ALIPAY-支付宝, WECHAT-微信, WINDOW-人工窗口,
SELF-自助)

status       订单状态 (PAYED-已支付, CANCEL-已退费, PART-部分退费,
NEGATIVE-退费负单)

dispenWininfor  发药窗口

mtguideinfor  检验检查执行地点

receiptId    收据 ID

*/

begin

```

```

open aCur_temp for

SELECT c.name as "patName", -- 患者姓名

       d.rcpt_no as "visitId", -- HIS 收据 id

       a.clinic_no as "visitNo", -- HIS 流水号备注说明

       a.visit_dept as "deptId", -- 接诊科室 ID

       g.dept_name || '(' ||

substr(e.clinic_type, 1, instr(e.clinic_type, '号', 1) - 1) || ')' as "deptName", -- 接诊科室名
称

       b.doctor_no as "doctorId", -- 接诊医生 ID

       (select s.name

        from comm.staff_dict s

        where s.user_name = b.doctor_no) as "doctorName", -- 接诊医生姓名

       round(sum(d.costs), 2) * 100 as "payAmount", -- 已缴费总金额, 单位: 分

       d.bill_date as "chargeDate", --收费日期

       f.money_type as "payMode", --支付方式,

       decode(d.charge_indicator, '0', '待支付', '1', '已支付') as "status", --订单状态,

       null as "dispenWininfor",

       null as "mtguideinfor",

       d.rcpt_no as "receiptId"

FROM OUTPADM.CLINIC_MASTER      a,

     OUTPDOCT.OUTP_ORDERS      b,

     OUTPDOCT.OUTP_ORDERS_COSTS d,

     MEDREC.Pat_Master_Index    c,

     OUTPADM.CLINIC_INDEX       e,

     OUTPBILL.OUTP_PAYMENTS_MONEY f,

     COMM.dept_dict             g

WHERE (b.VISIT_DATE = a.VISIT_DATE)

and (b.VISIT_NO = a.VISIT_NO)

and d.serial_no = b.SERIAL_NO

```



```

and (b.patient_id = c.patient_id)
and (a.clinic_label = e.clinic_label)
and d.rcpt_no = f.rcpt_no
and e.clinic_dept = g.dept_code
and d.order_class in
    ('A', 'B', 'C', 'Z', 'E', 'F', 'G', 'H', 'J', 'D', 'I')
and a.patient_id = patMedNo
and a.Visit_Date >= to_date(beginDate, 'yyyy-mm-dd')
and a.Visit_Date <= to_date(endDate, 'yyyy-mm-dd')
and d.charge_indicator = '1'
group by b.PATIENT_ID,
    c.name,
    a.clinic_no,
    a.visit_no,
    a.visit_dept,
    e.clinic_type,
    b.doctor_no,
    --b.DOCTOR,
    d.bill_date,
    f.money_type,
    d.charge_indicator,
    g.dept_name,
    d.rcpt_no
order by d.rcpt_no desc;
end;

-- 5.1.5 Paid Record Detail Query Interface 001 已缴费记录明细查询接口 001
procedure getPayDetailInfo001(visitId in varchar2, -- (必传) HIS 就诊流水号
    aCur_temp out sys_refcursor) is

```

```

/* 输出结果
* 字段      描述
* -----
detailTypeId  费用类别 Id
detailFee     缴费费别
detailId      缴费细目流水号，要求唯一
detailName    缴费细目名称
detailCount   缴费细目数量
detailUnit    缴费细目单位
detailAmount  缴费细目金额，单位：分
detailSpec    缴费规格
detailPrice   缴费细目单价，单位：分
*/

```

```
begin
```

```
open aCur_temp for
```

```
SELECT d.item_class as "detailTypeId", -- 费用类别 Id
```

```
(case
```

```
when d.item_class = 'A' then
```

```
'西药费'
```

```
when d.item_class = 'B' then
```

```
'中药费'
```

```
when d.item_class = 'C' then
```

```
'检验费'
```

```
when d.item_class = 'D' then
```

```
'检查费'
```

```
else
```

```

    '处置费'

end) as "detailFee", -- 缴费费别

d.serial_no as "detailId", -- 缴费细目流水号，要求唯一

d.item_name as "detailName", -- 缴费细目名称

d.amount as "detailCount", -- 缴费细目数量

d.units as "detailUnit", -- 缴费细目单位

d.costs * 100 as "detailAmount", -- 缴费细目金额，单位：分

d.item_spec as "detailSpec", -- 缴费规格

round(d.costs / ((case
    when d.item_class = 'B' then
        decode(nvl(d.repetition, 1), 0, 1, nvl(d.repetition, 1))
    else
        1
    end) * d.amount),

2) * 100 as "detailPrice", -- 缴费细目单价，单位：分

(select s.dept_name
 from dept_dict s
 where s.dept_code = d.performed_by) as "performDept"
FROM OUTPADM.CLINIC_MASTER    a,
    OUTPDOCT.OUTP_ORDERS    b,
    OUTPDOCT.OUTP_ORDERS_COSTS d,
    MEDREC.Pat_Master_Index  c,
    OUTPADM.CLINIC_INDEX     e
WHERE (b.VISIT_DATE = a.VISIT_DATE)
and (b.VISIT_NO = a.VISIT_NO)
and (b.patient_id = c.patient_id)
and (a.clinic_label = e.clinic_label)
--and d.order_class in

```

```

-- ('A', 'B', 'C', 'Z', 'E', 'F', 'G', 'H', 'J', 'D', 'I')
and d.serial_no = b.SERIAL_NO
and d.rcpt_no = visitId
and d.charge_indicator = '1'
and a.visit_date>sysdate-10
and b.visit_date>sysdate-10;
end;

-- 5.2.1 Exceptional Order Refund Application 001 异常订单退费申请 001
procedure applyRefund001(hospitalId in varchar2, -- (必传) 医院 ID
                        busType in varchar2, -- (必传) 业务类型 (YYGH-预约挂号缴费, DBGH-当
班挂号缴费, MZJF-门诊缴费, ZYYJ-住院押金缴费)
                        orderPayId in varchar2, -- 微信、支付宝商户号生成的交易流水号
                        orderId in varchar2, -- 第三方平台生成的订单 ID
                        payTime in varchar2, -- 交易时间
                        refundAmount in varchar2, -- 退费金额
                        aCur_temp out sys_refcursor) is
/* 输出结果
* 字段      描述
* -----
on_return    结果代码
on_message   结果描述
refundFlag   退费标志(Y-his 订单状态为已退费, N-his 订单为正常状态)
canRefund    是否存在该 orderPayId, 存在则不允许退费 (Y-是, N-否)
*/
n_record number;

begin

```

```

select count(1)
into n_record
from wxgh.wx_trans_his_log t
where 1 = 1 --t.order_type = busType
      and t.order_id = orderId
      and t.order_id_pay = orderPayId;
-- and t.order_pay_time = payTime;

if n_record > 0 then
  open aCur_temp for
    select orderPayId as "orderPayId", 'N' as "canRefund" from dual;
else
  open aCur_temp for
    select orderPayId as "orderPayId", 'Y' as "canRefund" from dual;
end if;

end;

-- 6.1.1 Laboratory Report List Query 001 检验报告列表查询 001
procedure getLisReportList001(hospitalId in varchar2, -- (必传) 医院 ID
                             patMedNo in varchar2, -- 患者诊疗卡号码
                             beginDate in varchar2, -- 查询开始日期, 格式: YYYY-MM-DD
                             endDate in varchar2, -- 查询结束日期, 格式: YYYY-MM-DD
                             aCur_temp out sys_refcursor) is
  /* 输出结果
  * 字段      描述
  * -----
  * reportId  检验报告 id
  */

```

```

reportNo    检验报告 no
reportType  报告单类别（超声、放射…）
patType     病人类型（门诊、住院、体检）
patName     病人姓名
patSex      性别（男/女）
patAge      年龄
reportName  报告单名称
diagnosis   临床诊断
applyDeptName 申请科室名称
applyDrName 申请医生姓名
specimenName 检验标本名称
reportStatus 报告审核状态 中文
collectTime 标本采集时间 YYYY-MM-DD HI24:MI
receiveTime 标本接收时间 YYYY-MM-DD HI24:MI
receiverName 接收人姓名
reportTime  报告时间 YYYY-MM-DD HI24:MI
reportDrName 报告人
*/
v_beginDate varchar2(20);
v_endDate   varchar2(20);

begin
v_beginDate := beginDate || ' 00:00:00';
v_endDate   := endDate || ' 23:59:59';

open aCur_temp for
select a.test_no as "reportId", -- 检验报告 id
       a.test_no as "reportNo", -- 检验报告 no

```

```

a.subject as "reportType", -- 报告单类别（超声、放射…）
(case
  when a.visit_id = '0' then
    '门诊'
  else
    '住院'
end) as "patType", -- 病人类型（门诊、住院、体检）
a.name as "patName", -- 病人姓名
a.sex as "patSex", -- 性别（男/女）
a.age as "patAge", -- 年龄
nvl(a.subject, a.specimen) as "reportName", -- 报告单名称
replace( replace( a.relevant_clinic_diag,'>','大于'), '<','小于') as "diagnosis", -- 临床诊断
b.dept_name as "applyDeptName", -- 申请科室名称
a.ordering_provider as "applyDrName", -- 申请医生姓名
a.specimen as "specimenName", -- 检验标本名称
(case a.result_status
  when '1' then
    '收到申请'
  when '2' then
    '已执行'
  when '3' then
    '初步报告'
  when '4' then
    '确认报告'
  when '6' then
    '条码已打'
  when '9' then
    '其他'

```

```

        else
        '0'
    end) as "reportStatus", -- 报告审核状态 中文
    a.spcm_received_date_time as "collectTime", -- 标本采集时间 YYYY-MM-DD HH24:MI
    a.spcm_sample_date_time as "receiveTime", -- 标本接收时间 YYYY-MM-DD HH24:MI
    null as "receiverName", --接收人姓名
    a.results_rpt_date_time as "reportTime", -- 报告时间 YYYY-MM-DD HH24:MI
    a.transcriptionist as "reportDrName" -- 报告人
from LAB.lab_test_master a, COMM.dept_dict b
where a.ordering_dept = b.dept_code(+)
and a.result_status in ('3', '4')
and a.patient_id IN
    (select patient_id
     from pat_master_index p
     where p.id_no = (select id_no
                     from pat_master_index
                     where patient_id = patMedNo))
and a.results_rpt_date_time >=
    to_date(v_beginDate, 'yyyy-mm-dd hh24:mi:ss')
and a.results_rpt_date_time <=
    to_date(v_endDate, 'yyyy-mm-dd hh24:mi:ss')
order by a.results_rpt_date_time desc;

end;

```

-- 6.1.2 Laboratory Report Detail Query 002 检验报告明细查询 002

```

procedure getLisReportDetail002(hospitalId in varchar2, -- （必传）医院 ID
                                reportId in varchar2, -- 检验报告 id

```



```

aCur_temp out sys_refcursor) is

/* 输出结果
* 字段      描述
* -----
*
itemId      结果指标 id
itemEnName  结果指标英文 name
itemCnName  结果指标中文 name
itemValue   检测结果
itemRef     参考值范围
itemUnit    计量单位
resultNote  异常提示
orderNum    打印顺序，输出结果按排序号排好序，界面直接显示
*/

begin

open aCur_temp for

select a.report_item_code as "itemId", -- 结果指标 id
       a.report_item_code as "itemEnName", -- 结果指标英文 name
       a.report_item_name as "itemCnName", -- 结果指标中文 name
       '<![CDATA[' || a.result || ']]>' as "itemValue", -- 检测结果
       '<![CDATA[' || a.print_context || ']]>' as "itemRef", -- 参考值范围
       a.units as "itemUnit", -- 计量单位
       (case a.abnormal_indicator
         when 'L' then
           ' ↓ '
         when 'H' then
           ' ↑ '

```

```

else
    "

end) as "resultNote", -- 异常提示

a.print_order as "orderNum" -- 打印顺序，输出结果按排序号排好序，界面直接显示

from LAB.lab_result a, LAB.lab_test_master b
where a.test_no = b.test_no
AND a.test_no = reportId
/* B.PATIENT_ID IN (select patient_id
    from pat_master_index p
    where p.id_no = (select DISTINCT id_no
        from pat_master_index P,LAB_TEST_MASTER L WHERE
P.PATIENT_ID=L.PATIENT_ID
        AND L.TEST_NO = reportId))*/
-- and b.test_no = reportId
order by a.print_order;

end;

```

-- 6.2.1 Examination Report List Query 002 检查报告列表查询 002

```

procedure getPacsReportList002(hospitalId in varchar2, -- （必传）医院 ID
    patMedNo in varchar2, -- 患者诊疗卡号码
    beginDate in varchar2, -- 查询开始日期，格式：YYYY-MM-DD
    endDate in varchar2, -- 查询结束日期，格式：YYYY-MM-DD
    aCur_temp out sys_refcursor) is

/* 输出结果
* 字段      描述
* -----
reportId    检查报告 id
reportNo    检查报告 no

```

reportType 报告单类别（超声、放射…）
 patType 病人类型（门诊、住院、体检）
 patName 病人姓名
 patSex 性别
 patAge 年龄
 reportName 报告单名称
 diagnosis 临床诊断
 applyDeptName 申请科室名称
 applyDrName 申请医生姓名
 reportStatus 报告审核状态 中文
 reportTime 报告时间
 reportDrName 报告人
 checkPart 检查部位
 checkMethod 检查方法
 reportSee 检查所见，为防止非法字符错误，将原始内容头尾用“<![CDATA[”和“]]>”标签括住
 reportResult 检查结论，为防止非法字符错误，将原始内容头尾用“<![CDATA[”和“]]>”标签括住
 */

v_beginDate varchar2(20);

v_endDate varchar2(20);

begin

v_beginDate := beginDate || ' 00:00:00';

v_endDate := endDate || ' 23:59:59';

open aCur_temp for

select a.exam_no as "reportId", -- 检查报告 id

```

a.exam_no as "reportNo", -- 检查报告 no
a.exam_class as "reportType", -- 报告单类别（超声、放射…）
(case a.patient_source
  when '1' then
    '门诊'
  when '2' then
    '住院'
  when '3' then
    '外来'
  else
    '0'
end) as "patType", -- 病人类型（门诊、住院、体检）
a.name as "patName", -- 病人姓名
a.sex as "patSex", -- 性别
a.date_of_birth as "patAge", -- 年龄
a.exam_class as "reportName", -- 报告单名称
a.clin_diag as "diagnosis", -- 临床诊断
d.dept_name as "applyDeptName", -- 申请科室名称
a.req_physician as "applyDrName", -- 申请医生姓名
(case a.result_status
  when '1' then
    '收到申请'
  when '2' then
    '已执行'
  when '3' then
    '初步报告'
  when '4' then
    '确认报告'

```

```

when '6' then
    '条码已打'
when '9' then
    '其他'
else
    '0'
end) as "reportStatus", -- 报告审核状态 中文*/
a.report_date_time as "reportTime", -- 报告时间
a.reporter as "reportDrName", -- 报告人
a.exam_sub_class as "checkPart", -- 检查部位
(case a.exam_class
    when 'A 硬化' then
        '脑血流图'
    when 'C T' then
        'CT'
    when 'DR' then
        'X 线'
    when 'DSCT' then
        'CT'
    when 'TCD' then
        '经颅多普勒'
    when 'X 射线' then
        'X 线'
    when '病理' then
        '病理'
    when '彩超' then
        '多普勒'
    when '磁共振' then

```

'磁共振'

when '定位 CT' then

'CT'

when '放疗室' then

'立体定向放射治疗'

when '肺功能' then

'胸外肺功能'

when '感染科' then

'综合'

when '骨密度' then

'骨密度仪'

when '核医学' then

'核医'

when '肌电图' then

'机电仪'

when '肌脑电' then

'肌脑电仪'

when '介入' then

'X 线'

when '钼钯' then

'X 线'

when '内窥镜' then

'消化内镜'

when '脑电图' then

'脑电仪'

when '热成像' then

'热成像仪'

when '心电图' then

```

        '心电图'

        when '阴超' then

        '阴道 B 超'

        when '支纤镜' then

        '纤维支气管镜'

        else

        '其他'

    end) as "checkMethod", -- 检查方法

    '<![CDATA[' || b.DESRIPTION || ']]>' as "reportSee", -- 检查所见，为防止非法字符
    错误，将原始内容头尾用 “<![CDATA[” 和 “]]>” 标签括住

    '<![CDATA[' || b.IMPRESSON || ']]>' as "reportResult" -- 检查结论，为防止非法字符
    错误，将原始内容头尾用 “<![CDATA[” 和 “]]>” 标签括住

    from EXAM.EXAM_MASTER a, EXAM.EXAM_REPORT b, COMM.dept_dict d

    where a.exam_no = b.exam_no

    and a.req_dept = d.dept_code


    and a.patient_id IN

    (select patient_id

    from pat_master_index p

    where p.id_no = (select id_no

    from pat_master_index

    where patient_id = patMedNo))


    and a.report_date_time >=

    to_date(v_beginDate, 'yyyy-mm-dd hh24:mi:ss')

    and a.report_date_time <=

    to_date(v_endDate, 'yyyy-mm-dd hh24:mi:ss');

end;

```

```

--患者建档 002

procedure newCard002(as_xmlParm in varchar2,
                    aCur_temp out sys_refcursor,
                    on_return out number,
                    on_message out varchar2) is
/* 输出结果
* 字段      描述
* -----
patIdCard    身份证号
patId        患者主索引 id
opPatNo      门诊病人号
ipPatNo      住院病人号
patMedNo     就诊卡号
homeAddress  患者住址
*/
--patMedNo varchar2(20);

patName      varchar2(20);
patIdCard    varchar2(18);
patMobile    varchar2(19);
patSex       varchar2(4);
patBirth     varchar2(10);
patNamePhonetic varchar2(40);
patNation    varchar2(10);
patBirthPlace varchar2(6);

relataionship varchar2(2);

```



```

guardName varchar2(20);
guardMobile varchar2(16);
guardIdCard varchar2(18);
homeAddress varchar2(200);

payAgentId varchar2(10);

n_record number;
n_likerecord number;
n_opera_sum number;

v_patId varchar2(20);
v_serialno varchar2(6);
v_tmp_seq varchar2(10);

v_opera_no varchar2(20); --操作工号
v_opera_name varchar2(20); --操作人

d_now date; --当前系统时间

n_current_no number;
begin

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patIdCard') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-1',
    'newCard002',
    as_xmlParm,
    '进入 newCard002-01' ||

```

```

        of_get_xmlval(as_xmlParm, 'patIdCard'),
        null,
        null);

on_return := 0;
on_message := '成功';

patName      := of_get_xmlval(as_xmlParm, 'patName');
patIdCard    := of_get_xmlval(as_xmlParm, 'patIdCard');
patMobile    := of_get_xmlval(as_xmlParm, 'patMobile');
patSex       := of_get_xmlval(as_xmlParm, 'patSex');
patBirth     := of_get_xmlval(as_xmlParm, 'patBirth');
patNamePhonetic := of_get_xmlval(as_xmlParm, 'patNamePhonetic');
patNation    := of_get_xmlval(as_xmlParm, 'patNation');
patBirthPlace := of_get_xmlval(as_xmlParm, 'patBirthPlace');

relataionship := of_get_xmlval(as_xmlParm, 'relataionship');
guardName     := of_get_xmlval(as_xmlParm, 'guardName');
guardMobile   := of_get_xmlval(as_xmlParm, 'guardMobile');
guardIdCard   := of_get_xmlval(as_xmlParm, 'guardIdCard');
homeAddress   := of_get_xmlval(as_xmlParm, 'homeAddress');

payAgentId := of_get_xmlval(as_xmlParm, 'payAgentId');

if patIdCard is null then
    on_return := -1;
    on_message := '参数 patIdCard 为空';
    goto end_program;
end if;

```

if patName is null then

on_return := -1;

on_message := '参数 patName 为空';

goto end_program;

end if;

if patMobile is null and guardMobile is null then

on_return := -1;

on_message := '参数 patMobile 和 guardMobile 不能同时为空';

goto end_program;

end if;

if patSex is null then

on_return := -1;

on_message := '参数 patSex 为空';

goto end_program;

end if;

if patBirth is null then

on_return := -1;

on_message := '参数 patBirth 为空';

goto end_program;

end if;

if patBirthPlace is null then

on_return := -1;

on_message := '参数 patBirthPlace 为空';

goto end_program;

end if;

if patMobile is not null and length(patMobile) > 11 then

on_return := -1;

on_message := 'patMobile 长度不正确';

goto end_program;

end if;

if guardMobile is not null and length(guardMobile) > 11 then

on_return := -1;

on_message := 'guardMobile 长度不正确';

goto end_program;

end if;

select count(1)

into n_record

from MEDREC.Pat_Master_Index b

where 1 = 1

and b.name = patName

and b.id_no = patIdCard

and (b.phone_number_home = patMobile or

b.phone_number_home = guardMobile)

and length(b.patient_id) >= 9;

if n_record > 0 then

on_return := 0;

on_message := '成功';

goto end_program;

else

```

select nvl(max(substr(t.patient_id, 4)), 0)
into v_serialno
from MEDREC.Pat_Master_Index t
where t.patient_id like '9' || to_char(sysdate, 'yy') || '%'
and t.create_date is not null;

if v_serialno = 0 then
select lpad('1', 6, '0') into v_tmp_seq from dual;
v_patId := '9' || to_char(sysdate, 'yy') || v_tmp_seq;
else
v_serialno := to_number(v_serialno) + 1;
select lpad(v_serialno, 6, '0') into v_tmp_seq from dual;
v_patId := '9' || to_char(sysdate, 'yy') || v_tmp_seq;
end if;

/*--构造患者 patient_id
select t.id_currently_value
into n_current_no
from auto_setting_id t
where t.type_name = '门诊病人';

v_patId := '0' || to_char(sysdate, 'yy') || n_current_no;*/

/*begin
update auto_setting_id t
set t.id_currently_value = t.id_currently_value + 1
where t.id_currently_value = n_current_no
and t.type_name = '门诊病人';

```

```

if sql%rowcount <> 1 then

    ROLLBACK;

    on_return := -1;

    on_message := '更新主索引当前值失败，[' || sql%rowcount || ']条记录被更新，取出
当前值' ||

        n_current_no;

    goto end_program;

end if;

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patIdCard') || '-' ||

    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',

    'newCard002',

    as_xmlParm,

    'newCard002 更新主索引当前值-02' ||

    of_get_xmlval(as_xmlParm, 'patIdCard') || '-' ||

    sql%rowcount || ']条记录被更新，取出当前值' ||

    n_current_no,

    on_return,

    on_message);

exception

when others then

    ROLLBACK;

    on_return := -1;

    on_message := '写入患者主索引记录出错' || SQLERRM;

    goto end_program;

end;*/

-- 取操作工号

```

```

select count(1)
into n_opera_sum
from COMM.staff_dict a
where 1 = 1 --job = '自助收款'
      and user_name = upper(payAgentId);
if n_opera_sum > 0 then
select a.id, a.name
into v_opera_no, v_opera_name
from COMM.staff_dict a
where 1 = 1 --job = '自助收款'
      and user_name = upper(payAgentId)
      and rownum = 1;
else
v_opera_no := upper(payAgentId);
v_opera_name := upper(payAgentId);
end if;

begin
insert into MEDREC.PAT_MASTER_INDEX
(PATIENT_ID,
NAME,
NAME_PHONETIC,
SEX,
DATE_OF_BIRTH,
CITIZENSHIP,
NATION,
ID_NO,
IDENTITY,
CHARGE_TYPE,

```

```

PHONE_NUMBER_HOME,
NEXT_OF_KIN,
RELATIONSHIP,
NEXT_OF_KIN_PHONE,
CREATE_DATE,
OPERATOR,
BIRTH_PLACE,
MAILING_ADDRESS,
VIP_INDICATOR,
ISSU_FLAG)
values
(v_patId,
patName,
substr(patNamePhonetic, 1, 16),
patSex,
to_date(patBirth, 'yyyy-mm-dd'),
'CN',
patNation,
patIdCard,
'一般人员',
'自费',
patMobile,
guardName,
relataionship,
guardMobile,
sysdate,
v_opera_name,
substr(patIdCard, 1, 6),
homeAddress,

```



```

2,
'1');
exception
when others then

    ROLLBACK;

    on_return := -1;

    on_message := '写入患者主索引记录出错' || SQLERRM;

    goto end_program;

end;
end if;

<<end_program>>

if on_return = 0 and n_record > 0 then
    insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patIdCard') || '-' ||
        to_char(systimestamp, 'yyyymmddhh24missxff3') || '-3',
        'newCard002',
        as_xmlParm,
        '[有记录]结束 newCard002-03' ||
        of_get_xmlval(as_xmlParm, 'patIdCard'),
        on_return,
        n_record);

    open aCur_temp for
        select b.id_no      as "patIdCard", -- 身份证号
               b.Patient_ID as "patMedNo", -- 患者主索引 id
               b.name       as "patName",
               b.sex        as "patSex", -- 性别
               b.phone_number_home as "patMobile", -- 本人联系电话

```

```

b.next_of_kin    as "guardName", -- 联系人
b.next_of_kin_phone as "guardMobile", -- 联系人电话
b.mailing_address as "homeAddress" -- 患者住址
from (select case
        when t.create_date is null then
            to_date('1900-01-01', 'yyyy-mm-dd')
        else
            t.create_date
    end,
    t.id_no,
    t.patient_id,
    t.name,
    t.sex,
    t.phone_number_home,
    t.next_of_kin,
    t.next_of_kin_phone,
    t.mailing_address
    from pat_master_index t
    where id_no = patIdCard
        and name = patName
        and length(t.patient_id) >= 9
    order by t.Patient_ID desc) b
where rownum = 1;
elsif on_return = 0 and n_record = 0 then
insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patIdCard') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-3',
    'newCard002',
    as_xmlParm,
    '[新建]结束 newCard002-03' ||

```

```

        of_get_xmlval(as_xmlParm, 'patIdCard'),
        on_return,
        v_patId);
open aCur_temp for
select b.id_no      as "patIdCard", -- 身份证号
       b.Patient_ID as "patMedNo", -- 患者主索引 id
       b.name       as "patName",
       b.sex        as "patSex", -- 性别
       b.phone_number_home as "patMobile", -- 本人联系电话
       b.next_of_kin as "guardName", -- 联系人
       b.next_of_kin_phone as "guardMobile", -- 联系人电话
       b.mailing_address as "homeAddress" -- 患者住址
from MEDREC.Pat_Master_Index b
where 1 = 1
and b.name = patName
and b.id_no = patIdCard
and b.patient_id = v_patId
and (b.phone_number_home = patMobile or
     b.phone_number_home = guardMobile)
and rownum = 1
order by b.create_date desc, b.Patient_ID desc;
else

insert_wx_webservice_err_logs('newCard002' ||
                               to_char(systimestamp,
                                         '-yyyymmddhh24missxff3'),
                               'newCard002',
                               as_xmlParm,

```

```

        ",
        on_message);

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patIdCard') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-3',
    'newCard002',
    as_xmlParm,
    '[错]结束 newCard002-03' ||
    of_get_xmlval(as_xmlParm, 'patIdCard'),
    on_return,
    on_message);

open aCur_temp for
select null as "patIdCard", -- 身份证号
       null as "patMedNo", -- 患者主索引 id
       null as "patName",
       null as "patSex", -- 性别
       null as "patMobile", -- 本人联系电话
       null as "guardName", -- 联系人
       null as "guardMobile", -- 联系人电话
       null as "homeAddress" -- 患者住址
from dual;

end if;

end;

--门诊排队信息查询接口 001
procedure getOpQueuingInfo001(patMedNo in varchar2,
    aCur_temp out sys_refcursor) is

```

begin

open aCur_temp for

```
select t.clinic_no "visitId",
       t.name "patName",
       to_char(t.visit_date, 'yyyy-mm-dd') "visitDate",
       t.clinic_label "clinicLabel",
       --t.visit_time_desc "timeDesc",
       case
         when length(t.visit_time_desc) > 4 then
           e.regtime || '(' || substr(t.visit_time_desc, 1, 2) || ':' ||
           substr(t.visit_time_desc, 3, 2) || '-' ||
           substr(t.visit_time_desc, 5, 2) || ':' ||
           substr(t.visit_time_desc, 7, 2) || ')'
         else
           t.visit_time_desc
       end as "timeDesc",
       t.serial_no "queuingSn",
       (select count(1)
        from clinic_master a
        where a.serial_no < t.serial_no
        and a.visit_date = t.visit_date
        and a.clinic_label = t.clinic_label
        and a.visit_time_desc = t.visit_time_desc
        and a.registration_status = '1'
        and a.returned_date is null) "waitPersons",
       s.serial_no "callSn"
from clinic_master t,
```

```

        outpbill.period_vs_regtime e,
        (select c.visit_date,
            c.clinic_label,
            c.visit_time_desc,
            c.serial_no,
            c.suq
        from (select b.visit_date,
            b.clinic_label,
            b.visit_time_desc,
            b.serial_no,
            row_number() over(partition by b.clinic_label, b.visit_time_desc order by
b.valid_date desc) suq
        from clinic_master b
        where b.returned_date is null
            and b.registration_status = '2'
            and b.visit_date = trunc(sysdate)) c
        where c.suq = 1) s
where 1 = 1
and t.visit_date = s.visit_date(+)
and t.clinic_label = s.clinic_label(+)
and t.visit_time_desc = s.visit_time_desc(+)
and t.visit_time_desc = e.period_time(+)
and t.patient_id = patMedNo
and t.returned_date is null
and t.visit_date = trunc(sysdate);
end;

```

--住院病人基本情况查询接口 002

```

procedure getlpPatientInfo002(as_xmlParm in varchar2,

```

```

aCur_temp out sys_refcursor,
on_return out number,
on_message out varchar2) is
ipPatNo varchar2(20);
patMedNo varchar2(20);
patName varchar2(50);

visitId      varchar2(20);
visitNum      pat_visit.visit_id%type; --住院次数
deptId      pats_in_hospital.dept_code%type;
deptName      dept_dict.dept_name%type;
bedNo      pats_in_hospital.bed_no%type;
doctorId      pats_in_hospital.doctor_in_charge%type;
doctorName      staff_dict.name%type;
admitDate      pats_in_hospital.admission_date_time%type;
patState      varchar2(30);
totalAmount      pats_in_hospital.total_costs%type;
closedAmount      pats_in_hospital.total_costs%type;
totalDeposit      pats_in_hospital.prepayments%type;
uncoveredDeposit      number;
returncode      number;
returnMes      varchar2(500);
v_charge_type      pat_visit.charge_type%type; --费别
v_insur_personal_id      pat_visit.insur_personal_id%type; --个人卡号
v_insurance_type      pat_visit.insurance_type%type; --身份
v_insur_medical_class_name      pat_visit.insur_medical_class_name%type; --住院类别
v_insur_card_class      varchar2(30); --医保类别

```

```

v_INSUR_COSTS  insur_budget.insur_costs%type; --已经上传费用
v_FreeAmount   insur_budget.insur_costs%type; --医保大概报销
v_INSUR_SELF_PAY insur_budget.insur_self_pay%type; --医保未报销金额
v_syyjj        insur_budget.insur_costs%type; --欠费金额
v_identity     pat_visit.identity%type; --身份

```

```

begin

```

```

    on_return := 0;

```

```

    on_message := '成功';

```

```

    patMedNo := of_get_xmlval(as_xmlParm, 'patMedNo');

```

```

    patName := of_get_xmlval(as_xmlParm, 'patName');

```

```

    ipPatNo := of_get_xmlval(as_xmlParm, 'ipPatNo');

```

```

    if patMedNo is null then

```

```

        on_return := -1;

```

```

        on_message := '没有住院或参数为空!';

```

```

    end if;

```

```

insert into outpbill.outpbill_refund_insur_log

```

```

    (app_name,

```

```

    operator_id,

```

```

    operator_user_name,

```

```

    operator_log,

```



```
operator_time,  
memo,  
XML_BODY)  
values  
( '预交金缴费',  
  '1968',  
  '预交查询',  
  on_message,  
  sysdate,  
  patMedNo,  
  as_xmlParm);commit;
```

```
begin  
  getIpPatientInfo001( --ipPatNo,  
    patmedno,  
    --patname,  
    visitid,  
    visitNum,
```

```
deptid,  
deptname,  
bedno,  
doctorid,  
doctorname,  
admitdate,  
patstate,  
totalamount,  
closedamount,  
totaldeposit,  
uncovereddeposit,  
returncode,  
returnmes,  
v_charge_type,  
v_insur_personal_id,  
v_insurance_type,  
v_insur_medical_class_name,  
v_insur_card_class,  
v_INSUR_COSTS,  
v_FreeAmount,  
v_INSUR_SELF_PAY,  
v_syyjj,  
v_identity);
```

```
on_return := returncode;  
if returncode <> 0 then  
    on_message := returnmes;  
end if;  
exception
```

```

when others then

    ROLLBACK;

    on_return := -1;

    on_message := '调用存储过程 getIpPatientInfo001 出错,' || returnmes ||

        SQLERRM;

    goto end_program;

end;

```

```

<<end_program>>

```

```

if on_return = 0 then

    open aCur_temp for

        select to_char(on_return) "resultCode",

            on_message "resultDesc",

            patmedno as "patId", --病人 ID,

            ipPatNo as "ipPatNo", --住院号,

            patname as "patName", --姓名,

            visitid as "visitId", --当前住院就诊流水号,

            bedno as "bedNo", --床标号,

            v_charge_type as "chargeType", --费别,

            admitdate as "admitDate", --入院时间,

            deptname as "areaName", -- 所在病区,

            nvl(totaldeposit, 0) * 100 as "totalDeposit", --预交金总额,

            nvl(totalamount, 0) * 100 as "totalAmount", --目前费用

            doctorName as "doctorName", --医生姓名

            patState as "patState", -- 在院状态

            v_insur_personal_id as "insurPersonalId", --医保个人电脑号

            v_insurance_type as "insuranceType", --医保身份

```

```

v_insur_medical_class_name as "insurMedicalClass", --住院类别
v_insur_card_class as "insurCardClass", --医保类别
nvl(v_INSUR_COSTS, 0) * 100 as "insurCosts", --已经上传费用
nvl(v_FreeAmount, 0) * 100 as "freeAmount", --医保大概报销
nvl(v_INSUR_SELF_PAY, 0) * 100 as "insurSelfPay", --医保未报销金额
nvl(v_syyjj, 0) * 100 as "owedMoney", --欠费金额

v_identity as "identity" --患者身份

from dual;

else
open aCur_temp for
select to_char(on_return) "resultCode",
       on_message "resultDesc",
       " as "patId", --病人 ID,
       " as "ipPatNo", --住院号,
       " as "patName", --姓名,
       " as "visitId", --当前住院就诊流水号,
       " as "bedNo", --床标号,
       " as "chargeType", --费别,
       " as "admitDate", --入院时间,
       " as "areaName", -- 所在病区,
       " as "totalDeposit", --预交金总额,
       " as "totalAmount", --目前费用
       " as "doctorName", --医生姓名
       " as "patState", -- 在院状态
       " as "insurPersonalId", --医保个人电脑号
       " as "insuranceType", --医保身份
       " as "insurMedicalClass", --住院类别
       " as "insurCardClass", --医保类别

```

```

        " as "insurCosts", --已经上传费用
        " as "freeAmount", --医保大概报销
        " as "insurSelfPay", --医保未报销金额
        " as "owedMoney", --欠费金额
        " as "identity" --患者身份
    from dual;
end if;
end;

--住院押金明细查询接口 001
procedure getIpDepositDetail002(as_xmlParm in varchar2,
                                aCur_temp out sys_refcursor,
                                on_return out number,
                                on_message out varchar2) is

    visitId varchar2(20);
begin
    --on_return := 0;
    --on_message := '成功';
    visitId := of_get_xmlval(as_xmlParm, 'visitId');

begin
    getIpDepositDetail001(visitId, aCur_temp, on_return, on_message);
    if on_return = 0 then
        on_message := '成功';
    end if;

exception

```

```

when others then

    ROLLBACK;

    on_return := -1;

    on_message := '调用存储过程 getIpDepositDetail001 出错' || SQLERRM;

    goto end_program;

end;

```

```

<<end_program>>

```

```

if on_return <> 0 then

    open aCur_temp for

        select " "patId",

            " "visitId",

            " "patName",

            " "payWay",

            " "payAmount",

            " "payDate",

            " "depositStatus"

        from dual;

    end if;

end;

```

--住院当前费用情况查询接口 002

```

procedure getIpChgInfo001(as_xmlParm in varchar2,

    aCur_temp out sys_refcursor,

    on_return out number,

    on_message out varchar2) is

    visitId varchar2(20);

    beginDate varchar2(20);

```

```

endDate varchar2(20);

begin
    on_return := 0;
    on_message := '成功';
    visitId := of_get_xmlval(as_xmlParm, 'visitId');
    beginDate := of_get_xmlval(as_xmlParm, 'beginDate');
    endDate := of_get_xmlval(as_xmlParm, 'endDate');

    begin
        getIpchgInfo002(visitId, aCur_temp, beginDate, endDate);

    exception
        when others then
            ROLLBACK;
            on_return := -1;
            on_message := '调用存储过程 getIpchgInfo002 出错' || SQLERRM;
            goto end_program;
    end;

    <<end_program>>

    if on_return <> 0 then
        open aCur_temp for
            select " "categoryId", " "categoryName", " "categoryAmount"
                from dual;
    end if;
end;

```

--住院一日清单费用分类项目汇总查询接口 001

```
procedure getIpDayExpenseItemTotal002(as_xmlParm in varchar2,  
                                     aCur_temp out sys_refcursor,  
                                     on_return out number,  
                                     on_message out varchar2) is
```

```
visitId  varchar2(20);  
categoryId varchar2(10);  
beginDate varchar2(20);  
endDate  varchar2(20);
```

```
begin  
    on_return := 0;  
    on_message := '成功';  
    visitId := of_get_xmlval(as_xmlParm, 'visitId');  
    categoryId := of_get_xmlval(as_xmlParm, 'categoryId');  
    beginDate := of_get_xmlval(as_xmlParm, 'beginDate');  
    endDate := of_get_xmlval(as_xmlParm, 'endDate');
```

```
begin  
    getIpDayExpenseItemTotal001(visitId,  
                                categoryId,  
                                aCur_temp,  
                                beginDate,  
                                endDate);
```

```
exception  
  
when others then  
    ROLLBACK;
```



```

on_return := -1;

on_message := '调用存储过程 getIpDayExpenseItemTotal001 出错' || SQLERRM;

goto end_program;

end;

```

```

<<end_program>>

```

```

if on_return <> 0 then
  open aCur_temp for
    select " "categoryName",
           " as "detailCode",
           " as "detailName",
           " as "detailSpec",
           " as "detailCount",
           " as "detailUnit",
           " as "detailAmount",
           " as "ybGrade"
    from dual;
end if;

end;

```

--住院押金支付接口 001

```

procedure payIpDeposit002(as_xmlParm in varchar2,
                           aCur_temp out sys_refcursor,
                           on_return out number, -- 结果编码
                           on_message out varchar2 -- 结果描述
                           ) is

  visitId varchar2(20);

  payAmount varchar2(10); --支付金额(单位 “分” )

```

```

patMobile varchar2(30); --患者手机号码
operMobile varchar2(30); -- 挂号人手机号码
orderPayId varchar2(50); --第三方平台的流水号
orderId varchar2(50); --渠道订单号
payAgentId varchar2(20); --虚拟操作员工号
payTime varchar2(20); --交易时间，格式：YYYY-MM-DD HH24:MI:SS
mchId varchar2(30); --商户号。微信商户号、支付宝商户号等
payAppId varchar2(30); --公众账号 ID

n_translog_count number; --交易记录校验结果标识

payMoney number(9, 2);
hisOrderId varchar2(30);
begin
insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'orderId') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-1',
    'payIpDeposit001',
    as_xmlParm,
    '进入 payIpDeposit001-01' ||
    of_get_xmlval(as_xmlParm, 'orderId'),
    null,
    null);

visitId := of_get_xmlval(as_xmlParm, 'visitId');
payAmount := of_get_xmlval(as_xmlParm, 'payAmount');
patMobile := of_get_xmlval(as_xmlParm, 'patMobile');
operMobile := of_get_xmlval(as_xmlParm, 'operMobile');

```

```

orderPayId := of_get_xmlval(as_xmlParm, 'orderPayId');
orderId    := of_get_xmlval(as_xmlParm, 'orderId');
payAgentId := of_get_xmlval(as_xmlParm, 'payAgentId');
payTime    := of_get_xmlval(as_xmlParm, 'payTime');
mchId      := of_get_xmlval(as_xmlParm, 'mchId');
payApplId  := of_get_xmlval(as_xmlParm, 'payApplId');

--是否存在相同一笔交易
select count(1)
into n_translog_count
from wxgh.WX_TRANS_HIS_LOG t
where t.order_id = orderId
and t.order_id_pay = orderPayId;
if n_translog_count > 0 then
on_return := -1;
on_message := '订单号' || orderId || '的交易已经存在';
goto end_program;
end if;

payMoney := payAmount / 100;

begin
payIpDeposit001(visitId, payMoney, on_return, on_message, hisOrderId);
if on_return = 0 then
on_message := '成功';
end if;
exception
when others then
ROLLBACK;

```

```

on_return := -1;

on_message := '调用存储过程 paylpDeposit001 出错' || SQLERRM;

goto end_program;

end;

-- 写入微信交易日志表

begin

insert into wxgh.wx_trans_his_log

(PATIENT_ID,

PATIENT_MOBILE,

ORDER_TYPE,

ORDER_ID,

ORDER_ID_PAY,

DEAL_MONEY,

PAY_APP_ID,

MCH_ID,

ORDER_PAY_TIME,

ORDER_DESC,

RCPT_NO,

SYS_DATETIME)

values

(substr(visitId, 0, 9),

nvl(patMobile, operMobile),

'INPRECHG',

orderId,

orderPayId,

payAmount,

payAppId,

mchId,

```

```

    payTime,
    null,
    hisOrderId,
    sysdate);
exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '写入微信交易日志表失败:' || SQLERRM;
    goto end_program;
end;

<<end_program>>

if on_return = 0 then
    insert_wx_ps_step_log(orderId || '-' ||
        to_char(systimestamp, 'yyyymmddhh24missxff3') || '-3',
        'paylpDeposit001',
        as_xmlParm,
        '结束 paylpDeposit001-03' || orderId,
        on_return,
        on_message);
    open aCur_temp for
        select to_char(on_return) as "resultCode",
            on_message as "resultDesc",
            hisOrderId as "receiptId"
        from dual;
else

```

```

insert_wx_webservice_err_logs('paylpDeposit001' ||
    to_char(systimestamp,
        '-yyyymmddhh24missxff3'),
    'paylpDeposit001',
    as_xmlParm,
    ",
    on_message);

insert_wx_ps_step_log(orderId || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-3',
    'paylpDeposit001',
    as_xmlParm,
    '[错]结束 paylpDeposit001-03' || orderId,
    on_return,
    on_message);

open aCur_temp for
    select to_char(on_return) as "resultCode",
        on_message as "resultDesc",
        null as "receiptId"
    from dual;
end if;
end;

-- 院内医保药品对照接口
procedure getPriceVsInsurDict(pageSize in number, -- 每页行数
    currentPage in number, -- 当前页码
    aCur_temp out sys_refcursor) is
    v_index number;
begin

```

v_index := 1;

open aCur_temp for

```
select item_class      as "itemClass",
       item_code       as "itemCode",
       item_spec       as "itemSpec",
       insur_item_class as "insurItemClass",
       insur_item_code  as "insurItemCode",
       insur_item_name  as "insurItemName",
       insur_item_spec  as "insurItemSpec",
       grade           as "grade",
       special_flag     as "specialFlag",
       units           as "units",
       qyzt            as "qyzt",
       insur_firm_id    as "insurFirmId",
       insur_center_code as "insurCenterCode",
       insur_item_units as "insurItemUnits",
       insur_item_class_bg as "insurItemClassBg",
       insur_item_code_bg as "insurItemCodeBg",
       insur_item_name_bg as "insurItemNameBg",
       insur_item_spec_bg as "insurItemSpecBg",
       grade_bg        as "gradeBg",
       insur_firm_id_bg as "insurFirmIdBg",
       insur_item_units_bg as "insurItemUnitsBg",
       totalPage       as "totalPage",
       med_chrgitm_type as "medChrgitmType",
       decode (item_class,'A', actPurcPric,'I', actPurcPric,'B',actPurcPric , '') actPurcPric
from (select rownum as rowno, t2.*
```

```

        from (select current_price_list.price actPurcPric, t1.*,round(((select count(1) as
totalRow from INSURANCE.PRICE_VS_INSUR t1) / pageSize), 0) as totalPage
from INSURANCE.PRICE_VS_INSUR t1,current_price_list
WHERE t1.ITEM_CLASS = current_price_list.ITEM_CLASS AND t1.ITEM_CODE =
current_price_list.ITEM_CODE AND
t1.ITEM_SPEC = current_price_list.ITEM_SPEC
order by t1.insur_item_code, t1.item_code,t1.item_spec) t2
where rownum <= pageSize * currentPage) t3
where t3.rowno >= (currentPage - 1) * pageSize + 1;
end;

-- 查询未设置扫码 url 的挂号记录
procedure getNoSetQrCode(beginDate in varchar2,
                        endDate in varchar2,
                        aCur_temp out sys_refcursor) is

begin
    if beginDate is null and endDate is null then

        open aCur_temp for
            select to_char(t.visit_date, 'yyyy-mm-dd') as "visitDate",
                t.clinic_no as "visitId",
                t.doctor as "doctorId",
                t.visit_dept as "deptId"
            from OUTPADM.clinic_master t
            where t.visit_date >= trunc(sysdate)
                and (t.qr_code is null or t.qr_code = '')
                and t.doctor is not null and t.doctor<>'XCXJF' AND T.CLINIC_LABEL<>'核酸检测专用门诊'
        ;
    else

```



```

open aCur_temp for
select to_char(t.visit_date, 'yyyy-mm-dd') as "visitDate",
       t.clinic_no as "visitId",
       t.doctor as "doctorId",
       t.visit_dept as "deptId"
from OUTPADM.clinic_master t
where t.visit_date >= to_date(beginDate, 'yyyy-mm-dd')
      and t.visit_date <= to_date(endDate, 'yyyy-mm-dd')
      and (t.qr_code is null or t.qr_code = '')
      and t.doctor is not null and t.doctor<>'XCXJF' AND T.CLINIC_LABEL<>'核酸检测专用门诊'
';

end if;

end;

-- 设置挂号记录的扫码 url
procedure setQrCodeByVisitId(as_xmlParm in varchar2,
                             aCur_temp out sys_refcursor,
                             on_return out number,
                             on_message out varchar2) is

visitId varchar2(20);
qrCode varchar2(60);

n_record number;

begin

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
                     to_char(systimestamp, 'yyyymmddhh24missxff3') || '-1',
                     'setQrCodeByVisitId',

```

```

        as_xmlParm,
        '进入 setQrCodeByVisitId-01' ||
        of_get_xmlval(as_xmlParm, 'visitId'),
        null,
        null);

on_return := 0;
on_message := '成功';

visitId := of_get_xmlval(as_xmlParm, 'visitId');
qrCode := of_get_xmlval(as_xmlParm, 'qrCode');

if visitId is null then
    on_return := -1;
    on_message := '参数 visitId 不能为空';
    goto end_program;
end if;

/*select count(1)
into n_record
from OUTPADM.clinic_master t
where 1 = 1
    and t.clinic_no = visitId
    and (t.qr_code is null or t.qr_code = '')
    and t.operator = '1968';

if n_record > 1 then
    on_return := -1;
    on_message := visitId || '的扫码 code 记录已存在';

```

```

        goto end_program;
    end if;*/

begin
    update OUTPADM.clinic_master t
        set t.qr_code = qrCode
    where t.clinic_no = visitId
        and (t.qr_code is null or t.qr_code = '')
        and t.doctor is not null AND t.visit_date >= trunc(sysdate)-1 AND T.CLINIC_LABEL<>'核酸
检测专用门诊' ;

    exception
    when others then
        ROLLBACK;

        on_return := -1;

        on_message := '更新挂号主记录[ || visitId || ]的 QrCode 出错' || SQLERRM;

        goto end_program;
    end;

    <<end_program>>

    if on_return = 0 then

        insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
            to_char(sysimestamp, 'yyyymmddhh24missxff3') || '-2',
            'setQrCodeByVisitId',
            as_xmlParm,
            '结束 setQrCodeByVisitId-02' ||
            of_get_xmlval(as_xmlParm, 'visitId'),
            on_return,

```

```
null);
```

```
open aCur_temp for
```

```
select to_char(on_return) "resultCode",
       on_message "resultDesc",
       to_char(t.visit_date, 'yyyy-mm-dd') as "visitDate",
       t.clinic_no as "visitId",
       t.doctor as "doctorId",
       t.visit_dept as "deptId",
       t.qr_code as "qrCode"
from OUTPADM.clinic_master t
where t.clinic_no = visitId AND t.visit_date >= trunc(sysdate)-3;
```

```
else
```

```
insert_wx_webservice_err_logs('setQrCodeByVisitId' ||
                               to_char(systimestamp,
                                         '-yyyymmddhh24missxff3'),
                               'setQrCodeByVisitId',
                               as_xmlParm,
                               ",
                               on_message);
```

```
insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'visitId') || '-' ||
                      to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
                      'setQrCodeByVisitId',
                      as_xmlParm,
                      '[错]结束 setQrCodeByVisitId-02' ||
                      of_get_xmlval(as_xmlParm, 'visitId'),
```

```

        on_return,
        on_message);

open aCur_temp for
select to_char(on_return) as "resultCode",
       on_message as "resultDesc",
       " as "visitDate",
       " as "visitId",
       " as "doctorId",
       " as "deptId",
       " as "qrCode"
from dual;
end if;

end;

-- 国家电子健康卡绑定
procedure countryEleHCNoBind001(as_xmlParm in varchar2,
                                aCur_temp out sys_refcursor,
                                on_return out number,
                                on_message out varchar2) is

    patMedNo    varchar2(20);
    patName     varchar2(20);
    patIdCard   varchar2(18);
    countryEleHCNo varchar2(80);

    v_patId     varchar2(20);
    v_idNo      varchar2(18);

```

```

v_name      varchar2(20);
v_eleHCNoVsPatId varchar2(20);
v_eleHCNoVsIdNo varchar2(20);
v_eleHCNoVsName varchar2(20);

n_record number;
n_eleRec number;
begin

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patMedNo') || '-' ||
                      to_char(systimestamp, 'yyyymmddhh24missxff3') || '-1',
                      'countryEleHCNoBind001',
                      as_xmlParm,
                      '进入 countryEleHCNoBind001-01' ||
                      of_get_xmlval(as_xmlParm, 'patMedNo'),
                      null,
                      null);

on_return    := 0;
on_message   := '绑定成功';
patMedNo     := of_get_xmlval(as_xmlParm, 'patMedNo');
patName      := of_get_xmlval(as_xmlParm, 'patName');
patIdCard    := of_get_xmlval(as_xmlParm, 'patIdCard');
countryEleHCNo := of_get_xmlval(as_xmlParm, 'countryEleHCNo');

if patMedNo is null and patIdCard is null then
    on_return := -1;
    on_message := '参数 patMedNo 和 patIdCard 不能同时为空';
    goto end_program;

```

```
end if;
```

```
if countryEleHCNo is null then
```

```
    on_return := -1;
```

```
    on_message := '参数 countryEleHCNo 不能为空';
```

```
    goto end_program;
```

```
end if;
```

```
if patMedNo is not null then
```

```
    if length(patMedNo) <> 9 then
```

```
        on_return := -1;
```

```
        on_message := '卡号' || patMedNo || '长度不正确';
```

```
        goto end_program;
```

```
    end if;
```

```
end if;
```

```
if patName is null then
```

```
    on_return := -1;
```

```
    on_message := '参数 patName 为空';
```

```
    goto end_program;
```

```
end if;
```

```
if patMedNo is not null then
```

```
    select count(1)
```

```
        into n_record
```

```
        from MEDREC.Pat_Master_Index b
```

```
        where 1 = 1
```

```
        and b.patient_id = patMedNo
```

```
        and b.name = patName
```

```

        and b.id_no is not null;
else
select count(1)
into n_record
from MEDREC.Pat_Master_Index b
where 1 = 1
and b.name = patName
and b.id_no = patIdCard;
end if;

if n_record < 1 then
on_return := -1;
on_message := '患者信息匹配失败，请至挂号窗口进行确认';
goto end_program;
end if;

if patMedNo is not null then
select b.id_no, b.Patient_ID, b.name
into v_idNo, v_patId, v_name
from MEDREC.Pat_Master_Index b
where 1 = 1
and b.patient_id = patMedNo
and b.name = patName
and b.id_no is not null;
else
select b.id_no, b.Patient_ID, b.name
into v_idNo, v_patId, v_name
from (select case
when t.create_date is null then

```



```

        to_date('1900-01-01', 'yyyy-mm-dd')
    else
        t.create_date
    end,
    t.name,
    t.id_no,
    t.patient_id,
    t.inp_no,
    t.mailing_address
from pat_master_index t
where id_no = patIdCard
    and name = patName
    and length(t.patient_id) >= 9
order by t.Patient_ID desc) b
where rownum = 1;
end if;

-- 首先检查电子健康码是否存在已绑定的门诊号
select count(1)
into n_eleRec
from MEDREC.Pat_Master_Index b
where b.health_card_no = countryEleHCNo;

if n_eleRec = 0 then
    -- 没有绑定记录，则直接绑定
    begin
        update MEDREC.Pat_Master_Index t
        set t.health_card_no = countryEleHCNo
        where t.patient_id = v_patId

```

```

        and t.name = v_name
        and t.id_no = v_idNo;
exception
when others then
    ROLLBACK;
    on_return := -1;
    on_message := '更新患者主索引记录[' || v_patId || ']出错' || SQLERRM;
    goto end_program;
end;
else
-- 比对传入门诊号与已绑定的门诊号是否一致
select b.id_no, b.Patient_ID, b.name
    into v_eleHCNoVsIdNo, v_eleHCNoVsPatId, v_eleHCNoVsName
    from MEDREC.Pat_Master_Index b
    where b.health_card_no = countryEleHCNo;

if patMedNo <> v_eleHCNoVsPatId then
-- 不一致
-- 先解绑原记录
begin
    update MEDREC.Pat_Master_Index t
        set t.health_card_no = null
        where t.patient_id = v_eleHCNoVsPatId
        and t.name = v_eleHCNoVsName
        and t.id_no = v_eleHCNoVsIdNo;
exception
when others then
    ROLLBACK;
    on_return := -1;

```

```

on_message := '更新患者主索引记录[' || v_eleHCNoVsPatId ||
              '], 解绑电子健康卡出错' || SQLERRM;

goto end_program;

end;

-- 绑定当前传入的门诊号记录

begin

update MEDREC.Pat_Master_Index t

set t.health_card_no = countryEleHCNo

where t.patient_id = v_patId

and t.name = v_name

and t.id_no = v_idNo;

exception

when others then

ROLLBACK;

on_return := -1;

on_message := '更新患者主索引记录[' || v_patId || '], 绑定电子健康卡出错' ||

              SQLERRM;

goto end_program;

end;

else

-- 一致则直接跳到返回成功逻辑

on_return := 0;

on_message := '绑定成功';

goto end_program;

end if;

end if;

<<end_program>>

```

if on_return = 0 then

```
insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patMedNo') || '-' ||  
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',  
    'countryEleHCNoBind001',  
    as_xmlParm,  
    '结束 countryEleHCNoBind001-02' ||  
    of_get_xmlval(as_xmlParm, 'patMedNo'),  
    on_return,  
    null);
```

open aCur_temp for

```
select to_char(on_return) as "resultCode",  
    on_message as "resultDesc"  
from dual;
```

else

```
insert_wx_webservice_err_logs('countryEleHCNoBind001' ||  
    to_char(systimestamp,  
        '-yyyymmddhh24missxff3'),  
    'countryEleHCNoBind001',  
    as_xmlParm,  
    "  
    on_message);
```

```
insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patMedNo') || '-' ||  
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
```

```

        'countryEleHCNoBind001',
        as_xmlParm,
        '[错]结束 countryEleHCNoBind001-02' ||
        of_get_xmlval(as_xmlParm, 'patMedNo'),
        on_return,
        on_message);

open aCur_temp for
select to_char(on_return) as "resultCode",
       on_message as "resultDesc"
from dual;
end if;
end;

-- 国家电子健康码查询
procedure countryEleHCNoInfo001(patIdCard in varchar2,
                                patMedNo in varchar2,
                                eleHCNo in varchar2,
                                aCur_temp out sys_refcursor) is

begin

open aCur_temp for

select b.id_no      as "patIdCard",
       b.Patient_ID as "patMedNo",
       b.health_card_no as "eleHCNo"
from MEDREC.Pat_Master_Index b
where 1 = 1

```

```

        and (b.patient_id = patMedNo or b.id_no = patIdCard or
             b.health_card_no = eleHCNo)
        and b.id_no is not null
        and b.health_card_no is not null;

end;

-- 核酸检测项目查询
procedure getHsjcltem001(as_xmlParm in varchar2,
                        aCur_temp out sys_refcursor,
                        on_return out number,
                        on_message out varchar2) is
    regDate varchar2(30);
begin
    on_return := 0;
    on_message := '成功';
    regDate := of_get_xmlval(as_xmlParm, 'regDate');

    begin
        self_help_order(to_date(regDate, 'yyyy-mm-dd'),
                        aCur_temp,
                        on_return,
                        on_message);

    exception
        when others then
            ROLLBACK;

            on_return := -1;

            on_message := '调用存储过程 self_help_order 出错' || SQLERRM;

```

```

        goto end_program;
end;

<<end_program>>

if on_return <> 0 then
    open aCur_temp for
        select " "clinicLabel",
               " "clinicItemCode",
               " "itemName",
               " "price"
        from dual;
    end if;
end;

-- 核酸检测自助登记
procedure selfRegHsjc001(as_xmlParm in varchar2,
                        aCur_temp out sys_refcursor,
                        on_return out number,
                        on_message out varchar2) is
    regDate    varchar2(30);
    patMedNo    varchar2(30);
    clinicItemCode varchar2(30);
    timeDesc    varchar2(30);
    hospitalId  varchar2(30);

    v_clinic_no varchar2(30);
    v_price_sum number(10, 2);
    v_doctor    varchar2(30);

```

```

v_dept_code varchar2(30);

begin

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patMedNo') || '-' ||
    to_char(systimestamp, 'yyyymmddhh24missxff3') || '-1',
    'selfRegHsjc001',
    as_xmlParm,
    '进入 selfRegHsjc001' ||
    of_get_xmlval(as_xmlParm, 'patMedNo'),
    on_return,
    on_message);

on_return    := 0;
on_message   := '成功';
regDate      := of_get_xmlval(as_xmlParm, 'regDate');
patMedNo     := of_get_xmlval(as_xmlParm, 'patMedNo');
clinicItemCode := of_get_xmlval(as_xmlParm, 'clinicItemCode');
timeDesc     := of_get_xmlval(as_xmlParm, 'timeDesc');
hospitalId   := of_get_xmlval(as_xmlParm, 'hospitalId');

begin

if hospitalId is null or hospitalId = '' then
    hospitalId := '0';
end if;

outp_clinic_orders_phone(to_date(regDate, 'yyyy-mm-dd'),
    patMedNo,
    clinicItemCode,

```



```

timeDesc,
to_number(hospitalId),
on_return,
on_message,
v_clinic_no,
v_price_sum,
v_doctor,
v_dept_code);

```

exception

when others then

```
ROLLBACK;
```

```
on_return := -1;
```

```
on_message := '调用存储过程 outp_clinic_orders_phone 出错' || SQLERRM;
```

```
goto end_program;
```

end;

<<end_program>>

if on_return = 0 then

```
insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patMedNo') || '-' ||
```

```
to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
```

```
'selfRegHsjc001',
```

```
as_xmlParm,
```

```
'结束 selfRegHsjc001' ||
```

```
of_get_xmlval(as_xmlParm, 'patMedNo'),
```

```
on_return,
```

```
on_message);
```

```

open aCur_temp for
select to_char(on_return) as "returnCode",
       on_message as "resultDesc",
       v_clinic_no as "visitId",
       v_price_sum * 100 as "priceSum",
       v_doctor as "doctorId",
       v_dept_code as "deptId"
from dual;
else
insert_wx_webservice_err_logs('selfRegHsjc001' ||
                               to_char(systimestamp,
                                         '-yyyymmddhh24missxff3'),
                               'selfRegHsjc001',
                               as_xmlParm,
                               ",
                               on_message);

insert_wx_ps_step_log(of_get_xmlval(as_xmlParm, 'patMedNo') || '-' ||
                      to_char(systimestamp, 'yyyymmddhh24missxff3') || '-2',
                      'selfRegHsjc001',
                      as_xmlParm,
                      '[错]结束 selfRegHsjc001' ||
                      of_get_xmlval(as_xmlParm, 'patMedNo'),
                      on_return,
                      on_message);

open aCur_temp for
select to_char(on_return) as "returnCode",
       on_message as "resultDesc",

```

```

        " as "visitId",
        " as "priceSum",
        v_doctor as "doctorId",
        v_dept_code as "deptId"
    from dual;
end if;
end;

-- 取 XML 串里的标记值
function of_get_xmlval(as_xml_string in varchar2, -- xml 串
    as_xml_mark in varchar2 -- xml 段标记
) return varchar2 is
    vs_bmark varchar2(64);
    vs_emark varchar2(64);
    vs_source varchar2(10240);
    vs_xmlval varchar2(256);
    vi_bpos integer;
    vi_epos integer;
begin

    vs_bmark := '<' || lower(as_xml_mark) || '>';
    vs_emark := '</' || lower(as_xml_mark) || '>';
    vs_source := lower(as_xml_string);

    vi_bpos := INSTR(vs_source, vs_bmark);
    vi_epos := INSTR(vs_source, vs_emark);

    if vi_bpos > 0 and vi_epos > 0 then
        vi_bpos := vi_bpos + length(vs_bmark);
    end if;
end;

```

```

vs_xmlval := SUBSTR(as_xml_string, vi_bpos, vi_epos - vi_bpos);
end if;

return vs_xmlval;

end;

-- 患者挂号规则
-- 1. 男性不允许挂妇科、产科
-- 2. 14 以上（含）不允许挂儿科
function check_reg_rule(patientId in varchar2,
                        clinicLabel in varchar2,
                        timeDesc in varchar2,
                        regDate in date) return varchar2 is
v_return varchar2(4);
v_sex varchar2(4);
v_birth date;
v_age number;

v_deptId varchar2(10);
v_label varchar2(20);
begin
v_return := '0';
-- 110201 产科普通门诊, 110202 产科专家门诊, 110203 产科特需门诊
-- 100201 妇科普通门诊, 100202 妇科专家门诊, 100203 妇科特需门诊
-- 130201 儿科普通门诊, 130202 儿科专家门诊, 130203 儿科特需门诊, 130204 儿科急诊门
诊
---
```

```

select s.clinic_dept, t.clinic_label
into v_deptId, v_label
from clinic_for_regist t, clinic_index s
where t.clinic_label = s.clinic_label
and t.clinic_date = regDate
and t.clinic_label = clinicLabel
and t.time_desc = timeDesc;

```

```

select t.sex, t.date_of_birth
into v_sex, v_birth
from pat_master_index t
where t.patient_id = patientId;

```

-- 判断是否可以挂妇科、产科

```

if v_deptId like '10%' or v_deptId like '11%' then
    if v_sex = '男' then
        v_return := '1';
    end if;
end if;

```

-- 判断是否可以挂儿科

```

if v_deptId like '13%' then
---  select Trunc(MONTHS_BETWEEN(regDate, v_birth) / 12)
        select regDate - ( v_birth + interval '14' year)
        into v_age
        from dual;
    if v_age >= 1 then
        v_return := '2';
    end if;

```

```

end if;

if v_deptId like '04%' then
    select Trunc(MONTHS_BETWEEN(regDate, v_birth) / 12)
        into v_age
        from dual;
    if v_age <= 14 then
        v_return := '2';
    end if;
end if;

```

```

return v_return;
end;

```

-- 5.1.8 查询待缴费医保处方信息

```

procedure getWaitPayMedInsInfo001(hospitalId in varchar2, -- 医院 ID
    visitId in varchar2, -- 就诊流水号
    aCur_temp out sys_refcursor) is
    /* 输出结果
    * 字段      描述
    * -----
    deptName  科室名称
    */

```

```

caty      科别
bkf004    执行科室编号
bkf005    执行科室名称
doctored  门诊诊断医生
diagType  诊断类别
diagSrtNo 诊断排序号
diagCode  诊断代码
diagName  诊断名称
diagdept  诊断科室
diagDorNo 诊断医生编号
diagDorName  诊断医生姓名
diagTime  诊断时间

```

```
*/
```

```
begin
```

```
open aCur_temp for
```

```

SELECT substr(t.dept_name,1,10) as "deptName", -- 科室名称
       substr(t.caty,1,10)      as "caty", -- 科别
       t.bkf004      as "bkf004", --执行科室编号
       substr(t.bkf005,1,10)    as "bkf005", -- 执行科室名称
       t.doctorid   as "doctorId", -- 门诊诊断医生
       t.diag_type  as "diagType", -- 诊断类别
       t.diag_srt_no as "diagSrtNo", -- 诊断排序号
       t.diag_code  as "diagCode", -- 诊断代码
       substr(t.diag_dept,1,10)  as "diagdept", -- 诊断科室
       t.diag_name  as "diagName", -- 诊断名称

```

```

t.dise_dor_no as "diagDorNo", -- 诊断医生编号
t.dise_dor_name as "diagDorName", -- 诊断医生姓名
t.diag_time as "diagTime" -- 诊断时间

FROM Outpdocr.outp_diagnosis_clinic_no t
WHERE t.clinic_no = visitId and rownum=1
order by diag_srt_no;
end;

```

--新增测试

```

procedure PRO_OUTPBILL_TJ_INSERT(v_xml_string in varchar2,
                                aCur_temp out sys_refcursor,
                                on_return OUT number,
                                on_message OUT varchar2) as

```

```

v_rcptno varchar2(20);

```

```

begin

```

```

begin

```



```

outpbill.pro_outpbill_tj_insert(v_xml_string,
                                on_return,
                                on_message,
                                v_rcptno);

exception
when others then

    on_return := 1;
    on_message := '调用存储过程出错' || SQLERRM;
    goto end_program;
end;

<<end_program>>

if on_return = 0 then

    open aCur_temp for
        select to_char(on_return) as "resultCode",
               on_message as "resultDesc",
               v_rcptno as "rcpt_no"
        from dual;
else

    open aCur_temp for
        select to_char(on_return) as "resultCode",
               on_message as "resultDesc"
        from dual;
end if;

```

end;

procedure PRO_OUTPBILL_INSUR_upload(v_xml_string in clob,

 aCur_temp out sys_refcursor,

 on_return OUT number,

 on_message OUT varchar2) as

begin

begin

 outpbill.pro_outpbill_insur_detail(v_xml_string,

 acur_temp,

 on_return,

 on_message);

exception

 when others then

 on_return := 1;

 on_message := '调用存储过程出错' || SQLERRM;

end;

end;

--体检退费，可用于核酸检测预约退费

procedure pro_outpbill_Tj_Ref(v_xml_string in varchar2,

 aCur_temp out sys_refcursor,

 on_return OUT number,

```

        on_message OUT varchar2) as

v_rcptno varchar2(20);

begin

insert into outpbill.outpbill_refund_insur_log
    (app_name,
      operator_id,
      operator_user_name,
      operator_log,
      operator_time,
      memo,
      XML_BODY)
values
    ('体检退费',
      '1968',
      '小程序计费',
      '99',
      sysdate,
      'pro_outpbill_Tj_Ref',
      '999');

commit;

begin

    outpbill.pro_outpbill_Tj_Refund(v_xml_string,
        on_return, -- 结果编码
        on_message -- 结果描述

```

```

        );

exception
when others then

    on_return := 1;
    on_message := '调用存储过程出错' || SQLERRM;
    goto end_program;
end;
<<end_program>>

if on_return = 0 then

    open aCur_temp for
        select to_char(on_return) as "resultCode",
               on_message as "resultDesc",
               " as "rcpt_no"
        from dual;
else

    open aCur_temp for
        select to_char(on_return) as "resultCode",
               on_message as "resultDesc"
        from dual;
end if;

end;

--新增测试 --体检退费，可用于核酸检测预约退费 缴费接口后面加个 1

```

```

procedure PRO_OUTPBILL_TJ_INSERT1(v_xml_string in varchar2,
                                aCur_temp out sys_refcursor,
                                on_return OUT number,
                                on_message OUT varchar2) as

begin

    on_return := 0;
    on_message := '成功';
    begin

        outpbill.pro_outpbill_Tj_Refund(v_xml_string,
                                       on_return, -- 结果编码
                                       on_message -- 结果描述
                                       );

    exception

        when others then

            on_return := 1;
            on_message := '调用存储过程出错' || SQLERRM;
            goto end_program;
    end;

    <<end_program>>

    if on_return = 0 then

        insert into outpbill.outpbill_refund_insur_log

```

```

(app_name,
operator_id,
operator_user_name,
operator_log,
operator_time,
memo,
XML_BODY)
values
('体检退费',
'1968',
'小程序计费',
'99',
sysdate,
'pro_outpbill_Tj_Ref 返回到包里面的' || on_return || on_message,
v_xml_string);

commit;
end if;

open aCur_temp for
select to_char(on_return) as "excCode", on_message as "excDesc"
from dual;
/*else

open aCur_temp for
select to_char(on_return) as "resultCode",
on_message as "resultDesc"
from dual;
end if;*/

```

```
end;
```

```
end ps_appreg_wx;
```