

CREATE OR REPLACE Package Body S_Ini_TableStruc Is

```
CREATE OR REPLACE Package Body S_Ini_TableStruc Is
```

```
v_CEt1      Constant Varchar2(200) := fun_get_etluser;
v_BTime     Constant Varchar2(40)  := '201001';
v_SWBUG     Constant Char(1)       := S_Ini_TableStruc.Fun_GetSwBug;
v_Nullable  Constant Char(1)       := '1'; --如果改变量='1'则所有业务字段（不包含主键）不允许为空设置在建表时不生效;
v_KeyDbint  Constant Char(1)       := '0'; --如果改变量='1' DB_INT对应接口表在创建时,会根据设置自动创建主键;
```

```
Function Fun_GetSwBug Return Varchar2 Is
```

```
/*获取sys_config_m中的调试开关信息*/
```

```
    vF_SWBUG  Char(1);
    Script Varchar2(4096);
```

```
Begin
```

```
    Script := 'Select SW_BUG From '||v_CEt1||'.SYS_CONFIG_M';
    Execute Immediate Script Into vF_SWBUG;
    Return nvl(vF_SWBUG, '0');
```

```
End Fun_GetSwBug;
```

```
Function Fun_GetDbUser (DbType Varchar2) Return Varchar2 Is
```

```
    vF_DbUser Varchar2(40);
    Script Varchar2(4096);
```

```
Begin
```

```
    vF_DbUser := 'N/A';
    If Upper(DbType)='DB_INT' Then
        Script := 'Select Upper(DB_INT) From '||v_CEt1||'.SYS_CONFIG_M';
    End If;
    If Upper(DbType)='DB_ORI' Then
        Script := 'Select Upper(DB_ORI) From '||v_CEt1||'.SYS_CONFIG_M';
    End If;
    If Upper(DbType)='DB_BUF' Then
        Script := 'Select Upper(DB_BUF) From '||v_CEt1||'.SYS_CONFIG_M';
    End If;
    If Upper(DbType)='DB_COR' Then
        Script := 'Select Upper(DB_COR) From '||v_CEt1||'.SYS_CONFIG_M';
    End If;
    If Upper(DbType)='DB_HIS' Then
        Script := 'Select Upper(DB_HIS) From '||v_CEt1||'.SYS_CONFIG_M';
    End If;
    If Upper(DbType)='DB_NOR' Then
        Script := 'Select Upper(DB_NOR) From '||v_CEt1||'.SYS_CONFIG_M';
    End If;
    If Upper(DbType)='DB_BAK' Then
        Script := 'Select Upper(DB_BAK) From '||v_CEt1||'.SYS_CONFIG_M';
    End If;
    Execute Immediate Script Into vF_DbUser;
    Return vF_DbUser;
End Fun_GetDbUser;
```

```
Function Fun_GetCharList (vString In Varchar2, vSymbol In Varchar2) Return S_Ini_TableStruc.Str_Split_Tab Pipelined As
```

```
    Counter  Number := 1;
    vStr_Split_Row  S_Ini_TableStruc.Str_Split_Row;
    vStr_Split_Tab  S_Ini_TableStruc.Str_Split_Tab := Str_Split_Tab();
    v_Length Number := Length(vString);
    v_Start  Number := 1;
    v_Index  Number;
```

```
Begin
```

```

While(v_Start <= v_Length)
Loop
    v_Index := Instr(vString, vSymbol, v_Start);
    If v_Index = 0 Then
        vStr_Split_Row.Col_Value :=Substr(vString, v_Start);
        v_Start := v_Length + 1;
    Else
        vStr_Split_Row.Col_Value :=Substr(vString, v_Start, v_Index - v_Start);
        v_Start := v_Index + 1;
    End If;
    vStr_Split_Tab.EXTEND;
    vStr_Split_Tab(Counter) := vStr_Split_Row;
    Pipe Row(vStr_Split_Tab(Counter));
    Counter := Counter + 1;
End Loop;
Return;
End Fun_GetCharList;

Function Fun_GetPrimaryKey(vTalbe Varchar2, vLinkStr Varchar2 Default ',', vType Int Default 0) Return Varchar2 Is
/*联接表中的主键字段名称*/
Cursor mycursor Is Select COL_CODE As Col_Name, DATA_TYPE From SYS_TABLE_LIST A, SYS_COLUMN_LIST B
    Where A.TAB_CODE = B.TAB_CODE And A.TAB_CODE=Upper(vTalbe) And A.TAB_USED = 1 And B.PK_SEQ>0
    Order By B.PK_SEQ;

vOutStr Varchar2(4000);
colName Varchar2(100);
colType Varchar2(100);
Begin
    Open mycursor;
    Loop Fetch mycursor Into colName, colType;
        Exit When mycursor%Notfound;
        vOutStr := Case When vType=1 Then vOutStr || '||''' ||Nvl(vLinkStr,',') ||'''||'
            Case When Upper(colType)=' DATE' Then 'To_Char(' ||colName||',''YYYYMMDD'' )'
                When Instr(Upper(colType),'CHAR')=0 Then 'To_Char(' ||colName||')'
            --Else colName End /*由于主键过长导致摘要信息超长影响系统运行，特此进行修改2014-06-05 Syc */
            Else 'substr(' ||colName||',1,32)' End
            When vType=2 Then vOutStr ||Nvl(vLinkStr,',') || '@T1@.' ||colName|| '=@T2@.' ||colName
        Else vOutStr ||Nvl(vLinkStr,',') || colName End;
    End loop;
    vOutStr := Case When vType=1 Then Ltrim(vOutStr, '||''' ||Nvl(vLinkStr,',') ||'''||')
        When vType=2 Then Ltrim(vOutStr, Nvl(vLinkStr,','))
        Else Ltrim(vOutStr, Nvl(vLinkStr,',')) End;

    Close mycursor;
    Return nvl(vOutStr, 'N/A');
End Fun_GetPrimaryKey;

Function Fun_GetColumn(vTalbe Varchar2) Return Varchar2 Is
/*联接表中的字段名称*/
Cursor mycursor Is Select COL_CODE As Col_Name From SYS_TABLE_LIST A, SYS_COLUMN_LIST B
    Where A.TAB_CODE = B.TAB_CODE And A.TAB_CODE=Upper(vTalbe) And A.TAB_USED = 1
    Order By B.COL_ID;

vOutStr Varchar2(4000);
colName varchar(100);
Begin
    Open mycursor;
    Loop Fetch mycursor Into colName;
        Exit When mycursor%Notfound;
        vOutStr := vOutStr || ', ' || colName;

```

```

End loop;
vOutStr := Ltrim(vOutStr , ',');

Close mycursor;

Return vOutStr;
End Fun_GetColumn;

```

Function Fun_OracleColType Return S_Ini_TableStruc.Str_Split_Tab Pipelined As

```

Counter Number := 1;
vString Varchar2(800) := ' DATE, NUMBER, CHAR, VARCHAR, VARCHAR2, NVARCHAR2' ;
vSymbol Varchar2(10) := ',';
vStr_Split_Row S_Ini_TableStruc.Str_Split_Row;
vStr_Split_Tab S_Ini_TableStruc.Str_Split_Tab := Str_Split_Tab();
v_Length Number := Length(vString);
v_Start Number := 1;
v_Index Number;
Begin
While(v_Start <= v_Length)
Loop
v_Index := Instr(vString, vSymbol, v_Start);
If v_Index = 0 Then
vStr_Split_Row.Col_Value :=Substr(vString, v_Start);
v_Start := v_Length + 1;
Else
vStr_Split_Row.Col_Value :=Substr(vString, v_Start, v_Index - v_Start);
v_Start := v_Index + 1;
End If;
vStr_Split_Tab.EXTEND;
vStr_Split_Tab(Counter) := vStr_Split_Row;
Pipe Row(vStr_Split_Tab(Counter));
Counter := Counter + 1;
End Loop;
Return;
End Fun_OracleColType;

```

Function Fun_GetScript_DrpTable (DbType varchar2, vTab_Code varchar2) Return Varchar2 Is

```

vString Varchar2(8000);
Lownum Number;
vOwner Varchar2(200);
Begin
vOwner := S_Ini_TableStruc.Fun_GetDbUser(DbType);
Select Count(1) Into Lownum From All_Tables Where Owner = vOwner And Table_Name = vTab_Code;
If vTab_Code Is Not Null And vOwner<>'N/A' And Lownum>0 Then
vString := vString||Case When vOwner Is Not Null Then vOwner||'.' Else '' End||vTab_Code;
vString := 'Drop Table ' ||vString||' Cascade Constraints';
End If;
Return nvl(vString,'--表'||vOwner||'.' ||vTab_Code||' 不存在');
End Fun_GetScript_DrpTable;

```

Procedure Prc_Drp_Table (Resulst Out Integer, DbType varchar2 Default '', vTableCode varchar2 Default '') Is

```

Type EmpCurTyp Is Ref Cursor;
Usecur EmpCurTyp;
Tabcur EmpCurTyp;
vString Varchar2(8000);
vDbType Varchar2(200);
vTab_Code Varchar2(200);
Lownum Number;

```

```

vPrc_Name      Varchar2(100);
Begin
vPrc_Name:='Prc_Drp_Table';
Resulst:= 0;

--DbType变量合理性扫描
If DbType Is Not Null Then
    vDbType := DbType;
    Select Count(1) Into Lownum From V_S_DUSER_LIST B Where B.DbType=vDbType;
    If Lownum=0 Then
        Commit;
        Dbms_Output.Put_Line('Error: DbType参数输入值异常，字段注释过程放弃执行。');
        Return;
    End If;
End If;

--接口表启用标志位合理性扫描
If vTableCode Is Null Then
    Select Count(1) Into Lownum From V_S_TABLE_LIST;
Else
    Select Count(1) Into Lownum From V_S_TABLE_LIST Where Tab_Code=vTableCode;
End If;
If Lownum=0 Then
    Commit;
    Dbms_Output.Put_Line('Error: V_S_TABLE_LIST中不存在已生效的接口表配置信息，字段注释过程放弃执行。');
    Return;
End If;

If vTableCode Is Null Then
    vString := 'Select A.Tab_Code From V_S_TABLE_LIST A Order By A.Tab_Note';
Else
    vString := 'Select '''||vTableCode||''' as Tab_Code From dual';
End If;
Open Tabcur For vString;
Loop Fetch Tabcur Into vTab_Code;
    Exit When Tabcur%Notfound;
    If DbType Is Null Then
        vString := 'Select B.DbType From V_S_DUSER_LIST B Order By B.DbType';
    Else
        vString := 'Select '''||DbType||''' as DbType From dual';
    End If;
    Open Usecur For vString;
    Loop Fetch Usecur Into vDbType;
        Exit When Usecur%Notfound;
        vString := S_Ini_TableStruc.Fun_GetScript_DrpTable(vDbType, vTab_Code);
        If vSWBUG='1' Then
            dbms_output.put_line(vString||';');
        End If;
        If Instr(vString,'不存在')=0 Then
            Execute Immediate vString;
        End If;
        Commit;
    End loop;
    --Update Sys_Table_List Set Tab_Used = 0 Where Tab_Code = vTab_Code;
    Commit;
End loop;
Resulst:= 1;

```

--异常处理

Exception

When Others Then

DBMS_OUTPUT.put_line(' ERROR SQLCODE: ' || SQLCODE);

DBMS_OUTPUT.put_line(' ERROR MSG: ' || SQLERRM);

Rollback;

RAISE_APPLICATION_ERROR(SQLCODE, SQLERRM);

Return;

End Prc_Drp_Table;

Function Fun_GetScript_CrtCol (vCol_Code varchar2, vData_Type varchar2, vCol_Default varchar2, vNullable number,
vData_Length number, vData_Scale number) Return Varchar2 Is

vString Varchar2(8000);

Begin

If Upper(vData_Type)=' DATE' Then

vString := vCol_Code || ' DATE'

|| Case When vCol_Default Is Null Then '' Else ' default ' || vCol_Default End

|| Case When vNullable=0 Then ' Not Null' Else '' End;

End If;

If Upper(vData_Type)=' NUMBER' And NVL(vData_Length,0)>0 And NVL(vData_Length,0)>NVL(vData_Scale,0) Then

If NVL(vData_Length,0)=22 And NVL(vData_Scale,0)=0 Then

vString := vCol_Code || ' INTEGER'

|| Case When vCol_Default Is Null Then '' Else ' default ' || vCol_Default End

|| Case When vNullable=0 Then ' Not Null' Else '' End;

Else

vString := vCol_Code || ' NUMBER(' || NVL(vData_Length,0) || Case When NVL(vData_Scale,0)>0 Then ',' || NVL(vData_Scale,0) Else

'' End || ')'

|| Case When vCol_Default Is Null Then '' Else ' default ' || vCol_Default End

|| Case When vNullable=0 Then ' Not Null' Else '' End;

End If;

End If;

If Upper(vData_Type)=' CHAR' And NVL(vData_Length,0)>0 Then

vString := vCol_Code || ' CHAR(' || NVL(vData_Length,1) || ')'

|| Case When vCol_Default Is Null Then '' Else ' default ' || vCol_Default End

|| Case When vNullable=0 Then ' Not Null' Else '' End;

End If;

If Upper(vData_Type)=' VARCHAR' And NVL(vData_Length,0)>0 Then

vString := vCol_Code || ' VARCHAR(' || NVL(vData_Length,1) || ')'

|| Case When vCol_Default Is Null Then '' Else ' default ' || vCol_Default End

|| Case When vNullable=0 Then ' Not Null' Else '' End;

End If;

If Upper(vData_Type)=' VARCHAR2' And NVL(vData_Length,0)>0 Then

vString := vCol_Code || ' VARCHAR2(' || NVL(vData_Length,1) || ')'

|| Case When vCol_Default Is Null Then '' Else ' default ' || vCol_Default End

|| Case When vNullable=0 Then ' Not Null' Else '' End;

End If;

If Upper(vData_Type)=' NVARCHAR2' And NVL(vData_Length,0)>0 Then

vString := vCol_Code || ' NVARCHAR2(' || NVL(vData_Length,1) || ')'

|| Case When vCol_Default Is Null Then '' Else ' default ' || vCol_Default End

|| Case When vNullable=0 Then ' Not Null' Else '' End;

End If;

Return nvl(vString, '--Col_Code[' || vCol_Code || '] 字段配置信息异常');

End Fun_GetScript_CrtCol;

Function Fun_GetScript_CrtPat (DbType varchar2, vTab_Code varchar2) Return Varchar2 Is

vString Varchar2(8000);

Lownum Number;

Begin

```

vString := '';
Select Count(1) Into Lownum From SYS_PART_LIST Where Upper(DB_TYPE)=Upper(DbType) And Upper(TAB_CODE)=Upper(vTab_Code);
If Lownum>0 Then
    Select Case When PAT_TYPE='1' Then 'partition by range('||PAT_COLU||') (partition POther values less than (MAXVALUE))'
        When PAT_TYPE='2' Then 'partition by hash('||PAT_COLU||') (partition POther)'
        When PAT_TYPE='3' Then 'partition by list('||PAT_COLU||') (partition POther values (DEFAULT))'
        Else '' End Into vString
    From SYS_PART_LIST Where Upper(DB_TYPE)=Upper(DbType) And Upper(TAB_CODE)=Upper(vTab_Code);
End If;
Return vString;
End Fun_GetScript_CrtPat;

```

```

Function Fun_GetScript_CrtKey (DbType varchar2, vTab_Code varchar2) Return Varchar2 Is
Type EmpCurTyp Is Ref Cursor;
mycursor      EmpCurTyp;
vString        Varchar2(8000);
vTab_Sno        Varchar2(100);
vCol_Code       Varchar2(200);
vPrimaryKey     Varchar2(400);
Lownum          Number;
vOwner          Varchar2(200);
Begin
vOwner := S_Ini_TableStruc.Fun_GetDbUser(DbType);
Select Count(1) Into Lownum From SYS_COLUMN_LIST Where PK_SEQ>0 And Upper(TAB_CODE)=Upper(vTab_Code);
If Lownum>0 And vOwner Is Not Null Then
    Select NVL(TAB_SNO,TAB_CODE) Into vTab_Sno From SYS_TABLE_LIST Where TAB_CODE=vTab_Code;

    Open mycursor For Select COL_CODE From SYS_COLUMN_LIST Where PK_SEQ>0 And TAB_CODE=vTab_Code Order By PK_SEQ, COL_ID;
    Loop Fetch mycursor Into vCol_Code;
        Exit When mycursor%Notfound;
        vPrimaryKey := Case When vPrimaryKey Is Not Null Then vPrimaryKey||',' Else '' End || vCol_Code;
    End loop;
    Close mycursor;
    vString := 'alter table '||vOwner||'.'||vTab_Code||' add constraint PK_'||vTab_Sno||' primary key ('||vPrimaryKey||')';
End If;
Return nvl(vString,'--Tab_Code['||vTab_Code||'] 主键配置信息异常');
End Fun_GetScript_CrtKey;

```

```

Function Fun_GetScript_CrtTable (DbType varchar2, vTab_Code varchar2) Return Varchar2 Is
Type EmpCurTyp Is Ref Cursor;
tabcolcur      EmpCurTyp;
syscolcur      EmpCurTyp;
vString        Varchar2(8000);
vInicol        Varchar2(8000);
vCol_Sql       Varchar2(400);
vCol_Code       Varchar2(200);
vData_Type     Varchar2(200);
vCol_Default   Varchar2(200);
vData_Length   Number(4);
vData_Scale    Number(4);
vNullable      Number(2);
vOwner          Varchar2(200);
Begin
vOwner := S_Ini_TableStruc.Fun_GetDbUser(DbType);
vString := 'create table '||vOwner||'.'||vTab_Code||' (';
Open tabcolcur For
Select Col_Code, Data_Type, Data_Length, Data_Scale, Col_Default,
    Case When Pk_Seq>0 Then 0

```

```

        When B.Col_Date Is Not Null Then 0
        When Upper(Data_Type)='DATE' Then Nullable
        When v_Nullable='1' Then 1 Else Nullable End As Nullable
From Sys_Column_List A Left Outer Join Sys_Table_List B On A.Tab_Code=B.Tab_Code And A.Col_Code=B.Col_Date
Where Upper(A.Tab_Code)=Upper(vTab_Code) Order By Col_Id;
Loop Fetch tabcolcur Into vCol_Code, vData_Type, vData_Length, vData_Scale, vCol_Default, vNullable;
    Exit When tabcolcur%Notfound;
    vCol_Sql := S_Ini_TableStruc.Fun_GetScript_CrtCol(vCol_Code, vData_Type, vCol_Default, vNullable, vData_Length,
vData_Scale);
    vString := vString||Chr(10)||vCol_Sql||',';
End loop;

vInicol := 'Select Col_Code, Data_Type, Data_Length, Data_Scale, Col_Default, Nullable From Sys_Config_Inicol Where
' || DbType || '= ' || '1' Order By Pkid';
Open syscolcur For vInicol;
Loop Fetch syscolcur Into vCol_Code, vData_Type, vData_Length, vData_Scale, vCol_Default, vNullable;
    Exit When syscolcur%Notfound;
    vCol_Sql := S_Ini_TableStruc.Fun_GetScript_CrtCol(vCol_Code, vData_Type, vCol_Default, vNullable, vData_Length,
vData_Scale);
    vString := vString||Chr(10)||vCol_Sql||',';
End loop;
vString := RTRIM(vString, ',') || ') ' ||
        Case When S_Ini_TableStruc.Fun_GetScript_CrtPat(DbType, vTab_Code) Is Null Then ''
        Else Chr(10)||S_Ini_TableStruc.Fun_GetScript_CrtPat(DbType, vTab_Code) End;
Return vString;
End Fun_GetScript_CrtTable;

```

```

Function Fun_GetScript_TabComment(DbType varchar2, vTab_Code varchar2) Return Varchar2 Is
vString      Varchar2(200);
Lownum       Number;
vOwner       Varchar2(200);
Begin
vOwner := S_Ini_TableStruc.Fun_GetDbUser(DbType);
Select Count(1) Into Lownum From SYS_COLUMN_LIST Where Upper(TAB_CODE)=Upper(vTab_Code);
If Lownum>0 And vOwner Is Not Null Then
    Select 'comment on table ' || vOwner || '.' || vTab_Code || ' is ''' || TAB_NOTE || '''' Into vString
    From SYS_TABLE_LIST Where Upper(TAB_CODE)=Upper(vTab_Code);
End If;
Return nvl(vString, '--Tab_Code[' || vTab_Code || '] 注释配置信息异常');
End Fun_GetScript_TabComment;

```

```

Function Fun_GetScript_GraTable (DbType varchar2, vTab_Code varchar2) Return Varchar2 Is
vString      Varchar2(200);
vEtIUser     Varchar2(200);
vOwner       Varchar2(200);
Begin
Select User Into vEtIUser From Dual;
vOwner := S_Ini_TableStruc.Fun_GetDbUser(DbType);
vString := 'Grant Select, Insert, Delete, Update On ' || vOwner || '.' || vTab_Code || ' To ' || Upper(vEtIUser);
Return (vString);
End Fun_GetScript_GraTable;

```

```

Procedure Prc_Crt_Table (Resulst Out Integer, DbType varchar2 Default '', vTableCode varchar2 Default '') Is
Type EmpCurTyp Is Ref Cursor;
Usecur        EmpCurTyp;
Tabcur        EmpCurTyp;
vString       Varchar2(8000);
vDbType       Varchar2(200);

```

```

vTab_Code      Varchar2(200);
Lownum         Number;
vPrc_Name      Varchar2(100);
Begin
vPrc_Name:='Prc_Crt_Table';
Resultst:= 0;

--DbType变量合理性扫描
If DbType Is Not Null Then
    vDbType := DbType;
    Select Count(1) Into Lownum  From V_S_DUSER_LIST B Where B.DbType=vDbType;
    If Lownum=0 Then
        Commit;
        Dbms_Output.Put_Line(' Error: DbType参数输入值异常，字段注释过程放弃执行。');
        Return;
    End If;
End If;

--数据类型匹配程度扫描
Select Count(1) Into Lownum  From Sys_Column_List Where Data_Type Not In (Select COL_VALUE From
Table(s_Ini_Tablestruc.Fun_Oraclecoltype));
If Lownum>0 Then
    Commit;
    Dbms_Output.Put_Line(' Error: Sys_Column_List中Data_Type字段中的数据类型超出了ORACLE数据库的允许范围，建表过程放弃执行。');
    Return;
End If;

--接口表启用标志位合理性扫描
If vTableCode Is Null Then
    Select Count(1) Into Lownum  From V_S_TABLE_LIST;
Else
    Select Count(1) Into Lownum  From V_S_TABLE_LIST Where Tab_Code=vTableCode;
End If;
If Lownum=0 Then
    Commit;
    Dbms_Output.Put_Line(' Error: V_S_TABLE_LIST中不存在已生效的接口表配置信息，建表过程放弃执行。');
    Return;
End If;

--接口表字段配置是否存在扫描
Select Count(1) Into Lownum From V_S_TABLE_LIST A
    Left Outer Join (Select Tab_Code, Count(1) As Col_Cunt From Sys_Column_List Group By Tab_Code) B On A.Tab_Code=B.Tab_Code
Where Nvl(B.Col_Cunt,0)=0;
If Lownum>0 Then
    Commit;
    Dbms_Output.Put_Line(' Error: Sys_Table_List已生效的接口表配置信息，在Sys_Column_List中缺乏相应的字段配置信息，建表过程放弃
执行。');
    Return;
End If;

--接口表主键配置是否存在扫描
Select Count(1) Into Lownum From V_S_TABLE_LIST A
    Left Outer Join (Select Tab_Code, Count(1) As Col_Cunt From Sys_Column_List Where Pk_Seq>0 Group By Tab_Code) B On
A.Tab_Code=B.Tab_Code
Where B.Tab_Code Is Null;
If Lownum>0 Then
    Commit;
    Dbms_Output.Put_Line(' Error: Sys_Table_List未生效的接口表配置信息，在Sys_Column_List中缺乏相应的主键字段配置信息，建表过程

```



```

放弃执行。');
    Return;
End If;

If vTableCode Is Null Then
    vString := 'Select A.Tab_Code From V_S_TABLE_LIST A Order By A.Tab_Note';
Else
    vString := 'Select '''||vTableCode||''' as Tab_Code From dual';
End If;
Open Tabcur For vString;
Loop Fetch Tabcur Into vTab_Code;
    Exit When Tabcur%Notfound;
    If DbType Is Null Then
        vString := 'Select B.DbType From V_S_DUSER_LIST B Order By B.DbType';
    Else
        vString := 'Select '''||DbType||''' as DbType From dual';
    End If;
    Open Usecur For vString;
    Loop Fetch Usecur Into vDbType;
        Exit When Usecur%Notfound;
        --表结构创建
        vString := S_Ini_TableStruc.Fun_GetScript_CrtTable(vDbType, vTab_Code);
        If vSWBUG='1' Then
            dbms_output.put_line(vString||';');
        End If;
        Execute Immediate vString;
        If vDbType='DB_BUF' Or vDbType='DB_COR' Or vDbType='DB_HIS' Or vDbType='DB_NOR' Or (vDbType='DB_INT' And v_KeyDbint='1')
Then
            --表主键创建
            vString := S_Ini_TableStruc.Fun_GetScript_CrtKey (vDbType, vTab_Code);
            If vSWBUG='1' Then
                dbms_output.put_line(vString||';');
            End If;
            Execute Immediate vString;
        End If;
        --表注释创建
        vString := S_Ini_TableStruc.Fun_GetScript_TabComment(vDbType, vTab_Code);
        If vSWBUG='1' Then
            dbms_output.put_line(vString||';');
        End If;
        Execute Immediate vString;
        --表授权创建
        vString := S_Ini_TableStruc.Fun_GetScript_GraTable(vDbType, vTab_Code);
        dbms_output.put_line(vString||';');
        --Execute Immediate vString;
        Commit;
    End loop;
    --Update Sys_Table_List Set Tab_Used = 1 Where Tab_Code = vTab_Code;
    Commit;
End loop;
Result:= 1;

--异常处理
Exception
When Others Then
    DBMS_OUTPUT.put_line('ERROR SQLCODE: '||SQLCODE);
    DBMS_OUTPUT.put_line('ERROR MSG: '||SQLERRM);
    Rollback;

```

```

        RAISE_APPLICATION_ERROR(SQLCODE, SQLERRM);
    Return;
End Prc_Crt_Table;

Function Fun_GetScript_ColComment(DbType varchar2, vTab_Code varchar2, vCol_Code varchar2) Return Varchar2 Is
    vString      Varchar2(600);
    Lownum       Number;
    vOwner       Varchar2(200);
Begin
    vOwner := S_Ini_TableStruc.Fun_GetDbUser(DbType);
    Select Count(1) Into Lownum From SYS_COLUMN_LIST Where Upper(TAB_CODE)=Upper(vTab_Code) And Upper(COL_CODE)=Upper(vCol_Code);
    If Lownum>0 And vOwner Is Not Null Then
        Select 'comment on column '||vOwner||'.'||vTab_Code||'.'||vCol_Code||' is '' '||Replace(COL_NOTE, ''', ''')||'''' Into
vString
        From SYS_COLUMN_LIST Where Upper(TAB_CODE)=Upper(vTab_Code) And Upper(COL_CODE)=Upper(vCol_Code);
    End If;
    Return nvl(vString, '--Tab_Code['||vTab_Code||'] Col_Code['||vCol_Code||'] 注释配置信息缺失');
End Fun_GetScript_ColComment;

Procedure Prc_Alt_ColComment (Result Out Integer, DbType varchar2 Default '', vTableCode varchar2 Default '') Is
    Type EmpCurTyp Is Ref Cursor;
    Usecur      EmpCurTyp;
    Tabcur      EmpCurTyp;
    vString     Varchar2(8000);
    vDbType     Varchar2(200);
    vTab_Code   Varchar2(200);
    Lownum      Number;
    vCol_Code   Varchar2(200);
    vPrc_Name   Varchar2(100);
Begin
    vPrc_Name:='Prc_Alt_ColComment';
    Result:= 0;

    --DbType变量合理性扫描
    If DbType Is Not Null Then
        vDbType := DbType;
        Select Count(1) Into Lownum From V_S_DUSER_LIST B Where B.DbType=vDbType;
        If Lownum=0 Then
            Commit;
            Dbms_Output.Put_Line('Error: DbType参数输入值异常，字段注释过程放弃执行。');
            Return;
        End If;
    End If;

    --接口表启用标志位合理性扫描
    If vTableCode Is Null Then
        Select Count(1) Into Lownum From V_S_TABLE_LIST;
    Else
        Select Count(1) Into Lownum From V_S_TABLE_LIST Where Tab_Code=vTableCode;
    End If;
    If Lownum=0 Then
        Commit;
        Dbms_Output.Put_Line('Error: V_S_TABLE_LIST中不存在已生效的接口表配置信息，字段注释过程放弃执行。');
        Return;
    End If;

    --接口表字段配置是否存在扫描
    Select Count(1) Into Lownum From V_S_TABLE_LIST A

```

```

        Left Outer Join (Select Tab_Code, Count(1) As Col_Cunt From Sys_Column_List Group By Tab_Code) B On A.Tab_Code=B.Tab_Code
Where Nvl(B.Col_Cunt,0)=0;
If Lownum>0 Then
    Commit;
    Dbms_Output.Put_Line(' Error: Sys_Table_List已生效的接口表配置信息，在Sys_Column_List中缺乏相应的字段配置信息，建表过程放弃
执行。');
    Return;
End If;

vString := 'Select Tab_Code, Col_Code From Sys_Column_List Where Col_Note Is Not Null '||
        Case When vTableCode Is Null Then ' And Tab_Code In (Select Tab_Code From V_S_TABLE_LIST)' Else ' And
Tab_Code='''||vTableCode|'''' ' End||
        'Order By Tab_Code, Col_Id';
Open Tabcur For vString;
Loop Fetch Tabcur Into vTab_Code, vCol_Code;
Exit When Tabcur%Notfound;
If DbType Is Null Then
    vString := 'Select B.DbType From V_S_DUSER_LIST B Order By B.DbType';
Else
    vString := 'Select '''||DbType|'''' as DbType From dual';
End If;
Open Usecur For vString;
Loop Fetch Usecur Into vDbType;
Exit When Usecur%Notfound;
--字段注释信息创建
vString := S_Ini_TableStruc.Fun_GetScript_ColComment(vDbType, vTab_Code, vCol_Code);
If vSWBUG='1' Then
    dbms_output.put_line(vString||';');
End If;
Execute Immediate vString;
Commit;
End loop;
Commit;
End loop;
Result:= 1;

--异常处理
Exception
When Others Then
    DBMS_OUTPUT.put_line(' ERROR SQLCODE: '||SQLCODE);
    DBMS_OUTPUT.put_line(' ERROR MSG: '||SQLERRM);
    Rollback;
    RAISE_APPLICATION_ERROR(SQLCODE,SQLERRM);
Return;
End Prc_Alt_ColComment;

Procedure Prc_Crt_SysIndex (Result Out Integer, vTableCode varchar2 Default '') Is
Type EmpCurTyp Is Ref Cursor;
Usecur      EmpCurTyp;
Tabcur      EmpCurTyp;
vString      Varchar2(8000);
vDbType      Varchar2(200);
vDbuser      Varchar(200);
vTab_Code    Varchar(200);
vTab_Sno     Varchar(200);
vKey_Col     Varchar(200);
vSys_IxName   Varchar(200);
vPrc_Name     Varchar2(100);

```

```

        Lownum          Number;
Begin
    vPrc_Name:='Prc_Crt_SysIndex';
    Resultst:= 0;

    vString := 'Select A.Tab_Code, NVL(TAB_SNO,TAB_CODE) As Tab_Sno From V_S_TABLE_LIST A ' ||
        Case When vTableCode Is Null Then '' Else 'Where Tab_Code=''' ||vTableCode||'''' End||
        ' Order By A.Tab_Note';
    Open Tabcur For vString;
    Loop Fetch Tabcur Into vTab_Code, vTab_Sno;
        Exit When Tabcur%Notfound;
        vString := 'Select DbType, Dbuser From V_S_DUSER_LIST Where DbType<>''DB_BUF'' And DbType<>''DB_COR'' And
        DbType<>''DB_HIS'' And DbType<>''DB_NOR'''' ||
            Case When v_KeyDbint='1' Then ' And DbType<>''DB_INT'''' Else '' End;
        Open Usecur For vString;
        Loop Fetch Usecur Into vDbType, vDbuser;
            Exit When Usecur%Notfound;
            vKey_Col := S_Ini_TableStruc.Fun_GetPrimaryKey(vTab_Code);
            If vKey_Col<>'N/A' Then
                vSys_IxName:='IX_S_' ||vTab_Sno;
                Select Count(1) Into Lownum From All_Indexes Where Table_Name=vTab_Code And Table_Owner=vDbuser And
                Index_Name=vSys_IxName;
                If (Lownum = 0) Then
                    Select Count(1) Into Lownum From SYS_PART_LIST Where (PAT_TYPE='1' Or PAT_TYPE='3') And DB_TYPE=vDbType And
                TAB_CODE=vTab_Code;
                    If (Lownum = 0) Then
                        vString := 'CREATE INDEX ' ||vDbuser||'.' ||vSys_IxName||' ON ' || vDbuser||'.' ||vTab_Code||' (' ||vKey_Col||')';
                    Else
                        vString := 'CREATE INDEX ' ||vDbuser||'.' ||vSys_IxName||' ON ' || vDbuser||'.' ||vTab_Code||' (' ||vKey_Col||') Local';
                    End If;
                    Select Count(1) Into Lownum From All_Tables Where OWNER=vDbuser And TABLE_NAME=vTab_Code;
                    If (Lownum = 1) Then
                        If vSWBUG='1' Then
                            dbms_output.put_line(vString);
                        End If;
                        Execute Immediate vString;
                    End If;
                End If;
            End If;
            Commit;
        End loop;
        Commit;
    End loop;
    Resultst:= 1;

    --异常处理
    Exception
        When Others Then
            DBMS_OUTPUT.put_line('ERROR SQLCODE: ' ||SQLCODE);
            DBMS_OUTPUT.put_line('ERROR MSG: ' ||SQLERRM);
            Rollback;
            RAISE_APPLICATION_ERROR(SQLCODE,SQLERRM);
        Return;
End Prc_Crt_SysIndex;

Procedure Prc_Crt_ExtIndex (Resultst Out Integer, vTableCode varchar2 Default '') Is
    Type EmpCurTyp Is Ref Cursor;
    Mycursor      EmpCurTyp;

```

```

vString      Varchar2(8000);
vTab_Code    Varchar2(200);
vIX_NAME     Varchar2(200);
vColList     Varchar2(200);
vDbType      Varchar2(40);
vIx_Local    Varchar(1);
vSys_IxName   Varchar(200);
vPrc_Name    Varchar2(100);
vOwner       Varchar2(200);
Lownum       Number;

Begin
    vPrc_Name:='Prc_Crt_ExtIndex';
    Result:= 0;

    vString := 'Select TAB_CODE, IX_NAME, COL_LIST, ''DB_INT'' As DbType, IX_LOCAL From Sys_Extindex_List Where DB_INT=''1'' '||
Case When vTableCode Is Null Then '' Else 'And Tab_Code='''||vTableCode||''' ' End ||'Union All '||
        'Select TAB_CODE, IX_NAME, COL_LIST, ''DB_ORI'' As DbType, IX_LOCAL From Sys_Extindex_List Where DB_ORI=''1'' '||
Case When vTableCode Is Null Then '' Else 'And Tab_Code='''||vTableCode||''' ' End ||'Union All '||
        'Select TAB_CODE, IX_NAME, COL_LIST, ''DB_BUF'' As DbType, IX_LOCAL From Sys_Extindex_List Where DB_BUF=''1'' '||
Case When vTableCode Is Null Then '' Else 'And Tab_Code='''||vTableCode||''' ' End ||'Union All '||
        'Select TAB_CODE, IX_NAME, COL_LIST, ''DB_COR'' As DbType, IX_LOCAL From Sys_Extindex_List Where DB_COR=''1'' '||
Case When vTableCode Is Null Then '' Else 'And Tab_Code='''||vTableCode||''' ' End ||'Union All '||
        'Select TAB_CODE, IX_NAME, COL_LIST, ''DB_HIS'' As DbType, IX_LOCAL From Sys_Extindex_List Where DB_HIS=''1'' '||
Case When vTableCode Is Null Then '' Else 'And Tab_Code='''||vTableCode||''' ' End ||'Union All '||
        'Select TAB_CODE, IX_NAME, COL_LIST, ''DB_NOR'' As DbType, IX_LOCAL From Sys_Extindex_List Where DB_NOR=''1'' '||
Case When vTableCode Is Null Then '' Else 'And Tab_Code='''||vTableCode||''' ' End ||'Union All '||
        'Select TAB_CODE, IX_NAME, COL_LIST, ''DB_BAK'' As DbType, IX_LOCAL From Sys_Extindex_List Where DB_BAK=''1'' '||
Case When vTableCode Is Null Then '' Else 'And Tab_Code='''||vTableCode||''' ' End;
    vString := 'Select TAB_CODE, IX_NAME, COL_LIST, DbType, IX_LOCAL From ('||vString||') A Where Exists (Select 1 From
V_S_DUSER_LIST B Where A.DbType=B.DbType)';
    Open Mycursor For vString;
    Loop Fetch Mycursor Into vTab_Code, vIX_NAME, vColList, vDbType, vIx_Local;
        Exit When Mycursor%Notfound;
        vOwner := S_Ini_TableStruc.Fun_GetDbUser(vDbType);
        Select TAB_SNO Into vSys_IxName From V_S_TABLE_LIST Where TAB_CODE =vTab_Code;
        vSys_IxName:=' IX_'||vIX_NAME||'_ '||vSys_IxName;
        Select Count(1) Into Lownum From All_Indexes Where Table_Name=vTab_Code And Table_Owner=vOwner And Index_Name=vSys_IxName;
        If (Lownum = 0) Then
            vString := 'CREATE INDEX '||vOwner||'.'||vSys_IxName||' ON '|| vOwner||'.'||vTab_Code||' ('||vColList||')'||
                Case When vIx_Local='1' Then ' Local' Else '' End;
            If vSWBUG='1' Then
                dbms_output.put_line(vString);
            End If;
            Execute Immediate vString;
        End If;
    End loop;

    Result:= 1;

--异常处理
Exception
    When Others Then
        DBMS_OUTPUT.put_line('ERROR SQLCODE: '||SQLCODE);
        DBMS_OUTPUT.put_line('ERROR MSG: '||SQLERRM);
        Rollback;
        RAISE_APPLICATION_ERROR(SQLCODE,SQLERRM);
    Return;
End Prc_Crt_ExtIndex;

```

```

Procedure Prc_Alt_TabNoLog (Resulst Out Integer, LogType Integer, DbType varchar2 Default '', vTableCode varchar2 Default '')
Is
    Type EmpCurTyp Is Ref Cursor;
    Usecur          EmpCurTyp;
    Tabcur          EmpCurTyp;
    vString         Varchar2(8000);
    vDbType         Varchar2(200);
    Lownum          Number;
    vTab_Code       Varchar2(200);
    vDbuser         Varchar(200);
    vPrc_Resulst    Integer;
    vPrc_Name       Varchar2(100);
Begin
    vPrc_Name:='Prc_Alt_TabNoLog';
    Resulst:= 0;

    --DbType变量合理性扫描
    If DbType Is Not Null And DbType Not In ('DB_ORI','DB_BUF','DB_NOR','DB_BAK') Then
        Commit;
        Dbms_Output.Put_Line('Error: DbType参数输入值不合理，表结构NoLog状态修改过程放弃执行。');
        Resulst:= 1;
        Return;
    End If;

    --DbType变量合理性扫描
    If DbType Is Not Null Then
        vDbType := DbType;
        Select Count(1) Into Lownum From V_S_DUSER_LIST B Where B.DbType=vDbType;
        If Lownum=0 Then
            Commit;
            Dbms_Output.Put_Line('Error: DbType参数输入值异常，表结构NoLog状态修改过程放弃执行。');
            Return;
        End If;
    End If;

    --接口表启用标志位合理性扫描
    If vTableCode Is Null Then
        Select Count(1) Into Lownum From V_S_TABLE_LIST;
    Else
        Select Count(1) Into Lownum From V_S_TABLE_LIST Where Tab_Code=vTableCode;
    End If;
    If Lownum=0 Then
        Commit;
        Dbms_Output.Put_Line('Error: V_S_TABLE_LIST中不存在已生效的接口表配置信息，表结构NoLog状态修改过程放弃执行。');
        Return;
    End If;

    If vTableCode Is Null Then
        vString := 'Select A.Tab_Code From V_S_TABLE_LIST A Order By A.Tab_Note';
    Else
        vString := 'Select '''||vTableCode||''' as Tab_Code From dual';
    End If;
    Open Tabcur For vString;
    Loop Fetch Tabcur Into vTab_Code;
    Exit When Tabcur%Notfound;
    If DbType Is Null Then
        vString := 'Select B.Dbuser From V_S_DUSER_LIST B Where B.DbType=''DB_ORI'' Or B.DbType=''DB_BUF'' Or B.DbType=''DB_NOR''

```

```
Or B.DbType='DB_BAK' Order By B.DbType;
```

```
Else
```

```
    vString := 'Select B.Dbuser From V_S_DUSER_LIST B Where B.DbType='''||DbType|'''';
```

```
End If;
```

```
Open Usecur For vString;
```

```
Loop Fetch Usecur Into vDbuser;
```

```
Exit When Usecur%Notfound;
```

```
--表结构NoLog状态修改
```

```
If (LogType = 1) Then
```

```
    vString := 'Alter Table '||vDbuser||'. '||vTab_Code||' Nologging';
```

```
Else
```

```
    vString := 'Alter Table '||vDbuser||'. '||vTab_Code||' Logging';
```

```
End If;
```

```
If vSWBUG='1' Then
```

```
    dbms_output.put_line(vString);
```

```
End If;
```

```
Execute Immediate vString;
```

```
Commit;
```

```
End loop;
```

```
Commit;
```

```
End loop;
```

```
Result:= 1;
```

```
--异常处理
```

```
Exception
```

```
When Others Then
```

```
    DBMS_OUTPUT.put_line(' ERROR SQLCODE: '||SQLCODE);
```

```
    DBMS_OUTPUT.put_line(' ERROR MSG: '||SQLERRM);
```

```
Rollback;
```

```
RAISE_APPLICATION_ERROR(SQLCODE,SQLERRM);
```

```
Return;
```

```
End Prc_Alt_TabNoLog;
```

```
--根据配置信息初始化接入数据表机构(主程序)
```

```
Procedure Prc_Main_IniTabStruc (Result Out Integer, DbType varchar2 Default '', vTableCode varchar2 Default '') Is
```

```
    vPrc_Result Integer;
```

```
    vPrc_Name Varchar2(100);
```

```
Begin
```

```
    vPrc_Name:='Prc_Main_IniTabStruc';
```

```
    Result:= 1;
```

```
    dbms_output.put_line(' 处理过程s_ini_tablestruc.prc_drp_table{参数DbType='''||DbType|'''' 参数vTableCode='''||vTableCode|''''}执行开始');
```

```
    vPrc_Result:= 0;
```

```
    s_ini_tablestruc.prc_drp_table(vPrc_Result, DbType, vTableCode);
```

```
    Result := Case When vPrc_Result > 0 Then Result Else vPrc_Result End;
```

```
    dbms_output.put_line(' 处理过程s_ini_tablestruc.prc_drp_table{参数DbType='''||DbType|'''' 参数vTableCode='''||vTableCode|''''}执行完成');
```

```
    dbms_output.put_line(' 处理过程s_ini_tablestruc.prc_crt_table{参数DbType='''||DbType|'''' 参数vTableCode='''||vTableCode|''''}执行开始');
```

```
    vPrc_Result:= 0;
```

```
    s_ini_tablestruc.prc_crt_table(vPrc_Result, DbType, vTableCode);
```

```
    Result := Case When vPrc_Result > 0 Then Result Else vPrc_Result End;
```

```
    dbms_output.put_line(' 处理过程s_ini_tablestruc.prc_crt_table{参数DbType='''||DbType|'''' 参数vTableCode='''||vTableCode|''''}执行完成');
```

```
    dbms_output.put_line(' 处理过程s_ini_tablestruc.prc_alt_colcomment{参数DbType='''||DbType|'''' 参数
```

```

vTableCode="" || vTableCode || ' " } 执行开始');
    vPrc_Resulst:= 0;
    s_ini_tablestruc.prc_alt_colcomment(vPrc_Resulst, DbType, vTableCode);
    Resultst := Case When vPrc_Resulst > 0 Then Resultst Else vPrc_Resulst End;
    dbms_output.put_line(' 处理过程s_ini_tablestruc.prc_alt_colcomment{参数DbType="" || DbType || ' " 参数
vTableCode="" || vTableCode || ' " } 执行完成');

    dbms_output.put_line(' 处理过程s_ini_tablestruc.prc_crt_sysindex{参数vTableCode="" || vTableCode || ' " } 执行开始');
    vPrc_Resulst:= 0;
    s_ini_tablestruc.prc_crt_sysindex(vPrc_Resulst, vTableCode);
    Resultst := Case When vPrc_Resulst > 0 Then Resultst Else vPrc_Resulst End;
    dbms_output.put_line(' 处理过程s_ini_tablestruc.prc_crt_sysindex{参数vTableCode="" || vTableCode || ' " } 执行完成');

    dbms_output.put_line(' 处理过程s_ini_tablestruc.prc_crt_extindex{参数vTableCode="" || vTableCode || ' " } 执行开始');
    vPrc_Resulst:= 0;
    s_ini_tablestruc.prc_crt_extindex(vPrc_Resulst, vTableCode);
    Resultst := Case When vPrc_Resulst > 0 Then Resultst Else vPrc_Resulst End;
    dbms_output.put_line(' 处理过程s_ini_tablestruc.prc_crt_extindex{参数vTableCode="" || vTableCode || ' " } 执行完成');

    dbms_output.put_line(' 处理过程s_ini_tablestruc.prc_alt_tabnolog{参数LogType="1" 参数DbType="" || DbType || ' " 参数
vTableCode="" || vTableCode || ' " } 执行开始');
    vPrc_Resulst:= 0;
    s_ini_tablestruc.prc_alt_tabnolog(vPrc_Resulst, 1, DbType, vTableCode);
    Resultst := Case When vPrc_Resulst > 0 Then Resultst Else vPrc_Resulst End;
    dbms_output.put_line(' 处理过程s_ini_tablestruc.prc_alt_tabnolog{参数LogType="1" 参数DbType="" || DbType || ' " 参数
vTableCode="" || vTableCode || ' " } 执行完成');

```

--异常处理

Exception

When Others Then

DBMS_OUTPUT.put_line(' ERROR SQLCODE: ' || SQLCODE);

DBMS_OUTPUT.put_line(' ERROR MSG: ' || SQLERRM);

Rollback;

RAISE_APPLICATION_ERROR(SQLCODE, SQLERRM);

Return;

End Prc_Main_InitabStruc;

--生成数据校验程序所需要的相关索引

Procedure Prc_Crt_ChkIndex (Resultst Out Integer, DbType varchar2, vIxFlag Varchar2, vCollList Varchar2, vTable Varchar2 Default
'') Is

Type EmpCurTyp Is Ref Cursor;

Tabcur EmpCurTyp;

vString Varchar2(8000);

vTab_Code Varchar(200);

vTab_Sno Varchar(200);

vSys_IxName Varchar(200);

vOwner Varchar2(200);

vPrc_Name Varchar2(100);

Lownum Number;

Begin

vPrc_Name:='Prc_Crt_ChkIndex';

Resultst:= 0;

vOwner := S_Ini_TableStruc.Fun_GetDbUser(DbType);

If vTable Is Null Then

vString:= 'Select A.Tab_Code, NVL(TAB_SNO,TAB_CODE) As Tab_Sno From V_S_TABLE_LIST A Order By A.Tab_Note';

Else

vString:= 'Select A.Tab_Code, NVL(TAB_SNO,TAB_CODE) As Tab_Sno From V_S_TABLE_LIST A Where A.Tab_Code=''' || vTable || ''''


```

Order By A.Tab_Note';
End If;
--dbms_output.put_line(vString);
Open Tabcur For vString;
Loop Fetch Tabcur Into vTab_Code, vTab_Sno;
Exit When Tabcur%Notfound;
vSys_IxName:=' IX_' || vIxFlag || '_' || vTab_Sno;
Select Count(1) Into Lownum From All_Indexes Where Table_Name=vTab_Code And Table_Owner=vOwner And Index_Name=vSys_IxName;
If (Lownum = 0) Then
    vString := 'CREATE INDEX ' || vOwner || '.' || vSys_IxName || ' ON ' || vOwner || '.' || vTab_Code || ' (' || vColList || ')';
    Select Count(1) Into Lownum From All_Tables Where OWNER=vOwner And TABLE_NAME=vTab_Code;
    If (Lownum = 1) Then
        If vSWBUG='1' Then
            dbms_output.put_line(vString);
        End If;
        Execute Immediate vString;
    End If;
End If;
End loop;
Resultst:= 1;

--异常处理
Exception
When Others Then
    DBMS_OUTPUT.put_line(' ERROR SQLCODE: ' || SQLCODE);
    DBMS_OUTPUT.put_line(' ERROR MSG: ' || SQLERRM);
    Rollback;
    RAISE_APPLICATION_ERROR(SQLCODE, SQLERRM);
Return;
End Prc_Crt_ChkIndex;

Procedure Prc_Crt_ExeInprc (Resultst Out Integer) Is
Type EmpCurTyp Is Ref Cursor;
Mycursor      EmpCurTyp;
vPRC_Com      Varchar2(800);
vDbType       Varchar2(40);
vPrc_Name     Varchar2(100);
vPrc_Resultst Integer;
Begin
    vPrc_Name:='Prc_Crt_ExeInprc';
    Resultst:= 1;

    Open Mycursor For
    Select PRC_COM, DbType From (
        Select PRC_COM, 'DB_INT' As DbType From Sys_Config_Iniprc Where DB_INT='1' Union All
        Select PRC_COM, 'DB_ORI' As DbType From Sys_Config_Iniprc Where DB_ORI='1' Union All
        Select PRC_COM, 'DB_COR' As DbType From Sys_Config_Iniprc Where DB_COR='1' Union All
        Select PRC_COM, 'DB_NOR' As DbType From Sys_Config_Iniprc Where DB_NOR='1' Union All
        Select PRC_COM, 'DB_BAK' As DbType From Sys_Config_Iniprc Where DB_BAK='1'
    ) A Where Exists (Select 1 From V_S_DUSER_LIST B Where A.DbType=B.DbType);
    Loop Fetch Mycursor Into vPRC_Com, vDbType;
    Exit When Mycursor%Notfound;
    vPRC_Com := 'begin ' || Replace(vPRC_Com, '@DbType@', '''' || vDbType || '''' ) || ';end;';
    dbms_output.put_line(' 数据校验处理过程: { ' || vPRC_Com || ' } 开始');
    vPrc_Resultst:= 0;
    Execute Immediate vPRC_Com Using Out vPrc_Resultst;
    Resultst := Case When vPrc_Resultst > 0 Then Resultst Else vPrc_Resultst End;
    dbms_output.put_line(' 数据校验处理过程: { ' || vPRC_Com || ' } 完成');

```

```

End loop;

--异常处理
Exception
When Others Then
    DBMS_OUTPUT.put_line(' ERROR  SQLCODE: ' ||SQLCODE);
    DBMS_OUTPUT.put_line(' ERROR  MSG: ' ||SQLERRM);
    Rollback;
    RAISE_APPLICATION_ERROR(SQLCODE,SQLERRM);
Return;
End Prc_Crt_ExeInprc;

Function Fun_GetCollLength (vTable Varchar2, vColumn Varchar2) Return Number Is
/*根据指定的“表名”vTable、“字段名”vColumn返回相应数据表字段的字段长度*/
vColLen Number(5);
Script Varchar2(4096);
Begin
    Script := 'Select DATA_LENGTH From ' ||v_Cet1||'.SYS_COLUMN_LIST Where TAB_CODE=:vTable And COL_CODE=:vColumn';
    Execute Immediate Script Into vColLen Using In vTable, vColumn;
    Return nvl(vColLen, 0);
End Fun_GetCollLength;

Procedure Prc_Crt_PartitionYwsj (Resulst Out Integer, vYear Varchar2) Is
    Type EmpCurTyp Is Ref Cursor;
    Usecur EmpCurTyp;
    Mycursor EmpCurTyp;
    Script Varchar2(8000);
    vDbType Varchar2(200);
    vColLen Number;
    vTable Varchar2(200);
    vPOther Varchar2(200);
    vDType Varchar2(200);
    vPName Varchar2(200);
    vYWSJ Varchar2(20);
    Monnum Number;
    Lownum Number;
    vPrc_Name Varchar2(100);
Begin
    vPrc_Name:='Prc_Crt_PartitionYwsj';
    Resulst:= 0;

    If S_Ini_TableStruc.Fun_IsDate(vYear||'0101', 'YYYYMMDD')=0 Then
        Dbms_Output.Put_Line(' Warning: 对vYear[' ||vYear||']参数所提供的数值，不是一个合理的年份值！处理过程自动跳出');
        Return;
    End If;

    If To_Date(vYear||'0101', 'YYYYMMDD')<To_Date(v_BTime||'01', 'YYYYMMDD') Then
        Dbms_Output.Put_Line(' Warning: 对vYear[' ||vYear||']参数所提供的数值，小于'||Substr(v_BTime,1,4)||'系统认为分区创建不合理！
处理过程自动跳出');
        Return;
    End If;

    Open Usecur For Select Distinct DB_TYPE From SYS_PART_LIST A Where Exists (Select 1 From V_S_DUSER_LIST B Where
A.DB_TYPE=B.DbType And PAT_TYPE='1') Order By DB_TYPE;
    Loop Fetch Usecur Into vDbType;
    Exit When Usecur%Notfound;
    Delete From Tmp_Partition_Name;
    Script := 'Insert Into Tmp_Partition_Name(Table_Owner, Table_Name, Partition_Name, High_Value)' ||Chr(10)||

```

```

        'Select Table_Owner, Table_Name, Partition_Name, To_lob(High_Value) From All_Tab_Partitions A' || Chr(10) ||
        'Where Table_Owner = ''' || S_Ini_TableStruc.Fun_GetDbUser(vDbType) || ''' ' ||
        'And Exists (Select 1 From sys_part_list B Where B.PAT_TYPE='''1''' And DB_TYPE='''' || vDbType || ''' And
B.TAB_CODE=A.Table_Name)';
    If vSWBUG='1' Then
        dbms_output.put_line(Script);
    End If;
    Execute Immediate Script;
    Open Mycursor For
    Select Table_Name, Partition_Name From Tmp_Partition_Name Where DBMS_LOB.INSTR(High_Value,'MAXVALUE')>0 And
Table_Owner=S_Ini_TableStruc.Fun_GetDbUser(vDbType);
    Loop Fetch Mycursor Into vTable, vPOther;
    Exit When Mycursor%Notfound;
    Select Count(1) Into Lownum From Tmp_Partition_Name Where Table_Name=vTable And Partition_Name<>vPOther And
Partition_Name>='M' || vYear || '01';
    If Lownum>0 Then
        Dbms_Output.Put_Line('Warning: 对vYear[' || vYear || ']参数所提供的数值,
表' || S_Ini_TableStruc.Fun_GetDbUser(vDbType) || '.' || vTable || '已经进行过将数据从默认分区分割的操作! 处理过程自动忽略');
        Goto NextLoop;
    End If;
    vDbType := 'N/A';
    Select B.DATA_TYPE, B.DATA_LENGTH Into vDbType, vColLen
    From Sys_Part_List A, ALL_Tab_Columns B
    Where Upper(A.DB_TYPE)=vDbType And A.Tab_Code=vTable And B.Table_Name=vTable And A.Pat_Type='1'
    And A.Pat_Colu = B.Column_Name And B.Owner = S_Ini_TableStruc.Fun_GetDbUser(vDbType)
    And ((Instr(Upper(B.Data_Type),'CHAR')>0) Or Upper(B.Data_Type)='DATE');
    If nvl(vDbType,'N/A')<>'N/A' Then
        Monnum := 1;
        While Monnum <= 12 Loop
            vPName := 'M' || vYear || Substr('00' || to_char(Monnum),-2);
            vWYSJ := To_Char(Add_Months(To_Date(vYear || Substr('00' || to_char(Monnum),-2) || '01','YYYYMMDD'),1),'YYYYMMDD');
            Select Count(1) Into Lownum From Tmp_Partition_Name
            Where Table_Owner=S_Ini_TableStruc.Fun_GetDbUser(vDbType) And Table_Name=vTable And Partition_Name=vPName;
            If Lownum = 0 Then
                Script := 'Alter Table ' || S_Ini_TableStruc.Fun_GetDbUser(vDbType) || '.' || vTable || ' ' ||
                Case When (vDbType='DB_COR' Or vDbType='DB_HIS') And Instr(Upper(vDbType),'CHAR')>0 Then 'Split Partition
' || vPOther || ' At(' || vWYSJ ||
                Case When vColLen=16 Then '/' || Lpad('0',6,'0') || '/'
                When vColLen>8 Then Lpad('0',vColLen-8,'0')
                Else ''End||''') '
                Else
                'Split Partition ' || vPOther || ' At(To_Date(' || vWYSJ || ''',''YYYYMMDD')) '
                End || Chr(10) ||
                'Into (Partition ' || vPName || ', Partition ' || vPOther || ');
            If vSWBUG='1' Then
                dbms_output.put_line(Script);
            End If;
            Execute Immediate Script;
        End If;
        Monnum := Monnum+1;
    End Loop;
End If;
<<NextLoop>>
Null;
End Loop;
Commit;
End loop;

```

```

Result:= 1;

--异常处理
Exception
When Others Then
    DBMS_OUTPUT.put_line(' ERROR  SQLCODE: ' ||SQLCODE);
    DBMS_OUTPUT.put_line(' ERROR  MSG: ' ||SQLERRM);
    Rollback;
    RAISE_APPLICATION_ERROR(SQLCODE,SQLERRM);
Return;
End Prc_Crt_PartitionYwsj;

Procedure Prc_Crt_PartitionList (Result Out Integer) Is
Type EmpCurTyp Is Ref Cursor;
Usecur          EmpCurTyp;
Listcur         EmpCurTyp;
Mycursor        EmpCurTyp;
Script          Varchar2(8000);
vDbType         Varchar2(200);
vTable          Varchar2(200);
vPOther         Varchar2(200);
vListVal        Varchar2(200);
vPatColu        Varchar2(200);
vPName          Varchar2(200);
vPVal           Varchar2(20);
Lownum          Number;
vPrc_Name       Varchar2(100);
Begin
    vPrc_Name:='Prc_Crt_PartitionList';
    Result:= 0;

    Open Usecur For Select Distinct DB_TYPE From SYS_PART_LIST A Where Exists (Select 1 From V_S_DUSER_LIST B Where
A.DB_TYPE=B.DbType And PAT_TYPE='3') Order By DB_TYPE;
    Loop Fetch Usecur Into vDbType;
        Exit When Usecur%Notfound;
        Delete From Tmp_Partition_Name;
        Script := 'Insert Into Tmp_Partition_Name(Table_Owner, Table_Name, Partition_Name, High_Value)' ||Chr(10)||
        'Select Table_Owner, Table_Name, Partition_Name, To_lob(High_Value) From All_Tab_Partitions A' ||Chr(10)||
        'Where Table_Owner = ''' ||S_Ini_TableStruc.Fun_GetDbUser(vDbType)||''' ' ||
        'And Exists (Select 1 From sys_part_list B Where B.PAT_TYPE='''3'' And DB_TYPE=''' ||vDbType||''' And
B.TAB_CODE=A.Table_Name)';
        If vSWBUG='1' Then
            dbms_output.put_line(Script);
        End If;
        Execute Immediate Script;
        Open Mycursor For
            Select Table_Name, Partition_Name From Tmp_Partition_Name Where DBMS_LOB.INSTR(High_Value,'DEFAULT')>0 And
Table_Owner=S_Ini_TableStruc.Fun_GetDbUser(vDbType);
        Loop Fetch Mycursor Into vTable, vPOther;
            Exit When Mycursor%Notfound;
            Select LIST_VALUES Into vListVal From SYS_PART_LIST Where PAT_TYPE='3' And DB_TYPE=vDbType And TAB_CODE=vTable;
            Select PAT_COLU Into vPatColu From SYS_PART_LIST Where PAT_TYPE='3' And DB_TYPE=vDbType And TAB_CODE=vTable;
            Script := 'Alter Table ' ||S_Ini_TableStruc.Fun_GetDbUser(vDbType)||'.' ||vTable||' Enable Row Movement';
            If vSWBUG='1' Then
                dbms_output.put_line(Script);
            End If;
            Execute Immediate Script;
            Open Listcur For Select COL_VALUE From Table(S_Ini_TableStruc.Fun_GetCharList(vListVal,', '));

```

```

Loop Fetch Listcur Into vListVal;
Exit When Listcur%Notfound;
vPName := substr(vListVal, Instr(vListVal, ',')+1, Length(vListVal)-Instr(vListVal, ','));
vPVal := substr(vListVal, 1, Instr(vListVal, ',')-1);
Select Count(1) Into Lownum From Tmp_Partition_Name Where Table_Name=vTable And Partition_Name=vPName;
If Lownum>0 Then
    Dbms_Output.Put_Line('Warning: 对列表分区枚举项的'||vListVal||']参数值,
表'||S_Ini_TableStruc.Fun_GetDbUser(vDbType)||'.'||vTable||'已经进行过将数据从默认分区分割的操作! 处理过程自动忽略');
Else
    Script := 'Alter Table '||S_Ini_TableStruc.Fun_GetDbUser(vDbType)||'.'||vTable||' Split Partition '||vPOther||'
Values('||vPVal||') '||
        'Into (Partition '||vPName||', Partition '||vPOther||')';
    If vSWBUG='1' Then
        dbms_output.put_line(Script);
    End If;
    Execute Immediate Script;
End If;
End Loop;
Null;
End Loop;
Commit;
End loop;
Result:= 1;

--异常处理
Exception
When Others Then
    DBMS_OUTPUT.put_line('ERROR SQLCODE: '||SQLCODE);
    DBMS_OUTPUT.put_line('ERROR MSG: '||SQLERRM);
    Rollback;
    RAISE_APPLICATION_ERROR(SQLCODE, SQLERRM);
Return;
End Prc_Crt_PartitionList;

Function Fun_IsDate(str varchar2, fmt varchar2 default null, nls varchar2 default null) Return Number Is
/*函数为自定义的 功能等同于sqlserver的isdate函数, 是时间格式就返回1, 否则返回0*/
v_date date;
v_fmt varchar2(100) default fmt;
v_nls varchar2(100) default nls;
Begin
    If fmt Is Null Then
        Select Value Into v_fmt From v$nls_parameters Where parameter='NLS_DATE_FORMAT';
    End If;
    If nls Is Null Then
        Select 'NLS_DATE_LANGUAGE='''||value||'''' Into v_nls From v$nls_parameters Where parameter='NLS_DATE_LANGUAGE';
    End If;
    v_date := to_date(str, v_fmt, v_nls);
    Return 1;
Exception
When Others Then
    Return 0;
End Fun_IsDate;
End S_Ini_TableStruc;

```