CREATE OR REPLACE PROCEDURE cen_interface_to_core(v_tab_name VARCHAR2) IS

```
CREATE OR REPLACE PROCEDURE cen_interface_to_core(v_tab_name VARCHAR2) IS
  v_col_code VARCHAR2(32);
  v_pk_seq   NUMBER;
  pk_columns VARCHAR2(128);
  pk_column  VARCHAR2(128);
  v_str      VARCHAR2(2000);
  v_str_set  VARCHAR2(2000);
  v_str_int1 VARCHAR2(2000);
  v_str_int2 VARCHAR2(2000);

  v_ind_name VARCHAR2(32);
  v_scripts   VARCHAR2(4096);

  v_count_row    PLS_INTEGER; --统计处理数据总量
  v_count_new    PLS_INTEGER; --统计新增记录数
  v_count_update PLS_INTEGER; --统计更新记录数
  v_count_cancel PLS_INTEGER; --统计撤销记录数

  --游标
  TYPE tp_cursor IS REF CURSOR;
  c_col tp_cursor;
  CURSOR c_ind_name IS
    SELECT index_name
    FROM   all_indexes
    WHERE  1 = 1
    AND    owner = 'CEN_INTERFACE_V20'
    AND    table_name = v_tab_name
    AND    index_type = 'NORMAL';
  --c_def_table tp_cursor;
  --c_date      tp_cursor;

  --处理医保数据
  v_yljgdm   VARCHAR2(16);
  v_ref_type NUMBER;
  v_script   VARCHAR2(9000);
  --v_col_date VARCHAR2(16);
  --v_date     VARCHAR2(6);

  v_exception_msg VARCHAR2(600);
  --v_processing_info VARCHAR2(100);

BEGIN
  v_count_update := 0;
  v_count_new    := 0;
  EXECUTE IMMEDIATE 'ALTER SESSION ENABLE PARALLEL DML';

  /*数据备份*/
  BEGIN
    v_script := 'INSERT /*+ append nologging */ INTO cen_original_v20.' ||
                v_tab_name || ' a (' ||
                s_ini_tablestruc.fun_getcolumn(v_tab_name) ||
                ',jlgxsj,posid) SELECT /*+ parallel (b,4) */' ||
                s_ini_tablestruc.fun_getcolumn(v_tab_name) ||
                ',jlgxsj,posid  FROM cen_interface_v20.' || v_tab_name ||
                ' b WHERE llbz = 1';
  --dbms_output.put_line(v_script || ';');
  EXECUTE IMMEDIATE v_script;
  v_count_row := SQL%ROWCOUNT;
  COMMIT;

  --记入日志
  INSERT INTO t_def_dataprocessing_log
    (id, processing_date, p_tablename, processing_info, process_count,
     log_type, info_class)
  VALUES
    (seq_dataprocessing_log.nextval, SYSDATE, v_tab_name,
     '完成对' || v_tab_name || '表的备份', v_count_row, '1', '1');
  COMMIT;
END;

/*清理重复数据*/
BEGIN
  SELECT a.pk_columns
  INTO   pk_column
  FROM   v_s_table_list a
  WHERE  a.tab_code = v_tab_name;

  v_str := 'DELETE FROM cen_interface_v20.' || v_tab_name || chr(13) ||
           'WHERE rowid IN (SELECT rd FROM (SELECT /*+ parallel(b.64) */ rowid rd ,row_number() OVER(PARTITION BY ' ||
           pk_column ||
           ' ORDER BY  jlgxsj DESC) rn FROM cen_interface_v20.' ||
           v_tab_name || ' b WHERE llbz = 1)  WHERE rn > 1)  ';
  --DBMS_OUTPUT.put_line(V_STR || ';' || CHR(13));
  EXECUTE IMMEDIATE v_str;
  COMMIT;
END;

/*数据整合*/
BEGIN
  /*医保数据预处理*/
  BEGIN
    SELECT a.ref_type
    INTO   v_ref_type
    FROM   sys_table_list a
    WHERE  a.tab_code = v_tab_name;

    IF v_ref_type = 0
    THEN

      SELECT a.col_jgdm
      INTO   v_yljgdm
      FROM   sys_table_list a
      WHERE  a.tab_code = v_tab_name;

      v_str := 'truncate table TMP_BUFV_YWSJLIST';
      EXECUTE IMMEDIATE v_str;

      v_str := 'INSERT INTO tmp_bufv_ywsjlist (yljgdm,ywsj)' || chr(13) ||
               'SELECT DISTINCT ' || v_yljgdm || '  , f_jysj' || chr(13) ||
               'FROM cen_interface_v20.' || v_tab_name || chr(13) ||
               'WHERE  llbz =1 ';
      --dbms_output.put_line(v_str);
      EXECUTE IMMEDIATE v_str;
    COMMIT;

    v_str := 'DELETE FROM cen_core_v20.' || v_tab_name || ' a ' ||
             chr(13) ||
             'WHERE exists (SELECT 1 FROM tmp_bufv_ywsjlist b WHERE a.f_jysj =b.ywsj AND a.' ||
             v_yljgdm || ' = b.yljgdm )';
    --DBMS_OUTPUT.put_line(V_STR || ';' || CHR(13));
    EXECUTE IMMEDIATE v_str;
    --取删除记录数(更新记录数)
    v_count_update := SQL%ROWCOUNT;
    COMMIT;
  END IF;
END;

--拼接并执行MERGE INTO动态SQL
/*ON连接内容拼接处理*/
BEGIN
  SELECT a.ext_columns
  INTO   pk_columns
  FROM   v_s_table_list a
  WHERE  tab_code = v_tab_name;

  OPEN c_col FOR
    SELECT a.col_code, a.pk_seq
    FROM   sys_column_list a
    WHERE  a.tab_code = v_tab_name;
  LOOP
    FETCH c_col
      INTO v_col_code, v_pk_seq;
    EXIT WHEN c_col%NOTFOUND;
    --循环拼接set语句段
    IF v_pk_seq = 0
    THEN
      v_str_set := v_str_set || 'a.' || v_col_code || '= b.' ||
                   v_col_code || ',';
    END IF;
    v_str_int1 := v_str_int1 || v_col_code || ',';
    v_str_int2 := v_str_int2 || 'b.' || v_col_code || ',';
  END LOOP;
  v_str_set  := substr(v_str_set, 1, length(v_str_set) - 1);
  v_str_int1 := substr(v_str_int1, 1, length(v_str_int1) - 1);
  v_str_int2 := substr(v_str_int2, 1, length(v_str_int2) - 1);
END;

--拼接merge语句（仅用于更新）
v_str := 'MERGE /*+ parallel(a,4) */ INTO cen_core_v20.' || v_tab_name ||
         ' a ' || chr(13) || 'USING (SELECT * FROM cen_interface_v20.' ||
         v_tab_name || ' b WHERE EXISTS (SELECT 1 FROM cen_core_v20.' ||
         v_tab_name || ' a WHERE ' ||
         REPLACE(REPLACE(pk_columns, '@T1@', 'A'), '@T2@', 'B') ||
         ') AND b.llbz = 1 ) b ' || chr(13) || 'ON (' ||
         REPLACE(REPLACE(pk_columns, '@T1@', 'A'), '@T2@', 'B') ||
         ') WHEN MATCHED THEN UPDATE set ' || v_str_set || chr(13) ||
         ',a.jlgxsj = SYSDATE ';

--DBMS_OUTPUT.put_line(V_STR || ';' || CHR(13));
EXECUTE IMMEDIATE v_str;
--取merge影响的记录数(更新记录数)
```

```
  v_count_update := v_count_update + SQL%ROWCOUNT;
  COMMIT;
  --拼接Insert语句(仅用于新增)
  v_str := 'INSERT /*+ append nologging */ INTO cen_core_v20.' ||
           v_tab_name || '(' || v_str_int1 || ' ) ' || chr(13) ||
           'SELECT ' || v_str_int2 || chr(13) ||
           'FROM cen_interface_v20.' || v_tab_name || ' b  ' || chr(13) ||
           'WHERE  NOT EXISTS (SELECT 1 FROM cen_core_v20.' || v_tab_name ||
           ' a WHERE ' ||
           REPLACE(REPLACE(pk_columns, '@T1@', 'A'), '@T2@', 'B') || ') ' ||
           chr(13) || 'AND b.llbz = 1';
  --DBMS_OUTPUT.put_line(V_STR || ';' || CHR(13));
  EXECUTE IMMEDIATE v_str;
  --取insert影响的记录数(新增记录数)
  v_count_new := SQL%ROWCOUNT;
  COMMIT;

  --写日志表
  INSERT INTO t_def_dataprocessing_log
    (id, processing_date, p_tablename, processing_info, process_count,
     new_count, update_count, cancel_count, log_type, info_class) --写日志记录运行状态
  VALUES
    (seq_dataprocessing_log.nextval, SYSDATE, v_tab_name,
     '完成对表' || v_tab_name || '入cen_core_v20库处理',
     v_count_new + v_count_update, v_count_new, v_count_update,
     v_count_cancel, '2', '1');

  --单表处理完成，拼接DELETE语句并执行
  v_script := 'DELETE from cen_interface_v20.' || v_tab_name || chr(13) ||
              'WHERE llbz = 1';
  --dbms_output.put_line(v_script || ';');
  EXECUTE IMMEDIATE v_script;
  COMMIT;
END;

/*空间清理*/
BEGIN
  v_script := 'ALTER TABLE cen_interface_v20.' || v_tab_name ||
              ' SHRINK SPACE';
  --dbms_output.put_line(v_script || ';');
  EXECUTE IMMEDIATE v_script;
END;

BEGIN
  OPEN c_ind_name;
  LOOP
    EXIT WHEN c_ind_name%NOTFOUND;

    FETCH c_ind_name
      INTO v_ind_name;

    v_scripts := 'ALTER INDEX cen_interface_v20.' || v_ind_name ||
                 ' REBUILD NOLOGGING';

    --dbms_output.put_line(v_scripts);
    EXECUTE IMMEDIATE v_scripts;

  END LOOP;
```

```
    CLOSE c_ind_name;
  END;

  --异常处理块
EXCEPTION
  WHEN OTHERS THEN
    v_exception_msg := substr(SQLERRM, 1, 500);
    v_script        := substr(v_script, 1, 3000);
    INSERT INTO t_def_dataprocessing_log
      (id, processing_date, p_tablename, processing_info, exception_msg,
       exception_script, log_type, info_class) --写日志记录异常信息
    VALUES
      (seq_dataprocessing_log.nextval, SYSDATE,
       'cen_interface_v20.' || v_tab_name,
       '处理过程中出现异常,详细信息见v_exception_msg和v_script字段的内容.', v_exception_msg,
       v_str, '2', '3');
    COMMIT;
END;
```