```
/* Fernando Medeiros


It generate an output to create XML file.


-----------------------------------------------------------------------------

How to use:

SELECT XML_TEXT FROM XML_GENERATOR( ' select * from users where user_id < 4

' )

or see the pictures


-----------------------------------------------------------------------------

*SP: XML_GENERATOR - 1/2*

(depends on the other 2/2)*/
CREATE PROCEDURE XML_GENERATOR (
    PHEADER VARCHAR(1024),
    PSQL VARCHAR(1024)
)
RETURNS (
    XML_TEXT VARCHAR(1024)
)
AS
    DECLARE VARIABLE FIELD VARCHAR(30);
DECLARE VARIABLE
fields VARCHAR(1024);
DECLARE VARIABLE
fields_begin SMALLINT;
DECLARE VARIABLE
fields_end SMALLINT;
DECLARE VARIABLE
from_end SMALLINT;
DECLARE VARIABLE
sql_end SMALLINT;
DECLARE VARIABLE
sql_right VARCHAR(1024);
DECLARE VARIABLE
i INTEGER;
DECLARE VARIABLE
records INTEGER;
DECLARE VARIABLE
table_name VARCHAR(30);
BEGIN
```

```
/* uppercase and trim psql parameter */
psql = UPPER(TRIM(
psql));



/* extract fields from psql parameter */
fields_begin = POSITION(' ',:
psql)+1;
fields_end = POSITION(' FROM ',
psql);
fields = SUBSTRING(
psql FROM :
fields_begin
              FOR :
fields_end-:
fields_begin) ||
              ',';/* for extract the last field */



/* extract table name, where, order by, etc... */
from_end =
fields_end + 6;
sql_end = CHAR_LENGTH(
psql)-
from_end+1;
sql_right = SUBSTRING(
psql FROM :
from_end FOR
sql_end) || ' ';



/* extract only table name for "all fields" and */

/* open/close tags in xml */
table_name = SUBSTRING(
sql_right FROM 1
                 FOR POSITION(' ',
sql_right)-1);



/* if fields = '*', fields = "all fields" of the table */
  IF (
fields = '*,') THEN BEGIN
fields = '';
    FOR
SELECT trim(rdb$field_name)
FROM rdb$relation_fields
WHERE
    rdb$relation_name =
    :table_name
ORDER BY rdb$field_id
    INTO :FIELD DO
```

```
fields =
fields || :FIELD || ',';
  END



/* if pheader xml isnull... */
  IF (:
pheader IS NULL) THEN pheader = '';



/* returns pheader to client... */
xml_text = :
pheader;
  SUSPEND;



/* open table */
xml_text = '<' || :
table_name || '>';
  SUSPEND;



/* record count... */
EXECUTE STATEMENT cast('select count(*) from ' ||
        :sql_right AS VARCHAR(1024))
INTO :records;



/* fields from another stored procedure */

/* loop first to last record */
i = 1;
  WHILE (
i <=
records ) DO
  BEGIN



/* open record */
xml_text = '';
    SUSPEND;



/* fields and values in another stored procedure */
    FOR
SELECT xml_text
FROM xml_generator_fields(:fields,
    :sql_right)
WHERE
```

```
    record =
    :i INTO :
xml_text DO
    SUSPEND;



/* close record */
xml_text = '';
    SUSPEND;



/* next record */
i =
i + 1;

  END



/* close table */
xml_text = ' || :
table_name || '>';
  SUSPEND;

END






/* *SP: XML_GENERATOR_FIELDS - 2/2* */

CREATE PROCEDURE XML_GENERATOR_FIELDS (
    FIELDS VARCHAR(1024),
    SQL_RIGHT VARCHAR(1024)
)
RETURNS (
    RECORD INTEGER,
    XML_TEXT VARCHAR(1024)
)
AS
    DECLARE VARIABLE i INTEGER;
    DECLARE VARIABLE len INTEGER;
    DECLARE VARIABLE c CHAR(1);
    DECLARE VARIABLE FIELD VARCHAR(100);
BEGIN



/* lenght fields */
len = CHAR_LENGTH(
fields);
```

```
/* loop in fields string */
i = 1;
  FIELD = ''; /* clear field */
  WHILE (
i <=
len) DO
  BEGIN


/* current char */
c = SUBSTRING(
fields FROM :
i FOR 1 );


/* if comma, select values of current field */
    IF (
c = ',') THEN BEGIN


/* trim fields */
      FIELD = TRIM(:FIELD);


/* order by record in the sp (xml_generator) */
record = 0;


/* values of current field */
      FOR
EXECUTE STATEMENT
cast('select ' ||
        : FIELD ||
' from ' ||
:sql_right AS VARCHAR(1024))
      INTO :
xml_text DO
      BEGIN


/* formatting the xml line */

/* value    */
xml_text = '<' || :FIELD || '>' ||
              :
xml_text || ' || :FIELD || '>';
```

```
/* order by in xml_generator */
record =
record + 1;
          SUSPEND;

      END



/* clear field */
      FIELD = '';

    END

    ELSE



/* char by char... */ FIELD = FIELD ||
c;



/* next char */
i =
i + 1;

  END

END
```

```
    /*Download completed SQL script and

    explain step by step instructions:



    http://fernandomedeiros.com.br/blog/xml-generator-stored-procedure-
fi... */
```