Scalar UDFs - In C

Value Generating Functions

ADD2

Add two integers

MMA5

Return the min, max, and average of five integers
NULL values are treated the same as a zero

REVERSE

Reverse a string
Input string must be VARCHAR and can be up to 32 characters

Note: If you are looking for table-valued functions, see here.

Trafodion UDFs are "trusted" at this point in time. See "Security" below.

C source

```
cat > udf.c <<EOF
#include "sqludr.h"

SQLUDR_LIBFUNC SQLUDR_INT32 add2(SQLUDR_INT32 *in1,
                                 SQLUDR_INT32 *in2,
                                 SQLUDR_INT32 *out1,
                                 SQLUDR_INT16 *inInd1,
                                 SQLUDR_INT16 *inInd2,
                                 SQLUDR_INT16 *outInd1,
                                 SQLUDR_TRAIL_ARGS)
{
  if (calltype == SQLUDR_CALLTYPE_FINAL)
    return SQLUDR_SUCCESS;
  if (SQLUDR_GETNULLIND(inInd1) == SQLUDR_NULL ||
      SQLUDR_GETNULLIND(inInd2) == SQLUDR_NULL)
    SQLUDR_SETNULLIND(outInd1);
  else
    (*out1) = (*in1) + (*in2);
  return SQLUDR_SUCCESS;
}

SQLUDR_LIBFUNC SQLUDR_INT32 mma5(SQLUDR_INT32 *in1,
                                 SQLUDR_INT32 *in2,
                                 SQLUDR_INT32 *in3,
                                 SQLUDR_INT32 *in4,
                                 SQLUDR_INT32 *in5,
                                 SQLUDR_INT32 *out1,
                                 SQLUDR_INT32 *out2,
                                 SQLUDR_INT32 *out3,
                                 SQLUDR_INT16 *inInd1,
                                 SQLUDR_INT16 *inInd2,
                                 SQLUDR_INT16 *inInd3,
                                 SQLUDR_INT16 *inInd4,
                                 SQLUDR_INT16 *inInd5,
                                 SQLUDR_INT16 *outInd1,
                                 SQLUDR_INT16 *outInd2,
                                 SQLUDR_INT16 *outInd3,
                                 SQLUDR_TRAIL_ARGS)
{
  int sum = 0, min = 0, max = 0, avg = 0;
```

```
  int args[5];
  int i;
  if (calltype == SQLUDR_CALLTYPE_FINAL)
    return SQLUDR_SUCCESS;
  args[0] = (SQLUDR_GETNULLIND(inInd1) == SQLUDR_NULL ? 0 : *in1);
  args[1] = (SQLUDR_GETNULLIND(inInd2) == SQLUDR_NULL ? 0 : *in2);
  args[2] = (SQLUDR_GETNULLIND(inInd3) == SQLUDR_NULL ? 0 : *in3);
  args[3] = (SQLUDR_GETNULLIND(inInd4) == SQLUDR_NULL ? 0 : *in4);
  args[4] = (SQLUDR_GETNULLIND(inInd5) == SQLUDR_NULL ? 0 : *in5);
  sum = min = max = args[0];
  for (i = 1; i < 5; i++)
    {
      sum += args[i];
      min = (args[i] < min ? args[i] : min);
      max = (args[i] > max ? args[i] : max);
    }
  avg = sum / 5;
  *out1 = min;
  *out2 = max;
  *out3 = avg;
  return SQLUDR_SUCCESS;
}

/* Helper function to reverse a string */
static void reverseBytes(void *out, void *in, unsigned int numBytes)
{
  int i;
  char *pOut = (char *) out;
  char *pIn = (char *) in;
  for (i = 0; i < numBytes; i++)
    pOut[i] = pIn[numBytes - (i + 1)];
}

SQLUDR_LIBFUNC SQLUDR_INT32 reverse(SQLUDR_VC_STRUCT *in1,
                                    SQLUDR_VC_STRUCT *out1,
                                    SQLUDR_INT16 *inInd1,
                                    SQLUDR_INT16 *outInd1,
                                    SQLUDR_TRAIL_ARGS)
{
  if (calltype == SQLUDR_CALLTYPE_FINAL)
    return SQLUDR_SUCCESS;
  if (SQLUDR_GETNULLIND(inInd1) == SQLUDR_NULL)
    {
      SQLUDR_SETNULLIND(outInd1);
      return SQLUDR_SUCCESS;
    }
  reverseBytes(out1->data, in1->data, in1->length);
  out1->length = in1->length;
  return SQLUDR_SUCCESS;
}
EOF
```

Compile

```
gcc -g -Wall -I$TRAF_HOME/export/include/sql -shared -o udf.so udf.c
```

Register

Do this from the directory containing udf.so

```
UDFLIB="'$(pwd)/udf.so'"

sqlci <<EOF

create library myudfs file $UDFLIB;
-- ADD2
drop function add2;
create function add2(int,int) returns (add2 int)
  external name 'add2' library myudfs
  deterministic no sql no transaction required;

-- MMA5
drop function mma5;
create function mma5(int,int,int,int,int)
  returns (mma_min int, mma_max int, mma_avg int)
  external name 'mma5' library myudfs
  deterministic no sql no transaction required;

-- REVERSE
drop function reverse;
create function reverse(varchar(32)) returns (reverse varchar(32))
  external name 'reverse' library myudfs
  deterministic no sql no transaction required;

EOF
```

Invoke

```
sqlci <<EOF

-- CREATE A TABLE
drop table t;
create table t (a int primary key not null, b int, c int, d int, e int, f varchar(32));
insert into t values (1,2,3,4,5,'abc'), (6,7,8,9,10,'def'), (11,12,13,14,15,'ghi');

-- ADD2
select a, b, add2(a,b) from t;

-- MMA5
select mma5(a,b,c,d,e) from t;

-- REVERSE
select * from t where reverse(f) > 'g';

EOF
```

SHOWDDL

```
sqlci <<EOF
showddl function add2;
showddl function mma5;
showddl function reverse;
EOF
```

Security

Trafodion UDFs are "trusted" at this point in time. A "trusted" UDF has full access to any resources of the Trafodion engine. That means that a malicious UDF writer could compromise the privacy and consistency of the data. Therefore, only trusted users can be allowed to write UDFs. See the "Security Considerations" section in the TMUDF wiki article for how to grant users the privileges needed to create UDFs.