

```

// OrderGen.cpp - ORDER DATABASE GENERATOR
//
// MODULE INDEX
// NAME                CONTENTS
// main                Main line
//
// MAINTENANCE HISTORY
// DATE                PROGRAMMER AND DETAILS
// 24-09-14    MPF    Original
//
//-----

#include <cstdlib>           // C-style standard library
#include <iostream>         // C++ input/output streams
using namespace std;       // Expand the standard namespace
exec sql include sqlca;    // Include SQL communications area

//-----

// DEFINITIONS

static const long    MAX_FLAVOURS = 50;    // Maximum number of flavours

//-----

// MAIN LINE

int
main ()
{
    long        stockCnt;    // Stock count
    long        orderCnt;    // Order count
    long        flavourCnt;  // Flavour count

    exec sql begin declare section;
        long        batchNo;    // Batch number
        long        flavourNo;  // Flavour number
        long        custNo;     // Customer number
        long        orderNo;    // Order number
        long        qty;        // Quantity
        short       delivered;  // Delivered flag
    exec sql end declare section;

    // Connect to the database

    exec sql connect to orderdb;

    // Drop tables

    exec sql whenever sqlerror continue;
    exec sql drop table custOrder;
    exec sql drop table orderItem;
    exec sql drop table stock;
    exec sql commit work;

    // Jump to DbError whenever an SQL error occurs

    exec sql whenever sqlerror goto DbError;

```

```

// Create the database tables

exec sql create table custOrder (
    orderNo          integer not null,
    orderCustNo     integer not null,
    delivered       smallint not null
);
exec sql create unique index orderNoInd on
    custOrder (orderNo);

exec sql create table orderItem (
    itemOrderNo     integer not null,
    itemFlavourNo   integer not null,
    itemQty         integer not null
);
exec sql create unique index orderNoFlavourNoInd on
    orderItem (itemOrderNo, itemFlavourNo);

exec sql create table stock (
    stockBatchNo    integer not null,
    stockFlavourNo  integer not null,
    stockQty        integer not null
);
exec sql create unique index batchNoInd on
    stock (stockBatchNo);
exec sql create index flavourNoInd on
    stock (stockFlavourNo);

exec sql commit work;

// Initialise the random number generator

srand48 (203607L);

// Generate the stock records

batchNo = 0;
stockCnt = lrand48() % 1000 + 100;
cout << "Generating stocks " << stockCnt << '\n';
while (stockCnt-- > 0) {
    batchNo += lrand48() % 10 + 1;
    flavourNo = lrand48() % MAX_FLAVOURS;
    qty = lrand48() % 100 + 1;

    exec sql insert into stock (
        stockBatchNo, stockFlavourNo, stockQty
    ) values (
        :batchNo, :flavourNo, :qty
    );
}

// Generate the customer orders

orderNo = 0;
orderCnt = lrand48() % 100 + 10;
cout << "Generating orders " << orderCnt << '\n';
while (orderCnt-- > 0) {

```

```

orderNo += lrand48() % 10 + 1;
custNo = lrand48() % 100;
delivered = (lrand48() % 2 == 0);

exec sql insert into custOrder (
    orderNo, orderCustNo, delivered
) values (
    :orderNo, :custNo, :delivered
);

flavourCnt = lrand48() % 10 + 1;
flavourNo = lrand48() % (MAX_FLAVOURS - flavourCnt);
while (flavourCnt-- > 0) {
    qty = lrand48() % 1000 + 1;
    flavourNo++;
    exec sql insert into orderItem (
        itemOrderNo, itemFlavourNo, itemQty
    ) values (
        :orderNo, :flavourNo, :qty
    );
}
}

// Commit the changes
exec sql commit work;

// And that's all
return 0;

// Process database errors
DbError:
cerr << "Error: SQLCODE=" << SQLCODE << endl;
return 1;
}

```