

```

// OrderGen.cpp - ORDER DATABASE GENERATOR
//
// MODULE INDEX
// NAME                CONTENTS
// main                Main line
//
// MAINTENANCE HISTORY
// DATE              PROGRAMMER AND DETAILS
// 03-10-17         SHT      Original
// 20-10-17         YGL      Corrected the above
// 27-10-17         SHT      Corrected datatype for resRestName
//                  Increased number of order from 100 to 120
//
//-----

#include <cstring>           // C-style string manipulation functions
#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

//-----

static const int RES_CNT = 12;           // Number of restaurants
static const int ORD_CNT = 120;        // Number of orders

// RESTAURANT NAME

static const char *RES_NAM[RES_CNT] = {
    "FOOD ZEBRA",
    "WALKING MAN",
    "DELICIOUS HUNTER",
    "EG CURRY HOUSE",
    "SHIN RESTAURANT",
    "PAPAPOOR",
    "KOKWONG",
    "SUBMARINE",
    "ORDER PROCESSOR",
    "YUMMY STORE",
    "FAULTY KITCHEN",
    "EPIC MAKANAN"
};

//-----

int
main()
{
    int    i;           // General purpose index

    exec sql begin declare section;
        long    restId;           // Restaurant ID
        char    restName[21];     // Restaurant name
        long    orderNo;         // Order number

```

```

    long    orderValue;        // Order value
    double  orderTim;         // Order time
    double  orderRspTim;     // Order response time
    long    orderVerified;    // Order verified flag
exec sql end declare section;

// Connect to database

exec sql connect to egpc;

// Drop tables

exec sql whenever sqlerror continue;
exec sql drop table ord;
exec sql drop table ordRsp;
exec sql drop table res;
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 ord (
    ordOrderNo    integer not null,
    ordTim        double precision not null,
    ordValue      integer not null,
    ordVerified   smallint not null
);
exec sql create unique index ordOrderNoInd on
    ord (ordOrderNo);

exec sql create table ordRsp (
    ordRspOrderNo integer not null,
    ordRspRestId  integer not null,
    ordRspTim     double precision,
    ordRspAccept  smallint
);
exec sql create index ordRspOrderNoInd on
    ordRsp (ordRspOrderNo, ordRspRestId, ordRspTim);

exec sql create table res (
    resRestId     integer not null,
    resRestName   varchar(21) not null
);
exec sql create unique index resRestIdInd on
    res (resRestId);

exec sql commit work;

// Initialize random number generator

srand48 (12176886L);

```

```

// Generate restaurant records

for (i = 0; i < RES_CNT; i++) {
    restId = i + 1;
    strcpy (restName, RES_NAM[i]);

    exec sql insert into res (
        resRestId, resRestName
    ) values (
        :restId, :restName
    );
}

// Generate order records

orderTim = 1510358400;           // Unix time at 11-11-2017 12:00AM
orderNo = lrand48() % 1000;     // Random order number

for (i = 0; i < ORD_CNT; i++) {
    // Generate order records

    orderNo = orderNo + 1;

    orderValue = lrand48() % 50000 + 1;
    orderTim = orderTim + (lrand48() % 1000);
    orderVerified = (lrand48() % 100 < 80) ? 1 : 0;

    exec sql insert into ord (
        ordOrderNo, ordTim, ordValue, ordVerified
    ) values (
        :orderNo, :orderTim, :orderValue, :orderVerified
    );

    // Generate order response records

    orderRspTim = orderTim;

    for (;;) {
        restId = lrand48() % RES_CNT + 1;
        orderRspTim = orderRspTim + (lrand48() % 100);

        // Generate accept record

        exec sql insert into ordRsp (
            ordRspOrderNo, ordRspRestId,
            ordRspTim, ordRspAccept
        ) values (
            :orderNo, :restId,
            :orderRspTim, 1
        );

        // Decide whether order to be declined by restaurant

```

```

        if (lrand48() % 100 < 80)
            break;

        // Generate decline record

        orderRspTim = orderRspTim + (lrand48() % 1000);

        exec sql insert into ordRsp (
            ordRspOrderNo, ordRspRestId,
            ordRspTim, ordRspAccept
        ) values (
            :orderNo, :restId,
            :orderRspTim, 0
        );
    }
}

// Commit the changes

exec sql commit work;

// And that's all

return 0;

// Process database error

```

```

DbError:
    cerr << "Error: SQLCODE=" << SQLCODE << endl;
    return 1;
}

```