

```

// RouteGen.cpp - GENERATE ROUTE DATABASE
//
// MAINTENANCE HISTORY
// DATE      PROGRAMMER AND DETAILS
// 29-09-16 MPF Original
//
//-----

#include <vector>           // C++ vector declaration
#include <set>              // C++ set declaration
#include <map>              // C++ map declaration
#include <iostream>        // C++ I/O stream declaration
using namespace std;
exec sql include sqlca;    // Include SQL communications area

//-----

// MAIN LINE

int main (
    int         argc,      // Argument count
    char        *argv[])  // Argument value
{
    long        numOfStats; // Number of stations
    bool        used[50];   // Used station identifier
    long        numOfLinks; // Number of links
    long        link;       // Link
    long        numOfTrains; // Number of trains
    vector<long> tracks[50]; // Valid tracks
    long        numOfStops; // Number of stops
    long        selectedStat; // Selected station

    exec sql begin declare section;
        long        trackId;    // Track identifier
        long        genStatId;  // Station identifier
        long        genNextStatId; // Next station identifier
        long        genDuration; // Duration
        long        genTrainId; // Train identifier
        long        startTime;  // Start time
        long        genSeqNo;   // Sequence number
    exec sql end declare section;

    exec sql connect to egpc;

    exec sql create table train(
        trainId          integer not null,
        trainFirstStatId integer not null,
        trainStartTime   integer not null
    );
    exec sql create unique index trainIdInd on train(trainId);

    exec sql create table route(
        routeTrainId     integer not null,
        routeSeqNo       integer not null,
        routeNextStatId integer not null,
        routeNextWaitTime integer not null
    );

```

```
exec sql create unique index routeTrainIdSeqNoInd on route(routeTrainId, routeSeqNo);

exec sql create table track(
    trackId            integer not null,
    trackFromStatId   integer not null,
    trackToStatId     integer not null,
    trackDuration     integer not null
);
exec sql create unique index trackIdInd on track(trackId);

exec sql create unique index trackFromToStatInd on track(trackFromStatId, trackToStatId
);

exec sql whenever sqlerror goto db_error;

srand (1234567L);

// Generate tracks

trackId = 0;
numOfStats = rand() % 40 + 10;
for (genStatId = 0; genStatId < numOfStats; genStatId++) {
    memset (used, 0, sizeof(used));
    used[genStatId] = 1;

    numOfLinks = rand() % (numOfStats / 2 - 2) + 2;
    for (link = 0; link < numOfLinks; link++) {
        do {
            genNextStatId = rand() % numOfStats;
        } while (used[genNextStatId]);
        used[genNextStatId] = 1;

        genDuration = rand() % 900 + 30;
        exec sql insert into track (
            trackId, trackFromStatId, trackToStatId,
            trackDuration
        ) values (
            :trackId, :genStatId, :genNextStatId,
            :genDuration
        );
        trackId++;
        tracks[genStatId].push_back(genNextStatId);
    }
}

// Generate trains

numOfTrains = rand() % (numOfStats * 3) + numOfStats;
for (genTrainId = 0; genTrainId < numOfTrains; genTrainId++) {
    genStatId = rand() % numOfStats;
    startTime = rand() % (6*60*60) + 30;
    exec sql insert into train (
        trainId, trainFirstStatId, trainStartTime
    ) values (
        :genTrainId, :genStatId, :startTime
    );
};
```

```
// Generate routes for the train

genSeqNo = 0;
numOfStops = rand() % 100 + 50;
for (genSeqNo = 0; genSeqNo < numOfStops; genSeqNo++) {
    genDuration = rand() % 60 + 15;
    selectedStat = rand() % tracks[genStatId].size();
    genNextStatId = tracks[genStatId][selectedStat];

    exec sql insert into route (
        routeTrainId, routeSeqNo, routeNextStatId,
        routeNextWaitTime
    ) values (
        :genTrainId, :genSeqNo, :genNextStatId,
        :genDuration
    );

    genStatId = genNextStatId;
}

exec sql commit work;

return 0;

db_error:
cerr << "Error: SQLCODE=" << SQLCODE << '\n';
return 1;
}
```