Задача: переконвертировать данные из xml файла в таблицы БД PostgreSQL.

1. Исходные данные: xml-файл с описанием трехмерной сцены, пример:

<mesh>

<submeshes>

<submesh material="tree.png">

<faces count="334">

<face v1="0" v2="1" v3="2" />

…

<face v1="211" v2="173" v3="228" />

</faces>

<geometry vertexcount="229">

<vertexbuffer>

…

<vertex>

<position x="49.6933" y="5.43665" z="-0.784625" />

</vertex>

…

</vertexbuffer>

</geometry>

</submesh>

…

</submeshes>

<nodes>

<node name="tree#1" id="176580096">

<position x="13.7081" y="8.14336" z="-33.9854" />

<rotation qw="1" qx="0" qy="0" qz="0" />

<entity name="tree#1" meshFile="tree.05.mesh" astShadows="false">

</node>

…

</nodes>

</mesh>

Теги <mesh> </mesh> определяют начало и конец данных.

Теги <submeshes> </submeshes> определяют начало и конец секции содержащей данные об объектах сцены.

Теги <nodes> </nodes> определяют начало и конец секции содержащей данные о точечных объектах сцены.

Теги <submesh material="materialname"></submesh> описывают объект из материала с названием "materialname".

Теги <faces count="334"></faces> задают начало и конец блока перечисления фацетов(полигонов, они всегда треугольные) и их вершин. Параметр count задает общее количество фацетов.

Тег <face v1="0" v2="1" v3="2" /> задает номера вершин одного фацета.

Теги <geometry vertexcount="229"> <vertexbuffer> </vertexbuffer> </geometry> Задают начало и конец блока перечисления вершин фацетов и их координат в метрических единицах. Параметр vertexcount задает общее количество вершин.

Теги <vertex><position x="49.6933" y="5.43665" z="-0.784625"/> </vertex> задают координаты вершины, номер вершины совпадает с порядковым номером данной конструкции в блоке между тегами <vertexbuffer> </vertexbuffer>.

Теги <node name="tree#1" id="176580096"></node> - блок описывающий точечный объект.

Тег <position x="13.7081" y="8.14336" z="-33.9854" /> - описывает координаты точечного объекта в метрических единицах.

Тег <rotation qw="1" qx="0" qy="0" qz="0" /> - описывает ориентацию точечного объекта в метрических единицах.

Некоторые блоки, в исходном файле могут отсутствовать, например секции <nodes> </nodes> может не быть.

2. Из данного файла необходимо извлечь информацию о всех полигонах трехмерной сцене и записать в таблицу с именем faces в БД PostgreSQL в следующем формате (список полей):

id\_file; X1; Y1; Z1; X2; Y2; Z2; X3; Y3; Z3; id\_mat;

где Xi, Yi, Zi – координаты i-той вершины фацета, где i=1,2,3;

id\_mat – тип материала = "materialname" из тега <submesh material="materialname">

Для координат тип float, id\_mat – текстовый

id\_file - целый числовой, уникальный для каждой новой выгрузки из файла в таблицу, например, автоинкрементный. 1 для первого файла, 2 для второго и т.д.

3. Выгрузить информацию об объектах в таблицу с именем objects в БД PostgreSQL в следующем формате (список полей):

id\_file; X; Y; Z; QW; QX; QY; QZ; id\_obj

X; Y; Z; координаты точечного объекта из <position x="13.7081" y="8.14336" z="-33.9854" />

QW; QX; QY; QZ; ориентация точечного объекта из <rotation qw="1" qx="0" qy="0" qz="0" />

id\_obj – название объекта, атрибут «name» из <node name="tree#1" id="176580096">

Примеры xml файлов result1.xml, result2.xml, result3.xml во вложении к письму.

4.Требования к программе.

Консольное приложение, Win32. Любой компилируемый язык, в идеале С/C++, но не критично. Входные параметры (имя xml файла, параметры подключения к БД) извлекаются из параметров командной строки. Загрузку параметров подключения к БД можно реализовать из настроечного файла, формат по желанию.

Использование сторонних библиотек приветствуется, например для парсинга XML и работы и БД. Версия PostgreSQL 9.1.

Для первоначального создания таблиц в БД должен быть sql-скрипт.