7. Exercise: B-Rep Geometrical Data

1. B-Rep Construction Methods

Explain the main construction methods
- basic topology,
- CSG, and
- sweeping
of the B-Rep model. What kind of data structures are required to represent these methods?

2. Winged Edge Data Structure

As introduced in the lecture the Winged Edge concept [Source: Winged Edge at wikipedia.org] can be implemented (e.g. as follows:

```java
class WE_Edge {
    WE_Vertex vert1, vert2;
    WE_Face aFace, bFace;
    WE_Edge aPrev, aNext, bPrev, bNext;
    // clockwise ordering
    WE_EdgeDataObject data;
}
class WE_Vertex {
    List<WE_Edge> edges;
    WE_VertexDataObject data;
}
class WE_Face {
    List<WE_Edge> edges;
    WE_FaceDataObject data;
}
```

How can the same data structure be implemented using a relational database system? How many tables would be required? Note that the List-valued attributes need to be normalized!
3. Free-form Curves

B-Splines are sets of polynomials described by control points and knots to described curves. Given the schema and data (excerpt) example below, discuss how according geometries can be described in STEP!

```plaintext
(* SCHEMA building design_schema; *)

ENTITY edge_curve
  SUBTYPE OF ( edge, geometric_representation_item );
  edge_geometry : curve;
  same_sense : BOOLEAN;
END_ENTITY; -- edge_curve

ENTITY b_spline_curve
  SUPERTYPE OF (ONEOF (uniform_curve,b_spline_curve_with_knots,
                        quasi_uniform_curve,bezier_curve) ANDOR
                        rational_b_spline_curve)
  SUBTYPE OF (bounded_curve);
  degree : INTEGER;
  control_points_list : LIST [2:?] OF cartesian_point;
  curve_form : b_spline_curve_form;
  closed_curve : LOGICAL;
  self_intersect : LOGICAL;
END_ENTITY; -- b_spline_curve

ENTITY b_spline_curve_with_knots
  SUBTYPE OF (b_spline_curve);
  knot_multiplicities : LIST [2:?] OF INTEGER;
  knots : LIST [2:?] OF parameter_value;
  knot_spec : knot_type;
END_ENTITY; -- b_spline_curve_with_knots
```

```plaintext
#1518 = EDGE_CURVE( '', #1790, #1800, #3352, .T. );
#1790 = VERTEX_POINT( '', #3656 );
#1800 = VERTEX_POINT( '', #3667 );
#3352 = B_SPLINE_CURVE_WITH_KNOTS( '', 3, ( #5569, #5570,
                      #5571, #5572, #5573, #5574, #5575, #5576, #5577, #5578 ),
                      .UNSPECIFIED., .F., .F., ( 4, 3, 3, 4 ),
                      ( 1.78358532591919, 3.30601186182344, 4.95901779273516,
                      5.57149748714495 ), .UNSPECIFIED. );
#5569 = CARTESIAN_POINT( '', ( 0.712774915050965,
                         -1.91744986484148, 0.541338582677161 ) );
... #5578 = CARTESIAN_POINT( '', ( 0.826771653543307,
                      -1.87007874015748, 0.623352874768626 ) );
```