

Canadian Board of Examiners for Professional Surveyors
Core Syllabus Item
E 1: SPATIAL DATABASES AND LAND INFORMATION SYSTEMS

Study Guide:

This study guide gives a complete example of the development of a spatial database from the feasibility study through the operation and maintenance of this spatial database. In this example, we use a Waterfall method but all other software development methods or database design methods could also be used in the E1 exam. For each database modeling level, more than one schema example is made in order to show solutions using different formalisms and approaches. Other formalisms than those used in this example can be used in an E1 exam. The vocabulary found in the standard OMG method called "[*Model-driven Architecture \(MDA\)*](#)" is used here to identify each level of modeling. According to this vocabulary, the CIM is the Computation Independent Model, the PIM is the Platform Independent Model and the PSM is the Platform Specific Model. These models also respectively correspond to the conceptual, logical and physical data models found in other methods.

Throughout this study guide applicable sections or chapters of the essential and supplementary references listed for the syllabus item are listed.

Planning Phase:

- Global understanding of business functions.
- Initial assessment of user requirements. Who is going to use the database and to do what?
- Feasibility study to implement the database. How much it should be cost? How many human resources are needed? What are the risks of failure?
- Choose a design and development method.

References on systems and database life cycle:

Essential a - chapter 3-4.2 in Yeung and Hall, [2007]

Essential d - Observations on Balancing Discipline and Agility

Essential e - Comparing Software Development Life Cycles, Hurst, J. 2007.

<http://www.giac.org/resources/whitepaper/application/257.php>

References on MDA:

Essential a - chapter 8-5.4 in Yeung and Hall, [2007]

Supplementary a: OMG Model Driven Architecture

References on organizational and institutional issues relating to Spatial Database Project Management:

Essential a - chapters 7, 8 and 9 in Yeung and Hall, [2007]

References on UML diagrams:

Essential c - Introduction to UML

Analysis Phase:

- Systematic and more detailed assessment of user requirements.
- Evaluation of existing business practices and operations. UML Use Cases are good diagrams to illustrate these practices and operations as well as some user requirements.

- Evaluation of existing data sources after having made their inventory and collected their metadata. These data sources are often used to load the spatial database, usually after some cleaning and transformation. The inventory of object classes and attributes of the datasets can be used as an input for the CIM. The best sources must be chosen regarding the user requirements, cost and legal issues.

References on metadata and standard:

Essential a - chapter 5 in Yeung and Hall, [2007]

References on legal issues of spatial data:

Essential a - chapter 7 in Yeung and Hall, [2007]

Design Phase:

- Design a CIM according to users' needs and that is technology-independent. This usually uses a modeling language such as UML (figure 1), extended UML for spatial data (figure 2) or E/R (figure 3). These models are not the only three possible solutions; other alternatives can also be used for the I-5 exam. A software package called CASE tool (Computer-Assisted Software Engineering) is often used to support the design phase. Typical CASE tools include a dictionary and provide automatic code generation as well as reverse engineering (i.e. generating models from code).
 - In figure 1, UML is used and the geometry is defined according to GM_object as defined in the ISO-TC211 19107 Spatial schema standard and in the Open GIS Reference Model.
 - In figure 2, Extended UML for Spatial reference is used. Spatial pictograms are used to show the geometry of the objects classes. Junction has 0D geometry, segment and structure have 1D geometries, building has a 0D, 1D or 2D geometry depending of the size of the building (as usually explained in the data dictionary) and each municipality is represented by an aggregate of one or many polygons. Perceptory CASE tool is used in this example.
 - In figure 3, the geometry is defined using a topological structure made of points, lines and polygons. Oracle Designer CASE tool is used.

References for UML:

Essential b and Supplementary c to e.

Essential a - chapter 3-4.5 in Yeung and Hall, [2007]

References on extended UML for spatial data:

Supplementary b: Using Spatial PVL for spatial database modeling

References on Entity-Relationship:

Essential a - chapter 3-3.1 in Yeung and Hall, [2007]

References on CASE tool:

Essential a - chapter 3-4.3 in Yeung and Hall, [2007]

References on OGC data types:

Essential f: Simple feature access - Part 1: Common architecture

Supplementary f: OpenGIS Reference Model (ORM)

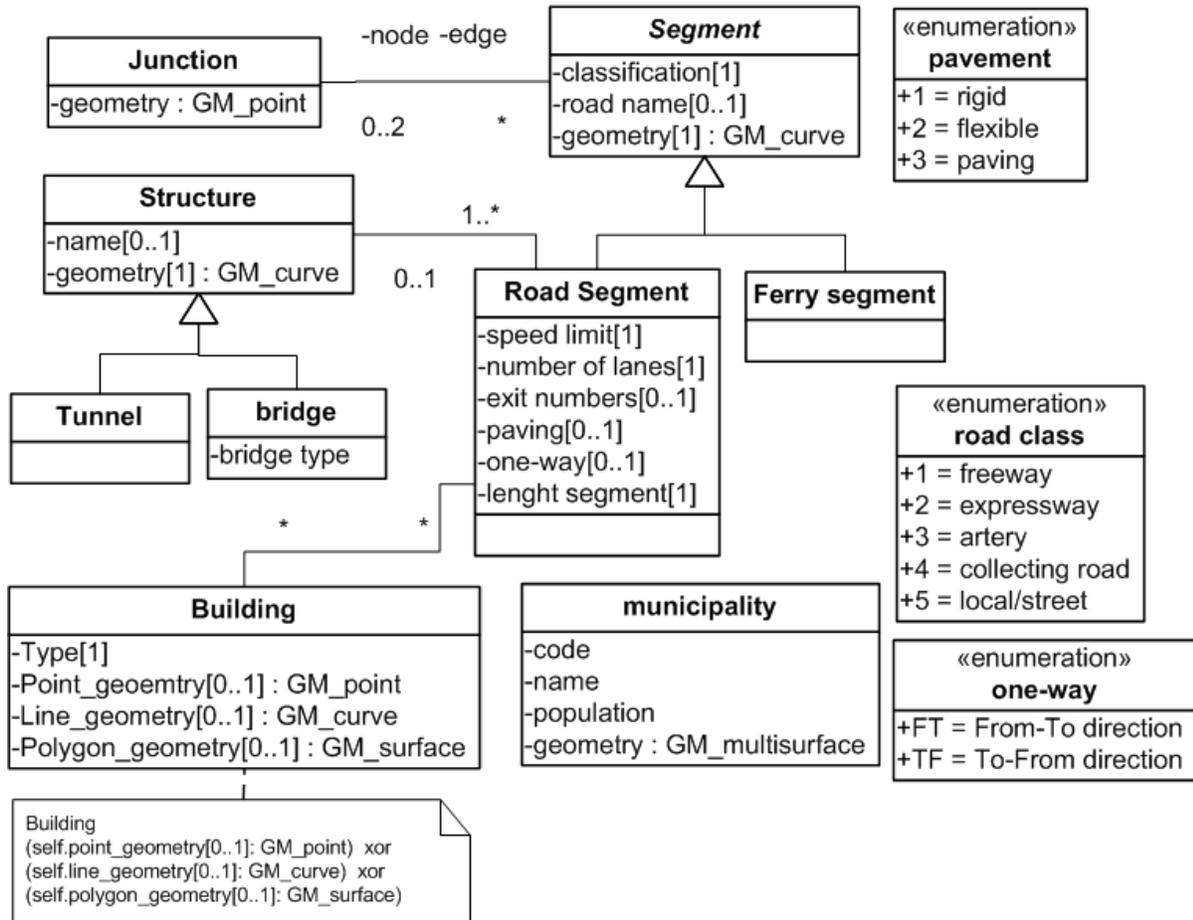


Figure 1: CIM designed with UML and ISO TC/211 19107 GM_object.

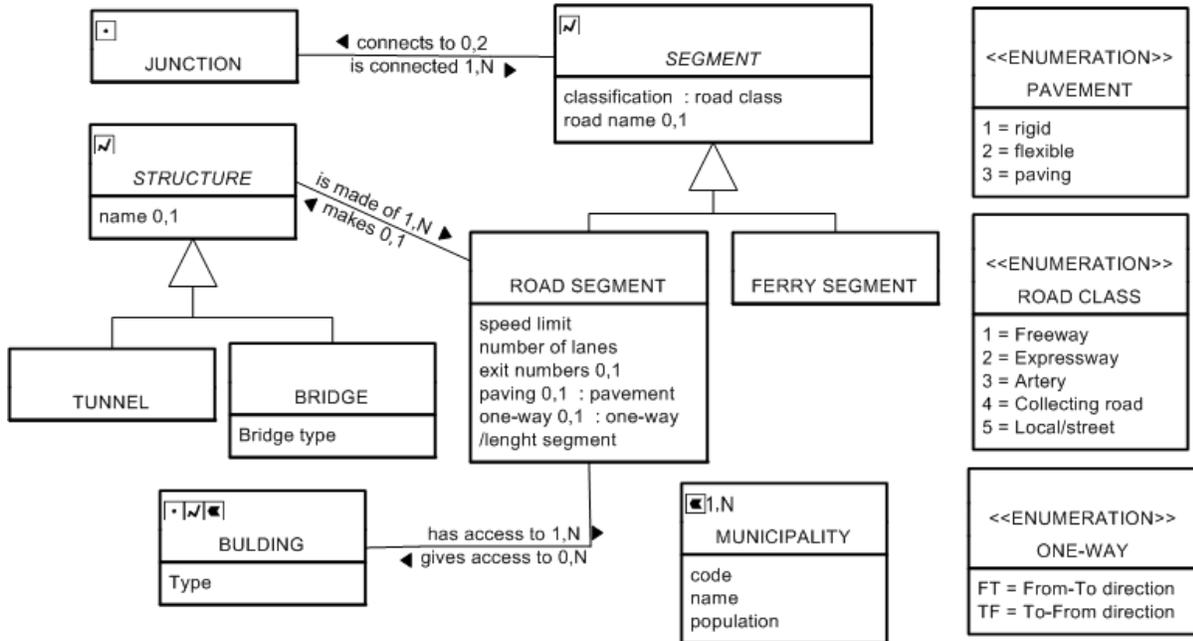


Figure 2: CIM designed with UML language extended for spatial reference.

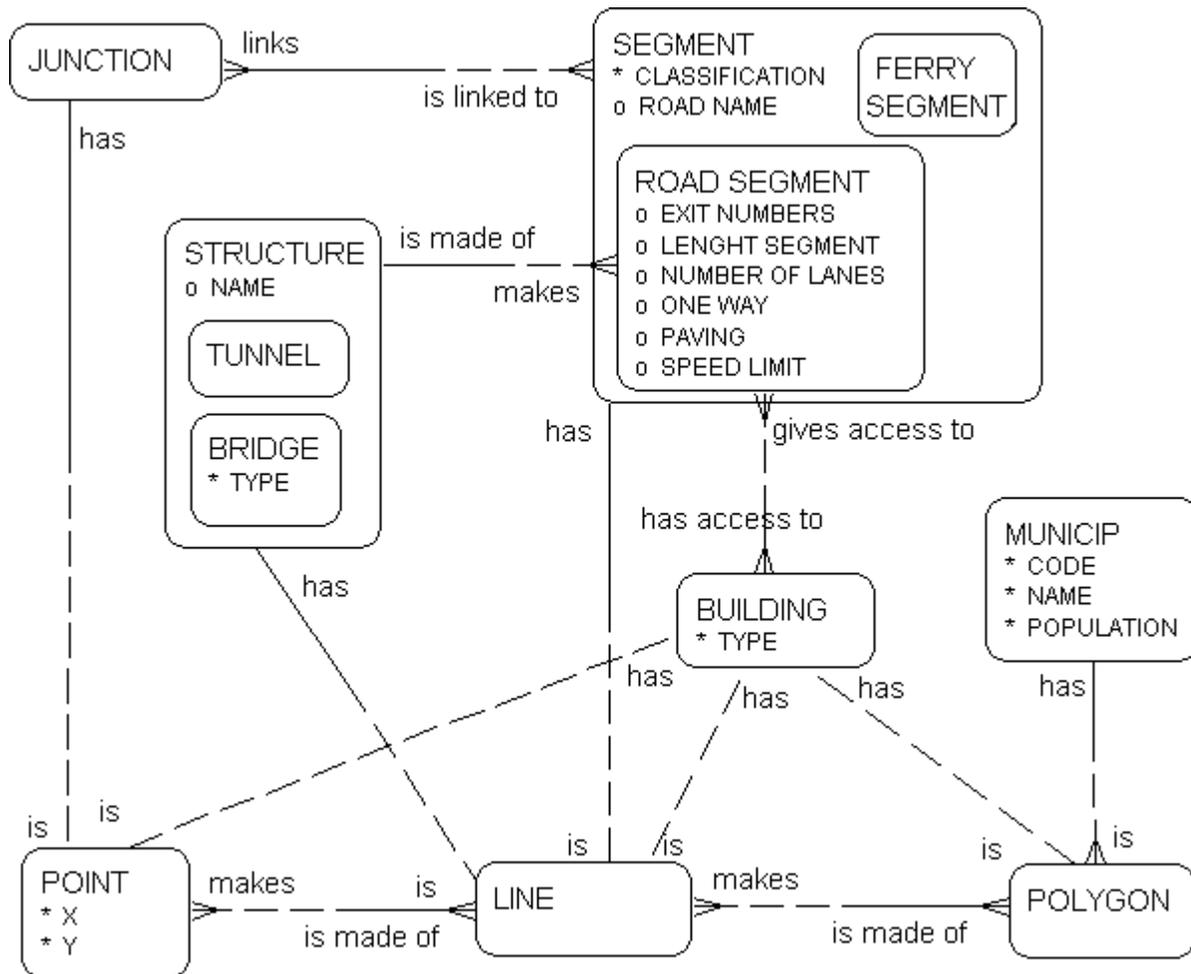


Figure 3: CIM designed with the Entity Relationship Diagrammer of Oracle Designer.

- A CIM is most of the time normalized, i.e. data redundancy is minimized or eliminated. The first three normal forms (NF) are usually respected.

References on normalization:

Essential a - chapter 3-3.2 in Yeung and Hall, [2007]

- Spatial integrity constraints should also be defined to guarantee the exactness of spatial relations between objects. The next figure illustrates such constraints for Junction using OGC operator. A junction must be connected to at most one road segment or one ferry segment and can be connected at many ones.

	Relation			Cardinality
Junction <input type="checkbox"/>	Touches ¹	Road Segment	<input checked="" type="checkbox"/>	0,N*
Junction <input type="checkbox"/>	Touches ¹	Ferry Segment	<input checked="" type="checkbox"/>	0,N*
Junction <input type="checkbox"/>	Disjoint of	Junction	<input type="checkbox"/>	

¹ must be touched at least by a Ferry or a Road segment

Table 1: Example of spatial integrity constraints using OGC operator.

¹ OpenGIS® Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture.

References on spatial integrity constraint:

Essential a -chapter 4-4.2.3 in Yeung and Hall, [2007]

Supplementary g: Natural Resources Canada, 1996. National Topographic Database Data dictionary: http://ftp2.cits.rncan.gc.ca/pub/bndt/doc/dictntd3_en.pdf. See Relation part of each entity for examples. Now, OGC operators are used instead of *Connect* and *Share* operators.

- Processes are also be modeled during this phase using UML activity diagrams and sequence diagram, or data flow diagrams (DFD). For example, design a process to calculate the number of buildings having access to a road segment.

Reference on activity diagram and sequence diagram:

Essential a - chapter 4-4.2.3 in Yeung and Hall, [2007]

- Choose the appropriate Spatial Database System Architecture (e.g. centralized, distributed, client/server, web-based).

Reference on Spatial Database System Architecture:

Essential a - chapter 2-4, chapter 6, chapter 10, chapter 11 in Yeung and Hall, [2007])

- Compare transactional and multidimensional families of database technology to select the most appropriate one (see table 2). Based on this comparison, you and your client decide to implement his database using transactional technology, more specify relational.

Transactional	Multidimensional
Build for transactions	Build for analysis
Prioritize security and data integrity	Prioritize analysis and exploration
Minimize data redundancy	Minimize the query response time
Detailed data	Detailed and abstract data
Integrated for applications	Integrated for the enterprise
Continually updated	Never updated, only new data added
Actual data	Actual and archived data
Legacy data	Imported data from legacy data
Normalized structure	Denormalized structure

Table 2: Comparison between transactional and multidimensional system.

Reference on differences between transactional and multidimensional:

Essential a -chapter 6-4.1.5 in Yeung and Hall, [2007]

Build Phase:

- Translate the CIM into a relational PIM, that is into a generic platform model for relational database management systems in our example. Here, two PIMs are used to illustrate this example, each one implementing the database generalization relationship differently and

using different formalisms. The examples showed in Figure 4 and in Figure 5 can be used with any relational database because generic data types are used. Integrity constraints are also defined in this example, for example Entity integrity (Primary key), Referential Integrity (Foreign key), Required data (bold fields), Domain constraints (data type and Check constraint) and Enterprise Constraint. Many steps are required to properly translate the CIM into a PIM.

References on integrity constraints:

Essential a - chapter 2-3.2 in Yeung and Hall, [2007]

References on steps to follow to translate a CIM into a PIM:

Essential b

Essential a - chapter 3-2.3, 3.2 in Yeung and Hall, [2007]

References on UML and PIM level:

Supplementary d: The UML and Data Modeling, Rational Software Whitepaper:

<http://www.uml.org.cn/object/Tp180.pdf>

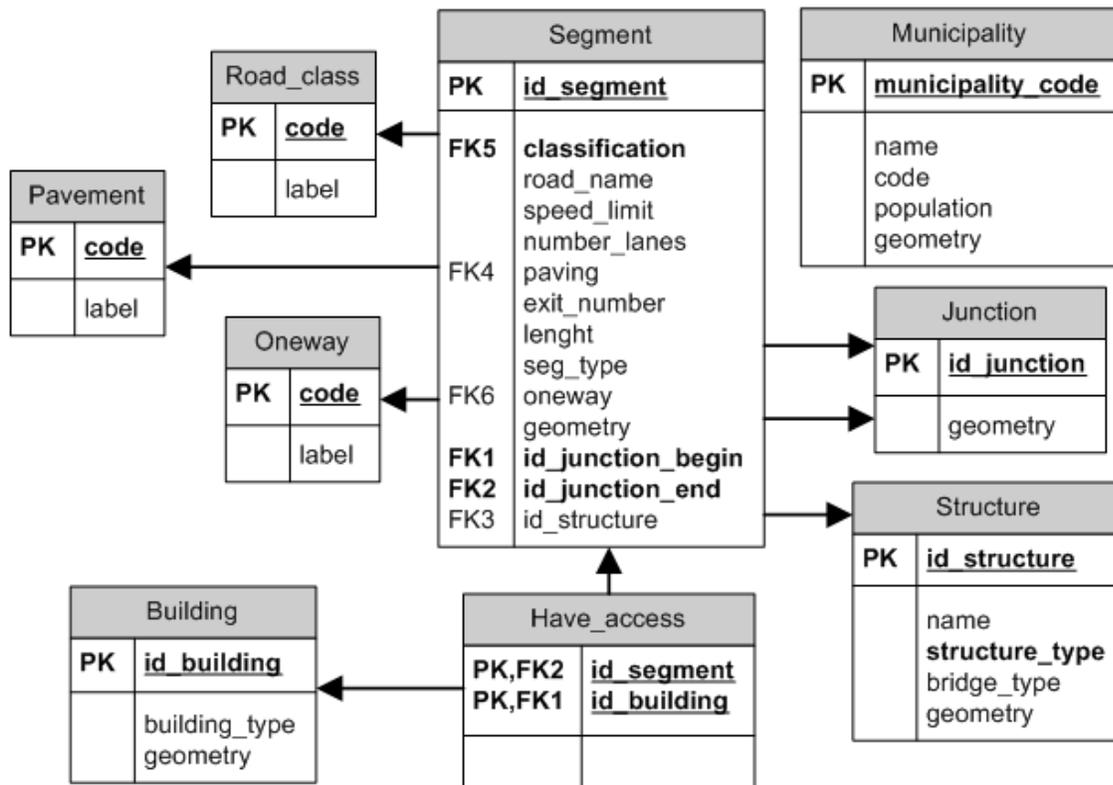


Figure 4: PIM Relational Data Model using Microsoft Visio Entity/Relationship diagram template. The data types are defined in the dictionary.

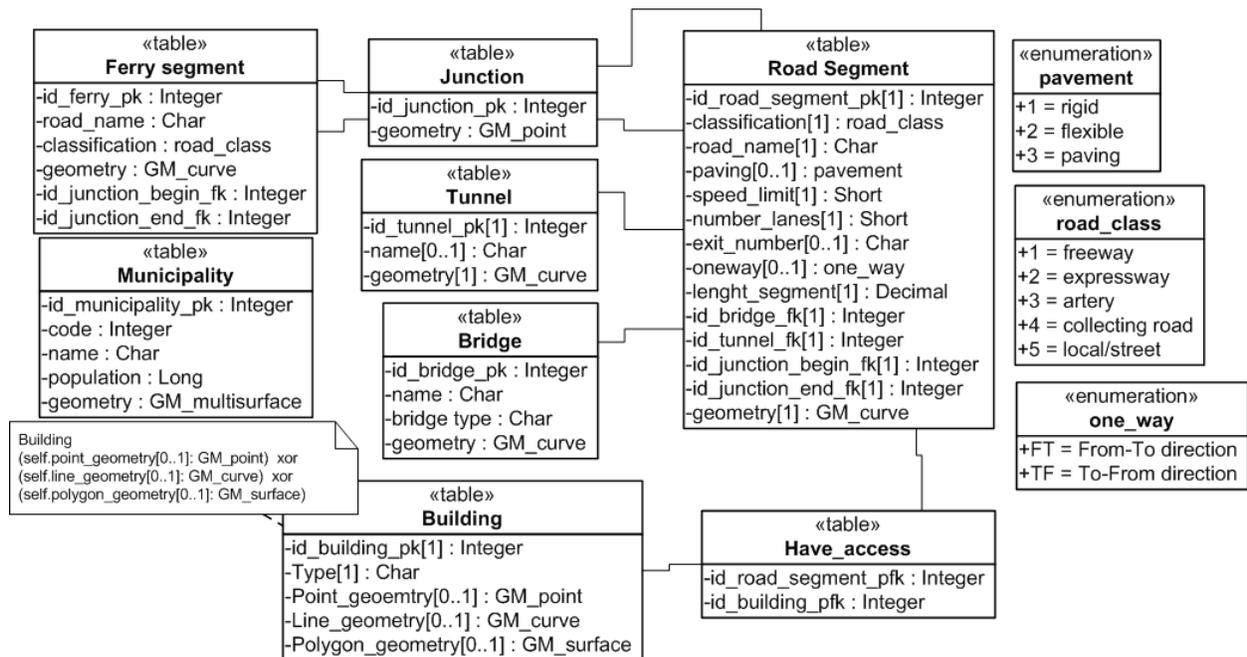


Figure 5: Another PIM example using UML notation (and Table stereotype) for the same CIM. The generalization relationship is implemented differently.

- Choose the platform on which the database will be implemented (e.g. Spatial Oracle, ESRI Geodatabase) and choose the appropriate spatial data structure to store geometric data (e.g. topological and non-topological data structure).

References on Spatial Data Structure:

Essential a - chapter 4 in Yeung and Hall, [2007]

- Translate the PIM into a PSM. The specific platform chosen in our example is Spatial Oracle and non-topological data structure. Data types supported by the selected platform are defined for each attributes and a DDL code specific to this platform can be generated. Here is the Spatial Oracle PSM associated with the PIM illustrated in Figure 4 (for the exam, be able to do reverse engineering but not to write this kind of SQL code).

```

CREATE TABLE Building
(
  Building_ID          Number(7) CONSTRAINT pk_building PRIMARY KEY,
  Building_type       Varchar2(30),
  Geometry             SDO_GEOMETRY
);
  
```

```

CREATE TABLE junction
(
  junction_ID         Number(7) CONSTRAINT pk_jonction PRIMARY KEY,
  Geometry            SDO_GEOMETRY
);
  
```

```

CREATE TABLE municipality
(
Municipality_code    Number(7) CONSTRAINT pk_mun PRIMARY KEY,
name                  Varchar2(6),
population            Number(7),
Geometry              SDO_GEOMETRY
);

```

```

CREATE TABLE segment
(
ID_segment           Number(7) CONSTRAINT pk_segment PRIMARY KEY,
classification       Number(1),
road_name            Varchar2(30),
speed_limit          Number(6),
number_lanes         Number(2),
exit_number          Varchar2(6),
paving               Number(1),
oneway               Char(2),
lenght               Float(38),
Seg_type             Char(40)    NOT NULL,
junction_begin_FK    Number(7),
junction_end_FK      Number(7),
structstruct_FK      Number(7)   NOT NULL,
Geometry             SDO_GEOMETRY,
CONSTRAINT fk_jonction1 FOREIGN KEY (junction_begin_FK)
REFERENCES junction (junction_ID),
CONSTRAINT fk_jonction2 FOREIGN KEY (junction_end_FK)
REFERENCES junction (junction_ID),
CONSTRAINT fk_structure FOREIGN KEY (structstruct_FK)
REFERENCES structure (ID_structure),
CONSTRAINT fk_oneway FOREIGN KEY (oneway)
REFERENCES oneway (code),
CONSTRAINT fk_pavement FOREIGN KEY (paving)
REFERENCES pavement (code),
CONSTRAINT fk_road_class FOREIGN KEY (classification)

```

REFERENCES road_class (code)

);

CREATE TABLE structure

(

ID_structure Number(7) CONSTRAINT pk_structure PRIMARY KEY,

name Varchar2(6),

sturcture_type Varchar2(30),

bridge_type Char(40) NOT NULL,

Geometry SDO_GEOMETRY

);

CREATE TABLE have_access

(

building_FK Number(7),

segment_FK Number(7),

CONSTRAINT pk_have_access PRIMARY KEY (building_FK, segment_FK),

CONSTRAINT fk_building FOREIGN KEY (building_fk)

REFERENCES building (building_id),

CONSTRAINT fk_segment FOREIGN KEY (segment_fk)

REFERENCES segment (id_segment)

);

CREATE TABLE oneway

(

code char(2) CONSTRAINT pk_oneway PRIMARY KEY,

label varchar2(5)

);

CREATE TABLE pavement

(

code number(1) CONSTRAINT pk_pavement PRIMARY KEY,

label varchar2(5)

);

CREATE TABLE road_class

```
(
code          number(1)    CONSTRAINT pk_road_class PRIMARY KEY,
label         varchar2(5)
);
```

References on PSM level:

Essential a - chapter 3-2.3 in Yeung and Hall, [2007]

References on SQL:

Essential a - chapter 2-3.7 in Yeung and Hall, [2007]

Essential b: Database design with UML and SQL

- Create indexes and spatial indexes to improve data retrieval.

References on indexes and spatial indexes:

Essential a - chapter 2-5.3 and chapter 4-4.2.2 in Yeung and Hall, [2007]

- Develop the procedures and triggers previously designed in PL/SQL or other programming language.

References on trigger:

Essential a - Example in Yeung and Hall, [2007], figure 2.5(e) p. 40.

Wikipedia: http://en.wikipedia.org/wiki/Database_trigger

Implementation Phase:

- Load data with SQL DML command such as INSERT and/or with IMPORT/EXPORT command. In Oracle, SQL Loader can be used to batch load data. Here are INSERT examples loading data in domain tables.

```
INSERT INTO pavement DOMAINS VALUES ('1', 'rigid');
```

```
INSERT INTO pavement DOMAINS VALUES ('2', 'flexible');
```

```
INSERT INTO pavement DOMAINS VALUES ('3', 'paving');
```

```
INSERT INTO oneway DOMAINS VALUES ('FT', 'From-To direction');
```

```
INSERT INTO oneway DOMAINS VALUES ('TF', 'To-From direction');
```

```
INSERT INTO road_class DOMAINS VALUES ('1', 'Freeway');
```

```
INSERT INTO road_class DOMAINS VALUES ('2', 'Expressway');
```

```
INSERT INTO road_class DOMAINS VALUES ('3', 'Artery');
```

```
INSERT INTO road_class DOMAINS VALUES ('4', 'Collecting road');
```

```
INSERT INTO road_class DOMAINS VALUES ('5', 'Local/street');
```

- Optimize the database structure using indexes, spatial indexes, clusters, partitions and denormalized the database structure

References on database optimization:

Essential a - chapter 4-4.2.2 , chapter 2-5.3 chapter 11-2.4.5, chapter 4-4.2.2 in Yeung and Hall, [2007]

- Different types of tests can be applied to see if the database is correct and if the processes work as planned.

References on SQL:

Essential a - chapter 2-3.7 in Yeung and Hall, [2007]

Essential b: Database design with UML and SQL

Maintenance/Operation Phase:

- Maintain the database (add new records, update and delete others) with Data Manipulation Language such as INSERT, UPDATE, DELETE and SELECT and operate the database with SQL queries such as:

- Select all the tunnels in the database:

```
SELECT *  
FROM structure  
WHERE type = 'Tunnel';
```

- Count the number of structure by type

```
SELECT COUNT(structure_type)  
FROM structure  
GROUP BY structure_type;
```

- Select all buildings having access to segment where speed limit is 30 km/h

```
SELECT building_id, building_type  
FROM building, have_access, segment  
WHERE building_id=building_fk AND segment_fk= segment_ID  
AND speed_limit = 30;
```

- Or spatial queries using OGC Spatial operator, GIS spatial operator or Oracle SDO_relate operator:

- Select buildings located within the municipality of Ottawa.

With OGC Operator

```
SELECT building WITHIN Municipality WHERE name= 'OTTAWA'
```

Or with ESRI ArcGIS operator

```
SELECT building ARE CONTAINED BY Municipality WHERE name=  
'OTTAWA'
```

Or with Spatial Oracle

```
SELECT a.building_id, a.building_type  
FROM building a, municipality b  
WHERE b.name='OTTAWA' AND SDO_RELATE(a.Geometry,  
b.Geometry,'mask=inside querytype=WINDOW') = 'TRUE';
```

Or any other working SQL statement and GIS operator.

- And verify if the database can be used to answer such questions:

- Which 0D (point) buildings are near to 2D (polygon) public buildings (building_type = 'Public')?
Yes, it can be done with “Within distance” operator.
- Which municipalities maintain selected Highway segments?
No, it can't be done because we do not have the relationship “Maintain” between Segment and Municipality.

References on SQL:

Essential a - chapter 2-3.7 in Yeung and Hall, [2007]

Essential b: Database design with UML and SQL

References for Spatial SQL:

Essential a - chapter 4-4.3 in Yeung and Hall, [2007]