

## Conseil canadien pour les examinateurs des arpenteurs géomètres

### Éléments du tronc commun

## E 1: BASES DE DONNÉES SPATIALE ET SYSTÈMES D'INFORMATION FONCIÈRE (SIT/LIS)

### Guide d'études :

Ce guide d'études fournit un exemple complet du développement d'une base de données spatiales à partir de l'étude de faisabilité jusqu'à son utilisation et sa maintenance. Pour cet exemple, nous utilisons une méthode chute d'eau mais, pour l'examen E1, toutes les autres méthodes de développement de logiciels ou de conception de bases de données pourraient être utilisées. Afin de bien illustrer les solutions utilisant divers formalismes et approches, nous avons utilisé plus d'un schéma pour chaque niveau de modélisation des bases de données. Lors de l'examen E1, on pourra utiliser des formalismes autres qu'utilisés dans cet exemple. Nous utilisons ici le vocabulaire issu de la méthode standard OMG intitulée "*Model-driven Architecture (MDA)*" afin d'identifier chaque niveau de modélisation. Selon ce vocabulaire, Le CIM est le Modèle indépendant de l'informatisation, le PIM est le Modèle indépendant de la plate-forme et le PSM est le Modèle spécifique à la plate-forme. Ces modèles correspondent de manière respectueuse aux modèles conceptuels, logiques et physiques des modèles de données issus d'autres méthodes.

Ce guide d'étude comporte des renvois aux chapitres et sections des références essentielles et supplémentaires de cet élément du tronc commun.

### Étape de planification :

- Compréhension globale des fonctions opérationnelles.
- Évaluation initiale des besoins des usagers. Qui utilisera la base de données et à quelles fins ?
- Étude de faisabilité de mise en oeuvre de la base de données. Quels devraient être les coûts afférents ? Quelles sont les ressources humaines requises ? Quels sont les risques d'échec ?
- Choix d'une méthode de conception et de développement.

### *Références sur les systèmes et cycle de vie des bases de données :*

Essentiel a - chapitre 3-4.2 dans Yeung and Hall, [2007]

Essentiel d - Observations on Balancing Discipline and Agility

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

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

### *Références sur le MDA (architecture dirigée par les modèles) :*

Essentiel a - chapitre 8-5.4 dans Yeung and Hall, [2007]

Supplémentaires a: OMG Model Driven Architecture

### *Référence sur les questions organisationnelles et institutionnelles pertinentes à la gestion de projets de données spatiales :*

Essentiel a - chapitres 7, 8 et 9 dans Yeung and Hall, [2007]

### *Références sur les schémas UML (langage de modélisation unifié):*

Essentiel c - Introduction au UML

## **Étape d'analyse :**

- Évaluation systématique et plus détaillée des besoins des usagers.
- Évaluation des pratiques d'affaires et d'exploitation déjà en place. Les UML Use Cases sont de bons schémas pour illustrer ces pratiques et opérations et quelques besoins des usagers.
- Évaluation des sources de données existantes suite à leur inventaire à la récolte de leurs métadonnées. Ces sources de données sont souvent utilisées afin de charger une base de données, habituellement suite à un nettoyage et une transformation. L'inventaire des classes d'objets et les attributs des ensembles de données peuvent être utilisés comme données d'entrée pour le CIM. On doit choisir les meilleures sources en fonction des exigences des usagers, des coûts et des questions de droit.

*Références sur les métadonnées et normes :*

Essentiel a - chapitre 5 dans Yeung and Hall, [2007]

*Références sur les aspects juridiques des données spatiales :*

Essentiel a - chapitre 7 dans Yeung and Hall, [2007]

## **Étape du design :**

- Concevez un CIM en fonction des besoins des usagers et qui est indépendant de la technologie. Ceci s'effectue habituellement en utilisant un langage de modélisation tel le UML (tableau 1), UML étendu pour les données spatiales (tableau 2) ou E/R (tableau 3). Ces modèles ne sont pas les seules solutions possibles; on pourra utiliser des alternatives pour l'examen I-5. Un progiciel de génie logiciel appelé CASE (Computer-Assisted Software Engineering) est souvent utilisé lors de l'étape de design. Parmi les outils qu'il comporte on peut trouver un dictionnaire et la génération de code automatique en plus d'outils de rétroingénierie (i.e. Modélisation à partir de code).
  - On utilise UML dans la figure 1 et la géométrie est définie selon l'objet GM tel que défini dans la norme de schéma spatial ISO-TC211 19107 et dans le modèle de référence Open GIS.
  - Dans la figure 2, on utilise le UML étendu pour références spatiales. Des pictogrammes spatiaux sont utilisés pour illustrer la géométrie des classes d'objets. La géométrie de la jonction est 0D, le segment et la structure, une géométrie de 1D l'édifice, une géométrie de 0D, 1D ou 2D selon les dimensions de l'édifice (habituellement expliqué dans le dictionnaire des données) et chaque municipalité est représentée par un, ou un agrégat de polygones. On utilise l'outil Perceptory pour CASE dans cet exemple.
  - Dans la figure 3, la géométrie est définie à l'aide d'une structure topologique composée de points, de lignes et de polygones. On utilise l'outil Oracle Designer pour CASE dans cet exemple.

*Références pour l' UML:*

Essentiel b et supplémentaires c à e.

Essentiel a - chapitre 3-4.5 dans Yeung and Hall, [2007]

*Références sur l'UML étendu pour données spatiales :*

Supplémentaires b: Using Spatial PVL for spatial database modeling

*Références sur les rapports entre entités :*

Essentiel a - chapitre 3-3.1 dans Yeung and Hall, [2007]

Références sur les outils CASE :

Essentiel a - chapitre 3-4.3 dans Yeung and Hall, [2007]

Références sur les types de données OGC :

Essentiel f: Simple feature access - Partie 1: Common architecture

Supplémentaires f: OpenGIS Reference Model (ORM)

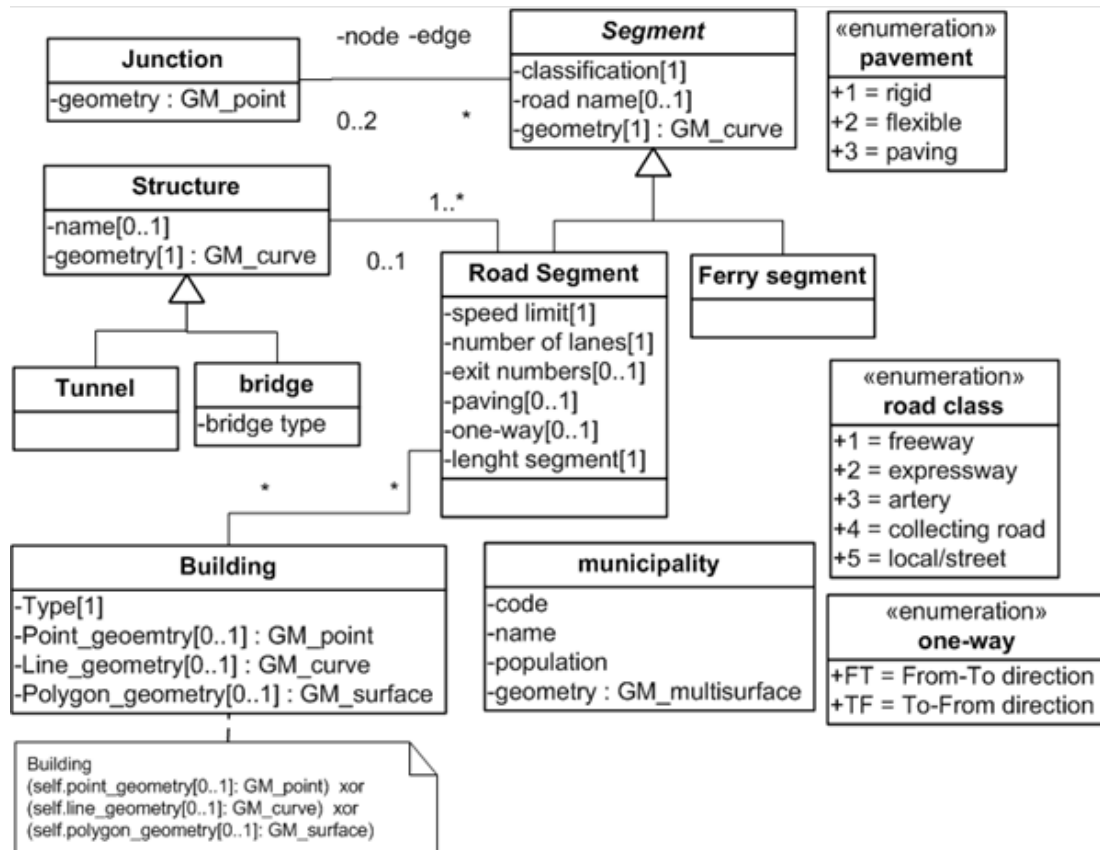


Figure 1 : CIM conçu à l'aide de UML et ISO TC/211 19107 dérivé de la classe GM\_object.

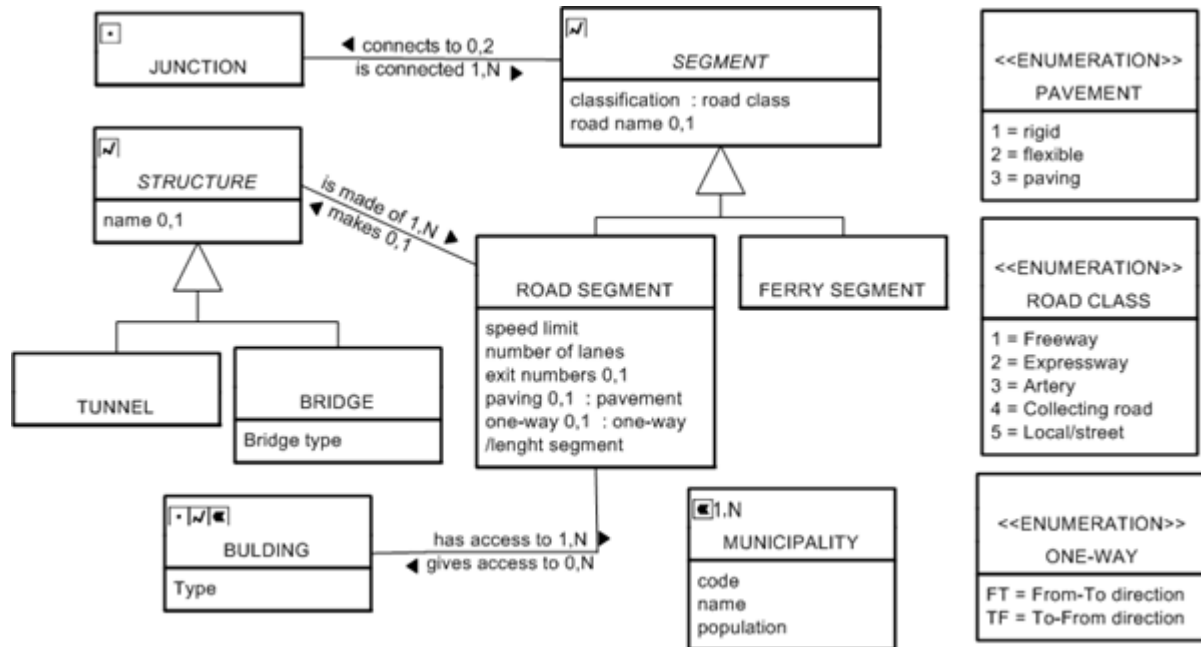


Figure 2 : CIM conçu à l'aide du langage UML étendu pour les références spatiales.

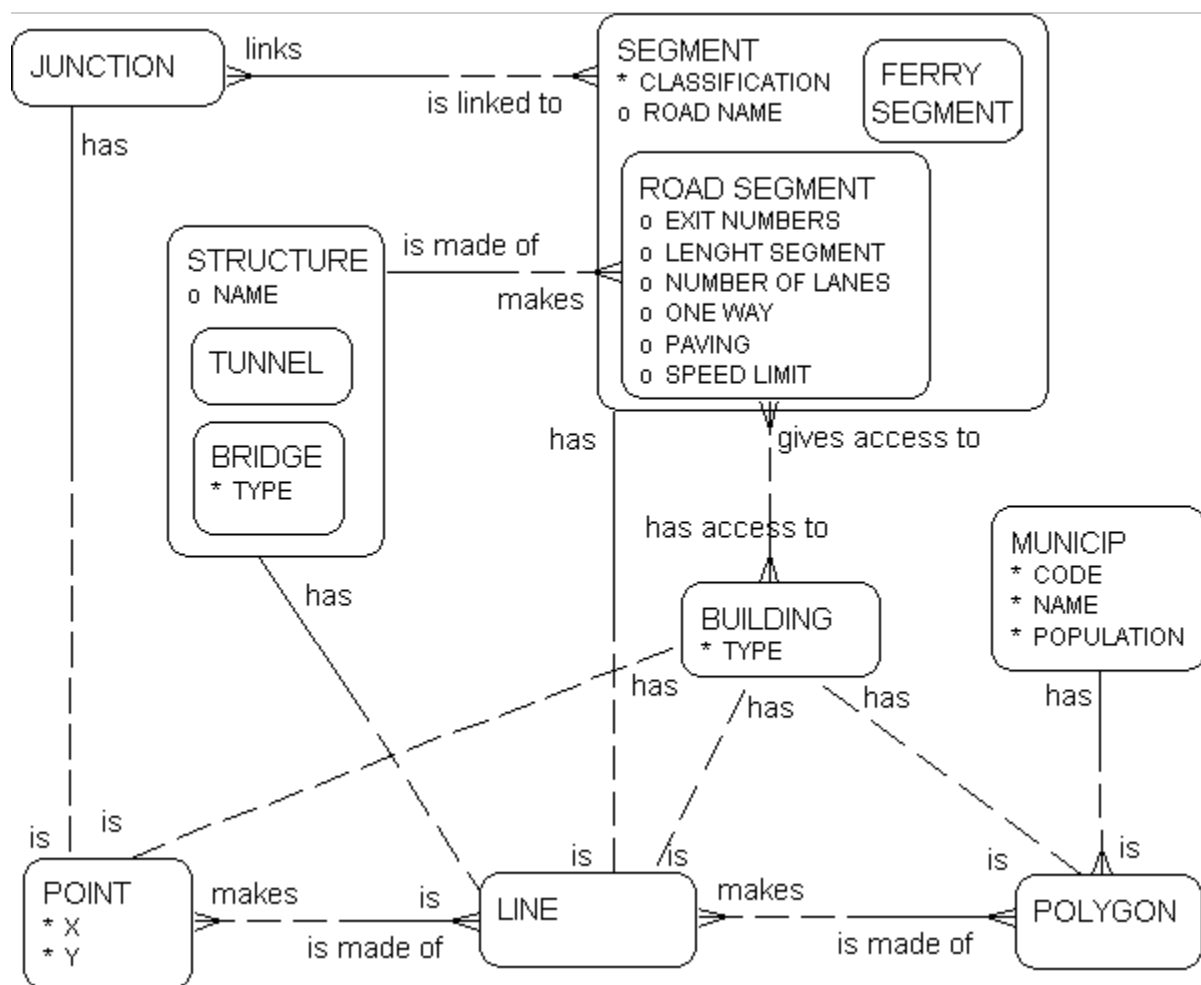


Figure 3: CIM conçu à l'aide du Entity Relationship Diagrammer de Oracle Designer.

- La plupart du temps, un CIM est normalisé - Les redondances des données sont réduites à un minimum ou éliminées. Les trois premières formes normales (NF) sont habituellement respectées.

*Références sur la normalisation :*

Essentiel a - chapitre 3-3.2 dans Yeung and Hall, [2007]

- Les contraintes d'intégrité spatiales devraient aussi être définies afin de garantir l'exactitude des relations spatiales entre les objets. la figure suivant illustre ces contraintes pour une jonction en utilisant l'opérateur OGC. Une jonction doit être reliée à au plus un segment de route ou un segment de bac/traversier et peut être relié à plusieurs d'entre eux.

	Relation		Cardinalité
Jonction	Touche <sup>1</sup>	Segment route	0,N*
Jonction +	Touche <sup>1</sup>	Segment bac	0,N*
Jonction	Détaché de	Jonction	

<sup>1</sup> doit être relié au moins à un segment de bac ou de route.

**Tableau 1: Exemple de contraintes d'intégrité spatiale utilisant l'opérateur OGC.**

*Références sur les contraintes d'intégrité spatiale :*

Essentiel a -chapitre 4-4.2.3 dans Yeung and Hall, [2007]

Supplémentaires g: Ressources naturelles Canada, 1996. BASE NATIONALE DE DONNÉES TOPOGRAPHIQUES dictionnaire :

[http://ftp2.cits.rncan.gc.ca/pub/bndt/doc/dictbnd3\\_fr.pdf](http://ftp2.cits.rncan.gc.ca/pub/bndt/doc/dictbnd3_fr.pdf). Voir la partie relation de chaque entité pour obtenir les exemples. On utilise maintenant les opérateurs OGC au lieu des opérateurs *Connect* et *Share*.

- Les processus sont aussi modélisés lors de cette étape en utilisant les schémas d'activité UML et le schéma séquentiel ou les diagrammes de cheminement des données (DFD). Concevez, par exemple, un processus destiné à calculer le nombre d'édifices ayant accès à un segment de route.

*Référence schémas d'activité et schémas séquentiels :*

Essentiel a - chapitre 4-4.2.3 dans Yeung and Hall, [2007]

- Choisir l'architecture de système de données spatiales appropriée ( centralisée, répartie, client/serveur, axée sur le web).

*Référence sur l'architecture de systèmes de base de données spatiale :*

Essentiel a - chapitre 2-4, chapitre 6, chapitre 10, chapitre 11 dans Yeung and Hall, [2007])

- Comparer les technologies des familles de bases de données transactionnelles et multidimensionnelles afin de pouvoir choisir celle qui s'applique le mieux (voir tableau 2). À partir de cette comparaison, votre client et vous décidez d'appliquer sa base de données selon un modèle transactionnel, plus spécifiquement relationnel.

---

<sup>1</sup> **OpenGIS® Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture.**

Transactionnel	Multidimensionnel
Version transactions	Version analyse
Priorité sur la sécurité et l'intégrité des données	Priorité sur l'analyse et l'exploration
Minimiser la redondance des données	Minimiser le temps de réponse aux interrogations
Données détaillées	Données détaillées et abstraites
Intégré pour les applications	Intégré pour l'entreprise
Mise à jour continue	Aucune mise à jour – ajout de nouvelles données seulement
Données actuelles	Données actuelles et d'archives
Données patrimoniales	Données importées de données patrimoniales
Structure normalisée	Structure dénormalisée

**Tableau 2: Comparaison entre les systèmes transactionnels et multidimensionnels.**

*Référence sur les différences entre transactionnel et multidimensionnel :*

Essentiel a - chapitre 6-4.1.5 dans Yeung and Hall, [2007]

### **Étape de construction de la version :**

- Traduire les CIM en un PIM relationnel, c'est à dire une modélisation de plateforme générique applicables aux systèmes de gestion des bases de données relationnelles de notre exemple. On utilise ici deux PIM pour illustrer l'exemple. Chacune d'entre elles effectuant l'implémentation de la généralisation des bases de données de manière différente et selon des formalismes différents. Les exemples illustrés aux figures 2 et 3 peuvent être utilisés avec toute base de données relationnelle puisqu'on y utilise des types de données génériques. De plus, cet exemple définit les contraintes d'intégrité. Par exemple: l'intégrité d'entité, (Clé primaire), intégrité référentielle (Clé étrangère), Données requises (champs caractères gras), contraintes de domaines (types de données et contrainte de contrôle des données) et contrainte d'entreprise. La traduction exacte de CIM à PIM requiert plusieurs étapes.

*Références sur les contraintes d'intégrité :*

Essentiel a - chapitre 2-3.2 dans Yeung and Hall, [2007]

*Références sur les étapes à suivre pour traduire de CIM à PIM :*

Essentiel b

Essentiel a - chapitre 3-2.3, 3.2 dans Yeung and Hall, [2007]

*Références sur les niveaux UML et PIM :*

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

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

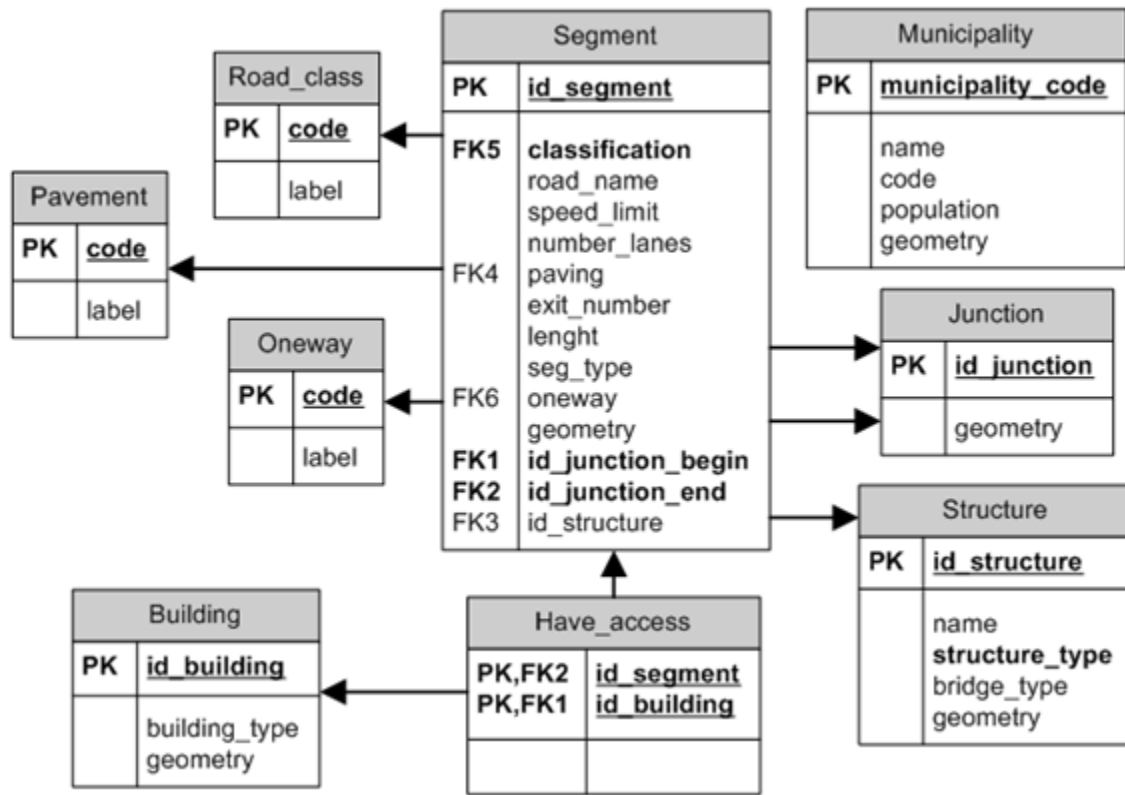


Figure 4: Modélisation de données relationnelles PIM utilisant le gabarit Microsoft Visio Entity/Relationship diagram. Les types de données sont définies au dictionnaire.

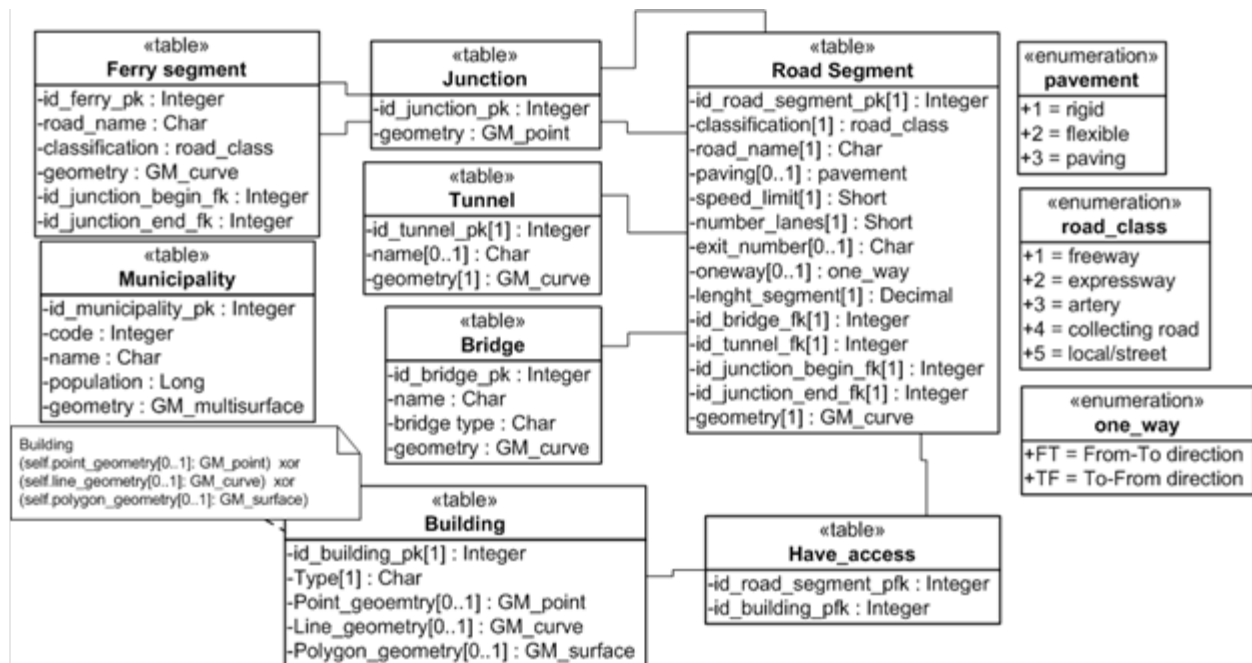


Figure 5: Un autre exemple PIM qui utilise l'annotation UML (et un tableau stéréotypé) pour le même CIM. La relation de généralisation est mise en place de manière différente.

- Choisir la plate-forme qui servira à la mise en place de la base de données (Par ex.: Spatial Oracle, ESRI Geodatabase) et choisissez la structure de données spatiales qui servira à entreposer les données géométriques (Par ex.: structure de données topologiques et non-topologiques).

*Références sur la structure de données spatiales :*

Essentiel a - chapitre 4 dans Yeung and Hall, [2007]

- Traduire le PIM en PSM. Pour notre exemple, nous avons choisi Spatial Oracle avec une structure de données non-topologiques. Les types de données supportées par la plate-forme choisies sont définis pour chaque attribut et un code DDL spécifique à cette plate-forme pourra être généré. Ici, le PSM Spatial Oracle est associé au PIM illustré dans la **Figure** (être capable, pour l'examen, d'effectuer la rétroingénierie sans nécessairement pouvoir écrire ce type de code SQL).

```
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)
);
```

*Références sur le niveau PSM :*

Essentiel a - chapitre 3-2.3 dans Yeung and Hall, [2007]

*Références sur le SQL :*

Essentiel a - chapitre 2-3.7 dans Yeung and Hall, [2007]

Essentiel b: Database design with UML and SQL

- Créer des index et index spatiaux afin d'améliorer l'extraction des données.

*Références sur les index et index spatiaux :*

Essentiel a - chapitre 2-5.3 et chapitre 4-4.2.2 dans Yeung and Hall, [2007]

- Développer les procédures et déclencheurs préalablement conçues en PL/SQL ou autre langage de programmation.

*Références sur les déclencheurs :*

Essentiel a - Exemple dans Yeung and Hall, [2007], figure 2.5(e) p. 40.

Wikipedia: [http://en.wikipedia.org/wiki/Database\\_trigger](http://en.wikipedia.org/wiki/Database_trigger)

## **Phase d'implémentation :**

- Charger les données en utilisant une commande SQL DML telle que INSERT et/ou avec la commande IMPORT/EXPORT. En Oracle, le chargeur SQL peut être utilisé pour charger les données en lots. Voici des exemples de l'utilisation de la commande INSERT pour charger des données dans des tables de domaines.

```
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');
```

- Optimiser la structure de la base de données à l'aide d'index, d'index spatiaux, de grappes de données, de partitions et une structure de base de données dénormalisée.

### *Références sur l'optimisation des bases de données :*

Essentiel a - chapitre 4-4.2.2 , chapitre 2-5.3 chapitre 11-2.4.5, chapitre 4-4.2.2 dans Yeung and Hall, [2007]

- On peut effectuer plusieurs tests pour vérifier l'intégrité de la base de données et pour s'assurer du bon déroulement des processus.

### *Références sur le SQL :*

Essentiel a - chapitre 2-3.7 dans Yeung and Hall, [2007]

Essentiel b: Database design with UML and SQL

## **Étape de Maintenance/Opération :**

- Maintien de la base de données (ajout de nouveaux enregistrements, mise à jour et élimination de certains autres) à l'aide d'un langage de manipulation de données tels INSERT, UPDATE, DELETE et SELECT et utiliser la base de données en utilisant des commande SQL telles:

- Choisir tous les tunnels de la base de données :

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

- Compter les nombre de structures par type :

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

- Choisir les édifices ayant accès à un segment où la limite de vitesse est 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;
```

- Ou des interrogations spatiales à l'aide de l'opérateur spatial OGC, GIS ou l'opérateur SDO\_relate Oracle :

- Choisir les édifices situés dans la municipalité de Ottawa :

**Avec opérateur OGC**

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

**ou avec l'opérateur ESRI ArcGIS**

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

**Ou avec 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';
```

**Ou toute autre commande SQL fonctionnelle et opérateur GIS.**

- Vérifier si la base de données peut être utilisée pour répondre à des questions telles que :
  - Quels édifices 0D (point) sont situés à proximité d'édifices publics 2D (polygone) (building\_type = 'Public')?
 

Ceci peut être réalisé à l'aide de l'opérateur “Within distance”.
  - Quelles municipalités maintiennent certains segments de routes sélectionnés ?
 

Ceci ne peut être réalisé puisque nous ne possédons pas la relation “Maintain” entre Segment et Municipality.

*Références sur le SQL:*

Essentiel a - chapitre 2-3.7 dans Yeung and Hall, [2007]

Essentiel b: Database design with UML and SQL

*Références sur le SQL Spatial:*

Essentiel a - chapitre 4-4.3 dans Yeung and Hall, [2007]