GEOG 5150/6150 – Fall 2020
Lab 2: Entity–Relationship (ER) Diagram
Due Sunday, 13 September by 11:59pm

Part 1

Overview
A municipal planning department has hired your GIS consulting firm "Rockstar Consulting" to build a database for their geospatial data. After meeting with the department, your project team has determined a set of data requirements, and it is your responsibility to create a conceptual model of the database given those requirements. The conceptual model will be in the form of an Entity-Relationship Diagram (ER Diagram), representing the data and their connections within the database.

Objectives
- Construct an ER Diagram based on provided data requirements
- Become familiar with the *draw.io* diagram tool
- Gain practice creating a professional quality product for an employer

draw.io
draw.io is one option for producing conceptual models for this course and is fairly intuitive to use once you are oriented with its basic features. Online tutorials for using draw.io can be found at User Manual for draw.io and Working With Diagrams (basic diagram construction). These tutorials discuss the draw.io environment, creating a diagram, and adding arrows and connectors.

Instructions
Using draw.io, create an ER Diagram for the municipality based on the information provided below. Note that these requirements do not specify every detail of your conceptual model; you will still need to make some decisions on your own. Be sure to look at the Additional Requirements section as well.

Data Requirements (Entities, Relations, Attributes, Integrity Constraints)
Note: Entity Types should be expressed as a singular noun in the ER Diagram (e.g. Parcel, not Parcels).

1. The **City** is its own entity type in the database, with attributes for **name** and **area**.
2. The **City** is also divided into **TAZ**, or Traffic Analysis Zones. **TAZ** entities have attributes for **ID** and **area**. The **City** also contains **Streets**, which have attributes for **basename**, **fullname**, **type**, and **capacity**. Since it is possible that two streets have the same name, **Streets** *does not have a unique identifier*. However, the planning department has determined that using the **TAZ ID** in conjunction with the street's **basename** will

uniquely identify **Streets**. **TAZ** have many streets, and a street can cross into multiple **TAZ**.

3. The city contains **Schools** and **Parks**, both with **ID** and **name** attributes. **Schools** have additional attributes for **type** and **capacity**. **Parks** have an additional attribute for **area**. The planners need to be able to distinguish between **Parks** with recreation facilities. This subgroup of **Parks**, called **Facility Parks**, needs an additional attribute for **facility type**, which can have multiple values. Planners also need to distinguish parks that are **Courtyards**. This subgroup **Courtyards** has an attribute for **building**. A park does not necessarily need to be a **Facility Park** or a **Courtyard**, but it cannot be both.

Hint: The resulting ER-Diagram contains a weak entity set and a subtype.

Additional Requirements

- Indicate the cardinality for each relationship type.
- Indicate the key field for each entity type.
- Your final diagram should also include the appropriate heading and title information, as you would include in any professional project. How to format this and what to include is up to you.

Scoring Rubric Part 1 (10 points total)

All entity types are included (2 points)
Attributes for each entity type are present, with a key field denoted for each (2 points)
All relationships included, with cardinality denoted (2 points)
Data Requirements are captured accurately in diagram (3 points)
Well organized and professionally presented (1 points)

Part 2

Overview

Now that the conceptual data model is finalized, you are tasked to translate it into a logical data model. For this assignment, you will translate the ER Diagram from Part 1 into a Relational Model Diagram. Draw.io is includes relation templates and a snapping function for drawing arrows.

Objectives

- Construct a Relational Model Diagram based on provided data requirements
- Build skills in translating a conceptual data model into a logical data model
- Gain practice creating a professional quality product for an employer/client

Instructions

Translate your ER Diagram from Part 1 into a Relational Model Diagram. The requirements for your Relational Model Diagram are as follows:

1. All entity types, relationship types, and attributes must be included in the diagram.
    a. See the Lab 2 Lecture to review specific translation techniques.
2. Follow the lecture material to appropriately translate unique features, such as Weak Entity Types, Supertypes/Subtypes, M:N relationships, etc.
3. Specify a data type for all of your attributes (abbreviations are fine).
4. Primary Keys must be underlined.
    a. For composite primary keys, underline all attributes that participate.
5. Foreign Keys need to be indicated and must be included in the diagram based on the model's specifications.
6. Include appropriate title and heading information as you would in a professionally produced product.

Scoring Rubric – Part 2 **(10 points total)**

- All entity types, relationship types, and attributes included **(3 point)**
- Unique Features are denoted appropriately **(2 points)**
- All attributes include a data type **(2 points)**
- All Primary Keys are underlined **(2 points)**
- Title/Header included **(1 point)**

Part 3

Overview

All of a sudden, your client asked you to design an object-oriented database rather than a relational database. In this scenario you will use the conceptual model (ER Diagram) developed in Part 1 to build a Class Diagram of the database. Draw.io has a section of diagram elements titled "UML" that contains many of the appropriate artifacts for your diagram (you may need a few artifacts from other sections as well).

Objectives

- Construct a Class Diagram based on provided data requirements
- Build skills in translating a conceptual data model into a logical data model
- Understand Unified Modeling Language (UML) and its application to database design
- Gain practice creating a professional quality product for an employer

Instructions

Develop a Class Diagram based on the ER Diagram from Part 1. Follow the guidelines for translation outlined during lecture. In addition, you are asked to add 3 new classes to the data model: Point, Line, and Polygon. You will need to connect the relevant classes to one of those 3 with an Inheritance Relationship based on which geometry you decide that class should have. In other words, your final diagram will use inheritance relationships to specify whether the

classes are points, lines, or polygons. Read the instructions below carefully and be sure to follow each step in completing the assignment.

1. All entity types, relationship types, and attributes must be included in the class diagram (as their object-oriented analogues).
2. Specify a data type for each attribute.
3. Specify multiplicity for each association/composition (no aggregations).
4. Add 3 new classes to the data model: Point, Line, and Polygon. Include these attributes and methods:
    - Point
        - Attributes: PointID (INT), X (DBL), Y (DBL)
        - Methods: drawPoint()
    - Line
        - Attributes: LineID (INT), Length (DBL)
        - Methods: drawLine(), getLength()
    - Polygon
        - Attributes: PolygonID (INT), Area (FLOAT)
        - Methods: drawPolygon(), getArea()
5. For each class, determine whether it is non-spatial or spatial data. If it is spatial, connect it to one of the geometry classes (point, line, polygon) with an Inheritance Relationship. Include at least 1 inheritance relationship for each geometry class (note: there is a good candidate for each in the data model).
6. Color code each class according to its spatial data class.
7. Include appropriate title and heading information.

Grading Rubric **(10 pints total)**

- All entity types, relationships, and attributes included **(1 point)**
- All attributes include a data type **(2 points)**
- Multiplicity is specified **(2 points)**
- Geometry classes are added, and spatial classes are connected appropriately **(4 points)**
- Title/Header included **(1 point)**

Deliverables

Submit three PDFs, one for each of your Diagrams. Name them as follows:

- *Lab2_Part1_yourLastName*
- *Lab2_Part2_yourLastName*
- *Lab2_Part3_yourLastName*