**CS 7330 Project Part 1**

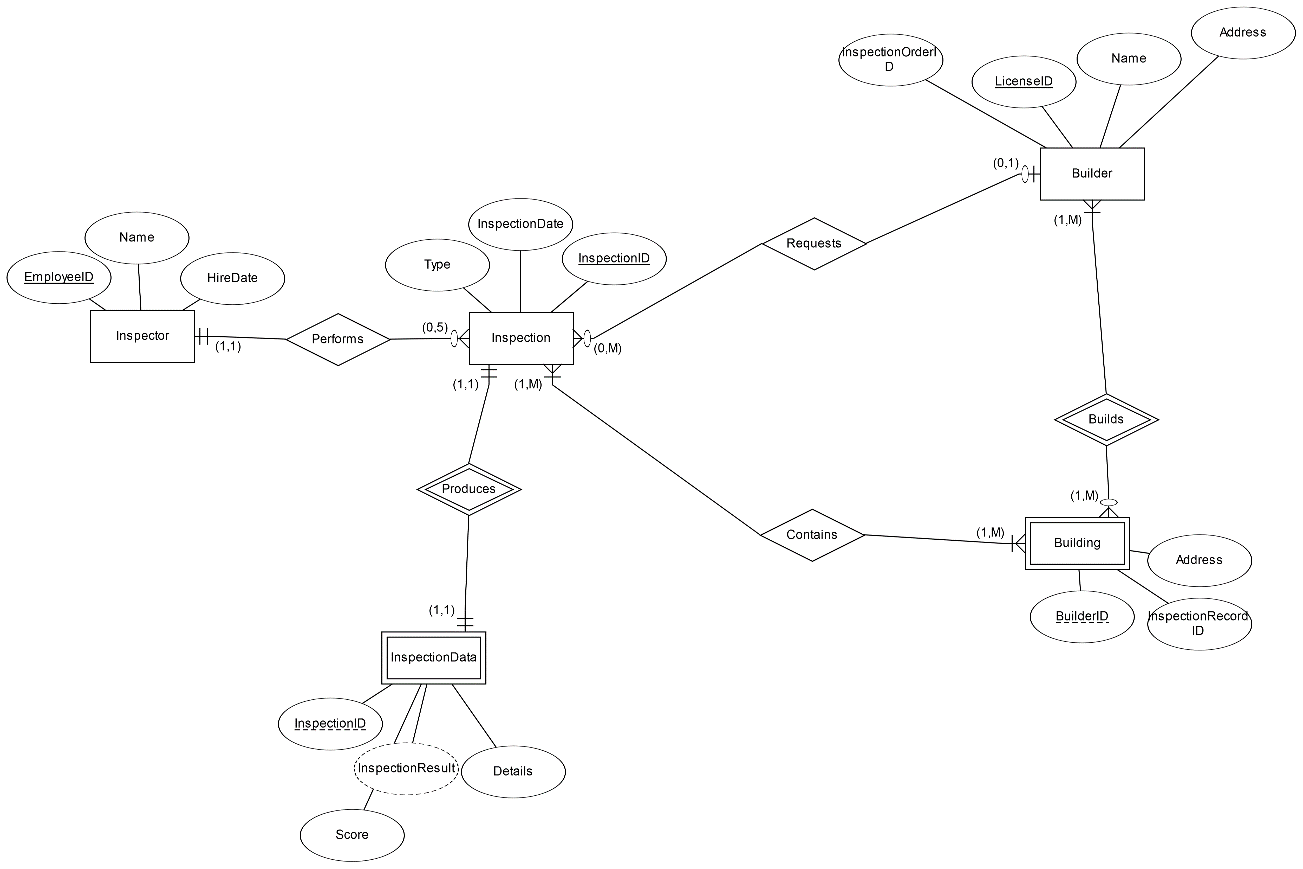
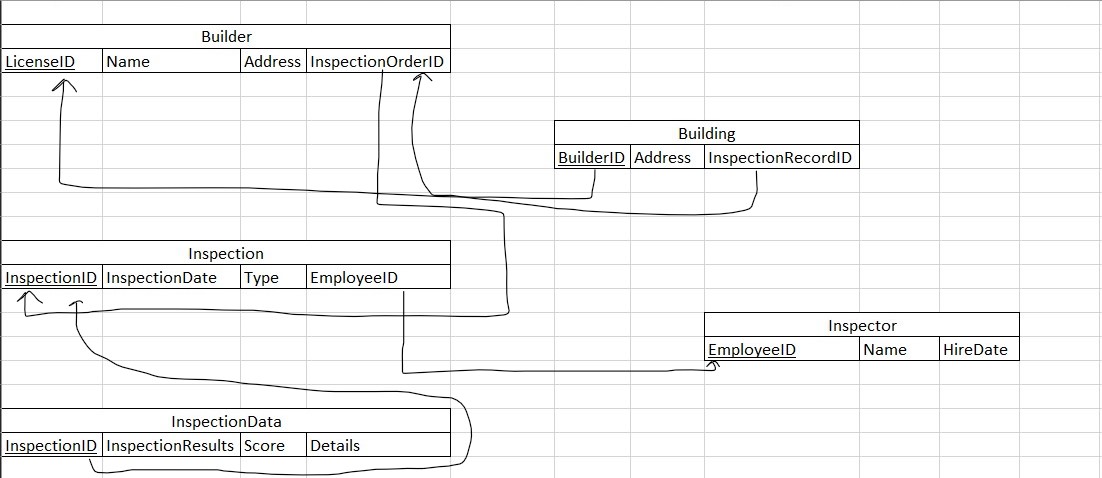
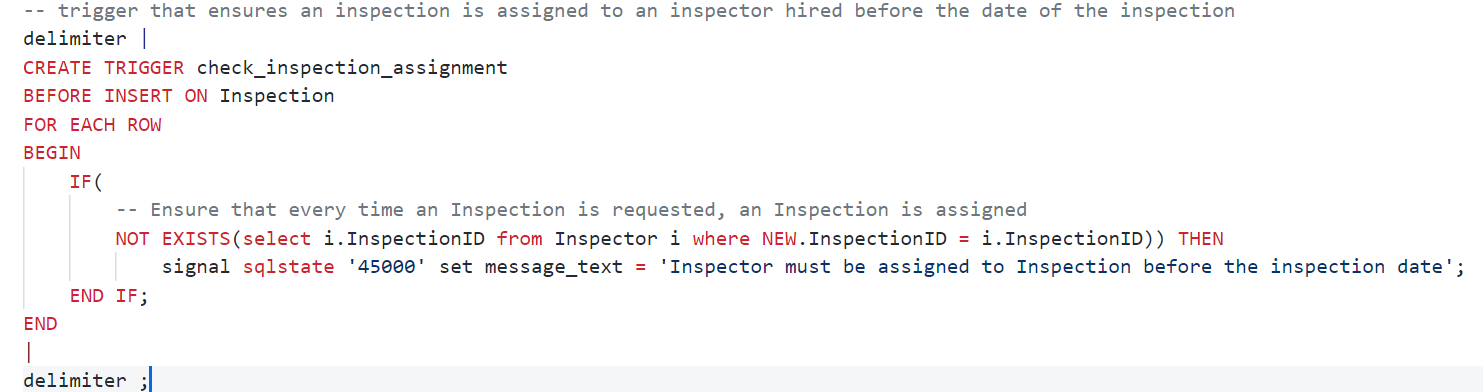
**Problem**:

You are to ***design*** a database to keep track of building inspections required by the department of Faulty or Damaged Buildings (FODB). Implementation will be done in part 2.

**Data Requirements**:

FODB coordinates building inspections requested by builders on a monthly basis. Each building inspection is either passed or not. Inspections have a type code (3 characters, e.g. PLU, FRM, ELE, etc.) and possibly sequencing requirements. Some inspections cannot be performed before other inspections, e.g. final plumbing inspection cannot be performed until the framing inspection is passed. Each inspection has a numeric score, with 75 or more out of 100 being sufficient for a pass status. Each inspection data contains the date of inspection, inspector identification, inspection score, and textual information about the inspection. The textual information can be updated later, but the score can never be changed. FODB maintains a pool of inspectors. Each inspector has a unique 5 digit employee ID, name, and date they were hired. They can conduct any type of inspection but can only perform at most 5 inspections per month. Any failed inspection can be repeated until passed. Particular information maintained about builders includes: Name (30 byte character string), address (40 byte character string), license# (5 digit number). A builder’s license# is unique. A builder and location must exist prior to requesting an inspection. A request for an inspection may be assigned to any available inspector assuming the prerequisite inspections have a pass status.

**Tasks:**

1. Construct an ER Diagram with attributes, being precise in your notation, including cardinality constraints. The ER diagram you create must support all requirements stated above. If you add any restrictions or information not stated above, please specify.
   1. 
   2. Additions:
      1. Inspection
         1. InspectionID – Identifies an inspection, primary key
         2. InspectionData
            1. Weak entity
            2. InspectionID – Weak key to refer to Inspection.InspectionID
            3. InspectionResult – To be derived from InspectionData.Score
      2. Builder
         1. InspectionOrderID – Identifies which inspection was requisitioned, primary key
      3. Building – new weak entity
         1. BuilderID – Identifies which Builder worked on the Building, partial key to foreign key with Builder.LicenseID
         2. Address
         3. InspectionRecordID – Identifies which Inspection is attached to the Building, to be foreign key’d to Inspection.InspectionID
2. Given your ER diagram, provide an initial description of the tables you plan to create, identifying keys and foreign keys (i.e. an initial relational schema)
   1. 
   2. Builder – This will represent a Builder having primary key of LicenseID and using InsepctionRecordOrderID as foreign key reference to Inspection.InspectionID to represent the “Orders” relationship.
   3. Building – This will represent a Building having only a foreign key reference of BuilderID to Builder.LicenseID as the primary key to represent the “Builds” relationship.
   4. Inspection – This will represent an Inspection having primay key of InspectionID, which also foreign key references Inspector.InspectorID to represent the “Performs” relationship.
   5. Inspector – This will represent an Inspector having primary key of EmployeeID
   6. InspectionData – This will represent what an Inspection produces, having only a foreign key reference of InspectionID to Inspection.InspectionID.
3. Write the code for a trigger that ensures an inspection is assigned to an inspector hired before the date of the inspection.
   1. 

Part 2 of the project will involve adding data and querying for particular results.