Entity Relationship Diagram for a College

John Wensink

(MIS300) Information Systems Design and Management

Colorado State University-Global Campus

Nathan Braun

December 1, 2018

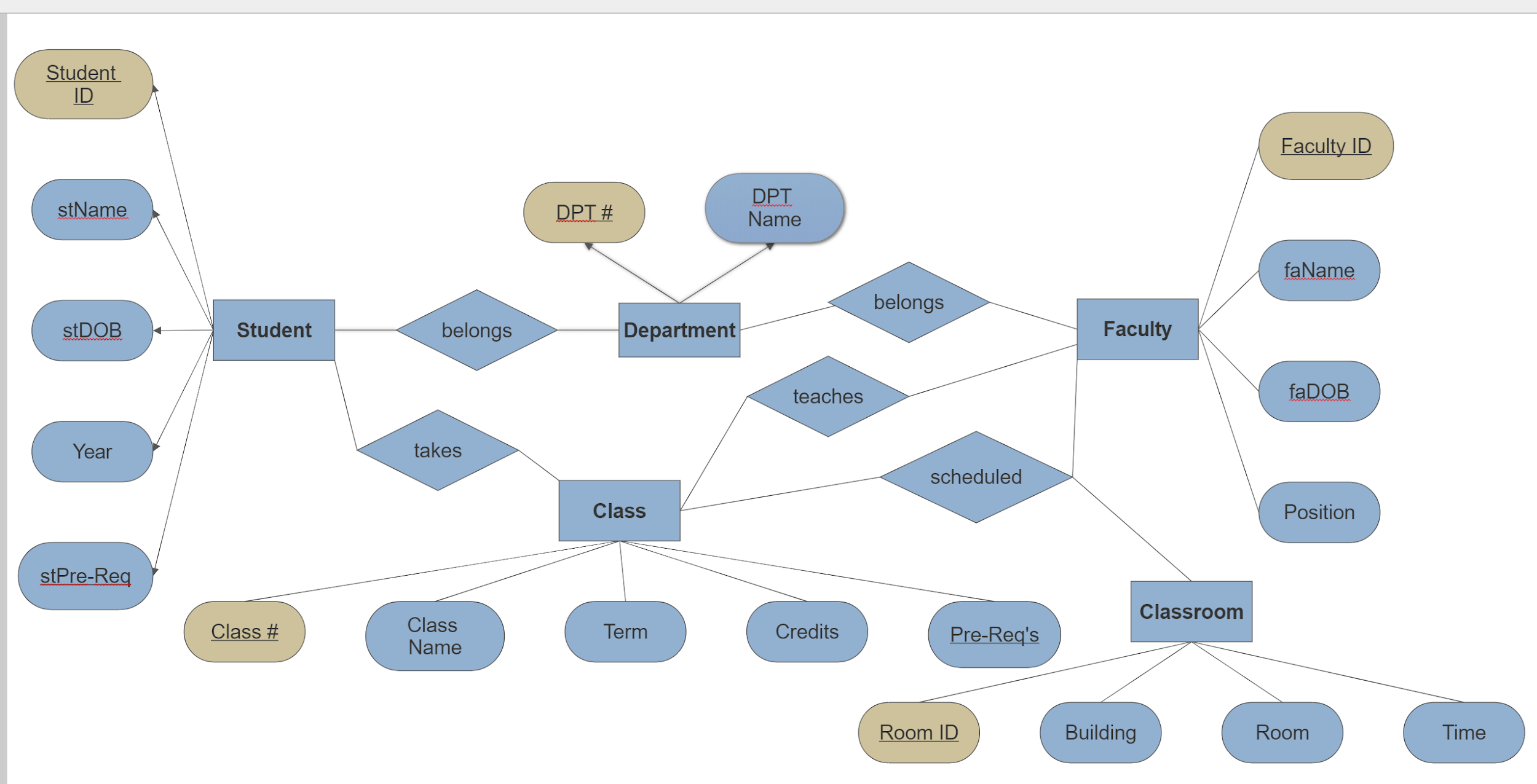
Entity Relationship Diagram for a College

The image below depicts an entity-relationship diagram that represents a basic database hierarchy for a small college’s class scheduling interface. Entities include the student, department, faculty, class, and classroom. Different entities are depicted with bold names in blue rectangles. Each entity has several attributes which can be found inside the oval boxes. For each entity, a primary key is selected and shaded tan. The primary keys selected are all based on unique ID numbers that are discrete and can be used as unique identifiers. Using these values as primary keys will allow for relational modeling and comparing fields in multiple tables. "In the relational model, data is placed in two-dimensional tables, or relations. As long as they share at least one common attribute, these relations can be linked to provide output useful information. In this example, all three tables include the Dept. number attribute." (Satir, 2018) There exist a number of conditionals (if/then rules) that are a part of this ER diagram such as that a student must have completed prerequisite courses before enrollment is allowed in a class; and that students, faculty, and classes are all schedule restricted by time, as a person cannot be two places at once. Using this basic framework, a student ought to be able to self-enroll without conflicts or prerequisite issues.

As I had never heard of an ER diagram before this week’s reading, I turned to stackoverflow.com to see how this type of modeling can be applied to real-world challenges. I have found that ER diagrams are useful in many types of databases such as medical and education as provided in our course, but also has applications in HR and online commerce. When researching ideas on how to make a good ER diagram, (Reaanb, 2017) provides several steps in order to produce a functional model:

For each column, determine whether it represents an entity set or a value set. Entity keys are normally primary keys in one table and foreign keys in others. Values sets generally represent labels and measurements and are independent columns. Identify relationships. Relationships are represented by two or more entity keys in the same table, at least one of which is part of the primary key. Identify attributes. Attributes are mappings from entity sets or relationships to value sets. The primary key of a table will determine the entity or relationship set (atomic or composite PK), with which the dependent columns (value sets) are associated, forming attributes. Make a diagram. Represent each entity set with a rectangle and each relationship set with a diamond. Connect relationships to the related entity sets. Draw keys and attributes as ovals attached to the determining entity or relationship set. We don't draw keys for relationships - they are determined by the keys of the associated entity sets. (Reaanb, 2017)

I chose the SmartDraw online javascript web app to produce my first ERD because of its simplicity, cost, and recommendation of my professor. I noticed from StackOverflow that Oracle SQL is what many pros use for production work. This app required vast knowledge of SQL language which I do not, at this time, have. SmartDraw is based on a customized bantam.js library (SmartDraw, n.d.) and has a dedicated template for flowcharts that are stylistically similar to the ER diagram we were assigned. I found the cloud app confusing and frustrating at first but was able to google anything I was confused about for help. While not the prettiest ERD ever created, I believe this introduction to entity relationship modeling delivered insight into a new world of database infrastructure to which I had not yet been exposed.



References

Reaanb. (2017, April 21). Relational Schema to ER Diagram. Retrieved December 2, 2018, from https://stackoverflow.com/questions/43537743/relational-schema-to-er-diagram

SmartDraw. (n.d.). Why We Ditched Angular JS. Retrieved from https://www.smartdraw.com/technology/ditching-angular-js.htm

Stair, R. M., & Reynolds, G. W. (2018). Fundamentals of Information Systems. Boston, MA: Cengage Learning.