

Unit 3 Research Project

Eddie S. Jackson

Kaplan University

IT525: Database Design and Data Modeling

05/26/2014

## Unit 3 Research Project

### Definitions

**Attribute Inheritance.** Attribute inheritance is a type of property associated with entities and relationships, specifically supertypes and subtypes. A supertype would contain common attributes that its subtypes would inherit. A good example of this can be observed in a supertype VEHICLE. VEHICLE would include attributes such as cost, color, and weight. Its subtypes could be CARS, TRUCKS, and MOTORCYCLES. Note, each of the subtypes would automatically inherit the attributes of VEHICLE. (Coronel, Morris, & and Rob, 2013)

**Supertype.** A supertype is a type of generic entity that shares common attributes with similar entities. An example of this is can be seen in a STUDENT supertype. In this supertype, some common values could be LastName, FirstName, and GraduateStatus. These attributes would be inherited by its subtypes. (Coronel, Morris, & and Rob, 2013)

**Subtype.** A subtype is a type of entity that forms a one-to-one relationship with a supertype. A subtype inherits attributes from a supertype. An example of this would be in the subtype HOURLY\_EMPLOYEE. HOURLY\_EMPLOYEE inherits Hire\_Date, Address, and Employee\_Name from the supertype EMPLOYEE. However, at the same time may have its own unique attributes such as Hourly\_Wage. (Coronel, Morris, & and Rob, 2013)

**Generalization.** Generalization refers to a bottom-up approach in the hierarchical structure of entities and relationships, where the common attributes are combined into a higher-level entity (a supertype entity). A simple example would be having multiple types of instruments, such as STRING and BRASS that have a bottom-up connection to the entity INSTRUMENT. (Coronel, Morris, & and Rob, 2013)

**Composite Key.** A composite is the combination of entity attributes, which are useful in representing a many-to-many and a weak-and-strong entity relationships. An example of a composite key can be observed in the entities of CLASS and STUDENT, both connected to an entity ENROLLMENT (in a one-to-many relationship). STUDENT will have a primary key such as classID, and STUDENT will have the primary key of stuID. The composite key can be seen in the ENROLLMENT entity as classID *and* stuID. (Coronel, Morris, & and Rob, 2013)

**Total Completeness.** Total completeness is a type of constraint placed upon supertypes and subtypes where the supertype must also be a member of at least one subtype entity. Additionally, the subtype discriminator cannot be empty or null, and the subtypes have unique sets. Note, total completeness is symbolized by a circle over a single line. An example of total completeness can be observed where STUDENT is connected to GRADUATE and UNDERGRAD subtype entities, and is required to be a graduate student or an undergraduate student. (Coronel, Morris, & and Rob, 2013)

**Partial Completeness.** Partial completeness is a type of constraint placed upon supertypes and subtypes where the supertype has optional subtypes. Additionally, the subtypes can be empty or null, and subtypes have unique sets. Note, partial completeness is symbolized by a circle over a double line. An example of partial completeness would be where EMPLOYEE is connected to ADMINISTRATOR and PROFESSOR; the employee could be an administrator and/or a professor (or neither). (Coronel, Morris, & and Rob, 2013)

**Surrogate Key.** A surrogate key is a type of key that is system-assigned which is usually in the form of numbers and automatically incremented. For example, in the table of EMPLOYEE, in the column EMP\_NUM, the surrogate key would be the consecutive identification number

signifying each additional item in the table; the surrogate would look like 101, 102, 103, 104, etc. (Coronel, Morris, & and Rob, 2013)

**Disjoint Subtype.** Disjoint subtypes specify a unique entity set that is nonoverlapping. For example, in an EMPLOYEE supertype connected to three subtypes defined as PILOT, MECHANIC, and ACCOUNTANT; the disjoint would dictate that the employee could only be in only one of the subtypes. (Coronel, Morris, & and Rob, 2013)

**Overlapping Subtype.** Overlapping subtypes specify a unique entity set that is overlapping. Using the example from a disjoint subtype, an EMPLOYEE supertype would have PILOT, MECHANIC, and ACCOUNTANT subtypes; however in an overlapping subtype, an employee could be both a pilot and mechanic. (Coronel, Morris, & and Rob, 2013)

### **Supertype/Subtype ERD**

**Author Attributes.** The Author attributes are Person\_LastName, Person\_FirstName

**Employee Attributes.** The Employee attributes are Person\_LastName, Person\_FirstName, and Hire\_Date.

**Entity Relationships.** Person – supertype. Employee – subtype. Author – subtype. Author\_Book – neither. Book – neither.

**Relationship Questions.** Can an employee also be an author? Yes. Can a person be neither an employee nor an author? Yes.

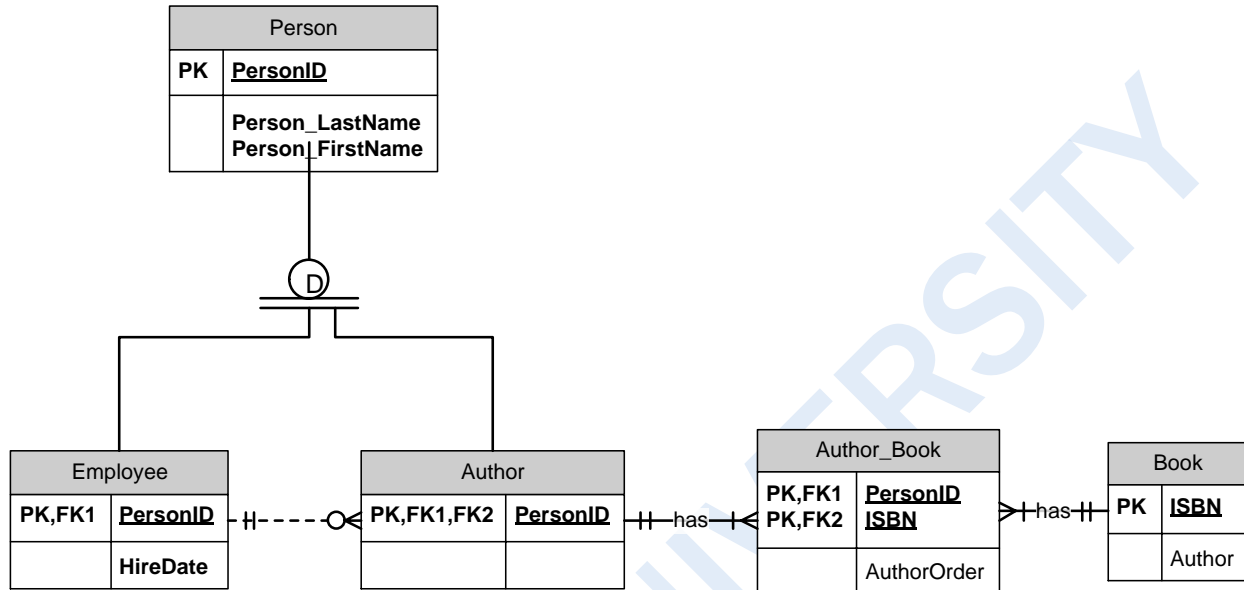
**Employee Manages Author.** The business rules are:

**Employee to Author**

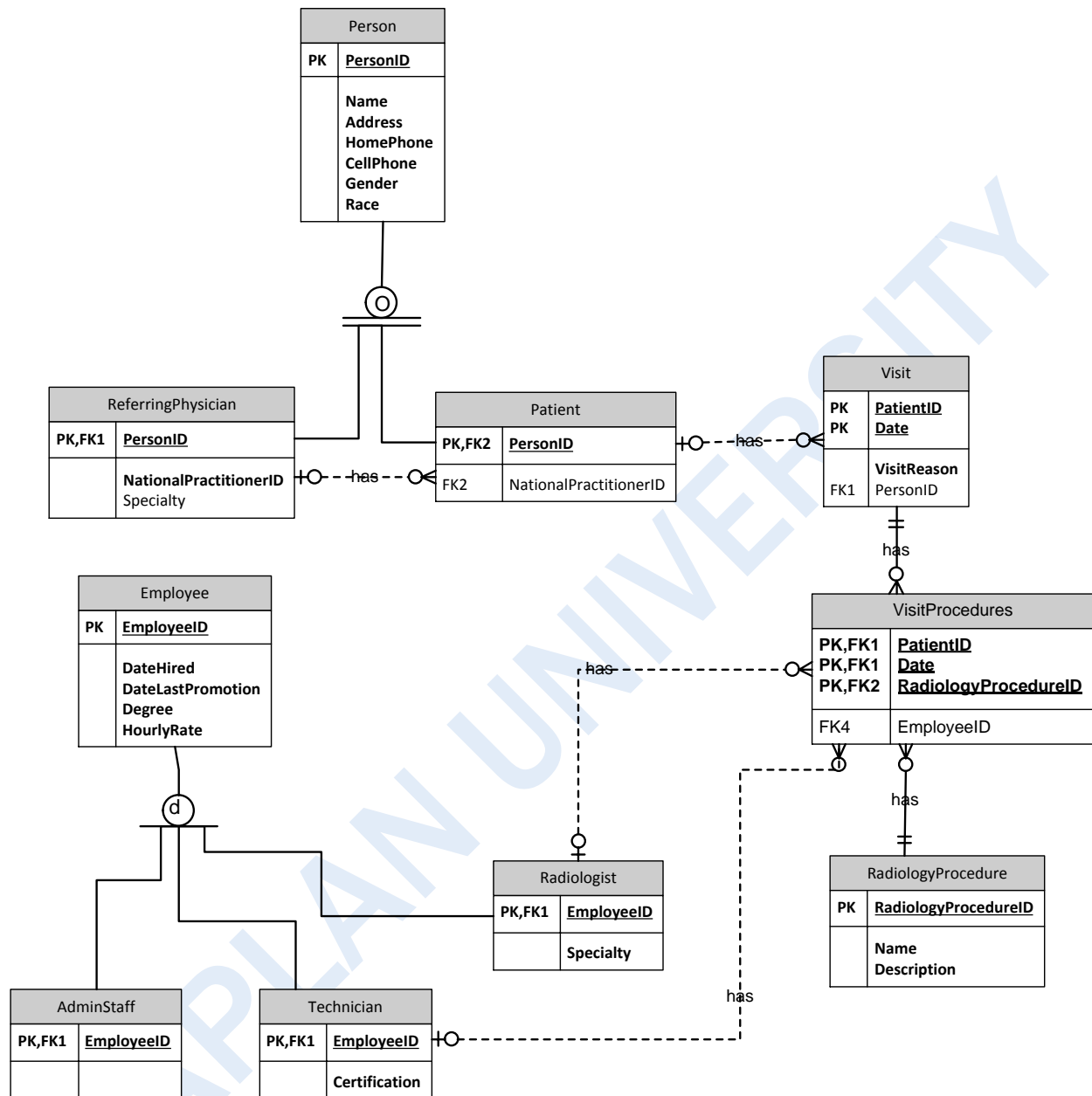
One Employee manages many Author [M]

Many Author are managed by Employee manager [1]

Employee 1:M Author



## Entity Relationship Diagram



### References

- Carnegie Mellon University. (2014). Carnegie Mellon Univ. Dept. of Computer Science 15-415 - Database Applications. Retrieved from <http://www.cs.cmu.edu/~christos/courses/dbms.S12/slides/02ERdiagrams.pdf>
- Coronel, C., Morris, S., & Rob, P. (2012). *Database systems: design, implementation, and management (10th ed.)*. Boston, MA: Cengage Learning.