Cesar Santiago

ID: 970403634

COP4710 Database Systems

Pf. J. Teichert

**Checkpoint 10.1:**

1) A specialization is an entity that inherits attributes from another entity. This makes the place of inheritance in database engineering. An example of specialization is to have a table called Animal, the specialization would come from another table called Cat and another called Dog, which bith take attributes from Animal.

2) A generalization would be the superclass to the specializations' subclass in an inheritance hierarchy. In the previous animal example, Animal would be the generalization to the Dog and the Cat entities.

3) A disjoint constraint means that the generalization can only participate in one specialization at a time. The dijoint symbol is a ‘d’ inside a circle to which the entities connected. The connection between this symbol and the specialization entities have to provide a subset like symbol.

4) The overlapping constraint means that the generalization may participate in more than one specialization. The symbol for this constraint is an ‘o’ encircled and the entities connect to it.

5) When an entity is connected to a constraint and has a subclass symbol it identifies it as one of the specializations to the generalization.

6) Because weak entities do not show the specific relationship that the generalization/specialization (G/S) relationships show. The disjunction and overlap relationships are important when specifying which type of entity a piece of data shows.

7) Inheritance only pertains to the holding of information and the sharing of attributes between a generalization and its specialization.

8) The generalization does not exist without any specialization. Otherwise, it would be a regular entity.

**Checkpoint 10.2:**

1) There is very little difference, except that the mapping rules for the G/S required the use of a relationship table because it can be thought of a Many to Many relationships. However, the weak entitites may be mapped directly to the dependant's table.

2) Firstly, this rule suits disjoint G/S relationships and Fif. 10.3 shows an overlapping relationship. The reason for this is because overlapping relationships require the use of many to many rules of mapping. Rule 17 makes the generalization be its specialization as well.

3) Mapping rule 15 shows the most advantage of mapping specializations that make use of many attributes because multivariable attributes can be mapped through different tables. However, any of the mapping rules can be used provided that the amount of attributes remains reasonable according to the relationship between generalization and specialization.

4) Mapping rule 16 is disadvantageous for overlapping relationships and rule 17 is not usable because it is meant to be used only on disjoint relationships.

5) Overlapping subclasses occur when the entities do not describe a concrete object but instead an abstraction of that object. For example, the athlete and sports database can be overlapping because one athlete may be able to compete in many different sports, here we can see that athlete is a real concrete object where sports are abstract thinking that describes an athlete.

6) Disjoint subclasses should be done when the entity describing the concrete idea are the subclasses and when the superclass describes an abstract idea. For example in the furniture database, furniture is abstract (because furniture does not describe a specific object) but table, chair, and desk are concrete objects.

**Checkpoint 10.3:**

1) Yes, this relationship should be done as a disjoint because a player may be either of the two specifications but because a student cannot be in faculty and because a faculty member may not be a student then the player may not be both.

2) A union is a complete opposite from a disjoint in the sense that it is the subclass can only inherit from one superclass instead of many whereas a disjunction is a superclass that can be interpreted as one subclass and not many.

3) I would make a separate key for identifying which superclass would come to the relationship.

4) I would map the keys the same way as the as the last question. I would make separate kays for each of the superclasses.

5) I think a generalization/specialization should be created when concrete classes depend on abstractions. Categories or Unions should be used when abstractions depend on concrete objects.

6) A shared subclass inherits the attributes of all of the superclasses above it in the hierarchy.

7) A category or union type inherits attributes from all of the superclasses above it in the hierarchy.

8) A shared class is a class that inherits attributes from another class that has inherited attributes, therefore, making it the third class down the hierarchy. A category or union type is a class that inherits attributes from multiple other classes that are in the same level of the hierarchy.