# horizontal lineDatabase System

Project Part 1

I confirm that this is my own work and that use of material from other sources, including the Internet, has been properly and fully acknowledged and referenced.

|  |  |
| --- | --- |
| Name: | Pang, Jinhao |
| Date: | 2022.11.10 |
| NYU ID: | N19475049 |
| Course Section Number: | csci-ga.2433-001 |



**Total in points** (100 points total): \_\_\_\_\_

**Professor’s Comments:**

|  |
| --- |
|  |

Database Systems Project Part I Report

1. **Tools**

Online diagram tools: draw.io. The diagram is represented in HTML, PNG, and draw.io format. The file, whose name is Part1.drawio.png, can be either opened as PNG or dragged into the draw.io platform to check the diagram.

1. **Documentation**

**Customer**: CustomerImage and CustomerAlias use all the keys of Customer as foreign keys. Hence, they can be regarded as weak entities. CustomerAddress and CustomerRelation can be abstracted into high-level concepts as Account, Claim and Associate-h also have similar relationships like that, but here we regard them as normal relationships.

**Claim**: we can consider Coverage\_Claim as a multi-level relationship, which has to contain a number of Participants, related ClaimEvents, and Contract. A triple can be used to record such activities. Here we use a weak entity to describe these specific cases because only these situations occur simultaneously which may become a piece of data. Claimant/Participant can be regarded as weak entity as well. All keys are foreign keys, and the other attributes are foreign keys to distinguish the data with others. This can also be used to design a table to store data in detail.

Diagram

Description automatically generated

Figure 1 Coverage Claim in ER diagram

**Product**: The diagrams are using the notation (A points to B), which means A partially identifies B. In this case, we can consider the flow as a series of weak entities from the Product-h, because each following entity is dependent on the Product. And the descendent entities (from Product-h) use the parent keys as the foreign keys. Then less attributes need to be drawn in the design, shown in Figure 2.

Diagram

Description automatically generated

Figure 2 Reduce the attributes.

**Account:** Similar to Customer, Relation, AccountLegacyAlias, AccountEligibility can be seen as a weak entity, but a little appropriate here to use that. Account\_billingAccount can be regarded this way. The diagram thinks AccountRelation is a weak entity, it may be not precious because the second half of foreign keys cannot be used in the weak entity. Therefore, a normal entity will be enough.

**Associate**: In ManagerContract, we have a relationship with ourselves. Therefore, we design it like Figure 3.

Diagram

Description automatically generated

Figure 3 Relation with self.

**Contract**: We noticed there is a notation with z near the black points, which means A partially identifies B and each A may be one and only one B. Each B must be one and only one A. Therefore, we can total participation somewhere in our diagram. We didn’t use it in the diagram and used weak entities instead in Figure 4. From my perspective, using total participation somewhere will be more precious to describe the relationship.

Diagram

Description automatically generated

Figure 4 Weak Entities instead of Total Participation