University of the Philippines Open University

CMSC206

Assignment 1
ER and relational models

Submitted by:	Angeline Chrisette C. Olegario
Date Submitted:	05/10/2020
Total Number of Pages:	10

1 Exercise 1: modelling

1.1 Problem definition

You have started playing this great new online game called Final Fantasy XIV[1]. You are having great fun with the game as it really appeals to your completionist instinct, as the game has many collectables: *mounts*, *minions* and *achievements* to collect. The game has a lot of content, as it also has four *content packs* (the base game and 3 expansions), each with a name and a release date and each of the collectables belongs to one of these packs.

You started tracking your collection in a notebook, but it's quickly getting messy, so you decide to design a proper database to store information on the collectables in the game. Here is the information you want to store:

- For each pack, its unique name and its release date
- For each mount, its unique name, whether it can fly, what pack it belongs to, and if you have unlocked it
- For each minion, its unique name, if it has a special animation (some can climb on your shoulder!), what pack it belongs to, and if you have unlocked it
- For each achievement, its unique name, the number of achievement points you earn by unlocking it, what pack it belongs to, and if you have unlocked it.

(a) A minion in FFXIV

(b) Mounts in FFXIV

Figure 1: Examples of in-game collectables

1.2 Tasks

1. Build a conceptual model of the the DB you will need using an Entity Relationships diagram.
2. You have played some more and you discovered that sometimes unlocking an achievement automatically unlocks a mount (exciting!). You want to track these cases in your database. Build a modified ER model to support this.

[1] https://en.wikipedia.org/wiki/Final_Fantasy_XIV

3. Your friends in your free company (a group of players in FFXIV) have heard of your great DB and they would like to use it too so they can track the things they have unlocked (but only they can see what they have unlocked). Each of your friends is identified by their unique character name, and they have a password to log into the app you built on top of your DB. Build a modified ER model to support this.
4. Your friends now want to share their collections, but only with people they choose. Build a modified ER model to support this.
5. Build a physical relational model from the last ER model.

ANSWERS FOR EXERCISE 1

1.2.1

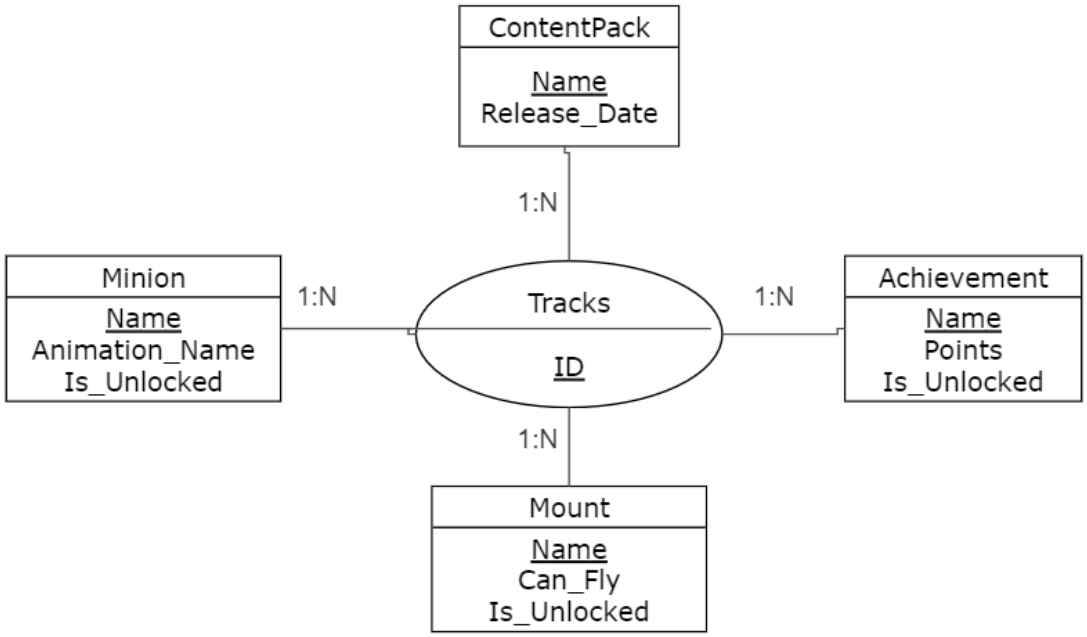

Figure 1 Entity Relationship Diagram for Task 1.2.1

Textual Diagram

ContentPack(Name, Release_Date)
Tracks(ID,ContentPackName,PrizeName)
ContentPackName -> ContentPack, PrizeName -> Minion,Mount,Achievement
Minion(Name, Animation_Name,Is_Unlocked,ContentPackName)
ContentPackName -> ContentPack
Mount(Name, Can_Fly, Is_Unlocked,ContentPackName)
ContentPackName -> ContentPack
Achievement(Name, Points, Is_Unlocked, ContentPackName)
ContentPackName -> ContentPack

1.2.2

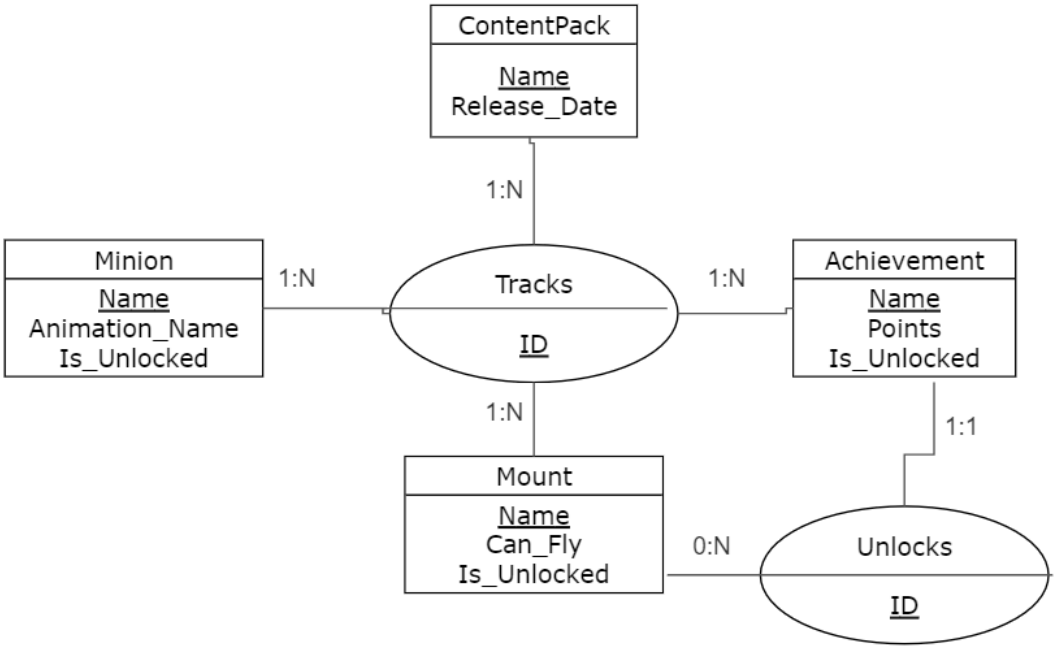

Figure 2 Entity Relationship Diagram for Task 1.2.2

Textual Diagram

ContentPack(Name, Release_Date)
Tracks(ID,ContentPackName,PrizeName)
ContentPackName -> ContentPack, PrizeName -> Minion,Mount,Achievement
Minion(Name, Animation_Name,Is_Unlocked,ContentPackName)
ContentPackName -> ContentPack
Mount(Name, Can_Fly, Is_Unlocked,ContentPackName)
ContentPackName -> ContentPack
Achievement(Name, Points, Is_Unlocked, ContentPackName)
ContentPackName -> ContentPack
Unlocks(ID,AchievementName, MountName)
AchievementName-> Achievement, MountName -> Mount

1.2.3

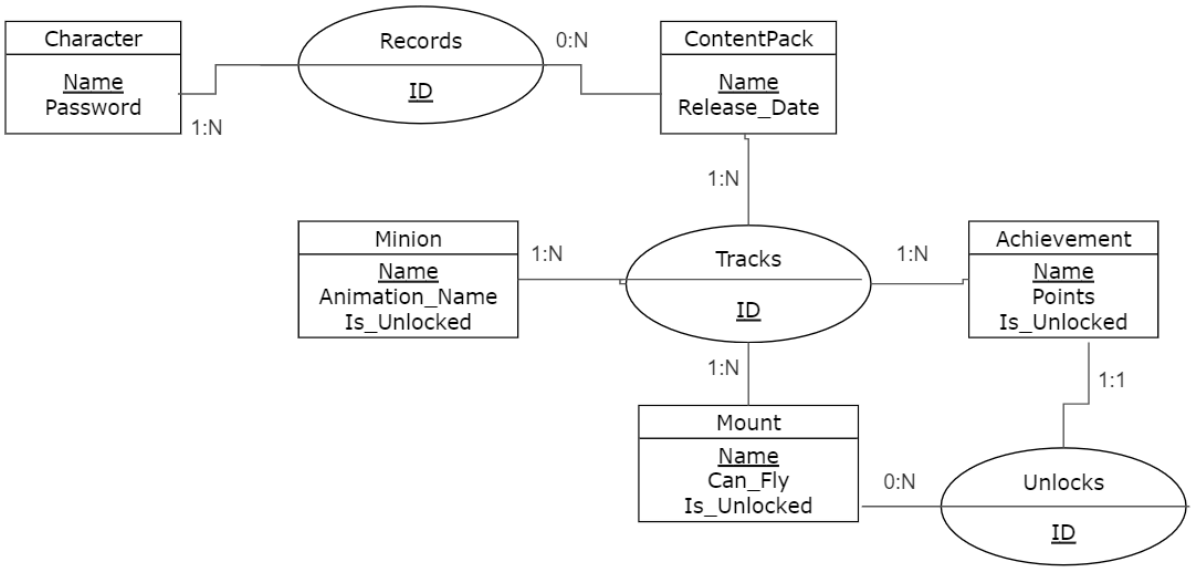

Figure 3 Entity Relationship Diagram for Task 1.2.3

Textual Diagram

Character(Name, Password)
Records(ID, CharacterName, ContentPackName)
CharacterName->Character, ContentPackName -> ContentPack
ContentPack(Name, Release_Date)
Tracks(ID,ContentPackName,PrizeName)
ContentPackName -> ContentPack, PrizeName -> Minion,Mount,Achievement
Minion(Name, Animation_Name,Is_Unlocked,ContentPackName)
ContentPackName -> ContentPack
Mount(Name, Can_Fly, Is_Unlocked,ContentPackName)
ContentPackName -> ContentPack
Achievement(Name, Points, Is_Unlocked, ContentPackName)
ContentPackName -> ContentPack
Unlocks(ID,AchievementName, MountName)
AchievementName-> Achievement, MountName -> Mount

1.2.4

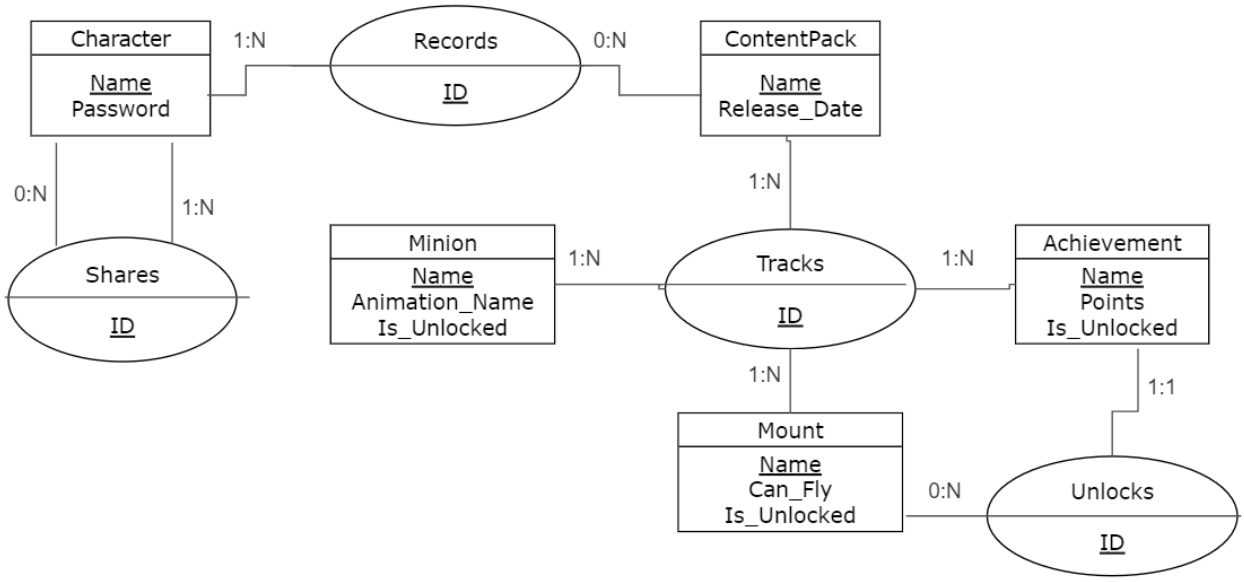

Figure 4: Entity Relationship Diagram for Task 1.2.4

Textual Diagram

Character(Name, Password)
Shares(ID, CharacterName,OtherCharacterName)
CharacterName-> Character, OtherCharacterName -> Character
Records(ID, CharacterName, ContentPackName)
CharacterName->Character, ContentPackName -> ContentPack
ContentPack(Name, Release_Date)
Tracks(ID,ContentPackName,PrizeName)
ContentPackName -> ContentPack, PrizeName -> Minion,Mount,Achievement
Minion(Name, Animation_Name,Is_Unlocked,ContentPackName)
ContentPackName -> ContentPack
Mount(Name, Can_Fly, Is_Unlocked,ContentPackName)
ContentPackName -> ContentPack
Achievement(Name, Points, Is_Unlocked, ContentPackName)
ContentPackName -> ContentPack
Unlocks(ID,AchievementName, MountName)
AchievementName-> Achievement, MountName -> Mount

Physical Diagram

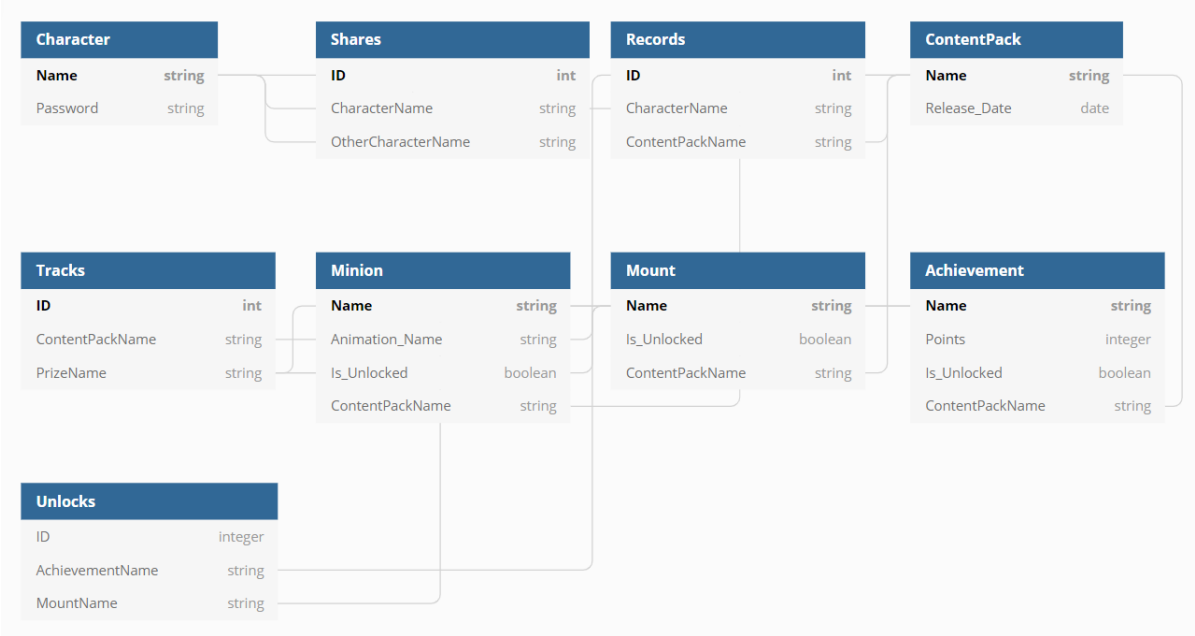

Figure 5 Physical Diagram for figure 4 ERD

END FOR EXERCISE 1

2 Exercise 2: Normalisation

2.1 Problem description

In this exercise we will consider the information system a library uses to keep track of their books. Books have a title, an author, a position (shelf number), and are divided in categories. The library may have multiple copies of the same book but no two books have the same title.

The library's database currently uses the following schema:
Book(Title, Author, Category, Shelf, Category, Copy number)

An extract from the database is shown in table 1:

Title	Author	Category	Shelf	Copy number
The Fellowship of the Ring	Tolkien	Fantasy	F10	1
Ulysses	Joyce	Novel	B50	1
The Hitchhiker's Guide to the Galaxy	Adams	SciFi	H20	1
The Hitchhiker's Guide to the Galaxy	Adams	SciFi	H20	2
Les Miserables	Hugo	Novel	B50	1
Les Miserables	Hugo	Novel	B50	2
Fundamentals of Database systems	Elmasri	TextBook	T30	1

Table 1: Extract from the library's database

2.2 Tasks

1. What are the (non trivial) functional dependencies in this model?
2. Is this model in the third normal form (3NF)?
3. If not, decompose the model so that it is. Is your decomposition in Boyce-Codd normal form (BCNF)?

ANSWERS FOR EXERCISE 2

1. What are the (non trivial) functional dependencies in this model?

   For the schema, Book (Title, Author, Category, Shelf, Copy_Number), the functional dependencies are:

   - Title -> Author, Category
   - Category -> Shelf
   - Copy_Number -> Title

   Assumptions

   - It is assumed that the Title is distinct
   - It is assumed that since Title is distinct, so is the author since there cannot be redundant titles, the name of the author is consistent throughout the database
   - It is assumed that since Title is distinct, so is the category; without the need of the author to be part of a composite key with the title.
   - It is assumed that Category defines Shelf and that Shelf disregards the title data and only cares about the category.
   - It is assumed that Copy_Number is only defined by Title since the rest of the headers are transitively descriptive to the Copy_Number through Title.

2. Is this model in the third normal form (3NF)?

   - No, it is not because transitive dependencies exist within the schema.

3. If not, decompose the model so that it is. Is your decomposition in Boyce-Codd normal form (BCNF)?

   - Book (Title, Copynumber)
     - Title -> Description.Title
   - Description (Title, Author, Category)
     - Category -> Categories.Category
   - Categories (Category, Shelf)

Assumptions

- It is assumed that it is BCNF because the schema was restructured in such a way that all the following superkeys can uniquely identify a row and that there are no more transitive dependencies nor prime attributes.
- It is assumed that it is okay to rename the restructured Database schema to less redundant names (e.g Category schema for Category)

END FOR EXERCISE 2