COMP 421 – DATABASE SYSTEMS

WINTER 2014

G.A.M.E

(**G**AME AND **A**SSOCIATED **M**ECHANDISE **E**-VENDOR)

BY GROUP 42

ANGELA CARRIERO, JEFFREY LEUNG, EMILIE ST-MARTIN

REQUIREMENT ANALYSIS

Application Description

This application will serve as an e-commerce hub where users can purchase video games, the consoles that play them, as well as the accessories attached to them. The formatting of the application is meant for it to also be used as the in-store sales recording system in order to keep accurate records of store inventory. As a bonus to the users, the rating system and order histories can be used to generate product recommendations specifically catered to each user according to their preferences.

Database Description

Entities and their attributes

AccountHolder: an AccountHolder is someone that can place orders and that can give a review of the products. An AccountHolder has a name, a phoneNumber, a billingAddress, a creditCardInfo and an email which is the primary key.
Order: An Order is placed by an AccountHolder and consists of ItemInstances. An Order is identified by an orderNumber (primary key) and is associated with a shippingAddress and a total representing the sum of the prices of the different ItemInstances.
ItemInstance: An AccountHolder buys ItemInstances which are identified by their UPC (primary key). An ItemInstance ISA ConsoleInstance, GameInstance or AccessoryInstance. It also has a price and number of copies associated with it.
ConsoleInstance: a ConsoleInstance is a subcategory of ItemInstance and inherits the UPC (primary key) and price attributes. It is also a weak entity and its owner is Console. A ConsoleInstance also has a colour attribute.
GameInstance: a GameInstance is a subcategory of ItemInstance and inherits a UPC as primary key and a price attribute. It is a weak entity and its owner is Game.
AccessoryInstance: an AccessoryInstance is a subcategory of ItemInstance and inherits a UPC as primary key and a price attribute. It also has a colour attribute. . It is a weak entity and its owner is Accessory.
Game: a Game is associated with a releaseDate, a maturityRating, a gameType and a developer. It is a subcategory of Product so it has an ID as primary key and is also associated with a name. It is the owner of the weak entity GameInstance.
Console: a Console is associated with a brand and a techSpecs along with an inherited ID (primary key) and a name from Product. It is the owner of the weak entity ConsoleInstance.

Accessory: An Anccessory has a brand and a type. It is also a subcategory of Product and has an ID as primary key and a name. It is the owner of the weak entity AccessoryInstance.
Product: A product is identified by an ID which is its primary key and has a name. A Product ISA Game, Console or Accessory. A Product is reviewed by an AccountHolder.
Warehouse: a WareHouse ISA StoreLocation and is identified by a StoreNumber (primary key). It also has an address. A warehouse has ItemInstances in stock.
StoreLocation: a StoreLocation is a subcategory of Warehouse so it inherits its primary key storeNumber and its attribute address. It also has openingHours.

<u>Relationships between Entities</u>

reviews : An *AccountHolder reviews* a product. This is a many-to-many relationship as many *AccountHolders* can *review* many *Products*, and a *Product* can be *reviewed* by many *AccountHolders*. Each *review* is characterized by a score rating given by the *AccountHolder* to indicate satisfaction in the *Product* on a given scale.

purchased: An *AccountHolder purchased* a given *Order*. This is a many-to-one relationship as an *AccountHolder* can have *purchased* many O*rders* but a given *Order* can only be purchased by a single *AccountHolder*. This relationship also contains the attribute quantity to represent the number of copies of a same ItemInstance are included in the Order.

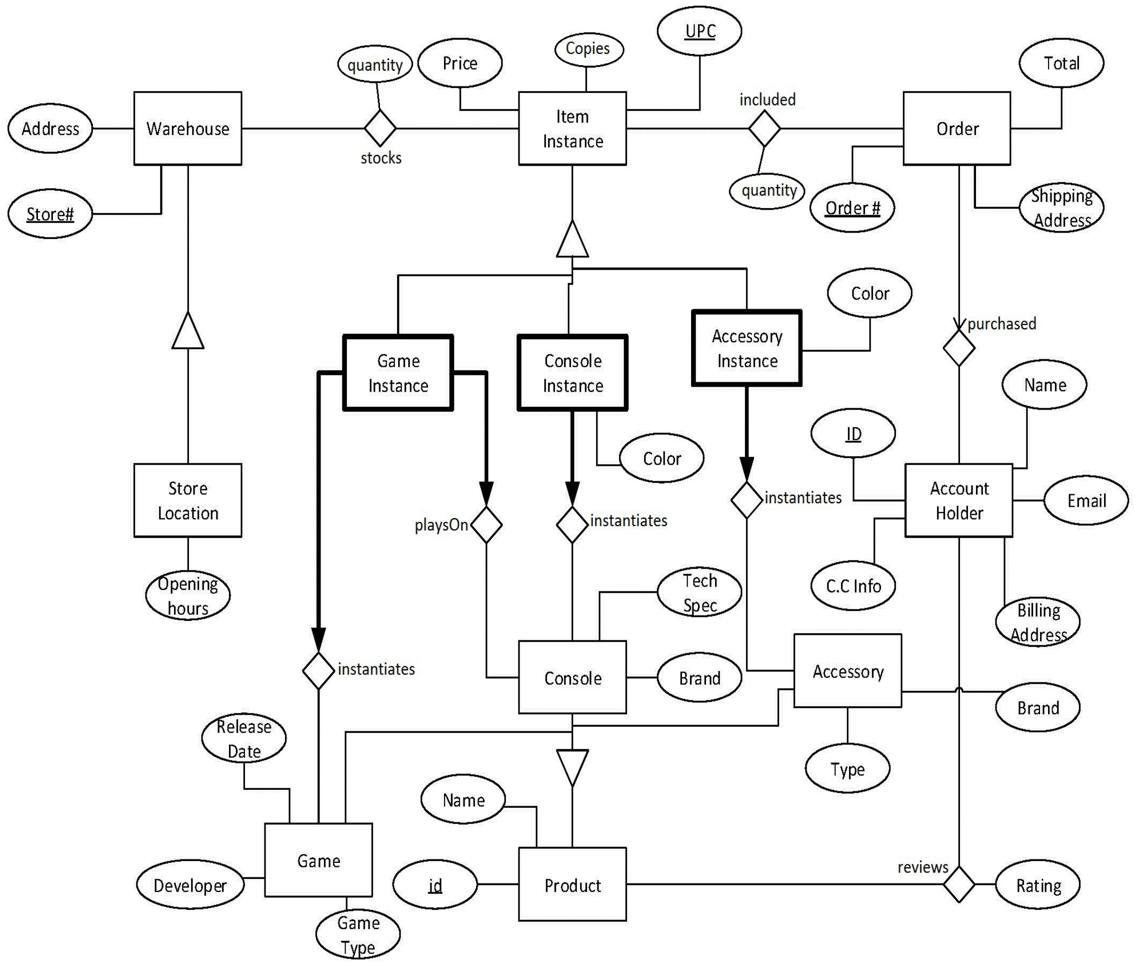
included: An *ItemInstance* is *included* in an Order. This represents a many-to-one relationship as many ItemInstances can be included in an Order but a given ItemInstance can only be included in one Order.

stocks : A Warehouse stocks an ItemInstance. This is a many-to-many relationship as many Warehouses can stock the same ItemInstance and many ItemInstances can be stocked in the same Warehouse. This relationship also carries the quantity attribute to represent how many copies of an ItemInstance the Warehouse has in stock.

instantiates: A GameInstance instantiates a Game, a ConsoleInstance instantiates a Console, and an AccessoryInstance instantiates an Accessory. I each case, this is a many-to-one relationship as an ProductInstance can only instantiate a single corresponding Product, while a Product can have many corresponding ItemInstances.

playsOn: This relationship is between a GameInstance and the console it can be playedOn. This is a many-to-one relationship as a GameInstance can only be played on one Console, while a console can play many GameInstances.

E/R DIAGRAM


quantity
Price
Copies
UPC
Total
Address
Warehouse
stocks
Item Instance
included
Order
quantity
Order #
Shipping Address
Store#
Color
purchased
Game Instance
Console Instance
Accessory Instance
Name
ID
Color
Account Holder
Email
instantiates
playsOn
instantiates
Store Location
Opening hours
Tech Spec
C.C Info
Billing Address
instantiates
Console
Brand
Accessory
Brand
Release Date
Type
Name
Game
reviews
Developer
id
Product
Rating
Game Type

Entities:

Warehouse (Store#, Address)

ItemInstance (UPC, Price)

Order (Order#, Total, Shipping Address)

StoreLocation (Opening hours)

AccountHolder (ID, Name, Email, contact info, Billing Address)

Accessory (Type, Brand)

Console (Tech spec, Brand)

Game (Release Date, Developer, Game Type)

Product (id, Name)

Weak Entities:

GameInstance

ConsoleInstance (Color)

AccessoryInstance (Color)

Relationships:

reviews(rating)

included(quantity)

stocks(quantity)

purchased(Order#,ID)

instantiates(id,UPC)

playsOn(id,UPC)

We feel that there is no way to merge any of the existing entities without sacrificing attributes which identify them. For example, if we merged Game and GameInstance, we would have to make a relationship between console and the Game/GameInstance entity which would cause must repetition for game titles which play on many consoles. Another example would be if we removed the ItemInstance entity. We would then have the same attributes(UPC, copies, price) repeated across multiple entities without them being linked in any way.