Coursework 1: Relational Algebra and Entity-Relationship Diagrams

201374125 *University of Liverpool*

Part A: Relational Algebra

1.

Person (name, age, gender)
Visits (name, restaurant)
Eats (name, dish)
Serves (restaurant, dish, price)

a)

$$\prod_{\text{restaurant}} (\text{Visits} \bowtie (\sigma_{\text{age} < 18} (\text{Person})))$$

b)

$$\prod_{\text{name}} ((\sigma_{\text{gender} = \text{'Female'}} (\text{Person})) \bowtie ((\sigma_{\text{dish} = \text{'Carbonara'}} (\text{Eats})) \cup (\sigma_{\text{dish} = \text{'Bolognese'}} (\text{Eats}))))$$

c)

$$\prod_{\text{name}} ((\sigma_{\text{gender} = \text{'Male'}} (\text{Person})) \bowtie ((\sigma_{\text{dish} = \text{'Pizza'}} (\text{Eats})) \cap (\sigma_{\text{dish} = \text{'Curry'}} (\text{Eats}))))$$

d)

$$\prod_{\text{restaurant}} ((\sigma_{\text{price} < £20} (\text{Serves})) \bowtie (\sigma_{\text{name} = \text{'Tom'}} (\text{Eats})))$$

e)

$$(\prod_{\text{restaurant}} (\text{Visits} \bowtie (\sigma_{\text{gender} = \text{'Male'}} (\text{Person}))) - \prod_{\text{restaurant}} (\text{Visits} \bowtie (\sigma_{\text{gender} = \text{'Female'}} (\text{Person}))))$$

$$\cup$$

$$(\prod_{\text{restaurant}} (\text{Visits} \bowtie (\sigma_{\text{gender} = \text{'Female'}} (\text{Person}))) - \prod_{\text{restaurant}} (\text{Visits} \bowtie (\sigma_{\text{gender} = \text{'Male'}} (\text{Person}))))$$

2.

Academic (aID, name, title, address)
Student (sID, name, course)
Module (mID, name, aID)
Enrolment (mID, sID)

a)

$$\prod_{sID} ((Enrolment \bowtie Module \bowtie \prod_{aID} (Academic)) - (\prod_{aID} (\sigma_{name = 'Mike\ Hodges'} (Academic))))$$

b)

$$\prod_{sID,\ name} (Student \bowtie (\prod_{name} (Student) \cap \prod_{name} (Academic)))$$

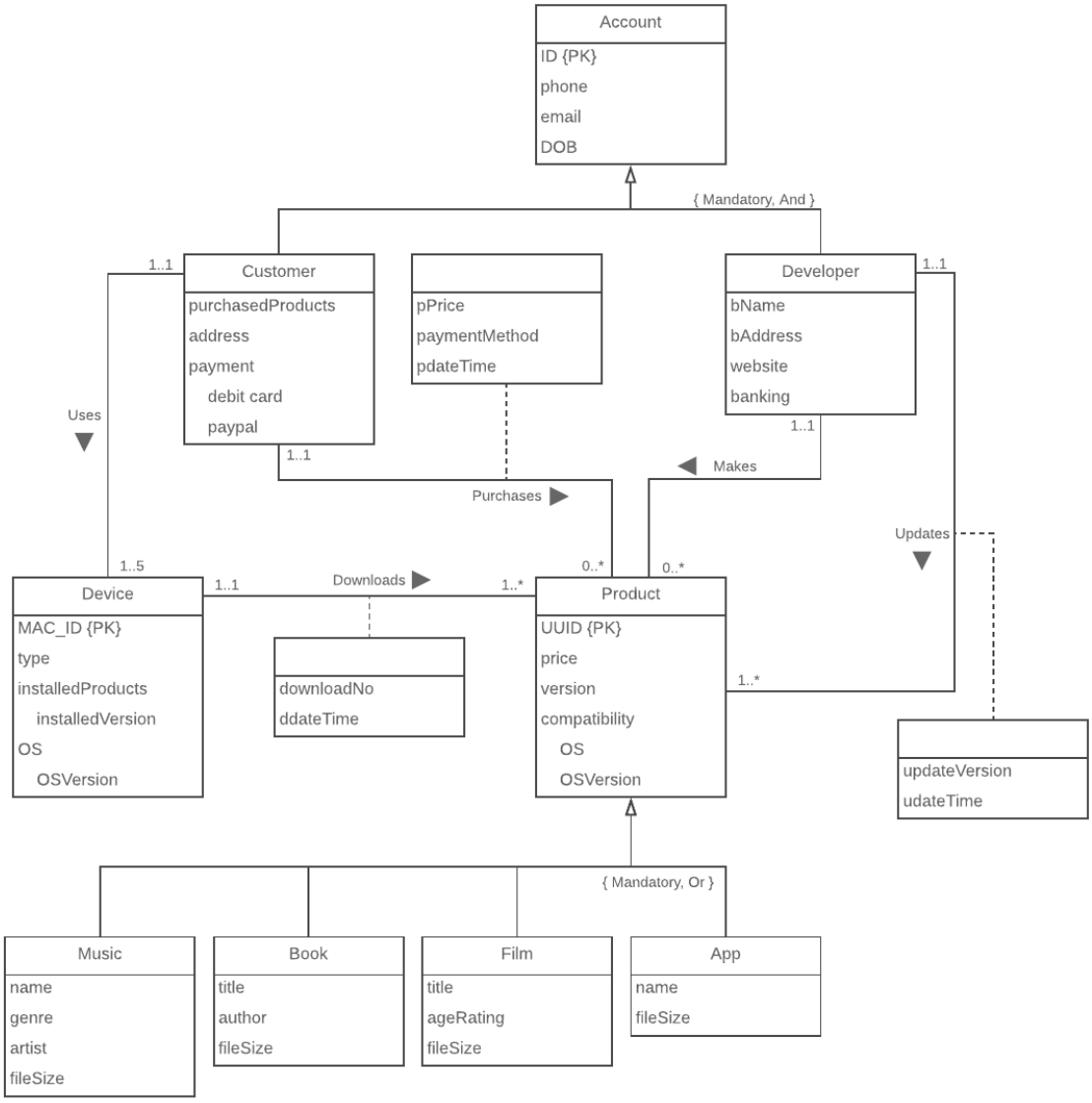
c)

$$\prod_{name} ((Module \bowtie Enrolment) \div (\prod_{sID} (\sigma_{course = 'Computer\ Science'} (Student))))$$

d)

$$\prod_{name} ((Student \bowtie Enrolment) \div (Module \bowtie \prod_{aID} (\sigma_{name = 'Shelia\ Gibney'} (Academic))))$$

Part B: Entity-Relationship Modelling



Assumptions made:

The purchases relationship between Customer and Product has been added to allow the purchase price (pPrice), payment method (pMethod), and time of purchase (pdateTime) attributes to record these details at the time of purchase. The pPrice in Purchases may differ from the price attribute listed in the Product entity as Products may go on sale, or a customer may use a voucher at the time of a purchase. pMethod examples are given as sub attributes in the payment attribute of the Customer entity.

A Customer may have between 1 and 5 Devices associated with their account. This limitation may prevent account sharing to a degree but still allows for a customer to use Products across their different devices.

The Downloads relationship between Device and Product tracks all Products downloaded onto a single Device. Products that are downloaded must have already been purchased from the associated Customer account. A Product may be downloaded to any device associated with a Customer account that has the Product listed under purchasedProducts. Any Device associated with a Customer may download the same product, provided it is compatible. Compatibility is checked against the compatibility attribute in Product with the OS and OSVersion attributes in Device. A product may only be downloaded provided the same product version has not already been downloaded and installed, installedProducts attribute checks what has already been downloaded by UUID (the primary key in Product).

A Developer may have between 0 and any number of Products as a developer may have an account while their first product is still under development.

A Product may only either be a Book, Music, Film or App. I have added fileSize to each of these Subclasses as this information should be indicated when a Product is downloaded onto a Device for any type.

A Developer may update a product and provide a new version number, updates can alter compatibility in Product. For example, an update may work with newer Operating Systems and begin to phase out older ones. Product version is tracked during an update and it must be updated in the Product entity (version). Price may also change if justifiable.

Information not captured:

The type of device such as a phone, tablet, or laptop has been consolidated into one attribute in the Device entity called type. Whether a product is free of paid for will be determined by the price, e.g. £0.00 in the price or pPrice attribute indicates that the product is free.

Not captured by my model is the revenue created through product purchases that will be directed towards the developer. Each individual app will have a total gross payment figure that is determined by the sum of each purchase price (pPrice) when a customer purchases a product. The app store will likely take a cut from each purchase and distribute the funds to the developer 'banking' through a separate process.

Additional information such as song length or app type may be used in the final database but the attributes given are used as an example.

Justification for the design:

Accounts have been split into the overall Account Superclass, and two Subclasses, Customer, and Developer to allow for a combination of the two that can both purchase and develop apps. This is indicated by the mandatory participation constraint paired with the nondisjoint, or "and" constraint.

The customer can associate up to 5 devices with a single account since many people own various types of devices that may support the same products that they purchase through the App store. The limitation is to prevent product sharing from one account between many different people.

Downloads and purchases have been split. This allows for a product to be purchased, then installed at a later date. In addition, it allows for a customer to install the same product onto multiple devices, provided they are compatible. The time of purchase and time of download are both noted.

The updates relationship allows for product updates without creating a new product entirely. Updates may alter product compatibility with devices, and the version number of the product being updated is required to be incrementally increased with each update. While two products of the same version may not be installed onto the same device, the sub attribute installedVersion means that if there is a product version available newer than the current installed version, it may be downloaded.

Products are split into the mandatory disjoint participations for each type which will define different sections in the app store and allow a product to either be an app, film, music or book.