

CS631, Sec #002

Prof. Dimitri Theodoratos

Group Project Phase 2

Group #2

Authors: Zachary Kaplan, Elizabeth Daudelin, John Daudelin

1. Outline the Goals of this Project Phase:

The goal of this phase of the project is to prepare a relational schema of the final database from the EER diagram created in phase 1 of the project. It was also meant to help us gain experience with the EER to Relational mapping algorithm presented in class.

We used our EER diagram from the phase 1 deliverable of the project except for the following changes we made.

- 1) In the HAS_CARD relationship type, we changed the participation of CREDIT_CARD from partial to total.

New EER diagram attached at end.

2. EER to Relational mapping:

Relational schema attached at the end.

Illustration of process as follows:

Step 1) Identified all the regular entity types (not including specialization/generalization entity types or weak types), and these were: CUSTOMER, CREDIT_CARD, and TRANSACTION. Created relations for each of these entity types and added all the attributes for these entity types except for "Total Bill," in the TRANSACTION entity type, since this attribute is calculated.

Step 2) Identified all weak entity types, which were: SHIPPING_ADDRESS. Created a relation for this entity type. Since CUSTOMER is the owner entity type of SHIPPING_ADDRESS, we added the primary key of CUSTOMER (ID) as a foreign key of our new SHIPPING_ADDRESS relation (CID). The new primary key of SHIPPING_ADDRESS is the combination of the partial key of SHIPPING_ADDRESS (Name) and the foreign key CID.

Step 3) Identified all 1:1 relationship types (NONE).

Step 4) Identified all 1:N binary relationship types, which were: HAS_CARD, PAID_WITH, and SHIPS_TO. For HAS_CARD, we took the primary key of CUSTOMER and added it as a foreign key to CREDIT_CARD. For PAID_WITH, we added the primary key of CREDIT_CARD as a

foreign key to TRANSACTION. For SHIPS_TO, we added the primary key from SHIPPING_ADDRESS (CID and Name) to TRANSACTION as foreign keys.

Step 5) Identified all M:N relationship types: SOLD and HAS_IN_BASKET. For SOLD, we made a new relation called SALE with primary key of PRODUCT and primary key of TRANSACTION (combining to form the new primary key of SALE). We also added the single valued attributes Qty and Price from the SOLD relationship type to the new relation. For HAS_IN_BASKET, we made a new relation called BASKET with primary key of CUSTOMER and primary key of PRODUCT (combining to form the new primary key of BASKET). We also added the single valued attribute of Qty.

Step 6) Identified all multivalued attributes (NONE).

Step 7) Identified all N-ary relationship types (NONE).

Step 8) We chose option A for dealing with PRODUCT and its subclasses, and option C for dealing with COMPUTER and its subclasses. For the relation COMPUTER that we created, attribute "Type" refers to either Desktop or Laptop.

Other keys not shown: none.

Constraints not shown:

- i. COMPUTER attribute Type can only be "desktop" or "laptop"
- ii. CUSTOMER attribute Type can only be "regular," "silver," "gold," or "platinum"
- iii. OfferPrice can only be used when the CUSTOMER Type is "gold" or "platinum"
- iv. All Qty attributes must be positive values
- v. MSRP, OfferPrice, and Price must be non-negative values
- vi. All attributes must be non-null except for Weight, Battery, and CPUType in the COMPUTER relation.

3. Difficulties:

Determining which option to use (A, B, C, or D) when mapping the PRODUCT specialization was slightly difficult, and we ended up using a combination of options A and C.