**44-560 Advanced Topics in Database Systems**

**The Grocery Store ER Model**

**Entities and attributes:**

**Store:** For each store in the grocery store chain, we store the store number (unique identifier), the store name, address (street, city, state and zip), floor plan type, the latest remodel date, the date the store originally opened, total square feet, square feet devoted to grocery products, square feet devoted to frozen foods, square feet devoted to meat, telephone, and fax number.

**Product:** For each product, we store the SKU number (unique identifier), the product description, the package height, width, and depth, weight, unit of measurement used for the weight, units per retail case, units per shipping case, number of cases per pallet, the unit cost (amount it cost the store), and unit price (price charged by the store to customers).

**District:** For each district, we store the district id, a description of the district, the number of square miles in the district, and a phone number for the district headquarters.

**Region:** For each region, we store the region id, a description of the region, and a phone number for regional headquarters.

**Brand:** For each brand, we store the brand id and a description of the brand.

**Department:** For each department, we store the department id, a description of the department, and the first and last name of the department manager.

**Transaction:** For each transaction, we store the transaction id, which is unique for a given store, and the date and time of the transaction. (Careful! What does it mean when we say the transaction id is unique for a given store?)

**Relationships:**

A district may have zero or more stores. Each store belongs to exactly one district.

A district is associated with exactly one region. Every region contains at least one district.

Every brand has at least one product. A product is associated with exactly one brand.

Every department has at least one product. A product belongs to exactly one department.

Every store carries at least one product. Each product is carried by zero or more stores. (Note that this relationship will result in an associative entity – call it **Carries**.) For each product carried by a store, we also store the quantity of the product currently on hand at the store.

A transaction includes information about products carried by the store. Every transaction includes at least one product carried by the store. A product carried by a store is included on zero or more transaction. For each product on a transaction, we store the quantity and the total price paid for that product. (Note that this relationship will result in an associative entity – call it **Includes** – between **Carries** and **Transaction**. Also, price paid is not simply the quantity \* unit price, since there may be promotions that reduce the price paid.) Note that you cannot make Includes a ternary relationship between product, store, and transaction, without losing information. If we want to keep track of which products are carried by which store, we must have the relationship Carries, separate from the Includes relationship. We might, for example, have a product carried by a store, but not have any sales for that store, and without the Carries relation we would lose that information.

