CS550-002 Project Assignment
Spring 2017

The project has two parts:

Part1: Database design (25%)

For the Supply Chain Information System (CSIP) described below:

1. Create an ER diagram, and include all integrity constraints that can be specified using database only tools. List constraints that cannot be specified as well as additional assumptions.
2. Translate the ER design into CREATE TABLE definitions using SQL and add CHECK constraints as necessary

General Requirements

Consider the following Supply Chain information system description. The system should support a collaborative supply chain composed of suppliers, manufacturers, shippers and end-customers. Items of different kinds are being moved in the supply chain. Manufacturers use Items of materials to manufacture Items of products for customers. Suppliers supply Items of materials to manufacturers; they also supply Items directly to Customers. Shippers (e.g., UPS, Fedex etc) move items from one business entity (supplier, manufacturer, customer etc) to another.

Items have a unique id and weight. Every business entity (suppliers, manufacturers, customers etc.) is identified by its id, and has a shipping location (to be used by Shippers for shipping orders), address, phone, web location, and contact information.

Specific Requirements

- Every product item (e.g., a table) has a number of associated material/part items in certain quantities necessary to produce 1 unit of the product item. For example, a table product item requires 1 table top item, 4 leg items and 8 screw items.
- Suppliers supply Items using price per unit, which may vary among different Suppliers for the same Item.
- Suppliers apply a volume discount computed on the dollar amount based on price per unit.
- Volume discount is described by a percentage deducted for an amount higher than a predetermined amount.
- Manufacturers produce product Items; this production has an associated setUpCost and product cost per unit.
- Manufacturers may offer volume discounts to customers applied the same way suppliers apply volume discounts.
- Shippers price shipping services per <source, destination> pairs, where sources and destinations are shipping locations of business entities.
- The pricing of each shipper is based on the total weight of shipment from source to destination, using price per lb, and a volume discount applied on the total dollar amount.
- Customers have demand quantity for certain Items.
- The orders are recorded separately for shipping, manufacturing and supply.
- Shipping orders capture information about a shipper, sender, and recipient (who are business entities) and the Item being shipped, and record the quantity of the Item shipped.
- Manufacturing orders capture information about a manufacturer, a manufactured Item and the ordered quantity
- Supply orders capture information about a supplier, Item and the quantity supplied.

Part 2: Database implementation (75%)

A. (5%) Create the following tables (using SQL) representing part of the supply chain information system and populate them with a database instance. For foreign key constraints definition, note that suppliers, manufacturers, shippers, customers, senders and recipients are business entities. Note also that product items, and material items are items. DDL and tested data will be provided, so this part only involves setting up the tables and inserting the provided data. As with previous assignments, be sure that the Parts 2B and 2C are written generically and not just tailored for the provided testbed.

1. items (item, unitWeight)
   - item has unitWeight
2. busEntities(entity, shipLoc, address, phone, web, contact)
   - every business entity has shipLoc (to be used by shipping companies), address, phone, web link, and contact info
3. billOfMaterials(prodItem, matItem, QtyMatPerItem)
   - to produce 1 unit of prodItem, manufacturers need QtyPerItem units of matItems
4. supplierDiscounts(supplier, amt1, disc1, amt2, disc2)
   - supplier gives discount disc1 for purchase amount between amt1 and amt2, and disc2 for purchase amount above amt2.
   - Note that discounts will be expressed as fractions rather then percentage, e.g., 0.15 rather than 15 (%).
5. supplyUnitPricing(supplier, item, ppu)
   - item supplied by sup has ppu (price per unit)
6. manufDiscounts(manuf, amt1, disc1)
   - manufacturer manuf gives discount disc1 for manufacturing cost in excess of amt1 of the base cost, which is computed according to manufUnitPricing table - see below.
7. manufUnitPricing(manuf, prodItem, setUpCost, prodCostPerUnit)
   - For manufacturing of prodItem by manuf, the manufacturer base cost is computed as setUpCost plus the prodPricePerUnit times the qty of the produced prodItem
8. shippingPricing(shipper, fromLoc, toLoc, minPackagePrice, pricePerLb, amt1, disc1, amt2, disc2)
   - The shipping cost for a shipper from fromLoc to toLoc is computed as follows: determine the total weight of all items shipped from fromLoc to toLoc (by all senders at fromLoc to all recipients at toLoc) base cost: is computed based on total weight of shipment and pricePerLb discounted cost: then for amount between amt1 and amt 2, disc 1 is applied; and to the amount above amt2, disc2 is applied total cost: the maximum of inPackagePrice and the discounted price
9. customerDemand(customer, item, qty)
   - The demand by customer is qty units of item; note that items may come from any combination of manufacturers and/or suppliers.
10. supplyOrders(item, supplier, qty)
    - qty units of item were ordered from supplier
11. manufOrders(item, manuf, qty)
    - qty units of item were ordered to be produced by manuf
12. shipOrders(item, shipper, sender, recipient, qty)
    - qty units of item were requested to be shipped by shipper from sender to recipient

B. (35%) Implement the following SQL views (use the given view names):

1. **shippedVsCustDemand:** For every (customer, item) in customerDemand, compute the total qty of this item shipped to this customer, along with the demand qty. Note that the items may come from manufacturers and/or suppliers

2. **totalManufItems:** For every item in manufOrders, compute the total qty of this (product) item produced
3. **matsUsedVsShipped:** For every manuf in manufOrders, and matItem used by this manuf (i.e., manuf ordered a prodItem that requires a matItem according to billOfMaterials) compute:
    a. the total qty of this matItem necessary to produce all the ordered (product) items by this manuf,
    b. the total qty of this matItem shipped (by all shippers) to this manufacturer
4. **producedVsShipped:** For every (item, manuf) in manufOrders compute the total qty of this item shipped out from this manuf (by all shippers to any recipient), along with the total qty of this item produced by this manufacturer (in manufOrders).
5. **suppliedVsShipped:** For every (item, supplier) in supplyOrders compute the total qty of this item shipped from this supplier (by all shippers to any recipient), along with the ordered qty of this item.
6. **perSupplierCost:** For each supplier in supplierDiscounts, compute the total cost of items supplied by this supplier (according to supplyOrders).
7. **perManufCost:** For each manufacturer in manufDiscounts, compute the total manufacturing cost of all items produced by this manufacturer (according to manufOrders).
8. **perShipperCost:** For each shipper in shippingPricing, compute the total shipping cost of this shipper.
9. **totalCostBreakdown:** Compute the total supply cost, manufacturing cost, shipping cost, and the overall cost.

C. (35%) Using the defined SQL views, implement the following SQL queries:

1. Find customers, whose demand is NOT satisfied, i.e., are not shipped all the quantities of items
2. Find suppliers, whose orders (in supplyOrders table) are not fully shipped out.
3. Find manufacturers who do NOT have enough materials to produce ordered product quantities, i.e., not enough materials were shipped to them
4. Find manufacturers whose orders are not fully shipped out

What and When to Submit

1. A cover sheet in pdf format containing your name, the course, the title “Semester Project”, and a list of parts you’re submitting (e.g., Part 1, Part 2a, Part 2b, and/or Part 2c)

2. For Part 1: (a) a pdf with the ERD and list of assumptions and constraints not covered in the ERD and (b) an sql file with the create table statements

3. For Part 2, submit three separate sql files- one for each part (A, B, and C). A ddl file will be provided that includes testbed data. As with previous assignments, be sure that Parts 2B and 2C are created to handle general data and not just tailored for the provided testbed.

4. The project is due Thursday May 4th at 11:59 pm. Late projects will be accepted up to May 8th at 11:59 pm, but will be assessed a late penalty of 5% off the score for each of the first two late days, and 10% for each of the second two late days. For example, submitting May 5th means the maximum score is 95%, May 6th is 90%, May 7th is 80%, and May 8th is 70%. Submissions after May 8th will not be accepted.

5. You have the option to submit only parts of the project – specifically any combination of Parts 1, 2A, 2B, or 2C. Note that Part 2B depends on completing 2A, and 2C depends on completing 2B. If you elect to do that the maximum score is limited to the parts you submit. For example, if you submit all parts for Part 2, your maximum score is 75%. If you submit all parts for Part 2 on May 5th, the max is 71.25% (.95 X 75%). Parts submitted must be listed on your cover sheet. (Course assignments are weighted as 35% of total course grade: 1 through 5 contribute 5% each, the project is 10% for a total of 35%.)