# *DATABASE MANAGENT PROJECT*

PART 1: Database Design

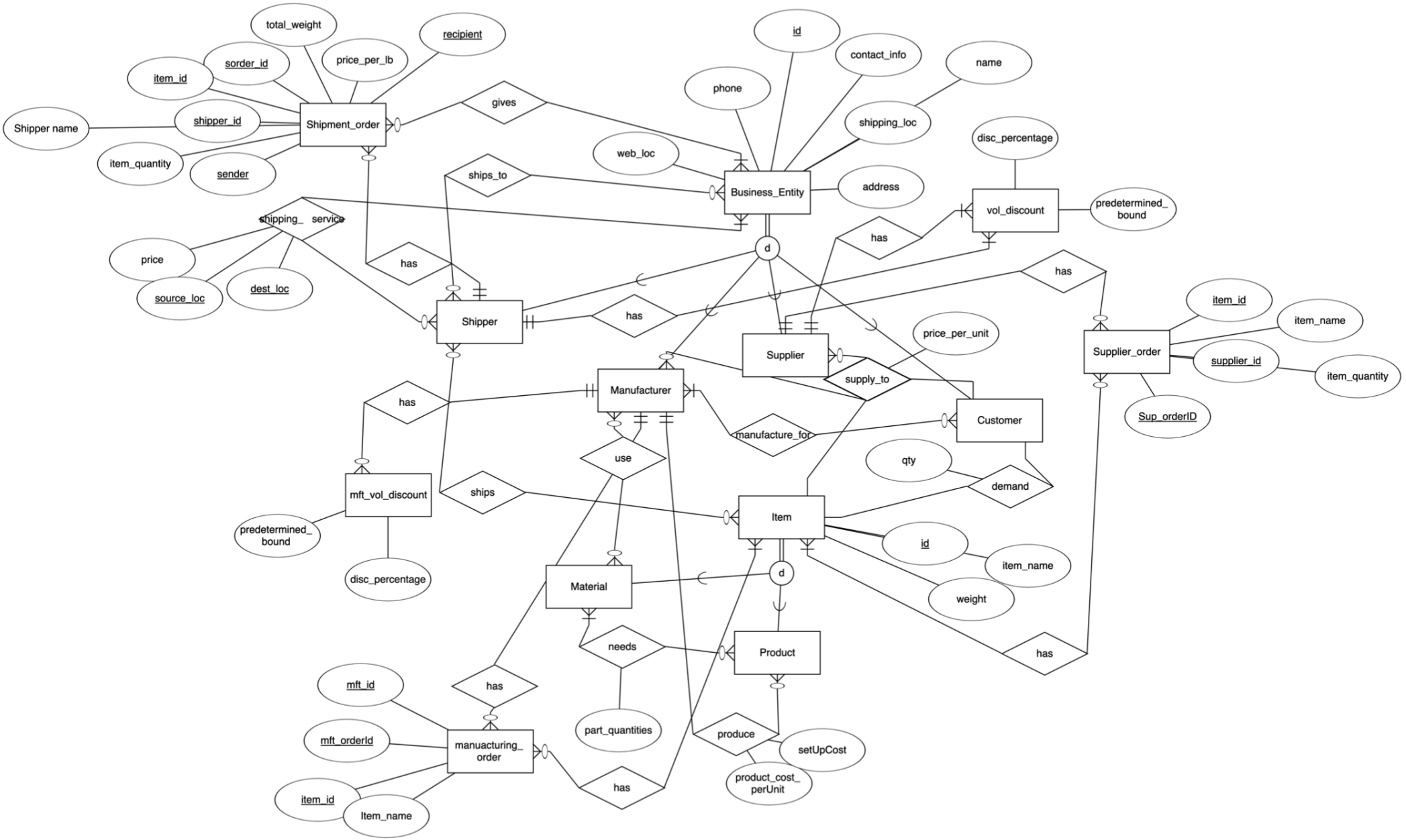
*Fatima Cheema*

*CS550-DL1 | HW 3(HA3)*

## Part 1: Database Design Instructions

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.
- Every product item (e.g., a table) has a number of associated material/part items in certain quantities necessary to produce 1unit of the product item. For example, a table product item, requires 1 tabletop item, 4 leg items, and 8screw items. Suppliers supply Items, using price per unit, which may vary among different Suppliers for the same Item.
- Suppliers have volume discount applied on the dollar amount computed based on price per unit.
- Volume discount is described by a percentage of deduction for amount above a predetermined bound.
- Manufacturers produce product Items; this production has an associated setup Cost 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 pairs of (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; and
    - Supply orders capture information about a supplier, Item and the quantity supplied.




The ER-diagram specifies integrity constraints and doesn't reflect the information to calclute the volume discount accurately.

```
CREATE TABLE Business_Entity (id int not null PRIMARY Key, contact_info varchar (50), shipping_loc
varchar(50), address varchar(50), name varchar(50), phone varchar(15), web_loc varchar(50));

CREATE TABLE Shipper (shipper_id int not null PRIMARY Key, FOREIGN Key(shipper_id) REFERENCES
Business_Entity(id) on DELETE cascade);

CREATE TABLE Customer (customer_id int not null PRIMARY Key, FOREIGN Key(customer_id) REFERENCES
Business_Entity(id) on DELETE cascade);

CREATE TABLE Supplier (supplier_id int not null PRIMARY Key, FOREIGN Key(supplier_id) REFERENCES
Business_Entity(id) on DELETE cascade);

CREATE TABLE Manufacturer (mft_id int not null PRIMARY Key, FOREIGN Key(mft_id) REFERENCES
Business_Entity(id) on DELETE cascade);

CREATE TABLE item (item_id int not null PRIMARY Key,  item_weight varchar(50));

CREATE TABLE Product(product_id int not null PRIMARY Key, PRODUCT_NAME VARCHAR(50), FOREIGN
Key(product_id) REFERENCES item(item_id) ON DELETE CASCADE);

CREATE TABLE MATERIAL(MATERIAL_id int not null PRIMARY Key, MATERIAL_NAME VARCHAR(50), FOREIGN
Key(MATERIAL_id) REFERENCES item(item_id) ON DELETE CASCADE);

CREATE TABLE MANUFACTURE_FOR_CSR(mft_id INT, customer_id INT, MFT_PRODUCT VARCHAR(50), PRIMARY
KEY(mft_id, customer_id), FOREIGN KEY(mft_id) REFERENCES Manufacturer(mft_id), FOREIGN KEY(customer_id)
REFERENCES Customer(customer_id));

CREATE TABLE PROD_NEEDS_MATERIAL(MATERIAL_id int, PRODUCT_ID INT, PART_QUANTITITES INT, PRIMARY
KEY(PRODUCT_ID, material_id), FOREIGN Key(MATERIAL_id) REFERENCES MATERIAL(material_id) , FOREIGN
Key(PRODUCT_id) REFERENCES Product(product_id));

CREATE TABLE SUPPLY_TO_MANUFACTURER(supplier_id INT, MFT_ID INT, material_id INT, price_per_unit
FLOAT(40), FOREIGN KEY(supplier_id) REFERENCES Supplier(supplier_id), FOREIGN KEY(MFT_id) REFERENCES
Manufacturer(MFT_id), FOREIGN KEY(material_id) REFERENCES MATERIAL(material_id));

CREATE TABLE SUPPLY_TO_CUSTOMER(supplier_id INT, CUSTOMER_ID INT, material_id INT, PRODUCT_ID INT,
price_per_unit FLOAT(40), FOREIGN KEY(supplier_id) REFERENCES Supplier(supplier_id), FOREIGN
KEY(customer_id) REFERENCES Customer(customer_id), FOREIGN KEY(product_id) REFERENCES
Product(product_id), FOREIGN KEY(material_id) REFERENCES MATERIAL(material_id));

CREATE TABLE SUPPLY_UNIT_PRICING(supplier_id INT, item_id INT, PRICE_PER_UNIT FLOAT(30), FOREIGN
KEY(supplier_id) REFERENCES Supplier(supplier_id), FOREIGN KEY(item_id) REFERENCES item(ITEM_ID));

CREATE TABLE SUPPLIER_DISCOUNT( supplier_id INT PRIMARY KEY, DISCOUNT_PERCENTAGE FLOAT(10),
PREDETERMINED_PRICE_BOUND FLOAT(10), FOREIGN KEY(supplier_id) REFERENCES Supplier(supplier_id));

CREATE TABLE MANUF_PRODUCE( product_id INT, MANUFACTURER_ID INT, SET_UP_COST FLOAT(50),
PRODUCT_COST_PER_UNIT FLOAT(50), PRIMARY KEY(product_id, Manufacturer_ID), FOREIGN KEY(product_id)
REFERENCES Product(product_id), FOREIGN KEY(Manufacturer_ID) REFERENCES Manufacturer(mft_id));

CREATE TABLE MANUFACTURER_DISCOUNT( MFT_id INT PRIMARY KEY, DISCOUNT_PERCENTAGE FLOAT(10),
PREDETERMINED_PRICE_BOUND FLOAT(10), FOREIGN KEY(MFT_id) REFERENCES Manufacturer(MFT_id));

create TABLE SHIP_SERVICE_PRICING(shipper_id INT, DESTINATION_ID INT, SOURCE_ID INT,
DESTINATION_LOCATION VARCHAR(50), SOURCE_LOCATION VARCHAR(50), MIN_PACKAGE_PRICE FLOAT(10),
PRICE_PER_LB FLOAT(10), AMT1 FLOAT(10), DISC1 FLOAT(10), AMT2 FLOAT(10), DISC2 FLOAT(10),  PRIMARY
KEY(shipper_id, SOURCE_ID, DESTINATION_ID), FOREIGN KEY(shipper_id) REFERENCES Shipper(shipper_id),
FOREIGN KEY(SOURCE_ID) REFERENCES Business_Entity(id), FOREIGN KEY(DESTINATION_ID) REFERENCES
Business_Entity(ID));

CREATE TABLE SHIPPER_DISCOUNT( shipper_id INT PRIMARY KEY, DISCOUNT_PERCENTAGE FLOAT(10),
PREDETERMINED_PRICE_BOUND FLOAT(10), FOREIGN KEY(shipper_id_id) REFERENCES Shipper(shipper_id));

CREATE TABLE CUSTOMER_DEMAND(customer_id INT, item_id INT, ITEM_QUANTITY INT, PRIMARY KEY(customer_id,
item_id), FOREIGN KEY(customer_id) REFERENCES Customer(customer_id), FOREIGN KEY(item_id) REFERENCES
item(item_id));

CREATE TABLE SHIP_ORDER (ORDER_ID INT ,SHIPPER_ID INT, SHIPPER_NAME VARCHAR(50), ITEM_ID INT,SENDER_ID
INT, RECIPIENT_ID INT,item_quantity INT, ,  price_per_lb FLOAT(10), TOTAL_WEIGHT FLOAT(10), PRIMARY
KEY( ORDER_ID, shipper_id, ITEM_ID, SENDER_ID, RECIPIENT_ID), FOREIGN KEY(shipper_id) REFERENCES
Shipper(shipper_id), FOREIGN KEY(item_id) REFERENCES item(ITEM_ID), FOREIGN KEY(SENDER_ID) REFERENCES
Business_Entity(id), FOREIGN KEY(RECIPIENT_ID) REFERENCES Business_Entity(id));

CREATE TABLE MANUF_ORDER (ORDER_ID INT , MFT_id INT, ITEM_ID INT, item_quantity INT, PRIMARY KEY(
ORDER_ID, MFT_id, ITEM_ID), FOREIGN KEY(MFT_id) REFERENCES Manufacturer(MFT_id), FOREIGN KEY(item_id)
REFERENCES item(ITEM_ID));
```

```
CREATE TABLE SUPPLIER_ORDER (ORDER_ID INT , supplier_id INT, ITEM_ID INT, item_quantity INT, PRIMARY
KEY( ORDER_ID, supplier_id, ITEM_ID), FOREIGN KEY(supplier_id) REFERENCES Supplier(supplier_id),
FOREIGN KEY(item_id) REFERENCES item(ITEM_ID));
```