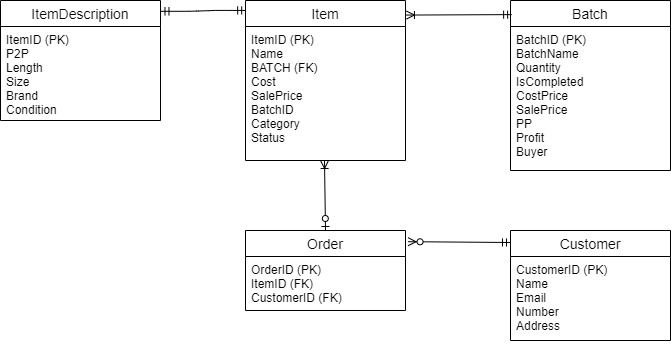
**532 Vintage Database**

**UML**

Explanation

1. Tables
   1. Batch – Details about batches we purchased, and we will be selling individually.
   2. Customer – Details about customers that have signed up to our website
   3. Item – Details about items from the batches we have purchased that we will be selling in relation to its batch
   4. ItemDescription – Item-related details about the items we will be selling.
   5. Order – Details about the Customer orders from our website
2. Attributes
   1. Batch
      1. BatchID – ID number for our batches, primary key and unique.
      2. BatchName – The name of the seller we purchased from.
      3. Quantity – Number of items in the batch.
      4. IsCompleted – Boolean implemented into an integer value; 1 = True and 0 = False.
      5. CostPrice – Cost of how much the batch cost in dollar value.
      6. SalePrice – Cost of how much we sold items in the batch for in dollar value.
      7. PP – Original price per piece in dollar value (Cost/Quantity).
      8. Profit – Net value in dollar value (SalePrice - CostPrice).
      9. Buyer – Who purchased the batch.
   2. Customer
      1. CustomerID – ID number for customers who have signed up to our website, primary key and unique
      2. Name – Customer’s name
      3. Email – Customer’s Email address
      4. Number – Customer’s contact number
      5. Address - Customer’s living address
   3. Item
      1. ItemID - ID number for items we will be selling, primary key and unique
      2. Name – Item’s name
      3. BATCH – Batch name which the Item is part of
      4. Cost – How much the item cost, but better denoted in the Batch Table.
      5. SalePrice – How much an individual item sold for.
      6. BatchID – Foreign key, denotes which batch the Item is part of.
      7. Category – Type of product: E.g. Shirts, Hoodies, etc.
      8. Status – Denotes which products have been uploaded onto Social Media, sold and are available.
   4. ItemDescription
      1. ItemID- Foreign key, denotes which item is being described.
      2. P2P – Measurement of the item from armpit to armpit in inches.
      3. Length – Measurement of the item from top of the shoulder to the bottom in inches.
      4. Size – Size on the tag or what it fits like
      5. Brand – Brand name on the tag
      6. Condition – Denotes any faults within the item. E.g. holes, rips stains, etc.
   5. Order
      1. OrderID – ID order for orders on our website, primary key and unique.
      2. ItemID – Foreign key, denotes which Item was part of the order.
      3. CustomerID - Foreign key, denotes which customer ordered the item(s).
3. Relationships
   1. Item & Item Description
      1. To every item there is must be only one set of descriptions.
   2. Item & Batch
      1. An item will only come from one batch.
      2. A batch will have one to many items.
   3. Item & Order
      1. An order may have many items.
      2. An item may have only one order.
   4. Order & Customer
      1. A customer may have many orders.
      2. An order may only have one customer.

**Views**

1. finishedbatches  
     
   Code:  
     
   SELECT (sum(i.saleprice) - sum(i.cost)) AS Net, sum(i.cost) AS Cost, sum(i.saleprice) AS Sale, count(i.itemid) As 'Sold Pieces'  
   from Batch b, item i  
   where b.batchid = i.batchid  
   and b.IsCompleted = 1  
   and i.status = 'Sold';  
     
   Explanation:  
     
   These are all of the batches that we have completed into a table. We would like to view how well we are doing in terms of sales. We only record batches that we have sold altogether. This is because, it is hard to predict how much profit a batch will do.
2. soldinbatch  
     
   Code:  
     
   SELECT b.BatchName, count(i.name) as "Is Sold", b.Quantity, b.quantity - count(i.name) as "Remaining", pp as "Original PP", round(((b.costprice - b.saleprice)/(b.quantity - count(i.name))),2) as "New PP", Profit, b.IsCompleted, b.buyer  
   FROM item i, batch b  
   WHERE i.BatchID = b.BatchID  
   AND i.saleprice > 0  
   group by i.batchid  
   order by b.IsCompleted desc, profit desc, "New PP" desc;  
     
   Explanation:  
     
   An table on products that have sold at least one piece. We believe that being having an update on the cost per piece as we sell products allows us more pricing flexibility.

**Triggers**

1. getperpiece  
     
   Code:  
     
   CREATE TRIGGER getperpiece  
   after update on batch  
   begin  
   update batch  
   set pp = round(costprice/quantity) + 1;  
   end  
     
   Explanation:   
     
   A trigger that sets the cost per piece + 1 once the cost and quantity have been entered. The ‘+1’ has been entered to get the ceiling value.
2. getprofit  
     
   Code:  
     
   CREATE TRIGGER getprofit  
   after update on batch  
   begin  
   update batch  
   set Profit = round(SalePrice - CostPrice, 2);  
   end  
     
   Explanation:  
     
   A trigger that updates the profit on the batch table once a SalePrice on the item table entry is updated.
3. updatesaleprice  
     
   Code:  
     
   CREATE TRIGGER updatesaleprice  
   after update on item  
   begin  
   update batch  
   SET saleprice =  
   (SELECT SUM(SALEPRICE)  
   FROM ITEM  
   WHERE Item.BatchID = Batch.BatchID   
   GROUP BY BatchID);  
   END  
     
   Explanation:  
     
   A trigger that updates the SalePrice on the batch table once a SalePrice on the item table entry is updated.