Part 2: Database Design

Gaps:

1. How actually should the bundle work? Only as the mapping or as a real product?

   Suppose we have a bundle named Back to School with products such as a school bag, geometric box, pen, pencil, eraser set.

   Then Back to School should only act as a viewing source or a separate product, which can be purchased by the user?
2. Good to have an InventoryHistory table to record and analyze how actually the stocks have updated and changed.
3. Does a single product is supplied by a single supplier or multiple suppliers?
4. As it is a B2B business, is there some kind of minimum order criteria for the company for the products?

Relations

Here we have 3 base tables Supplier, Company and product.
Base tables are the independent tables which represent the complete entity.

1. Supplier
    a. id – Primary Key, String(255) UUID
    b. name – String(3, 100) not null
    c. address – String(10, 100) not null
    d. contact_no – String(13) not null
    e. registered_email – String(10, 100) not null

2. Company
    a. id – Primary Key, String(255) UUID
    b. name – String(3, 100) not null
    c. address – String(10, 100) not null
    d. contact_no – String(13) not null
    e. registered_email – String(5, 50) not null

3. Product
    a. id – Primary Key, String(255) UUID
    b. company_id – Foreign Key -> Company.id and not null
    c. sku – String(5, 100) – Not Null
    d. price – Number(10, 2) not null
    e. name – String(5, 100) not null
    f. description – Text not null
    g. created_at – DateTime Default Current Time Stamp
    h. updated_at – DateTime Default Current Time Stamp
    i. is_bundle boolean Default false
    j. Unique Constrain (sku, company_id) - Each company haves unique SKU’s

This relation is used to manage the Bundled/grouped items.
A grouped item is treated as a single product.

4. BundleItem
    a. bundle_id – Foreign Key → Product.id not null
    b. product_id – Foreign Key → Product.id not null
    c. quantity – Number(10), default 1
    d. Primary Key - (bundle, product)

Product supplier is a Junction table which is used to handle the Many to Many relation among the supplier and product.

It is used to manage ‘single product can be purchased from the multiple suppliers’

5. ProductSupplier
    a. id – Primary Key, UUID
    b. product_id – Foreign Key → Product.id not null
    c. supplier_id – Foreign Key → Supplier.id not null
    d. created_at – DateTime Default Current Time Stamp

Warehouse, Inventory and InventoryHistory tables are used to keep track of the products in each warehouse.

A single company can have multiple warehouses.

Each warehouse can have a separate Inventory and also we can track the inventory history.

6. Warehouse
    a. id – Primary Key, String(255) UUID
    b. name – String(10, 100) not null
    c. company_id – Foreign Key → company.id not null

7. Inventory
    a. id – Primary Key, String(255) UUID
    b. product_id – Foreign Key → Product.id not null
    c. warehouse_id – Foreign Key → Warehouse.id not null
    d. quantity – Number(5) not null
    e. threshold – Number(5) not null
    f. created_at – DateTime Default Current Time Stamp
    g. updated_at – DateTime Default Current Time Stamp
    h. Unique Key → (product, warehouse)

8. InventoryHistory
    a. id – Primary Key, String(255) UUID
    b. inventory_id – Foreign Key → Inventory.id not null
    c. change_type – Enum : (ADD, REMOVE, RETURN,CANCEL) not null
    d. quantity_changed – Number(5) not null
    e. previous_quantity – Number(5) not null
    f. new_quantity – Number(5) not null
    g. note – Text - Optional
    h. created_at – DateTime Default Current Time Stamp

B2B Order Processing

PurchaseOrder relation takes complete order details, while PurchasedOrderItems contains details of each product in an order.

9. PurchaseOrder
    a. Id - Primary Key, String(255) UUID
    b. supplier_id - Foreign Key : Supplier.id
    c. company_id - Foreign Key : Company.id
    d. status - Enum (PLACED, APPROVED, SHIPPED, DELIVERED)
    e. purchase_date - DateTime Default Current Time Stamp
    f. updated_at - DateTime Default Current Time Stamp

10. PurchasedOrderItem
    a. id - Primary Key, String(255) UUID
    b. order_id - Foreign Key : PurchaseOrder.id not null
    c. product_id - Foreign Key : Product.id not null
    d. warehouse_id - Foreign Key : Warehouse.id not null
    e. quantity - Number(5) not null
    f. is_bundled - boolean Default false
    g. price_per_unit - Number(10, 2)
    h. received_quantity - Actual received quantity updated when order is received by customer.

Index Explanation:

1. For product relation we have a composite unique key constraint on (sku, company_id) which will help to have unique SKUs for each company.
2. We can have a separate index for product name as many times a product is searched using its name.
3. Find stock for a product in a warehouse is also a very frequently used query. As are already having a composite unique index for it on Inventory(product_id, warehouse_id)
4. Apart from indexing we for getting the complete product information we can even consider the views.