Carpetero database design document

Introduction

CARPETERO is a carpet distributor company. They always restock by batches of 100x100m (i.e. 100 sqm) rolls of carpet and deliver by cutting a rectangular area (e.g. 30x20m) of carpet from one of their inventory carpet rolls, and store the remaining carpet area for future cuts. The company prefers cutting from an already cut carpet roll. They have 4 carpet types: Wool, Nylon, Fiber and Acrylic. Width and Length are always specified in meters. Carpet orders from customers are always rectangular meters.

Requirement

Need to design a database schema that can track CARPETERO's inventory which can satisfy the below use cases
1. Restock carpet rolls (i.e. Add a stock of 100x100m roll of carpet to inventory)
2. Cut from a roll of carpet (i.e. Remove an area of size MxN meters from a roll of carpet in the inventory)
3. Select which roll of carpet to cut from.

Database design

Keeping all the use cases in mind I have design the below database schema

Database Name : **Carpeto**

Table List:

1. Company
2. CarpetType
3. Carpet
4. CarpetRoll
5. CarpetInventory
6. PurchaseOrder
7. PurchaseOrderCarpet
8. Customer
9. Order
10. Orderdetails

Table Details

Company

1. **Introduction**: This table is a base table which will be used to store company data.
2. **Entity List**:
   a. name
   b. email
   c. phone_number
   d. gst_number
   e. status

CarpetType

1. **Introduction**: This table will be used to store the different carpet type (Wool, Nylon, Fiber and Acrylic)
2. **Entity List**:
   a. carpet_type
   b. status

Carpet

1. **Introduction**: This table will be used to store the carpet base data. All the different type of carpet will be stored in different rows and if each row will have the overall sum of roll dimension available for the particular carpet row. Once there is some deduction from the roll this table dimension will be updated. If there is a new purchase for the existing carpet this table dimension will be updated.
2. **Entity List**:
   a. name
   b. carpet_type (Foreign key of carpet type)
   c. company_id (Foreign key of company)
   d. available_carpet
   e. colour
   f. in_stock
   g. status

CarpetRoll

1. Introduction: This table is a sub table of carpet. Since company will be purchasing the carpet roll in 100*100 dimension and all the sales will be happening in m*n dimension so keeping that in mind this table is one which will be updated for any new roll addition and any sales happen. This table consists of four entities (roll_r1_width, roll_r1_length, roll_r2_width, roll_r2_length) which will be updated in every cut and the table main logic will also work with the table. Roll cutting method which I have thought about behind this table is explained below.

Step 1 : Once roll is purchased it will be added to carpet roll table with the dimension (100*100)

L	W
100	100

Step 2 : Cutting 20 * 30 dimension (Since the company got the order for 20 * 30 dimension , will first check the oldest carpet which will have the required dimension and will deduct the required dimension and will deduct 20 from 100 and update 80(100-20=80) in the first row and second row will save 20 in r2 length. r1_width will have 100 dimensions and r2_width will be updated with 100-30 =70).

L	W
80	100
20	70

Step 3 : Cutting 20 * 30 Dimension

L	W
80	100
20	40

Step 4 : Cutting 70 * 100 Dimension

L	W
10	100
20	40

Step 5: Cutting 60 * 80 Dimension (Not possible since this roll is not having enough dimension so this will search for the next roll and will take the carpet from the other roll.

Step5 : Cutting 10 * 80 Dimension

L	W
10	80
10	20

The same process will continue until the dimension becomes 0 for each carpet.

2. **Entity List:**
   a. carpet (ForeignKey with carpet)
   b. roll_r1_width
   c. roll_r1_length
   d. roll_r2_width
   e. roll_r2_length
   f. available_area
   g. added_date
   h. updated_date
   i. status

CarpetInventory

1. **Introduction**: This table will be used to store the inventory of carpet.
2. **Entity List**:
   a. carpet (ForeignKey of carpet)
   b. available_roll_count
   c. sold_roll_count
   d. total_roll_count

PurchaseOrder

1. **Introduction**: This table will be used to store the purchase order of roll and carpet.
2. **Entity List**:
   a. company_id (ForeignKey Company)
   b. purchase_date
   c. order_cost
   d. status

PurchaseOrderCarpet

1. **Introduction**: This table is a sub part of the purchase order which will store the details of carpet purchased.
2. **Entity List**:
   a. carpet (ForeignKey Carpet)
   b. purchase_order (ForeignKey Purchase)
   c. carpet_width
   d. carpet_length
   e. carpet_cost
   f. status

Customer

1. **Introduction**: This is a customer table which will be used to store the customer information.
2. **Entity List**:
    a. company_id
    b. customer_name
    c. customer_phone
    d. customer_email
    e. status

Order

1. **Introduction**: This table is used to store the carpet order, this is base table which will store the base order detail and final cost
2. **Entity List**:
    a. customer_id
    b. order_date
    c. total_cost
    d. discount
    e. final_cost
    f. status

Orderdetails

1. **Introduction**: This table will be used to store the carpet order details , In one order there can be multiple carpets so the main order will be stored in the main order table and the list of ordered carpet will be stored in a list of rows in order detail table.
2. **Entity List**:
    a. order_id
    b. roll_id
    c. carpet_id
    d. carpet_width
    e. carpet_length
    f. mrp
    g. discount
    h. price
    i. status