Project Name: Digital Textile Production & Monitoring Platform

Short Description: This system is designed to manage and track each stage of textile production, from order creation to product delivery, ensuring efficiency, accuracy, and realtime monitoring throughout the workflow.

Target Users:

• Textile company staff (production managers, quality control, warehouse team)

• Company administrators and supervisors

• Employees responsible for production tracking

The Digital Textile Production & Monitoring Platform is a complete and easy-to-use system that helps textile companies manage, track, and control every stage of their production process from the first design idea to the final sale of the finished product.
Each product in the system is given a unique product code that allows it to be followed through every step of production. This code helps the company know exactly where each product is in the process and what its current status is.
The process starts with the design stage, where the designer creates the product plan, selects the fabric type, and decides on colors and sizes. After the design is completed, the product moves to the production phase, which is divided into several smaller parts such as sewing, ironing, and finishing (quality control). Each step in production is recorded in the system so that managers can see how the work is progressing in real time.
The platform also keeps detailed records about the number of items produced, how many of them failed quality control, and how many were approved and ready for sale. By collecting this information, the company can analyze performance, find production problems quickly, and improve efficiency.
In addition, the system stores important dates such as when the order was created, when production started and ended, and when the products became ready for sale. These records make it easy to check the timeline of each order and ensure that everything stays on

schedule.
Finally, the platform connects the production data with sales information, allowing the company to see the full life cycle of each product — from design to sale. This provides transparency, better organization, and more accurate decision-making for managers and employees at every level of the textile company.

1. Product

Key Attributes:

• Product_ID (Primary Key)

• Product_Code

• Design_ID (Foreign Key)

• Order_ID (Foreign Key)

• Product_Name

• Status (e.g., In Design, In Production, Ready for Sale)

Description: Represents each textile product in the system. Each product has a unique code and links to its design and related order.

2. Design

Key Attributes:

• Design_ID (Primary Key)

• Designer_Name

• Design_Date

• Fabric_Type

• Color

• Description

Description: Contains all details related to product design, including designer information, materials, and initial specifications before production begins.

3. Production

Key Attributes:

- Production_ID (Primary Key)
- Product_ID (Foreign Key)
- Start_Date
- End_Date
- Sewing_Status
- Ironing_Status
- Finish_Status (Quality Control)
- Total_Quantity
- Defective_Quantity
- Approved_Quantity

Description: Tracks the entire manufacturing process, including the main steps — sewing, ironing, and quality control. Records how many units were produced, defective, or approved for sale.

4. Order

Key Attributes:

- Order_ID (Primary Key)
- Customer_Name
- Order_Date
- Total_Ordered
- Sales_Ready_Date
- Delivery_Status

Description: Represents customer orders, including when the order was placed, how many units were requested, and when products became ready for sale.

5. Sales

Key Attributes:

• Sales_ID (Primary Key)

• Product_ID (Foreign Key)

• Order_ID (Foreign Key)

• Sales_Date

• Quantity_Sold

• Revenue

Description: Handles the sales process after products pass quality control. Links sales data with products and orders to track revenue and performance.

1. Functional Requirements

1. The system must create a unique Product_Code for every product.
2. The system must track each product through the design, production, and sales stages.
3. Every product must be linked to its related Design record.
4. The system must record all production steps, including Sewing, Ironing, and Quality Control.
5. The system must store the total quantity produced, the number of defective units, and the number of approved units.
6. The system must save customer orders with information such as customer name, order date, total ordered amount, and delivery status.
7. The sales module must record sales data, including quantity sold and revenue, linked to the correct product and order.
8. The system must show the current status of each product (for example: “In Design”, “In Production”, “Ready for Sale”).
9. The system must store production start and end dates for each product.
10. The system must show the full life cycle of a product from design to final sale.

2. Business Rules

1. Every product must have one unique Product_Code.

2. A product cannot start production unless its design is completed.

3. Production must follow this order: Sewing, Ironing, Quality Control.

4. Each production record must include Total_Quantity, Defective_Quantity, and Approved_Quantity.

5. Products that do not pass Quality Control cannot be sold.

6. Sales can only be recorded if the product has an Approved_Quantity greater than zero.

7. An order can only be marked as "Delivered" when all ordered products are completed and ready for sale.

8. A product can belong to only one order, but an order may include multiple products.

All date values must follow logical order: Design_Date, Production_Start_Date, Production_End Date, Sales_Date

9. A product's status must always match its real progress.

10. A product's status must always match its real progress.

11. A design record cannot be deleted if it is already linked to an active product.

12. Defective_Quantity cannot be greater than Total_Quantity for any production record.

13. Approved_Quantity must always equal Total_Quantity minus Defective_Quantity.

14. A production record must be created for every product before it can be marked as "Ready for Sale."

15. Sales records cannot be edited after they are confirmed.

16. Each order must have at least one product linked to it.

17. Delivery_Status cannot be changed to "Delivered" unless Sales_Ready_Date is filled.

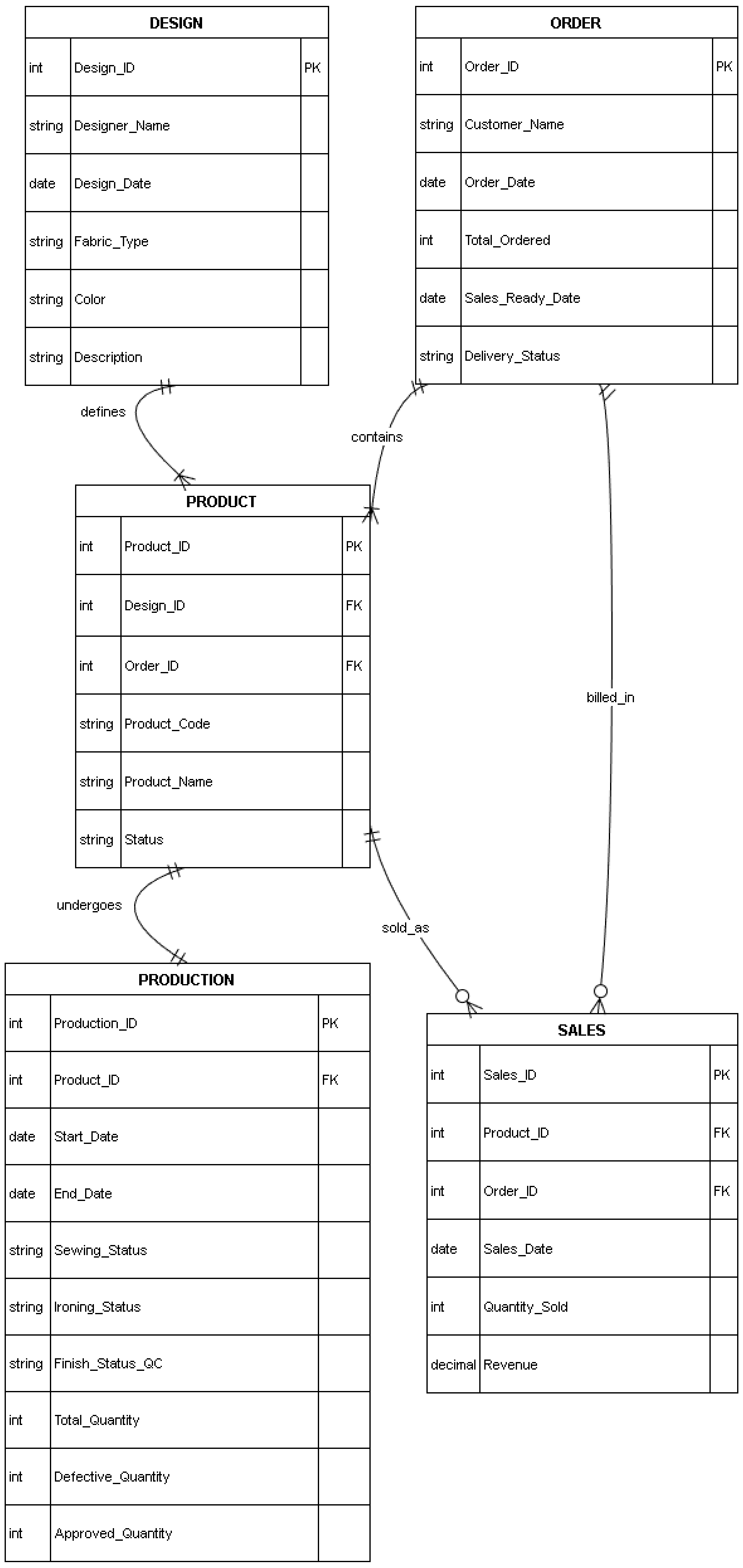
18. A product cannot be moved back to "In Design" after production has started.

In this project, I used Crow’s Foot Notation to clearly show the connections between the entities. The diagram follows these rules:

1-to-Many (Mandatory): This is used for the Design and Order tables. For example, one Design is linked to many Products. The vertical line (|) shows that a product must have a design to exist.

1-to-1 (Mandatory): There is a strict one-to-one relationship between Product and Production. This means every single product has exactly one unique production record to track its status.

Zero-or-Many (Optional): The Sales table uses a circle symbol (o). This allows a flexible relationship. It means a product can have zero sales (if it is still in stock) or many sales records (once it is sold).


DESIGN
int Design_ID PK
string Designer_Name
date Design_Date
string Fabric_Type
string Color
string Description
ORDER
int Order_ID PK
string Customer_Name
date Order_Date
int Total_Ordered
date Sales_Ready_Date
string Delivery_Status
defines
contains
PRODUCT
int Product_ID PK
int Design_ID FK
int Order_ID FK
string Product_Code
string Product_Name
string Status
billed_in
undergoes
sold_as
PRODUCTION
int Production_ID PK
int Product_ID FK
date Start_Date
date End_Date
string Sewing_Status
string Ironing_Status
string Finish_Status_QC
int Total_Quantity
int Defective_Quantity
int Approved_Quantity
SALES
int Sales_ID PK
int Product_ID FK
int Order_ID FK
date Sales_Date
int Quantity_Sold
decimal Revenue

Normalization and Design Evolution

Normalization Demonstration (1NF, 2NF, 3NF)

In this section, I describe the logical process of organizing the data for the Digital Textile Production & Monitoring Platform. I will explain how I transformed the raw data view into the final normalized entities found in my ER Diagram.

Step 1: First Normal Form (1NF)

Goal: Ensure that every attribute contains only a single (atomic) value and remove repeating groups.

Before (Raw Data Analysis): Imagine that we initially stored order information in a simple list format. In this "unnormalized" state, a single order record (e.g., Order ID 101) would contain a list of multiple products (e.g., "Shirt, Pants") all squeezed into one cell. This makes it impossible to query specific items effectively.

After (Applying 1NF): To solve this, I restructured the data so that each product has its own separate record. For example, instead of one row for Order 101 containing two items, I created two separate rows: one for the "Shirt" and one for the "Pants". Now, every attribute holds a single value, satisfying the First Normal Form. However, data such as the "Customer Name" repeats for every product in the same order.

Step 2: Second Normal Form (2NF)

Goal: Remove partial dependencies (data that does not depend on the entire Primary Key).

Reasoning: In the 1NF structure, the Primary Key is a combination of Order ID and Product Code. I observed that the Customer Name depends only on the Order ID, not on the specific product. Storing the customer name with every product creates unnecessary repetition.

After (Applying 2NF): I split the data into two distinct entities:

1. **ORDER Entity:** This stores only the information related to the order itself, such as Order ID and Customer Name.
2. **PRODUCT LIST Entity:** This stores the product details (like Product Code, Fabric, and Designer) separately from the customer data. Now, the customer information is recorded only once per order, satisfying the Second Normal Form.

Step 3: Third Normal Form (3NF)

Goal: Remove transitive dependencies (non-key attributes that depend on other non-key attributes).

Reasoning: In the "Product List" entity, I noticed that attributes like Designer Name and Fabric Type are actually dependent on the Design ID, not directly on the Product Code. If a designer changes their details, we should not have to update every single product record associated with them.

After (Applying 3NF - Final Design): To fix this, I extracted the design details into a new, separate entity called **DESIGN**.

1. **DESIGN Entity:** Stores specific design attributes like Designer Name and Fabric Type.
2. **PRODUCT Entity:** Now only contains the product's unique code and references the Design ID as a Foreign Key.

This structure matches my final ER Diagram, ensuring that design details are managed independently and no data is duplicated.

System Architecture: 3-Tier Architecture

In a real-life scenario, the Digital Textile Production & Monitoring Platform is implemented using a 3-tier architecture. This structure separates the user interface, the logic, and the data to provide transparency and better organization.

1. Presentation Tier (User Interface)

This is the top layer where users interact with the system. In this project, it includes two different platforms to help different staff members:

- **Desktop Application:** This app is for **Company Administrators and Supervisors**. They use it on their office computers to create new product designs , manage customer orders , and check the timeline of each order to ensure everything stays on schedule.
- **Mobile Application:** This app is for Employees on the production floor. They use mobile devices to track production in real time. Workers can quickly update the status of each step, such as Sewing, Ironing, and Quality Control.

2. Application Tier (Business Logic)

This middle layer is the "brain" of the platform. It controls how data moves and makes sure all Business Rules are followed:

- **Step Control:** It ensures that a product cannot start the production phase until the designer completes the design.
- **Workflow Order:** it makes sure the production process follows the correct logical order: first Sewing, then Ironing, and finally Quality Control.
- **Automatic Calculations:** This layer calculates the Approved_Quantity by taking the Total_Quantity and subtracting the Defective_Quantity.
- **Status Management:** It ensures the product status (like "In Design" or "Ready for Sale") always matches the real progress of the work.

3. Data Tier (Database)

This is the bottom layer and the memory of the system. It uses a MySQL Database to store all information:

- **Organized Tables:** It stores all the entities described in the ER Diagram, including Product, Design, Production, Order, and Sales.
- **Relationships:** It uses Foreign Keys to link records, such as connecting every product to its specific Design record and Customer Order.
- **Data Integrity:** By using Normalization (1NF, 2NF, 3NF), the database ensures that information like "Designer Name" or "Customer Name" is not duplicated unnecessarily.
- **Tracking:** It stores unique Product_Codes for every item, allowing the company to follow each product from the first design idea to the final sale.

Capacity Planning and Performance Forecasting (Data Analytics)

In this section, I added an analytical layer to the project. This part shows that the system is not only for recording data but also for helping managers make smart decisions for the future. Using the **analytics.py** module, the system processes raw data to create future predictions.

How it Works (Methodology)

- **Data Source:** The analysis uses information from the PRODUCTION and SALES tables.

- **Prediction Algorithm:** I analyzed the real data from the first six months (January to June) to calculate an "Average Growth Rate." The system uses this rate to predict production and revenue for the next six months.
- **Visualization:** To make the results easy to understand, I used the **Matplotlib** library in Python to create professional charts.

Business Value

This data-driven approach helps the textile company in three main ways:

1. **Capacity Management:** By seeing future production growth, the company can plan for materials and workers in advance.
2. **Performance Tracking:** Managers can compare real sales revenue with the predicted results to measure production efficiency.
3. **Better Decision Making:** The system provides clear data to help administrators decide on new investments or expansion plans.