AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

FACULTY OF SCIENCE & TECHNOLOGY

INTRODUCTION TO DATABASE

Spring 2022 – 2023

PROJECT ON

GARMENTS FACTORY MANAGEMENT SYSTEM

Supervised By

JUENA AHMED NOSHIN

Submitted By

Name	ID	Contribution
1. MARUF MUNTASIR	22-46620-1	Scenario Description, ER Diagram, Normalization, Table creation, Query writing
2. TALHA HOSSAIN SIFAT	22-46344-1	Scenario Description, Normalization, Data insertion
3. SUMAIYA SARKAR SHIMLA	22-46172-1	Scenario Description, Introduction, Normalization, Data insertion, Schema diagram
4. MD. ATIKUR RAHMAN	22-47944-2	Scenario Description, ER Diagram, Data insertion, Schema diagram
5. P.M. TASRIFUL ISLAM	22-47885-2	Scenario Description, Data insertion, Relational Algebra, Conclusions

Introduction to Database

Section: F

Group: 6

Date of Submission: 5/15/23

TABLE OF CONTENTS

Introduction:

This database project is about creating a database for garments management system. Implementing database in the sector of garments factory will provide many benefits over the traditional methods of managing data. The main objective of this database project is to CREATE a database management system based on garments factory management. The database can store and provide ways to access and use various information about the garments or the matters within the garments.

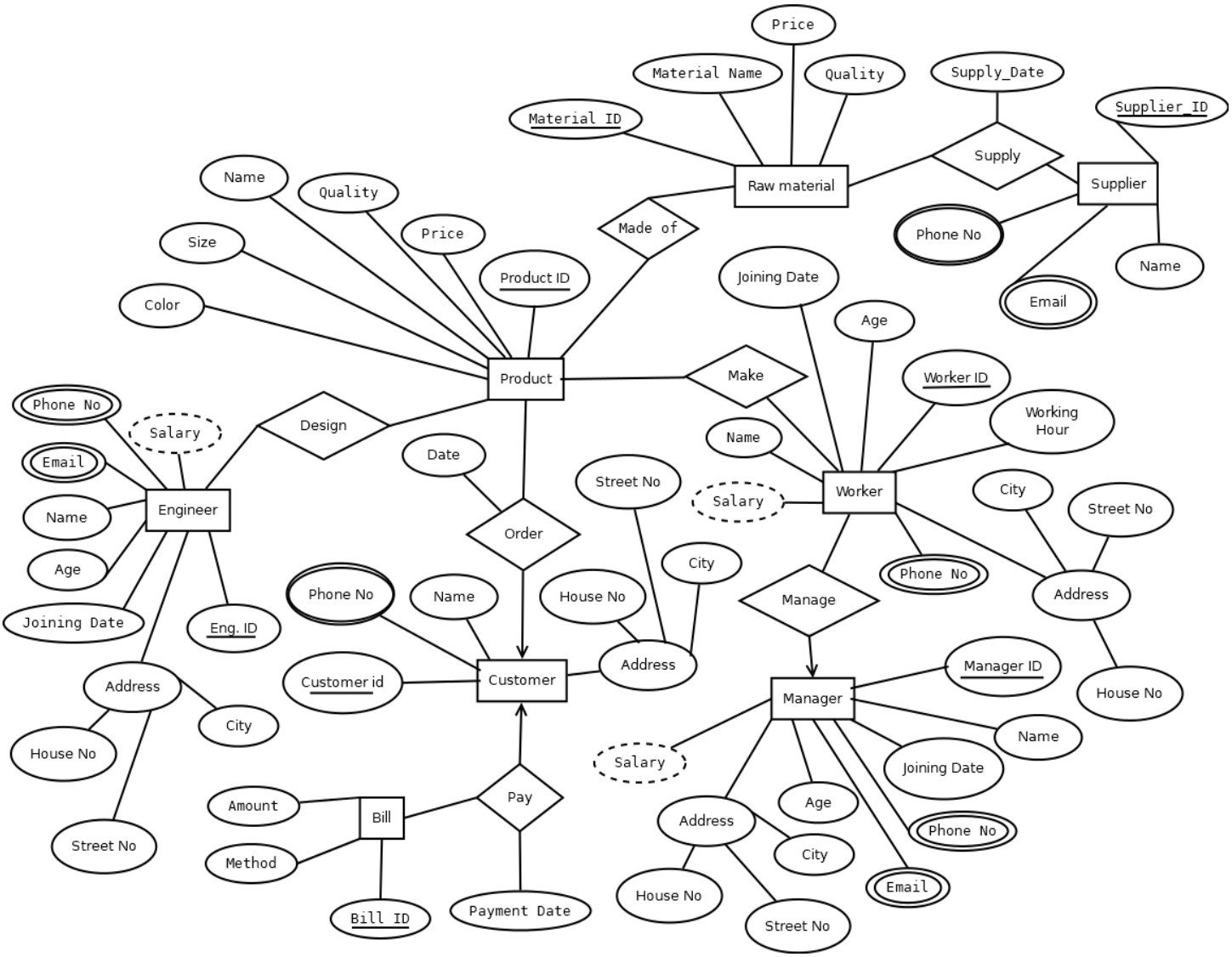
There are many entities which work managing a garment factory. The entities can be,

- Customers
- Products
- Workers
- Production lines
- Engineers or Designers
- Raw materials
- Suppliers
- Bills

Scenario Description:

In a garment management system, a customer can order many products, but a product can be ordered by one customer. A customer is identified by the customer id. The system also stores customer names, phone NUMBERs and addresses. A customer address is composed of house NUMBER, street name and city. A product is identified by the product id. Product also has product name, product price, product category, product size and color. There can be multiple sizes and colors of a product. While placing the order the order date is stored. A product can be made by many workers and a worker can make many products. A worker is identified by worker id. The worker's age, name, joining date and salary and the work hours are also stored in the system. The worker also has a worker address. Worker address is composed of city, street_no and house_no. According to the working hours the salary is calculated. A product can be designed by many engineers, and an engineer can design many products. An engineer can be identified by engineer id. An engineer also has name, age, salary, joining date and address. The address is composed of city, street_no and house_no. One manager can handle many workers, but one worker can work under only one manager. The system stores the manager's name, id, age and, joining date, salary and address. The address contains city, street_no and house_no. Raw materials can be supplied by many suppliers and one supplier can supply many raw materials. Supplier is identified by supplier id. The name phone NUMBER and email of the supplier are also stored. Raw materials contain material id, price, material name and quality. Raw materials are dyes, threads, buttons, zippers, and cloths. A product can be made by many raw materials, and one raw material can be used to make many products. A customer can pay many bills, but a bill can be paid by only one customer. A bill contains bill id, amount, and payment method. While paying, the payment date is recorded.

ER Diagram:


Price
Material Name
Quality
Supply_Date
Supplier_ID
Material ID
Supply
Raw material
Supplier
Name
Quality
Size
Price
Made of
Phone No
Name
Product ID
Joining Date
Email
Color
Age
Product
Make
Worker ID
Phone No
Salary
Design
Working Hour
Name
Date
Email
Street No
Worker
City
Engineer
Salary
Order
Street No
Name
Age
City
Phone No
Address
Joining Date
Eng. ID
Phone No
Name
House No
Manage
Address
Customer id
Customer
Address
Manager ID
Manager
House No
City
Name
Salary
House No
Joining Date
Amount
Bill
Pay
Age
Address
Phone No
Street No
Method
City
House No
Email
Bill ID
Payment Date
Street No

Normalization:

Supply:

UNF

supply (supplier_id, supplier_name, email, phone_no, supply_date, material_id, material_name, quality, price)

1NF

s_email and phone are multivalued attributes.

1. supplier_id, name, email, phone_no, supply_date, material_id, material_name, price, quality

2NF

1. supplier_id, name, email, phone_no, supply_date
2. material_id, material_name, quality, price

3NF

There is no transitive dependency. Relation already in 3NF.

1. supplier_id, name, email, phone_no, supply_date
2. m_id, name, m_quality, m_price

Table Creation

1. supplier_id, name, email, phone_no, s_date
2. material_id, material_name, price, quality
3. **supplier_id**, **material_id**

Made of

UNF

made of (product_id, name, price, quality, size, color, category, material_id, material_name, price, quality)

1NF

size and color are multivalued attributes.

1. product_id, name, price, quality, size, color, category, material_id, material_name, price, quality

2NF

1. product_id, name, price, size, color, category, quality
2. material_id, material_name, price, quality

3NF

There is no transitive dependency. Relation already in 3NF.

1. product_id, name, price, size, color, category, quality
2. material_id, material_name, price, quality

Table Creation

1. product_id, name, price, size, color, category, quality
2. material_id, material_name, price, quality
3. **product_id**, **material_id**

Make:

UNF

make (worker_id, name, age, salary, joining_date, phone_no, working_hour, city, street_no, house_no, product_id, name, price, quality, size, color, category)

1NF

Phone_no is multivalued attribute.

1. worker_id, name, age, salary, joining_date, phone_no, working_hour, city, street_no, house_no, product_id, name, price, quality, size, color, category

2NF

1. worker_id, name, age, salary, joining_date, phone_no, working_hour, city, street_no, house_no
2. product_id, name, price, size, color, category, quality

3NF

1. worker_id, name, age, salary, joining_date, phone_no, working_hour
2. city, street_no, house_no
3. product_id, name, price, size, color, category, quality

Table Creation

1. worker_id, name, age, salary, joining_date, phone_no, working_hour, **w_address_id**
2. w_address_id, city, street_no, house_no
3. product_id, name, price, quality, size, color, category
4. **worker_id, product_id**

Design:

UNF

design (eng_id, name, age, salary, phone_no, email, joining_date, city, street_no, house_no, product_id, name, price, size, color, category, quality)

1NF

email and phone_no are multivalued attributes.

1. eng_id, name, age, salary, phone_no, email, joining_date, city, street_no, house_no, product_id, name, price, size, color, category, quality

2NF

1. eng_id, name, age, salary, phone_no, email, joining_date, city, street_no, house_no
2. product_id, name, price, size, color, category, quality

3NF

1. eng_id, name, age, salary, phone_no, email, joining_date
2. city, street_no, house_no
3. product_id, name, price, size, color, category, quality

Table Creation

1. eng_id, name, age, salary, phone_no, email, joining_date, **eng_address_id**
2. eng_address_id, city, street_no, house_no
3. product_id, name, price, size, color, category, quality
4. **eng_id**, **product_id**

Manage

UNF

manage (manager_id, name, age, joining_date, salary, email, phone_no, city, street_no, house_no, worker_id, name, age, salary, joining_date, phone_no, working_hour)

1NF

email, phone_no and phone_no are multivalued attributes.

1. manager_id, name, age, joining_date, salary, email, phone_no, city, street_no, house_no, worker_id, name, age, salary, joining_date, phone_no, working_hour

2NF

1. manager_id, name, age, joining_date, salary, email, phone_no, city, street_no, house_no
2. worker_id, name, age, salary, joining_date, phone_no, working_hour

3NF

1. manager_id, name, age, joining_date, salary, email, phone_no
2. city, street_no, house_no
3. worker_id, name, age, salary, joining_date, phone_no, working_hour
4. city, street_no, house_no

Table Creation

1. manager_id, name, age, joining_date, salary, email, phone_no, **m_address_id**
2. m_address_id, city, street_no, house_no
3. worker_id, name, age, salary, joining_date, phone_no, working_hour, **manager_id, w_address_id**
4. w_address_id, city, street_no, house_no

Order

UNF

order (product_id, name, price, size, color, category, quality, order_date, customer_id, name, phone_no, city, street_no, house_no)

1NF

phone_no is attribute.

1. product_id, name, price, size, color, category, quality, customer_id, name, phone_no, city, street_no, house_no

2NF

1. product_id, name, price, size, color, category, quality, order_date
2. customer_id, name, phone_no, city, street_no, house_no

3NF

1. product_id, name, price, size, color, category, quality, order_date
2. customer_id, name, phone_no
3. city, street_no, house_no

Table Creation

1. product_id, name, price, size, color, category, quality, order_date, **customer_id**
2. customer_id, name, phone_no, **c_address_id**
3. c_address_id, city, street_no, house_no

Pay

UNF

pay (customer_id, name, phone_no, city, street_no, house_no, bill_id, amount, method, payment_date)

1NF

phone_no is a multivalued attribute

1. customer_id, name, phone_no, city, street_no, house_no, bill_id, amount, method, payment_date

2NF

1. customer_id, name, phone_no, city, street_no, house_no
2. bill_id, amount, method, payment_date

3NF

1. customer_id, name, phone_no
2. bill_id, amount, method, payment_date
3. city, street_no, house_no

Table Creation

1. customer_id, name, phone_no, **c_address_id**
2. bill_id, amount, method, payment_date, **customer_id**
3. c_address_id, city, street_no, house_no

Temporary Tables:

1. supplier_id, name, email, phone_no, s_date
2. material_id, material_name, price, quality
3. **supplier_id**, **material_id**
4. ~~product_id, name, price, size, color, category, quality~~
5. ~~material_id, material_name, price, quality~~
6. **product_id**, **material_id**
7. ~~worker_id, name, age, salary, joining_date, phone_no, working_hour, **w_address_id**~~
8. w_address_id, city, street_no, house_no
9. ~~product_id, name, price, quality, size, color, category~~

10. **worker_id, product_id**
11. eng_id, name, age, salary, phone_no, email, joining_date, **eng_address_id**
12. eng_address_id, city, street_no, house_no
13. ~~product_id, name, price, size, color, category, quality~~
14. **eng_id**, **product_id**
15. manager_id, name, joining_date, salary, email, phone_no, **m_address_id**
16. m_address_id, city, street_no, house_no
17. worker_id, name, age, salary, joining_date, phone_no, working_hour, **manager_id, w_address_id**
18. ~~w_address_id, city, street_no, house_no~~
19. product_id, name, price, size, color, category, quality, order_date, **customer_id**
20. customer_id, name, phone_no, **c_address_id**
21. c_address_id, city, street_no, house_no
22. ~~customer_id, name, phone_no, **c_address_id**~~
23. bill_id, amount, method, payment_date, **customer_id**
24. ~~c_address_id, city, street_no, house_no~~

Final Tables:

1. supplier_id, name, email1, email2, email3, phone_no1, phone_no2, phone_no3, s_date
2. material_id, material_name, price, quality
3. worker_id, name, age, joining_date, salary, phone_no1, phone_no2, phone_no3, working_hour, **w_address_id, manager_id**
4. w_address_id, city, street_no, house_no
5. manager_id, name, age, joining_date, salary, phone_no1, phone_no2, phone_no3, email1, email2, email3, **m_address_id**
6. m_address_id, city, street_no, house_no
7. product_id, product_size, product_color, name, price, category, quality, order_date**, customer_id**
8. eng_id, name, age, joining_date, salary, phone_no1, phone_no2, phone_no3, email1, email2, email3, **eng_address_id**
9. eng_address_id, city, street_no, house_no
10. customer_id, name, phone_no1, phone_no2, phone_no3, **c_address_id**
11. c_address_id, city, street_no, house_no
12. bill_id, amount, method, payment_date, **customer_id**
13. **supplier_id**, **material_id**
14. **product_id**, **material_id**
15. **worker_id, product_id**
16. **eng_id**, **product_id**

Sehema Diagram:

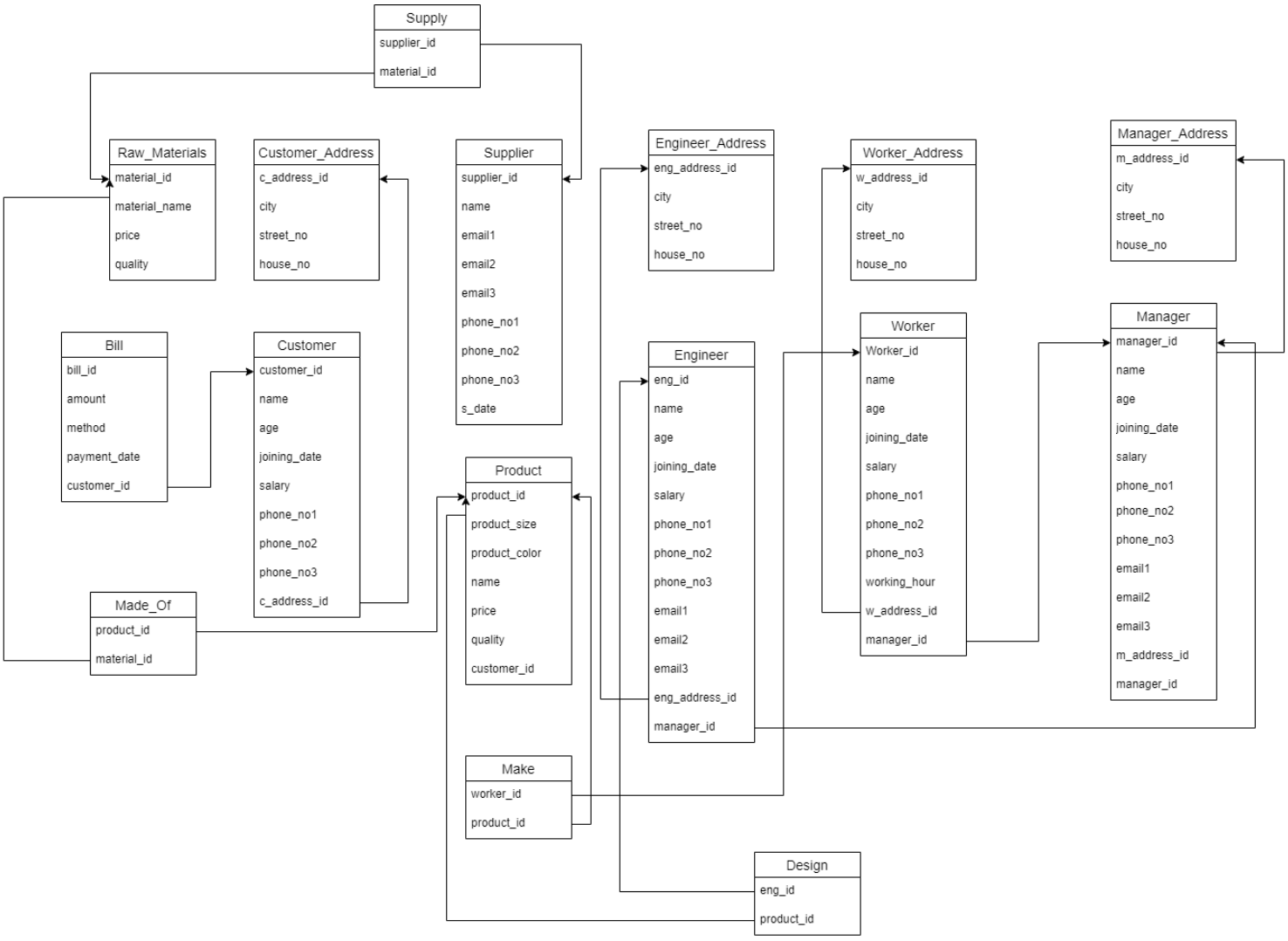


Table Creation Queries:

1. Supplier Table:

CREATE TABLE Supplier (supplier_id NUMBER (6) PRIMARY KEY, name VARCHAR2(20), phone_no1 NUMBER (12) NOT NULL, phone_no2 NUMBER (12), phone_no3 NUMBER (12), email1 VARCHAR2(30), email2 VARCHAR2(30), email3 VARCHAR2(30), s_date date);

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SUPPLIER	SUPPLIER_ID	Number	-	6	0	1	-	-	-
	NAME	Varchar2	20	-	-	-	✓	-	-
	PHONE_NO1	Number	-	12	0	-	-	-	-
	PHONE_NO2	Number	-	12	0	-	✓	-	-
	PHONE_NO3	Number	-	12	0	-	✓	-	-
	EMAIL1	Varchar2	30	-	-	-	✓	-	-
	EMAIL2	Varchar2	30	-	-	-	✓	-	-
	EMAIL3	Varchar2	30	-	-	-	✓	-	-
	S_DATE	Date	7	-	-	-	✓	-	-
									1 - 9

2. Raw_Material Table:

CREATE TABLE Raw_Material (material_id NUMBER (6) PRIMARY KEY, material_name VARCHAR2(20) NOT NULL, price NUMBER (4), quality VARCHAR2(20));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RAW_MATERIAL	MATERIAL_ID	Number	-	6	0	1	-	-	-
	MATERIAL_NAME	Varchar2	20	-	-	-	-	-	-
	PRICE	Number	-	4	0	-	✓	-	-
	QUALITY	Varchar2	20	-	-	-	✓	-	-
									1 - 4

3. Worker_Address Table:

CREATE TABLE Worker_Address (w_address_id NUMBER (6) PRIMARY KEY, city VARCHAR2(20), street_no VARCHAR2(20), house_no VARCHAR2(20));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
WORKER_ADDRESS	W_ADDRESS_ID	Number	-	6	0	1	-	-	-
	CITY	Varchar2	20	-	-	-	✓	-	-
	STREET_NO	Varchar2	20	-	-	-	✓	-	-
	HOUSE_NO	Varchar2	20	-	-	-	✓	-	-
									1 - 4

4. Manager_Address Table:

CREATE TABLE Manager_Address (m_address_id NUMBER (6) PRIMARY KEY, city VARCHAR2(20), street_no VARCHAR2(20), house_no VARCHAR2(20));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANAGER_ADDRESS	M_ADDRESS_ID	Number	-	6	0	1	-	-	-
	CITY	Varchar2	20	-	-	-	✓	-	-
	STREET_NO	Varchar2	20	-	-	-	✓	-	-
	HOUSE_NO	Varchar2	20	-	-	-	✓	-	-
									1 - 4

5. **Manager Table:**

CREATE TABLE Manager (manager_id NUMBER(6) PRIMARY KEY, name VARCHAR2(20), age NUMBER(3), joining_date date, salary NUMBER(6), phone_no1 NUMBER(12) NOT NULL, phone_no2 NUMBER(12), phone_no3 NUMBER(12), email1 VARCHAR2(30), email2 VARCHAR2(30), email3 VARCHAR2(30), m_address_id NUMBER(6), constraint m_fk1 FOREIGN KEY(m_address_id) REFERENCES Manager_Address(m_address_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANAGER	MANAGER_ID	Number	-	6	0	1	-	-	-
	NAME	Varchar2	20	-	-	-	✓	-	-
	AGE	Number	-	3	0	-	✓	-	-
	JOINING_DATE	Date	7	-	-	-	✓	-	-
	SALARY	Number	-	6	0	-	✓	-	-
	PHONE_NO1	Number	-	12	0	-	-	-	-
	PHONE_NO2	Number	-	12	0	-	✓	-	-
	PHONE_NO3	Number	-	12	0	-	✓	-	-
	EMAIL1	Varchar2	30	-	-	-	✓	-	-
	EMAIL2	Varchar2	30	-	-	-	✓	-	-
	EMAIL3	Varchar2	30	-	-	-	✓	-	-
	M_ADDRESS_ID	Number	-	6	0	-	✓	-	-
									1 - 12

6. **Worker Table:**

CREATE TABLE Worker (worker_id NUMBER(6) PRIMARY KEY, name VARCHAR2(20), age NUMBER(3), joining_date date, salary NUMBER(6), phone_no1 NUMBER(12) NOT NULL, phone_no2 NUMBER(12), phone_no3 NUMBER(12), working_hour NUMBER(10), w_address_id NUMBER(6) , manager_id NUMBER(6), constraint w_fk1 FOREIGN KEY (w_address_id) REFERENCES Worker_Address(w_address_id), constraint w_fk2 FOREIGN KEY (manager_id) REFERENCES Manager(manager_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
WORKER	WORKER_ID	Number	-	6	0	1	-	-	-
	NAME	Varchar2	20	-	-	-	✓	-	-
	AGE	Number	-	3	0	-	✓	-	-
	JOINING_DATE	Date	7	-	-	-	✓	-	-
	SALARY	Number	-	6	0	-	✓	-	-
	PHONE_NO1	Number	-	12	0	-	-	-	-
	PHONE_NO2	Number	-	12	0	-	✓	-	-
	PHONE_NO3	Number	-	12	0	-	✓	-	-
	WORKING_HOUR	Number	-	10	0	-	✓	-	-
	W_ADDRESS_ID	Number	-	6	0	-	✓	-	-
	MANAGER_ID	Number	-	6	0	-	✓	-	-

1 - 11

7. Customer_Address Table:

CREATE TABLE Customer_Address (c_address_id NUMBER (6) PRIMARY KEY, city VARCHAR2(20), street_no VARCHAR2(20), house_no VARCHAR2(20));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER_ADDRESS	C_ADDRESS_ID	Number	-	6	0	1	-	-	-
	CITY	Varchar2	20	-	-	-	✓	-	-
	STREET_NO	Varchar2	20	-	-	-	✓	-	-
	HOUSE_NO	Varchar2	20	-	-	-	✓	-	-

1 - 4

8. Customer Table:

CREATE TABLE Customer (customer_id NUMBER (6) PRIMARY KEY, name VARCHAR2(20), age NUMBER (3), phone_no1 NUMBER (12) NOT NULL, phone_no2 NUMBER (12), phone_no3 NUMBER (12), c_address_id NUMBER (6), constraint c_fk1 FOREIGN KEY(c_address_id) REFERENCES Customer_Address(c_address_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	CUSTOMER_ID	Number	-	6	0	1	-	-	-
	NAME	Varchar2	20	-	-	-	✓	-	-
	AGE	Number	-	3	0	-	✓	-	-
	PHONE_NO1	Number	-	12	0	-	-	-	-
	PHONE_NO2	Number	-	12	0	-	✓	-	-
	PHONE_NO3	Number	-	12	0	-	✓	-	-
	C_ADDRESS_ID	Number	-	6	0	-	✓	-	-

1 - 7

9. Product Table:

CREATE TABLE Product (product_id NUMBER (6) PRIMARY KEY, product_size VARCHAR2(6), product_color VARCHAR2(10), name VARCHAR2(30), price NUMBER (6), quality VARCHAR2(20), order_date DATE, customer_id NUMBER (6), constraint p_fk1 FOREIGN KEY (customer_id) REFERENCES Customer(customer_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRODUCT	PRODUCT_ID	Number	-	6	0	1	-	-	-
	PRODUCT_SIZE	Varchar2	6	-	-	-	✓	-	-
	PRODUCT_COLOR	Varchar2	10	-	-	-	✓	-	-
	NAME	Varchar2	30	-	-	-	✓	-	-
	PRICE	Number	-	6	0	-	✓	-	-
	QUALITY	Varchar2	20	-	-	-	✓	-	-
	ORDER_DATE	Date	7	-	-	-	✓	-	-
	CUSTOMER_ID	Number	-	6	0	-	✓	-	-

1 - 8

10. Engineer_Address Table:

CREATE TABLE Engineer_Address (eng_address_id NUMBER (6) PRIMARY KEY, city VARCHAR2(20), street_no VARCHAR2(20), house_no VARCHAR2(20));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ENGINEER_ADDRESS	ENG_ADDRESS_ID	Number	-	6	0	1	-	-	-
	CITY	Varchar2	20	-	-	-	✓	-	-
	STREET_NO	Varchar2	20	-	-	-	✓	-	-
	HOUSE_NO	Varchar2	20	-	-	-	✓	-	-

1 - 4

11. Engineer Table:

CREATE TABLE Engineer (eng_id NUMBER(6) PRIMARY KEY, name VARCHAR2(20), age NUMBER(3), joining_date date, salary NUMBER(6), phone_no1 NUMBER(12) NOT NULL, phone_no2 NUMBER(12), phone_no3 NUMBER(12), email1 VARCHAR2(30), email2 VARCHAR2(30), email3 VARCHAR2(30), eng_address_id NUMBER(6), constraint e_fk1 FOREIGN KEY(eng_address_id) REFERENCES Engineer_Address(eng_address_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ENGINEER	ENG_ID	Number	-	6	0	1	-	-	-
	NAME	Varchar2	20	-	-	-	✓	-	-
	AGE	Number	-	3	0	-	✓	-	-
	JOINING_DATE	Date	7	-	-	-	✓	-	-
	SALARY	Number	-	6	0	-	✓	-	-
	PHONE_NO1	Number	-	12	0	-	-	-	-
	PHONE_NO2	Number	-	12	0	-	✓	-	-
	PHONE_NO3	Number	-	12	0	-	✓	-	-
	EMAIL1	Varchar2	30	-	-	-	✓	-	-
	EMAIL2	Varchar2	30	-	-	-	✓	-	-
	EMAIL3	Varchar2	30	-	-	-	✓	-	-
	ENG_ADDRESS_ID	Number	-	6	0	-	✓	-	-

1 - 12

12. Bill Table:

CREATE TABLE Bill (bill_id NUMBER (6) PRIMARY KEY, amount NUMBER (10), method VARCHAR2(20), payment_date date, customer_id NUMBER (6), constraint b_fk1 FOREIGN KEY (customer_id) REFERENCES Customer(customer_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BILL	BILL_ID	Number	-	6	0	1	-	-	-
	AMOUNT	Number	-	10	0	-	✓	-	-
	METHOD	Varchar2	20	-	-	-	✓	-	-
	PAYMENT_DATE	Date	7	-	-	-	✓	-	-
	CUSTOMER_ID	Number	-	6	0	-	✓	-	-
									1 - 5

13. Supply Table:

CREATE TABLE Supply (supplier_id NUMBER (6), material_id NUMBER (6), CONSTRAINT s_pk PRIMARY KEY (supplier_id, material_id), CONSTRAINT s_fk1 FOREIGN KEY (supplier_id) REFERENCES Supplier(supplier_id), CONSTRAINT s_fk2 FOREIGN KEY (material_id) REFERENCES Raw_Material(material_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SUPPLY	SUPPLIER_ID	Number	-	6	0	1	-	-	-
	MATERIAL_ID	Number	-	6	0	2	-	-	-
									1 - 2

14. Made_Of Table:

CREATE TABLE Made_Of(product_id NUMBER(6), material_id NUMBER(6), CONSTRAINT madeof_pk PRIMARY KEY (product_id, material_id), CONSTRAINT madeof_fk1 FOREIGN KEY (product_id) REFERENCES Product(product_id), CONSTRAINT madeof_fk2 FOREIGN KEY (material_id) REFERENCES Raw_Material(material_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MADE_OF	PRODUCT_ID	Number	-	6	0	1	-	-	-
	MATERIAL_ID	Number	-	6	0	2	-	-	-
									1 - 2

15. Make Table:

CREATE TABLE Make (worker_id NUMBER (6), product_id NUMBER (6), CONSTRAINT make_pk PRIMARY KEY (worker_id, product_id), CONSTRAINT make_fk1 FOREIGN KEY (worker_id) REFERENCES Worker(worker_id), CONSTRAINT make_fk2 FOREIGN KEY (product_id) REFERENCES Product(product_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MAKE	WORKER_ID	Number	-	6	0	1	-	-	-
	PRODUCT_ID	Number	-	6	0	2	-	-	-
									1 - 2

16. Design Table:

CREATE TABLE Design (eng_id NUMBER (6), product_id NUMBER (6), CONSTRAINT design_pk PRIMARY KEY (eng_id, product_id), CONSTRAINT design_fk1 FOREIGN KEY (eng_id) REFERENCES Engineer(eng_id), CONSTRAINT design_fk2 FOREIGN KEY (product_id) REFERENCES Product(product_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DESIGN	ENG_ID	Number	-	6	0	1	-	-	-
	PRODUCT_ID	Number	-	6	0	2	-	-	-
									1 - 2

Sequence:

1. CREATE SEQUENCE sq_supplier INCREMENT BY 1 START WITH 1 MAXVALUE 10000 NOCACHE NOCYCLE;
2. CREATE SEQUENCE sq_material INCREMENT BY 1 START WITH 1 MAXVALUE 10000 NOCACHE NOCYCLE;
3. CREATE SEQUENCE sq_worker INCREMENT BY 1 START WITH 1 MAXVALUE 10000 NOCACHE NOCYCLE;
4. CREATE SEQUENCE sq_waddress INCREMENT BY 1 START WITH 1 MAXVALUE 10000 NOCACHE NOCYCLE;
5. CREATE SEQUENCE sq_manager INCREMENT BY 1 START WITH 1 MAXVALUE 10000 NOCACHE NOCYCLE;
6. CREATE SEQUENCE sq_maddress INCREMENT BY 1 START WITH 1 MAXVALUE 10000 NOCACHE NOCYCLE;
7. CREATE SEQUENCE sq_product INCREMENT BY 1 START WITH 1 MAXVALUE 10000 NOCACHE NOCYCLE;
8. CREATE SEQUENCE sq_customer INCREMENT BY 1 START WITH 1 MAXVALUE 10000 NOCACHE NOCYCLE;
9. CREATE SEQUENCE sq_caddress INCREMENT BY 1 START WITH 1 MAXVALUE 10000 NOCACHE NOCYCLE;
10. CREATE SEQUENCE sq_engineer INCREMENT BY 1 START WITH 1 MAXVALUE 10000 NOCACHE NOCYCLE;
11. CREATE SEQUENCE sq_eaddress INCREMENT BY 1 START WITH 1 MAXVALUE 10000 NOCACHE NOCYCLE;
12. CREATE SEQUENCE sq_bill INCREMENT BY 1 START WITH 1 MAXVALUE 10000 NOCACHE NOCYCLE;

SELECT sequence_name, min_value, max_value, increment_by, last_number FROM user_sequences;

User access control:

CREATE user garments identified by management;

GRANT connect, resource, unlimited tablespace to garments;

GRANT create table, create sequence, create view TO garments;

ALTER USER GARMENTS DEFAULT TABLESPACE USERS;

ALTER USER GARMENTS TEMPORARY TABLESPACE TEMP;

CREATE ROLE manager;

GRANT create table, create view TO manager;

ALTER user garments identified by management;

Data Insertion:

Data for Supplier table:

1. INSERT INTO Supplier (supplier_id, name, phone_no1, phone_no2, phone_no3, email1, email2, email3, s_date) VALUES (sq_supplier.NEXTVAL, 'Oceanic Logistics', 01712345678, 01812345678, 01912345678, 'info@oceaniclogistics.com', 'sales@oceaniclogistics.com', 'contact@oceaniclogistics.com', TO_DATE('2023-05-13', 'YYYY-MM-DD'));
2. INSERT INTO Supplier (supplier_id, name, phone_no1, phone_no2, phone_no3, email1, email2, email3, s_date) VALUES (sq_supplier.NEXTVAL, 'SeaLink Shipping', 01723456789, 01823456789, 01923456789, 'info@sealinkshipping.com', 'sales@sealinkshipping.com', 'contact@sealinkshipping.com', TO_DATE('2023-05-13', 'YYYY-MM-DD'));
3. INSERT INTO Supplier (supplier_id, name, phone_no1, phone_no2, phone_no3, email1, email2, email3, s_date) VALUES (sq_supplier.NEXTVAL, 'Swift Maritime', 01734567890, 01834567890, 01934567890, 'info@swiftmaritime.com', 'sales@swiftmaritime.com', 'contact@swiftmaritime.com', TO_DATE('2023-05-13', 'YYYY-MM-DD'));
4. INSERT INTO Supplier (supplier_id, name, phone_no1, phone_no2, phone_no3, email1, email2, email3, s_date) VALUES (sq_supplier.NEXTVAL, 'CargoExpress Ltd', 01745678901, 01845678901, 01945678901, 'info@cargoexpress.com', 'sales@cargoexpress.com', 'contact@cargoexpress.com', TO_DATE('2023-05-13', 'YYYY-MM-DD'));
5. INSERT INTO Supplier (supplier_id, name, phone_no1, phone_no2, phone_no3, email1, email2, email3, s_date) VALUES (sq_supplier.NEXTVAL, 'Globe Shipping', 01756789012, 01856789012, 01956789012, 'info@globeshipping.com', 'sales@globeshipping.com', 'contact@globeshipping.com', TO_DATE('2023-05-13', 'YYYY-MM-DD'));

SUPPLIER_ID	NAME	PHONE_NO1	PHONE_NO2	PHONE_NO3	EMAIL1	EMAIL2	EMAIL3	S_DATE
1	Oceanic Logistics	1712345678	1812345678	1912345678	info@oceaniclogistics.com	sales@oceaniclogistics.com	contact@oceaniclogistics.com	13-MAY-23
2	SeaLink Shipping	1723456789	1823456789	1923456789	info@sealinkshipping.com	sales@sealinkshipping.com	contact@sealinkshipping.com	13-MAY-23
3	Swift Maritime	1734567890	1834567890	1934567890	info@swiftmaritime.com	sales@swiftmaritime.com	contact@swiftmaritime.com	13-MAY-23
4	CargoExpress Ltd	1745678901	1845678901	1945678901	info@cargoexpress.com	sales@cargoexpress.com	contact@cargoexpress.com	13-MAY-23
5	Globe Shipping	1756789012	1856789012	1956789012	info@globeshipping.com	sales@globeshipping.com	contact@globeshipping.com	13-MAY-23

Data for Raw_Material

1. INSERT INTO Raw_Material (material_id, material_name, price, quality) VALUES (sq_material.NEXTVAL, 'Cotton Fabric', 100, 'High');
2. INSERT INTO Raw_Material (material_id, material_name, price, quality) VALUES (sq_material.NEXTVAL, 'Polyester Thread', 50, 'Medium');
3. INSERT INTO Raw_Material (material_id, material_name, price, quality) VALUES (sq_material.NEXTVAL, 'Zippers', 30, 'Low');
4. INSERT INTO Raw_Material (material_id, material_name, price, quality) VALUES (sq_material.NEXTVAL, 'Buttons', 20, 'Medium');
5. INSERT INTO Raw_Material (material_id, material_name, price, quality) VALUES (sq_material.NEXTVAL, 'Elastic Bands', 10, 'High');

MATERIAL_ID	MATERIAL_NAME	PRICE	QUALITY
1	Cotton Fabric	100	High
2	Polyester Thread	50	Medium
3	Zippers	30	Low
4	Buttons	20	Medium
5	Elastic Bands	10	High

Data of Worker Address:

1. INSERT INTO Worker_Address (w_address_id, city, street_no, house_no) VALUES (sq_waddress.NEXTVAL, 'Dhaka', '123/A', '25');
2. INSERT INTO Worker_Address (w_address_id, city, street_no, house_no) VALUES (sq_waddress.NEXTVAL, 'Chittagong', '456/B', '10');
3. INSERT INTO Worker_Address (w_address_id, city, street_no, house_no) VALUES (sq_waddress.NEXTVAL, 'Sylhet', '789/C', '5');
4. INSERT INTO Worker_Address (w_address_id, city, street_no, house_no) VALUES (sq_waddress.NEXTVAL, 'Rajshahi', '234/D', '15');
5. INSERT INTO Worker_Address (w_address_id, city, street_no, house_no) VALUES (sq_waddress.NEXTVAL, 'Khulna', '567/E', '8');

W_ADDRESS_ID	CITY	STREET_NO	HOUSE_NO
1	Dhaka	123/A	25
2	Chittagong	456/B	10
3	Sylhet	789/C	5
4	Rajshahi	234/D	15
5	Khulna	567/E	8

Data of Manager_Address:

1. INSERT INTO MANAGER_ADDRESS (m_address_id, city, street_no, house_no) VALUES (sq_maddress.NEXTVAL, 'Dhaka','9/A',2);
2. INSERT INTO MANAGER_ADDRESS (m_address_id, city, street_no, house_no) VALUES (sq_maddress.NEXTVAL, 'Comilla','10/A',9);
3. INSERT INTO MANAGER_ADDRESS (m_address_id, city, street_no, house_no) VALUES (sq_maddress.NEXTVAL, 'Rajshahi','11/A',8);
4. INSERT INTO MANAGER_ADDRESS (m_address_id, city, street_no, house_no) VALUES (sq_maddress.NEXTVAL, 'Barishal','3/A',7);
5. INSERT INTO MANAGER_ADDRESS (m_address_id, city, street_no, house_no) VALUES (sq_maddress.NEXTVAL, 'Narshindi','5/A',5);

M_ADDRESS_ID	CITY	STREET_NO	HOUSE_NO
1	Dhaka	9/A	2
2	Comilla	10/A	9
3	Rajshahi	11/A	8
4	Barishal	3/A	7
5	Narshindi	5/A	5

Data of Manager:

1. INSERT INTO MANAGER (manager_id, name, age, joining _date, salary, phone_no1, phone_no2, phone_no3, email1, email2, email3, m_address_id) VALUES (sq_manager.NEXTVAL, 'Ashiq', 35, TO_DATE('2021-05-18', 'YYYY-MM-DD'), 3000,01712567845, 01812045678, 01912355678, 'manager@ashiq.com', 'sales@ashiq1.com', 'contact@ashiq2.com', 1);
2. INSERT INTO MANAGER (manager_id, name, age, joining _date, salary, phone_no1, phone_no2, phone_no3, email1, email2, email3, m_address_id) VALUES (sq_manager.NEXTVAL, 'Shuvho',35, TO_DATE('2018-11-21', 'YYYY-MM-DD'), 4000,01712567855, 01812145678, 01942355678, 'manager@shuvho.com', 'sales@shuvho1.com', 'contact@shuvho2.com', 4);
3. INSERT INTO MANAGER (manager_id, name, age, joining _date, salary, phone_no1, phone_no2, phone_no3, email1, email2, email3, m_address_id) VALUES (sq_manager.NEXTVAL, 'Ashiqur', 32, TO_DATE('2019-05-17', 'YYYY-MM-DD'), 3000,01712867845, 01912045678, 01912455678, 'manager@ashiqur.com', 'sales@ashiqur1.com', 'contact@ashiqur2.com', 5);
4. INSERT INTO MANAGER (manager_id, name, age, joining _date, salary, phone_no1, phone_no2, phone_no3, email1, email2, email3, m_address_id) VALUES (sq_manager.NEXTVAL, 'Ayan',32, TO_DATE('2020-07-13', 'YYYY-MM-DD'), 3500,01712567845, 01812045678, 01912355678, 'manager@ayan.com', 'sales@ayan1.com', 'contact@ayan2.com', 2);

5. INSERT INTO MANAGER (manager_id, name, age, joining _date, salary, phone_no1, phone_no2, phone_no3, email1, email2, email3, m_address_id) VALUES (sq_manager.NEXTVAL, 'Ashfi', 33, TO_DATE('2018-05-13', 'YYYY-MM-DD'),4000,01712567845, 01812045678, 01912355678, 'manager@ashfi.com', 'sales@ashfi1.com', 'contact@ashfi2.com', 3);

MANAGER_ID	NAME	AGE	JOINING_DATE	SALARY	PHONE_NO1	PHONE_NO2	PHONE_NO3	EMAIL1	EMAIL2	EMAIL3	M_ADDRESS_ID
1	Ashiq	35	18-MAY-21	3000	1712567845	1812045678	1912355678	manager@ashiq.com	sales@ashiq1.com	contact@ashiq2.com	1
2	Shuvho	35	21-NOV-18	4000	1712567855	1812145678	1942355678	manager@shuvho.com	sales@shuvho1.com	contact@shuvho2.com	4
3	Ashiqur	32	17-MAY-19	3000	1712867845	1912045678	1912455678	manager@ashiqur.com	sales@ashiqur1.com	contact@ashiqur2.com	5
4	Ayan	32	13-JUL-20	3500	1712567845	1812045678	1912355678	manager@ayan.com	sales@ayan1.com	contact@ayan2.com	2
5	Ashfi	33	13-MAY-18	4000	1712567845	1812045678	1912355678	manager@ashfi.com	sales@ashfi1.com	contact@ashfi2.com	3

Data for Worker:

1. INSERT INTO Worker (worker_id, name, age, joining_date, salary, phone_no1, phone_no2, phone_no3, working_hour, w_address_id, manager_id) VALUES (sq_worker.NEXTVAL, 'Rahim', 25, TO_DATE ('2020-05-13', 'YYYY-MM-DD'), 25000, 0171234567, 0181234567, 0191234567, 8, 1, 1);
2. INSERT INTO Worker (worker_id, name, age, joining_date, salary, phone_no1, phone_no2, phone_no3, working_hour, w_address_id, manager_id) VALUES (sq_worker.NEXTVAL, 'Karim', 28, TO_DATE ('2019-07-18', 'YYYY-MM-DD'), 28000, 0172345678, 0182345678, 0192345678, 8, 2, 1);
3. INSERT INTO Worker (worker_id, name, age, joining_date, salary, phone_no1, phone_no2, phone_no3, working_hour, w_address_id, manager_id) VALUES (sq_worker.NEXTVAL, 'Fatima', 22, TO_DATE ('2021-06-13', 'YYYY-MM-DD'), 20000, 0173456789, 0183456789, 0193456789, 8, 3, 2);
4. INSERT INTO Worker (worker_id, name, age, joining_date, salary, phone_no1, phone_no2, phone_no3, working_hour, w_address_id, manager_id) VALUES (sq_worker.NEXTVAL, 'Aminul', 30, TO_DATE ('2020-05-19', 'YYYY-MM-DD'), 30000, 0174567890, 0184567890, 0194567890, 8, 4, 2);
5. INSERT INTO Worker (worker_id, name, age, joining_date, salary, phone_no1, phone_no2, phone_no3, working_hour, w_address_id, manager_id) VALUES (sq_worker.NEXTVAL, 'Nadia', 27, TO_DATE ('2019-05-22', 'YYYY-MM-DD'), 27000, 0175678901, 0185678901, 0195678901, 8, 5, 3);

WORKER_ID	NAME	AGE	JOINING_DATE	SALARY	PHONE_NO1	PHONE_NO2	PHONE_NO3	WORKING_HOUR	W_ADDRESS_ID	MANAGER_ID
1	Rahim	25	13-MAY-20	25000	171234567	181234567	191234567	8	1	1
2	Karim	28	18-JUL-19	28000	172345678	182345678	192345678	8	2	1
3	Fatima	22	13-JUN-21	20000	173456789	183456789	193456789	8	3	2
4	Aminul	30	19-MAY-20	30000	174567890	184567890	194567890	8	4	2
5	Nadia	27	22-MAY-19	27000	175678901	185678901	195678901	8	5	3

Data of Customer_Address:

1. INSERT INTO Customer_Address (c_address_id, city, street_no, house_no) VALUES (sq_caddress.NEXTVAL, 'New York', '123 Main St', 'Apt 1A');
2. INSERT INTO Customer_Address (c_address_id, city, street_no, house_no) VALUES (sq_caddress.NEXTVAL, 'Los Angeles', '456 Elm St', 'Unit 2B');
3. INSERT INTO Customer_Address (c_address_id, city, street_no, house_no) VALUES (sq_caddress.NEXTVAL, 'Chicago', '789 Oak St', 'Suite 3C');
4. INSERT INTO Customer_Address (c_address_id, city, street_no, house_no) VALUES (sq_caddress.NEXTVAL, 'Houston', '111 Pine St', 'Apt 4D');
5. INSERT INTO Customer_Address (c_address_id, city, street_no, house_no) VALUES (sq_caddress.NEXTVAL, 'Miami', '222 Beach Rd', 'Unit 5E');

C_ADDRESS_ID	CITY	STREET_NO	HOUSE_NO
1	New York	123 Main St	Apt 1A
2	Los Angeles	456 Elm St	Unit2B
3	Chicago	789 Oak St	Suite 3C
4	Houston	111 Pine St	Apt 4D
5	Miami	222 Beach Rd	Unit 5E

Data of Customer:

1. INSERT INTO Customer (customer_id, name, age, phone_no1, phone_no2, phone_no3, c_address_id) VALUES (sq_customer.NEXTVAL, 'John Smith', 35, 1234567890, 9876543210, NULL, 1);
2. INSERT INTO Customer (customer_id, name, age, phone_no1, phone_no2, phone_no3, c_address_id) VALUES (sq_customer.NEXTVAL, 'Jane Doe', 28, 1112223333, NULL, NULL, 2);
3. INSERT INTO Customer (customer_id, name, age, phone_no1, phone_no2, phone_no3, c_address_id) VALUES (sq_customer.NEXTVAL, 'Bob Johnson', 42, 5556667777, NULL, NULL, 3);
4. INSERT INTO Customer (customer_id, name, age, phone_no1, phone_no2, phone_no3, c_address_id) VALUES (sq_customer.NEXTVAL, 'Sarah Lee', 25, 4445556666, 2223334444, NULL, 4);
5. INSERT INTO Customer (customer_id, name, age, phone_no1, phone_no2, phone_no3, c_address_id) VALUES (sq_customer.NEXTVAL, 'Tom Brown', 42, 7778889999, NULL, NULL, 5);

CUSTOMER_ID	NAME	AGE	PHONE_NO1	PHONE_NO2	PHONE_NO3	C_ADDRESS_ID
1	John Smith	35	1234567890	9876543210	-	1
2	Jane Doe	28	1112223333	-	-	2
3	Bob Johnson	42	5556667777	-	-	3
4	Sarah Lee	25	4445556666	2223334444	-	4
5	Tom Brown	42	7778889999	-	-	5

Data of Engineer_Address:

1. INSERT INTO Engineer_Address (eng_address_id, city, street_no, house_no) VALUES (sq_eaddress.NEXTVAL, 'New York', '123 Main St', 'Apt 5');
2. INSERT INTO Engineer_Address (eng_address_id, city, street_no, house_no) VALUES (sq_eaddress.NEXTVAL, 'Los Angeles', '456 Elm St', 'Unit 10');
3. INSERT INTO Engineer_Address (eng_address_id, city, street_no, house_no) VALUES (sq_eaddress.NEXTVAL, 'Chicago', '789 Oak St', 'Suite 200');
4. INSERT INTO Engineer_Address (eng_address_id, city, street_no, house_no) VALUES (sq_eaddress.NEXTVAL, 'Houston', '1010 Pine St', 'Apt 3B');
5. INSERT INTO Engineer_Address (eng_address_id, city, street_no, house_no) VALUES (sq_eaddress.NEXTVAL, 'Miami', '1111 Beach Ave', 'Unit 15');

ENG_ADDRESS_ID	CITY	STREET_NO	HOUSE_NO
1	New York	123 Main St	Apt 5
2	Los Angeles	456 Elm St	Unit 10
3	Chicago	789 Oak St	Suite 200
4	Houston	1010 Pine St	Apt 3B
5	Miami	1111 Beach Ave	Unit 15

Data of Engineer:

1. INSERT INTO Engineer (eng_id, name, age, joining_date, salary, phone_no1, phone_no2, phone_no3, email1, email2, email3, eng_address_id) VALUES (sq_engineer.NEXTVAL, 'John Smith', 32, TO_DATE('2022-02-05', 'yyyy-mm-dd'), 60000, 1234567890, 'john.smith@email.com', NULL, NULL, 1);
2. INSERT INTO Engineer (eng_id, name, age, joining_date, salary, phone_no1, phone_no2, phone_no3, email1, email2, email3, eng_address_id) VALUES (sq_engineer.NEXTVAL, 'Jane Doe', 27, TO_DATE('2022-05-06', 'yyyy-mm-dd'), 55000, 1112223333, 'jane.doe@email.com', NULL, NULL, 2);
3. INSERT INTO Engineer (eng_id, name, age, joining_date, salary, phone_no1, phone_no2, phone_no3, email1, email2, email3, eng_address_id) VALUES (sq_engineer.NEXTVAL, 'Bob Johnson', 45, TO_DATE('2022-03-01', 'yyyy-mm-dd'), 70000, 5554443333, 'bob.johnson@email.com', NULL, NULL, 3);

4. INSERT INTO Engineer (eng_id, name, age, joining_date, salary, phone_no1, phone_no2, phone_no3, email1, email2, email3, eng_address_id) VALUES (sq_engineer.NEXTVAL, 'Alice Lee', 29, TO_DATE('2022-04-15', 'yyyy-mm-dd'), 50000, 4443332222, 'alice.lee@email.com', NULL, NULL, 4);
5. INSERT INTO Engineer (eng_id, name, age, joining_date, salary, phone_no1, phone_no2, phone_no3, email1, email2, email3, eng_address_id) VALUES (sq_engineer.NEXTVAL, 'Mike Chen', 38, TO_DATE ('2022-10-19', 'yyyy-mm-dd'), 65000, 9998887777, 'mike.chen@email.com', NULL, NULL, 5);

ENG_ID	NAME	AGE	JOINING_DATE	SALARY	PHONE_NO1	PHONE_NO2	PHONE_NO3	EMAIL1	EMAIL2	EMAIL3	ENG_ADDRESS_ID
1	John Smith	32	05-FEB-22	60000	1234567890	-	-	john.smith@email.com	-	-	1
2	Jane Doe	27	06-MAY-22	55000	1112223333	-	-	jane.doe@email.com	-	-	2
3	Bob Johnson	45	01-MAR-22	70000	5554443333	-	-	bob.johnson@email.com	-	-	3
4	Alice Lee	29	15-APR-22	50000	4443332222	-	-	alice.lee@email.com	-	-	4
5	Mike Chen	38	19-OCT-22	65000	9998887777	-	-	mike.chen@email.com	-	-	5

Data of Product:

1. INSERT INTO PRODUCT (product_id, product_size, product_color,name, price, quality, order_date, customer_id) VALUES (sq_product.NEXTVAL, 9, 'BLUE', 'Salwar', 100, 'High', Sysdate, 2);
2. INSERT INTO PRODUCT (product_id, product_size, product_color, name, price, quality, order_date, customer_id) VALUES (sq_product.NEXTVAL, 22 ,'BLACK','Jeans',121, 'MEDIUM', Sysdate, 1);
3. INSERT INTO PRODUCT (product_id, product_size, product_color, name, price, quality, order_date, customer_id) VALUES (sq_product.NEXTVAL, 22, 'BLUE', 'Shirt',100,'High', Sysdate, 4);
4. INSERT INTO PRODUCT (product_id, product_size, product_color, name, price, quality, order_date, customer_id) VALUES (sq_product.NEXTVAL 23, 'PINK', 'saree', , 1000, 'High', Sysdate, 2);
5. INSERT INTO PRODUCT (product_id, product_size, product_color, name, price, quality, order_date, customer_id) VALUES (sq_product.NEXTVAL, 14, 'BLUE', 'punjabi', 1500, 'LOW', Sysdate, 3);

PRODUCT_ID	PRODUCT_SIZE	PRODUCT_COLOR	NAME	PRICE	QUALITY	ORDER_DATE	CUSTOMER_ID
1	9	BLUE	Salwar	100	High	15-MAY-23	2
2	22	BLACK	Jeans	121	MEDIUM	15-MAY-23	1
3	22	BLUE	Shirt	100	High	15-MAY-23	4
4	14	BLUE	punjabi	1500	LOW	15-MAY-23	3

Data of Bill:

1. INSERT INTO Bill (bill_id, amount, method, payment_date, customer_id) Values (sq_bill.NEXTVAL, 15000, 'card', TO_DATE ('2023-06-13', 'YYYY-MM-DD'),3);
2. INSERT INTO Bill (bill_id, amount, method, payment_date, customer_id) Values (sq_bill.NEXTVAL, 5000, 'card', TO_DATE (Sysdate, 'YYYY-MM-DD'),2);
3. INSERT INTO Bill (bill_id, amount, method, payment_date, customer_id) Values (sq_bill.NEXTVAL, 15000, 'cash', TO_DATE (Sysdate, 'YYYY-MM-DD'),1);

4. INSERT INTO Bill (bill_id, amount, method, payment_date, customer_id) Values (sq_bill.NEXTVAL, 1500, 'cash', TO_DATE (Sysdate, 'YYYY-MM-DD'),3);
5. INSERT INTO Bill (bill_id, amount, method, payment_date, customer_id) Values (sq_bill.NEXTVAL, 1000, 'bank', TO_DATE (Sysdate, 'YYYY-MM-DD'),4);

BILL_ID	AMOUNT	METHOD	PAYMENT_DATE	CUSTOMER_ID
1	15000	card	13-JUN-23	3
2	5000	card	23-MAY-15	2
3	15000	cash	23-MAY-15	1
4	1500	cash	23-MAY-15	3
5	1000	bank	23-MAY-15	4

Data of Supply:

1. INSERT INTO Supply (supplier_id, material_id) Values (3,5);
2. INSERT INTO Supply (supplier_id, material_id) Values (5,4);
3. INSERT INTO Supply (supplier_id, material_id) Values (1,3);
4. INSERT INTO Supply (supplier_id, material_id) Values (4,2);
5. INSERT INTO Supply (supplier_id, material_id) Values (2,1);

SUPPLIER_ID	MATERIAL_ID
1	3
2	1
3	5
4	2
5	4

Data of Made_Of:

1. INSERT INTO Made_Of (product_id, material_id) Values (3,4);
2. INSERT INTO Made_Of (product_id, material_id) Values (5,4);
3. INSERT INTO Made_Of (product_id, material_id) Values (2,3);
4. INSERT INTO Made_Of (product_id, material_id) Values (3,5);
5. INSERT INTO Made_Of (product_id, material_id) Values (1,2);

PRODUCT_ID	MATERIAL_ID
1	2
2	3
3	1
3	4
3	5

Data of Make:

1. INSERT INTO Make (worker_id, product_id) Values (5,2);
2. INSERT INTO Make (worker_id, product_id) Values (2,3);
3. INSERT INTO Make (worker_id, product_id) Values (1,4);
4. INSERT INTO Make (worker_id, product_id) Values (3,5);
5. INSERT INTO Make (worker_id, product_id) Values (4,1);

WORKER_ID	PRODUCT_ID
1	4
2	3
3	1
4	1
5	2

Data of Design:

1. INSERT INTO Design (eng_id, product_id) Values (1,3);
2. INSERT INTO Design (eng_id, product_id) Values (5,2);
3. INSERT INTO Design (eng_id, product_id) Values (3,4);
4. INSERT INTO Design (eng_id, product_id) Values (4,5);
5. INSERT INTO Desig n (eng_id, product_id) Values (2,1);

ENG_ID	PRODUCT_ID
1	3
2	1
3	4
4	1
5	2

Query Writing:

Single row function:

1. SELECT name, salary FROM Engineer WHERE salary BETWEEN 55000 and 60000 ORDER BY joining_date;

NAME	SALARY
John Smith	60000
Jane Doe	55000

2. SELECT name, phone_no1 FROM Customer WHERE name LIKE '_o%';

NAME	PHONE_NO1
John Smith	1234567890
Bob Johnson	5556667777
Tom Brown	7778889999

Group function:

1. SELECT age, AVG (salary) FROM Manager GROUP BY age;

AGE	AVG(SALARY)
32	3250
35	3500
33	4000

2. SELECT product_color, MAX (price) MAX FROM Product GROUP BY product_color;

PRODUCT_COLOR	MAX
BLACK	121
BLUE	1500

Subquery:

1. SELECT name FROM Engineer WHERE salary > (SELECT salary FROM Engineer WHERE name = 'Alice Lee');

NAME
John Smith
Jane Doe
Bob Johnson
Mike Chen

2. SELECT name FROM Worker WHERE joining_date > (SELECT joining_date FROM Worker WHERE name = 'Aminul');

NAME
Fatima

Joining:

1. SELECT w.name, m.name FROM worker w, manager m WHERE w.manager_id=m.manager_id;

NAME	NAME
Karim	Ashiq
Rahim	Ashiq
Aminul	Shuvho
Fatima	Shuvho
Nadia	Ashiqur

2. SELECT c.name ||' ordered '|| p.name FROM customer c, product p WHERE c.customer_id(+)=p.product_id;

C.NAME||'ORDERED'||P.NAME
John Smith ordered Salwar
Jane Doe ordered Jeans
Bob Johnson ordered Shirt
Sarah Lee ordered punjabi

View:

1. CREATE VIEW ManagerView AS SELECT worker_id, name, age, joining_date, phone_no1, salary FROM Worker;

   SELECT*FROM ManagerView

WORKER_ID	NAME	AGE	JOINING_DATE	PHONE_NO1	SALARY
1	Rahim	25	13-MAY-20	171234567	25000
2	Karim	28	18-JUL-19	172345678	28000
3	Fatima	22	13-JUN-21	173456789	20000
4	Aminul	30	19-MAY-20	174567890	30000
5	Nadia	27	22-MAY-19	175678901	27000

2. CREATE VIEW GoodProductView AS SELECT product_id, name, product_size, product_color FROM Product WHERE quality = 'High';

   SELECT * FROM GoodProductView;

PRODUCT_ID	NAME	PRODUCT_SIZE	PRODUCT_COLOR
3	Shirt	22	BLUE
1	Salwar	9	BLUE

Relational Algebra:

1. SELECT name FROM Engineer WHERE salary = 60000;

   $\prod_{\text{name}} (\sigma_{\text{salary = 60000}} (\text{Engineer}))$

2. SELECT name FROM Product WHERE product_size>9;

   $\prod_{\text{name}} (\sigma_{\text{product\_size>9}} (\text{Product}))$

3. SELECT name, age FROM Worker WHERE joining_date < '17-MAY-2020';

   $\prod_{\text{name, age}} (\sigma_{\text{joining\_date<”17-MAY-2020”}} (\text{Worker}))$

4. SELECT name FROM Customer WHERE phone_no2 IS NULL;

   $\prod_{\text{name}} (\sigma_{\text{phone\_no2 IS NULL}} (\text{Customer}))$

5. SELECT salary FROM Manager WHERE name = 'Shuvho';

   $\prod_{\text{salary}} (\sigma_{\text{name = 'Shuvho'}} (\text{Manager}))$

Conclusion:

In conclusion, creating a garments management system database brings several benefits for effective data management in a garment factory. The implemented database effectively organizes and stores information about customers, products, workers, managers, engineers, raw materials, suppliers, and bills. By following normalization principles, we ensured data integrity and reduced unnecessary duplication. We demonstrated the database's functionality through tasks like creating tables, inserting data, and writing queries. The use of relational algebra operations allowed for logical data manipulations. Overall, this project improves operational efficiency, aids decision-making, and proves valuable to the garment industry.