

Database Systems-I

(Project Report)

Project Members:

Husnain Ahmed (SP20-BCS-033)

Muaz Muhammad (SP20-BCS-039)

Yalmaz Ali (SP20-BCS-093)

Asfar Aman (SP20-BCS-140)

Section: C

Date: 5-1-2022

1) Abstract:

This report is for IVI Textile Industries, a small-scale fabric manufacturing factory. The primary focus of this report is on the company's sales, transactions, and employees. The current situation is evaluated, and a full requirement specification is prepared. Based on the user needs, a conceptual database design is offered, and other recommendations are provided.

2) Domain Description:

Company Profile and Operations:

IVI Textile Industries is a small-scale factory of fabric manufacturing. They are in the business since 1988, it's an old factory with few resources.

The textile is being operated by managers team. Sales manager receives the orders and clients' details from salesman (who receives orders from clients) and provides the order details to production manager. Sales manager takes care of the transactions of all the orders by clients. HR officer keeps the employee's data and manages their salary. Production manager is responsible for the fabric resources and completion of order in due time.

Current Situation:

Currently, all the managers are operating manually, which is creating problems for the company in these modern times. The managers require much time to process information and pass the information to other managers.

Sales manager is operating manually to store order and client details. Sales manager keeps the transactions of cash payments but becomes difficult for them to manage online payments manually. Keeping the record of workers is tough for HR officer to maintain and update on registers. Moreover, the company is experiencing several issues in the transfer of information between managers including lack of accuracy, ambiguity in information, a sluggish reaction time, and non-systematic data arrangement. So, the management work needs to be automated.

3) Requirement Specification:

In this report the whole system for textile management including Sales, HR and Production work is broken down as these functions for specification:

Client:

Clients are Customers who gives Orders to the salesman and are handled by the Sales manager.

Fabric:

Details of a fabric like material and cost are stored. And are used by the Production manager.

Order:

Orders are then stored in the database given by clients with reference to Fabric and are forwarded to Sales manager for further processing.

Transaction:

Transaction function is being used by sales manager with reference to Client, it stores the transaction details like advance, total payment and on which date it was occurred.

Employee:

This function is used by HR Officer to store details of the employees and stores the salary of each employee which can be used at the end of a month.

Functional Requirements:

Id	Functions	Entity	Priority
<i>Client</i>			
1	Insert / Update / Delete / View the Client details.	Client, Order, Transaction	High
2	Get Details of a client from his name or ID.	Client	Normal

3	Check balance of a specified client.	Client, Transaction	Low
<i>Fabric</i>			
1	Insert / Update / Delete / View the Fabric and its details.	Fabric	High
2	Get fabric rate history to check at what rate the client has purchased a particular fabric.	Fabric	Normal
3	Get fabric name whose order is done by a Specified Client.	Fabric, Order Detail	Normal
4	Get rate of a specific Fabric.	Fabric	Low
<i>Order:</i>			
1	Insert / Update / Delete / View the Orders given by Clients.	Order, Client, Employee	High
2	Get Order details of a specific Client.	Order, Client	Normal
3	Get Order details of an Order id.	Order, Order Detail	Normal
<i>Transaction:</i>			
1	Insert / Update / Delete / View the Transactions done by a client.	Transaction, Employee, Client	High
2	Get monthly and yearly revenue.	Transactions	High
3	Get transaction details by a specific Order Id or Client Id.	Transaction, Order, Client	Normal
<i>Employee:</i>			
1	Insert / Update / Delete / View the Employee data.	Employee	High
2	Get details of a specific Employee by his name or ID.	Employee, Order	Low

Non-Functional Requirements:

1. System can store data without any hinderance efficiently and is increased by adding data in the database.
2. System can function on any system having a display and input device.
3. Each process should be time efficient to load.
4. System is reliable and should not crash on such small database.
5. It is based on a friendly User Interface.

4) Assumptions:

Client:

- Client must deal with salesman and the salesman gives the orders to sales manager to add new Order.
- Client should not get his order until his dues are cleared.

Fabric:

- Raw Material for a fabric to be produced is already in the mill.

Order:

- An Order should have 1 minimum employee working on it.
- Order for a client must be done in a specified days in the policy.

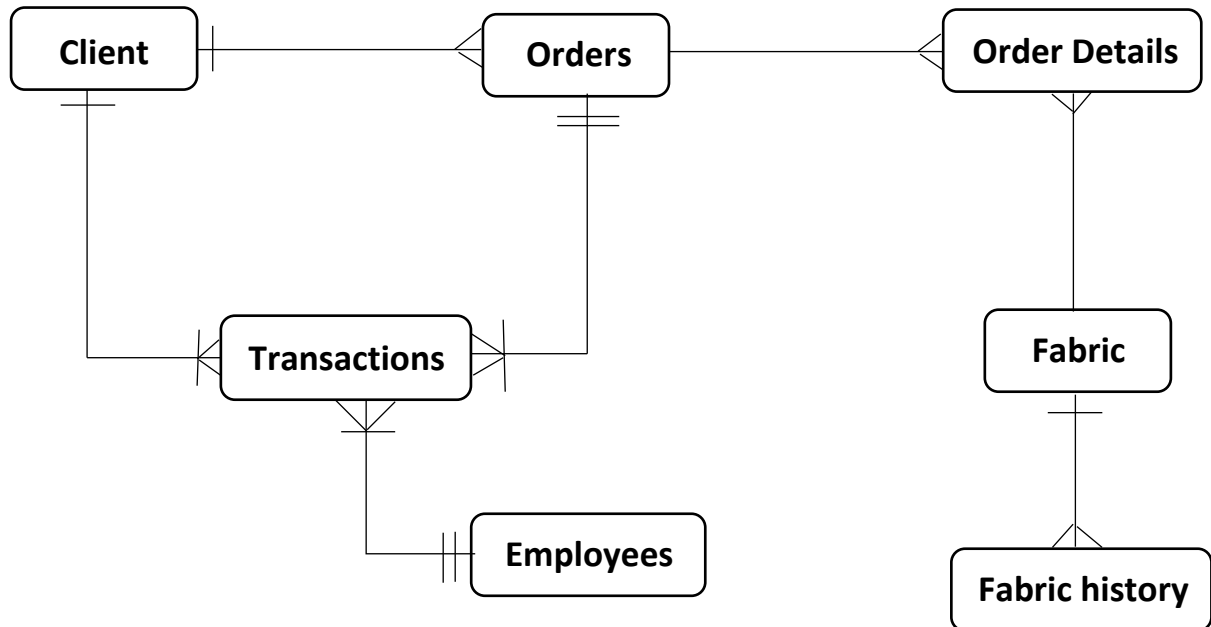
Transaction:

- Client may pay total amount or some amount of the order.

Employee:

- Employees and managers detail is stored in Employee table by different designation.
- Bonus are added in salary if an employee works overtime.

5) Entity Relationship Diagram:



5.1) Entities and Attributes

Clients:

C_id	varchar (10)
C_name	varchar (20)
C_contact	Integer (15)
C_address	varchar (80)
C_company	varchar (20)

Clients will be a common entity with Orders and Transactions. Client's entity relationship with Orders entity is one to many and with Transaction entity is again one to many. C_id is a Primary Key of Clients entity. C_name will be used to store the names of clients. Similarly, C_contact will keep the contact number of clients so that in case of any emergency

contact will be easier. C_address will be used to keep the information of client's complete address. C_company will be used to store the information of company who ordered for fabrics.

Fabrics:

F_id	Integer (10)
F_name	varchar (20)
F_rate	Integer (5)
F_type	Varchar (15)

Fabrics will be a common entity with Orders entity. And its relationship with Orders entity is many to many. F_id will be used as a Primary Key in fabrics entity, and it cannot be repeat in any case. F_name will keep the information about type of fabric either it is chiffon, cotton, silk etc. And F_rate will be used to keep the information of fabric rate per meter because every single fabric type will have different cost. F_type will be used to assign the fabric family or type of the fabric.

Fabric History:

F_id	Integer (10)
F_rate	Integer (5)
H_date	date

Fabrics will be a common entity with Orders entity. And its relationship with Orders entity is many to many. F_id will be used as a Primary Key in fabrics entity, and it cannot be repeat in any case. F_name will keep the information about type of fabric either it is chiffon, cotton, silk etc. And F_rate will be used to keep the information of fabric rate per meter because every single fabric type will have different cost. F_type will be used to assign the fabric family or type of the fabric.

Orders:

O_id	varchar (10)
------	--------------

C_id	varchar (10)
Order_date	date
Due_date	date

Orders will be a common entity with other four entities like Clients, Fabrics, Employees and Transactions. Orders table relationship with Clients table is many to one. Its relationship with Fabrics is many to many and with Transactions is one to many. O_id is a Primary Key and it will be unique, and it cannot be repeat in any case. C_id is a Foreign Key of Clients table. Order_date will be used to remember the date of order and it will help the company to complete their task in time. Due_date will be used to keep due date information of order.

Order Details:

O_id	varchar (10)
F_id	varchar (10)
Meters	Integer (10)
Color	varchar (10)

Relationship between Orders entity and Fabrics entity is many to many therefore we must break both tables. O_id is a foreign key from table orders and F_id is a foreign key from table Fabrics. Meters attribute will be used to keep the information of fabrics in meter which clients have been ordered. And total_amount will be used to keep total price of order which client will have to pay for order.

Transactions:

T_id	varchar (10)
C_id	varchar (10)
O_id	varchar (10)
E_id	varchar (10)
Payment	Integer (20)
Transaction_date	date

Transaction entity will be a common entity with Orders entity and Client's entity. Its relationship with Orders table will be one to one and with clients will be many to one. C_id is a Foreign Key from Clients table. similarly, O_id is a Primary Key from Orders table. E_id is Foreign Key from Employees table. Payment attribute will be the total price of the order. And Transaction_date will be used to keep transaction date information.

Employee:

E_id	Varchar (10)
E_name	varchar (30)
E_contact	Integer (15)
CNIC	varchar (20)
E_address	varchar (80)
designation	varchar (20)
DOJ	date
Salary	Integer (10)

Employee entity will be used to keep all information of employee and its relationship with other table is not acceptable. E_id will be a Primary Key of this entity. E_name attribute will be storing names of employees and similarly, E_contact will store their contact number. CNIC attribute will be used to store the information of their national identity card and E_address will keep their complete residential address. E_designation will show their post this employee is working on this rank. DOJ will be used to store the information about employees joining date in company. And last Salary attribute will be used to store the information of their monthly salary according to their production and designation.

5.2) Relationship & Labels

Clients:

- A single client can place no order or many orders, but a single order is placed by only one customer.
- A single client can do many transactions but at least one, and single transaction is done by single client that single transaction cannot be done by many customers.

Orders:

- A single order can have single Fabric and single order can also have multiple Fabrics. Similarly, a single Fabric can place by single order and single Fabric can also place by many orders.
- A single order can have single Transaction and that single order can also have many Transactions but at least one. But single Transaction is done by one and only one order it cannot place by many orders.

Transaction:

- A specific transaction is done by one and only one Employee, but a single Employee can do single Transaction and that single Employee can do multiple Transactions but at least one.

5.3) Validation:

The above system is broken down into five Functions to meet the User requirements, the relationship defined above is stating all the information necessary for this business. So, the model suggested is validated against the User Requirement.

There exists no conflict in this system as each department is handling its own functions, and all relationships of this model is resolved in an authentic manner so that it can be non-redundant and manageable at the same time.

When C-1 Client gives and Order O-1, A manager E-1 manages the Order and gives the details into Finance Department. There comes a transaction and order is placed on which F-1 fabric. When the production is ended, it is reported to the Sales Department and then the client can take its order by giving its pending dues.

Every function including Client, Order, Transaction, Fabric, and Employee are handled by 3 main departments. hence this business break down has eliminated the chances of redundancy and the repetition of any entities or functions.

Each Function is having Primary keys. Order and Transaction consist of composite primary key so keeping the data natural and more user Friendly. This system can be implemented easily throughout the departments. As producing it in a final product it is easy to maintain.

5.4) Description:

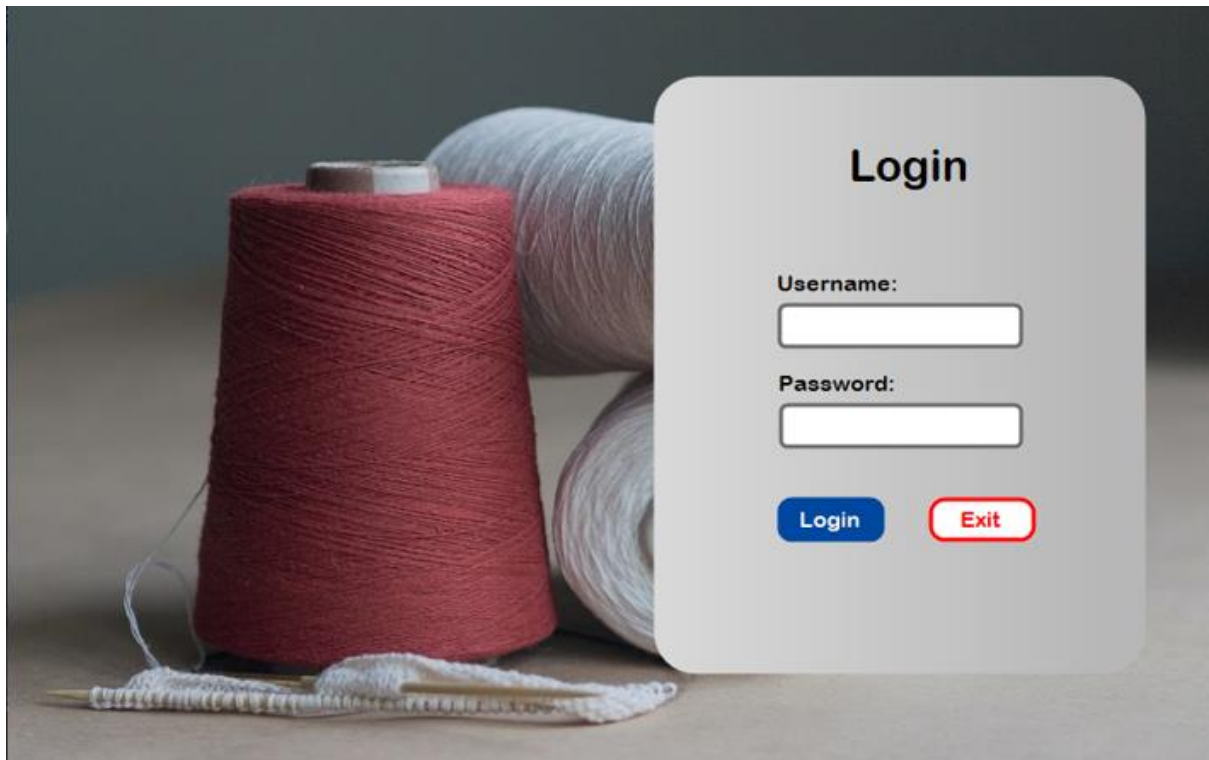
Entity Relationship Diagram illustrates the structure of the Sales, Accounts and HR section of the proposed system. It can be observed from the diagram that IVI's sales department is performing two main functions. Those are of receiving the orders from clients, storing it appropriately and then issuing it to the Accounts department.

In Orders, it is a must requirement to record the order details. Since any Orders can have one or more Fabrics and there could be many Orders of any one Fabric; therefore, to resolve this M:M relationship an Order_Details link entity is used once the order is received and its details recorded then comes the stage of Transaction of Orders. HR department maintains the Employee's data and one employee is assigned to one and only one order, an employee is also involved in Transaction of an order.

6) GUI Screenshots:

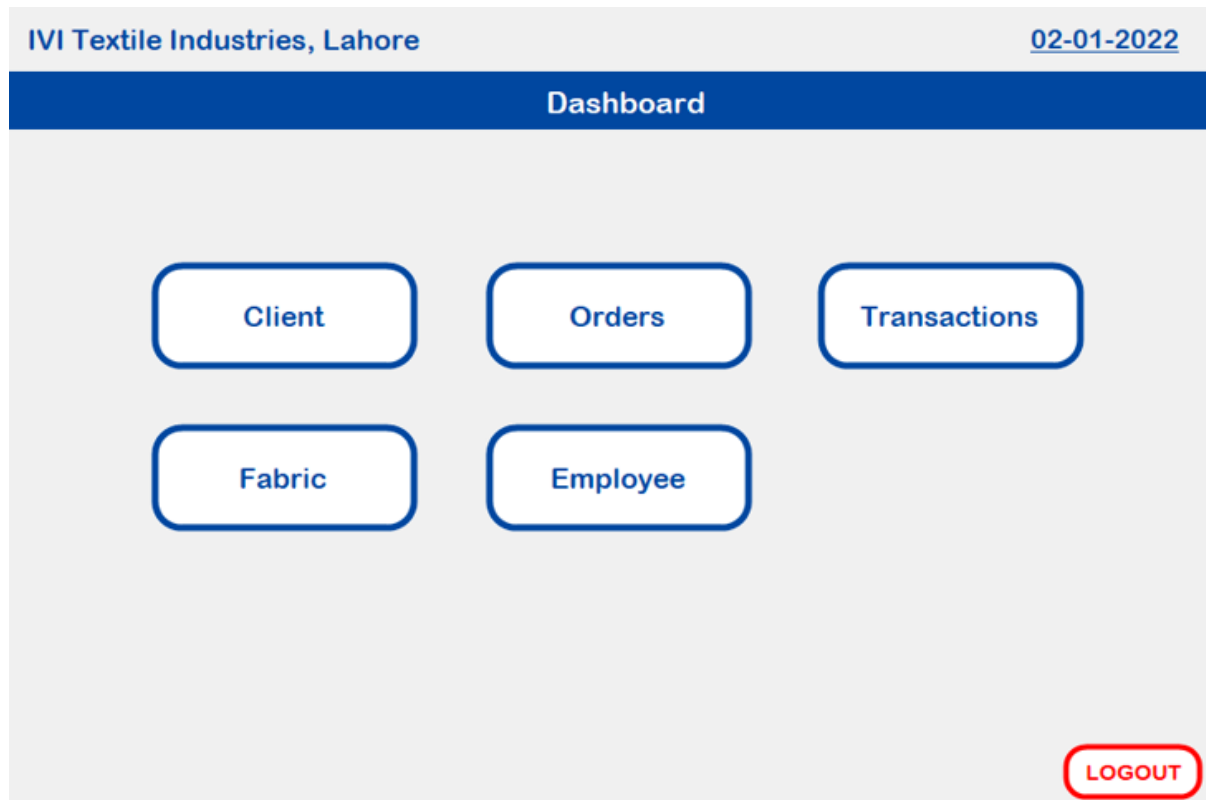
(1) Login page

This is login page from where user can get access to enter in IVI Textile management system application.



(2) Dashboard

This is dashboard of system from where user can go anywhere in the application either he/she wants to go in clients page, orders page, transactions page, fabric page or employee's page.



(3) Clients page

This page is for the information of clients from where we can see the details of clients, add new clients, and can update them.

The screenshot shows a web application interface for 'IVI Textile Industries, Lahore'. The top navigation bar includes the company name and the date '02-01-2022'. A blue sidebar on the left contains the IVI logo and navigation links: 'New Client', 'Update Client', 'Client Details', and 'Dashboard'. The main content area is titled 'Client' and contains a form with the following fields:

- Client Name:
- Company Name:
- Contact:
- City:
- Email:

Below the form is a blue button labeled 'Enter Data'.

(4) Orders page

This page will keep the information of orders placed by clients, meters of fabric and color of that fabric.

It is also used for adding new orders, update orders, delete orders.

OrderForm

IVI Textile Industries, Lahore

02-01-2022

Orders

Client ID: C01

Order ID:

Due Date: 2022-01-02

Fabric ID: F01

Meters:

Color:

Select

Add Order

Dashboard

(5) Transaction's page

This page will keep the details of all transactions and will generate monthly and yearly revenue.

The screenshot shows a web application interface for "IVI Textile Industries, Lahore". The page title is "Transactions". On the left, there is a blue sidebar with the IVI logo and three menu items: "New Transaction", "Transaction Details", and "Revenue". At the bottom of the sidebar is a "Dashboard" link. The main content area contains a form with the following fields: "Client ID:" with a dropdown menu showing "C01", "Employee ID:" with a dropdown menu showing "E03", "Order ID:" with a dropdown menu, "Employee Name:" with a text input field, and "Payment:" with a text input field. Below these fields is a blue button labeled "Complete Transaction". The top right corner of the page shows the date "02-01-2022".

Revenue Page:

The screenshot shows the "Revenue" page in the same application. The sidebar is identical to the previous page. The main content area has two tabs: "Monthly Revenue" and "Yearly Revenue". The "Monthly Revenue" tab is active. It contains a form with a "Month:" label, a calendar icon, and a dropdown menu showing "01-2022". Below this is a blue "Check" button. At the bottom of the form, it says "Total Revenue:" followed by a rounded rectangular box containing the value "930000". The "Yearly Revenue" tab is also visible, with a "Year:" label, a calendar icon, and a dropdown menu showing "2022". It also has a "Check" button and a "Total Revenue:" label with a rounded rectangular box containing the value "930000". The top right corner of the page shows the date "02-01-2022".

(6) Fabric page

This page will keep the information of fabrics, fabric types and fabric per meter details. It is also used to add new fabric, update, and delete.

FabricForm

IVI Textile Industries, Lahore

02-01-2022

Fabric

Fabric Name:

Rate (per meter):

Fabric Type:

Enter Data

Dashboard

Fabric Rate History:

FabricForm

IVI Textile Industries, Lahore

02-01-2022

Fabric

Fabric ID:

Month:

View Rates

	Fabric ID	Fabric	Rate	Update
▶	F01	Corduroy	350	01/01/2022
*				

Dashboard

(7) Employee's page

This page will keep the information of all employees their name, contact number, address, CNIC, designation and salary.

It is also used to add new employees and update them.

Employee

IVI Textile Industries, Lahore

02-01-2022

Employee

Employee Name:

CNIC:

Contact:

Designation:

Address:

Salary:

Enter Data

Dashboard

7) Normalization Logic:

Orders table:

In this table as there are no multivalued attribute and no partial and transitive dependencies, so it is in 3rd normal form.

Clients table:

In Client table as there are no multivalued attribute and no partial and transitive dependencies, so it is in 3rd normal form. All non-key attributes are fully functional dependent on primary key.

Fabric table:

Similarly in Fabric table there are no multivalued attributes all non-key attributes are dependent on primary key, and all non-primary attributes are determined by primary key attribute.

Fabric History table:

In Fabric table there is no multivalued attributes all non-key attributes are dependent on primary key, and all non-primary attributes are determined by primary key attribute.

Order Details table:

This table is already in 3rd normal form, so no changes are required

Transactions table:

In transactions table there are no functional and transitive dependencies, so it is in 3rd normal form.

Employees table:

This table is also in 3rd normal form.

8) Denormalization Opportunities:

After analysis all the tables, the transactions table can be denormalized.

Transactions table:

T_id	C_id	O_id	E_id	Payment	Tranaction_date
------	------	------	------	---------	-----------------

Orders table:

O_id	C_id	Order_date	Due_date
------	------	------------	----------

By denormalizing the attributes of transactions table with Orders table we get the following table.

O_id	C_id	E_id	Order_date	Due_date	Payment	Tranaction_date
------	------	------	------------	----------	---------	-----------------

9) Conclusion:

This model here tries to display the information needed in the framework according to the perspective of the three departments. At the point when every one of the three perspectives will be coordinated, a portion of the substances and connections may require some alteration regarding the necessities of different frameworks and functions.

10) Recommendation:

Getting User requirements, we can recommend that backup should be available for the database and admin should get some training in advance to how to use the system. Thinking about the simplicity of data access with the help of a mechanized framework it is suggested that a security strategy ought to be implemented, restricting the framework admittance to approved staff as for admins only. By this we can efficiently improve speed, accuracy, and security of our system for the future.

WORD COUNT: 2651