AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

FACULTY OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

ADVANCE DATABASE MANAGEMENT SYSTEM

Spring 2022-2023

Section: B

Supervised By

Juena Ahmed Noshin

GARMENTS FACTORY MANAGEMENT SYSTEM

Group Members

Name	ID	Contribution
Kamil Ahmed	20-42058-1	Table Creation 100%, Query Writing 100%, PL/SQL 50%
Md. Aman Ulla Shawon	20-42028-1	Class Diagram, Use Case Diagram and Activity Diagram 100%
Farhan Sadik Ferdous	20-42072-1	Introduction 50%, Project Proposal 50%, Scenario Description 50%, ER Diagram 50%, Normalization 50%, Scheme Diagram 50%, PL/SQL 20%, Relational Algebra 50%, Conclusion 50%
Tapu Biswas	20-42073-1	Introduction 50%, Project Proposal 50%, Scenario Description 50%, ER Diagram 50%, Normalization 50%, Scheme Diagram 50%, PL/SQL 20%, Relational Algebra 50%, Conclusion 50%
Md. Morshedul Islam	20-42645-1	User Interface 100%, Data Insertion 100%

Date of Submission: May 14, 2023

Content Page

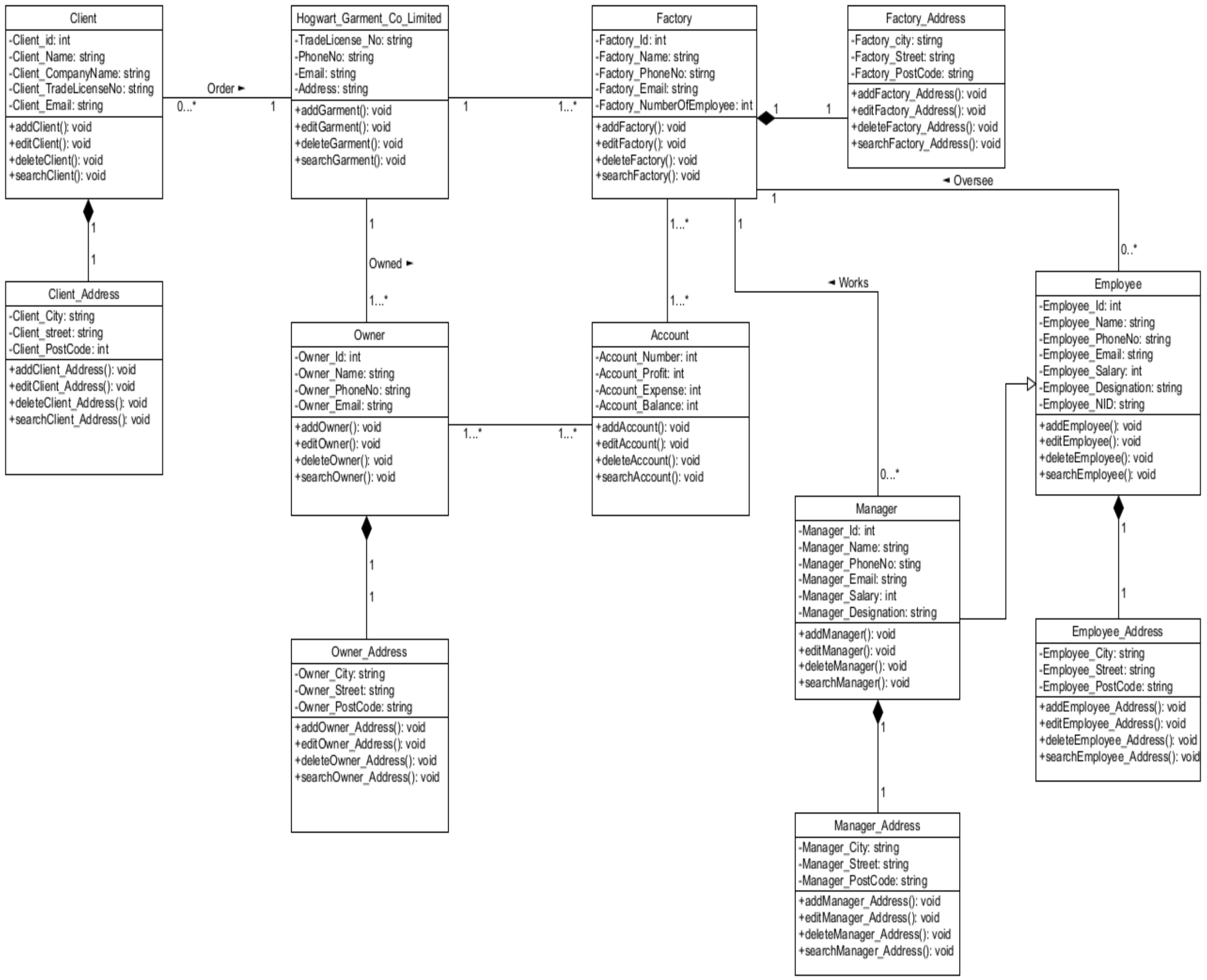
Introduction

Database is basically a collection of data or information from an organization. A database management system (DBMS) is an application that is used to store, retrieve, and modify users' data. In a database, the data is stored in the table. A DBMS helps the user to collect data that is easily accessible in a protected environment. A DBMS helps the user to use the data efficiently. In our project, we are going to discuss the garment factory management system. Hogwart garment Co Limited is situated on Privet Drive, Little Whinging, Surrey. Nowadays the garment factory has a large number of branches, employees, and customers along with many managers to maintain them efficiently. To record their details and supervision we need to use the garment factory management system. A garment factory management system keeps the details of the owner, company, customer, manager, employee, and account details. Everything becomes well-organized and time-efficient. With the help of the garment factory management system, we can record the present data along with the previous data orderly. So, the garment factory management system is required for users to use the data effectively which can be accessed easily.

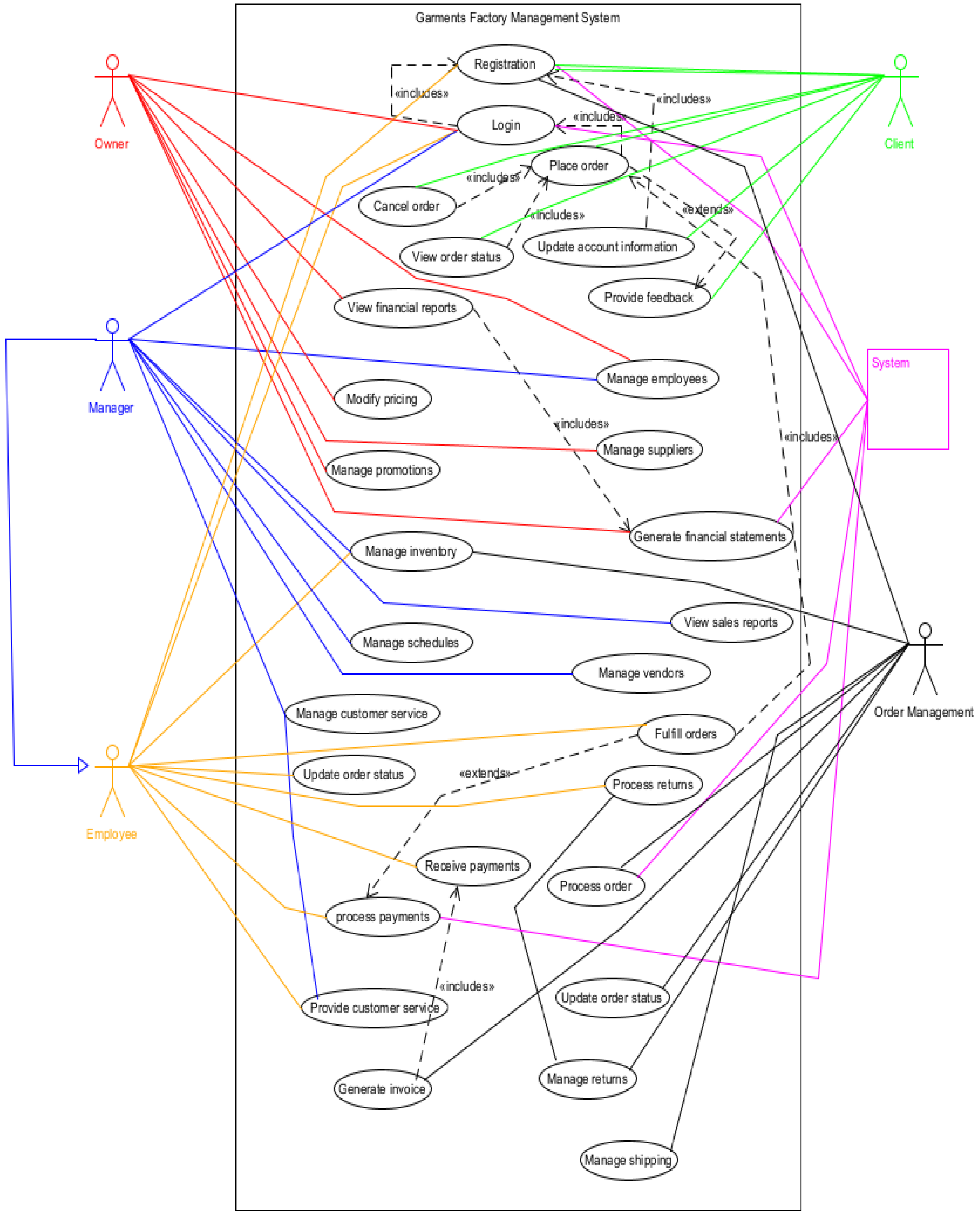
Project Proposal

Garment factory management system is a database management system that keeps the track of the inside activity of a garment factory. In our project, we are introducing a garment factory management system that will represent the management system of the Hogwart garment factory. Here, Hogwart garment Co Limited is owned by multiple owners. It also takes orders from the client and the factory manufactures the products ordered by the client. Managers manage the factory as well as oversees the employee working in the factory. Each branch has its single account and the company has a single account.

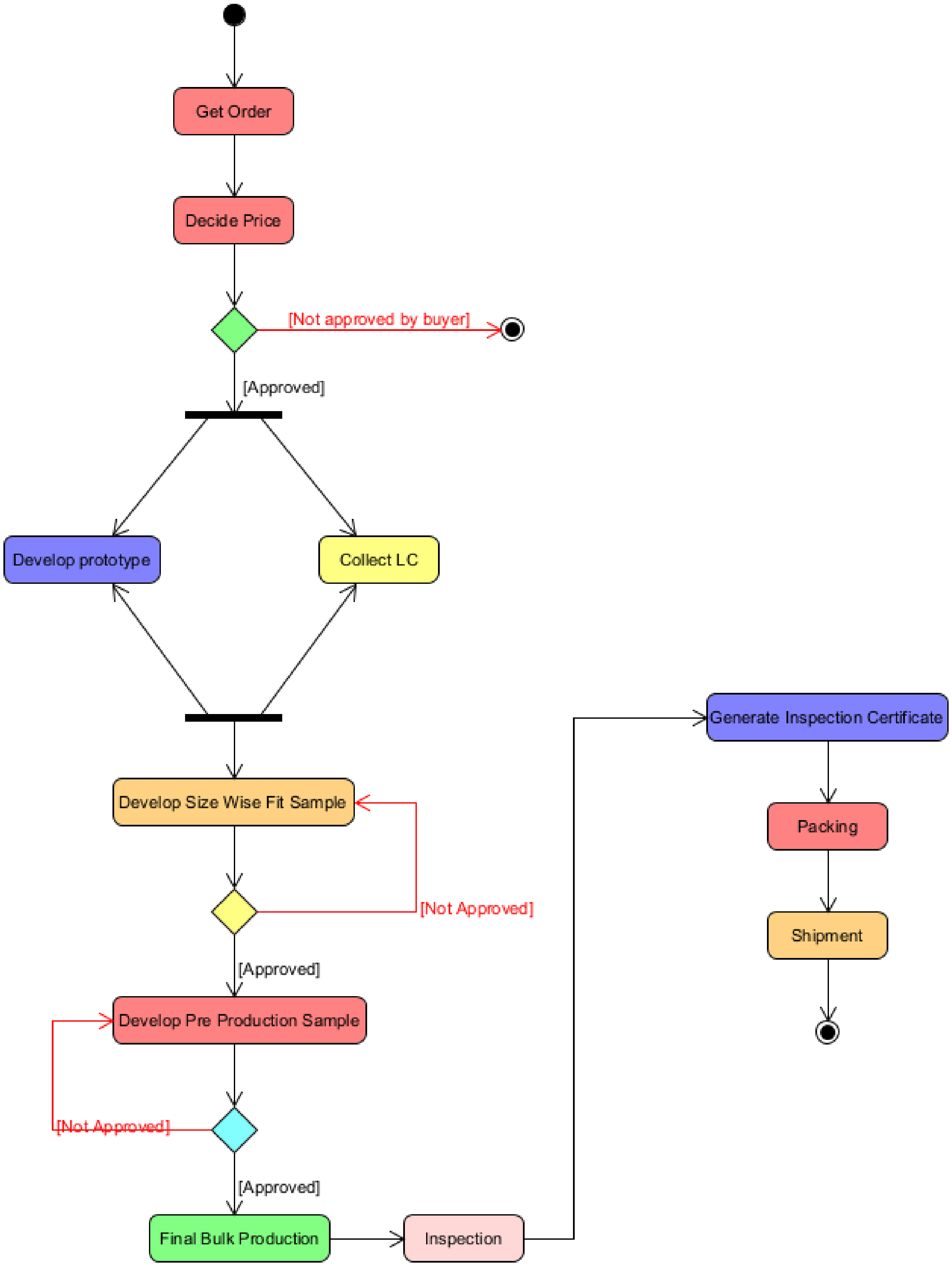
Class Diagram



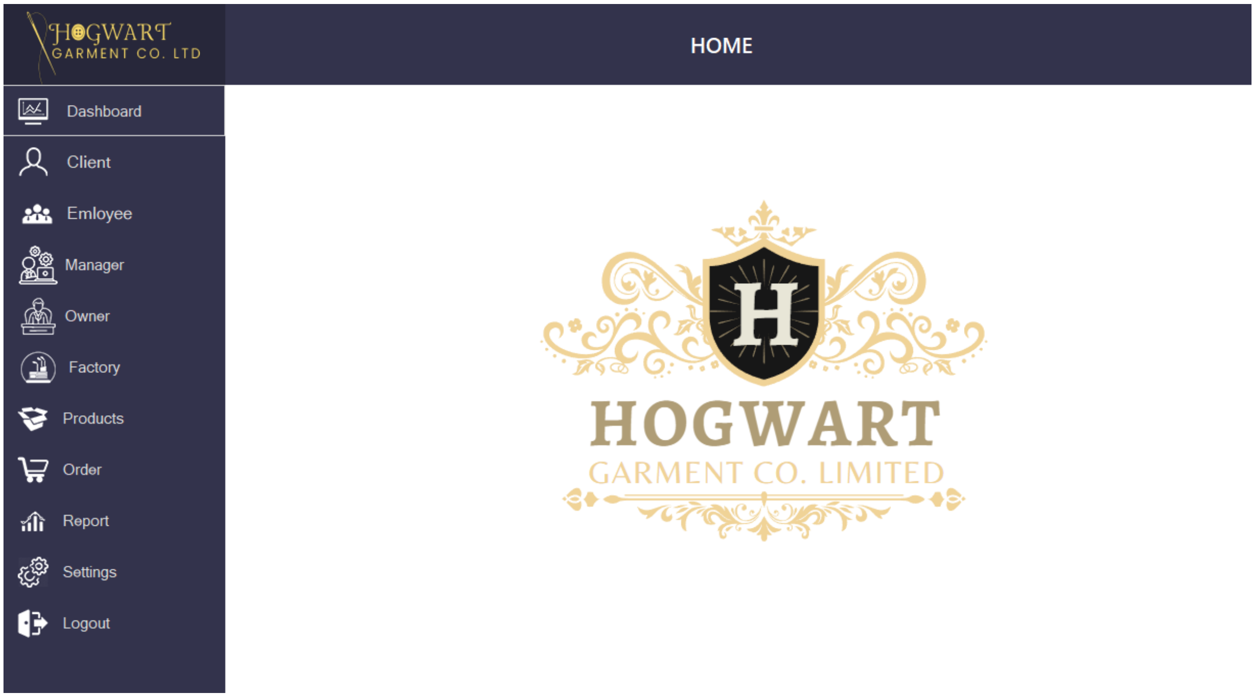
Use Case Diagram

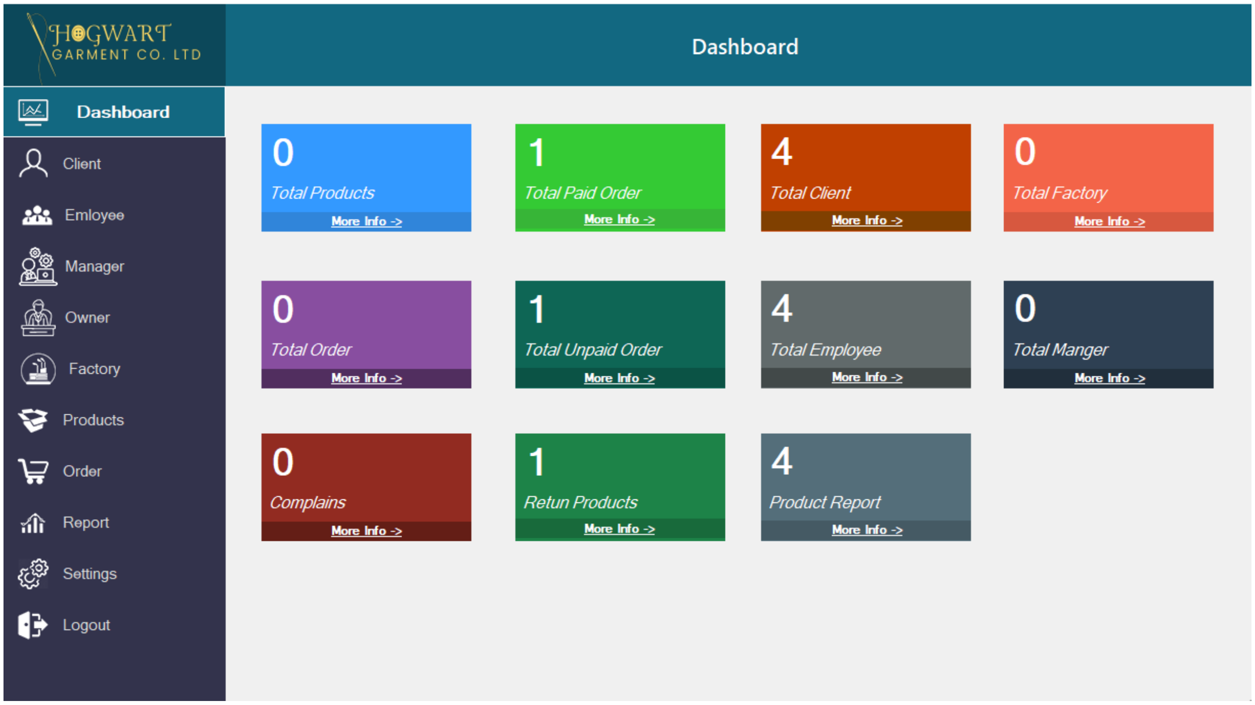


Activity Diagram



User Interface





0 Total Products More Info ->

1 Total Paid Order More Info ->

4 Total Client More Info ->

0 Total Factory More Info ->

0 Total Order More Info ->

1 Total Unpaid Order More Info ->

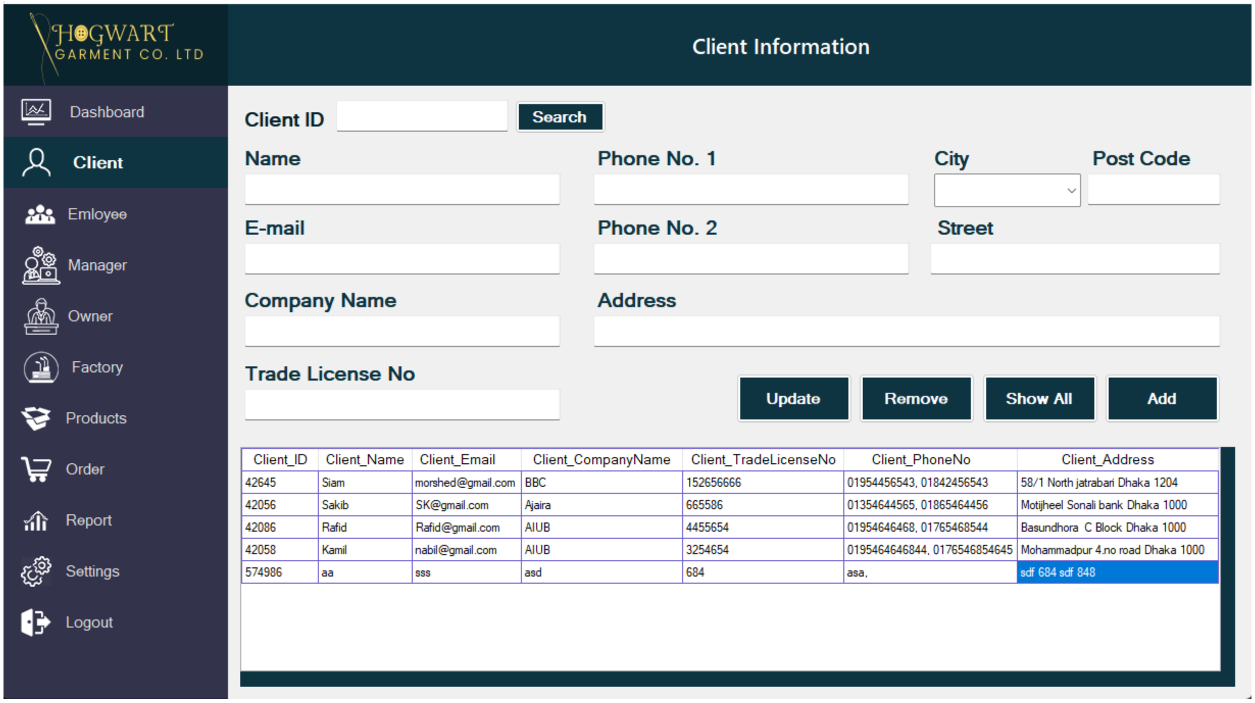
4 Total Employee More Info ->

0 Total Manger More Info ->

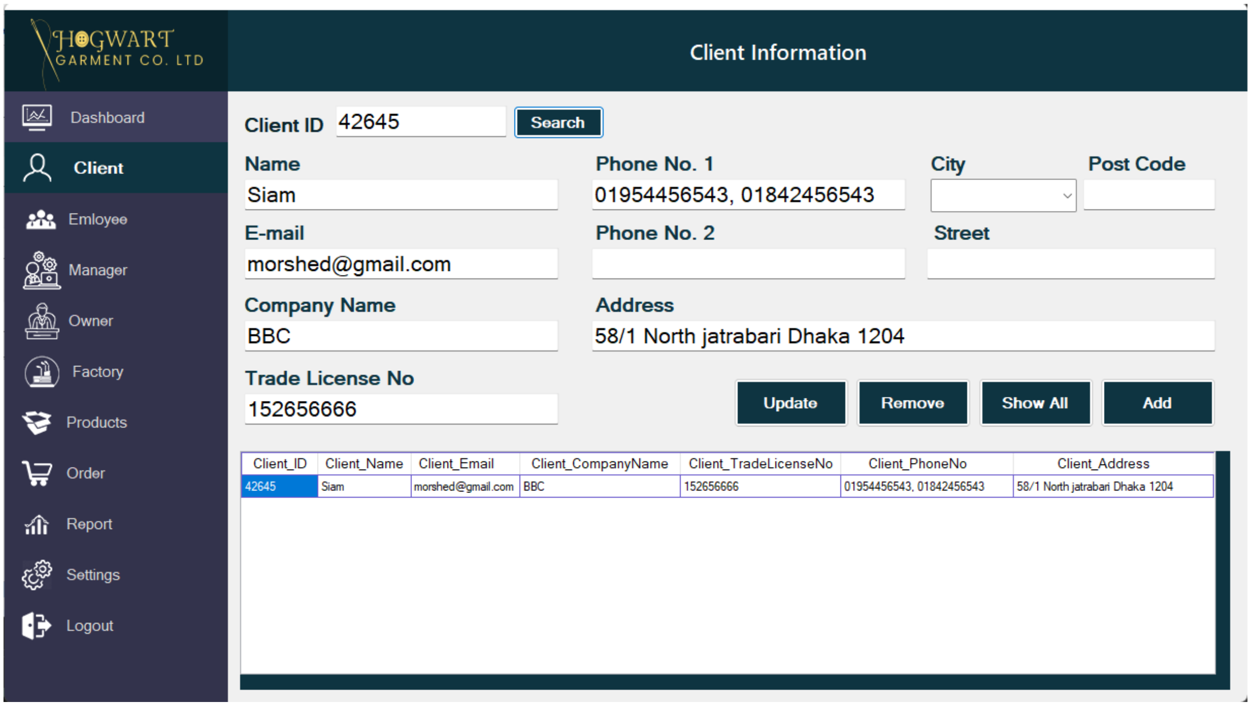
0 Complains More Info ->

1 Retun Products More Info ->

4 Product Report More Info ->



Client_ID	Client_Name	Client_Email	Client_CompanyName	Client_TradeLicenseNo	Client_PhoneNo	Client_Address
42645	Siam	morshed@gmail.com	BBC	152656666	01954456543, 01842456543	58/1 North jatrabari Dhaka 1204
42056	Sakib	SK@gmail.com	Ajaira	665586	01354644565, 01865464456	Motijheel Sonali bank Dhaka 1000
42086	Rafid	Rafid@gmail.com	AIUB	4455654	01954646468, 01765468544	Basundhora C Block Dhaka 1000
42058	Kamil	nabil@gmail.com	AIUB	3254654	0195464646844, 0176546854645	Mohammadpur 4.no road Dhaka 1000
574986	aa	sss	asd	684	asa,	sdf 684 sdf 848



Client ID 42645

Name: Siam

Phone No. 1: 01954456543, 01842456543

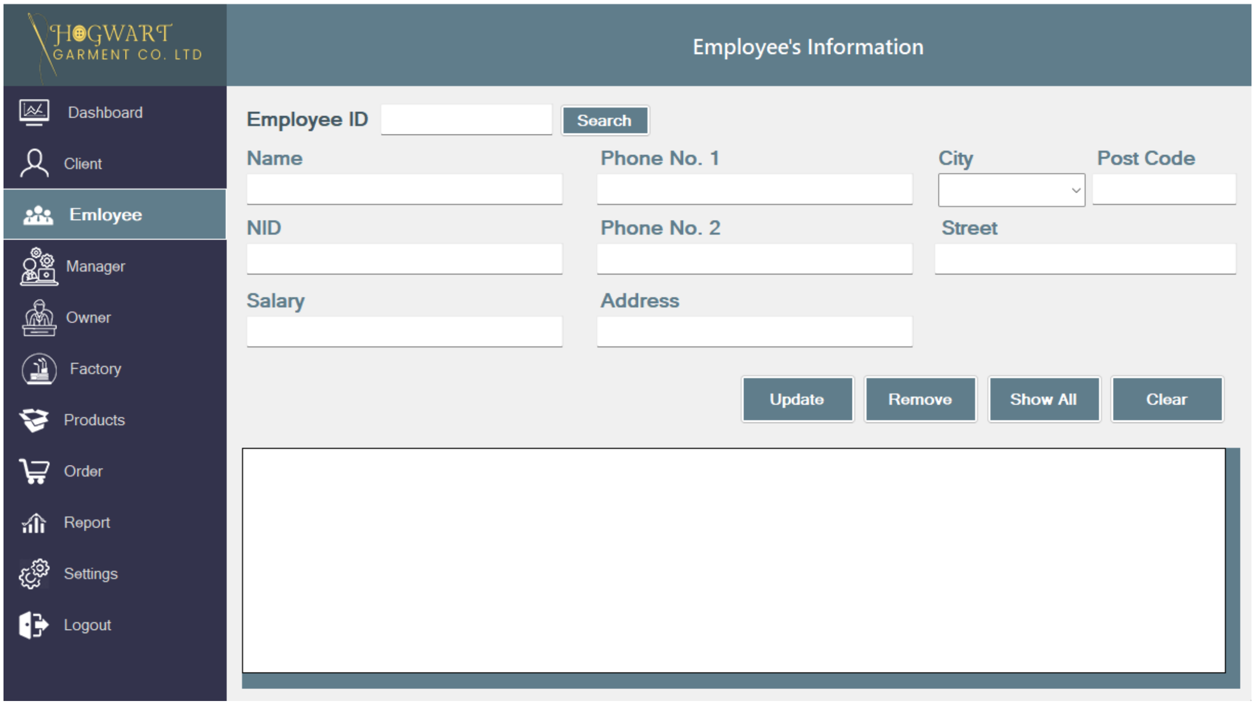
E-mail: morshed@gmail.com

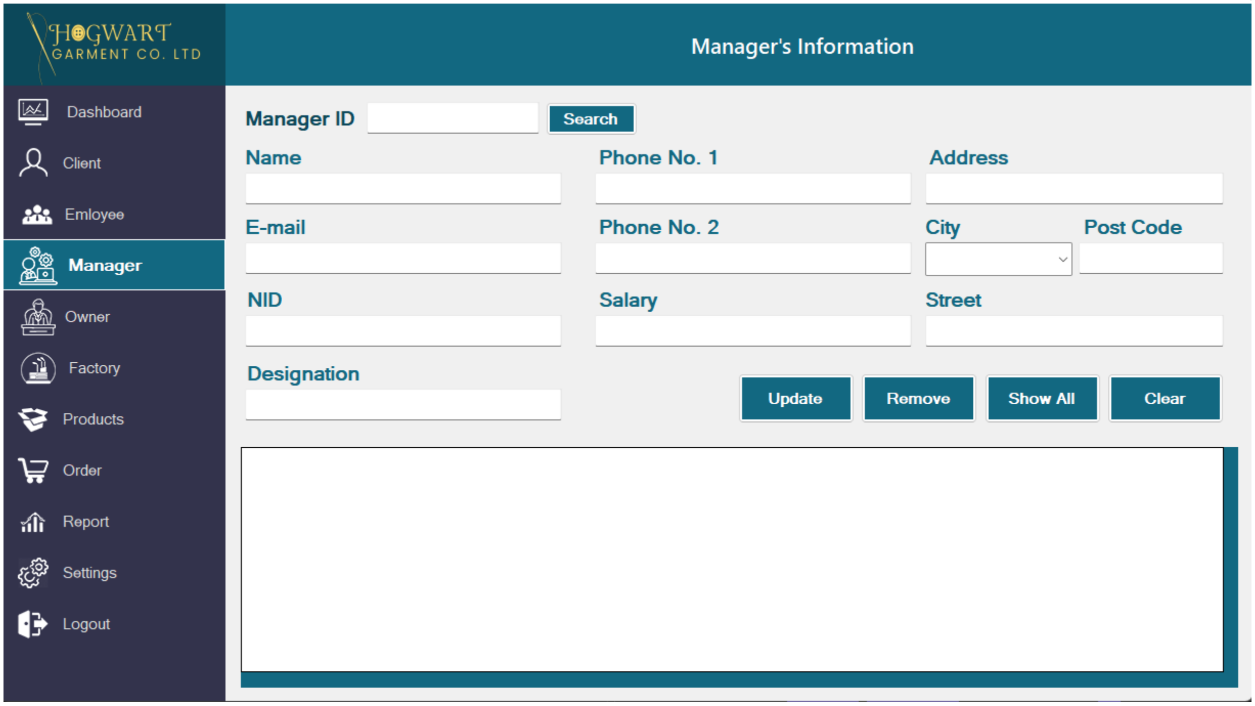
Company Name: BBC

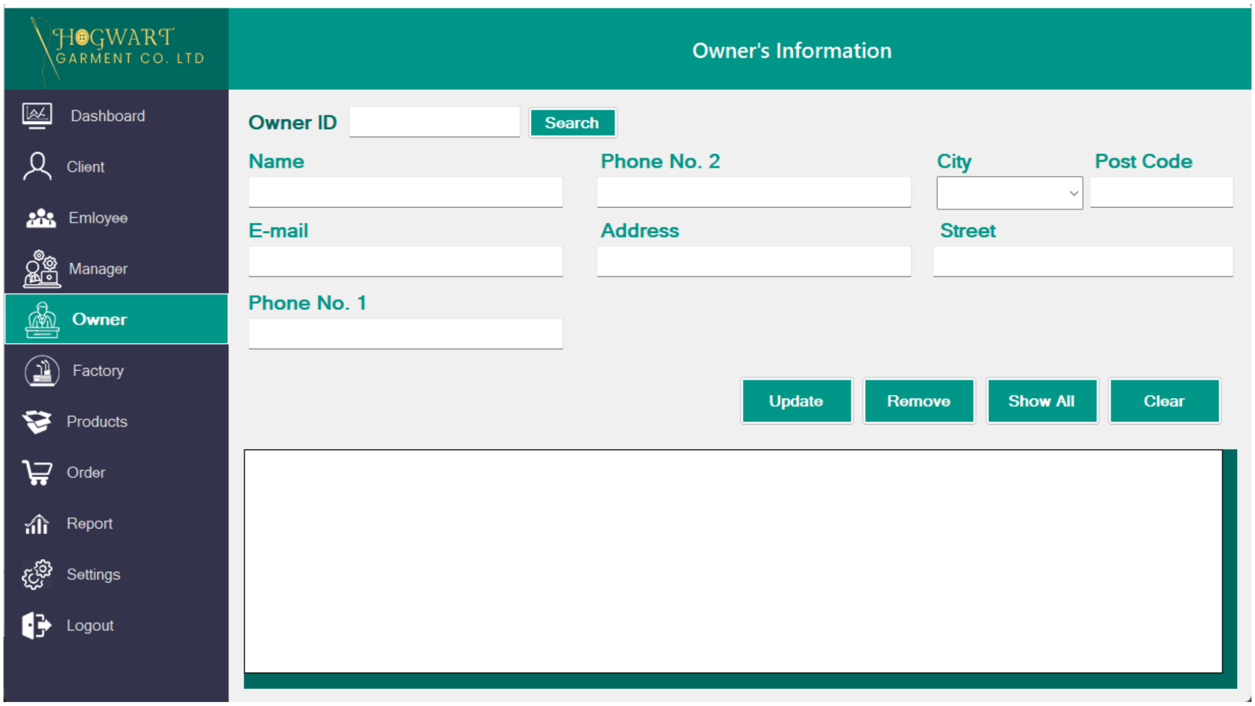
Address: 58/1 North jatrabari Dhaka 1204

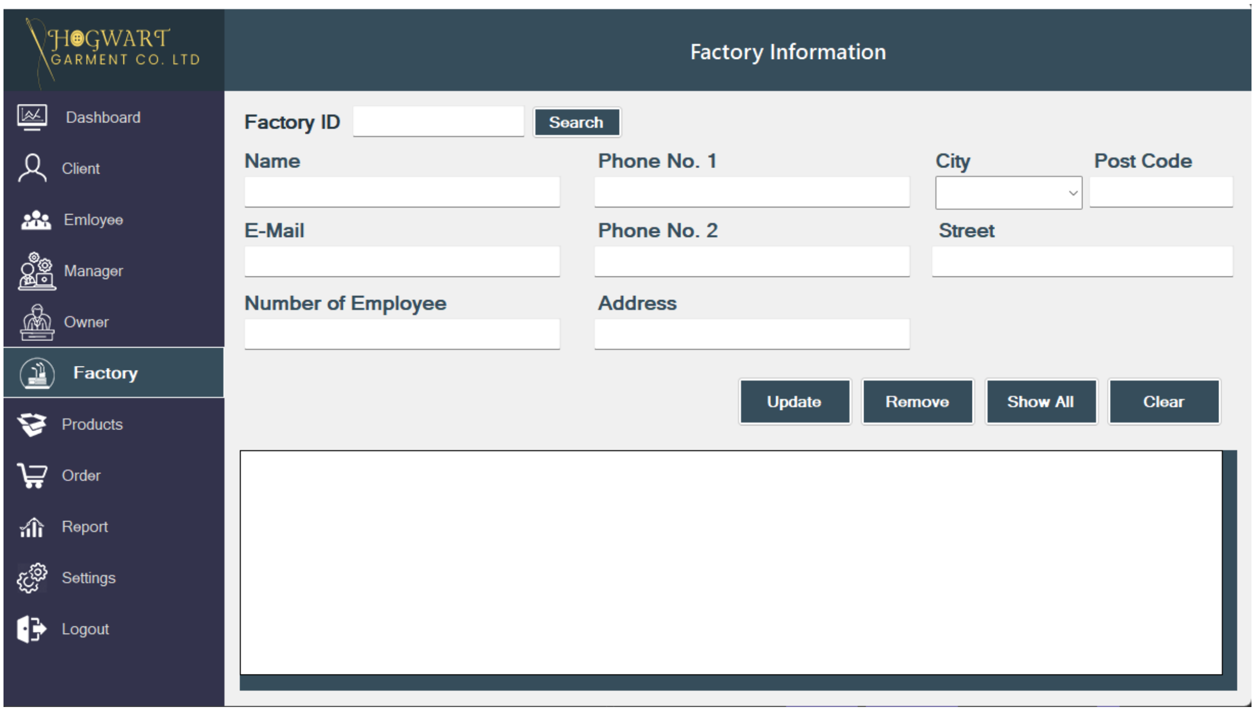
Trade License No: 152656666

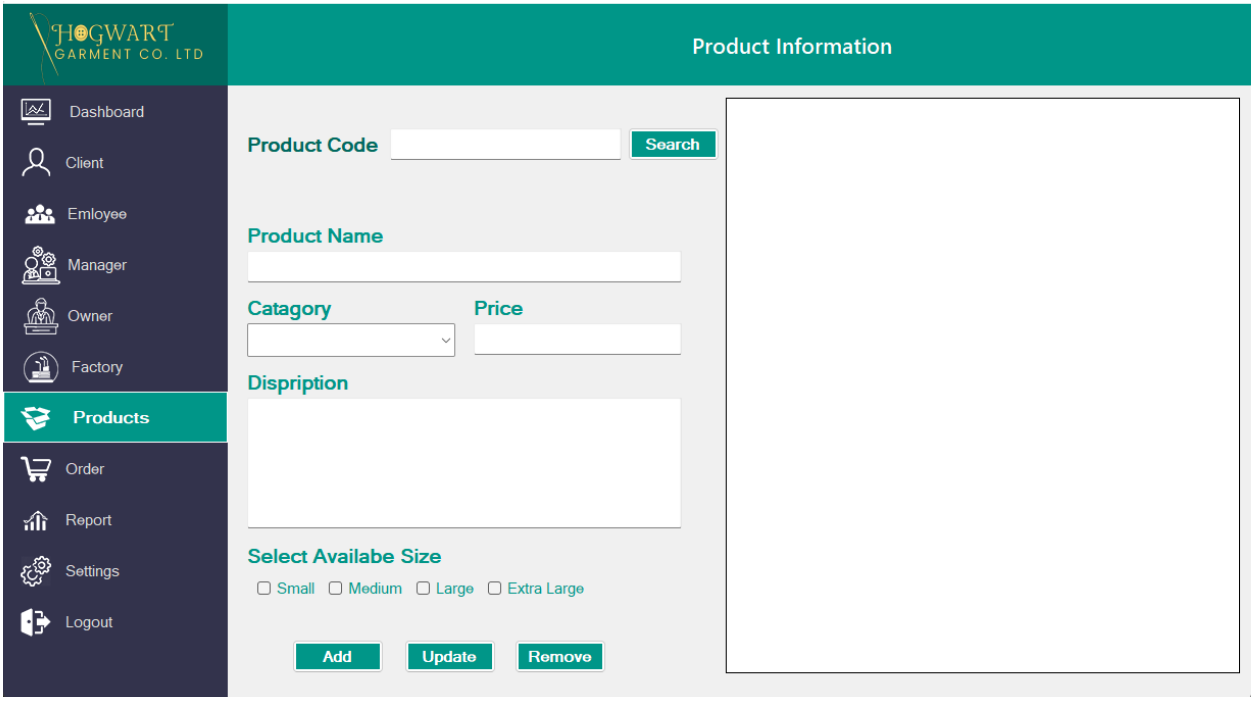
Client_ID	Client_Name	Client_Email	Client_CompanyName	Client_TradeLicenseNo	Client_PhoneNo	Client_Address
42645	Siam	morshed@gmail.com	BBC	152656666	01954456543, 01842456543	58/1 North jatrabari Dhaka 1204

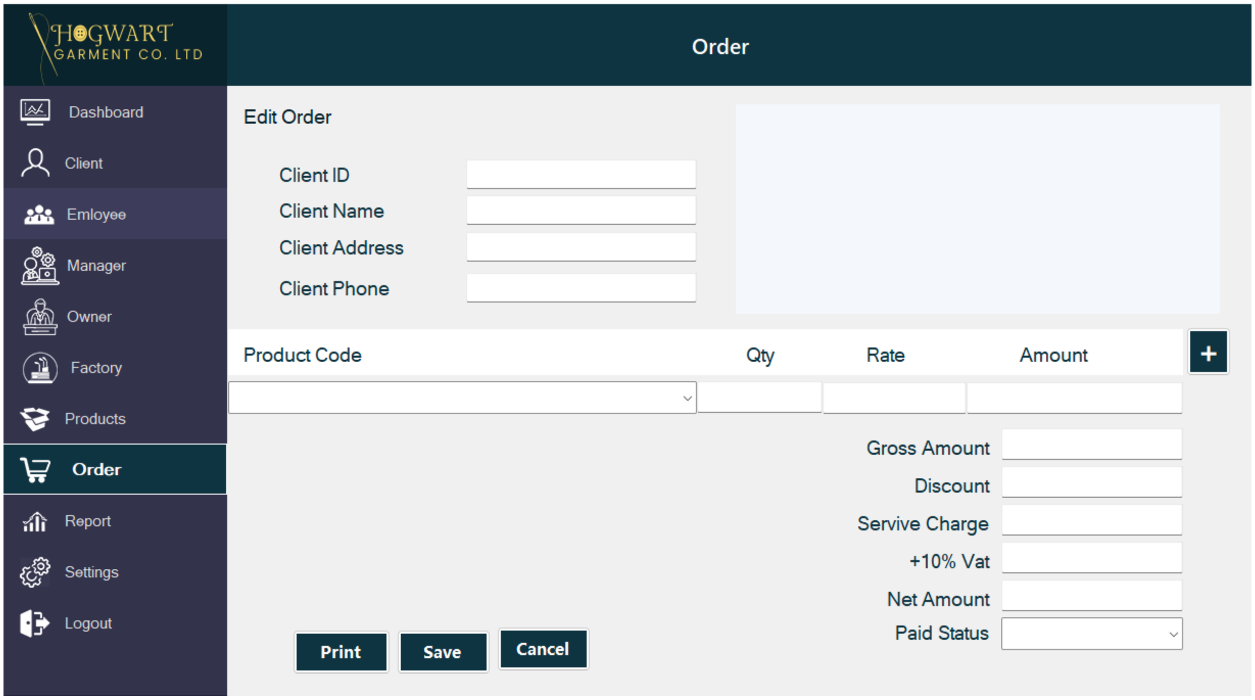

HOGWART
GARMENT CO. LTD
Employee's Information
Dashboard
Client
Emloyee
Manager
Owner
Factory
Products
Order
Report
Settings
Logout
Employee ID
Search
Name
Phone No. 1
City
Post Code
NID
Phone No. 2
Street
Salary
Address
Update
Remove
Show All
Clear

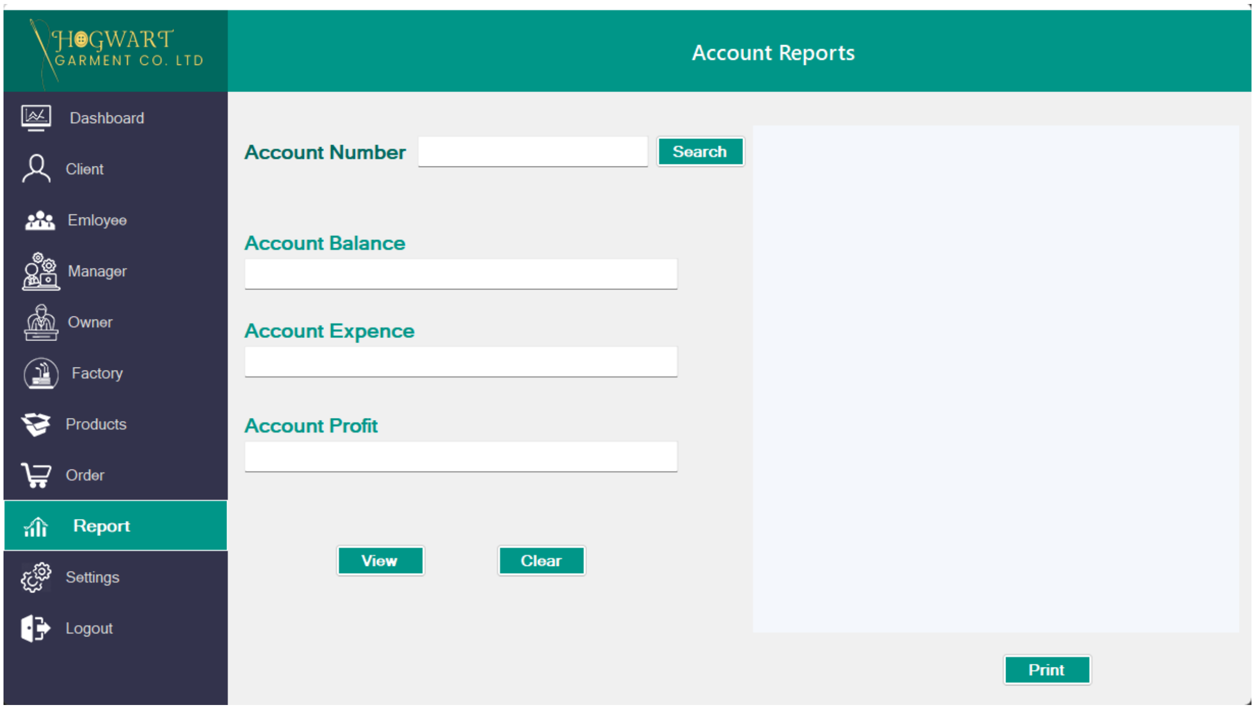

HOGWART
GARMENT CO. LTD
Manager's Information
Dashboard
Client
Emloyee
Manager
Owner
Factory
Products
Order
Report
Settings
Logout
Manager ID
Search
Name
Phone No. 1
Address
E-mail
Phone No. 2
City
Post Code
NID
Salary
Street
Designation
Update
Remove
Show All
Clear


HOGWART
GARMENT CO. LTD
Owner's Information
Dashboard
Client
Emloyee
Manager
Owner
Factory
Products
Order
Report
Settings
Logout
Owner ID
Search
Name
Phone No. 2
City
Post Code
E-mail
Address
Street
Phone No. 1
Update
Remove
Show All
Clear


HOGWART
GARMENT CO. LTD
Factory Information
Dashboard
Client
Emloyee
Manager
Owner
Factory
Products
Order
Report
Settings
Logout
Factory ID
Search
Name
Phone No. 1
City
Post Code
E-Mail
Phone No. 2
Street
Number of Employee
Address
Update
Remove
Show All
Clear


HOGWART
GARMENT CO. LTD
Product Information
Dashboard
Client
Emloyee
Manager
Owner
Factory
Products
Order
Report
Settings
Logout
Product Code
Search
Product Name
Catagory
Price
Disprition
Select Availabe Size
□ Small □ Medium □ Large □ Extra Large
Add
Update
Remove


HOGWART
GARMENT CO. LTD
Order
Dashboard
Client
Emloyee
Manager
Owner
Factory
Products
Order
Report
Settings
Logout
Edit Order
Client ID
Client Name
Client Address
Client Phone
Product Code
Qty
Rate
Amount
+
Gross Amount
Discount
Servive Charge
+10% Vat
Net Amount
Paid Status
Print
Save
Cancel



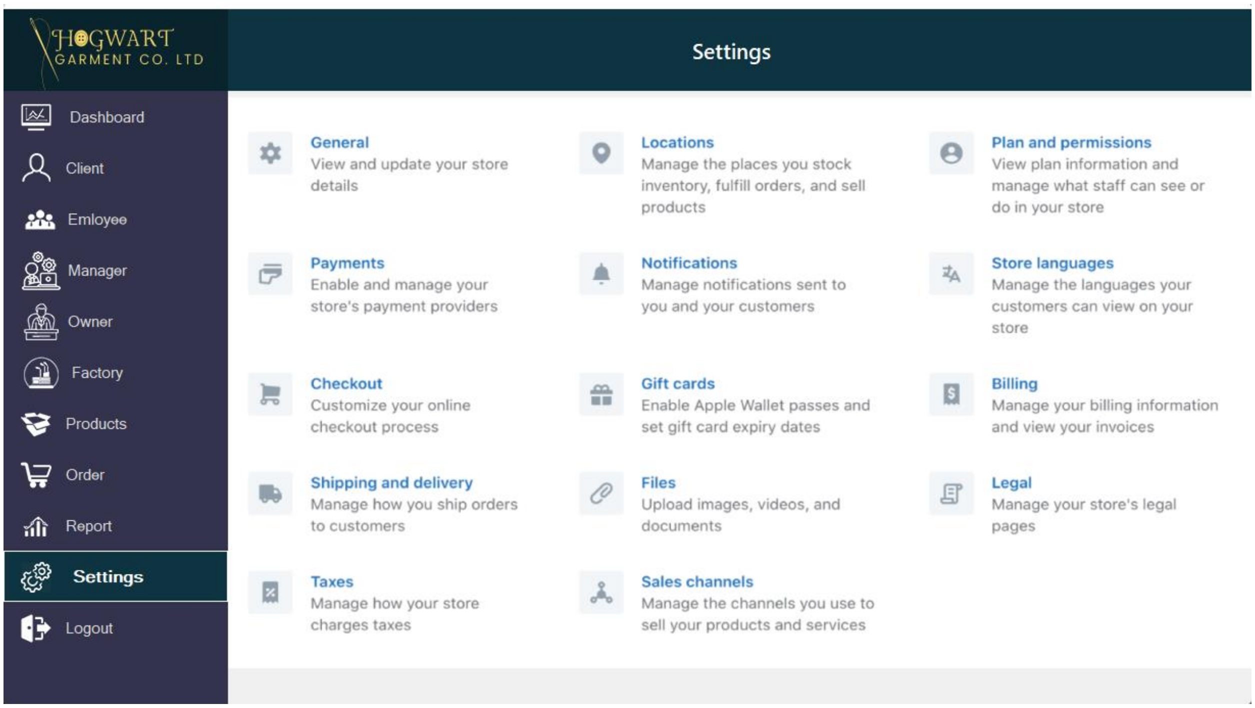
Account Reports

Account Number

Account Balance

Account Expence

Account Profit



Settings

General
View and update your store details

Locations
Manage the places you stock inventory, fulfill orders, and sell products

Plan and permissions
View plan information and manage what staff can see or do in your store

Payments
Enable and manage your store's payment providers

Notifications
Manage notifications sent to you and your customers

Store languages
Manage the languages your customers can view on your store

Checkout
Customize your online checkout process

Gift cards
Enable Apple Wallet passes and set gift card expiry dates

Billing
Manage your billing information and view your invoices

Shipping and delivery
Manage how you ship orders to customers

Files
Upload images, videos, and documents

Legal
Manage your store's legal pages

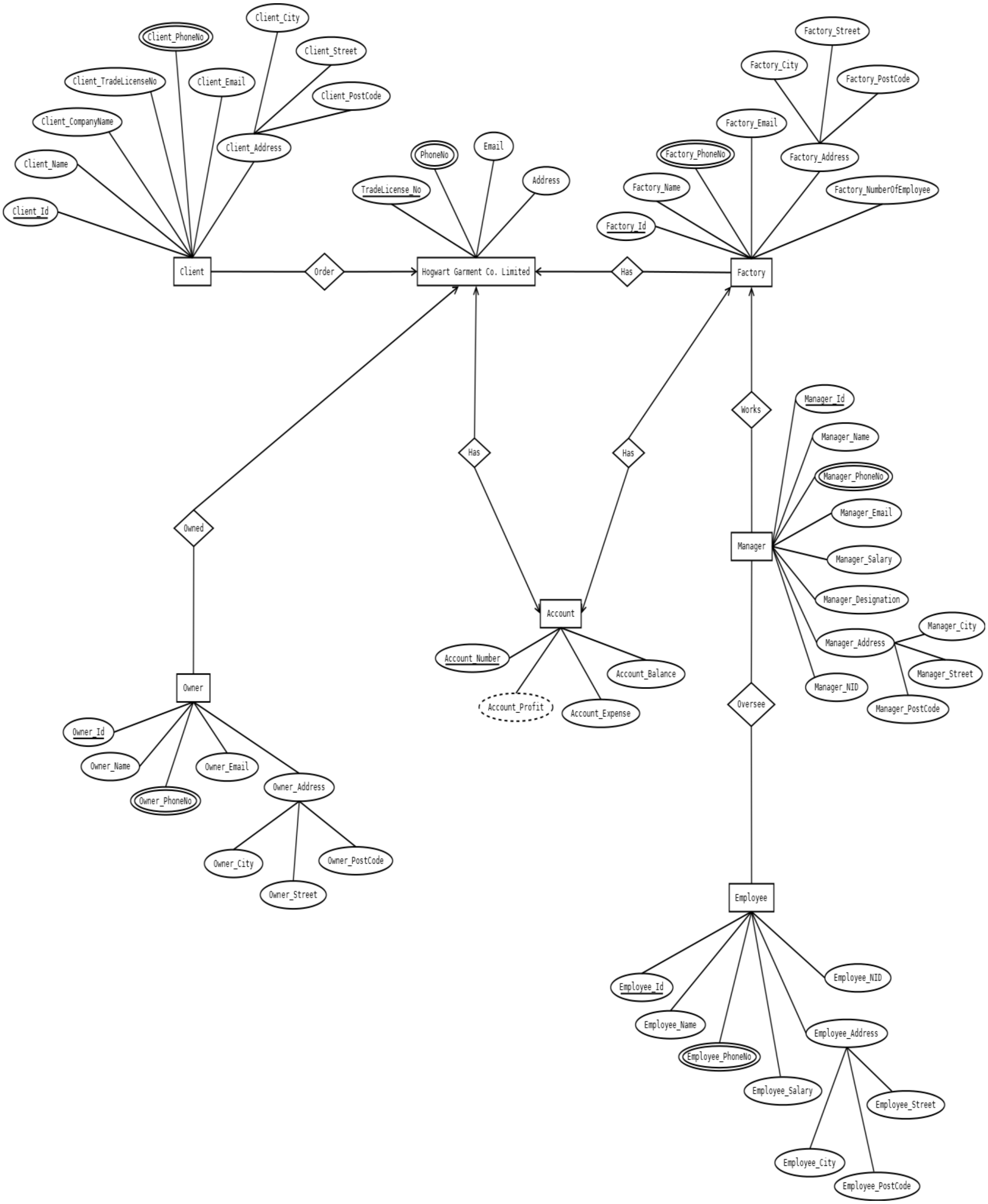
Taxes
Manage how your store charges taxes

Sales channels
Manage the channels you use to sell your products and services

Scenario description

Hogwart Garment Co. Limited takes orders from multiple clients and all the clients are overseen by Hogwart Garment Co. Limited. It is identified by a unique trade license no. The system also stores phone no, email, and address. Each client is identified by their unique id and it also stores the client’s name, company name, trade license number, phone number, email, and address. The client's address is composed of city, street, and postcode. Hogwart Garment Co. Limited and clients can have multiple phone no. Hogwart Garment Co. Limited has many factories and all the factories are managed by Hogwart Garment Co. Limited. Factory is identified by a unique id and can have more than one phone no. The system also stores the name, email, address, and number of employees. The factory address is composed of city, street, and postcode. Many managers work for a factory and all the managers work under that factory. Each manager is identified by their unique id and it also stores the manager's name, phone number, email, salary, designation, address, and NID. The managers can have multiple phone no. Managers oversee employees, while multiple managers can oversee specific one factory’s employees but every employee of that factory has to report to the particular managers. Each employee has an individual id to recognize them. it also stores the employee's name, phone number, salary, address, and NID. The employees can have multiple phone no. Each factory has its own single account and Hogwart Garment Co. Limited has a single account. In the system account number, profit, expenses, and balance are also stored. The profit is calculated from the garments and the company’s profit and expenses. The whole company is owned by multiple owners. Each of the owners is identified by their owner id. Other data such as name, phone no, email, and address are also stored in the system. The owner's address is composed of city, street, and postcode The owners can have multiple phone no.

ER Diagram


Client_City
Client_PhoneNo
Client_Street
Client_TradeLicenseNo
Client_Email
Client_PostCode
Client_CompanyName
Client_Address
Client_Name
Client_Id
Client
Order
PhoneNo
Email
TradeLicense_No
Address
Hogwart Garment Co. Limited
Factory_Street
Factory_City
Factory_PostCode
Factory_Email
Factory_PhoneNo
Factory_Address
Factory_Name
Factory_NumberOfEmployee
Factory_Id
Has
Factory
Works
Manager_Id
Manager_Name
Manager_PhoneNo
Manager_Email
Manager
Manager_Salary
Manager_Designation
Manager_City
Manager_Address
Manager_Street
Manager_NID
Manager_PostCode
Oversee
Has
Has
Owned
Account
Account_Number
Account_Balance
Account_Profit
Account_Expense
Owner
Owner_Id
Owner_Name
Owner_Email
Owner_Address
Owner_PhoneNo
Owner_City
Owner_PostCode
Owner_Street
Employee
Employee_Id
Employee_NID
Employee_Name
Employee_Address
Employee_PhoneNo
Employee_Salary
Employee_Street
Employee_City
Employee_PostCode

Normalization

Order

UNF

Order (Client_Id, Client_Name, Client_CompanyName, Client_TradeLicenseNo, Client_PhoneNo, Client_Email, Client_City, Client_Street, Client_PostCode, TradeLicense_No, PhoneNo, Email, Address)

1NF

Client_PhoneNo and PhoneNo are multivalued attribute.

1. Client_Id, Client_Name, Client_CompanyName, Client_TradeLicenseNo, Client_PhoneNo, Client_Email, Client_City, Client_Street, Client_PostCode, TradeLicense_No, PhoneNo, Email, Address

2NF

1. Client_Id, Client_Name, Client_CompanyName, Client_TradeLicenseNo, Client_PhoneNo, Client_Email, Client_City, Client_Street, Client_PostCode

2. TradeLicense_No, PhoneNo, Email, Address

3NF

1. Client_Id, Client_Name, Client_CompanyName, Client_TradeLicenseNo, Client_PhoneNo, Client_Email

2. Client_City, Client_Street, Client_PostCode

3. TradeLicense_No, PhoneNo, Email, Address

Table Creation

1. Client_Id, Client_Name, Client_CompanyName, Client_TradeLicenseNo, Client_PhoneNo, Client_Email, **C_Id**, **TradeLicense_No**

2. C_Id, Client_City, Client_Street, Client_PostCode

3. TradeLicense_No, PhoneNo, Email, Address

Has

UNF

Has (TradeLicense_No, PhoneNo, Email, Address, Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_City, Factory_Street, Factory_PostCode, Factory_NumberOfEmployee)

1NF

PhoneNo and Factory_PhoneNo are multivalued attribute.

1. TradeLicense_No, PhoneNo, Email, Address, Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_City, Factory_Street, Factory_PostCode, Factory_NumberOfEmployee

2NF

1. TradeLicense_No, PhoneNo, Email, Address

2. Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_City, Factory_Street, Factory_PostCode, Factory_NumberOfEmployee

3NF

1. TradeLicense_No, PhoneNo, Email, Address

2. Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_NumberOfEmployee

3. Factory_City, Factory_Street, Factory_PostCode

Table Creation

1. TradeLicense_No, PhoneNo, Email, Address

2. Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_NumberOfEmployee, **F_Id**, **TradeLicense_No**

3. F_Id, Factory_City, Factory_Street, Factory_PostCode

Owned

UNF

Owned (Owner_Id, Owner_Name, Owner_PhoneNo, Owner_Email, Owner_City, Owner_Street, Owner_PostCode, TradeLicense_No, PhoneNo, Email, Address)

1NF

Owner_PhoneNo and PhoneNo are multivalued attribute.

1. Owner_Id, Owner_Name, Owner_PhoneNo, Owner_Email, Owner_City, Owner_Street, Owner_PostCode, TradeLicense_No, PhoneNo, Email, Address

2NF

1. Owner_Id, Owner_Name, Owner_PhoneNo, Owner_Email, Owner_City, Owner_Street, Owner_PostCode

2. TradeLicense_No, PhoneNo, Email, Address

3NF

1. Owner_Id, Owner_Name, Owner_PhoneNo, Owner_Email

2. Owner_City, Owner_Street, Owner_PostCode

3. TradeLicense_No, PhoneNo, Email, Address

Table Creation

1. Owner_Id, Owner_Name, Owner_PhoneNo, Owner_Email, **O_Id**, **TradeLicense_No**

2. O_Id, Owner_City, Owner_Street, Owner_PostCode

3. TradeLicense_No, PhoneNo, Email, Address

Has

UNF

Has (TradeLicense_No, PhoneNo, Email, Address, Account_Number, Account_Profit, Account_Expense, Account_Balance)

1NF

PhoneNo is multivalued attribute

1. TradeLicense_No, PhoneNo, Email, Address, Account_Number, Account_Profit, Account_Expense, Account_Balance

2NF

1. TradeLicense_No, PhoneNo, Email, Address

2. Account_Number, Account_Profit, Account_Expense, Account_Balance

3NF

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

1. TradeLicense_No, PhoneNo, Email, Address

2. Account_Number, Account_Profit, Account_Expense, Account_Balance

Table Creation

1. TradeLicense_No, PhoneNo, Email, Address, **Account_Number**

2. Account_Number, Account_Profit, Account_Expense, Account_Balance

Has

UNF

Has (Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_City, Factory_Street, Factory_PostCode, Factory_NumberOfEmployee, Account_Number, Account_Profit, Account_Expense, Account_Balance)

1NF

Factory_PhoneNo is multivalued attribute

1. Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_City, Factory_Street, Factory_PostCode, Factory_NumberOfEmployee, Account_Number, Account_Profit, Account_Expense, Account_Balance

2NF

1. Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_City, Factory_Street, Factory_PostCode, Factory_NumberOfEmployee

2. Account_Number, Account_Profit, Account_Expense, Account_Balance

3NF

1. Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_NumberOfEmployee

2. Factory_City, Factory_Street, Factory_PostCode

3. Account_Number, Account_Profit, Account_Expense, Account_Balance

Table Creation

1. Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_NumberOfEmployee, **F_Id**, **Account_Number**

2. F_Id, Factory_City, Factory_Street, Factory_PostCode

3. Account_Number, Account_Profit, Account_Expense, Account_Balance

Works

UNF

Works (Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_City, Factory_Street, Factory_PostCode, Factory_NumberOfEmployee, Manager_Id, Manager_Name, Manager_PhoneNo, Manager_Email, Manager_Salary, Manager_Designation, Manager_City, Manager_Street, Manager_PostCode, Manager_NID)

1NF

Factory_PhoneNo and Manager_PhoneNo multivalued attribute

1. Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_City, Factory_Street, Factory_PostCode, Factory_NumberOfEmployee, Manager_Id, Manager_Name, Manager_PhoneNo, Manager_Email, Manager_Salary, Manager_Designation, Manager_City, Manager_Street, Manager_PostCode, Manager_NID

2NF

1. Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_City, Factory_Street, Factory_PostCode, Factory_NumberOfEmployee

2. Manager_Id, Manager_Name, Manager_PhoneNo, Manager_Email, Manager_Salary, Manager_Designation, Manager_City, Manager_Street, Manager_PostCode, Manager_NID

3NF

1. Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_NumberOfEmployee

2. Factory_City, Factory_Street, Factory_PostCode

3. Manager_Id, Manager_Name, Manager_PhoneNo, Manager_Email, Manager_Salary, Manager_Designation, Manager_NID

4. Manager_City, Manager_Street, Manager_PostCode

Table Creation

1. Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_NumberOfEmployee, **F_Id**

2. F_Id, Factory_City, Factory_Street, Factory_PostCode

3. Manager_Id, Manager_Name, Manager_PhoneNo, Manager_Email, Manager_Salary, Manager_Designation, Manager_NID, **M_Id**, **Factory_Id**

4. M_Id, Manager_City, Manager_Street, Manager_PostCode

Oversee

UNF

Oversee (Manager_Id, Manager_Name, Manager_PhoneNo, Manager_Email, Manager_Salary, Manager_Designation, Manager_City, Manager_Street, Manager_PostCode, Manager_NID, Employee_Id, Employee_Name, Employee_PhoneNo, Employee_Salary, Employee_City, Employee_PostCode, Employee_Street, Employee_NID)

1NF

Manager_PhoneNo and Employee_PhoneNo are multivalued attribute

1. Manager_Id, Manager_Name, Manager_PhoneNo, Manager_Email, Manager_Salary, Manager_Designation, Manager_City, Manager_Street, Manager_PostCode, Manager_NID, Employee_Id, Employee_Name, Employee_PhoneNo, Employee_Salary, Employee_City, Employee_PostCode, Employee_Street, Employee_NID

2NF

1. Manager_Id, Manager_Name, Manager_PhoneNo, Manager_Email, Manager_Salary, Manager_Designation, Manager_City, Manager_Street, Manager_PostCode, Manager_NID

2. Employee_Id, Employee_Name, Employee_PhoneNo, Employee_Salary, Employee_City, Employee_PostCode, Employee_Street, Employee_NID

3NF

1. Manager_Id, Manager_Name, Manager_PhoneNo, Manager_Email, Manager_Salary, Manager_Designation, Manager_NID

2. Manager_City, Manager_Street, Manager_PostCode

3. Employee_Id, Employee_Name, Employee_PhoneNo, Employee_Salary, Employee_NID

4. Employee_City, Employee_PostCode, Employee_Street

Table Creation

1. Manager_Id, Manager_Name, Manager_PhoneNo, Manager_Email, Manager_Salary, Manager_Designation, Manager_NID, **M_Id**

2. M_Id, Manager_City, Manager_Street, Manager_PostCode

3. Employee_Id, Employee_Name, Employee_PhoneNo, Employee_Salary, Employee_NID, **E_Id**

4. E_Id, Employee_City, Employee_PostCode, Employee_Street

5. **Manager_Id**, **Employee_Id**

Temporary Table

1. Client_Id, Client_Name, Client_CompanyName, Client_TradeLicenseNo, Client_PhoneNo, Client_Email, **C_Id**, **TradeLicense_No**

2. C_Id, Client_City, Client_Street, Client_PostCode

~~3. TradeLicense_No, PhoneNo, Email, Address~~

~~4. TradeLicense_No, PhoneNo, Email, Address~~

5. Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_NumberOfEmployee, **F_Id**, **TradeLicense_No**

6. F_Id, Factory_City, Factory_Street, Factory_PostCode

7. Owner_Id, Owner_Name, Owner_PhoneNo, Owner_Email, **O_Id**, **TradeLicense_No**

8. O_Id, Owner_City, Owner_Street, Owner_PostCode

~~9. TradeLicense_No, PhoneNo, Email, Address~~

10. TradeLicense_No, PhoneNo, Email, Address, **Account_Number**

11. Account_Number, Account_Profit, Account_Expense, Account_Balance

12. Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_NumberOfEmployee, **F_Id**, **Account_Number**

~~13. F_Id, Factory_City, Factory_Street, Factory_PostCode~~

~~14. Account_Number, Account_Profit, Account_Expense, Account_Balance~~

~~15. Factory_Id, Factory_Name, Factory_PhoneNo, Factory_Email, Factory_NumberOfEmployee, **F_Id**~~

~~16. F_Id, Factory_City, Factory_Street, Factory_PostCode~~

17. Manager_Id, Manager_Name, Manager_PhoneNo, Manager_Email, Manager_Salary, Manager_Designation, Manager_NID, **M_Id**, **Factory_Id**

18. M_Id, Manager_City, Manager_Street, Manager_PostCode

~~19. Manager_Id, Manager_Name, Manager_PhoneNo, Manager_Email, Manager_Salary, Manager_Designation, Manager_NID, **M_Id**~~

~~20. M_Id, Manager_City, Manager_Street, Manager_PostCode~~

21. Employee_Id, Employee_Name, Employee_PhoneNo, Employee_Salary, Employee_NID, **E_Id**

22. E_Id, Employee_City, Employee_PostCode, Employee_Street

23. **Manager_Id**, **Employee_Id**

Final Table

1. Client_Id, Client_Name, Client_CompanyName, Client_TradeLicenseNo, Client_PhoneNo1, Client_PhoneNo2, Client_Email, **C_Id**, **TradeLicense_No**

2. C_Id, Client_City, Client_Street, Client_PostCode

3. Factory_Id, Factory_Name, Factory_PhoneNo1, Factory_PhoneNo2, Factory_Email, Factory_NumberOfEmployee, **F_Id**, **TradeLicense_No**

4. F_Id, Factory_City, Factory_Street, Factory_PostCode

5. Owner_Id, Owner_Name, Owner_PhoneNo1, Owner_PhoneNo2, Owner_Email, **O_Id, TradeLicense_No**

6. O_Id, Owner_City, Owner_Street, Owner_PostCode

7. TradeLicense_No, PhoneNo1, PhoneNo2, Email, Address, **Account_Number**

8. Account_Number, Account_Profit, Account_Expense, Account_Balance

9. Factory_Id, Factory_Name, Factory_PhoneNo1, Factory_PhoneNo2, Factory_Email, Factory_NumberOfEmployee, **F_Id**, **Account_Number**

10. Manager_Id, Manager_Name, Manager_PhoneNo1, Manager_PhoneNo2, Manager_Email, Manager_Salary, Manager_Designation, Manager_NID, **M_Id**, **Factory_Id**

11. M_Id, Manager_City, Manager_Street, Manager_PostCode

12. Employee_Id, Employee_Name, Employee_PhoneNo1, Employee_PhoneNo2, Employee_Salary, Employee_NID, **E_Id**

13. E_Id, Employee_City, Employee_PostCode, Employee_Street

14. **Manager_Id**, **Employee_Id**

Schema Diagram

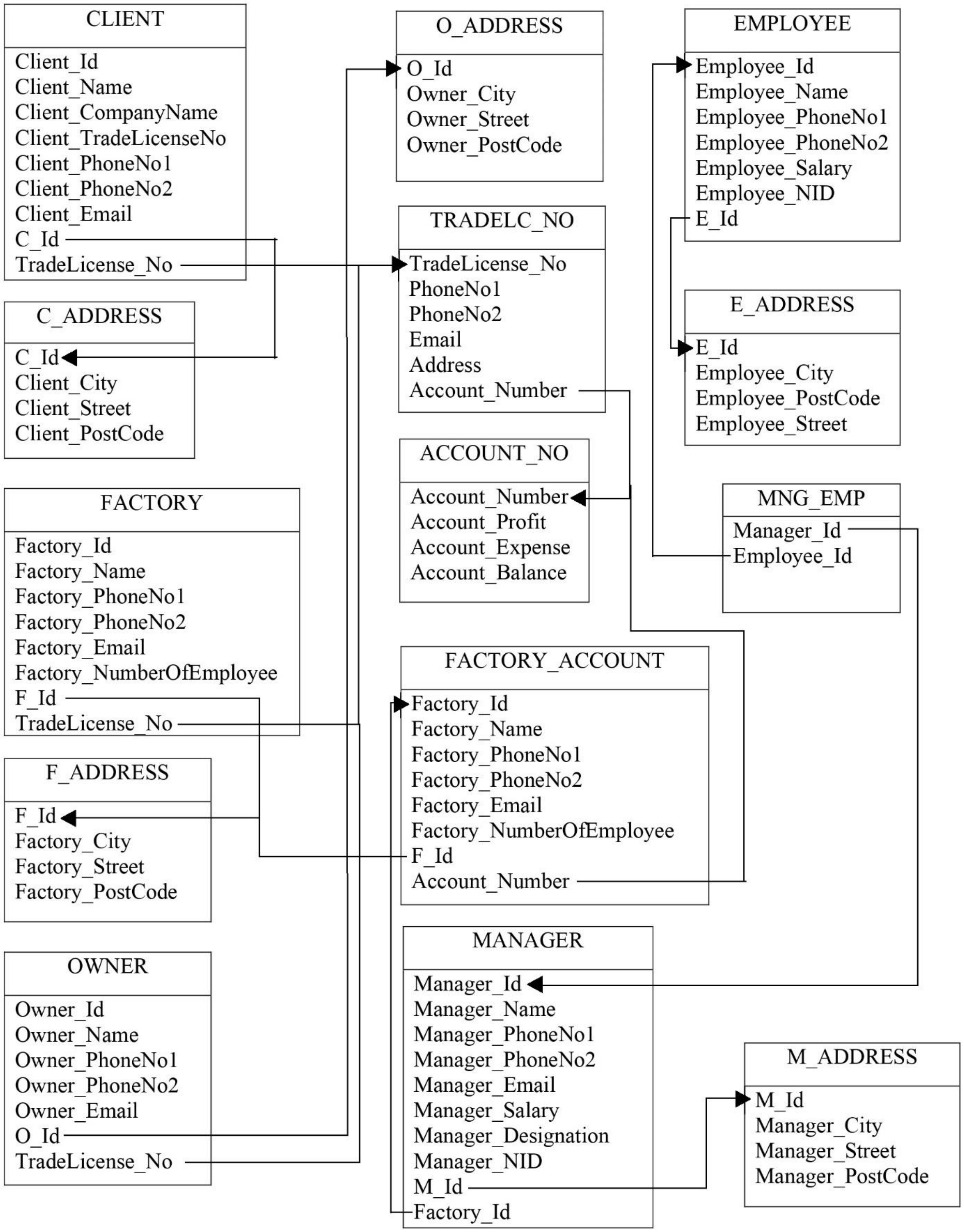


Table Creation

Create Table Account_No (Account_Number Number, Account_Profit Number, Account_Expense Number, Account_Balance Number, Primary Key (Account_Number));

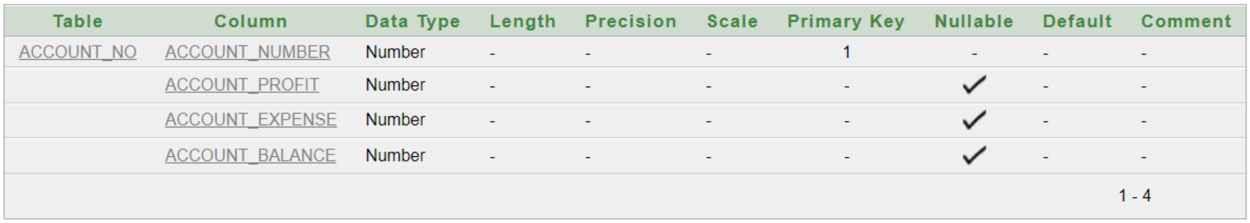


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ACCOUNT_NO	ACCOUNT_NUMBER	Number	-	-	-	1	-	-	-
	ACCOUNT_PROFIT	Number	-	-	-	-	✓	-	-
	ACCOUNT_EXPENSE	Number	-	-	-	-	✓	-	-
	ACCOUNT_BALANCE	Number	-	-	-	-	✓	-	-

1 - 4

create table tradelc_no (tradelicense_no number, phoneno1 number, phoneno2 number, email varchar2(255), address varchar2(255), account_number number, primary key (tradelicense_no), foreign key (account_number) references account_no);

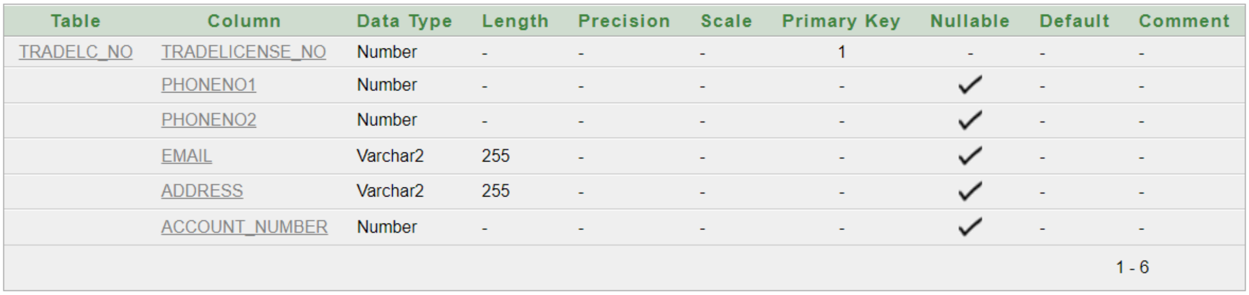


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TRADELC_NO	TRADELICENSE_NO	Number	-	-	-	1	-	-	-
	PHONENO1	Number	-	-	-	-	✓	-	-
	PHONENO2	Number	-	-	-	-	✓	-	-
	EMAIL	Varchar2	255	-	-	-	✓	-	-
	ADDRESS	Varchar2	255	-	-	-	✓	-	-
	ACCOUNT_NUMBER	Number	-	-	-	-	✓	-	-

1 - 6

Create Table C_Address (C_Id Number, Client_City Varchar2 (255), Client_Street Varchar2 (255), Client_Postcode Varchar2 (255), Primary Key (C_Id));

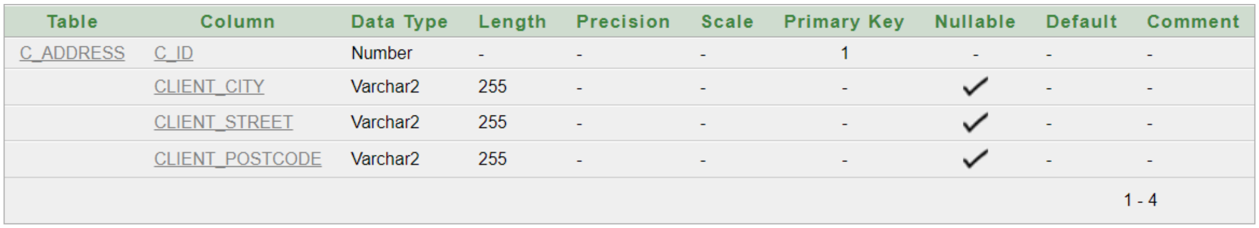


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
C_ADDRESS	C_ID	Number	-	-	-	1	-	-	-
	CLIENT_CITY	Varchar2	255	-	-	-	✓	-	-
	CLIENT_STREET	Varchar2	255	-	-	-	✓	-	-
	CLIENT_POSTCODE	Varchar2	255	-	-	-	✓	-	-

1 - 4

Create Table Client (Client_Id Number, Client_Name Varchar2 (255), Client_Companyname Varchar2 (255), Client_Tradelicenseno Number, Client_Phoneno1 Number, Client_Phoneno2 Number, Client_Email Varchar2 (255), C_Id Number, Tradelicense_No Number,Primary Key (Client_Id),Foreign Key (C_Id) References C_Address,Foreign Key (Tradelicense_No) References Tradelc_No);

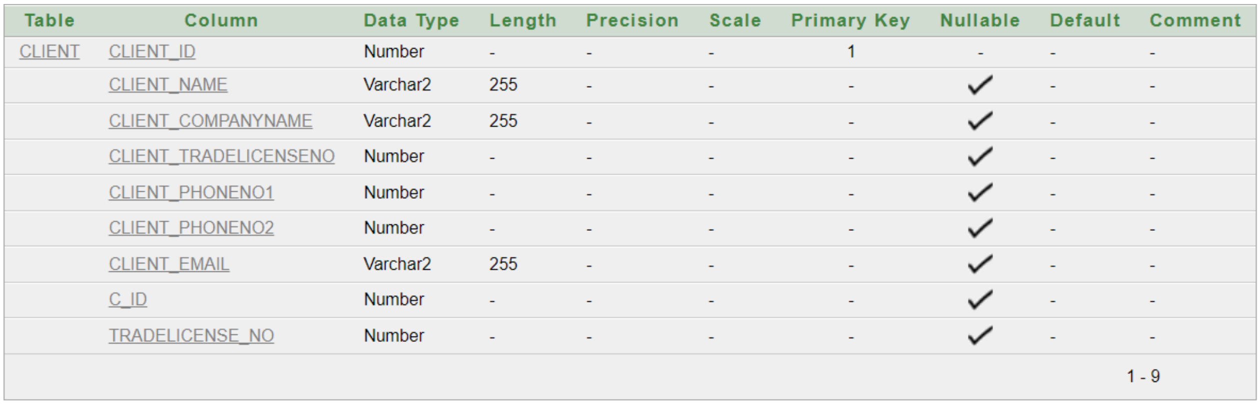


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CLIENT	CLIENT_ID	Number	-	-	-	1	-	-	-
	CLIENT_NAME	Varchar2	255	-	-	-	✓	-	-
	CLIENT_COMPANYNAME	Varchar2	255	-	-	-	✓	-	-
	CLIENT_TRADELICENSENO	Number	-	-	-	-	✓	-	-
	CLIENT_PHONENO1	Number	-	-	-	-	✓	-	-
	CLIENT_PHONENO2	Number	-	-	-	-	✓	-	-
	CLIENT_EMAIL	Varchar2	255	-	-	-	✓	-	-
	C_ID	Number	-	-	-	-	✓	-	-
	TRADELICENSE_NO	Number	-	-	-	-	✓	-	-

1 - 9

Create Table F_Address (F_Id Number, Factory_City Varchar2(255), Factory_Street Varchar2(255), Factory_Postcode Varchar2(255), Primary Key (F_Id));

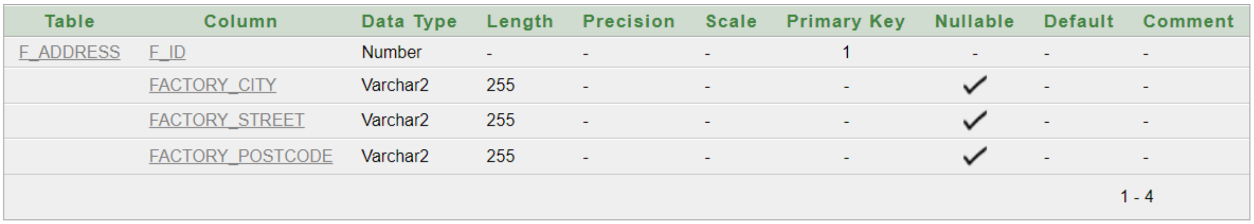


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
F_ADDRESS	F_ID	Number	-	-	-	1	-	-	-
	FACTORY_CITY	Varchar2	255	-	-	-	✓	-	-
	FACTORY_STREET	Varchar2	255	-	-	-	✓	-	-
	FACTORY_POSTCODE	Varchar2	255	-	-	-	✓	-	-

1 - 4

Create Table Factory (Factory_Id Number, Factory_Name Varchar2 (255), Client_Companyname Varchar2 (255), Factory_Phoneno1 Number, Factory_Phoneno2 Number, Factory_Email Varchar2 (255), Factory_Numberofemployee Number, F_Id Number, Tradelicense_No Number, Primary Key (Factory_Id), Foreign Key (F_Id) References F_Address, Foreign Key (Tradelicense_No) References Tradelc_No);

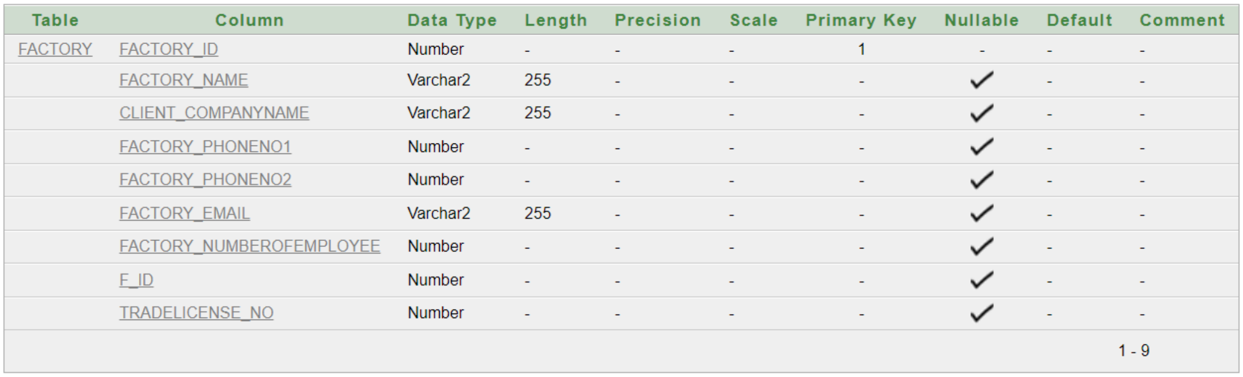


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FACTORY	FACTORY_ID	Number	-	-	-	1	-	-	-
	FACTORY_NAME	Varchar2	255	-	-	-	✓	-	-
	CLIENT_COMPANYNAME	Varchar2	255	-	-	-	✓	-	-
	FACTORY_PHONENO1	Number	-	-	-	-	✓	-	-
	FACTORY_PHONENO2	Number	-	-	-	-	✓	-	-
	FACTORY_EMAIL	Varchar2	255	-	-	-	✓	-	-
	FACTORY_NUMBEROFEMPLOYEE	Number	-	-	-	-	✓	-	-
	F_ID	Number	-	-	-	-	✓	-	-
	TRADELICENSE_NO	Number	-	-	-	-	✓	-	-
									1 - 9

Create Table Factory_Account (Factory_Id Number, Factory_Name Varchar2 (255), Client_Companyname Varchar2 (255), Factory_Phoneno1 Number, Factory_Phoneno2 Number, Factory_Email Varchar2 (255), Factory_Numberofemployee Number, F_Id Number, Account_Number Number, Primary Key (Factory_Id), Foreign Key (F_Id) References F_Address, Foreign Key (Account_Number) References Account_No);

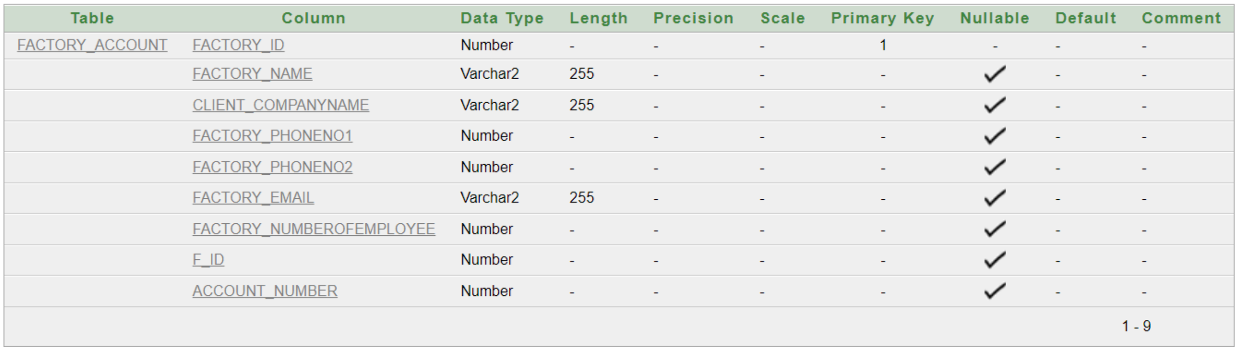


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FACTORY_ACCOUNT	FACTORY_ID	Number	-	-	-	1	-	-	-
	FACTORY_NAME	Varchar2	255	-	-	-	✓	-	-
	CLIENT_COMPANYNAME	Varchar2	255	-	-	-	✓	-	-
	FACTORY_PHONENO1	Number	-	-	-	-	✓	-	-
	FACTORY_PHONENO2	Number	-	-	-	-	✓	-	-
	FACTORY_EMAIL	Varchar2	255	-	-	-	✓	-	-
	FACTORY_NUMBEROFEMPLOYEE	Number	-	-	-	-	✓	-	-
	F_ID	Number	-	-	-	-	✓	-	-
	ACCOUNT_NUMBER	Number	-	-	-	-	✓	-	-
									1 - 9

Create Table M_Address (M_Id Number, Manager_City Varchar2(255), Manager_Street Varchar2(255), Manager_Postcode Varchar2(255), Primary Key (M_Id));

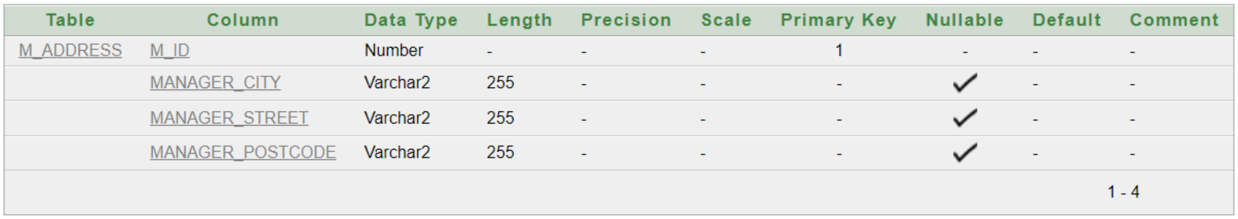


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
M_ADDRESS	M_ID	Number	-	-	-	1	-	-	-
	MANAGER_CITY	Varchar2	255	-	-	-	✓	-	-
	MANAGER_STREET	Varchar2	255	-	-	-	✓	-	-
	MANAGER_POSTCODE	Varchar2	255	-	-	-	✓	-	-
									1 - 4

Create Table Manager (Manager_Id Number, Manager_Name Varchar2 (255), Manager_Phoneno1 Number, Manager_Phoneno2 Number, Manager_Email Varchar2 (255), Manager_Salary Number, Manager_Designation Varchar2 (255), Manager_Nid Number, M_Id Number, Factory_Id Number, Primary Key (Manager_Id), Foreign Key (M_Id) References M_Address, Foreign Key (Factory_Id) References Factory);

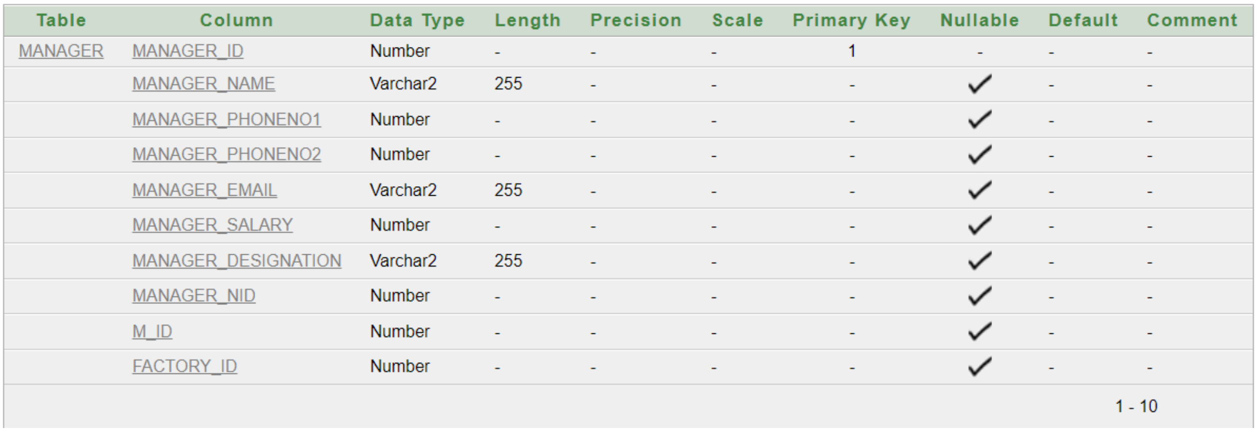


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANAGER	MANAGER_ID	Number	-	-	-	1	-	-	-
	MANAGER_NAME	Varchar2	255	-	-	-	✓	-	-
	MANAGER_PHONENO1	Number	-	-	-	-	✓	-	-
	MANAGER_PHONENO2	Number	-	-	-	-	✓	-	-
	MANAGER_EMAIL	Varchar2	255	-	-	-	✓	-	-
	MANAGER_SALARY	Number	-	-	-	-	✓	-	-
	MANAGER_DESIGNATION	Varchar2	255	-	-	-	✓	-	-
	MANAGER_NID	Number	-	-	-	-	✓	-	-
	M_ID	Number	-	-	-	-	✓	-	-
	FACTORY_ID	Number	-	-	-	-	✓	-	-
									1 - 10

Create Table E_Address (E_Id Number, Employee_City Varchar2 (255), Employee_Street Varchar2 (255), Employee_Postcode Varchar2(255), Primary Key (E_Id));

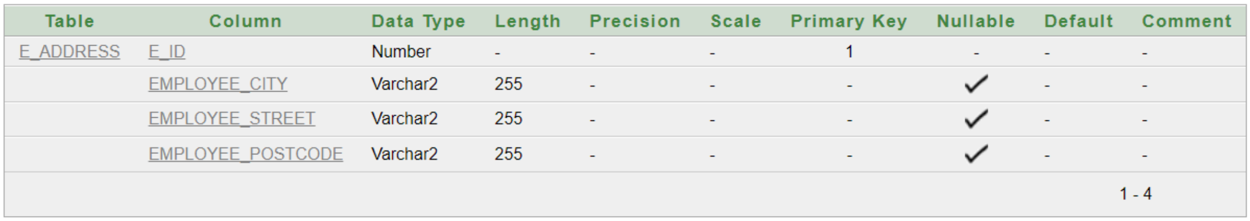


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
E_ADDRESS	E_ID	Number	-	-	-	1	-	-	-
	EMPLOYEE_CITY	Varchar2	255	-	-	-	✓	-	-
	EMPLOYEE_STREET	Varchar2	255	-	-	-	✓	-	-
	EMPLOYEE_POSTCODE	Varchar2	255	-	-	-	✓	-	-
									1 - 4

Create Table Employee (Employee_Id Number, Employee_Name Varchar2 (255), Employee_Phoneno1 Number, Employee_Phoneno2 Number, Employee_Salary Number, Employee_Nid Number, E_Id Number, Primary Key (Employee_Id), Foreign Key (E_Id) References E_Address);

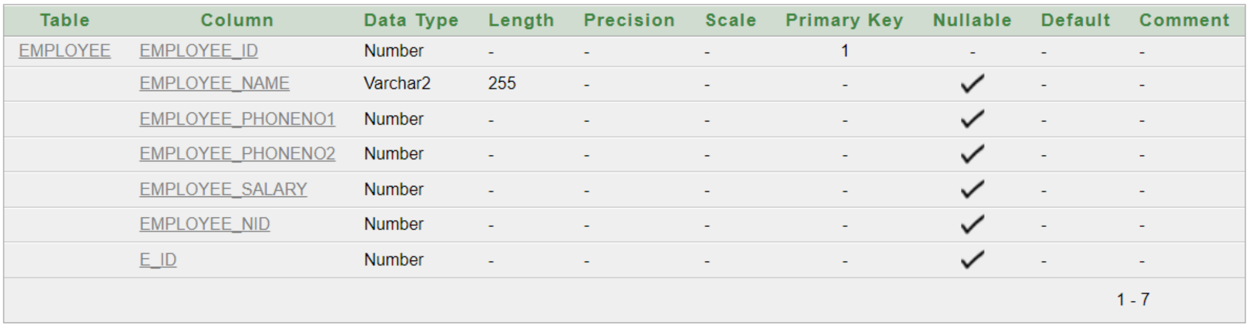


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE	EMPLOYEE_ID	Number	-	-	-	1	-	-	-
	EMPLOYEE_NAME	Varchar2	255	-	-	-	✓	-	-
	EMPLOYEE_PHONENO1	Number	-	-	-	-	✓	-	-
	EMPLOYEE_PHONENO2	Number	-	-	-	-	✓	-	-
	EMPLOYEE_SALARY	Number	-	-	-	-	✓	-	-
	EMPLOYEE_NID	Number	-	-	-	-	✓	-	-
	E_ID	Number	-	-	-	-	✓	-	-
								1 - 7	

Create Table O_Address (O_Id Number, Owner_City Varchar2 (255), Owner_Street Varchar2 (255), Owner_Postcode Varchar2 (255), Primary Key (O_Id));

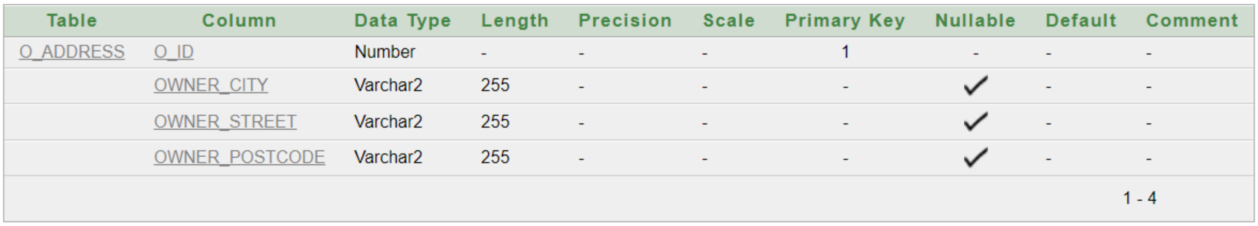


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
O_ADDRESS	O_ID	Number	-	-	-	1	-	-	-
	OWNER_CITY	Varchar2	255	-	-	-	✓	-	-
	OWNER_STREET	Varchar2	255	-	-	-	✓	-	-
	OWNER_POSTCODE	Varchar2	255	-	-	-	✓	-	-
								1 - 4	

Create Table Owner (Owner_Id Number, Owner_Name Varchar2 (255), Owner_Phoneno1 Number, Owner_Phoneno2 Number, Owner_Email Varchar2 (255), O_Id Number, Tradelicense_No Number, Primary Key (Owner_Id), Foreign Key (O_Id) References O_Address, Foreign Key (Tradelicense_No) References Tradelc_No);

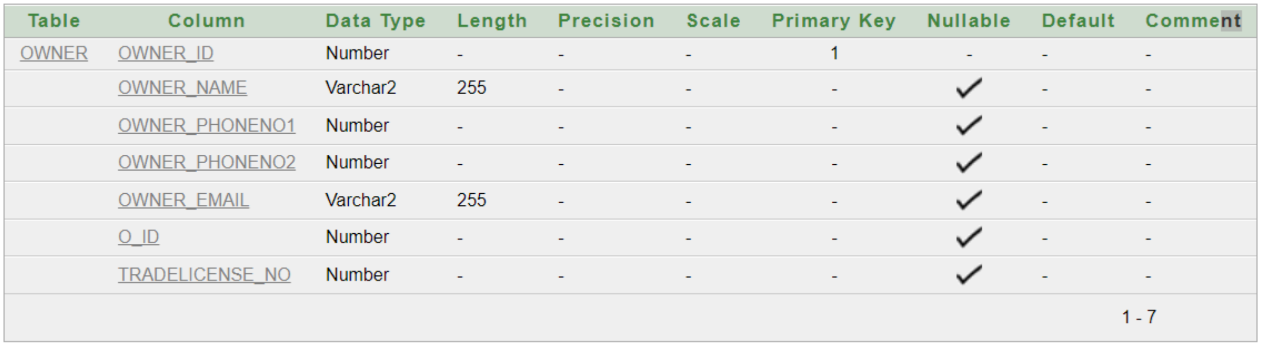


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
OWNER	OWNER_ID	Number	-	-	-	1	-	-	-
	OWNER_NAME	Varchar2	255	-	-	-	✓	-	-
	OWNER_PHONENO1	Number	-	-	-	-	✓	-	-
	OWNER_PHONENO2	Number	-	-	-	-	✓	-	-
	OWNER_EMAIL	Varchar2	255	-	-	-	✓	-	-
	O_ID	Number	-	-	-	-	✓	-	-
	TRADELICENSE_NO	Number	-	-	-	-	✓	-	-
								1 - 7	

```
Create Table Mng_Emp (Manager_Id Number, Employee_Id Number, Foreign Key
(Manager_Id) References Manager, Foreign Key (Employee_Id) References
Employee);
```

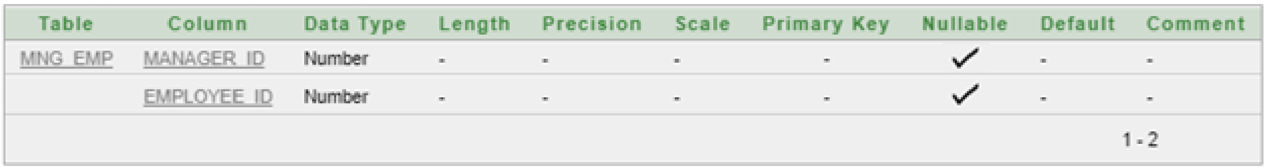


Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MNG_EMP	MANAGER_ID	Number	-	-	-	-	✓	-	-
	EMPLOYEE_ID	Number	-	-	-	-	✓	-	-
									1 - 2

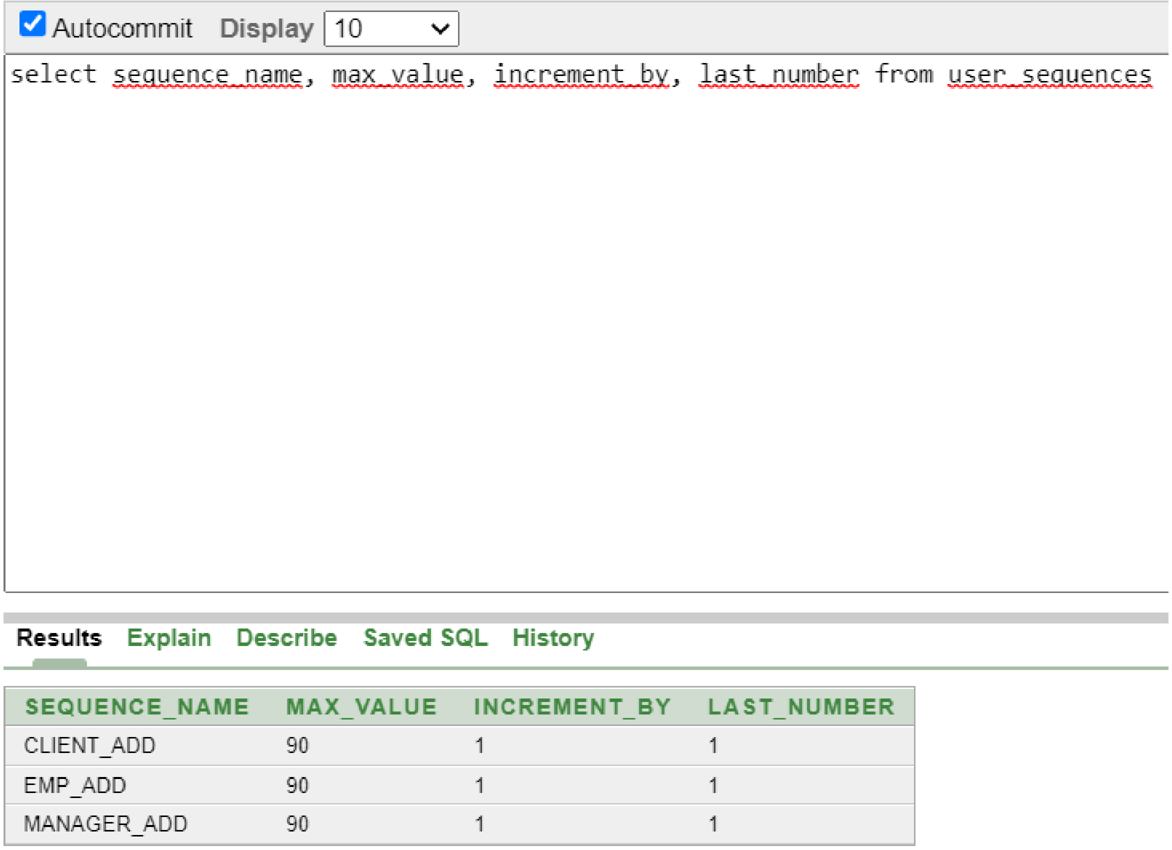
Sequence

```
Create Sequence Client_Add
        Increment By 1
        Start With 1
        Maxvalue 90
        Nocache
        Nocycle;

Create Sequence Emp_Add
        Increment By 1
        Start With 1
        Maxvalue 90
        Nocache
        Nocycle;



Create Sequence Manager_Add
        Increment By 1
        Start With 1
```

Maxvalue 90

Nocache

Nocycle;



```
select sequence_name, max_value, increment_by, last_number from user_sequences
```

SEQUENCE_NAME	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
CLIENT_ADD	90	1	1
EMP_ADD	90	1	1
MANAGER_ADD	90	1	1

Index

Create Index Manager_Address_Idx On M_Address (M_Id,Manager_City);

Create Index Employee_Address_Idx On E_Address (E_Id,Employee_City);

Create Index Owner_Address_Idx On O_Address (O_Id,Owner_City);

Create Index Factory_Address_Idx On F_Address (F_Id);

Create Index Manager_Idx On Manager (Manager_Id,Manager_Name);

Create Index Employee_Idx On Employee (Employee_Id,Employee_Name);

Create Index Owner_Idx On Owner (Owner_Id,Owner_Name);

Create Index Factory_Idx On Factory (Factory_Id,Factory_Name);

Create Index Account_Idx On Account_No (Account_Number,Account_Profit);

Create Index Trade_Licsense_Idx on Tradelc_No (Tradelicense_No, Account_Number);

Create Index Client_Address_Idx On C_Address (C_Id,Client_City);

Create Index Client_Idx On Client (Client_Id,Client_Name);

Create Index Factory_Acc_Idx On Factory_Account (Factory_Id,Factory_Name);

Create Index Mng_Emp_Idx on Mng_Emp (Manager_Id);

Create users, assign roles and grant privileges

login as System

Create user Project identified by pass;

Grant unlimited tablespace to Project;

Grant create all privileges,session,create role,create table,create sequence,create view,create procedure,connect, resource to Project with admin option;

login as Project

Create Role Admin;

Grant create session,create role,create table,create sequence,create view,create procedure,connect, resource to Admin with admin option;

Create user Emp identified by password;

Grant unlimited tablespace to Emp;

Grant Admin to Emp;

Data Insertion

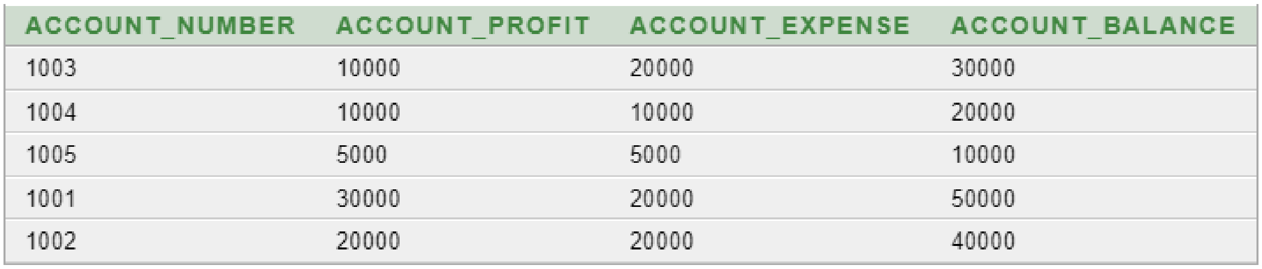
1.insert into account_no values (1001, 30000,20000,50000)

2.insert into account_no values (1002, 20000,20000, 40000)

3.insert into account_no values (1003, 10000,20000, 30000)

4.insert into account_no values (1004, 10000,10000, 20000)

5.insert into account_no values (1005, 5000,5000, 10000)



ACCOUNT_NUMBER	ACCOUNT_PROFIT	ACCOUNT_EXPENSE	ACCOUNT_BALANCE
1003	10000	20000	30000
1004	10000	10000	20000
1005	5000	5000	10000
1001	30000	20000	50000
1002	20000	20000	40000

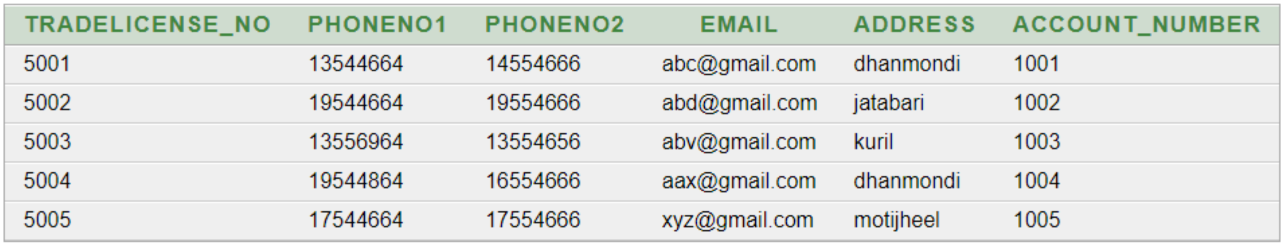
1.insert into tradelc_no values (5001,013544664,014554666, 'abc@gmail.com','dhanmondi',1001)

2.insert into tradelc_no values (5002,019544664,019554666, 'abd@gmail.com','jatabari',1002)

3.insert into tradelc_no values (5003,013556964,013554656, 'abv@gmail.com','kuril',1003)

4.insert into tradelc_no values (5004,019544864,016554666, 'aax@gmail.com','dhanmondi',1004)

5.insert into tradelc_no values (5005,017544664,017554666, 'xyz@gmail.com','motijheel',1005)



TRADELICENSE_NO	PHONENO1	PHONENO2	EMAIL	ADDRESS	ACCOUNT_NUMBER
5001	13544664	14554666	abc@gmail.com	dhanmondi	1001
5002	19544664	19554666	abd@gmail.com	jatabari	1002
5003	13556964	13554656	abv@gmail.com	kuril	1003
5004	19544864	16554666	aax@gmail.com	dhanmondi	1004
5005	17544664	17554666	xyz@gmail.com	motijheel	1005

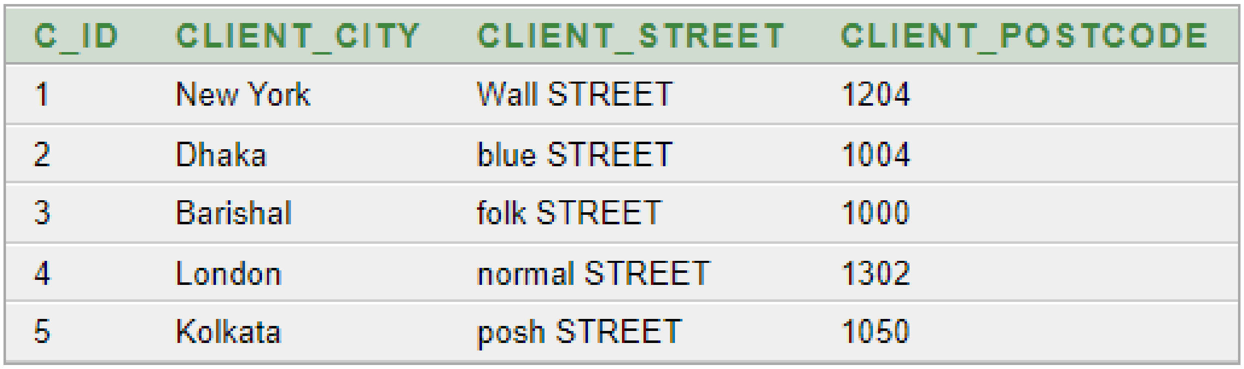
1.insert into c_address values (Client_Add.nextval,'New York','Wall STREET','1204')

2.insert into c_address values (Client_Add.nextval,'Dhaka','blue STREET','1004')

3.insert into c_address values (Client_Add.nextval,'Barishal','folk STREET','1000')

4.insert into c_address values (Client_Add.nextval,'London','normal STREET','1302')

5.insert into c_address values (Client_Add.nextval,'Kolkata','posh STREET','1050')



C_ID	CLIENT_CITY	CLIENT_STREET	CLIENT_POSTCODE
1	New York	Wall STREET	1204
2	Dhaka	blue STREET	1004
3	Barishal	folk STREET	1000
4	London	normal STREET	1302
5	Kolkata	posh STREET	1050

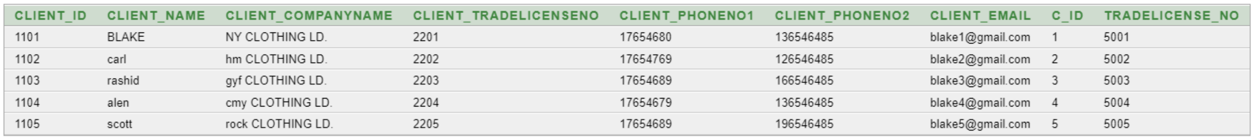
1.insert into client values (1101,'BLAKE','NY CLOTHING LD.',2201,017654680,0136546485, 'blake1@gmail.com',1,5001)

2.insert into client values (1102,'carl','hm CLOTHING LD.',2202,017654769,0126546485, 'blake2@gmail.com',2,5002)

3.insert into client values (1103,'rashid','gyf CLOTHING LD.',2203,017654689,0166546485, 'blake3@gmail.com',3,5003)

4.insert into client values (1104,'alen','cmy CLOTHING LD.',2204,017654679,0136546485, 'blake4@gmail.com',4,5004)

5.insert into client values (1105,'scott','rock CLOTHING LD.',2205,017654689,0196546485, 'blake5@gmail.com',5,5005)



CLIENT_ID	CLIENT_NAME	CLIENT_COMPANYNAME	CLIENT_TRADELICENSENO	CLIENT_PHONENO1	CLIENT_PHONENO2	CLIENT_EMAIL	C_ID	TRADELICENSE_NO
1101	BLAKE	NY CLOTHING LD.	2201	17654680	136546485	blake1@gmail.com	1	5001
1102	carl	hm CLOTHING LD.	2202	17654769	126546485	blake2@gmail.com	2	5002
1103	rashid	gyf CLOTHING LD.	2203	17654689	166546485	blake3@gmail.com	3	5003
1104	alen	cmy CLOTHING LD.	2204	17654679	136546485	blake4@gmail.com	4	5004
1105	scott	rock CLOTHING LD.	2205	17654689	196546485	blake5@gmail.com	5	5005

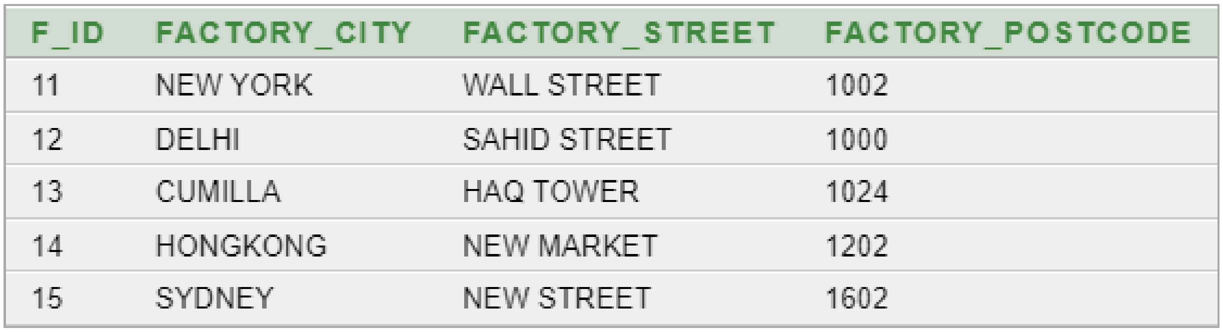
1.insert into f_address values (11,'NEW YORK','WALL STREET','1002')

2.insert into f_address values (12,'DELHI','SAHID STREET','1000')

3.insert into f_address values (13,'CUMILLA','HAQ TOWER','1024')

4.insert into f_address values (14,'HONGKONG','NEW MARKET','1202')

5.insert into f_address values (15,'SYDNEY','NEW STREET','1602')



F_ID	FACTORY_CITY	FACTORY_STREET	FACTORY_POSTCODE
11	NEW YORK	WALL STREET	1002
12	DELHI	SAHID STREET	1000
13	CUMILLA	HAQ TOWER	1024
14	HONGKONG	NEW MARKET	1202
15	SYDNEY	NEW STREET	1602

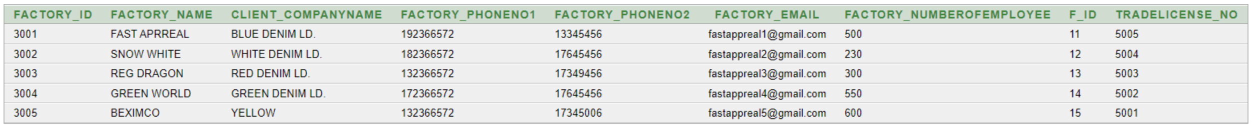
1.insert into factory values (3001,'FAST APRREAL','BLUE DENIM LD.',0192366572,013345456, 'fastappreal1@gmail.com',500,11,5005)

2.insert into factory values (3002,'SNOW WHITE','WHITE DENIM LD.',0182366572,017645456, 'fastappreal2@gmail.com',230,12,5004)

3.insert into factory values (3003,'REG DRAGON','RED DENIM LD.',0132366572,017349456, 'fastappreal3@gmail.com',300,13,5003)

4.insert into factory values (3004,'GREEN WORLD','GREEN DENIM LD.',0172366572,017645456, 'fastappreal4@gmail.com',550,14,5002)

5.insert into factory values (3005,'BEXIMCO','YELLOW',0132366572,017345006, 'fastappreal5@gmail.com',600,15,5001)



FACTORY_ID	FACTORY_NAME	CLIENT_COMPANYNAME	FACTORY_PHONENO1	FACTORY_PHONENO2	FACTORY_EMAIL	FACTORY_NUMBEROFEMPLOYEE	F_ID	TRADELICENSE_NO
3001	FAST APRREAL	BLUE DENIM LD.	192366572	13345456	fastappreal1@gmail.com	500	11	5005
3002	SNOW WHITE	WHITE DENIM LD.	182366572	17645456	fastappreal2@gmail.com	230	12	5004
3003	REG DRAGON	RED DENIM LD.	132366572	17349456	fastappreal3@gmail.com	300	13	5003
3004	GREEN WORLD	GREEN DENIM LD.	172366572	17645456	fastappreal4@gmail.com	550	14	5002
3005	BEXIMCO	YELLOW	132366572	17345006	fastappreal5@gmail.com	600	15	5001

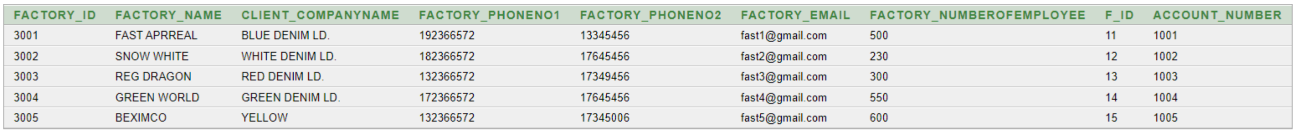
1.insert into factory_account values (3001,'FAST APRREAL','BLUE DENIM LD.',0192366572,013345456, 'fast1@gmail.com',500,11,1001)

2.insert into factory_account values (3002,'SNOW WHITE','WHITE DENIM LD.',0182366572,017645456, 'fast2@gmail.com',230,12,1002)

3.insert into factory_account values (3003,'REG DRAGON','RED DENIM LD.',0132366572,017349456, 'fast3@gmail.com',300,13,1003)

4.insert into factory_account values (3004,'GREEN WORLD','GREEN DENIM LD.',0172366572,017645456, 'fast4@gmail.com',550,14,1004)

5.insert into factory_account values (3005,'BEXIMCO','YELLOW',0132366572,017345006, 'fast5@gmail.com',600,15,1005)



FACTORY_ID	FACTORY_NAME	CLIENT_COMPANYNAME	FACTORY_PHONENO1	FACTORY_PHONENO2	FACTORY_EMAIL	FACTORY_NUMBEROFEMPLOYEE	F_ID	ACCOUNT_NUMBER
3001	FAST APRREAL	BLUE DENIM LD.	192366572	13345456	fast1@gmail.com	500	11	1001
3002	SNOW WHITE	WHITE DENIM LD.	182366572	17645456	fast2@gmail.com	230	12	1002
3003	REG DRAGON	RED DENIM LD.	132366572	17349456	fast3@gmail.com	300	13	1003
3004	GREEN WORLD	GREEN DENIM LD.	172366572	17645456	fast4@gmail.com	550	14	1004
3005	BEXIMCO	YELLOW	132366572	17345006	fast5@gmail.com	600	15	1005

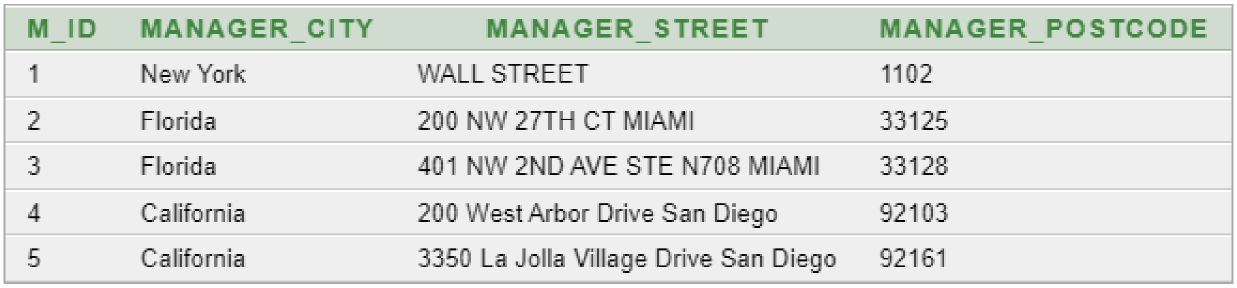
1.insert into m_address values (Manager_Add.nextval,'New York','WALL STREET','1102')

2.insert into m_address values (Manager_Add.nextval,'Florida','200 NW 27TH CT MIAMI','33125')

3.insert into m_address values (Manager_Add.nextval,'Florida','401 NW 2ND AVE STE N708 MIAMI','33128')

4.insert into m_address values (Manager_Add.nextval,'California','200 West Arbor Drive San Diego','92103')

5.insert into m_address values (Manager_Add.nextval,'California','3350 La Jolla Village Drive San Diego','92161')



M_ID	MANAGER_CITY	MANAGER_STREET	MANAGER_POSTCODE
1	New York	WALL STREET	1102
2	Florida	200 NW 27TH CT MIAMI	33125
3	Florida	401 NW 2ND AVE STE N708 MIAMI	33128
4	California	200 West Arbor Drive San Diego	92103
5	California	3350 La Jolla Village Drive San Diego	92161

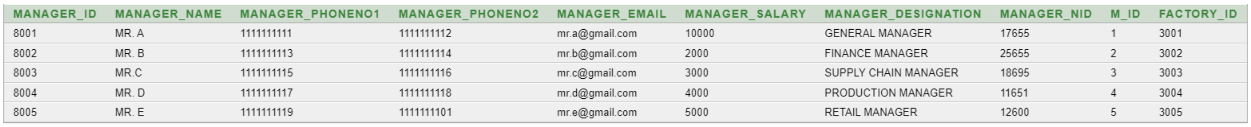
1.insert into manager values (8001,'MR. A',1111111111, 1111111112, 'mr.a@gmail.com',10000,'GENERAL MANAGER',17655,1,3001)

2.insert into manager values (8002,'MR. B', 1111111113, 1111111114, 'mr.b@gmail.com',2000,'FINANCE MANAGER',25655,2,3002)

3.insert into manager values (8003,'MR.C', 1111111115, 1111111116, 'mr.c@gmail.com',3000,'SUPPLY CHAIN MANAGER',18695,3,3003)

4.insert into manager values (8004,'MR. D', 1111111117, 1111111118, 'mr.d@gmail.com',4000,'PRODUCTION MANAGER',11651,4,3004)

5.insert into manager values (8005,'MR. E', 1111111119, 1111111101, 'mr.e@gmail.com',5000,'RETAIL MANAGER',12600,5,3005)



MANAGER_ID	MANAGER_NAME	MANAGER_PHONENO1	MANAGER_PHONENO2	MANAGER_EMAIL	MANAGER_SALARY	MANAGER_DESIGNATION	MANAGER_NID	M_ID	FACTORY_ID
8001	MR. A	1111111111	1111111112	mr.a@gmail.com	10000	GENERAL MANAGER	17655	1	3001
8002	MR. B	1111111113	1111111114	mr.b@gmail.com	2000	FINANCE MANAGER	25655	2	3002
8003	MR.C	1111111115	1111111116	mr.c@gmail.com	3000	SUPPLY CHAIN MANAGER	18695	3	3003
8004	MR. D	1111111117	1111111118	mr.d@gmail.com	4000	PRODUCTION MANAGER	11651	4	3004
8005	MR. E	1111111119	1111111101	mr.e@gmail.com	5000	RETAIL MANAGER	12600	5	3005

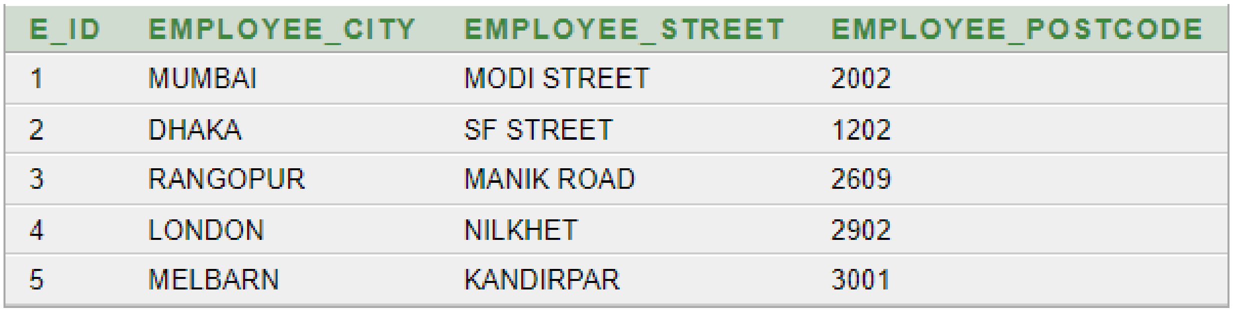
1.insert into e_address values (Emp_Add.nextval,'MUMBAI','MODI STREET','2002')

2.insert into e_address values (Emp_Add.nextval,'DHAKA','SF STREET','1202')

3.insert into e_address values (Emp_Add.nextval,'RANGPUR',' MANIK ROAD','2609')

4.insert into e_address values (Emp_Add.nextval,'LONDON','NILKHET','2902')

5.insert into e_address values (Emp_Add.nextval,'MELBARN','KANDIRPAR','3001')



E_ID	EMPLOYEE_CITY	EMPLOYEE_STREET	EMPLOYEE_POSTCODE
1	MUMBAI	MODI STREET	2002
2	DHAKA	SF STREET	1202
3	RANGOPUR	MANIK ROAD	2609
4	LONDON	NILKHET	2902
5	MELBARN	KANDIRPAR	3001

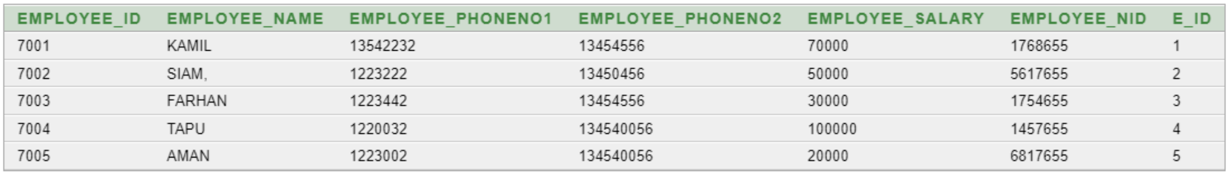
1.insert into employee values (7001,'KAMIL',013542232,013454556,70000,1768655,1)

2.insert into employee values (7002,'SIAM,',01223222,013450456,50000,5617655,2)

3.insert into employee values (7003,'FARHAN',01223442,013454556,30000,1754655,3)

4.insert into employee values (7004,'TAPU',01220032,0134540056,100000,1457655,4)

5.insert into employee values (7005,'AMAN',01223002,0134540056,20000,6817655,5)



EMPLOYEE_ID	EMPLOYEE_NAME	EMPLOYEE_PHONENO1	EMPLOYEE_PHONENO2	EMPLOYEE_SALARY	EMPLOYEE_NID	E_ID
7001	KAMIL	13542232	13454556	70000	1768655	1
7002	SIAM,	1223222	13450456	50000	5617655	2
7003	FARHAN	1223442	13454556	30000	1754655	3
7004	TAPU	1220032	134540056	100000	1457655	4
7005	AMAN	1223002	134540056	20000	6817655	5

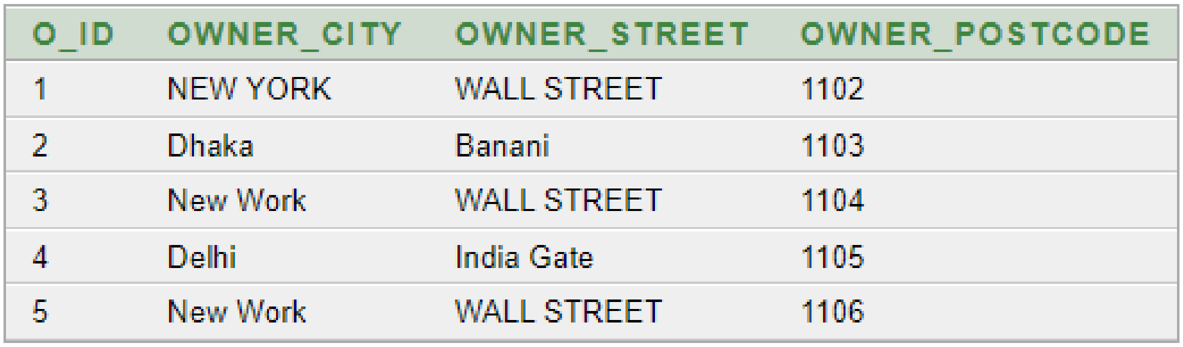
1.insert into o_address values (1,'New York','WALL STREET','1102')

2.insert into o_address values (2,'Dhaka','Banani','1103')

3.insert into o_address values (3,'New Work','WALL STREET','1104')

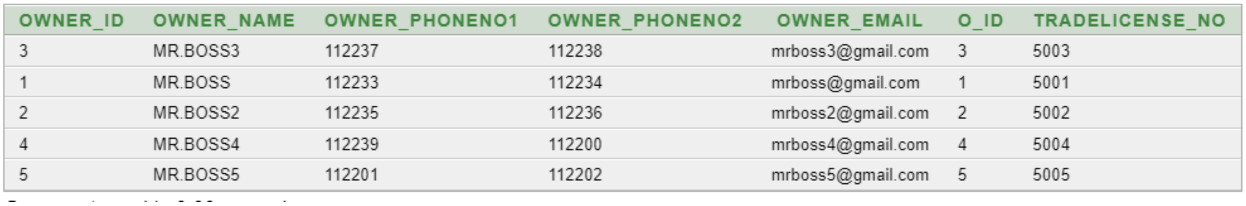
4.insert into o_address values (4,'Delhi','India Gate','1105')

5.insert into o_address values (5,'New Work','WALL STREET','1106')



O_ID	OWNER_CITY	OWNER_STREET	OWNER_POSTCODE
1	NEW YORK	WALL STREET	1102
2	Dhaka	Banani	1103
3	New Work	WALL STREET	1104
4	Delhi	India Gate	1105
5	New Work	WALL STREET	1106

```
1.insert into owner values (1,'MR.BOSS',112233, 112234, 'mrboss@gmail.com',1,5001)

2.insert into owner values (2,'MR.BOSS2', 112235, 112236, 'mrboss2@gmail.com',2,5002)

3.insert into owner values (3,'MR.BOSS3', 112237, 112238, 'mrboss3@gmail.com',3,5003)

4.insert into owner values (4,'MR.BOSS4', 112239, 112200, 'mrboss4@gmail.com',4,5004)

5.insert into owner values (5,'MR.BOSS5', 112201, 112202, 'mrboss5@gmail.com',5,5005)
```



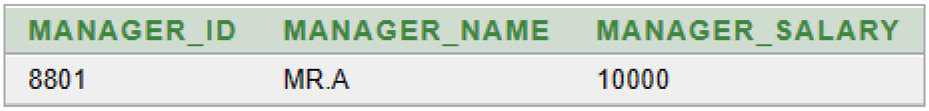
OWNER_ID	OWNER_NAME	OWNER_PHONENO1	OWNER_PHONENO2	OWNER_EMAIL	O_ID	TRADELICENSE_NO
3	MR.BOSS3	112237	112238	mrboss3@gmail.com	3	5003
1	MR.BOSS	112233	112234	mrboss@gmail.com	1	5001
2	MR.BOSS2	112235	112236	mrboss2@gmail.com	2	5002
4	MR.BOSS4	112239	112200	mrboss4@gmail.com	4	5004
5	MR.BOSS5	112201	112202	mrboss5@gmail.com	5	5005

Query Writing

Single Row Function

1. Print manager ID,name and salary using Single Row Function (CASE CONVERSION FUNCTION)

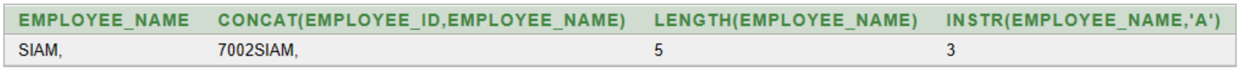
select manager_id,manager_name,manager_salary from manager where lower(manager_name)='mr.a'



MANAGER_ID	MANAGER_NAME	MANAGER_SALARY
8801	MR.A	10000

2. Print Employee name,concat name and id,lenght of employee name and the position of 'A' in employee name using Single Row Function(CHARACTER MANIPULATION FUNCTIONS)

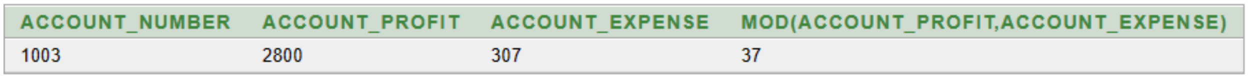
select employee_name, concat (employee_id, employee_name), length(employee_name), instr(employee_name, 'a') from employee where employee_id=7002



EMPLOYEE_NAME	CONCAT(EMPLOYEE_ID,EMPLOYEE_NAME)	LENGTH(EMPLOYEE_NAME)	INSTR(EMPLOYEE_NAME,'A')
SIAM,	7002SIAM,	5	3

3. Print the account number and the modulus of account profit and account expense whose account number is 1003 by using Single Row Function(NUMBER FUNCTION-MOD)

select account_number, account_profit,account_expense, mod(account_profit, account_expense) from account_no where account_number = 1003

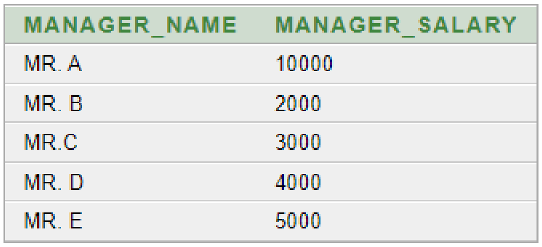


ACCOUNT_NUMBER	ACCOUNT_PROFIT	ACCOUNT_EXPENSE	MOD(ACCOUNT_PROFIT,ACCOUNT_EXPENSE)
1003	2800	307	37

Group Function

1.Display the name and salary of the managers who has the max salary group by m_id

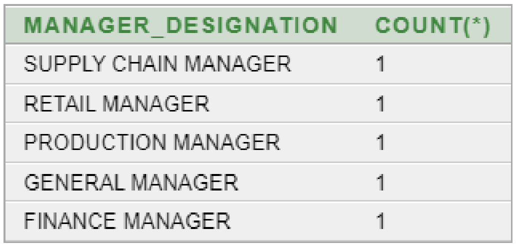
select manager_name, manager_salary from manager where manager_salary in (select max(manager_salary) from manager group by m_id)



MANAGER_NAME	MANAGER_SALARY
MR. A	10000
MR. B	2000
MR.C	3000
MR. D	4000
MR. E	5000

2.Display the number of managers from each designation group by their designation in descending order

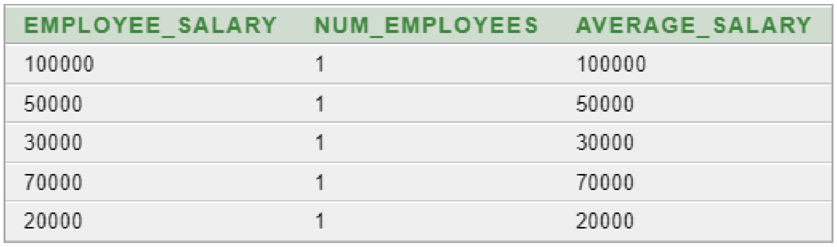
select manager_designation, count(*) from manager group by manager_designation order by manager_designation desc



MANAGER_DESIGNATION	COUNT(*)
SUPPLY CHAIN MANAGER	1
RETAIL MANAGER	1
PRODUCTION MANAGER	1
GENERAL MANAGER	1
FINANCE MANAGER	1

3.Display the employee salary,average salary,no of emplyees group by salay

select employee_salary, count(*) as num_employees, avg(employee_salary) as average_salary from employee group by employee_salary

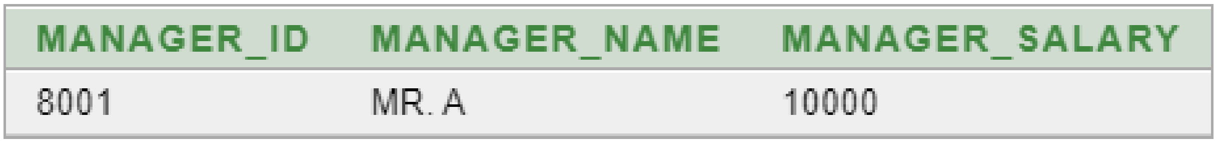


EMPLOYEE_SALARY	NUM_EMPLOYEES	AVERAGE_SALARY
100000	1	100000
50000	1	50000
30000	1	30000
70000	1	70000
20000	1	20000

Subqueries

1.Write a subquery that displays manager's name, id, salary those salaries are greater than RETAIL MANAGER.

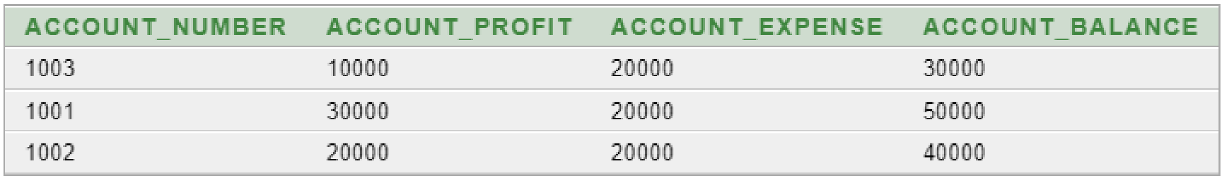
select manager_id, manager_name, manager_salary from manager where manager_salary > (select manager_salary from manager where manager_designation='RETAIL MANAGER')



MANAGER_ID	MANAGER_NAME	MANAGER_SALARY
8001	MR. A	10000

2.Display the account information from account table which has the highest account expense

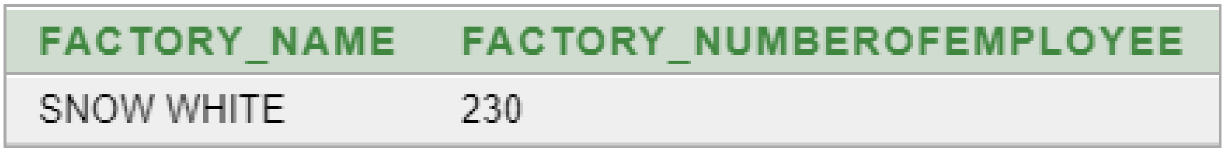
select *from account_no where account_expense in (select max(account_expense) from account_no)



ACCOUNT_NUMBER	ACCOUNT_PROFIT	ACCOUNT_EXPENSE	ACCOUNT_BALANCE
1003	10000	20000	30000
1001	30000	20000	50000
1002	20000	20000	40000

3.Display the factory name, employee numbers which has the lowest number of employee

select factory_name, factory_numberofemployee from factory where factory_numberofemployee in (select min(factory_numberofemployee) from factory)

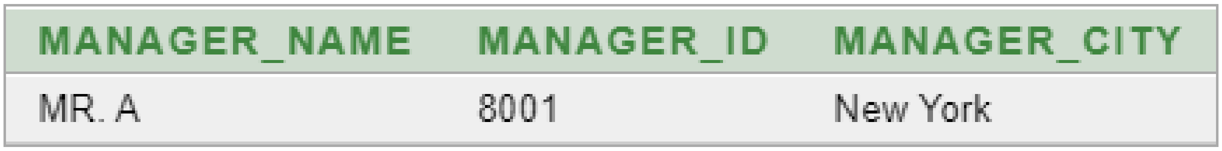


FACTORY_NAME	FACTORY_NUMBEROFEMPLOYEE
SNOW WHITE	230

Joining

1.Display the manager name, id, city who lives in New York.

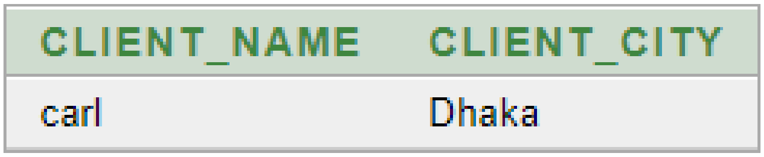
select m.manager_name, m.manager_id, n.manager_city from manager m join m_address n on m.m_id = n.m_id where n.manager_city = 'New York';



MANAGER_NAME	MANAGER_ID	MANAGER_CITY
MR. A	8001	New York

2.Display client name and client of only those clients who lives in DHAKA using natural join

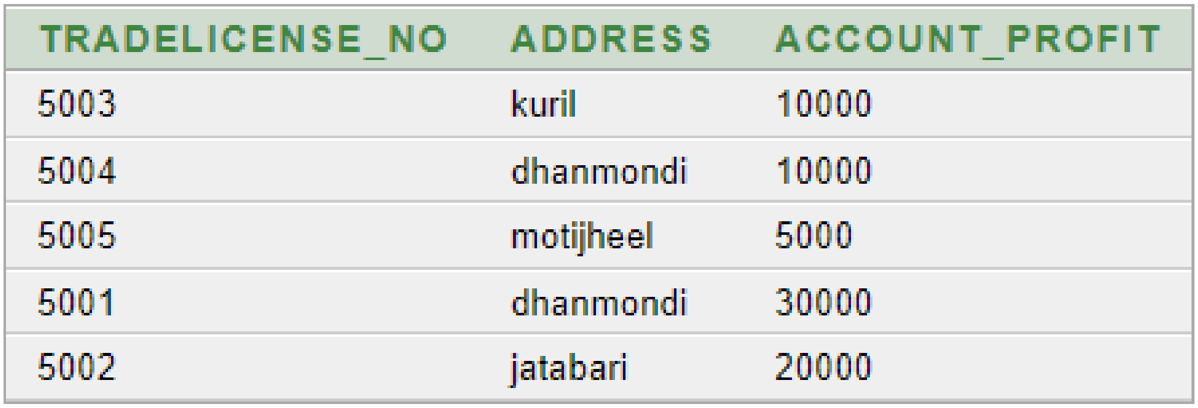
select c.client_name, d.client_city from client c natural join c_address d where d.client_city = 'Dhaka';



CLIENT_NAME	CLIENT_CITY
carl	Dhaka

3.Display the trade license number address and account_profit

select t.tradelicense_no, t.address, a.account_profit from tradelc_no t join account_no a on t.account_number = a.account_number

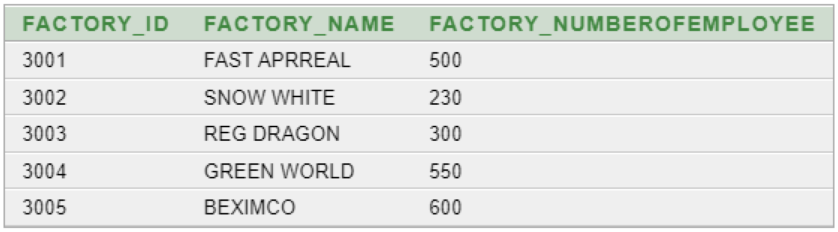


TRADELICENSE_NO	ADDRESS	ACCOUNT_PROFIT
5003	kuril	10000
5004	dhanmondi	10000
5005	motijheel	5000
5001	dhanmondi	30000
5002	jatabari	20000

View

1.Create a view named FACTORY_VIEW based on FACTORY table which shows the factory id, name, number of employees.

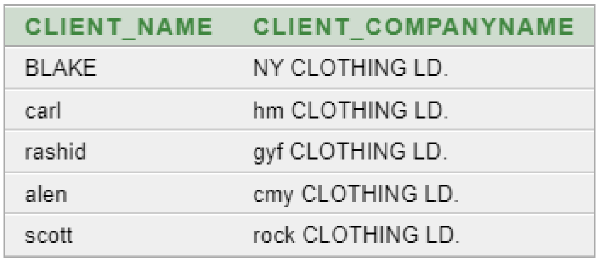
create view factory_view as select factory_id, factory_name, factory_numberofemployee from factory



FACTORY_ID	FACTORY_NAME	FACTORY_NUMBEROFEMPLOYEE
3001	FAST APRREAL	500
3002	SNOW WHITE	230
3003	REG DRAGON	300
3004	GREEN WORLD	550
3005	BEXIMCO	600

2.Create a view named CLIENT_VIEW based on CLIENT table which shows the client name,company name.

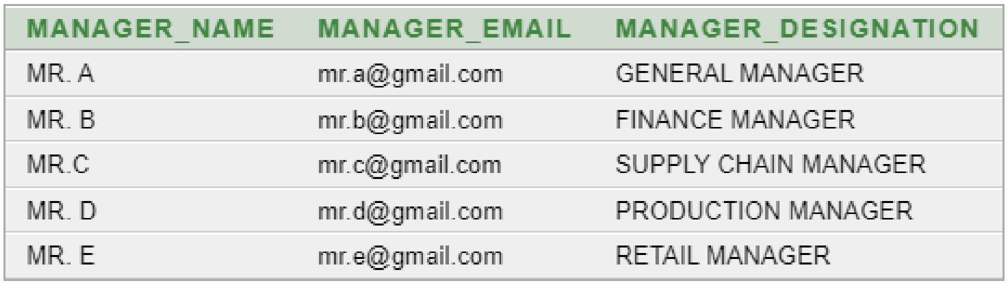
create view client_view as select client_name, client_companyname from client



CLIENT_NAME	CLIENT_COMPANYNAME
BLAKE	NY CLOTHING LD.
carl	hm CLOTHING LD.
rashid	gyf CLOTHING LD.
alen	cmy CLOTHING LD.
scott	rock CLOTHING LD.

3.Create a view named MANAGER_VIEW based on MANAGER table which shows the manager name,email,designation.

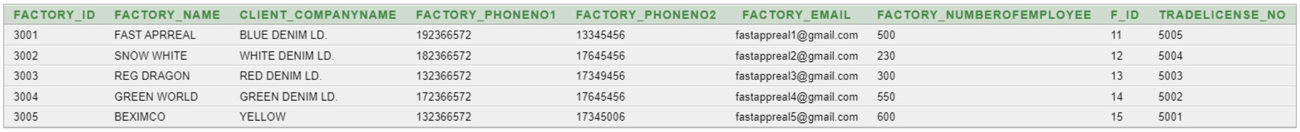
create view manager_view as select manager_name, manager_email, manager_designation from manager



MANAGER_NAME	MANAGER_EMAIL	MANAGER_DESIGNATION
MR. A	mr.a@gmail.com	GENERAL MANAGER
MR. B	mr.b@gmail.com	FINANCE MANAGER
MR.C	mr.c@gmail.com	SUPPLY CHAIN MANAGER
MR. D	mr.d@gmail.com	PRODUCTION MANAGER
MR. E	mr.e@gmail.com	RETAIL MANAGER

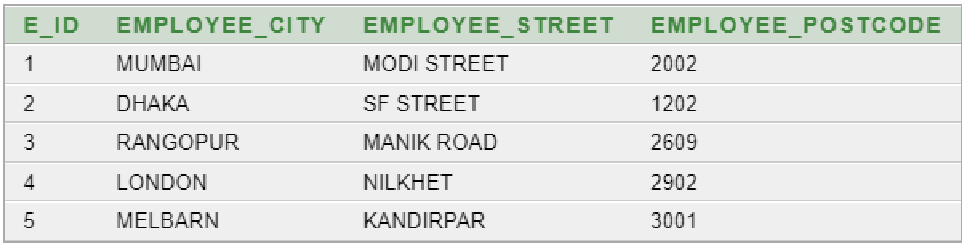
Synonym

1.create synonym production_house for factory



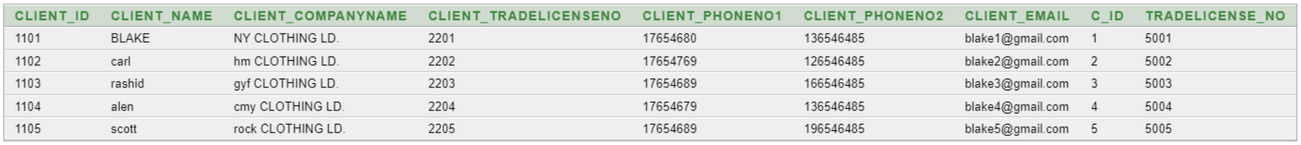
FACTORY_ID	FACTORY_NAME	CLIENT_COMPANYNAME	FACTORY_PHONENO1	FACTORY_PHONENO2	FACTORY_EMAIL	FACTORY_NUMBEROFEMPLOYEE	F_ID	TRADELICENSE_NO
3001	FAST APRREAL	BLUE DENIM LD.	192366572	13345456	fastappreal1@gmail.com	500	11	5005
3002	SNOW WHITE	WHITE DENIM LD.	182366572	17645456	fastappreal2@gmail.com	230	12	5004
3003	REG DRAGON	RED DENIM LD.	132366572	17349456	fastappreal3@gmail.com	300	13	5003
3004	GREEN WORLD	GREEN DENIM LD.	172366572	17645456	fastappreal4@gmail.com	550	14	5002
3005	BEXIMCO	YELLOW	132366572	17345006	fastappreal5@gmail.com	600	15	5001

2.create synonym employee_address for e_address



E_ID	EMPLOYEE_CITY	EMPLOYEE_STREET	EMPLOYEE_POSTCODE
1	MUMBAI	MODI STREET	2002
2	DHAKA	SF STREET	1202
3	RANGOPUR	MANIK ROAD	2609
4	LONDON	NILKHET	2902
5	MELBARN	KANDIRPAR	3001

3.create synonym consumers for client



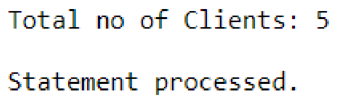
CLIENT_ID	CLIENT_NAME	CLIENT_COMPANYNAME	CLIENT_TRADELICENSENO	CLIENT_PHONENO1	CLIENT_PHONENO2	CLIENT_EMAIL	C_ID	TRADELICENSE_NO
1101	BLAKE	NY CLOTHING LD.	2201	17654680	136546485	blake1@gmail.com	1	5001
1102	carl	hm CLOTHING LD.	2202	17654769	126546485	blake2@gmail.com	2	5002
1103	rashid	gyf CLOTHING LD.	2203	17654689	166546485	blake3@gmail.com	3	5003
1104	alen	cmy CLOTHING LD.	2204	17654679	136546485	blake4@gmail.com	4	5004
1105	scott	rock CLOTHING LD.	2205	17654689	196546485	blake5@gmail.com	5	5005

PL/SQL

Function

1.Create a function that returns the total number of clients.

```
create or replace function total_clients
return number as
  total number := 0;
begin
  select count(*) into total from client;
  return total;
end;

declare
  c number;
begin
  c := total_clients();
  dbms_output.put_line('Total no of Clients: ' || c);
end;
```

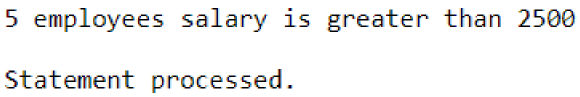


```
Total no of Clients: 5

Statement processed.
```

2.Create a function that returns where employee salary is greater than 2500.

```
create or replace function salary_employees
return number
is
  v_count number := 0;
```

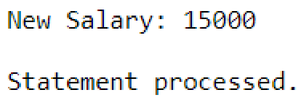
```
begin
  select count(*) into v_count from employee where employee_salary > 2500;
  return v_count;
end;


declare
  s number;
begin
  s := salary_employees();
  dbms_output.put_line(s || ' employees salary is greater than 2500');
end;
```



```
5 employees salary is greater than 2500

Statement processed.
```

3.Create a fuction that increases the manager salary with 2500 by using only manager's id number.

```
create or replace function increase_salary(p_manager_id number)
return number
is
  v_new_salary number;
begin
  select manager_salary + 2500 into v_new_salary from manager where manager_id = p_manager_id;


  update manager set manager_salary = v_new_salary where manager_id = p_manager_id;
  return v_new_salary;
end;


declare
  n_new_salary number;
begin
  n_new_salary := increase_salary(8001);
  dbms_output.put_line('New Salary: ' || n_new_salary);
end;
```



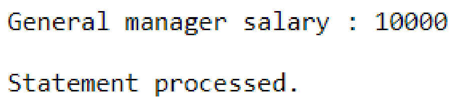
```
New Salary: 15000

Statement processed.
```

<u>**Procedure**</u>

1.Create a procedure to update the salary of GENERAL MANAGER to 10000.

```
create or replace procedure update_salary
as
begin
   update manager set manager_salary=10000 where manager_designation='GENERAL MANAGER';
End;


Declare
   sal number(6);
begin
   update_salary ;
```

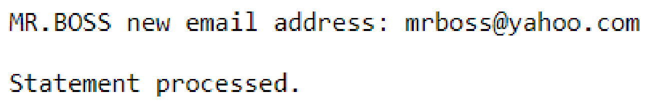
```
  select manager_salary into sal from manager where manager_designation = 'GENERAL MANAGER';
  dbms_output.put_line('General manager salary : '||sal);
End;
```



```
General manager salary : 10000

Statement processed.
```

2. Create a procedure to update the MR.BOSS email address to mrboss@yahoo.com

```
create or replace procedure update_owner_email (n_email in varchar2)
as
begin
   update owner set owner_email = n_email where owner_name = 'MR.BOSS';
End;



Declare
  v_email varchar2(50);
begin
  update_owner_email('mrboss@yahoo.com');
  select owner_email into v_email from owner where owner_name = 'MR.BOSS';
  dbms_output.put_line('MR.BOSS new email address: ' || v_email);
End;
```

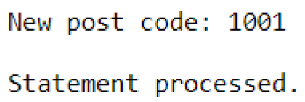


```
MR.BOSS new email address: mrboss@yahoo.com

Statement processed.
```

3. Create a procedure to update Factory 11's postcode to 1001

```
create or replace procedure update_postcode (n_postcode in number)
as
```

```
Begin
   update f_address set factory_postcode = n_postcode where f_id = 11;
End;


Declare
  v_factory_postcode number;
Begin
  update_postcode(1001);
  select factory_postcode into v_factory_postcode from f_address where f_id = 11;
  dbms_output.put_line('New post code: ' || v_factory_postcode);
End;
```

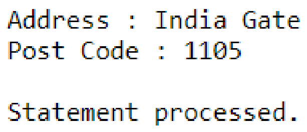


```
New post code: 1001

Statement processed.
```

Record

1.Create a record that can show the address and postcode where city is Delhi.

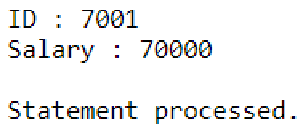
```
declare
  o_address_rec o_address%rowtype;
begin
  select *
  into o_address_rec
  from o_address
  where owner_city = 'Delhi';
  dbms_output.put_line('Address : ' || o_address_rec.Owner_Street);
  dbms_output.put_line('Post Code : ' || o_address_rec.Owner_Postcode);
end;
```



```
Address : India Gate
Post Code : 1105

Statement processed.
```

2.Create a record that can output the id and salary of the employee whose name is KAMIL.

```
declare
  cursor c_employee is
    select * from employee where employee_name = 'KAMIL';
  rec_employee employee%rowtype;
begin
  open c_employee;
  fetch c_employee into rec_employee;
  dbms_output.put_line('ID : ' || rec_employee.Employee_Id);
  dbms_output.put_line('Salary : ' || rec_employee.Employee_Salary);
  close c_employee;
end;
```

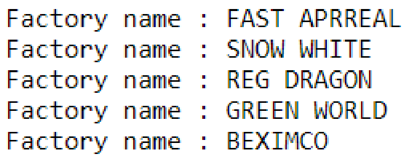


```
ID : 7001
Salary : 70000

Statement processed.
```

3.Create a record that can output the factory names.

```
declare
  cursor c_factory is select factory_name from factory;
  factory_name factory.factory_name%type;
begin
  open c_factory;
```

```
 loop
   fetch c_factory into factory_name;
   exit when c_factory%notfound;
   dbms_output.put_line('Factory name : ' || factory_name);
 end loop;
 close c_factory;
end;
```

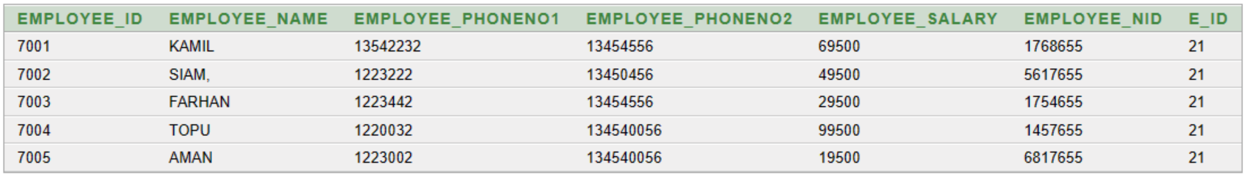


```
Factory name : FAST APRREAL
Factory name : SNOW WHITE
Factory name : REG DRAGON
Factory name : GREEN WORLD
Factory name : BEXIMCO
```

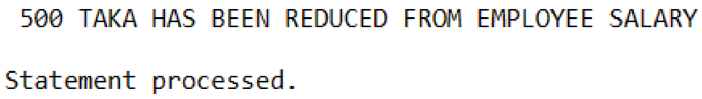
Cursor

1. Create a cursor that will reduce 500 takas from the employee salary.

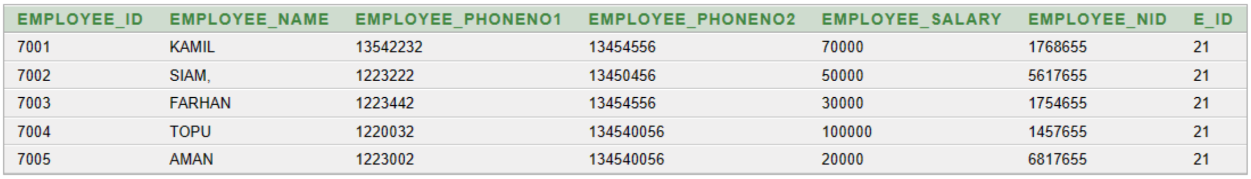
```
begin
update employee
set employee_salary = employee_salary - 500;
if sql%notfound then
dbms_output.put_line('NO SALARY HAS BEEN RDUCED');
 ELSIF sql%found THEN
dbms_output.put_line(' 500 TAKA HAS BEEN REDUCED FROM EMPLOYEE SALARY ');
 end if;
end;
rollback;
select * from employee
```



EMPLOYEE_ID	EMPLOYEE_NAME	EMPLOYEE_PHONENO1	EMPLOYEE_PHONENO2	EMPLOYEE_SALARY	EMPLOYEE_NID	E_ID
7001	KAMIL	13542232	13454556	69500	1768655	21
7002	SIAM,	1223222	13450456	49500	5617655	21
7003	FARHAN	1223442	13454556	29500	1754655	21
7004	TOPU	1220032	134540056	99500	1457655	21
7005	AMAN	1223002	134540056	19500	6817655	21



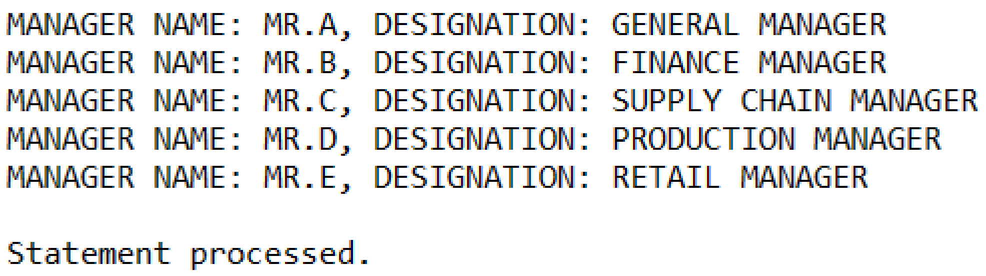
```
 500 TAKA HAS BEEN REDUCED FROM EMPLOYEE SALARY

Statement processed.
```



EMPLOYEE_ID	EMPLOYEE_NAME	EMPLOYEE_PHONENO1	EMPLOYEE_PHONENO2	EMPLOYEE_SALARY	EMPLOYEE_NID	E_ID
7001	KAMIL	13542232	13454556	70000	1768655	21
7002	SIAM,	1223222	13450456	50000	5617655	21
7003	FARHAN	1223442	13454556	30000	1754655	21
7004	TOPU	1220032	134540056	100000	1457655	21
7005	AMAN	1223002	134540056	20000	6817655	21

2. Create a cursor that will display the name and designation of all the managers.

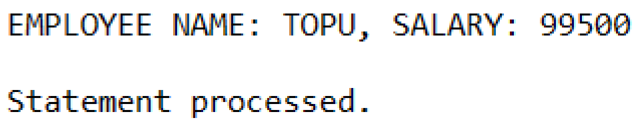
```
declare
  cursor mgr_cursor is
    select manager_name, manager_designation from manager;
  ename_var manager.manager_name%type;
  job_var manager.manager_designation%type;
begin
  open mgr_cursor;
  loop
    fetch mgr_cursor into ename_var, job_var;
    exit when mgr_cursor%notfound;
    dbms_output.put_line('MANAGER NAME: ' || ename_var || ', DESIGNATION: ' || job_var);
  end loop;
  close mgr_cursor;
End;
```



```
MANAGER NAME: MR.A, DESIGNATION: GENERAL MANAGER
MANAGER NAME: MR.B, DESIGNATION: FINANCE MANAGER
MANAGER NAME: MR.C, DESIGNATION: SUPPLY CHAIN MANAGER
MANAGER NAME: MR.D, DESIGNATION: PRODUCTION MANAGER
MANAGER NAME: MR.E, DESIGNATION: RETAIL MANAGER

Statement processed.
```

3. Create a cursor that will display the name, salary of employee who has the highest salary.

```
declare
 cursor emp_cursor is
  select employee_name,employee_salary
  from employee
  where employee_salary = (select max(employee_salary) from employee);
 ename_var employee.employee_name%type;
 sal_var employee.employee_salary%type;
begin
 open emp_cursor;
 fetch emp_cursor into ename_var, sal_var;
 if emp_cursor%found then
  dbms_output.put_line('EMPLOYEE NAME: ' || ename_var || ', SALARY: ' || sal_var);
 ELSE
  dbms_output.put_line('NO EMPLOYEE FOUND');
 END IF;
 close emp_cursor;
End;
```

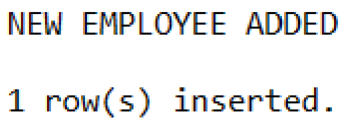


```
EMPLOYEE NAME: TOPU, SALARY: 99500

Statement processed.
```

Trigger

1. Create a trigger which will execute when we add a new employee.

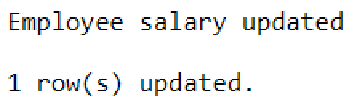
```
create or replace trigger employee_insert_trigger
after insert on employee
for each row
begin
  dbms_output.put_line('NEW EMPLOYEE ADDED');
End;


insert into employee values
(7006,'BATMAN',010743598,01642241177,20000,43547578,1)
```



```
NEW EMPLOYEE ADDED

1 row(s) inserted.
```

2. Create a trigger which will execute when we update the employee's salary.

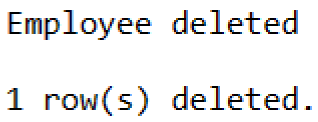
```
create or replace trigger emp_salary_update_trigger
after update of employee_salary on employee
for each row
begin
  dbms_output.put_line('Employee salary updated');
End;


update employee set employee_salary=1111 where employee_id=7001
```



```
Employee salary updated

1 row(s) updated.
```

3. Create a trigger which will execute when we delete an employee.

```
create or replace trigger emp_delete_trigger
after delete on employee
for each row
begin
  dbms_output.put_line('Employee deleted');
End;
delete from employee where employee_id =7005;
```



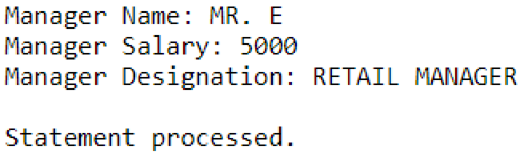
```
Employee deleted

1 row(s) deleted.
```

Package

1. Create a package which will display the name, salary, designation of manager

```
create or replace package mgr_details_pkg
is
  procedure get_mgr_details(mgr_id in number);
end;


create or replace package body mgr_details_pkg
is
  procedure get_mgr_details(mgr_id in number)
  is
    mgr_name manager.manager_name%type;
    mgr_salary manager.manager_salary%type;
    mgr_designation manager.manager_designation%type;
  begin
```

```
    select manager_name,manager_salary, manager_designation
    into mgr_name, mgr_salary, mgr_designation
    from manager
    where manager_id = mgr_id;
    dbms_output.put_line('Manager Name: ' || mgr_name);
    dbms_output.put_line('Manager Salary: ' || mgr_salary);
    dbms_output.put_line('Manager Designation: ' || mgr_designation);
  End;
End;


Begin
  mgr_details_pkg.get_mgr_details(8005);
End;
```

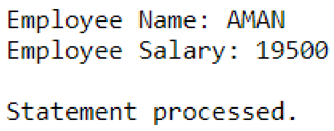


```
Manager Name: MR. E
Manager Salary: 5000
Manager Designation: RETAIL MANAGER

Statement processed.
```

2. Create a package to display the name, salary of employee who has the lowest salary from employee table.

```
create or replace package employee_info_pkg
is
  procedure get_lowest_salary_employee;
end;


create or replace package body employee_info_pkg
is
  procedure get_lowest_salary_employee
```

```
 is
  emp_name employee.employee_name%type;
  emp_salary employee.employee_salary%type;
 begin
  select employee_name, employee_salary
  into emp_name, emp_salary
  from employee
  where employee_salary = (select min(employee_salary) from employee);
  dbms_output.put_line('Employee Name: ' || emp_name);
  dbms_output.put_line('Employee Salary: ' || emp_salary);
 End;
End;


Begin
 employee_info_pkg.get_lowest_salary_employee;
End;
```

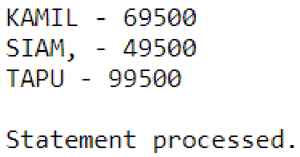


```
Employee Name: AMAN
Employee Salary: 19500

Statement processed.
```

3. Create a package to display the employee name,salary from employee table who has slary more than 30000.

```
create or replace package empl_pkg is
  procedure display_high_salary_emp;
end empl_pkg;


create or replace package body empl_pkg is
```

```
  procedure display_high_salary_emp is
    cursor emp_cur is
      select employee_name, employee_salary
      from employee
      where employee_salary > 30000;
    emp_rec emp_cur%rowtype;
  begin
    for emp_rec in emp_cur loop
      dbms_output.put_line(emp_rec.employee_name || ' - ' ||
emp_rec.employee_salary);
    End Loop;
  End display_high_salary_emp;
End empl_pkg;


Begin
  empl_pkg.display_high_salary_emp;
End;
```



```
KAMIL - 69500
SIAM, - 49500
TAPU - 99500

Statement processed.
```

Relational Algebra

1.Find the factory Id and factory name from the Factory.

$$\prod_{Factory\_Id,Factory\_Name}(Factory)$$

2. Find the street name where city is New York from Client Address.

$$\prod_{Client\_Street}(\sigma_{Client\_City\ =\ \text{“New York”}}(C\_Address))$$

3. Find the manager name where salary is 10000 from Manager.

$$\prod_{Manager\_Name}(\sigma_{Manager\_Salary=\text{“10000”}}(Manager))$$

4. Find the factory name where factory trade license no is 5005.

$$\prod_{Factory\_Name}(\sigma\ _{Tradelicense\_no=\ “5005”}(Factory))$$

5. Find the account number whose account balance is greater than 10000.

$$\prod_{Account\_number}(\sigma_{Account\_balance\ >\ 10000}(Account\_no))$$

Conclusion

Database Management System is a software that analyses and store data to use later according to our requirements. Our project is about Garment Factory Management System where we stored information about the office, factory, client, manager, employee, owner, and account. In our database, we can also retrieve and modify data. In the future, we can add biometrics and Artificial intelligence to the system and organize it professionally. We can capitalize the cache and use virtualizing technology in our project. We can add more memory to minimize the time of the process. We can also improve the network system to work more efficiently.