**INVENTORY STORE MANAGEMENT SYSTEM**

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

**Bachelor of Technology/Master of Technology**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

**K . Jaya Prakash**

**(AP20110010403)**

**A picture containing text

Description automatically generated**

Under the Guidance of

**Prof. Dr Rajiv Senapati**

Department of Computer science Engineering

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**[December, 2022]**

# Certificate

Date: 15-Dec-22

This is to certify that the work present in this Project entitled “**INVENTORY STORE MANAGEMENT SYSTEM**” has been carried out by **K. Jaya Prakash** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

**Supervisor**

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

**Co-supervisor**

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

# Acknowledgements

We thank **Dr**. **Rajiv** **Senapati**, our professor in charge, for the help and direction finishing the project on the subject inventory store management system. It was a fantastic learning opportunity. This subject has given me an opportunity to explore the field I always have been curious about. Your insightful counsel and recommendations were quite beneficial to me as I finished the assignment. I will always be grateful to you for this.

# Table of Contents

[Certificate i](#_Toc122018809)

[Acknowledgements iii](#_Toc122018810)

[Table of Contents iv](#_Toc122018811)

[Abstract vi](#_Toc122018812)

[1. Introduction 1](#_Toc122018813)

[2. Methodology 3](#_Toc122018814)

[3. List of Entities : 3](#_Toc122018815)

[4. ER MODEL: 6](#_Toc122018817)

[5. ER MODEL TO RELATIONAL MODEL 7](#_Toc122018818)

[6. Normalization : 9](#_Toc122018819)

[7. SQL Code for the project: 19](#_Toc122018820)

[8. SQL QUERIES : 32](#_Toc122018821)

[9. Concluding Remarks 36](#_Toc122018822)

[10. References : 37](#_Toc122018823)

# Abstract

Data needs to be stored in an organized way so that it is easy to use and manipulate the information. This Project talks about inventory store management system which is an combination of (software and hardware) and process and procedures that oversee the monitoring and maintenance of stocked products, which includes management of raw materials, components and finished products. This system can be widely used by normal shops and departmental stores for keeping proper track of stock, it contains manager details, customer details etc. Checking requirements, finding all entities and their properties, ER diagram, converting ER to relational model, Normalizing, and SQL code are the steps followed. The main theme is to decrease the redundancy and inconsistency of the data.

# Introduction

There are various ways to store and retrieve the data. Older versions like using files for data organizing have many issues and take more manpower and time. Data redundancy, Data inconsistency, Data isolation, Atomicity, Concurrent access, Security are the few tasks which become an anomaly by using files. Database management system (DBMS) helps to hold all above issues.

Store management system is one of the applications where maintaining a database is required. In this report the simple requirements were considered. This database system stores the data about the availability of the products, information regarding sale statistics, details of manager so that customers can contact with them easily.

# Methodology

**Step 1 :**

# List of Entities :

1.Manager

2.Supplyers

3.Products

4.Store

5.Workers

6.Billing

7.Sales

8.Customer

9.Payment

10.Tax

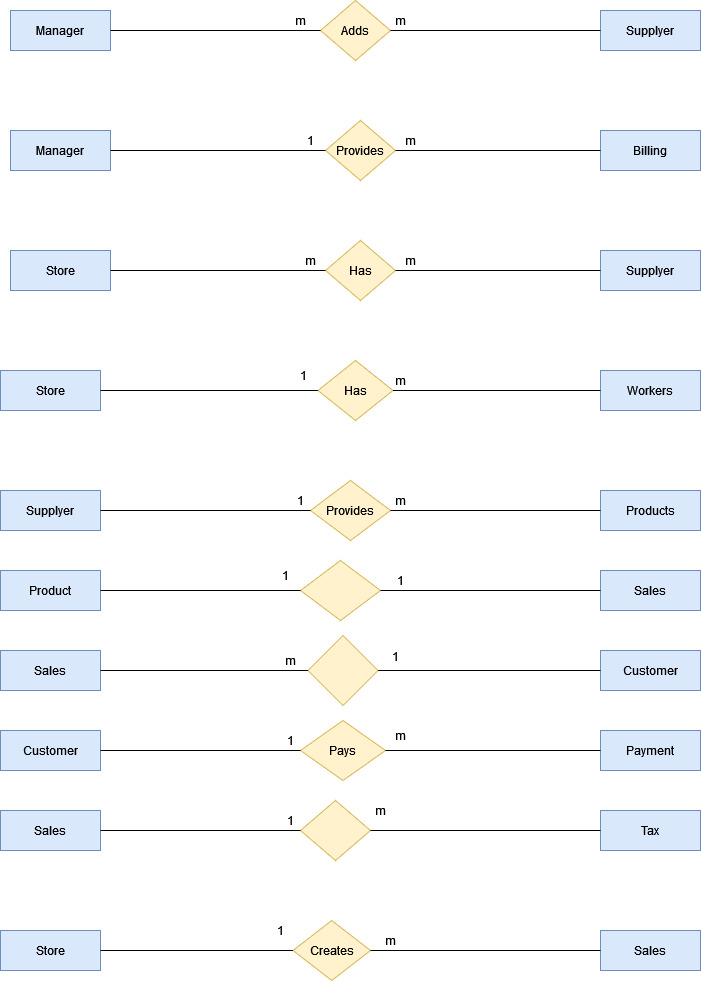


**Step 2:**

**List of attributes :**

1. Manager(manager\_id , manager\_name , manager\_Qualification , manager\_ContactNumber, manager\_dateofbirth, , manager\_gender, manager\_address).
2. Supplyer(supplyer\_id , supplyer\_name, supplyer\_address, supplyer\_contactnumber).
3. Products(product\_id , product\_name , product\_type , product\_cost , product\_discription).
4. Store(store\_id , store\_name , store\_contactnumber , store\_address).
5. Workers(worker\_id , worker\_name , worker\_type , worker\_salary , worker\_contact , worker\_gender).
6. Billing(billing\_id , billing\_date , billing\_amount).
7. Sales(sales\_id , sales\_cost , sales\_discription).
8. Customer(customer\_id , customer\_name , customer\_address , customer\_contact , customer\_email\_id).
9. Payment(payment\_id , payment\_date , payment\_amount , payment\_type , payment\_status).
10. Tax(tax\_id , tax\_amount , tax\_type).

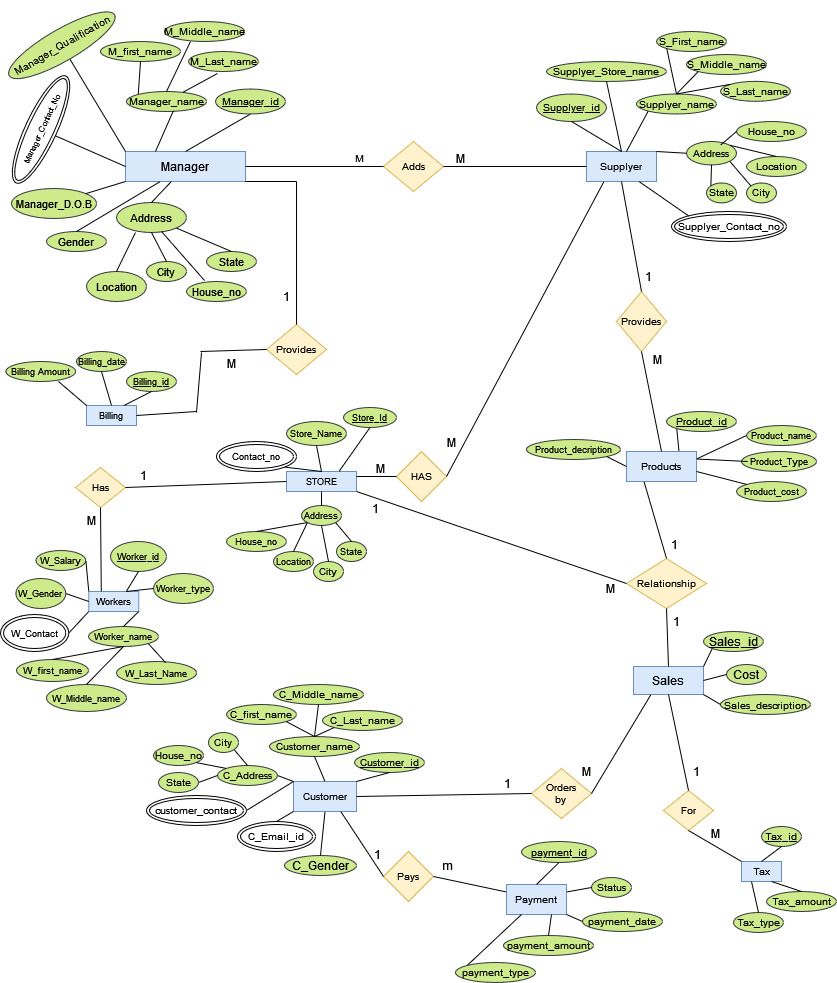
**Step 3 :**

****

# ER MODEL:

**ER Diagram:**

# 

****

# ER MODEL TO RELATIONAL MODEL

**RELATIONAL MODEL :** Data is represented as relation (or table) which consists of rows and columns.

Below are the rules used for conversion of ER model to Relational model

1. An entity in the ER model is represented by a relational table in the relational model.

2. All the attributes of the ER model are represented in different columns as an attribute.

3. Primary key attribute of the ER model is represented as the primary key in the relational model.

4. Composite key of the ER model is split in different columns in the relational model.

5. Derived attributes must be dropped in relational models.

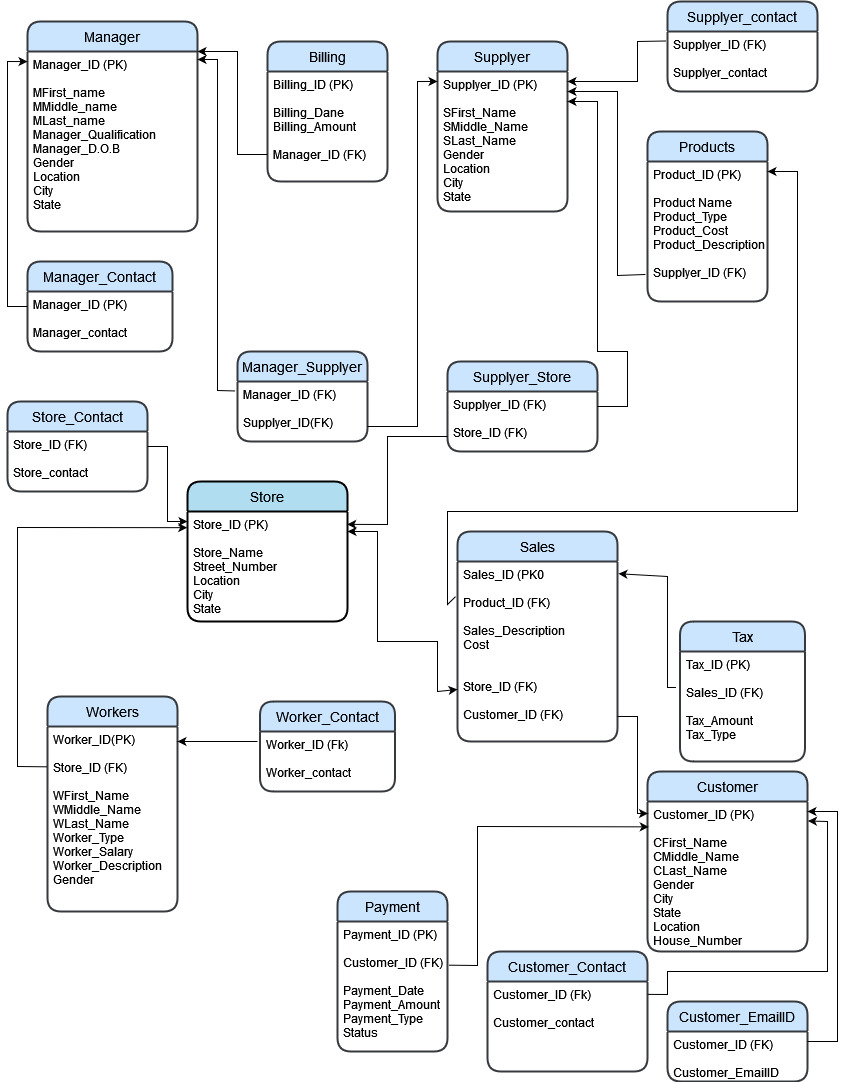
6. Multi valued attributes need to be kept separate in a new table.

7. If the relationship between two entities is 1:1 then the primary key of one relation becomes foreign key in another relation.

8. If the relationship between entities is 1:M or M:1 then the primary key attribute of one-sided relations becomes a foreign key in many-sided relations.

9. If the relationship between entities is M:M then a new table needs to be created to represent that relation, where the new relation will have the primary key of both relations as the foreign keys.

**Relational model:**

****

# Normalization :

Normalization is a process of analyzing and decomposing the complex relation which satisfies some constraint to form a simple relation.

There are 5 Normal forms:

1. First Normal Form (1NF)

2. Second Normal Form (2NF)

3. Third Normal Form (3NF) or Boyes Cord Normal Form (BCNF)

4. Fourth Normal Form (4NF)

5. Fifth Normal Form (5F) or Project Join Normal Form (PJNF)

**Manager Table:**

Attributes :

Manager-ID, Mfirst\_name, Mmiddle\_name, Mlast-name, Manager\_qualification, Manager-DOB, House-number, Location, City, State.

functional Dependencies:-

Manager – ID 🡪 Mfirst\_name

Manager ID 🡪 Mmiddle\_name

Manager ID 🡪 Mlast\_name

Manager ID 🡪 Manager\_qualification

Manager-ID 🡪 Manager\_D.O.B

Manager-ID 🡪 Location

Manager-ID 🡪 City

Manager-ID 🡪 State

Manager-ID 🡪 Gender

Primary\_Key: Manager\_ID

**First Normal Form(1NF):**

CONDITION: If a relation is said to be in 1NF only if it has no non atomic attributes in it.

The table is already in First Normal Form because there are no non atomic attributes present in it.

**Second Normal Form(2NF):**

If a relation is said to be in 2NF then the relation should be in 1NF and should satisfy any one of the below conditions.

CONDITIONS: 1. If the primary\_key consists of only one attribute.

2.If there exists no non key attributes.

3. Every non key attribute present in the relation should be fully functionally dependent on the full set of the primary\_key.

The table is already in 2NF because it is 1NF and satisfies the first condition of the second normal form.

**Third Normal Form(3NF):**

CONDITION: If a relation is said to be in 3NF only if the relation is in 1NF, 2NF & it has no transitive functional dependency in it.

The table is already in 3NF because it is in 1NF, 2NF and there is no transitive functional dependency in the table.

**Boyce Codd Normal Form(BCNF):**

CONDITION: If a relation is said to be in BCNF only if it is in 1NF, 2NF, 3NF and satisfies any one of the below conditions.

1. X 🡪 Y, Y be a subset of X
2. X is a Super Key

The table is already in BCNF because it is in 1NF, 2NF, 3NF & it satisfies the first condition of BCNF.

**Fourth Normal Form(4NF):**

If a relation is said to be in fourth normal form(4NF) only if it has no multivalued dependencies.

The table is already in 4NF because it has no multivalued dependencies.

**Manager\_Contact Table:**

Attributes:

Manager\_ID, Manager\_contact

Functional Dependencies:

Manager\_ID 🡪 Manager\_Contact

Primary & Foreign Key: Manager\_ID

**First Normal Form(1NF):**

The table is already in 1NF because it has no non atomic attributes present in it.

**Second Normal Form(2NF):**

The table is already in 2NF because it is in 1NF & satisfies the third condition of 2NF.

**Third Normal Form(3NF):**

The table is already in 3NF because the table is in 1NF, 2NF and it does not have any transitive functional dependency int it.

**Boyce Codd Normal Form(BCNF):**

The table is already in BCNF because the table is in 1NF, 2NF, 3NF & it satisfies the first condition of the Boyce Codd Normal Form.

**Fourth Normal Form(4NF):**

The table is already in 4NF because the table has no multivalued dependencies in it.

**Products Table:**

Attributes:

Product\_ID, Product\_Name , Product\_Type, Product\_cost, Product\_description

Functional Dependencies:

Product\_ID 🡪 Product\_name

Product\_ID 🡪 Product\_Type

Product\_ID 🡪 Product\_cost

Product\_ID 🡪 Product\_description

Primary Key: Product\_iD

**First Normal Form(1NF):**

The table is already in 1NF because the table has no non atomic attributes present in it.

**Second Normal Form(2NF):**

The table is already in 2NF because the table is in 1NF & has no partial functional dependencies in it.

**Third Normal Form(3NF):**

The table is already in 3NF because the table is in 1NF, 2NF & has no transitive functional dependency among the attributes.

**Fourth Normal Form(4NF):**

The table is already in 4NF because it has no multivalued dependency in it.

**Sales Table:**

Attributes:

Sales\_ID, cost, Sales\_description

Functional Dependencies:

Sales\_ID 🡪 cost

Sales\_ID 🡪 Sales\_description

Primary Key: Sales\_ID

**First Normal Form(1NF):**

The table is already in 1NF because the table has no non atomic attributes in it.

**Second Normal Form(2NF):**

The table is already in 2NF because the table is in 1NF & has only one Primary Key in it which satisfies the first condition of 2NF.

**Third Normal Form(3NF):**

The table is already in 3NF because the table is in 1NF, 2NF & has no transitive functional dependency among the attributes.

**Fourth Normal Form(4NF):**

The table is already in 4NF because it has no multivalued dependency in it.

**Store Table:**

Attributes:

Store\_ID, Store\_Name, Street\_name , Location, city, state

Functional Dependencies:  
Store\_ID 🡪 Store\_Name

Store\_ID 🡪 Street\_name

Store\_ID 🡪 Location

Store\_ID 🡪 city

Store\_ID 🡪 State

Primary Key: Showroom\_ID

**First Normal Form(1NF):**

The table is already in 1NF because the table has no non atomic attributes in it.

**Second Normal Form(2NF):**

The table is already in 2NF because the table is in 1NF & has no partial dependencies in it which satisfies the third condition of 2NF.

**Third Normal Form(3NF):**

The table is already in 3NF because the table is in 1NF, 2NF & has no transitive functional dependency among the attributes.

**Fourth Normal Form(4NF):**

The table is already in 4NF because it has no multivalued dependency in it.

**Worker\_Table:**

Attributes:

Worker\_ID, Wfirst\_name, Wmiddle\_name, Wlast\_name, Worker\_type, Worker\_salary, Worker\_description, Gender.

Worker\_ID 🡪 Wfirst\_name

Worker\_ID 🡪 Wmiddle\_name

Worker \_ID 🡪 Wlast\_name

Worker \_ID 🡪 Worker\_type

Worker \_ID 🡪 Worker\_salary

Worker \_ID 🡪 Worker\_description

Worker \_ID 🡪 Gender

Primary Key: Worker\_ID

**First Normal Form(1NF):**

The table is already in 1NF because the table has no non atomic attributes in it.

**Second Normal Form(2NF):**

The table is already in 2NF because the table is in 1NF & has only one primary key in it which satisfies the first condition of 2NF.

**Third Normal Form(3NF):**

The table is already in 3NF because the table is in 1NF, 2NF & has no transitive functional dependency among the attributes.

**Fourth Normal Form(4NF):**

The table is already in 4NF because it has no multivalued dependency in it.

**Payment Table:**

Attributes:

Payment\_ ID, Payment\_ date, Payment\_ Amount, Payment\_ type, Status

Functional Dependencies:

Payment \_ID 🡪 Payment \_date

Payment \_ID 🡪 Payment \_Amount

Payment \_ID 🡪 Payment \_type

Payment \_ID 🡪 Status

Primary Key: Payment \_ID

**First Normal Form(1NF):**

The table is already in 1NF because the table has no non atomic attributes in it.

**Second Normal Form(2NF):**

The table is already in 2NF because the table is in 1NF & has only one primary key in it which satisfies the first condition of 2NF.

**Third Normal Form(3NF):**

The table is already in 3NF because the table is in 1NF, 2NF & has no transitive functional dependency among the attributes.

**Fourth Normal Form(4NF):**

The table is already in 4NF because it has no multivalued dependency in it.

**Customer Table:**

Attributes:

Customer\_ID, Cfirst\_ name, Cmiddle\_ name, Clast\_name, Location, city, state.

Functional Dependencies:

Customer \_ID 🡪 Cfirst\_ name

Customer \_ID 🡪 Cmiddle\_ name

Customer \_ID 🡪 Clast\_name

Customer \_ID 🡪 Location

Customer \_ID 🡪 city

Customer \_ID 🡪 State

Primary Key: Customer \_ID

**First Normal Form(1NF):**

The table is already in 1NF because the table has no non atomic attributes in it.

**Second Normal Form(2NF):**

The table is already in 2NF because the table is in 1NF & has only one primary key in it which satisfies the first condition of 2NF.

**Third Normal Form(3NF):**

The table is already in 3NF because the table is in 1NF, 2NF & has no transitive functional dependency among the attributes.

**Fourth Normal Form(4NF):**

The table is already in 4NF because it has no multivalued dependency in it.

**Tax Table:**

Attributes:

Tax\_ ID, Tax\_ Amount, Tax\_ Type

Functional Dependencies:

Tax \_ID 🡪 Tax \_Amount

Tax \_ID 🡪 Tax \_Type

Primary Key: Payment \_ID

**First Normal Form(1NF):**

The table is already in 1NF because the table has no non atomic attributes in it.

**Second Normal Form(2NF):**

The table is already in 2NF because the table is in 1NF & has only one primary key in it which satisfies the first condition of 2NF.

**Third Normal Form(3NF):**

The table is already in 3NF because the table is in 1NF, 2NF & has no transitive functional dependency among the attributes.

**Fourth Normal Form(4NF):**

The table is already in 4NF because it has no multivalued dependency in it.

**Multivalued Tables:**

**First Normal Form(1NF):**

All the multivalued tables are already in 1NF because all the tables has no non atomic attributes in them.

**Second Normal Form(2NF):**

All the multivalued tables are already in 2NF because all the tables are in 1NF & has no partial dependencies in them which satisfies the first condition of 2NF.

**Third Normal Form(3NF):**

All the tables are already in 3NF because the tables are in 1NF, 2NF & has no transitive functional dependency among the attributes in the multivalued tables.

**Fourth Normal Form(4NF):**

All the tables are already in 4NF because the tables has no multivalued dependency in them.

# SQL Code for the project:

create database project;

use project;

-----------------------------#CREATINGTABLES:-------------------------------------

#1.MANAGER TABLE....

create table manager (

Manager\_id int primary key,

Mfirst\_name varchar(20),

Mmiddle\_name varchar(20),

Mlast\_name varchar(20),

Manager\_Qualification varchar(20),

Manager\_DOB date,

Gender varchar(15),

Location varchar(20),

city varchar(20),

State varchar(20)

);

create table Billing\_details (

Billing\_id int primary key,

Billing\_date date,

Billing\_amount int,

Manager\_id int references manager(Manager\_id)

);

create table manager\_contact (

Manager\_id int references manager(Manager\_id),

Manager\_contact int

);

create table supplyer (

Supplyer\_id int primary key,

Supplyer\_store\_name varchar(20),

Supplyer\_first\_name varchar(20),

Supplyer\_middle\_name varchar(20),

Supplyer\_last\_name varchar(20),

Supplyer\_location varchar(20),

city varchar(20),

State varchar(20),

Supplyer\_Gender varchar(10)

);

create table Supplyer\_contact (

Supplyer\_id int references supplyer(Supplyer\_id),

Supplyer\_contact int

);

create table manager\_supplyer (

Manager\_id int references manager(Manager\_id),

Supplyer\_id int references supplyer(Supplyer\_id)

);

create table products (

Product\_id int primary key,

Product\_name varchar(20),

Product\_type varchar(20),

Product\_cost varchar(20),

Product\_discription varchar(20),

Supplyer\_id int references supplyer(Supplyer\_id)

);

create table store (

Store\_id int primary key,

Store\_name varchar(20),

Store\_street\_name varchar(20),

location varchar(20),

city varchar(20),

state varchar(20)

);

create table store\_contact (

Store\_id int references store(Store\_id),

Store\_contact\_nummber int

);

create table Supplyer\_store (

Supplyer\_id int references supplyer(Supplyer\_id),

Store\_id int references store(Store\_id)

);

create table workers

(

Workers\_id int primary key,

Workers\_first\_name varchar(20),

Workers\_middle\_name varchar(20),

Workers\_last\_name varchar(20),

Store\_id int references store(Store\_id),

Workers\_type varchar(20),

Workers\_salary int,

Workers\_discription varchar(20)

);

create table workers\_contact

(

Workers\_id int references workers(Workers\_id),

Workers\_contact int

);

create table customer

(

Customer\_id int primary key,

Customer\_first\_name varchar(20),

Customer\_middle\_name varchar(20),

Customer\_last\_name varchar(20),

Customer\_street\_name varchar(20),

location varchar(20),

city varchar(20),

state varchar(20),

Customer\_gender varchar(10)

);

create table sales

(

Sales\_id int primary key,

Product\_id int references products(Product\_id),

cost int ,

Sales\_discription varchar(20),

Store\_id int references store(Store\_id),

Customer\_id int references customer(Customer\_id)

);

create table tax

(

Tax\_id int primary key,

Tax\_amount int ,

Tax\_type varchar(20),

Sales\_id int references sales(Sales\_id)

);

create table customer\_email

(

Customer\_id int references customer(Customer\_id),

email\_id varchar(20)

);

create table customer\_contact

(

Customer\_id int references customer(Customer\_id),

Customer\_contact\_number int

);

create table payment

(

Payment\_id int primary key ,

Customer\_id int references customer(Customer\_id) ,

Payment\_date date ,

Payment\_amount int ,

Payment\_type varchar(20) ,

Payment\_status varchar(20)

);

insert into manager values(1,'kadiyala','jaya','prakash','PHD','2003-06-14','male','gannavaram','vijayawada','AP');

select\*from manager;



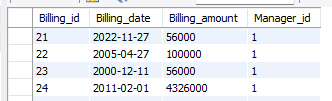
insert into Billing\_details values(11,'2022-11-27','56000',2);

insert into Billing\_details values(16,'2005-04-27','100000',1);

insert into Billing\_details values(120,'2000-12-11','56000',4);

insert into Billing\_details values(17,'2011-02-01','4326000',3);

select\*from Billing\_details;



insert into manager\_contact values(1,'90146447');

select\*from manager\_contact;



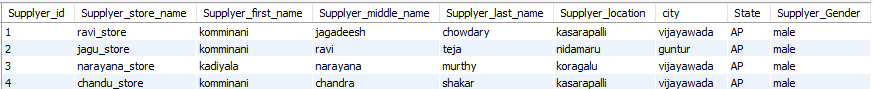
insert into supplyer values(1,'ravi\_store','komminani','jagadeesh','chowdary','kasarapalli','vijayawada','AP','male');

insert into supplyer values(2,'jagu\_store','komminani','ravi','teja','nidamaru','guntur','AP','male');

insert into supplyer values(3,'narayana\_store','kadiyala','narayana','murthy','koragalu','vijayawada','AP','male');

insert into supplyer values(4,'chandu\_store','komminani','chandra','shakar','kasarapalli','vijayawada','AP','male');

select\*from supplyer;



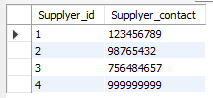
insert into Supplyer\_contact values(1,'123456789');

insert into Supplyer\_contact values(2,'098765432');

insert into Supplyer\_contact values(3,'756484657');

insert into Supplyer\_contact values(4,'999999999');

select\*from Supplyer\_contact;



insert into manager\_supplyer values(1,1);

select \*from manager\_supplyer;

insert into manager\_supplyer values(1,4);

insert into manager\_supplyer values(4,3);

insert into manager\_supplyer values(22,7);

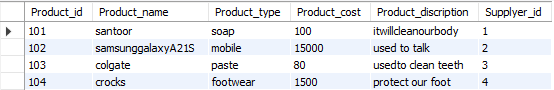
insert into products values(101,'santoor','soap','100','itwillcleanourbody',1);

insert into products values(102,'samsunggalaxyA21S','mobile','15000','used to talk',2);

insert into products values(103,'colgate','paste','80','usedto clean teeth',3);

insert into products values(104,'crocks','footwear','1500','protect our foot',4);

select\*from products;



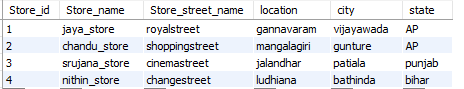
insert into store values(1,'jaya\_store','royalstreet','gannavaram','vijayawada','AP');

insert into store values(2,'chandu\_store','shoppingstreet','mangalagiri','gunture','AP');

insert into store values(3,'srujana\_store','cinemastreet','jalandhar','patiala','punjab');

insert into store values(4,'nithin\_store','changestreet','ludhiana','bathinda','bihar');

select \* from store;



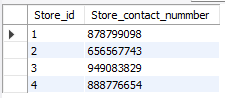
insert into store\_contact values(1,'878799098');

insert into store\_contact values(2,'656567743');

insert into store\_contact values(3,'949083829');

insert into store\_contact values(4,'888776654');

select \* from store\_contact;



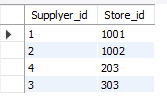
insert into Supplyer\_store values(1,1001);

insert into Supplyer\_store values(2,1002);

insert into Supplyer\_store values(4,203);

insert into Supplyer\_store values (3,303);

select\*from Supplyer\_store;



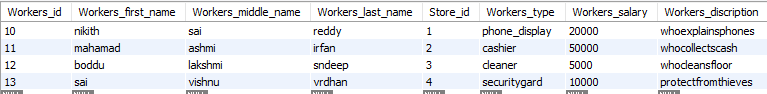
insert into workers values(10,'nikith','sai','reddy',1,'phone\_display','20000','whoexplainsphones');

insert into workers values(11,'mahamad','ashmi','irfan',2,'cashier','50000','whocollectscash');

insert into workers values(12,'boddu','lakshmi','sndeep',3,'cleaner','5000','whocleansfloor');

insert into workers values(13,'sai','vishnu','vrdhan',4,'securitygard','10000','protectfromthieves');

select \* from workers;



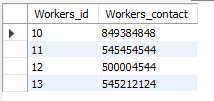
insert into workers\_contact values(10,'849384848');

insert into workers\_contact values(11,'545454544');

insert into workers\_contact values(12,'500004544');

insert into workers\_contact values(13,'545212124');

select \* from workers\_contact;



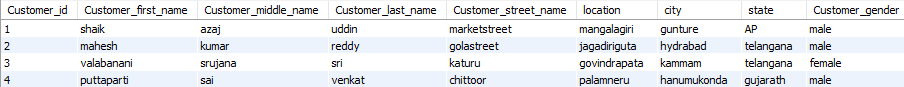
insert into customer values(1,'shaik','azaj','uddin','marketstreet','mangalagiri','gunture','AP','male');

insert into customer values(2,'mahesh','kumar','reddy','golastreet','jagadiriguta','hydrabad','telangana','male');

insert into customer values(3,'valabanani','srujana','sri','katuru','govindrapata','kammam','telangana','female');

insert into customer values(4,'puttaparti','sai','venkat','chittoor','palamneru','hanumukonda','gujarath','male');

select \* from customer;



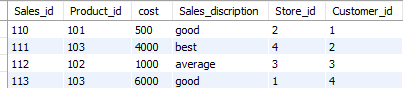
insert into sales values(110,101,'500','good',2,1);

insert into sales values(111,103,'4000','best',4,2);

insert into sales values(112,102,'1000','average',3,3);

insert into sales values(113,103,'6000','good',1,4);

select \* from sales;



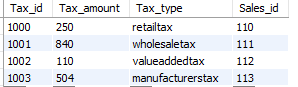
insert into tax values(1000,'250','retailtax',110);

insert into tax values(1001,'840','wholesaletax',111);

insert into tax values(1002,'110','valueaddedtax',112);

insert into tax values(1003,'504','manufacturerstax',113);

select\*from tax;



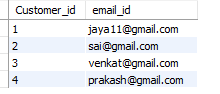
insert into customer\_email values(1,'jaya11@gmail.com');

insert into customer\_email values(2,'sai@gmail.com');

insert into customer\_email values(3,'venkat@gmail.com');

insert into customer\_email values(4,'prakash@gmail.com');

select\* from customer\_email;



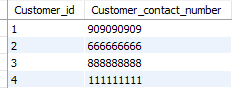
insert into customer\_contact values(1,'909090909');

insert into customer\_contact values(2,'666666666');

insert into customer\_contact values(3,'888888888');

insert into customer\_contact values(4,'111111111');

select\*from customer\_contact ;



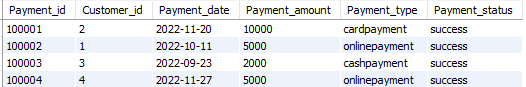
insert into payment values(100001,2,'2022-11-20','10000','cardpayment','success');

insert into payment values(100002,1,'2022-10-11','5000','onlinepayment','success');

insert into payment values(100003,3,'2022-09-23','2000','cashpayment','success');

insert into payment values(100004,4,'2022-11-27','5000','onlinepayment','success');

select \* from payment;



# SQL QUERIES :

-------------------------------------------------------------------------#QUERIES-----------------------------------------------------

select all Product\_name

from products

where Product\_type ='soap';



select all Product\_name

from products

where Product\_type ='mobile';

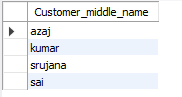


select customer.Customer\_middle\_name

from customer,sales

where

sales.Customer\_id=customer.Customer\_id;



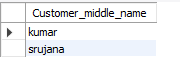
select customer.Customer\_middle\_name

from customer,sales

where

sales.Customer\_id=customer.Customer\_id and

State='telangana';



select Product\_name, max(Product\_cost)

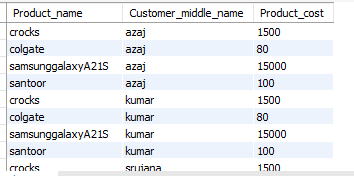
from products;

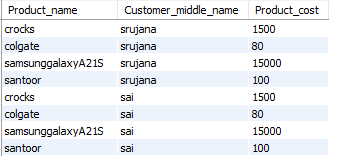
select Product\_name, min(Product\_cost)

from products;

select Product\_name,Customer\_middle\_name,Product\_cost

from products,customer;



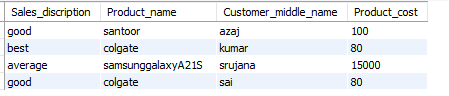


select Sales\_discription,Product\_name,Customer\_middle\_name,Product\_cost

from sales,products,customer

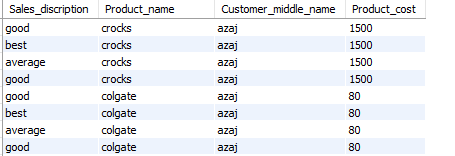
where sales.Product\_id=products.Product\_id and

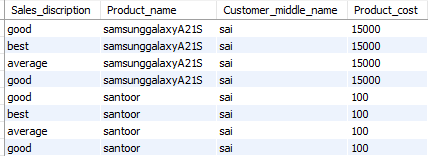
sales.Customer\_id=customer.Customer\_id;



select Sales\_discription,Product\_name,Customer\_middle\_name,Product\_cost

from sales,products,customer;





# Concluding Remarks

A description of the background and context of the project and its relation to work already done in the area. Made statement of the aims and objectives of the project. The description of Purpose, Scope, and applicability. I have defined the problem on which I had worked in the project. I have described the requirement Specifications of the system and the actions that can be done on these things. I had understood the problem domain and produce a model of the system, which describes operations that can be performed on the system. I have included the features and operations in detail, including screen layouts.

# Future work

Today India is flooded with various super markets and stores for customers to buy In the last couple of years the growth of inventory store management system has been phenomenal as more store managers and customers started discovering the benefits of using this platform. There is enough scope for inventory store management system in the future if they understand the Indian shoppers psyche and cater to their needs. So, I would like to develop a webpage for this project in the coming days.

# References :

1. <https://github.com/topics/inventory-management-system>
2. <https://www.geeksforgeeks.org/dbms/>
3. <https://www.unleashedsoftware.com/inventory-management-guide/inventory-management-systems>
4. <https://youtu.be/4eSjYd68AI0>
5. <https://www.camcode.com/blog/what-is-an-inventory-management-system/>