Team member:

Cheng-Yuan Wang

Lingyun Zhao

Inventory Management System

• Problem Statement: Our idea is to create an inventory management system in order to help small business’s owner to stay on track with their current inventory of the product. Therefore, we need database system to control the backend information

2. Our database will consist of two portions: data that represent the employees information and data that represents the product information.

The ‘’Store’’ entity is a store, with attributes store\_id, store\_phone, store\_address, store\_manager and inventory. The ‘’Store’’ entity contains two relationships, Sells and Work For. The relationship ‘’Sell’’ will connect “store” entity to “Product” entity and ‘’Work For” will connect “employee” entity.

‘’Product” entity is an entity of item information, the attributes are item ID and item name and it has a weak entity which is “Customer”. In “Customer” entity contains three attributes: name, phone and address, this entity is connected to ‘’Product’’ with a ‘’Buy” relationship.

“Employee” entity is an entity containing attributes of employee’s information which is, Employee ID, Name, SSN, Phone, Address. “Employee” entity also connects to 3 other weak entities, Manager, Accounting and Worker, they all have the same attribute which is Employee ID. There is an “emergency contact” entity under “employee” which contains 2 attributes: phone and name.

**product**

Item\_ID

Item\_name

**Employee**

Employee\_ID

Name

SSN

Phone

Address

**Emergency Contact**

Employee\_ID

Name

Phone

**Manager**

employee\_ID

**Accounting**

**Worker**

Employee\_ID

Employee\_Id

**Customer**

Name

Phone

address

**Buy**

C\_name

Item\_Id

amount

**Sells**

SID

Item\_id

amount

**store**

Store\_ID

Store\_phone

Store\_address

Store\_manager

inventory

**Work\_For**

E\_id

S\_ID

**Table**

**Attribute**

**Type**

**Constraint**

Product

Item\_id

CHAR(80)

Primary Key

Product

Item\_name

CHAR(80)

NOT NULL

Employee

Employee\_id

CHAR(80)

Primary key

Employee

Name

CHAR(80)

NOT NULL

Employee

SSN

CHAR(80)

NOT NULL

Employee

Phone

CHAR(80)

Employee

address

CHAR(80)

Emergency Contact

Employee\_id

CHAR(80)

Foreign key

Emergency Contact

phone

CHAR(80)

Emergency Contact

address

CHAR(80)

Manager

Employee\_id

CHAR(80)

Foreign key

Worker

Employee\_id

CHAR(80)

Foreign key

Accounting

Employee\_id

CHAR(80)

Foreign key

customer

Name

CHAR(80)

NOT NULL

customer

Phone

CHAR(80)

store

Store\_id

CHAR(80)

Primary key

store

Store\_address

CHAR(80)

store

Store\_phone

CHAR(80)

store

Store\_manager

CHAR(80)

store

inventory

CHAR(80)

NOT NULL

buy

C\_name

CHAR(80)

Foreign Key

Buy

Item\_id

CHAR(80)

Foreign Key

Buy

Amount

INTEGER

NOT NULL

Sell

SID

CHAR(80)

Foreign Key

Sell

Item\_id

CHAR(80)

Foreign Key

Sell

Amount

INTEGER

NOT NULL

Work\_for

E\_id

CHAR(80)

Foreign Key

Work\_for

S\_id

CHAR(80)

Foreign Key

4.

Function 1: edit\_product\_inventory

// This function is to edit the inventory of the product.

// It accesses ‘’store’’ table and ‘’product’’ table

Input: Store ID, the ID of selected product

Steps:

• Check the store information in ‘’store’’ table base on store ID

• Then find the inventory of product by item ID

• Modify the amount of product and save it to the ‘’product’’ table

• display the product

Function 2: print\_product\_inventory

// This function is to print the product inventory

// It accesses ‘’store’’ table and ‘’product’’ table

Input: Store ID, the ID of selected product

Steps:

• Check the store product base on the store ID

• Print the product table

Function 3: add\_new\_employee

// This function is to add new employee into ‘’employee’’ table

// It accesses ‘’store’’ table and ‘’employee’’ table

Input: Store ID, the employee ID

Steps:

• Find the store base on the Store ID

• Insert a new itinerary under Employee table

• Display the itinerary

5.

**Installation instruction**

• Intended operating system: windows

• Steps:

• Install pymysql for connecting database server and python code

• Install mysql to build database server.

• In mysql workbench, create a new connection, import the SQL file into the connction.

• Change the host, user, password information in python code to match the database server.

• Run python code to gain access to the database

**User manual:**

• Product:

• This category includes the information of all products in records.

• It is a list of all products with information about the name, store that sells them, and inventory at the store.

• It allows you to add product information as well as search a certain product

• To add, simple type in the blank the information of the product, then click edit inventory. If there exists the same product id, it will pop up a warning that there’s duplicated value. Otherwise, it will add a new product in the database.

•

• When users click the search button, it will list all the data stored in the database.

• Inventory

• this category allows users to check all inventory at each store.

• It has a filter function which allows users to check all inventory at certain store.

• To use the filter function, click the down arrow and select the store you want. The result will show all the products at that store.

•

• Employee:

• This category contains the information of all employees working in the company.

• It is a list that shows the information, including name, ssn, phone number, address and position, of all employees in that company.

• It provides search and add function.

•

• To search, click the search button. When users click the search button, it will list all the data stored in the database.

• To add, like for product, simple type in the blank the information of the employee, then click add new If there exists the same employee id, it will pop up a warning. Otherwise, it will add a new employee in the database.