[Database design and Development]

[LAYAN SAEED AQEL] | [21110405]

[Final technical documentation]

ENG: **Lina Hammad**

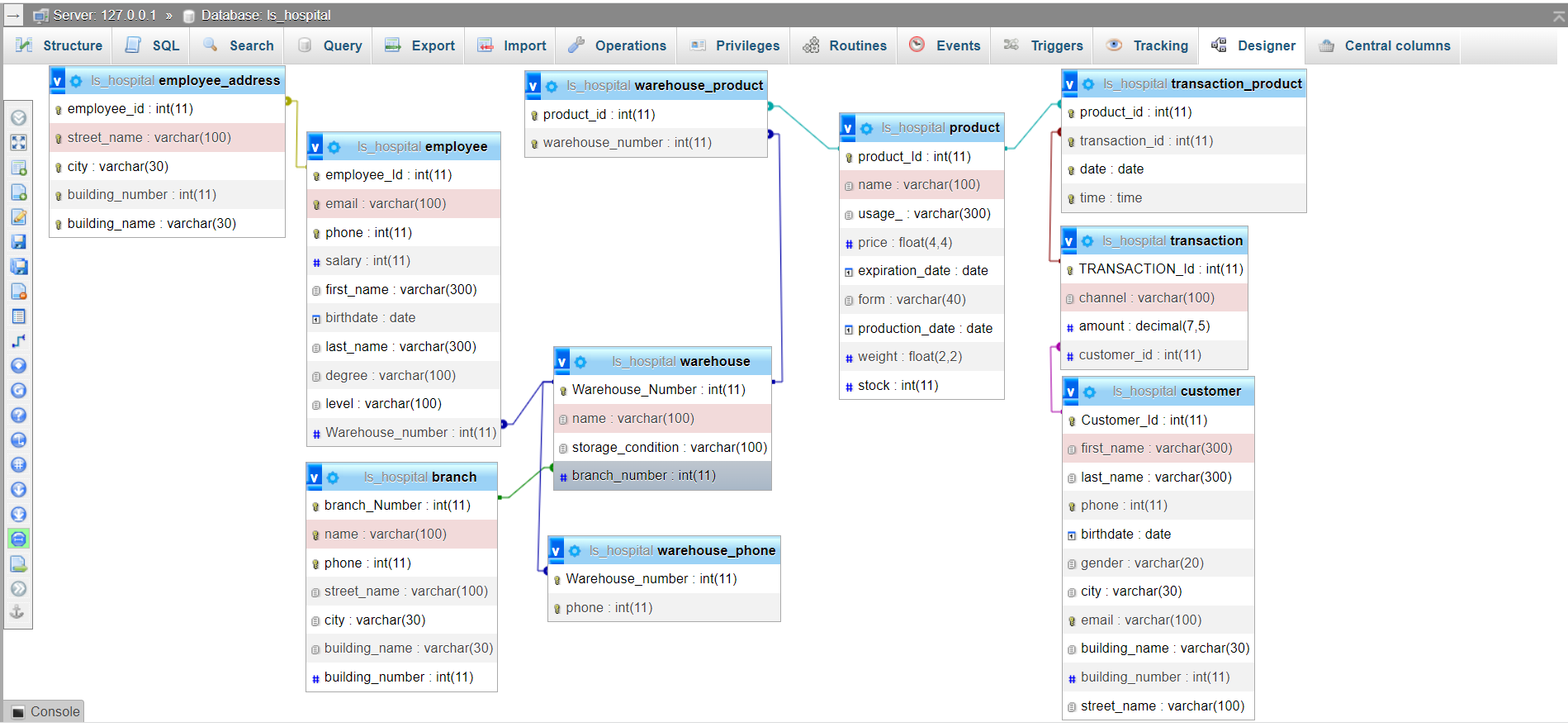
[2023]

# Technical Documentation

# Introduction

This is a technical document which is a written report about the way of creating database for the warehouses of LS hospital and its purpose. It is meant to be read by someone who has knowledge about database or expert in the field. In my technical documentation, I wrote about the different aspects of my database such as physical schema, description of the tables, views, procedure. Also, in the third part, I created users and explained the given privileges for each user. Then, Drawing DFD and flow chart for each user. I explained every page in the GUI and tested the privileges of every user on each page. After that, I tested the validation of the data type and constraints in my database, and I tested the validation of the output, security, and GUI. Also, I explained how I can make pickup and recovery from PHPMYADMIN, then how I can make maintenance for database in general. Finally, I evaluated the previously mentioned testing processes and my database. Then I offer some suggestions for improving the database and the business and how database should deal with these improvements in business.

# Physical Schema



# Database Development

## Database Overview

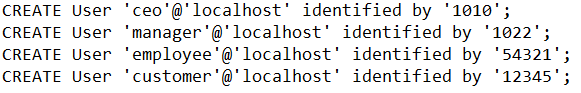
|  |  |  |
| --- | --- | --- |
| **Table** | **Name** | **Description** |
|  | Branch | This table stores information about the branches of LS\_Hospital such as number, name, phone , address which includes (city , Streat name, building name, building number). |
|  | Warehouse | This table stores information about the warehouse in the branches in the hospital such as number, name, storage condition, the number of the branch which the warehouse exists in |
|  | Warehouse\_phone | This table stores the phone numbers of warehouses because each warehouse may have one or more phone number, so this table stores the numbers of warehouses and the phones. The benefit of having this separate table is to reduce redundancy of data. |
|  | Employee | This table stores information about employees which work in the warehouses for LS\_Hospital in all branches. Any employee should have these information such as ID of employee, email, phone, salary, name(first and last name), birthdate, degree, level , and the number of warehouse which the employee works in . |
|  | Employee\_address | This table stores information about employees’ addresses because employees may have one or more address, so this table stores the ID of employees and their different addresses. The benefit of having this separate table is to reduce redundancy of data. |
|  | Transaction | This table stores the information about the transaction made by the customer when he buys the medicine such as the ID of transaction, amount, channel, and Id of customer who does the transaction. |
|  | Product | This table stores the information about any product in the warehouse in LS\_Hospital such as the ID for the product, name, usage, price, form, expiration and production date , wight ,and stock . |
|  | Transaction\_product | This table stores the ID of products which are found in any transaction and the ID of transactions which include any medicine. The benefit of having this separate table is to reduce redundancy of data. |
|  | Warehouse\_product | This table stores the ID of products which is found in any warehouse and the ID of warehouses which may include any product. The benefit of having this separate table is to reduce redundancy of data. |
|  | Customer | This table stores information about customers when they buy the medicine such as the ID of customers, name (first and last name), phone, birthdate, address which includes (city , Streat name, building name, building number), email, and gender. |

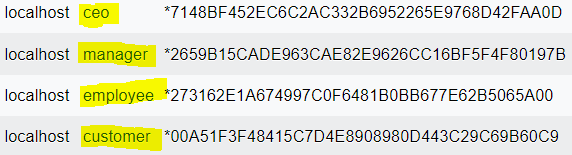
|  |  |  |
| --- | --- | --- |
| **View** | **Name** | **Description** |
|  | CEO\_employee | This view contains select statement which allows the CEO of LS\_hospital to see all the information of employees in any warehouse. |
|  | CEO\_warehouse | This view contains select statement which allows the CEO of LS\_hospital to see all the information of warehouses in all branches. |
|  | product\_transaction02 | This view contains select statement which allows the CEO of LS\_hospital to see the amount of each transaction of all the products in all warehouses. |
|  | customer\_select | This view contains select statement which allows the customer to see the information about the product in the warehouses. |
|  | employee\_select | This view contains select statement which allows the employees to see their information. |
|  | stock\_product\_ceo | This view contains select statement which allows the CEO of LS\_hospital to see the stock of all products. |

|  |  |  |
| --- | --- | --- |
| **Procedure** | **Name** | **Description** |
|  | product\_transaction | This procedure contains select statement which allows the manager of any warehouse to see the amount of each transaction according to the ID of the product in it. |
|  | warehouse\_stock\_product | This procedure contains select statement which allows the manager of any warehouse to see the stock of products according to the ID of product and number of the warehouse. |
|  | num\_warehouse\_product | This procedure contains select statement which allows the manager of any warehouse to see the number of warehouses which has this product according to the ID of the product. |
|  | num\_transaction\_product | This procedure contains select statement which allows the manager of any warehouse to see the number of transactions which has this product according to the ID of the product. |
|  | Manager\_update | This procedure contains update statement which allows the manager of any warehouse to update important information of employees such as salary, degree, level. |
|  | manager\_insert | This procedure contains insert statement which allows the manager of any warehouse to insert all the information of employees. |
|  | manager\_delete | This procedure contains delete statement which allows the manager of any warehouse to delete all the information of employees according to the ID of employee and number of the warehouse. |
|  | manager\_select | This procedure contains select statement which allows the manager of any warehouse to see all the information of employees according to the number of the warehouse. |
|  | ADD\_customer | This procedure contains insert statement which allows customers to insert their information |
|  | UPDATE\_customer | This procedure contains UPDATE statement which allows customers to UPDATE their information. |
|  | employee\_insert | This procedure contains insert statement which allows employees to insert their information. |
|  | employee\_product | This procedure contains select statement which allows employees to see all the information about the products according to the ID of product. |
|  | employee\_update | This procedure contains UPDATE statement which allows employees to UPDATE their information. |
|  | INSERT\_transaction\_customer | This procedure contains insert statement which allows customers to do transactions. |
|  | manager\_transaction | This procedure contains select statement which allows manager of any warehouse to see the number of transactions in his warehouse. |
|  | Employee\_insert\_product | This procedure contains insert statement which allows employees to insert all the information about the products. |

## Security

First, I created four users.



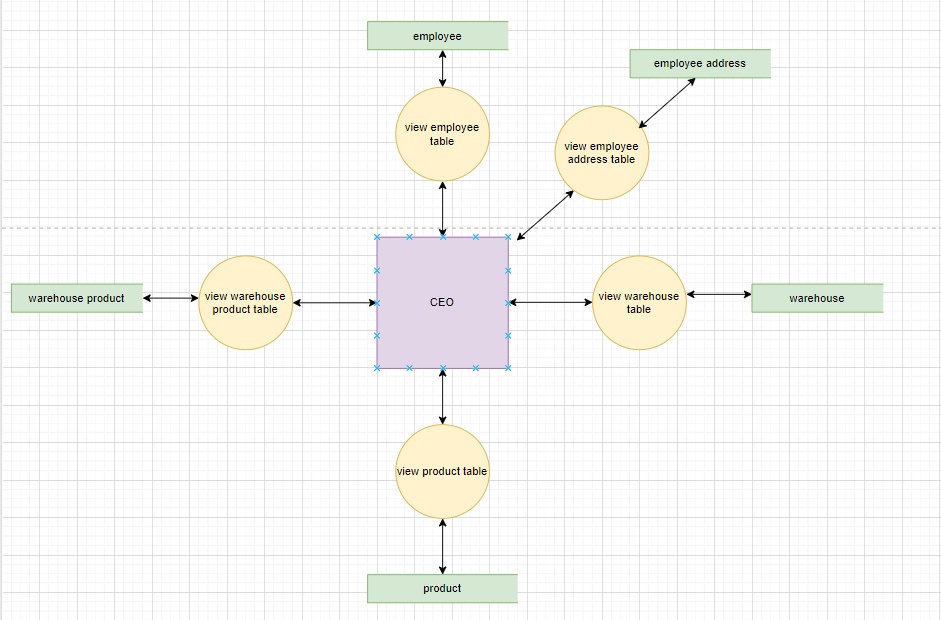


|  |  |  |  |
| --- | --- | --- | --- |
| **User name** | **Privilege Command** | **Description** | **Screenshot** |
| CEO | GRANT SELECT on ls\_hospital2.ceo\_employee to 'ceo'@'localhost'; | This privilege allows the CEO of LS\_hospital to see all the information of employees in any warehouse |  |
| GRANT SELECT on ls\_hospital2.  product\_transaction02 to 'ceo'@'localhost'; | This privilege allows the CEO of LS\_hospital to see the amount of each transaction of all the products in all warehouses. |  |
| GRANT SELECT on ls\_hospital2.  stock\_product\_ceo to 'ceo'@'localhost'; | This privilege allows the CEO of LS\_hospital to see the stock of all products |  |
| GRANT SELECT on ls\_hospital2.  ceo\_warehouse to 'ceo'@'localhost'; | This privilege allows the CEO of LS\_hospital to see all the information of warehouses in all branches. |  |
| manager | GRANT EXECUTE on PROCEDURE ls\_hospital2.  product\_transaction to 'manager'@'localhost'; | This privilege allows the manager of any warehouse to see the amount of each transaction according to the ID of the product in it. |  |
| GRANT EXECUTE on PROCEDURE ls\_hospital2.  num\_warehouse\_product to 'manager'@'localhost'; | This privilege allows the manager of any warehouse to see the number of warehouses which has this product according to the ID of the product. |  |
| GRANT EXECUTE on PROCEDURE ls\_hospital2.  num\_transaction\_product to 'manager'@'localhost'; | This privilege allows the manager of any warehouse to see the number of transactions which has this product according to the ID of the product. |  |
| GRANT EXECUTE on PROCEDURE ls\_hospital2.  manager\_update to 'manager'@'localhost'; | This privilege allows the manager of any warehouse to update important information of employees such as salary, degree, level. |  |
| GRANT EXECUTE on PROCEDURE ls\_hospital2.  manager\_insert to 'manager'@'localhost'; | This privilege allows the manager of any warehouse to insert all the information of employees. |  |
| GRANT EXECUTE on PROCEDURE ls\_hospital2.  manager\_delete to 'manager'@'localhost'; | This privilege allows the manager of any warehouse to delete all the information of employees according to the ID of employee and number of the warehouse |  |
| GRANT EXECUTE on PROCEDURE ls\_hospital2.manager\_select to 'manager'@'localhost'; | This privilege allows the manager of any warehouse to see all the information of employees according to the number of the warehouse. |  |
| GRANT EXECUTE on PROCEDURE ls\_hospital2.  manager\_transaction to 'manager'@'localhost'; | This privilege allows manager of any warehouse to see the number of transactions in his warehouse. |  |
| GRANT EXECUTE on PROCEDURE ls\_hospital2.  warehouse\_stock\_product to 'manager'@'localhost'; | This privilege allows the manager of any warehouse to see the stock of products according to the ID of product and number of the warehouse. |  |
| employee | GRANT SELECT on ls\_hospital2.employee\_select to 'employee'@'localhost'; | This privilege allows the employees to see their information. |  |
| GRANT EXECUTE on PROCEDURE ls\_hospital2.  employee\_insert to 'employee'@'localhost'; | This privilege allows employees to insert their information. |  |
| GRANT EXECUTE on PROCEDURE ls\_hospital2.  employee\_product to 'employee'@'localhost'; | This privilege  allows employees to see all the information about the products according to the ID of product. |  |
| GRANT EXECUTE on PROCEDURE ls\_hospital2.  employee\_update to 'employee'@'localhost'; | This privilege  allows employees to UPDATE their information. |  |
| GRANT EXECUTE on PROCEDURE ls\_hospital2.  employee\_insert\_product to 'employee'@'localhost'; | This privilege allows employees to insert all the information about the products. |  |
| customer | GRANT SELECT on ls\_hospital2.customer\_select to 'customer'@'localhost'; | This privilege allows the customer to see the information about the product in the warehouses |  |
| GRANT EXECUTE on PROCEDURE ls\_hospital2.  add\_customer to 'customer'@'localhost' | This privilege allows the customers to insert their information |  |
| GRANT EXECUTE on PROCEDURE ls\_hospital2.  update\_customer to 'customer'@'localhost' | This privilege allows the customers to UPDATE their information |  |
| GRANT EXECUTE on PROCEDURE ls\_hospital2.  insert\_transaction\_customer to 'customer'@'localhost' | This privilege allows the customers to do transactions. |  |

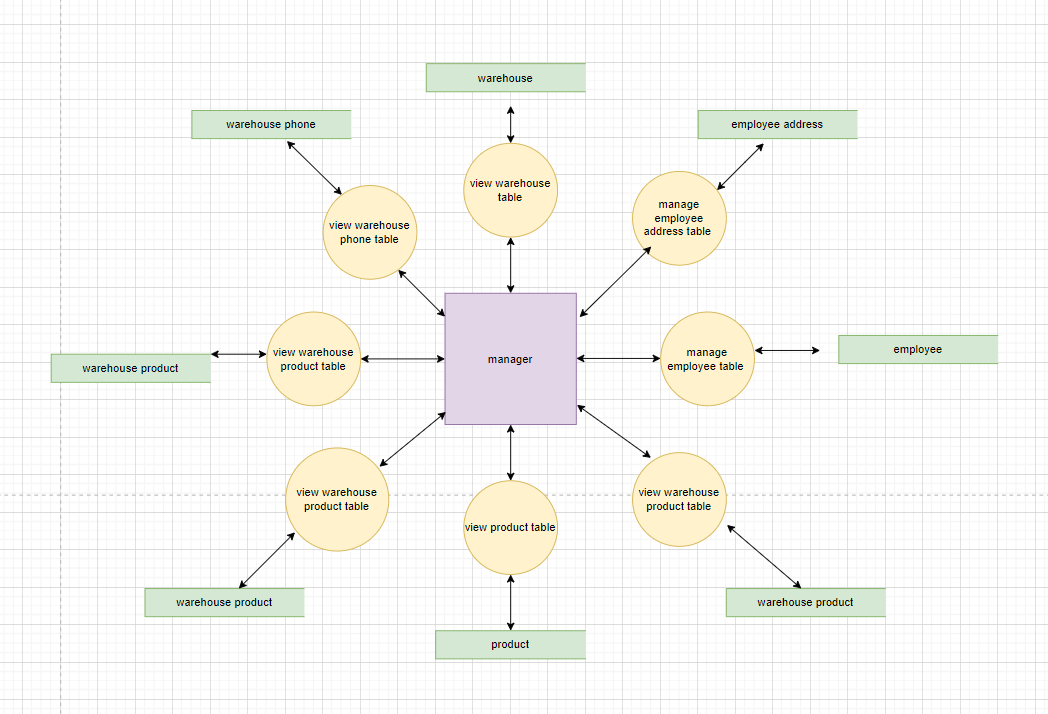
## User Interface

### Flowchart and Data Movement Diagrams

* DFD for CEO



* DFD for manager

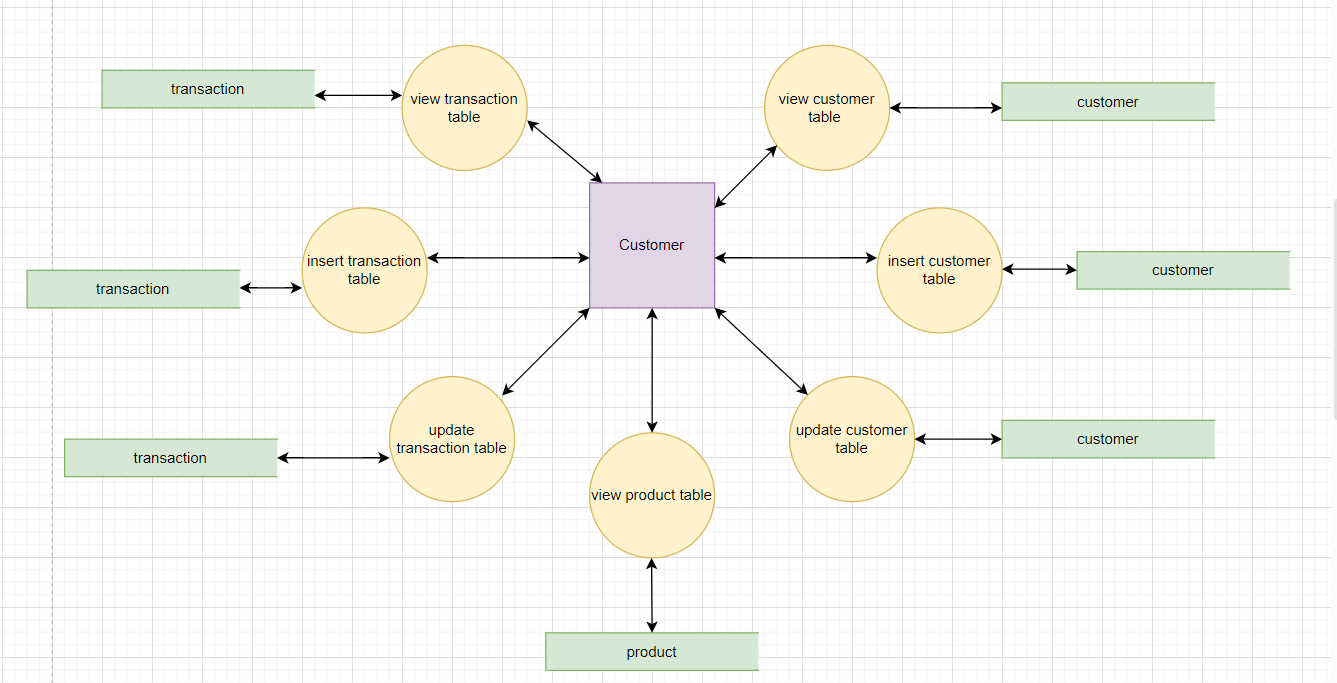


* DFD for employee

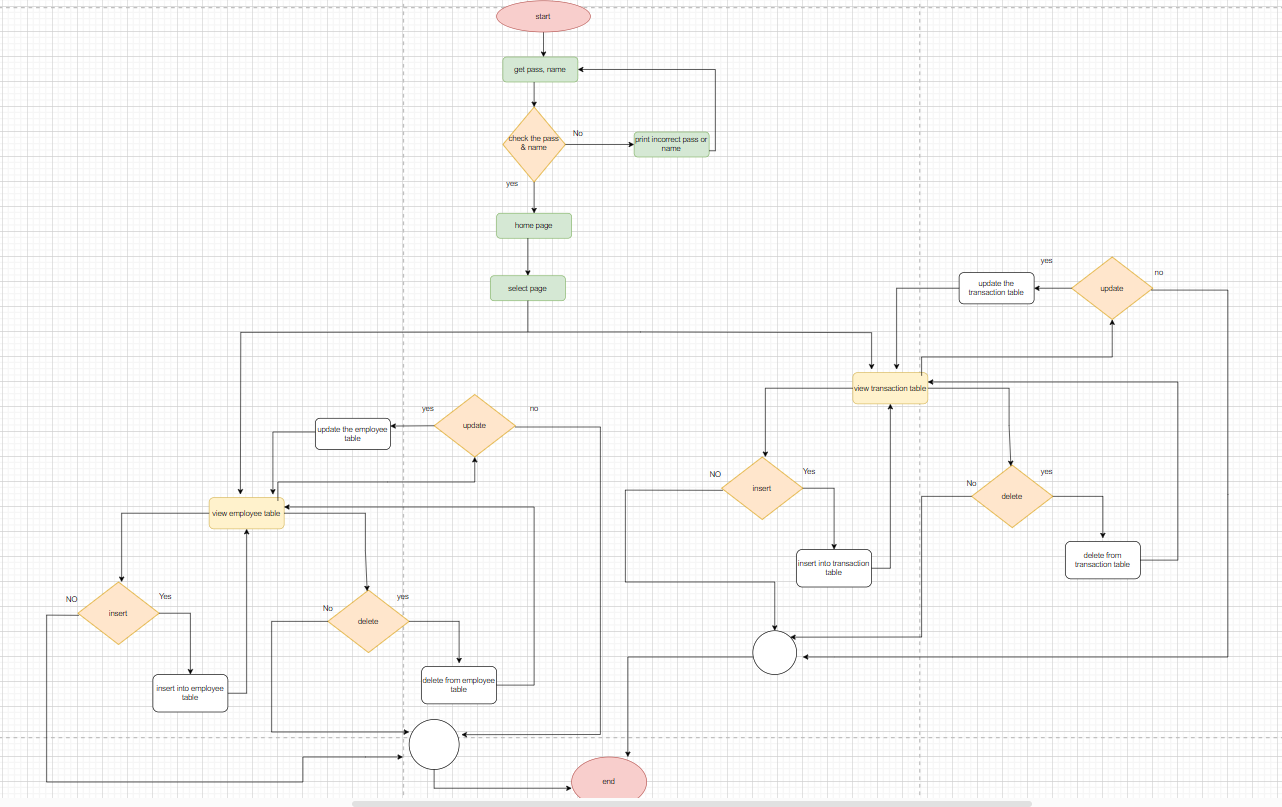
A picture containing text, screenshot, diagram, line

Description automatically generated

* DFD for customer



* Flowchart for employee in case enter the employee table or the transaction table.



### Interfaces Development

|  |  |  |  |
| --- | --- | --- | --- |
| **Page ID** | **Title** | **Description** | **Screenshot** |
|  | Home Page | This page is for presenting all the pages. By them the employee can move from one page to another. |  |
|  | Employee  Page | This page consists of the information of the employee where the employee can insert, update, view the information (the data in this table). he/she can’t insert and update the salary, degree, level, ID because it’s the responsibility of the manager. |  |
|  | Employee Address page | This page consists of the information which is the addresses of the employee where the employee can insert, update, view the information (the data in this table). But the employee can’t update, insert his/her ID |  |
|  | Product page | This page consists of all information about the product where the employee can view, insert, update the information of the product |  |
|  | Transaction  Page | This page consists of the information about the transaction such as the id, channel, amount, the id of customer who does this transaction. In this page the employee can just view the information but not insert, and update the transaction |  |
|  | Transaction Product | This page consists of information about the time and date of the transaction and the products in each transaction. in this page the employee can just view the information but not edit or insert information |  |

# Maintenance (M2)

## **Database recovery & backups**

* **The benefits of the recovery and the backups:**

They are very important in any change or system updates to ensure its reliability, safety, and continuity. They can help developers to reduce the effects of problems on your system and prevent loss of data. They are also a crucial part of any disaster recovery plan. WHICH HELPS to deal with attacks or other dangerous situations. And so, you can recover your data, safety, and continuity of your business. Moreover, they are very important to reduce risk and ensure compliance with regulations. It is very important to regularly check your backup to make sure that your system is working in the right way.

* **The reasons of the failure:**

**As I said before backup and recovery is very important because data may damage for different reasons:**

The first reason of damage is the problems related to software such as viruses, bugs and coding mistakes which may lead d to data loss or corruption. Another reason is related to human behaviour and errors or lack of training. Natural disasters and sipper attacks are other dangers for data or systems. Network issues and connectivity may break connection and reduce quality of services. Any system should be regularly maintained and optimized to prevent any failure in the system or data.

* **Now, I will talk about how to make backup and recovery from PHPMyAdmin:**

**Export 🡪 backup**

If we want to make backup for our data, we should go to Export button in PHPMyAdmin.

I need back up to make extra copies of my data base to prevented from loss and corruption. And so that I can go back to it whenever I need to.



First, I open PHPMyAdmin then choose the data base which I want to make backup for then press on the export button to install the data base on my device.

**Import 🡪 recover.**

if we want to make recovery for our data, we should go to Import button in PHPMyAdmin.

I need to make recovery of my database to recover important information of my system or to move data from one place to another.



First, I open PHPMyAdmin then press on the import button and choose the data base file which I want to make recovery for.

One of the ways that help to make backup and recovery is using the Notepad. I use it to store all the queries that I apply in PHPMyAdmin. It helps me to keep another copy of data in case my data is lost, or damaged, I can get another copy of my data. Also, it helps me to make recovery of my data that is to get my data back after the data is lost, or damaged or it was not accessible.

## Database maintenance in general

It is a number of procedures and tasks that a developer makes to make sure that the performance of the data base system is efficient, integrate, and reliable. It includes regular testing, monitoring, and improving the performance of the system to avoid any situation which could negatively affect the performance, reliability, and efficiency of the system. These tasks involve backup, disaster recovery plan, software updates, integrity check, security management, which will help to reduce risk and increase functionality of the system.

**The first technique** of maintaining database is **Integrity check** is a type of testing to fix any damage in the database which includes ensuring that inserted data and the relationships are valid.

**Another technique** is **Backup and recovery** which are very important in any change or system updates to ensure its reliability, safety, and continuity. They can help developers to reduce the effects of problems on your system and prevent loss of data. They are also a crucial part of any disaster recovery plan. Which help to deal with attacks or other dangerous situations. And so, you can recover your data, safety, and continuity of your business. Moreover, they are very important to reduce risk and ensure compliance with regulations. It is very important to regularly check your backup to make sure that your system is working in the right way.

Also, we can make regular **software updates** which is very important to improve performance of the system by making new version, fix bugs, vulnerability of the security, adding new features and minimize the errors. other benefits of updating software system are improving security which includes activities to protect the system from possible threats and deal with attacking thus we can prevent illegal access or data violation. Also, improving performance of the system is another benefit of software updates. For example, we can apply an efficient algorithm, query, and faster techniques of retrieving data.

An efficient technique of maintaining data is **monitoring data and fixing errors** such as regular schedule testing and analysing the result of the testing and tracking. Also, error detection can be found out from any failure in the query or the system, violation, or contradiction in data. Then, diagnosing errors and finding resolution is very important throw many tasks such as analysing the error messages and any related data and then finding resolution depending on the type of error. Some of these ways of resolution could be data correction, configuration and updating of the system. The continuous monitoring of the system can prevent potential errors and threats.

To conclude, database maintenance is very essential for any organisation for many reasons, It helps to achieve data integration and improving performance. Also, It ensures reliability of any system and enhance its security, which will result in reliability and complying with the requirements. Also, it leads to the organisation’s future growth and expansion and success.

# Testing

## Data Validation

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Type** | **Description** | **Screenshot** |
|  | All cases of PK | In this case, I check the validity of the primary key is unique, so that the value will not be repeated. | Text  Description automatically generated with medium confidence |
| In this case, I check the validity of primary key is not null, so that the value will not be null |  |
|  | All cases of FK | On update cascade which means any change in the primary key will automatically happen on the foreign key. | PK in branch table before updating:    Fk in warehouse table before updating:    \*\*UPDATE branch SET branch\_Number=10 WHERE branch\_Number=1;  PK in branch table after updating:    Fk in warehouse table after updating: |
| ON delete cascade which means any deletion on the primary key will automatically happen on the foreign key. | PK in branch table before deleting:    FK in warehouse table before deleting:    \*\*DELETE FROM branch WHERE branch\_Number=10;  PK in branch table after deleting:    FK in branch table after deleting: |
|  |  |
|  |  |
|  | Unique | In this case, I check if the validity of column is unique , so the column which has unique constraint will have unique values such as (name, phone, and time) as columns , so I can prevent repeating data. | 1. Unique name      1. Unique phone      1. Unique time |
|  | Default | In this case, I check validity of the default constraint . I didn’t enter the value in the column which has default constraint. The values will be default values. In this case, the male and cash are the default values | INSERT INTO customer (Customer\_Id,first\_name,last\_name,phone,birthdate,city,email,building\_name,building\_number,street\_name )  VALUES (2, 'Jane', 'Smith' ,0785050667, '1992-12-05','London', 'jane.smith@example.co.uk', 'The Shard' ,32 ,'London Bridge');    INSERT INTO transaction (TRANSACTION\_Id ,amount ,customer\_id)  VALUES (21 ,'50.00' ,1); |
|  | Not null | In this case, I check the validity of not null constraint. I should enter the value in columns which have not null constraint | 1. City |
|  | Varchar data type | In this case, I check the validity of varchar data type. If the value is anything except integer, it should be written between single quotations. |  |
|  | datedata type | In this case, I check the validity of (date) data type. If the value is inserted wrongly, it will be stored 0000-00-00 |  |
|  | Time data type | In this case, I check the validity of (time) data type. If the value is inserted wrongly, it will be stored 00:00:00 |  |

## Output Validation

|  |  |  |
| --- | --- | --- |
| **Query Description** | **Screenshot (query + result)** | **Result validation** |
| This query will show all the information of the warehouses. | A screenshot of a computer  Description automatically generated with low confidence  **The query of the view which consist of the select statement.** | I checked this query by going to the warehouse table and warehouse\_phone table, and then compared them and made sure that all information in the two tables match the result of the query. Also, I made sure that the phone is related to the appropriate warehouse which means that the PK in the warehouse table quals the FK in the warehouse\_phone table. |
| This query will show the name of product and the id of transaction which has this product and the amount of this transaction. | A screen shot of a computer  Description automatically generated with low confidence  **The query of the procedure which consist of the select statement.** | I checked this query by going to the transaction\_product table and then finding the id of transactions which has id of the product and then went to transaction table and saw the amount of these id transactions. The amount equals the result of the query, so the query is correct. |
| This query shows the name of all products and the id of all transactions which have these products and the amount of these transactions. | A picture containing text, screenshot, number, font  Description automatically generated  **The query of the view which consist of the select statement.** | I checked this query by going to the transacrion\_product table and see all id of transactions and then going to the transaction table and see the amount of the transaction depending on id of the transaction and product together. The amount equals the result of the query, so the query is correct. |
| This query shows the name of product and the stock for each one. | A picture containing text, screenshot, font, number  Description automatically generated  **The query of the view which consist of the select statement.** | I checked this query by going to the product table and see the stock for all products and compared the stock with the result of the query. |
| This query shows the number of warehouse and the name of the product and stock in this warehouse depending on the number of the warehouse and the id of the product. | A picture containing text, screenshot, font, line  Description automatically generated  **The query of the producer which consist of the select statement.** | I checked this query by going to the warehouse\_product table and see if the product is already there in the warehouse, and then going to the product table and see the stock of the id of the product. I found that what I checked equals the result of the query, so the query is correct. |
| This query will show some pieces of information about the product which the customer can see. | A screenshot of a computer  Description automatically generated with low confidence  **The query of the producer which consist of the select statement.** | I checked this query by going to the product table and see if the query returns the information in the right way and order |
| This query will insert information for the new product. |  | I checked this query by going to product table and checked if the information is added in the right way. |

## Security Validation

**Note**: you need to test the given and not given privileges.

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **User Name** | **Description of privilege/no privilege** | **Screenshot (query + result)** |
|  | CEO | The CEO has a privilege which allows him to see all the information of all employees, but he doesn’t have the privilege to allow him to delete their data. also, he can’t insert the information of employees |  |
|  | CEO | The CEO has a privilege which allows him to see all the information of all warehouses, but he doesn’t have a privilege to insert their information but just see it |  |
|  | CEO | The CEO has a privilege which allows him to see the amount of each transaction of all the products in all warehouses. But he can’t delete any amount in any transaction. |  |
|  | manager | The manager has the privilege which allows him to see the stock of product depending on the id of product and the number of warehouse which the product exists in . |  |
|  | Manager | The manager has the privilege which allows him to see the number of warehouses which have this product according to the ID of the product. |  |
|  | employee | The employee has the privilege which allows him to see all the information about the products according to the ID of product. But he can’t delete the information about the product. |  |
|  | Employee | The employee has a privilege which allows him to insert all the information about the products. |  |
|  | customer | The customer has a privilege which allows him to see the information about the product, but he can’t delete it. |  |
|  | manager | The manager doesn’t have the privilege which allows him to insert information about the customer |  |
|  | employee | The employee doesn’t have the privilege which allows him to insert information about the warehouse |  |
|  | customer | The customer doesn’t have the privilege which allows him to insert information about the warehouse |  |
|  | Customer | The customer doesn’t have the privilege which allows him to update the important information about the employee |  |
|  | CEO | The CEO doesn’t have the privilege which allows him to delete information about the customer |  |

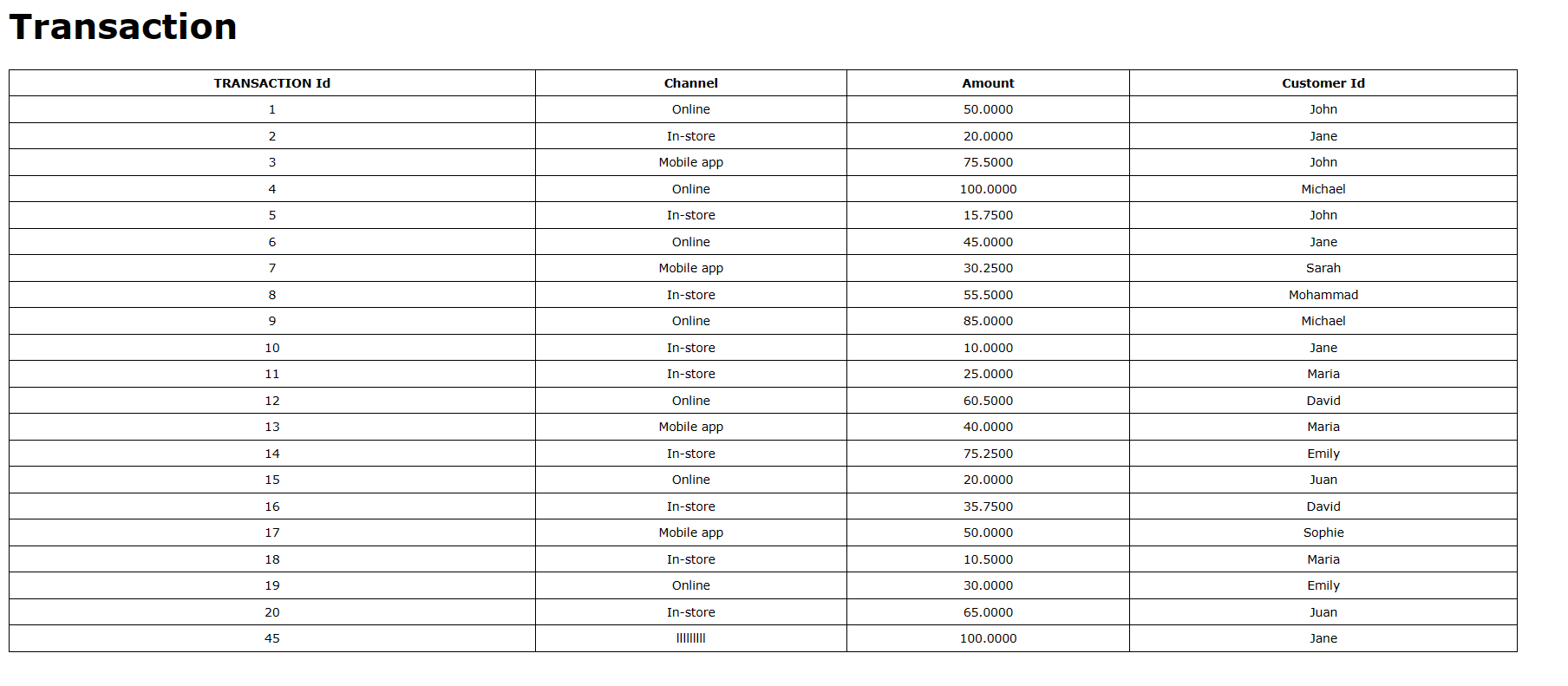
### 

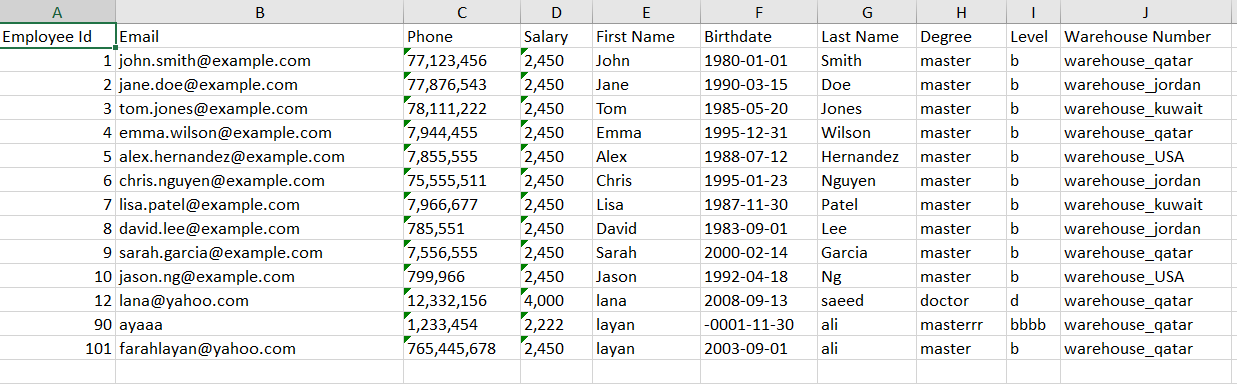
## GUI Validation

|  |  |  |
| --- | --- | --- |
| **Number** | **Description** | **screenshot** |
|  | **In employee page**, the employee can insert, update, view his/her information, but he/she can’t update and insert the salary, degree, level, ID. So, **the first picture** shows when the employee clicks on edit (update) icon  the employee will not see the ID, salary, degree, level information to edit them. **the second picture** shows when the employee clicks on the because he/she want to insert her/his data, the employee will not see the ID, salary, degree, level information to insert them. **the third picture,** the employee can see (view) all her/his information by click on the employee page from home page. | the first picture    the second picture    The third picture |
|  | **In employee address page**, the employee can insert, update, view his/her address information, but she/he can’t update the ID. So, **the first picture shows** when the employee clicks on edit (update) icon.  the employee will not see the ID, to edit it. **the second picture** shows when the employee clicks on the because he/she wants to insert her/his data. **the third picture shows** the employee can see (view) all her/his information by clicking on the employee address page from home page. | The first picture    The second picture    The third picture |
|  | **In product page**, the employee can edit the information of the product by clicking on edit icon  as we can see in **the first picture**. **the second picture** shows the employee can insert all the information of the product by clicking on the  **the third picture shows** the employee can see (view) all the information of the product by clicking on the product page from the home pages. | The first picture    The second picture    After inserting data    The third picture |
|  | **In transaction page**, the employee can just see the information but not edit or insert them because this information about the transaction is only inserted when the customer buys the product.  the first picture shows the employee can view all the information about the transaction by clicking on the transaction page from home page. **the second picture** shows that the employee can’t edit the information when he/she clicks on edit icon    **The third picture** shows that the employee can’t insert the information when he/she clicks on | The first picture    The second picture    The third picture |
|  | **In transaction product page**, the employee can just see the information but not edit or insert them because the information of this page is inserted when the customer buys the product in specific time and date.  **the first picture** shows that the employee can view all the information about the transaction product page by clicking on this page from home page. **the second picture** shows that the employee can’t edit the information when he/she clicks on edit icon  **The third picture** shows that the employee can’t insert the information when he/she clicks on | The first picture    The second picture    The third picture |

Note: the employee can print and export the data **from all pages.**

**Print the data from transaction page.**



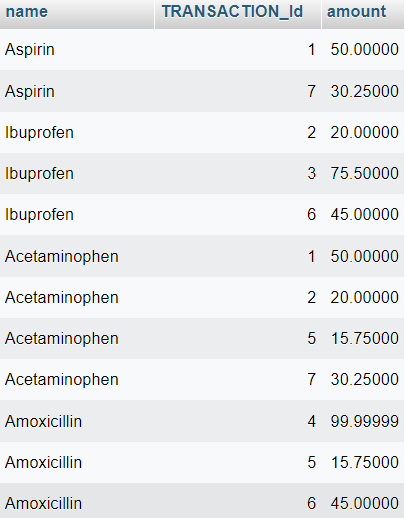
**Export to excel.** 

## Assess whether meaningful data has been extracted.

I want to check if the that I will get from the select statement is logical and meaningful. The select statement could be written in view and the procedure. So, I want to make sure that the view and procedure that I gave to my users return the meaningful data.

**The first View:**

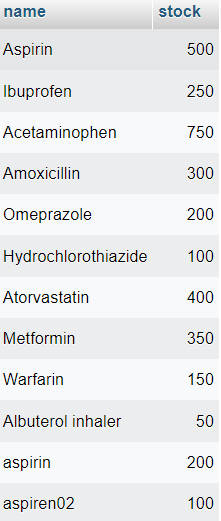
The CEO can Generate a report to see detailed data on all products such as the sales and the number of transactions. So, I created view which enables CEO to see this information.



In the table above, we can see the result of the view. We can see that the return data is meaningful and logical because it returns only the data that the user is expected to read. In the table above I can see that more that more than one product have the same transaction and the many transactions have the same product this is logical because the relationship between the transaction and product tables is many to many.

**The second View:**

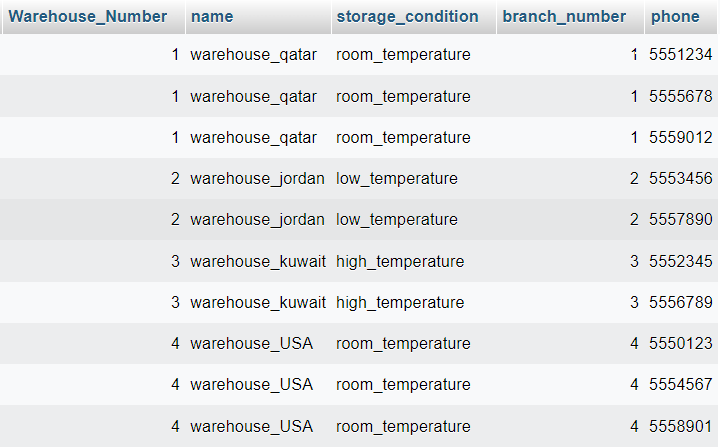
The CEO can view the stock of products in any warehouse. So, I created view which enables CEO to see this information.



In the table above, we can see the result of the view. We can see that the return data is meaningful and logical because it returns only the data that the user is expected to read.

**The third View:**

The CEO can view the warehouse information. So, I created view which enables CEO to see this information.



In the table above, we can see the result of the view. we can see that the return data is meaningful and logical because it returns only the data that the user is expected to read. For example, When the CEO wants to see the warehouse information, he may get one or more number for each warehouse this is logical because the phone attributes is multivalve and each warehouse may have one or more phone number.

**The fourth View:**

The customer can view available medicine, prices, and expiry dates. So, I created view which enables CEO to see this information.



In the table above, we can see the result of the view. We can see that the return data is meaningful and logical because it returns only the data that the user is expected to read.

I expect that this the select statement will return each product how many transactions include this product.



It the table above, we can see the result of the select statement. we can see the returned data is meaningful and logical because it returns only the data that the user is expected to read. Also, if I count the number of the product in each transaction manually, it will get the same result.

## Assess the effectiveness of testing.

IN my data base I made a many testing processes.

**The First testing process** 🡪 I want to make sure of the validation of the type of the attributes and the constraints on them before data is inserted because we should make sure that all data inserted in the tables is logical and meaningful. For example, I checked the validity of the PK constraint because it is not logical that more than one attribute has the same PK because its restrictions are unique (not repeated) and not null (not empty). Another testing is related to date and time type, If I put date or time as a type on the attributes, data and time should be inserted in a particular form. For example, the form of the date should be yyyy\_mm\_dd and the form of the time should be hh: mm:ss. Also, I checked the validity of the FK constraint because it’s not logical that the attribute which has FK constraint has FK value that is more than the range of PK because any FK is taken from PK, and it points to the PK that it is taken from. All these of ways of testing are useful for ensuring that the system works efficiently, as well as preventing inserting illogical data.

For example, if the user inters numbers in the name filed instead of letters the inserted data will be illogical so I should check that the type of data is suitable for the content of the field.

Also, I checked that the validation of not null constraint because in case the customer wants to buy a product and he/she didn’t inter the name of the product, that is not logical because we will not know which product, he / she wants. In other words, this field is required, and the customer shouldn’t skip**. Finally,** checking that the validation of the type and constraints of the attributes is correct will increase the integrity of the data, compliance with requirements and regulations, and error prevention.

**The Second testing process🡪** I want to make sure of the validation of the select statements, which helps me to retrieve and check data. This will be done by testing the views and the procedures which include select statements or by testing select statement directly. This will help me to make sure that the data that we got is logical, meaningful, and it is what the user wants exactly; not more and not less. For example, if the employee wants to see the products in the warehouse where he warks, there is no need to see information about products in other warehouses. As a result, this will ensure data accuracy, consistency, and error prevention. Also, it will help better quality of fata and more reliable content.

**The Third testing process🡪** I want to make sure of the validation of the privileges that the user has or not have. I checked that the privileges for each user are logical and compliant with the requirements. For example, I should make sure that the employee can view his salary, level and degree but he can’t edit or insert them because this is illogical, and it is the responsibility of the manager. Also, I made sure that the manager can view all information about the employees in his warehouse, but he can’t view, edit, insert information about the employees in other warehouse. This will help to ensure that sensitive data is protected and complies with the requirements and regulations, as well as preventing errors and ensuring that the system is auditable and stable. Other benefits are security enhancement, access control and not wasting time on features which are not accessible. Thus, this will help to make sure that the actions of the system align with the responsibilities of the legal users of the system.

**The Fourth testing process🡪** I want to make sure of the validation of the user privileges on the GUI. I checked what the user privileges and what he can or can’t do on each page in order to make sure that he/she can only do what he/she is privileged to and doesn’t affect what other users can do or can’t do as stated in the system. **For example**, the employee can’t edit the information about the transaction in the transaction page. He can view them because the information about the transaction is made when the customer buy the product such as time, date, and the amount of the transaction. Also, the customer can’t edit the information about the product on the product page. he can view and buy them by doing transactions because that is the responsibility of the employee who works in the warehouse. As a result, this will help to achieve many benefits such as clarity, empowerment of users and increasing his experience. Other benefits are related to data such as protecting data and maintaining security, which will help error prevention and complying with requirements and regulations. These all will make the system more secure, user friendly and compliant with the work environment.

# Evaluation of database solution

## Effectiveness of the database solution based on user and system requirement.

First, After I finished designing data base in the first part, I implemented my data base, so I used Phpmyadmine to create the data base on my device and I used Apache which is a virtual server that stores my data base. This is because data base is backend so it needs a server to store database in.

I created tables in a logical way depending on requirements of the system for my database which is compliant with my design, so I crated 10 tables, each one includes the needed data.

**Branch table:** This table stores information about the branches of LS\_Hospital such as number, name, phone, address which includes (city, Streat name, building name, building number). No one has a privilege to access this table but in the final improvement I plan to give the CEO the privileges to insert, view, update, and delete because he is responsible for data management in this table.

**Warehouse table:** This table stores information about the warehouse in the branches in the hospital such as number, name, storage condition, the number of the branch which the warehouse exists in. In this table, I give the manger the privileges to insert, view, update, and delete because he is responsible for data management in this table. This is because it is not logical to give other users such as employees and customers the privileges to deal with the data related to the warehouse.

**Warehouse phone table:** This table stores the phone numbers of warehouses because each warehouse may have one or more phone number, so this table stores the numbers of warehouses and the phones. The benefit of having this separate table is to reduce redundancy of data. This table is related to the warehouse table, so I designed it in the same way and I give the privileges only to the manager.

**Employee table:** This table stores information about employees who work in the warehouses for LS\_Hospital in all branches. Any employee should have this information such as ID of employee, email, phone, salary, name (first and last name), birthdate, degree, level, and the number of warehouses which the employee works in.

On this table, I just gave the employee and the manager the privilege on this table. Also, I give the employee the privilege of updating specific attributes, not all of the attributes taking into account that it is not logical that the employee has access to sensitive data such as salary, degree, level and this is the responsibility of the manager.

**Employee address table:** This table stores information about employees’ addresses because employees may have one or more address, so this table stores the ID of employees and their different addresses. The benefit of having this separate table is to reduce redundancy of data. This table is related to the employee table, so I give the privileges to the manager and the employee on this table because it’s not logical to have access on the employee table and not to have access on the employee address table because information in these two tables is connected.

**Transaction table:** This table stores the information about the transaction made by the customer when he buys the medicine such as the ID of transaction, amount, channel, and Id of customer who does the transaction. I give the manager, employee, customer privilege on this table but off courser each user has different privileges depending on the position for each user. For example, the customer can insert the data on this table; the employee and the manager can only view the data, but they cannot update the data because it is not logical to update the data of the transaction because this data is inserted when the customer makes the transaction.

**Product table:** This table stores the information about any product in the warehouse in LS\_Hospital such as the ID for the product, name, usage, price, form, expiration and production date, wight, and stock . I give the manager, employee, customer privileges on this table but off courser each user has different privileges depending on the position of each user. For example, the customer has the view privilege on this table because it is not logical to insert, update, and delete the information about the product, but the employee has the privileges to insert, update, delete, and view data. Because that is the responsibility for the employee.

**Transaction product table:** This table stores the ID of products which are found in any transaction and the ID of transactions which include any medicine. The benefit of having this separate table is to reduce redundancy of data. I give the employee and manager the privilege of viewing the data so that they can see how many transactions which have the product.

**Warehouse product table**: This table stores the ID of products which is found in any warehouse and the ID of warehouses which may include any product. The benefit of having this separate table is to reduce redundancy of data. I give the manager the privilege of viewing data so he can see how many warehouses which have the product.

**Customer table:** This table stores information about customers when they buy the medicine such as the ID of customers, name (first and last name), phone, birthdate, address which includes (city, Streat name, building name, building number), email, and gender. I give the customer and employee privileges on this table. Customers can view, insert, update their information. The employee can view the information of all the customers.

after I created tables with their attributes and test their type and constraints to ensure that all inserted data is logical and meaningful. Then I inserted data to the data base, then I tested if the database returns the required appropriate data to ensure that the database works efficiently.

Then I created the views and the producers to create the privileges for the users. After that, I tested them, and I created the 4 users to give them access on my database. then I granted each user the privileges depending on his position and the user requirements. The privileges are taken from the views and the producers.

I tested that each privilege and non-privilege for each user are correct to ensure that each user has the correct privilege and ensure security of the data and system. I don’t miss any case in testing, so my testing was efficient.

Finally, I used “PHP generator for MySql professional” to connect the database with the HTML to

create GUI. In the GUI, each user has privileges depending on the user requirements. Each user has

limited access to each page of the GUI to protect data and maintain security, which will help in

error prevention and complying with requirements and regulations. These all will make the system.

more secure, user friendly and compliant with the work environment.

## Suggested improvements

Developers though of improving the data base system in order to cope with the recent technologies in many different ways. Also, these changes will make database more efficient, and reliable.

**One of these improvements is** allowing employees to make transactions using the system to buy any product of the warehouse, that is not only the customers who can get this service. Another improvement is giving the CEO the privileges of updating, inserting, deleting, viewing the branch table because he is responsible for this table. In addition to limiting the possibility of using the system to people who are +18 by providing the birthdate attributes with check constraint. Also, giving the customer the privilege to update his transaction and giving the employee the privilege to delete the transaction if it lasted for a long time without performing.

## Evaluation based on improvements needed

In order to cope with the increase of demand for the medical products and increase productivity of the warehouse, I thought of improving the services provided by the warehouse by adding more products such as apparatus and medical tools. Consequently, I will add a new table which stores information related to the added products. This table will be treated as a product table with regard to which users can use it or who has privilege on it as well as the relationships which connect it with other tables.

Another improvement is adding the delivery service and allowing the employees and customers to track the orders in order to deal with issue of delivery delay and do what is required. As a result, I will add new table which stores information about delivery such as amount, current location, date, time, name if the customer, the product included. Also, It will be connecting with others table. Employees and customers will be given privileges to use this table.

Another improvement is adding online payment feature by improving a payment getaway within the system taking into account the safety and security of this process. Also, this feature will allow developers to store data related to this transaction securely such as bank account number or visa account number, IDs, amount of money, and state of payment. Consequently, a new table related to the previous points will be added to ensure that the database can deal with this service securely and comply with standards and regulations.

A new branch may be added to deal with large demand and overload on the previous branches. In this case, database will be able to deal with expansion because the tables which deal with the branches’ information are already there. All we need to do is to update the database migrate data from the previous system within a carefully designed database.

Finally, the new system will be able to deal with customers’ feedback and enquiries, which necessitates employing workers to follow-up web enquiries and send more information to customers. As a result, two tables will be added to the database; one to store feedback information about the transactions and customers, and another to store information about people who receive follow up.

**References:**

1. <https://www.makeuseof.com/what-is-database-maintenance/>
2. [Database maintenance tasks - IBM Documentation](https://www.ibm.com/docs/en/product-master/12.0.0?topic=database-maintenance-tasks)
3. [Database Maintenance Explained - OfficeTools](https://www.officetools.com/knowledgebase/database-maintenance-explained/)
4. [What Is Technical Documentation? (And How To Create It) | Indeed.com](https://www.indeed.com/career-advice/career-development/technical-documentation)