

CHAPTER I

INTORDUCTION

1.1 Introduction

Tupperware Management System is a web-based application that provide ordering service. Staffs able to manage the system through the system. Public can view the detail of our product. Ordering service just valid for registered member only.

Tuppershop releases standard and quality products which lead public towards higher level of lifestyle. The products available in Tuppershop are bottle, container, tableware and others. Tuppershop encourages public to use their products instead of polystyrene lunch box, plastic tableware or bottle. It may harm to us when we use it in the long term.

The used of Tuppershop Management System can enhance the services that happens in company where it helps in reducing the workload of staff, make order management become more manageable and easier to control and centralized the customer and order database.

In a nutshell, Tuppershop Management System provides better platform to clients and staffs for better management of the transactions. Besides that, Tuppershop can keep track on the latest quantity in hand of product and analyse the overall selling record through the system. This helps them to make better decision in future for better organization development.

1.2 Problem statement

The TupperShop Management System is tending to develop a new ordering system because the system will bring more benefits for the company. Below are the problem that identified from the current system:

- i. The data of the company maybe will save on different relational database according to department.
- ii. The oldest data are very useful when then database occurring the data lost problem.
- iii. Some department maybe need the data of other department which use the different relational databases.

1.3 Objective

- i. Show the integration of multiple relational databases.
- ii. Manage backup and restore data from multiple source of relation database.
- iii. Demonstrate the features of data integration of relation database by using simple user interface.

1.4 Scope

1.4.1 Scope of user

a. Staff

- Manage the orders of the Tupperware.
- Manage the information of supplier
- Calculate the total stocks get from the supplier

b. Customer

- View the stock of Tupperware
- View the detail of the Tupperware
- Make orders of the Tupperware
- Get the detail of the orders

c. Supplier

- Contact with the staff to get the product need to supplied
- Provide the details of the product
- Supply the stocks to the client and company

1.4.2 Scope of system

a. Staff Department

- Staff information
- Department information

b. Inventory Department

- Tupperware stock
- Tupperware information
- Stock management
- Tupperware price
- Details of Tupperware
- Tupperware quantity

c. Data Audit Department

- All the backup data from other department
- Recovery when some data are lost
- Record all the action of database

d. Marketing Department

- Quantity of order
- Orders price
- Orders type
- Customer information

1.5 Project significant

As we all know, database has availability of storage system, the most obvious of these being persistence. Without database project, staff will face problem to search details about equipment because record manual system normally easy to lost. Therefore, it is important to have Tupperware Management System for helping staff to search data about all the Tupperware and customer details for future reference.

Besides that, with this database project a lot of time can be reducing for searching details about Tupperware and order details. Staff will just need to key in the product id and supplier id, then found all the record about the product and order compared to manual filling which take more time to find record. If there are more order added in manual fill, it will increase the time to search.

Other than that, database project is also important to share data of all kinds between users, between tools working on them, and usually between different computers. All should be able to work with the same set of data. We need persistent storage and storage must be reliable. Several users and/or several tools must be able to safely access the same data concurrently.

In addition, database project may provide more efficient access to large amounts of data, through indexing and other optimizations. Staff will be able to keep many record without finding hard to find the saved record.

1.6 Conclusion

In conclusion, Tupperware Management System is very useful and convenient in searching, viewing, inserting, deleting and updating the order, product and supplier of tupperware.

Furthermore, this system is easy-to-use and increases the data consistency. Problems with data redundancies can be avoided to a very minimum state with a good database design. Staffs and customers will benefit with the interactive tupperware records and can perform data manipulation faster. This database system will also increase customer's data security and privacy.

This system is developed by using different database such as SQL Server, MySQL and Oracle 11g for Windows, and PostgreSQL for Ubuntu. It involves data transmissions between different database, data manipulation, and data addition. Good management is important to keep the business running without any problems.

CHAPTER II

DATABASE SYSTEM METHODOLOGY AND PLANNING

2.1 Introduction

To ensure the system has been developed efficiently, Waterfall Model is applied as the methodology in this project. The Waterfall Model consists of five stages which are Analysis, Design, Testing and maintenance. Each of the stage is a starting stage of the next stage. So if there are any problem arise, we can refer back to stages that we gone through. Therefore, we decided to use Waterfall Model based on Database Development Life Cycle (DBLC) as our methodology to develop our system.

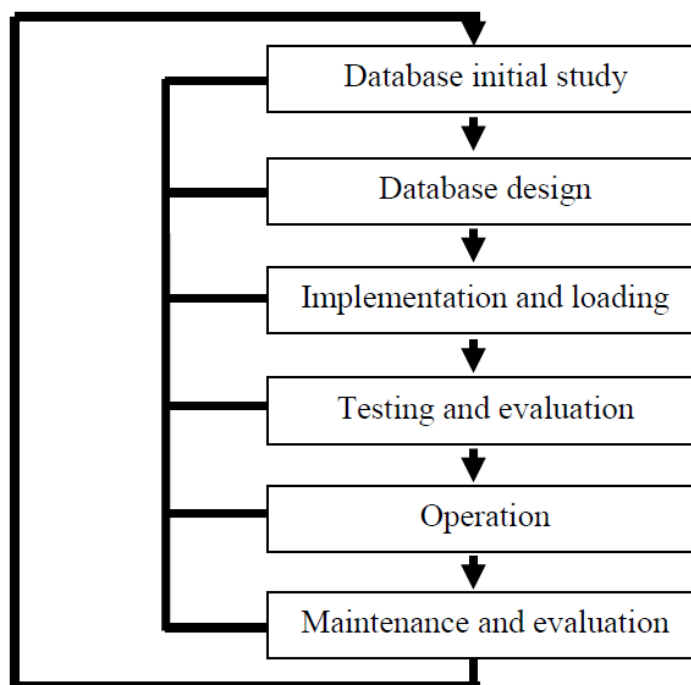


Figure 2.1: Database life cycle

2.2 Database development methodology

Tuppershop Management System was a system need developed by the Tuppershop Company to make the order and supplier details of Tupperware more efficient. So the Tuppershop Management System is a new system that want develop by company to manage their Tupperware order. There are four department to be make better which are Staff Department, Inventory Department, Marketing Department and Data Audit Department.

Staff Department will be developed in Windows operating system with Oracle 11g database. Inventory Department will be developed in Ubuntu operating system with PostgreSQL database. Besides, Marketing Department will developed with SQL Server database in Windows operating system. Finally, Data Audit Department will be developed in Windows operating system with MYSQL database.

Database Life Cycle contains six (6) phases which is database initial study, database design, implementation and loading, testing and evaluation, operation, maintenance and evaluation.

- i) Database initial study
 - a) Analyse the current company situation to get the requirements. Analyse the current general operating environments of the company.
 - b) Identify and specify the problems that have in a current system and justify need for new data and databases in support of management.
- ii) Database design
 - a) Analyse the transaction, forms, displays and inquiries required by the management functions supported by the database.

- b) Design the conceptual design like ERD, logical design like data dictionary and physical design like DDL statement.
 - c) Integrate database views into conceptual data model and define entities, attributes, primary keys, and foreign keys.
 - d) Identify data integrity and security requirements and populate repository.
- iii) Implementation and loading
 - a) Install the requirement software for the development process
 - b) Design database system and convert the data from the current system to the new system.
- iv) Testing and evaluation
 - a) Code and test database processing programs and evaluate the database and its application programs.
- v) Operation
 - a) Starts the process of system evaluation and produce the required information flow.
- vi) Maintenance and Evaluation
 - a) Introduce changes and make enhancements.

2.2.1 Methodology in developing database

a. Implementation

In the implementation stage, we will be installing four different databases which are Oracle 11g, MySQL, SQL Server and PostgreSQL. Then we will be using the Data Definition Language (DDL) to create the database and include all the tables we needed. Finally, we will use Data Manipulation Language (DML) to enter few data to make sure the database is working properly. The programming language use to develop the interface is JSP and ASP.

b. Testing

We will conduct two type of test to test our system which are unit testing and system testing. Unit testing will be conducted to each of the subsystem individually which include Staff Department, Inventory Department, Marketing Department and Data Audit Department. We run the unit testing by input sample data to the subsystem to test the connection between interface and database and error message. If the database implementation is fails to meet the system's evaluation criteria or requirement, several options will be considered to enhance the system such as follows:-

- For performances related issues, the designer must consider fine tuning specific system and DBMS configuration parameters. The best sources of information are the hardware and software technical reference manuals.
- Modify the logical design and upgrade or change the DBMS software or the hardware platform.

As for system testing, the four subsystems will be integrated to one system. System testing will be conducted to the whole system to test whether the data can be transfer or migrate from one database to another database using different platform.

c. Maintenance

At this stage, the system will be put into practical use. There might be problem that were not discovered in early stages arise. Therefore, maintenance involve correcting error, improving the implementation of system and enhancing the system's services as new requirement are discovered.

2.3 Database development project requirement

2.3.1 Software requirement

There are listed the requirement and specification of software components, which have been used in TupperShop Management System. There are:

i. Oracle 11g

One of the databases that used in developing the system is Oracle 11g. It can run on many platforms such as Window 8.1, Window 10 and Linux. The Oracle Database also commonly referred as Oracle RDBMS consists of a relational database management system (RDBMS). Oracle 11g Database provides efficient, reliable and secure data management.

ii. MySQL (My Structured Query Language)

One of the databases that used in developing the system is MySQL. It can run on many platforms such as Window 8.1, Window 10 and Linux. It is a

relational database management system. Its scope includes data query and update, scheme creation and modification, and data access control.

iii. PostgreSQL

One of the database that used in developing the system is PostgreSQL. It can run on many platforms such as Window 8.1, Window 10 and Linux. PostgreSQL is a sophisticated Object-Relational DBMS. It supports almost all SQL constructs, including sub selects, transactions, and user-defined types and functions. It is the most advanced open-source database available.

iv. NetBeans

NetBeans is a software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components called modules.

v. Eclipse

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE.[3] It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications,

vi. Microsoft SQL Server

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with

the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet).

vii. Windows

Windows is one of the platforms that used in developing the system.

Windows is a very famous platform that frequently use in public

viii. Linux

Linux is one of the platforms that used in developing the system. Linux is a free platform to use.

ix. Microsoft Visio Office 2007

Microsoft Visio Office 2007 is a software diagramming program for Microsoft Windows that uses vector graphics to create diagrams. These diagrams include flow chart, context diagram, data flow diagram (DFD) and entity relationship diagram (ERD).

2.3.2 Hardware requirement

Table 2.1: Hardware requirement

No.	Hardware	Description
1.	Processor	Intel (3.40Ghz)
2.	Memory	2GB
3.	Hard Disk	80 GB
4.	Drive	DVD-ROM drive

2.4.2. The segregation duties of team

Table 2.3: The segregation duties of team

Name of member	Position	Module	Software using
MUHAMMAD AMMAR BIN MUHAMMAD SANI	Project Manager	Marketing Department	-Windows 8.1 -SQL Server
NG KOK SOON	Software Analysts	Inventory Department	-Linux -PostgreSQL
LIAU SHUK YEE	Programmer	Data Audit Department	-Windows 8.1 -MySQL
MUHAMAD SYAHIRAN BIN MOHD SHAHNIN	System Architect	Staff Department	-Windows 8.1 -Oracle 11g

2.5 Conclusion

At the end, the literature review and project methodology that are using to define the planning of the project is discussed in this chapter. We develop our TupperShop Management System by using the Waterfall model. During analysis phase, we analyse the problems of the current existing of TupperShop Management System. Then, we try to design the solutions for the problem in the design phase. After that, we will start our implementation phase by implement databases and interfaces for our system. After completing our system, we will test our system by unit testing and system testing during testing phase. Lastly, we will maintain our system as in the maintenance phase.

CHAPTER III

DATABASE SYSTEM ANALYSIS

3.1 Introduction

The database analysis report to explain the current system analysis and system analysis. Database analysis is an initial overview of the database will be developed. Analysis carried out the existing system weakness and improve the quality of the system throughout the time.

3.2 Current system analysis

The current system has many flaw that we detect, such as lack of security when user register or update, their password were stored in plain text in database and not encrypted. On the other hands, the system server was only in one place and does not have backup and recovery function. This is very crucial because just in case if the only server down or has disaster, there were no backup and need to build from scratch again, this time wasteful. Another flaws are the database server is only in one place, meaning that if staff that in charge of sales were transfer, and the database that stored sales information store in only the headquarter, its make the system difficult to access.

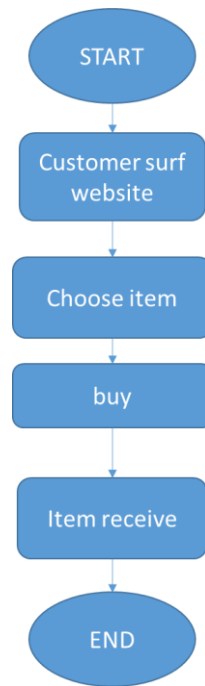


Figure 3.1: Current system flow

Based on the flow above, the flow events are:

- i. Customer surf the web to view the product.
- ii. Customer choose the item to buy.
- iii. Customer buy without login account for tracking and history purchase purpose.
- iv. Customer did not receive notice of the status of shipment, instead customer suddenly receive the item and does not has receipt as proof of purchase.
- v. The operation end.

3.3 To-be system analysis

From current system analysis, we know that, some improvement could be made with adding extra functionalities and features. So, as the analysis so far, we would like to create a web based application, TupperShop Management System. This proposed system will use Object Oriented Programming as the foundation development model for the application.

3.3.1 Use case diagram

TupperShop Management System use case diagram is as follow:

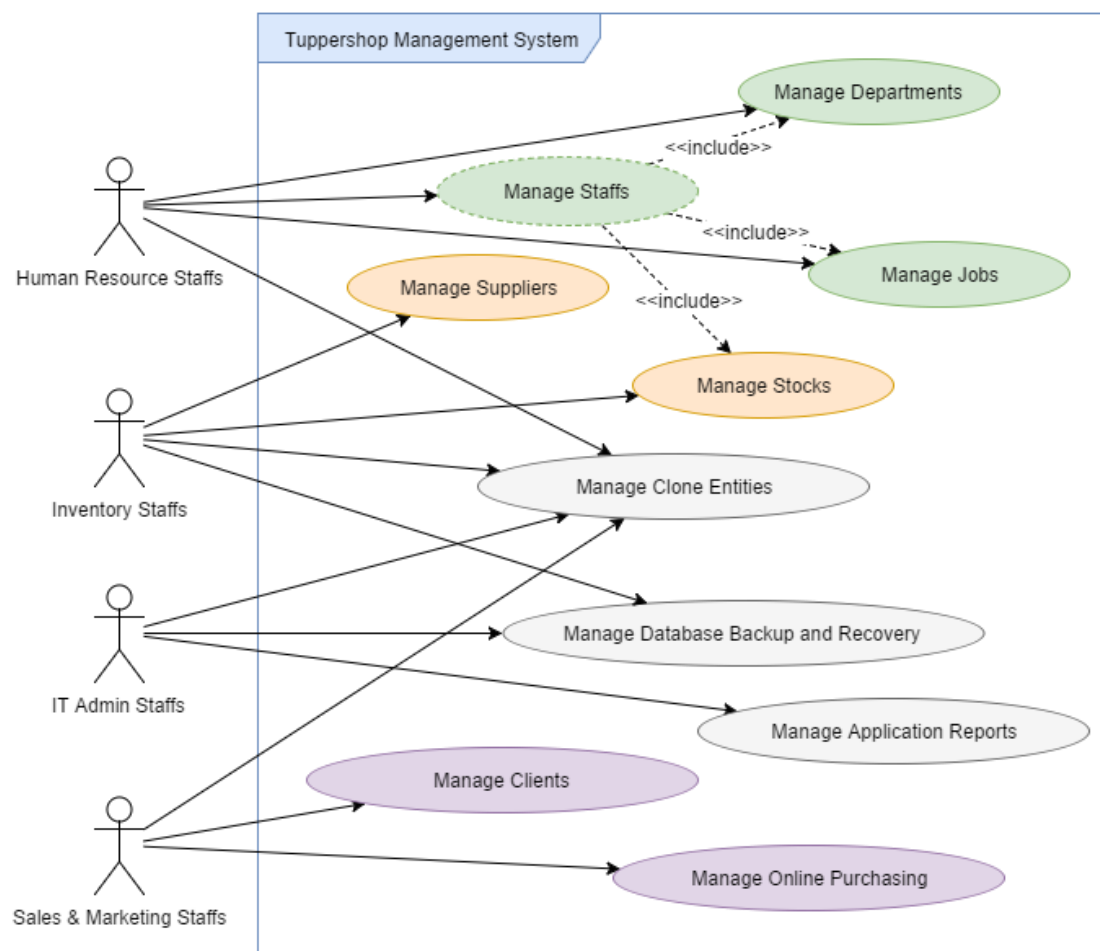


Figure 3.2: TupperShop use case diagram

3.3.2 Use case specification

1 Manage Departments

1.1 Description

The use case describes the process for the actor to manage departments information.

1.2 Pre Condition

The actor must have been authenticated.

The actor was Human Resources Department staff.

1.3 Post Condition

The actor able to view department list or create new department or update department information or delete any existing department.

1.4 Actor

Human Resource Staff

1.5 Flow of Events

1.5.1 Primary Flow

P: View Departments

1. The Use Case starts when the actor navigated to the Manage Department Screen from the Navigation Menu.
2. The screen fetch data from Department table in database.
3. The screen shows a department list with certain action on the same screen such as New Department (A1), Update Department (A2) and Delete Department (A3).

4. The Use Case ends.

1.5.2 Alternatives Flow

A1: Create New Department

1. The Use Case starts when the actor clicked a button/link for creating new department.
2. The screen displays a form with some information needed to fill up in order to create new department.
3. A complete form will submit for inserting process into database.
4. The Use Case ends.

A2: Update Department

1. The Use Case starts when the actor clicked on any edit button/link resides at any of the records row.
2. The screen then requests a department info from database based on selected department to update department info.
3. Then the info retrieved will be fill up on the form fields.
4. After changes made by the actor, the latest info will be submitting and save the updates to database depends on which department is currently being updated.
5. The Use Case ends.

A3: Remove Department

1. The Use Case starts when the actor clicked on any remove/delete button/link on any row the actor wants to delete.

2. The application would validate the actor either want to proceed with removing the department record or cancel the process (E1).
3. The department would be delete if the actor chose to continue remove it.
4. The Use Case ends.

1.5.3 Exceptional Flow

E1: Department Remove Cancellation

1. The Use Case starts when the actor wanted to remove a department.
2. The process cancelled when the actor denies to continue with the deletion process.
3. The Use Case ends.

1.6 <<include>> Use Cases

Not Applicable

1.7 <<extend>> Use Cases

Not Applicable

1.8 Rules

Not Applicable

1.9 Constraint

Not Applicable

1.10 Comments

Not Applicable

2 Manage Jobs

2.1 Description

The use case describes the process for the actor to manage jobs information.

2.2 Pre Condition

The actor must have been authenticated.

The actor was Human Resources Department staff.

2.3 Post Condition

The actor able to view job list or create new job or update job or delete any existing job.

2.4 Actor

Human Resource Staff

2.5 Flow of Events

2.5.1 Primary Flow

P: View Jobs

1. The Use Case starts when the actor navigated to the Manage Jobs Screen from the Navigation Menu.
2. The screen fetch data from Job table in database.
3. The screen shows a job list with certain action on the same screen such as New Job (A1), Update Job (A2) and Delete Job (A3).
4. The Use Case ends.

2.5.2 Alternatives Flow

A1: Create New Job

1. The Use Case starts when the actor clicked a button/link for creating new job.
2. The screen displays a form with information needed to fill up in order to create new job.
3. A complete form will submit for inserting process into database.
4. The Use Case ends.

A2: Update Job

1. The Use Case starts when the actor clicked on any edit button/link resides at any of the records row.
2. The screen then requests a job info from database based on selected job to update the selected job info.
3. Then the info retrieved will be fill up on the form fields.
4. After changes made by the actor, the latest info will be submitting and save the updates to database depends on which job is currently being updated.
5. The Use Case ends.

A3: Remove Job

1. The Use Case starts when the actor clicked on any remove/delete button/link on any row the actor wants to delete.
2. The application would validate the actor either want to proceed with removing the department record or cancel the process (E1).
3. The job would be delete if the actor chose to continue remove it.
4. The Use Case ends.

2.5.3 Exceptional Flow

E1: Job Remove Cancellation

1. The Use Case starts when the actor wanted to remove a job.
2. The process cancelled when the actor denies to continue with the deletion process.
3. The Use Case ends.

2.6 <<include>> Use Cases

Not Applicable

2.7 <<extend>> Use Cases

Not Applicable

2.8 Rules

Not Applicable

2.9 Constraint

Not Applicable

2.10 Comments

Not Applicable

3 Manage Staffs

3.1 Description

The use case describes the process for the actor to manage staffs' information.

3.2 Pre Condition

The actor must have been authenticated.

The actor was Human Resources Department staff.

3.3 Post Condition

The actor able to view staff list or create new staff or update staff or remove any existing staff.

3.4 Actor

Human Resource Staff

3.5 Flow of Events

3.5.1 Primary Flow

P: View Staff List

1. The Use Case starts when the actor navigated to the Manage Staffs Screen from the Navigation Menu.
2. The screen fetch data from Staff table in database.
3. The screen shows a staff list with certain action on the same screen such as New Staff (A1), Update Staff (A2) and Delete Staff (A3).
4. The Use Case ends.

3.5.2 Alternatives Flow

A1: Create New Staff

1. The Use Case starts when the actor clicked a button/link for creating new staff.
2. The screen displays a form with information needed to fill up in order to create new staff.
3. The screen also requests for department list from Department table in database to use by the actor to assign the new user with his/her departments assigned to.
4. A job list will also request to load on the screen as a selection for assign job for the new staff.
5. A complete form will submit for inserting process into database.
6. The Use Case ends.

A2: Update Staff

1. The Use Case starts when the actor clicked on any edit button/link resides at any of the records row.
2. The screen then requests a staff info from database based on selected staff to update the selected staff info.
3. Then the staff retrieved will be fill up on the form fields.
4. After changes made by the actor, the latest info will be submitting and save the updates to database depends on which staff is currently being updated.
5. The Use Case ends.

A3: Remove Staff

1. The Use Case starts when the actor clicked on any remove/delete button/link on any row the actor wants to delete.
2. The application would validate the actor either want to proceed with removing the staff record or cancel the process (E1).
3. The staff would be delete if the actor chose to continue remove it.
4. The Use Case ends.

3.5.3 Exceptional Flow

E1: Staff Remove Cancellation

1. The Use Case starts when the actor wanted to remove a staff.
2. The process cancelled when the actor denies to continue with the deletion process.
3. The Use Case ends.

3.6 <<include>> Use Cases

Manage Departments Use Case.

Manage Jobs Use Case.

3.7 <<extend>> Use Cases

Not Applicable

3.8 Rules

Not Applicable

3.9 Constraint

Not Applicable

3.10 Comments

Not Applicable

4 Manage Suppliers

4.1 Description

The use case describes the process for the actor to manage supplier's information.

4.2 Pre Condition

The actor must have been authenticated.

The actor was Inventory Department staff.

4.3 Post Condition

The actor able to view list of suppliers and take action on it such as update or delete.

The actor can also create new supplier if he/she need to do it.

4.4 Actor

Inventory Staff

4.5 Flow of Events

4.5.1 Primary Flow

P: View The List of Suppliers

1. The Use Case starts when the actor navigated to the Manage Suppliers from the Navigation Menu.
2. The screen fetch data from Suppliers table in database.

3. The screen shows a staff list with certain action on the same screen such as New Supplier (A1) and Update Supplier (A2).
4. The product then would be assigned to the staff who in charge the specific products. (A3)
5. The Use Case ends.

4.5.2 Alternatives Flow

A1: Create New Supplier

1. The Use Case starts when the actor clicked a button/link for creating new supplier.
2. The screen displays a form with information needed to fill up in order to create new supplier.
3. A complete form will submit for inserting process into database.
4. The Use Case ends.

A2: Update Supplier Information

1. The Use Case starts when the actor clicked on any edit button/link resides at any of the records row.
2. The screen then requests a supplier info from database based on selected supplier from the list earlier.
3. After changes made by the actor, the latest info will be submitting and save the updates to database depends on which supplier is currently being updated.
4. The Use Case ends.

A3: Assigning Staff Product In Charged

1. The Use Case starts after the actor add new product to the existing list.
2. The screen would retrieve a list of staff from Staff table in database.
3. Then, the actor must choose who's in charge by using pairing assignment model.
4. The actor would submit the pairing once the assignment settled.
5. The pairing would be store in the database.
6. The Use Case ends.

4.5.3 Exceptional Flow

Not Applicable

4.6 <<include>> Use Cases

Not Applicable

4.7 <<extend>> Use Cases

Not Applicable

4.8 Rules

Not Applicable

4.9 Constraint

Not Applicable

4.10 Comments

Not Applicable

5 Manage Stocks

5.1 Description

The use case describes the process for the actor to manage stocks of Tupperware for inventory monitoring and sales.

5.2 Pre Condition

The actor must have been authenticated.

The actor was Inventory Department staff.

5.3 Post Condition

The actor able to view product list or create new product or update product details or delete any existing product. The products have been categorised to their own category.

5.4 Actor

Inventory Staff

5.5 Flow of Events

5.5.1 Primary Flow

P: View Product List

1. The Use Case starts when the actor navigated to the List of Product Screen from the Navigation Menu.
2. The screen fetch data from Product table in database.
3. The screen shows a job list with certain action on the same screen such as Create New Product (A1), Update Product (A2) and Delete Product (A3).
4. The actor may also utilize a function to filter the product list by category.
(A4)

5. The Use Case ends.

5.5.2 Alternatives Flow

A1: Create New Product

1. The Use Case starts when the actor clicked a button/link for creating new product.
2. A list of product category loads from database.
3. The screen displays a form with information needed to fill up in order to create new product.
4. A complete form will submit for inserting process into database.
5. The Use Case ends.

A2: Update Product Information

1. The Use Case starts when the actor clicked on any edit button/link resides at any of the records row.
2. The screen then requests a product information from database based on the selected product to update from the list early.
3. Then the info retrieved will be fill up on the form fields.
4. After changes made by the actor, the latest info will be submitting and save the updates to database depends on which product is currently being updated.
5. The Use Case ends.

A3: Remove Product

1. The Use Case starts when the actor clicked on any remove/delete button/link on any row the actor wants to delete.

2. The application would validate the actor either want to proceed with removing the product record or cancel the process (E1).
3. The product would be delete if the actor chose to continue to remove it.
4. The Use Case ends.

5.5.3 Exceptional Flow

E1: Job Remove Cancellation

1. The Use Case starts when the actor wanted to remove a job.
2. The process cancelled when the actor denies to continue with the deletion process.
3. The Use Case ends.

5.6 <<include>> Use Cases

Manage Staff Use Case

5.7 <<extend>> Use Cases

Not Applicable

5.8 Rules

Not Applicable

5.9 Constraint

Not Applicable

5.10 Comments

Not Applicable

6 Manage Clone Entities

6.1 Description

The use case describes the process for the actor to manage clone entities. Clone entities means, all table in the other departments would have a copy such them in the IT Admin database.

6.2 Pre Condition

The changes happen in any of the source table.

6.3 Post Condition

The actor able to see all changes from any source tables would also applied to the IT Admin database too.

6.4 Actor

Human Resource Staff

Inventory Staff

IT Admin Staff

Sales and Marketing Staff

6.5 Flow of Events

6.5.1 Primary Flow

P: View Any Cloned Tables

1. The Use Case starts when the actor need to review any clone table.
2. The screen fetch data from any table as the actor request from database.
3. The data inside all these table donate by all the source table. (A1)

4. The Use Case ends.

6.5.2 Alternatives Flow

A1: Outsource Donate Data

1. The Use Case starts when the actor established a database link.
2. The changes, (insert, update, delete) happened in the source table may try to execute the same changes at the clone tables.
3. The Use Case ends.

6.5.3 Exceptional Flow

Not Applicable

6.6 <<include>> Use Cases

Not Applicable

6.7 <<extend>> Use Cases

Not Applicable

6.8 Rules

Not Applicable

6.9 Constraint

Not Applicable

6.10 Comments

Not Applicable

7 Manage Database Backup and Recovery

7.1 Description

The use case describes the process for the actor to manage database backup and recovery. The backup and recovery happened implicitly and explicitly. These could be done using two types of backup and recovery, full-backup and incremental backup.

7.2 Pre Condition

The actor does not take any action on the database.

7.3 Post Condition

The actor able to backup or recovery any tables in database.

7.4 Actor

Inventory Staff

IT Admin Staff

7.5 Flow of Events

7.5.1 Primary Flow

P: Full-Backup

1. The Use Case starts when the actor need to backup any table.
2. The application fetches all data from database as requested by the actor.
3. The data fetched stored in csv file. Then it can use to restore when it is needed. (A1)
4. The audit table (A2) may flush after the full backup executed by the actor.
5. The Use Case ends.

7.5.2 Alternatives Flow

A1: Full-Recovery

1. The Use Case starts when any possible issue happens on any table.
2. The actor need to load the latest backup file to recover any lost.
3. The table then should be restored as the last time the table was backup.
4. The Use Case ends.

A2: Incremental Backup

1. The Use Case starts when any changes made on any row of any table.
2. The changes then stored automatically inside an audit table with the state of changes such as insert, update, delete.
3. The data inside the audit table may be useful during Incremental Recovery later on. (A3)
4. The Use Case ends.

A3: Incremental Recovery

1. The Use Case starts when any possible issue happens on any of table in the database.
2. The actor must choose the latest full backup for any issue table.
3. After the table recover up to the last backup, the application will refer to the audit table to recover the data up to the latest changes happened before the issue.
4. The Use Case ends.

7.5.3 Exceptional Flow

Not Applicable

7.6 <<include>> Use Cases

Not Applicable

7.7 <<extend>> Use Cases

Not Applicable

7.8 Rules

Not Applicable

7.9 Constraint

To execute incremental recovery, full-backup of the table in issued must have first. If not, the incremental backup is useless.

7.10 Comments

Not Applicable

8 Manage Application Report

8.1 Description

The use case describes the process for the actor to manage application report. The actor may see the data represented in graphics such as graph or chart. The actor also able to get all those data including the diagram in PDF format too.

8.2 Pre Condition

The actor opens a screen to review data or details of any clone table.

8.3 Post Condition

The actor able to review the data inside clone table and review diagram of the chosen clone table. The actor also able to get a copy of data in pdf format containing data and diagrams of the selected clone table.

8.4 Actor

IT Admin Staff

8.5 Flow of Events

8.5.1 Primary Flow

P: Review Clone Table Data

1. The Use Case starts when the actor need to review data in the database.
2. The records retrieve from database for display.
3. The application displays the records on the screen.
4. On the same screen, user able to get a pdf copy for the data.
5. The Use Case ends.

8.5.2 Alternatives Flow

A1: Generate PDF

1. The Use Case starts when the actor clicked on button/link for generating pdf file.
2. The data for the selected clone table load to the screen.
3. The data is display on it including a diagram that describe the data inside the table.
4. The screen is convert by a function provided by the programming language to generate the screen into pdf format.
5. The Use Case ends.

8.5.3 Exceptional Flow

Not Applicable

8.6 <<include>> Use Cases

Not Applicable

8.7 <<extend>> Use Cases

Not Applicable

8.8 Rules

Not Applicable

8.9 Constraint

Not Applicable

8.10 Comments

Not Applicable

9 Manage Clients

9.1 Description

The use case describes the process for the actor to manage clients. The manage consist of login for existing user and registration for any user.

9.2 Pre Condition

The actor came to TupperShop Web Application

9.3 Post Condition

The actor able to login to the system or register if he/she never registered.

9.4 Actor

Sales and Marketing Staff

Public Guest

9.5 Flow of Events

9.5.1 Primary Flow

P: Login Client

1. The Use Case starts when the actor navigated to login page of the application or after the user want to check out with their cart.
2. The screen shows information needed such as email and password on the screen.
3. If the actor never registers, he/she may click on registration link. (A1)
4. After the actor key in all information needed and submit it.
5. The application would check the actor validity then authenticated him/her to proceed to the next page.
6. The Use Case ends.

9.5.2 Alternatives Flow

A1: Registration

1. The Use Case starts when the actor clicked a link for registration.
2. The screen displays a form with some information needed to fill up in order to register as a new client.
3. A complete form will submit for inserting process into the database.

4. The Use Case ends.

9.5.3 Exceptional Flow

Not Applicable

9.6 <<include>> Use Cases

Not Applicable

9.7 <<extend>> Use Cases

Not Applicable

9.8 Rules

Not Applicable

9.9 Constraint

The email address should be entered correctly.

9.10 Comments

Not Applicable

10 Manage Online Purchasing

10.1 Description

The use case describes the process for the actor to manage online purchasing. The application offers a list of products that the actor able to buy. The actor may proceed to check out directly or buy multiple item at once. Then, the staff should be able to updates the information of the order made by the clients.

10.2 Pre Condition

The actor came to TupperShop Web Application and make new order or he/she want to review the past orders.

The actor wants to update the order status.

10.3 Post Condition

The order made by the actor successfully stored in database.

The status changes.

10.4 Actor

Sales and Marketing Staff

Public Guest

10.5 Flow of Events

10.5.1 Primary Flow

P: Add to My Cart

1. The Use Case starts when the actor navigated to product pages.
2. The screen will load data from product table to display on the product page.
3. The actor may see the product name, short description, price and the availability of the product.
4. After the actor made decision, he/she can buy it using add to cart button or if she want to directly purchase it, they can directly proceed it to check out.
5. If the user authenticated (9.5.1 P), the order will be store into database.
6. The Use Case ends.

10.5.2 Alternatives Flow

A1: Status Update

1. The Use Case starts when the actor wanted to update order status. The job could be done for staff/client.
2. After authenticated, the actor able to review the order details together with the order status. For status update, the application would determine by itself where the user current state for ordering process. Then, the button would be dynamically change depends on the user order's status.
3. When the actor updates the status, the application will submit a changes to database.
4. The Use Case ends.

10.5.3 Exceptional Flow

Not Applicable

10.6 <<include>> Use Cases

Not Applicable

10.7 <<extend>> Use Cases

Not Applicable

10.8 Rules

Not Applicable

10.9 Constraint

Not Applicable

10.10 Comments

Not Applicable

3.4 Conclusion

In conclusion, this chapter require us to really understand the problem that currently happen in the real application that exist nowadays. From there, we could improve with the idea for solving the problems.

The current system workflow we shown by using flow chart as above. That is the general workflow that happen in almost any application out there.

As we plan to develop our application using OOP. Instead using DFD, flow chart or any model else, we use UML to describe our application a little bit deeper. As we can see, we use Use Case Diagram to come out with the foundation idea for the application. Then, the detailed description is appending together to describe the process could happen for each Use Case. By reading the description, we could understand the behind story of the application that we wanted to develop.

CHAPTER IV

DATABASE DESIGN

4.1 Introduction

A database usually is a main component of the information system, then the process of database design is important in system development. The process of database design brings up the idea of data models. Data models represent different data level requirements in the form of diagrams or schemas. There are three phases of database design which are conceptual database design, logical database design and physical database design.

First and foremost, entities, attributes, relationship types are identified in the phase of conceptual database design. The product of this phase will be the entity relationship diagram (ERD). ERD is a diagram which represent all the physical considerations and it act as the conceptual data model which is the high-level overview of the database system.

The second phase of database system design is logical database design. Logical architecture of the database system is generated during this phase. Basically, data dictionary of every entity in the database system contains logical components of the database system. This logical data model describe how the data is stored into database. It will provide more information about the database system such as the data type of attribute, existence of primary key or foreign key, description of attribute and others.

Physical database design is the last phase of database system design. The description of implementation is generated during physical database design. This phase describe the base relations, file organization, indexes, integrity constraints, security and others. SQL language is used to build up the database.

4.2 Conceptual database design

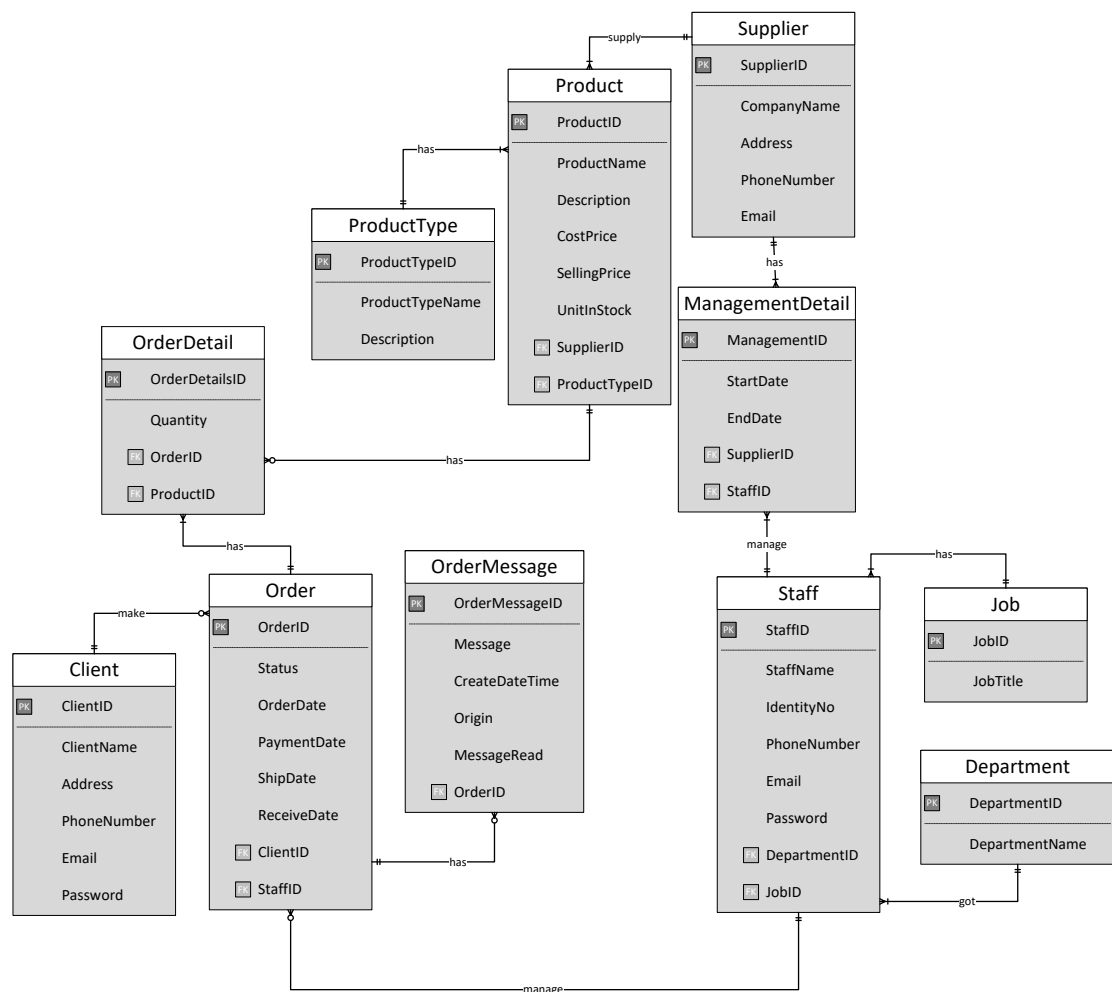


Figure 4.1: Entity Relationship Diagram (ERD)

4.3 Logical database design

a) PRODUCTTYPE

Products can be categorised into different product types.

Table 4.1: Data Dictionary of table PRODUCTTYPE

No	Attribute	Data Type	PK/ FK	Contents
1	ProductTypeID	VARCHAR (10)	PK	Product type identification number
2	ProductTypeName	VARCHAR (45)		Product type name
3	Description	VARCHAR (80)		Product type description

b) SUPPLIER

Information of every supplier of the product is stored for better communication.

Table 4.2: Data Dictionary of table SUPPLIER

No	Attribute	Data Type	PK/ FK	Contents
1	SupplierID	VARCHAR (10)	PK	Supplier identification number
2	CompanyName	VARCHAR (45)		Supplier company name
3	Address	VARCHAR (80)		Supplier company address
4	PhoneNumber	VARCHAR (15)		Supplier contact number
5	Email	VARCHAR (45)		Supplier email

c) PRODUCT

Details of the product is stored and keep track on the quantity in hand.

Table 4.3: Data Dictionary of table PRODUCT

No	Attribute	Data Type	PK/ FK	Contents
1	ProductID	VARCHAR (10)	PK	Product identification number
2	ProductName	VARCHAR (45)		Product name
3	Description	VARCHAR (150)		Product description
4	CostPrice	DECIMAL(10,2)		Product cost price
5	SellingPrice	DECIMAL(10,2)		Product selling price
6	UnitInStock	INT		Product unit in stock
7	SupplierID	VARCHAR (10)	FK	Supplier identification number
8	ProductTypeID	VARCHAR (10)	FK	Product type identification number

d) JOB

Every staff will be assigned to a job title.

Table 4.4: Data Dictionary of table JOB

No	Attribute	Data Type	PK/ FK	Contents
1	JobID	VARCHAR (10)	PK	Job identification number
2	JobTitle	VARCHAR (20)		Job title

e) DEPARTMENT

Every staff must belongs to a department.

Table 4.5: Data Dictionary of table DEPARTMENT

No	Attribute	Data Type	PK/ FK	Contents
1	DepartmentID	VARCHAR (10)	PK	Department identification number
2	DepartmentName	VARCHAR (30)		Department name

f) STAFF

Personal details of every staff is stored in this table.

Table 4.6: Data Dictionary of table STAFF

No	Attribute	Data Type	PK/ FK	Contents
1	StaffID	VARCHAR (10)	PK	Staff identification number
2	StaffName	VARCHAR (45)		Staff name
3	IdentityNo	VARCHAR (14)		Staff identity card number
4	PhoneNumber	VARCHAR (15)		Staff contact number
5	Email	VARCHAR (45)		Staff login email
6	Password	VARCHAR (255)		Staff login password
7	DepartmentID	VARCHAR (10)	FK	Department identification number
8	JobID	VARCHAR (10)	FK	Job identification number

g) MANAGEMENTDETAIL

Management detail is about the supplier management. The staff who handle which supplier can be determined in this table.

Table 4.7: Data Dictionary of table MANAGEMENTDETAIL

No	Attribute	Data Type	PK/ FK	Contents
1	ManagementID	VARCHAR (10)	PK	Management identification number
2	StartDate	DATE		Management start date
3	EndDate	DATE		Management end date
4	SupplierID	VARCHAR (10)	FK	Supplier identification number
5	StaffID	VARCHAR (10)	FK	Staff identification number

h) CLIENT

All client's details is stored in this table.

Table 4.8: Data Dictionary of table CLIENT

No	Attribute	Data Type	PK/ FK	Contents
1	ClientID	VARCHAR (10)	PK	Client identification number
2	ClientName	VARCHAR (45)		Client name
3	Address	VARCHAR (80)		Client address
4	PhoneNumber	VARCHAR (15)		Client contact number

5	Email	VARCHAR (45)		Client login email
6	Password	VARCHAR (255)		Client login password

i) ORDER

Every single order record will be stored in this table.

Table 4.9: Data Dictionary of table ORDER

No	Attribute	Data Type	PK/ FK	Contents
1	OrderID	VARCHAR (10)	PK	Order identification number
2	Status	VARCHAR (20)		Order status
3	OrderDate	DATE		Order date
4	PaymentDate	DATE		Payment date
5	ShipDate	DATE		Shipping date
6	ReceiveDate	DATE		Receive date
7	ClientID	VARCHAR (10)	FK	Client identification number
8	StaffID	VARCHAR (10)	FK	Staff identification number

j) ORDERDETAIL

The exact quantity of the product from order will be stored in this table.

Table 4.10: Data Dictionary of table ORDERDETAIL

No	Attribute	Data Type	PK/ FK	Contents
1	OrderDetailsID	VARCHAR (10)	PK	Order detail identification number

2	Quantity	INT		Quantity of product ordered
3	OrderID	VARCHAR (10)	FK	Order identification number
4	ProductID	VARCHAR (10)	FK	Product identification number

k) ORDERMESSAGE

Client can send message to organization through the system and all the contents is stored in this table.

Table 4.11: Data Dictionary of table ORDERMESSAGE

No	Attribute	Data Type	PK/ FK	Contents
1	OrderMessageID	VARCHAR (10)	PK	Order message identification number
2	Message	VARCHAR (80)		Message from clients
3	CreateDateTime	DATETIME		Date and time of message created from clients
4	Origin	VARCHAR (80)		Details of origin
5	MessageRead	VARCHAR (10)		Status of message
6	OrderID	VARCHAR (10)	FK	Order identification number

4.4 Physical database design

a) PRODUCTTYPE

Table 4.12: Data Definition Language for creating table PRODUCTTYPE

PRODUCTTYPE
<pre>CREATE TABLE `workshop2`.`producttype` (`productTypeID` VARCHAR(10) NOT NULL, `productName` VARCHAR(45) NOT NULL, `description` VARCHAR(80) NOT NULL, PRIMARY KEY (`productTypeID`), UNIQUE INDEX `productTypeID_UNIQUE` (`productTypeID` ASC));</pre>

b) SUPPLIER

Table 4.13: Data Definition Language for creating table SUPPLIER

SUPPLIER
<pre>CREATE TABLE `workshop2`.`supplier` (`supplierID` VARCHAR(10) NOT NULL, `companyName` VARCHAR(45) NOT NULL, `address` VARCHAR(80) NOT NULL, `phoneNumber` VARCHAR(15) NOT NULL, `email` VARCHAR(45) NOT NULL, PRIMARY KEY (`supplierID`), UNIQUE INDEX `supplierID_UNIQUE` (`supplierID` ASC));</pre>

c) PRODUCT

Table 4.14: Data Definition Language for creating table PRODUCT

PRODUCT
<pre>CREATE TABLE `workshop2`.`product` (`productID` VARCHAR(10) NOT NULL, `productName` VARCHAR(45) NOT NULL, `description` VARCHAR(150) NOT NULL, `costPrice` DECIMAL(10,2) NOT NULL, `sellingPrice` DECIMAL(10,2) NOT NULL, `unitInStock` INT NOT NULL, `supplierID` VARCHAR(10) NOT NULL, `productTypeID` VARCHAR(10) NOT NULL, PRIMARY KEY (`productID`),</pre>

```

UNIQUE INDEX `productID_UNIQUE` (`productID` ASC),
INDEX `productTypeID_idx` (`productTypeID` ASC),
INDEX `supplierID_idx` (`supplierID` ASC),
CONSTRAINT `supplierID` FOREIGN KEY (`supplierID`)
REFERENCES `workshop2`.`supplier` (`supplierID`),
CONSTRAINT `productTypeID` FOREIGN KEY (`productTypeID`)
REFERENCES `workshop2`.`producttype` (`productTypeID`));

```

d) JOB

Table 4.15: Data Definition Language for creating table JOB

JOB
<pre> CREATE TABLE `workshop2`.`job` (`jobID` VARCHAR(10) NOT NULL, `jobTitle` VARCHAR(20) NOT NULL, PRIMARY KEY (`jobID`), UNIQUE INDEX `jobID_UNIQUE` (`jobID` ASC)); </pre>

e) DEPARTMENT

Table 4.16: Data Definition Language for creating table DEPARTMENT

DEPARTMENT
<pre> CREATE TABLE `workshop2`.`department` (`departmentID` VARCHAR(10) NOT NULL, `departmentName` VARCHAR(30) NOT NULL, PRIMARY KEY (`departmentID`), UNIQUE INDEX `departmentID_UNIQUE` (`departmentID` ASC)); </pre>

f) STAFF

Table 4.17: Data Definition Language for creating table STAFF

STAFF
<pre> CREATE TABLE `workshop2`.`staff` (`staffID` VARCHAR(10) NOT NULL, `staffName` VARCHAR(45) NOT NULL, `identityNo` VARCHAR(14) NOT NULL, `phoneNumber` VARCHAR(15) NOT NULL, `email` VARCHAR(45) NOT NULL, `password` VARCHAR(255) NOT NULL, `departmentID` VARCHAR(10) NOT NULL, </pre>

```

`jobID` VARCHAR(10) NOT NULL,
PRIMARY KEY (`staffID`),
UNIQUE INDEX `ATTRIBUTES_UNIQUE` (`staffID`,`email` ASC)
INDEX `departmentID_idx` (`departmentID` ASC),
INDEX `jobID_idx` (`jobID` ASC),
CONSTRAINT `jobID` FOREIGN KEY (`jobID`)
REFERENCES `workshop2`.`job` (`jobID`),
CONSTRAINT `departmentID` FOREIGN KEY (`departmentID`)
REFERENCES `workshop2`.`department` (`departmentID`));

```

g) MANAGEMENTDETAIL

Table 4.18: Data Definition Language for creating table

MANAGEMENTDETAIL

MANAGEMENTDETAIL
<pre> CREATE TABLE `workshop2`.`managementdetail` (`managementID` VARCHAR(10) NOT NULL, `startDate` DATE NULL, `endDate` DATE NULL, `supplierID` VARCHAR(10) NULL, `staffID` VARCHAR(10) NULL, PRIMARY KEY (`managementID`), UNIQUE INDEX `managementID_UNIQUE` (`managementID` ASC) INDEX `supplierID_idx` (`supplierID` ASC), INDEX `staffID_idx` (`staffID` ASC), CONSTRAINT `supplierID` FOREIGN KEY (`supplierID`) REFERENCES `workshop2`.`supplier` (`supplierID`), CONSTRAINT `staffID` FOREIGN KEY (`staffID`) REFERENCES `workshop2`.`staff` (`staffID`)); </pre>

h) CLIENT

Table 4.19: Data Definition Language for creating table CLIENT

CLIENT
<pre> CREATE TABLE `workshop2`.`client` (`clientID` VARCHAR(10) NOT NULL, `clientName` VARCHAR(45) NOT NULL, `address` VARCHAR(80) NOT NULL, `phoneNumber` VARCHAR(15) NOT NULL, `email` VARCHAR(45) NOT NULL, </pre>


```

`password` VARCHAR(45) NOT NULL,
PRIMARY KEY (`clientID`),
UNIQUE INDEX `clientID_UNIQUE` (`clientID` ASC));

```

i) ORDER

Table 4.20: Data Definition Language for creating table ORDER

ORDER
<pre> CREATE TABLE `workshop2`.`order` (`orderID` VARCHAR(10) NOT NULL, `clientID` VARCHAR(10) NOT NULL, `staffID` VARCHAR(10) NOT NULL, `status` VARCHAR(45), `orderDate` DATE, `paymentDate` DATE, `shipDate` DATE, `receiveDate` DATE, PRIMARY KEY (`orderID`), UNIQUE INDEX `orderID_UNIQUE` (`orderID` ASC) INDEX `clientID_idx` (`clientID` ASC), INDEX `staffID_idx` (`staffID` ASC), CONSTRAINT `clientID` FOREIGN KEY (`clientID`) REFERENCES `workshop2`.`client` (`clientID`), CONSTRAINT `staffID` FOREIGN KEY (`staffID`) REFERENCES `workshop2`.`staff` (`staffID`)); </pre>

j) ORDERDETAIL

Table 4.21: Data Definition Language for creating table ORDERDETAIL

ORDERDETAIL
<pre> CREATE TABLE `workshop2`.`orderdetail` (`orderDetailsID` VARCHAR(10) NOT NULL, `quantity` INT NULL, `orderID` VARCHAR(10) NULL, `productID` VARCHAR(10) NULL, PRIMARY KEY (`orderDetailsID`), UNIQUE INDEX `orderDetailsID_UNIQUE` (`orderDetailsID` ASC) INDEX `orderID_idx` (`orderID` ASC), INDEX `productID_idx` (`productID` ASC), CONSTRAINT `orderID` FOREIGN KEY (`orderID`) REFERENCES `workshop2`.`order` (`orderID`), </pre>

CONSTRAINT `productID` FOREIGN KEY (`productID`) REFERENCES `workshop2`.`product` (`productID`));
--

k) ORDERMESSAGE

Table 4.22: Data Definition Language for creating table ORDERMESSAGE

ORDERMESSAGE
CREATE TABLE `workshop2`.`ordermessage` (`orderMessageID` VARCHAR(10) NOT NULL, `message` VARCHAR(45) NOT NULL, `createDateTime` DATETIME NOT NULL, `origin` VARCHAR(80) NOT NULL, `messageRead` VARCHAR(45) NOT NULL, `orderID` VARCHAR(10) NOT NULL, PRIMARY KEY (`orderMessageID`), UNIQUE INDEX `orderMessageID_UNIQUE` (`orderMessageID` ASC) INDEX `orderID_idx` (`orderID` ASC), CONSTRAINT `orderID` FOREIGN KEY (`orderID`) REFERENCES `workshop2`.`order` (`orderID`));

4.5 Conclusion

In a nutshell, the process of database system design is producing the detailed data model of database. The main product of database system design is design document. Description of logical and physical design is stated in this document. This document similar with the blue print of the system then it will be used in implementation, testing and maintenance.

CHAPTER V

INDIVIDUAL DATABASE SYSTEM IMPLEMENTATION

5.1 Introduction

The installation steps of the four databases that we selected are shown in this chapter. The individual task in implementing the database design in different operating system is explained in more details too in this chapter. In the TupperShop Management System, it has several subsystems to be developed and implemented to accomplish the whole system. For instance, marketing management, staff management, inventory management and data audit management.

5.2 Database software installation

Firstly, database installation of SQL Server on Windows which in charge in marketing management of this workshop 2. Secondly, database installation of Oracle 11g on Windows which in charge in staff management of this workshop 2. Thirdly, database installation of PostgreSQL on Ubuntu which in charge in inventory management of this workshop 2. Lastly, database installation of MySQL on Windows which in charge in data audit management of this system.

5.2.1 Installation steps of SQL Server on Windows

Step 1: Upon starting the installation, you will get this window:



Figure 5.1: Installation first page

Step 2: Choose the Installation section in order to start the installation.



Figure 5.2: Product key insertion

Step 3: Mark the check box in order to use Microsoft Update to check for an update after you finished the installation, and then click Next.

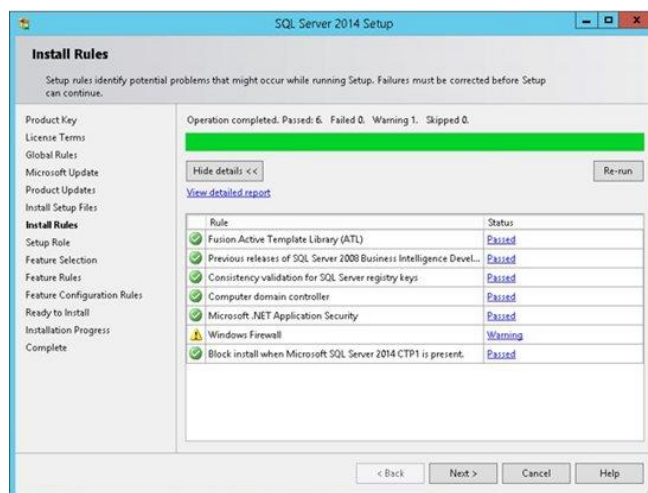


Figure 5.3: Page of install rules

Step 4: The SQL Server installation program checks your machine to make sure it meets the hardware and software requirements to install SQL Server. If you get any Errors in the results, please use the link in the error message to get more information.

Do not move to the next step if you get any errors in this report.

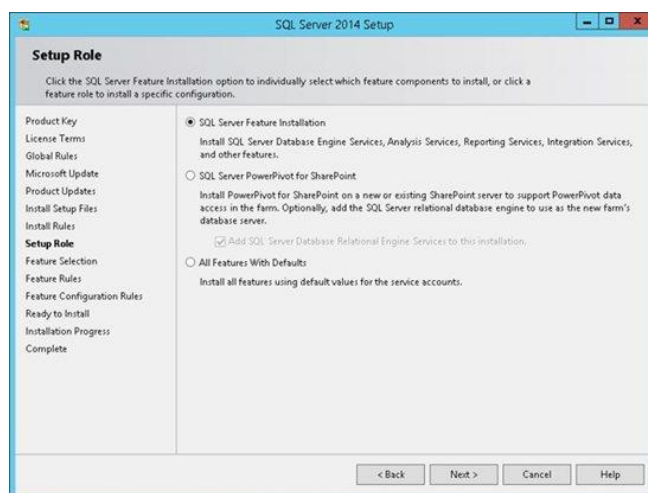


Figure 5.4: Page of setup role

Step 5: Choose the first option in order to install SQL Server, and then select which features to install. We are going to choose our features manually in the next step. Click Next to continue.

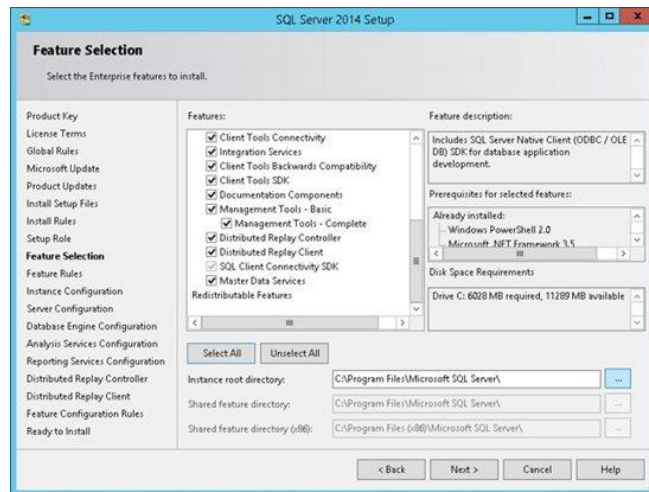


Figure 5.5: Page of feature selection

Step 6: You can choose what features to install. It is highly recommend NOT to choose all the features on a production server if you don't need them. Choose only the features that you need or might need. Conversely you might want to select all using a development server in order to give you more flexibility in your development environment. Once you choose to use another feature you will be able to add it in production later on.

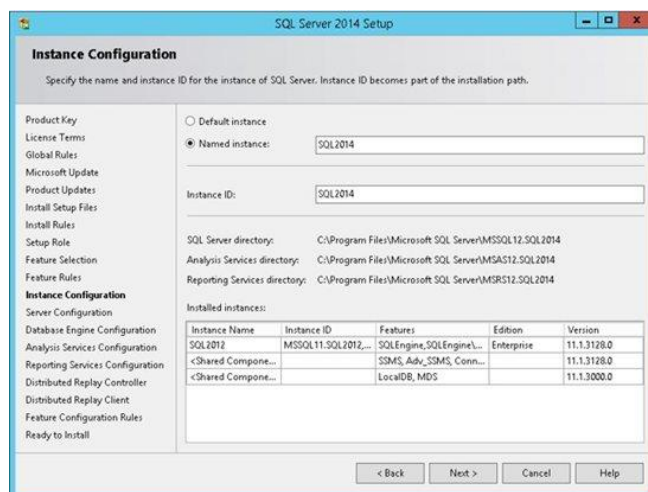


Figure 5.6: Page of instance configuration

Step 7: SQL Server allows you to install multiple instances of the same version or different versions of the server. Make sure you choose a name that will help you in the future to recognize the instance which you are working with. Theoretically you can use any instance name that you want.

*** Remember the name of the instance!**

You will need to use this name every time that you want to connect to the server.

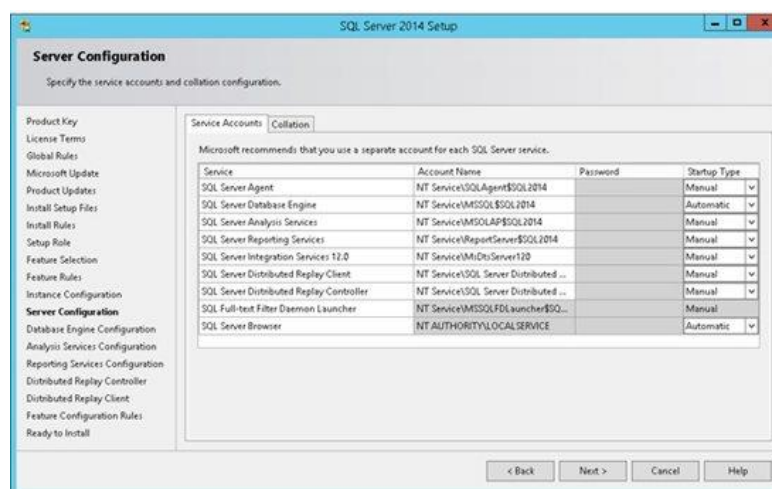


Figure 5.7: Page of server configuration

Step 8: Security is important for every product and every business. By following simple best practices, you can avoid many security vulnerabilities. SQL Server works as a group of services in the background. In this step you can select the service accounts for the SQL Server actions. Each service in SQL Server represents a process or a set of processes to manage authentication of SQL Server operations with Windows. Each service can be configured to use its own service account. Once you configure this step move to the collation tab.

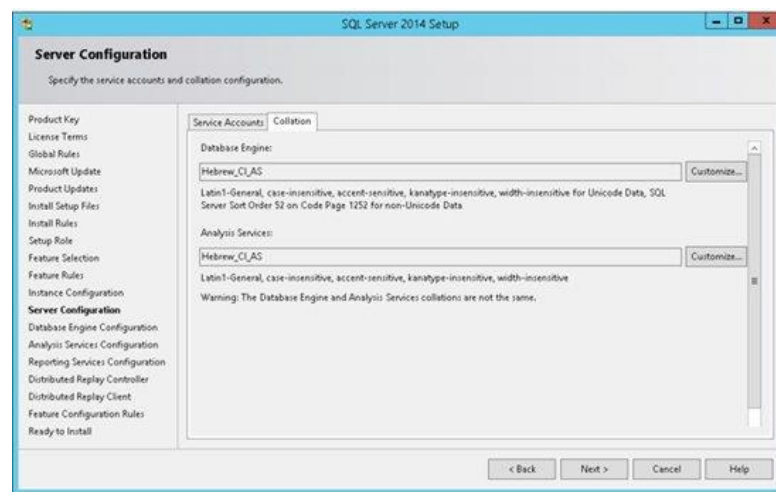


Figure 5.8: Collation tab in page of server configuration

Step 9: SQL Server supports several collations. A collation encodes the rules governing the proper use of characters for either a language, such as Hebrew or Polish, or an alphabet, such as Latin1_General (the Latin alphabet used by western European languages). Typically, you should choose a SQL Server collation that supports most languages commonly used by users at your organization. Select the collation and press next.

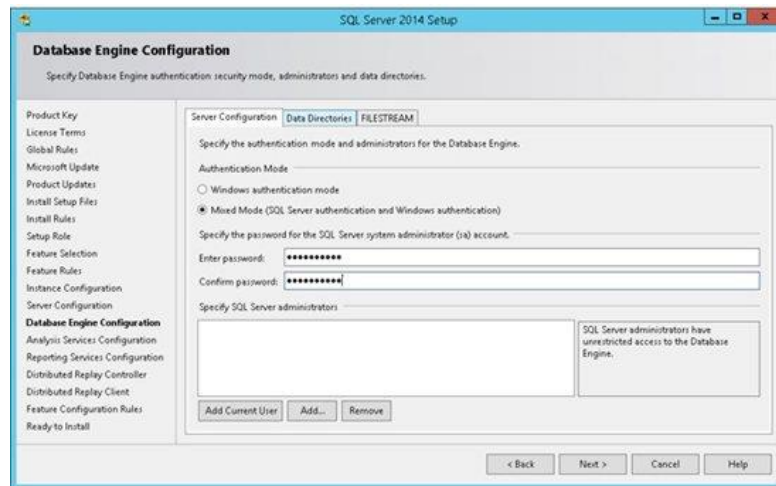


Figure 5.9: Page of database engine configuration

Step 10: FILESTREAM enables SQL Server-based applications to store unstructured data, such as documents and images, on the file system. FILESTREAM integrates the SQL Server Database Engine with an NTFS file system by storing varbinary(max) binary large object (BLOB) data as files on the file system.

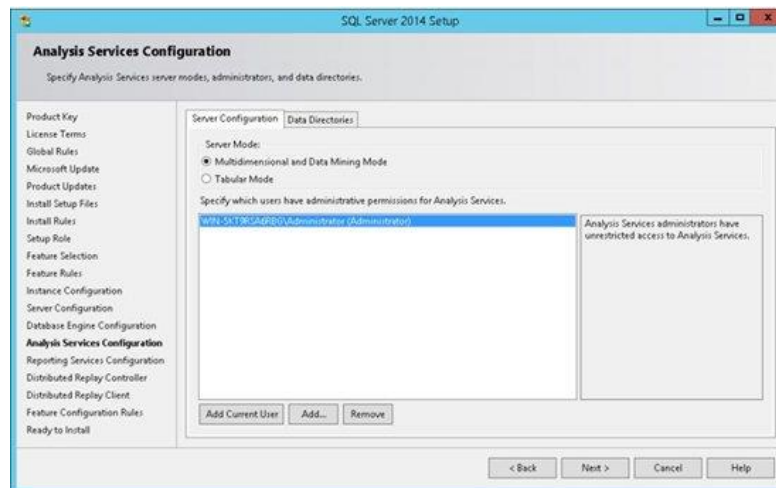


Figure 5.10: Page of analysis services configuration

Step 11: As mentioned in step 13 the data directories can have significant influence on the server performance. Press Next to continue the installation configuration.

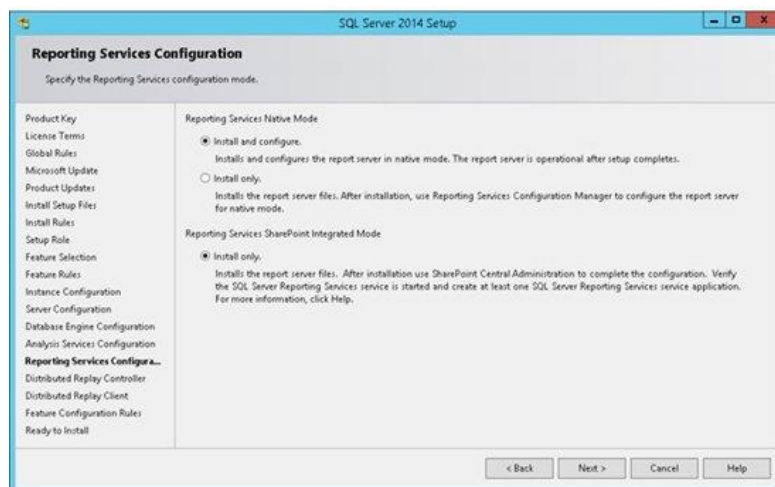


Figure 5.11: Page of reporting services configuration

Step 12: Choose if you want to install and configure the Reporting Service or just Install, which mean you will need to configure it later on.

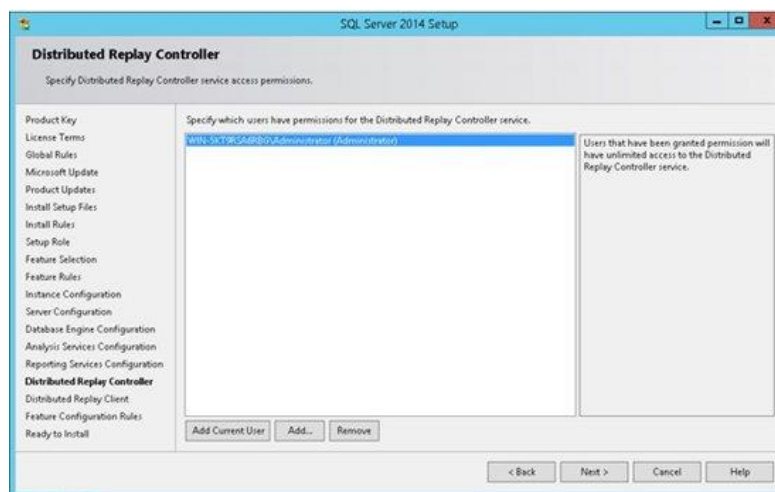


Figure 5.12: Page of distributed replay controller

Step 13: That is all :-). The SQL Server Installation program will show us a report of all our configuration (steps 1-19). Once you select Next, the installation will start.

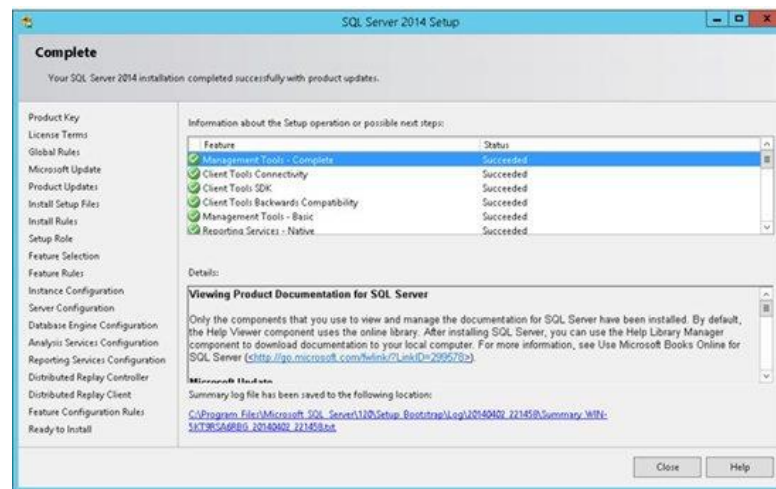


Figure 5.13: Page of installation complete

Connecting to the Server

Open the new SSMS application which we have just installed (in step 8 we choose which features to install, if you have chosen Management Tools it includes the SQL Server Management Studio).



Figure 5.14: Login into Microsoft SQL Server 2014

5.2.2 Installation steps of Oracle 11g on Windows

Step 1: Open Oracle 11g installer, fill in the form and follow the next instruction.

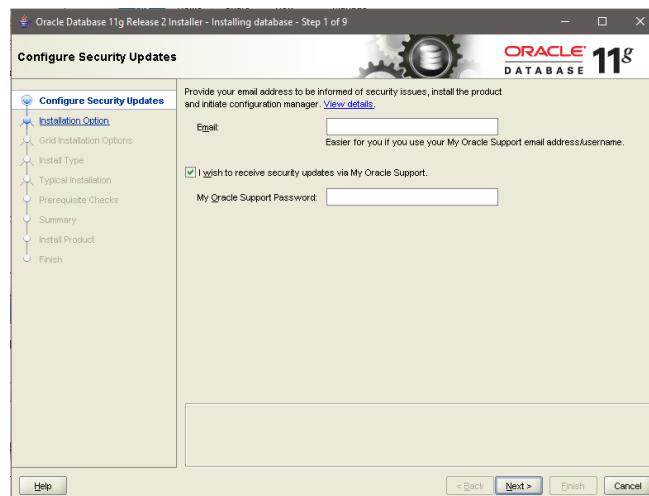


Figure 5.15: Installation process

Step 2: Select install database software only, then click ‘Next’ until installation complete.

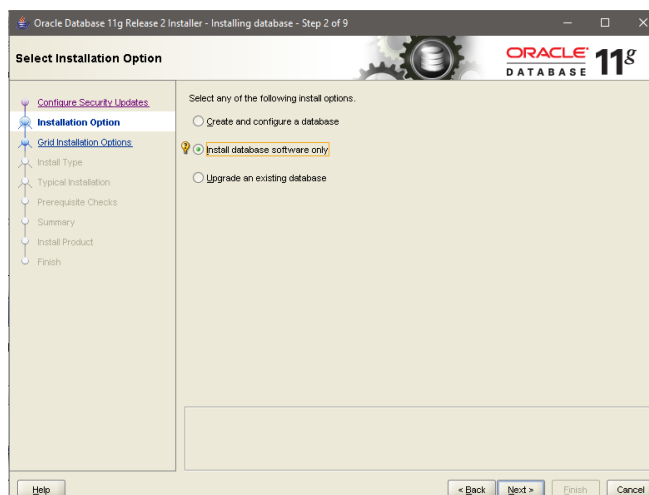


Figure 5.16: Installation choice

Step 3: After complete the installation, the database need to configure listener. Find *Net configuration assistant* start menu. Choose Listener configuration then choose add new and follow the instruction until finish.

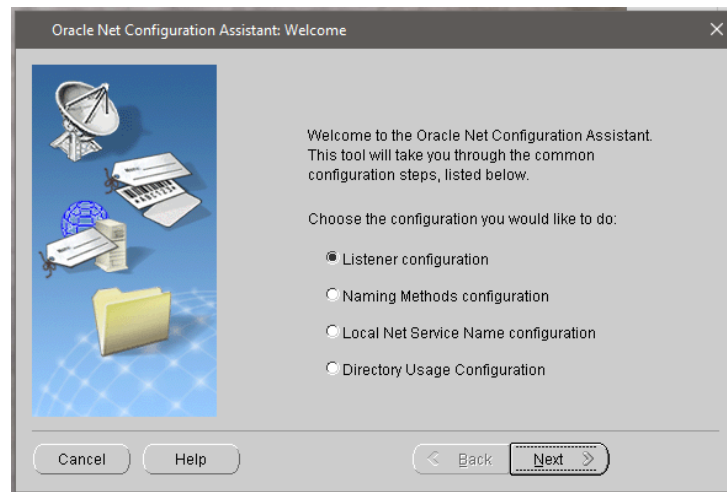


Figure 5.17: Listener configuration

Step 4: After configuring the listener, open *Database configuration assistant* to start creating the database. Select *Create a Database* and follow the instruction until its finish.

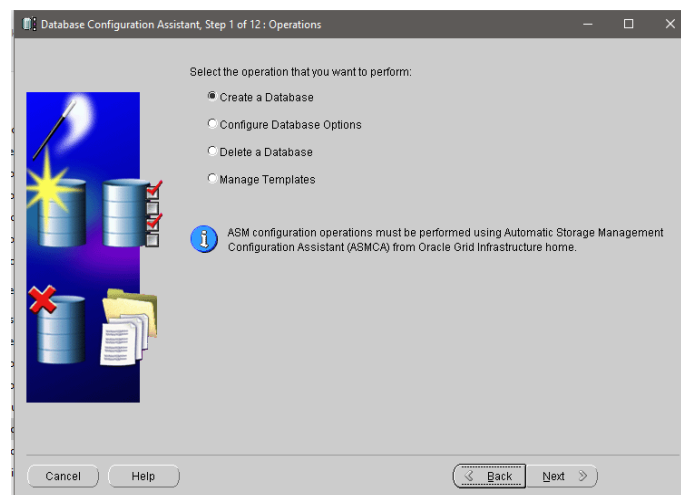


Figure 5.18: Database creation

Step 5: Open command prompt as Administrator, type **SQLPLUS /NOLOG** to start creating the table and development.

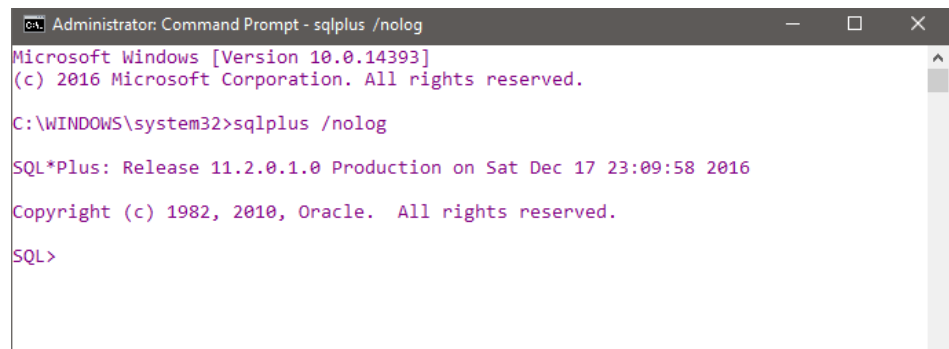


Figure 5.19: CMD command

Step 6: This step is how to install and configure Oracle 11g to heterogeneous link to MYSQL. Download and install **MYSQL ODBC DRIVER**. Open ODBC Data Source Administrator corresponding to what version of Oracle 11g you are. Open **System DSN** and click **Add** and find MYSQL option to create new data source.

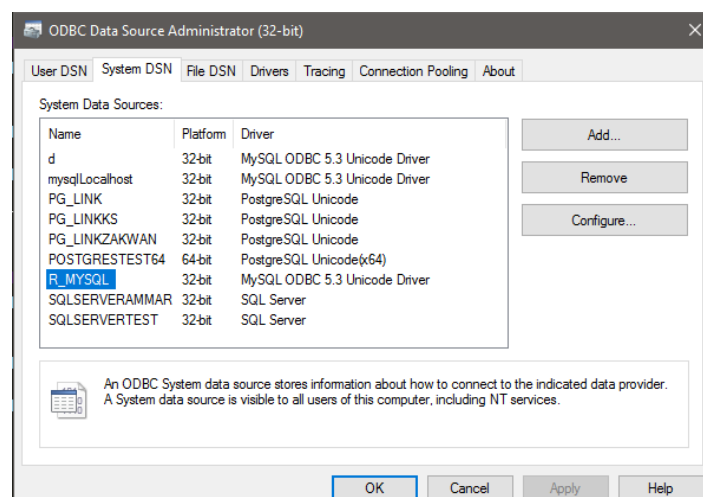


Figure 5.20: ODBC setup

Step 7: This step for configuring ODBC details. Fills in the detail such as **Data Source Name**, **TCP/IP SERVER**, **User**, **Password** and **Database**. **Data source Name** is used for next configuration and need named carefully. **TCP/IP SERVER** is used for source **ip address** or **pc name**. **User** is for MYSQL username. **Password** is for MYSQL password. Click test after fill all the details and notification of successful connection will be prompted. Then choose the **Database** from the list given.

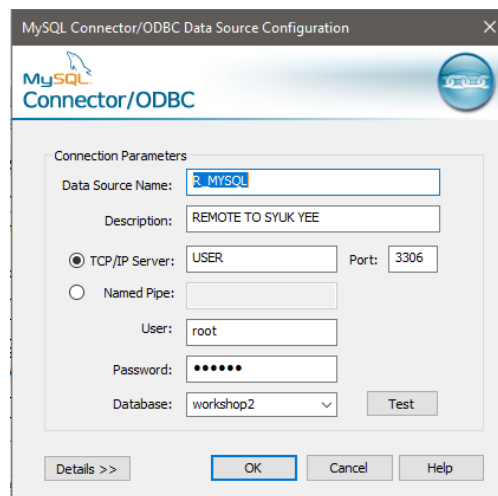
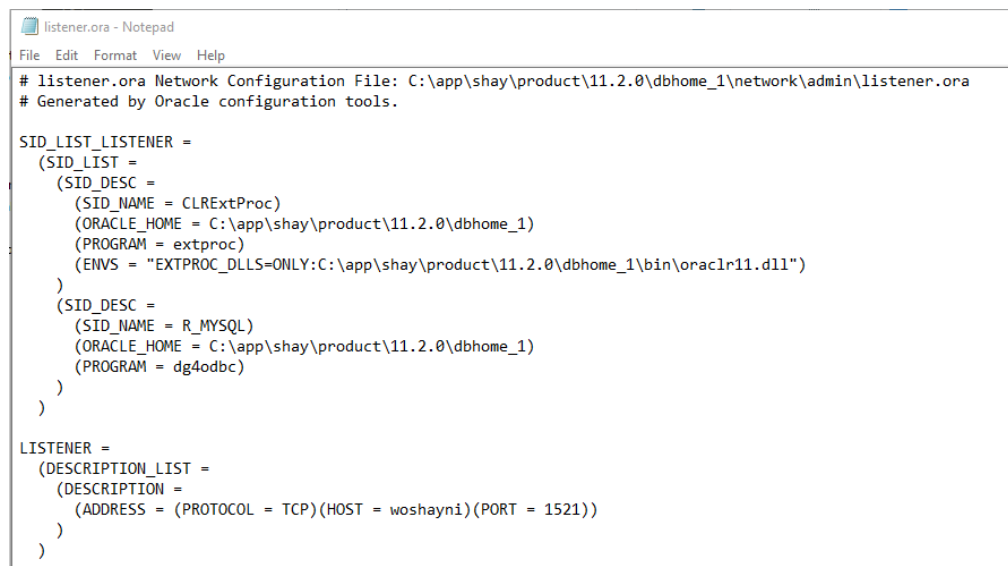


Figure 5.21: ODBC details

Step 8: This Step is for configuring *Listener.ora* which is most one of the most importance part. Go to directory of Oracle 11g installation file, which is *C:\app\shay\product\11.2.0\dbhome_1\NETWORK\ADMIN*. Open *listener.ora* and add under the *SID_LIST_LISTENER*

```
(SID_DESC =
    (SID_NAME = R_MYSQL)
    (ORACLE_HOME = C:\app\shay\product\11.2.0\dbhome_1)
    (PROGRAM = dg4odbc)
)
```

Note that *SID_NAME* is name of data source that has been configure earlier.



```
listener.ora - Notepad
File Edit Format View Help
# listener.ora Network Configuration File: C:\app\shay\product\11.2.0\dbhome_1\network\admin\listener.ora
# Generated by Oracle configuration tools.

SID_LIST_LISTENER =
(SID_LIST =
  (SID_DESC =
    (SID_NAME = CLRExtProc)
    (ORACLE_HOME = C:\app\shay\product\11.2.0\dbhome_1)
    (PROGRAM = extproc)
    (ENVS = "EXTPROC_DLLS=ONLY:C:\app\shay\product\11.2.0\dbhome_1\bin\oraclr11.dll")
  )
  (SID_DESC =
    (SID_NAME = R_MYSQL)
    (ORACLE_HOME = C:\app\shay\product\11.2.0\dbhome_1)
    (PROGRAM = dg4odbc)
  )
)

LISTENER =
(DESCRIPTION_LIST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = woshayni)(PORT = 1521))
  )
)
```

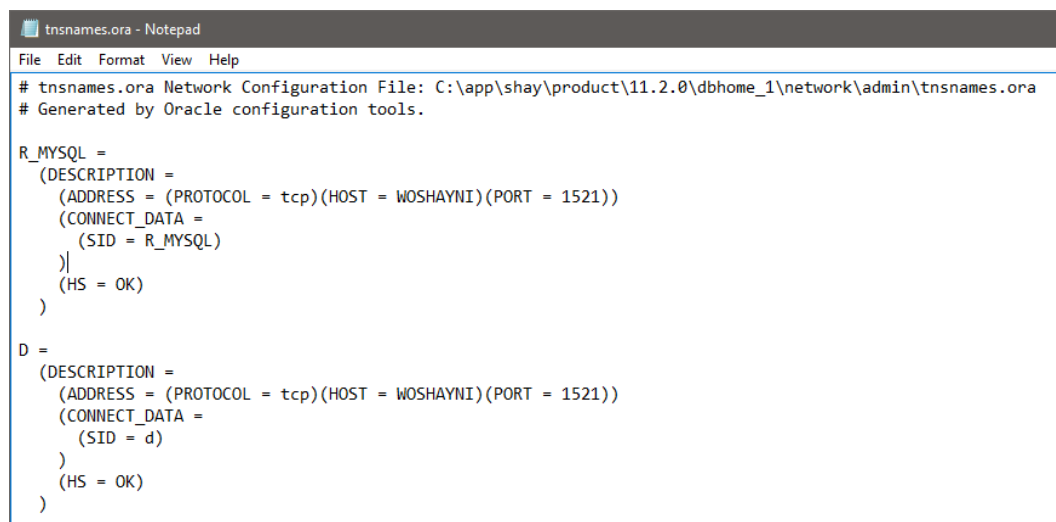
Figure 5.22: Configuration of listener.ora file

Step 9: This step is for configuring *Tnsname.ora* which is one of the importance part.

Go to directory of Oracle 11g installation file, which is *C:\app\shay\product\11.2.0\dbhome_1\NETWORK\ADMIN*. Open *Tnsname.ora* and add

```
R_MYSQL =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = tcp)(HOST = WOSHAYNI)(PORT = 1521))
(CONNECT_DATA =
(SID = R_MYSQL)
)
(HS = OK)
)
```

Note that *R_MYSQL* is the name of data source that were created earlier and for *HOST* is the name of your PC where Oracle 11g installed and *PORT* is the port number of your Oracle 11g. Default *PORT* number for Oracle 11g is 1521.



```
tnsnames.ora - Notepad
File Edit Format View Help
# tnsnames.ora Network Configuration File: C:\app\shay\product\11.2.0\dbhome_1\network\admin\tnsnames.ora
# Generated by Oracle configuration tools.

R_MYSQL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = tcp)(HOST = WOSHAYNI)(PORT = 1521))
    (CONNECT_DATA =
      (SID = R_MYSQL)
    )
  )
  (HS = OK)
)

D =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = tcp)(HOST = WOSHAYNI)(PORT = 1521))
    (CONNECT_DATA =
      (SID = d)
    )
  )
  (HS = OK)
)
```

Figure 5.23: Configuration of tnsname.ora file

Step 10: Go to *C:\app\shay\product\11.2.0\dbhome_1\hs\admin* and create a *NEW file* named with *init<YOUR DATA SOURCE NAME>.ora*, for this,create a new ora file called *iniR_MYSQL.ora* and add these two line.

```
HS_FDS_CONNECT_INFO = R_MYSQL  
HS_FDS_TRACE_LEVEL = 0
```

Note that *R_MYSQL* is the name of data source that has been created earlier.

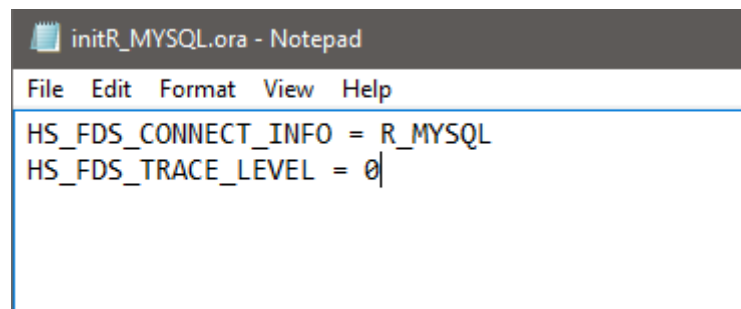


Figure 5.24: Configuration of init file

5.2.3 Installation steps of PostgreSQL on Ubuntu

Since Ubuntu is a type of Linux operating system, public usually own the laptop with Windows operating system. Therefore, we have to install the Ubuntu manually first before proceed to another stage. After that, database selected can install on Ubuntu.

5.2.3.1 Installation operating system (Ubuntu)

Step 1: Double click at wubi software. This Ubuntu installation doesn't need any hardware such as hard drive, burn into cd and etc.

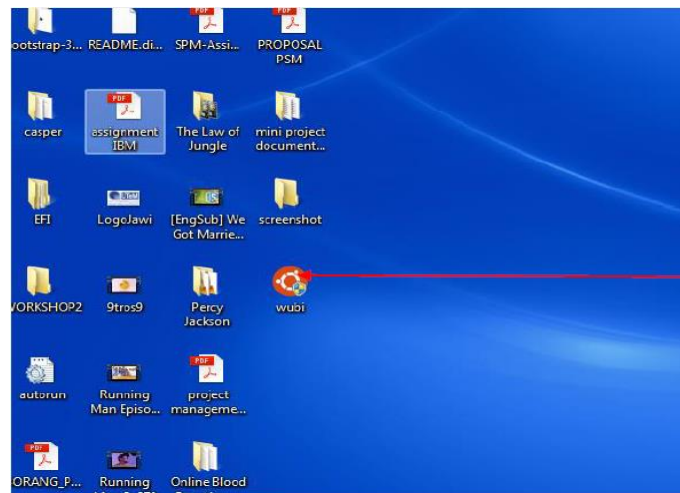


Figure 5.25: Installing Ubuntu

Step 2: After click the wubi, it will pop out like below. You need to enter username and password and make sure you remember your own username and password. Set the installation drive, installation size, desktop environment and language also. Then click install to proceed.

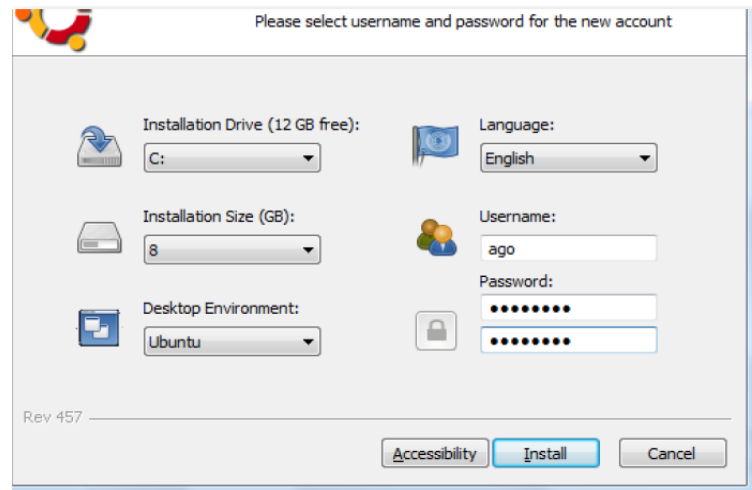


Figure 5.26: Set the installation

Step 3: The installation is loading. It may take long time if the internet connection is slow. What are it do during installation is checking installation files.

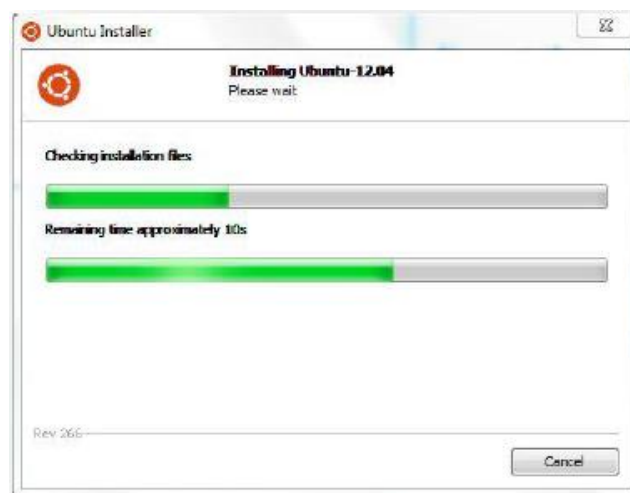


Figure 5.27: Loading installation

Step 4: After that, you need to reboot your computer to completing the Ubuntu setup by click 'finish' button.

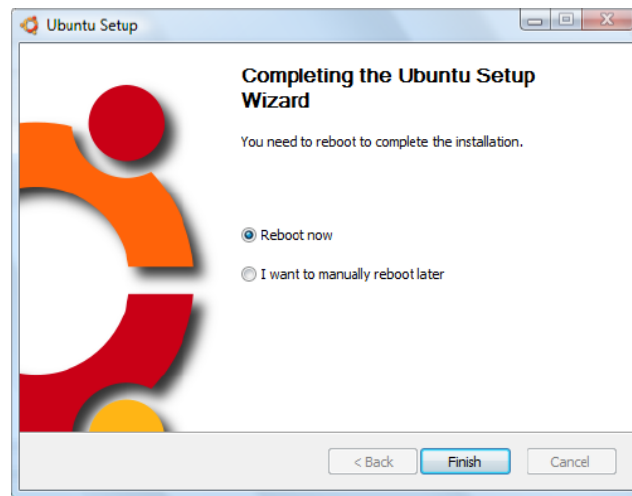


Figure 5.28: Installation finish

Step 5: When reboot is done, you have dual boot which is you can choose what operating system you want use either window 7 or Ubuntu. Select and press "enter".

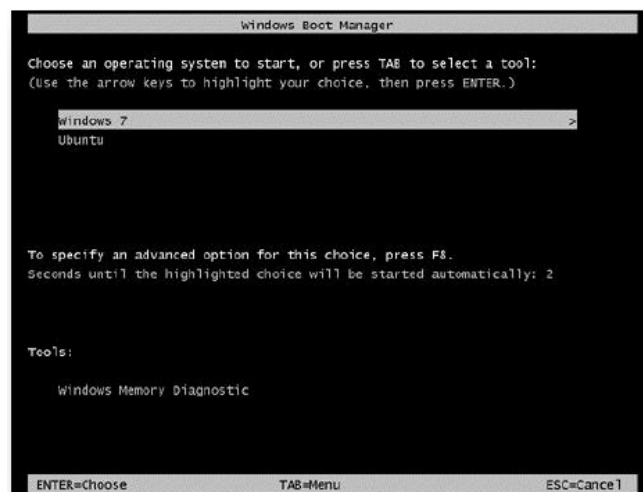


Figure 5.29: Ubuntu desktop

5.2.3.2 Installation database

Step 1: Open you terminal in Ubuntu and install PostgreSQL. Write command on terminal like this:

```
sudo apt-get update
```

```
sudo apt-get install postgresql postgresql-contrib
```

```
sudo -i -u postgres
```

```
psql
```

Step 2: Install java and netbeans. Write command on terminal like this:

```
sudo apt-get update
```

```
sudo apt-get install oracle-java8-installer
```

```
sudo add-apt-repository "deb http://archive.ubuntu.com/ubuntu $(lsb_release -sc)  
universe"
```

```
sudo apt-get install netbeans
```

5.2.4 Installation steps of MySQL on Windows

Step 1: Accept the agreement and you will be prompted with a selection of installation type. Choose Custom as shown below.

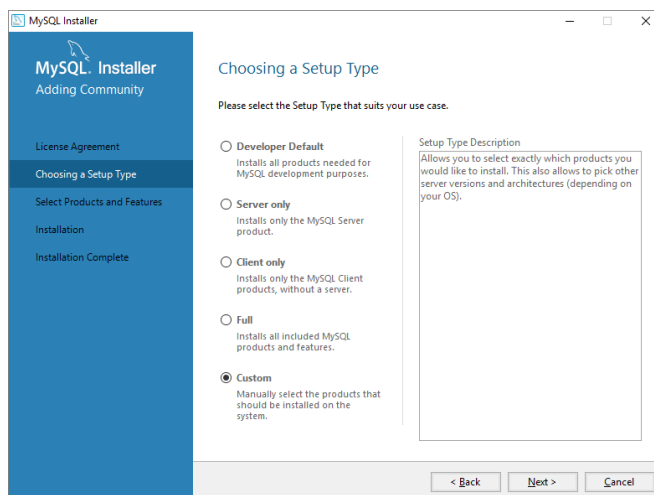


Figure 5.30: Choosing a setup type

Step 2: Click Next to bring up a screen to choose products for installation. Expand MySQL Server and choose MySQL Server 5.7.10.

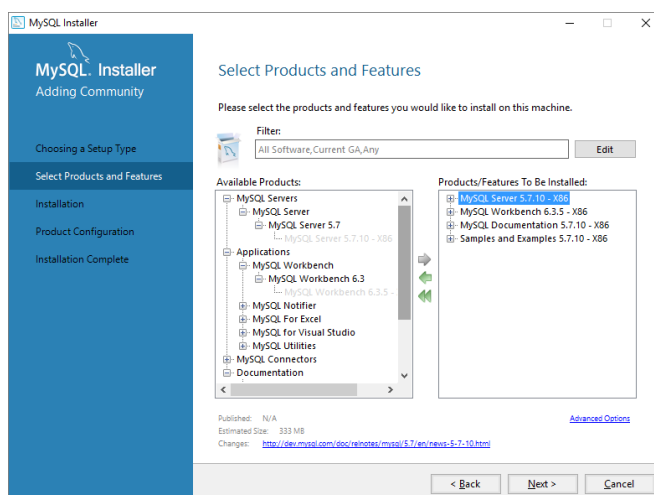


Figure 5.31: Select products and features

Step 3: After installation is complete, click Next. Product Configuration screen will indicate the need to configure MySQL Server and Samples. Click Next. Keep the configuration type as Development Machine. Keep TCP/IP and 3306 port as-is. Allow installer to open firewall port for network access and choose Show Advanced Options.

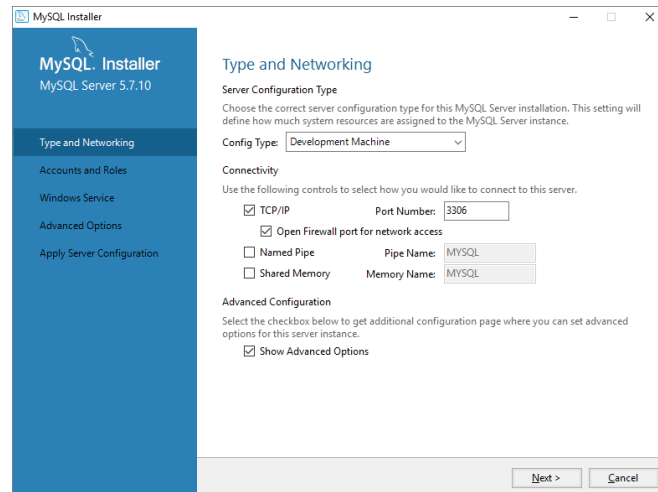


Figure 5.32: Setting type and networking

Step 4: Set up MySQL root password. Type in a complex password. Do not create any user at this time.

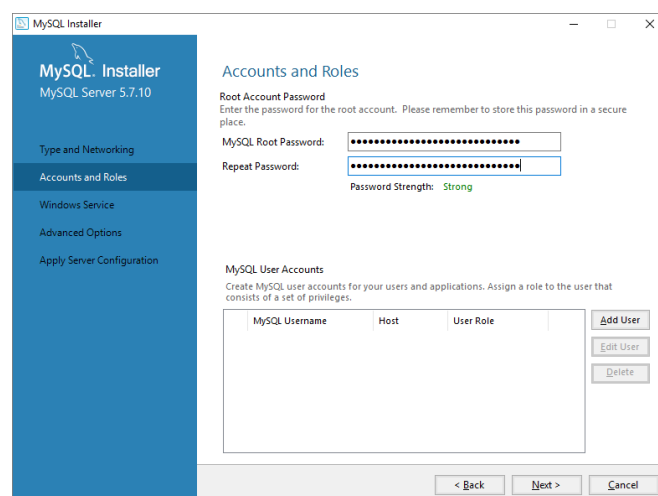


Figure 5.33: Set up MySQL root password

Step 5: Click Next. Install MySQL as a Windows Service. This is the easiest manner to start, stop, restart MySQL and allow Windows to start MySQL when Windows starts. Leave the advanced setting as is. Click Next and in the screen that follows, click Execute. If everything is successful, click Finish. Otherwise, use the log to determine what happened.

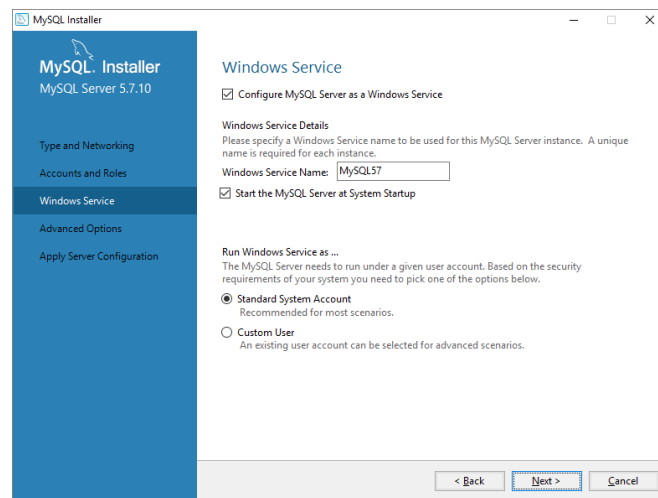


Figure 5.34: Setting Windows Service

Step 6: Click Finish to move to configuration of Sample.

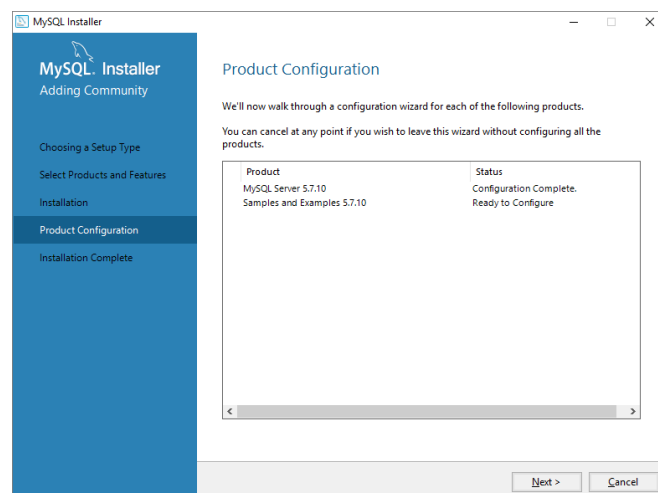


Figure 5.35: List of Product Configuration

Step 7: Click Next. In the screen that follows, click Check button. That enables the Next button. Click Execute. This process will run all the necessary scripts to configure sample. If it is successful, click Finish. Otherwise, analyse log tab to figure out what went wrong. Click Next. Leave Start MySQL Workbench after Setup checked. Click Finish. Workbench will start by itself in a few seconds.

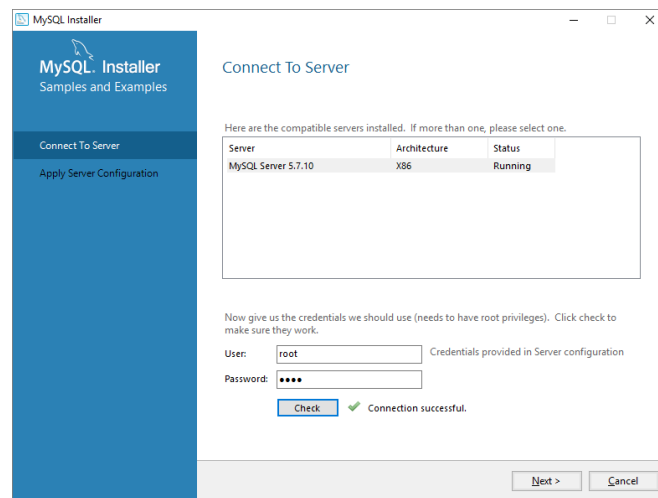


Figure 5.36: Connecting to Server

Step 8: At this point, MySQL Server has been installed and it is running. To access MySQL Server, we will use MySQL Workbench as a client. When Workbench starts, a connection to local database would have been created by default like so. Double-click on Local instance MySQL57 connection and type in root password to gain access to the server.

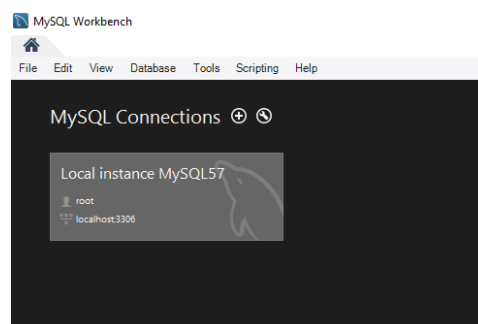


Figure 5.37: Workbench is started

5.3 Database development

5.3.1 SQL Server on Windows

TrgNewOrder Trigger

```

USE [Tuppershop]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TRIGGER [dbo].[TrgNewOrder] ON [dbo].[Order]
INSTEAD OF INSERT
AS
    DECLARE @NextID int;
    DECLARE @OrderID varchar(5);

    SET @NextID = NEXT VALUE FOR dbo.SeqOrderID;

    BEGIN

        SET NOCOUNT ON;

        IF @NextID < 10
            SET @OrderID = CONCAT('0000', @NextID);
        ELSE IF @NextID < 100
            SET @OrderID = CONCAT('000', @NextID);
        ELSE IF @NextID < 1000
            SET @OrderID = CONCAT('00', @NextID);
        ELSE
            SET @OrderID = CONCAT('0', @NextID);

        INSERT INTO [dbo].[Order] (
            [OrderID],
            [ClientID]
        )
        SELECT
            @OrderID,
            [ClientID]
        FROM
            INSERTED;

    END;

GO

```

UfnEncryptPassword Function

```

USE [Tuppershop]
GO

SET ANSI_NULLS ON
GO

```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE FUNCTION [dbo].[UfnEncryptPassword]
(
    @Password varchar(MAX)
)
RETURNS varchar(MAX)
AS
BEGIN
    RETURN CONVERT(VARCHAR(MAX), HASHBYTES('MD5',
LTRIM(RTRIM(@Password))),2);
END

GO
```

UspAccountExist Stored Procedure

```
USE [Tuppershop]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[UspAccountExist]
    @Type varchar(7),
    @Email varchar(115),
    @Exist bit OUTPUT
AS
BEGIN
    IF @Type = 'Client'
        SELECT @Exist = CASE WHEN COUNT(*) = 0 THEN 0 ELSE 1 END FROM
[Client] WHERE [Email] = @Email;

    ELSE IF @Type = 'Staff'
        SELECT @Exist = CASE WHEN COUNT(*) = 0 THEN 0 ELSE 1 END FROM
[Staff] WHERE [Email] = @Email;
END

GO
```

5.3.2 Oracle 11g on Windows

5.3.2.1 Create database link

Table 5.1: Creating database link MYSQL_LINK

DATABASE LINK
CREATE DATABASE LINK MYSQL_LINK CONNECT TO “root” IDENTIFIED BY “123456” USING ‘R_MYSQL’;

5.3.2.2 Create procedure to insert data into different databases

Table 5.2: Creating procedure DUAL INSERT

PROCEDURE
<pre>CREATE OR REPLACE PROCEDURE dual_ins_department (p_id IN varchar2,p_name IN varchar2) IS BEGIN INSERT INTO department@MYSQL_LINK VALUES(p_id,p_name); commit; INSERT INTO department VALUES(p_id,p_name); commit; END dual_ins_department; /</pre>
<pre>CREATE OR REPLACE PROCEDURE dual_ins_job (p_id IN varchar2,p_name IN varchar2) IS BEGIN INSERT INTO job@MYSQL_LINK VALUES(p_id,p_name); commit; INSERT INTO job VALUES(p_id,p_name); commit; END dual_ins_job; /</pre>

```

CREATE OR REPLACE PROCEDURE dual_ins_staff
(p_id IN varchar2,p_name IN varchar2, p_ic IN varchar2,p_pnum IN
number,p_email IN varchar2,p_pass IN varchar2,p_did IN varchar2,p_jid
IN varchar2)
IS
BEGIN
    INSERT INTO staff@MYSQL_LINK
VALUES(p_id,p_name,p_ic,p_pnum,p_email,p_pass,p_did,p_jid);
    commit;
    INSERT INTO staff
VALUES(p_id,p_name,p_ic,p_pnum,p_email,p_pass,p_did,p_jid);
    commit;
    END dual_ins_staff;
/

```

5.3.2.3 Create procedure to call MySQL's procedure about delete data

Table 5.3: Creating procedure DELETE CALLING MYSQL PROCEDURE

PROCEDURE
<pre> CREATE OR REPLACE PROCEDURE del_job_mysql (p_id IN varchar2) AS rows number; BEGIN rows := DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@MYSQL_LIN K('CALL `workshop2`.`del_job`('" p_id '"');); END del_job_mysql; / </pre>
<pre> CREATE OR REPLACE PROCEDURE del_department_mysql (p_id IN varchar2) AS rows number; BEGIN rows := DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@MYSQL_LIN K('CALL `workshop2`.`del_department`('" p_id '"');); END del_department_mysql; / </pre>

```

CREATE OR REPLACE PROCEDURE del_staff_mysql
    (p_id IN varchar2)
    AS rows number;
    BEGIN
        rows :=
DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@MYSQL_LINK('CALL `workshop2`.`del_staff`('' || p_id || '');');
        END del_staff_mysql;
/

```

5.3.3 MySQL on Windows

5.3.3.1 Create CSV file

Table 5.4: Create CSV file of table PRODUCT and saved at default directory

PRODUCT
SET SQL_SAFE_UPDATES=0; DELETE FROM product_h; SELECT * INTO OUTFILE 'C:\workshop2\product.csv' FIELDS TERMINATED BY ',' ENCLOSED BY '''' ESCAPED BY '\\' LINES TERMINATED BY '\n' FROM product;

5.4 Database access

We can access own database through the interface of system. Moreover, we can access other databases that are far apart from us but all have to connect to the same network. Different databases connect to each other by using the same network for data integration.

5.4.1 Access to own database

The following code is the connection to own database and login to homepage from the interface that build up by using Java Server Page (JSP).

```
<%
    java.sql.Connection con;
    java.sql.Statement s;
    java.sql.ResultSet rs;
    java.sql.PreparedStatement pst;
    con=null;
    s=null;
    pst=null;
    rs=null;

    String user=request.getParameter("username");
    String pass=request.getParameter("userpass");
    Class.forName("org.postgresql.Driver");
    con
    DriverManager.getConnection("jdbc:postgresql://localhost:5432/postgres","postgres", "postgres");

    String sql = "select * from workshop2.staff where staffid='"+user+"' and
    password= '"+pass+"'";

    s = con.createStatement();
    rs = s.executeQuery(sql);
    boolean empty = true;
```



```

while( rs.next() ) {
    empty = false;
}

if(!empty)
{
    session.setAttribute("islogin","plz sign in first");
    session.setAttribute("admin","admin");
}%>

```

5.4.2 Access to other database

5.4.2.1 From Oracle 11g access to MySQL

This part is to access MySQL database through the database link from Oracle 11g. An example to select data from Oracle 11g to MySQL as shown below:

```
SELECT * FROM "department"@MYSQL_LINK;
```

```

DEPARTMENTID
-----
DEPARTMENTNAME
-----
DPT01
SALES

DPT02
HUMAN RESOURCE

DPT03
INVENTORY

DEPARTMENTID
-----
DEPARTMENTNAME
-----
DPT04
INFORMATION TECHNOLOGY

```

Figure 5.38: Result of selecting data

```
SELECT * FROM "department"@MYSQL_LINK
```

```
WHERE departmentid = 'DPT01';
```



The image shows a query result in a table format. The first row contains the column headers 'DEPARTMENTID' and 'DEPARTMENTNAME', separated by a dashed line. The second row contains the data values 'DPT01' and 'SALES'.

DEPARTMENTID	DEPARTMENTNAME
DPT01	SALES

Figure 5.39: Result of selective data

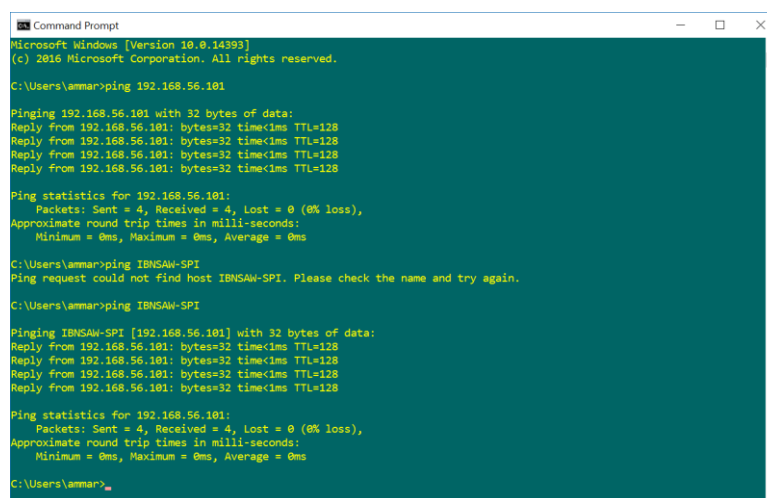
From these two example code, it is shown that the database MySQL is successfully being access by database Oracle 11g through database link.

5.4.2.2 From SQL Server access to Oracle 11g

This is how the SQL Server manages to connect to all the other DBMS.

Configuration

Step 1: First, we must make sure the connection between the local and remote established. Typically, we use command prompt to ping each other. Make sure the ping succeeds.



```

Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\ammar>ping 192.168.56.101

Pinging 192.168.56.101 with 32 bytes of data:
Reply from 192.168.56.101: bytes=32 time<1ms TTL=128
Reply from 192.168.56.101: bytes=32 time<1ms TTL=128
Reply from 192.168.56.101: bytes=32 time<1ms TTL=128
Reply from 192.168.56.101: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.56.101:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\ammar>ping IBNSAW-SPI
Ping request could not find host IBNSAW-SPI. Please check the name and try again.

C:\Users\ammar>ping IBNSAW-SPI

Pinging IBNSAW-SPI [192.168.56.101] with 32 bytes of data:
Reply from 192.168.56.101: bytes=32 time<1ms TTL=128
Reply from 192.168.56.101: bytes=32 time<1ms TTL=128
Reply from 192.168.56.101: bytes=32 time<1ms TTL=128
Reply from 192.168.56.101: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.56.101:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\ammar>

```

Figure 5.40: Ping to another device from own device

Step 2: For connecting to Oracle we need to download an ODAC from Oracle Official Site. Download and install it.

32bit - <http://www.oracle.com/technetwork/database/windows/downloads/utilsoft-087491.html>

64bit - <http://www.oracle.com/technetwork/database/windows/downloads/index-090165.html>

Steps to install Oracle ODAC xcopy:

2.1. Download the zip file and extract it. (eg: C:/Users/User/Desktop/odac_xcopy)

2.2. Open command prompt and go to the extracted path.

2.3. Type **install all C:/Oracle odac**. Wait until the installation finish.

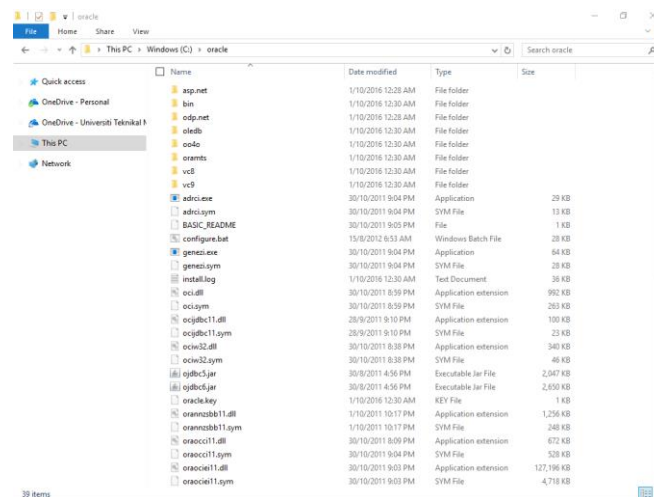


Figure 5.41: Oracle directory

2.4. Open environment path, click at path and add in C:/Oracle/ and C:/Oracle/bin/

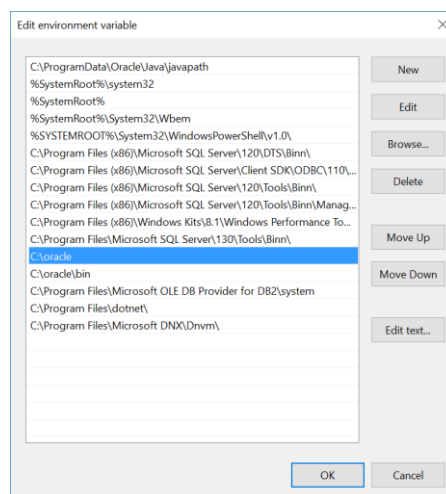


Figure 5.42: Edit environment variable

2.5. Restart Computer

Query to retrieve data in Oracle 11g and PostgreSQL

```

SELECT * FROM LINK-NAME..SCHEMA-NAME.TABLE-NAME
INSERT INTO FROM LINK-NAME..SCHEMA-NAME.TABLE-NAME
VALUES (VALUE1, VALUE2, ...)
UPDATE LINK-NAME..SCHEMA-NAME.TABLE-NAME
SET COLUMN = VALUE
WHERE COLUMN = VALUE;
DELETE LINK-NAME..SCHEMA-NAME.TABLE-NAME
WHERE COLUMN = VALUE;

```

5.4.2.3 From SQL Server access to MySQL

Configuration

Step 1: To connect with the other DBMS such as MySQL, Postgres, MariaDB, you must install ODBC Driver to establish the connection between SQL Server and all those DBMS.

ODBC Drivers:

MySQL - <https://dev.mysql.com/downloads/connector/odbc/>

MariaDB - <https://downloads.mariadb.org/connector-odbc/>

PostgreSQL - <https://odbc.postgresql.org/>

Install it until finish. Then, below are the steps needed to establish the connection (e.g.: MySQL).

Step 2: Open ODBC (Search windows)

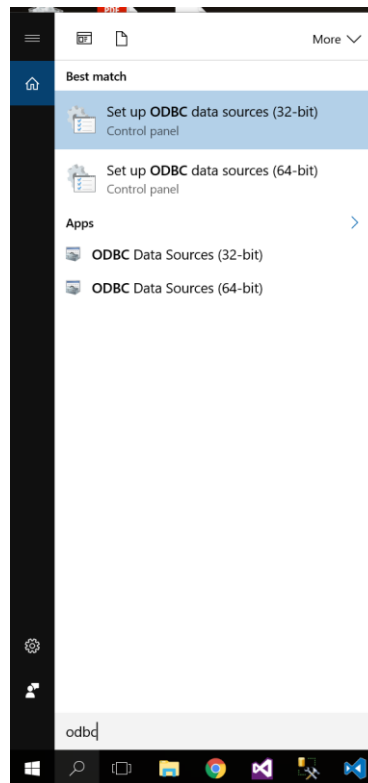


Figure 5.43: Search ODBC on Windows

Step 3: Open System DSN and click Add.

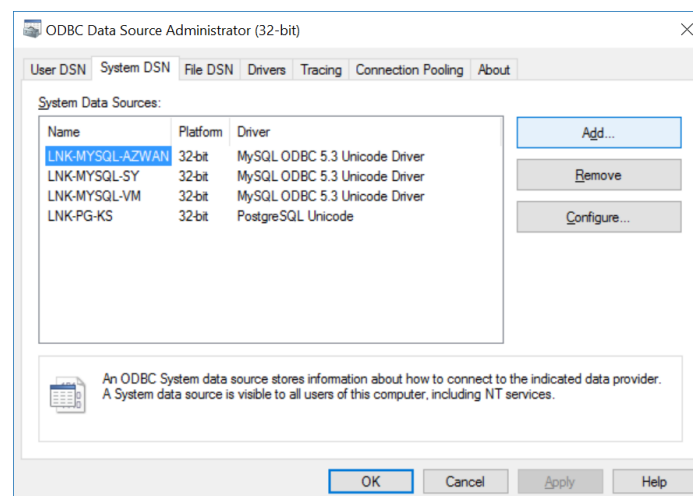


Figure 5.44: View system DSN in ODBC on Windows

Step 4: Choose MySQL in the list and type the required details.

Data Source Name: << PUT ANYTHING HERE. BUT PLEASE REMEMBER >>

TCP/IP Server: << HOSTNAME/IP ADDRESS >>

User: << USERNAME IN MYSQL >>

Password: << USER'S PASSWORD SET UP IN MYSQL >>

Database:

<< MUST CHOOSE THE DATABASE TO ENSURE THE ABILITY TO ACCESS
LATER >>

Click on Test button. If the connection succeeds, a prompt message will appear.

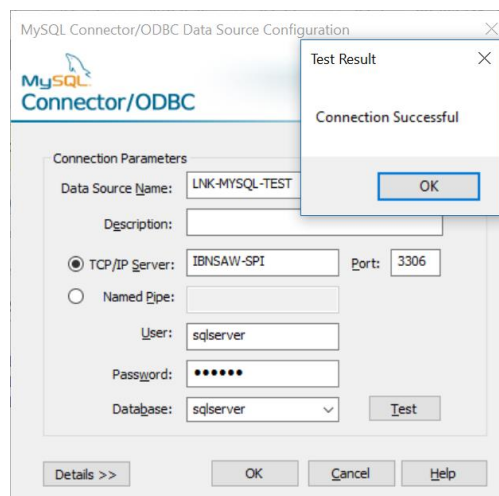


Figure 5.45: Test connection to MySQL

Step 5: Open SSMS and connect to your account. In the Object Explorer window, expand Server Objects > Linked Servers > Providers. Double click on MSDASQL. Here, tick on Dynamic Parameters and allow in process.

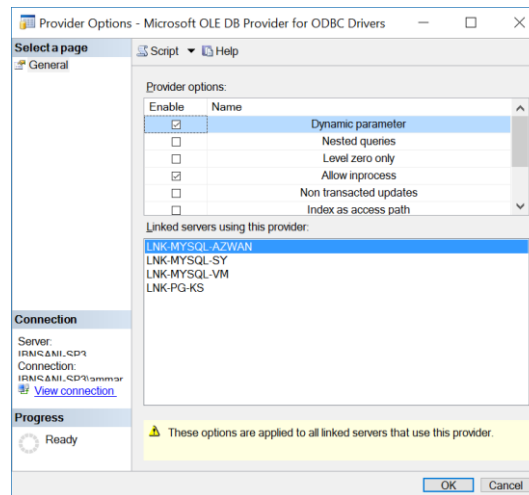


Figure 5.46: Edit provider options

Step 6: On Linked Servers menu, right click and choose New Linked Server.

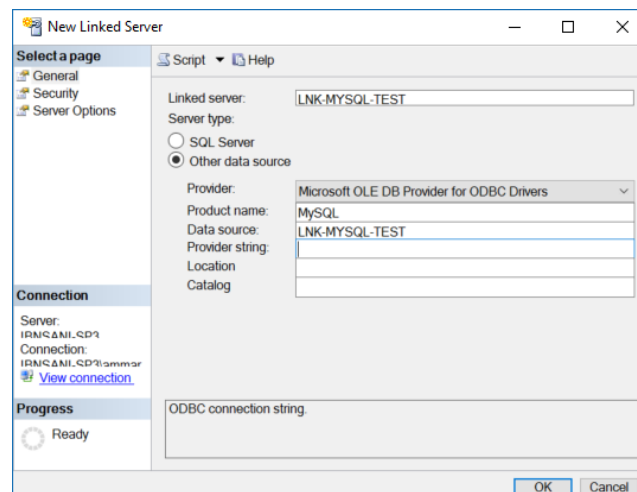


Figure 5.47: Edit details in new linked server

Step 7: Click on Security and select **Be made using this security context**, type username and password as in ODBC earlier. Then click OK. If no error is showing, it's means the connection succeed.

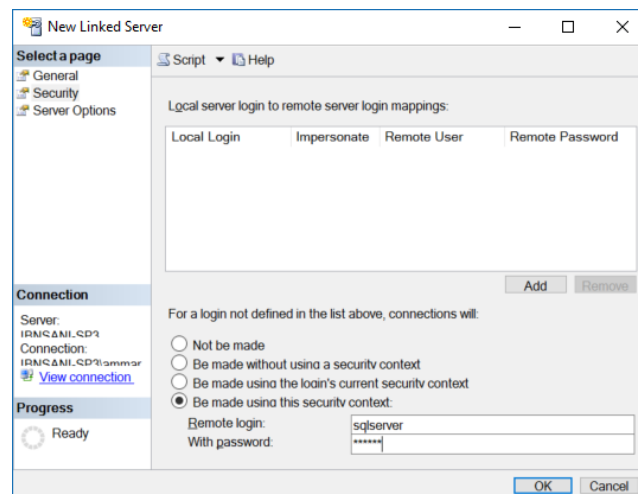


Figure 5.48: Account authentication in ODBC

Query to retrieve data in MySQL

```
SELECT * FROM OPENQUERY(LINK-NAME, 'SQL QUERY IN
MySQL');
```

```
INSERT INTO OPENQUERY(LINK-NAME, 'SQL QUERY IN MySQL')
VALUES (VALUE1, VALUE2, ...);
```

```
UPDATE OPENQUERY(LINK-NAME, 'SQL QUERY IN MySQL')
SET COLUMN = VALUE
```

```
WHERE COLUMN = VALUE;
```

```
DELETE OPENQUERY(LINK-NAME, 'SQL QUERY IN MySQL')
WHERE COLUMN = VALUE;
```

5.5 Database maintenance

5.5.1 Oracle 11g on Windows

This part is for maintenance if the source pc is changing the IP Address. Quickly change the *TCP/IP SERVER* for the new IP Address or just put the source *PC NAME* so that even if the IP address is change, the connection still can be made.

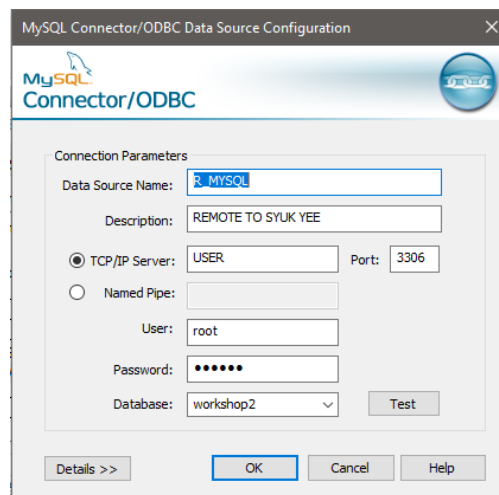


Figure 5.49: Changes of IP server

5.5.2 MySQL on Windows

5.5.2.1 Import CSV file

Table 5.14: Import data from CSV file to table PRODUCT

PRODUCT
<pre> set @trig_product_insert = 1; LOAD DATA INFILE 'C:\workshop2\product.csv' INTO TABLE product FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n'; SET @trig_product_insert = 0; </pre>

5.5.2.2 Incremental recovery from audit trail table PRODUCT

Table 5.14: Recover data from audit trail table PRODUCT

PRODUCT
<pre> CREATE DEFINER=`root`@`%` PROCEDURE `incremental_recover_product`() BEGIN DECLARE n INT DEFAULT 0; DECLARE i INT DEFAULT 0; DECLARE xaction varchar(45); DECLARE xproductID varchar(10); DECLARE xproductName varchar(45); DECLARE xdescription varchar(80); DECLARE xcostPrice decimal(10,2); DECLARE xsellingPrice decimal(10,2); DECLARE xunitInStock int(11); DECLARE xsupplierID varchar(10); DECLARE xproductTypeID varchar(10); SET SQL_SAFE_UPDATES=0; SELECT COUNT(*) FROM product_h INTO n; SET i=0; </pre>

```

WHILE i<n DO

select action into xaction from product_h LIMIT i,1;
select productID into xproductID from product_h LIMIT i,1;
select productName into xproductName from product_h LIMIT i,1;
select description into xdescription from product_h LIMIT i,1;
select costPrice into xcostPrice from product_h LIMIT i,1;
select sellingPrice into xsellingPrice from product_h LIMIT i,1;
select unitInStock into xunitInStock from product_h LIMIT i,1;
select supplierID into xsupplierID from product_h LIMIT i,1;
select productTypeID into xproductTypeID from product_h LIMIT i,1;

if(xaction = 'INSERT') then
set @trig_product_insert = 1;
INSERT INTO product
VALUES(xproductID,xproductName,xdescription,xcostPrice,
      xsellingPrice,xunitInStock,xsupplierID,xproductTypeID);
SET @trig_product_insert = 0;

elseif (xaction = 'UPDATE') then
set @trig_product_delete = 1;
set @trig_product_insert = 1;
delete from product
where productID = xproductID;
INSERT INTO product
VALUES(xproductID,xproductName,xdescription,xcostPrice,
      xsellingPrice,xunitInStock,xsupplierID,xproductTypeID);
SET @trig_product_delete = 0;
SET @trig_product_insert = 0;

else
set @trig_product_delete = 1;
delete from product
where productID = xproductID;
SET @trig_product_delete = 0;

end if;

SET i = i + 1;

END WHILE;
DELETE FROM product_h;
END

```

5.6 Database management and administration

5.6.1 SQL Server on Windows

Create New SQL Server User

Step 1: In SSMS, Expand Security > Logins

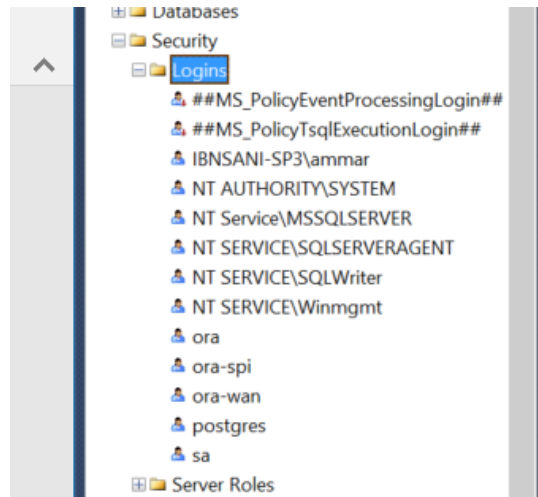


Figure 5.50: Logins tab in SQL Server

Step 2: Right click on Logins, Click New Login. Fill up as following:

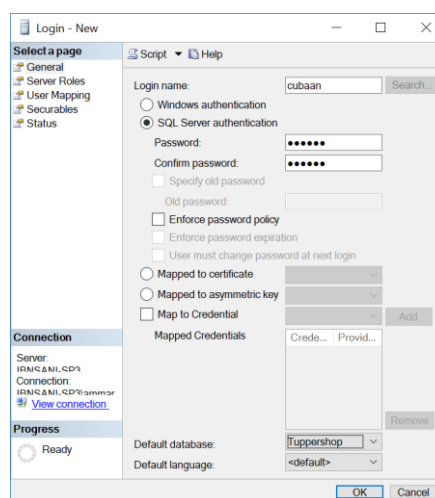


Figure 5.51: Edit the details in Logins tab in SQL Server

Step 3: Then click at User Mapping and follow as below:

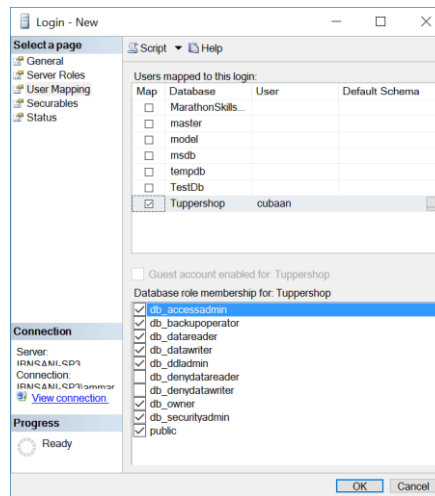


Figure 5.52: Manage the roles in Logins tab in SQL Server

Step 4: Click OK. Then click File in Menu Bar. Choose Connect Object Explorer and key in using the new user as below:

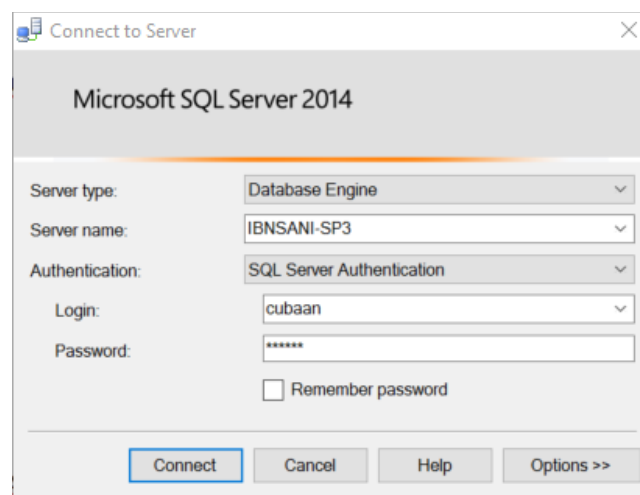


Figure 5.53: Login into SQL Server

Step 5: The SSMS will login using the account.

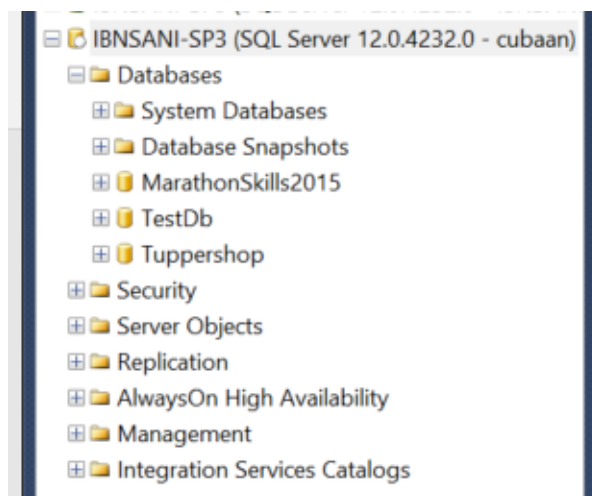


Figure 5.54: Login details in SQL Server

5.6.2 Oracle 11g on Windows

This part shows how granting privileges to user “shay” so that user can connect to Oracle 11g and perform task.

Table 5.16: Granting all privileges to user shay

CONN /as sysdba;
GRANT ALL PRIVILEGES TO USER shay IDENTIFIED BY shay WITH GRANT OPTION;

After granting all privileges, user shay can perform various task such as creating a new table, inserting data, updating data, and deleting data and most importantly connecting to Oracle 11g.

Table 5.17: Granting all privileges to user shay in the MYSQL Database

```
GRANT ALL PRIVILEGES ON *.* TO
'root'@'WOSHAYNI' identified by '123456';
```

These privileges are important because, to make remote access through *Database link* the privileges on user root to be remotely connect from another IP address is needed. “*WOSHAYNI*” is the Oracle 11g server Address.

5.7 Conclusion

A database can be accessed from another server called as remote connection, and to do query such as select, update, delete and insert from another type of DBMS is called Database link. In Oracle 11g there may steps that need to be done to allow the heterogeneous link between various DBMS.

CHAPTER VI

DATABASE INTEGRATION AND TESTING

6.1 Introduction

In this phase, integration of different database has been done. Database replication, recovery and migration are implemented between databases to produce the heterogeneous DBMS environments. There are two types of backup which are logical and physical backup. Logical backup is done by using the database utility to read sets of records and write it to a file. While Physical backup is using the O/S file backup commands to backup the database.

6.2 Methods of backup and recovery

a) Full Backup

productID	productName	description	costPrice	sellingPrice	unitInStock	supplierID	productTypeID
P0033	Tupperware Multicolour Plates	Inclusive of : Name and Number of Pieces:4	500.00	550.00	30	S0001	PT0009
P0034	Tupperware Green Serving Dishes	Inclusive of : Name and Number of Pieces:1 Tray	200.00	250.00	30	S0002	PT0009
P0035	Plastic Cutlery Holder	The base of the Hold 'N Rest is wide and stable.	400.00	450.00	30	S0006	PT0008
P0036	Tupperware Cuterly spoon	Material : Polypropylene (PP) Colour : Red	300.00	350.00	30	S0006	PT0008
P0037	Tupperware fresh Set	12pcs Colorful Stackable Fridge Containers	600.00	650.00	30	S0008	PT0004
P0038	Tupperware RiceSmart	The surface of the piece peels	300.00	350.00	30	S0006	PT0004
P0039	Elegance Lunch Set	FREE Eco FlipTop Transparent (1) 500 ML	200.00	250.00	30	S0008	PT0004
P0040	Tupperware Bottle	the breaking line passes through the mold injec...	100.00	150.00	30	S0007	PT0004
P0041	Elegance Dinner Set	FREE Eco FlipTop Transparent (1) 750 ML	200.00	250.00	30	S0007	PT0004
P0042	Pokemon model	Very cute and suitable for child	100.00	150.00	30	S0007	PT0010
P0043	Avenger tupperware model	Suitable for the child and cheap	100.00	150.00	30	S0007	PT0010
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 6.1: Current data in table PRODUCT












	workshop2client	13/12/2016 8:56 PM	Microsoft Excel C...	1 KB
	workshop2department	14/12/2016 9:00 AM	Microsoft Excel C...	1 KB
	workshop2job	13/12/2016 8:53 PM	Microsoft Excel C...	1 KB
	workshop2managementdetail	13/12/2016 8:54 PM	Microsoft Excel C...	1 KB
	workshop2order	13/12/2016 8:55 PM	Microsoft Excel C...	1 KB
	workshop2orderdetail	13/12/2016 8:55 PM	Microsoft Excel C...	1 KB
	workshop2ordermessage	13/12/2016 8:59 PM	Microsoft Excel C...	1 KB
	workshop2product	14/12/2016 1:00 PM	Microsoft Excel C...	7 KB
	workshop2producttype	13/12/2016 8:54 PM	Microsoft Excel C...	1 KB
	workshop2staff	13/12/2016 8:54 PM	Microsoft Excel C...	1 KB
	workshop2supplier	13/12/2016 8:54 PM	Microsoft Excel C...	1 KB

Figure 6.2: The CSV file created is saved at default directory

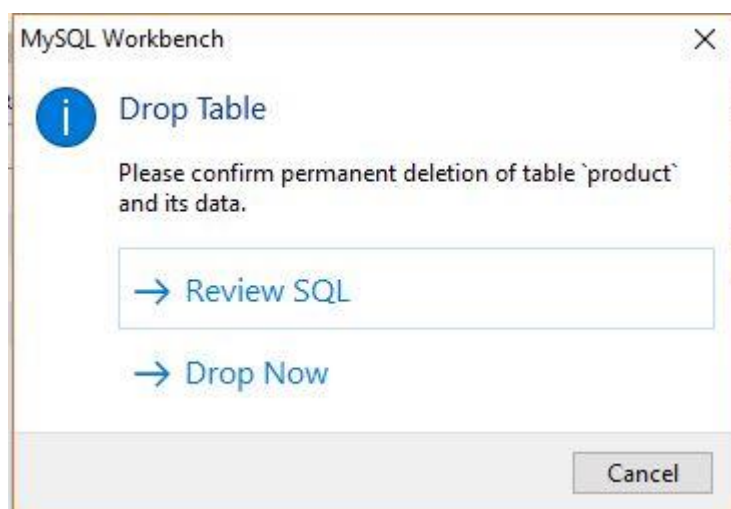


Figure 6.3: Drop the table PRODUCT

```

CREATE TABLE `workshop2`.`product` (
  `productID` VARCHAR(10) NOT NULL,
  `productName` VARCHAR(45) NOT NULL,
  `description` VARCHAR(150) NOT NULL,
  `costPrice` DECIMAL(10,2) NOT NULL,
  `sellingPrice` DECIMAL(10,2) NOT NULL,
  `unitInStock` INT NOT NULL,
  `supplierID` VARCHAR(10) NOT NULL,
  `productTypeID` VARCHAR(10) NOT NULL,
  PRIMARY KEY (`productID`),
  UNIQUE INDEX `productID_UNIQUE` (`productID` ASC));

set @trig_product_insert = 0;
set @trig_product_update = 0;
set @trig_product_delete = 0;

DELIMITER $$

DROP TRIGGER IF EXISTS workshop2.product_AFTER_INSERT$$
USE `workshop2` $$
CREATE DEFINER=`root`@`%` TRIGGER `workshop2`.`product_AFTER_INSERT` AFTER INSERT ON `product` FOR EACH ROW
BEGIN
  IF @trig_product_insert = 0 THEN
    insert into product_h(dateTime,action,productID,productName,description,costPrice,
    sellingPrice,unitInStock,supplierID,productTypeID)
    values

```

Figure 6.4: Create table PRODUCT, related triggers and import data from CSV file

productID	productName	description	costPrice	sellingPrice	unitInStock	supplierID	productTypeID
P0033	Tupperware Multicolour Plates	Inclusive of : Name and Number of Pieces:4	500.00	550.00	30	S0001	PT0009
P0034	Tupperware Green Serving Dishes	Inclusive of : Name and Number of Pieces:1 Tray	200.00	250.00	30	S0002	PT0009
P0035	Plastic Cutlery Holder	The base of the Hold 'N Rest is wide and stable.	400.00	450.00	30	S0006	PT0008
P0036	Tupperware Cutlerly spoon	Material : Polypropylene (PP) Colour : Red	300.00	350.00	30	S0006	PT0008
P0037	Tupperware fresh Set	12pcs Colorful Stackable Fridge Containers	600.00	650.00	30	S0008	PT0004
P0038	Tupperware RiceSmart	The surface of the piece peels	300.00	350.00	30	S0006	PT0004
P0039	Elegance Lunch Set	FREE Eco Flitop Transparent (1) 500 ML	200.00	250.00	30	S0008	PT0004
P0041	Elegance Dinner Set	FREE Eco Flitop Transparent (1) 750 ML	200.00	250.00	30	S0007	PT0004
P0042	Pokemon model	Very cute and suitable for child	100.00	150.00	30	S0007	PT0010
P0043	Avenger tupperawre model	Suitable for the child and cheap	100.00	150.00	30	S0007	PT0010
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 6.5: Early state of data in table PRODUCT

b) Incremental Backup:

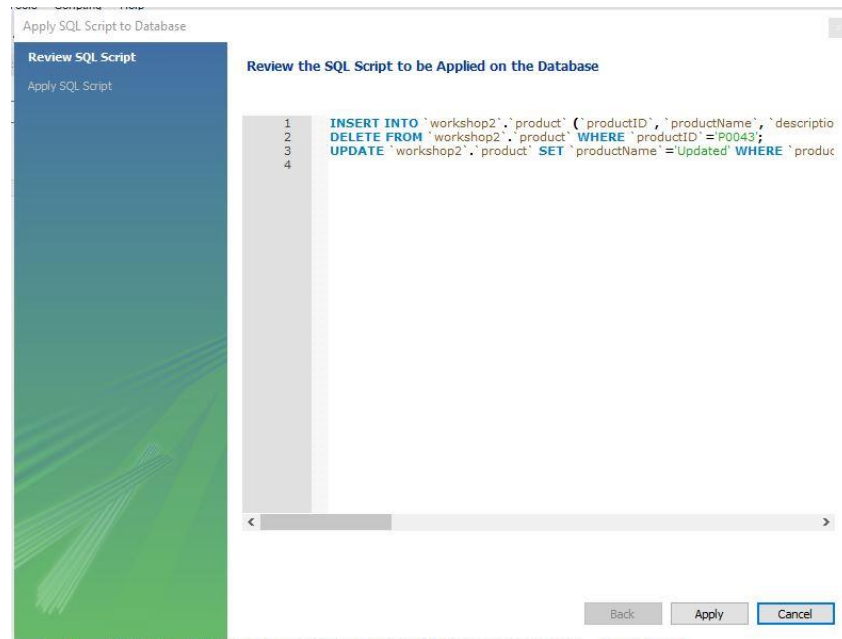


Figure 6.6: Edit data in table PRODUCT

idproduct_h	dateTime	action	productID	productName	description	costPrice	sellingPrice	unitInStock	supplierID	produc
27	2016-12-18 10:43:11	INSERT	P0044	Ice cream bowl	All time flavour for dessert lover	15.00	25.00	45	S0008	PT0005
28	2016-12-18 10:43:11	DELETE	P0043	Avenger tupperawre model	Suitable for the child and cheap	100.00	150.00	30	S0007	PT0010
29	2016-12-18 10:43:11	UPDATE	P0042	Updated	Very cute and suitable for child	100.00	150.00	30	S0007	PT0010
*										

Figure 6.7: Any changes of the data will be stored in audit trail table PRODUCT

```

if(xaction = 'INSERT') then
set @trig_product_insert = 1;
INSERT INTO product
VALUES(xproductID,xproductName,xdescription,xcostPrice,
      xsellingPrice,xunitInStock,xsupplierID,xproductTypeID);
SET @trig_product_insert = 0;

elseif (xaction = 'UPDATE') then
set @trig_product_delete = 1;
set @trig_product_insert = 1;
delete from product
where productID = xproductID;
INSERT INTO product
VALUES(xproductID,xproductName,xdescription,xcostPrice,
      xsellingPrice,xunitInStock,xsupplierID,xproductTypeID);
SET @trig_product_delete = 0;
SET @trig_product_insert = 0;

```

Figure 6.8: Recover data from audit trail table

productID	productName	description	costPrice	sellingPrice	unitInStock	supplierID	productTypeID
P0033	Tupperware Multicolour Plates	Inclusive of : Name and Number of Pieces:4	500.00	550.00	30	S0001	PT0009
P0034	Tupperware Green Serving Dishes	Inclusive of : Name and Number of Pieces:1 Tray	200.00	250.00	30	S0002	PT0009
P0035	Plastic Cutlery Holder	The base of the Hold 'N Rest is wide and stable.	400.00	450.00	30	S0006	PT0008
P0036	Tupperware Cuterly spoon	Material : Polypropylene (PP) Colour : Red	300.00	350.00	30	S0006	PT0008
P0037	Tupperware fresh Set	12pcs Colorful Stackable Fridge Containers	600.00	650.00	30	S0008	PT0004
P0038	Tupperware RiceSmart	The surface of the piece peels	300.00	350.00	30	S0006	PT0004
P0039	Elegance Lunch Set	FREE Eco Fliptop Transparent (1) 500 ML	200.00	250.00	30	S0008	PT0004
P0041	Elegance Dinner Set	FREE Eco Fliptop Transparent (1) 750 ML	200.00	250.00	30	S0007	PT0004
P0042	Updated	Very cute and suitable for child	100.00	150.00	30	S0007	PT0010
P0044	Ice cream bowl	All time flavour for dessert lover	15.00	25.00	45	S0008	PT0005
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 6.9: Latest data in table PRODUCT

6.3 Database integration and testing result analysis

The following are the steps to setting integration between MySQL in Windows operating system and PostgreSQL in Linux operating system.

i) Setup static IPv4 address in each PC

- a. Network and sharing center > Network Connections > Change adapter settings
- b. Disable WLAN by right click on the icon (to revoke from receive an WLAN IP address)
- c. Right click in Local Area Connection > Properties
- d. Select IPv4 and Click Properties.

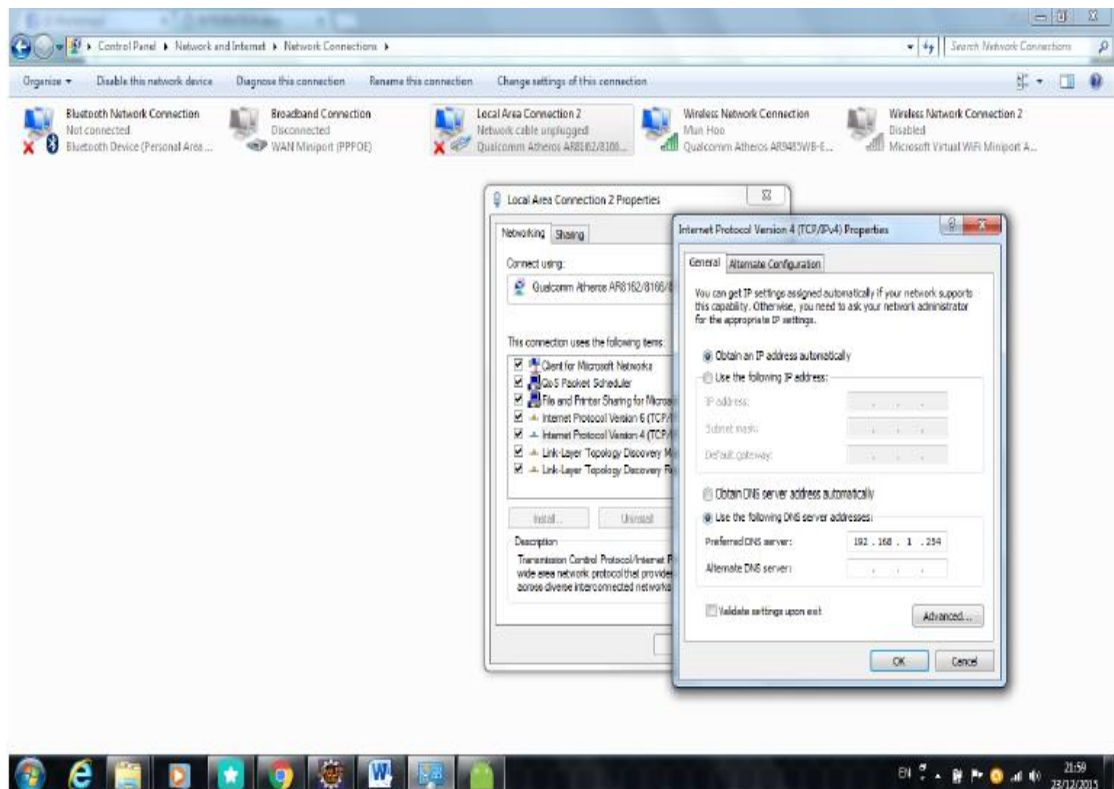


Figure 6.10: Internet Protocol Version 4 (TCP/IPv4) Properties

e. State the following IP address :

IP : 192.168.1.1, 192.168.1.2, 192.168.1.3, 192.168.1.4

Subnet Mask : 255.255.255.0

Default gateway and DNS server : 192.168.1.254

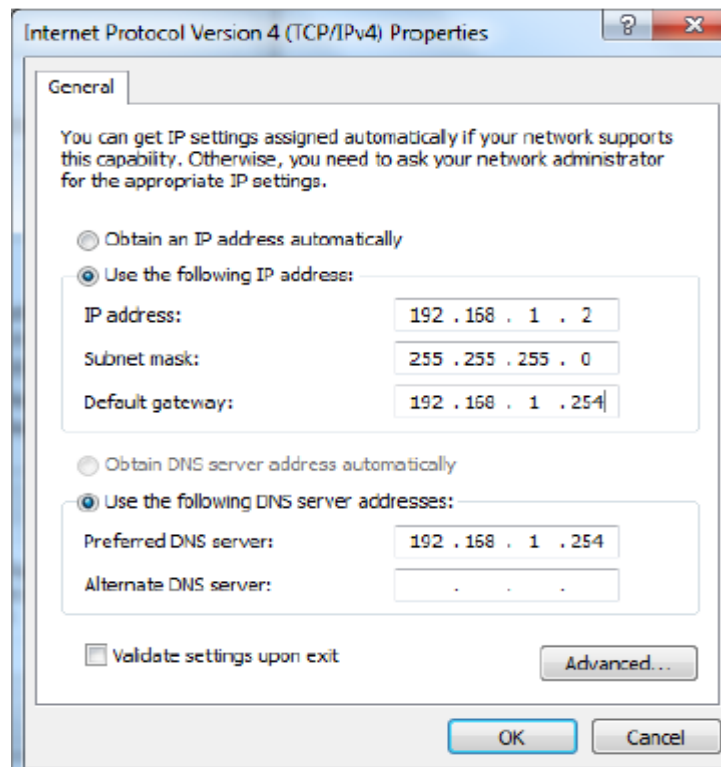


Figure 6.11: Edit the Internet Protocol Version 4 (TCP/IPv4) Properties

f. Click OK and Close.

ii) Disable firewall after "disable WLAN" connection and only enable "Local Area Connection".

- a. Control Panel > System and Security > Windows Firewall
- b. Left side menu bar click on "Turn Windows Firewall on or off"

- c. Choose turn off Windows Firewall for private and public network. If this firewall enable it will block other connection to connect with the PC)



Figure 6.12: Adjust the setting of Windows firewall

iii) Step for PostgreSQL to access data from MySQL

```
create extension mysql_fdw;
create server mysql_server
foreign data wrapper mysql_fdw
options (address '192.168.1.2', port '3306');
create user mapping for postgres
server mysql_server
options (username 'root', password '123456');
create foreign table workshop2.mysql_orderdetail(
orderdetailid varchar(10),
productid integer,
quantity integer ) server mysql_server
options (dbname 'workshop2', table_name 'orderdetail');
```


6.4 Conclusion

As the conclusion, integration between heterogeneous databases is important because the data from various sources can be accessed and it can be used for operational of system. In this part of integration of databases, we are able to do the database link which is the direct integration (online) and import and export the data through CSV files. The CSV files able to make as the backup and recovery for the database.

CHAPTER VII

CONCLUSION

7.1 Introduction

The accomplishment of this workshop 2 meets the objectives and scope as the beginning. There are 4 main subsystems in this TupperShop Management System which are inventory management, staff management, marketing management and data audit management. These 4 managements operate in distinct database systems such as SQL Server, Oracle 11g, PostgreSQL and MySQL. Moreover, PostgreSQL is running on Ubuntu only. However, the other 3 database systems are running on Windows. In this workshop 2, we decide to use Java Server Page (JSP) and Active Server Page (ASP) as the main languages in implementation of this TupperShop Management System.

7.2 Achievement

There are 4 subsystems in this TupperShop Management System which are marketing management, staff management, inventory management and data audit management. This system implemented successfully and provides various features. The functionalities of different subsystems are stated as following:

a) Marketing management

- i. Clients can make order after login into system

- ii. Clients can pay after the order is complete
- iii. Staffs manage to view the order from clients
- iv. Staffs manage to organise the allocation of products before shipping from the order

b) Staff management

- i. Administrators can insert the information of new staff
- ii. Administrators can update the latest details of staffs
- iii. Administrators can manage the information of departments
- iv. Administrators can manage the information of jobs

c) Inventory management

- i. Staffs can update the latest information of products
- ii. Staffs can check the latest quantity in hand of products
- iii. Staffs can order the product when the quantity in hand of product is lacking
- iv. Staffs can contact the supplier when they detect the flaw of products

d) Data audit management

- i. All data in the database can view at here
- ii. Analysis and documentation able to make at here
- iii. Full backup that save all the data in a table into CSV file can be done through system interface
- iv. Staffs can recover the data by using the saved CSV file
- v. Staffs can incremental recover the data to the latest data by using the records in audit trail table in database

7.3 Project limitation

There are some limitations in this this system. For instance, this system cannot make recovery automatically when data lost. This system recover the data in database manually. Staffs have to backup the data into CSV file first then recover the data by using that file manually.

7.4 Suggestion for improvement

Selective recovery is recommended to implement in this system. When the system detects the different between every database, then recover the data and make synchronize data. Hence, data integrity is enhanced in the system and the data in different databases is updated all the time without any manual operation.

7.5 Potential commercialization

This online ordering system provides a better platform to replace the tradition business. Those tradition businesses usually operate at physical booth or store. By using this system, organization manages to save a lot of budget on rental, wage for staff, insurance and others. This is due to the fact that the system can operate at anywhere as long as there is internet connection. Hence, this system leads the business to the latest trend now. On the other hand, the selling records are stored in the database and provide better review in the future. Therefore, organization can keep track on the changes of sales and analyse whole record for better decision making.

7.6 Conclusion

In conclusion, it is important and beneficial when the system has data integration plan. Data integration makes data more available without considering the physical distance between every subsystem. In addition, it increases the value of data through unified systems. Data from different subsystems merge and collaborate together to perform well in such huge system. Finally, it is a main concern to allow different databases to transfer and receive data between them.

BIBLIOGRAPHY

Fatih Acar (2013). *Oracle DbLink With Odbc Connection to MsSQL Server With FreeTDS*. Retrieved from

<http://www.fatihacar.com/blog/oracle-dblink-with-odbc-connection-from-oracle-to-mssql-server-with-freetds/>

GitHub (2016). *A PostgreSQL foreign data wrapper to connect to TDS databases (Sybase and Microsoft SQL Server)*. Retrieved from

https://github.com/tds-fdw/tds_fdw

GitHub (2016). *PostgreSQL foreign data wrapper for MySQL*. Retrieved from

https://github.com/EnterpriseDB/mysql_fdw

Javaguicodeexample (2016). *An Introduction to Developing Java Web Applications*. Retrieved from

<http://www.javaguicodeexample.com/javawebapplication1.html>

MySQLTutorial (2016). *MySQL Export Table to CSV*. Retrieved from

<http://www.mysqltutorial.org/mysql-export-table-to-csv/>

O7planning (2016). *Create a Simple Java Web Application Using Servlet, JSP and JDBC*. Retrieved from

<http://o7planning.org/en/10285/create-a-simple-java-web-application-using-servlet-jsp-and-jdbc>

Stackoverflow (2011). *Connecting Oracle to SQL Server via database link*. Retrieved from

<http://stackoverflow.com/questions/6112848/connecting-oracle-to-sql-server-via-database-link>

Stackoverflow (2014). *How to create an alert message in jsp page after submit process is complete*. Retrieved from

<http://stackoverflow.com/questions/25286967/how-to-create-an-alert-message-in-jsp-page-after-submit-process-is-complete>

Vidhyasagar (2009). *Creating linked server to MySQL from SQL Server*. Retrieved from

<http://sql-articles.com/articles/dba/creating-linked-server-to-mysql-from-sql-server/>

W3schools (2016). *PHP md5() Function*. Retrieved from

http://www.w3schools.com/php/func_string_md5.asp