2814ICT – DATA MANAGEMENT
7003ICT – DATABASE DESIGN
School of Information & Communication Technology
Trimester 1, 2021

Assignment Part 1:
Designing a Database for BigM

ASSIGNMENT TITLE: Logical Database model for BigM

Student 1	**s-number: s5210032**	**Full name: Jingdi Lin**
Student 2	**s-number: s5204340**	**Full name: Heang SOK**
Course Code: 7003ICT		**Workshop/Lab day & time: 8am Thu/ 8am Wed**
Sessional's name: John Wang/ Fahimeh Alaei		**Date submitted: 23-04-2021**

Table of Contents

List of Illustrations

Statement of Completion:

All tasks have been completed.

Acknowledgements:

1) Mohammad Awrangjeb
2) John Wang
3) Fahimeh Alaei
4) Nosheen munir

Entity Relationship Diagram

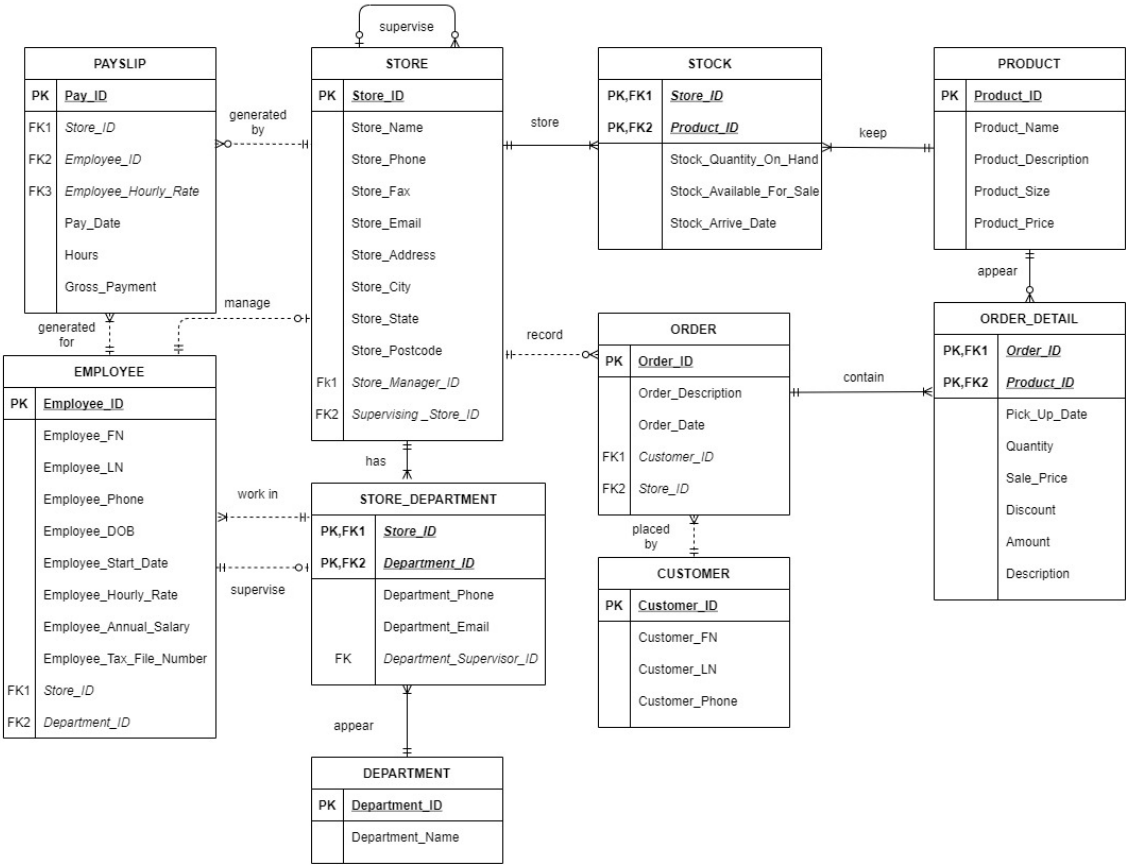


Figure 1: Entity Relationship Diagram

Assumptions

- One store may supervise one or more stores. A store may also supervise no store if no need. A store may not be assigned a supervising store.
- One store must have one or more departments.
- One kind of departments can appear in one or many stores.
- One or many employees can work in one department of a store.
- One employee may supervise one department. Not every employee supervises a department. A department must be supervised by at least one employee.
- One employee may manage one store. Not every employee manages a store. A store must be managed by only at least employee.
- A supervising store generates pay slips for all employees. Not every store can generate pay slips.
- One or many pay slips can be generated for each employee.
- A store can store one or many kinds of products.
- One kind of product can be kept by one or more stores.
- One customer must place at least one orders.
- A store can record one or many orders. A store may receive no order. An order should be recorded by at least one store.
- An order may contain one or more kinds of product.
- One kind of product can appear in one or more orders. One product may not be ordered.

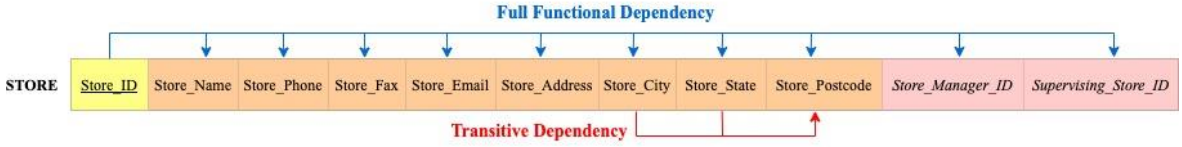
Normalisation

a) Relation Schema

1. STORE (Store_ID, Store_Name, Store_Phone, Store_Fax, Store_Email, Store_Address, Store_City, Store_State, Store_Postcode, *Store_Manager_ID, Supervising_Store_ID*)
2. STORE_DEPARTMENT (*Store_ID*, *Department_ID*, Department_Phone, Department_Email, *Department_Supervisor_ID)*
3. DEPARTMENT (Department_ID, Department_Name)
4. PAYSLIP (Pay_ID, *Store_ID, Employee_ID, Employee_Hourly_Rate*, Pay_Date, Hours, Gross_Payment*)*
5. EMPLOYEE (Employee_ID, Employee_FN, Employee_LN, Employee_Phone, Employee_DOB, Employee_Start_Date, Employee_Hourly_Rate, Employee_Annual_Salary, Employee_Tax_File_Number, *Store_ID, Department_ID)*
6. STOCK (*Store_ID*, *Product_ID*, Stock_Quantity_On_Hand, Stock_Available_For_Sale, Stock_Arrive_Date)
7. PRODUCT (Product_ID, Product_Name, Product_Description, Product_Size, Product_Price)
8. ORDER (Order_ID, Order_Description, Order_Date, *Customer_ID, Store_ID*)
9. ORDER_DETAIL (*Order_ID*, *Prouduct_ID*, Pick_Up_Date, Quantity, Sale_Price, Discount, Amount, Description)
10. CUSTOMER (Customer_ID, Customer_FN, Customer_LN, Customer_Phone)

b) Normalisation

1. For **STORE** table, the dependency diagram is:



This table is in a 2NF because it has a transitive dependency, yet it does not have partial dependency:

- **Full Functional Dependency:** Store_ID → Store_Name, Store_Phone, Store_Fax, Store_Email, Store_Address, Store_City, Store_State, Store_Postcode, *Store_Manager_ID, Supervising_Store_ID*
- **Transitive Dependency:** {Store_City, Store_State} → Store_Postcode

The relationship of transitive dependency among Store_ID, Store_City, Store_State, and Store_Postcode is shown as:

- Store_ID → Store_City, Store_State
- {Store_City, Store_State} → Store_Postcode

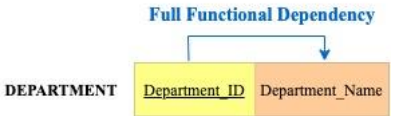
Since this table includes the transitive dependency, it is not in 3NF. However, there is no need to decompose this table into two because Store_Postcode does not lead to big redundancy.

2. For **STORE_DEPARTMENT** table, the dependency diagram is:

This table is in 3NF because it comprises only the full functional dependency, and there is no sign of partial and transitive dependencies:

- **Full Functional Dependency:** {*Store_ID*, *Department_ID*} → Department_Phone, Department_Email, Department_Supervisor_ID

3. For **DEPARTMENT** table, the dependency diagram is:



This table is already in 3NF because it comprises only the full functional dependency, and there is no sign of partial and transitive dependencies:

- **Full Functional Dependency:** Department_ID →Department_Name

4. For **PAYSLIP** table, the dependency diagram is:

The **PAYSLIP** table is in 2NF because it comprises of a transitive dependency and does not have partial functional dependency:

- **Full Functional Dependency:** Pay_ID → *Store_ID*, *Employee_ID*, *Employee_Hourly_Rate*, Pay_Date, Hours, Gross_Payment
- **Transitive Dependency: {***Employee_Hourly_Rate*, Hours} → Gross_Payment

The relationship of transitive dependency among Pay_ID, *Employee_Hourly_Rate*, Hours, and Gross_Payment is shown as:

- Pay_ID → *Employee_Hourly_Rate*, Hours
- {*Employee_Hourly_Rate*, Hours} → Gross_Payment

For this table, it is acceptable to be in 2NF because Gross_Payment does not introduce big redundancy (only one attribute). Therefore, there is no need to decompose this table into two.

5. For **EMPLOYEE** table, the dependency diagram is:

The **EMPLOYEE** table is in 3NF because it comprises only the full functional dependency, and there is no sign of partial and transitive dependencies:

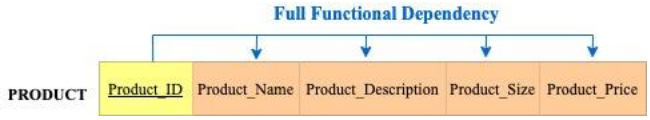
- **Full Functional Dependency:** Employee_ID → Employee_FN, Employee_LN, Employee_Phone, Employee_DOB, Employee_Start_Date, Employee_Hourly_Rate, Employee_Annual_Salary, Employee_Tax_File_Number, *Store_ID*, *Department_ID*

6. For **STOCK** table, the dependency diagram is:

This table is already in 3NF because it comprises only the full functional dependency, and there is no sign of partial and transitive dependencies:

- **Full Functional Dependency:** {*Store_ID*, *Product_ID*}→ Stock_Quantity_On_Hand, Stock_Available_For_Sale, Stock_Arrive_Date

7. For **PRODUCT** table, the dependency diagram is:



This table is already in 3NF because it comprises only the full functional dependency, and there is no sign of partial and transitive dependencies:

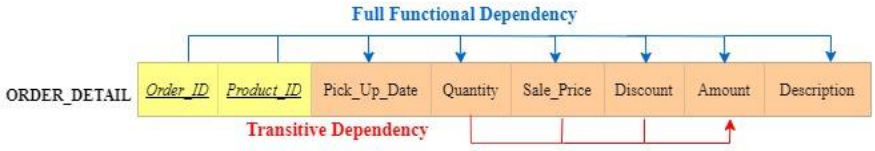
- **Full Functional Dependency:** Product_ID → Product_Name, Product_Description, Product_Size, Product_Price

8. For **ORDER** table, the dependency diagram is:

This table is already in 3NF because it comprises only the full functional dependency, and there is no sign of partial and transitive dependencies:

- **Full Functional Dependency:** Order_ID → Order_Description, Order_Date, *Customer_ID, Store_ID*

9. For **ORDER_DETAIL** table, the dependency diagram is:



This table is in a 2NF because it has a transitive dependency, yet it does not have partial dependency:

- **Full Functional Dependency:** {*Order_ID*, *Product_ID*} →Pick_Up_Date, Quantity, Sale_Price, Discount, Amount, Description
- **Transitive Dependency:** {Quantity, Sale_Price, Discount} → Amount

The relationship of transitive dependency among *Order_ID*, *Product_ID*, Quantity, Sale_Price, Discount, Amount is shown as:

- {*Order_ID*, *Product_ID*} → Quantity, Sale_Price, Discount
- { Quantity, Sale_Price, Discount} → Amount

Since this table includes the transitive dependency, it is not in 3NF. However, there is no need to decompose this table into two because Amount does not lead to big redundancy (only one attribute).

10. For **CUSTOMER** table, the dependency diagram is:

This table is already in 3NF because it comprises only the full functional dependency, and there is no sign of partial and transitive dependencies:

- **Full Functional Dependency:** Customer_ID → Customer_FN, Customer_LN, Customer_Phone

Appendix A

This page is not application.

Bibliography

Coronel. C., & Morris. S. (2017). *Database systems: Design, implementation, and management* (12th ed.). Retrieved from http://www.cengage.com

Oracle. (n.d.). Database. Retrieved from https://www.oracle.com/database/what-is-database/

Study Tonight. (n.d.). Normalization of database. Retrieved from https://www.studytonight.com/dbms/database-normalization.php