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: s5196431**	**Full name: Duwon Ha**
Student 2	**s-number: s5213262**	**Full name: Shinzo Tanimoto**
Student 3	**s-number: s5203114**	**Full name: Kavya Krishnakumar**
Course Code: 2814ICT		**Workshop/Lab day & time: Friday / 9am-10:45am**
Sessional's name: Emon Kumar Dey		**Date submitted: 22th April 2021**

Table of Contents

Table of Contents

List of Illustrations

Acknowledgements:

1) **Emon Kumar Dey**

Entity Relationship Diagram

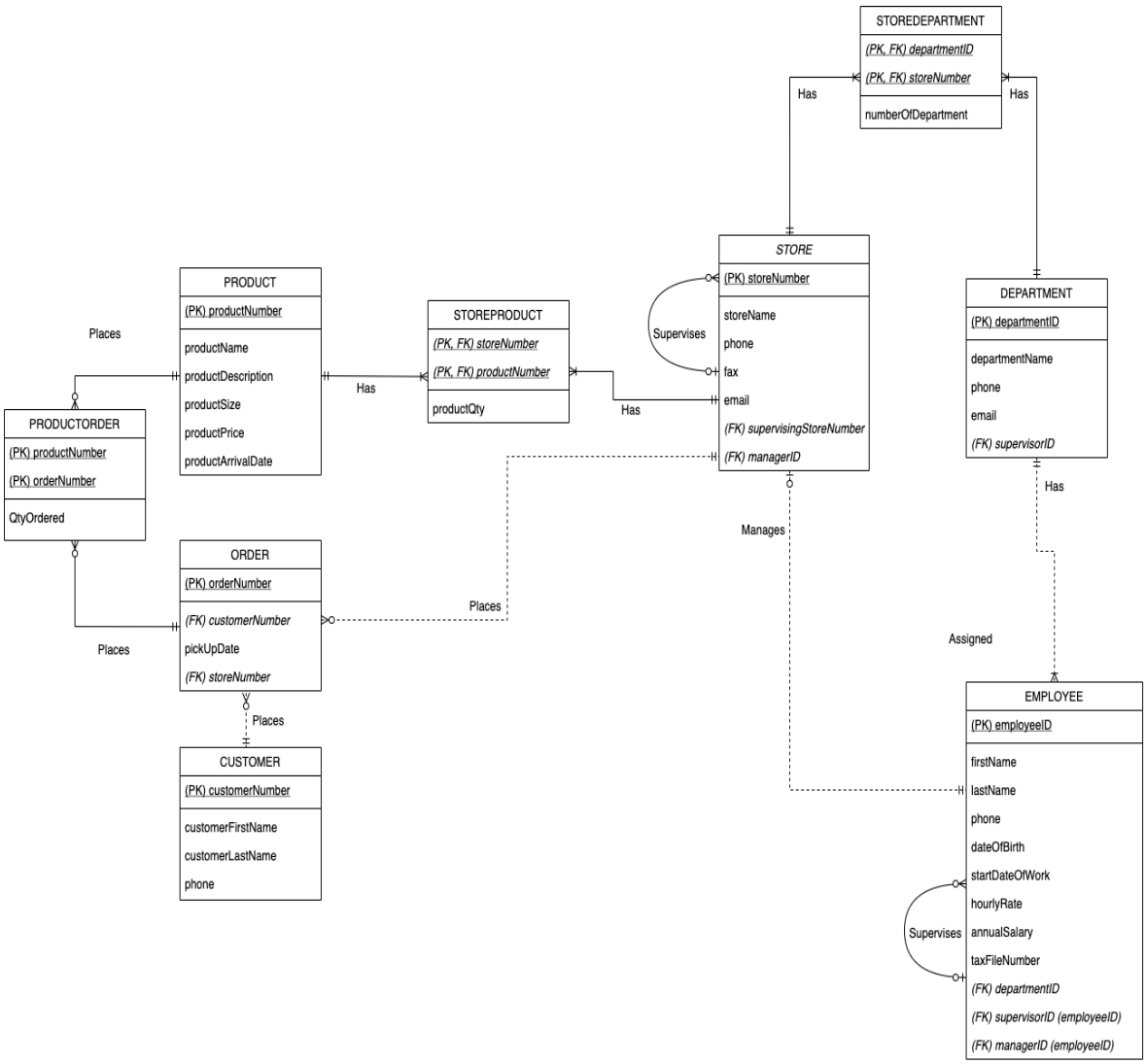


Figure1: Entity Relationship Diagram

Assumptions

• Store Managers manage all supervisors in the store.

• One employee can only work in one department.

• Customers can only pick up orders as a whole, not as individual products.

• Every order has a unique order number assigned to it which is the primary key.

• When an order is taken, date identifying when the order will arrive at the store will be noted under the name orderArrivalDate.

• Similarly, date identifying when the order is ready to be picked up by the customer at the store is noted under pickUpDate.

• Big M does not make any deliveries.

Normalisation

a) Relation Schema

1. PRODUCT (productNumber, productName, productDescription, productSize, productPrice, productArrivalDate)
2. PRODUCT ORDER (productNumber, orderNumber, qtyOrdered)
3. ORDER (orderNumber, *customerNumber*, pickUpDate, *storeNumber*)
4. CUSTOMER (customerNumber, customerFirstName, customerLastName)
5. STORE PRODUCT (*storeNumber*, *productNumber*, productQty)
6. STORE (storeNumber, storeName, phone, fax, email, *supervisingStoreNumber*, *managerID*)
7. STORE DEPARTMENT (*departmentID*, *storeNumber*, numberOfDepartment)
8. DEPARTMENT (departmentID, departmentName, phone, email, *supervisorID*)
9. EMPLOYEE (employeeID, firstName, lastName, phone, dateOfBirth, startDateOfWork, hourlyRate, annualSalary, taxFileNumber, *departmentID*, *supervisorID*, *managerID*)

b) Normalisation

Store

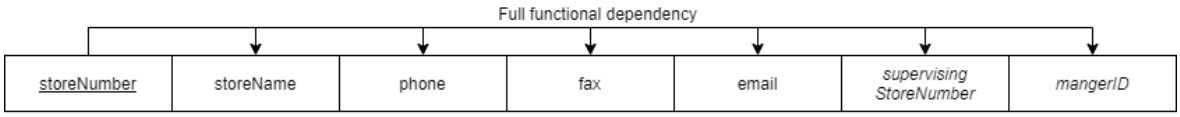


Table: **Store**

This table is in 3NF because it has full dependency without any transitive and partial dependencies.

- **Full Dependency:** storeNumber → storeName, phone, fax, email, *supervisingStoreNumber*, *managerID*

Store Department

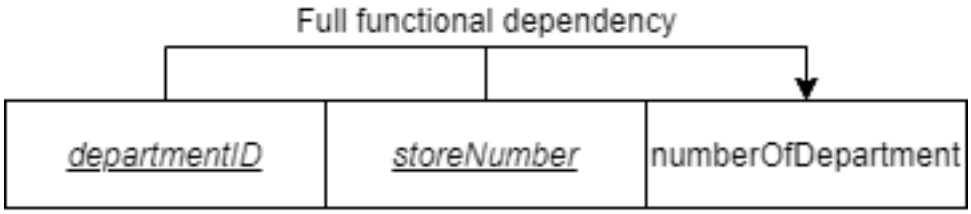


Table: **Store Department**

This table is in 3NF because it has full dependency without any transitive and partial dependencies.

- **Full Dependency:** *departmentID*, *storeNumber* → numberOfDepartment

Department

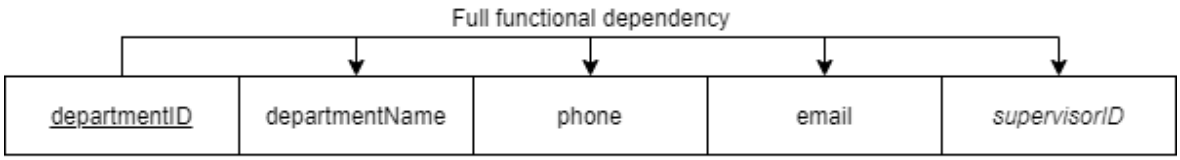


Table: **Department**

This table is in 3NF because it has full dependency without any transitive and partial dependencies.

- **Full Dependency:** departmentID → departmentName, phone, email, *supervisorID*

Product

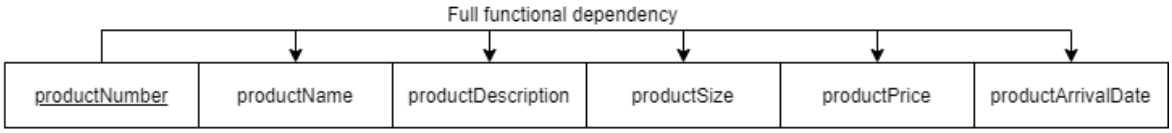


Table: **Product**

This table is in 3NF because it has full dependency without any transitive and partial dependencies.

- **Full Dependency:** productNumber → productName, productDescription, productSize, productPrice, productArrivalDate

Product Order

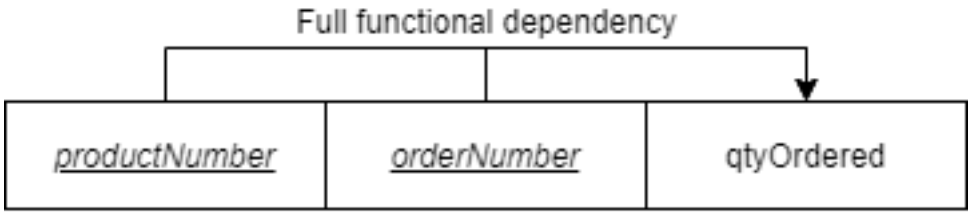


Table: **Product Order**

This table is in 3NF because it has full dependency without any transitive and partial dependencies.

- **Full Dependency:** *productNumber*, *orderNumber* → qtyOrdered

Customer

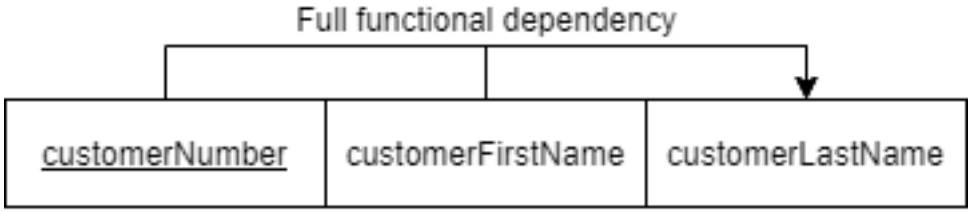
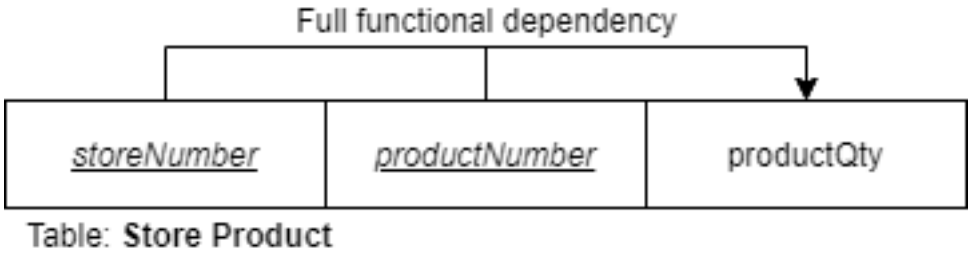


Table: **Customer**

This table is in 3NF because it has full dependency without any transitive and partial dependencies.

- **Full Dependency:** customerNumber → customerFirstName, customerLastName

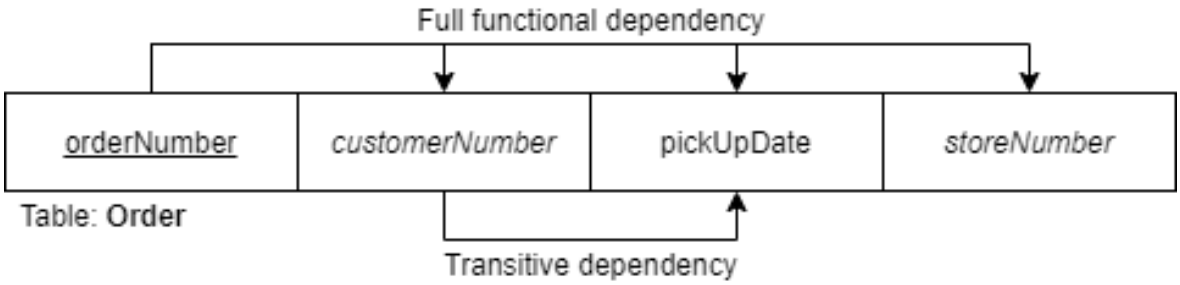
Store Product



This table is in 3NF because it has full dependency without any transitive and partial dependencies.

- **Full Dependency:** *storeNumber*, *productNumber* → productQty

Order



This table is in a 2NF because it has a transitive dependency, but no partial dependency.

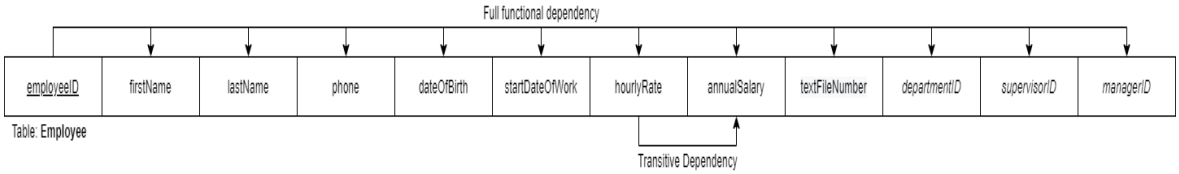
- **Full Dependency:** orderNumber → *customerNumber*, pickUpDate, *storeNumber*
- **Transitive Dependency:** {*customerNumber*} → pickUpDate

There is transition dependency among orderNumber, customerNumber and pickUpDate.

orderNumber -> *customerNumber*
{*customerNumber*} -> pickUpDate

However, there is no need to decompose this table into two because pickUpDate doesn't introduce big redundancy.

Employee



This table is in a 2NF because it has a transitive dependency, but no partial dependency.

- **Full Dependency:** employeeID → firstName, lastName, phone, dateOfBirth, startDateOfWork, hourlyRate, annualSalary, taxFileNumber, *departmentID*, *supervisorID*, *managerID*
- **Transitive Dependency:** {hourlyRate} -> annualSalary

There is transition dependency among employeeID, hourlyRate and annualSalary.

employeeID -> hourlyRate
{hourlyRate} -> annualSalary

However, there is no need to decompose this table into two because annualSalary doesn't introduce big redundancy

Appendices

Bibliography