



Sebastian Huynh
Project 1 Reflection
October 5, 2023
CIS 3050.04
Fall 2023

Table of Contents

1. Table of Contents.....	1
2. Statement of Academic Honesty.....	2
3. Introduction.....	3
4. Project Descriptions and Requirements.....	3
5. Business Rules, Additional Business Rules.....	3
6. EERD Functional Dependencies.....	4-5
7. Lesson Learned and Conclusion.....	6

Statement of Academic Honesty

My name is: Sebastian Huynh, I declare that, except where fully referenced no aspect of this project has been copied from any other source. I understand that any act of Academic Dishonesty such as plagiarism or collusion may result in serious offense and punishments. I promise not to lie about my academic work, to cheat, or to steal the words or ideas of others, nor will I help fellow students to violate the Code of Academic Honesty.

Name: Sebastian Huynh

Date: Oct 5, 2023

Introduction

In this project, I will demonstrate my knowledge of relational database concepts using database modeling tools like Erwin and LucidCharts. Good database design allows for users in the company to have a clear understanding of how data is structured and the design should reflect company expectations.

Project Descriptions and Requirements:

- Developing a normalized relational data model for Quantum Company at the conceptual, logical, and physical model levels.
- Initial Entities required include Employee, Department, Supervisor, and Project.
- Make sure data model fulfills all required and additional business rules.
- Create a relation schema for all entities, which displays the primary key and foreign key integrity between them.
- Explain the dependencies present in the data model along with the process in achieving normal form.

Business Rules and Additional Business Rules:

1. Each DEPARTMENT has exactly one SUPERVISOR.
2. A SUPERVISOR is in charge for one and only one DEPARTMENT.
3. Each DEPARTMENT is assigned at least one EMPLOYEE.
4. Each EMPLOYEE works for at least one DEPARTMENT.
5. Each PROJECT has at least one EMPLOYEE working on it.
6. An EMPLOYEE is assigned to 0 or more Project.
7. A SUPERVISOR supervises at least one EMPLOYEE
8. Each EMPLOYEE has at least one SUPERVISOR
9. A DEPARTMENT must be classified by OPERATIONS or LOGISTICS
10. An EMPLOYEE must be either FULL-TIME or an INTERN
11. A PROJECT consists of at least one TASK
12. Each TASK must belong to only one PROJECT

EERD Functional Dependencies Explained

Partial Dependencies (before any normalization, if the primary keys from both entities form a composite key, partially-dependent attributes rely only on a portion of the composite key):

employee with department:

1. EmployeeID → FirstName, LastName, Address, BirthDate, ContactEmail, EmployeeStatus, YearlySalary, HrsPerWeek
2. DepartmentID → DeptName, DeptLocation, DeptType, ProductionCost, ServiceCost, ServiceRating

employee with project:

1. EmployeeID → FirstName, LastName, Address, BirthDate, ContactEmail, EmployeeStatus, YearlySalary, HrsPerWeek
2. ProjectID → ProjectName, TaskID

employee with supervisor:

1. EmployeeID → FirstName, LastName, Address, BirthDate, ContactEmail, EmployeeStatus, YearlySalary, HrsPerWeek
2. SupervisorID → FirstName, LastName, Address, BirthDate, ContactEmail

Transitive Dependencies (before any normalization, the attribute(s) listed at the end of the flow can be derived from a non-primary attribute instead of only the primary key):

employee with department:

1. EmployeeID → EmployeeStatus → YearlySalary, HrsPerWeek
2. DepartmentID → DeptType → ProductionCost, ServiceCost, ServiceRating

employee with project:

1. ProjectID → TaskID → TaskName

Full Dependencies (attributes listed at the end depend solely on the primary key or all portions of a composite key functioning as the primary key):

employee with department:

1. DepartmentID + EmployeeID \rightarrow AssignedDate
2. EmployeeID \rightarrow FirstName, LastName, Address, BirthDate, ContactEmail, EmployeeStatus
3. DepartmentID \rightarrow DeptName, DeptLocation, DeptType

employee with project:

1. EmployeeID + ProjectID \rightarrow AssignedDate (when an employee was assigned to a project), GroupNumber (identifies a group of employees working on the same project)
2. ProjectID \rightarrow ProjectName
3. EmployeeID \rightarrow FirstName, LastName, Address, BirthDate, ContactEmail, EmployeeStatus

employee with supervisor:

1. EmployeeID + SupervisorID \rightarrow AssignedDate (when a supervisor was assigned over an employee), DisciplinaryAction (the response to an employee's potential misbehavior)
2. EmployeeID \rightarrow FirstName, LastName, Address, BirthDate, ContactEmail, EmployeeStatus
3. SupervisorID \rightarrow FirstName, LastName, Address, BirthDate, ContactEmail

Lessons Learned

The order of parent-to-child entities can affect the way that the entire model comes out to be. I initially started the project with Department as the parent entity in the one-to-one relationship with Supervisor being the child entity. Because Supervisor entity held the foreign key, this caused an issue when attempting to make an associative entity to bridge Supervisor and Employee, as the Department foreign key would be placed into the new entity's composite key against intention.

Conclusion

With the rise of huge amounts of data present for businesses to use in order to analyze and make decisions for firm survival and success, the implementation of the database approach along with the proper understanding and improvement on relational database designs allows for companies to keep a competitive edge in an increasingly interconnected and globalized market.