COMP353: Databases
Final Report

Team: lzc353_4

Prof. N. Shiri

Gabriel Harel 40006469
Philippe Beaudry 27883676
Emanuel Mateus 40004325
Daniel Stroppolo 40006025

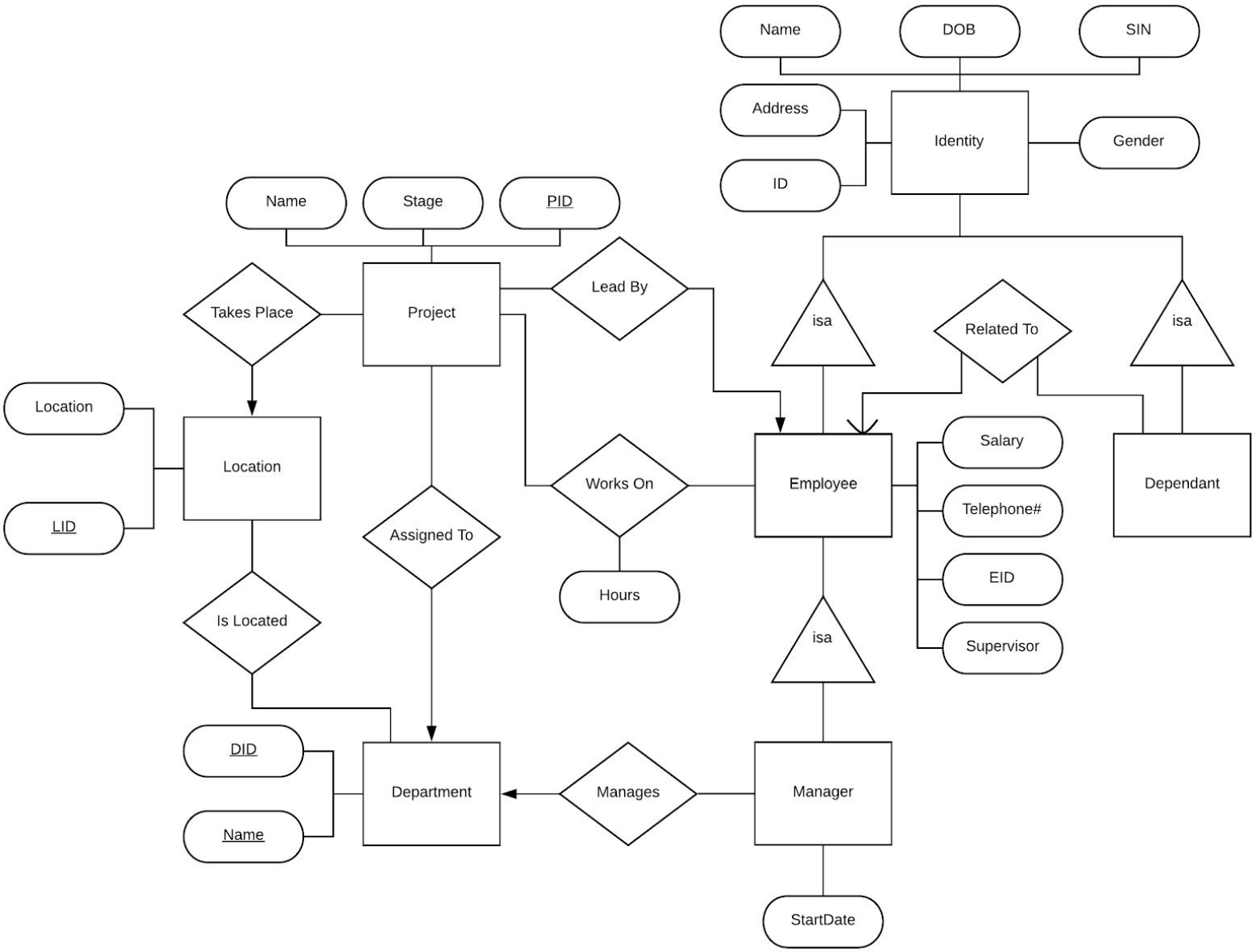
April 11, 2018

Assumptions

Some assumptions were made for the proposed database design that affect the structure. First off, it is assumed that every manager is also an employee, who are both identities. Dependents also are an Identity. It is also assumed that an employee can only be assigned to one department and that a manager may only manage a single department.

Looking at projects it is assumed that each project can only happen in one location and that they can only be assigned to one department, whereas an employee can work on multiple projects but can only be the lead on one project. For departments it is assumed that every for every location there is one department.

E/R Diagram



Full size diagram included in file: COMPANY3.png

Normalization Process

The normalization process usually includes the decomposition of tables into their Boyce-Codd normal form to eliminate anomalies that lead to errors in the data. Such anomalies can arise, for example, in situations where attributes are used as the key in a database table. Below is an example of such a situation.

Video Games				
Title	Year	Budget	Length	Main Character
Zelda	2011	12,000,000	14	Link
Mario Party	2006	30,000,000	56	Mario
Mario Party	2006	30,000,000	56	Donkey Kong
Mario Part	2006	30,000,000	56	Boo

In this example, we would first assume the key would be (Title, Year). However, this would not work because multiple tuples exist with an identical key. We would need to add (Main Character) to the key for it to be unique.

This creates a second problem of having multiple tuples with the same information in the table. This can create update anomalies. For example, if an error was made when entering the length, and it needs to be changed it might be possible to update only 1 of the tuples, thus creating inconsistent data in the table (2 of the rows would have the same length, whereas another would be different for the same game).

In the company schema, there is the potential for errors in the tables storing information on people. In our model, we include the following attributes:

SIN	Name	DOB	Gender	Supervisor	Address	Phone	Dept.

These attributes would be attributed to both employees and their dependants. If a single table were made for both employees and their dependants, it could introduce multiple tuples with repeating information such as follows:

Company People								
SIN	Name	DOB	Gender	Supervisor	Address	Phone	Dependant	Dept.
420	Daniel	11/19	M	Allan	Montreal	555-1234	Martha	123
421	Daniel	11/19	M	Allan	Montreal	555-1234	Steve	123
422	Martha	08/07	F		Montreal	555-1254		
423	Steve	03/12	M		Montreal	555-6542		

In this case, the key would need to be (SIN, Dependant), ie. (SIN, Dependant → Name DOB Gender Supervisor Address Phone Dept).

We also need to include managers in our schema. These are the same as employees with the added attribute (start date). This would extend our hypothetical table above to:

Company People									
SIN	Name	DOB	Gender	Supervisor	Address	Phone	Dependant	Dept.	Start Date
420	Daniel	11/19	M	Allan	Montreal	555-1234	Martha	123	01/01
421	Daniel	11/19	M	Allan	Montreal	555-1234	Steve	123	01/01
422	Martha	08/01	F		Montreal	555-1254			
423	Steve	03/12	M		Montreal	555-6542			
424	Dave	04/10	M	Daniel	Ottawa	555-6231	Jessy	123	

This table exhibits similar problems to the previous one.

We will decompose this table into three separate tables, using IDs to achieve BC normal form. The following steps assume a new attribute has been added to the table, which we will call “Identity ID” or “IID” and is unique for each tuple.

By having a single, unique ID for each tuple, the set of functional dependencies becomes:
{ IID → SIN, Name, DOB, Gender, Supervisor, Address, Phone, Dependant, Dept, Start Date}.

We observe the following attributes are not necessary for dependants: Supervisor, Dependant, Department, Start Date and the Start date attribute is only necessary for managers. The following attributes are mandatory for all people: SIN, Name, DOB, Gender.

With this, we can decompose the table into 4 seperate models: Identities, Employee, Dependant, Manager.

Identity				
IID	SIN	Name	DOB	Gender

{ IID → SIN Name, DOB, Gender }

Employee				
IID	Dept.	Supervisor	Address	Phone

{ IID → Dept, Supervisor, Address, Phone }

Dependant	
IID	Employee

{ IID → Employee }

Manager		
EID[1]	Dept	Start Date

{ EID[1] → Dept, Start Date }

[1] Employee ID

The condition for BCNF states that any nontrivial FD in a relation is a superkey. This is the case for all the above relations because there is only one functional dependency and the left hand side is the key.

E/R Model to Relation Conversion

An identity is highest on the hierarchy of people in the company. It's properties are shared across Employees, Dependents and Managers.

Identity (IID, Address, Name, DOB, SIN, Gender)

An employee is an Identity with added properties. The supervisor property is an employee ID and represents the employee's manager, of which there can only be one.

Employee (IID, EID, Salary, Telephone#, Supervisor, Department)

A manager is an employee with the added property of start date and Department ID. A manager can only manage a single department and thus the aforementioned relation is described by the Department ID property.

Manager (EID, DID, Start Date)

Dependents can only exist if there is an employee with whom the dependent is related. They simply have their identity ID and matching employee ID.

Dependent (IID, EID)

Employees are assigned to departments. The assigned to relation is a many employees to one department relationship. This relationship is represented by the DID (department ID) which is a property of employees.

Projects are many to many with employees. This relationship is better represented by a table called Assignments which relates employees to projects they are working on. It has its own property called hours worked which depends on each employee and project pair.

Assignments(EID, PID, hours)

Departments have a name and an ID that is referenced in the manager table.

Department(DID, Name)

Locations have an ID that is referenced from projects. Locations have a many to many relationship with departments. This relationship is denoted by the Department ID in the Location table. They also have the name of the location.

Location(LID, DID, Location)

Projects have a many to one relationship with locations, and thus store the location ID as an attribute. A project also has a many to one relationship with Departments and stores the department ID as an attribute. There is also a many to one relationship with employees, meaning one employee can lead multiple projects but not the other way around. This relationship is stored as an Employee ID under the attribute lead. A project also contains name and stage attributes.

Project(PID, DID, LID, Name, Lead, Stage)

Website Features:

The Company website offers several features in order to to better navigate the records of the Company. The homepage of the website off offers an at a glance look at the Company's records from projects, departments, and employees. Also on the homepage towards the bottom of the page, the website offers a custom reports self-service query feature. What this feature allows the user to do is write their own query on the Company's database. The page will then be displayed with a result set in tabular format if the query was valid. The website also has dedicated pages for the employees, departments and projects with more data related to each of the attributes.

For the employees page a table with the list of employees in the Company is displayed. The page also displays information about the employees displayed in tabular format underneath the list of employees table. This information includes total project work hours, project involvement, and direct subordinates for the related employees.

For the departments page a table with the list of departments in the Company is displayed. The page also displays the information about the departments cost break down in a separate table underneath the list of departments table.

For the projects page a table with the list of projects in the Company is displayed. The page also shows the information about the projects each with their own table from project cost breakdown, project progress, and project participants.

The final page is the admin page which allows the users to manipulate data in the Company database. The user can add or modify records from the employee, departments, and projects tables. They can also log employee hours for a specific project and promote employees to become a department's manager.

Work Distribution

Gabriel Harel and Philippe Beaudry worked on the implementation of the database. That is to say, they designed the website and the query system. Gabriel worked on the admin delete, employee edit and admin log hours features. Philippe worked on the admin edit, and admin create features. The reporting queries and and the rest of the frontend tasks were evenly shared.

Daniel Stroppolo and Emanuel Mateus were responsible for the documentation. An initial E/R model was created by Daniel and later revised by Emanuel. The normalization and model conversion was a joint effort with both teammates working on it together. The layout and presentation of the report was also done as a joint effort.