

CS-A1150 Databases

Project part 2

Yrjö Koski, 526076, yrjo.koski@aalto.fi
Viet Tran, 432267, viet.2.tran@aalto.fi
Max Müller, 588441, max.muller@aalto.fi

May 12, 2020

1 Solution of the first part

1.1 UML table of Construction company

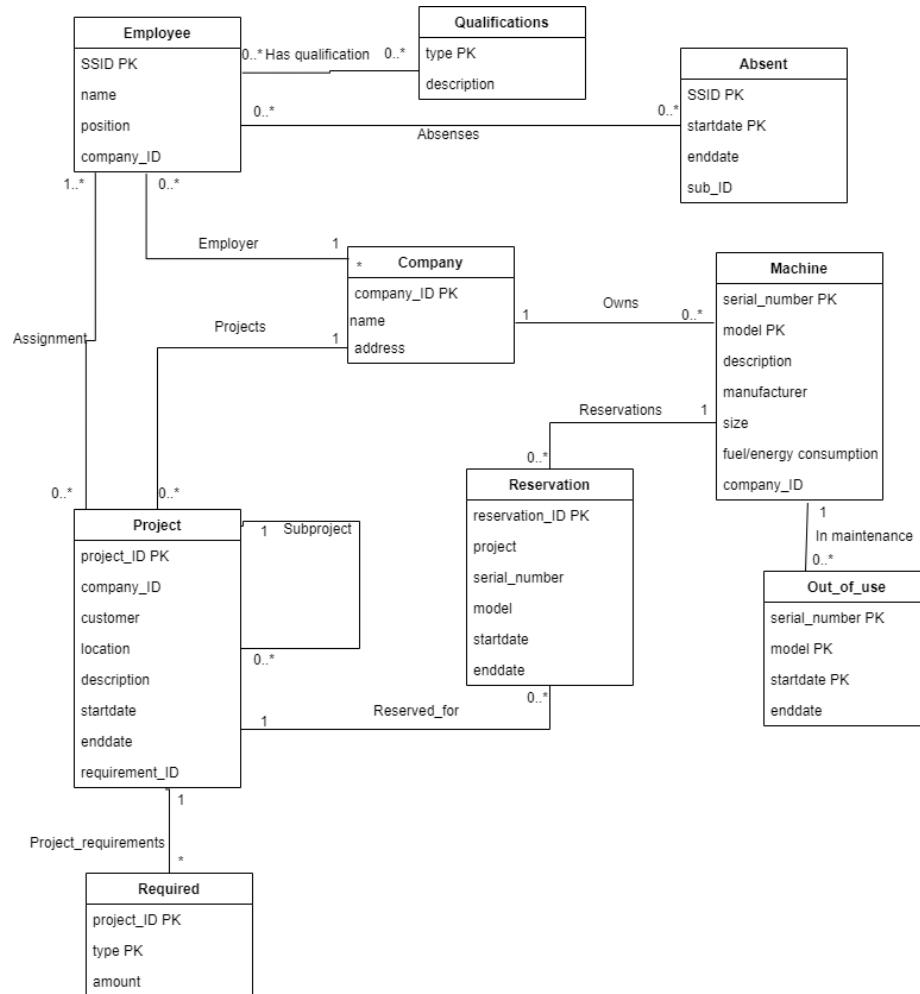


Figure 1: UML table for Construction Company

1.2 Relation tables of Construction company

Employee(SSID, position, name, company_ID)
Has_qualification(SSID, type)
Qualification(type, description)
Absent(SSID, startdate, enddate, sub_ID)
Project(project_ID, mainproject_ID, company_ID, customer, location, startdate, enddate, requirement_ID)
Reservation(reservation_ID, project, serial_number, model, startdate, enddate)
Company(company_ID, address, name)
Machine(serialNumber, model, description, manufacturer, size, fuel_consumption)
Out_of_use(serial_number, model, startdate, enddate)
Assignment(project_ID, SSID)
Required(project_ID, type, amount)
Project_manager(subproject_ID, mainproject_ID)

1.3 Functional dependencies and BCNF of relations

1.3.1 Functional dependencies

In the relation Employee:

$$\{SSID\}^+ = \{SSID, position, name, company_ID\}$$

In the relation Has_qualification:

$$\{SSID\}^+ = \{SSID\}$$

$$\{type\}^+ = \{type\}$$

In the relation Qualification:

$$\{type\}^+ = \{type, description\}$$

In the relation Absent:

$$\{SSID\}^+ = \{SSID, startdate, enddate, sub_ID\}$$

In the relation Project:

$$\{project_ID\}^+ = \{project_ID, company_ID, customer, location, description, startdate, enddate, requirement_ID\}$$

In the relation Reservation:

$$\{reservation_ID\}^+ = \{reservation_ID, project, serial_number, model, startdate, enddate\}$$

In the relation Company:

$$\{company_ID\}^+ = \{company_ID, address, name\}$$

In the relation Machine:

$$\{serial_number, model\}^+ = \{serial_number, model, description, manufacturer, size, fuel_consumption\}$$

In the relation Out_of_use:

$$\{serial_number, model, startdate\}^+ = \{serial_number, model, startdate, enddate\}$$

In the relation Assignment:

$$\{project_ID\}^+ = \{project_ID\}$$

$$\{SSID\}^+ = \{SSID\}$$

In the relation Required:

$$\{project_ID, type\}^+ = \{project_ID, type, amount\}$$

In the relation Project_manager:

$$\{subproject_ID\}^+ = \{subproject_ID, mainproject_ID\}$$

1.3.2 Anomalies and redundancies of relations

Deleting project, company, qualification causes deletion anomalies as removing company there is no knowing which company the project is contracted to, which machine is owned by which company and there is no employer for the employee. Project is needed for Required and Reservation tables and for assignments for employees.

Having a nonexistent type of qualification in the required table can halt the project as there are no people who has the needed qualifications and the project can't start.

Having a nonexistent project attribute in Reservation table can cause a machine's reservation in a deadlock if the known enddate isn't known and the machine isn't released.

There are four relations where there is company's business ID as attribute as we want to know in which company the employee works in and to which company project is contracted to. We also want to have information which company owns which machine.

Project ID is needed in Required and Reservation tables because we want to connect the required workers with the required qualifications for the assignment and connect the reservation with the project and machine.

Machine's primary keys are used in Out_of_use and Reservation relations as one would like to know which machine is going to be used or is reserved.

Employees' SSID is found in Employees, Assignment and Absent relations as SSID is needed in knowing if the employee is assigned to which projects and if the employee's absent and have a record who is substituting the regular employee.

1.3.3 BCNF

All the relations are in the BCNF format since the primary keys determine all other attributes in a class with no other dependencies in any classes. The

relations `Has_qualification` and `Assignment` are an exception as they have no dependencies at all.

1.4 Explanation of solution

The database we have designed can store the necessary information of a construction company. It can be used to manage projects, divide larger projects into subprojects and handle employee and machine information and assignments. In this section We'll discuss how we came up with our UML model and relational data model and how our design can function as desired.

1.4.1 Description of relations

Company relation has the basic information about the construction company. It will contain information about the company's company_ID (PK) (y-tunnus in Finnish), name and address and it will be linked to the employees, to the company's machines and to company's projects via company_ID attribute. A company can have any amount of employees, machines and projects.

The employee relation has Social security number as a primary key. It will also contain name and position of the employee. It is connected to company via company_ID attribute. The qualifications of each employee are stored in "Has_qualification" relation which has Qualification type and employee's SSID as Primary keys. The Qualifications relation has type as its primary key and a short description of the qualification. The type is a the qualification title e.g. carpenter. An employee can/ be absent for various reasons. The absences of employees are stored in Absent relation which has employee SSID and start date as a primary key and it also holds information about the end of absence and the SSID of a substitute employee.

The Machine relation holds information about a machine owned by a company. A combination of a serial_number and a model form the primary key of each machine (different manufacturer's may have overlapping serial numbers for different machines). Machine also has description of it's usage, manufacturer, size, energy_consumption and it's linked to the company via company_ID. The relation Out_of_use holds information about machine's that are out of use. It's primary key consists of a serial number and type of a machine and starting date of its out of use time. It also holds information about the end date i.e. the time a machine was fixed. This way a certain machine can be marked as out-of-use multiple times.

The projects of a company are stored in Project relation. The primary key is project_ID which is an identifier of a project. The relation also has attributes: company_ID which links the project to the company, customer which gives information about the client who ordered the project, location of the project site which may be different if different subprojects have different locations, description what is being done, startdate of the project, enddate of a finished project and required_ID which contains the project ID of a project that needs to be finished before the current one can be started. We'd like to point out

that the project and subproject are contained within the same relation meaning that there is no separate relation for subprojects. The subprojects are managed via ProjectManager relation which has attributes mainproject_ID and subproject_ID. This way it links the subprojects to a whole project.

We can assign machines and employees to a project by using Assignment and Reservation relations. Assignment relation has attributes ssID (of an employee) and a project ID which combined form the primary key. This way it links an employee to a project/subproject for its duration. The Machines are reserved to a project via Reservation relation which has also information about the reservation time period, since the machine are not necessarily needed for the whole duration of the project. Reservation has attributes reservation_ID (PK), project_ID, machine serial number and model, start date and end date.

To make sure that employees assigned to a project have the necessary qualifications we have a Required relation that has attributes: project_ID, qualification type and amount needed. The primary key consists of project_ID and the qualification type.

1.4.2 Example usage cases

A user can create a new company (or multiple companies) by adding a new company tuple into company relation. New projects can be created by adding them to the project relation with appropriate information. The hierarchy between main and subprojects can be handled with ProjectManager relation as described previously. New employees can be added to Employee relation and their qualifications can be updated with HasQualification and Qualification relations. New machines can be added to Machine relation. It is assumed that the same models have different serial numbers so that by using combination of serial number and model we can indentify both totally new machines and new items of already existing models.

The projects can be managed with Project and ProjectManager relations. Ongoing projects can be found by status attribute of project. It can mean that the project is finished, ongoing or pending. The temporal order of projects can be found by looking at requirement_ID of Project tuples.

To make sure that every project has employees with correct qualifications we can compare the Required relation of a project with the qualifications of employees assigned to that project via the Assignment relation. Similarly for checking the correct machinery for each project we can use the Reservation relation. As mentioned before, here we'll also specify the time period of reservation.

By looking at the Assignment and Reservation relations we can see the current state of each employee and each machine. We can see if there are people who are not assigned to any projects at a time or machines that are available

at a certain time. To see absences for personnel and maintenances for machines we can use Absent and Out_of_use relations. To see the time periods for when people with certain qualifications are available we can get the time duration of current or future assignment for an employee from project's time window and we can reference that to the employees qualifications.

If we are interested to see what an employee has done in the past we can look at which projects he/she has been assigned previously and get the information of the projects by their project id. This project id can be further used to get information about other employees that worked at that project and the machinery that was used.

2 Solution of the second part

2.1 Modifications and supplements of first part

2.2 Updated UML table

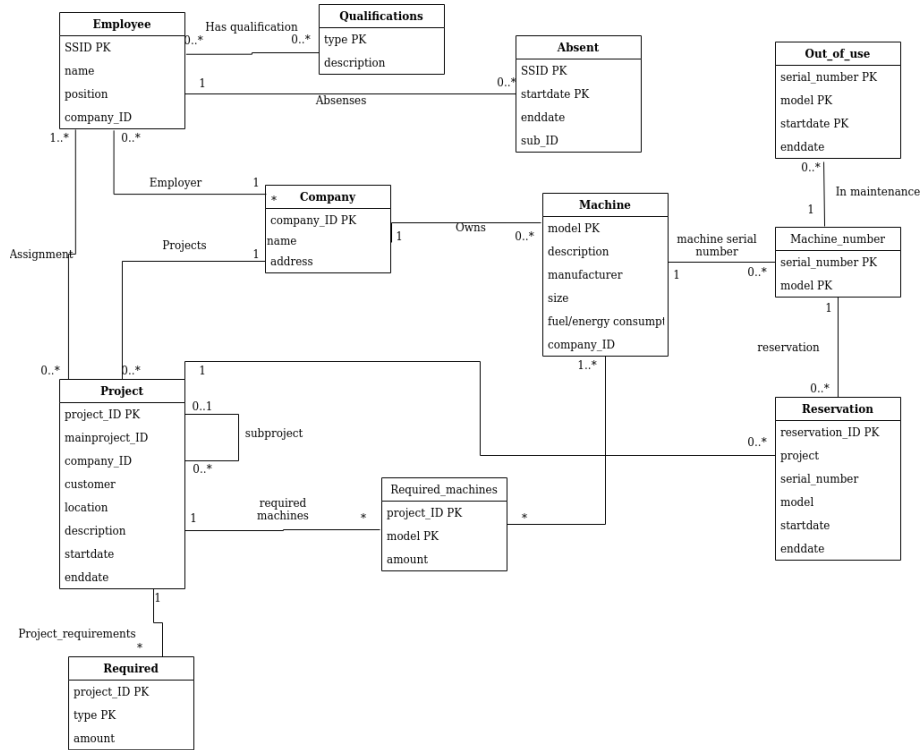


Figure 2: UML table for Construction Company

The differences:

Separated the different serial numbers of a single machine model as a Machine_numbers table.

Added the table Required_machines to represent the need of certain machine models in a project.

Linked the required machines of a project to a certain machine model and the reservations of a project to a distinct serial number of that model.

Added the column main_project to the project table that is used by a sub project to reference its main project.

Linked the Out_of_use table to a specific machine.

2.3 Updated Relation tables

Qualification(type, description)

Has_qualification(SSID, type)

Company(company_ID, address, name)

Employee(SSID, position, name, company_ID)

Project(project_ID, main_project, company_ID, description, customer, location, startdate, enddate)

Absent(SSID, startdate, enddate, sub_ID)

Machine(model, description, manufacturer, size, fuel_consumption)

Machine_number(serial_number, model)

Reservation(reservation_ID, project, serial_number, model, startdate, enddate)

Out_of_use(serial_number, model, startdate, enddate)

Assignment(project_ID, SSID)

Required(project_ID, type, amount)

Required_machine(project_ID, model, amount)

The Differences:

We removed the Project_manager relation since it did not work in the way we intended it to and moved its function to the project relation instead in the form of the column main_project. If the column main_project is not 0 it means that the project in question is a sub project.

We also removed the requirement_ID column since it was not necessary.

We added the separate Machine_number Relation in order to divide the machines into models and specific units of those models.

We added the Required_machines relation to represent the need of certain machine models for a project where we had previously used the reservations to both reserve the machines and represent their need in a certain project. Now the reservations is a separate relation.

2.4 Creating the tables

2.5 Example use cases

We want to create an example database and do some queries on it to show how it works in action. We'll use every relation that we have and do some queries to fetch and combine data from our relations. Some of the queries will fail on purpose to show some of the constraints. We will go through them so that nothing should fail without a reasonable explanation.

First we will set up a company. We'll insert a company called "Work bois" with ID 123 and with Address "Moonlight avenue 14". This should be done first because without an existing company we can't add any employees or machines due to foreign key references.

Next, we'll add four employees with different SSIDs, positions and names but with the same company ID. Then we add 4 different qualifications with a simple descriptive type such as "Excel skills" and a more detailed explanation. We insert Has_qualification relations with employee SSIDs and Qualification types so that each employee has at least one qualification, where some have multiple.

Then we'll move on to machines. We insert three different types of machines each with their own model, description, maker, size and fuel consumption. Fuel consumption is an integer value and all others are text. They all have the same company ID since they belong to the same company. We also insert Machine_number relations to keep track of individual machines. Our company has two pieces of excavator with model "E-001" and 1 piece of other machines.

Next, we'll need some projects for the company. We insert two projects with some subprojects. The first one is "Rewiring of electrics and waterpipe installation of a summer cottage" which is done for a client "Mr. Virtanen" with his personal address. It's start and end dates are defined and its project ID is set to 700. This is further divided into 3 subprojects: wiring, digging and installing pipes. Each have their own project IDs and own employee and machine requirements. The second project is "demolition of and old storage building" with two subprojects.

Next we set up the required qualifications, assigned employees, required machines and machine reservations for our projects. For the first project we need someone with Electrical installation qualification for wiring subproject, someone with Machine operation qualification for the digging subproject and Plumbing skills for the pipe installation subproject. We assign one person with electrical installation and machine operation qualifications to do the first and the second subproject and one employee with a plumber qualification to do the third subproject. We need an excavator to dig ground to get to the water pipes in the second subproject and a pipe connector to install pipes in the third subproject.

We assume that we need the machines for the whole durations of the subprojects so we create reservations for them by inserting a new tuple to Reservation relation. Now the project is properly set up.

The second project is set up similarly. The first subproject is paperwork at the company's office which requires someone with Excel skills. We add a requirement and assign an employee with proper qualification. The second subproject is demolition at client's address which needs some machines and someone with Machine operation qualification. We add the necessary qualification to the Requirements and necessary machines to required_machines. We assign a qualified employee on the project and add reservations for the machines. This time we don't need an excavator for the whole duration while we need a hammer machine for the whole time so we set up different start and end dates for the reservations. Now this project is also set up.

A construction site is always in motion and things won't always go as planned. In the first project the excavator breaks down in the middle of the second subproject. We insert a tuple to Out_of_use relation to handle the situation. The excavator is in the shop for one week to get repaired so we add start date to the time it went out of order and the end date for a week later. We then reserve the other excavator of the company to the project duration. In the second project one employee becomes ill. We use Absent relation to mark the sick leave. We then assign the substitute worker to the subproject.

Now we have inserted data to all our tables and everything works as planned. Then let's move on to search queries.

These INSERTs and our SELECT queries can be found in our .sql file. We commented what our SELECT queries do in the .sql instead of explaining them here to minimize repetition and to offer a clear representation. With our queries we try to show what kinds of operations are possible with our implementation and what kind of key restrictions are in place to prevent misuse of the database.

2.6 Possible improvements

At the moment our database does not deal well with deleting and updating data. We have not implemented any DELETE CASCADE or UPDATE CASCADE commands since it was not in the scope of this project but for a practical implementation these should be considered. For even better event handling we could implement triggers which could perform various actions when data is inserted, updated or deleted.