

ECE356 F20 – Lab 2 – Logical Schema Design

Due date: October 30th at 11pm Waterloo time

Group work policy: In this lab, students are expected to work in groups no larger than 3. Groups of 1 or 2 are fine.

Weight: This lab is worth 10% of your final course grade (1/5 of the weight in the Assignments/Labs category).

Objective: In this lab, you are given a database model for the Employee database. We ask that you apply database design principles to improve this model. Specifically, you will need to:

- Identify primary and foreign key constraints in the database model.
- Identify any non-atomic value domains.
- Identify any redundancies due to functional dependencies.
- If needed, normalize the schema to achieve 3NF and BCNF.

Submission: All parts of this lab must be submitted to the Lab 2 dropbox in LEARN by the due date, which is shown below the title of this document. **You must enroll in a Lab 2 group in LEARN before you can submit to the group dropbox.** Your submission will contain three separate files since the lab manual has three parts; each part of the manual describes the required submission format. Should you choose to resubmit the deliverables prior to the deadline, please ensure that you **upload all three files each time** as the dropbox is programmed to overwrite the old submission with the new.

Overview of Employee Database

In Lab 1, you worked with the Employee database. The schema for Lab 1 had four tables: Employee, Project, Assigned and Department. For Lab 2, we provide a modified version of the Lab 1 schema in the starter code file *part1.sql*. The highlights of the new data model are as follows:

- The Employee name is a compound attribute comprising a first name (emp_fname), last name (emp_lname), and initials (emp_initials).
- The Department location is a single attribute comprising the street number (streetNum), street name (streetName), city name (cityName), province (province), and postal code (postalcode).
- The Assigned table has additional columns: roleID, start_date, end_date, assigned_status, and role_level.

The tables in the Lab 2 model are illustrated in Figure 1. The primary and foreign keys are missing in this figure because adding them is a requirement of this lab.

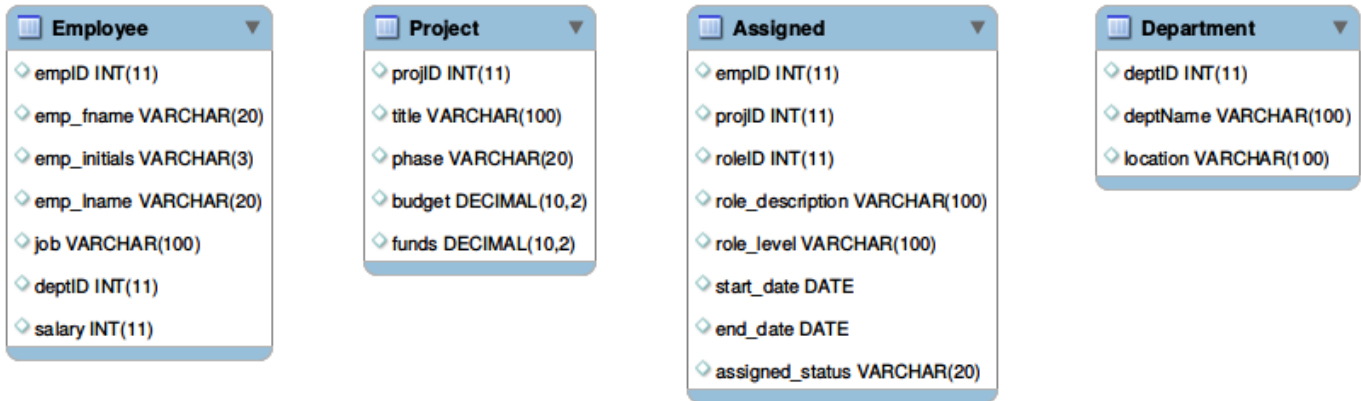


Figure 1: Initial schema for Lab 2.

Table 2 describes the functional dependencies that you should consider for this lab.

Table Name	Functional dependencies
<i>Department</i>	deptID \rightarrow deptName, location
<i>Project</i>	projID \rightarrow title, phase, budget, funds
<i>Employee</i>	empID \rightarrow emp_fname, emp_initials, emp_lname, job, deptID, salary
<i>Assigned</i>	empID, projID, roleID \rightarrow role_description, role_level, start_date, end_date, assigned_status roleID \rightarrow role_description, role_level

Table 2: Functional dependencies for the schema in Figure 1.

Part 1: Adding Primary Key and Foreign Key Constrains (10%)

Add to the database schema all the relevant primary key constrains and foreign key constrains, as determined using the information provided below:

- 1) The empID uniquely identifies one employee.
- 2) The deptID uniquely identifies one department.
- 3) The projID uniquely identifies one project.
- 4) An employee may be assigned to any number (including zero) of projects.
- 5) An employee **must** be assigned to at least one role in a project.

Using the provided *part1.sql* file as the starting point, modify the SQL DDL statements to incorporate the required constraints. Do not add or remove any tables or columns in this part of the lab. Also, please preserve the drop table statements at the top of the *part1.sql* file so that the script can be run repeatedly without raising errors.

Submission instructions: For this part of the lab, the deliverable is your modified version of the provided *part1.sql* file with the appropriate primary and foreign key constraints added. This is one of several files you will submit for Lab 2.

Part 2: Normalization to 1NF, 3NF, BCNF (80%)

In this part of the lab, the starting point is your modified schema from Part 1, which includes the primary keys and foreign keys you added. You will transform this schema by applying different normalization procedures. For each of the questions below (A, B1, and B2), you will need to:

- Decide if the schema already satisfies the required normal form.
- If not, apply the techniques taught in lecture to transform the schema so that it satisfies the required normal form.
- Draw a relational schema diagram, similar to Figure 1 but showing primary and foreign keys, representing your normalized schema.

A) What modifications to your schema from Part 1, if any, are needed to ensure that it is in 1NF? (10%)

B1) What modifications to your 1NF schema from (A), if any, are needed to ensure that it is in 3NF? (35%)

B2) What modifications to your 1NF schema from (A), if any, are needed to ensure that it is in BCNF? (35%)

Note 1: For A, if you find that the schema is not in 1NF then address the problem **without adding any tables to the schema**.

Note 2: For B1 and B2, please show all your work including the 3NF and BCNF test discussed in lecture, as well as the steps of any decomposition, if applicable. The grader will deduct marks for missing details. If you are using the simplified 3NF or BCNF test, please justify why such a test is applicable.

Note 3: B1 and B2 both build on A, which builds on your schema from Part 1. Do **not** use the schema from B1 as the starting point for B2, or vice versa. (I.e., do not simply use the argument that a BCNF schema also satisfies 3NF.)

Note 4: When analyzing the schema for different normal forms, you may ignore any redundancy (e.g., due to non-atomic value domains or due to functional dependencies) related to postal codes.

Submission instructions: For this part of the lab, please write a report and save it as a PDF file named *part2.pdf*. This document will be graded manually, so please **include the names and student IDs** of all group members on the title page. The document should be submitted to the same dropbox as your SQL script from Part 1. The dropbox will be configured to accept multiple files per submission.

Part 3: SQL DDL for BCNF (10%)

Write SQL DDL code for the BCNF schema you created in Part 2 question B2. Use your modified *part1.sql* file from Part 1 as the starting point, and save the new script in a file named *part3.sql*. Note that the script contains several INSERT INTO statements, which must be modified to accommodate any changes you have made to the schema.

Submission instructions: For this part of the lab, the deliverable is your *part3.sql* file containing a script that creates the BCNF schema. This is the third and final file required for submission.