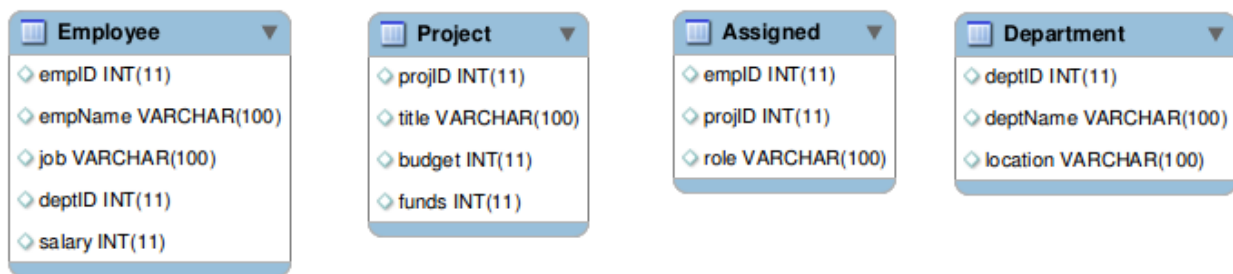


ECE 356 Winter 2019: Lab 2

Goals: The goals of this lab are to

- (a) Recognize when a database is not in BCNF
- (b) Decompose a database into BCNF, creating the necessary tables
- (c) Create and invoke a stored procedure
- (d) Use explain to understand the database query plan
- (e) Create indexes to improve the database query plan

Part 1: Consider the database from Part 1 of Lab 1:



The constraints on the attributes and functional dependencies between attributes were left undefined in that schema. We now define them as follows:

- (1) empID is unique per employee
- (2) projID is unique per project
- (3) The empName is the full name of each employee, and consists of a First and Last Name, and optionally a Middle Name.
- (4) The deptID is unique per departments
- (5) A department location is a full address, comprising the street number, street name, city name, province, and postal code
- (6) An employee may have more than one role on a project
- (7) An employee may be assigned to more than one project
- (8) A department may have multiple locations
- (9) An employee can be in more than one department

You are required to take this initial database schema and:

- (1) decompose it, per BCNF
- (2) determine primary keys
- (3) determine foreign keys
- (4) write the necessary SQL to create tables for this database, together with the necessary primary and foreign keys
- (5) if your decomposition has resulted in the loss of any of the above four tables, write the necessary SQL to create a view that corresponds to that table (note: in the case of the empName and department location you should use “concat” to create a single attribute from the atomic components)

(6) You are required to create a stored procedure “payRaise” that takes two input parameters “inEmpID” (Int) and “inPercentageRaise” (double 4,2) and one output parameter “errorCode” (int). In normal operation the procedure should raise the salary of the associated employee by the input percentage and return an errorCode of 0. However, if the payRaise is by more than 10% or less than 0% (*i.e.*, it is a pay cut), it should return -1. If the employee does not exist, it should return an errorCode of -2. You should create the necessary query to increase the salary of all employees at the Waterloo location by 5%.

Part 1 Submission: For any portions of your solution to this that is SQL, write the SQL in a single file titled `employee.sql` and submit that file to the Lab 2 Dropbox on Learn within two weeks of your scheduled lab. Any written component and/or explanation for your schema should be written in a single file titled `employee.pdf` and likewise be submitted to the Lab 2 Dropbox on Learn within two weeks of your scheduled lab date.

Part 2: In Lab 1 you had to compute several queries on the Sean Lahman baseball database. There were no explicit indexes on that database, though you should have added primary and foreign keys. Using explain on the queries you created for Lab 1, determine if any additional explicit indexes would help in solving those queries.

Likewise, you had to compute several queries on the Yelp database. Again, using explain on the queries you created for Lab 1, determine what indexes would help in solving those queries.

Part 2 Submission: Since your answers to this question are in the form of creating indexes, your submissions should be two `sql` files: `baseball.sql` and `yelp.sql`. Your explanation for why you chose the indexes you chose should be submitted in corresponding files `baseball.pdf` and `yelp.pdf`. All four files should be submitted to the Lab 2 Dropbox on Learn within two weeks of your scheduled lab date.