# Exam Databases 1, 28 mei 2008

| For free | 10 |
|---|---|
| 1.A | 15 |
| 1.B | 10 |
| 2.A | 12 |
| 2.B | 13 |
| 2.C | 15 |
| 3.A | 10 |
| 3.B | 5 |
| 4.A | 5 |
| 4.B | 5 |
| Total | 100 |

## 1. UML class diagrams and the relational scheme

Our administration involves faculty members of a university. The university is divided into departments (each department has a unique name), each member is working for a department. Most members work full time, some of them work part time and some of them may even work for more then one department. Hence, each involvement of a member with a department can be given by a percentage, within each department a member has a certain level determining his/her salary.

The faculty campus consists of a number of buildings, each belonging to a department. The rooms in the building each have a room number, which is unique within that building. A faculty member may have a room within a building that belongs to its department. Members may share a room, their telephone number is unique for the room.

Most faculty members are staff members, before a staff member gets his/her job his/her degrees are checked. A typical degree is contains information about the

level (like "MS" for Master), the study (like "Computer Science"), the year (like "1990") and the university (like "University of Utrecht"). These degrees have to be archived, in case of future promotions. Next to staff members, there are student members who assist the staff members in their tasks. For each student we maintain its study (like "Computer Science", or "mathematics") and its student number as a link with the student administration. In order to assign student members properly to their tasks we have to keep track of the courses they have successfully taken. Each course has a name, and for each course and student member we maintain the grade and the year he/she got this grade.

Finally we have the support members, as final category of faculty members. Each support member has a specialization, like "secretary", "administration", or "IT-helpdesk". For each member we maintain the first name, last name, a SOFI number and a unique internal faculty number. There are some faculty members who fall in neither of the three subcategories, like the department manager. Certain members fall in more subcategories, like student members who work as a support member on the IT-helpdesk.

**A. Give a UML class diagram**
**B. Give the associated relational scheme**

## 2. SQL: the Project Company

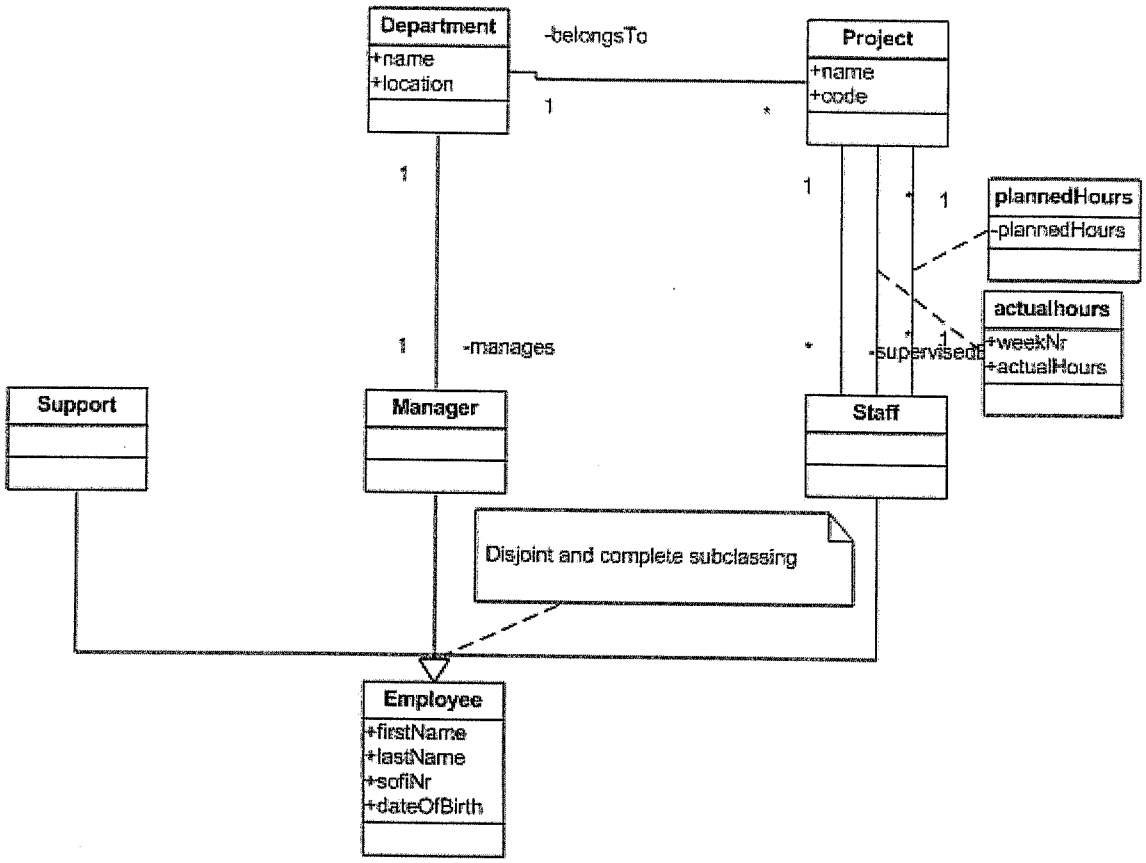
This case is taken from the exercises that are discussed in the class. One can also read the exercises first, before reading the complete case.

A company exists of a number of departments, each department has a name and a location. The company has a number of employees, all part of a department. An employee has a name (first, last), a unique SOFI-number and a date of birth. There are three types of employees: staff, support, and managers (each employee belongs to exactly one of these types). Each department is supervised by a manager.

The company also has a number of projects. Each project has a name, a unique project code and belongs to exactly one department. A project is supervised by an employee from the staff. Furthermore there are a number of staff-members working on the project. If an employee is assigned to a project, he/she is supposed to work a fixed amount of hours per week on that project, this is the

amount of planned hours. During the project for each week the hours an employee actually worked on the project are recorded.

**The UML-diagram of the Project Company**




In this version we assume that the amount of planned hours is fixed for every week, at the end we will give another interpretation.

## The relational scheme

For each table we indicate the PK (Primary Key) and the potential other CK's (Candidate Keys).

Department: (depId, depName, location, managerId)
- PK: depId
- CK: depName
- FK: managerId to Employee (empType must be manager)

Project: (projectId, projectName, projectCode, depId, supervisorId)
- PK: projectId
- CK1: projectName
- CK2: projectCode

FK1: depId to Department
FK2: superVisorId to Employee (empType must be staff)

Employee: (empId, firstName, lastName, sofiNr, dateOfBirth, empType)
PK: empId
CK1: sofiNr
CK2: (firstName, lastName, dateOfBirth)
Note, empType is an enumerated type with three possible values: support, manager or staff

PlannedHours: (projectId, empId, plannedHours)
PK: (projectId, empId)
FK1: projectId to Project
FK2: empId to Employee

ActualHours: (projectId, empId, weekNr, actualHours)
PK(projectId,empId,weekNr)
FK1: projectId to Project
FK2: empId to Employee

After giving the UML class diagram and the relational scheme, we can give the SQL assignments.

A. **Give the names of all the projects that belong to the department that is managed by John Doe.**

B. **Give for each staff member (given by first name and last name) the total amount of hours that he/she has worked on any of the projects in week 24, if this total number is greater than 40.**

C. **Give the names (first name and last name) of the staff members that (in every week) spent more hours on their projects than planned.**

## 3. Concurrency

A. **Explain the four aspects of ACID-transactions**
B. **Explain the three types of locks**

## 4. SQL and Java

A. **Describe briefly some advantages of the new JPA (Java Persistency API) with respect to the "old" JDBC approach.**
B. **Describe briefly the notion of "a managed object" in the context of JPA.**