Assignment 1

\_ Add, delete, or modify an employee, including the data about his/her

address, education, and job position.

\_ Add, modify, delete a project, including its location.

\_ Assign an existing employee to a project.

\_ Check what projects cannot pay the rent, that is, the rent is higher than

10% of their budget.

Hand-in

The assignment must be handed in in natschool using the right submission

folder. The hand-in must include:

\_ The SQL code to generate the database tables.

\_ A report that explains in what normal form each table is and why, and

eventual normalization steps performed to normalize the table up to 3NF

and BCNF.

\_ The source code of the management application.

\_ Optional: the ERD diagram used to model the database.

Assignment 3

Instructions

\_ The assignment must be implemented using Neo4j.

\_ Deliver the cypher queries into a text \_le (not word!).

\_ Deliver also the database folder of Neo4j (your database name.graphdb).

Assignment A

Consider the Entity-Relationship diagram in Figure 1 representing the model

for an airport database.

\_ Give a graph database implementation in Neo4j. Fill in the database with

data satisfying the following constraints:

{ At least 6 airports, two of which must be name 'Schiphol', and

'Venezia Marco Polo', two must be located in 'London', and one

in 'Rome'.

{ The values of size in airports must be 'Small', 'Medium', or 'Large'.

{ At least one airport must be 'Large'.

{ Each airport must have at least 5 terminals.

{ 'Venezia Marco Polo' must have a terminal 'B'.

{ At least 5 companies, two of which must be 'Lufthansa' and 'KLM'.

{ At least 5 ights, one of which must be scheduled before 15:00.

{ At least 3 gates per terminal.

{ The values for state in gates must be either 'Boarding' or 'Closed'.

{ There must be a 'Boarding' gate for terminal 'B' in 'Venezia Marco

Polo'.

\_ Implement the following queries in cypher:

1. Find the name and the capacity of all 'Large' airports.

2. Find the total capacity of the airports in the same city. Output the

name of the city and the total capacity.

3. Find the name of the airport with the highest capacity. Output the

name and the capacity.

4. Find all the opened terminals in 'Schiphol'. Output the code.

5. Find all the terminals of the airports in 'London'. Output the code.

6. Find all the gates that are boarding in 'Venezia Marco Polo' for

terminal 'B'. Print the number and the state.

7. Find all the ights landing in 'Rome' for 'Lufthansa' and 'KLM'.

Print the code and the plane.

8. For each company \_nd the amount of ights going to 'Rome' leaving

before 15:00. Print the company name and the total of ights.

Assignment 2

Task description

Create a document database in MongoDB for the described context. Create

an application (in Java or C#) that inserts random data inside the database.

Strings de\_ning names should be picked randomly inside a speci\_ed lists of

1

names. Furthermore, the numerical data should have likely values, for example

the age of an employee cannot be 1500, or the our fee 100000. Insert at least

10000 employees, and 100 projects. Your application should use a \_xed random

seed which can be set to generate the random data, in order to able to debug

the application.

Implement the following queries using the map-reduce paradigm in Mon-

goDB and output the data on either a GUI or the shell.

1. An employee is underworking if he is working less than 5 hours on his

projects, he is working normally if he is working between 5 and 20 hours,

and he is overworking if he is working more than 20 hours. Output the

number of employees that are overworking per project.

2. Find the total working hours of all employees and average working hours

per employee.

3. Find the total fee of an employee. This number is obtained by multiplying

the working hours on all the projects by the hour fee of his position.

You are allowed to use only the map-reduce functions seen in class.

Hand-in

The assignment must be handed in in natschool using the right submission

folder. The hand-in must include:

\_ The code to generate the database.

\_ The source code of the program to populate the database.