FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
Mestrado Integrado em Engenharia Informática e Computação

Database Technology

Assignment nr. 3

Version 3 – Student applications

NoSQL Assignment

Goals

Build a small DB using a NoSQL approach and compare with the results obtained from the relational approach.

Work groups

The assignment should be executed by a group of one or two elements.

Deadline

To submit in the Moodle by 2016-06-04.

Subject

The database to be used in this assignment is about applications, enrolments and results of students in programs, during a certain number of years. The tables are: the table **lics**, containing the list of programs; the table **cands** with the applications; the table **alus** with the annual enrolments of students; and the table **anos** with the list of years with recorded enrolments. The data is about all the students that were active in 1992 or later and it was frozen in 2002 with the enrolments in 2002/2003 already done.

Before a student can be enrolled in a program in a certain academic year, his application to the program in that same year must be recorded. The student may apply for more than one program. In this phase, he is identified by the personal id (BI), as he will get a student number only in the enrolment process. With respect to each application, the intended program is known, the application outcome, and, sometimes, the application grade (calculated from the high school results). Not all the accepted candidates actually register in the program. Once enrolled, the student gets a status indicating whether he is F(ollowing) the program or he C(oncluded) it.

The database schema is like follows:

- Table **lics (**list of programs)
    - codigo NUMBER(4) NOT NULL primary key,
    - sigla VARCHAR2(5) NOT NULL,
    - nome VARCHAR2(80) NOT NULL
- Table **cands** (all the applications)
    - bi VARCHAR2(12) NOT NULL,
    - curso NUMBER(4) NOT NULL references lics,
    - ano_lectivo NUMBER(4) NOT NULL,
    - resultado VARCHAR2(1), -- C – accepted; E – excluded; S – waiting list
    - media NUMBER(5,2),
    - PRIMARY KEY (bi,curso, ano_lectivo)
- Table **alus** (the annual enrolments). Notice that if a person changes between programs, he will get two different student number, but the same personal id (BI). Academic years are represented by the first year, i. e., 1999 means ‘1999/2000’.
    - numero VARCHAR2(9) primary key,
    - bi VARCHAR2(12) NOT NULL,
    - curso NUMBER(4) references lics,

  - a_lect_matricula NUMBER(4) NOT NULL,
  - estado VARCHAR2(2), -- F – following; C – concluded
  - a_lect_conclusao NUMBER(4),
  - med_final NUMBER(4,2) – final grade
  - FOREIGN KEY (bi, curso, a_lect_matricula) REFERENCES cands
- Table **anos** (list of years with enrolments)
  - ano NUMBER(4).

The tables are in Oracle user GTD2.

TASKS

The following tasks should be performed, described in a report and presented on June, 6th:

1) Design a Mongo document model for the student applications example, explaining the decisions made. The information available under this model should be equivalent to the information in the relational database.
2) Migrate the data from the Oracle user GTD2 into the NoSQL database using one or more times the following method: write a PL/SQL package able to extract the data using an appropriate SQL query, deliver it in XML format, and process the XML result using a dynamic XSL stylesheet to produce in a browser a set of Mongo method calls to populate the NoSQL database designed in 1). Include in the report the SQL query, a sample of the XML result, the XSL stylesheet, and a sample of the method calls.
3) Prepare Mongo queries for the following questions:
   a. Calculate the total number of students enrolled in each program, in each year, after 1991.
   b. Obtain the BI and the student number of the students with a final grade (med_final) higher than the application grade (media).
   c. Find the average of the final grades of all the students finishing their program in a certain number of years, 5 years, 6 years, …
4) Compare the Mongo and the Oracle implementations from the viewpoints of data size, processing time, and query easiness.