Data Base and Data Mining Group of Politecnico di Torino

Data Warehousing

Politecnico di Torino

Versioning services

Problem specifications

The software development process for medium- and large-sized applications is typically performed by exploiting services for code versioning (e.g. SVN, Git). These services enable tracking changes on code during its development. Suppose you want to exploit data from a code versioning website to realize a data warehouse and obtain statistics on developer activities.

Code generated by developers is divided into repositories. Each repository represents a project under development and it is associated to a specific software company. A repository is characterized by the visibility in the website (public or private), a software category (e.g. “app”, “videogame”, “driver”, ...) and one or more programming languages. The website recognizes 15 different programming languages (i.e. ‘Scala’, ‘Python’, ‘Java’, ecc...).

Each repository consists of one or more parallel branches. A branch is defined as a workflow where one or more developers make code updates. Each atomic set of code updates saved onto the site by a developer is called commit. For each branch, the creation date and the repository it belongs to are known. A branch belongs to one repository only.

The system tracks information about commits from developers. Specifically, it stores the developers’ role (e.g. ‘test’, ‘development’, ‘design’, ...), and their work team (e.g. ‘backend development’, ‘UI development’, ...). Each developer belongs to a single work team only. Commits are timestamped.
Suppose you want to analyze statistics about the number of commits made by developers, the number of additions and deletions of code lines.

The analysis must be carried on based on:

- branch name, branch creation date, branch creation year
- repository name, company, visibility, category, programming languages
- developer, their role, their work team
- hour, date, month, 2-months, 4-months, trimester, year, month of the year, 4-months of the year

Design

Design the data warehouse to address the specifications and to efficiently answer to the provided frequent queries. Draw the conceptual schema of the data warehouse and the logical schema (fact and dimension tables).

Query

Write the following frequent queries using the extended SQL language.

a) Consider only private repositories. Separately for month and repository name, analyze: the number of commits made on average in a day, the number of commits made on average in a branch, the monthly cumulative number of commits from the beginning of the year.

b) Consider data related to repositories which include the ‘Scala’ language. Separately for branch and work team, analyze: the ratio between the number of additions and the number of deletions, the percentage of additions with respect to the total of the repository the branch belongs to, assign a rank to work teams based on the ratio between the number of additions and the number of deletions, separately for each branch.