CtrlPrint

CtrlPrint *Coding Task*

Thank you for taking the time to do the CtrlPrint coding task. We expect it to take 2-4 hours to complete. Please make sure you can work with it without major interruptions and send us an email as soon as you have completed the task. We will schedule a short follow up session for you to present your work to us. Good luck!

The task

The task is to implement a REST endpoint in Go that extracts the full version history of a single chapter from the provided database, and returns it as a json document. The endpoint should expect the chapter_id as input parameter. The result should be returned as a json document, containing the following information:

- The name of the chapter
- The name of the project that the chapter belongs to
- The name of the company that the project belongs to
- An array with entries for each version, ordered so that the first version (version 0 - a placeholder version) comes first, and then each subsequent version in order until the latest version. Each version entry should contain the following information:
    - The chapter_version_id
    - The date the chapter version was created
    - The version number of the chapter version
    - The username of the person that created the chapter version
    - The application version that was used to create the chapter version, converted to the marketing name, ie 11.0 should be converted to CC 2015

Example

Query

```
curl http://localhost/chapter_versions/27
```

Result:

```
{
      "company": "Nesbo Farmers",
      "project": "AR 2016",
      "chapter": "02 SV Omslag insida fram-5",
      "versions": [
            {
                  "created_by": "haye-olle",
                  "chapter_version_id": 356692,
                  "version_number": 0,
                  "created": "2016-12-22T08:37:27.922596",
                  "appversion": "CC 2015"
            },
            /*...*/
      ]
}
```

What we will be looking for

There are of course a number of different ways to implement the solution. While there are numerous third party packages that can help in implementing REST-based services and database marshalling, it is also quite straightforward to implement everything without using any external packages as well. It is possible to implement an object model in go, and use that to unmarshall the database objects and then marshall the structure as json, or it is even possible to have the PostgreSQL database return the requested dataset as a json encoded string with one single query. We will be looking at various aspects of your implementation when evaluating it, but the main focus is of course the go code. We value simplicity, readability and security primarily. You don't need to worry about authentication, but please make sure to sanitize input data.

Data model

This section details the data model for the exercise. The data model is based on a subset of the actual data model used in CtrlPrint, but simplified. It is also detailed in the ER-diagram DB, and available as an sql script which creates the necessary tables and populates them with sample data.

language

This table contains the various languages in the system used for the user interface. For this exercise this can also be used to differentiate between Swedish and English users.

person

The table of users in the system. The column **`person_language_id`** is a foreign key mapping to **`language.language_id`**. A person belongs to a company as mapped via the foreign key **`person_company_id`** to **`company.company_id.`**

company

The companies in the system. A company can contain zero, one or more projects, and has zero, one or more persons in it.

project

The projects in the system. A project always belongs to a company, using **`project_company_id`** as foreign key to **`company.company_id`**.

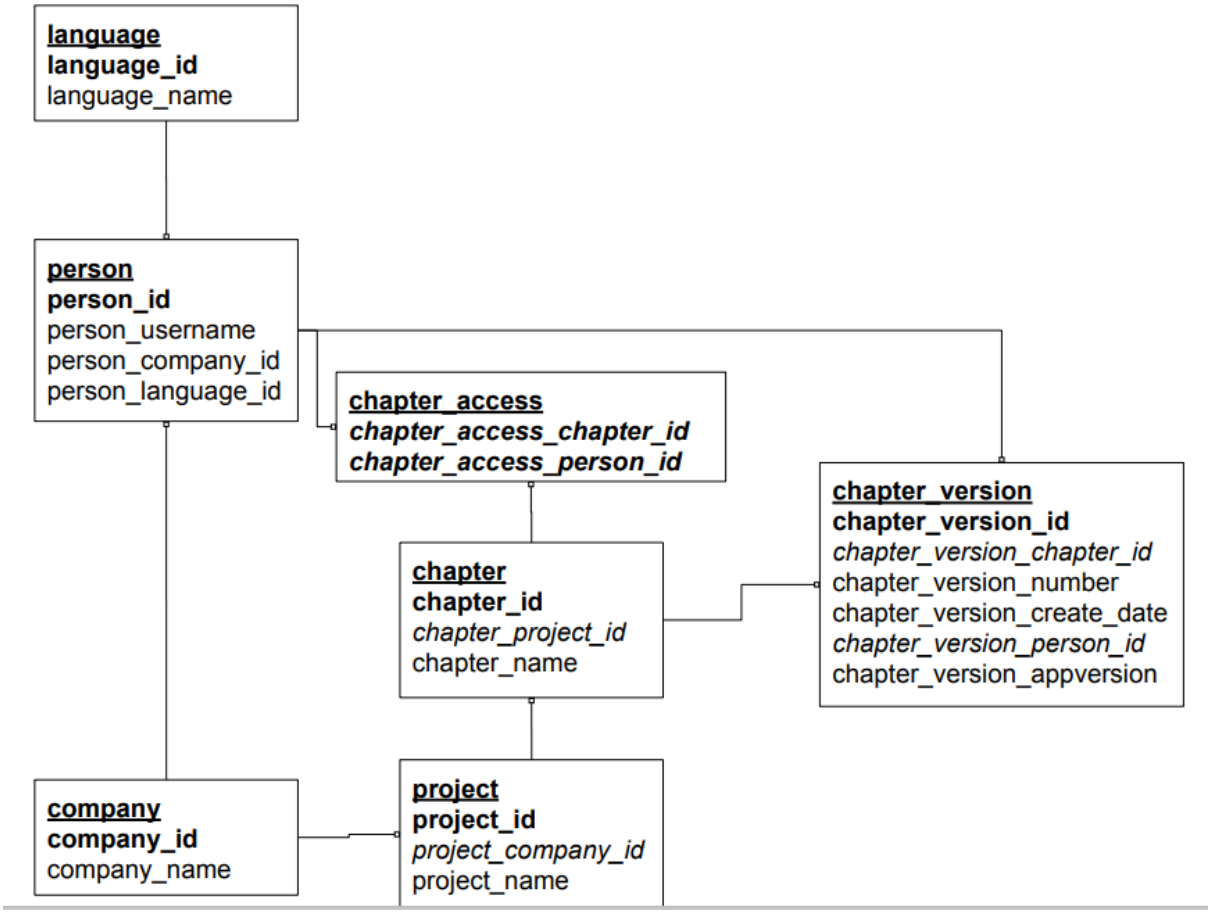
chapter

A chapter refers to a document, and belongs to a project. This relationship is mapped through the foreign key **`chapter_project_id`** to **`project.project_id`**.

chapter_access

The chapter_access table maps a person to a chapter, and thereby grants the person access to the chapter. chapter_version Each chapter has zero, one or more versions. Every time a document is opened and then saved back, a new chapter_version is created, numbering starts at 1 and then increases with 1 for each new version, which is represented in the column **`chapter_version_number`**. The person who created a specific version is tracked in the **`chapter_version_person_id`** column which maps to **`person.person_id`**.

CtrlPrint supports using various versions of Adobe InDesign/InCopy (such as CC 2015, CC 2017 and so on), and this is tracked in the column **`chapter_version_appversion`**. The content contains a version number internal to the application, so CC 2015 for instance is represented as 11.0, and CC 2017 as 12.0


language
language_id
language_name
person
person_id
person_username
person_company_id
person_language_id
chapter_access
chapter_access_chapter_id
chapter_access_person_id
chapter_version
chapter_version_id
chapter_version_chapter_id
chapter_version_number
chapter_version_create_date
chapter_version_person_id
chapter_version_appversion
chapter
chapter_id
chapter_project_id
chapter_name
company
company_id
company_name
project
project_id
project_company_id
project_name