

Repository Schema

Software Engineering Principles



January 23, 2023

Group 15

Table of Contents

[Introduction 2](#_Toc125279330)

[Data Store Design 2](#_Toc125279331)

[Tables 2](#_Toc125279332)

[Data dictionary 2](#_Toc125279333)

## Introduction

A repository schema is the structure and organization of a database or collection of data within a version control system. It defines the tables, fields, and relationships that make up the data and how it is stored and accessed. The schema can include information on data types, constraints, and other metadata that is used to ensure the integrity and consistency of the data. In version control system, it is the blueprint of the data, how it is structured and organized, which is used to maintain version and track changes over time.

## Data Store Design

Data store design is the process of creating a blueprint for organizing and structuring data in a way that is efficient, scalable, and easy to understand. This includes identifying the types of data that will be stored, defining relationships between data entities, selecting an appropriate data model, normalizing the data, optimizing for performance, testing, and implementing the design, and monitoring and maintaining the data store over time. The goal of data store design is to ensure that the data is stored in a way that is optimized for performance and meets the requirements of the application or system that uses the data.

### Tables

The list of tables that has been used for completion of the system are listed below:

|  |  |
| --- | --- |
| **Table Name** | **Description** |
| users | This table stores data for user like name, email, password, is\_superuser, and timestamps. |
| categories | This table stores the categories of different places. The data stored in it is name of the category. |
| places | This table stores the details of different places. The data can be added to this by the back office. |
| reviews | This table stores the reviews of different places by different users. The data can be added to this by any registered user. |
| comments | This table stores the replies made for the reviews. The data can be added to this by any registered user. |

### Data dictionary

A data dictionary is a collection of metadata that describes the structure, organization, and characteristics of data within a database or other data storage system. It is a reference tool that provides information about the data elements, data relationships, and data constraints that make up the data.

**User Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| id | Integer | Stores the user id |
| name | String | Stores the name of a user |
| email | String | Stores an email of a user |
| password | String | Stores an encrypted hashed password of a user |
| is\_superuser | Boolean | Stores the superuser status of the user |

**Category Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| id | Integer | Stores the category id |
| name | String | Stores the name of a category |

**Place Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| id | Integer | Stores the place id |
| name | String | Stores the name of a place |
| image | String | Stores an image location of a place |
| address | String | Stores an address for a place |
| post\_code | String | Stores the post code for a place |
| description | Long Text | Stores the post code for a place |
| category\_id | Foreign Key | Stores the category id to which place belongs |

**Review Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| id | Integer | Stores the review id |
| comment | String | Stores the message |
| rating | Integer | Stores the rating |
| is\_active | Boolean | Stores the status of the review |
| place\_id | Foreign Key | Stores the place id for which review has been added |
| user\_id | Foreign Key | Stores the user id that has added the review |

**Comment Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| id | Integer | Stores the comment id |
| comment | String | Stores the reply for the review |
| from\_id | Foreign Key | Stores the user id of the user that added the reply |
| to\_id | Foreign Key | Stores the user id of the user that added the review |
| review\_id | Foreign Key | Stores the review id to which this reply is linked |