

Mr. Hajiqorbani

Presented to:

by Sepehr Norouzi, Bahar Navab, Asma Hossein Fakhar

DATA DICTIONARY

**SEMUNI B☻OK CLUB**



ارائه شده به:

استاد محسن حاجی‌قربانی

نویسندگان:

سپهر نوروزی

بهارالسادات نواب صفی‌الدین

اسماء حسین فخار

فرهنگ داده‌ی SemUni Book Club

فهرست

[مقدمه 1](#_Toc86176037)

[Book 2](#_Toc86176038)

[User 3](#_Toc86176039)

[Messages & Complaints 4](#_Toc86176040)

[Favorites 6](#_Toc86176041)

[ارتباط userموجودیت‌ها 6](#_Toc86176042)

[ارتباط user با book 6](#_Toc86176043)

[ارتباط user با toReads 6](#_Toc86176044)

[ارتباط user با favorites 7](#_Toc86176045)

[ارتباط user با messages 7](#_Toc86176046)

# مقدمه

پروژه‌ی SemUni Book Club یک وب‌اپلیکیشن برای رویت و جستجوی کتاب‌های معروف، نویسنده‌های آن، امتیاز جهانی آن و ISBN13 آن‌ها می‌‌باشد. همچنین کاربران می‌توانند لیستی از کتاب‌های مورد علاقه و کتاب‌هایی که قصد خواندن آن‌ها را دارند نیز ایجاد کنند و به مطالعه خود نظم ببخشند.

این برنامه تحت وب و بر پایه‌ی فرانت اند VueJs که یک framework جاوااسکریپ می‌باشد و بک اند Django و Django REST framework که بر پایه‌ی زبان python می‌باشند و همچنین Sqlite3 است، که به عنوان پایگاه داده‌ی برنامه از آن استفاده شده است. در این متن ما به توضیح تمامی عناصر داده‌ای مرتبط با برنامه می‌پردازیم. لازم به ذکر است لینک تمامی نمودارهای مربوط به این پروژه به پیوست همین سند آمده است.

# Book

یک موجودیت داده‌ای برای ذخیره‌ی کتاب‌ها در Database می‌باشد. این موجودیت هیچ کلید خارجی ندارد ولی از کلید اصلی آن (Book\_ID) به عنوان کلید خارجی در جداول favorite و ToRead استفاده می‌شود که راجع‌به نوع ارتباط آن‌ها در خود بخش آن جداول مربوط بحث خواهد شد. فیلد‌های Book را در جدول زیر مشاهده می‌کنیم.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Size(B) | PK | FK | DESC |
| Book\_ID | INTEGER | 8 |  |  | کلید اصلی برای مشخص کردن کتاب در جداول دیگر |
| Title | VARCHAR | DYNAMIC |  |  | یک فیلد متنی برای ذخیره‌ی عنوان کتاب در پایگاه داده |
| Author(s) | ARRAY(VARCHAR) | DYNAMIC |  |  | یک آرایه از فیلدهای متنی برای ذخیره‌ی نویسنده یا نویسندگان یک کتاب. |
| ISBN13 | VARCHAR(13) | 13 |  |  | یک مشخصه جهانی برای پیدا کردن یک کتاب که ترکیبی از اعداد و حروف می‌باشد و دقیقاً باید 13 کاراکتر باشد. |
| Rating | DOUBLE | 8 |  |  | یک فیلد عدد صحیح برای امتیاز یک کتاب از 5 |
| ImageURL | URL | 128 |  |  | یک فیلد برای ذخیره‌ی آدرس تصویر یک کتاب که در پایگاه به صورت مستقیم ذخیره نمی‌شود. |

# User

یک موجودیت داده‌ای برای ذخیره‌ی کاربران در database استفاده می‌شود. درست‌تر است که admin هم یک user معرفی شود و ما نیز در این برنامه این کار را کردیم ولی به صورت جداگانه ادمین را در ERD و DFD یک موجودیت معرفی کردیم که از پیچیدگی نمودار کمتر شود. ادمین بودن یک user با فیلد is\_staff در database مشخص می‌شود و در بک اند privilege های مربوط به آن داده می‌باشد.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Size(B) | PK | FK | DESC |
| Id | INTEGER | 8 |  |  | کلید اصلی برای مشخص کردن کاربر در جداول دیگر |
| Username | VARCHAR | DYNAMIC |  |  | یک فیلد متنی برای ذخیره‌ی نام کاربری |
| Password | NVARCHAR | DYNAMIC |  |  | یک فیلد متنی برای ذخیره پسورد که با استفاده از جنگو رمزگزاری می‌شود |
| profile\_image | Image\_field | DYNAMIC |  |  | یک فیلد مخصوص جنگو برای ذخیره عکس در دیتابیس |
| is\_staff | Boolean | 1 |  |  | یک فیلد برای تمایز ادمین و کاربر ساده |
| Email | email\_field | DYNAMIC |  |  | یک فیلد مخصوص جنگو برای ذخیره ایمیل در دیتابیس |

# Messages & Complaints

یک جدول در پروژه که قرار است پیام‌های کاربران را ذخیره کند و طبیعتاً تنها ادمین است که می‌تواند به آن دسترسی داشته باشد. این جدول علاوه‌بر پیام‌ها کاربر مربوط به آن پیام را نیز ذخیره می‌کند تا ادمین بتواند به او پاسخ بدهد. باید توجه داشت که رابطه‌ی پیام و کاربر یک به چند می‌باشد. یعنی User\_ID ها می‌توانند در این جدول تکرار شوند ولی هر message\_ID یک user\_ID خواهد داشت.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Size(B) | PK | FK | DESC |
| Message\_ID | INTEGER | 8 |  |  | کلید اصلی این موجودیت |
| User\_ID | INTEGER | 8 |  |  | کلید خارجی این جدول که مشخص می‌کند کدام کاربر این پیام را ثبت کرده است |
| Title | VARCHAR | DYNAMIC |  |  | یک فیلد متنی است که قرار است موضوع پیام را ذخیره کند |
| Body | VARCHAR | DYNAMIC |  |  | متن اصلی پیام را ذخیره می‌کند |
| Created\_at | DATETIMEFIELD | DYNAMIC |  |  | یک فیلد مخصوص جنگو است که زمان ایجاد این شئ را ذخیره می‌کند |

To reads

اولین موجودیت ضعیف این پروژه به این معنا که از خود Primary Key یا کلید اصلی ندارد و با توجه به user\_ID که Foreign Key یا کلید خارجی آن است بیان می‌شود. با این حال خود Django به خاطر راحتی کار برنامه‌نویس یک فیلد ID به آن اضافه می‌کند که ما آن را بیان نمی‌کنیم. درکل لیستی از کتبی است که کاربر قصد خواندن آن‌ها را دارد و می‌خواهد آن‌ها را ذخیره کند.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Size | PK | FK | DESC |
| User\_ID | INTEGER | 8 |  |  | کلید خارجی این جدول که مشخص می‌کند کدام کاربر این کتاب را ذخیره کرده |
| Book\_ID | INTEGER | 8 |  |  | کلید خارجی این جدول که مشخص می‌کند کدام کتاب توسط کاربر در این لیست ذخیره شده. |
| Created\_at | DATETIMEFIELD | DYNAMIC |  |  | یک فیلد مخصوص جنگو است که زمان ایجاد این شئ را ذخیره می‌کند |

# Favorites

موجودیتی ضعیف است که باز با User\_ID بیان می‌شود و تفاوت آن با لیست بالا این است که کتب موجود در این لیست را کاربر خوانده و مورد پسندش بوده است.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Size | PK | FK | DESC |
| User\_ID | INTEGER | 8 |  |  | کلید خارجی این جدول که مشخص می‌کند کدام کاربر این کتاب را ذخیره کرده |
| Book\_ID | INTEGER | 8 |  |  | کلید خارجی این جدول که مشخص می‌کند کدام کتاب توسط کاربر در این لیست ذخیره شده. |
| Created\_at | DATETIMEFIELD | DYNAMIC |  |  | یک فیلد مخصوص جنگو است که زمان ایجاد این شئ را ذخیره می‌کند |

# ارتباط userموجودیت‌ها

## ارتباط user با book

در این پروژه user می‌تواند کتاب‌ها را ببیند و تعدادی از آن‌ها را در لیست toReads و favorites خود قرار دهد. حال اگر user ادمین باشد، می‌تواند علاوه‌بر این، کتاب اضافه کند، به‌روزرسانی کند و حذف کند.

## ارتباط user با toReads

هر کاربر می‌تواند چند فیلد در جدول toReads را به خود اختصاص دهد که یک book\_Id نیز دارد و نشان می‌دهد کدام کتاب را در این لیست قرار داده است

## ارتباط user با favorites

هر کاربر می‌تواند چند فیلد در جدول favorites را به خود اختصاص دهد که یک book\_Id نیز دارد و نشان می‌دهد کدام کتاب را در این لیست قرار داده است.

## ارتباط user با messages

در این مورد ما باید به دو دسته user هارا تقسیم کنیم. یک کاربران عادی هستند که می‌توانند پیام بفرستند و با user\_id خود در system ثبت بکنند. و دوم کاربران ادمین هستند که می‌توانند این پیام‌ها را مشاهده کنند و به آن‌ها پاسخ دهند.

یک نکته قابل توجه این است که این document تا پیاده‌سازی کامل پروژه همواره تغییر خواهد کرد و ممکن است چیزی به آن اضافه یا از آن کم شود. با این وجود کلیت موضوع پروژه بدون تغییر باقی خواهد ماند.

پایان

\*