



Project 1

Lecture 1, 2

Course Principles of Database Design

Dr. Shahriari

Spring 2023



مقدمه

در درس با یکی از پایگاه داده‌های غیر ساختارمند آشنا شدیم. یکی از این پایگاه داده‌ها [Redis](#) است که از آن عموماً به عنوان یک `cache` `database` استفاده می‌شود. این پایگاه داده به دلیل ذخیره سازی اطلاعات بر روی حافظه اصلی (RAM) ذخیره می‌کند که گذرا است. موارد استفاده از این پایگاه داده به دلیل قابلیت های متعدد آن زیاد است که می‌خواهیم با ساختن یک برنامه چت ساده با برخی از آنها آشنا شویم.

بخش اول - راه اندازی Redis

در این بخش شما باید یک پایگاه داده Redis را راه اندازی کرده و به آن متصل شوید. شما می‌توانید از هر زبان برنامه نویسی دلخواه استفاده کنید (ولی سعی کنید از پایتون و از فریمورک جنگو استفاده کنید). بعد از اتصال، موفقیت آمیز بودن آن را با ارسال `ping` و نمایش پیام مناسب تایید کنید.

نکات:

- استفاده از [Docker](#) و یا [Kubernetes](#) برای راه اندازی Redis نمره اضافی دارد.

بخش دوم - پیاده سازی گروه

شما قابلیت ایجاد تعداد نامحدود پایگاه داده که با اعداد صحیح مشخص می‌شوند را دارید. یک پایگاه داده به دلخواه خودتان ایجاد کرده و به گروه ها اختصاص دهید به طوری که هر کلید (`key`) اسم گروه و مقدار (`value`) آن یک Dictionary از اطلاعات زیر باشد:

- نام سازنده (`creator`): اسم فردی که این کانال را ساخته است.
- تاریخ ساخته شدن (`created_at`): زمان و تاریخی که این کانال ساخته شده است.
- توضیح (`description`): یک یا چند جمله توصیف کننده کانال.
- اعضا (`members`): لیستی از اسامی افرادی که در این کانال عضو هستند.
- نمایش تاریخچه پیام ها (`load_last_n_hour_messages`): تمام پیام های که در `n` ساعت گذشته ارسال شده برای عضو جدید نمایش داده شود (به ترتیب زمانی)

شما با استفاده از قابلیت [Publish/Subscribe](#) می‌بایستی مکانیسمی طراحی کنید که در آن افراد مختلف بتوانند به یک گروه اضافه شده (`subscribe`) کرده و پیام اعضای دیگر را دریافت کنند) و همچنین بتوانند در گروه پیام ارسال کنند (`publish` کرده). توجه کنید که پیام‌های ارسال شده باید شامل اطلاعات زیر به صورت Dictionary بوده و نمایش واضحی از محتوا بسته های دریافت شده توسط هر عضو داشته باشید:

- ارسال کننده (`sender`): اسم فرد ارسال کننده پیام
 - تاریخ ارسال (`sent_at`): زمان و روزی که پیام ارسال شده
 - محتوا (`text`): متن ارسالی
- نمونه نمایش بسته دریافتی:

```
[2023-02-27 21:29:41] Keivan (databaseGP): "Hello Everyone!"
[2023-02-27 21:31:12] Arad (databaseGP): "Howdy!"
```



نکات:

- هنگام ساختن کانال ها، به تکراری بودن اسم آن توجه کنید، زیرا اسم کانال باید یکتا باشد. شما نمی توانید کانالی با اسم تکراری بسازید! (چرا؟!...)
- هر کاربر در باید لیستی از گروه های موجود همراه مشخصات آنها (اسم، تعداد عضو، توضیح، ...) گرفته و بتواند وارد آنها شود. (گروه خصوصی نداریم)
- در صورت اضافه شدن یک کاربر به گروه اسم آن در لیست اعضای آن گروه باید اضافه شود. همچنین هنگام جدا شدن از گروه، اسم آن باید از لیست هم حذف شود.
- یک کاربر می تواند در چندین گروه به صورت همزمان عضو باشد. (بدیهی است نحوه ارسال پیام به هر گروه باید طراحی شود)

بخش سوم- تاریخچه پیام ها

در این بخش می خواهیم پیام های ارسال شده را در یک پایگاه داده جدید در قالب یک تاریخچه پیام ها ذخیره کنیم. از آنجایی که Redis یک موتور key-value است، برای انتخاب کلید ها به این صورت عمل میکنیم:

<group name>-<sender name>-<timestamp>

که در آن groupname اسم گروه، sendername اسم فرستنده و timestamp همان [unix timestamp](#) است. نمونه این کلید در عمل مشابه زیر است:

databaseGP-Bardia-2022-02-27-21:29:41

برای دریافت پیام ها باید کلیدهای مرتبط را با کمک کوئری ای بهینه دریافت کنید تا Redis فضای حالت کوچکتری را جست و جو کند. با بدست آوردن کلید های مرتبط می توانید مقادیر پیام ها را دریافت کرده و و نمایش دهید، اما می خواهیم با استفاده از قابلیت [Pipelining](#) این کار را سریع تر انجام دهیم.



به نکات زیر توجه کنید.

- این پروژه باید به صورت انفرادی انجام شده و کد های مشابه چک خواهند شد.
- استفاده از Redis برای این پروژه اجباری است.
- این پروژه تحویل آنلاین داشته و کارکرد برنامه شما به صورت عملی تست خواهد شد.
- استفاده از معماری مناسب و OOP برای پیاده سازی پروژه نمره امتیازی دارد.
- پروژه را در قالب یک فایل زیپ به نام db-proj۰-studentnumber.zip بارگزاری کنید.
- در صورت داشتن هرگونه ابهام و سوال از تدریساران درس کمک بگیرید.