به نام خدا

کوروش مسلمی – ۹۸۱۳۰۲۷

کلاس های نوشته شده عبارتند از:

- Query Loader

وظیفه این کلاس خواندن فایل commands.sql می باشد. در این فایل قالب کلی کوئری هایی که برای ساخت جداول، تریگر ها، ویو ها و ... در برنامه استفاده می شود، موجود است. لازم به ذکر است کوئری های فایل مذکور hard code شده نیستند و با استفاده از آرگومان هایی که به متد load فرستاده می شود می توان کوئری متفاوتی تولید کرد. همچنین برای افزایش کارآیی این کلاس از پترن singleton استفاده می کند تا تنها یک نمونه از این کلاس ساخته شود و هر بار خواندن از فایل تکرار نشود.

- Table / Version Table

این دو کلاس از کلاس BaseTable ارث می برند که وظیفه نگهداری اطلاعاتی از ساختار یک جدول مانند اسم جدول، کلید های داخلی و خارجی، ستون ها و تعداد رکورد های موجود در جدول را دارد. هر کلاس متد generateQuery مخصوص به خود را دارد. در کلاس Table این متد تنها کوئری مخصوص ساخت یک جدول را تولید می کند ولی در کلاس Version Table این متد به مراتب پیچیده تر است و همراه کوئری هر جدول، کوئری تریگر و ویو هایی را می سازد که برای پیاده سازی ویژگی سفر در زمان مورد نیاز هستند. جزئیات این پیاده سازی در فایل مربوطه شرح داد شده است.

- Database

اولین وظیفه این کلاس اتصال به یک دیتابیس Postgres می باشد. همچنین متد هایی برای CRUD در دیتابیس وجود دارد که بخشی از آن ها در کلاس Pipeline نیز مورد استفاده قرار می گیرند.

در صورتی که کاربر قصد استفاده از ویژگی سفر در زمان را دارد باید در دیتابیس مقصدی که به آن متصل می شود ویژگی versioning را فعال کرده باشد تا با آن مانند یک انبار داده رفتار شود.

این کلاس دو متد کلیدی با نام های extract و construct دارد. متد extract وظیفه استخراج شمای دیتابیس مبدا و محاسبه topological order جداول آن را دارد.

متد construct وظیفه ساخت دیتابیس مقصد و درون ریزی اطلاعات جداول آن را دارد. اگر کاربر قصد استفاده از سفر در زمان در دیتابیس مقصد را داشته باشد این متد از extract روی دیتابیس مبدا استفاده می کند و جداول آن را از نوع VersionTable دریافت می کند در غیر این صورت جداول از نوع Table استخراج می شوند.

- Pipeline

این کلاس وظیفه ETL را انجام می دهد و در دو حالت قابل اجرا است. در حالت اول دیتابیس مقصدی که به آن وصل می شویم حاوی هیچ اطلاعاتی نیست و باید ویژگی constructive این کلاس را فعال کنیم تا پس از اتصال با استفاده از متد construct دیتابیس مقصد، ساخت آن انجام شود. در حالت دیگر دیتابیس مقصد قبلا ساخته شده و این کلاس، دیتابیس مقصد را به گونه ای تغییر می دهد که با دیتابیس مبدا همگام باشد. در فرآیند همگام سازی عملیات های درج در جهت topological order و عملیات های حذف و آپدیت در خلاف جهت آن انجام می شود.

کتابخانه های مورد استفاده نیز در requirements.txt قابل مشاهده هستند.