

## توضیحات فایل ETL:

برای پیاده سازی پروژه از زبان پایتون (PYTHON) استفاده شده است و لازم است در ابتدا کتابخانه PSYCOPG2 را نصب کنید. در تمام پروژه از کلمه PG به جای نام طولانی این کتابخانه استفاده کرده ام.

در پوشه ETL سه فایل پایتون وجود دارد. یک فایل با نام create\_library.py برای ساخت دیتابیس LIBRARY (همان دیتابیس اصلی) و یک فایل با نام create\_warehouse.py برای ساخت دیتابیس WAREHOUSE (همان دیتابیس مقصد) استفاده شده است. هر دو این فایلها بر اساس SCHEMA و DATA MODEL طراحی شده اند (آموزش این قسمت از یک کد مشابه در گیت هاب گرفته شده است). یک فایل ETL.PY وجود دارد که منطق فاز دوم برنامه را اجرا کرده ام.

ابتدا قبل توضیح کارکرد ETL، یک توضیح مختصری راجع به ساختار DELETE و UPDATE دیتابیس که طراحی کردم بدهم. به ازای هر جدول در دیتابیس دو جدول آپدیت و حذف برای هر کدام طراحی کرده ام. به این صورت که اگر در دیتابیس اصلی داده ای از یک جدول حذف یا آپدیت شود به ترتیب به جدول DELETE یا UPDATE مربوط به آن انتقال می یابد.

مثلا جدول X جدول DELETE\_X و جدول UPDATED\_X. برای راحتی نیز در ادامه از همین نام ها استفاده میکنم. پس هر جا X دیدیم صرفا یک جدول دلخواه است و طبعاً DELETE\_X و UPDATED\_X جداول همراه آن می باشند.

برای ساختار ETL که در ادامه توضیح می دهم من نیاز به متغیرهایی با نوع زمان داشتم که بتوانم بر اساس زمان تصمیم گیری کنم. این متغیرها شامل CREATE, UPDATE, DELETE, و TRANSFER می باشد. اما بعد از اجرا کردن کد ETL.PY، تمام 3 حالت ممکن را در کد به صورت شرطی پیاده سازی کردم.

### حالت اول - DELETE:

اگر داده ای در X حذف شود، تمام رکوردهای داخل DELETE\_X از دیتابیس اصلی که در قبل در آن بودند به دیتابیس WAREHOUSE انتقال پیدا می کنند (بر اساس DELETE) و مقدار TRANSFER آن ها ثبت میشود. بر اساس پرایمری کی (PRIMARY KEY) در WAREHOUSE داده ها در X نیز حذف میکنیم. بعد از این که تمام تغییرات انجام شد تمام داده ها داخل DELETE\_X دیتابیس اصلی را پاک می کنیم.

### حالت دوم - UPDATE:

بروزرسانی یا همان آپدیت کمی پیچیده تر می باشد. گام اول مشابه قبلی است یعنی اگر داده ای در X آپدیت شود، تمام رکوردهای داخل UPDATE\_X از دیتابیس اصلی که در قبل در آن بودند به دیتابیس WAREHOUSE انتقال پیدا می کنند (بر اساس UPDATE) و مقدار TRANSFER آن ها ثبت میشود. بر اساس پرایمری کی (PRIMARY KEY) داده ها، آخرین نسخه از هر داده دیتابیس اصلی را میگیریم و به WAREHOUSE انتقال میدهیم. بعد از این که تمام تغییرات انجام شد تمام داده ها داخل UPDATE\_X دیتابیس اصلی را پاک می کنیم.

### حالت سوم - INSERT:

اگر داده ای در X اضافه شود، تمام رکوردهای داخل INSERT\_X از دیتابیس اصلی که در قبل در آن بودند به جدول آن در دیتابیس WAREHOUSE انتقال پیدا می کنند (براساس CREATE) و مقدار TRANSFER آن ها ثبت می شود.

بعد از هندل کردن تمام حالت های ذکر شده، هیچ داده ای از بین نرفته است و در دیتابیس اصلی نیز UPDATE و DELETE خالی می باشد.

### **توضیح راجع به وصل شدن دیتابیس در ETL.PY:**

در چند خط اول با استفاده از دستورات موجود دیتابیس اصلی و دیتابیس WAREHOUSE را به فایل اتصال دادیم و سپس با دستور SQL ای که نوشتم داده های مورد نیاز را گرفتم. در آخر کد نیز اتصال را از دیتابیس متوقف کردم.

### **توضیح تابع REFACTOR:**

یکی از توابعی که استفاده کردم تابع REFACTOR است که برای تبدیل فایل دیتابیس به مورد استفاده در پایتون، لازم بود تمام مولفه های با مقدار NONE را با استرینگ خالی جا به جا می کند و هرچه داده با تایپ DATE را به استرینگ آن تبدیل می کند.

اشکان ودادی گرگری - 9713032