

1-کد برنامه ای را بنویسید که یک task به صورت دوره ای هر روز ساعت 14 اجرا شود

2-کد برنامه ای را بنویسید که یک task به صورت دوره ای و هر 20 دقیقه اجرا شود.

درون فایل schedule\_tasks.py قرار دارد که برای انجام هر یک از تسک ها از کتابخانه schedule و thread استفاده شده و برای اجرای تسک های برنامه ریزی شده از یک thread دیگر استفاده میشود.

3-دیتا مدل موارد زیر را طراحی کنید.

1-محصولات

2-دیدگاهها

3-سفارش خرید مشتری

4-کاربران

5-notification

مدل های مورد نظر درون فایل models در دایرکتوری task موجود میباشد.

4-گامهای مهاجرت از mysql به mongodb را بنویسید.

پیشنهاد من ایجاد یک فرآیند ساختار یافته سه مرحله ای برای انتقال داده ها از MySQL به MongoDB با استفاده از ابزارهای RabbitMQ، Change Data Capture (CDC) برای بافر و ابزارهای ETL میباشد :

مرحله 1: تغییرات MySQL را با ابزار CDC داشته باشیم

در این مرحله، ابزارهای CDC (Change Data Capture) را برای ثبت تغییرات لحظه ای از MySQL تنظیم می کنیم.

ابزارهای CDC مانند Maxwell، Debezium یا Canal را انتخاب میکنیم که می تواند تغییرات (درج، به روز رسانی، حذف) را در زمان واقعی از پایگاه داده MySQL شما نظارت و ضبط کند.

Configure CDC Tool: ابزار CDC را برای اتصال به پایگاه داده MySQL و گرفتن جداول و رویدادهای مورد نظر پیکربندی می کنیم. اطمینان حاصل می کنیم که رویدادهای تغییر را در قالبی ایجاد می می کنیم که به راحتی توسط RabbitMQ قابل استفاده باشد.

مرحله 2: تغییرات بافر با RabbitMQ

در این مرحله از RabbitMQ برای بافر کردن تغییرات دریافتی از ابزار CDC قبل از انتقال به MongoDB استفاده می کنیم.

RabbitMQ را نصب و راه اندازی می کنیم: RabbitMQ را نصب کرده و صف ها و تبادلات لازم را راه اندازی می کنیم. اطمینان حاصل می کنیم که RabbitMQ قادر به مدیریت حجم رویدادهای تغییر ایجاد شده توسط ابزار CDC است.

یکپارچه سازی CDC Tool to RabbitMQ: یک پل یا رابط را پیاده سازی می کنیم که می تواند رویدادهای تغییر تولید شده توسط ابزار CDC را بگیرد و آنها را در RabbitMQ منتشر کند. این تضمین می کند که تغییرات بافر می شوند و می توانند با نرخ کنترل شده پردازش شوند.

مرحله 3: انتقال داده ها به MongoDB با ابزار ETL

در این مرحله از ابزارهای ETL برای انتقال داده ها از RabbitMQ به MongoDB استفاده می کنیم که یک فرآیند انتقال داده ساختار یافته و کنترل شده را ارائه می دهد.

یک ابزار ETL را انتخاب می کنیم: یک ابزار ETL مانند Talend، Apache Nifi یا Apache Camel را انتخاب می کنیم که می تواند پیام های RabbitMQ را مصرف کند و داده ها را در MongoDB بارگذاری کند. این ابزارها اغلب اتصالاتی را برای RabbitMQ و MongoDB فراهم می کنند.

پیکربندی گردش کار ETL: یک گردش کار ETL طراحی می کنیم که پیام ها را از RabbitMQ می خواند، در صورت لزوم آنها را پردازش می کند (به عنوان مثال، تبدیل داده ها)، و داده ها را در MongoDB وارد یا به روز می کند. برای اطمینان از همگام سازی منظم داده ها، برنامه ریزی را تنظیم می کنیم.

گزارش و بررسی اجمالی راه حل:

هدف از این راه حل انتقال داده ها از MySQL به MongoDB در عین حصول اطمینان از sync بودن لحظه ای و یکپارچگی داده ها است. این از طریق یک فرآیند سه مرحله ای شامل ابزارهای RabbitMQ، CDC برای بافر و ابزارهای ETL به دست می آید.

نقاط قوت:

همگام سازی لحظه ای: تغییرات در MySQL به محض وقوع، ضبط و به MongoDB منتقل می شوند.

Sync بودن داده ها: بافر کردن با RabbitMQ به مدیریت تغییرات با حجم بالا کمک می کند و اطمینان می دهد که داده ها از دست نمی روند.

انعطاف پذیری: ابزارهای ETL انعطاف پذیری را برای تبدیل و سفارشی سازی داده ها ارائه می دهند.

نقاط ضعف و هزینه :

هزینه های زیرساخت برای میزبانی سرورهای RabbitMQ و ETL.

هزینه های توسعه و نگهداری برای اتصالات سفارشی و گردش کار ETL.

هزینه های احتمالی مجوز برای ابزارهای تجاری ETL.

نتیجه :

راه حلی که پیشنهاد شده صرفا با یک سرچ کوتاه و با توجه به عدم آگاهی نسبت ارزش داده ها و مشکلات پیش رو بیان شده و برای پیاده سازی باید سرچ گسترده تر و عمیق تری برای انتخاب ابزارها و پیچیدگی های آن انجام شود و همچنین روشهای دیگر نیز بررسی شود و مزایا و معایب آن ها با این روش مقایسه شود.

## 5-یک api با متد post بنویسید برای ذخیره یکی از دیتا مدل های بالا

درون فایل view در دایرکتوری task یک api مربوط به ایجاد محصولات موجود می باشد.

## 6-روش پیشنهادی برای ایجاد branch چیست؟

گیت فلو پیشنهادی دو branch اصلی دارد که عبارت اند از :

Develop branch-1

Main branch-2

و سه نوع branch دیگر وجود دارد:

**Release branch-3**

**Hotfix-4**

**Feature branch -5**

**Feature branch :**

این branch از Branch Develop ایجاد میشود.

**Develop branch:**

پس از اتمام کد، توسعه دهنده branch خود را با branch Develop مرج می کند.

**Main Branch and Release Branch:**

پس از اینکه یک ویژگی انجام شد. یک Branch Release از Branch Develop ایجاد میشود و آن را با Branch

Develop و Main ادغام می کند،

**Hotfix Branch:**

پس از انتشار یک نسخه، اگر اشکالی در یک نسخه یافت شود، یک Branch از Branch Release ایجاد می شود

و پس از رفع اشکال، Branch Hotfix به Branch Develop و Branch Main ادغام می شود.