به نام خدا

رایانش ابری

تمرین سوم

آشنایی با Spark (فاز دو)

طراح تمرین:

محمد رحمانیان

استاد درس:

دکتر جوادی

مهلت نهایی ارسال پاسخ:

30 آذر ساعت ۲۳:۵۹

مقدمه:

در بخش دوم شما با **Spark** آشنا خواهید شد.Apache Spark یک فریمورک نرم‌افزاری متن‌باز و چند منظوره است که قابلیت پردازش داده‌های بزرگ را در سرعت بالا با استفاده از محاسبات موازی فراهم می‌کند. Spark نسبت به Hadoop MapReduce، که از مدل MapReduce برای پردازش داده‌ها استفاده می‌کند و داده‌ها را در دیسک ذخیره می‌کند، عملکردی تا 100 برابر سریع‌تر ارائه می‌دهد، زیرا داده‌ها را در حافظه پردازش می‌کند که چرخه‌های پردازش داده را به‌طور قابل‌توجهی سرعت می‌بخشد. اسپارک از مدل RDD استفاده می‌کند که امکان پردازش موازی و یکپارچه‌سازی را فراهم می‌آورد تا عملکرد بهینه‌ای داشته باشد. این پلتفرم به دلیل عملکرد بالا، قابلیت پردازش در زمان واقعی، پشتیبانی از مجموعه‌ای از الگوریتم‌ها و کتابخانه‌های مختلف (برای یادگیری ماشین و تحلیل داده) و امکانات موازی‌سازی، بسیار محبوب است و می‌تواند بر روی یک کلاستر یا یک محیط ابری اجرا شود.

شرح تمرین:

این تکلیف طراحی شده است تا شما را با جنبه‌های عملی کار با **Apache Spark** آشنا کند و بر عملکرد برتر آن نسبت به فناوری‌های کلان داده سنتی مانند Hadoop، به ویژه از نظر سرعت به دلیل پردازش داده‌های درون حافظه، تأکید دارد. در طول این تمرین، شما درگیر وظایف مختلفی خواهید بود که در بخش‌های زیر هستند:

- راه اندازی و کانفیگ spark session
- Load و تغییر داده ها با استفاده از Spark DataFrames
- استفاده از user-defined functions برای گسترش قابلیت های اسپارک

هدف این تمرین، ارائه درکی جامع از قابلیت‌ها و ابزارهای Apache Spark است تا شما را برای پردازش کارآمد داده‌های حجیم و انجام تحلیل‌های پیشرفته آماده کند.

Apache Spark یکی از ابزارهای قدرتمند در حوزه پردازش داده‌های کلان است. در این تمرین، تمرکز شما بر استفاده از **Spark DataFrame API** خواهد بود تا توانایی‌های این ابزار را در تحلیل و پردازش داده‌ها بیاموزید. دقت داشته باشید که استفاده از **Spark SQL** در این تمرین مجاز نیست.

برای کار با **DataFrame** در Spark، باید از زبان‌هایی که این API را پشتیبانی می‌کنند، مانند **Java**، **Python** یا **Scala** استفاده کنید. با این حال، **Python** به دلیل سادگی و محبوبیت، به عنوان گزینه پیشنهادی توصیه می‌شود.

بخش اول: معرفی دیتاست:

دیتاست مورد نظر شامل اطلاعات کاملی از چارت‌های "Top 200" و "Viral 50" است که توسط اسپاتیفای در سطح جهانی منتشر می‌شود. اسپاتیفای هر 2 تا 3 روز یکبار چارت جدیدی منتشر می‌کند و این دیتاست شامل تمام چارت‌ها از تاریخ 1 ژانویه 2017 تا کنون می‌باشد.

ستون‌های دیتاست:

1. **title:** str : نام آهنگ.
2. **rank:** int: رتبه آهنگ در چارت (از 1 تا 200).
3. **date:** date: تاریخ انتشار چارت.
4. **artist**:str: نام هنرمند(ها)ی آهنگ.
5. **url**:str: لینک آهنگ در اسپاتیفای.
6. **region**:str: منطقه یا کشور مربوط به چارت.
7. **category**:str: آیا آهنگ در چارت "Top 200" یا "Viral 50" قرار دارد.
8. **trend**:str: روند آهنگ (مثلاً "Same Position"، "Move Down"، "Move Up").
9. **streams**:int: تعداد استریم‌های آهنگ.

برای مشاهده اطلاعات بیشتر در مورد دیتاست میتوانید به این لینک مراجعه کنید.

بخش دوم: راه‌اندازی Spark Session

در این بخش لازم است که یک اپلیکیشن اسپارک بسازید و **master** آن را بر روی **localhost** تنظیم کنید. **SparkSession** به عنوان نقطه شروع برنامه اسپارک شما عمل می‌کند.

امتیازی (استفاده از کلاستر اسپارک):

استفاده از کلاستر اسپارک به جای **localhost** در بخش master، نمره امتیازی دارد. این کار به شما این امکان را می‌دهد که از قدرت پردازشی بیشتر استفاده کنید و سرعت پردازش را افزایش دهید.

بخش سوم: خواندن داده

فایل دیتاست شما به صورت **Parquet** ارائه شده است.در این مرحله، شما باید ابتدا فایل Parquet را با استفاده از Spark DataFrame API بخوانید. این فایل شامل تمام اطلاعات مربوط به چارت‌های اسپاتیفای است.

برای آشنایی بیشتر با فرمت Parquet و ویژگی‌های آن، می‌توانید به لینک مربوطه مراجعه کنید.

بخش چهارم: پردازش داده و انجام تبدیل‌های مختلف

مرحله اول:

سپس باید دیتاست اصلی را بر اساس چارت **"Top 200" فیلتر** کنید. به این معنا که تنها رکوردهایی که مربوط به چارت "Top 200" هستند و همچنین **سال و ماه مشخص شده** در **کانفیگ ورودی** برنامه را شامل می‌شوند، باقی بمانند. با این کار، فقط داده‌های مربوط به چارت و زمان مشخص شده در دیتاست قرار می‌گیرند.

مرحله دوم:

هدف این مرحله این است که با استفاده از دیتا فریمی که در مرحله قبل بدست آوردیم برای هر منطقه و تاریخ، هنرمندی را پیدا کنیم که **بیشترین میانگین استریم** را داشته باشد.

توجه داشته باشید که **همکاری‌ها** به عنوان یک هنرمند واحد در نظر گرفته می‌شوند و نیازی به جداسازی آن‌ها نیست.

مرحله سوم:

در این مرحله، برای هر هنرمند در هر منطقه و تاریخ که بیشترین میانگین استریم را داشته باشد، باید **کمترین رتبه** (بهترین موقعیت) او را پیدا کنید. منظور از رتبه حداقل، بهترین موقعیتی است که هنرمند در آن تاریخ در چارت "Top 200" کسب کرده است. این رتبه می‌تواند به شما کمک کند که بدانید برای کدام موسیقی آن هنرمند بهترین جایگاه خود را در چارت به دست آورده است.

مرحله چهارم:

سپس در این مرحله، باید اطلاعات بیشتری از هنرمندانی که در هر منطقه و تاریخ بیشترین میانگین استریم را داشته‌اند، نمایش دهید. این اطلاعات باید شامل **نام آهنگی که کمترین رتبه** را داشته، **نام هنرمند**، **رتبه (رنک) آهنگ**، **تاریخ** و **منطقه** باشد. علاوه بر این، باید **track_id** را برای هر آهنگ از URL آن استخراج کنید. این کار با استفاده از یک **User-Defined Function** انجام می‌شود که از URL آهنگ، **track_id** مربوطه را استخراج کرده و به عنوان یک ستون جدید به دیتافریم اضافه می‌کند.

برای آشنایی بیشتر با udf به این لینک مراجعه کنید.

مرحله پنجم:

در این مرحله، دیتافریم نهایی که شامل تمام اطلاعات پردازش شده است، باید در یک پایگاه داده **MySQL** ذخیره شود.این مرحله به شما کمک می‌کند که نتایج پردازش‌شده را در یک پایگاه داده مرتب و ذخیره کنید تا امکان دسترسی و استفاده از آن در آینده فراهم باشد.

بخش پنجم: ساخت و اجرای اپلیکیشن با Docker

در این بخش، هدف این است که اپلیکیشن پردازش داده‌های خود را با استفاده از **Docker** بسازید. به این منظور می‌توانید از **داکرفایل نمونه موجود** در تمرین برای بیلد اپلیکیشن استفاده کنید.

اپلیکیشن شما باید بتواند از ماه اول سال 2017 تا ماه آخر سال 2021 **به طور خودکار** اجرا شود. این به معنای **تنظیم زمان‌بندی** اجرای اپلیکیشن به‌طوری که در هر ماه از این بازه زمانی به‌طور خودکار اجرا شود و پردازش‌های مورد نظر را انجام دهد.

برای این منظور، باید از **قابلیت‌های cronjob** برای ساخت، تنظیم و اجرای اپلیکیشن به طور خودکار استفاده کنید. به این صورت که در cronjob شما اجرای اسکریپت به ازای ساعت های مختلف و ورودی ماه و سال های مختلف انجام شود.

برای مدیریت و اجرای همزمان **اپلیکیشن Spark** و **پایگاه داده MySQL** از **Docker Compose** استفاده کنید.

گزارش:

به سوالات زیر در گزارش مربوط به این فاز پاسخ دهید:

- تفاوت های معماری و عملکردی بین RDDs و DataFrames در اسپارک را توضیح دهید.
- مفهوم partitioning داده را در اسپارک توضیح دهید. چرا partitioning در پردازش داده توزیع شده اهمیت دارد.

نکات مربوط به تمرین تحویلی:

- تمرین شما تحویل اسکایپی خواهد داشت؛ بنابراین از استفاده از کدهای یکدیگر یا کدهای موجود در وب که قادر به توضیح داده عملکرد آنها نیستید، بپرهیزید.
- در صورت داشتن هرگونه مشکل، سوالی یا ابهام، آن را در با تدریس یاران درس مطرح کنید تا آن‌ها در سریع‌ترین زمان ممکن به شما پاسخ دهند.

مواردی که باید ارسال شود:

- یک فایل زیپ با نام studentID_HW3_2.zip که شامل گزارش شما به همراه کد شما است.

موفق باشید

تیم تدریس‌یاری مبانی رایانش ابری