

## بسمه تعالی

۱- از چه فرمتی برای مدیریت داده ها استفاده می کنید. دلیل خود را بیان و تحلیل کنید. (برای

دلایل خود نیازمندی های Nonfunctional را هم در نظر بگیرید.)

با توجه به نیازمندی های پروژه، "NoSQL Data Stores" بهترین گزینه برای مدیریت داده ها در این پروژه به نظر می رسد.

دلایل انتخاب NoSQL Data Stores :

1. کارایی: (Performance)

- جستجوی سریع: NoSQL (به ویژه پایگاه های داده اسناد مانند MongoDB) برای جستجوی سریع در داده های غیرساخت یافته و نیمه ساخت یافته بهینه است. این موضوع پاسخگویی زیر 1 ثانیه را برای جستجوی کتاب ها تضمین می کند.
- مقیاس پذیری افقی: با افزایش تعداد کاربران یا ناشران، می توان به سادگی سرورها را اضافه کرد تا بار سیستم بین آن ها تقسیم شود.
- کش داخلی: برخی از NoSQL ها قابلیت کش داخلی دارند که زمان پاسخ دهی را کاهش می دهد.

2. دسترسی پذیری: (Accessibility)

- ساختار منعطف NoSQL برای ذخیره داده های متنوع مانند اطلاعات کتاب، کاربران، ناشران، و بازخوردها مناسب است. داده های مرتبط با کاربران و کتاب ها را می توان به صورت مستقیم در قالب JSON یا مشابه ذخیره کرد و در API های موبایل یا وب به کار برد.

### 3. امنیت: (Security)

- پایگاه‌های داده NoSQL امکاناتی برای رمزنگاری داده‌ها در سطح پایگاه داده و انتقال دارند. همچنین، امکان مدیریت دسترسی نقش‌محور (Role-Based Access Control) برای محدودسازی دسترسی به داده‌ها وجود دارد.
- اکثر پایگاه‌های داده NoSQL مانند MongoDB از احراز هویت دو مرحله‌ای، رمزنگاری SSL، و سازگاری با GDPR پشتیبانی می‌کنند.

### 4. مقیاس پذیری: (Scalability)

- NoSQL به راحتی می‌تواند از حجم بالای داده‌ها (مانند بیش از یک میلیون کاربر یا ده هزار ناشر) پشتیبانی کند. توزیع داده‌ها بین سرورهای مختلف و افزودن سرورهای جدید در آن ساده است.

### 5. پایداری و قابلیت اطمینان: (Reliability)

- بسیاری از سیستم‌های NoSQL مانند Cassandra یا MongoDB از معماری توزیع شده استفاده می‌کنند و از دسترسی مداوم (High Availability) و تحمل خطا (Fault Tolerance) پشتیبانی می‌کنند. این ویژگی می‌تواند نیاز به 99% Uptime را تضمین کند.

### 6. مدیریت داده‌های پیچیده:

- داده‌های مرتبط با کاربران، تاریخچه خرید، و رفتارهای مطالعه می‌توانند به صورت ساختاریافته در پایگاه‌های NoSQL ذخیره شوند، که امکان پردازش داده‌ها برای الگوریتم‌های یادگیری ماشینی را تسهیل می‌کند.

## ۲. پایگاه داده‌ی رابطه‌ای (Relational Databases)

- دلایل انتخاب:
- ساختار منظم: پایگاه داده‌های رابطه‌ای مانند PostgreSQL یا MySQL برای داده‌هایی با ساختار ثابت و منظم، مانند اطلاعات کاربران، ناشران، کتاب‌ها، و تراکنش‌های مالی، مناسب هستند.

- **پشتیبانی از تراکنش ها: (ACID)** برای عملیات پرداخت و مدیریت تراکنش ها، نیاز به پایگاه داده ای دارید که از ویژگی های **ACID** (اتمیگ بودن، سازگاری، ایزوله بودن، و دوام) پشتیبانی کند.
- **امنیت:** پایگاه داده های رابطه ای امکانات پیشرفته ای برای رمزنگاری داده ها، مدیریت نقش ها-Role (Based Access Control)، و جلوگیری از حملات رایج مانند **SQL Injection** ارائه می دهند.
- **پشتیبانی از زبان فارسی:** این پایگاه داده ها به خوبی از کاراکترهای **Unicode** پشتیبانی می کنند، که برای پشتیبانی از زبان فارسی و تاریخ هجری شمسی ضروری است.
- **تحلیل نیازمندی های غیرعملکردی:**
- **کارایی: (Performance)** با استفاده از **ایندکس گذاری (Indexing)** و بهینه سازی کوئری ها، می توان زمان پاسخ دهی برای جستجو و نمایش اطلاعات کتاب ها را به کمتر از ۱ ثانیه کاهش داد.
- **دسترسی پذیری: (Accessibility)** این پایگاه داده ها با ابزارهای مدیریتی مدرن و رابط های کاربری ساده، امکان دسترسی آسان برای ناشران و ادمین ها را فراهم می کنند.
- **مقیاس پذیری: (Scalability)** با استفاده از تکنیک هایی مانند **Sharding** و **Replication**، می توان پایگاه داده ای رابطه ای را برای پشتیبانی از تعداد زیادی کاربر و ناشر مقیاس پذیر کرد.
- **امنیت: (Security)** این پایگاه داده ها از رمزنگاری داده های حساس و کنترل دسترسی پیشرفته پشتیبانی می کنند.

## مقایسه NoSQL با سایر گزینه ها:

### 1. Sequential and Random Access Files:

- ناکارآمد برای جستجوهای پیچیده و داده های مرتبط.
- عدم پشتیبانی مناسب از مقیاس پذیری و قابلیت اطمینان.

## 2. Relational Databases:

- مناسب برای داده‌های ساخت‌یافته، اما برای مقیاس‌پذیری افقی و داده‌های متنوع کمتر مناسب است.
- در مقابل NoSQL، ساختار سخت‌گیرانه‌ای دارد که ممکن است با تغییر نیازها و داده‌ها به سرعت انعطاف‌پذیر نباشد.

## 3. Object-Relational Databases:

- نسبت به پایگاه‌های داده رابطه‌ای سنتی انعطاف‌پذیرتر هستند، اما پیچیدگی بالاتری دارند و بهینه‌سازی آن‌ها سخت‌تر است.

## 4. Object-Oriented Databases:

- مناسب برای برنامه‌های خاص منظوره، اما برای سیستم‌های عمومی مانند فروشگاه کتاب، انعطاف‌پذیری کمتری دارد.

---

### نتیجه‌گیری:

NoSQL، به‌ویژه پایگاه‌های داده سندمحور مانند MongoDB، انتخاب مناسبی برای این پروژه است. این پایگاه داده می‌تواند نیازهای غیرعملکردی را با ارائه مقیاس‌پذیری بالا، عملکرد سریع، امنیت مناسب، و سازگاری با معماری توزیع‌شده برآورده کند.

برای هر نوع پایگاه داده پیشنهادی، بهترین DBMS را بررسی می کنیم:

## 1. برای: NoSQL.

### • MongoDB:

#### ○ چرا مناسب است؟

- ساختار سندمحور آن برای ذخیره اطلاعات کتاب‌ها، کاربران، ناشران و تاریخچه خرید مناسب است.
- پشتیبانی از مقیاس‌پذیری افقی (Sharding) برای مدیریت حجم بالای داده‌ها.
- API های JSON بومی که با نیاز اپلیکیشن‌های وب و موبایل به خوبی همخوانی دارد.
- امکانات امنیتی مانند رمزنگاری داده‌ها و سازگاری با GDPR.

○ **معایب:** نیاز به مدیریت دقیق در مقیاس‌های خیلی بزرگ برای جلوگیری از پیچیدگی در طراحی.

### • Cassandra:

#### ○ چرا مناسب است؟

- به طور خاص برای سیستم‌های با مقیاس بالا و معماری توزیع شده طراحی شده است.
- قابلیت تحمل خطا و دسترسی 7/24.
- مناسب برای ذخیره داده‌های بزرگ مانند تاریخچه خرید یا لاگ‌های کاربران.
- **معایب:** عملکرد بهینه آن نیازمند پیکربندی پیچیده است.

## 2. برای پایگاه داده رابطه‌ای: (Relational Databases)

اگر نیاز به استفاده از پایگاه داده‌های رابطه‌ای دارید:

### • PostgreSQL:

#### ○ چرا مناسب است؟

- بسیار قدرتمند، متن‌باز و انعطاف‌پذیر.
- پشتیبانی از JSON و JSONB برای ذخیره داده‌های نیمه‌ساخت یافته.
- امکانات امنیتی پیشرفته (مانند احراز هویت دو مرحله‌ای و رمزنگاری).
- عملکرد خوب برای کوئری‌های پیچیده و گزارش‌گیری.

○ **معایب:** ممکن است در مقیاس‌های بسیار بزرگ نیاز به تنظیمات پیشرفته داشته باشد.

### • MySQL:

#### ○ چرا مناسب است؟

- یکی از محبوب‌ترین پایگاه داده‌های رابطه‌ای.
- رابط کاربری ساده و مناسب برای توسعه‌دهندگان با تجربه متوسط.
- پشتیبانی گسترده از انجمن‌های کاربری و منابع آموزشی.

○ **معایب:** در مدیریت داده‌های پیچیده و نیمه‌ساخت یافته محدودتر از PostgreSQL است.

انتخاب نهایی بستگی به مقیاس پروژه، نیاز به تحلیل داده‌ها، بودجه و زیرساخت موجود دارد.