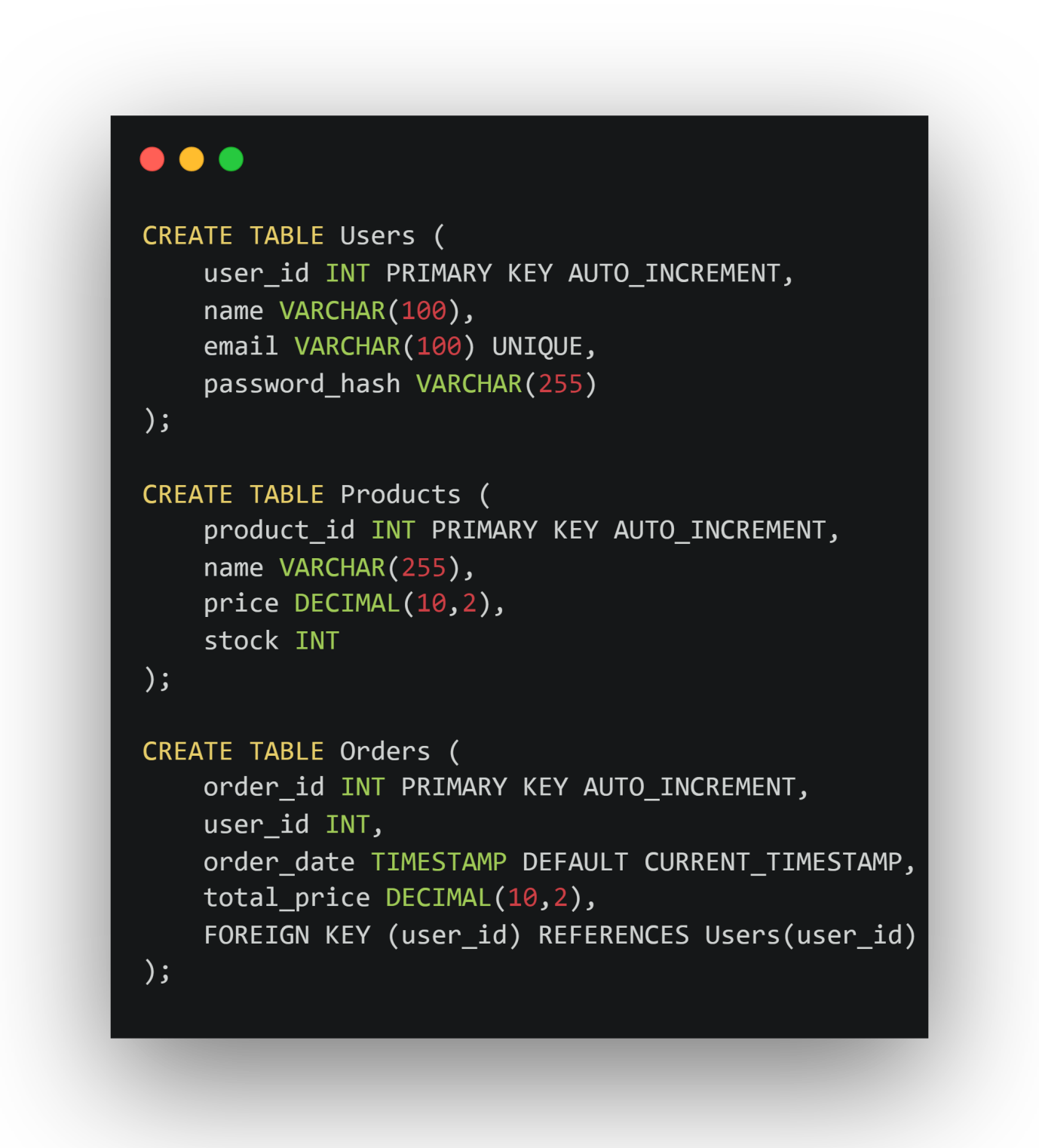
**.1 طراحی یک Relational Database Schema برای فروشگاه آنلاین**

در یک فروشگاه آنلاین، باید جداولی مانند کاربران، محصولات، سفارشات و پرداخت‌ها داشته باشیم.

یک فروشگاه آنلاین شامل جداول کاربران، محصولات، سفارشات و جزئیات سفارش است. جداول با کلیدهای اولیه و خارجی به هم مرتبط می‌شوند. داده‌ها نرمال‌سازی شده تا افزونگی کاهش یابد. از ایندکس‌ها برای بهبود سرعت جستجو استفاده می‌شود. هر سفارش شامل چندین محصول بوده و ارتباط چندبه‌چند بین محصولات و سفارشات وجود دارد.

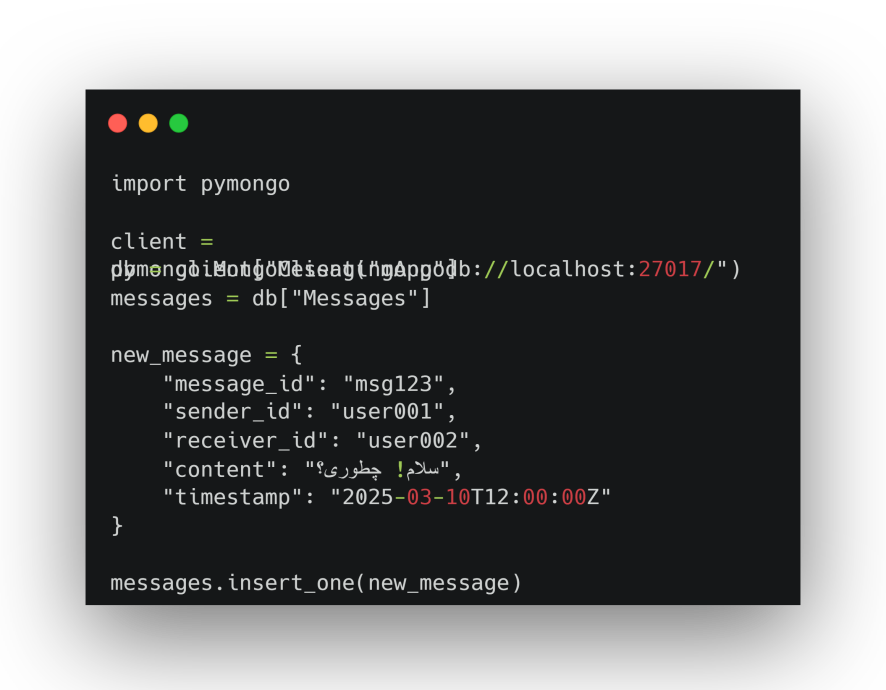
### ****2 .طراحی یک No SQL Database Schema برای سیستم پیام‌رسانی****

در پایگاه داده‌های No SQL مثلاً (Mongo DB)، داده‌ها به‌صورت اسناد ذخیره می‌شوند

در پایگاه داده‌های No SQL مانند MongoD3، داده‌ها در قالب اسناد ذخیره می‌شوند. برای سیستم پیام‌رسانی، هر پیام شامل شناسه فرستنده، گیرنده، متن و زمان ارسال است. داده‌ها به صورت توزیع‌شده در سرورهای مختلف ذخیره می‌شوند. طراحی **Schema-less** انعطاف‌پذیری بیشتری برای تغییرات آینده ایجاد می‌کند. پیام‌ها می‌توانند در مجموعه‌های جداگانه برای هر کاربر ذخیره شوند.

**(تصویر زیر مربوط به طرح کلی پایگاه داده پیام‌رسانی هستش)**

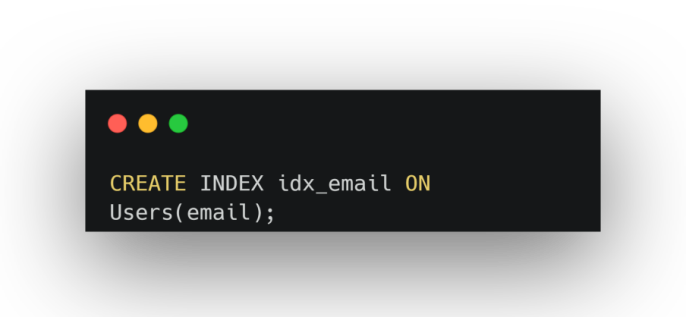


(تصویر زیر مربوط به کد ایجاد مجموعه در Mongo DB هستش)

### ****3 .طراحی یک Indexing Strategy برای پایگاه داده****

ایندکس‌ها باعث افزایش سرعت جستجو می‌شوند. مثلاً اگر زیاد بر اساس ایمیل کاربران جستجو انجام شود، می‌توان ایندکس ایجاد کرد:

در جداولی که حجم داده زیاد است، ایندکس روی فیلدهای پرکاربرد مانند **email** یا **order\_ date** اعمال می‌شود. در MySQL از CREATE INDEX و در Mongo DB از create\_ index() استفاده می‌شود. استفاده بیش‌ازحد از ایندکس می‌تواند باعث افزایش هزینه ذخیره‌سازی شود. انتخاب ایندکس مناسب بستگی به الگوی جستجوی داده‌ها دارد.

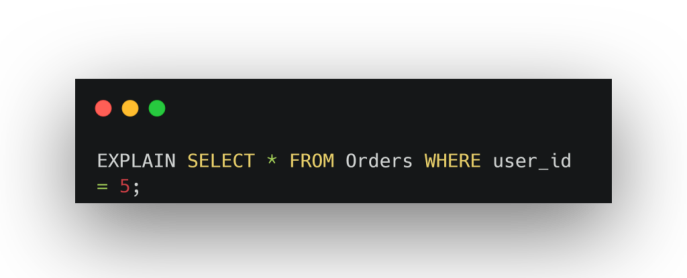


### ****.4 بهینه‌سازی Query (Query Optimization)****

برای بهینه‌سازی کوئری‌ها می‌توان از **EXPLAIN** در MySQL استفاده کرد تا عملکرد کوئری را تحلیل کنیم.

#### ****نکات بهینه‌سازی:****

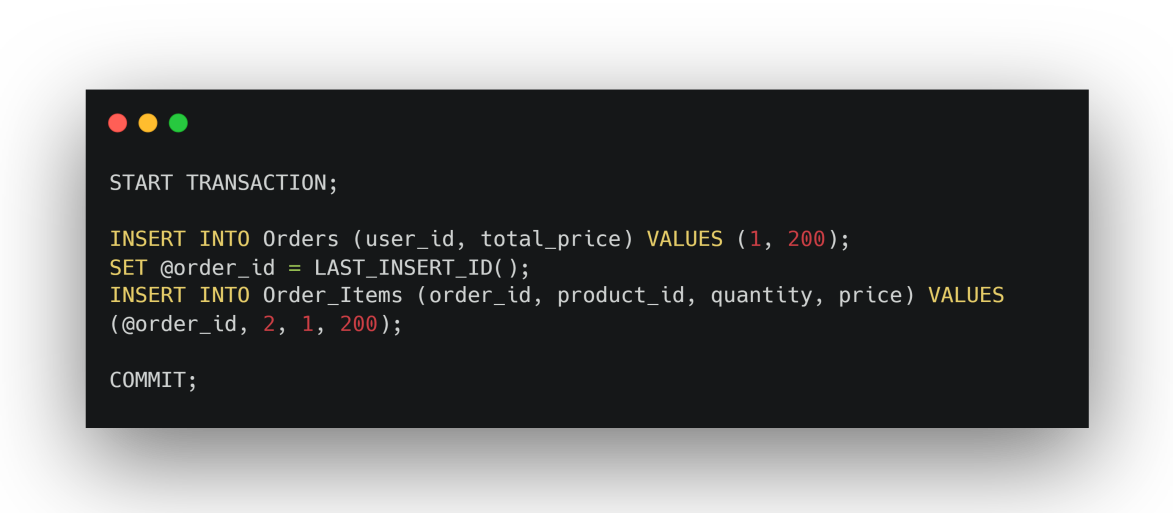
* از **ایندکس** برای فیلدهای پرتکرار استفاده کنید.
* از  **LIMIT** برای کاهش تعداد رکوردهای بازیابی شده استفاده کنید.
* از **JOIN**‌ های بهینه و فیلترها استفاده کنید.

بهینه‌سازی کوئری شامل کاهش زمان پردازش و افزایش کارایی پایگاه داده است. استفاده از EXPLAIN در SQL برای تحلیل عملکرد کوئری‌ها مفید است. استفاده از ایندکس، فیلترهای مناسب، و بهینه‌سازی **JOIN** می‌تواند کارایی را بهبود بخشد. کاهش تعداد رکوردهای بازگردانده شده با استفاده از **LIMIT** سرعت اجرا را افزایش می‌دهد. ذخیره نتایج موقت در کش (Cache) باعث کاهش فشار بر پایگاه داده می‌شود.

### ****.5 طراحی یک Transaction Management System****

تراکنش‌ها باید ویژگی‌های **ACID** را داشته باشند. در MySQL، تراکنش به این صورت مدیریت می‌شود:

تراکنش‌ها باید ویژگی‌های**) ACID** اتمی بودن، سازگاری، ایزوله بودن، دوام) را رعایت کنند. در SQL از START TRANSACTION, COMMIT و ROLLBACK برای مدیریت تراکنش‌ها استفاده می‌شود. اگر در یک عملیات چندین تغییر باید هم‌زمان انجام شوند، از تراکنش‌ها استفاده می‌کنیم. در سیستم‌های مالی، تراکنش‌ها باید به‌گونه‌ای مدیریت شوند که از دست رفتن داده جلوگیری شود Mongo DB. از **multi-document transactions** برای عملیات چندگانه پشتیبانی می‌کند.

****

### ****6 .طراحی یک Backup and Recovery Plan****

بکاپ‌گیری منظم از پایگاه داده ضروری است. می‌توان از my SQL dump استفاده کرد.

بکاپ‌گیری از داده‌ها برای جلوگیری از از دست رفتن اطلاعات ضروری است. از my SQL dump در MySQL و mongodump در Mongo DB برای گرفتن نسخه پشتیبان استفاده می‌شود. نسخه‌های پشتیبان باید در مکان‌های جداگانه و با برنامه‌ریزی منظم ذخیره شوند. برای بازیابی، می‌توان از my SQL < backup. SQL یا mongorestore استفاده کرد. پیاده‌سازی یک سیستم **incremental backup** می‌تواند فضای ذخیره‌سازی را بهینه کند.

### ****7 .طراحی یک Sharding Strategy****

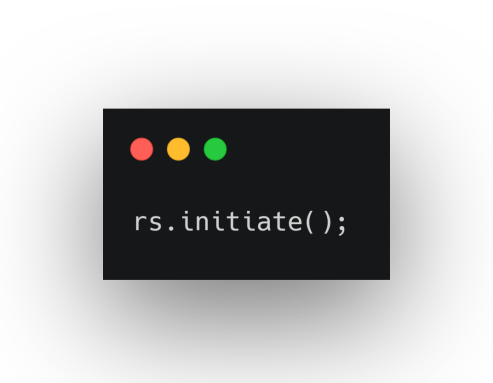
**Sharding** یعنی تقسیم داده‌ها روی چندین سرور و در Mongo DB، این کار به صورت زیر انجام می‌شود

Sharding داده‌ها را بر اساس یک کلید خاص بین چندین سرور توزیع می‌کند و در پایگاه داده‌های توزیع‌شده مانند Mongo DB، از  **hashed sharding** یا  **range sharding** استفاده می‌شود . کلید Shard باید به‌گونه‌ای انتخاب شود که توزیع متوازن داده‌ها حفظ شود. این روش باعث افزایش سرعت خواندن و نوشتن داده‌ها در حجم بالا می‌شود. انتخاب نادرست Shard Key می‌تواند باعث ایجاد **hotspot** و کاهش عملکرد شود.



### ****8 .طراحی یک Replication Strategy****

**Replication** به معنای نگه داشتن یک نسخه از پایگاه داده روی چندین سرور است

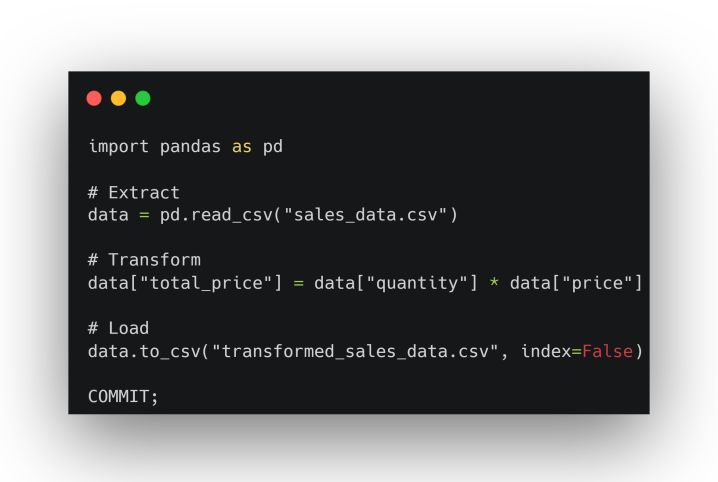
Replication باعث نگه‌داری نسخه‌های کپی از پایگاه داده در چندین سرور می‌شود ، این روش باعث افزایش قابلیت اطمینان و دسترس‌پذیری سیستم می‌شود ، در MySQL از **Master-Slave Replication** و در Mongo DB از  **Replica Set** استفاده می‌شود ، در صورت خرابی سرور اصلی، یکی از نسخه‌های کپی به‌طور خودکار جایگزین می‌شود. Replication به بهبود عملکرد خواندن کمک می‌کند اما نوشتن را کمی کندتر می‌کند

### ****.9 پیاده‌سازی فرآیند ETL برای Data Warehouse****

فرآیند **ETL** شامل استخراج داده‌ها، تبدیل آن‌ها و ذخیره در انبار داده است.

ETL شامل سه مرحله  **Extract استخراج، Transform تبدیل، Loadبارگذاری** است. داده‌ها از منابع مختلف مانند پایگاه داده‌های عملیاتی و فایل‌های CSV استخراج می‌شوند. در مرحله تبدیل، داده‌ها پاک‌سازی، تجمیع و نرمال‌سازی می‌شوند. در نهایت، داده‌های پردازش‌شده به

انبار داده (Data Warehouse) منتقل می‌شوند. ابزارهایی مانند **Apache Nifi، Talend، و Pandas** برای پیاده‌سازی ETL استفاده می‌شوند.

****

### ****10 .طراحی یک Data Migration Plan****

انتقال داده‌ها از یک پایگاه داده به دیگری نیاز به برنامه‌ریزی دارد.

#### ****مثال: انتقال از MySQL به**** Post gre SQL

انتقال داده‌ها شامل استخراج از پایگاه داده قدیمی، تبدیل به فرمت جدید و بارگذاری در سیستم جدید است. فرآیند مهاجرت باید بدون از دست رفتن داده و کمترین Downtime انجام شود. ابزارهایی مانند **AWS DMS،** **PG Loader و MySQL Workbench** برای مهاجرت داده‌ها استفاده می‌شوند. تست و اعتبارسنجی داده‌ها پس از مهاجرت ضروری است تا از صحت انتقال اطمینان حاصل شود. در صورت بروز خطا، امکان بازگردانی به نسخه قبل باید فراهم باشد.