

## مقایسه پایگاه داده‌ها:

- مقایسه نسبت به relational schema فاز اول:

مزایا	معایب
جدول مجزا برای اعضای گروه و اعضای کامیونیتی	رابطه یک‌به‌یک میان User و Post
اضافه کردن بخش ROLE به جدول‌های COMMUNITYMEMBERSHIP و GROUPMEMBERSHIP	نبود جدول‌های مربوط به تنظیمات امنیتی (PrivacySetting)
اضافه کردن TIMESTAMP به جدول‌ها	نبود جدول مربوط به پست‌های مورد علاقه کاربر (Favorite)
جدول‌های جدا برای HASHTAG	نبود جدول TimeLine
جدا بودن جدول USER و USERPROFILE	نبود جدول Notification
جدول جدا برای BLACKLIST	نبود جدول Reaction

## مزایا:

- ایجاد جدول مجزا برای اعضای گروه و اعضای کامیونیتی: با این کار از تکرار داده‌ها جلوگیری شده و داده‌های مرتبط به جدول‌های کوچک‌تر و متمرکزتر کپسوله می‌شوند.

- اضافه کردن بخش role به جدول‌های CommunityMembership و GroupMembership: انعطاف‌پذیری بیشتری برای داده‌ها اعمال می‌کند.

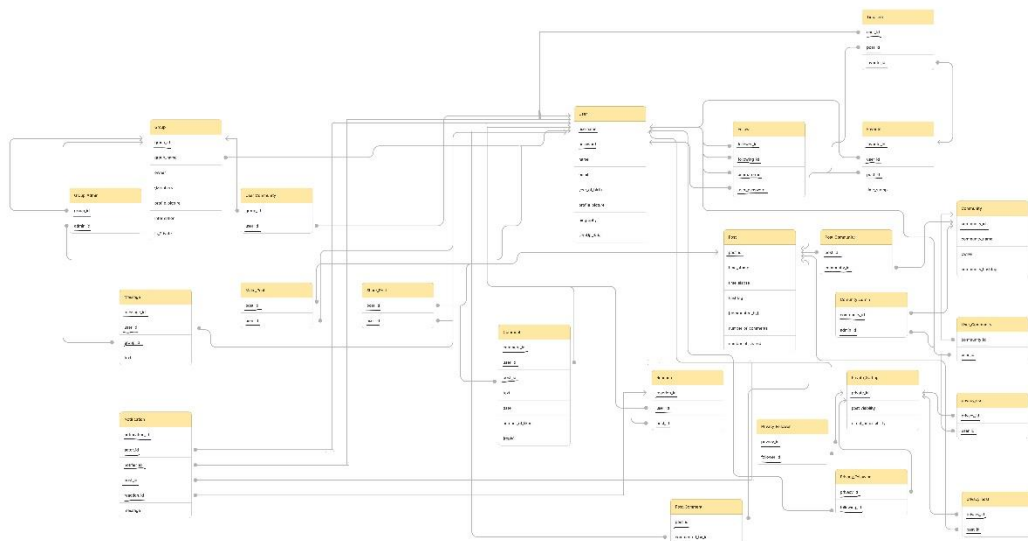
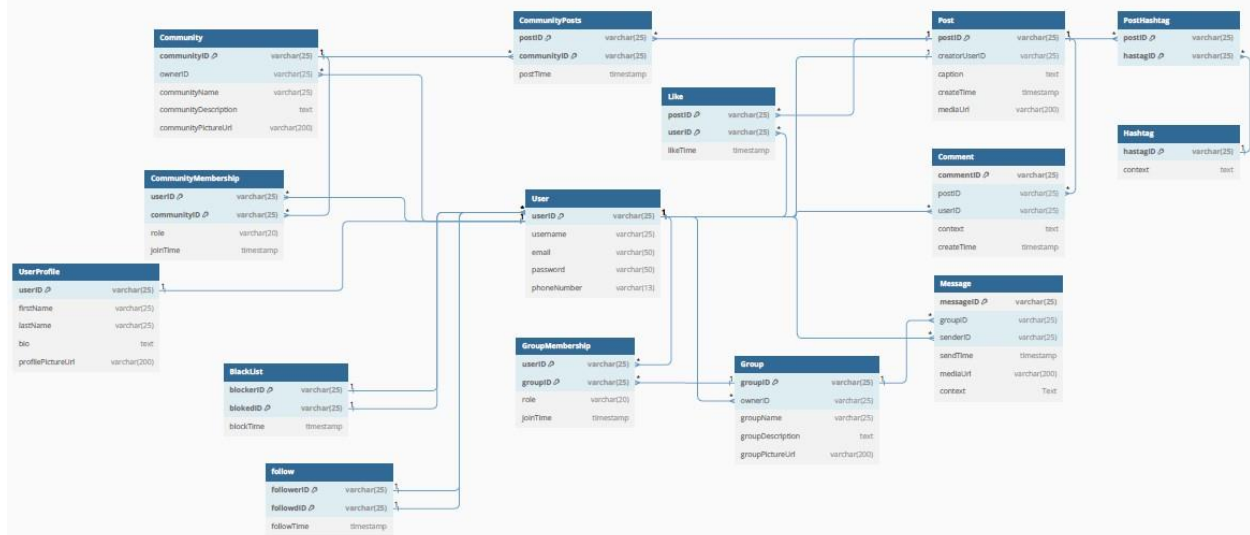
- اضافه کردن بخش Timestamp به جدول‌های Like، Post، Comment، Message و Follow و دو جدول Membership: به دنبال کردن فعالیت‌های کاربر و برخی الزامات امنیتی کمک می‌کند.
- جدول‌های جدا برای Hashtag و PostHashtag: جداسازی مفهوم هشتگ و پست پایگاه داده را کارآمدتر و مقیاس‌پذیرتر می‌کند و مدیریت روابط آسان‌تر می‌شود.
- جدا بودن جدول User و UserProfile: این قابلیت به تفکیک بهتر داده‌ها و جلوگیری از بار عملیاتی بیشتر، کمک می‌کند.
- جدول جدا برای BlackList: با این قابلیت جداسازی کاربرها ساده‌تر و از تکرار داده‌ها جلوگیری می‌شود.

#### معایب:

- رابطه یک‌به‌یک میان User و Post: این رابطه منطقی نیست زیرا هر کاربر می‌تواند یک پست ایجاد کند. در صورتی که رابطه کاربر و پیغام یک‌به‌چند و منطقی‌تر است.
- نبود جدول‌های مربوط به تنظیمات امنیتی: یکی از الزامات فاز اول قیدهای امنیتی بود که در این شما لحاظ نشده است.
- نبود جدول مربوط به پست‌های مورد علاقه کاربر: وجود این جدول به دنبال کردن فعالیت‌های کاربر کمک می‌کند.
- نبود جدول مربوط به TimeLine: وجود این جدول دنبال کردن فعالیت‌های کاربران را راحت‌تر می‌کند.

○ نبود جدول مربوط به اعلان‌ها: از الزامات فاز اول و یک نرم‌افزار شبکه‌ی اجتماعی، جدول مربوط به اعلان‌ها و داده‌ها و تاریخچه‌ی آن‌هاست.

○ نبود جدولی مربوط به ری‌اکشن‌های کاربر فراتر از لایک: اگر واکنش‌های کاربر حالتی جامع‌تر داشته باشند، به مقیاس‌پذیری سیستم کمک می‌کند.



- مزایا و معایب کلی (با توجه به اصول نرمال سازی):

مزایا	معایب
ذخیره و بازیابی ساختاریافته و کارآمد داده‌ها	احتمال بروز Data Bloat
کپسوله سازی موجودیت‌ها	عدم پشتیبانی از پیگیری فعالیت کاربر
پشتیبانی از روابط کاربر	پشتیبانی محدود از چندرسانه‌ای
مدیریت کارآمد هشگ	نبود ملاحظات امنیتی
مقیاس پذیری	پشتیبانی محدود برای تعاملات کاربر فراتر از لایک و کامنت
سازماندهی داده‌ها حول محور کاربر	نبود جدول Reaction
ضبط روابط بین موجودیت‌ها	پتانسیل کاهش عملکرد در کوئری‌های پیچیده
	افزایش پیچیدگی
	بار عملیاتی احتمالی

مزایا:

○ ذخیره و بازیابی ساختاریافته و کارآمد داده‌ها: این پایگاه داده دارای ساختار منظمی است و از مدل رابطه‌ای استفاده می‌کند، که به ذخیره و بازیابی کارآمد داده‌ها کمک می‌کند. نحوه استفاده از کلیدهای خارجی، اطمینان از صحت داده و ارتباطات بین جداول را فراهم می‌آورد.

○ کپسوله‌سازی موجودیت‌ها: این پایگاه داده به طور مؤثری موجودیت‌های مختلف را مانند کاربران، پست‌ها، نظرات، لایک‌ها، گروه‌ها و هشتک‌ها کپسوله می‌کند. این جداسازی مدیریت داده را ساده‌تر کرده و کوئری‌ها را کارآمدتر می‌سازد.

○ پشتیبانی از روابط کاربر: پایگاه داده روابط مختلف کاربران را از جمله دنبال‌کننده-دنبال‌شونده، عضویت در گروه و عضویت در کامیونیتی را ثبت می‌کند. این ویژگی امکان پیگیری تعاملات کاربر و ساختارهای گروه را فراهم می‌آورد.

○ مدیریت کارآمد هشتک: جدول مجزا از هشتک و پست‌ها (PostHashtag) به طور کارآمد هشتک‌ها و ارتباط آن‌ها با پست‌ها را مدیریت می‌کنند. این امر باعث کاهش تکرار داده و بهبود عملکرد کوئری می‌شود.

○ مقیاس‌پذیری: طراحی پایگاه داده توسعه‌پذیر است، و این اجازه را می‌دهد که جداول و داده‌های جدید بدون فدا کردن عملکرد یا صحت داده اضافه شوند.

○ سازماندهی داده‌ها حول محور کاربر: ساختار پایگاه داده به منظور سازماندهی داده‌ها بر اساس کاربران طراحی شده است. برای هر کاربر، جداول مجزا برای پروفایل کاربر، پست‌ها، نظرات، لایک‌ها و عضویت‌های گروه وجود دارد. این ساختار به مدیریت و تجزیه و تحلیل محتوای تولیدشده توسط کاربران و تعاملات آنها سادگی بخشیده است.

○ ضبط روابط بین موجودیت‌ها: پایگاه داده به طور مؤثر روابط بین موجودیت‌های مختلف در زمینه رسانه‌های اجتماعی را ضبط می‌کند. جدول PostHashtag پست‌ها را به هشتک‌های مرتبط متصل می‌کند، جدول Comment نظرات را به پست‌ها مرتبط می‌کند و جدول GroupMembership کاربران را به گروه‌ها متصل می‌کند. این کار تجزیه و تحلیل رفتار کاربر و الگوهای به اشتراک گذاری محتوا را تسهیل می‌کند.

## معایب:

- احتمال بروز بیش‌ترین داده: اگر پایگاه داده به طور چشمگیری رشد کند، احتمال بروز بیش‌ترین داده (Data Bloat) ممکن است، به ویژه در جدول PostHashtag اگر تعداد زیادی هشتگ استفاده شود. این ممکن است منجر به مشکلات عملکردی شود.
- عدم پشتیبانی از پیگیری فعالیت کاربر: پایگاه داده به طور صریح فعالیت کاربر را مانند زمان‌های مشاهده پست، تعاملات نظر، یا شرکت در گروه را پشتیبانی نمی‌کند. این اطلاعات می‌تواند برای تجزیه و تحلیل و شخصی‌سازی ارزشمند باشند.
- پشتیبانی محدود از چندرسانه‌ای: پایگاه داده اصولاً بر روی داده‌های مبتنی بر متن تمرکز دارد، در حالی که محتوای چندرسانه‌ای مانند تصاویر، ویدیوها و فایل‌های صوتی به طور مستقیم ذخیره نمی‌شود. این ممکن است محدود عملکردی پایگاه داده را محدود کند.
- ملاحظات امنیتی: طراحی پایگاه داده به طور صریح به مسائل امنیتی مانند تأیید هویت کاربر، مجوزدهی و حفاظت از داده پاسخ نمی‌دهد. اقدامات امنیتی اضافی برای حفاظت اطلاعات حساس کاربر نیاز است.
- پشتیبانی محدود برای تعاملات کاربر فراتر از لایک و کامنت: پایگاه داده تنها بر روی لایک و کامنت به عنوان راه‌های تعامل کاربر متمرکز است. در حالی که این معیارها مهم هستند، اضافه کردن تعاملات دیگر مانند Share، Mention و Reaction می‌تواند دیدی جامع‌تر از کاربر ارائه دهد.
- پتانسیل کاهش عملکرد در کوئری‌های پیچیده: با افزایش پایگاه داده و تعداد کاربران و پست‌ها، کوئری‌های پیچیده که شامل جدول‌ها و join‌های چندگانه هستند، ممکن است هزینه محاسباتی بیشتری داشته باشند. این ممکن است بر عملکرد سیستم تأثیر بگذارد و نیاز به تکنیک‌های بهینه‌سازی داشته باشد.

○ افزایش پیچیدگی: یک پایگاه داده بسیار نرمال سازی شده می تواند به دلیل تعداد بسیار بالای جداول و روابط افزایش یافته، پیچیده تر شود. این ممکن است باعث کاهش قابلیت فهم و نگهداری پایگاه داده برای کاربران شود.

○ بار عملیاتی احتمالی: در برخی موارد، نیاز به اتصالات اضافی برای بازیابی داده از چندین جدول می تواند منجر به کاهش کارایی نسبی نسبت به پایگاه داده کمتر نرمال سازی شده شود.

## • سطح نرمال سازی:

ساختار این پایگاه داده به فرم نرمال (BCNF) که یک فرم سخت گیرانه تر از فرم نرمال سوم (3NF) است، نزدیک تر است. BCNF نیازمند آن است که هر ویژگی غیر اصلی به طور کامل وابسته به کلید اصلی باشد و وابستگی انتقالی (Transitive Dependency) به کلید اصلی نداشته باشد.

در این پایگاه داده، به عنوان مثال ستون های `firstName`، `lastName`، `bio` و `profilePictureUrl` در جدول `UserProfile` به طور کامل وابسته به کلید خارجی `userId` هستند و این نیازمندی های BCNF را برآورده می کند. این امر باعث حذف احتمال وابستگی های انتقالی می شود که ممکن است منجر به ناهمسانی های داده شود.