

**طراحی پایگاه داده**

**دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف**

زمستان 1402



**معین آعلی - 401105561­**

**فهرست عناوین**

[**1. سوال شماره 1 2**](#_Toc160853246)

[**2. سوال شماره 2: 3**](#_Toc160853247)

[**3. سوال شماره 3: 6**](#_Toc160853248)

## سوال شماره 1

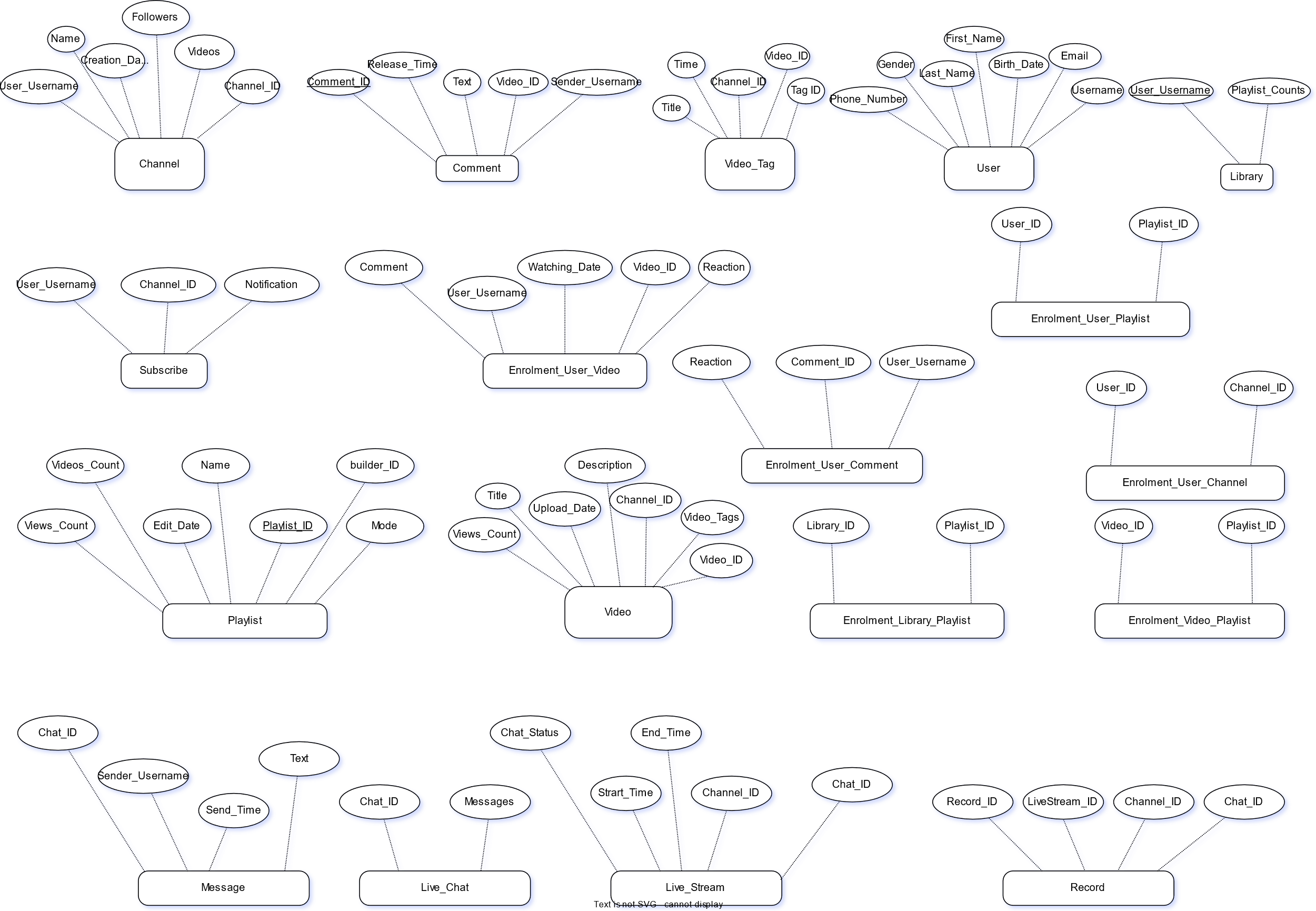
## سوال شماره 2:

در این سوال قرار است ما یک پایگاه‌داده برای پلتفرم یوتیوب طراحی کنیم.

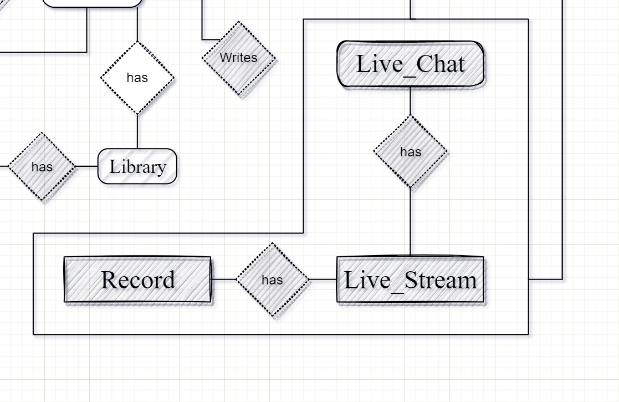
در ابتدا با توجه به سناریو توضیح داده شده در صورت سوال، Entityها را مشخص می‌کنیم:

* **User**
* **Library**
* **Video\_Tag**
* **Comment**
* **Channel**
* **Account**
* **Enrollment\_User\_Playlist**
* **Enrollment\_User\_Channel**
* **Enrollment\_User\_Comment**
* **Enrollment\_User\_Video**
* **Subscribe**
* **Playlist**
* **Video**
* **Enrollment\_Library\_Playlist**
* **Enrollment\_Video\_Playlist**
* **Message**
* **Live\_Chat**
* **Live\_Stream**
* **Record**

در این عکس تمامی Entityها به همراه Attributeهایشان نمایش داده شده است:



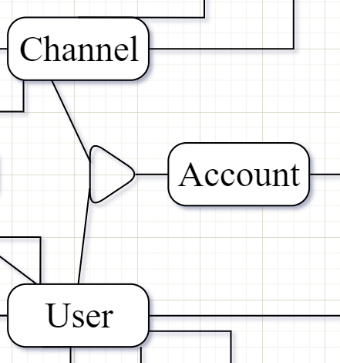
حال به سراغ روابط تجمع در موجودیت‌ها می‌رویم:

* **Live\_Chat**
* **Live\_Stream**
* **Record**

این موجودیت‌ها با یک دیگر رابطه‌ی Aggregation دارند.

حال به سراغ روابط تعمیم و تخصیص(Specialization / Generalization) در بین موجودیت‌ها می‌رویم:

* **Account**
* **User**
* **Channel**

این موجودیت‌ها دارای رابطه Specialization هستند.

در ERD موجود در صفحه‌ی بعد، Weak Entity **با مستطیل دولایه** نمایش داده شده‌اند.

همچنین Weak Relationها با **مستطیل دولایه** نمایش داده شده‌اند.

همچنین کاردینالیتی روابط سه نوع دارند:

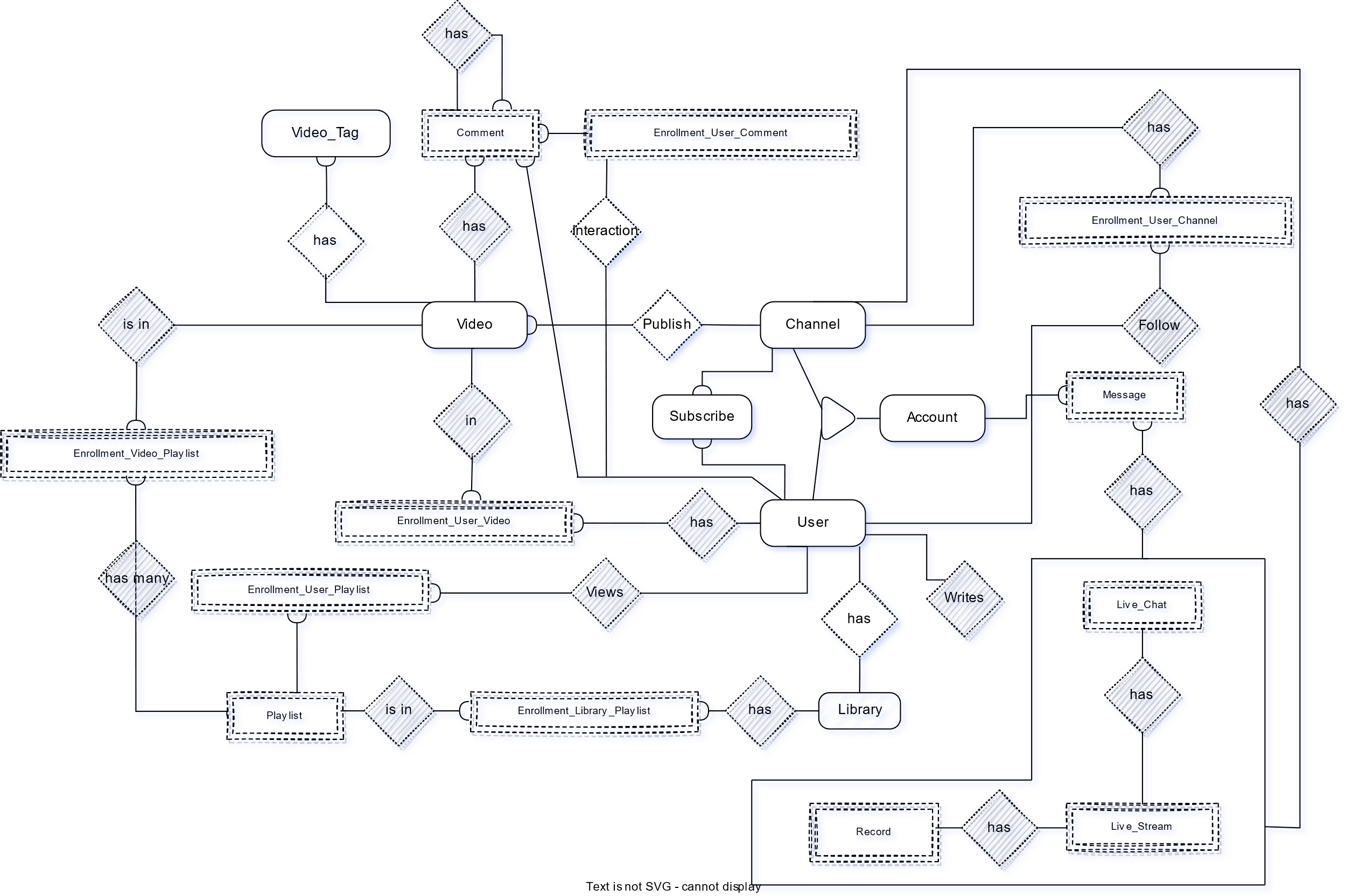
* **Many to Many**
* **One to Many**
* **One to One**

**نکته‌ی مهم:**

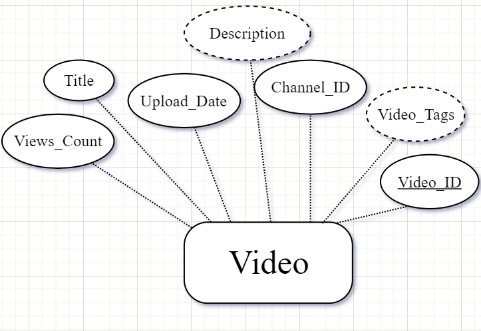
در سناریوی صورت سوال، برخی صفات مد نظر است که نیاز نیست برایشان Attribute جدا تعریف کنیم. آن صفات، صفت مشتق(Derived attributes) هستند و از باقی موارد قابل بدست آمدن هستن.

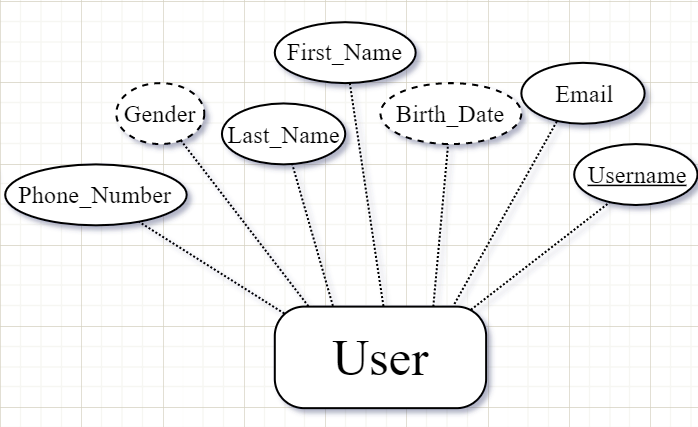
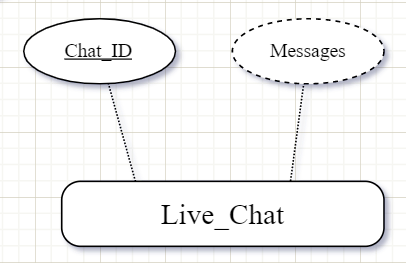
به عنوان مثال عمر یک کانال، نیاز نیست که ذخیره شود و مداوم آپدیت شود، بلکه از روی Creation\_Date قابل محاسبه است.

حال در ادامه روابط بین موجودیت‌ها را نمایش می‌دهیم:



حال در این مرحله که نمودار را کشیدیم، بعضی ویژگی‌ها که ضروری نیستند را به صورت Nullable در نظر می‌گیریم:





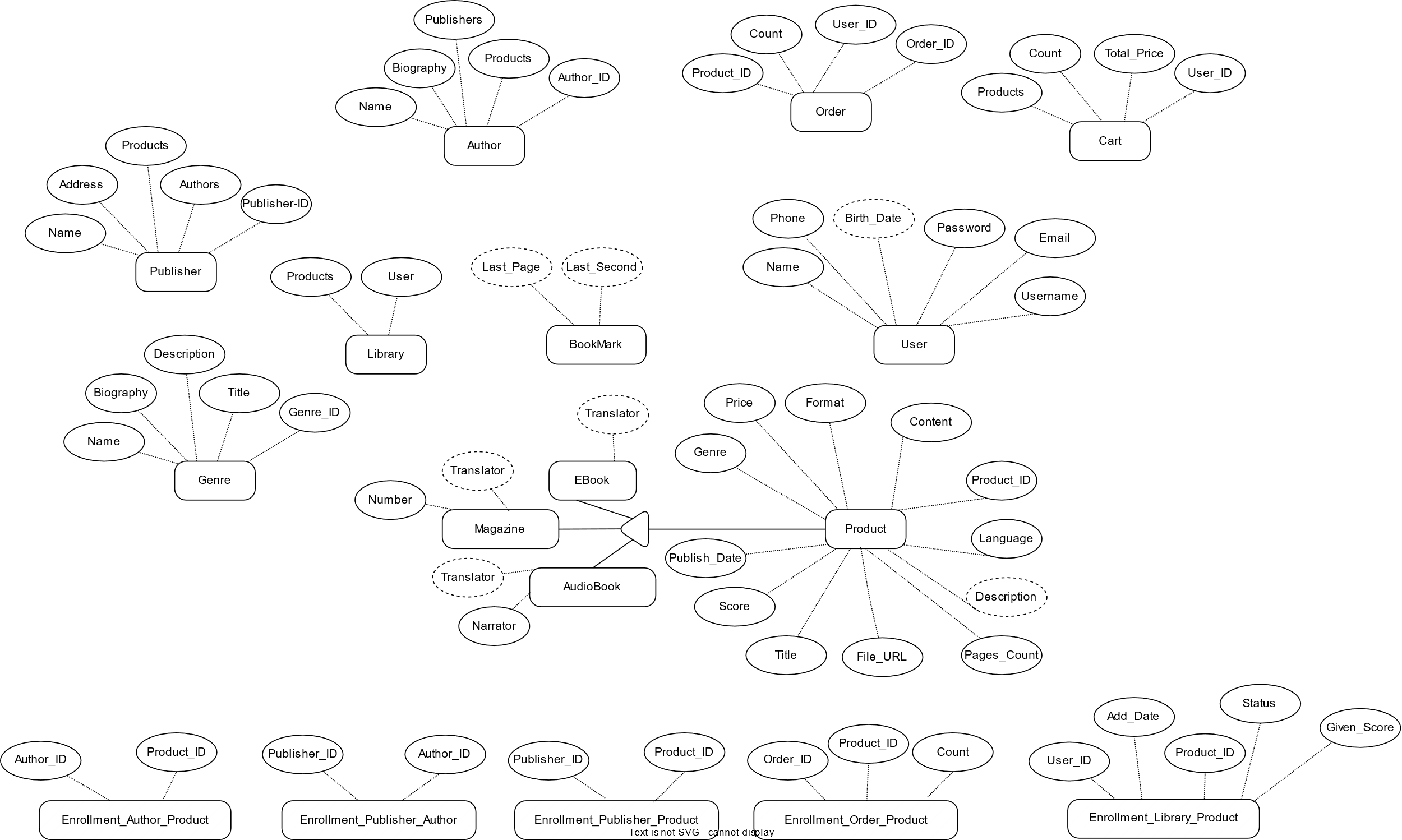
## سوال شماره 3:

در این سوال قرار است ما یک پایگاه‌داده برای پلتفرم فیدیبو طراحی کنیم.

در ابتدا با توجه به سناریو توضیح داده شده در صورت سوال، Entityها را مشخص می‌کنیم:

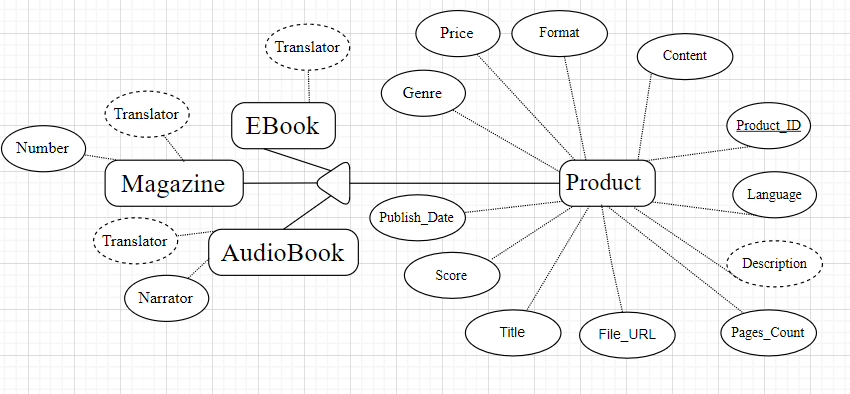
* **Publisher**
* **Author**
* **Library**
* **Genre**
* **Bookmark**
* **Order**
* **User**
* **Product**
* **EBook**
* **AudioBook**
* **Magazine**
* **Cart**
* **Enrollment\_Library\_Product**
* **Enrollment\_Order\_Product**
* **Enrollment\_Publisher\_Product**
* **Enrollment\_Author\_Product**

در عکس زیر تمامی Entityها به همراه Attributeهایشان آمده است:



حال به سراغ روابط تعمیم و تخصیص(Specialization / Generalization) در بین موجودیت‌ها می‌رویم:

* **Product**
* **EBook**
* **Magazine**
* **AudioBook**

این موجودیت‌ها دارای رابطه‌ی تخصیص (Specialization) هستند.

در ERD موجود در صفحه‌ی بعد، Weak Entity **با مستطیل دولایه** نمایش داده شده‌اند.

همچنین Weak Relationها با **مستطیل دولایه** نمایش داده شده‌اند.

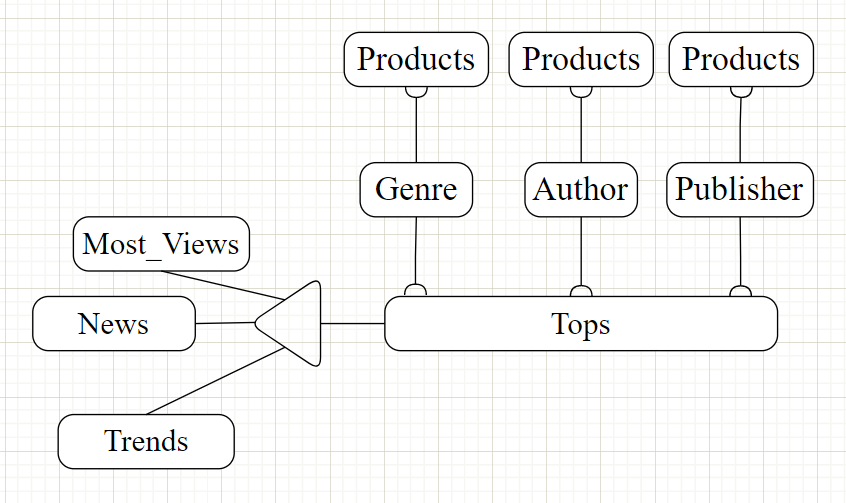
همچنین کاردینالیتی روابط سه نوع دارند:

* **Many to Many**
* **One to Many**
* **One to One**

**نکته‌ی مهم:**

در سناریوی صورت سوال، برخی صفات مد نظر است که نیاز نیست برایشان Attribute جدا تعریف کنیم. آن صفات، صفت مشتق(Derived attributes) هستند و از باقی موارد قابل بدست آمدن هستن.

به عنوان مثال، لیست کتاب‌های محبوب این هفته نیاز به ذخیره‌سازی ندارند و از روی روابط زیر قابل دست‌یابی هستند:

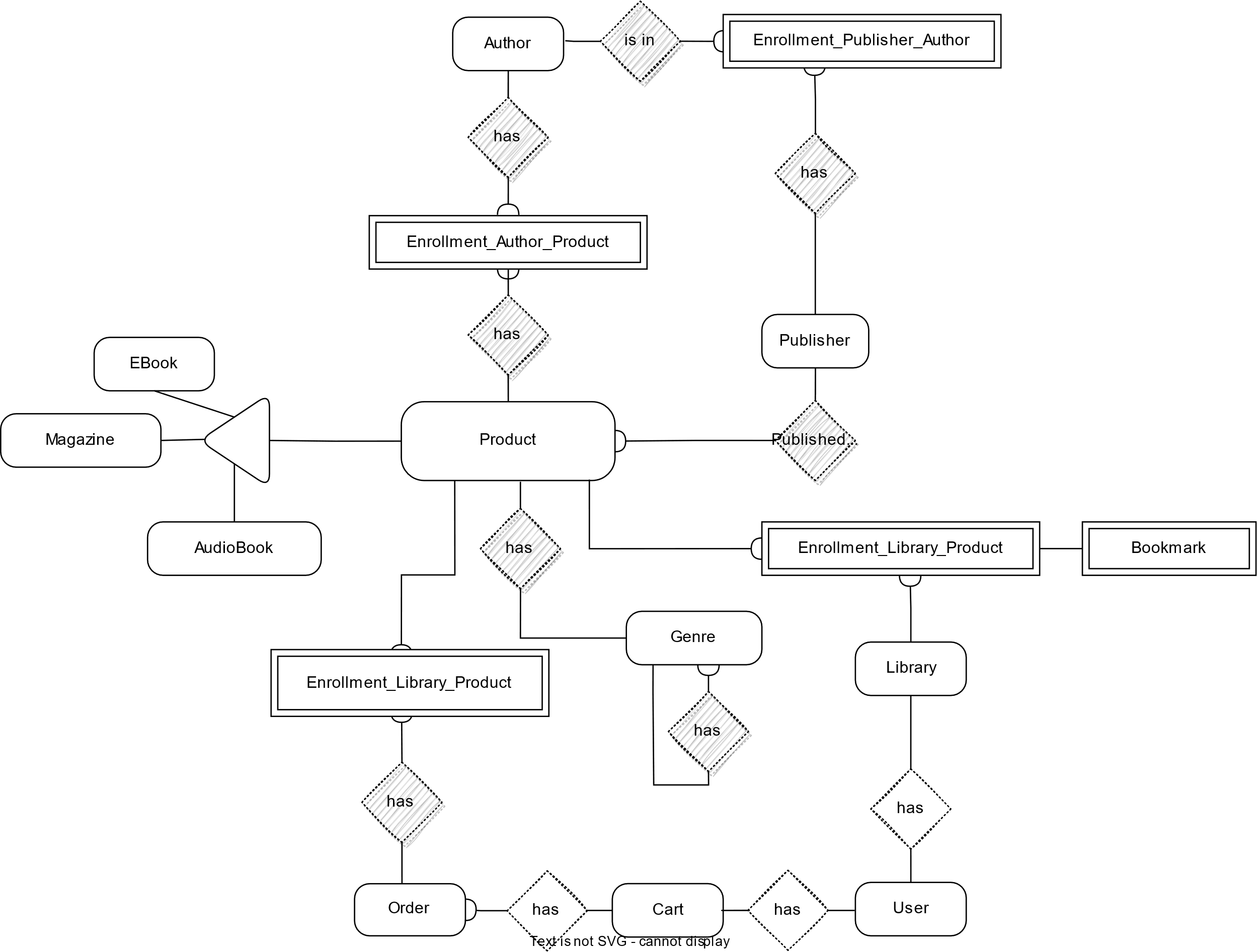


اگر لیست کتاب‌های محبوب هر ژانر، نویسنده و یا انتشارات را بخواهیم، کافیست ازطریق آن به Productهایی که با آن رابطه دارند دست یابیم و از طریق Simple Attributeیی به نام Score، آن‌ها را شناسایی کنیم و به Tops بدهیم.

بعضی از صفت‌ها هم از نوع Composite هستند، می‌توان آن‌ها با ترکیب‌ کردن صفت‌های ساده‌ای که داریم، بدست آوریم.

به عنوان مثال: اسم کامل یک نویسنده.

حال در ادامه روابط بین موجودیت‌ها را نمایش می‌دهیم:



**توجه:** آن صفت‌هایی که دورشان به صورت خط-چین است، قابلیت Null بودن دارند. به عبارتی آن‌ها Nullable هستند.