# به نام خدا

# سند نهایی پروژهٔ طراحی پایگاهدادهها

# استاد: دکتر امینی

اعضای گروه: امیرعباس عسگری ۹۷۱۰۲۰۸۸ محمدرضا عبدی ۹۷۱۰۲۸۵ محمدجواد حمزه ۹۷۱۰۱۵۵۳

عنوان پروژه: سامانهٔ پایگاهی اطلاعات فیلم (مشابه وبسایت IMDb)

تمر کز اعضای گروه در انجام بخشهای مختلف پروژه: در فازهای صفر، یک، و دو پروژه تقریباً همهٔ اعضا تمرکز یکسانی در طراحی و اجرای بخشهای مختلف پروژه: در فازهای صفر، یک، و دو پروژه تقریباً همهٔ اعضا تمرکز یکسانی در طراحی و اجرای بخشهای مختلف بروژه: در فازهای برکردن فاز سوم محمدرضا عبدی پیاده سازی جدولها، توابع و رهاناهای پایگاه داده را انجام داده، امیرعباس عسگری نوشتن برنامهٔ جاوایی که هدفش تولید داده های نمونهٔ موردنیاز برای پرکردن جدولهای پایگاه داده داشته است، و نهایتاً محمد جواد حمزه بخش نرمال ترسازی و توضیحات مربوط به آن را به انجام رسانده است.

#### نوع موجوديتها

#### کاربر سایت (عادی یا ویژه)

- صفات: ايميل (صفت شناسهٔ اول)- نام- گذرواژه- شمارهٔ تلفن
- زیرنوعها (تخصیص کامل و مجزا) : عادی و ویژه با این تفاوت که کاربر عادی صرفاً امکان ویرایش تعداد محدودی فیلم (حداکثر به تعداد edit\_limit\_num که حضوب میشود) را دارد. همچنین امکان انحصاری مشاهدهٔ جزئیات فروش فیلمها به کاربران ویژه از طریق تعریف دیدی با دسترسی کسترده تر به اطلاعات پایگاهداده فراهم میشود که کاربران عادی به آن دسترسی نخواهند داشت. لذا این تفاوت در سطح پیادهسازی مورد بحث قرار خواهد گرفت.
- نقد کردن فیلمها توسط کاربر به این صورت است که هر کاربر می تواند برای هر فیلم یک نمره، یک نظر متنی، با هر دو را درج کند که موضوع نظردهی در قالب
   رابطهای بین کاربر و فیلم مدل شدهاست.
- o علاقمندی هر کاربر به فیلمها در قالب رابطهای به نام interested\_in مدل شده است و به این ترتیب نیازی نیست لیست علاقمندی ها به شکل یک نوع موجودیت مجزا در نظر گرفته شود بلکه هر کاربر از طریق یک پرسمان به لیست فیلمهای موردعلاقهٔ خود دسترسی پیدا می کند.

#### · فیلم (سریال و غیر سریال)

- صفات: شناسهٔ فیلم (صفت شناسهٔ اول) آخرین تاریخ تغییر اطلاعات فیلم- سال تولید (صفت مرکب شامل شروع و پایان بازهٔ تولید)- نام- زبان- مدت- ژانر- بودجهفروش- سود (صفت مشتق حاصل از تفاضل فروش و بودجه)- لینک خلاصهٔ فیلم- کشور(های) سازنده (صفت چندمقداری)- ردهٔ سنی- لینک تیزر فیلم (صفت چندمقداری)
   چندمقداری)- لینک تصاویر فیلم (صفت چندمقداری)
- ⊙ زیرنوع (تخصیص ناقص و مجزا): سریال که نوع موجودیت ضعیف «فصل» به آن وابستگی وجودی دارد که نوع موجودیت فصل شامل شمارهٔ فصل (صفت ممیزه)
   و تعداد قسمتها است. همچنین سریال تعداد کل قسمتها را به عنوان یک صفت مشتق دارد که از جمع تعداد قسمتهای همهٔ فصلها به دست میآید.
- توجه کنید که یک فیلم مشخص جایزهٔ یک عنوان معین از یک جشنوارهٔ خاص را حداکثر یک بار (در یک سال) می تواند ببرد و به همین دلیل صفت سال برای رابطهٔ fwins صفت ممیزه نیست.

# - عوامل فیلم (بازیگر، کارگردان، تهیهکننده، و نویسنده)

- صفات: ایمیل (صفت شناسهٔ اول) نام شمارهٔ تلفن تاریخ تولد (صفت مرکب) تاریخ مرگ (صفت مرکب، هیچمقدارپذیر) جنسیت سن (صفت مشتق قابل محاسبه از روی تاریخ تولد و تاریخ مرگ یا تاریخ کنونی)
  - 🔾 زیرنوعها (نخصیص کامل و همپوشا): کارگردان، بازیگر، نویسنده، و تهیه کننده که در نوع ارتباط با نوع موجودیت فیلم تفاوتهایی دارند.
    - هر فیلم یک یا دو نقش اول دارد که اگر دو نقش اول داشته باشد، حتماً یکی مرد و دیگری زن است.

### جایزه (نوع موجودیت ضعیف وابسته به فیلم یا عوامل)

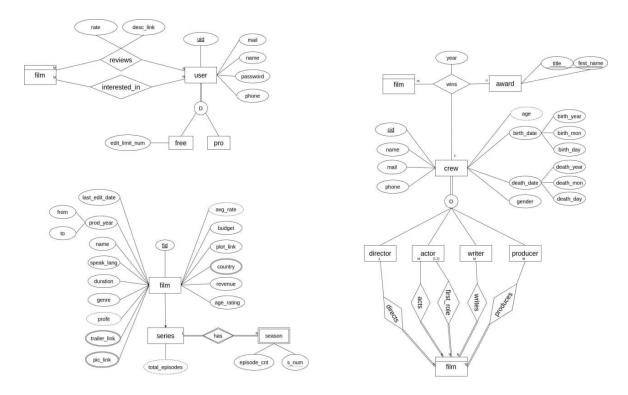
- صفات: نام جشنواره- عنوان (ترکیب این دو صفت ممیزه است)
- به عنوان مثال جایزهٔ اسکار (نام جشنواره) بهترین بازیگر نقش اول مرد (عنوان) به یک بازیگر مشخص اهدا می شود و سال دریافت جایزه به عنوان صفت ممیزهٔ رابطهٔ بردن جایزه (cwins) شمرده می شود. دقت کنید که بر خلاف فیلم، عوامل فیلم می توانند یک عنوان جایزه از یک جشنواره را در چند سال مختلف ببرند و لذا صفت سال برای رابطهٔ cwins صفت ممیزه است.

### نیازمندیها:

- جستجو بر اساس نام فیلم، کارگردان، تهیه کننده، بازیگر، نقش اول، سال تولید، بازهٔ فروش، بازهٔ بودجهٔ ساخت، ژانر، کشور (های) سازنده، زبان فیلم، درجهٔ سنی، بازهٔ مدت فیلم، نوع جایزه (ها)
  - امکان نمره دهی و اصلاح نمرهٔ فیلم (۱ تا ۱۰)، نمره دهی به نقدهای آن و مشاهدهٔ میانگین نمرهٔ کل کاربران به یک فیلم

- امكان افزودن و حذف فيلم از ليست علاقمنديها
- امكان مشاهدهٔ ليستى مرتب از فيلمها بر اساس معيارهايي مثل كل فروش، ميانگين نمره، سال توليد، بودجهٔ ساخت، تعداد بازيگران، و ...
  - امکان مشاهدهٔ خلاصهٔ فیلم و نقدهای کاربران به ترتیب نزولی نمرهٔ دادهشده به آن نقدها
    - ثبت نقد دربارهٔ هر فیلم
    - امكان ويرايش اطلاعات حداكثر پنج فيلم براى كاربران عادى
      - قابلیت قرار دادن لینک تیزر و تصاویر فیلم
  - امكان مشاهدهٔ جزئيات فروش فيلمها و ويرايش اطلاعات تعداد نامحدودي فيلم صرفاً براي كاربران ويژه
  - اگر فیلم از نوع سریال باشد، تعداد قسمت و تعداد فصلهای آن نیز ثبت شده و قابل مشاهده خواهد بود.
- دسترسیهای کلی کاربران برای تغییر اطلاعات فیلمها شامل درج عنوان فیلم، اصلاح برخی اطلاعات فیلم از جمله عوامل اصلی (بازیگران، کارگردان، تهیه کننده، نویسنده)، جوایز، تیزر و تصاویر فیلم، خلاصهٔ داستان، و ... میشود.
  - دسترسیهای مدیریت سیستم شامل مشاهدهٔ لیست تمام کاربران و نمرهدهیها و نظرات ثبتشدهٔ هر یک از آنها و حذف کاربران در صورت لزوم میشود.

# آخرین نسخهٔ نمودار ER:



لينك EER اصلاحشده:

https://drive.google.com/file/d/196pKPNzoHJAI14lYJpNd1yrduH55LMbc/view?usp=sharing

<u>F</u>	I <u>D</u>	FYR	TYR	NAME	LANG	DUR	GENRE	BUDG	PLOTL	REVEN	AGER	TSTMP	ISSER	DIRID
þ	k													fk

#### Season

<u>FID</u>	<u>SNUM</u>	EPCNT
fk	pk	
pk		

# TrailerLink

<u>FID</u>	<u>LINK</u>
fk	pk
pk	

PictureLink

<u>FID</u>	<u>LINK</u>
fk	pk
pk	
Country	
<u>FID</u>	<u>NAME</u>
fk	pk
pk	

UID	NAME	MAIL	PASS	PHONE	UTYPE	ELIMIT
pk						

#### Review

UID	<u>FID</u>	RATE	DESCL
fk	fk		
pk	pk		

# InterestedIn

<u>UID</u>	<u>FID</u>
fk	fk
pk	pk

Crew

CID	NAME	MAIL	PHONE	BYEAR	BMON	BDAY	DYEAR	DMON	DDAY	SEX	ISDIR	ISPRO	ISACT	ISWRT
pk														

FirstRole

<u>CID</u>	<u>FID</u>
fk	fk
pk	pk

Acts

CID	<u>FID</u>
fk	fk
pk	pk
·	·

Writes

CID	<u>FID</u>
fk	fk
pk	pk

Produces

CID	<u>FID</u>
fk	fk
pk	pk

#### Award

<u>TITLE</u>	<u>FEST</u>	YEAR	<u>FID</u>	<u>CID</u>
pk	pk		fk	fk
			pk	pk

```
create view Pro_Acc_Film
```

as select FYR, TYR, TSTMP, NAME, LANG, DUR, GENRE, PLOTL, AGER, ISSER, DIRID, SNUM, EPCNT, BUDG, REVEN, avg(RATE) As AVRATE from (Film left outer join Season on Film.fid = Season.fid) inner join Review on film.fid = review.fid group by Film.fid, snum, epcnt;

create view Free\_Acc\_Film

as select FYR, TYR, TSTMP, NAME, LANG, DUR, GENRE, PLOTL, AGER, ISSER, DIRID, SNUM, EPCNT, AVRATE from Pro\_Acc\_Film;

create view All\_Actors

as select NAME, FID

from Acts natural join Crew;

create view Male\_First\_Role

as select NAME, FID

from FirstRole natural join Crew

where Crew.SEX = 'M';

create view Female\_First\_Role
 as select NAME, FID

from FirstRole natural join Crew
 where Crew.SEX = 'F';

create view All\_Directors

as select NAME, FID

from Film natural join Crew;

create view All\_Writers
as select NAME, FID
from Writes natural join Crew;

create view All\_Producers
as select NAME, FID

```
from Produces natural join Crew;
create view All_Pictures
as select LINK, FID
from PictureLink natural join Film;
create view All_Trailers
as select LINK, FID
from TrailerLink natural join Film;
create view All_Country
as select Country.NAME, Film.FID
from Country inner join Film on Country.FID = Film.FID;
create view All_Award
       as select FID, TITLE, FEST, YEAR, Crew.NAME
       from Award inner join Crew on Crew.CID = Award.CID;
```

create view Film\_Reviews

as select Review.FID, Users.NAME, RATE, DESCL

from Review inner join Users on Review.UID = Users.UID;

create view User\_Reviews

as select Review.UID, NAME, RATE, DESCL

from Review inner join Film on Review.FID = Film.FID;

create view Favorite\_List

as select UID, Film.NAME

from Film inner join InterestedIn on Film.FID = InterestedIn.FID;

### توضيح ديدها:

در این بخش دیدهای تعریفشده روی جدولهای مبنایی پایگاهداده را به اختصار معرفی میکنیم.

- ۱) Pro\_Acc\_Film: اطلاعات کامل فیلمها و سریالهای مختلف را برای کاربر Pro در اختیار میگذارد که شامل میانگین نمرهٔ داده شده به هر فیلم یا سریال نیز هست.
- Y :Free\_Acc\_Film برای تعریف این دید از تکنیک دید روی دید استفاده شده و همهٔ اطلاعات قابل مشاهده در دید Pro\_Acc\_Film به جز تعداد سرمایهٔ تولید محصول :Free\_Acc\_Film به جز تعداد سرمایهٔ تولید محصول (BUDG) و فروش آن (REVEN) که مختص کاربران ویژه (Pro) است- در این دید قابل دسترسی است.

- All\_Actors (۳ نیگر نام او و کد فیلمهایی که در آنها بازی کرده را در سطرهایی جداگانه دارد تا برای بازیابی نام همهٔ بازیگران یک فیلم یا سریال استفاده شود.
- ۴) Male\_First\_Role: شامل لیست نام همهٔ بازیگران نقش اول مرد و کذ فیلم مربوطه است که از آن می توان برای بازیابی نام بازیگر نقش اول مرد (در صورت وجود) یک اثر از طریق داشتن کد آن استفاده کرد.
  - ∆ Female\_First\_Role: مشابه مورد قبلی برای نقش اول زن.
  - All\_Directors (۶ برای یافتن نام کارگردان هر اثر (البته توجه شود که بر خلاف موضوع بازیگران، هر اثر دقیقاً یک کارگردان دارد).
    - All\_Writers (۷ مشابه All\_Actors برای یافتن نویسندهٔ (های) یک اثر.
    - All\_Producers (های) یک اثر. All\_Actors مشابه All\_Actors برای یافتن تهیه کنندهٔ (های) یک اثر.
    - All\_Pictures (۹: برای نمایش لینک همهٔ تصاویر مربوط به یک اثر از طریق داشتن کد (FID) آن استفاده می شود.
      - All\_Trailers (۱۰ مشابه All\_Pictures برای نمایش لینک تریلرهای یک اثر.
      - All\_Country (۱۱: با استفاده از آن می توان نام تمام کشورهای دخیل در تولید یک اثر را با داشتن FID آن یافت.
- All\_Award (۱۲ یک فیلم می توان جزئیات همهٔ جوایزی را که هر یک از عوامل تولید آن فیلم (بازیگر، کارگردان، نویسنده، یا تهیه کننده) بردهاند، از روی این دید به دست آورد. این جزئیات شامل عنوان جایزه، نام جشنواره، سال برنده شدن، و نام فرد برنده است. توجه کنید که طبق طراحی اولیه هر جایزهٔ ممکن به یک فرد تعلق می گیرد؛ حتی جایزهای جمعی مثل بهترین فیلم به تهیهٔ کنندهٔ آن داده می شود.
- Film\_Reviews (۱۳: جزئیات تمام نظرات ثبتشده برای هر فیلم را از طریق داشتن FID آن، میتوان از این دید مشاهده کرد که شامل نام نظردهنده، لینک متن نظر او، و نمرهای می شود که به فیلم دادهاست.
  - User\_Reviews (۱۴: کارکردی مشابه Film\_Reviews دارد اما برای مشاهدهٔ جزئیات همهٔ نظراتی که یک کاربر دربارهٔ فیلمهای مختلف دادهاست، استفاده می شود.

Favorite\_List: لیست علاقمندیهای کاربران را نمایش میدهد که با داشتن کد کاربر (UID) میتوان نام همهٔ فیلمهای موردعلاقهٔ یک کاربر خاص را از طریق آن مشاهده کرد.

# نکاتی در ارتباط با پیادهسازی جداول:

- در تمامی جداولی که حاوی کلید خارجی بودهاند، محدودیت انتشار در حذف به صورتon update cascade on delete cascade در نظر گرفته شدهاست.
- در جداول Film, User و fid, uid و cid از نوع uuid-ossp هستند و توسط رهاناها در پایگاه داده مقداردهی می شوند. (از افزونهٔ fid, uid و cid و cid و rostgreSQL در Film, User استفاده شده است.)
  - صفت گذرواژه (pass) توسط تابع crypt موجود در افزونهٔ pgcrypto در PostgreSQL قبل از درج یا بهروزرسانی بهوسیله رهانا hash و سپس ذخیره می شود.
- به دلیل تعداد زیاد کشورها و زبانهای ممکن برای هر فیلم, برای دامنهٔ این صفات جداول country\_list و country\_list را که حاوی نام تمامی کشورها و زبانها است ساخته و هنگام درج یا بهروزرسانی کشور یا زبان برای هر فیلم به وسیله رهانا اعتبارشان را بررسی میکنیم.

#### دامنههای صفات:

دامنههای زیر در پروژه تعریف شدهاند.

```
۱) شماره تلفن معتبر (براساس استاندارد ۱۶۴E):
create domain phone number as text
                 check ( value ~ '^[+]*[(]{0,1}[0-9]{1,4}[)]{0,1}[-\s\./0-9]*$'
);
                                                                                                                                                                                                                                                                                                                                                  ۲) نشانی ایمیل معتبر:
create domain email address as text
                check ( value ~
'^(([^<>()\[\]\\.,;:\s@"]+(\.[^<>()\[\]\\.,;:\s@"]+)*)|(".+"))@((\
[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}])|(([a-zA-Z\-0-y]{1,3})]|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z])|)|(([a-zA-z
9]+\.)+[a-zA-Z]{2,}))$');
                                                                                                                                                                                                                                                                            ۳) نام کاربری معتبر (تنها حاوی حروف، ارقام، ے و -):
create domain user name as text
                 check ( value \sim '^{A-Za-z0-9}+(?:[ -][A-Za-z0-9]+)*$');
                                                                                                                                                                                                                                                                                                                                                                   ۴) لینک معتبر:
create domain link as text
                check ( value ~
'^(http:\/\/www\.|https:\/\/www\.|http:\/\/|https:\/\/)?[a-z0-
9]+([\-\.]{1}[a-z0-9]+)*\.[a-z]{2,5}(:[0-9]{1,5})?(\/.*)?$');
```

```
create domain valid password as text
Z]).{8,}$');
                                                            ۶) سال (چهار رقم بزرگتر از ۱۸۸۸ و کمتر از امسال ۲۰۲۱):
create domain valid year as smallint
    check (value >= 1888 and value <= date_part('year', now()));</pre>
                                                                             ٧) ماه (مخففهای سه حرفی):
create domain valid month as char(3)
   check (upper(value) in ('JAN', 'FEB', 'MAR', 'APR', 'MAY',
'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC'));
                                                                          ۸) روز (عدد طبیعی بین ۱ تا ۳۱):
create domain valid day as smallint
    check (value <= 31 and value >= 1);
                                                                          ۹) جنسیت (یک کاراکتر دوحالتی):
create domain sex as char(1)
    check (upper(value) in ('M', 'F'));
```

۵) گذرواژهٔ معتبر (شامل حداقل ۸ کاراکتر و حاوی رقم و حروف):

```
create domain user type as char(1) default 'F'
    check (upper(value) in ('P', 'F'));
                                                            ۱۱) دفعات تغییر اطلاعات یک فیلم برای یک کاربر عادی (عددی بین ۰ تا ۵):
create domain user_edit_number_limitation as smallint default 5
    check (value <= 5 and value >= 0);
                                                                                                   ۱۲) عدد طبيعي:
create domain normal number as smallint
    check (value > 0);
                                                     ۱۳), ده بندی سنی فیلم (بر اساس سیستم رتبه بندی انجمن Motion Picture):
create domain mpa film rating as varchar(5)
    check (value in ('G', 'GP', 'PG', 'PG-13', 'R', 'NC-17', 'X',
'M'));
                                                                       ۱۴) واحد یول (بر حسب میلیون دلار با دقت سه رقم اعشار):
create domain million dollar as numeric(10, 5)
    check (value >= 0.0);
                                                                            ۱۵) تاریخ (حاوی سال, ماه, روز, ساعت, دقیقه, ثانیه):
create domain modify_date as text
```

۱۰) نوع کاربر (یک کاراکتر دوحالتی):

```
check (value \sim '^\d{4}-\d{2}-\d{2}:\d{2}:\d{2}:\d{2}$');
                                                                     ۱۶) امتیاز (عددی با دقت یک رقم اعشار بین ۰ تا ۱۰):
create domain rate as numeric(3, 1)
    check (value \geq= 0.0 and value \leq= 10.0);
                                                                                              ١٧) نام فيلم:
create domain movie name as text
    check (value \sim '[a-zA-Z0-9 \cdot :\s@(())]*$');
                                                                                              ۱۸) ژانر فیلم:
create domain genre as text
    check (upper(value) in
('ACTION', 'ADVENTURE', 'COMEDY', 'DRAMA', 'CRIME', 'THRILLER',
'SCI-FI', 'HORROR', 'ROMANCE', 'SCIENCE FICTION', 'CRIME',
'WESTERN', 'MUSIC', 'FICTION', 'MYSTERY', 'DOCUMENTARY', 'EPIC',
'MATERIAL ARTS', 'SCIENCE', 'FANTASY', 'ANIMATION', 'SPORTS',
'BIOGRAPHICAL', 'EXPERIMENTAL', 'SILENT', 'SPY', 'COMEDY-DRAMA',
'MONSTER', 'ZOMBIE', 'CARTOON', 'GORE', 'HEIST', 'SLASHER'));
```

### محدودیتهای جامعیتی و رهانا:

در PostgreSQL هر رهانا بایستی در نهایت تابعی را صدا کند. لذا رهاناها و توابع مرتبط با محدودیتها به صورت زیر در پروژه درنظر گرفته شدهاند.

- ۱) هنگام درج هر عضو از عوامل ساخت فیلم (Crew) در جدول بایستی موارد زیر رعایت شود:
  - صفت نامش نباید null باشد.
  - باید حداقل یک شغل داشته باشد.
  - در صورت درج تاریخهای تولد و مرگ، نباید تاریخ مرگ از تاریخ تولد عقبتر باشد.

در هنگام درج یک Crew با استفاده از تابع uuid\_generate\_v3، یک uuid یکتا برای درج به عنوان صفت cid تولید می شود.

تابع مربوطه:

```
create or replace function crewConstructor() returns trigger as
$crew insert init$
begin
    if new.name is NULL then
       raise exception 'crew name cannot be empty';
    end if;
   if new.cid is null then
        new.cid := uuid generate v3(uuid ns oid(), new.name);
   end if;
    if new.isdir is false and new.ispro is false and new.isact is
false and new.iswrt is false then
       raise exception 'crew should have at least one job';
    end if;
    if not isCrewAgeValid(new.byear, new.bmon, new.bday,
new.dyear, new.dmon, new.dday) then
       raise exception 'death date is sooner than birth date';
```

```
end if;
    return new;
end;
$crew insert init$ language plpgsql;
                                                                                                    رهانای مربوطه:
create trigger insert_crew_initial
    before Insert
    on crew
    for each row
execute procedure crewConstructor();
           ۲) هنگام بهروزرسانی تاریخهای تولد و مرگ یک عضو از عوامل ساخت فیلم (Crew), بایستی همواره بررسی شود که سن مرگ از سن تولد عقبتر نباشد.
                                                                                                     توابع مربوطه:
create or replace function isCrewAgeValid(by valid year, bm
valid month, bd valid day, dy valid year, dm valid month, dd
valid_day) returns boolean as
$$
begin
    if by is not null and dy is not null and by < dy then
        return true;
    end if;
    if bm is not null and dm is not null and by = dy and
monthToInt(bm) < monthToInt(dm) then</pre>
        return true;
```

```
end if;
    if bd is not null and dd is not null and by = dy and
monthToInt(bm) = monthToInt(dm) and bd <= dd then</pre>
        return true;
    end if;
    if by is not null and bm is not null and bd is not null and dy
is null and dm is null and dd is null then
        return true;
    end if;
    return false;
end;
$$ language plpgsql;
create or replace function crewUpdateAge() returns trigger as
$crew_age_update$
begin
    if not isCrewAgeValid(new.byear, new.bmon, new.bday, new.dyear, new.dmon, new.dday) then
        raise exception 'death date is sooner than birth date';
    end if;
    return new;
end;
$crew age update$ language plpgsql;
                                                                                            رهانای مربوطه:
create trigger update crew age
    before update of byear, bmon, bday, dyear, dmon, dday
```

```
on crew
     for each row
execute procedure crewUpdateAge();
۳) هنگام بهروزرسانی صفتهای مرتبط با شغل برای یک عضو از عوامل ساخت فیلم (Crew), بایستی همواره بررسی شود که حالتی پیش نیاید که شغلی نداشته باشد.
                                                   (صفتهای isdir ,ispro ,isact ,iswrt نباید در یک لحظه همه با هم false باشند.)
                                                                                                         تابع مربوطه:
create or replace function crewUpdateJob() returns trigger as
$crew job update$
begin
     if new.isdir is false and new.ispro is false and new.isact is
false and new.iswrt is false then
         raise exception 'crew should have at least one job';
     end if;
     return new;
end;
$crew job update$ language plpgsql;
                                                                                                       رهانای مربوطه:
create trigger update_crew_job
     before update of isact, isdir, ispro, iswrt
     on crew
     for each row
execute procedure crewUpdateJob();
```

```
۴) صفت cid برای یک عضو از عوامل ساخت فیلم (Crew), یک صفت read-only بوده و هرگز بعد از تولید نمی تواند تغییر کند.
                                                                                                                   تابع مربوطه:
 create or replace function crewIDModifying() returns trigger as
 $user id changing$
begin
      raise exception 'you cannot modify crew ID';
end;
 $user id changing$ language plpgsql;
                                                                                                                رهانای مربوطه:
 create trigger update_crew_id
      before update of cid
      on crew
     for each row
 execute procedure crewIDModifying();
                                                                     ۵) هنگام درج هر کاربر (User) در جدول بایستی موارد زیر رعایت شود:
                                                                                                • صفت نامش نباید null باشد.
                                                                                             • صفت گذرواژهاش نباید null باشد.
در هنگام درج یک User با استفاده از تابع uuid_generate_v3، یک uuid یکتا برای درج به عنوان صفت uid تولید می شود. ضمناً صفت گذرواژه نیز طبق روش hash ،bf
                                                                                                   شده و بعد در جدول قرار می گیرد.
                                                                                                                  تابع مربوطه:
 create or replace function userConstructor() returns trigger as
 $insert user initial$
```

```
begin
    if new.name is NULL then
        raise exception 'username cannot be empty';
    end if;
    if new.pass is NULL then
        raise exception 'user password cannot be empty';
    end if;
    if new.uid is null then
        new.uid := uuid_generate_v3(uuid_ns_oid(), new.name);
    end if;
    new.pass := crypt(new.pass, gen_salt('bf'));
    return new;
end
$insert user initial$ language plpgsql;
                                                                                                  رهانای مربوطه:
create trigger insert_user_initial
    before Insert
    on users
    for each row
execute procedure userConstructor();
                                        ۷) صفت uid برای کاربر (User)، یک صفت read-only بوده و هرگز بعد از تولید نمی تواند تغییر کند.
                                                                                                    تابع مربوطه:
```

create or replace function userIDModifying() returns trigger as

```
$user id changing$
begin
    raise exception 'you cannot modify user ID';
end;
$user id changing$ language plpgsql;
                                                                                                  رهانای مربوطه:
create trigger update user id
    before update of uid
    on users
    for each row
execute procedure userIDModifying();
                 ۷) در هنگام بهروزرسانی صفت گذرواژهٔ یک کاربر (User), بایستی علاوه بر null نبودن آن، غیرتکراریبودن گذرواژهٔ واردشده هم بررسی شود.
                                                                                                    تابع مربوطه:
create or replace function updatePassword() returns trigger as
$user update password$
begin
    if new.pass is NULL then
        raise exception 'user password cannot be empty';
    end if;
    if old.pass = crypt(new.pass, old.pass) then
        raise exception 'new password is repetitive';
    end if;
    new.pass := crypt(new.pass, gen_salt('bf'));
    return new;
end;
```

```
$user update password$ language plpgsql;
                                                                                                   رهانای مربوطه:
create trigger update user pass
    before update of pass
    on users
    for each row
execute procedure updatePassword();
                           ۸) هنگام ارتقای نوع حساب کاربری یک کاربر (User) به نوع pro, می بایستی بررسی شود که کاربر قبلا pro نبوده و برعکس.
                                                                                                    تابع مربوطه:
create or replace function promoteDemoteUser() returns trigger as
$make_user_pro_free$
begin
    if upper(old.utype) ~ upper(new.utype) then
        raise exception 'redundant promotion/demotion process';
    end if;
    new.utype = upper(new.utype);
    return new;
end;
$make_user_pro_free$ language plpgsql;
                                                                                                   رهانای مربوطه:
create trigger promote demote user
    before update of utype
    on users
```

for each row
execute procedure promoteDemoteUser();

- ۹) هنگام درج هر فیلم (Film) در جدول بایستی موارد زیر رعایت شود:
- در صورت درج سال شروع ساخت و پایانش, سال شروع ساخت فیلم نباید جلوتر از سال پایان ساخت آن فیلم باشد.
  - بایستی بررسی شود که زبان محاوره در فیلم یک زبان معتبر باشد.
- صفت آخرین تاریخ ویرایش فیلم (tstmp) صفتی است که نباید مقداری برایش هنگام درج بیاید و read-only است.

در هنگام درج یک Film با استفاده از تابع uuid\_generate\_v3، یک uuid یکتا برای درج به عنوان صفت fid تولید میشود. ضمناً صفت tstmp نیز از روی زمان جاری در جدول ثبت میشود.

تابع مربوطه:

```
create or replace function filmConstructor() returns trigger as
$film insert init$
begin
    if not isFilmYearsValid(new.fyr, new.tyr) then
       raise exception 'the start year is more then finish year';
    end if;
   if not isFilmLanguageValid(new.lang) then
        raise exception 'the language doesnt exist';
    end if;
    if new.tstmp is not null then
       raise exception 'time stamp of film is read only field';
    end if:
    if new.fid is null then
        new.fid = uuid generate v1mc();
    end if;
   if new.isser is null then
       new.isser = false;
    end if;
    new.tstmp = to_char(current_timestamp, 'YYYY-MM-DD HH:MI:SS');
```

```
return new;
end;
$film insert init$ language plpgsql;
                                                                                                  رهانای مربوطه:
create trigger insert_film_initial
    before Insert
    on film
    for each row
execute procedure filmConstructor();
                   ۱۰) بعد از بهروزرسانی صفتهای یک فیلم (Film) بایستی صفت آخرین تاریخ ویرایش آن (tstmp), معادل با تاریخ کنونی بهروزرسانی شود.
                                                                                                     تابع مربوطه:
create or replace function filmUpdateTimeStamp() returns trigger
$film update timestamp$
begin
    if new.tstmp is not null and new.tstmp != old.tstmp then
        raise exception 'time stamp of film is read only field';
    end if;
    new.tstmp = to_char(current_timestamp, 'YYYY-MM-DD HH:MI:SS');
    return new;
end;
$film update timestamp$ language plpgsql;
```

رهانای مربوطه:

```
create trigger update_film_timestamp
    before update
    on film
    for each row
execute procedure filmUpdateTimeStamp();
                                           ۱۱) صفت fid برای هر فیلم (Film)، یک صفت read-only بوده و هرگز بعد از تولید نمی تواند تغییر کند.
                                                                                                            تابع مربوطه:
create or replace function filmIDModifying() returns trigger as
$user_id_changing$
begin
    raise exception 'you cannot modify film ID';
end;
$user id changing$ language plpgsql;
                                                                                                          رهانای مربوطه:
create trigger update_film_id
    before update of fid
    on film
    for each row
execute procedure filmIDModifying();
           ۱۲) هنگام بهروزرسانی سال شروع و پایان ساخت یک فیلم (Film), بایستی در صورت ثبت هر دو سال بررسی کنیم سال پایان از سال شروع عقب تر نباشد.
                                                                                                            توابع مربوطه:
```

```
create or replace function isFilmYearsValid(start year smallint,
finish_year smallint) returns boolean
    language plpgsql
    immutable as
$$
begin
    if (start_year is not null and finish_year is not null and
finish year < start year) then</pre>
        return false;
    end if;
    return true;
end;
$$;
create or replace function filmUpdateProductYears() returns
trigger as
$film update product year$
begin
    if not isFilmYearsValid(new.fyr, new.tyr) then
        raise exception 'the start year is more then finish year';
    end if;
    return new;
end;
$film_update_product_year$ language plpgsql;
create trigger update_film_fyr_tyr
```

رهانای مربوطه:

```
before update of fyr, tyr
    on film
    for each row
execute procedure filmUpdateProductYears();
۱۳) هنگام بهروزرسانی صفت زبان مربوط به یک فیلم (Film), بایستی همواره بررسی شود که این زبان یک زبان موجود و معتبر باشد. (به وسیلهٔ تطابق زبان ورودی با عناصر
                                                                                        جدول language list).
                                                                                                      توابع مربوطه:
create or replace function isFilmLanguageValid(lang varchar(64))
returns boolean
    language plpgsql
    immutable as
$$
begin
    if (lang is not null and
         not exists(select * from language_list where upper(id) =
upper(lang) or upper(value) = upper(lang))) then
         return false;
    end if;
    return true;
end;
$$;
create or replace function filmUpdateLanguage() returns trigger as
$film_update_language$
begin
```

```
if not isFilmLanguageValid(new.lang) then
         raise exception 'the language doesn't exist';
     end if;
     return new;
end;
$film update language$ language plpgsql;
                                                                                                      رهانای مربوطه:
create trigger update film language
     before update of lang
     on film
     for each row
execute procedure filmUpdateLanguage();
۱۴) هنگام درج یا بهروزرسانی صفت کشورهای سازندهٔ مربوط به یک فیلم (Film) در جدول Country, بایستی همواره بررسی شود که نام آن کشور, یک کشور موجود و معتبر
                                                              باشد. (از طریق تطابق کشور ورودی با عناصر جدول country list).
                                                                                                       توابع مربوطه:
create or replace function isFilmCountryValid(country varchar(64))
returns boolean
     language plpgsql
     immutable as
$$
begin
     if (country is not null and
         not exists(select * from country list where upper(id) =
upper(country) or upper(value) = upper(country))) then
         return false;
```

```
end if;
    return true;
end;
$$;
create or replace function filmAddUpdateCountry() returns trigger
as
$film_update_country$
begin
    if not isFilmCountryValid(new.name) then
        raise exception 'the country doesn't exist';
    end if;
    return new;
end;
$film update country$ language plpgsql;
                                                                                               رهانای مربوطه:
create trigger insert_update_film_country
    before insert or update
    on country
    for each row
execute procedure filmAddUpdateCountry();
                                                              - تابع کمکی برای تبدیل ماه سه حرفی به ماه عددی (۱ تا ۱۲):
create or replace function monthToInt(m valid month) returns
smallint
```

```
language plpgsql
    immutable as
$$
declare
    res smallint;
begin
    if m is null then
        raise exception 'null input field exception';
    end if;
    case upper(m)
        when 'JAN' then res = 1;
        when 'FEB' then res = 2;
        when 'MAR' then res = 3;
        when 'APR' then res = 4;
        when 'MAY' then res = 5;
        when 'JUN' then res = 6;
        when 'JUL' then res = 7;
        when 'AUG' then res = 8;
        when 'SEP' then res = 9;
        when 'OCT' then res = 10;
        when 'NOV' then res = 11;
        when 'DEC' then res = 12;
        else res = 0;
        end case;
    return res;
end;
$$;
```

```
create or replace function intToMonth(m integer) returns char(3)
    language plpgsql
    immutable as
$$
declare
    res char(3);
begin
   if m is null then
        raise exception 'null input field exception';
    end if;
    case m
        when 1 then res = 'JAN';
       when 2 then res = 'FEB';
        when 3 then res = 'MAR';
        when 4 then res = 'APR';
        when 5 then res = 'MAY';
        when 6 then res = 'JUN';
        when 7 then res = 'JUL';
        when 8 then res = 'AUG';
        when 9 then res = 'SEP';
        when 10 then res = 'OCT';
        when 11 then res = 'NOV';
        when 12 then res = 'DEC';
        else res = null;
        end case;
    return res;
end;
$$;
```

# تولید دادههای نمونه:

یک برنامه به زبان جاوا نوشته شده که کدهای آن در پوشهٔ Dataset\_Handling موجود است و هدف آن تولید نمونه داده هایی سازگار با محدودیت های جامعیتی و ساختار پایگاه دادهٔ الله به زبان جاوا نوشته شده که کدهای آن در پوشهٔ Dataset\_Handling موجود است و هدف آن تولید می شوند، نام برخی صفات مثل نام فیلمها و عوامل آنها از قالبهای مشخصی شبیه + film + مراحی شده بوده است که نکتهٔ مذکور خللی در این هدف (index یا یا actor + (index پیروی می کند. هرچند که هدف از تولید این داده ها صوفاً صحت سنجی عملکرد پایگاه دادهٔ طراحی شده بوده است که نکتهٔ مذکور خللی در این هدف وارد نمی کند. خروجی این برنامه در دو قالب، یکی به صورت فایل هایی sql جداگانه برای پر کردن هر یک از جدول ها و به نام هر یک از جدول ها، و دیگری در قالب یک فایل یکپارچه به نام الله علی عداگانهٔ هر یک از جدول هاست.

# نرمالتر سازي جداول:

در ابتدا قابل ذکر است که تمام صفات جداول تک مقداریاند، لذا شمای ما در فرم نرمال 1NF قرار دارد و همچنین کلید اصلی هر جدول خود کلید کاندید محسوب می شود.

حال جداول را بررسی می کنیم:

۱) جدول Users به صورت زیر است:

Users (UID, NAME, MAIL, PASS, PHONE, UTYPE, ELIMIT)

در این جدول UID کلید اصلی و NAME و MAIL کلید های کاندید محسوب میشوند. از NAME ،UID و NAME به علت کلید بودن به سایر صفات FD داریم و از صفات غیر کلید به سایر صفات FD وجود ندارد.

۲ جدول Crew به صورت زیر است:

Crew (CID, NAME, MAIL, PHONE, BYEAR, BMON, BDAY, DYEAR, DMON, DDAY, SEX, ISDIR, ISPRO, ISACT, ISWRT)

FD تامنی و MAIL کلید اصلی و MAIL کلید کاندید محسوب می شود. از UID و MAIL به علت کلید بودن به سایر صفات FD کلید اصلی و و از صفات غیر کلید به سایر صفات وجود ندارد.

۳) جدول Crew به صورت زیر است:

Film (FID, NAME, FYR, TYR, LANG, DUR, GENRE, BUDGE, PLOTL, REVEN, AGER, TSTMP, ISSER, DIRID)

در این جدول FID کلید اصلی و ترکیب FYR ، NAME و TYR کلید کاندید محسوب می شود و DIRID کلید خارجی برای صفت CID در جدول FYR ، NAME و FYR ، NAME و FYR ، NAME داریم و از صفات غیرکلید و زیرمجموعه محض از ترکیب FYR ، NAME و FYR ، NAME داریم و از صفات غیرکلید و زیرمجموعه محض از ترکیب TYR و TYR به سایر صفات FD وجود ندارد.

۴) جدول Season به صورت زیر است:

Season (FID, SNUM, EPCENT)

در این جدول ترکیب FID و SNUM کلید اصلی محسوب می شود و از این ترکیب به سایر صفات FD داریم (FID کلید خارجی برای صفت FID در جدول Film محسوب می شود).

جداول TrailerLink و PictureLink به صورت زير هستند:

TrailerLink (FID, LINK)

PictureLink (FID, LINK)

در این جداول ترکیب FID و LINK کلید اصلی محسوب می شود و در واقع تنها FD های بدیهی در این جداول وجود دارند (FID در این دو جدول کلید خارجی برای صفت FID در جدول Film محسوب می شود).

۵) جدول Country به صورت زیر است:

Country (FID, NAME)

در این جدول ترکیب FID و NAME کلید اصلی محسوب میشود و در واقع تنها FD های بدیهی در این جدول وجود دارد (FID کلید خارجی برای صفت FID در جدول FIm محسوب میشود).

جدول Review به صورت زیر است:

Review (UID, FID, RATE, DESCL)

در این جدول ترکیب FID و UID کلید اصلی محسوب می شود و از این ترکیب به سایر صفات FD هست و از صفات غیرکلید به سایر صفات FD وجود ندارد (FID کلید خارجی به صفت UID و USers محسوب می شود).

۶) جدول InterestedIn به صورت زیر است:

InterestedIn (UID, FID)

در این جدول ترکیب UID و UID کلید اصلی محسوب میشود و در واقع تنها FDهای بدیهی در این جدول وجود دارد (FID کلید خارجی به صفت FID در جدول FID در جدول USers و UID در جدول USers محسوب میشود).

Writes ، Acts ، FirstRole و Produces به صورت زير مي باشند:

FirstRole (CID, FID)

Acts (CID, FID)

Writes (CID, FID)

Produces (CID, FID)

در این جداول ترکیب CID و CID کلید اصلی محسوب میشوند و در واقع تنها FD های بدیهی در این جدول وجود دارد (در این جداول FID کلید خارجی به صفت CTD در جدول Crew و CTD کلید خارجی به صفت CTD در جدول CTD محسوب میشود).

به صورت زیر است:  $\Lambda$  جدول  $\Lambda$ 

Award (TITLE, FEST, YEAR, FID, CID)

در این جدول ترکیب FID ، FEST ، TITLE و CID کلید اصلی محسوب می شود و از این ترکیب به صفت YEAR وابستگی تابعی وجود دارد (در این جدول FID کلید خارجی به صفت FID در جدول Film و CID کلید خارجی به صفت CID در جدول Crew محسوب می شود).

در طراحی بالا تمام صفات تک مقداریاند در نتیجه شما در فرم 1NF است . هر صفت غیرکلید با کلید اصلی رابطهاش FD کامل دارد در نتیجه شما در فرم 2NF نیز قرار دارد و هم چنین هر صفت غیرکلید با کلید اصلی رابطهاش فقط FD بی واسطه دارد (معادلاً در هیچ کدام از روابط از یک صفت غیر کلید به صفت غیر کلید دیگری FD وجود ندارد)؛ در نتیجه شما در فرم 3NF قرار دارد. نهایتاً دترمینان هر FD مهم و کاهش ناپذیر آن کلید کاندید است؛ در نتیجه شما در فرم 3NF قرار دارد.

# نمودار وابستگیهای تابعی روابط طراحی در زیر آمدهاست:

