

توضیح پروژه

پروژه تعریف شده مربوط به مدیریت سامانه فیلم برای کاربران است. به طوری که هر کاربر میتواند با توجه به موجودی (credit) خود فیلمی را خریده (purchase) و یا اجاره (rent) کند. هر فیلم میتواند تعدادی بازیگر (cast) و یک کارگردان (director) داشته باشد. همچنین یک بازیگران و یا کارگردان ها به واسطه ی فیلم هایی که در آنها مشارکت کردند میتوانند جایزه دریافت کنند که در جدول awards ذخیره شده است. همچنین هر فیلم میتواند دو یا چند ژانر (genre) مختلف داشته باشد.

دیتا این پروژه از 250 فیلم برتر از IMDB گرفته شده است که با توجه به نیازمندی های پروژه preprocessing های لازم بر روی آن انجام شده است.

گزارش کوئری های زده شده

کوئری actors_crosstab_genre

برای هر بازیگر، ژانر های بازی کرده را نمایش میدهد. یک بازیگر میتواند فقط در فیلم هایی بازی کرده باشد که 3 ژانر مختلف داشته باشند. پس تنها 3 ستون برای این بازیگر دارای تعداد دفعات بازی کردن در آن ژانر آمده است. پس میتوان فهمید هر بازیگر در چه ژانر هایی و به چه تعداد بازی کرده است.

دستورات استفاده شده: crosstab, inner-join, with, group-by, cast

کوئری actors_rollup_genres

مانند کوئری قبل اما به view متفاوت به قضیه برای هر بازیگر و تعداد دفعات بازیگری برای هر ژانر آمده شده است. تفاوت این کوئری با کوئری قبل این است که این دستور قبل یک گزارش کامل و جامع حتی برای ژانر های بازی نکرده میدهد در صورتی که این کوئری فقط برای جفت بازیگر و ژانر های موجود گزارش میدهد. برای گزارش کامل تر میتوان از دستوری cube استفاده کرد که حاصل تقریباً مانند کوئری قبل میشود.

دستورات استفاده شده: concat, inner-join, with, sum, count, roll-up, order-by

کوئری genre_popularity

در این کوئری میزان معروفیت هر ژانر بر اساس تعداد کاربرانی که فیلمی از آن ژانر ها خریده یا اجاره کرده اند نمایش میدهد. به طوری که اسم ژانر، تعداد کاربرانی که فیلمی از آن ژانر گرفته اند و رتبه آن ژانر بخصوص نمایش داده شده است. خروجی بر اساس رتبه مرتب شده است.

دستورات استفاده شده: with, inner-join, union-all, count, dense-rank, group-by, order-by

توابع نوشته شده

تابع `awards_per_actor`

این تابع یک آرگومان به عنوان `threshold` برای فیلتر تعداد جوایز برگردانده شده گرفته و خروجی یک جدول بر اساس تعداد جوایز کسب شده توسط هر بازیگر است.

تابع `movies_by_actor`

این تابع دو آرگومان به عنوان `first_name` و `last_name` گرفته و یک جدول حاوی اسم فیلم، نویسنده آن، تاریخ انتشار و موضوع کلی فیلم هایی که بازیگر مورد نظر در آنها بازی کرده است را برگرداند.

تابع `rents_between`

این تابع دو تاریخ به عنوان بازه های شروع و پایان `rents` که نیاز داریم گرفته و `rent` هایی که بین این دو تاریخ ثبت شده اند را خروجی بر میگرداند (خروجی از نوع جدول است)

آغازگر های نوشته شده

آغازگر `check_age_trigger`

این آغازگر چک میکند که آیا فیلم یا سریالی که میخواهیم بخریم در بازه سنی ما است یا خیر، به طوری که اگر نبود پیام مناسب چاپ خروجی داده شود. در غیر این صورت میتوانیم یک خرید ثبت کنیم.

آغازگر `check_runtime_trigger`

این آغازگر مدت زمان پخش یک فیلم یا سریال را بررسی میکند و اگر بیش تر از یک `threshold` بود مانع `insert` شدن آن در جدول میشود، در غیر این صورت به جدول اضافه میشود.

آغازگر `permit_to_send`

این آغازگر چک میکند در صورت تمایل یک کاربر برای انتقال شارژ به کاربر دیگر، دارای میزان شارژ مورد نیاز باشد.

رویه های نوشته شده

رویه `add_purchase`

این رویه ابتدا پول از حساب کاربر کم کرده و سپس یک خرید برای کاربر در جدول مربوطه ثبت میکند.

رویه `add_rent`

این رویه ابتدا به ازای نصف هزینه یک فیلم از کاربر کم کرده و سپس یک اجاره برای آن کاربر و فیلم مورد نیاز در جدول مربوطه ثبت میکند.

رویه `send_gift`

این رویه اجازه میدهد یک کاربر برای کاربر دیگر خریدی انجام دهد، به طوری که از حساب خریدار مبلغ کسر شده و برای کاربر مهمان یک فیلم اضافه میشود.

رویه `send_money`

با دریافت سه آرگومان کاربر فرستنده، کاربر گیرنده و مقدار انتقال، پولی از کاربر به کاربر دیگر انتقال میدهد.

