آرمان سمیعی

۹۵۳۱۰۳۹

گزارش پروژه آزمایشگاه پایگاه داده

یک نقش مدیریتی به نام game_manager وجود داردکه وظیفه کنترل پایگاه داده game را بر عهده دارد. هر کاربری که بخواهد بازی کند باید ابتدا نقشی برای او توسط game_manager ایجاد شود.مثلا فرض کنید کاربر arman بخواهد بازی کند در اینصورت کاربر game_manager ابتدا login میکند و کد زیر را اجرا میکند.

```
create role arman
login
password 'arman';
```

حال کاربر arman با رمز 'arman' وارد پایگاه داده game میشود.برای آنکه بتواند کاری کند ابتدا باید در جدول players اطلاعات خود را وارد کند.این جدول دارای فیلدهای

```
player_id serial primary key,
first_name varchar(30),
last_name varchar(30),
continent varchar(50),
country varchar(50),
city varchar(50),
phone varchar(11),
score int default 0,
role_name varchar(20)
```

میباشد که game_manager تنها اجازه وارد کردن اطلاعات first_name و last_name و continent و country و city و phone را به او میدهد. مابقی اطلاعات به صورت اتوماتیک با استفاده از trigger پر میشوند.هیچ کاربری اجازه آپدیت کردن این جدول را ندارد تا جلوی تقلب گرفته شود.

رویه ها:

رویه words_init() :

این رویه وظیفه پر کردن جدول words را دارد تا کلمات مسابقه در جدول words وارد شوند.

رویه bycotted_countries_init() :

این رویه وظیف دارد تا کشورهای تحریمی ابتدایی را وارد جدول boycotted_countries کند.

رویه reset_matches() :

این رویه وظیفه دارد تا تا جدول matches را reset کند.از آنجا که جدول matches تنها باید یک سطر داشته باشد.

در ابتدای راه اندازی پایگاه داده و پس از انجام هر مسابقه تنها سطر این جدول حذف و یک سطر خالی به آن اضافه میشود و اجازه هر گونه تغییر در این جدول به جز select از کاربران گرفته میشود.

رویه grant_update_to_user1(user_role_name name) :

این رویه وظیفه دارد وقتی یک کاربر شناسه خود را به عنوان بازیکن اول در جدول matches قرار داد اجازه آپدیت کردن کلمات بازیکن اول را به آن کاربر بددهد تا بتواند به عنوان نفر اول بازی کند.همچنین اجازه تغییر شناسه first_player_id تا اتمام مسابقه از تمام کاربران گرفته میشود.

رویه grant_update_to_user2(user_role_name name) :

مانند رویه قبل است برای بازی کردن به عنوان بازیکن دوم.

رویه revoke_update_from_user1(user_role_name name) :

پس از پایان بازی دیگر بازیکنان نباید اجازه آپدیت کردن کلمات جدول matches را داشته باشند. این رویه پس از پایان بازی برای همین مورد فراخوانی میشود.

رویه revoke_update_from_user2(user_role_name name)

مانند رویه قبل برای بازپس گیری اجازه آپدیت کردن کلمات از بازیکن دوم به کار میرود.

View ها:

CREATE VIEW rankings :

این view برای آن به وجود آمده تا اطلاعات غیر شخصی از جدول players را به بازیکنان نشان دهد.چون جدول Players دارای چند فیلد شخصی است نباید اجازه دیدن تمام ستونهای آن را به همه بازیکنان داد. برای همین برای نشان دادن ستونهای عمومی از این view استفاده میشود.

توابع:

public.get_rankings_based_on_location() :

این تابع برای گزارش گیری ساخته شده.

به کمک آن میتوان مشاهده کرد که کدام قاره بیشترین امتیاز را دارد یا کدام کشور از کدام قاره بیشترین امتیاز را دارد و... .

از دستور rollup برای این منظور استفاده شده.

public.get_matches_based_on_time() :

این تابع مشخص میکند بازیهایی را که در سالها و ماه ها و روزهای مختلف انجام شده.

public.top3() :

این تابع مشخص میکند کدام ۳ بازیکن بیشترین امتیاز را دارند.

public.one_hundred_club_countries() :

این تابع مشخص میکند کدام کشورها از مرز ۱۰۰ امتیاز عبور کرده اند.

how_many_times_win(player_identity int) :

این تابع شناسه یک بازیکن را میگیرد و مشخص میکند آن بازیکن چند بازی را برده است.

public.number_of_players_from_same_country() :

این تابع تعداد بازیکنان را به تفکیک هر کشور نشان میدهد.

public.score_calc() :

این تابع وظیفه محاسبه امتیازات برای دو بازیکن پس از اتمام مسابقه را دارد.اگر یکی تعداد کلمات بیشتری را درست وارد کرد ۳ امتیاز میگیرد و دیگری ۰ امتیاز. اگر هردو بازیکن تعداد کلمات یکسانی را درست وارد کردند هردو ۱ امتیاز میگیرند و فیلد score در جدول players برای دو بازیکن آپدیت میشود. همچنین این تابع یک مقدار برمیگرداند تا تابعی که ان را فراخوانی کرده با استفاده از آن جدول matches_history را پر کند.

آغازگرها:

game_ended_trigger:

این آغازگر وظیف دارد تا در پایان مسابقه کارهای لازم را انجام دهد.

reset_matches() را فراخوانی کند.

دوباره اجازه آپدیت کردن first_player_id وsecond_player_id را به بازیکنان بدهد تا دوباره همه بازیکنان بتوانند در مسابقه شرکت کنند.

در جدول matches_history یک سطر مربوط به همین مسابقه اخیر اضافه کند.

role_name_setter:

این آغازگر وظیفه دارد پس از وارد کردن اطلاعات توسط کاربر، ستون role_name را مقداردهی کند.

check_matches_first_player_validity_trigger:

این آغازگر وظیفه دارد پس از آپدیت شدن فیلد first_player_id مشخص کند که آیا کاربری که این فیلد را تغییر داده شناسه خود را وارد کرده یا شناسه شخص دیگری را وارد کرده.

یا اگر قبلا همین شناسه را به عنوان second_player_id وارد کرده دیگر نتواند دوباره همان را در first_player_id وارد کند.

همچنین با فراخوانی رویه grant_update_to_user1(current_user) باعث میشود تا اجازه واردکردن کلمات را در first_player_words به دست آورد. همچنین اجازه تغییر second_player_id از این بازیکن گرفته میشود تا دیگر نتواند با استفاده از یک شناسه دیگر خود هم به عنوان بازیکن دوم در مسابقه شرکت کند.

check_matches_second_player_validity_trigger:

همان نقش آغازگر قبلی را برای بازیکن دوم دارد.

revoke_from_first_player_end_of_game_trigger:

این آغازگر کنترل میکند که اگر بازیکن اول کلمه آخر خود را وارد کرد دیگر نتواند در جدول matches تغییر ایجاد کند و منتظر بررسی امتیازاتش شود.

revoke_from_second_player_end_of_game_trigger:

همان نقش آغازگر قبلی را برای بازیکن دوم دارد.

bycotted_countries_check_trigger

این آغازگر هنگام ورود اطلاعات توسط کاربر بررسی میکند که کشور او جزو کشورهای تحریمی نباشد.