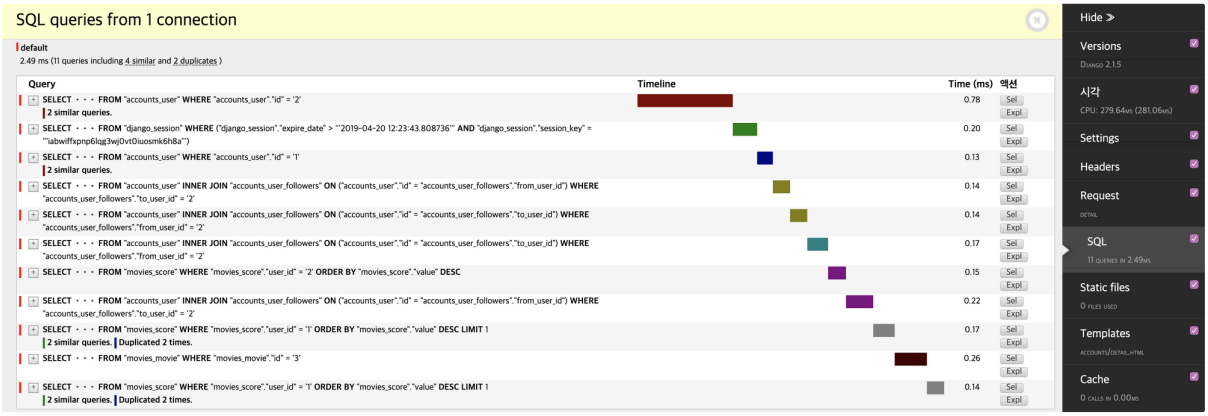
Improve your query

1. 기본 사항(11 Query, 4 similar and 2 duplicates)




```python
def detail(request, user_pk): User = get_user_model() user =
get_object_or_404(User, pk=user_pk) context = {'user_info': user}
return render(request, 'accounts/detail.html', context) def
follow(request, user_pk): User = get_user_model() target_user =
get_object_or_404(User, pk=user_pk) if request.user in
target_user.followers.all():
target_user.followers.remove(request.user) else:
target_user.followers.add(request.user) return
redirect('accounts:detail', user_pk)
```

Python

```
{% extends 'base.html' %} {% block body %} <p>팔로잉 : {{
user_info.followings.all.count }}</p> <p>팔로워 : {{
user_info.followers.all.count }}</p> {% if user != user_info %} {% if
user in user_info.followers.all %} <a href="{% url 'accounts:follow'
user_info.pk %}">언팔!</a> {% else %} <a href="{% url
'accounts:follow' user_info.pk %}">팔로우!</a> {% endif %} {% endif %}
<hr> <p>내가 작성한 리뷰 목록</p> {% for score in user_info.score_set.all
%} <p>영화명 : {{ score.movie.title }}</p> <p> 평점 : {{ score.value }}
</p> <hr> {% endfor %} <hr> {% for follow in
user_info.followings.all %} <p>{{ follow.username }}의 최고 평점</p>
<p>{{ follow.score_set.first.movie.title }}</p> <p>{{
follow.score_set.first.value }}</p> <hr> {% endfor %} {% endblock %}
```

HTML

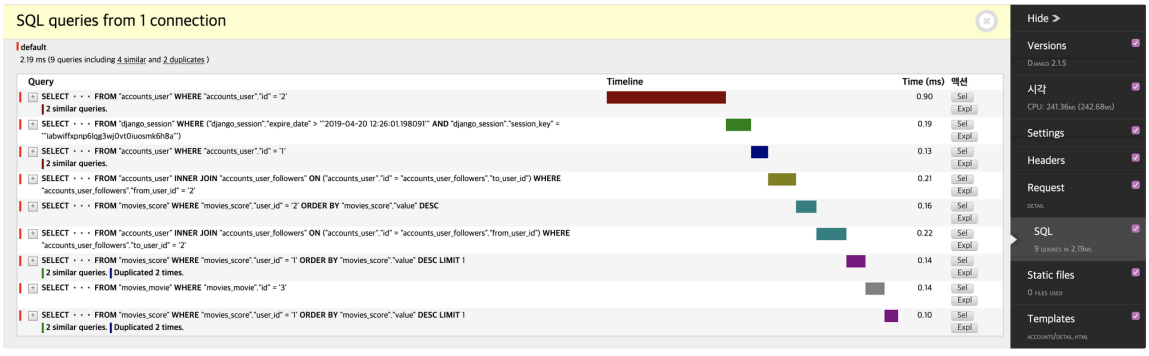
2-1. annotate

▶ 단순 `count`, `max` 값을 위해서라면 DB에서 계산해서 가져오자.

```
from django.db.models import Count def detail(request, user_pk):
User = get_user_model() user = get_object_or_404(
User.objects.annotate( followers_count=Count('followers'),
followings_count=Count('followings') ), pk=user_pk) context =
{'user_info': user} return render(request, 'accounts/detail.html',
context)
```

Python

▼ 결과



- 여기까지는 중복을 제거하지 않고 단순히 쿼리 갯수만 날린거! 근데 이 것보다 더 큰 문제는 `반복문을 도는 상황` 에서의 1:N, M:N 호출 관계에서이다.

2-2. prefetch_related

▼ 현재의 코드에서 중복되는 내용은 아래의 코드이다.

```
{% for follow in followings %} <p>{{ follow.username }}의 최고 평점
</p> <p>{{ follow.score_set.first.movie.title }}</p> <p>{{
follow.score_set.first.value }}</p> <hr> {% endfor %}
```

HTML

- 지금 follow 오브젝트가 정확히 동일한 쿼리인 `score_set` 을 두번 던지게 되는 것이 문제이다.
- 현재의 상황은 내가 팔로우 한 인원이 한명이어서 망정이지.. 사람이 한 명 늘어날 때마다 쿼리가 계속 추가될 것이다.

```
from django.db.models import Count, Prefetch def detail(request,
user_pk): User = get_user_model() user = get_object_or_404(
User.objects.annotate( followers_count=Count('followers'),
followings_count=Count('followings'), ), pk=user_pk) followings =
user.followings.prefetch_related( Prefetch( 'score_set',
queryset=Score.objects.order_by('-value'),
to_attr='score_set_value_order' )) context = {'user_info': user,
'followings': followings} return render(request,
'accounts/detail.html', context)
```

Python

```
{% for follow in followings %} <p>{{ follow.username }}의 최고 평점</p>
<p>{{ follow.score_set_value_order.0.movie.title }}</p> <p>{{
follow.score_set_value_order.0.value }}</p> <hr> {% endfor %}
```

HTML

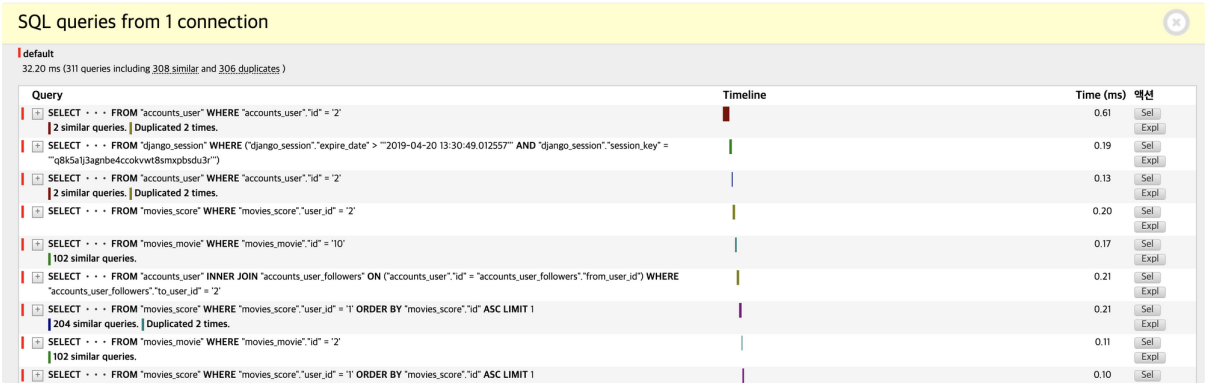
짠! 중복된 내용이 없단다. 이게 얼마나 큰 이슈냐고?

아래의 코드로 shell_plus에서 100명을 추가하고 평점 남기게끔하고 내가 팔로우를 하도록 하자.

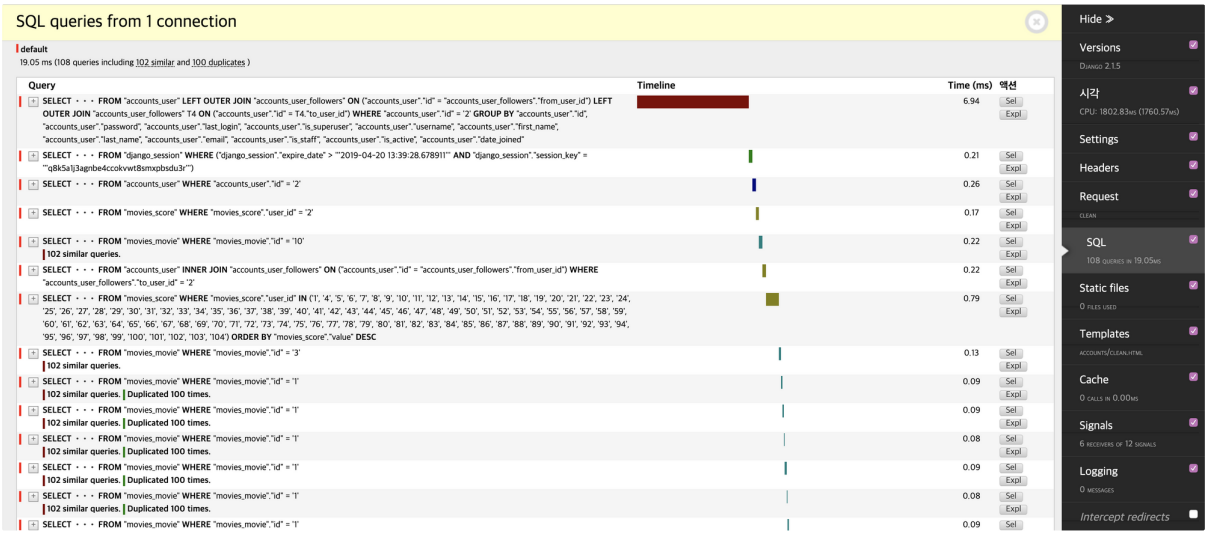
```python
me = User.objects.get(pk=2) for i in range(100): u = User.objects.create_user(f'user{i}', f'user{i}@gmail.com', 'govlgozld!') Score.objects.create( content='hi', value=3, movie_id=1, user=u ) u.followers.add(me)
```

Python

- 원래 코드대로 돌렸다면 이렇게 날아갔을 거다.
- 모든 팔로워들이 `score_set` 에 미친듯이 쿼리를 날리고 있다.



- 개선한 상황에서는 다음과 같다.



- 반토막 났다! 근데 또 중복이라고? 해결 가능하다.

2-3. select_related

▼ 현재의 코드에서 중복되는 내용은 아래의 코드이다.

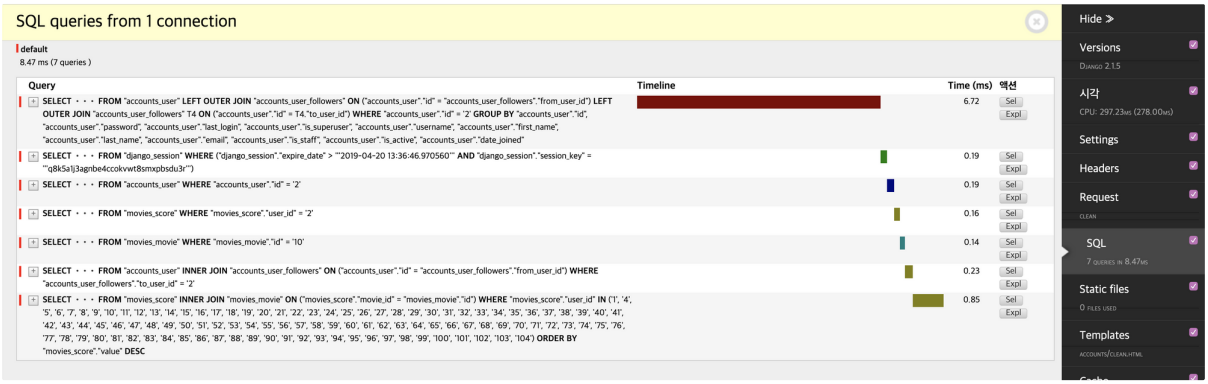
```
<p>{{ follow.score_set_value_order.0.movie.title }}</p>
```

Python

- 각각 스코어 셋의 첫번째가 해당하는 영화를 찾는 쿼리를 미친듯이 날리기 때문.
- 이때는 `select_related('movie')` 를 통해서 평점 정보 가져올 때 영화까지 함께 들고 오도록 하자!

```
followings = user.followings.prefetch_related( Prefetch(
'score_set',
queryset=Score.objects.select_related('movie').order_by('-value'),
to_attr='score_set_value_order' ))
```

Python



- 짠@_@