Its very simple we have to go to **BlogPost.php** and add

public function scopeLatest(Builder $query){

return $query->orderBy(static::CREATED\_AT,'desc');

}

In case of showing relations

One way is go to **PostsController.php**

return view('posts.show',['posts'=>BlogPost::with(['comment'=>function($query){

return $query->latest();

}])->findorFail($id)]);

Much easir way go to **BlogPost.php**

**public function comment()**

**{**

**return $this->hasMany('App\Comment')->latest();**

**}**

**Local Query Scope-Most Commented Posts**

BlogPost::mostCommented()->get()->pluck('comment\_count');

**Problem**

Illuminate/Database/QueryException with message 'SQLSTATE[42S22]: Column not found: 1054 Unknown column 'comments\_count' in 'order clause'

(SQL: select `blog\_posts`.\*, (select count(\*) from `comments` where `blog\_posts`.`id` = `comments`.`blog\_post\_id` and `comments`.`deleted\_at` is null) as `comment\_count` from `blog\_posts` where `blog\_posts`.`deleted\_at` is null order by `comments\_count` desc)'

**Problem solved**

BlogPost::mostCommented()->get()->pluck('comment\_count');

BlogPost.php

public function scopemostCommented(Builder $query){

//we will produce a field name comments count

// return $query->withCount('comment')->orderBy('comments\_count','desc');

return $query->withCount('comment')->orderBy('comment\_count', 'desc');

}

'comment\_count' the first letter you need to check the relation if relation name is comment and you write comments you will face the above error Illuminate/Database/QueryException with message 'SQLSTATE[42S22]: Column not found: 1054 Unknown column 'comments\_count' in 'order clause'

**In PostsController you will write**

**return view(**

**'posts.index',**

**[**

**'posts' => BlogPost::latest()->withCount('comment')->get(),**

**'most\_commented' => BlogPost::mostCommented()->take(5)->get(),**

**'mostActive' => User::WithMostBlogPosts()->take(5)->get(),**

**// 'mostActiveLastMonth' => User::withMostBlogPostsLastMonth()->take(5)->get(),**

**]**

**);**

**Users with most Blog posts in Last month**

**public function scopeWithMostBlogPostslastMonth(Builder $query)**

**{**

**{**

**return $query->withCount(['blogPost' => function (Builder $query) {**

**$query->whereBetween(static::CREATED\_AT, [now()->subMonths(1), now()]);**

**}])->having('blog\_post\_count', '>=', 2)**

**->orderBy('blog\_post\_count', 'desc');**

**}**

**Practical Global Query Scope**

**-Admin can see deleted posts**

**At first we will create a DeletedAdminScope.php inside scope folder**

**public static function boot()**

**{**

**// static::addGlobalScope(new LatestScope);**

**static::addGlobalScope(new DeletedAdminScope);**

**parent::boot();**

**static::deleting(function (BlogPost $blogPost) {**

**$blogPost->comment()->delete();**

**});**

**static::restoring(function (BlogPost $blogPost) {**

**$blogPost->comment()->restore();**

**});**

**}**

**If post is trashed than if we want that admin cant delete it again**