# Ccd LARAVEL WITH PHP. (MY PERSONAL NOTES)

* **LARAVEL BLADE:**

This is a php templating engine that makes it easy to embed php in laravel’s html codes. To use this engine, the file should be named as follow [filename.blade.php]. Thus the .blade extension must be in the file name.

**@yield(‘’), @extends(‘’) and @section(‘’)/@stop:**

This is used to refactor codes in laravel to reduce the bulkiness of the code. @yield(‘sectionName’) is used in the common file containing most of the html, to label a part that would be refered to from another file.

@extends(‘common file’s path and initial name’) is used to extend or include the common file in the page of choice.

@section(‘sectionName’)/@stop is used to embed a section that wants to be changed in the common file.

* **LARAVEL RAW SQL QUERY::**

**Insert ::** This is done in the route or controller section using PDO query format. Here a static DB::insert(); class is used. Example below:-

**DB::insert(“insert into post(title, content) values (?, ?)”, [“This is the title”, ”This is the content”]);**

**Read ::** This is done in the route or controller section using PDO query format. Here a static DB::select(); class is used. Example below:-

**$results = DB::select(“select \* from post where id = ?”, [2]);**

**return $results;**

**Update ::** This is done in the route or controller section using PDO query format. Here a static DB::update(); class is used. Example below:-

**$updated = DB::update(“update posts set title = ? where id = ?)”, [“Updated title”, 1]); return $updated;**

**Delete ::** This is done in the route or controller section using PDO query format. Here a static DB::delete(); class is used. Example below:-

**$delete = DB::delete(“delete from posts where id = ?)”, [1]);**

**return $delete;**

* **Eloquent [OBJECT RELATIONAL MAPPER] ::**

This allows us to deal with laravel database with ease.

**[How to create a model** :: **php artisan make:model Post -m] :-**

Here the flag[-m] creates a migration that’s related to the post model. The model created above is assumed to deal with the **posts** table in the data base (that is, the default model table is the model’s name in lowercase ending with an ‘s’).

The newly created model is an extention of the model class.

**MODEL PATH ::/app/.**

**[How to change default table name and primary key] :-**

The newly created model assume the name of the table as mentioned above.. it also assume that every table has and “id” column which is a primary key .. Hence the following code changes the default values..

**Protected $table = ‘RealNameOfTable’;**

**Protected $priamarykey = ‘postsid’;**

**[How to retrieve data from database using ELOQUENT] :-**

use App\Post

Route::get(‘/display’, function(){

$posts = Post::all();

OR

//$posts = Post::find(2);

foreach ($posts as $post){

return $post->title; }

**});**

**[How to retrieve more specific data from db using ELOQUENT]:-**

Route::get(‘/find’, function (){

$posts = Post::where(‘id’, 2)->orderBy(‘id’, ‘desc’)->take(1)->get();

return $posts;

});

**[How to retrieve data using ELOQUENT with exceptions]:-**