|  |  |
| --- | --- |
| Command | Description |
| $table->bigIncrements('id'); | Incrementing ID using a "big integer" equivalent. |
| $table->bigInteger('votes'); | BIGINT equivalent to the table |
| $table->binary('data'); | BLOB equivalent to the table |
| $table->boolean('confirmed'); | BOOLEAN equivalent to the table |
| $table->char('name', 4); | CHAR equivalent with a length |
| $table->date('created\_at'); | DATE equivalent to the table |
| $table->dateTime('created\_at'); | DATETIME equivalent to the table |
| $table->decimal('amount', 5, 2); | DECIMAL equivalent with a precision and scale |
| $table->double('column', 15, 8); | DOUBLE equivalent with precision |
| $table->enum('choices', array('foo', 'bar')); | ENUM equivalent to the table |
| $table->float('amount'); | FLOAT equivalent to the table |
| $table->increments('id'); | Incrementing ID to the table (primary key). |
| $table->integer('votes'); | INTEGER equivalent to the table |
| $table->longText('description'); | LONGTEXT equivalent to the table |
| $table->mediumInteger('numbers'); | MEDIUMINT equivalent to the table |
| $table->mediumText('description'); | MEDIUMTEXT equivalent to the table |
| $table->morphs('taggable'); | Adds INTEGER taggable\_id and STRING taggable\_type |
| $table->nullableTimestamps(); | Same as timestamps(), except allows NULLs |
| $table->smallInteger('votes'); | SMALLINT equivalent to the table |
| $table->tinyInteger('numbers'); | TINYINT equivalent to the table |
| $table->softDeletes(); | Adds **deleted\_at** column for soft deletes |
| $table->string('email'); | VARCHAR equivalent column |
| $table->string('name', 100); | VARCHAR equivalent with a length |
| $table->text('description'); | TEXT equivalent to the table |
| $table->time('sunrise'); | TIME equivalent to the table |
| $table->timestamp('added\_on'); | TIMESTAMP equivalent to the table |
| $table->timestamps(); | Adds **created\_at** and **updated\_at** columns |
| $table->rememberToken(); | Adds remember\_token as VARCHAR(100) NULL |
| ->nullable() | Designate that the column allows NULL values |
| ->default($value) | Declare a default value for a column |
| ->unsigned() | Set INTEGER to UNSIGNED |
| $table->primary('id'); | Adding a primary key |
| $table->primary(array('first', 'last')); | Adding composite keys |
| $table->unique('email'); | Adding a unique index |
| $table->index('state'); | Adding a basic index |
| $table->integer('user\_id')->unsigned();  $table->foreign('user\_id')  ->references('id')->on('users'); | Laravel also provides support for adding foreign key constraints to your tables: n this example, we are stating that the user\_id column references the id column on the users table. Make sure to create the foreign key column first! |
| $table->foreign('user\_id')  ->references('id')->on('users')  ->onDelete('cascade'); | ou may also specify options for the "on delete" and "on update" actions of the constraint: |
| $table->dropForeign('posts\_user\_id\_foreign'); | To drop a foreign key, you may use the dropForeign method. A similar naming convention is used for foreign keys as is used for other indexes: |
| $table->dropPrimary('users\_id\_primary'); | Dropping a primary key from the "users" table |
| $table->dropUnique('users\_email\_unique'); | Dropping a unique index from the "users" table |
| $table->dropIndex('geo\_state\_index'); | Dropping a basic index from the "geo" table |
| $table->dropTimestamps(); | Dropping the **created\_at** and **updated\_at** columns from the table |
| $table->dropSoftDeletes(); | Dropping **deleted\_at** column from the table |
| Schema::create('users', function($table)  {  $table->engine = 'InnoDB';  $table->string('email');  }); | To set the storage engine for a table, set the engine property on the schema builder: |