

Artesian Command	Uses
php artisan make:migration create_users_table	To Create Table
php artisan migrate	To Execute Migration
php artisan migrate --force	Force Execute Migration
php artisan migrate:rollback	Rolling Back Migration
php artisan migrate:rollback --step=5	Rolling Back Step 5
php artisan migrate:reset	Reset Migration
php artisan migrate:refresh	Reset And Execute Again
php artisan migrate:refresh --step=5	Rolling Back 5 and Execute Again
php artisan migrate:fresh	Drop All Tables & Migrate
Table Command	Uses
\$table->bigIncrements('id');	Auto-incrementing UNSIGNED BIGINT (primary key) equivalent column.
\$table->bigInteger('votes');	BIGINT equivalent column.
\$table->binary('data');	BLOB equivalent column.
\$table->boolean('confirmed');	BOOLEAN equivalent column.
\$table->char('name', 100);	CHAR equivalent column with a length.
\$table->date('created_at');	DATE equivalent column.
\$table->dateTime('created_at', 0);	DATETIME equivalent column with precision (total digits).
\$table->dateTimeTz('created_at', 0);	DATETIME (with timezone) equivalent column with precision (total digits).
\$table->decimal('amount', 8, 2);	DECIMAL equivalent column with precision (total digits) and scale (decimal digits).
\$table->double('amount', 8, 2);	DOUBLE equivalent column with precision (total digits) and scale (decimal digits).
\$table->enum('level', ['easy', 'hard']);	ENUM equivalent column.
\$table->float('amount', 8, 2);	FLOAT equivalent column with a precision (total digits) and scale (decimal digits).
\$table->geometry('positions');	GEOMETRY equivalent column.
\$table->geometryCollection('positions');	GEOMETRYCOLLECTION equivalent column.
\$table->increments('id');	Auto-incrementing UNSIGNED INTEGER (primary key) equivalent column.
\$table->integer('votes');	INTEGER equivalent column.
\$table->ipAddress('visitor');	IP address equivalent column.
\$table->json('options');	JSON equivalent column.
\$table->jsonb('options');	JSONB equivalent column.
\$table->lineString('positions');	LINestring equivalent column.
\$table->longText('description');	LONGTEXT equivalent column.
\$table->macAddress('device');	MAC address equivalent column.
\$table->mediumIncrements('id');	Auto-incrementing UNSIGNED MEDIUMINT (primary

	key) equivalent column.
<code>\$table->mediumInteger('votes');</code>	MEDIUMINT equivalent column.
<code>\$table->mediumText('description');</code>	MEDIUMTEXT equivalent column.
<code>\$table->multiLineString('positions');</code>	MULTILINESTRING equivalent column.
<code>\$table->multiPoint('positions');</code>	MULTIPOINT equivalent column.
<code>\$table->multiPolygon('positions');</code>	MULTIPOLYGON equivalent column.
<code>\$table->point('position');</code>	POINT equivalent column.
<code>\$table->polygon('positions');</code>	POLYGON equivalent column.
<code>\$table->smallInteger('votes');</code>	SMALLINT equivalent column.
<code>\$table->text('description');</code>	TEXT equivalent column.
<code>\$table->time('sunrise', 0);</code>	TIME equivalent column with precision (total digits).
<code>\$table->timeTz('sunrise', 0);</code>	TIME (with timezone) equivalent column with precision (total digits).
<code>\$table->timestamp('added_on', 0);</code>	TIMESTAMP equivalent column with precision (total digits).
<code>\$table->timestampTz('added_on', 0);</code>	TIMESTAMP (with timezone) equivalent column with precision (total digits).
<code>\$table->tinyIncrements('id');</code>	Auto-incrementing UNSIGNED TINYINT (primary key) equivalent column.
<code>\$table->tinyInteger('votes');</code>	TINYINT equivalent column.
<code>\$table->unsignedBigInteger('votes');</code>	UNSIGNED BIGINT equivalent column.
<code>\$table->unsignedDecimal('amount', 8, 2);</code>	UNSIGNED DECIMAL equivalent column with a precision (total digits) and scale (decimal digits).
<code>\$table->unsignedInteger('votes');</code>	UNSIGNED INTEGER equivalent column.
<code>\$table->unsignedMediumInteger('votes');</code>	UNSIGNED MEDIUMINT equivalent column.
<code>\$table->unsignedSmallInteger('votes');</code>	UNSIGNED SMALLINT equivalent column.
<code>\$table->unsignedTinyInteger('votes');</code>	UNSIGNED TINYINT equivalent column.
<code>\$table->uuid('id');</code>	UUID equivalent column.
<code>\$table->year('birth_year');</code>	YEAR equivalent column.
Column Modifier Command	Uses
<code>->after('column')</code>	Place the column "after" another column (MySQL)
<code>->autoIncrement()</code>	Set INTEGER columns as auto-increment (primary key)
<code>->charset('utf8')</code>	Specify a character set for the column (MySQL)
<code>->collation('utf8_unicode_ci')</code>	Specify a collation for the column (MySQL/PostgreSQL/SQL Server)
<code>->comment('my comment')</code>	Add a comment to a column (MySQL/PostgreSQL)
<code>->default(\$value)</code>	Specify a "default" value for the column
<code>->first()</code>	Place the column "first" in the table (MySQL)
<code>->nullable(\$value = true)</code>	Allows (by default) NULL values to be inserted into the column

->storedAs(\$expression)	Create a stored generated column (MySQL)
->unsigned()	Set INTEGER columns as UNSIGNED (MySQL)
->useCurrent()	Set TIMESTAMP columns to use CURRENT_TIMESTAMP as default value
->virtualAs(\$expression)	Create a virtual generated column (MySQL)
->generatedAs(\$expression)	Create an identity column with specified sequence options (PostgreSQL)
->always()	Defines the precedence of sequence values over input for an identity column (PostgreSQL)