Crear nuevo índice compuesto único, en una tabla con duplicados MySQL y Laravel Eloquent

Introducción

Una de las cosas que mas enamorado me tiene, son las migraciones de Laravel ya que te permite trabajar con software en producción, siendo un poco cuidadoso, por supuesto.

En el caso de este artículo, se trataba de añadir un índice compuesto, sobre una o varias tablas, que contenían cientos de miles de filas, y que por necesidad y eficacia, deberían haber tenido este índice compuesto único.

Que decir tiene, que la razón del mismo era, que al no tenerlo se habían producido duplicados no deseados, que fueron detectados en un bug abierto.

Procedimiento

- Crear un branch del proyecto
- **Crear un backup de mysql** del proyecto para trabajar con los datos en local.
- Generar la migración de actualización.
- Ejecutarla para ver el resultado.

Migración fallida

En el escenario normal escribimos una migración para añadir un índice compuesto en nuestra app Laravel

```
$ php artisan make:migration add_index_modem_plus_job_modems_table --table="logger_modems"
```

Donde escribiremos el índice compuesto que queremos añadir.

```
Schema::table('logger_modems', function (Blueprint $table) {
   $table->unique(['command_center_id', 'job']);
});
```

Pero al ejecutarlo ya sabemos lo que ocurrirá.

```
Migrating: 2022_04_23_124738_add_index_modem_plus_job_modems_table


   Illuminate\Database\QueryException


  SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry '8-2022-04-21
02:00:01' for key 'logger_modems.logger_modems_command_center_id_job_unique' (SQL: alter table
`logger_modems` add unique `logger_modems_command_center_id_job_unique`(`command_center_id`,
`job`))
```

Comprobación del problema

En Mysql (shell) o en nuestra app preferida (MySQLWorkbench, TablePlus,...) compobamos el problema

```
SELECT command_center_id, COUNT(command_center_id) job, COUNT(job)
FROM sitelight.logger_analyzers
GROUP BY command_center_id, job HAVING COUNT(*) > 1;
```

MysqlWorkbench - Duplicados

Modificación de la migración

La secuencia en formato SQL sería la de abajo, pero hemos dicho que queremos trabajar con eloquent, por muchos motivos.

```
CREATE TABLE tmp_data SELECT * FROM mytable;
TRUNCATE TABLE mytable;
ALTER TABLE mytable ADD UNIQUE INDEX myindex (A, B, C, D);
INSERT IGNORE INTO mytable SELECT * from tmp_data;
DROP TABLE tmp_data;
```

Así que aquí abajo lo adaptamos

```
public function up()
{
  DB::statement('CREATE TABLE sitelight.tmp_data SELECT * FROM sitelight.logger_modems');
  LoggerModem::query()->delete(); // TRUNCATE via eloquent
  Schema::table('logger_modems', function (Blueprint $table) {
      $table->unique(['command_center_id', 'job']);
  });
  DB::statement('INSERT IGNORE INTO sitelight.logger_modems SELECT * FROM
sitelight.tmp_data');
  DB::statement('DROP TABLE sitelight.tmp_data');
}
```

Agradecimientos

- Removing duplicates with unique index
- Find and remove duplicate rows by two columns

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Revision #1
Created 2022-04-23 17:40:48 UTC by Abkrim
Updated 2022-04-23 17:44:06 UTC by Abkrim