



**Centro de Electricidad Electrónica Y Telecomunicaciones (CEET)**

**Análisis y desarrollo en sistemas de información**

**Documento de instalación del aplicativo web**

**Ficha:**

2049891

**Aprendices:**

Andrés Felipe Sáenz Salazar

Lady Tatiana Chivito Caicedo

Andrés Esteban Sossa Rodríguez

Yesid Steven Valencia Rodríguez

**Instructor:**

Helberth Rodrigo Rojas Gacha

**Marzo 11 de 2021, Bogotá D.C**

## TABLA DE CONTENIDO

<b>¿A QUIÉN VA DIRIGIDO?</b> .....	<b>3</b>
<b>ALCANCE DEL DOCUMENTO</b> .....	<b>4</b>
<b>RESUMEN</b> .....	<b>5</b>
<b>PASOS PARA LA MIGRACION DE MySQL A PostgreSQL</b> .....	<b>6</b>
<b>PASO 1. Generar un respaldo o copia de seguridad</b> .....	<b>6</b>
<b>PASO 2. Traducir SQL de MySQL a SQL de PostgreSQL</b> .....	<b>8</b>
<b>PASO 3. Enviar los datos generados desde MySQL a PostgreSQL.</b> .....	<b>10</b>

## **¿A QUIÉN VA DIRIGIDO?**

El presente documento va dirigido al equipo técnico o quien tenga conocimientos en programación y experiencia en bases de datos. Este manual contiene información detallada del procedimiento que se debe realizar para la correcta migración y configuración de otro gestor de bases de datos como es PostgreSQL.

## **ALCANCE DEL DOCUMENTO**

El presente documento pretende definir una serie de pasos y/o actividades que permitirán especificar procesos para realizar la migración de datos, desde el sistema de gestión de bases de datos MySQL a PostgreSQL.

## RESUMEN

La migración de bases de datos consiste en la transferencia de información útil de un origen a un destino, para esto, se deben realizar una serie de pasos específicos, los cuales son:

- Identificar las fuentes de datos a migrar.
- Identificar las estructuras diseñadas para almacenar los datos.
- Observar el formato que tiene los datos, para identificar las conversiones necesarias para adaptarse a las nuevas necesidades
- Realizar pruebas previas a la migración con el fin de identificar las excepciones presentes propiamente en las estructuras y datos actuales que pudiesen surgir en el proceso de la migración
- Ejecución de la migración en su totalidad y su validación.

También se debe identificar en qué momento se va a realizar la migración. Después de esto tenemos que tener en cuenta los datos que se van a migrar. Para esto debemos de tener la planeación y preparación. Lo cual con lleva a verificar que tengamos una copia de la base de datos, analizar los datos y así evitar errores durante la migración de datos, por último, elegir la plataforma mas adecuada.

## PASOS PARA LA MIGRACIÓN DE MySQL A PostgreSQL

Para empezar, debemos definir ¿qué es PostgreSQL?

PostgreSQL es un gestor de bases de datos relacional open-Source, es considerado un motor de bases de datos más avanzado en la actualidad. Una característica de este gestor es el control de concurrencias multiversión; este método agrega una imagen del estado de la base de datos a consistentes, ofreciendo grandes ventajas en el rendimiento. También aporta mucha flexibilidad en nuestros proyectos. Por ejemplo, nos permite definir funciones personalizadas por medio de varios lenguajes de programación, como:

PL/ Java

PL/Ruby

PL/PHP

PL/Python

Ya que tenemos claro que es PostgreSQL, procedemos con el primer paso para la migración de la base de datos.

**PASO 1. Generar un respaldo o copia de seguridad** de la estructura de la base de datos de MySQL

Para esto debemos escribir la siguiente línea de código en la consola:

---

```
mysqldump -u [usuario] -p [base_de_datos] --no-data > [archivo_salida].sql
```

Donde:

- [usuario] es el nombre de usuario con el que ingresamos a la base de datos.
- [base\_de\_datos] es el nombre de la base de datos que vamos a migrar en este caso nuestra base de datos se llama effective.
- [archivo\_salida] es el nombre de nuestro archivo de salida que contendrá los comandos SQL generados por mysqldump.
- El parámetro -p es para que mysqldump nos pregunte por la contraseña del usuario. si no hay contraseña la base de datos entonces obviar este parámetro.
- El parámetro --no-data omitirá los datos, porque solo necesitamos la estructura de las tablas, una vez migrada la estructura seguimos con los datos mas adelante.

El comando se vería de la siguiente manera:

```
mysqldump -u root -p effective --no-data > effective-mysql.sql
```

este comando debe generar un archivo que se vería de la siguiente forma:

```
-- MySQL dump 10.13  Distrib 5.1.54, for debian-linux-gnu (i686)
--
-- Host: localhost    Database: world
--
-- Server version      5.1.54-1ubuntu4
--
-- Table structure for table `City`
--
DROP TABLE IF EXISTS `City`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `City` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `Name` char(35) NOT NULL DEFAULT '',
  `CountryCode` char(3) NOT NULL DEFAULT '',
  `District` char(20) NOT NULL DEFAULT '',
  `Population` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=4080 DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `Country`
--
```

```

DROP TABLE IF EXISTS `Country`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Country` (
  `Code` char(3) NOT NULL DEFAULT '',
  `Name` char(52) NOT NULL DEFAULT '',
  `Continent` enum('Asia','Europe','North America','Africa','Oceania','Antarctica','South America') NOT NULL DEFAULT 'Asia',
  `Region` char(26) NOT NULL DEFAULT '',
  `SurfaceArea` float(10,2) NOT NULL DEFAULT '0.00',
  `IndepYear` smallint(6) DEFAULT NULL,
  `Population` int(11) NOT NULL DEFAULT '0',
  `LifeExpectancy` float(3,1) DEFAULT NULL,
  `GNP` float(10,2) DEFAULT NULL,
  `GNPOld` float(10,2) DEFAULT NULL,
  `LocalName` char(45) NOT NULL DEFAULT '',
  `GovernmentForm` char(45) NOT NULL DEFAULT '',
  `HeadOfState` char(60) DEFAULT NULL,
  `Capital` int(11) DEFAULT NULL,
  `Code2` char(2) NOT NULL DEFAULT '',
  PRIMARY KEY (`Code`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `CountryLanguage`
--

DROP TABLE IF EXISTS `CountryLanguage`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `CountryLanguage` (
  `CountryCode` char(3) NOT NULL DEFAULT '',
  `Language` char(30) NOT NULL DEFAULT '',
  `IsOfficial` enum('T','F') NOT NULL DEFAULT 'F',
  `Percentage` float(4,1) NOT NULL DEFAULT '0.0',
  PRIMARY KEY (`CountryCode`,`Language`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

-- Dump completed on 2011-09-17 11:36:57

```

**PASO 2. Traducir SQL de MySQL a SQL de PostgreSQL** utilizando mysql2pgsql.perl.

mysql2pgsql.perl es un script escrito en Perl que «traduce» la sintaxis del código SQL de MySQL a un SQL que pueda entender PostgreSQL. Escribiremos la siguiente línea de código:

```
perl mysql2pgsql.perl [opciones] sql_formato_mysql.sql sql_formato_postgresql.sql
```

Para nuestro proyecto la línea que debemos escribir es:

`perl mysql2pgsql.perl --nodrop effective --mysql.sql effective-postgresql.sql`



```

) ;
/*!40101 SET character_set_client = @saved_cs_client */;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;

--
-- Table structure for table CountryLanguage
--

CREATE TABLE "countrylanguage" (
  "countrycode"  char(3) NOT NULL DEFAULT '',
  "language"     char(30) NOT NULL DEFAULT '',
  "isofficial"   varchar CHECK ("isofficial" IN ( 'T
', 'F' )) NOT NULL DEFAULT 'F',
  "percentage"   double precision NOT NULL DEFAULT '0.0',
  primary key ("countrycode", "language")
) ;

```

---

la opción `--nodrop` nos permite que el script no incluya los comandos `DROP TABLE` antes de crear las tablas.

<http://pgfoundry.org/projects/mysql2pgsql/> este link es para descartar el archivo `mysql2pgsql.perl`

el archivo que nos quedaría se vería de la siguiente manera:

```

-- MySQL dump 10.13  Distrib 5.1.54, for debian-linux-gnu (i686)
--
-- Host: localhost    Database: world
-- -----
-- Server version      5.1.54-1ubuntu4
--
-- Dumping data for table `City`
--
INSERT INTO `City` (`ID`, `Name`, `CountryCode`, `District`, `Population`)
VALUES (1, 'Kabul', 'AFG', 'Kabol', 1780000),
       (2, 'Qandahar', 'AFG', 'Qandahar', 237500),
       (3, 'Herat', 'AFG', 'Herat', 186800)
...

--
-- Dumping data for table `Country`
--

```

```
-- please see http:// archives.postgresql.org/pgsql-novice/2004-10/msg00158.php

-- #####

-- MySQL dump 10.13  Distrib 5.1.54, for debian-linux-gnu (i686)
--
-- Host: localhost    Database: world
-- -----
-- Server version      5.1.54-1ubuntu4

--
-- Table structure for table City
--

DROP SEQUENCE "city_id_seq" CASCADE ;

CREATE SEQUENCE "city_id_seq"  START WITH 4080 ;

CREATE TABLE "city" (
    "id" integer DEFAULT nextval('"city_id_seq"') NOT NULL,
    "name"   char(35) NOT NULL DEFAULT '',
    "countrycode"   char(3) NOT NULL DEFAULT '',
    "district"   char(20) NOT NULL DEFAULT '',
    "population"  int NOT NULL DEFAULT '0',
    primary key ("id")
) ;

/*!40101 SET character_set_client = @saved_cs_client */;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;

--
-- Table structure for table Country
--

CREATE TABLE "country" (
    "code"   char(3) NOT NULL DEFAULT '',
    "name"   char(52) NOT NULL DEFAULT '',
    "continent" varchar CHECK ("continent" IN ( 'Asia','Europe','North
America','Africa','Oceania','Antarctica','South America' )) NOT NULL DEFAULT 'Asia',
    "region"   char(26) NOT NULL DEFAULT '',
    "surfacearea" double precision NOT NULL DEFAULT '0.00',
    "indepyear" smallint DEFAULT NULL,
    "population"  int NOT NULL DEFAULT '0',
    "lifeexpectancy" double precision DEFAULT NULL,
    "gnp" double precision DEFAULT NULL,
    "gnpold" double precision DEFAULT NULL,
    "localname"   char(45) NOT NULL DEFAULT '',
    "governmentform" char(45) NOT NULL DEFAULT '',
    "headofstate" char(60) DEFAULT NULL,
    "capital"    int DEFAULT NULL,
    "code2"   char(2) NOT NULL DEFAULT '',
    primary key ("code")
) ;
```

### **Paso 3. Enviar los datos generados desde MySQL a PostgreSQL.**

Ahora debemos colocar el contenido del archivo generado hacia postgresql. Para ello nos logueamos con el usuario postgres y ejecutamos los comandos necesarios:

1. `$su postgres`
- 2.
3. `createdb effective`
- 4.
5. `psql -f effective_postgresql.sql -u usuario-db word`

Ahora solo nos queda migrar los datos. Al igual que en el paso 1, tenemos que sacar un respaldo pero esta vez solamente de los datos sin la creación de la estructura de las tablas, para esto vamos a escribir el siguiente comando:

```
mysqldump -u root -p effective --no-create-info --complete-insert --skip-add-locks  
> effective-mysql-data.sql
```

Donde:

- `-no-create-info` omite los scripts de creación de tabla (no las necesitamos)
- `-complete-insert` Hace que los comandos INSERT contenga también los nombres de las columnas. Ya que PostgreSQL no permite comandos insert sin los nombre de las columnas.
- `-skip-add-locks` omite el comando LOCK TABLE (este comando también es diferentes en PostgreSQL).

Como resultado tendremos el archivo de effective-mysql-data.sql va a ser algo parecido a como lo que vamos a mostrar a continuación:

```
INSERT INTO `Country` ( `Code`, `Name`, `Continent`, `Region`, `SurfaceArea`, `IndepYear`,
                        `Population`, `LifeExpectancy`, `GNP`, `GNPOld`, `LocalName`,
                        `GovernmentForm`,
                        `HeadOfState`, `Capital`, `Code2`)
VALUES ('ABW','Aruba','North America','Caribbean',193.00,NULL,103000,78.4,828.00,793.00,
        'Aruba','Nonmetropolitan Territory of The Netherlands','Beatrix',129,'AW'),
        ('AFG','Afghanistan','Asia','Southern and Central Asia',652090.00,1919,22720000,
        45.9,5976.00,NULL,'Afganistan/Afqanestan','Islamic Emirate','Mohammad Omar',1,'AF'),
        ('AGO','Angola','Africa','Central
Africa',1246700.00,1975,12878000,38.3,6648.00,7984.00,
        'Angola','Republic','José Eduardo dos Santos',56,'AO'),
        ('AIA','Anguilla','North America','Caribbean',96.00,NULL,8000,76.1,63.20,NULL,
        'Anguilla','Dependent Territory of the UK','Elisabeth II',62,'AI'),
        ('ALB','Albania','Europe','Southern Europe',28748.00,1912,3401200,71.6,3205.00,2500.00,
        'Shqipëria','Republic','Rexhep Mejdani',34,'AL'),
        ...

--
-- Dumping data for table `CountryLanguage`
--

INSERT INTO `CountryLanguage` (`CountryCode`, `Language`, `IsOfficial`, `Percentage`)
VALUES ('ABW','Dutch','T',5.3),
        ('ABW','English','F',9.5),
        ('ABW','Papiamentu','F',76.7),
```

Para no tener problemas con el carácter ' , vamos a reemplazarlo con el siguiente comando:

1. `sed -i 's/"/"/g' effective-mysql-data.sql`

el resultado es el siguiente:

```
-- MySQL dump 10.13  Distrib 5.1.54, for debian-linux-gnu (i686)
--
-- Host: localhost    Database: world
--
-----
-- Server version      5.1.54-1ubuntu4
--
-- Dumping data for table "City"
--

INSERT INTO "City" ("ID", "Name", "CountryCode", "District", "Population")
VALUES (1,'Kabul','AFG','Kabul',1780000),
       (2,'Qandahar','AFG','Qandahar',237500),
       (3,'Herat','AFG','Herat',186800)
...

--
-- Dumping data for table "Country"
--

INSERT INTO "Country" ( "Code", "Name", "Continent", "Region", "SurfaceArea", "IndepYear",
                        "Population", "LifeExpectancy", "GNP", "GNPOld", "LocalName",
                        "GovernmentForm",
                        "HeadOfState", "Capital", "Code2")
VALUES ('ABW','Aruba','North America','Caribbean',193.00,NULL,103000,78.4,828.00,793.00,
        'Aruba','Nonmetropolitan Territory of The Netherlands','Beatrix',129,'AW'),

        ('AFG','Afghanistan','Asia','Southern and Central Asia',652090.00,1919,22720000,
        45.9,5976.00,NULL,'Afganistan/Afqanestan','Islamic Emirate','Mohammad Omar',1,'AF'),
        ('AGO','Angola','Africa','Central
Africa',1246700.00,1975,12878000,38.3,6648.00,7984.00,
        'Angola','Republic','José Eduardo dos Santos',56,'AO'),
        ('AIA','Anguilla','North America','Caribbean',96.00,NULL,8000,76.1,63.20,NULL,
        'Anguilla','Dependent Territory of the UK','Elisabeth II',62,'AI'),
        ('ALB','Albania','Europe','Southern Europe',28748.00,1912,3401200,71.6,3205.00,2500.00,
        'Shqipëria','Republic','Rexhep Mejdani',34,'AL'),
...

--
-- Dumping data for table "CountryLanguage"
--

INSERT INTO "CountryLanguage" ("CountryCode", "Language", "IsOfficial", "Percentage")
VALUES ('ABW','Dutch','T',5.3),
       ('ABW','English','F',9.5),
       ('ABW','Papiamentu','F',76.7),
       ('ABW','Spanish','F',7.4)
...

-- Dump completed on 2011-09-18 17:41:21
```

Una vez que tengamos el archivo listo, lo colocamos en postgresql

```
1.  $su postgres
2.
3.  psql -f world_mysql-data.sql -u usuario-db word
```

Con esto finalizamos la migración de base de datos de MySQL a PostgreSQL.