

DESERCIÓN EN EDUCACIÓN MEDIA PARA LA REGIÓN CARIBE EN EL AÑO 2021

John Garcia¹, July Pedroza²,

¹⁻²Ingeniería y Ciencias Básicas, ³Dpto. de Ingeniería,

Universidad Central

Maestría en Analítica de Datos

Curso de Bases de Datos

Bogotá, Colombia

{¹*JohnDavidGarcia*,²*JulyPedroza*}*jgarcia9@ucentral.edu.co*,³*jpedrozac@ucentral.edu.co*

November 12, 2022

Contents

1	Introducción	3
2	Características del proyecto de investigación	3
2.1	Titulo del proyecto de investigación	3
2.2	Objetivo general	3
2.2.1	Objetivos especificos	3
2.3	Alcance	4
2.4	Pregunta de investigación	4
2.5	Hipotesis	4
3	Reflexiones sobre el origen de datos e información	5
3.1	¿Cual es el origen de los datos e información ?	5
3.2	¿Cuales son las consideraciones legales o eticas del uso de la información?	6
3.3	¿Cuales son los retos de la información y los datos que utilizara en la base de datos en terminos de la calidad y la consolidación?	6
3.4	¿Que espera de la utilización de un sistema de Bases de Datos para su proyecto?	6
4	Diseño del Modelo de Datos del SMBD (Sistema Manejador de Bases de Datos)	7
4.1	Características del SMBD (Sistema Manejador de Bases de Datos) para el proyecto	7

4.2	Diagrama modelo de datos	8
4.3	Imágenes de la Base de Datos	9
4.4	Código SQL - lenguaje de definición de datos (DDL)(<i>código en anexos</i>)	9
4.5	Código SQL - Manipulación de datos (DML) (<i>código en anexos</i>)	9
4.6	Código SQL + Resultados: Vistas (<i>código en anexos</i>)	10
4.7	Código SQL + Resultados: Triggers (<i>código en anexos</i>)	10
4.8	Código SQL + Resultados: Funciones (<i>código en anexos</i>)	11
4.9	Código SQL + Resultados: procedimientos almacenados (<i>código en anexos</i>)	12
5	Bases de Datos No-SQL	13
5.1	Diagrama Bases de Datos No-SQL (<i>código en anexos</i>)	13
5.2	SMBD utilizado para la Base de Datos No-SQL (<i>Segunda entrega</i>)	13
6	Aplicación de ETL (Extract, Transform, Load) y Bodega de Datos (<i>Tercera entrega</i>)	15
6.1	Ejemplo de aplicación de ETL y Bodega de Datos (<i>Tercera entrega</i>)	15
7	Lecciones aprendidas (<i>Tercera entrega</i>)	16
8	Bibliografía	17
9	Anexos Códigos (<i>Tercera entrega</i>)	18
9.1	Tablas (<i>Bases de datos</i>)	18
9.2	Eliminación de datos nulos (<i>Bases de datos - Conflicto armado</i>)	19
9.3	Eliminación de datos nulos (<i>Bases de datos - Pobreza</i>)	19
9.4	Rutinas y llaves de las bases de datos(<i>Bases de datos -</i>)	20
9.5	DDL, DML, llaves, Vistas, Triggers, Funciones, procedimientos almacenados (<i>Bases de datos organizadas</i>)	20
9.6	Código NoSQL (<i>Bases de datos No Relacional</i>)	22

1 Introducción

Para el año 2021 los estudiantes matriculados en Colombia fueron 9.797.677 alumnos, cifra inferior al año 2020 con 9.712.511 alumnos, muestra una disminución de 85.166 estudiantes (DANE,2021). El presupuesto asignado para educación en el país fue de 49,4 billones de pesos de los 350 billones de pesos aprobados del Presupuesto General de la Nación (DNP,2021).

De estos 49,4 billones el 89 por ciento se destinan para funcionamiento, es decir 43,9 billones deben ser usados para que 9.797.677 estudiantes e instituciones universitarias, brinden garantías y den resultados óptimos de calidad, algo injusto en algunos aspectos teniendo en cuenta los años de atraso en materia de infraestructura, en actualización de contenidos, de mejoras en los accesos a la educación, en garantías para docentes, entre tantas otras.

De igual forma, se complica aún más el acceso y las garantías cuando en un país como Colombia, se presenta casos como el de Centro Poblados en el cual se perdieron 70 mil millones de pesos (Revista Portafolio, 2021), destinados para para garantizar el acceso a internet en zonas alejadas del país, o el PAE (plan de alimentación Escolar), destinado a brindar una alimentación adecuada a los niños de bajos recursos.

Teniendo en cuenta este panorama, se hace un poco más complicado garantizar un adecuado acceso a una educación de calidad en zonas excluidas históricamente, con presencia de conflicto, con carencias generales y alejadas del bogo centrismo que gobierna Colombia, es decir, en departamentos ubicados en la región caribe (Atlántico, Bolívar, Cesar, Córdoba, Sucre, La Guajira, Magdalena, San Andrés y Providencia)

2 Características del proyecto de investigación

2.1 Título del proyecto de investigación

DESERCIÓN EN EDUCACIÓN MEDIA PARA LA REGIÓN CARIBE EN EL AÑO 2021

2.2 Objetivo general

Determinar el aporte realizado en Cobertura Media para la Región Caribe para el año 2021, en comparación con la Cobertura Media del resto del País.

2.2.1 Objetivos específicos

- Determinar el departamento de la región caribe que presentó la menor deserción en educación Media para el año 2021.
- Definir el departamento de la región caribe que tiene la menor de reprobación en educación Media para el año 2021.

- Identificar qué departamento de la región caribe presentó la más alta de deserción en educación Media para el año 2021.
- Determinar qué departamento de la región caribe presentó la más alta de repitencia en educación Media para el año 2021.

2.3 Alcance

La educación colombiana se mantiene en vilo no solo por los coletazos que ha dejado la pandemia del año 2020 por un déficit histórico que tiene el país y por un giro a la izquierda en las políticas nacionales con el cambio de gobierno en el año 2022, porque, aunque el gobierno saliente indica que la cobertura a nivel general “aumento”, en contraposición un estudio brindado por el DANE indica que aun hacen falta 206.260 mil cupos que deberían ayudar a garantizar el acceso a la educación de los 793.311 niños hasta los 5 años. El presente escrito busca determinar cuál es el porcentaje de Cobertura Neta en educación media en la región Caribe, en comparación con el porcentaje de Cobertura Neta Nacional, para el 2021, teniendo en cuenta variables como la corrupción y la violencia en el desarrollo académico de los niños de esta región.

2.4 Pregunta de investigación

¿La Cobertura Neta en educación media en la región Caribe es mayor que la Cobertura Neta del resto de departamentos de Colombia para el año 2021?

2.5 Hipotesis

Por Antonomasia se considera que los departamentos, las regiones, los lugares en los cuales se presentan mayores niveles de corrupción, de violencia, de carencias, son los lugares donde la niñez tiene una menor proyección académica, pero que sucedería si los números, las estadísticas rompieran la tendencia o esta falsa creencia. Por medio de este escrito se procurará comprobar con el uso de bases de datos académicas, de violencia y de pobreza. Sí los estudiantes de la Región Caribe presentan números más bajos en calidad en el nivel de educación media en comparación con el resto del País.

3 Reflexiones sobre el origen de datos e información

La fuente de datos de la operación es el registro realizado, normalmente, por los rectores de las sedes y establecimientos educativos de las Entidades Territoriales Certificadas (ETC) en el Sistema Integrado de Matrícula (SIMAT), es decir, que la información plasmada en la base de datos que presenta el Ministerio de Educación Nacional (MEN), debe estar actualizada y debe ser lo más fehaciente posible.

Esta base de datos se comenzó a publicar en el año 2016, es de carácter anualizado y se va actualizando con base en la información que se va agregando al SIMAT por medio de las Entidades Territoriales Certificadas, el rector de establecimiento educativo o el personal administrativo responsable de la entidad territorial.

De igual forma, se usan 3 bases de datos adicionales para darle continuidad a los objetivos, Pobreza, Conflicto armado y seguridad ciudadana, Educación, las cuales son brindadas por la página de la Dirección Nacional de Planeación (DNP) y su aplicativo TERRIDATA, en la cual reposan los datos brindados y los cuales son de carácter público.

Dentro del proceso de recolección y publicación de la información se debe realizar, el proceso de planeación de las entidades, sus capacidades institucionales, así como las proyecciones de cupos estudiantiles para el siguiente año, de igual forma, se deben tener en cuenta la cantidad de solicitudes, la asignación de los cupos educativos, la cantidad real de matriculados, lo cual deriva en una auditoria directa sobre la información brindada por cada ETC, como lo establece la Resolución 7797 de 2015 del Ministerio de Educación.

Realizado el proceso de recolección de la información se procede a desagregar de forma regional, luego departamental, municipal, zonal (sea urbano o rural), posteriormente por sedes y por establecimiento. De igual forma, se desagrega por sexo, grado, nivel educativo (transición, básica primaria, básica secundaria y media), el sector (público o privado), la zona, la jornada (mañana, tarde, nocturna) y por modelo educativo.

Esta forma tan detallada de recolectar y desagregar la información debería mostrarle al país los datos más precisos y exactos posibles, pero como no es del todo posible acceder a ciertas zonas de la geografía nacional o en algunas ocasiones la información suministrada en el SIMAT es manipulada por los mismos encargados de brindarla, resulta muy complicado y aunque es la más acercada a la realidad no implica que no se presenten fallos en la información consignada o que haya fallos de juicio al momento de desagregarla.

3.1 ¿Cual es el origen de los datos e información ?

Se toma como base la información de DATOS ABIERTOS acerca de las estadísticas de educación en los niveles de preescolar, primaria y media, brindadas por el ministerio de educación de Colombia, para los años 2019 y 2021, de igual forma se usan los datos para violencia y pobreza brindados por TERRIDATA de la Dirección Nacional de planeación (DNP).

3.2 ¿Cuales son las consideraciones legales o eticas del uso de la información?

Al tratarse de un tema tan delicado como la educación en menores de edad, se presenta el dilema legal y ético por el uso de la información, pero al tratarse de un conjunto de datos, en los cuales no se reflejen datos personales, sino el manejo de cifras bases de datos públicas y al no contener información sensible, confidencial o que vulnere los derechos de los menores de edad, no se estaría incurriendo en una agresión a los estatutos de manejo de información ni a la ética

3.3 ¿Cuales son los retos de la información y los datos que utilizara en la base de datos en terminos de la calidad y la consolidación?

Al realizar el proceso de combinar la base de datos principal de educación, con las bases de datos de Pobreza y Violencia se pueda consolidar la información requerida para poder observar en que medida afectan estos dos detonadores influyen directamente en el rendimiento y los estándares de Educación de la Región Caribe en comparación con otras zonas que hayan sufrido en mayor, menor o igual medida el impacto de estos dos detonantes.

3.4 ¿Que espera de la utilización de un sistema de Bases de Datos para su proyecto?

Se espera que el uso de MySQL facilite no solo la consolidación de la información, sino la forma en que se pueda estructurar, dando un margen de mejora de esta, sin tener que dar inicio desde ceros a todo el proceso. De igual forma, se espera que al ser un sistema de gestión permita una mejor lectura e interpretación de las de datos los datos usados para el estudio.

4 Diseño del Modelo de Datos del SMBD (Sistema Manejador de Bases de Datos)

4.1 Características del SMBD (Sistema Manejador de Bases de Datos) para el proyecto

MySQL es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto respaldado por Oracle y basado en el lenguaje de consulta estructurado (SQL) más extendido en la actualidad para almacenar y administrar datos.

- Admite muchos tipos de datos, como enteros con o sin signo, float, texto, data time, entre otros.
- MySQL permite almacenar y acceder a los datos a través de múltiples motores de almacenamiento, incluyendo InnoDB, CSV y NDB.
- Es capaz de replicar datos y particionar tablas para mejorar el rendimiento y la durabilidad
- Los usuarios de MySQL no tienen que aprender nuevos comandos; pueden acceder a sus datos utilizando comandos SQL estándar
- MySQL está escrito en C y C++ y es accesible y está disponible en más de 20 plataformas, como Mac, Windows, Linux y Unix
- Soporta grandes bases de datos con millones de registros y admite muchos tipos de datos
- Para la seguridad se utiliza un sistema de privilegios de acceso y contraseñas encriptadas que permite la verificación basada en el host.
- MySQL es posible conectarse a MySQL Server utilizando varios protocolos, incluyendo sockets TCP/IP en cualquier plataforma
- Es un componente de LAMP, plataforma de desarrollo web que utiliza Linux como sistema operativo, Apache como servidor web.

Sentencias básicas: SELECT es usada para consultar datos. DISTINCT Sirve para eliminar los duplicados de las consultas de datos. WHERE Es usada incluir las condiciones de los datos que queremos consultar. AND y OR es usada para incluir 2 o más condiciones a una consulta. ORDER BY Es usada para ordenar los resultados de una consulta. INSERT Es usada para insertar datos. UPDATE Es usada actualizar o modificar datos ya existentes. DELETE Es usada borrar datos.

4.2 Diagrama modelo de datos

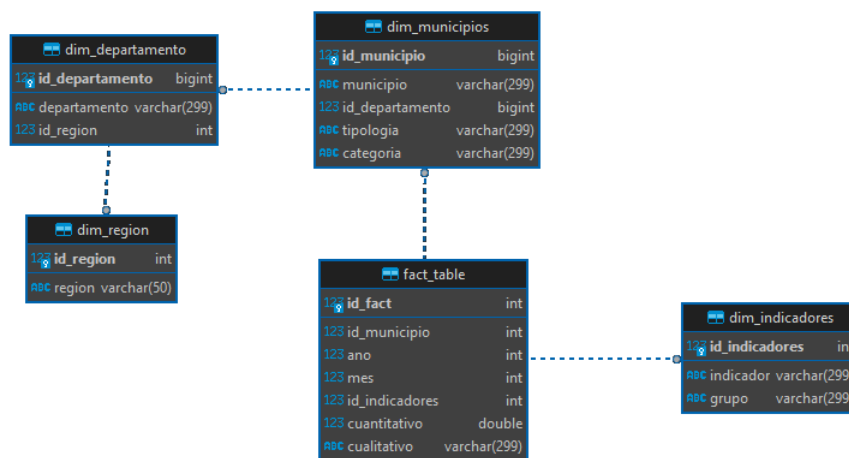
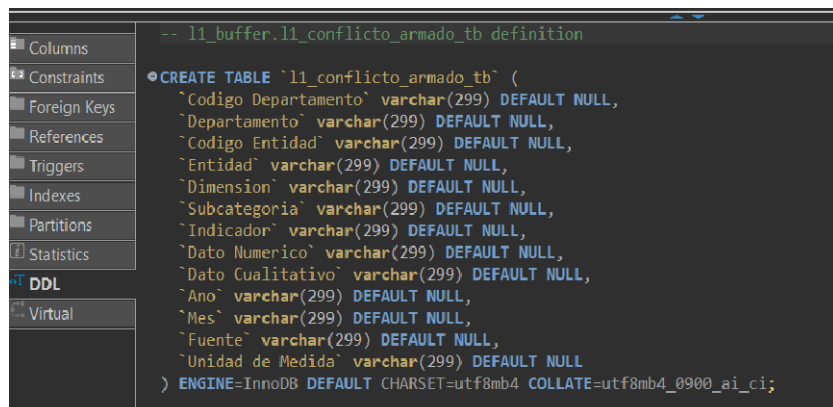


Figure 1: Modelo entidad de relación

EL modelo construido consta de 5 tablas, de las cuales 4 de ellas son dimensiones y una única tabla de hechos donde se plasman todos los indicadores medidos para los departamentos y municipios. En las dimensiones geográficas tenemos las segmentaciones propias de la DIVIPOLA a nivel nacional obteniendo, en la tabla de municipio, una tipología asociada al de nivel de ruralidad/urbanidad de este y una categoría en terminos de factores económicos.

4.3 Imágenes de la Base de Datos

4.4 Código SQL - lenguaje de definición de datos (DDL)(*código en anexos*)



```
-- l1_buffer.l1_conflicto_armado_tb definition
CREATE TABLE `l1_conflicto_armado_tb` (
  `Codigo Departamento` varchar(299) DEFAULT NULL,
  `Departamento` varchar(299) DEFAULT NULL,
  `Codigo Entidad` varchar(299) DEFAULT NULL,
  `Entidad` varchar(299) DEFAULT NULL,
  `Dimension` varchar(299) DEFAULT NULL,
  `Subcategoria` varchar(299) DEFAULT NULL,
  `Indicador` varchar(299) DEFAULT NULL,
  `Dato Numerico` varchar(299) DEFAULT NULL,
  `Dato Cualitativo` varchar(299) DEFAULT NULL,
  `Año` varchar(299) DEFAULT NULL,
  `Mes` varchar(299) DEFAULT NULL,
  `Fuente` varchar(299) DEFAULT NULL,
  `Unidad de Medida` varchar(299) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Figure 2: DDL

De las 4 bases de datos usadas para el desarrollo del proyecto se un operando CREATE para crear la tabla principal l3.edw, en la cual se ven consignados todos los cambios realizados a las bases iniciales, cambios como eliminación de datos nulos, organización de las palabras, en pocas palabras limpieza y organización de la data.

4.5 Código SQL - Manipulación de datos (DML) (*código en anexos*)

Se uso la sentencia SELECT para la selección adecuada de la información, en especial las columnas que se repiten en las 4 bases de datos, al igual que JOIN e INNER JOIN, para juntar las bases de datos si generar una repitencia de las columnas generadas ni una saturación de la información contenida en la tabla final l3.edw.

```

create table if not exists l3_edw.dim_municipios (primary key (id_municipio)) as
select
    cast(id_municipio as signed) id_municipio,
    municipio,
    cast(id_departamento as signed) id_departamento,
    b.id_region,
    tipologia as id_tipologia,
    categoria as id_categoria
from (
    select *,
    row_number() over(partition by id_municipio order by categoria desc) rn
    from l2_stage.distinct_municipios
) a
join l3_edw.dim_region b on a.region = b.region
where a.rn = 1;

```

Figure 3: DML

4.6 Código SQL + Resultados: Vistas *(código en anexos)*

se procede a agregar la vista y el código usado, con enfoque en la sentencia SELECT distinct

```

create table if not exists l3_edw.dim_departamento (primary key (id_departamento)) as
select distinct
    cast(id_departamento as signed) id_departamento,
    departamento
from l2_stage.l2_territorios_tb;

create temporary table if not exists l2_stage.distinct_municipios as
select distinct *
from l2_stage.l2_territorios_tb ltt;

```

Figure 4: Vistas

4.7 Código SQL + Resultados: Triggers *(código en anexos)*

En el código no se generaron triggers, se generaron índices de las tablas creadas con las bases de datos, en los cuales se pueden observar las llaves primarias de las tablas.

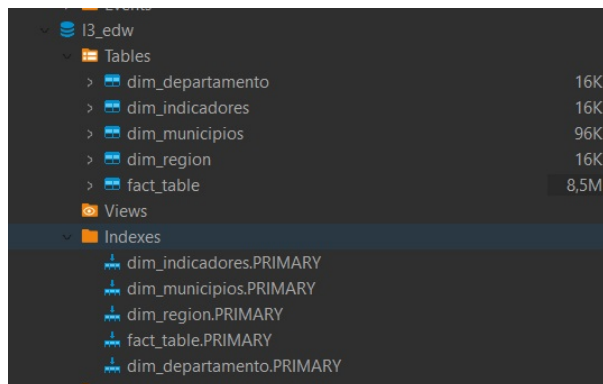


Figure 5: Indices

4.8 Código SQL + Resultados: Funciones *(código en anexos)*

se procedio a usar la función distinct para eliminar los datos repetidos en las tablas existentes en las tablas indicadores y municipios de las bases de datos.

```

)
select
  cast(a.id_municipio as signed) id_municipio,
  cast(a.ano as signed) ano,
  cast(a.mes as signed) mes,
  b.id_indicadores,
  a.cuantitativo,
  a.cualitativo
from l3_edw.fact_temp a
inner join l3_edw.dim_indicadores b on a.indicador = b.indicador;

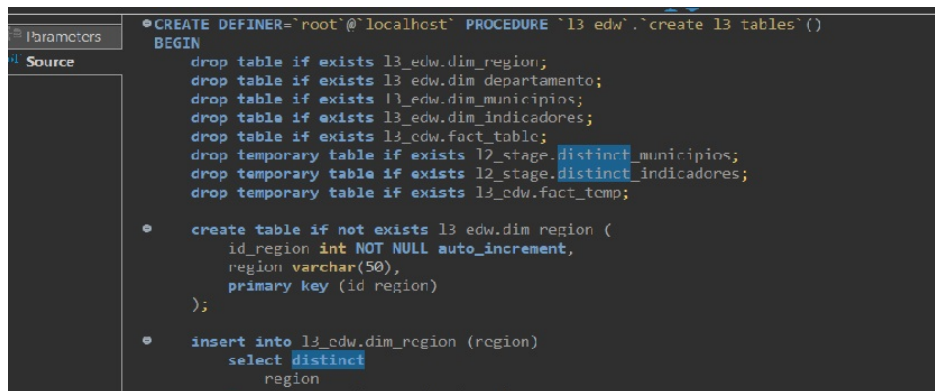
drop temporary table if exists l2_stage.distinct_indicadores;
drop temporary table if exists l2_stage.distinct_municipios;
drop temporary table if exists l3_edw.fact_temp;
END

```

Figure 6: Funciones código

4.9 Código SQL + Resultados: procedimientos almacenados (código en anexos)

Se procede a mostrar el concepto general del conjunto de datos que fueron manipulados y los cuales generaron el desarrollo general del concepto de la base l3.edw En la segunda imagen se puede ver el resultado del procedimiento aplicado y como este compila en l3.edw la creación general de la tabla con las modificaciones y limpieza anteriormente aplicadas.



```
CREATE DEFINER='root'@'localhost' PROCEDURE `l3_edw`.`create l3 tables`()
BEGIN
    drop table if exists l3_edw.dim_region;
    drop table if exists l3_edw.dim_departamento;
    drop table if exists l3_edw.dim_municipios;
    drop table if exists l3_edw.dim_indicadores;
    drop table if exists l3_edw.fact_table;
    drop temporary table if exists l2_stage.distinct_municipios;
    drop temporary table if exists l2_stage.distinct_indicadores;
    drop temporary table if exists l3_edw.fact_temp;

    create table if not exists l3_edw.dim_region (
        id_region int NOT NULL auto_increment,
        region varchar(50),
        primary key (id_region)
    );

    insert into l3_edw.dim_region (region)
    select distinct
        region
```

Figure 7: Procedimiento

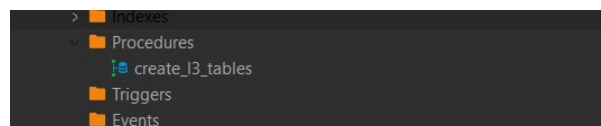


Figure 8: Procedimiento Resultados

5 Bases de Datos No-SQL

5.1 Diagrama Bases de Datos No-SQL (*código en anexos*)

Se procede a mostrar el concepto general del conjunto de datos que fueron manipulados y los cuales generaron el desarrollo general del concepto de la base en MongoDB.

```
_id: ObjectId('63617e0d8ebcc8fbf3a7ff12')
AÑO: 2021
MUNICIPIO: "Medellín"
DEPARTAMENTO: "Antioquia"
COBERTURA_NETA: 9694
COBERTURA_NETA_MEDIA: 5438
COBERTURA_BRUTA: 10993
COBERTURA_BRUTA_MEDIA: 10121
DESERCIÓN_MEDIA: 265
APROBACIÓN: 8789
REPROBACIÓN: 884
REPROBACIÓN_MEDIA: 855
REPITENCIA_MEDIA: 363
```

Figure 9: Modelo NoSQL

5.2 SMBD utilizado para la Base de Datos No-SQL (*Segunda entrega*)

MongoDB es un sistema de base de datos NoSQL, orientado a documentos y de código abierto.

En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

MongoDB es una base de datos adecuada para su uso en producción y con múltiples funcionalidades. Esta base de datos se utiliza mucho en la industria,¹ contando con implantaciones en empresas como MTV Network,² Craigslist,³ Foursquare.⁴

Consultas ad hoc

MongoDB soporta la búsqueda por campos, consultas de rangos y expresiones regulares. Las consultas pueden devolver un campo específico del documento pero también puede ser una función definida por el usuario para su mejor ocupación.

Indexación

Cualquier campo en un documento de MongoDB puede ser indexado, al igual que es posible hacer índices secundarios. El concepto de índices en MongoDB

es similar al empleado en base de datos relacionales..

Replicación

MongoDB soporta el tipo de replicación primario-secundario. Cada grupo de primario y sus secundarios se denomina replica set.¹³ El primario puede ejecutar comandos de lectura y escritura. Los secundarios replican los datos del primario y sólo se pueden usar para lectura o para copia de seguridad, pero no se pueden realizar escrituras. Los secundarios tienen la habilidad de poder elegir un nuevo primario en caso de que el primario actual deje de responder.

Balanceo de carga

MongoDB puede escalar de forma horizontal usando el concepto de [shard].¹⁴ El desarrollador elige una clave de sharding, la cual determina cómo serán distribuidos los datos de una colección. Los datos son divididos en rangos (basado en la clave de sharding) y distribuidos a través de múltiples shard. Cada shard puede ser una réplica set. MongoDB tiene la capacidad de ejecutarse en múltiples servidores, balanceando la carga y/o replicando los datos para poder mantener el sistema funcionando en caso de que exista un fallo de hardware. La configuración automática es fácil de implementar bajo MongoDB y se pueden agregar nuevos servidores a MongoDB con el sistema de base de datos funcionando.

Almacenamiento de archivos

MongoDB puede ser utilizado como un sistema de archivos, aprovechando la capacidad de MongoDB para el balanceo de carga y la replicación de datos en múltiples servidores. Esta funcionalidad, llamada GridFS¹⁵ e incluida en la distribución oficial, implementa sobre los drivers, no sobre el servidor,¹⁶ una serie de funciones y métodos para manipular archivos y contenido. En un sistema con múltiples servidores, los archivos pueden ser distribuidos y replicados entre los mismos de forma transparente, creando así un sistema eficiente tolerante de fallos y con balanceo de carga.

Agregación

MongoDB proporciona un framework de agregación que permite realizar operaciones similares al "GROUP BY" de SQL. El framework de agregación está construido como un pipeline en el que los datos van pasando a través de diferentes etapas en las cuales estos datos son modificados, agregados, filtrados y formateados hasta obtener el resultado deseado. Todo este procesamiento es capaz de utilizar índices si existieran y se produce en memoria. Asimismo, MongoDB proporciona una función MapReduce que puede ser utilizada para el procesamiento por lotes de datos y operaciones de agregación.

6 Aplicación de ETL (Extract, Transform, Load) y Bodega de Datos *(Tercera entrega)*

6.1 Ejemplo de aplicación de ETL y Bodega de Datos *(Tercera entrega)*

7 Lecciones aprendidas (*Tercera entrega*)

8 Bibliografía

<https://www.mineducacion.gov.co/portal/normativa/Resoluciones/351282:Resolucion-No-07797-de-2015>

<https://www.dane.gov.co/files/investigaciones/educacion/educacion-formal/2021/bol-EDUC-21-pdf>

https://www.dane.gov.co/files/investigaciones/boletines/educacion/bol_EDUC20.pdf

https://www.dane.gov.co/files/investigaciones/boletines/educacion/bol_EDUC19.pdf

<https://www.dane.gov.co/index.php/estadisticas-por-tema/pobreza-y-condiciones-de-vida/pobreza-multidimensional/pobreza-multidimensional-2020>

https://www.datos.gov.co/Educacion/MEN_ESTADISTICAS_EN_EDUCACION_EN_PREESCOLAR-B-SICA/nudc-7mev

9 Anexos Códigos (Tercera entrega)

9.1 Tablas (Bases de datos)

```
/*!40101 SET @OLD_CHARACTERSET_CLIENT = @@CHARACTERSET_CLIENT*
/;
/*!40101 SET @OLD_CHARACTERSET_RESULTS = @@CHARACTERSET_RESULTS*
/;
/*!40101 SET @OLD_COLLATION_CONNECTION = @@COLLATION_CONNECTION*
/;
/*!50503 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE = @@TIME_ZONE */;
/*!40103 SET TIME_ZONE = '+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS = @@UNIQUE_CHECKS, UNIQUE_CHECKS =
0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS =
@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS = 0
*/;
/*!40101 SET @OLD_SQL_MODE = @@SQL_MODE, SQL_MODE = 'NO_AUTO_VALUE_ON_ZERO'
/;
/*!40111 SET @OLD_SQL_NOTES = @@SQL_NOTES, SQL_NOTES = 0 */;
/;
CREATE DATABASE 'l_bu_fer' DEFAULT CHARACTER SET utf8mb4;
USE 'l_bu_fer';
-- Table structure for table 'l_conflicto_armado_ib' --
DROP TABLE IF EXISTS 'l_conflicto_armado_ib';
/*!40101 SET @saved_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE 'l_conflicto_armado_ib' (
  'Codigo Departamento' varchar(299) DEFAULT NULL,
  'Departamento' varchar(299) DEFAULT NULL,
  'Codigo Entidad' varchar(299) DEFAULT NULL,
  'Entidad' varchar(299) DEFAULT NULL,
  'Dimension' varchar(299) DEFAULT NULL,
  'Subcategoria' varchar(299) DEFAULT NULL,
  'Indicador' varchar(299) DEFAULT NULL,
  'Dato Numerico' varchar(299) DEFAULT NULL,
  'Dato Cualitativo' varchar(299) DEFAULT NULL,
  'Año' varchar(299) DEFAULT NULL,
  'Mes' varchar(299) DEFAULT NULL,
  'Fuente' varchar(299) DEFAULT NULL,
  'Unidad de Medida' varchar(299) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_client */;
```

9.2 Eliminación de datos nulos (*Bases de datos - Conflicto armado*)

```
DROP TABLE IF EXISTS '11_estadisticas_academicas_t'b';
/*!40101 SET @saved_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE '11_estadisticas_academicas_t'b'(
  'Codigo Departamento' varchar(299) DEFAULT NULL,
  'Departamento' varchar(299) DEFAULT NULL,
  'Codigo Entidad' varchar(299) DEFAULT NULL,
  'Entidad' varchar(299) DEFAULT NULL,
  'Dimension' varchar(299) DEFAULT NULL,
  'Subcategoria' varchar(299) DEFAULT NULL,
  'Indicador' varchar(299) DEFAULT NULL,
  'Dato Numerico' varchar(299) DEFAULT NULL,
  'Dato Cualitativo' varchar(299) DEFAULT NULL,
  'Año' int DEFAULT NULL,
  'Mes' int DEFAULT NULL,
  'Fuente' varchar(299) DEFAULT NULL,
  'Unidad de Medida' varchar(299) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_client */;
```

9.3 Eliminación de datos nulos (*Bases de datos - Pobreza*)

```
DROP TABLE IF EXISTS '11_pobreza_t'b';
/*!40101 SET @saved_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE '11_pobreza_t'b'(
  'Codigo Departamento' varchar(299) DEFAULT NULL,
  'Departamento' varchar(299) DEFAULT NULL,
  'Codigo Entidad' varchar(299) DEFAULT NULL,
  'Entidad' varchar(299) DEFAULT NULL,
  'Dimension' varchar(299) DEFAULT NULL,
  'Subcategoria' varchar(299) DEFAULT NULL,
  'Indicador' varchar(512) DEFAULT NULL,
  'Dato Numerico' varchar(299) DEFAULT NULL,
  'Dato Cualitativo' varchar(299) DEFAULT NULL,
  'Año' varchar(299) DEFAULT NULL,
  'Mes' varchar(299) DEFAULT NULL,
  'Fuente' varchar(512) DEFAULT NULL,
  'Unidad de Medida' varchar(299) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_client */;
```

9.4 Rutinas y llaves de las bases de datos(*Bases de datos -*)

```
– Dumping routines for database 'lbuffer' – –
/*!40103 SET TIME_ZONE = @OLD_TIME_ZONE * /;
/*!40101 SET SQL_MODE = @OLD_SQL_MODE * /;
/*!40014 SET FOREIGN_KEY_CHECKS = @OLD_FOREIGN_KEY_CHECKS*
/;
/*!40014 SET UNIQUE_CHECKS = @OLD_UNIQUE_CHECKS * /;
/*!40101 SET CHARACTER_SET_CLIENT = @OLD_CHARACTER_SET_CLIENT*
/;
/*!40101 SET CHARACTER_SET_RESULTS = @OLD_CHARACTER_SET_RESULTS*
/;
/*!40101 SET COLLATION_CONNECTION = @OLD_COLLATION_CONNECTION*
/;
/*!40111 SET SQL_NOTES = @OLD_SQL_NOTES * /;
```

9.5 DDL, DML, llaves, Vistas, Triggers, Funciones, procedimientos almacenados (*Bases de datos organizadas*)

```
CREATE DEFINER='root'@'localhost' PROCEDURE 'create_tables'() BEGIN
drop table if exists l3_dw.dim_region;
drop table if exists l3_dw.dim_departamento;
drop table if exists l3_dw.dim_municipios;
drop table if exists l3_dw.dim_indicadores;
drop table if exists l3_dw.fact_table;
drop temporary table if exists l2_stage.distinct_municipios;
drop temporary table if exists l2_stage.distinct_indicadores;
drop temporary table if exists l3_dw.fact_emp;
create table if not exists l3_dw.dim_region(
id_region int NOT NULL auto_increment,
region varchar(50),
primary key (id_region));
insert into l3_dw.dim_region(region) select distinct region from l2_stage.l2_territorios_b;
create table if not exists l3_dw.dim_departamento(primarykey
(id_departamento)) as
select distinct
cast(id_departamento assigned) id_departamento,
departamento
from l2_stage.l2_territorios_b;
create temporary table if not exists l2_stage.distinct_municipios as
select distinct *
from l2_stage.l2_territorios_b;
create table if not exists l3_dw.dim_municipios(primarykey(id_municipio))
as
select
cast(id_municipio assigned) id_municipio,
```

```

municipio,
cast(idddepartamentoassigned)idddepartamento,
b.idrregion,
tipologia as idttipologia,
categoria as idccategoria
from (
select *,
rownumber()over(partitionbyidmmunicipioorderbycategoriadesc)rn
from l2stage.distinctmmunicipios
) a
join l3edw.dimregionbona.region = b.region
where a.rn = 1;
create temporary table if not exists l2stage.distinctiindicadoresas
select distinct indicador, grupo
from (
select *, 'CONFLICTO' grupo
from l2stage.l2conflictoarmadotblcat
union all
select *, 'POBREZA' grupo
from l2stage.l2pobrezaatblpt
union all
select *, 'EDUCACION' grupo
from l2stage.l2estadisticasacademicastbleat
) a;
create table if not exists l3edw.dimiindicadores(
idiindicadoresintNOTNULLautoincrement,
indicador varchar(299),
grupo varchar(299),
primary key (idiindicadores));
insert into l3edw.dimiindicadores(indicador, grupo)
select distinct
indicador,
grupo
from l2stage.distinctiindicadores;
create temporary table if not exists l3edw.facttempas
select *
from (
select
idddepartamento, departamento, idmmunicipio, municipio, ano, mes,
indicador, cast(cuantitativo as double) cuantitativo, cualitativo
from l2stage.l2estadisticasacademicastbleat
union all
select
idddepartamento, departamento, idmmunicipio, municipio, ano, mes,
indicador, cast(replace(cuantitativo, ',', '.') as double)
cuantitativo, cualitativo

```

```

from l2stage.l2pobrezatblpt
union all
select
idddepartamento, departamento, idmmunicipio, municipio, ano, mes,
indicador,
cast(replace(replace(replace(cuantitativo,','//'),','.''),'//','.'') as double) cuantitativo, cualitativo
from l2stage.l2eonflictoarmadotblcat
) a;
create table if not exists l3edw.facttable(
idfactint NOT NULL autoincrement,
idmmunicipioint,
ano int,
mes int,
idiindicadoresint,
cuantitativo double,
cualitativo varchar(299),
primary key (idfact)
);
insert into l3edw.facttable
( idmmunicipio,
ano,
mes,
idiindicadores,
cuantitativo,
cualitativo
)
select
cast(a.idmmunicipioassigned)idmmunicipio,
cast(a.ano as signed) ano,
cast(a.mes as signed) mes,
b.idiindicadores,
a.cuantitativo,
a.cualitativo
from l3edw.facttempa
inner join l3edw.dimiindicadoresbona.indicador = b.indicador;
drop temporary table if exists l2stage.distinctiindicadores;
drop temporary table if exists l2stage.distinctmmunicipios;
drop temporary table if exists l3edw.facttemp;
END ;;

```

9.6 Código NoSQL (*Bases de datos No Relacional*)

```

[
  "AÑO": 2021,
  "MUNICIPIO": "Medellín",

```

"DEPARTAMENTO": "Antioquia",
 "COBERTURA_NETA" : 9694,
 "COBERTURA_NETA_MEDIA" : 5438,
 "COBERTURA_BRUTA" : 10993,
 "COBERTURA_BRUTA_MEDIA" : 10121,
 "DESERCIÓN_MEDIA" : 265,
 "APROBACIÓN": 8789,
 "REPROBACIÓN": 884,
 "REPROBACIÓN_MEDIA" : 855,
 "REPITENCIA_MEDIA" : 363,
 "AÑO": 2021,
 "MUNICIPIO": "Abejorral",
 "DEPARTAMENTO": "Antioquia",
 "COBERTURA_NETA" : 8006,
 "COBERTURA_NETA_MEDIA" : 3991,
 "COBERTURA_BRUTA" : 8835,
 "COBERTURA_BRUTA_MEDIA" : 7451,
 "DESERCIÓN_MEDIA" : 293,
 "APROBACIÓN": 8697,
 "REPROBACIÓN": 905,
 "REPROBACIÓN_MEDIA" : 439,
 "REPITENCIA_MEDIA" : 122