

Proyecto de Bases de datos para XXX

Daniel Salgado¹, Juan Andres Benavides², Jorfan Vargas³

¹⁻²Dpto. de matematicas , ³Dpto. de matematicas,
Universidad Uexternado
Pregrado Ciencia de Datos
Curso de Bases de Datos
Bogotá, Colombia

Jorfan Vargas, Daniel Salgado, Juan Benavides

September 11, 2023

Contents

1	Introducción	3
2	Características del proyecto de investigación que hace uso de Bases de Datos	3
2.1	Titulo del proyecto de investigación	3
2.2	Objetivo general	3
2.2.1	Objetivos especificos	3
2.3	Alcance	3
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?	5
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?	5
3.4	¿Que espera de la utilización de un sistema de Bases de Datos para su proyecto? (Max 100 Palabras) - (<i>Primera entrega</i>)	5
4	Diseño del Modelo de Datos del SMBD (Sistema Manejador de Bases de Datos)(<i>Primera entrega</i>)	7
4.1	Características del SMBD (Sistema Manejador de Bases de Datos) para el proyecto (<i>Primera entrega</i>)	7
4.2	Diagrama modelo de datos (<i>Primera entrega</i>)	7

4.3	Imágenes de la Base de Datos (<i>Primera entrega</i>)	7
4.4	Código SQL - lenguaje de definición de datos (DDL) (<i>Primera entrega</i>)	8
4.5	Código SQL - Manipulación de datos (DML) (<i>Primera entrega</i>)	8
4.6	Código SQL + Resultados: Vistas (<i>Primera entrega</i>)	9
4.7	Código SQL + Resultados: Triggers (<i>Primera entrega</i>)	10
4.8	Código SQL + Resultados: Funciones (<i>Primera entrega</i>)	11
4.9	Código SQL + Resultados: procedimientos almacenados (<i>Primera entrega</i>)	11
5	Bases de Datos No-SQL (<i>Segunda entrega</i>)	13
5.1	Diagrama Bases de Datos No-SQL (<i>Segunda entrega</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>)	14
6.1	Ejemplo de aplicación de ETL y Bodega de Datos (<i>Tercera entrega</i>)	14
6.2	Automatización de Datos (<i>Tercera entrega</i>)	14
6.3	Integración de Datos (<i>Tercera entrega</i>)	14
7	Proximos pasos (<i>Tercera entrega</i>)	15
8	Lecciones aprendidas (<i>Tercera entrega</i>)	16
9	Bibliografía	17

1 Introducción

La hora del día, el tipo de vehículo y la ubicación del accidente son factores significativos e importantes que afectan la gravedad y la frecuencia de los accidentes de tráfico. Un análisis detallado y exploratorio de los datos de accidentes revelará patrones y tendencias clave en el contexto de la siniestralidad vial, lo que permitirá proponer medidas efectivas para mejorar la seguridad en las carreteras y reducir el número de accidentes y víctimas.

2 Características del proyecto de investigación que hace uso de Bases de Datos

2.1 Título del proyecto de investigación

Prevención y Control de Accidentes de Tráfico:

2.2 Objetivo general

realizar un análisis cuantitativo exhaustivo de los datos de accidentes de tráfico para identificar patrones y tendencias en la siniestralidad vial, con el propósito de proponer medidas efectivas para mejorar la seguridad en las carreteras y reducir el número de accidentes y víctimas. Este análisis incluirá el estudio de variables como la hora del día, el tipo de vehículo, la dirección del accidente y la gravedad de estos, entre otros factores relevantes.

2.2.1 Objetivos específicos

- Analizar la distribución temporal de los accidentes de tráfico, estudiando variables como la hora del día y el día de la semana, para identificar los momentos de mayor riesgo de accidente.
- Investigar la relación entre el tipo de vehículo y la gravedad de los accidentes, con el fin de determinar si ciertos tipos de vehículos están asociados con accidentes más graves.
- Examinar la distribución geográfica de los accidentes, identificando las direcciones con mayor número de accidentes, para proponer medidas de seguridad vial específicas para estas áreas.

2.3 Alcance

Fuentes de datos: datos abiertos Bogotá. Superintendencia de movilidad y transporte
Métodos: Encontrar diferentes relaciones que hay entre los posibles factores que causan mas accidentes de transito y los siniestros viales. exploración de los datos de accidentes de tráfico analizando detalladamente cada uno de los factores que pueden ser relacionados con la frecuencia o gravedad de los

accidentes. Después de identificar los factores mas relacionados con los accidentes de transito identificar las posibles estrategias para reducir el numero de accidentes o la gravedad del accidente.

2.4 Pregunta de investigación

¿Cuáles son los factores que contribuyen a la gravedad de los accidentes de tráfico en Bogotá?

2.5 Hipotesis

La hora del día, el tipo de vehículo y la ubicación del accidente son factores significativos e importantes que afectan la gravedad y la frecuencia de los accidentes de tráfico. Un análisis detallado y exploratorio de los datos de accidentes revelará patrones y tendencias clave en el contexto de la siniestralidad vial, lo que permitirá proponer medidas efectivas para mejorar la seguridad en las carreteras y reducir el número de accidentes y víctimas.

3 Reflexiones sobre el origen de datos e información

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

Los datos son registrados a través de la plataforma de datos abiertos de Bogotá realizados por la Secretaría Distrital de Movilidad, al ser una fuente de una entidad gubernamental basamos gran parte de nuestra confianza en este centro de datos, como se detalla de mejor manera en el punto anterior, a nuestro criterio la base tiene elementos muy completos e interesantes para nuestro rango de investigación, en ese sentido, cada dato registrado se refiere a un siniestro ocurrido en la ciudad de Bogotá y cuyo situación haya sido reportada a la Secretaría, no solo eso, además las decisiones usualmente tomadas o datos publicados por parte de la alcaldía, son tomados de los registros de este tipo de entidades.

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

En este caso si bien es un tema un tanto delicado, creemos que la publicación de estos datos ante toda la internet es una gran decisión, si bien con un arduo estudio acerca de los datos se puede llegar a prestar para malas prácticas usualmente las decisiones tomadas en este tipo de entidades esta mas que fundamentada, aun así nuestro papel sobre los datos es para generar en gran parte conocimiento y tal vez rectificar o modificar una hipótesis principal que tenemos acerca del desarrollo de los siniestros viales.

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?

La secretaria tanto como nosotros entiende que hay siniestros que no son reportados ante ellos, a pesar de esto pensamos que la información tan detallada y completa, es suficiente para hacer una generalización en la ciudad. En este sentido creemos que quizás este sea uno de los mayores retos, no esto, sino que la terminología que usa para referirse a cada sujeto, elemento, vehículo, barrio, etc. La base de datos la registra con un número de identificación, esto provoca que debamos tomar mayor precaución y realizar mayor investigación para no caer en errores de nomenclatura.

3.4 ¿Que espera de la utilización de un sistema de Bases de Datos para su proyecto? (Max 100 Palabras) - (Primera entrega)

Esperamos desarrollar una base de datos mucho mas entendible de la que tenemos en nuestras manos, en ese sentido, definir de una mejor manera cuales son las entidades principales y como a raíz de estas podemos, no solo responder

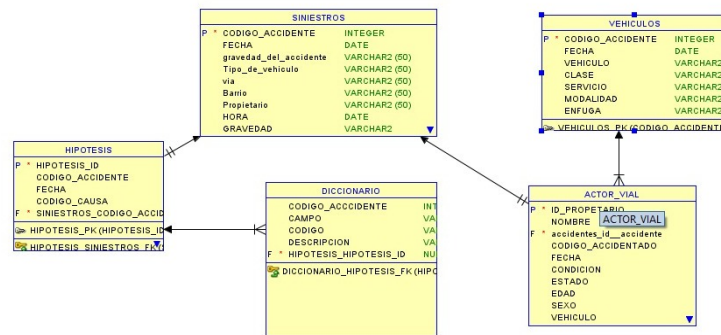
nuestra pregunta propuesta, sino que también, si en algún punto nuestro interés se va por otras ramas involucradas en la base, a través de las relaciones y vistas poder también extraer información relevante para las partes interesadas. En general, esperamos que el proceso de desarrollo con los datos y su implementación en la base sea de gratificación a la hora de referirnos a la absorción de conocimiento.

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

Oracle se refiere generalmente a "Oracle Database", que es un sistema de gestión de bases de datos relacional (RDBMS) desarrollado por Oracle Corporation. Oracle Database es uno de los sistemas de gestión de bases de datos más populares y ampliamente utilizados en el mundo empresarial y en aplicaciones críticas. Decidimos usar este programa principalmente por el conocimiento que hemos desarrollado en estas ultimas semanas en este.

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

4.2 Diagrama modelo de datos (Primera entrega)



4.3 Imágenes de la Base de Datos (Primera entrega)

196131	10530484	31/12/2020	13:40:00	3	1	1	CL 6-KR 708 02	8	1	
196132	10530494	31/12/2020	13:40:00	3	1	1	KR 37-CL 7 49	16	1	
196133	10530442	31/12/2020	14:00:00	2	1	1	KR 81C-CL 53 S 02	8	2	
196134	10530458	31/12/2020	14:40:00	3	1	1	CL 1-KR 18 02	14	2	
196135	10530487	31/12/2020	14:40:00	3	1	1	KR 72-CL 79 02	10	1	
196136	10530489	31/12/2020	14:59:00	3	1	1	CL 127-KR 45 02	1	4	
196137	10530486	31/12/2020	15:00:00	3	1	1	KR 81-CL 23F 02	9	2	
196138	10530475	31/12/2020	15:22:00	3	1	1	AV AVENIDA DE LAS AMERICAS-KR 62 02	16	1	
196139	10530482	31/12/2020	15:53:00	3	1	1	CL 31A-KR 120 S 02	18	2	
196140	10530482	31/12/2020	16:30:00	3	1	1	CL 13-KR 96 02	9	1	

4.4 Código SQL - lenguaje de definición de datos (DDL) (Primera entrega)

```
CREATE TABLE actor_vial (
    id_propetario          INTEGER NOT NULL,
    nombre                 VARCHAR2(20),
    accidentes_id_accidente INTEGER NOT NULL,
    codigo_accidentado     INTEGER,
    fecha                  DATE,
    condicion              VARCHAR2(15),
    estado                 VARCHAR2(20),
    edad                   INTEGER,
    sexo                   VARCHAR2(10),
    vehiculo                INTEGER,
    codigo_accidente        VARCHAR2(1000000) NOT NULL,
    vehiculos_codigo_accidente INTEGER NOT NULL
);

CREATE UNIQUE INDEX propietario_idx ON
actor_vial (
    accidentes_id_accidente
ASC );

ALTER TABLE actor_vial ADD CONSTRAINT propietario_pk PRIMARY KEY ( id_propetario );

CREATE TABLE diccionario (
    codigo_accidente      INTEGER,
    campo                 VARCHAR2(20),
    codigo                VARCHAR2(20),
    descripcion            VARCHAR2(20),
    hipotesis_hipotesis_id NUMBER NOT NULL
);

CREATE TABLE hipotesis (
    hipotesis_id          NUMBER NOT NULL,
    codigo_accidente       INTEGER,
    fecha                  DATE,
    codigo_causa            INTEGER,
    siniestros_codigo_accidente INTEGER NOT NULL
);

CREATE UNIQUE INDEX hipotesis_idx ON
hipotesis (
    siniestros_codigo_accidente
ASC );

ALTER TABLE hipotesis ADD CONSTRAINT hipotesis_pk PRIMARY KEY ( hipotesis_id );

CREATE TABLE siniestros (
    codigo_accidente      INTEGER NOT NULL,
    fecha                  DATE,
    gravedad_del_accidente VARCHAR2(50),
    tipo_de_vehiculo       VARCHAR2(50),
    via                    VARCHAR2(50),
    barrio                 VARCHAR2(50),
    propietario            VARCHAR2(50),
    hora                   DATE,
    gravedad                VARCHAR2(60),
    clase                  VARCHAR2(20),
    choque                 VARCHAR2(20),
    objeto_fijo             VARCHAR2(30),
    codigo_localidad        VARCHAR2(20),
    diseño_lugar            VARCHAR2(50)
);
```

4.5 Código SQL - Manipulación de datos (DML) (Primera entrega)

```
ALTER TABLE siniestros ADD CONSTRAINT accidentes_pk PRIMARY KEY ( codigo_accidente );

CREATE TABLE vehiculos (
    codigo_accidente      INTEGER NOT NULL,
    fecha                  DATE,
    vehiculo               VARCHAR2(16),
    clase                  VARCHAR2(20),
    servicio                VARCHAR2(20),
    modalidad              VARCHAR2(20),
    enfuga                 VARCHAR2(20)
);

ALTER TABLE vehiculos ADD CONSTRAINT vehiculos_pk PRIMARY KEY ( codigo_accidente );

ALTER TABLE actor_vial
ADD CONSTRAINT actor_vial_vehiculos_fk FOREIGN KEY ( vehiculos_codigo_accidente )
REFERENCES vehiculos ( codigo_accidente );

ALTER TABLE diccionario
ADD CONSTRAINT diccionario_hipotesis_fk FOREIGN KEY ( hipotesis_hipotesis_id )
REFERENCES hipotesis ( hipotesis_id );

ALTER TABLE hipotesis
ADD CONSTRAINT hipotesis_siniestros_fk FOREIGN KEY ( siniestros_codigo_accidente )
REFERENCES siniestros ( codigo_accidente );

ALTER TABLE actor_vial
ADD CONSTRAINT propietario_accidentes_fk FOREIGN KEY ( accidentes_id_accidente )
REFERENCES accidentes ( codigo_accidente );
```



```

ALTER TABLE actor_vial
  ADD CONSTRAINT actor_vial_vehiculos_fk FOREIGN KEY ( vehiculos_codigo_accidente )
  REFERENCES vehiculos ( codigo_accidente );

ALTER TABLE diccionario
  ADD CONSTRAINT diccionario_hipotesis_fk FOREIGN KEY ( hipotesis_hipotesis_id )
  REFERENCES hipotesis ( hipotesis_id );

ALTER TABLE hipotesis
  ADD CONSTRAINT hipotesis_siniestros_fk FOREIGN KEY ( siniestros_codigo_accidente )
  REFERENCES siniestros ( codigo_accidente );

ALTER TABLE actor_vial
  ADD CONSTRAINT propietario_accidentes_fk FOREIGN KEY ( accidentes_id_accidente )
  REFERENCES siniestros ( codigo_accidente );

CREATE SEQUENCE hipotesis_hipotesis_id_seq START WITH 1 NOCACHE ORDER;
CREATE OR REPLACE TRIGGER hipotesis_hipotesis_id_trg BEFORE
  INSERT ON hipotesis
  FOR EACH ROW
  WHEN ( new.hipotesis_id IS NULL )
  BEGIN
    :new.hipotesis_id := hipotesis_hipotesis_id_seq.nextval;
  END;
/

```

4.6 Código SQL + Resultados: Vistas (*Primera entrega*)

1. 5 Primeros vehiculos de la base de datos

```

CREATE OR REPLACE VIEW vista_vehiculos AS
SELECT *
FROM vehiculos
WHERE ROWNUM <= 5;

```

2. 10 primeros actores viales de la base de datos

```

CREATE OR REPLACE VIEW vista_actores_viales AS
SELECT *
FROM actor_vial
WHERE ROWNUM <= 10;

```

3. 10 primeras direcciones de la base

```
CREATE OR REPLACE VIEW vista_direcciones AS
SELECT *
FROM direcciones
WHERE ROWNUM <= 10;
```

4.7 Código SQL + Resultados: Triggers (*Primera entrega*)

1. Registro de auditoría de cambios en los datos

```
CREATE OR REPLACE FUNCTION vehiculo_mas_recurrente RETURN NUMBER IS
    v_vehiculo_id NUMBER;
    v_max_count NUMBER := 0;
BEGIN
    FOR rec IN (SELECT codigo_accidente, COUNT(*) AS count_vehiculo
                FROM vehiculos
                GROUP BY codigo_accidente
                ORDER BY COUNT(*) DESC)
    LOOP
        IF rec.count_vehiculo > v_max_count THEN
            v_max_count := rec.count_vehiculo;
            v_vehiculo_id := rec.codigo_accidente;
        END IF;
    END LOOP;

    RETURN v_vehiculo_id;
END vehiculo_mas_recurrente;
```

2. Validación de datos antes de la inserción

```
CREATE OR REPLACE TRIGGER validar_edad_actor_vial_trg
BEFORE INSERT ON actor_vial
FOR EACH ROW
BEGIN
    IF :new.edad < 18 THEN
        RAISE_APPLICATION_ERROR(-20001, 'La edad del actor vial debe ser mayor de 18 años.');
```

3. Control de restricciones de integridad referencial

```

CREATE OR REPLACE TRIGGER validar_integridad_hipotesis_trg
BEFORE INSERT ON hipotesis
FOR EACH ROW
BEGIN
    IF NOT EXISTS (SELECT 1 FROM siniestros WHERE codigo_accidente = :new.siniestros_codigo_accidente) THEN
        RAISE_APPLICATION_ERROR(-20002, 'El código de accidente en hipotesis no existe en la tabla siniestros.');
```

4.8 Código SQL + Resultados: Funciones (Primera entrega)

1. Localidad que mas aparece

```

CREATE OR REPLACE FUNCTION localidad_mas_comun RETURN VARCHAR2 IS
    v_localidad_mas_comun VARCHAR2(50);
    v_max_count NUMBER := 0;
BEGIN
    FOR rec IN (SELECT codigo_localidad, COUNT(*) AS count_localidad
                FROM siniestros
                GROUP BY codigo_localidad
                ORDER BY COUNT(*) DESC)
    LOOP
        IF rec.count_localidad > v_max_count THEN
            v_max_count := rec.count_localidad;
            v_localidad_mas_comun := rec.codigo_localidad;
        END IF;
    END LOOP;

    RETURN v_localidad_mas_comun;
END localidad_mas_comun;
/
```

2. Vehiculo mas recurrente

```

CREATE OR REPLACE FUNCTION vehiculo_mas_recurrente RETURN NUMBER IS
    v_vehiculo_id NUMBER;
    v_max_count NUMBER := 0;
BEGIN
    FOR rec IN (SELECT codigo_accidente, COUNT(*) AS count_vehiculo
                FROM vehiculos
                GROUP BY codigo_accidente
                ORDER BY COUNT(*) DESC)
    LOOP
        IF rec.count_vehiculo > v_max_count THEN
            v_max_count := rec.count_vehiculo;
            v_vehiculo_id := rec.codigo_accidente;
        END IF;
    END LOOP;

    RETURN v_vehiculo_id;
END vehiculo_mas_recurrente;
/
```

4.9 Código SQL + Resultados: procedimientos almacenados (Primera entrega)

1. Procedimiento para insertar un nuevo actor vial

```

CREATE OR REPLACE PROCEDURE insertar_actor_vial(
    p_nombre IN VARCHAR2,
    p_accidente_id IN INTEGER,
    p_codigo_accidentado IN INTEGER,
    p_fecha IN DATE,
    p_condicion IN VARCHAR2,
    p_estado IN VARCHAR2,
    p_edad IN INTEGER,
    p_sexo IN VARCHAR2,
    p_vehiculo IN INTEGER,
    p_codigo_accidente IN VARCHAR2,
    p_vehiculos_codigo_accidente IN INTEGER
)
IS
BEGIN
    INSERT INTO actor_vial (
        nombre,
        accidentes_id__accidente,
        codigo_accidentado,
        fecha,
        condicion,
        estado,
        edad,
        sexo,
        vehiculo,
        codigo_accidente,
        vehiculos_codigo_accidente
    ) VALUES (
        p_nombre,
        p_accidente_id,
        ,
        vehiculos_codigo_accidente
    ) VALUES (
        p_nombre,
        p_accidente_id,
        p_codigo_accidentado,
        p_fecha,
        p_condicion,
        p_estado,
        p_edad,
        p_sexo,
        p_vehiculo,
        p_codigo_accidente,
        p_vehiculos_codigo_accidente
    );
    COMMIT;
END insertar_actor_vial;
/

```

2. Procedimiento para actualizar la condición de un actor vial

```

CREATE OR REPLACE PROCEDURE actualizar_condicion_actor_vial(
    p_id_propietario IN INTEGER,
    p_nueva_condicion IN VARCHAR2
)
IS
BEGIN
    UPDATE actor_vial
    SET condicion = p_nueva_condicion
    WHERE id_propetario = p_id_propietario;
    COMMIT;
END actualizar_condicion_actor_vial;

```

5 Bases de Datos No-SQL (*Segunda entrega*)

5.1 Diagrama Bases de Datos No-SQL (*Segunda entrega*)

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

- 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*)
 - 6.2 Automatización de Datos** (*Tercera entrega*)
 - 6.3 Integración de Datos** (*Tercera entrega*)

7 Proximos pasos (*Tercera entrega*)

8 Lecciones aprendidas (*Tercera entrega*)

9 Bibliografía