TEC | Tecnológico de Costa Rica

Escuela de Administración de Tecnologías de la Información

Bases de Datos Avanzados - TI4601

IMPLEMENTACIÓN DE UN DATA WAREHOUSE PARA EL ANÁLISIS DE ACCIDENTES DE TRÁNSITO

Profesor: Michael Sanchez Soto

Anthony Montero Roman	2019275097
Daniel Andrés Rayo Díaz	2020158181
Kun Kin Zheng Liang	2022205015

Verano 2024-2025

Índice

Objetivo General

Diseñar e implementar un Data Warehouse a partir de un conjunto de datos transaccionales, utilizando un proceso completo de ETL (Extracción, Transformación y Carga) para optimizar la consulta y el análisis de datos agregados, facilitando la toma de decisiones estratégicas basadas en información estructurada y procesada.

Objetivos Específicos

- Seleccionar y analizar un conjunto de datos transaccionales, asegurándose de que provenga de una fuente sin depuración previa y que sea adecuado para el modelado de un Data Warehouse, con una cantidad mínima de registros de 150.000.
- Diseñar un modelo dimensional que incluya al menos dos tablas de hechos y cuatro tablas de dimensiones, definiendo correctamente claves primarias y foráneas para garantizar la integridad y eficiencia del sistema.
- Implementar un proceso ETL eficiente, que permita la extracción de datos desde la fuente seleccionada, su limpieza y transformación para alinearlos con el modelo dimensional, y su carga en la base de datos del Data Warehouse.
- Ejecutar consultas analíticas clave sobre el Data Warehouse para calcular indicadores de rendimiento (KPIs) y obtener insights relevantes que faciliten el análisis de patrones y tendencias en los datos.
- Documentar detalladamente el proceso, incluyendo la descripción del dataset, el diseño del modelo dimensional, las transformaciones aplicadas en el ETL, los resultados de las consultas analíticas y las instrucciones para ejecutar el proyecto.
- Preparar y presentar una exposición efectiva, explicando el proceso seguido, los principales resultados obtenidos y los desafíos enfrentados, utilizando material de apoyo visual adecuado para facilitar la comprensión del trabajo realizado.

Descripción del Dataset Seleccionado

Este conjunto de datos contiene información detallada sobre los conductores de vehículos involucrados en colisiones de tránsito en las carreteras locales y dentro del condado de Montgomery, Maryland. Estos datos son fueron obtenidos a través del Sistema de Informe Automatizado de Accidentes (ACRS) de la Policía Estatal de Maryland y son reportados por diversas agencias policiales, incluyendo la Policía del Condado de Montgomery, la Policía de Gaithersburg, la Policía de Rockville y la Policía de Maryland-National Capital Park.

El conjunto de datos documenta cada colisión registrada en estas áreas, proporcionando detalles sobre los incidentes, los conductores involucrados y otros factores relevantes. Sin embargo, es importante tener en cuenta que la información contenida en estos registros proviene de reportes preliminares entregados por las partes involucradas y los agentes que cubrieron la escena.

Cada fila representa un incidente y proporciona información sobre:

- Identificación del accidente (número de reporte, agencia responsable)
- Fecha, hora y ubicación (coordenadas, nombre de la calle, intersecciones)
- Condiciones ambientales (clima, iluminación, estado de la superficie)
- Factores humanos (distracciones, consumo de sustancias, culpabilidad del conductor)
- Vehículos involucrados (tipo, daños, dirección, movimiento en el momento del choque)
- Severidad del accidente (nivel de lesiones, tipo de colisión)

Este dataset permite analizar patrones de accidentes, identificar factores de riesgo y desarrollar estrategias para mejorar la seguridad vial.

Se ha realizado un proceso de limpieza y transformación para eliminar datos duplicados, corregir valores inconsistentes y manejar valores nulos, asegurando la calidad del dataset final.

Diagrama del Modelo Dimensional

Visión General del Esquema en Estrella

Para cubrir los dos niveles de análisis planteados en el proyecto, se han definido dos esquemas en estrella (Star Schemas) separados. Uno se centra en la tabla de hechos relacionada con el accidente en general, mientras que el otro se enfoca en la participación de cada vehículo en dichos accidentes. A continuación, se describe cada uno de manera general.

En la Figura 1, se ilustra el Esquema Crash, que incluye la tabla de hechos denominada FactCrash, y las dimensiones DimDateTime_Crash, DimLocation_Crash, DimCondition_Crash y DimCrashType. Por otra parte, en la Figura 2, se presenta el Esquema Vehicle, conformado por la tabla de hechos FactVehicleInvolment y las dimensiones DimDateTime_Veh, DimLocation_Veh, DimDriver y DimVehicle.

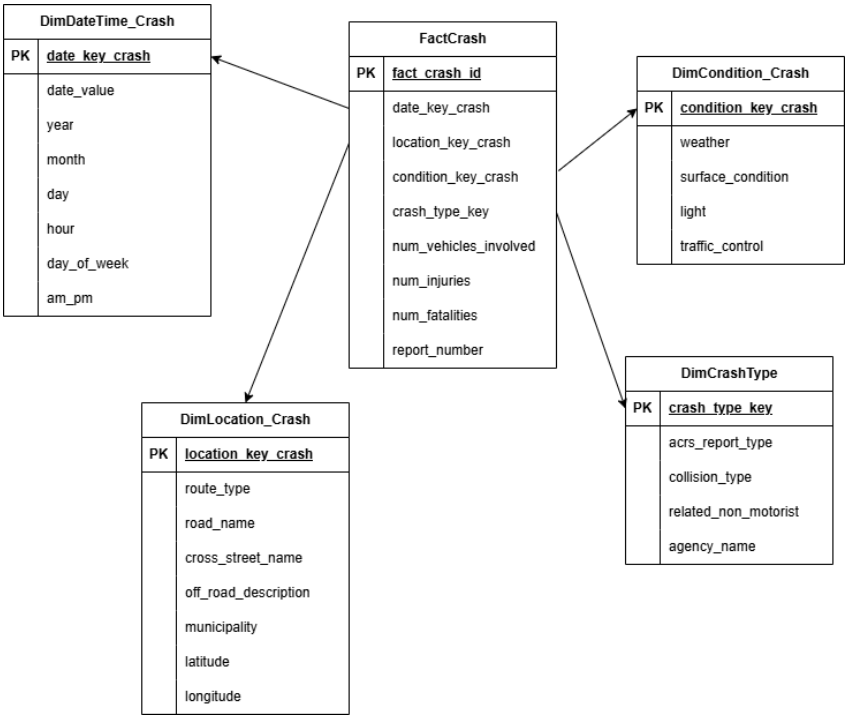

Figura 1: Esquema Estrella de Accidentes (Crash).

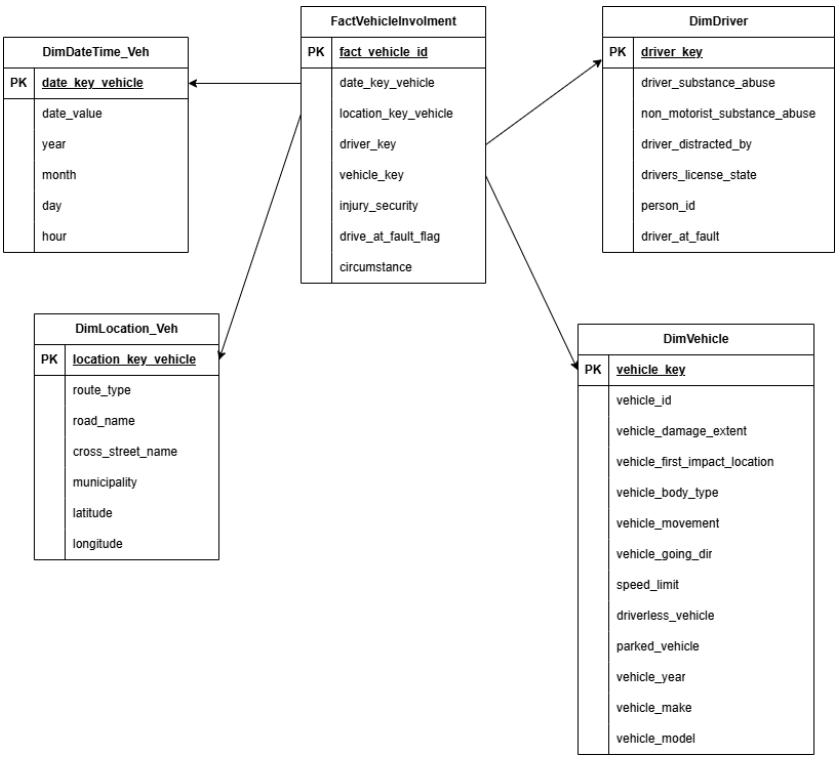

Figura 2: Esquema Estrella de Involucramiento Vehicular (Vehicle).

Descripción de las Tablas

A continuación se describen las tablas de hechos y tablas de dimensiones que conforman cada esquema, indicando sus atributos principales y el propósito de cada una.

1. **Esquema de Accidentes (Crash DW)**

a. **Tabla de Hechos: FactCrash**

Esta tabla almacena la información central de cada accidente, consolidada a partir del conjunto de datos transaccionales. Cada fila corresponde a un accidente único, identificado por la clave primaria fact_crash_id. Dentro de esta tabla, se incluyen los siguientes campos clave:

➔ Claves Foráneas:

- ◆ date_key_crash (relacionada a la dimensión de tiempo DimDateTime_Crash)

- location_key_crash (relacionada a la dimensión de ubicación DimLocation_Crash)
- condition_key_crash (relacionada a la dimensión de condiciones DimCondition_Crash)
- crash_type_key (relacionada a la dimensión de tipo de choque DimCrashType)

➔ Medidas:

- num_vehicles_involved (cantidad de vehículos asociados al accidente)
- num_injuries (número de personas lesionadas)
- num_fatalities (número de fallecimientos, si la información está disponible)

➔ Campo de referencia:

- report_number (identificador que proviene del sistema original, útil para rastrear el registro del accidente)

b. **DimDateTime_Crash**

La dimensión de fecha y hora facilita el análisis temporal de los accidentes, evitando repetir en la tabla de hechos toda la información asociada a la fecha. Su clave primaria es date_key_crash. Además, se incluyen los atributos year, month, day, hour, day_of_week y am_pm. Esto permite, por ejemplo, agrupar o filtrar accidentes por año, mes o franja horaria.

c. **DimLocation_Crash**

Para la gestión de la información geográfica del accidente, se definió la dimensión de localización con la clave primaria location_key_crash. Almacena atributos como route_type, road_name, cross_street_name, off_road_description, municipality, latitude y longitude. De esta forma, resulta sencillo hacer consultas geoespaciales o comparar accidentes por municipio y tipo de vía.

d. **DimCondition_Crash**

Los factores ambientales asociados a la ocurrencia de un accidente se concentran en esta dimensión. Su clave primaria es condition_key_crash. Incluye

atributos como weather (clima), surface_condition (estado de la carretera), light (condición de luz) y traffic_control (tipo de control de tráfico). Esta separación permite analizar, por ejemplo, cuántos accidentes ocurren bajo lluvia o en carreteras con hielo.

e. **DimCrashType**

En esta dimensión se clasifican los tipos de colisión y otros aspectos contextuales del choque, como la participación de no motorizados (peatones o ciclistas) y la agencia encargada de reportarlo. La clave primaria es crash_type_key, mientras que los atributos principales incluyen acrs_report_type, collision_type, related_non_motorist y agency_name.

2. Esquema de Involucramiento Vehicular (Vehicle DW)

a. Tabla de Hechos: FactVehicleInvolment

La segunda gran tabla de hechos, FactVehicleInvolment, almacena la información de cada vehículo que participó en un accidente. Cada fila registra los datos de un vehículo en particular asociado a un suceso, lo que permite un nivel de detalle mayor al de FactCrash. Su clave primaria es fact_vehicle_id, y en ella destacan:

➔ Claves Foráneas:
- ◆ date_key_vehicle (dimensión temporal DimDateTime_Veh)
- ◆ location_key_vehicle (dimensión de localización DimLocation_Veh)
- ◆ driver_key (dimensión del conductor DimDriver)
- ◆ vehicle_key (dimensión del vehículo DimVehicle)

➔ Medidas - Campos Relevantes:
- ◆ injury_security (nivel de lesión reportada en la persona que conducía o iba a bordo)
- ◆ drive_at_fault_flag (indicador de responsabilidad del conductor)
- ◆ circumstance (texto descriptivo de la circunstancia del vehículo)

b. DimDateTime_Veh

Esta dimensión cumple una función análoga a la de DimDateTime_Crash, pero enfocada a la perspectiva de cada vehículo involucrado. Tiene como clave primaria date_key_vehicle, y sus atributos se basan en la fecha y hora del evento (por ejemplo, year, month, day y hour).

c. DimLocation_Veh

De igual modo, DimLocation_Veh replica parte de la lógica de localización, pero aplicada específicamente al registro de cada vehículo. Se identifican atributos como route_type, road_name, cross_street_name, municipality, latitude y longitude, asociados a la clave primaria location_key_vehicle.

d. DimDriver

Para representar los factores humanos que intervienen en un accidente, se define la dimensión DimDriver, cuya clave primaria es driver_key. Aquí se almacenan campos como driver_substance_abuse, driver_distracted_by, drivers_license_state, driver_at_fault y person_id. Así, es posible agrupar accidentes según el estado de distracción del conductor o la presencia de sustancias.

e. DimVehicle

Por último, la dimensión de vehículo (clave primaria vehicle_key) incluye atributos como vehicle_id, vehicle_damage_extent, vehicle_body_type, vehicle_year, vehicle_make, vehicle_model y otros indicadores (por ejemplo, parked_vehicle o driverless_vehicle). Esta información permite ver la composición del parque vehicular involucrado en los choques, así como la relación entre ciertos tipos de vehículos y el grado de daño.

Justificación del Diseño

El punto de partida de este diseño en estrella surge de la necesidad de realizar análisis OLAP (Online Analytical Processing) sobre un gran volumen de datos de accidentes, de manera ágil y flexible. Para ello, se han definido dos tablas de hechos

principales—FactCrash y FactVehicleInvolment—que responden a la diferencia esencial entre estudiar la ocurrencia global del accidente (esquema Crash) y el detalle de cada vehículo y conductor (esquema Vehicle).

En primer lugar, se busca la simplicidad en las consultas. Al extraer la información clave (fecha, ubicación, condiciones ambientales, tipo de choque) a dimensiones específicas, se evitan repeticiones y se facilita la ejecución de agregaciones, por ejemplo, el número de accidentes por mes o la relación entre siniestros y clima. Esta estructura clara y ordenada hace más sencillo tanto el acceso como la interpretación de los datos.

Asimismo, el modelo prioriza la escalabilidad. Al contar con dos tablas de hechos, es posible incluir nuevos atributos de vehículos o conductores en las dimensiones DimVehicle o DimDriver sin verse obligado a modificar la estructura de la tabla FactCrash. Gracias a ello, el sistema puede crecer y adaptarse a futuras necesidades, ampliando progresivamente la cobertura de datos sin romper el diseño inicial.

Otro aspecto relevante es la claridad en el análisis de factores humanos y del tipo de choque. Las dimensiones DimDriver y DimVehicle permiten detallar aspectos como distracciones al volante, uso de sustancias, año de fabricación del vehículo y tipo de carrocería. Esta información enriquece los análisis al ayudar a formular hipótesis de riesgo o tendencia (por ejemplo, si ciertas marcas o modelos son más propensos a colisiones), lo cual contribuye a la toma de decisiones basadas en datos.

Por último, se refuerza la independencia lógica al mantener separadas las dimensiones de fecha y hora para Crash (DimDateTime_Crash) y para cada vehículo (DimDateTime_Veh). Aunque ambas comparten la misma naturaleza (fechas y horas), la separación evita conflictos en el proceso de ETL y en las claves primarias. Una estrategia similar se sigue con las dimensiones de localización, con el fin de garantizar la coherencia y la flexibilidad de la arquitectura a largo plazo.

Descripción de las Transformaciones Aplicadas y del Proceso ETL

1. Extracción de los Datos

La primera etapa de este proceso consistió en extraer la información de un archivo CSV donde se detalla cada accidente reportado, junto con la información relevante del lugar, fecha, condiciones ambientales, vehículos y conductores implicados. Para ello, se empleó la librería pandas de Python, aprovechando la función pd.read_csv(). Esta función se configuró cuidadosamente mediante:

- Un diccionario de tipos de datos (dtype) que forzaba a leer ciertas columnas como texto, otras como enteros (Int64) o flotantes, y así evitar conversiones inadecuadas.
- Un conjunto de valores nulos (na_values) para identificar adecuadamente celdas vacías o inconsistentes, incluyendo “N/A”, “UNKNOWN” y variantes similares.
- La directiva keep_default_na=False, con el fin de desactivar los valores nulos por defecto y tener un control total sobre cuáles cadenas se interpretarían como “missing”.

Como resultado, se obtuvo un DataFrame de pandas con la información cargada, pero aún en bruto, lista para ser limpiada y normalizada en la siguiente fase del proceso.

2. Transformación y Limpieza

La transformación se inició con la definición de varias funciones escritas en Python para depurar y homogeneizar cada columna. Por ejemplo, clean_string_value() elimina espacios en blanco innecesarios y asigna cadenas vacías en lugar de valores nulos. También se diseñaron clean_numeric_value() y clean_vehicle_year() para asignar valores numéricos válidos (como cero) cuando se encuentre texto inválido, o para descartar años de fabricación de vehículo fuera de un rango razonable (por debajo de 1900 o más allá del año actual). Para las columnas que manejaban información booleana —como “Driverless Vehicle” o “Parked

Vehicle”— se empleó la función normalize_boolean(), la cual transformaba valores dispares (Yes, True, 1) a la notación estándar “Y”, y los equivalentes de No a “N”.

Estas reglas de limpieza se aplicaron columna por columna al DataFrame, asegurando así la uniformidad de los datos. De forma opcional, se consideró el parseo de la fecha y hora original con clean_datetime() para convertir las cadenas, como “05/27/2021 07:40:00 PM”, en objetos datetime de Python; sin embargo, en este caso se decidió dejar la columna como cadena y emplear dicha información para generar claves de fecha y hora (date_key) en el momento de la carga.

3. Carga en PostgreSQL

Tras la transformación, se procedió a cargar los datos en dos bases de datos, crashDW y vehicleDW, alojadas en PostgreSQL. Para ello, se utilizó la librería psycopg2, que permite crear conexiones y ejecutar sentencias SQL. En primer lugar, se definió la función create_database_if_not_exists() para verificar si cada base de datos existía y, en caso contrario, crearla automáticamente. Luego, get_db_connection() se encargó de establecer la conexión (y un cursor) con dichas bases de datos.

Una vez conectados a crashDW y vehicleDW, se ejecutaron sentencias CREATE TABLE IF NOT EXISTS para crear las tablas del modelo dimensional. En la primera base se incluyeron DimDateTime_Crash, DimLocation_Crash, DimCondition_Crash, DimCrashType y la tabla de hechos FactCrash. En la segunda, se definieron DimDateTime_Veh, DimLocation_Veh, DimDriver, DimVehicle y la tabla de hechos FactVehicleInvolment.

Para insertar la información, se empleó un mecanismo de diccionarios en memoria que prevenía duplicaciones en las dimensiones. Cada columna o conjunto de columnas único en las dimensiones se insertaba solo si no existía un registro igual, devolviendo la clave primaria que luego se usaba en la tabla de hechos. De esta manera, se mantenía la unicidad y no se desperdiciaban registros repetidos.

En cuanto a la tabla de hechos FactCrash, los registros se agregaron agrupando los datos del CSV por “Report Number”. Allí se contaron los vehículos involucrados y las lesiones por cada accidente. Para la tabla de hechos

FactVehicleInvolment, se recorrió cada fila del CSV insertando detalles de cada vehículo, junto con sus llaves foráneas a las dimensiones de fecha, localización, conductor y vehículo, de modo que se tuviera un nivel de detalle mucho mayor sobre la participación de cada unidad.

4. Herramientas Utilizadas

- Python 3.11.0: Se empleó como lenguaje principal para orquestar todo el proceso, desde la lectura del CSV hasta la inserción de datos en PostgreSQL.
- pandas: Fundamental para la fase de Extracción y Transformación, dado que simplifica enormemente la lectura, filtrado y limpieza de datos en formatos tabulares.
- psycopg2: Librería que permite comunicarse con PostgreSQL, ejecutar sentencias SQL y controlar las conexiones a la base de datos.
- PostgreSQL 17.2: Escogido por su robustez, manejo eficiente de integridad referencial y su idoneidad para alojar un Data Warehouse de tamaño moderado o grande.

Resultados de las Consultas Analíticas

Esta sección presenta el análisis y la interpretación de las consultas SQL ejecutadas sobre el Data Warehouse de accidentes de tráfico. Cada consulta se diseñó para responder preguntas clave sobre la ocurrencia, severidad y factores asociados a los accidentes, proporcionando información valiosa para la toma de decisiones en seguridad vial.

Los resultados obtenidos permiten identificar patrones temporales, evaluar el impacto de las condiciones climáticas, analizar la severidad de los incidentes y determinar tendencias relacionadas con el tipo de colisión y los vehículos involucrados. Estos hallazgos pueden ser utilizados por entidades de tránsito, aseguradoras y organismos de planificación para mejorar estrategias de prevención y respuesta ante accidentes.

A continuación, se detallan los resultados y su aplicación en cada uno de los análisis realizados.

1. Análisis Temporal: Accidentes por Año y Mes

Esta consulta tiene como objetivo realizar un análisis temporal de la frecuencia de accidentes en función del año y mes en que ocurrieron. Utiliza la tabla de hechos FactCrash, que contiene información sobre los accidentes, y la tabla de dimensiones DimDateTime_Crash, que proporciona los datos relacionados con la fecha y hora de los incidentes. Al agrupar los accidentes por año y mes, la consulta proporciona una visión clara de la evolución temporal de los accidentes a lo largo del tiempo.

El resultado de esta consulta permite identificar patrones estacionales y tendencias a largo plazo en la frecuencia de accidentes. Por ejemplo, puede ser útil observar si hay meses con un aumento significativo de accidentes, lo que podría estar relacionado con factores estacionales como el clima o el tráfico. Además, este análisis puede ayudar a identificar años con un número anómalo de accidentes, lo que puede indicar situaciones excepcionales o la necesidad de revisar políticas de seguridad vial implementadas en esos años.

TotalCrashes	Mes												
Año	1	2	3	4	5	6	7	8	9	10	11	12	Total general
2015	906	873	824	835	1022	894	832	879	993	1119	1049	1104	11330
2016	993	822	901	933	1155	974	984	959	1074	1123	1047	1068	12033
2017	965	854	967	944	1047	1009	930	970	1050	1123	1061	1105	12025
2018	1034	803	904	860	1039	940	969	928	1077	1133	1080	1061	11828
2019	926	822	933	932	1064	958	963	874	967	1118	997	1049	11603
2020	942	920	649	370	455	570	617	702	677	724	686	711	8023
2021	564	504	636	714	787	781	847	783	911	977	898	899	9301
2022	728	711	829	804	836	812	785	777	853	974	925	964	9998
2023	826	790	878	860	941	870	820	843	969	981	1002	942	10722
2024	854	755	872	879	952	883	841	845	958	964	862	858	10523
2025	461												461
Total general	9199	7854	8393	8131	9298	8691	8588	8560	9529	10236	9607	9761	107847

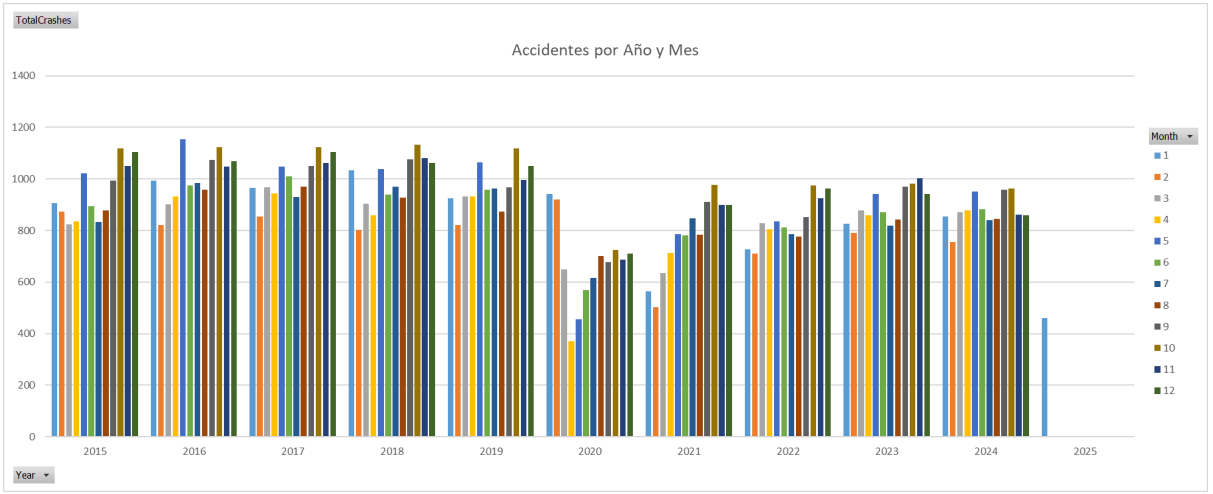

Figura 3: Resultados del Análisis Accidentes por Año y Mes.

El uso de estos resultados puede ser clave para la planificación de recursos y la toma de decisiones en cuanto a la asignación de patrullas de tránsito o la implementación de campañas de seguridad en momentos de mayor incidencia de accidentes. Además, esta consulta proporciona información relevante para las autoridades viales al evaluar la efectividad de las políticas y acciones preventivas en diferentes períodos. Con base en los resultados, también se pueden desarrollar estrategias de concienciación y prevención dirigidas a los meses con mayor riesgo de accidentes, contribuyendo así a mejorar la seguridad vial.

2. Análisis de Severidad por Día de la Semana

La consulta "Análisis de Severidad por Día de la Semana" se centra en evaluar cómo varía la severidad de los accidentes según el día de la semana en que ocurren. Utilizando las tablas FactCrash (que contiene datos sobre los accidentes) y

DimDateTime_Crash (que proporciona información sobre la fecha y hora de los incidentes), la consulta calcula el número total de lesiones ocurridas en cada día de la semana. Además, ofrece el porcentaje de lesiones respecto al total de lesiones ocurridas, lo que permite ver de forma más clara el impacto relativo de los accidentes según el día.

El análisis de los resultados ayuda a identificar posibles patrones en la gravedad de los accidentes a lo largo de la semana. Por ejemplo, es posible que los accidentes ocurran con mayor frecuencia y/o severidad durante los días laborales, cuando el tráfico es más denso y los comportamientos de conducción pueden verse influenciados por factores como el estrés o la fatiga. También puede evidenciarse un aumento de accidentes graves durante los fines de semana, cuando las actividades recreativas o el consumo de alcohol pueden ser factores influyentes.

Week	Injuries	Percentage %
Friday	5533	15,7
Thursday	5484	15,56
Tuesday	5433	15,41
Wednesday	5370	15,24
Monday	5196	14,74
Saturday	4379	12,42
Sunday	3850	10,92

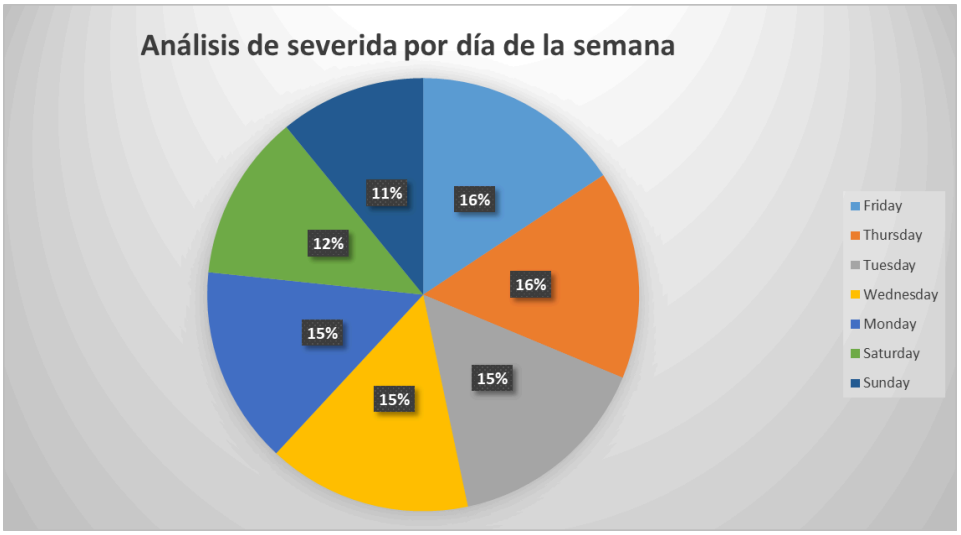

Figura 4: Resultados del Análisis de Severidad por Día de la Semana.

3. Análisis de Clima y Tipo de Colisión

La consulta "Análisis de Clima y Tipo de Colisión" tiene como objetivo correlacionar las condiciones climáticas con los tipos de colisiones ocurridas, proporcionando información valiosa sobre cómo el clima puede influir en la naturaleza de los accidentes. Para llevar a cabo este análisis, se utilizan las tablas FactCrash (que contiene los registros de accidentes), DimCrashType (que categoriza los tipos de colisiones) y DimCondition_Crash (que describe las condiciones climáticas y otros factores en el momento del accidente).

La consulta muestra la cantidad de accidentes registrados para cada combinación de tipo de colisión y condición climática, ordenándolos según la cantidad de accidentes. Esto permite identificar qué condiciones climáticas están asociadas con los tipos de colisiones más frecuentes. Por ejemplo, es posible observar si los accidentes por colisiones traseras son más comunes en días lluviosos o si las condiciones de visibilidad reducida están relacionadas con choques en intersecciones.

Colisión	Clima	# de accidente
SAME DIR REAR END	CLEAR	17742
STRAIGHT MOVEMENT ANGLE	CLEAR	10341
SINGLE VEHICLE	CLEAR	9721
OTHER	CLEAR	8889
SAME DIRECTION SIDESWIPE	CLEAR	6278
HEAD ON LEFT TURN	CLEAR	4345
SAME DIR REAR END	RAINING	3073
SINGLE VEHICLE	RAINING	2601
SAME DIR REAR END	CLOUDY	2533
Front to Rear	Clear	2167

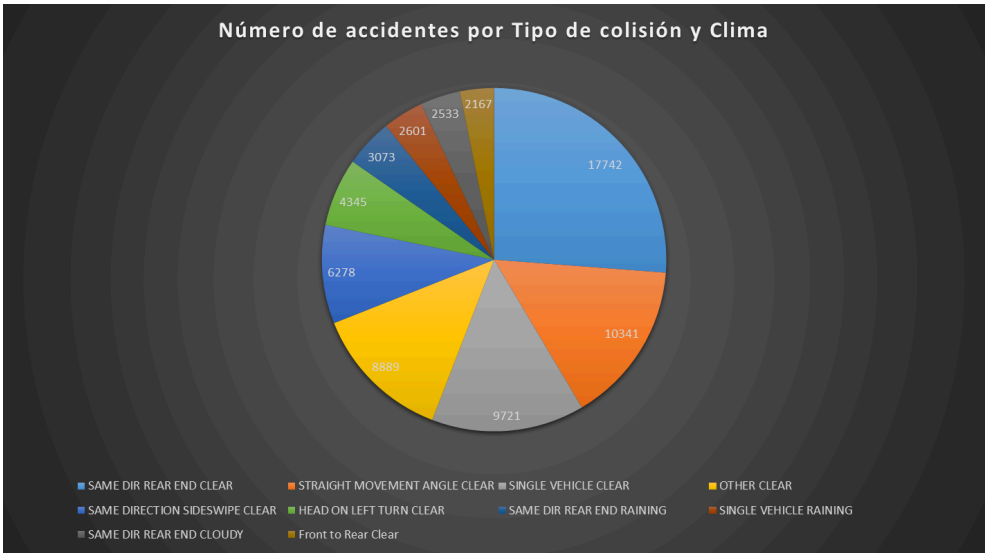

Figura 5: Resultados del Análisis de Clima y Tipo de Colisión.

4. Análisis de Vehículos por Tipo de Colisión

La consulta "Análisis de Vehículos por Tipo de Colisión" tiene como propósito evaluar la relación entre los diferentes tipos de colisión y la cantidad promedio de vehículos involucrados en cada tipo. Esta información se obtiene utilizando las tablas FactCrash, que contiene los registros de accidentes, y DimCrashType, que categoriza los tipos de colisiones. El análisis se enfoca en calcular el número promedio de vehículos involucrados en cada tipo de accidente.

El resultado de la consulta muestra los tipos de colisiones más frecuentes y el número promedio de vehículos que suelen estar involucrados en cada uno de estos accidentes. Este análisis proporciona una visión crucial sobre la magnitud de los accidentes, ya que algunos tipos de colisiones, como las colisiones en cadena o los accidentes múltiples, involucran a un mayor número de vehículos, mientras que otros, como los choques frontales o laterales, pueden implicar solo a un vehículo o un par de vehículos.

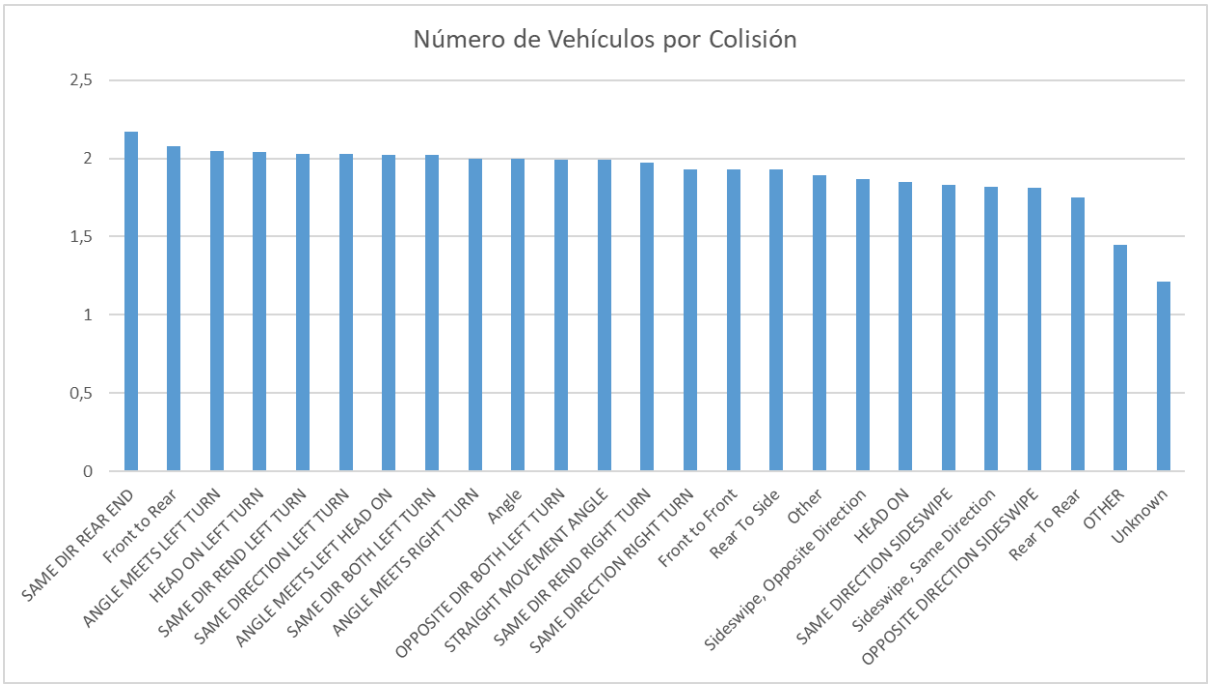

Figura 6: Resultados del Análisis de Vehículos por Tipo de Colisión.

Los resultados pueden ser utilizados por diferentes actores dentro del ámbito de la seguridad vial, como los servicios de emergencia, autoridades de tránsito y diseñadores de políticas de seguridad. Por ejemplo, los datos de accidentes con más vehículos involucrados pueden ayudar a mejorar la planificación de recursos y la logística de emergencia, asegurando que haya suficiente personal y equipos disponibles para hacer frente a accidentes más complejos.

5. Análisis de Vehículos Involucrados en Accidentes

La consulta "Análisis de Vehículos Involucrados en Accidentes" está diseñada para identificar las marcas y modelos de vehículos que se encuentran más frecuentemente involucrados en accidentes. Este análisis proporciona información crucial sobre la relación entre los vehículos y su participación en incidentes viales, lo que puede ser de utilidad para fabricantes de automóviles, aseguradoras y organismos reguladores.

La consulta se realiza utilizando los datos de las tablas DimVehicle y FactVehicleInvolment. En DimVehicle, se almacenan los detalles relacionados con los vehículos, como la marca (vehículo_make) y el modelo (vehicle_model). Por otro lado, FactVehicleInvolment registra la relación de cada vehículo con los accidentes, proporcionando un identificador único para cada participación en un accidente.

Etiquetas de fila	Suma de # de accidentes
FORD	**3554**
EXPLORER	1636
TK	1918
HONDA	**13783**
ACCORD	5612
CIVIC	5780
CRV	2391
NISSAN	**2457**
ALTIMA	2457
TOYOTA	**14179**
CAMRY	5998
COROLLA	5452
RAV4	2729
TOYT	**2472**
4S	2472
Total general	**36445**

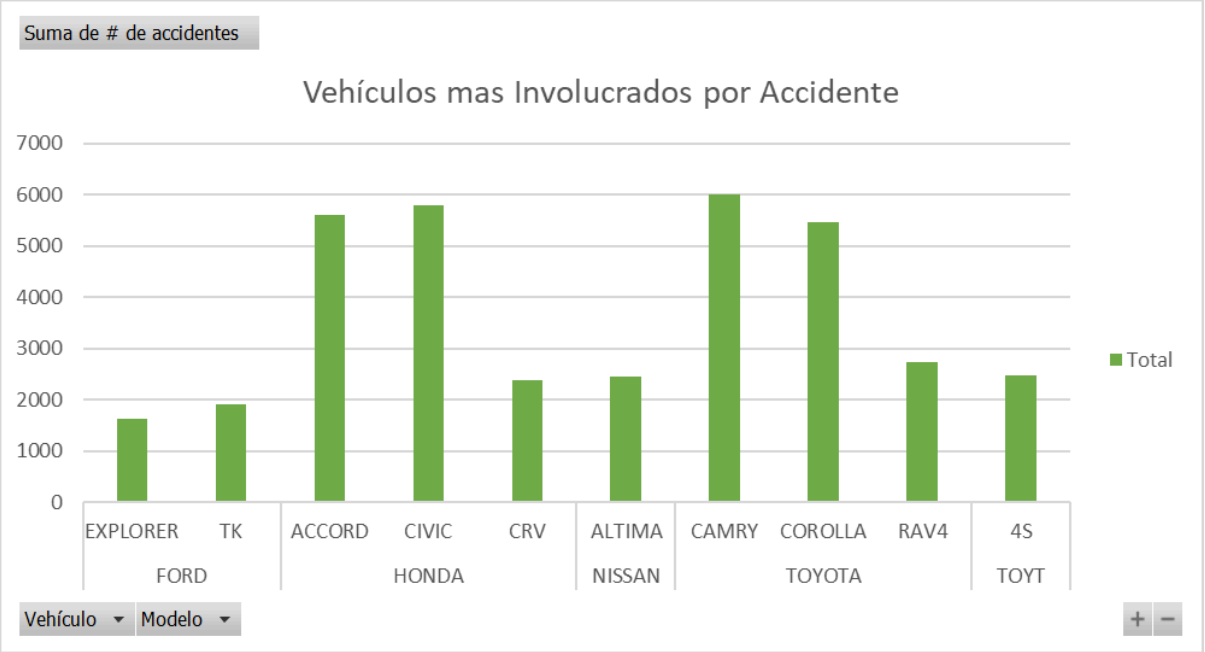

Figura 7: Resultados del Análisis de Vehículos Involucrados en Accidentes.

A través de esta consulta, se obtiene un conteo de las participaciones de cada vehículo en accidentes, agrupado por marca y modelo. Los resultados de este análisis son valiosos para identificar qué tipos de vehículos están más involucrados en accidentes, lo que podría sugerir posibles patrones de riesgo asociados con ciertas marcas o modelos. Estos patrones podrían ser útiles para las aseguradoras a la hora de ajustar primas, para los fabricantes en términos de seguridad vehicular, o incluso para los reguladores al identificar qué tipos de vehículos podrían necesitar revisiones o mejoras en sus características de seguridad.

Instrucciones para Ejecutar el Proyecto

Requisitos Previos

1. PostgreSQL instalado en su sistema
2. Python 3.x instalado
3. Acceso a internet para descargar el repositorio
4. Permisos de administrador para crear bases de datos

Pasos de Instalación

1. Obtener el Código Fuente

1. Visite el repositorio en Github: https://github.com/DARD172002/data-warehouse/tree/master
2. Descargue el archivo .zip del proyecto
3. Extraiga el contenido del archivo zip en una ubicación de su preferencia

2. Configuración del Entorno

1. Navegue hasta la carpeta 'project' dentro de los archivos extraídos
2. Localice el archivo dataWarehouse.py
3. Abra el archivo con un editor de texto o IDE de su preferencia

3. Configuración de la Base de Datos

1. Abra el archivo dataWarehouse.py
2. Localice la sección DB_CONFIG
3. Modifique los siguientes parámetros según su configuración local:
    - user: Su nombre de usuario de PostgreSQL
    - password: Su contraseña de PostgreSQL
    - host: La dirección del servidor (normalmente 'localhost')
    - port: El puerto de PostgreSQL (por defecto 5432)

4. Configuración del Dataset

1. En el mismo archivo dataWarehouse.py

2. Localice la variable csv_path
3. Asegúrese de que la ruta apunte correctamente al archivo CSV que desea procesar
4. Si es necesario, modifique la ruta para que coincida con la ubicación de su archivo CSV

5. Ejecución del Proyecto

1. Abra una terminal o línea de comandos
2. Navegue hasta la carpeta donde se encuentra dataWarehouse.py
3. Ejecute el script con el comando “python dataWarehouse.py”

6. Verificación

1. Abra pgAdmin o su cliente PostgreSQL preferido
2. Verifique que se hayan creado las siguientes bases de datos:
    - crashDW
    - vehicleDW
3. Examine las tablas creadas en cada base de datos
4. Ejecute algunas consultas de prueba para verificar la correcta carga de datos