



## ***Proyecto Final - SQL***

**Comisión 47365**

**Profesor Camilo Andres Rodondo**

**Tutor asignado Nicolás Javier Olivera**

**Alumno Agustín Del Río**

**Año Lectivo 2023**

# Índice

02 - Introducción

02 - Objetivos del Proyecto

03 - Herramientas utilizadas

03 - Composición de la carpeta del proyecto

03 - Scripts

04 - Scripts : 0. Creación de base de datos y tabla de hecho

05 - Scripts : 1. Creación de tabla madre y tablas hijas ( Normalización

06 - Vistas

07 - Funciones

07 - Procedimientos Almacenados

08 - Triggers

09 - DML + TCL

10 - Back Up

10 - Creacion de usuarios

11 - Diagrama Entidad Relación

12 - Conclusion - Comentarios Finales

## Introducción

El presente proyecto tiene como objetivo principal realizar un análisis detallado y comparativo de los precios de autos usados, centrándose en diferentes marcas y modelos. En un mercado automotriz dinámico y diverso, el estudio de los precios de vehículos usados despierta un interés significativo tanto para los consumidores como para la industria en general. Este análisis comparativo se basa en datos recopilados y almacenados en una base de datos gestionada en MySQL.



## Objetivos del Proyecto

El proyecto se enfoca en los siguientes objetivos (3):

**Recopilación y almacenamiento de datos** Se han recopilado datos sobre precios de autos usados de diversas fuentes confiables para su almacenamiento y gestión en una base de datos MySQL.

**Análisis comparativo:** Se llevará a cabo un análisis exhaustivo de los precios de autos usados, desglosados por marcas y modelos, con el fin de identificar tendencias, variaciones y relaciones entre los diferentes segmentos del mercado.

**Presentación de resultados:** Los hallazgos obtenidos se presentarán de manera clara y visualmente atractiva a través de consultas SQL, análisis estadísticos y gráficos, con el objetivo de brindar una comprensión profunda y accesible de los datos.

## Herramientas Utilizadas

MySQL para consultas de base de datos, visualización de datos PowerBI.



## Composición de la carpeta

El archivo consta de 9 scripts , 1 diagrama , un instructivo de instalación y el dataset ( plantilla en CSV donde estan los datos primitivos a analizar)

El instructivo de instalación consiste en un archivo de bloc de notas donde está el orden de ejecución de los scripts de mysql para generar la base de datos , las tablas , vistas , funciones, procedimientos almacenados, disparadores, backup y creación de usuarios.

El diagrama ya esta realizado, solo tiene que abrir el archivo "Diagrama Entidad Relacion.mwb" y visualizar todos los objetos de la base de datos.

## Metodología utilizada

Se creará una tabla de hecho coches, donde a partir de esa tabla principal se hará la ingesta de datos del dataset en csv donde se encuentran los datos primitivos. Luego a partir de la tabla coches se crearán la tabla madre auto y las tablas hijas, con todos los registros relacionados.

---

## Scripts

A continuación se describirán los objetos que se irán formando a medida que se ejecuten los scripts.

### 0 . Creación de base de datos y tabla de hecho

Este script SQL tiene como objetivo crear una base de datos llamada "proyecto" y una tabla principal llamada "COCHES" para almacenar datos relacionados con autos usados. Aquí está una explicación detallada:

#### **Creación del esquema (schema) proyecto:**

- DROP SCHEMA proyecto;: Esta línea eliminará el esquema (base de datos) llamado "proyecto" si existiera previamente. Este comando eliminará todos

los objetos (tablas, vistas, procedimientos almacenados, etc.) dentro del esquema "proyecto".

- `CREATE SCHEMA proyecto;` Luego, se crea un nuevo esquema (base de datos) llamado "proyecto" utilizando el comando `CREATE SCHEMA`.

#### **Uso del esquema proyecto:**

- `USE proyecto;` Esta instrucción cambia el contexto de trabajo al esquema recién creado "proyecto", lo que significa que todos los comandos SQL posteriores se ejecutarán en este esquema.

#### **Creación de la tabla COCHES:**

- Se crea una tabla llamada "COCHES" que actúa como la tabla principal ("tabla de hecho") para almacenar datos relacionados con autos usados.
- La estructura de la tabla incluye columnas como `ID_COCHE`, `REGION`, `PRECIO`, `ANIO`, `MARCA`, `MODELO`, entre otros, cada una con su respectivo tipo de datos.

#### **Ingesta de datos en la tabla COCHES:**

- Utilizando el comando `LOAD DATA INFILE`, se carga o ingiere datos desde un archivo CSV ubicado en la ruta `"C:/Users/Agdel/Downloads/SQL/Coder/Bloque 12 - Entrega Trabajo final/dataset/Dataset_Coches1.csv"`.
- Se establece que el archivo CSV está delimitado por comas y cada línea está terminada por un salto de línea.
- La cláusula `IGNORE 1 LINES` indica que la primera línea del archivo CSV debe ser ignorada, asumiendo que contiene encabezados.

Este script crea una nueva base de datos llamada "proyecto", crea una tabla "COCHES" para almacenar datos relacionados con autos usados y carga información desde un archivo CSV específico a esta tabla para su posterior análisis y manipulación dentro del esquema "proyecto".

# 1 . Creación de tabla madre y tablas hijas ( Normalización)

Este script SQL realiza una serie de acciones para estructurar la tabla principal AUTO a partir de la tabla de hecho COCHES y, a su vez, crea tablas secundarias normalizadas utilizando los datos de la tabla AUTO. Aquí está la explicación:

Creación de la tabla AUTO a partir de COCHES:

- Se crea la tabla AUTO utilizando la estructura y datos de la tabla de hecho COCHES. Los datos se seleccionan y se insertan directamente en AUTO desde COCHES.

Ajustes en la estructura de la tabla AUTO:

- Utilizando ALTER TABLE, se modifican las columnas de AUTO para ajustar sus tipos de datos y restricciones, como modificar el tamaño de ciertas columnas, cambiar el tipo de dato, establecer claves primarias, entre otros.

Creación de tablas normalizadas:

- Se crean varias tablas normalizadas (COCHE\_TIPO, TIPO\_TRANSMISION, COLOR, FECHA\_PUBLICACION, MARCA, TIPO\_COMBUSTIBLE, UBICACION) a partir de la tabla AUTO. Cada una de estas tablas extrae información específica de AUTO para evitar redundancias y mantener la base de datos normalizada.
- Para cada tabla normalizada, se agrega una restricción de clave foránea (FOREIGN KEY) que hace referencia a la columna ID\_COCHE en la tabla AUTO.

Eliminación de columnas redundantes:

- Se eliminan las columnas de AUTO que se han normalizado y se han movido a las tablas secundarias para evitar redundancia y mantener la normalización de la base de datos.

Descripción de la tabla AUTO:

- Al final del script, se muestra la descripción actualizada de la tabla AUTO utilizando el comando DESCRIBE, lo que proporciona información sobre la estructura final de la tabla después de las modificaciones y normalizaciones.

Este script transforma la tabla de datos inicial COCHES en una estructura más normalizada y eficiente, dividiendo la información en tablas secundarias y eliminando columnas redundantes de la tabla principal AUTO. Esto ayuda a optimizar la base de datos, reducir la redundancia de datos y mejorar la consistencia en el esquema de la base de datos.

## Vistas

Las vistas en este proyecto ofrecen una forma simplificada y estructurada de acceder a la información almacenada en la base de datos "proyecto". Estas vistas fueron diseñadas para proporcionar diferentes perspectivas de los datos y facilitar consultas específicas sin la necesidad de escribir consultas complejas repetidamente.

### VW\_VISTA\_MAS\_BARATOS

Esta vista muestra los vehículos cuyo precio es inferior a \$5000. Proporciona detalles sobre el modelo, precio, descripción, año y kilometraje de los vehículos más económicos.

### VW\_VISTA\_COLORES\_DISPONIBLES

Ofrece una lista de colores de pintura disponibles en los vehículos almacenados, mostrando una lista distinta de colores sin repetición.

### VW\_VEHIC\_POR\_ANIO

Proporciona un recuento de vehículos disponibles por año de fabricación, lo que ofrece una visión general del número de modelos disponibles en cada año.

### Vista\_AutoPorAnio

Muestra la cantidad de vehículos y el precio promedio agrupados por año de fabricación, proporcionando una perspectiva del volumen y el precio promedio de los vehículos por año.

### Vista\_AutoPorModelo

Ofrece una visión general del número de vehículos y el kilometraje promedio agrupados por modelo de vehículo.

Estas vistas permiten analizar rápidamente datos específicos y proporcionan resúmenes útiles para entender mejor la información almacenada en la base de datos "proyecto".

---

## Funciones

Estas funciones fueron creadas para realizar tareas específicas dentro del proyecto de base de datos "proyecto".

`contar_modelos_por_marca(marca_param VARCHAR)`

Esta función cuenta el número de modelos asociados a una marca específica en la tabla "auto".

`obtener_modelo_mas_caro()`

Ofrece el modelo más costoso almacenado en la tabla "auto", proporcionando información sobre el modelo con el precio más elevado.

Estas funciones permiten realizar consultas específicas y obtener información útil sobre modelos y precios dentro de la base de datos "proyecto".

---

## Procedimientos Almacenados

Estos procedimientos almacenados tienen la finalidad de realizar operaciones específicas en la base de datos "proyecto".

`ACTUALIZAR_PRECIO (id_coche bigint, nuevoPrecio decimal)`

Este procedimiento actualiza el precio de un vehículo con un valor nuevo y registra el cambio en la tabla "REGISTRO\_PRECIOS".

`SP_MODELO_POR_RANGO(PRECIO_MIN INT, PRECIO_MAX INT)`

Al ingresar dos límites de valores, este procedimiento muestra los modelos de vehículos disponibles dentro de ese rango de precios.

`sp_update_precios(variacion_p float)`

Realiza una actualización en porcentaje (x%) de los precios de todos los vehículos almacenados en la tabla "auto".

`sp_seleccionar_por_color(p_color varchar)`

Selecciona vehículos por color, mostrando detalles como año, modelo, kilometraje y precio basado en el color especificado.

`sp_InsertarRegistroAuto(p_id_coche bigint, p_modelo VARCHAR, p_precio INT, p_anio INT)`

Inserta un nuevo registro en la tabla "auto" con los datos proporcionados: ID del coche, modelo, precio y año.



Estos procedimientos proporcionan funcionalidades específicas y operaciones útiles para gestionar la información de vehículos dentro de la base de datos "proyecto".

---

## Triggers ( disparadores)

Este script de SQL contiene una serie de comandos que están creando tablas de auditoría y triggers (disparadores) para registrar ciertas acciones en estas tablas de auditoría. Aquí tienes una explicación breve de cada parte:

Creación de la tabla log\_auditoria\_1:

- Esta tabla se crea para registrar acciones relacionadas con la tabla proyecto.auto. Guardará información como el tipo de acción (nombre\_accion), nombre de la tabla (nombre\_tabla), usuario que realizó la acción (user\_), y la fecha de la acción (fecha\_accion).

Trigger trg\_insert\_new:

- Este trigger se activa después de realizar una inserción en la tabla proyecto.auto.
- Registra en la tabla de auditoría log\_auditoria\_1 que se ha realizado una inserción ('INSERT') en la tabla auto, registrando el usuario actual y la fecha y hora de la acción.

Creación de la tabla log\_reg\_eliminados y trigger log\_reg\_eliminados:

- Esta tabla se crea para registrar eliminaciones de registros en la tabla proyecto.coches.
- El trigger log\_reg\_eliminados se activa antes de una eliminación en la tabla proyecto.coches. Guarda información sobre el registro eliminado en la tabla log\_reg\_eliminados.

Creación de la tabla log\_reg\_eliminados\_color y trigger log\_color\_eliminado:

- Esta tabla se utiliza para registrar eliminaciones de registros en la tabla proyecto.color.
- El trigger log\_color\_eliminado se activa antes de eliminar registros de la tabla proyecto.color y registra la acción en log\_reg\_eliminados\_color.

Este script establece tablas de auditoría y triggers que registran acciones específicas (inserciones y eliminaciones) en algunas tablas del esquema proyecto. Los triggers capturan estas acciones y registran información relevante en las tablas de auditoría para fines de seguimiento y control de cambios.

---

## DML + TCL

Este conjunto de comandos realiza y simula cambios en los datos de la tabla "auto", mostrando cómo se pueden agrupar estos cambios, revertirlos o confirmarlos, todo dentro de una transacción controlada.

Configuración del entorno:

- Se selecciona el "proyecto" como el espacio de trabajo.
- Se revisa y configura cómo se confirma la información después de cambios en la base de datos.

Exploración de datos:

- Se echa un vistazo rápido a algunos datos de la tabla "auto" para verificar y mostrar algunos registros.

Control de transacciones:

- Se inicia una transacción para agrupar ciertos cambios juntos.
- Se realizan una serie de inserciones, actualizaciones y eliminaciones en la tabla "auto" para simular cambios en los datos.
- Se utilizan puntos de guardado (savepoint) para marcar ciertos estados durante las transacciones.
- Se verifica la información ingresada y se revierten algunos cambios usando rollback para regresar a puntos anteriores.

Confirmación de transacciones:

- Se realizan más inserciones de datos.
- Se utiliza commit para confirmar todos los cambios realizados en la transacción, haciendo que los cambios sean permanentes y visibles para todos.

Restauración de configuración:

- Se restauran las verificaciones de clave foránea (foreign\_key\_checks) a su estado original para asegurar la integridad de los datos.

## Backup de la Base de Datos

El proceso de backup se refiere a la creación de una copia de seguridad de la base de datos "proyecto", lo que permite guardar una versión exacta de los datos en un momento específico. Esto se realiza para resguardar la integridad y disponibilidad de la información en caso de pérdida de datos, fallos del sistema o necesidades de restauración.

El archivo de backup contiene la estructura y los datos de las tablas almacenadas en la base de datos. Al utilizar esta copia de seguridad, es posible restaurar la base de datos a un estado previo en el tiempo, asegurando la continuidad del proyecto y la preservación de los datos críticos.

Este proceso es fundamental para garantizar la disponibilidad y la recuperación de la información en situaciones adversas, proporcionando una capa adicional de seguridad y resguardo para la base de datos "proyecto".

Esta descripción resalta la importancia y el propósito del proceso de backup en la gestión de la base de datos, enfocándose en su utilidad para la seguridad y recuperación de datos.

---

## Creacion de usuarios

La creación de usuarios es muy útil para evitar errores accidentales ya que al limitar privilegios pueden evitarse pérdidas importantes de información , para auditoría con el fin de identificar quién realizó modificaciones en los registros, y limitar el acceso según su rol o actividad .

El script básicamente se divide en dos órdenes; creación de los usuarios y los privilegios otorgados a cada uno.

Creación de Usuarios:

- Se están creando dos usuarios: `guest_1` con la contraseña 1234 y `guest_2` con la contraseña 56789. Estos usuarios se están creando para conectarse desde localhost.
- La sintaxis `CREATE USER IF NOT EXISTS 'username'@'hostname' IDENTIFIED BY 'password';` crea un usuario con el nombre especificado ('`guest_1`' y '`guest_2`') y la contraseña correspondiente ('1234' y '56789') para acceder desde localhost.

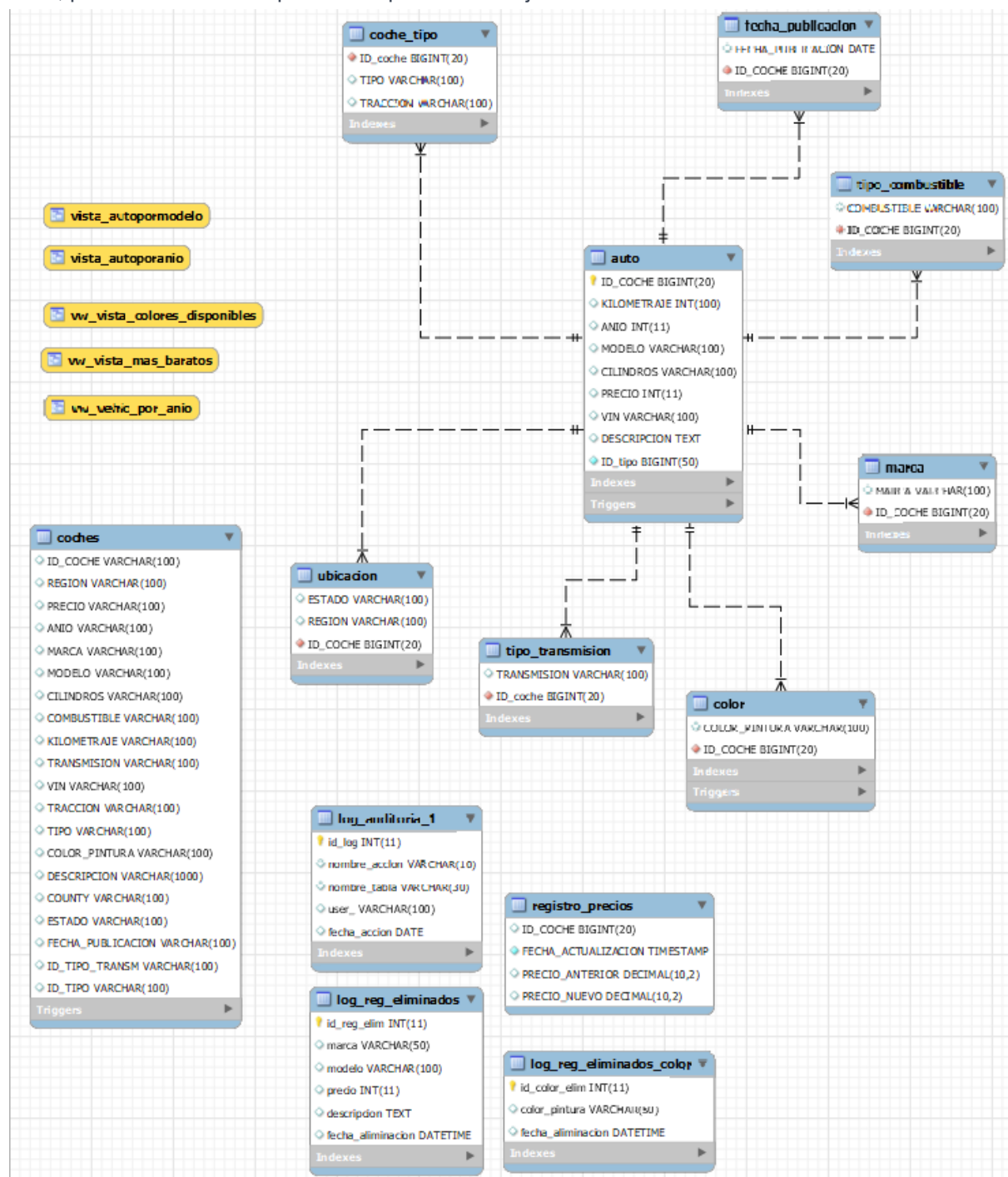
Concesión de Privilegios:

- Para el usuario `guest_1`, se otorga el permiso `SELECT` en la tabla `proyecto.auto`. Esto significa que este usuario puede realizar consultas de selección en la tabla `auto` dentro del esquema `proyecto`.
- Para el usuario `guest_2`, se otorgan permisos `SELECT`, `INSERT`, y `UPDATE` en la tabla `proyecto.auto`. Esto permite al usuario `guest_2` realizar consultas de selección, inserciones y actualizaciones en la tabla `auto` dentro del esquema `proyecto`.

En resumen, este script crea dos usuarios diferentes (guest\_1 y guest\_2) con diferentes niveles de acceso y privilegios en la base de datos proyecto. guest\_1 tiene permiso solo para seleccionar datos, mientras que guest\_2 tiene permisos para seleccionar, insertar y actualizar datos en la tabla auto dentro de ese esquema.

## Grafico Entidad Relacion

Este diagrama proporciona una vista clara y organizada de cómo se interrelacionan estos elementos clave, permitiendo una comprensión rápida de los objetos creados.



---

## Conclusion - Comentarios finales

El informe del proyecto de SQL se centra en el análisis de precios de autos usados, destacando las herramientas utilizadas, la estructura de la carpeta del proyecto y los objetos creados en la base de datos. Los objetivos principales incluyen recopilar datos, realizar un análisis comparativo de precios de autos usados, y presentar los resultados de manera clara y visual.

El proyecto utiliza MySQL para consultas de base de datos y PowerBI para la visualización de datos. La carpeta del proyecto contiene scripts, un diagrama, un instructivo de instalación y un dataset en CSV. La metodología se enfoca en la normalización de datos y la creación de tablas relacionadas para almacenar la información recopilada.

Se describen varios objetos creados en la base de datos, incluyendo tablas principales como "auto" y tablas hijas para categorías específicas, así como vistas, funciones y procedimientos almacenados que ofrecen diferentes perspectivas y funcionalidades útiles para el análisis de datos.

Además, se implementan triggers (disparadores) para registrar acciones como inserciones y eliminaciones en tablas de auditoría, asegurando un seguimiento detallado de los cambios realizados en la base de datos.

Por último, se destaca la importancia del backup de la base de datos para asegurar la integridad de los datos y se detalla la creación de usuarios con diferentes niveles de acceso y privilegios para mantener la seguridad y el control de la base de datos.

En resumen, el proyecto abarca desde la recopilación y almacenamiento de datos hasta la implementación de medidas de seguridad y control, todo orientado a un análisis detallado y completo de los precios de autos usados.

---