



PROYECTO FINAL CONCESIONARIA

Curso SQL - 50055

Profesor: *Nancy Elizabeth Villena Reines*

Tutor: *Ariel Annone*

Alumno: *Valentín Rymaszewski*



Introducción

El sistema está pensado para una concesionaria de vehículos. Es una herramienta clave para la administración integral de la empresa, permitiendo un control preciso de todas las operaciones y recursos de esta. Entre las funciones clave que aborda se incluye la gestión de inventario, seguimiento de clientes, entre otros aspectos más generales.

Objetivo

Para garantizar la eficiencia y el éxito en el proceso de ventas, es esencial contar con una base de datos robusta que almacene una amplia gama de información.

Una base de datos bien estructurada se convierte en una valiosa fuente de datos estadísticos que permitan optimizar las estrategias de ventas a través de la recopilación y análisis de datos, podemos identificar tendencias, preferencias de los clientes, y oportunidades de mejora.

Situación problemática

La situación problemática surge de la falta de una base de datos, esto implica una gran dificultad para gestionar el inventario, realizar seguimiento de los clientes y por otra parte la falta de datos precisos y oportunos para la correcta toma de decisiones. Esto implica directamente en la capacidad de posicionamiento de la empresa en un mercado competitivo.

Resolver esta situación problemática es crucial para el éxito a largo plazo de la empresa.



Modelo de negocio

El modelo de negocio se basa en la venta de vehículos y servicios relacionados
Algunos elementos esenciales del modelo:

1. Venta de vehículos

- La principal fuente de ingresos
- Estrategias de ventas como financiamiento y servicio post-venta

2. Servicio Asociados

- Servicios como mantenimiento, reparaciones y venta de accesorios

3. Gestión del Inventario

- Sistema de gestión del inventario que permita un seguimiento en tiempo real de la disponibilidad de los vehículos.


4. Base de Datos Integral:

- No solo gestiona el inventario y clientes sino que también es utilizada como herramienta estratégica para analizar datos y optimizar acciones.

5. Gestión de Proveedores y Fabricantes:

- Necesario para tener una relación sólida con los fabricantes y estar al tanto de sus próximos lanzamientos y decisiones

Diagrama conceptual

Link:  Diagrama_Conceptual.jpg

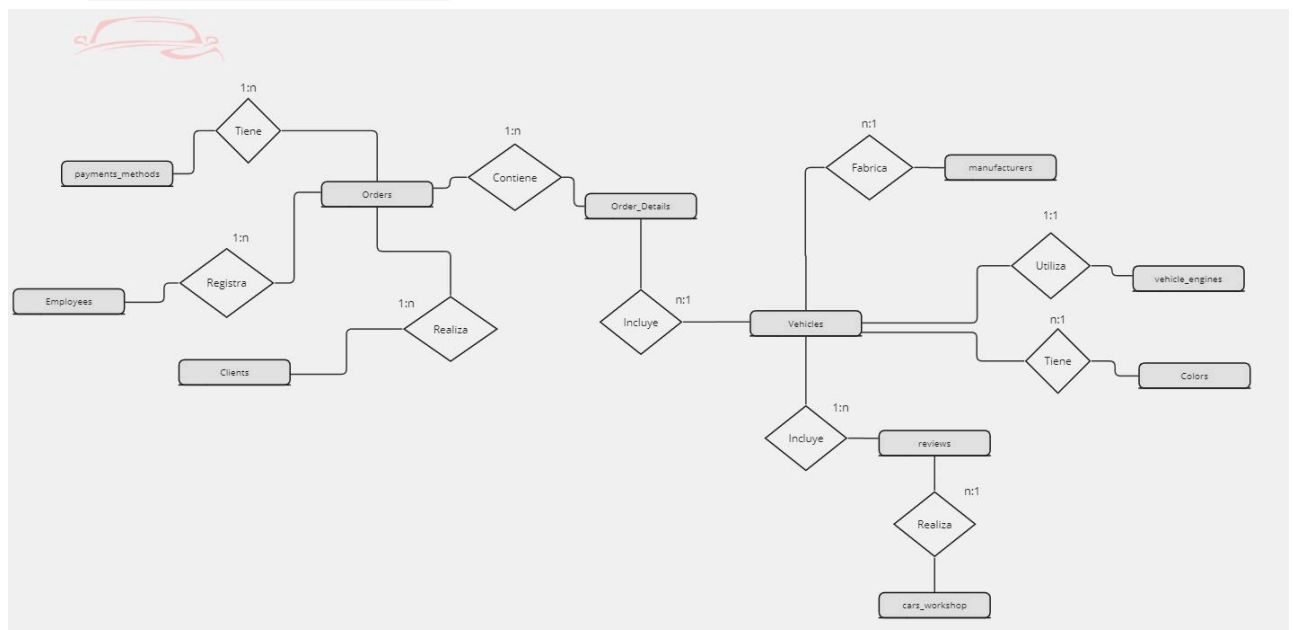
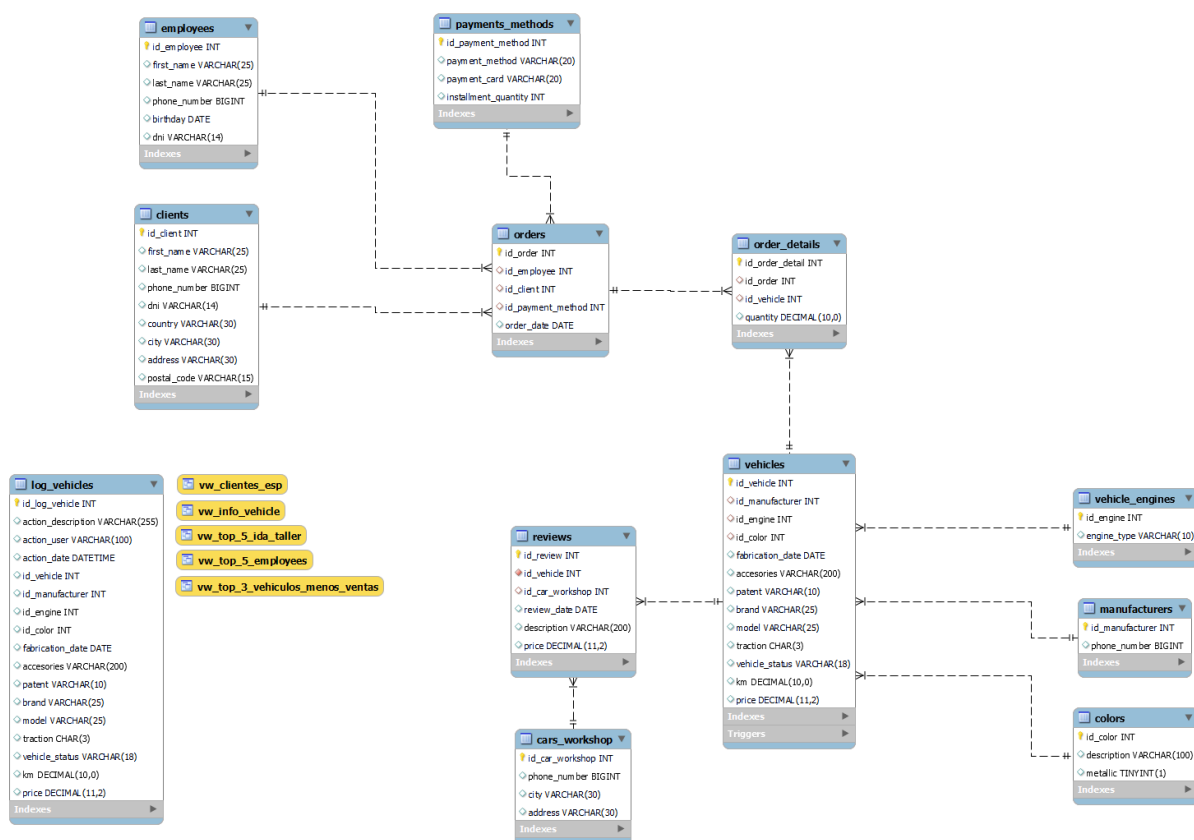


Diagrama entidad relación



Link: [Diagrama_Esquematico.png](#)



Tablas



Link: [+ Tablasv2](#)

Tabla	EMPLOYEE		
Descripción	Almacena información sobre los empleados		
KEY	COLUMNA	TIPO	UNIQUE
PK	id_employee	NUMERIC	Y
	first_name	VARCHAR	
	last_name	VARCHAR	
	dni	VARCHAR	
	phone_number	NUMERIC	
	birthday	DATE	

Tabla	CLIENTS		
Descripción	Almacena información sobre los clientes		
KEY	COLUMNA	TIPO	UNIQUE
PK	id_client	NUMERIC	Y
	first_name	VARCHAR	
	last_name	VARCHAR	
	dni	VARCHAR	
	phone_number	NUMERIC	
	country	VARCHAR	
	city	VARCHAR	
	address	VARCHAR	
	postal_code	NUMERIC	

Tabla	ORDERS		
Descripción	Contiene la información de todas las Orders realizadas		
KEY	COLUMNA	TIPO	UNIQUE
PK	id_order	NUMERIC	Y
FK	id_employee	NUMERIC	
FK	id_client	NUMERIC	



FK	id_payment_method	NUMERIC	
	order_date	DATE	

Tabla	PAYMENTS_METHODS		
Descripcion	Contiene todos los posibles tipos de pago (y nro. cuotas)		
KEY	COLUMNA	TIPO	UNIQUE
PK	id_payment_method	NUMERIC	Y
	payment_method	VARCHAR	
	payment_card	VARCHAR	
	installment_quantity	NUMERIC	

Tabla	ORDERS_DETAILS		
Descripcion	Almacena información detallada sobre las órdenes		
KEY	COLUMNA	TIPO	UNIQUE
PK	id_order_detail	NUMERIC	Y
FK	id_order	NUMERIC	
FK	id_vehicle	NUMERIC	
	quantity	NUMERIC	

Tabla	VEHICLES		
Descripcion	Almacena información de los vehículos, es la tabla principal		
KEY	COLUMNA	TIPO	UNIQUE
PK	id_vehicle	NUMERIC	Y
FK	id_manufacturer	NUMERIC	
FK	id_engine	NUMERIC	
FK	id_color	NUMERIC	
	fabrication_date	DATE	
	accessories	VARCHAR	
	patent	VARCHAR	Y
	brand	VARCHAR	
	model	VARCHAR	
	traction	VARCHAR	
	status	VARCHAR	
	km	NUMERIC	
	price	DECIMAL	



Tabla	MANUFACTURERS		
Descripción	Almacena el contacto de los fabricantes de vehiculos		
KEY	COLUMNA	TIPO	UNIQUE
PK	id_maufacturer	NUMERIC	Y
	phone_number	NUMERIC	

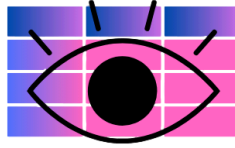
Tabla	COLORS		
Descripción	Almacena los posibles colores de un vehículo		
KEY	COLUMNA	TIPO	UNIQUE
PK	id_color	NUMERIC	Y
	metallic	BOOLEAN	
	description	VARCHAR	

Tabla	REVIEWS		
Descripción	Almacena la información de las revisiones de un vehículo.		
KEY	COLUMNA	TIPO	UNIQUE
PK	id_review	NUMERIC	Y
FK	id_vehicle	NUMERIC	
FK	id_car_workshop	NUMERIC	
	review_date	DATE	
	description	VARCHAR	
	price	DECIMAL	

Tabla	CARS_WORKSOP		
Descripción	Almacena la información de contacto del taller que realiza la revisión		
KEY	COLUMNA	TIPO	UNIQUE
PK	id_car_workshop	NUMERIC	Y
	phone_number	NUMERIC	
	city	VARCHAR	
	address	VARCHAR	



Vistas



1. vm_top_5_employees:

- Descripción:** Devuelve los 5 empleados que más dinero han generado
- Objetivo:** Se podría establecer un sistema de Bonos para los empleados que más dinero han recaudado a modo de incentivo para aumentar las ventas de la concesionaria.
- Tablas:** Employees, orders, vehicles.

```
CREATE VIEW vw_top_5_employees AS (
  SELECT o.id_employee,
         (SELECT e.first_name FROM employees e WHERE o.id_employee = e.id_employee) AS first_name_employee,
         (SELECT e.last_name FROM employees e WHERE o.id_employee = e.id_employee) AS last_name_employee,
         SUM(v.price * od.quantity) as total_generado
  FROM order_details od
  JOIN vehicles v
  ON v.id_vehicle = od.id_vehicle
  JOIN orders o
  ON od.id_order = o.id_order
  GROUP BY id_employee
  ORDER BY total_generado DESC
  LIMIT 5
);
```

2. vm_top_3_vehiculos_menos_ventas:

- Descripción:** Devuelve los 3 vehículos que menos dinero generan.
- Objetivo:** Aporta información sobre los vehículos menos convenientes de vender, o a los que se le podría aplicar una oferta para aumentar sus ventas.
- Tablas:** Vehicles, orders, order_details.




```
CREATE VIEW vw_top_3_vehiculos_menos_ventas AS (
  SELECT v.id_vehicle,
    (SELECT v.brand FROM vehicles v WHERE od.id_vehicle = v.id_vehicle) as brand,
    (SELECT v.model FROM vehicles v WHERE od.id_vehicle = v.id_vehicle) as model,
    SUM(v.price * od.quantity) as total_generado
  FROM order_details od
  JOIN vehicles v
  ON v.id_vehicle = od.id_vehicle
  JOIN orders o
  ON od.id_order = o.id_order
  GROUP BY v.id_vehicle
  ORDER BY total_generado ASC
  LIMIT 3
);
```

3. vm_clientes_esp:

- Descripción:** Devuelve los clientes provenientes de un país hispanohablante, los cuales no tengan código postal
- Objetivo:** Se les podría enviar una notificación al número de contacto para que completen esta información.
- Tablas:** Clients

```
CREATE VIEW vw_clientes_esp AS (
  SELECT *
  FROM clients c
  WHERE country in ('Argentina', 'España', 'Mexico', 'Peru', 'Bolivia', 'Colombia')
  AND postal_code = '' or postal_code = NULL
);
```

4. vm_top_5_ida_taller:

- Descripción:** Devuelve los 5 vehículos que más visitas al taller tienen.
- Objetivo:** Saber cuáles son los vehículos que más inspecciones necesitan, para poder darles un descuento en el taller e intentar aumentar las ventas.
- Tablas:** Vehicles, reviews



```

CREATE VIEW vw_top_5_ida_taller AS (
    SELECT r.id_vehicle,
           v.brand,
           v.model,
           count(r.id_vehicle) as ida_taller
    FROM reviews r
    JOIN vehicles v
    ON v.id_vehicle = r.id_vehicle
    GROUP BY id_vehicle
    ORDER BY ida_taller DESC
    LIMIT 5
);

```

5. vm_info_vehicles:

- Descripción:** Devuelve la información ampliada de vehículo, como el el tipo de motor, o el color.
- Objetivo:** Ver de manera sencilla información detallada / relacionada de un vehículo.
- Tablas:** Vehicles, vehicle_engines, colors

```

CREATE VIEW vw_info_vehicle AS (
    SELECT v.id_vehicle,
           ve.engine_type,
           v.brand,
           v.model,
           c.description as color
    FROM vehicles v
    JOIN vehicle_engines ve
    ON v.id_engine = ve.id_engine
    JOIN colors c
    ON v.id_color = c.id_color
    ORDER BY id_vehicle
);

```



Store procedure



1. **sp_ordenar_vehicles:** Ordena la tabla vehiculo por un campo pasado como parámetro, ya sea de manera ASC o DESC según se le especifique por parámetro

```

DROP PROCEDURE IF EXISTS sp_ordenar_vehicles;

DELIMITER //
CREATE PROCEDURE `sp_ordenar_vehicles` (IN campo VARCHAR(25), IN orden VARCHAR(4))
BEGIN
    IF (orden != 'ASC' AND orden != 'DESC') THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El parametro orden debe ser: ASC o DESC';
    ELSE
        SET @ordenar = CONCAT('SELECT * FROM vehicles', ' ', 'ORDER BY', ' ', campo, ' ', orden);
    END IF;

    PREPARE consulta FROM @ordenar;
    EXECUTE consulta;
    DEALLOCATE PREPARE consulta;
END //
DELIMITER ;

```

2. **sp_insertar_order:** Toma 3 parámetros de entrada que son el id del cliente, id del empleado y el id del método de pago, luego corrobora que los 3 existan, le agrega la fecha del momento y por último registra una nueva orden.

```

DROP PROCEDURE IF EXISTS sp_insertar_order;

DELIMITER //
CREATE PROCEDURE `sp_insertar_order` (IN sp_id_employee INT,
                                     IN sp_id_client INT,
                                     IN sp_id_payment_method INT)
BEGIN
    IF (NOT
        (SELECT sp_id_employee in (SELECT id_employee FROM employees)) AND
        (SELECT sp_id_client in (SELECT id_client FROM clients)) AND
        (SELECT sp_id_payment_method in (SELECT id_payment_method FROM payments_methods))
    ) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Parametro/s invalido/s, corrobore que existan todos los id.';
    ELSE
        INSERT INTO orders
        (id_employee, id_client, id_payment_method, order_date)
        VALUES
        (sp_id_employee, sp_id_client, sp_id_payment_method, (CURRENT_DATE()));
    END IF;
END
//
DELIMITER ;

```



Functions



1. f_vehicle_iva:

- Descripción:** Devuelve el precio mas el iva de un vehículo, es decir el costo final
- Objetivo:** Ahorrar recursos en la base de datos, ya que el precio + el iva es una operación que se realiza constantemente.
- Tablas:** Vehicles

```
DELIMITER //
CREATE FUNCTION `f_vehicle_iva` (`vehicle_id` INT) RETURNS DECIMAL (11, 2)
NO SQL
BEGIN
    DECLARE precio DECIMAL(11, 2);
    SET precio = ((SELECT price FROM vehicles WHERE id_vehicle = vehicle_id) + ((SELECT price FROM vehicles WHERE id_vehicle = vehicle_id) * (21 / 100)));
    RETURN precio;
END;
//
DELIMITER ;
```

```
SELECT price AS Precio,
        f_vehicle_iva(5) as Precio_Final
FROM vehicles
WHERE id_vehicle = 5;
```

	Precio	Precio_Final
▶	475331.42	575151.02

2. f_descuento:

- Descripción:** Otorga un descuento a los clientes cuyo vehículo haya tenido más de dos (2) revisiones en el taller.
- Objetivo:** Mejorar el servicio de post venta y la relación con los clientes.
- Tablas:** Vehicles, reviews



```

DELIMITER //
CREATE FUNCTION `f_descuento` (`vehicle_id` INT) RETURNS TINYINT(1)
NO SQL
BEGIN
    DECLARE `discount` TINYINT(1);
    SET discount = IF((
        SELECT count(*) as times_in_workshop
        FROM reviews
        WHERE id_vehicle = vehicle_id
        GROUP BY id_vehicle
        ) > 2, 1, 0);
    -- El WHERE devuelve la cantidad de veces que un vehiculo tuvo una review (Fue al taller)
    RETURN discount;
END;
//
DELIMITER ;

SELECT f_descuento(832) AS discount;

```

```

SELECT id_vehicle,
       count(*) as times_in_workshop
FROM reviews
WHERE id_vehicle = 832
GROUP BY id_vehicle;

SELECT f_descuento(832) AS discount;

```

id_vehicle	times_in_workshop
832	5

discount
1

Triggers



TRIGGER	TABLA MONITOREADA	ACCIÓN	TABLA LOG	DESCRIPCIÓN
tr_b_u_vehicles	vehicles	BEFORE UPDATE	log_vehicles	Almacenar en la tabla log_vehicles los datos antiguos de la tabla vehiculo
tr_a_d_vehicles	vehicles	AFTER DELETE	log_vehicles	Almacena en la tabla log_vehicles los datos de un vehículo y la fecha y hora antes de eliminarlo

Herramientas y tecnologías utilizadas

Tecnologías y herramientas utilizadas tanto para la creación de la DB en cuestión, como para la generación del proyecto completo.

- MySQL Workbench: Creación y gestión de la base de datos.
- Miro: Generación del diagrama conceptual.
- Google Sheets: Descripción de los elementos de la DB y generación de informes.



- Google Docs: Documentación.
- Obsidian: Toma de apuntes y planificación del proyecto.
- Mockaroo: Generación de datos aleatorios.
- Google Drive: Repositorio.

Futuras líneas

Futuras medidas a implementar en el proyecto:

- Despliegue de la base de datos en sobre un sistema web funcional
- Gestión de usuarios: Ampliar los usuarios para las diferentes áreas de la empresa ej: usuario para el responsable de contactar a los proveedores para que este los pueda cargar, usuario empleados para registrar las ventas, usuario de post venta para registrar las revisiones mecánicas de los vehículos de los clientes
- Ampliación de la base de datos: Expandir el sistema a diversas áreas como la venta de accesorios, lavandería, personalización, etc.

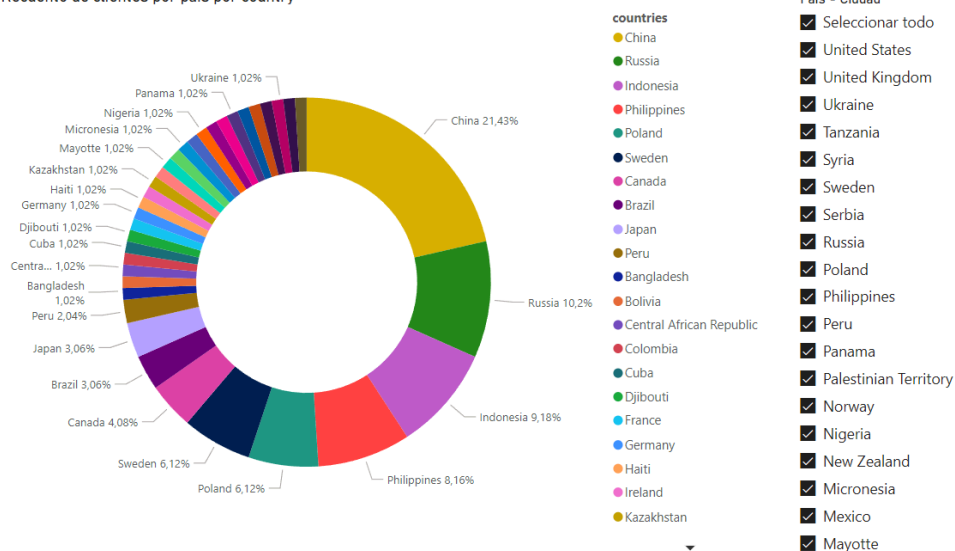
Informes

Link: [Concecionria_rymaszewski.pbix](https://concecionria-rymaszewski.pbix)



Informe de Clientes

Recuento de clientes por país por country

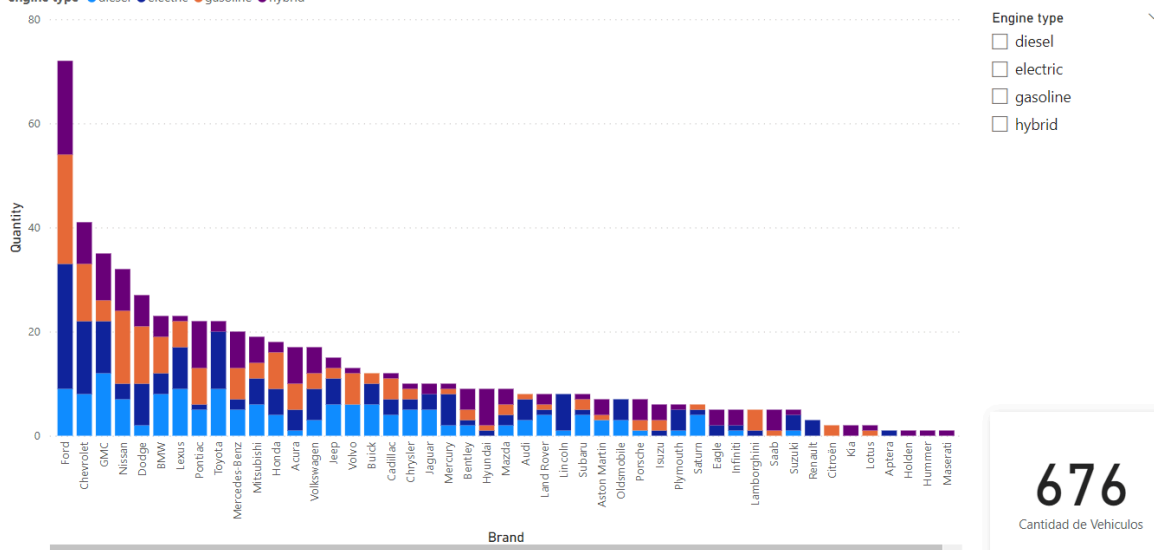


Informe sobre la cantidad de clientes en los diferentes países.

Informe de Reviews

Informe de visitas al taller por marca

engine type ● diesel ● electric ● gasoline ● hybrid



Informe sobre la cantidad de vehículos que realizan visitar al taller, agrupadas por marca y tipo de motor

Script

Link: [script.sql](#)

