

REPORTE DE PRÁCTICA NO. 1.4

GESTIÓN DE FLOTILLA DE AUTOS

ALUMNO: Israel Campos Vázquez
Dr. Eduardo Cornejo-Velázquez



1. Introducción

El proceso de hacer consultas a la Base de Datos es algo que debemos tener muy claro como administradores de bases de datos, ya que el cliente puede solicitar información específica dentro de la Base de Datos que nosotros deberemos saber manipular y consultar para devolverla.

En esta práctica pondremos a prueba las consultas SQL que ya sabemos hacer, pero también, investigaremos más formas de consultar información a la Base de datos de ser necesario. Aprenderemos a usar funciones como `order by`, `group by`, `join`, etc.

2. Marco teórico

SQL, MySQL

La gestión de una flotilla de autos implica tener el control de aspectos como el gasto de gasolina, mantenimiento de los autos, verificación de rutas y regulación de documentos. Por ello, debemos definir con precisión los requerimientos del sistema.

Una vez hecho esto, el siguiente poblar nuestra Base de Datos con datos prueba para poder saber con exactitud como acceder a determinada información solicitada.

Para hacer consultas que sean un poco más complejas podemos hacer uso de inner join, join, left join, right join. En SQL, JOIN se usa para unir filas de dos o más tablas basándose en una condición común. Se utilizan claves primarias y foráneas para relacionar las tablas.

Un INNER JOIN devuelve solo las filas donde hay coincidencias en ambas tablas.

Cuando usas simplemente JOIN, en la mayoría de los motores de base de datos (incluyendo MySQL), se interpreta como un INNER JOIN, por lo que el comportamiento es el mismo: solo muestra los registros que coinciden en ambas tablas.

El LEFT JOIN devuelve todas las filas de la tabla de la izquierda, aunque no haya coincidencias en la tabla de la derecha.

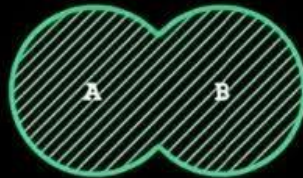
Por otro lado, el RIGHT JOIN hace lo contrario: devuelve todos los registros de la tabla de la derecha y solo las coincidencias de la tabla de la izquierda .



```

1 SELECT *
2 FROM A
3 INNER JOIN B ON A.key = B.key

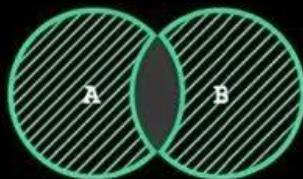
```



```

1 SELECT *
2 FROM A
3 FULL JOIN B ON A.key = B.key

```



```

1 SELECT *
2 FROM A
3 FULL JOIN B ON A.key = B.key
4 WHERE A.key IS NULL OR
5 B.key IS NULL

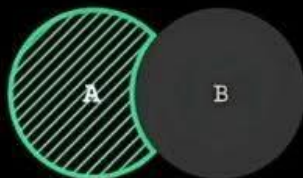
```



```

1 SELECT *
2 FROM A
3 LEFT JOIN B ON A.key = B.key

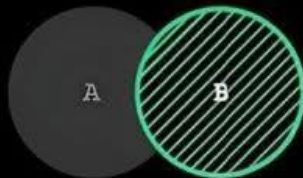
```



```

1 SELECT *
2 FROM A
3 LEFT JOIN B ON A.key = B.key
4 WHERE B.key IS NULL

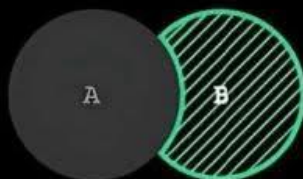
```



```

1 SELECT *
2 FROM A
3 RIGHT JOIN B ON A.key = B.key

```



```

1 SELECT *
2 FROM A
3 RIGHT JOIN B ON A.key = B.key
4 WHERE B.key IS NULL

```

Figure 1: JOIN.

Algebra Relacional

En SQL, el uso de JOIN sigue de cerca las operaciones de álgebra relacional.

INNER JOIN: Representa la intersección de dos relaciones en álgebra relacional. Solo se devuelven las filas que cumplen con la condición de unión en ambas tablas.

Operación Relacional: Producto cartesiano seguido de una selección () que filtra las filas que cumplen con la condición de igualdad entre columnas.

LEFT JOIN: Es una combinación de un INNER JOIN seguido de una diferencia () entre la tabla izquierda y el resultado del INNER JOIN para obtener las filas que están en la tabla izquierda pero no tienen coincidencias en la tabla derecha.

Operación Relacional: Producto cartesiano seguido de una selección () y una diferencia () entre las tablas.

RIGHT JOIN: Similar al LEFT JOIN, pero devuelve todas las filas de la tabla derecha, incluso si no hay coincidencias en la tabla izquierda.

Operación Relacional: Producto cartesiano seguido de una selección () y una diferencia () entre las tablas.

FULL OUTER JOIN: Devuelve todas las filas de ambas tablas, incluyendo aquellas que no tienen coincidencias en la otra tabla. Esto se puede lograr mediante la unión () de un LEFT JOIN y un RIGHT JOIN.

Operación Relacional: Unión () de los resultados de un LEFT JOIN y un RIGHT JOIN.

4. Desarrollo

Respaldo Base de Datos

Haz clic aquí para ver el respaldo de la Base de Datos

Sentencias SQL y Algebra Relacional

placa,marca,año,modelo,fechaMantenimiento(year(fechaMantenimiento)=2025month(fechaMantenimiento)=1 (Vehiculo×Mantenimiento))

	placa	V.marca	año	modelo	V.marca	fechaMantenimiento
1	AS12-AS3	Peugeot	2020	Rifter	Peugeot	2025-01-12
2	FD3-45G	Renault	2020	Oroch	Renault	2025-01-10
3	FDL-42K	RAM	2023	RAM 1200	RAM	2025-01-12

```
1      select V.placa, V.marca, V.año, V.modelo, V.marca, M.fecha as fechaMantenimiento
2      from Vehiculo as V
3      join
4      Mantenimiento as M on V.placa=M.placa
5      where year(M.fecha)=2025 and month(M.fecha) = 1
6      order by V.año;
```

Listing 1: Consulta 1.

placa,marca,año,modelo,fechaVerificacion (year(fechaVerificacion)=2025month(fechaVerificacion)=3 (Vehiculo Vehiculo.placa = Verificacion.placa Verificacion))

	placa	V.marca	año	modelo	V.marca	fechaVerificacion
1	AS12-AS3	Peugeot	2020	Rifter	Peugeot	2025-03-10
2	FDS32-12	Volkswagen	2021	Saveiro	Volkswagen	2025-03-15
3	FD3-45G	Renault	2020	Oroch	Renault	2025-03-20

```

1      select V.placa, V.marca, V.año, V.modelo, V.marca, e.fecha as fechaVerificacion
2      from Vehiculo as V
3      join
4      Verificacion as e on V.placa=e.placa
5      where year(e.fecha)=2025 and month(e.fecha) = 3
6      order by e.fecha asc ;

```

Listing 2: Consulta 2.

nombre,fechaRuta (year(fechaRuta)=2025month(fechaRuta)=2day(fechaRuta)=19 (Conductor Conduc-
tor.curp = Ruta.curp Ruta))

	nombre	fechaRuta
1	Alejandro Gonzalez Cruz	2025-02-19

```

1      select C.nombre, R.fecha as fechaRuta
2      from Conductor as C
3      join
4      Ruta as R on C.curp = R.curp
5      where year(R.fecha)=2025 and month(R.fecha)=02 and day(R.fecha)=19
6      order by R.fecha asc;

```

Listing 3: Consulta 3.

placa,marca,modelo,totalgasto (totalgasto;2300 (placa,fecha;SUM(costoViaje) → totalgasto(RutaRuta.placa = Vehiculo.

```

1      SELECT r.placa,v.marca,v.modelo,SUM(r.costoViaje) AS total_gasto
2      FROM Ruta r
3      JOIN Vehiculo v on r.placa = v.placa
4      WHERE r.fecha >= DATE_SUB(CURDATE(), INTERVAL 2 MONTH)
5      GROUP BY r.placa, r.fecha
6      HAVING total_gasto > 2300
7      ORDER BY total_gasto DESC;

```

Listing 4: Consulta 4.

	placa	marca	modelo	total_gasto
1	SDF-12	Peugeot	Rifter	2606.0999603271484

marca,modelo,placa,tipo,costo (fechadatesub(curdate(), interval 2 month) (Refaccion Refaccion.placa = Ve-
hiculo.placa Vehiculo))

	marca	modelo	placa	tipo	costo
1	Volkswagen	Saveiro	FDS32-12	Valvulas	3000
2	Renault	Oroch	FD3-45G	Llantas	6000
3	Peugeot	Rifter	SDF-12	Frenos	8000
4	RAM	RAM 1200	FDL-42K	Caja	10000
5	Peugeot	Rifter	AS12-AS3	Motor	20000

```

1      select v.marca,v.modelo,r.placa,r.tipo,r.costo
2      from Refaccion r
3      join Veh culo v on r.placa = v.placa
4      where r.fecha >= date_sub(curdate(), interval 2 month)
5      order by r.costo asc;

```

Listing 5: Consulta 5.

nombre,placa,idRuta,fecha,horaLlegada,horaSalida,destino,costoViaje ((Ruta Ruta.placa = Vehiculo.placa Vehiculo) Ruta.curp = Conductor.curp Conductor))

	nombre	placa	idRuta	fecha	horaLlegada	horaS...	destino	costoViaje
4	Alejandro Gonzal...	AS12-AS3	19	2025-02-19	21:55:59	17:55:59	Cuautitlan	247.9
5	Alejandro Gonzal...	FDL-42K	4	2025-02-08	19:55:59	18:51:39	Lomas Verdes	88.5357
6	Alejandro Gonzal...	FDL-42K	9	2025-02-08	19:55:59	18:51:39	Lomas Verdes	973.893
7	Ruben Ruiz Diaz	FDS32-12	2	2025-02-06	15:55:59	14:31:45	Coapa	141.944
8	Ruben Ruiz Diaz	FDS32-12	7	2025-02-06	15:55:59	14:31:45	Coapa	141.944
9	Ruben Ruiz Diaz	SDF-12	5	2025-02-08	20:55:59	19:52:13	Satelite	102.2
9	Ruben Ruiz Diaz	SDF-12	10	2025-02-08	20:55:59	19:52:13	Satelite	783.533
10	Ruben Ruiz Diaz	SDF-12	11	2025-02-07	21:55:59	17:55:59	Cuautitlan	340.667
11	Alejandro Gonzal...	SDF-12	11	2025-02-07	21:55:59	17:55:59	Cuautitlan	340.667
12	Ruben Ruiz Diaz	SDF-12	12	2025-02-06	14:31:45	14:31:45	Coapa	170.333
13	Antonio Cruz Ros	SDF-12	13	2025-02-06	14:31:45	13:55:12	Santa Fe	579.133

```

1      SELECT C.nombre, V.placa, R.idRuta, R.fecha, R.horaLlegada, R.horaSalida, R.destino, R.costoViaje
2      FROM Ruta R
3      JOIN Veh culo V ON R.placa = V.placa
4      JOIN Conductor C ON R.curp = C.curp;

```

Listing 6: Consulta Resumen de Viaje.

nombre,conductor,totalrutas (curp,nombre;COUNT(*) → totalrutas(RutaRuta.curp = Conductor.curpConductor))

```

1      SELECT c.nombre, r.curp AS conductor, COUNT(*) AS total_rutas
2      FROM Ruta r
3      JOIN Conductor c ON r.curp = c.curp
4      GROUP BY r.curp, c.nombre
5      ORDER BY total_rutas DESC;

```

Listing 7: Consulta Total de Viajes Chofer.

	nombre	conductor	total_rutas
1	Alejandro Gonzalez Cruz	60CA040227HDFMZSA1	8
2	Ruben Ruiz Diaz	RUDR040227HDFMZSA1	6
3	Antonio Cruz Rosas	CRRA040227HDFMZSA1	3

placa,modelo,marca,a nio,totalrutas (placa,modelo;COUNT(*) → totalrutas(RutaRuta.placa = Vehiculo.placaVehiculo))

```

1      select r.placa, v.modelo, v.marca,v.a o ,count(*) as total_rutas
2      from Ruta r
3      JOIN Veh culo v on r.placa = v.placa
4      group by r.placa,v.modelo
5      order by total_rutas desc;

```

Listing 8: Consulta Total de Viajes Vehiculo.

	placa	modelo	marca	año	total_rutas
1	SDF-12	Rifter	Peugeot	2024	7
2	AS12-AS3	Rifter	Peugeot	2020	4
3	FD3-45G	Oroch	Renault	2020	2
4	FDL-42K	RAM 1200	RAM	2023	2
5	FDS32-12	Saveiro	Volkswagen	2021	2

origen,destino,kilometraje (Ruta)

	origen	destino	kilometraje
1	Toreo	Santa Fe	40
2	San Mateo	Lomas Verdes	50
3	San Mateo	Satelite	60
4	Toreo	Coapa	100
5	Toreo	Coapa	100
6	Toreo	Coapa	

```

1      select origen,destino,kilometraje
2      from Ruta
3      order by kilometraje asc;
```

Listing 9: Consulta Rutas mas cortas.

origen,destino,kilometraje (Ruta)

	origen	destino	kilometraje
1	San Mateo	Lomas Verdes	550
2	San Mateo	Lomas Verdes	550
3	San Mateo	Satelite	460
4	San Mateo	Satelite	460
5	Toreo	Santa Fe	340
6	Toreo	Santa Fe	

```

1      select origen,destino,kilometraje
2      from Ruta
3      order by kilometraje desc;
```

Listing 10: Consulta Rutas mas largas.

5. Conclusiones

Al realizar esta práctica, pude recordar la sintaxis y funcionamiento de sentencias SQL que había utilizado en clases anteriores. En este caso fueron sentencias para hacer consultas a la Base de Datos.

Además de utilizar consultas SQL que ya conocía, tuve que investigar nuevas consultas y modificadores para cumplir con los requisitos de la práctica.

Fue muy interesante conocer las múltiples opciones que te otorga SQL, había funciones que ni me imaginaba se podían hacer.

Referencias Bibliográficas

References

- [1] Grabowska, S.; Saniuk, S. (**2022**). Business models in the industry 4.0 environment—results of web of science bibliometric analysis. *J. Open Innov. Technol. Mark. Complex*, 8(1), 19.
- [2] FUNDAMENTOS DE BASES DE DATOS (4.a ed.). (**2002**). Silberschatz, Korth y Sudarshan. <http://biblioteca.univalle.edu.ni/files/original/01aebde3cc06dce33f2538aa2724eb2541cb9473.pdf>
- [3] Fundamentos de Sistemas de Bases de Datos (5.a ed.). (2007). Elmasri y Navathe. <https://gc.scalahed.com/recursos/files/r161r/w24566w/FundamentosDeSistemasDeBasesDeDatos.-1-69.pdf>
- [4] Database Systems (6.a ed.). (2015). Conolly, Begg. <https://dl.ebooksworld.ir/motoman/Pearson.Database.Systems.A.Pr>