

Revisión de Sentencias

Sitio: [Agencia de Aprendizaje a lo largo de la Vida](#)
Curso: Administración de Base de Datos 1° F
Libro: Revisión de Sentencias

Imprimido por: MARIO DAVID GONZALEZ BENITEZ
Día: sábado, 21 de septiembre de 2024, 20:04

Tabla de contenidos

- 1. Introducción y recursos
- 2. Básica : "Distinct" - "Cambio de nombre"
- 3. Básica: "Between" - "Order by"
- 4. Básica "Like"
- 5. Funciones de agregación
- 6. Funciones de agrupación
- 7. Funciones de pertenencia a conjuntos

Introducción y recursos



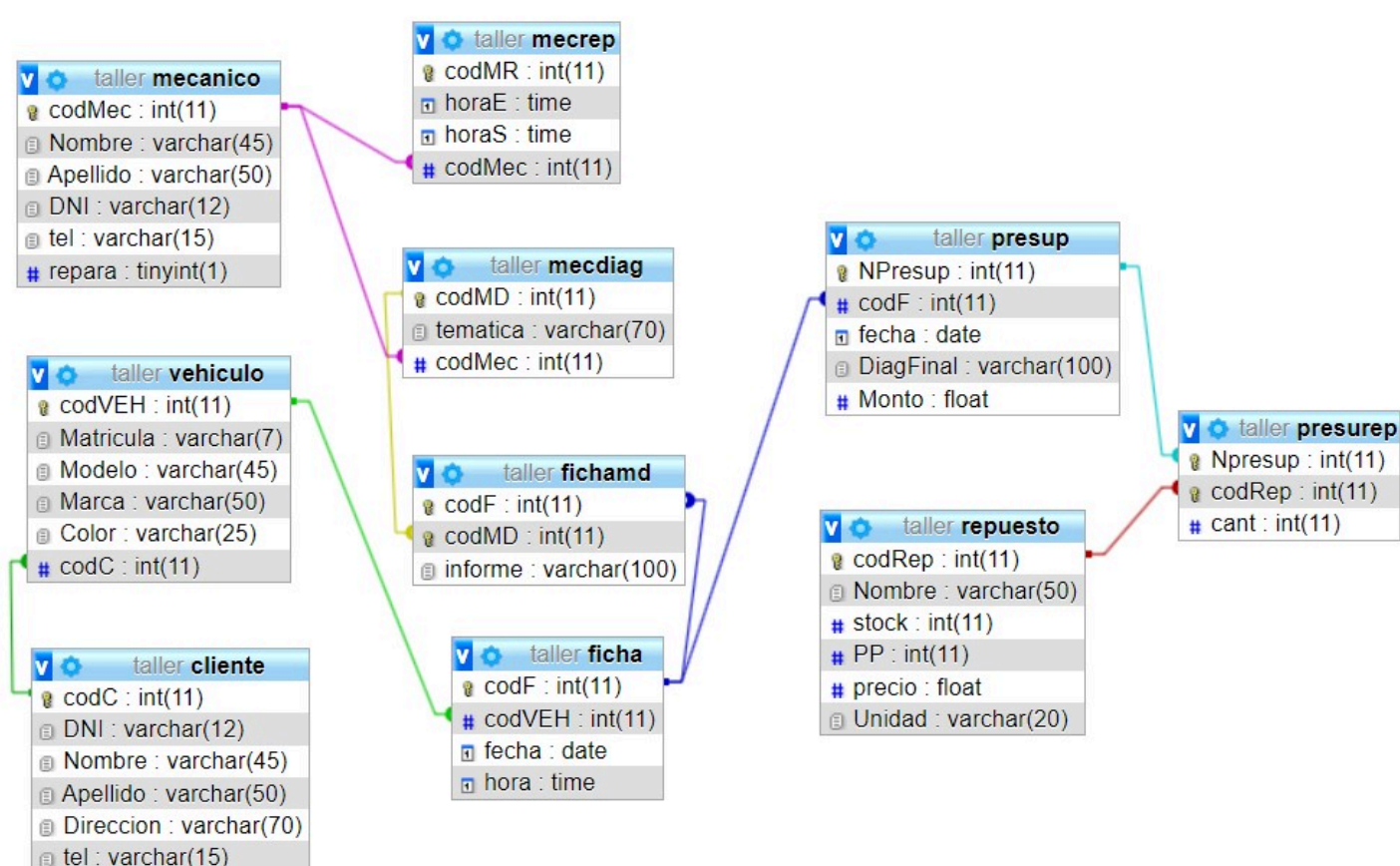
Esta parte del repaso vamos a trabajar con SQL, la base de datos se llama *"Taller"* y corresponde al análisis que vimos en el libro anterior.

Realizaremos un ejemplo modelo para cada caso.



Descargá el archivo [Semana7_BaseTaller.sql](#) en la sección bibliografía de este curso. El archivo tiene la creación y los *insert* correspondientes.

La gráfica es la siguiente:



Distinct - Cambio de nombre



¿Recordás para que utilizamos [Distinct](#) y [Cambio de nombre](#) ?

[Distinct](#) : elimina tuplas repetidas en el resultado

Mostrar los números de presupuestos que usaron repuestos

```
MariaDB [taller]> select Npresup from presurep;
+-----+
| Npresup |
+-----+
| 70100   |
| 70101   |
| 70101   |
| 70104   |
| 70102   |
| 70102   |
+-----+
6 rows in set (0.00 sec)
```

Observamos que el presupuesto **70101** se repite ¿Por qué?
Porque ese presupuesto necesita más de un repuesto

```
MariaDB [taller]> select distinct Npresup from presurep;
+-----+
| Npresup |
+-----+
| 70100   |
| 70101   |
| 70102   |
| 70104   |
+-----+
4 rows in set (0.00 sec)
```

Aquí no aparece por el uso del **distinct**

[Cambio de nombre](#): Cambia el nombre de las tablas o de las columnas que se proyectan. Se usa "as".

Mostrar los números de presupuestos que usaron repuestos

(en lugar de proyectar el nombre del atributo colocar PRESUPUESTO)

```
MariaDB [taller]> select distinct Npresup AS PRESUPUESTO from presurep;
+-----+
| PRESUPUESTO |
+-----+
| 70100       |
| 70101       |
| 70102       |
| 70104       |
+-----+
4 rows in set (0.00 sec)
```

Si el cambio de nombre tiene **más de una palabra** debe ir entre **comillas**

Between - Order by



¿Recordás en que caso utilizamos [Between](#) y en que caso [Order by](#)?

[Between](#): cuando el dominio del predicado pertenece a un rango de valores.

Mostrar los repuestos cuyo precio oscile entre 3000 y 6500 **inclusive**

MariaDB [taller]> select * from repuesto
-> where precio between 3000 and 6500;

BETWEEN <<incluye>> los extremos

codRep	Nombre	stock	PP	precio	Unidad
65	Cigüeñal	10	8	3000	Cada uno
89	Embrague de auto 250	4	3	6250	Por unidad
90	Puerta	50	9	5600	Por unidad
91	Bocina	14	6	3000	Por unidad
93	Carburador	6	3	5678	Por unidad
96	Amortiguador Peugeot 308	4	2	4000	Por unidad
97	Pack de Ruedas Todo terreno	2	2	4000	Por unidad
123	Cilindro	10	5	3045	Por unidad

8 rows in set (0.00 sec)

Mostrar los repuestos cuyo precio oscile **entre** 3000 y 6500

MariaDB [taller]> select * from repuesto
-> where precio between 3001 and 6499;

Dice "entre" es un intervalo abierto

codRep	Nombre	stock	PP	precio	Unidad
89	Embrague de auto 250	4	3	6250	Por unidad
90	Puerta	50	9	5600	Por unidad
93	Carburador	6	3	5678	Por unidad
96	Amortiguador Peugeot 308	4	2	4000	Por unidad
97	Pack de Ruedas Todo terreno	2	2	4000	Por unidad
123	Cilindro	10	5	3045	Por unidad

6 rows in set (0.00 sec)

[Order by](#): o rdena el resultado por un atributo o varios.

*Mostrar los nombre y precio de los repuestos cuyo precio oscile entre 3000 y 6500
ordenados alfabéticamente*

```
MariaDB [taller]> select nombre as REPUESTO, precio as PRECIO from repuesto  
-> where precio between 3001 and 6499  
-> order by nombre;
```

Resultado ordenado **ascendente**

REPUESTO	PRECIO
Amortiguador Peugeot 308	4000
Carburador	5678
Cilindro	3045
Embrague de auto 250	6250
Pack de Ruedas Todo terreno	4000
Puerta	5600

6 rows in set (0.00 sec)

```
MariaDB [taller]> select nombre as REPUESTO, precio as PRECIO from repuesto  
-> where precio between 3001 and 6499  
-> order by nombre desc;
```

Resultado ordenado **descendente**

REPUESTO	PRECIO
Puerta	5600
Pack de Ruedas Todo terreno	4000
Embrague de auto 250	6250
Cilindro	3045
Carburador	5678
Amortiguador Peugeot 308	4000

6 rows in set (0.00 sec)



¿Recordás para que utilizamos Like?

Like: Tuplas que contengan determinada información, se usa el comodín que es %

Mostrar los códigos de mecánicos que se dedican a diagnosticar problemas de “motor”

```
MariaDB [taller]> select * from mecdiag;
```

codMD	tematica	codMec
2122	Motor roto	
2127	Motores en general	
2143	Resortes del diferencial rotos	
2151	Puertas	
2155	Neumaticos	
2160	Motores	
2161	Analista de chasis	
2163	Analista de piston	
2164	Electricista	
2165	Amortiguadores y direccion	
2166	Direccion	
2168	Tren delantero	
2190	Mecanica general	
2197	Neumatica	542
2199	Tren delantero	544

15 rows in set (0.00 sec)

Vemos que los mecánicos que diagnostican problemas de **motor** están ingresados de diferente forma, pero en definitiva pueden diagnosticar

El comodín % puede ser
 “motor%” → Comienza con motores
 “%motor%” → contiene
 “%motor” → termina

```
MariaDB [taller]> select * from mecdiag
-> where tematica like "motor%";
```

codMD	tematica	codMec
2122	Motor roto	483
2127	Motores en general	488
2160	Motores	476

3 rows in set (0.00 sec)

Funciones de agregación



¿Para que sirven?

Operan sobre un conjunto de tuplas y arrojan un único resultado.

Repasemos caso por caso los diferentes operadores.

- AVG

Mostrar el promedio de los precios de los repuestos

```
MariaDB [taller]> select avg(precio) as "Precio promedio"
-> from repuesto;
+-----+
| Precio promedio |
+-----+
|      9324.21875 |
+-----+
1 row in set (0.00 sec)
```

Renombramos la columna

- COUNT

Mostrar la cantidad de vehículos registrados en el taller

```
MariaDB [taller]> select count(codveh) from vehiculo;
+-----+
| count(codveh) |
+-----+
|             27 |
+-----+
1 row in set (0.00 sec)

MariaDB [taller]> select count(*) from vehiculo;
+-----+
| count(*) |
+-----+
|       27 |
+-----+
1 row in set (0.00 sec)
```

Se coloca el nombre de la columna
Si hay un Null → no lo cuenta

Se coloca * (todas las filas)

- MAX

Mostrar el código del último cliente registrado

```
MariaDB [taller]> select max(codc) as CLIENTE from cliente;
+-----+
| CLIENTE |
+-----+
|      1104 |
+-----+
1 row in set (0.00 sec)
```

Es el código más **grande**

- MIN

Mostrar el número de la primera ficha registrada

```
MariaDB [taller]> select min(codf) as "Primer ficha" from ficha;
+-----+
| Primer ficha |
+-----+
|          8003 |
+-----+
1 row in set (0.00 sec)
```

Es el código más **chico**

- SUM

Mostrar la los montos de los presupuestos

```
MariaDB [taller]> select sum(monto) as "Montos Totales" from presup;
+-----+
| Montos Totales |
+-----+
|          200950 |
+-----+
1 row in set (0.00 sec)
```

Se **suma** todos los dominios de la columna monto



¿Para que sirven?

- GROUP BY:

Agrupar las tuplas por algún criterio para obtener un resultado.

Mostrar la cantidad de vehículos de cada marca

```
MariaDB [taller]> select marca, count(*) from vehiculo
-> group by marca;
```

marca	count(*)
Audi	1
Benztruk	1
Chery	1
Chevrolet	1
Ferrari	2
Fiat	4
Ford	6
Nissan	1
Peugeot	4
Renault	3
Subaru	1
Volkswagen	2

12 rows in set (0.00 sec)

Primero se agrupa por **marca** y después se proyecta lo solicitado.

- HAVING:

Es el filtro del grupo (group by).

Mostrar la marca de vehículos que tienen 4 autos registrados

```
MariaDB [taller]> select marca from vehiculo
-> group by marca
-> having count(marca) = 4;
```

marca
Fiat
Peugeot

2 rows in set (0.00 sec)

La condición de **4 autos** por marca que es el filtro se aplica al **grupo**, por eso se coloca en el having

Funciones de pertenencia a conjuntos



¿Para que sirven?

Se emplean en subconsultas.

- **IN**: comprueba si el atributo es parte de un conjunto.

Mostrar nombre y apellido de los mecánicos que solo reparan vehículos

```
MariaDB [taller]> select nombre, apellido from mecanico
-> where codmec in (select codmec from mecprep);
```

nombre	apellido
Marcelo	Juarez
Melchor	Moreno
Jose Maria	Mayo
Raul	Damonte
Gustavo	Lopez
Joaquin	Ragonese
Luciano	Florian
Samuel	Condori
Javier	Benitez
Sofia	Moreyra

10 rows in set (0.00 sec)

La subconsulta proyecta los códigos de mecánicos que reparan vehículos.

La consulta principal proyecta los datos solicitados de los códigos que proyecta la subconsulta

- **NOT IN**: comprueba si el atributo no es parte de un conjunto.

Mostrar número de presupuesto, ficha y monto de los presupuestos que NO se realizaron entre el 1 de abril y el 5 de abril del 2023

```
MariaDB [taller]>
MariaDB [taller]> select NPresup, codf as FICHA, monto
-> from presup
-> where Npresup NOT IN (select Npresup from presup
-> where fecha between '2023/04/01' and '2023/04/05');
```

NPresup	FICHA	monto
70100	8003	150000
70103	8006	18500
70104	8007	9700

3 rows in set (0.00 sec)

La subconsulta muestra los presupuestos del rango de fechas del enunciado.

La consulta principal los ignora