

PARTICION DE TABLAS EN MYSQL

Victor Manuel Albornoz Panti

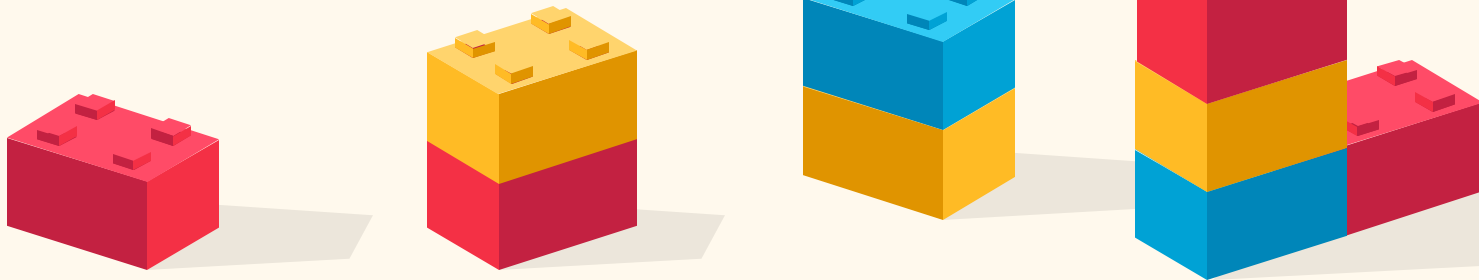


TABLA DE CONTENIDO

01

INTRODUCCIÓN

Explicación básica de qué es la partición de tablas.

03

IMPLEMENTACIÓN

Pasos para la creación de particiones en una tabla Mysql existente.

02

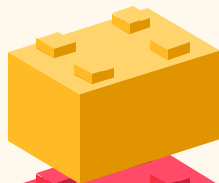
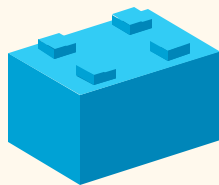
TIPOS

Descripción de los tipos de particiones.

04

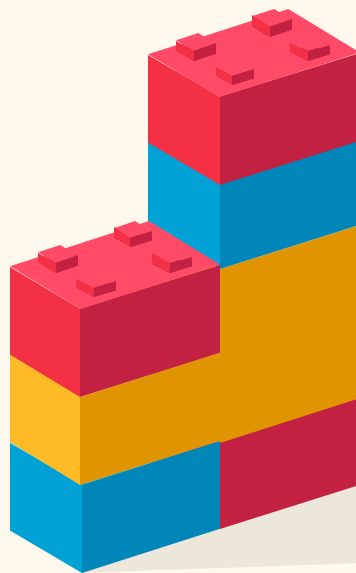
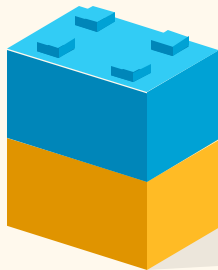
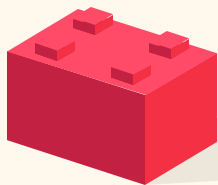
GESTIÓN

Métodos para agregar, eliminar o reorganizar particiones.



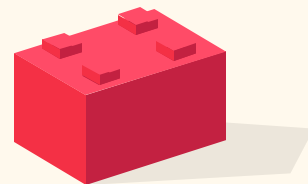
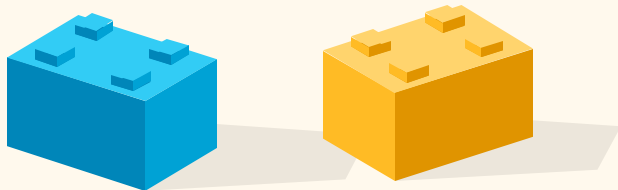
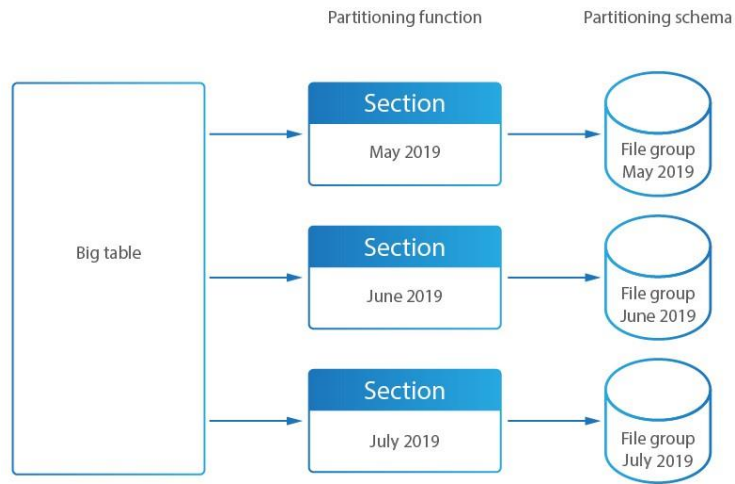
01

INTRODUCCIÓN A PARTICIONES



QUE ES LA **PARTICIÓN DE TABLAS?**

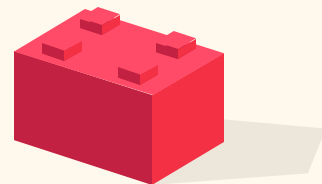
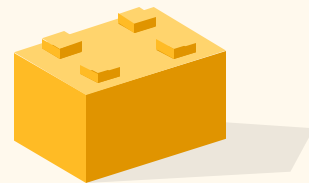
La partición de tablas en Mysql es una técnica que divide una tabla grande en partes mas pequeñas llamadas particiones. Cada partición actúa como una tabla independiente, pero todas están relacionadas como una única tabla lógica. Esto puede mejorar el rendimiento y la administración de grandes conjuntos de datos.



EXPLIQUEMOS CON LEGOS!



EXPLIQUEMOS CON LEGOS!



EXPLIQUEMOS CON LEGOS!



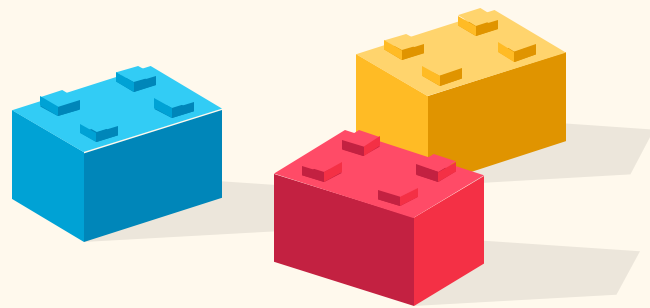
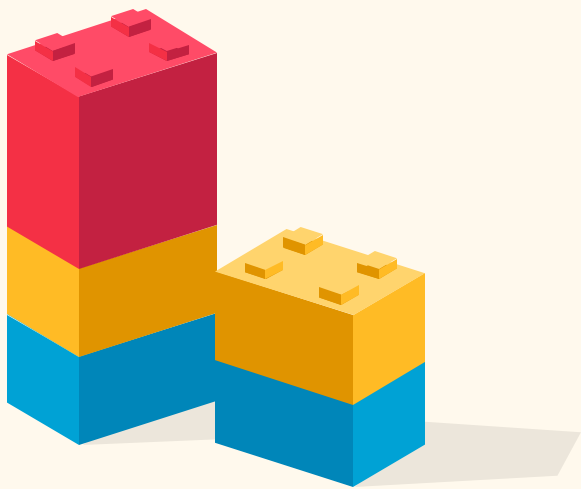
WHOA!!!

Mas rapido! Mas organizado! Mas elegante!



02

TIPOS

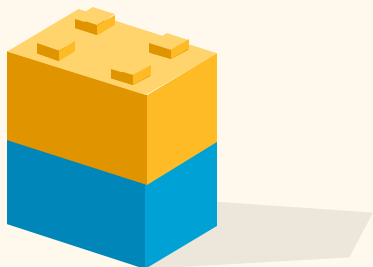


TIPOS DE PARTICIÓN



RANGO

Los datos se dividen en rangos basados en un valor específico.



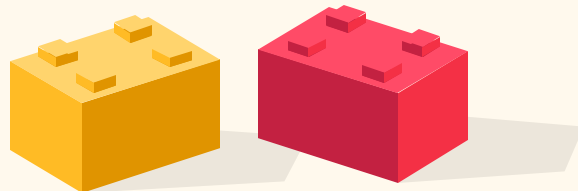
LISTA

Los datos se agrupan en listas definidas de datos.
(Ejem. País)



HASH

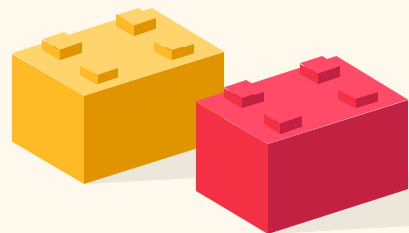
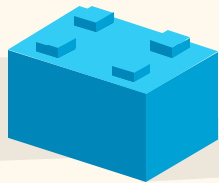
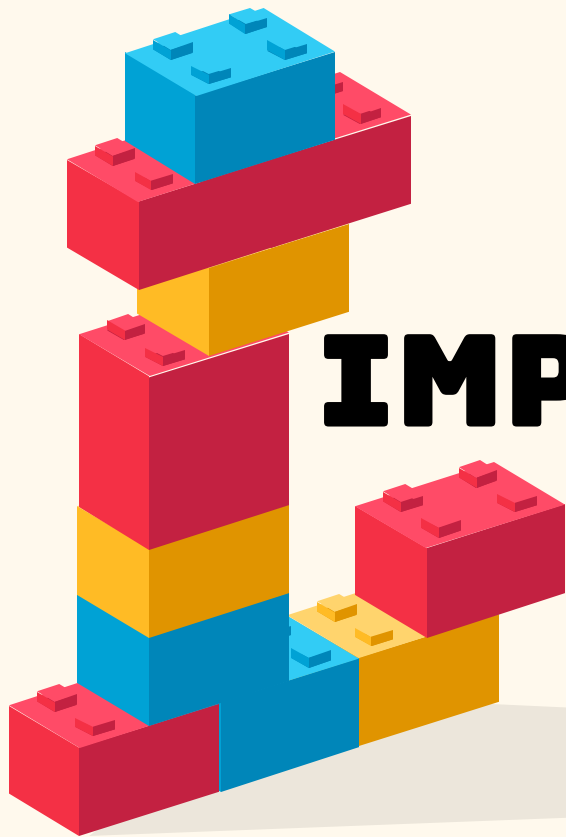
Se aplica una función hash a una columna y se asigna a una partición según el resultado del hash





03

IMPLEMENTACIÓN



COMO PUEDO IMPLEMENTARLO



LISTA

Particionamiento a base de una lista de departamentos.

```
-- Crear una tabla de empleados particionada por lista
CREATE TABLE empleados (
  id INT,
  nombre VARCHAR(50),
  salario DECIMAL(10, 2),
  departamento VARCHAR(50)
)
PARTITION BY LIST (departamento) (
  PARTITION p_ventas VALUES IN ('Ventas'),
  PARTITION p_marketing VALUES IN ('Marketing'),
  PARTITION p_it VALUES IN ('IT'),
  PARTITION p_otro VALUES IN (DEFAULT)
);
```



RANGO

Particionamiento a base de un rango de meses.

```
-- Crear una tabla particionada por rango de fechas por mes
CREATE TABLE registros (
  id INT,
  nombre_cliente VARCHAR(50),
  fecha_compra DATE
)
PARTITION BY RANGE (YEAR(fecha_compra), MONTH(fecha_compra)) (
  PARTITION p_2022_01 VALUES LESS THAN (2022, 2),
  PARTITION p_2022_02 VALUES LESS THAN (2022, 3),
  PARTITION p_2022_03 VALUES LESS THAN (2022, 4),
  -- ... continúa con más particiones para cada mes
  PARTITION p_2023_01 VALUES LESS THAN (2023, 2),
  PARTITION p_2023_02 VALUES LESS THAN (2023, 3),
  -- ... y así sucesivamente
  PARTITION p_future VALUES LESS THAN (MAXVALUE, MAXVALUE)
);
```

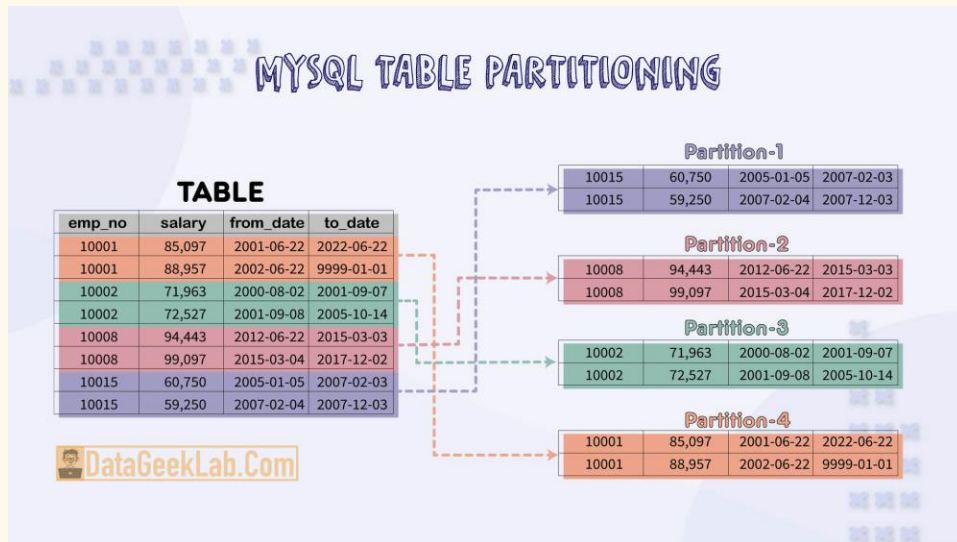
COMO PUEDO IMPLEMENTARLO



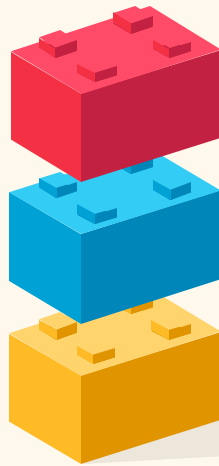
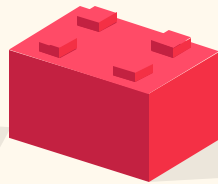
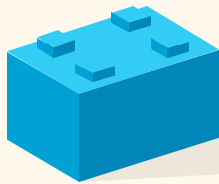
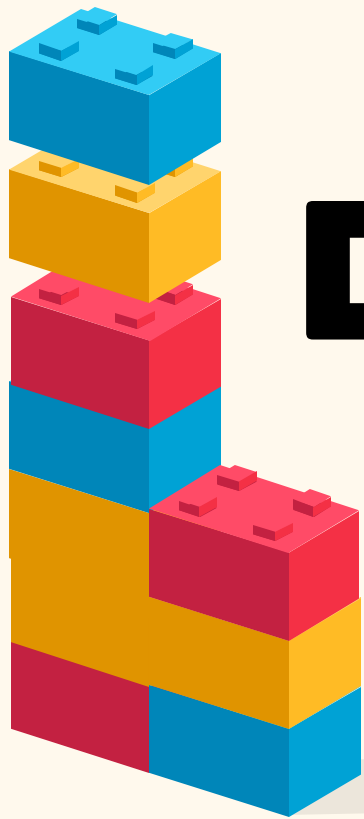
HASH

Particionamiento a base de un hash que identifica lotes de datos

```
CREATE TABLE usuarios (  
  user_id INT,  
  nombre VARCHAR(50),  
  email VARCHAR(50)  
)  
PARTITION BY HASH(user_id)  
PARTITIONS 4;
```

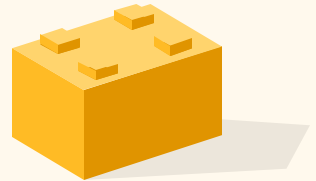
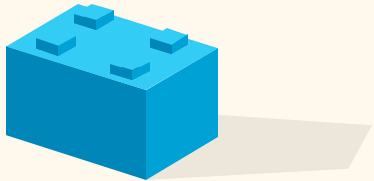


HORA DE PRUEBAS



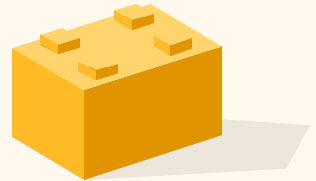
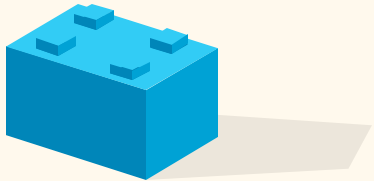
PRUEBAS SIN PARTICION

```
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
755 rows in set (13.93 sec)
```



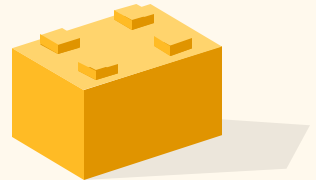
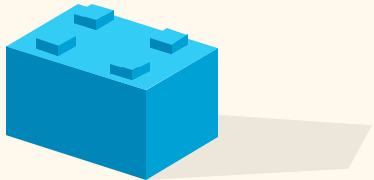
PRUEBAS CON PARTICION

```
+-----+-----+-----+-----+-----+-----+-----+-----+
|-----+-----+-----+-----+-----+-----+-----+-----|
|-----+-----+-----+-----+-----+-----+-----+-----|
|-----+-----+-----+-----+-----+-----+-----+-----|
|-----+-----+-----+-----+-----+-----+-----+-----|
|-----+-----+-----+-----+-----+-----+-----+-----|
|-----+-----+-----+-----+-----+-----+-----+-----|
|-----+-----+-----+-----+-----+-----+-----+-----|
|-----+-----+-----+-----+-----+-----+-----+-----|
755 rows in set (1.20 sec)
```



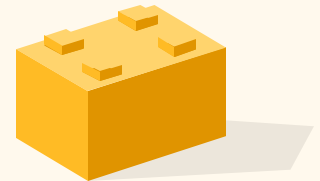
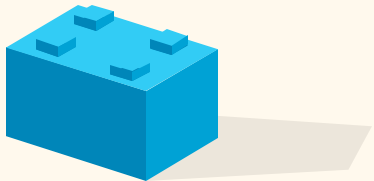
PRUEBAS CON PARTICION

```
| 0 | sp-508@atenea | ANA MARINA | SA  
+-----+-----+-----+-----+-----+-----+-----+  
-----+-----+-----+-----+-----+-----+-----+  
-----+-----+-----+-----+-----+-----+-----+  
+-----+-----+-----+-----+-----+-----+-----+  
-----+-----+-----+-----+-----+-----+-----+  
-----+-----+-----+-----+-----+-----+-----+  
-----+-----+-----+-----+-----+-----+-----+  
+-----+-----+-----+-----+-----+-----+-----+  
2543 rows in set (1.96 sec)
```



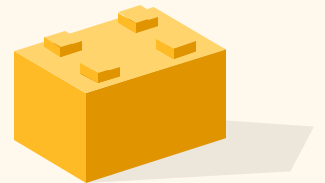
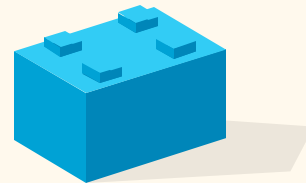
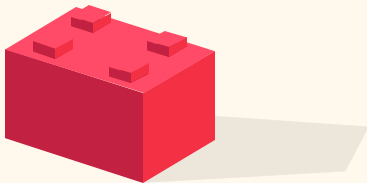
PRUEBAS CON PARTICION

```
o | sp-500@atenea | ANA MARIA  
+-----+-----+-----+-----+  
+-----+-----+-----+-----+  
+-----+-----+-----+-----+  
+-----+-----+-----+-----+  
+-----+-----+-----+-----+  
+-----+-----+-----+-----+  
+-----+-----+-----+-----+  
+-----+-----+-----+-----+  
2543 rows in set (0.86 sec)
```



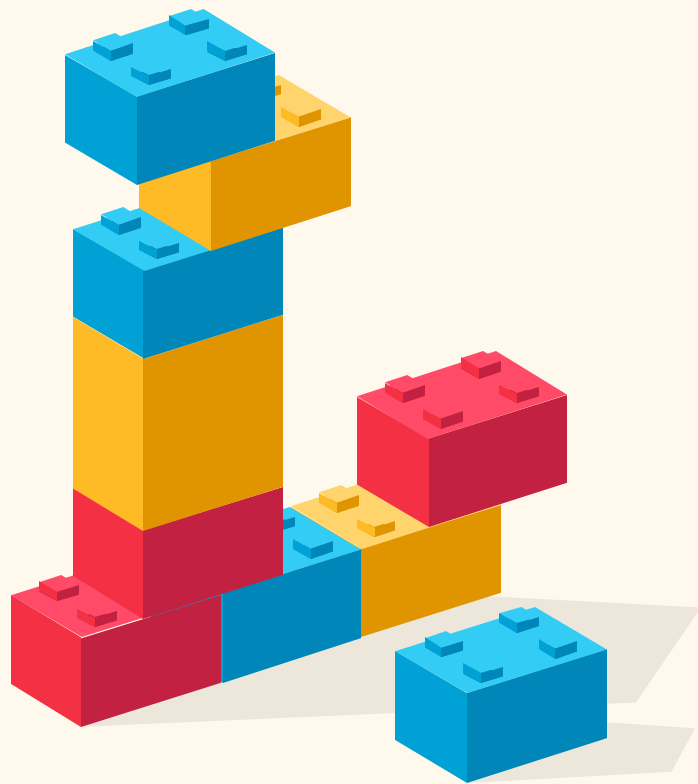
TIEMPOS

TABLA	REGISTROS	TIEMPO
V1	755	27.76s
V1	717	19.16s
V2	755	3.30s
V2	717	0.74s



04

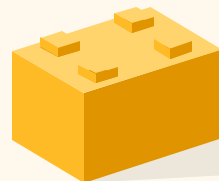
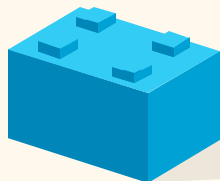
GESTION



COMO REVISAR PARTICIONES

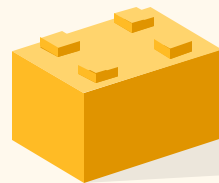
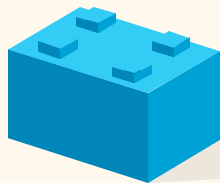
```
SELECT
  table_name,
  subpartition_name,
  table_rows,
  subpartition_ordinal_position
FROM
  information_schema.partitions
WHERE
  table_name = 'wisp_factura_facturas2';
```

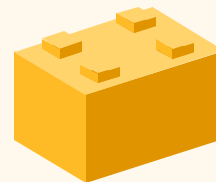
```
SHOW CREATE TABLE wisp_factura_facturas2;
```



COMO REVISAR PARTICIONES

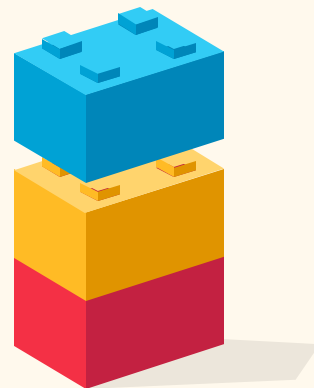
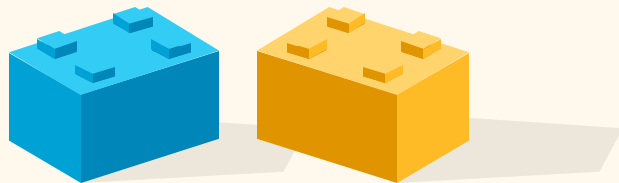
```
) ENGINE=InnoDB DEFAULT CHARSET=utf8
/*!50100 PARTITION BY RANGE (YEAR(fecha_pago)*100 + MONTH(fecha_pago))
(PARTITION p0 VALUES LESS THAN (201601) ENGINE = InnoDB,
PARTITION p1 VALUES LESS THAN (201602) ENGINE = InnoDB,
PARTITION p2 VALUES LESS THAN (201603) ENGINE = InnoDB,
PARTITION p3 VALUES LESS THAN (201604) ENGINE = InnoDB,
PARTITION p4 VALUES LESS THAN (201605) ENGINE = InnoDB,
PARTITION p5 VALUES LESS THAN (201606) ENGINE = InnoDB,
PARTITION p6 VALUES LESS THAN (201607) ENGINE = InnoDB,
PARTITION p7 VALUES LESS THAN (201608) ENGINE = InnoDB,
PARTITION p8 VALUES LESS THAN (201609) ENGINE = InnoDB,
PARTITION p9 VALUES LESS THAN (201610) ENGINE = InnoDB,
PARTITION p10 VALUES LESS THAN (201611) ENGINE = InnoDB,
PARTITION p11 VALUES LESS THAN (201612) ENGINE = InnoDB,
PARTITION p12 VALUES LESS THAN (201701) ENGINE = InnoDB,
PARTITION p13 VALUES LESS THAN (201702) ENGINE = InnoDB,
PARTITION p14 VALUES LESS THAN (201703) ENGINE = InnoDB,
PARTITION p15 VALUES LESS THAN (201704) ENGINE = InnoDB,
PARTITION p16 VALUES LESS THAN (201705) ENGINE = InnoDB,
PARTITION p17 VALUES LESS THAN (201706) ENGINE = InnoDB,
PARTITION p18 VALUES LESS THAN (201707) ENGINE = InnoDB,
```





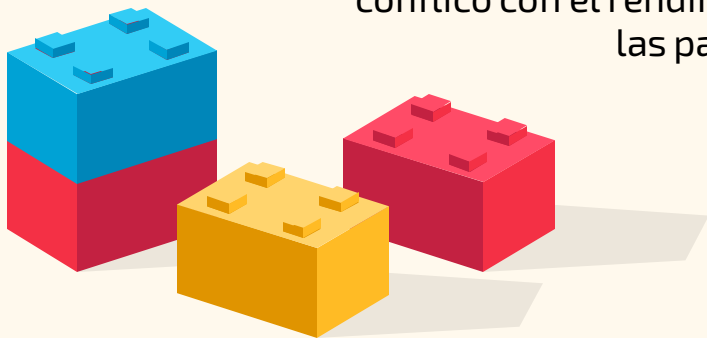
FILTRAR

Cuidado con al no realizar los filtros incluyendo el campo que se uso para la partición.



CONFIGURACIÓN DE MYSQL

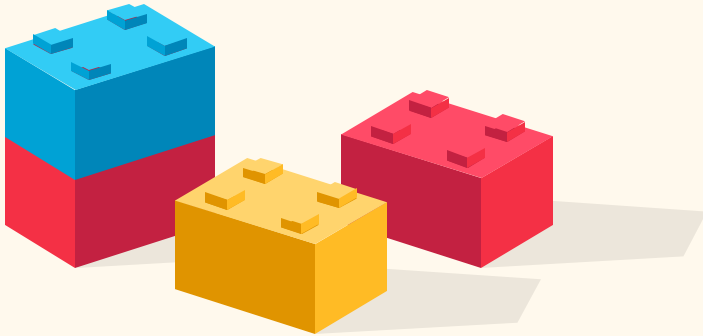
Es necesario tener cuidado con la configuración de mysql, ya que esta puede entrar en conflicto con el rendimiento de las particiones



CREACIÓN DE NUEVAS PARTICIONES

```
-- Añadir particiones para el año 2022 a una tabla particionada por rango de fecha
ALTER TABLE registros
ADD PARTITION (
  PARTITION p_2022_01 VALUES LESS THAN (2022, 2),
  PARTITION p_2022_02 VALUES LESS THAN (2022, 3),
  PARTITION p_2022_03 VALUES LESS THAN (2022, 4),
  -- ... continúa con más particiones para cada mes del año 2022
  PARTITION p_2022_12 VALUES LESS THAN (2023, 1)
);
```

Es necesario tener cuidado con la configuración de mysql, ya que esta puede entrar en conflicto con el rendimiento de las particiones



PROPIEDADES DE TABLAS MYSQL

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
	int(10) unsigned	NO	MUL	NULL	
	date	NO	MUL	NULL	
	date	NO	MUL	NULL	
	datetime	NO	MUL	NULL	
	smallint(5) unsigned	NO	MUL	NULL	
	decimal(15,2)	NO		NULL	
	decimal(15,2)	NO		NULL	
	decimal(15,2)	NO		NULL	
	decimal(15,2)	NO		NULL	
	decimal(15,2)	NO		NULL	
	varchar(200)	NO	MUL	NULL	
	tinyint(1)	NO	MUL	NULL	
	int(11)	YES	MUL	NULL	
	int(11)	YES	MUL	NULL	
	int(11)	YES	MUL	NULL	
	int(11)	YES	MUL	NULL	
	int(11)	NO	MUL	NULL	
	longtext	NO		NULL	
	tinyint(1)	NO		NULL	
	varchar(200)	NO	MUL	NULL	
	decimal(10,2)	NO		NULL	
	varchar(200)	NO		NULL	
	decimal(15,2)	NO		NULL	
	varchar(100)	NO	MUL	NULL	
	varchar(50)	NO	MUL	NULL	
	decimal(10,2)	NO		NULL	
	varchar(300)	NO		NULL	
	varchar(255)	YES		NULL	
	decimal(15,2)	NO		NULL	
	decimal(15,2)	NO		NULL	
	decimal(10,2)	NO		NULL	
	decimal(15,2)	NO		NULL	
	decimal(15,2)	NO		NULL	
	decimal(15,2)	NO		NULL	
	tinyint(1)	NO		NULL	
	tinyint(1)	NO		NULL	
	int(10) unsigned	YES	MUL	NULL	
	longtext	YES		NULL	
	varchar(10)	YES	MUL	NULL	
	int(11)	YES	MUL	NULL	
	int(11)	YES	MUL	NULL	

Field	Type	Null	Key	Default	Extra
id	int(11)	NO		0	
	int(10) unsigned	NO		NULL	
	date	NO		NULL	
	date	NO		NULL	
	datetime	NO		NULL	
	smallint(5) unsigned	NO		NULL	
	decimal(15,2)	NO		NULL	
	decimal(15,2)	NO		NULL	
	decimal(15,2)	NO		NULL	
	decimal(15,2)	NO		NULL	
	decimal(15,2)	NO		NULL	
	varchar(200)	NO		NULL	
	tinyint(1)	NO		NULL	
	int(11)	YES		NULL	
	int(11)	YES		NULL	
	int(11)	YES		NULL	
	int(11)	YES		NULL	
	int(11)	NO		NULL	
	longtext	NO		NULL	
	tinyint(1)	NO		NULL	
	varchar(200)	NO		NULL	
	decimal(10,2)	NO		NULL	
	varchar(200)	NO		NULL	
	decimal(15,2)	NO		NULL	
	varchar(100)	NO		NULL	
	varchar(50)	NO		NULL	
	decimal(10,2)	NO		NULL	
	varchar(300)	NO		NULL	
	varchar(255)	YES		NULL	
	decimal(15,2)	NO		NULL	
	decimal(15,2)	NO		NULL	
	decimal(10,2)	NO		NULL	
	decimal(15,2)	NO		NULL	
	decimal(15,2)	NO		NULL	
	tinyint(1)	NO		NULL	
	tinyint(1)	NO		NULL	
	int(10) unsigned	YES		NULL	
	longtext	YES		NULL	
	varchar(10)	YES		NULL	
	int(11)	YES		NULL	
	int(11)	YES		NULL	



GRACIAS!

