



El empleo
es de todos

Mintrabajo

Clase SQL update, delete, limit, like



@SENAcomunica

www.sena.edu.co

UPDATE -ACTUALIZAR



UPDATE nombre_tabla

SET campo_columna1 = "valor_nuevo",

campo_columna1 = "valor_nuevo"

WHERE campo_principal = valor

palabra reservada

nombre de la tabla

columna a cambiar

valor nuevo

llave principal o condición

EJEMPLO UPDATE



Si queremos cambiar el apellido2 'BENITO' por 'RODRIGUEZ' ejecutaremos:

```
UPDATE personas  
SET apellido2 = 'RODRIGUEZ'  
WHERE nombre = 'ANTONIO'  
AND apellido1 = 'GARCIA'  
AND apellido2 = 'BENITO'
```

Ahora la tabla 'personas' quedará así:

nombre	apellido1	apellido2
ANTONIO	PEREZ	GOMEZ
LUIS	LOPEZ	PEREZ
ANTONIO	GARCIA	RODRIGUEZ
PEDRO	RUIZ	GONZALEZ

EJEMPLO UPDATE

1

```
UPDATE cartelera SET titulo_pelicula = "Rapidos y furiosos 9", fecha_estreno = "2020"  
WHERE codigo_cartelera = 10
```

2

```
UPDATE cartelera SET titulo_pelicula ="Saw 9",  
fecha_estreno = "2019"  
WHERE id_cartelera = 10 and formato = "2D"
```

3

```
UPDATE empleados SET estado_sistema = "inactivo"  
WHERE id_empleado = 123
```

DELETE -BORRAR



palabra reservada

nombre de la tabla

```
DELETE FROM nombre_tabla  
WHERE campo_principal = valor
```

Condición para poder borrar

EJEMPLO DELETE

Si queremos borrar a la persona LUIS LOPEZ PEREZ, podemos ejecutar el comando:

```
DELETE FROM personas  
WHERE nombre = 'LUIS'  
AND apellido1 = 'LOPEZ'  
AND apellido2 = 'PEREZ'
```

La tabla 'personas' resultante será:

nombre	apellido1	apellido2
ANTONIO	PEREZ	GOMEZ
ANTONIO	GARCIA	RODRIGUEZ
PEDRO	RUIZ	GONZALEZ

EJEMPLO DELETE

1

```
DELETE FROM cartelera WHERE id_cartelera = 2
```

2

```
DELETE FROM series  
where nombre_serie = "Dark" AND temporada = "3"  
.
```

3

```
DELETE FROM marca_producto WHERE id-marca = 1 and  
estado_marca = "repetida"
```

LIMIT



```
SELECT * FROM nombre_tabla  
LIMIT 2
```

palabra reservada

cantidad de
registros

EJEMPLO LIMIT



Ejemplo SQL TOP para MySQL:

Dada la siguiente tabla 'personas', quiero obtener los 2 primeros valores.

nombre	apellido1	apellido2
ANTONIO	PEREZ	GOMEZ
ANTONIO	GARCIA	RODRIGUEZ
PEDRO	RUIZ	GONZALEZ

```
SELECT * FROM personas LIMIT 2
```

Obtendríamos el siguiente resultado:

nombre	apellido1	apellido2
ANTONIO	PEREZ	GOMEZ
ANTONIO	GARCIA	RODRIGUEZ

LIKE



El operador **LIKE** se utiliza en la cláusula **WHERE** para buscar por un patrón.

Sintaxis de SQL LIKE

```
SELECT columna(s) FROM tabla WHERE columna LIKE '%patron%'
```

Ejemplos del uso de SQL LIKE

Dada la siguiente tabla 'personas'

nombre	apellido1	apellido2
ANTONIO	PEREZ	GOMEZ
ANTONIO	GARCIA	RODRIGUEZ
PEDRO	RUIZ	GONZALEZ

Si quiero seleccionar los nombres que empiezan por 'AN' en la tabla 'personas', ejecutaría el comando siguiente:

```
SELECT * FROM personas  
WHERE nombre LIKE 'AN%'
```

LIKE



El caracter '%' es un comodín, que sirve para uno o más caracteres.

Este es el resultado

nombre	apellido1	apellido2
ANTONIO	PEREZ	GOMEZ
ANTONIO	GARCIA	RODRIGUEZ

Otro ejemplo de SQL LIKE

Para seleccionar las personas que tienen un 'Z' en su apellido1, ejecutaríamos:

```
SELECT * FROM personas  
WHERE apellido1 LIKE '%Z%'
```

nombre	apellido1	apellido2
ANTONIO	PEREZ	GOMEZ
PEDRO	RUIZ	GONZALEZ

EJEMPLO like

1

```
SELECT * FROM cartelera WHERE titulo_pelicula like "A%"
```

2

```
SELECT * FROM cartelera WHERE titulo_pelicula like "%A%"
```

3

```
SELECT titulo_pelicula, pais_pelicula, genero FROM  
cartelera WHERE genero like "T%"
```

IN



Con "in" averiguamos si el valor de un campo dado está incluido en la lista de valores especificada

```
SELECT * FROM pedido where PaisDestinatario
in ("Alemania","Brazil","Suiza")
```

Diagram illustrating the SQL query structure with annotations:

- `SELECT *`: todos las columnas
- `FROM pedido`: tabla
- `where`: condicional
- `in`: palabra reservada para lista
- `("Alemania","Brazil","Suiza")`: lista entre " " y separada por comas (,)

Todos los datos de pedidos enviados a Alemania, Brazil y Suiza

ejemplos



Datos relevantes de pedidos enviados en estas fechas 31 de agosto, septiembre 7,8 y 10

```
SELECT FechaPedido, IdCliente, FechaEntrega, DireccionDestinatario,  
CiudadDestinatario FROM pedido where FechaEntrega  
in ("2018-08-31","2018-09-07","2018-09-08","2018-09-10")
```

Datos de los empleados que trabajan en la dependencia de Sistemas, Contabilidad y Recursos Humanos

```
SELECT nombre_empleado, correo_empleado, dir_empleado  
FROM empleado where Dependencia_trabajo  
in ("Sistemas", "Contabilidad","Recursos Humanos" )
```

ejemplos



Datos de los empleados que trabaja en el bloque 5,7 y 9 cuyos jefes son Yuber Ferrero, Diego Marín

```
SELECT * FROM empleado where bloque in (5,7,9) and  
Nombre_jefe in("Yuber Ferrero","Diego Marin")
```

Datos de aprendices de la formación de análisis y sistemas donde tienen como instructores a Donald Beltrán y Arle Morales

```
SELECT * FROM aprendiz where formacion in("ADSI","Sistemas")  
and Instructor in ("Donald Beltran", "Arle Morales")
```

ejemplos



Datos de instructores que no tengan como profesión ingeniero civil ni arquitectos

```
SELECT * FROM Instructor  
where profesion not in ("Ingeniero civil", "Arquitecto")
```

Placas de vehículos que salieron a las 10 de la mañana y a las 2 de la tarde

```
SELECT placa, hora_salida from tabla_salida_vehiculo  
where hora_salida in ("10:00", "14:00")
```


between



"between" significa "entre". Averiguamos si el valor de un campo dado está entre los valores mínimo y máximo especificados

```
SELECT numero_factura, valor_venta, nombre_cliente  
from listado_facturas where fecha_venta  
between "2018-08-15" and "2018-09-15"
```

Diagram illustrating the SQL query using the `between` operator. Red arrows point from the word `entre` to the `between` keyword and the date range `"2018-08-15" and "2018-09-15"`.

ejemplos



Numero de la ficha y nombre de la formación que estén entre la 1095200 y la 1095300

```
SELECT numero_ficha, nombre_ficha from Fichas_Sena  
where numero_ficha between 1095200 and 1095300
```

Datos de salarios de los empleados que ganan entre 1.000.000 y 2.000.000

```
select * from salarios  
where valor_salario between 1000000 and 2000000
```

ejemplos



Libros que el precio esta entre los 20000 mil pesos y los 40000 mil pesos

```
select * from libros
  where precio between 20000 and 40000;
```

Placas de carros con placas entre ass214 y cab513 que entraron entre el 1 y el 19 de septiembre

```
select placa, fecha_entrada from entrada_parqueo
where placa between "ass214" and "cab513" and
fecha_entrada between "2018-09-01" and "2018-09-19"
```

alias



Un "alias" se usa como nombre de un campo o de una expresión o para referenciar una tabla cuando se utilizan más de una tabla (tema que veremos más adelante).

```
select count(nombre_empleados)
```

```
as 'nombre' from empleados
```



palabra reservada



alias o nombre para el conteo

ejemplos




```
select * from empleados as e
```

esta tabla tiene un alias de "e"

A red arrow originates from the text "esta tabla tiene un alias de 'e'" and points to the alias "e" in the SQL query above.

```
SELECT IdPedido, PrecioUnidad,  
       Cantidad as 'cantidad de productos pedidos'  
FROM detalles_de_pedido
```

A red arrow originates from the text "Deben ser para alias con comilla simple (')" and points to the string "'cantidad de productos pedidos'" in the SQL query above.

Deben ser para alias con comilla simple (')

Funciones de fechas



1

`adddate(fecha, interval expresion)`: retorna la fecha agregándole el intervalo especificado. Ejemplos:
`adddate('2006-10-10',interval 25 day)` retorna "2006-11-04".
`adddate('2006-10-10',interval 5 month)` retorna "2007-03-10".

```
select adddate("2020-09-01", interval 10 day)
as 'fecha siguiente'
```

```
select adddate("2020-09-01", interval 2 month)
as 'meses siguientes'
```

3

`current_time`: retorna la hora actual con formato "HH:MM:SS" o "HHMMSS".

```
select CURRENT_TIME() as 'hora actual'
```

2

`current_date`: retorna la fecha de hoy con formato "YYYY-MM-DD" o "YYYYMMDD".

```
select Current_date()
as Fecha_actual
```

4

`datediff(fecha1,fecha2)`: retorna la cantidad de días entre fecha1 y fecha2.

```
select datediff("2020-09-01","2020-09-17")
as 'cantidad dias'
```

Funciones de fechas



5

dayname(fecha): retorna el nombre del día de la semana de la fecha. Ejemplo:

```
select dayname("2020-09-01")  
AS 'día de la semana'
```

7

extract(tipo from fecha): extrae partes de una fecha.

```
select extract(year from "2020-08-31")  
as 'año de la fecha'
```

6

dayofweek(fecha): retorna el índice del día de semana para la fecha pasada como argumento. Los valores de los índices son: 1=domingo, 2=lunes,... 7=sábado). Ejemplo: dayofweek('2006-08-10') retorna 5, o sea jueves.

```
select dayofweek("2020-09-01")  
AS 'numero de dia'
```

8

hour('18:25:09') retorna "18";
-minute(hora): retorna los minutos de la hora dada, en el rango de 0 a 59.
-monthname(fecha): retorna el nombre del mes de la fecha dada.

```
select hour("14:02:01")  
as horas
```

```
select minute("14:02:01")  
as horas
```

1. Elabore la siguiente la tabla producto: y realice las siguientes consultas:

Idproducto	Nombrepo	categoria	cantiunidad	Preciounidad	Estado
1	Cerveza	Bebida	20	1800	Activo
2	Gaseosa	Bebida	30	1000	Activo
3	Energizante	Bebida	40	2500	Activo
4	Leche	Bebida	30	2200	Activo
5	Te	Bebida	30	800	Activo
6	Malboro	cigarrillos	50	2000	Activo
7	Kool	cigarrillos	50	2000	Inactivo
8	Derby	cigarrillos	50	2000	Activo
9	Lentejas	Granos	18	1200	Activo
10	frijoles	Granos	25	1200	Activo
11	Garbanzo	Granos	4	1500	Activo
12	Arvejas	Granos	1	2500	Activo
13	Tomate	verduras	100	300	Activo
14	cebolla	Verduras	100	500	Activo
15	Zanahoria	Verduras	150	300	Activo
16	Ajo	Verduras	1	100	Inactivo
17	Banano	fruta	80	200	Activo
18	Piña	futa	30	5000	Activo
19	fresa	fruta	800	100	Activo

consultas:

- 1) Listado de categorías.
- 2) Nombre de los productos ordenados descendientemente.
- 3) Nombre de los productos con su cantidad que son granos o verduras
- 4) Productos con su categoría que valen más de mil pesos y que se encuentran activos.
- 5) Productos que su precio es menor igual a 100 pesos
- 6) Nombre del producto, categoría y precio de los productos cuya cantidad es menor de 10 artículos y cuyo estado es activo.
- 7) todos los datos de los artículos cuyo estado es inactivo.
- 8) Frutas y verduras cuyo precio está entre los 100 y los 2000 y las cantidades son 100 0 300 ordenados ascendentemente.
- 9) id del producto, nombre del producto, precio; donde el precio es igual a 100 pesos y están activos.

2. Elabore la siguiente la tabla canción y realice las siguientes consultas:



Id	nombrecancion	cantante	duracion	clase
1	Una vida para recordar	Piso 21	3.52	Reggaetón
2	No vaya a ser	Pablo Alboran	4.10	pop
3	Con calma	Daddy Yankee	4.98	Reggaetón
4	expectativas	Bunbury	3.80	pop
5	Amor prohibido	Luis Fonsi	4.10	pop
6	Calma	Farruko	4.98	Reggaetón
8	Cuando lleguen los millones	Grupo Kvrass	3.90	vallenato
9	No Te Vayas	Carlos vives	4.98	vallenato
10	Amigos nada mas	Altafulla	3.10	champeta
11	Pa todo el año	Pipe Bueno	4.03	popular
12	Dulce pecado	Jessi Uribe	4.00	popular
13	Azul	J Balvin	5.2	Reggaetón
14	si yo no te vuelvo a ver	Yatra	4.80	Reggaetón

Consultas:



- 1) Listado de los géneros (sin repetir)
- 2) Listado de cantantes (sin repetir)
- 3) Listado de canciones
- 4) Canciones que canta yatra
- 5) Canciones que canta Luis Fonsi y altafulla
- 6) Canciones cuyo género es el rock
- 7) Nombre del cantante de la canción desconocidos
- 8) Listado de canciones de Farruco, Bunbury y Carlos Vives
- 9) Listado de películas con duración entre 3.50 y 4.20

Actividad 4 para realizar

Desarrolle la
base de datos
en
phpmyadmin

Empleado								
id	Nom_emp	Ape_emp	Salario_e	Subsidio_tran	Eps	Pension	Riesgo	Seguro_fune
1	Pedro	Gonzalez	700.000	80.000	Medimas	colfondos	1	olivos
2	Maria	Perez	900.000	85.000	saludtotal	colfondos	1	olivos
3	Maria	Martinez	800.000	82.000	SOS	colfondos	1	olivos
4	Andrea	suarez	500.000	80.000	colsanitas	colpension	1	olivos
5	Rosalba	figueroa	1.000.000	90.000	nueva_eps	colpension	1	ofrenda
6	Patricia	Martinez	1.000.000	90.000	nueva_eps	colpension	2	ofrenda
7	Alex	Pelaez	1.000.000	90.000	nueva_eps	colpension	1	ofrenda
8	Felipe	Martinez	900.000	85.000	nueva_eps	proteccion	1	ofrenda
9	Andres	Perez	450.000	80.000	SOS	proteccion	2	ofrenda
10	Maria	Gonzalez	3.200.000	-	SOS	proteccion	3	ofrenda
11	Teresa	suarez	700.000	80.000	SOS	proteccion	1	ofrenda
12	Sebastian	Pereira	700.000	80.000	Medimas	proteccion	1	sura
13	Alexa	Ortiz	700.000	80.000	Medimas	proteccion	1	sura
14	Leonel	alvarez	380.000	80.000	saludtotal	proteccion	2	sura
15	Andrea	cespedez	250.000	80.000	colsanitas	proteccion	2	ofrenda
16	Efrain	Rodriguez	4.200.000	-	colsanitas	proteccion	2	sura
17	Rubiela	Rodriguez	3.400.000	-	SOS	colfondos	1	sura

Elabore la base de datos para la tabla empleado y genere el siguiente script de código en SQL.



1

Inserte los datos partir de la fila 10 por SQL

2

Borre los datos del registro 17 con SQL.

3

Actualice las personas que tienen seguro funerario SURA por Colseguros

4

Cambie el apellido "Martinez" por "Bejarano"

5

Consulte todos los datos de los empleados solo hasta el registro 9

6

Consulte nombre y salarios de los empleados solo de los 5 primeros registros

7

Consulte los nombre de las personas que empiezan por "A"

8

Consulte los nombres de las personas que empiezan por "M" o su apellido empieza por "S"

9 Consulte las personas cuyo apellido es Suarez o pelaez.

10 Consulte los nombres de las empresas de seguros funerarios sin repetir.

11 Consulte los empleados que su apellido es "Pereira"

12 Consulte las personas que ganas menos de 1.000.000

13 Consulte listado de personas que tienen su pensión en protección y cuyo seguro funerario es de la ofrenda.

14 Consulte los empleados que ganan menos de 1.000.000 cuya salud la pagan en Medimas y su seguro funerario lo tienen en Colseguros.

15 Saque una lista de empleados que su subsidio de transporte es de 80.000

16 Sacar listado de EPS a las que están afiliados los empleados de la empresa.

17 Consulte el listado de personas que tienen pensión en protección y ganan más de 2.500.000



GRACIAS

Línea de atención al ciudadano: 018000 910270
Línea de atención al empresario: 018000 910682



@SENAcomunica

www.sena.edu.co