Obteniendo la fecha de hoy

Con la función DATE() podemos obtener la fecha de hoy. Por ejemplo, la podemos utilizar en la claúsula WHERE para obtener todos lo registros de hoy.

SELECT \* FROM usuarios WHERE fecha\_registro = DATE();

También es posible indicar explícitamente a la función que la fecha deseada es la de hoy. Ejemplo:

SELECT \* FROM usuarios WHERE fecha\_registro = DATE('now');

Todo lo que hemos aprendido de SQL hasta ahora sirve en todos los motores de base de datos como SQLITE, PostgreSQL o MySQL. Esta función es un caso especial ya que recibe distintos nombres en cada uno de los motores. Por ejemplo, en MySQL se utiliza CURDATE(), y en Microsoft SQL Server se utiliza GETDATE() A la hora de buscar documentación es importante dejar claro que motor se está ocupando. En este tutorial interactivo estamos ocupando SQLite.

Ejercicio

Se tiene una tabla llamada tareas con las siguientes columnas: "id" (identificador único), "descripcion" (descripción de la tarea) y "fecha\_limite" (fecha límite para completar la tarea).

Obtén la descripción de todas las tareas que tengan fecha\_limite igual a la fecha actual .

select descripcion from tareas where fecha\_limite = date('now');

*Obteniendo la fecha de mañana*

EN SQL es posible sumar fechas para obtener fechas futuras. En SQLite lo podemos lograr pasando un segundo argumento a la función DATE. Esto suena complicado pero es mas sencillo de lo que parece:

DATE('now', '1 day')

En este ejemplo, estamos sumando 1 día a la fecha de hoy (now). Si queremos sumar más días, por ejemplo 5 días, utilizaremos DATE('now', '5 day').

También es posible sumar semanas y meses con:

2 Semanas: DATE('now', '2 week') 3 Meses: DATE('now', '3 month')

En una consulta esto se vería de la siguiente forma:

SELECT \* FROM tabla where fecha > DATE('now', '2 week')

Al sumar el intervalo de tiempo, el sistema calculará automáticamente la fecha correcta.

Ejercicio

Se tiene una tabla de tareas con los campos id, descripcion y fecha\_limite. Se pide seleccionar todos los campos de las tareas que tienen como fecha límite el día de mañana.

select \* from tareas where fecha\_limite = DATE ('now', '1 day');

Obteniendo la fecha de ayer

Así como es posible sumar fechas, también es posible restarlas:

DATE('now', '-1 day') DATE('now', '-1 week')

Es importante aclarar que cuando no especificamos el signo, se asume que es positivo, esto quiere decir que

DATE('now', '1 day')

es lo mismo que

DATE('now', '+1 day')

Ejercicio

Supongamos que tenemos una tabla llamada ganancias con las columnas "id" (identificador único), "fecha" (fecha de registro) y "monto" (ganancia del día).

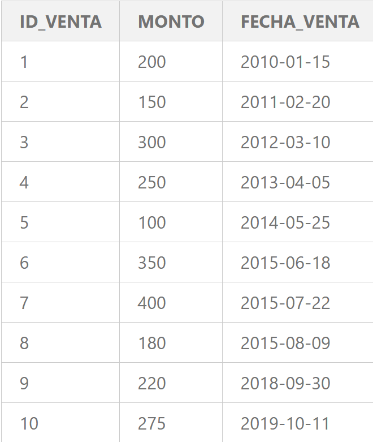
Muestra el monto correspondiente al día de ayer.

select monto from ganancias where fecha = date ('now', '-1 day');

*Extracción del año*

Para ciertos reportes, es muy probable que nos pidan extraer información de una fecha, como, por ejemplo, el año en que se hizo una transacción.

Analicemos el siguiente escenario:

Se tiene la tabla ventas con la siguiente información:

Nos piden mostrar toda la información de la tabla y adicionalmente agregar una columna con el año de la venta.

SELECT \*, strftime('%Y', fecha\_venta) as año\_venta FROM ventas El resultado de esta consulta será el siguiente:



Para mostrar los resultados de este tipo de funciones, es necesario asignar un nombre a la nueva columna, ya que, de lo contrario, la columna resultante mantendrá el nombre de "strftime('%Y', fecha\_venta)", lo cual resultaría en una denominación poco legible para un informe.

Ejercicio

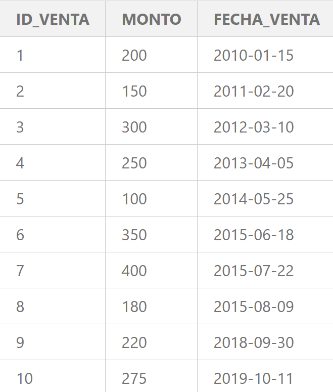
Dada una **tabla ventas** con las columnas monto y fecha\_venta, crea una consulta que muestre únicamente el monto y el año de la venta. La columna que muestre el año de la venta debe llamarse año\_venta

SELECT monto, strftime('%Y', fecha\_venta) as año\_venta FROM ventas

Extracción del mes

Podemos extraer el mes de una fecha de manera similar a la extracción del año, utilizando nuevamente la función strftime.

Siguiendo con nuestro ejemplo de la tabla de ventas, si deseamos agregar una columna que indique únicamente el mes de la venta, podemos utilizar la siguiente consulta:



SELECT strftime('%m', columna) FROM tabla

En este caso, para obtener el mes, pasamos %m como argumento a la función strftime.

Ejercicio

Dada la **tabla ventas** previamente presentada con las columnas monto y fecha\_venta, crea una consulta que muestre una tabla con el monto, el mes de la venta y el año de la venta, en ese mismo orden. La columna para el mes de la venta debe llamarse mes\_venta y aquella para el año de la venta debe llamarse año\_venta

SELECT monto, strftime('%m', fecha\_venta) as mes\_venta ,

strftime('%Y',fecha\_venta) as año\_venta

FROM ventas

*Extracción del mes y año*

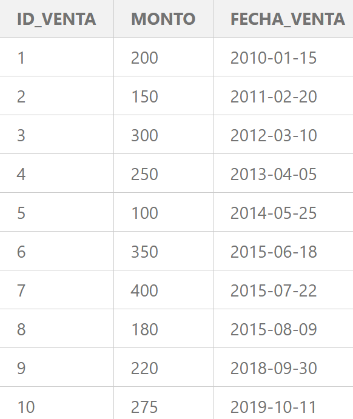
Ya aprendimos a extraer el mes y el año de una fecha. Sin embargo, cómo podriamos extraer ambos datos en una sola columna?

Para extraer tanto el mes como el año de una fecha en una sola columna, puedes utilizar la función strftime('%m-%Y'). Esto te permitirá obtener un resultado en el formato "mes-año". Veamos un ejemplo utilizando una tabla de ventas:

*Ejercicio*

Dada la tabla ventas con las columnas monto y fecha\_venta, crea una consulta que muestre las siguientes dos columnas:

Monto

El mes y año de la fecha de venta. Esta columna debe llamarse mes\_año

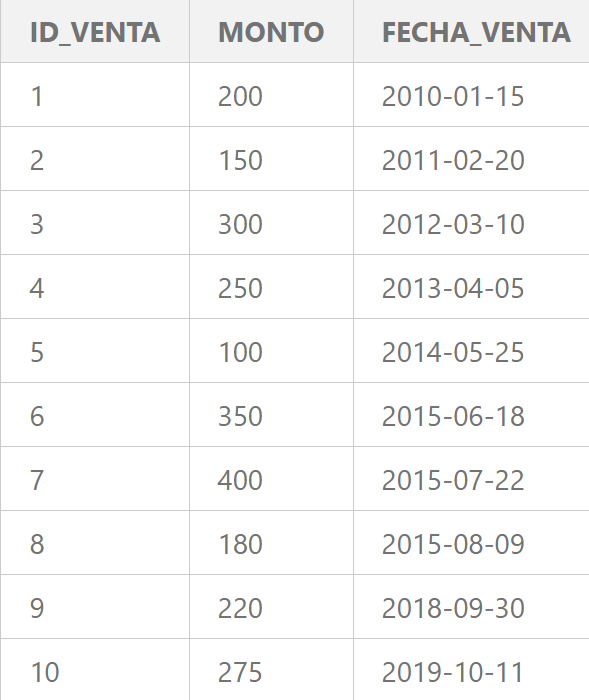
Resultado.

select monto, strftime('%m-%Y',fecha\_venta) as mes\_año from ventas

*Extracciones y where*

Previamente aprendimos a filtrar utilizando como parámetro una fecha. Ahora utilizaremos lo aprendido para filtrar fechas de un año o mes en específico.

Se tiene la tabla ventas con la siguiente información:



Nos piden mostrar todas las ventas del año 2012. Para esto utilizaremos la función strftime para extraer el año de las fechas, y luego filtraremos por el año indicado:

SELECT \* FROM ventas WHERE strftime('%Y', fecha\_venta) = '2012';

*Ejercicio*

Dada una tabla ventas con las columnas monto y fecha\_venta, selecciona toda la información de las ventas del 2015

select \* from ventas WHERE strftime('%Y', fecha\_venta) = '2015';