

## SQL -Structured Query Language (lenguaje de consulta estructurada)

**SQL (Structured Query Language)** es un lenguaje de programación diseñado para almacenar, manipular y recuperar datos almacenados en bases de datos relacionales. La primera encarnación de SQL apareció en 1974, cuando un grupo de IBM desarrolló el primer prototipo de una base de datos relacional. Relational Software (luego se convirtió en Oracle) lanzó la primera base de datos relacional comercial.

### CONSULTAS BÁSICAS

**SELECT** Campo1, Campo2/\*(que significa TODOS)...  
**FROM** Tabla  
**ORDER BY** Campo1 le decimos que esté ordenado por campo1  
**DESC** (que sirve para que haga el orden inverso,)  
Campo2 (poner [ ] si hay espacios en el nombre del campo)

**WHERE** Especifica criterios de campo que cada registro debe cumplir

**TOP** La cláusula TOP indica que en el resultado no deben aparecer todas las filas resultantes sino un cierto número de registros, las n primeras.

**BETWEEN** expresión **AND** expresión

**DATE()** devuelve la fecha del día  
**YEAR**(fecha) Devuelve el año de una fecha  
**MONTH**(fecha) Devuelve el mes de una fecha  
**DAY**(fecha ) Devuelve el día de una fecha

### CARACTERES COMODÍN

CARÁCTER	DESCRIPCIÓN	EJEMPLO
*	Hace coincidir cualquier número de caracteres. Puede utilizar el asterisco (*) en cualquier sitio de una cadena de caracteres.	qu* encuentra qué, quién y quizás pero no aquellos ni aunque.
?	Hace coincidir un carácter alfabético individual en una posición concreta.	b?l encuentra bala, billete y bola.
[ ]	Hace coincidir los caracteres incluidos entre los corchetes.	b[ao]l encuentra bala y bola pero no billete.
!	Excluye los caracteres incluidos entre los corchetes.	r[!oc]a encuentra risa y rema pero no roca ni rosa. Igual que "[a]*" encuentra todos los elementos que no empiezan con la

## SQL -Structured Query Language (lenguaje de consulta estructurada)

		letra a.
-	Hace coincidir cualquier intervalo de caracteres. Recuerde que debe especificar los caracteres en orden ascendente (de la A a la Z, no de la Z a la A).	a[m-s]a encuentra ama, ata y asa.
#	Hace coincidir cualquier carácter numérico.	1#3 encuentra 103, 113 y 123.

**LIKE** para buscar valores en un campo que coincidan con un patrón. Por ejemplo un nombre que tenga 5 letras camponombre LIKE '?????'

**AS** Crea un alias para una tabla

### OPERADORES EN ACCESS

Los **operadores aritméticos** sirven para calcular un valor de dos o más números o cambiar el signo de un número de positivo a negativo o viceversa.

OPERADOR	FINALIDAD	EJEMPLO
+	Sumar dos números.	[Subtotal] + [Impuesto sobre las ventas]
-	diferencia entre dos números o indicar el valor negativo de un número.	[Precio] - [Descuento]
*	Multiplicar dos números	[Cantidad] * [Precio]
/	Dividir el primer número entre el segundo número.	[Total] / [Número_de_elementos]
\	Redondear ambos números a enteros, dividir el primer número entre el segundo número y, a continuación, truncar el resultado en un entero.	[Registros] \ [Habitaciones]
Residuo	Dividir el primer número entre el segundo número y, a continuación, devolver solo el resto.	[Registros] Residuo [Habitaciones]
^	Elevar un número a la potencia de un exponente.	Número ^ Exponente

Operadores de **comparación** para comparar valores y devuelven un resultado que es verdadero, Falso o nulo.

OPERADOR	FINALIDAD	EJEMPLO
<, <=	Devuelve verdadero si el primer valor es menor o, menor o igual que el segundo valor.	Valor1 < valor2 Valor1 < = valor2

## SQL -Structured Query Language (lenguaje de consulta estructurada)

>, >=	Devuelve verdadero si el primer valor es mayor o, mayor o igual que el segundo valor.	Valor1 > valor2 Valor1 > = valor2
=	Devuelve verdadero si el primer valor es igual que el segundo valor.	Valor1 = valor2
<>	Devuelve verdadero si el primer valor no es igual que el segundo valor. NOTA: En todos los casos, si el primer valor o el segundo valor es null, el resultado es, a continuación, también null. Dado que nulo representa un valor desconocido, el resultado de cualquier comparación con un valor nulo es también desconocido.	Valor1 <> valor2

Utilizar los operadores **lógicos** para combinar dos valores booleanos y devolver un resultado verdadero, Falso o nulo. Operadores lógicos también se conocen como operadores booleanos.

OPERADOR	FINALIDAD	EJEMPLO
AND	Devuelve True cuando Expr1 y Expr2 son verdaderas.	Expr1 AND Expr2
OR	Devuelve True cuando Expr1 OR Expr2 es true.	Expr1 OR Expr2
Eqv	Devuelve True cuando Expr1 y Expr2 son verdaderas, o cuando Expr1 y Expr2 tienen el valor false.	Expr1 Eqv Expr2
Not	Devuelve True si Expr no es true.	NOT Expr
Xor	Devuelve True cuando Expr1 es true o Expr2 es true, pero no en ambos.	Expr1 OEx Expr2

Operadores de **concatenación** para combinar dos valores de texto en uno solo.

OPERADOR	FINALIDAD	EJEMPLO
&	Combina dos cadenas en una sola cadena.	cadena1 & cadena2
+	Combina dos cadenas para formar una cadena y propaga los valores nulos (si un valor nulo, toda la expresión se evalúa como null).	cadena1 + cadena2