

RESUMEN SQL

COMO SE ESCRIBE UNA CONSULTA

SELECT: campos

FROM + JOIN: tablas de las que saca los datos

WHERE: criterios que se aplican directamente a los datos (**aplica a TODOS los registros**)

GROUP BY: agrupación de los datos seleccionados

HAVING: criterios que se aplican a los datos ya agrupados/con una FDA-función de dominio agregado (max, min, avg, sum, count) (**aplica/filtra SOLO los resultados de una FDA**)

ORDER BY: ordena los datos según algún criterio

CÓMO LO PROCESA LA COMPUTADORA

FROM + JOIN: primero rescata los datos de la/las tablas

WHERE: después aplica los criterios simples/FDA

GROUP BY: luego agrupa los datos

HAVING: luego aplica los criterios los resultados del primer filtro

ORDER BY: ordena los datos

SELECT: finalmente extrae los campos que queremos

SUBCONSULTA

Se usa para incluir campos de una tabla en la consulta (generalmente en el WHERE) sin tener que hacer un JOIN entre las dos tablas. Se puede incluir en el ORDER BY para ordenar sin que sea visible.

SELECT: campos Tabla A

WHERE: campos Tabla A OPERADOR (SELECT fda/campos FROM Tabla B)

SELECT

SELECT ALL (o SELECT): muestra todos los resultados de esos campos

SELECT TOP X: selecciona los x resultados de arriba de la tabla, después de ordenarlos

SELECT TOP X PERCENT: selecciona el x% de resultados de arriba de la tabla, después de ordenarlos

SELECT DISTINCT: selecciona solo las combinaciones de datos distintos, es decir, evita que se repitan

SELECT *: selecciona todos los campos de la tabla en el FROM

OPERADORES (WHERE o HAVING)

LIKE: indicar que un campo sea similar a algo (Ej: nombres LIKE "a*", "*a", "*a*")

Tipo de coincidencia	Modelo Planteado	Coincide	No coincide
Varios caracteres	'a*a'	'aa', 'aBa', 'aBBBa'	'aBC'
Carácter especial	'a[*]a'	'a*a'	'aaa'
Varios caracteres	'ab**'	'abcdefg', 'abc'	'cab', 'aab'
Un solo carácter	'a?a'	'aaa', 'a3a', 'aBa'	'aBBBa'
Un solo dígito	'a#a'	'a0a', 'a1a', 'a2a'	'aaa', 'a10a'
Rango de caracteres	'[a-z]'	'f', 'p', 'j'	'2', '&'

Fuera de un rango	'[!a-z]'	'9', '&', '%'	'b', 'a'
Distinto de un dígito	'[!0-9]'	'A', 'a', '&', '~'	'0', '1', '9'
Combinada	'a[!b-m]#'	'An9', 'az0', 'a99'	'abc', 'aj0'

IN/NOT IN: que el campo esté o no dentro de otra tabla/consulta/grupo de datos (Ej: nombre IN ("juan", "pedro", "martin"); titulo NOT IN (SELECT titulo FROM tabla))

BETWEEN: que un número/fecha este entre otros (Ej: fechainicio BETWEEN "fecha1" AND "fecha2")

AND: un campo tiene que cumplir **todas** las condiciones

OR: un campo tiene que cumplir **una** de las condiciones

NOT: un campo no es algo (Ej: WHERE NOT nombre = "juan") - puede ser escrito como nombre <> "juan")

DES/IGUALDADES: < (menor), > (mayor), <> (diferente), = (igual), <= (menor o igual), >= (mayor o igual)

TIPOS DE ORDER BY

DESC: descendente (Z-A, 10-0)

ASC: ascendente (A-Z, 0-10)

FUNCIONES SQL

TEXTO Y CARACTERES

LEN: largo/cantidad de caracteres que contiene un texto (Ej: LEN("casa")=4; LEN("yo voy")=6)

TRIM: Elimina los espacios delante o detrás del texto (Ej: TRIM("casa ") "casa"; TRIM("yo voy")="yo voy")

LTRIM: Elimina los espacios delante (izquierda) del texto

RTRIM: Elimina los espacios detrás (derecha) del texto

UCASE: convierte a mayúsculas un texto (Ej: UCASE("Hola")="HOLA")

LCASE: convierte a minúsculas un texto (Ej: LCASE("Hola")="hola")

LEFT: devuelve la cantidad indicada de caracteres de un texto desde la izquierda (Ej:

LEFT("hola",1)="h"; LEFT("hola",3)="hol")

RIGHT: devuelve la cantidad indicada de caracteres de un texto desde la derecha (Ej:

RIGHT("hola",1)="a"; RIGHT("hola",3)="ola")

MID: devuelve una cantidad de caracteres de un texto a partir de un número de carácter (Ej:

MID("hola",2,1)="o"; MID("hola",2,2)="ol")

NUMÉRICAS Y DE FECHA

INT: parte entera de un número, redondea para abajo (Ej: INT(36.7)=36; INT(1.99989)=1; INT(-10.85)=-11)

ROUND: redondea un número a la cantidad de decimales especificados (Ej: ROUND(36.7579,3)=36.757; ROUND(1.97989,0)=2; ROUND(-10.85,1)=-10.9)

DAY: devuelve el día de una fecha (Ej: MONTH(12/5/2008)=12)

MONTH: devuelve el mes de una fecha (Ej: MONTH(12/5/2008)=5)

YEAR: devuelve el año de una fecha (Ej: MONTH(12/5/2008)=2008)

DATE(): devuelve la fecha de hoy

NOW(): devuelve la fecha y la hora de ahora

HOUR(NOW()): devuelve la hora de ahora

MINUTE(NOW()): devuelve los minutos de la hora actual

SECOND(NOW()): devuelve los segundos de la hora actual

DATEDIFF: calcula la diferencia entre dos fechas - DATEDIFF("formato", fecha inicial, fecha final)

FORMATOS DE DATEDIFF

"yyyy": años

"m": meses

"d": días

"h": horas

"n": minutos

"s": segundos

"q": trimestres

LÓGICAS

IIF: iif(condición, verdadero, falso)

OPERADORES ARITMÉTICOS

ESTÁNDAR: + - * /

\: cociente de la división entera entre 2 números (Ej: 9\2=4)

MOD: resto de la división entera entre 2 números (9 MOD 2 = 1)

FUNCIÓN DE DOMINIO AGREGADO (FDA)

AVG: promedio de los valores de un campo

COUNT: número de registros de la selección

SUM: suma de todos los valores de un campo

MAX: valor más alto de un campo

MIN: valor más bajo de un campo

TIPOS DE CONSULTA

FUNCIÓN FDA DENTRO DEL WHERE

Para insertar una función de dominio agregado en una consulta que no tiene GROUP BY, esta se puede agregar al WHERE haciendo una subconsulta:

SELECT función de valor agregado FROM tabla

EJ: WHERE importe < (SELECT avg(importe) FROM compras)

TRANSFORM (tablas de doble entrada)

TRANSFORM: función de dominio agregado/FDA (max, min, avg, sum, count)

SELECT: campos (primera entrada - FILAS)

FROM + JOIN: tablas de las que saca los datos

GROUP BY: primera entrada - FILAS

PIVOT: segunda entrada - COLUMNAS

PARAMETERS (tablas de doble entrada)

PARAMETERS: parámetros y el tipo de dato que se ingresa (datetime, numeric, text[num caracteres], currency, etc)

SELECT: campos

FROM + JOIN: tablas de las que saca los datos

GROUP BY: agrupación de los datos seleccionados

HAVING: criterios aplicados a la FDA

CREAR TABLAS NUEVAS CON CONSULTA

SELECT: campos que quiero en la tabla nueva

INTO: nombre de la tabla nueva

FROM: tabla de la que saco los datos y campos

GENERAR UN REGISTRO NUEVO EN UNA TABLA

PARAMETERS: [parametro1] tipo, [parametro2] tipo, etc

INSERT INTO: tabla a la que se deben agregar los registros (campo1, campo2, etc)

VALUES: (valor.campo1, valor.campo2, etc) ó (parametro1, parametro2, etc) ó mezcla

UPDATE (actualizar tabla)

PARAMETERS: [parametro1] tipo, [parametro2] tipo, etc

UPDATE tabla/s a actualizar

SET: (campo1=[parametro1], campo2=[parametro2], etc) ó (campo1=valor1, campo2=valor2, etc)

DELETE (borrar filas enteras de datos)

PARAMETERS: [parametro1] tipo, [parametro2] tipo, etc

DELETE: campos (generalmente *)

FROM: tabla

WHERE: campo1=[parametro1], campo2=[parametro2], condicion1, condicion2, etc

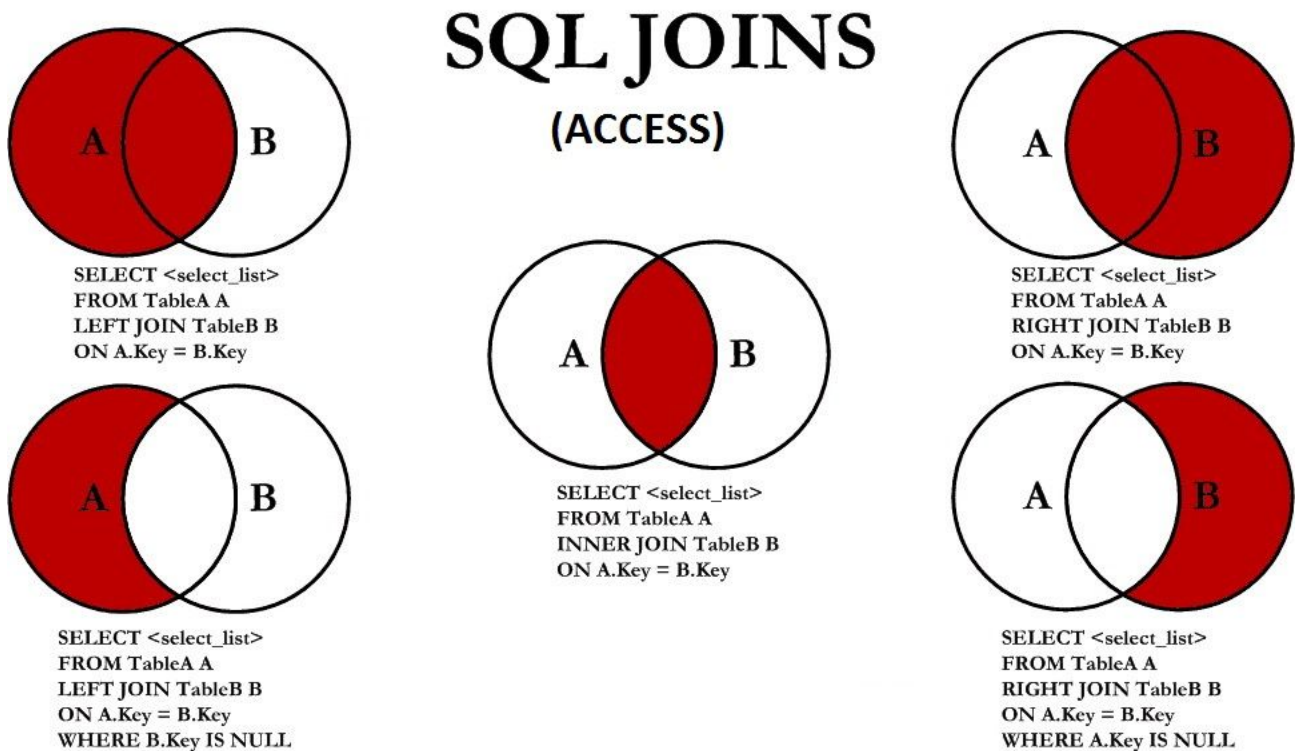
IMPORTANTE

- Si se usa SELECT TOP, **siempre** se usa ORDER BY
- PARAMETERS → [nombre campo] tipo.dato → **siempre** se vuelve a usar en el resto del código
- DELETE **nunca** se usa con JOIN
- UPDATE **nunca** se usa con JOIN en consultas, **solo** en tablas
- Si en un JOIN un campo esta en mas de una tabla **siempre** se pone TableA.Campo en todo el código

- & → CONCATENAR <> → DIFERENTE (≠)
- **Todos** los campos del SELECT que no son FDA **siempre tienen** que estar en el GROUP BY
- Una FDA **nunca** puede estar en el WHERE, hay que hacer una subconsulta
- HAVING **siempre** va después de GROUP BY
- SELECT ... INTO TableA crea la TableA
- En datediff → "m"=meses, "h"=horas, "n"=minutos

TIPOS DE JOIN

Cuando se unen dos tablas con inner join, el campo que los vincula (TableA.key=TableB.key) debe ser de valor único en alguna de las tablas (que no se repita en alguna de las tablas)



NULL LEFT/RIGHT JOIN ($A \cap B'$)

```
SELECT
FROM TableA LEFT/RIGHT JOIN TableB
ON A.key=B.key
WHERE B.    key is null
```

ó

```
SELECT
FROM TableA
WHERE A.key NOT IN (key from TableB)
```