

SQL- Apuntes

SET DATEFORMAT

Utilizamos esto cuando queremos insertar fecha pero **lo insertamos con otro formato**, distinto al que nos pide el proyecto/s → evita problemas de compatibilidades

Especificamos que queremos insertar una fecha en este caso con el mes, día y año:

```
SET DATEFORMAT 'MDY'  
INSERT INTO CONTACTOS (apellido, nombre, fecha_Nacimiento)  
VALUES ('Welz', 'Alex', '09/10/1996')
```

De esta forma nos deja realizar el INSERT sin problemas.

DATEDIFF → EDAD

Es el resultado del cálculo de la edad utilizando la función **DATEDIFF** que se **encarga de realizar restas entre fechas**

```
DATEDIFF(YEAR, U.FechaNacimiento, GETDATE()) AS EDAD FROM Usuarios U
```

```
SELECT Legajo, Apellido, Nombre DATEDIFF(YEAR,0, GETDATE()-fecha_Nacimiento)  
AS EDAD FROM ALUMNOS
```

YEAR, MONTH

De esta forma podemos obtener columnas que indiquen el mes y año de una tabla:

```
SELECT APELLIDO, 1 AS UNO, 'Hola mundo' AS HOLA, MONTH(FECHA_NACIMIENTO) AS  
MES, YEAR(FECHA_NACIMIENTO) AS ANIO FROM ALUMNOS
```

Consultas de selección

- ▼ Las consultas de Selección nos permiten obtener datos de una o más tablas de una base de datos.
- ▼ Permite, mediante un lenguaje intuitivo y de alto nivel, obtener información de la base de datos para luego procesarlos, resumirlos, ordenarlos, agruparlos, etc.
- ▼ Permite obtener todos los registros de una tabla o aplicar un filtro y obtener un conjunto de ellos.
- ▼ Permite obtener información de todas las columnas de una tabla o especificar un conjunto de ellas.

▼ **SELECT ALL**

Devuelve todos los registros de una tabla

▼ **SELECT DISTINCT**

Ignora los registros cuyos campos seleccionados coincidan completamente

▼ **SELECT TOP**

Devuelve una determinada cantidad de registros de la tabla.

SELECT ALL

Anteriormente, obteníamos **todos los registros** de la tabla de la cual seleccionábamos datos. Sin importar si existieran registros duplicados o no

SELECT DISTINCT → Evita tener datos duplicados.

SELECT TOP

Utilizaremos la expresión TOP para indicar que queremos que sólo un grupo principal de registros. Se puede utilizar de estas dos formas:

- SELECT TOP (10) o SELECT TOP (10%)

- Se puede incorporar la sentencia **WITH TIES** si se desea que la última fila repita los registros que se encuentran igualados.

Utilizamos PERCENT siempre con TOP o LIMIT, permite a su vez poder tener el porcentaje de los resultados.

```
SELECT TOP (50) PERCENT YEAR(FECHA_NACIMIENTO) AS 'FECHA_NAC' FROM ALUMNOS
```

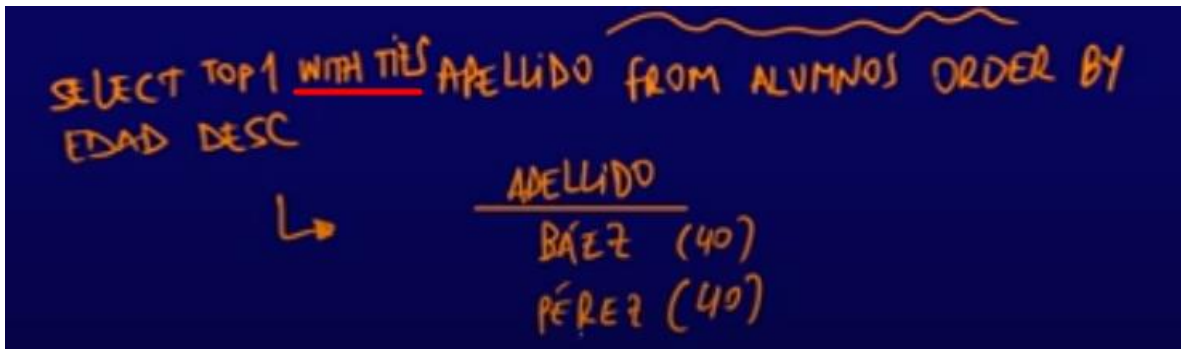
	FECHA_NAC
1	1990
2	1984

Por otro lado, WITH TIES también se utiliza solo con TOP o LiMIT pero igualmente este mismo busca también los que **están empatados**

```
SELECT TOP (1) WITH TIES YEAR(FECHA_NACIMIENTO) AS 'FECHA_NAC' FROM ALUMNOS ORDER BY YEAR(FECHA_NACIMIENTO) ASC
```

	FECHA_NAC
1	1984
2	1984

Todos los alumnos que tengan 40 años pertenecerán a la lista por ser los alumnos mas viejos y tener WITH TIES



ORDER BY

```
SELECT <col1, col2, colN> FROM <tabla>
ORDER BY colX [ASC | DESC]
```

Siendo colX la columna por la cual se ordena y el criterio está dado por ASC → ascendente ó DESC → descendente.

ASC → Menor a Mayor – A a la Z

DESC → Mayor a Menor – Z a la A

Existen funciones:

IN, BETWEEN (puede ser n° o cadenas), YEAR, MOUNTH

OPERADOR LIKE

'%ER%' → contenga ER en alguna parte de la cadena

Todos los que finalicen con EZ:

```
-SELECT * FROM CONTACTOS WHERE APELLIDO LIKE '%EZ'
```

Todos los que tengan la 3er letra R por tener 2 gion bajo:

```
SELECT * FROM CONTACTOS WHERE NOMBRE LIKE '__R%'
```

Todos los que contengan 2da letra una vocal y al tener% después cualquier letra:

```
SELECT * FROM CONTACTOS WHERE MAIL LIKE '_[AEIOU]%'
```

Todas las direcciones que no tengan ningún numero:

```
SELECT * FROM CONTACTOS WHERE DIRECCION NOT LIKE '%[0-9]%'
```

LIKE 'Ale_' --> Busca todos los que tengan Ale y despues 1 carater cualquiera

Ejemplo: Alex, Ale4, etc.

LIKE '[a-m]ario' --> Busca registros que comiencen con a y m y luego continue con ario

Ejemplo: Dario, Mario.

LIKE '[kcmh]arina' --> comiencen con k,c,m o h y luego continue con ariana

Ejemplo: Marina, Karina, Carina, etc.

LIKE 'an[^g]%' --> comience con an que La 3er letra no sea g y continue con cualquier caracter

Ej: 'Analía', 'Antonio', 'Ana' pero NO 'Angel', 'Angie'

```
SELECT LEGAJO, APELLIDO, NOMBRE FROM ALUMNOS WHERE APELLIDO LIKE '%[A-Z][^R]ez'
```

	LEGAJO	APELLIDO	NOMBRE
1	1100	MENDEZ	JULIETA
2	1300	FERNANDEZ	NATALIA

El ejemplo anterior obtiene los datos de los alumnos cuyo apellido comience con cualquier caracter de la A a la Z y finalice con 'EZ' siempre en cuando la antepenúltima letra no sea la R.

```
SELECT * FROM TABLA WHERE
NOMBRE <> 'ANGEL'
```

Obtiene todos los registros de TABLA cuyo valor de NOMBRE sea distinto a 'ANGEL'

Otra forma es:

```
SELECT * FROM CONTACTOS WHERE NOT CODIGOPOSTAL = 1111 → ES IGUAL QUE <> DISTINTO
```

```
SELECT * FROM TABLA WHERE
APELLIDO LIKE '%EZ'
```

Obtiene todos los registros de TABLA cuyos apellidos finalicen con los caracteres 'EZ'.

```
SELECT * FROM TABLA WHERE
NOMBRE LIKE '_A%'
```

Obtiene todos los registros de TABLA cuyos nombres tengan como segundo caracter la letra A.

TIPO	MODELO	COINCIDE (DEVUELVE TRUE)	NO COINCIDE (DEVUELVE FALSE)
Caracteres múltiples (0-muchos)	a%a	Aa, aBa, aBBBa	aBC
	%ab%	abc, AABBB, Xab	aZb, bac
	ab%	ab, abc, abcdef	cab, aab
Caracter múltiple (solo 1)	a_a	aaa, aba, axa, a5a	aaaa, a123a, bac
Intervalo de caracteres	[a-z]	f, p, j, a, z	1, \$, &
Fuera de un intervalo	[^a-z]	1, 9, \$	f, p, j, a, z
Conjunto de caracteres	[axz]	a, x, z	b, c, d, 1, 5

```
--De esta forma podemos obtener nombres que tengan la 3er letra con una R por tener 2 guion bajo
SELECT * FROM CONTACTOS WHERE NOMBRE LIKE '__R%'
```

```
--Busca mails que tengan como 2do carater una vocal y despues cualquier caracter
SELECT * FROM CONTACTOS WHERE MAIL LIKE '_[AEIOU]%'
```

```
--Busca todas las direcciones que no tengan N°s
SELECT * FROM CONTACTOS WHERE DIRECCION NOT LIKE '%[0-9]%'
```