Funciones de sistema

CAST

CAST: es una función genérica que lo que hace es convertirnos un tipo de dato en otro siempre y cuando sea del mismo género, es decir, de un decimal a un número entero (numérico). Este cambio es solo a nivel de la operación NO a nivel real.

La columna UnitPrice en un inicio estaba configurada como MONEY

SELECT CAST(UnitPrice AS INT)
FROM Purchasing.PurchaseOrderDetail

Convertimos dos textos y los sumamos

SELECT CAST('5' AS INT) + CAST('5' AS INT)

CONVERT

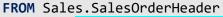
CONVERT(TipoDato, NombreColumna, Estilo)

Podemos utilizar un MIN y un MAX en el mismo SELECT. También podemos convertir formatos de fechas que vienen con la hora, a formato solo de fechas y en el orden que necesitemos.

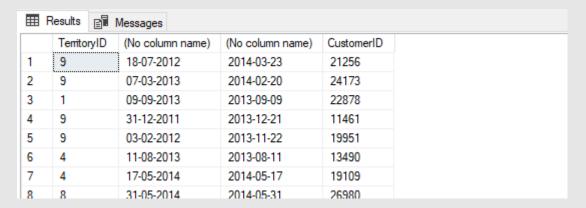
USE AdventureWorks2019
SELECT OrderDate
FROM Sales SalesOrderHeader

	Results 📳	Messages
	OrderDate	
1	2011-05-31	00:00:00.00
2	2011-05-31	00:00:00.00
3	2011-05-31	00:00:00.00
4	2011-05-31	00:00:00.00
5	2011-05-31	00:00:00.00
c	2011 05 21	nn-nn-nn nr

USE AdventureWorks2019



GROUP BY TerritoryID, CustomerID



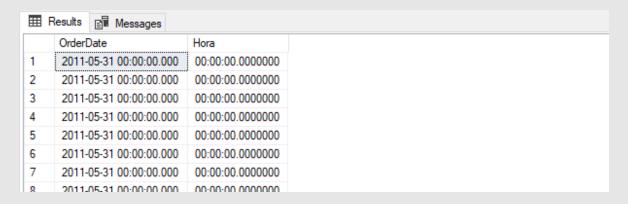
Convierto un tipo de datos DATETIME (traen la fecha y la hora) a DATE (solo la fecha)

USE AdventureWorks2019
SELECT OrderDate, CONVERT(DATE,OrderDate) AS Fecha
FROM Sales.SalesOrderHeader

****	Results Messag	es
	OrderDate	Fecha
1	2011-05-31 00:00:0	00.000 2011-05-31
2	2011-05-31 00:00:0	00.000 2011-05-31
3	2011-05-31 00:00:0	00.000 2011-05-31
4	2011-05-31 00:00:0	00.000 2011-05-31
5	2011-05-31 00:00:0	00.000 2011-05-31
6	2011-05-31 00:00:0	00.000 2011-05-31
7	2011-05-31 00:00:0	00.000 2011-05-31
8	2011-05-31 00:00:0	00.000 2011-05-31
q	2011-05-21 00-00-0	00 000 2011-05-21

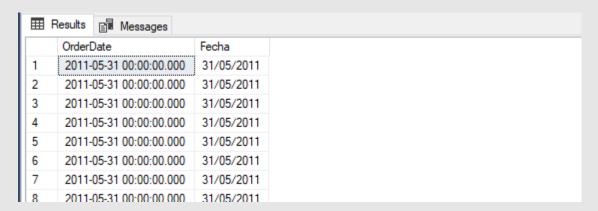
Extraigo la hora, conviertiendo el dato DATETIME a TIME

USE AdventureWorks2019
SELECT OrderDate, CONVERT(TIME,OrderDate) AS Hora
FROM Sales.SalesOrderHeader



Convertimos la fecha al formato latinoamericano

USE AdventureWorks2019
SELECT OrderDate, CONVERT(VARCHAR(10),OrderDate,103) AS Fecha
FROM Sales.SalesOrderHeader



FORMAT

Nos trae el formato de fecha en **TIPO TEXTO**

USE sample
SELECT enter_date, FORMAT(enter_date,'dd-MM-yyyy') AS Fecha
FROM works_on

===	Results 📳 M	lessages
	enter_date	Fecha
1	2017-10-15	15-10-2017
2	2017-04-15	15-04-2017
3	2016-11-15	15-11-2016
4	2016-10-01	01-10-2016
5	2018-01-01	01-01-2018
6	2017-06-01	01-06-2017
7	2017-02-15	15-02-2017
Q	2017_08_01	01-08-2017

USE sample
SELECT enter_date, FORMAT(enter_date,'yyyy/MM') AS Fecha
FROM works_on

 	Results	el M	lessages
	enter_c	date	Fecha
1	2017-	10-15	2017/10
2	2017-0	04-15	2017/04
3	2016-	11-15	2016/11
4	2016-	10-01	2016/10
5	2018-0	01-01	2018/01
6	2017-0	06-01	2017/06
7	2017-0	02-15	2017/02
8	2017-0	18-N1	2017/08

COALESCE

COALESCE: Devuelve para una lista dada de expresiones a1,a2,... el valor de la primera expresión que no es NULL. Acepta más de 2 parámetros.

COALESCE(float, int) --> Si el primer valor tiene un tipo de dato definido y el segundo también el resultado tendrá el tipo de dato con mayor prioridad. En este caso devolvería el resultado con el tipo de dato float.

Orden de tipos de datos según prioridad:

- 1. sql_variant
- 2. xml
- 3. datetimeoffset
- 4. datetime2
- 5. datetime
- 6. smalldatetime
- 7. date
- 8. time
- 9. float
- 10. real
- 11. decimal
- 12. money
- 13. smallmoney
- 14. bigint
- 15. int
- 16. smallint
- 17. tinyint
- 18. bit
- 19. ntext
- 20. text
- 21. image
- 22. timestamp
- 23. uniqueidentifier
- 24. nvarchar (including nvarchar(max))
- 25. nchar
- 26. varchar (including varchar(max))
- 27. char
- 28. varbinary (including varbinary(max))
- 29. binary (lowest)

COALESCE(NULL, NULL) --> Si el primer valor tiene un tipo NULL y el segundo parámetro también, el resultado devolverá un ERROR.

```
DECLARE @ValorNulo INT = NULL,
       @Valor1 = 0,
       DATETIME = GETDATE()
/* INT(NULL) & INT */
SELECT COALESCE(@ValorNulo, @Valor2) AS [COALESCE]
Results Resages
    COALESCE
    10
/* INT(NULL) & FLOAT & DATETIME */
SELECT COALESCE(@ValorNulo, @Valor3) AS [COALESCE],
        COALESCE(@ValorNulo, @Valor3, @Fecha) AS [COALESCE]
Results 📳 Messages
    COALESCE COALESCE
     10,5
             1900-01-11 12:00:00.000
/* NULL & NULL */
SELECT COALESCE(NULL, NULL) AS [COALESCE]
```

ISNULL

ISNULL: Acepta 2 parámetros.

ISNULL(float, int) --> Si el primer valor tiene un tipo de dato definido, el resultado tendrá este tipo de dato

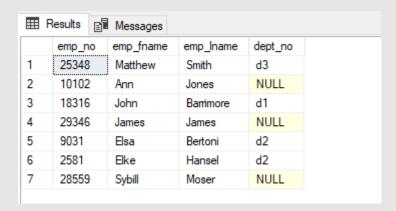
ISNULL(NULL, int) --> Si el primer valor tiene un tipo NULL, el resultado tendrá el tipo de dato del segundo parámetro (aunque este también sea nulo)

ISNULL(NULL, NULL) --> Si el primer valor tiene un tipo NULL y el segundo parámetro también, el resultado tendrá el tipo de dato INTEGER.

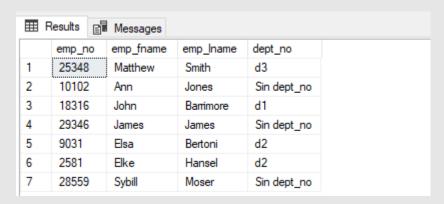
Ejemplo 1

USE Registro

SELECT * **FROM** employee



SELECT emp_no, emp_fname, emp_lname, ISNULL(dept_no,'Sin dept_no') 'dept_no'
FROM employee



CONCAT_WS

```
USE AdventureWorks2019
SELECT P.FirstName, P.LastName,
CONCAT(P.FirstName,' ',P.LastName)'Nombre Completo',
CONCAT_WS(' ',P.FirstName,P.LastName)'Nombre Completo',
P.FirstName + ' ' + P.LastName 'Nombre Completo'
FROM Person.Person P
```

===	Results 🗐	Messages			
	FirstName	LastName	Nombre Completo	Nombre Completo	Nombre Completo
1	Syed	Abbas	Syed Abbas	Syed Abbas	Syed Abbas
2	Catherine	Abel	Catherine Abel	Catherine Abel	Catherine Abel
3	Kim	Abercrombie	Kim Abercrombie	Kim Abercrombie	Kim Abercrombie
4	Kim	Abercrombie	Kim Abercrombie	Kim Abercrombie	Kim Abercrombie
5	Kim	Abercrombie	Kim Abercrombie	Kim Abercrombie	Kim Abercrombie
6	Hazem	Abolrous	Hazem Abolrous	Hazem Abolrous	Hazem Abolrous
7	Sam	Abolrous	Sam Abolrous	Sam Abolrous	Sam Abolrous
8	Humberto	Acevedo	Humberto Acevedo	Humberto Acevedo	Humberto Acevedo
9	Gustavo	Achong	Gustavo Achong	Gustavo Achong	Gustavo Achong
10	Pilar	Ackeman	Pilar Ackeman	Pilar Ackeman	Pilar Ackeman
11	Pilar	Ackeman	Pilar Ackeman	Pilar Ackeman	Pilar Ackeman
12	Aamn	Adams	Aamn Adams	Aaron Adams	Aamn Adams

Ejercicios

USE sample
SELECT enter_date
FROM works_on

	Results		M
	enter_	date	
1	2017-	10-1	,
2	2017-	04-1	,
3	2016-	11-1	,
4	2016-	10-0	
5	2018-	01-0	
6	2017-	06-0	
7	2017-	02-15	,
8	2017-	08-0	

III	Results Message
	(No column name)
1	2017-10
2	2017-4
3	2016-11
4	2016-10
5	2018-1
6	2017-6
7	2017-2
8	2017-8
9	2018-2