

# LES TYPES DE DONNÉES : **DATE ET HEURE**

Transact SQL

# PLAN

## I. Présentation

## II. La saisie de date et heure

1. L'instruction **SET DATEFORMAT**
2. La fonction **CONVERT**

## III. Afficher une date et heure

1. Afficher la date et heure au format par défaut
2. Convertir la date et heure

## IV. Les différents type de date et heure

## V. Manipuler les informations de type date et heure

1. La fonction **DATEADD**
2. La fonction **DATEDIFF**
3. La fonction **DAY**
4. La fonction **MONTH**
5. La fonction **YEAR**
6. La fonction **GETDATE**
7. La fonction **DATENAME**
8. La fonction **DATEPART**

## VI. Conclusion

# PRÉSENTATION

Pour gérer les données de type date et heure SQL Server propose les types de données suivants :

- **datetime2** : qui offre plus de précision que le simple type *datetime*.
- **date** : qui permet de stocker uniquement une date (jour, mois et année) et donc aucune information de type *heures*, *minutes* ou *secondes*.
- **time** : qui permet de stocker des informations de type *heure*, *minutes*, *secondes* et fraction de secondes avec une précision de 100 nanosecondes.
- **datetimeoffset** : qui permet de stocker une information *date* et *heure* ainsi que le décalage par rapport à l'heure *GMT*.
- **datetime** : hérité des versions précédentes de SQL Server, il est préférable de ne pas utiliser ce type de données pour les nouveaux développements
- **smalldatetime** : peu lourd (4 octets); ce type permet de stocker une information de type *date* et *heure* (entre le 1<sup>er</sup> Janvier 1900 et le 6 juin 2079) avec une précision d'une minute.

Les types **datetime2**, **date**, **time** et **datetimeoffset** sont introduits avec SQL Server 2008. Ils sont à privilégier pour tous nouveaux développement car ils permettent une gestion plus fine des données de type **date** et **heure**.

Par exemple avec le type **date** seule les informations **année**, **mois** et **jour** sont conservées. Ainsi non seulement l'information est plus compact, mais en plus de nombreux effets de bords peuvent être évités lorsque les informations de type **heure**, **minutes** et **secondes** sont incorrect.

Le travail avec les données de type **date** et **time** est facile et la comparaison devient plus facile.

# LA SAISIE DE DATE ET HEURE

Il est possible de saisir la **date** sous forme de chaîne de caractères puis de convertir cette chaîne de caractères en données de type **datetime**.

La conversion peut être faite de façon implicite (**SET DATEFORMAT**) ou bien de façon explicite avec la fonction **CONVERT**.

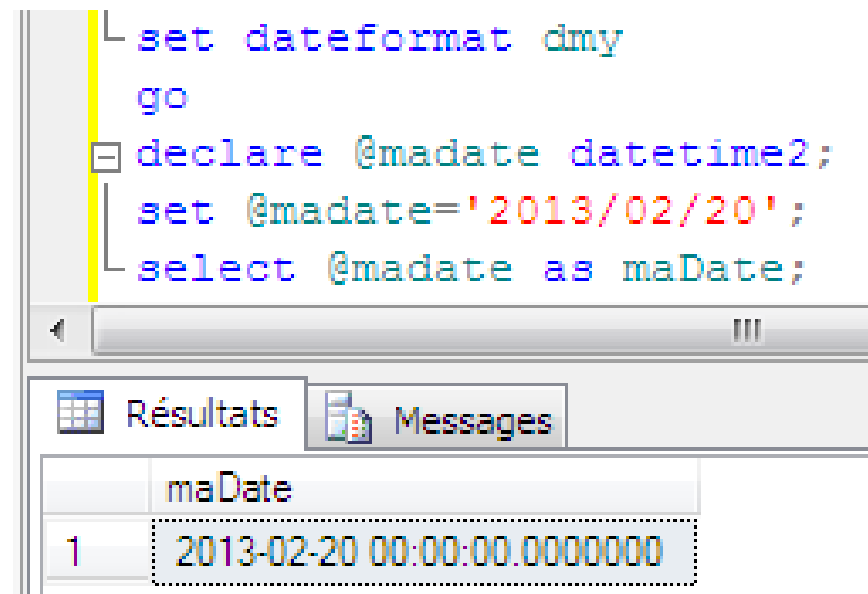
## L'instruction SET DATEFORMAT

Avec cette instruction, il est possible de définir son propre format de date pour la session en cours.

L'instruction **SET DATEFORMAT** ne concerne que les saisies de dates.

**SET DATEFORMAT** formatDate;

Le format de date permet de savoir dans quel ordre sont fournies les valeurs jour (d), mois (m) et année (y).



```
set dateformat dmy
go
declare @madata datetime2;
set @madata='2013/02/20';
select @madata as maDate;
```

	maDate
1	2013-02-20 00:00:00.0000000

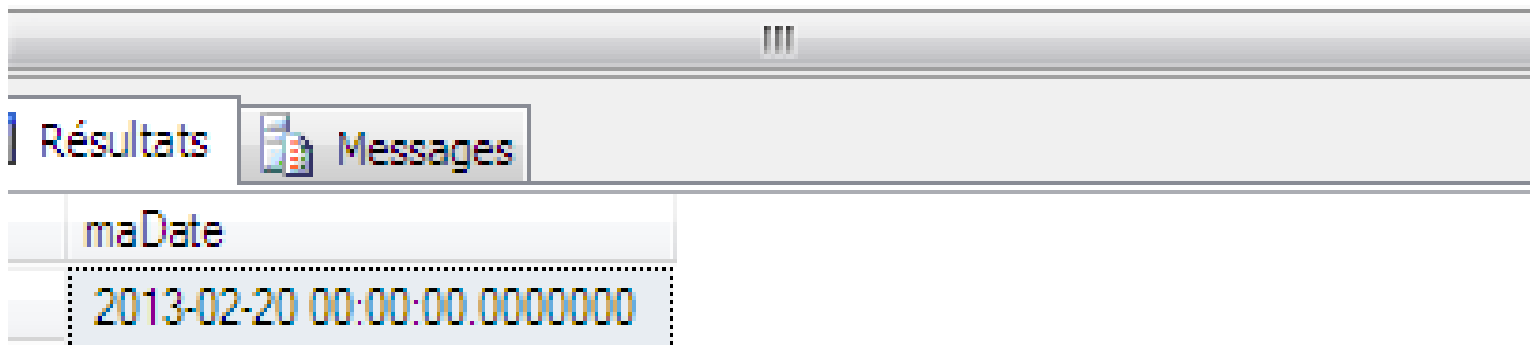
## La fonction **CONVERT**

La fonction **convert** accepte 3 paramètres :

- le type de destination,
- l'expression à convertir
- et le format de date.

Pour saisir une date au format *jj/mm/aaaa* il faut utiliser le style **103** ou bien le style **3** si on se contente d'une année saisie sur 2 chiffres simplement.

```
declare @maDate datetime2;  
set @maDate=convert(datetime2, '20/02/2013', 103);  
select @maDate as maDate;
```



Résultats		Messages
maDate	2013-02-20 00:00:00.0000000	

# AFFICHER UNE DATE ET HEURE

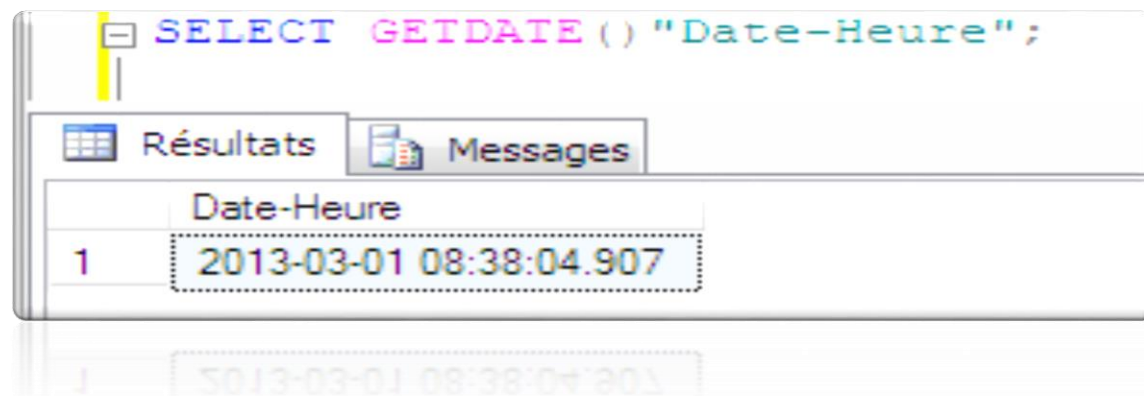
## Afficher la date et heure au format par défaut

Pour afficher une date et heure, il est nécessaire de convertir les informations au format texte.

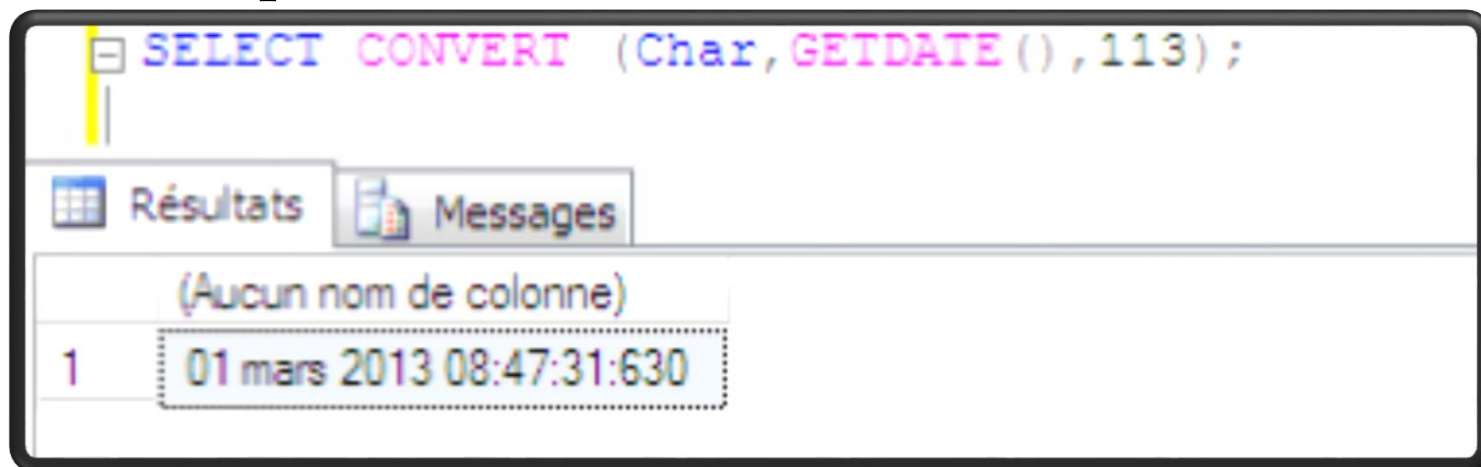
Cette opération est effectuée de façon implicite par le paramètre de langue ou bien **dateformat** (*s'il a été précisé*). Il est possible de réaliser cette opération de conversion de façon explicite avec la fonction **CONVERT**. Il est également possible de faire appel à la fonction **DATENAME** pour afficher la date à un format spécifique.



Dans l'exemple suivant, la date et heure courant est affichée au format par défaut.



La fonction **CONVERT** est utilisée pour convertir la date et heure courante de façon explicite en chaîne de caractère en respectant un format prédéfini.



# LES DIFFÉRENTS TYPE DE DATE ET HEURE

Type	Octets	Précision	Mini	Maxi	test
Datetime2 (précision)	6 à 8 octets suivant la précision	100 nanosecondes	01/01/0001	31/12/9999	select CONVERT (datetime2,getdate() );
Datetimeoffset (precision)	10 octets	100 nanosecondes avec un décalage horaire compris entre -14h et +14h	01/01/0001	31/12/9999	select CONVERT (datetimeoffset,getd ate());
date	3 octets	1 jour	01/01/0001	31/12/9999	select CONVERT (date,getdate());
time	5 octets	100 nanosecondes	00:00:00.000 0000	23:59:59.9999 999	select CONVERT (time,getdate());
datetime	8 octets	Arrondi à 0, 3 et 7 millième de seconde	01/01/1753	31/12/9999	select CONVERT (datetime,getdate());
smalldatetime	4 octets	1 minute	01/01/1900	06/06/2079	select CONVERT (smalldatetime,getda te());

## MANIPULER LES INFORMATIONS DE TYPE DATE ET HEURE

Pour manipuler les informations contenues dans des colonnes de type **datetime** et **smalldatetime** SQL Server propose différentes fonctions Transact SQL.

## La fonction DATEADD

**DATEADD** : Ajouter à une date un nombre de minutes, d'heures, de jours, ....

Exemple :

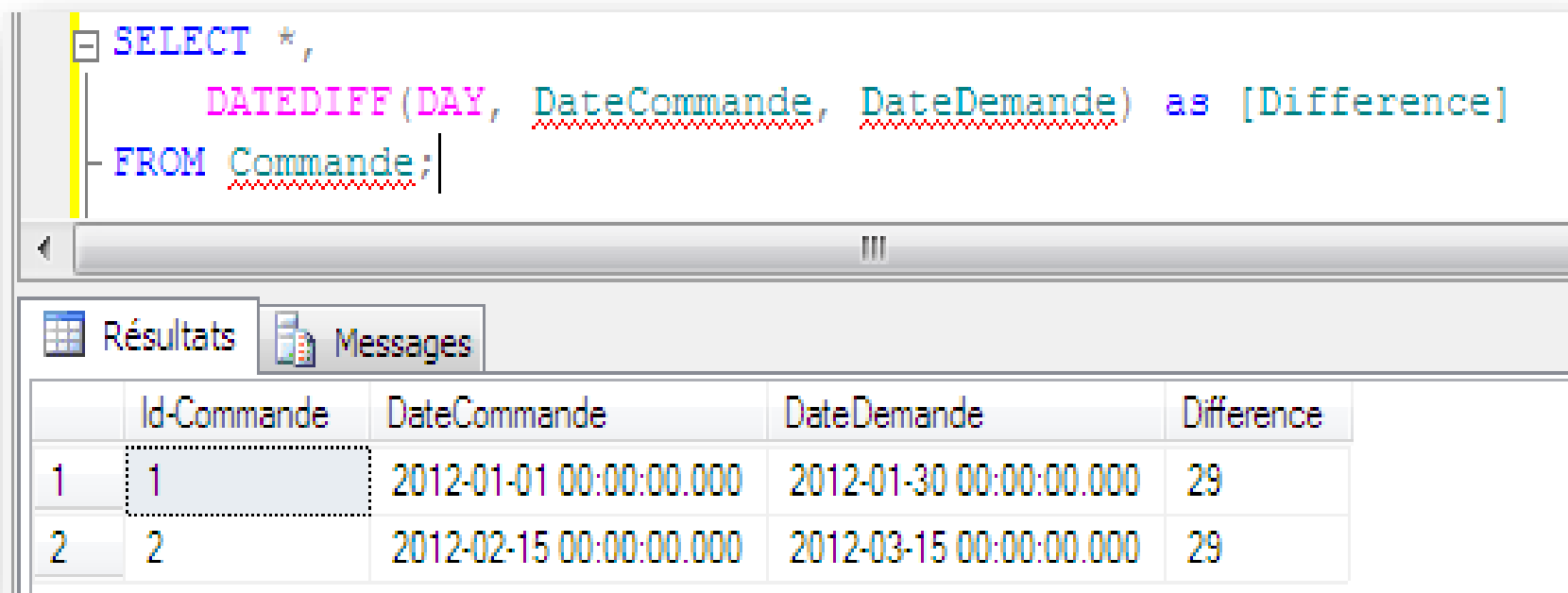
```
SELECT DateCommande,  
DATEADD(DAY, 30, DateCommande) as Plus30  
FROM Commande;
```

Résultats Messages

	DateCommande	Plus30
1	2012-01-01 00:00:00.000	2012-01-31 00:00:00.000
2	2012-02-15 00:00:00.000	2012-03-16 00:00:00.000

**DATEDIFF** : écart entre deux dates en respectant l'unité de mesure (année, mois, jour, heures, ...)

Exemple :



The screenshot shows a SQL query window with the following text:

```
SELECT *,  
    DATEDIFF(DAY, DateCommande, DateDemande) as [Difference]  
FROM Commande;
```

Below the query window, the 'Résultats' (Results) tab is active, displaying a table with the following data:

	Id-Commande	DateCommande	DateDemande	Difference
1	1	2012-01-01 00:00:00.000	2012-01-30 00:00:00.000	29
2	2	2012-02-15 00:00:00.000	2012-03-15 00:00:00.000	29

## La fonction DAY

DAY : Pour obtenir le numéro du jour dans le mois (*équivalent à datepart (d, ...)*)

Exemple :

```
SELECT *,  
       DAY(DateCommande) as jour  
FROM Commande;
```

Résultats Messages

	Id-Commande	DateCommande	DateDemande	jour
1	1	2012-01-01 00:00:00.000	2012-01-30 00:00:00.000	1
2	2	2012-02-15 00:00:00.000	2012-03-15 00:00:00.000	15

## La fonction MONTH

**MONTH** : Pour obtenir le numéro du mois dans l'année.

Exemple :

```
SELECT *,  
    MONTH(DateCommande) as Mois  
FROM Commande;
```

Résultats Messages

	Id-Commande	DateCommande	DateDemande	Mois
1	1	2012-01-01 00:00:00.000	2012-01-30 00:00:00.000	1
2	2	2012-02-15 00:00:00.000	2012-03-15 00:00:00.000	2

# La fonction YEAR

**YEAR** : Pour obtenir le numéro de l'année

Exemple :

```
SELECT *,  
        YEAR(DateCommande) as Annee  
FROM Commande;
```

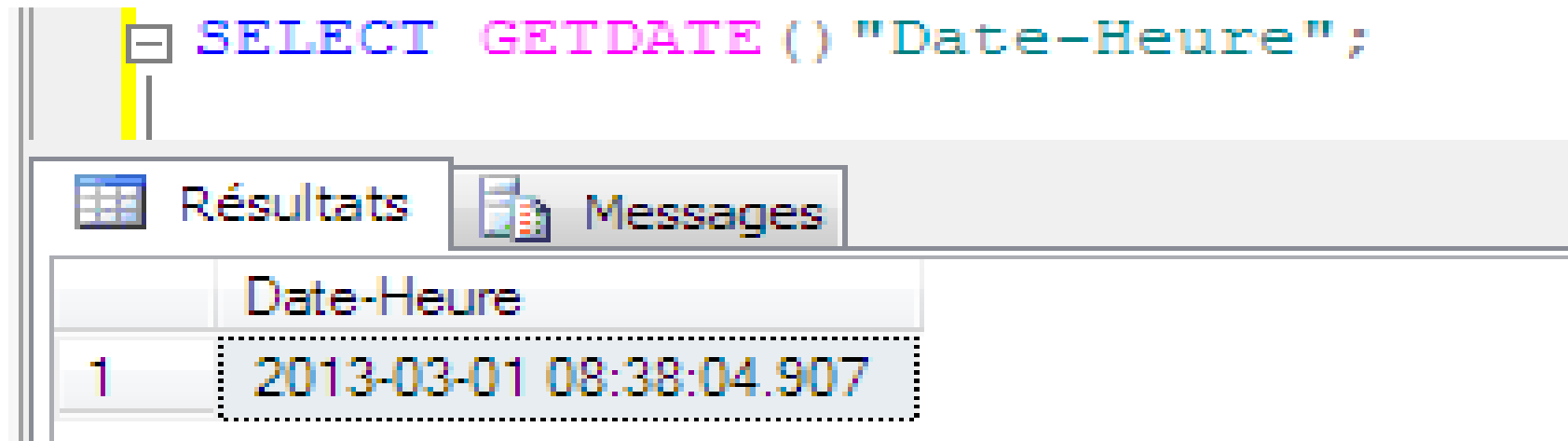
Résultats		Messages		
	Id-Commande	DateCommande	DateDemande	Annee
1	1	2012-01-01 00:00:00.000	2012-01-30 00:00:00.000	2012
2	2	2012-02-15 00:00:00.000	2012-03-15 00:00:00.000	2012



## La fonction GETDATE

**GETDATE** : Date et heure du système

Exemple :



The screenshot shows the SQL Server Enterprise Manager interface. At the top, a query is entered in the query window: `SELECT GETDATE () "Date-Heure";`. Below the query window, there are two tabs: "Résultats" (Results) and "Messages" (Messages). The "Résultats" tab is active, displaying a table with one column named "Date-Heure" and one row containing the value "2013-03-01 08:38:04.907".

	Date-Heure
1	2013-03-01 08:38:04.907

## La fonction DATENAME

**DATENAME** : Extraire une partie de la date au format caractère.

La fonction **DATENAME** accepte 2 arguments :

- la partie de la date à extraire ;
- l'information de type **datetime** avec laquelle travailler.

**DATENAME** (*élémentDeDate*, *donnéeDeTypeDatetime*)

Les éléments de dates sont codés de la façon suivante :

Elément	Abréviation
année	yy, yyyy
trimestre	qq, q
mois	mm, m
jour dans l'année	dy, y
jour	dd,d
semaine	wk, ww
jour de la semaine	dw
heures	hh
minutes	mi, n
secondes	ss, s
millisecondes	ms



## La fonction DATEPART

**DATEPART** : Extraire une partie de la date au format numérique. Elle fonctionne de façon similaire mais retourne un nombre entier au lieu d'une chaîne de caractères.

**Exemple :**

```
SELECT DATEPART (yyyy, getdate ());
```



Résultats



Messages

(Aucun nom de colonne)

1

2013

## CONCLUSION

Les différentes bases de données de date et heure relationnelles sont conçues pour recevoir différents types de données, il est donc sage de consulter au préalable les références spécifiques aux bases de données.