

Convertir un entier en timestamp ou date/heure GMT / locale

Date de dernière mise à jour : 13/05/2014 à 12:39

Source : <http://www.vulgarisation-informatique.com/conversion-date-timestamp-sql-server.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)

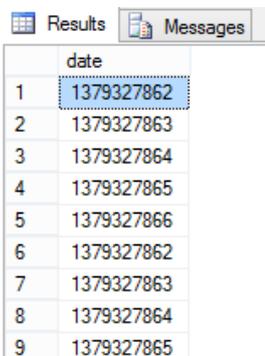
Là où de nombreux SGBD (Systèmes de gestion de bases de données) proposent des fonctions simples de conversion de dates, SQL Server est un peu plus compliqué à appréhender. Pour vous éviter de longues recherches, voici quelques compilations de mes trouvailles pour **convertir un timestamp en datetime** ou, plus généralement, **convertir un entier en date/heure locale ou GMT**.

Pour l'ensemble de nos exemples, nous utiliserons une table **alertes** avec une colonne **temps** de type **bigint**. J'ai utilisé SQL Server 2012 dans ce tutoriel, mais les exemples devraient fonctionner avec SQL Server 2008.

Convertir une chaîne de caractères en entier

Normalement, vous devriez stocker vos timestamps dans des types appropriés, pas en **Varchar**, mais, peu importe, si vous êtes dans ce cas, vous devez d'abord convertir un varchar en nombre. On va utiliser la fonction **CONVERT()** de SQL Server :

```
SELECT
convert
(
bigint
,
temps)
AS
date
FROM
alertes
```



	date
1	1379327862
2	1379327863
3	1379327864
4	1379327865
5	1379327866
6	1379327862
7	1379327863
8	1379327864
9	1379327865

A ce stade, vous obtenez des entiers de type **bigint**. Si votre champ est déjà de type **bigint**, vous pouvez sauter cette étape.

Convertir un entier de type bigint en date/heure GMT

L'heure GMT ne tient pas compte du décalage horaire. Pour effectuer ces conversions, nous aurons besoin de la fonction **DATEADD()**.

```
SELECT
dateadd
(
s
,
temps,
'1-1-1970 00:00:00'
)
FROM
```

alertes;

	(No column name)
1	2013-09-16 10:37:42.000
2	2013-09-16 10:37:43.000
3	2013-09-16 10:37:44.000
4	2013-09-16 10:37:45.000
5	2013-09-16 10:37:46.000
6	2013-09-16 10:37:42.000
7	2013-09-16 10:37:43.000
8	2013-09-16 10:37:44.000
9	2013-09-16 10:37:45.000
10	2013-09-16 10:37:46.000
11	2013-09-16 10:37:42.000
12	2013-09-16 10:37:43.000

En cas de soucis avec votre version de SQL Server, il vous faudra peut-être effectuer une conversion de type pour le paramètre de la fonction **DATEADD()** qui, normalement, est une expression de type **datetime**. Essayez alors la requête suivante :

```
SELECT
dateadd
(
s
,
temps,
CONVERT
(
datetime,
'1-1-1970 00:00:00'
)
)
FROM
alertes;
```

A ce stade, vous obtenez des dates au format **GMT**.

Convertir une date/heure GMT en date/heure locale

Maintenant que vous avez des dates au format GMT, vous souhaitez sûrement convertir votre timestamp original en date/heure locale. SQL Server ne permet pas directement de le faire. Toutefois, nous allons ruser. La première étape consiste à récupérer la date/heure GMT et la date/heure locale, de faire la différence entre les deux pour obtenir le **décalage horaire** (ou **OFFSET**). On ajoutera ensuite ce décalage à notre date GMT, pour obtenir une date/heure locale.

La date/heure GMT est obtenue avec la fonction **GETDATE()**, tandis que la date/heure locale est obtenue avec la fonction **GETUTCDATE()**. La différence entre deux dates s'effectue avec la fonction **DATEDIFF**, dont le premier argument est l'intervalle utilisée (ici la seconde, symbole **s**).

```
SELECT
DATEDIFF
(
second
,
GETUTCDATE(
)
)
```

```
,
```

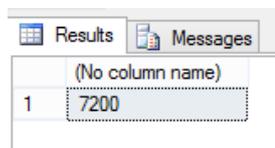
```
GETDATE
```

```
(
```

```
)
```

```
)
```

```
;
```



	(No column name)
1	7200

Nous obtenons ici un résultat de 7200 secondes de décalage (GMT +2h en France à l'heure d'été).

Il faut maintenant ajouter notre différence à la date GMT calculée précédemment. Pour ce faire, nous allons utiliser la fonction **DATEADD()** de SQL Server.

Le premier argument de la fonction **DATEADD()** correspond au type passé en second paramètre. En gros, nous avons calculé une différence en secondes, il faudra donc indiquer **second**.

```
SELECT
```

```
DATEADD
```

```
(
```

```
second
```

```
,
```

```
DATEDIFF
```

```
(
```

```
second
```

```
,
```

```
GETUTCDATE(
```

```
)
```

```
,
```

```
GETDATE
```

```
(
```

```
)
```

```
)
```

```
,
```

```
dateadd
```

```
(
```

```
s
```

```
,
```

```
temps,
```

```
'1-1-1970 00:00:00'
```

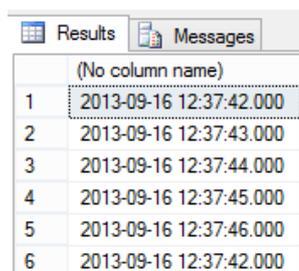
```
)
```

```
)
```

```
FROM
```

```
alertes;
```

A ce stade, vous vous retrouvez avec une date/heure locale.



	(No column name)
1	2013-09-16 12:37:42.000
2	2013-09-16 12:37:43.000
3	2013-09-16 12:37:44.000
4	2013-09-16 12:37:45.000
5	2013-09-16 12:37:46.000
6	2013-09-16 12:37:42.000

Source : <http://www.vulgarisation-informatique.com/conversion-date-timestamp-sql-server.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)