

Les fonctions spéciales

Il y a plusieurs fonctions spéciales en SQL qui peuvent être ajoutée plus ou moins n'importe où dans la requête.

Conversion des dates

Pour convertir un `DATE` en chaîne de caractère, on peut utiliser `TO_CHAR()`

```
SELECT localite, TO_CHAR(date_mise_en_vente, 'month-yyyy') FROM Bien;
```

Pour ce qui est de la syntaxe voici un extrait :

Syntaxe SQL	Correspond à	Exemple
<code>dd</code>	Jour du mois	<code>05</code>
<code>mm</code>	Mois en numéro	<code>02</code>
<code>yyyy</code>	Année en 4 numéros	<code>2022</code>
<code>yy</code>	Année en 2 numéros	<code>22</code>
<code>month</code>	Mois (épilé)	<code>décembre</code>
<code>mon</code>	Nom du mois en abrégé	<code>JAN</code>
<code>ddd</code>	Jour de l'année	<code>365</code>
<code>day</code>	Le jour en lettres	<code>lundi</code>
<code>mi</code>	Minutes	<code>56</code>
<code>hh</code>	Heures en format 12h	<code>1</code>
<code>ss</code>	Secondes	<code>59</code>
<code>hh24</code>	Heures en format 24h	<code>13</code>
<code>am</code>	Afficher AM si matin ou PM si après midi	<code>PM</code>

Pour la liste complète du format de `TO_CHAR` [cliquez ici](#)

Le même principe peut être utilisé pour convertir des chaînes de caractères en date aussi avec `TO_DATE()` qui a la même syntaxe.

- Si on veut juste extraire sans formater certaines valeurs de `DATE` ou `TIME` on peut utiliser la fonction `EXTRACT()`

```
-- YEAR peut être remplacé par MONTH, DAY. Et dans les cas où c'est `TIME` également par HOUR, MINUTE ou SECOND
```

```
SELECT EXTRACT(YEAR FROM date_mise_en_vente) AS annee FROM Bien;
```

```
-- OUTPUT: 2012
```

- On peut aussi récupérer la date actuelle du système avec la fonction `SYSDATE()`

```
-- On ajoute dans une table 'Leaderboard' la date avec SYSDATE
```

```
INSERT INTO Leaderboard (date, nom, prenom, score) VALUES (SYSDATE, 'Roger', 'Pierre', 144);
```

- On peut arrondir une date à une certaine unité avec `ROUND()`

```
-- On va arrondir la date au mois ce qui va retourner des choses tel que '01-12-2022' à la place de '13-12-2022'
```

```
SELECT ROUND(date_mise_en_vente, 'month') FROM Bien;
```

```
-- \_ Les valeurs ici sont les mêmes que celle vue dans le tableau au dessus
```

- On peut ajouter un certain nombre de mois à une date avec `ADD_MONTHS()`

```
-- Affiche la date d'aujourd'hui, la date d'aujourd'hui + 5 mois et le nombre de jours entre les 2
```

```
SELECT SYSDATE AS aujourd'hui,
```

```
ADD_MONTHS(SYSDATE, 5) AS deadline,
```

```
ADD_MONTHS(SYSDATE, 5) - SYSDATE AS tempsrestant
```

```
FROM Bien;
```

```
-- OUTPUT: AUJOURDH DEADLINE JOURSRESTANTS
```

```
-- -----
```

```
-- 30/12/22 30/05/23 151
```

- Affiche le dernier jour du mois avec `LAST_DAY()`

```
SELECT SYSDATE AS aujourd'hui, LAST_DAY(SYSDATE) AS findumois
```

```
FROM Bien;
```

```
-- OUTPUT: AUJOURDH FINDUMOI
```

```
-- -----
```

```
-- 30/12/22 31/12/22
```

- Affiche le nombre de mois de différence entre deux dates avec `MONTHS_BETWEEN()`

```
-- Ici je vais reprendre un exemple précédent mais changer le 'tempsrestant'
SELECT SYSDATE AS aujourd'hui,
       ADD_MONTHS(SYSDATE, 5) AS deadline,
       MONTHS_BETWEEN(ADD_MONTHS(SYSDATE, 5), SYSDATE) AS tempsrestant
FROM Bien;
-- OUTPUT: AUJOURDH DEADLINE TEMPSRESTANT
--
--      -----
--      30/12/22 30/05/23      5
```

- Pour avoir la date du premier jour de la semaine donné à partir d'une date on peut utiliser `NEXT_DAY()`

```
-- Donne la date du prochain lundi à partir d'aujourd'hui
SELECT SYSDATE AS aujourd'hui,
       NEXT_DAY(SYSDATE, 'LUNDI') AS prochainlundi
FROM Bien;
-- OUTPUT: AUJOURDH PROCHAIN
--
--      -----
--      30/12/22 02/01/23
```

Conversion des chaines de caractères

On peut utiliser `LOWER()` pour mettre en minuscule, ou `UPPER()` pour mettre en majuscule.

```
-- Voici un exemple de SQL avec un LIKE qui est case-insensitive en le transformant en lowercase
SELECT * FROM Bien WHERE LOWER(description) = '%calme%';
```

Enfin on peut aussi utiliser `SUBSTR()` pour couper une chaine de caractère (c'est très similaire au *substring* en Java)

```
-- Va donner l'initiale du prenom + le nom des candidats et renommer la colonne 'candidat'
-- On donne ici la position de début (1) et la position de fin (1) dans la chaine.
-- Mais si on veut prendre tout à partir de la position 2 par exemple, on peut juste écrire SUBSTR(nom, 2)
SELECT SUBSTR(prenom, 1, 1) || ' ' || nom AS candidat FROM Candidat;
```

On peut utiliser la fonction `replace` pour remplacer une chaine par une autre

```
-- Remplace tous les "robert" par des "roger"
SELECT REPLACE(prenom, 'Robert', 'Roger') AS prenom FROM candidat;
```

Mais dans certains cas on veut éliminer toute une série de caractère de nos entrées (pour pouvoir les utiliser comme nom de fichier par exemple) en utilisant `TRANSLATE`

```
-- Remplace les caractères espace, apostrophe, point, slash, backslash et étoile par des underscore
SELECT TRANSLATE(nom, ' ""'.\*', ' _____') AS filename FROM fichiers;
```

Pour savoir la longueur d'une chaîne de caractère on peut utiliser la fonction `LENGTH`

```
-- Donne pour chaque candidat, son nom et la longueur de son nom
SELECT nom, LENGTH(nom) AS longueurdunom FROM candidat;
```

Pour savoir la position d'une chaîne de caractère dans une autre chaîne, on peut utiliser `INSTR`

```
-- L'exemple ici va donner 14.
-- Il va rechercher la 2e occurrence de OR dans la chaîne à partir du 3e caractère
SELECT INSTR('CORPORATE FLOOR','OR', 3, 2) FROM Bien;

--      |          |  |  \_Quelle occurrence
--      |          |  \_A partir de X caractère
--      \_Input      \_Chaîne à rechercher
```

Pour simplement retirer des caractères, sans utiliser `REPLACE` on peut utiliser `TRIM` et ses variantes

```
-- Pour remplacer le début d'une chaîne de caractères
SELECT LTRIM('foo bar', 'foo ') FROM Bien;
-- OUTPUT: 'bar'

-- Pour remplacer la fin d'une chaîne de caractères
SELECT RTRIM('foo bar', ' bar') FROM Bien;
-- OUTPUT: 'foo'

-- Pour remplacer le début et la fin d'une chaîne de caractères
-- ATTENTION: TRIM ne fonctionne que pour 1 caractère
SELECT TRIM(BOTH 'o' FROM 'oobaroo') FROM Bien;
-- OUTPUT: 'bar'

-- Note: 'BOTH' peut être remplacé par LEADING pour isoler le début ou par TRAILING pour isoler la fin
```

On peut convertir une chaîne de caractère en nombre avec `TO_NUMBER`

```
SELECT TO_NUMBER('1210.73', '9999.99') FROM Bien;
```

Gestion des nombres

- La fonction `ROUND()` fonctionne également pour les nombres

```
-- Ceci donne le nombre de jour qui sépare la date de mise en vente de chaque bien de la date d'aujourd'hui
-- Ce nombre est normalement un nombre à virgule mais on l'arrondi avec ROUND
SELECT ROUND(SYSDATE - date_mise_en_vente) AS nbr_jours
FROM BIEN;
-- OUTPUT: NBR_JOURS
--
--      -----
--      3858
```

En savoir plus

- [En savoir plus sur les fonctions dans Oracle SQL](#)
- [Voir tous les formats de TO_CHAR et TO_DATE](#)
- [Voir tous les formats de ROUND et TRUNC](#)

Revision #1

Created 26 April 2023 19:01:33 by SnowCode

Updated 26 April 2023 19:01:33 by SnowCode