



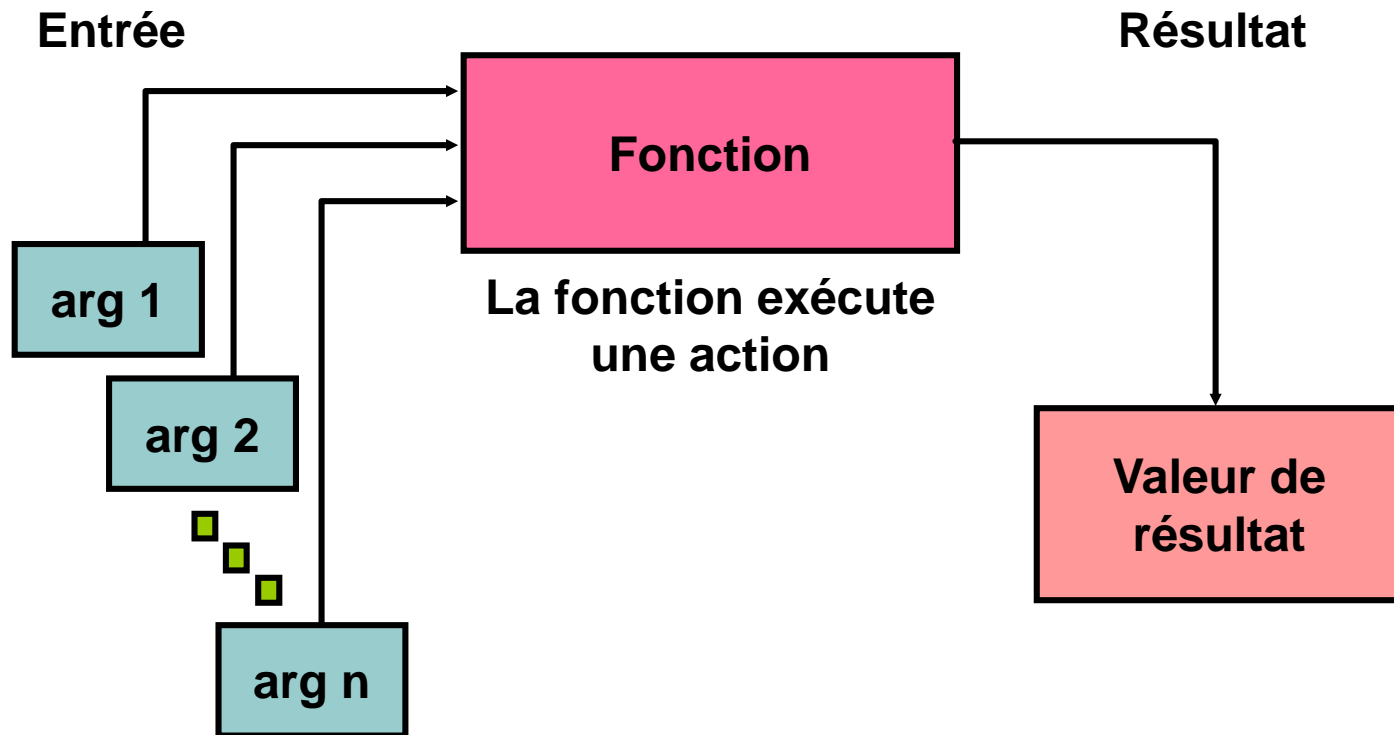
**Utiliser des fonctions monoligne afin de
personnaliser la sortie**

Objectifs

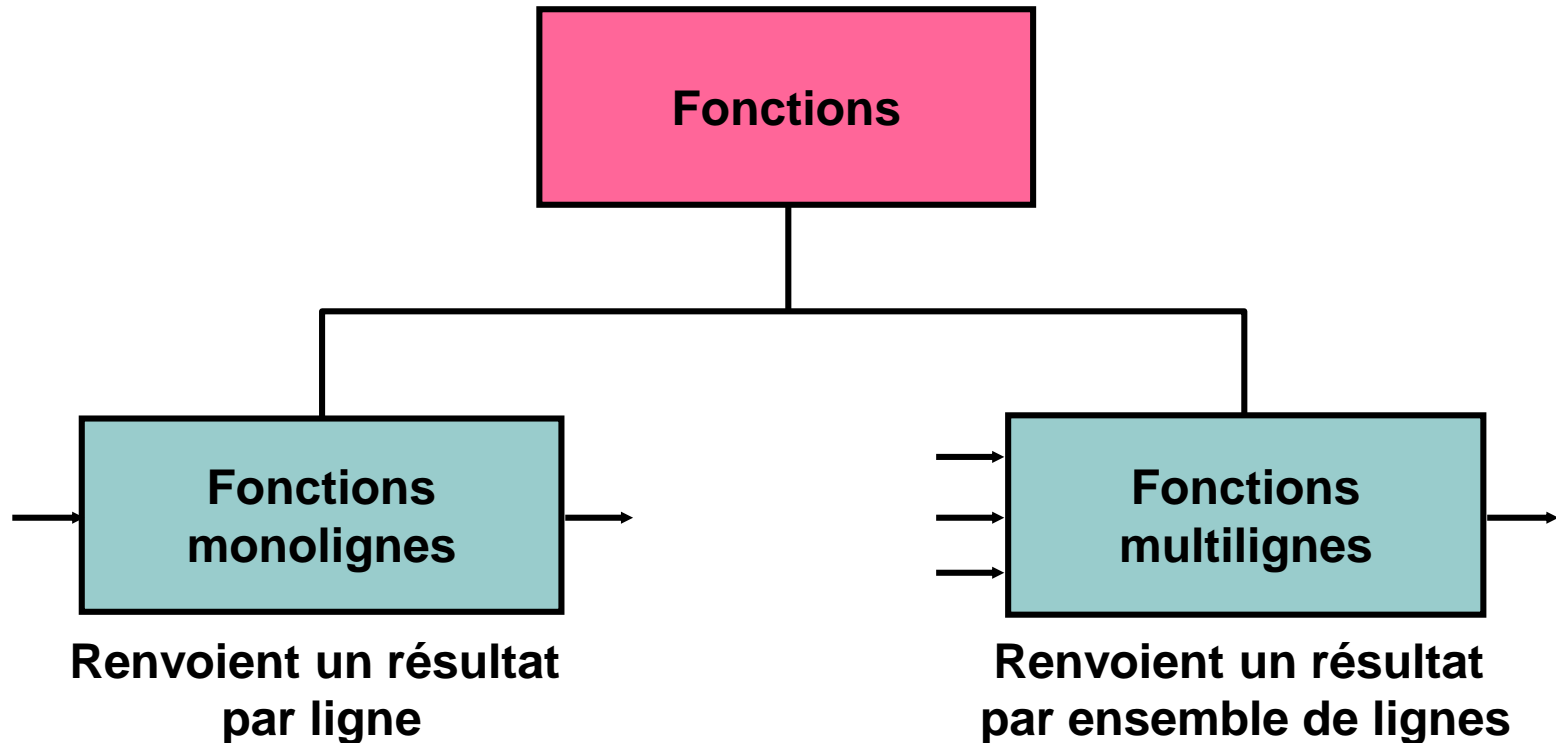
A la fin de ce TP, vous pourrez :

- **décrire les divers types de fonction du langage SQL**
- **utiliser des fonctions de type caractère, numérique et date dans les instructions `SELECT`**
- **utiliser des fonctions de conversion**
- **utiliser des fonctions générales**

Fonctions SQL



Deux types de fonction SQL



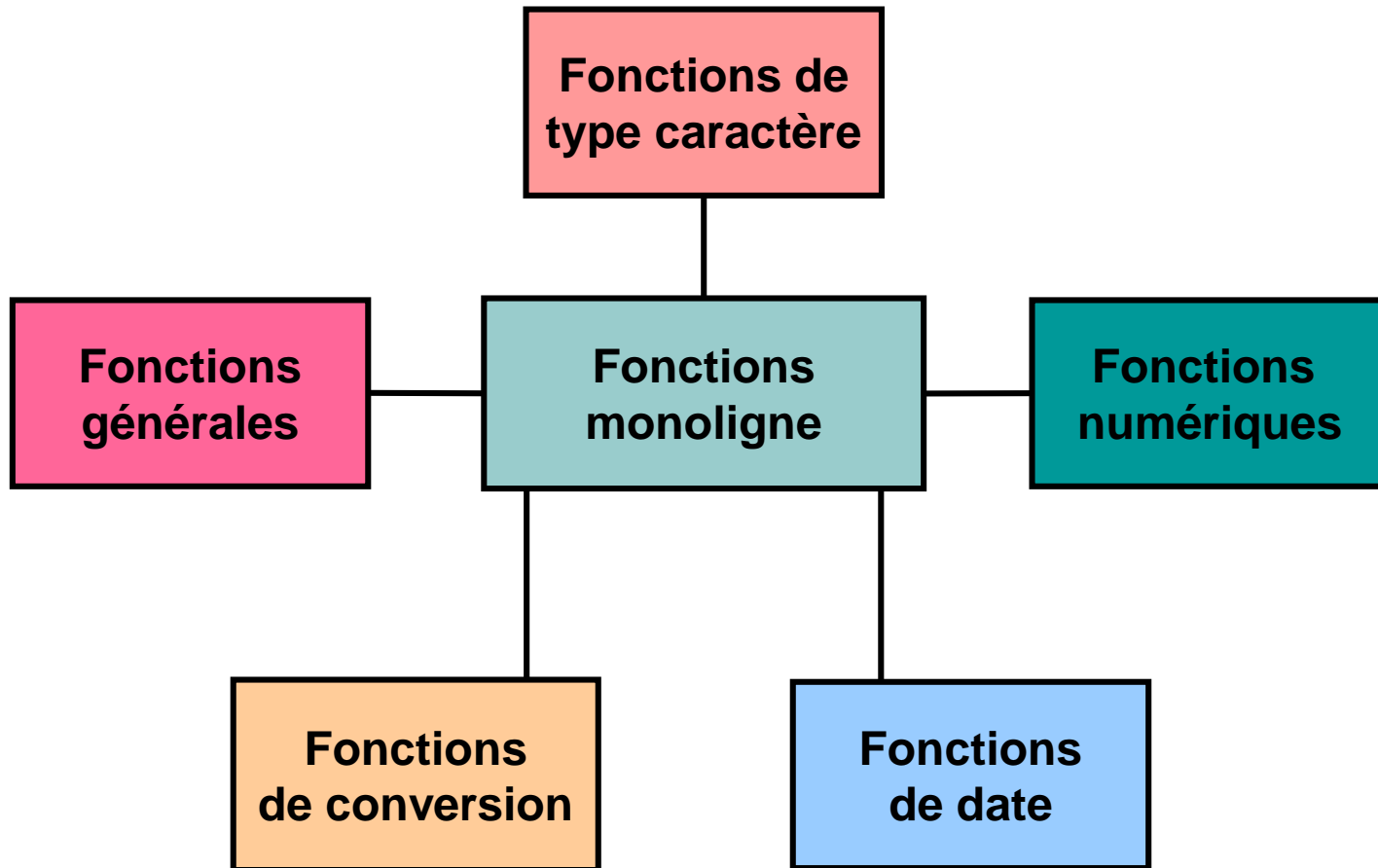
Fonctions monolignes

Les fonctions monolignes :

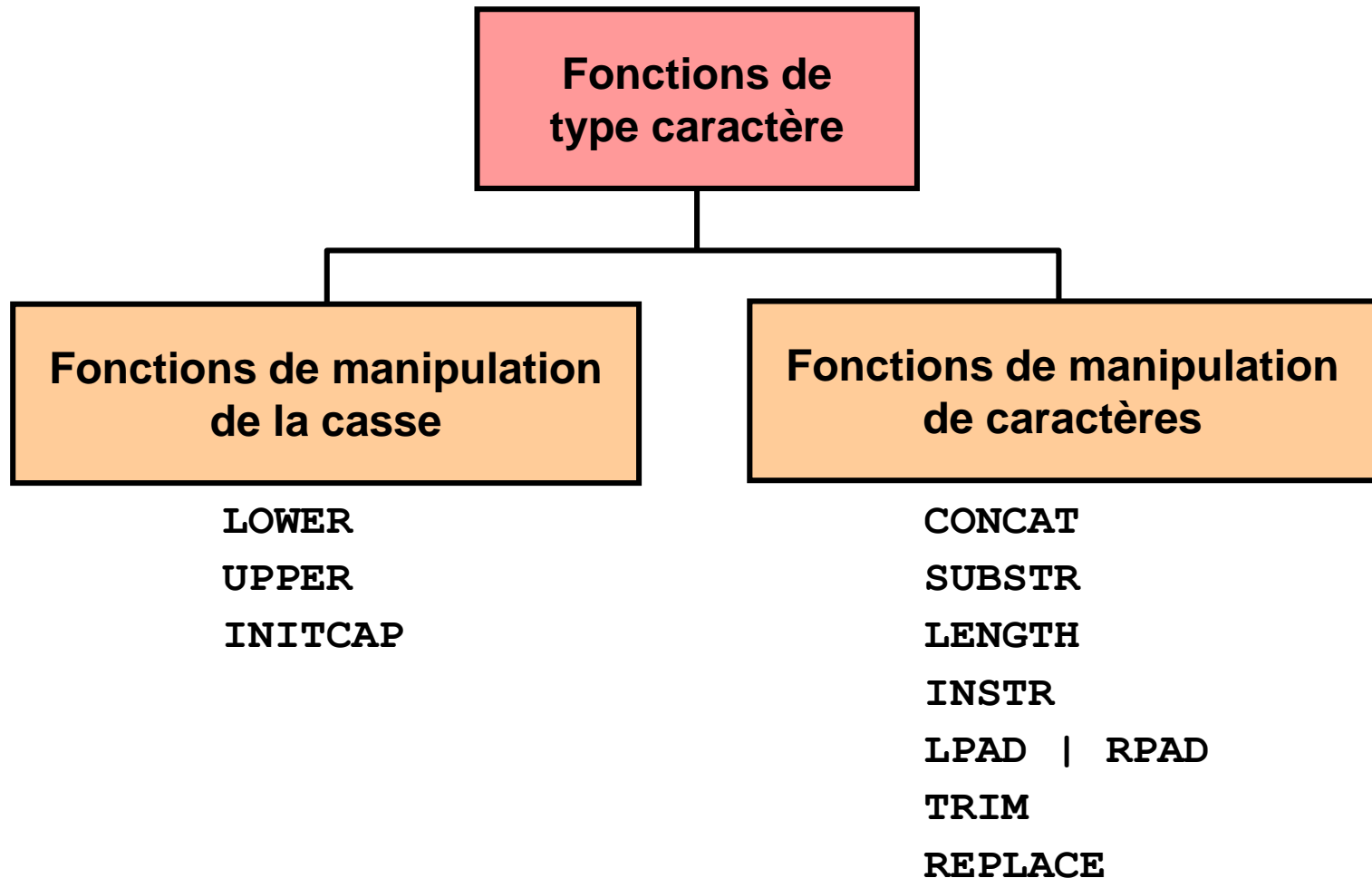
- Manipulent des données
- Acceptent des arguments pouvant être une constante, une variable, une colonne ou une expression
- Opèrent sur chaque ligne renvoyée par la requête
- Renvoient une seule valeur, un seul résultat par ligne
- Peuvent modifier le type de données
- peuvent être utilisées dans les clauses SELECT, WHERE et ORDER BY
- Peuvent être imbriquées

```
function_name [(arg1, arg2, ...)]
```

Fonctions monolignes



Fonctions de type caractère



Fonctions de manipulation de la casse

Ces fonctions convertissent la casse de chaînes de caractères :

Fonction	Résultat
<code>LOWER('SQL Course')</code>	sql course
<code>UPPER('SQL Course')</code>	SQL COURSE
<code>INITCAP('SQL Course')</code>	Sql Course

Utiliser les fonctions de manipulation de la casse

Afficher le numéro, le nom et le numéro de département de l'employé Higgins :

```
SELECT employee_id, last_name, department_id
FROM   employees
WHERE  last_name = 'higgins';
no rows selected
```

```
SELECT employee_id, last_name, department_id
FROM   employees
WHERE  LOWER(last_name) = 'higgins';
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
205	Higgins	110

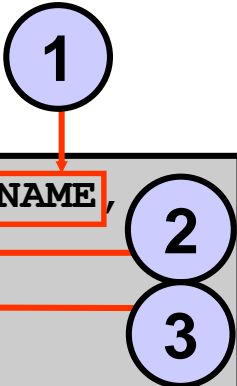
Fonctions de manipulation de caractères

Ces fonctions manipulent des caractères :


Fonction	Résultat
<code>CONCAT('Hello', 'World')</code>	HelloWorld
<code>SUBSTR('HelloWorld',1,5)</code>	Hello
<code>LENGTH('HelloWorld')</code>	10
<code>INSTR('HelloWorld', 'W')</code>	6
<code>LPAD(salary,10,'*')</code>	*****24000
<code>RPAD(salary,10,'*')</code>	24000*****
<code>REPLACE ('JACK and JUE', 'J', 'BL')</code>	BLACK and BLUE
<code>TRIM('H' FROM 'HelloWorld')</code>	elloWorld

Utiliser les fonctions de manipulation de caractères

```
SELECT employee_id, CONCAT(first_name, last_name) NAME,  
       job_id, LENGTH(last_name),  
       INSTR(last_name, 'a') "Contains 'a'?"  
FROM employees  
WHERE SUBSTR(job_id, 4) = 'REP';
```



EMPLOYEE_ID	NAME	JOB_ID	LENGTH(LAST_NAME)	Contains 'a'?
174	EllenAbel	SA_REP	4	0
176	JonathonTaylor	SA_REP	6	2
178	KimberelyGrant	SA_REP	5	3
202	PatFay	MK_REP	3	2

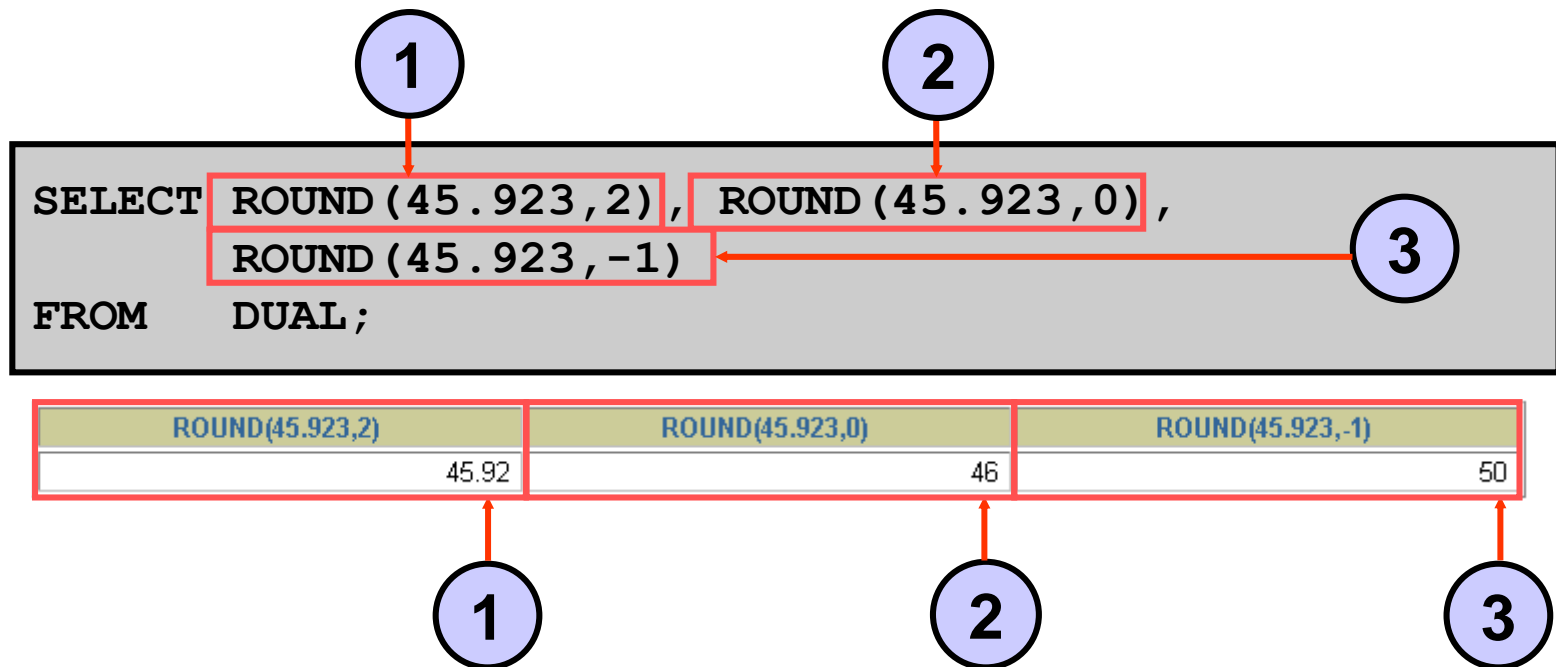


Fonctions numériques

- **ROUND** : arrondit la valeur à une décimale donnée
- **TRUNC** : tronque la valeur à une décimale donnée
- **MOD** : renvoie le reste d'une division

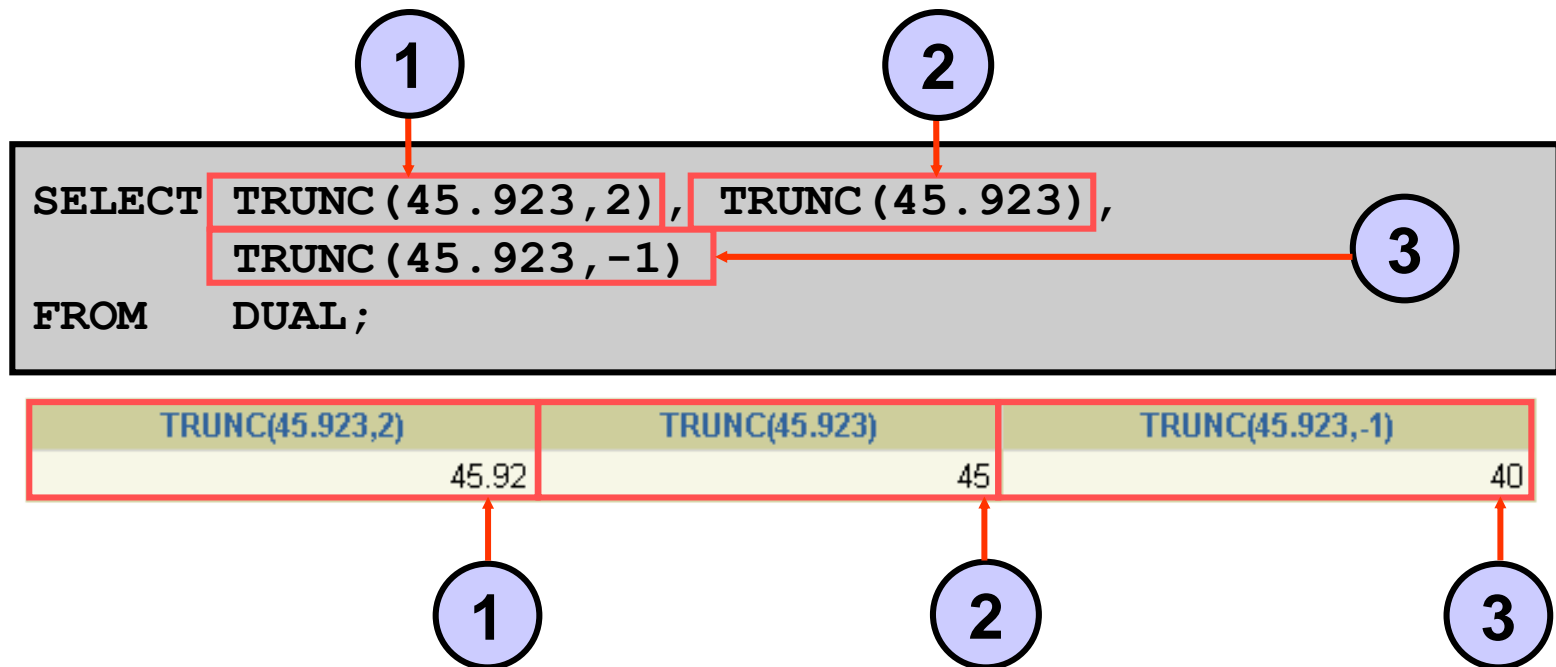
Fonction	Résultat
ROUND (45 . 926 , 2)	45 . 93
TRUNC (45 . 926 , 2)	45 . 92
MOD (1600 , 300)	100

Utiliser la fonction ROUND



DUAL est une table factice que vous pouvez utiliser pour afficher les résultats de fonctions et de calculs.

Utiliser la fonction TRUNC



Utiliser la fonction MOD

Pour tous les employés dont le poste est Sales Representative, calculer le reste du salaire après division par 5 000 :

```
SELECT last_name, salary, MOD(salary, 5000)
FROM employees
WHERE job_id = 'SA_REP';
```

LAST_NAME	SALARY	MOD(SALARY,5000)
Abel	11000	1000
Taylor	8600	3600
Grant	7000	2000

Manipuler les dates

- Dans la base de données Oracle, les dates sont stockées dans un format numérique interne : siècle, année, mois, jour, heures, minutes et secondes.
- Le format de date par défaut est DD-MON-RR.
 - Ce format permet le stockage des dates du 21ème siècle dans le 20ème siècle via l'indication les deux derniers chiffres de l'année uniquement.
 - De la même façon, il permet le stockage des dates du 20ème siècle dans le 21ème siècle.

```
SELECT last_name, hire_date
FROM employees
WHERE hire_date < '01-FEB-88';
```

LAST_NAME	HIRE_DATE
King	17-JUN-87
Whalen	17-SEP-87

Utiliser les dates

SYSDATE est une fonction qui renvoie :

- La date
- L'heure

Exemple

Affichez la date actuelle à l'aide de la table **DUAL** :

```
SELECT SYSDATE  
FROM DUAL;
```

Calculs arithmétiques sur des dates

- Ajoutez un nombre à une date ou soustrayez un nombre d'une date afin d'obtenir une date résultante.
- Soustrayez une date d'une autre afin de déterminer le nombre de jours entre ces dates.
- Ajoutez des heures à une date en divisant le nombre d'heures par 24.

Utiliser des opérateurs arithmétiques avec des dates

```
SELECT last_name, (SYSDATE-hire_date)/7 AS WEEKS  
FROM employees  
WHERE department_id = 90;
```

LAST_NAME	WEEKS
King	744.245395
Kochhar	626.102538
De Haan	453.245395

Fonctions de date

Fonction	Résultat
MONTHS_BETWEEN	Nombre de mois entre deux dates
ADD_MONTHS	Ajout de mois calendaires à une date
NEXT_DAY	Jour qui suit la date indiquée
LAST_DAY	Dernier jour du mois
ROUND	Date arrondie
TRUNC	Date tronquée

Utiliser des fonctions de date

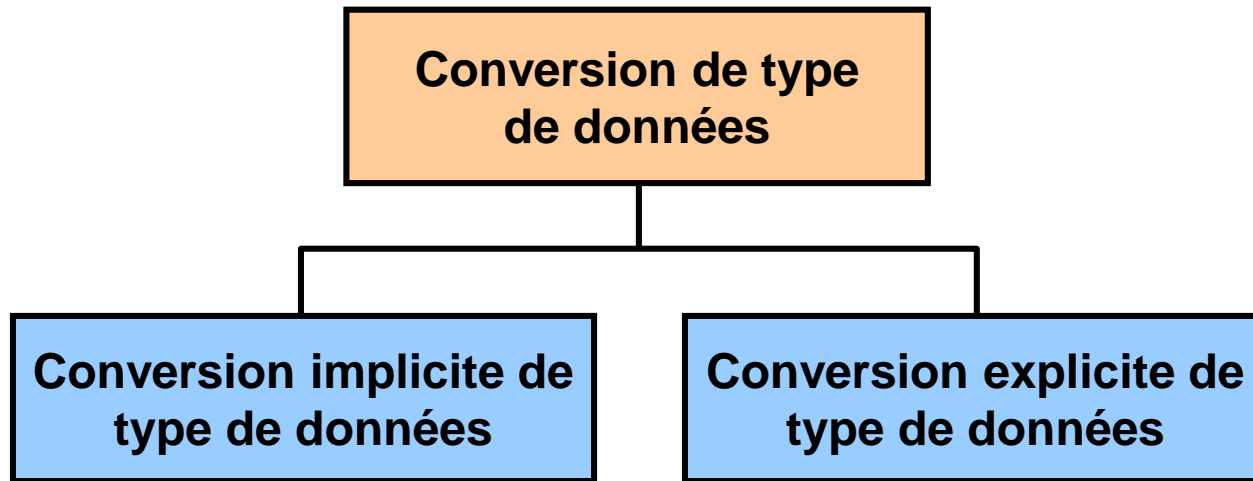
Fonction	Résultat
MONTHS_BETWEEN ('01-SEP-95', '11-JAN-94')	19.6774194
ADD_MONTHS ('11-JAN-94', 6)	'11-JUL-94'
NEXT_DAY ('01-SEP-95', 'FRIDAY')	'08-SEP-95'
LAST_DAY ('01-FEB-95')	'28-FEB-95'

Utiliser des fonctions de date

Supposons que SYSDATE = '25-JUL-03' :

Fonction	Résultat
ROUND (SYSDATE , 'MONTH')	01-AUG-03
ROUND (SYSDATE , 'YEAR')	01-JAN-04
TRUNC (SYSDATE , 'MONTH')	01-JUL-03
TRUNC (SYSDATE , 'YEAR')	01-JAN-03

Fonctions de conversion



Conversion implicite de type de données

Pour les affectations, le serveur Oracle peut convertir automatiquement les types de données suivants :

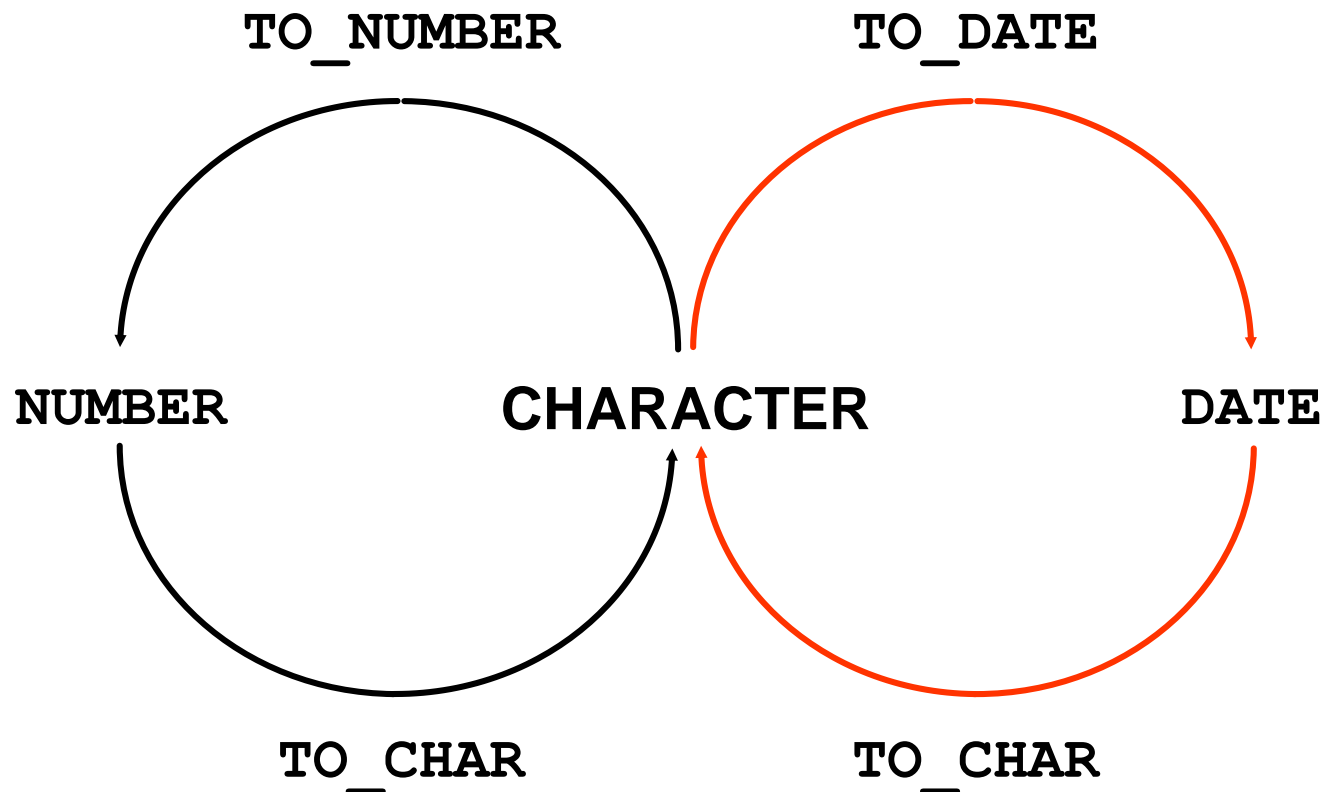
De	Vers
VARCHAR2 ou CHAR	NUMBER
VARCHAR2 ou CHAR	DATE
NUMBER	VARCHAR2
DATE	VARCHAR2

Conversion implicite de type de données

Pour l'évaluation des expressions, le serveur Oracle peut convertir automatiquement les types de données suivants :

De	Vers
VARCHAR2 ou CHAR	NUMBER
VARCHAR2 ou CHAR	DATE

Conversion explicite de type de données



Utiliser la fonction TO_CHAR avec des dates

```
TO_CHAR(date, 'format_model')  

```

Le modèle de format :

- Doit être inclus entre apostrophes
- Distingue les majuscules des minuscules
- Peut inclure n'importe quel élément de format de date valide
- Comporte un élément `fm` permettant de supprimer les espaces de remplissage ou les zéros de début
- Est séparé de la valeur de date par une virgule

Éléments du modèle de format de date

Élément	Résultat
YYYY	Année complète en chiffres
YEAR	Année en lettres (en anglais)
MM	Valeur à deux chiffres du mois
MONTH	Nom complet du mois
MON	Abréviation à trois lettres du mois
DY	Abréviation à trois lettres du jour de la semaine
DAY	Nom complet du jour de la semaine
DD	Valeur numérique du jour du mois

Éléments du modèle de format de date

- Les éléments d'heure formatent la partie heure de la date :

HH24 : MI : SS AM	15 : 45 : 32 PM
-------------------	-----------------

- Ajoutez des chaînes de caractères en les incluant entre guillemets :

DD "of" MONTH	12 of OCTOBER
---------------	---------------

- Utilisez des suffixes de nombre pour écrire les nombres en toutes lettres :

ddspth	fourteenth
--------	------------

Utiliser la fonction TO_CHAR avec des dates

```
SELECT last_name,  
       TO_CHAR(hire_date, 'fmDD Month YYYY')  
       AS HIREDATE  
FROM   employees;
```

LAST_NAME	HIREDATE
King	17 June 1987
Kochhar	21 September 1989
De Haan	13 January 1993
Hunold	3 January 1990
Ernst	21 May 1991
Lorentz	7 February 1999
Mourgos	16 November 1999

...

20 rows selected.

Utiliser la fonction TO_CHAR avec des nombres

```
TO_CHAR(number, 'format_model') ddspth
```

Voici quelques-uns des éléments de format que vous pouvez utiliser avec la fonction TO_CHAR pour afficher une valeur numérique sous forme de caractères :

Élément	Résultat
9	Représente un nombre
0	Force l'affichage d'un zéro
\$	Insère un signe dollar flottant
L	Utilise le symbole monétaire local flottant
.	Affiche un point en tant que séparateur décimal
,	Affiche une virgule en tant que séparateur de milliers

Utiliser la fonction TO_CHAR avec des nombres

```
SELECT TO_CHAR(salary, '$99,999.00') SALARY  
FROM   employees  
WHERE  last_name = 'Ernst';
```

SALARY
\$6,000.00

Utiliser les fonctions TO_NUMBER et TO_DATE

- Convertir une chaîne de caractères en format numérique à l'aide de la fonction TO_NUMBER :

```
TO_NUMBER(char[, 'format_model'])
```

- Convertir une chaîne de caractères en format de date à l'aide de la fonction TO_DATE :

```
TO_DATE(char[, 'format_model'])
```

- Ces fonctions comportent un modificateur **fx**. Ce modificateur indique la correspondance exacte de l'argument de type caractère et du modèle de format de date d'une fonction TO_DATE.

Format de date RR

Année en cours	Date indiquée	Format RR	Format YY
1995	27-OCT-95	1995	1995
1995	27-OCT-17	2017	1917
2001	27-OCT-17	2017	2017
2001	27-OCT-95	1995	2095

		Si l'année à deux chiffres indiquée est :	
		0–49	50–99
Si les deux chiffres de l'année actuelle sont :	0–49	La date renvoyée correspond au siècle actuel	La date renvoyée correspond au siècle avant le siècle actuel
	50–99	La date renvoyée correspond au siècle après le siècle actuel	La date renvoyée correspond au siècle actuel

Exemple de format de date RR

Pour afficher les employés embauchés avant 1990, utilisez le format de date RR, qui donne les mêmes résultats que la commande soit exécutée en 1999 ou maintenant :

```
SELECT last_name, TO_CHAR(hire_date, 'DD-Mon-YYYY')  
FROM employees  
WHERE hire_date < TO_DATE('01-Janv.-90', 'DD-Mon-RR');
```

LAST_NAME	TO_CHAR(HIR
King	17-Jun-1987
Kochhar	21-Sep-1989
Whalen	17-Sep-1987

Imbriquer des fonctions

- Les fonctions monoligne peuvent être imbriquées à un niveau quelconque.
- Les fonctions imbriquées sont évaluées du niveau le plus profond au niveau le moins profond.

F3 (F2 (F1 (col, arg1) , arg2) , arg3)

Etape 1 = résultat 1

Etape 2 = résultat 2

Etape 3 = résultat 3

Imbriquer des fonctions

```
SELECT last name,  
       UPPER(CONCAT(SUBSTR (LAST_NAME, 1, 8), '_US'))  
FROM   employees  
WHERE  department_id = 60;
```

LAST_NAME	UPPER(CONCAT(SUBSTR(LAST_NAME,1,8
Hunold	HUNOLD_US
Ernst	ERNST_US
Lorentz	LORENTZ_US

Fonctions générales

Les fonctions suivantes peuvent utiliser n'importe quel type de données, y compris les valeurs NULL :

- **NVL (expr1, expr2)**
- **NVL2 (expr1, expr2, expr3)**
- **NULLIF (expr1, expr2)**
- **COALESCE (expr1, expr2, ..., exprn)**

Fonction NVL

Convertit une valeur NULL en une valeur réelle :

- **Les types de données pouvant être utilisés sont les dates, les caractères et les valeurs numériques.**
- **Les types de données doivent correspondre :**
 - `NVL(commission_pct,0)`
 - `NVL(hire_date,'01-JAN-97')`
 - `NVL(job_id,'No Job Yet')`

Utiliser la fonction NVL

```
SELECT last name, salary, NVL(commission_pct, 0),  
       (salary*12) + (salary*12*NVL(commission_pct, 0)) AN_SAL  
FROM employees;
```

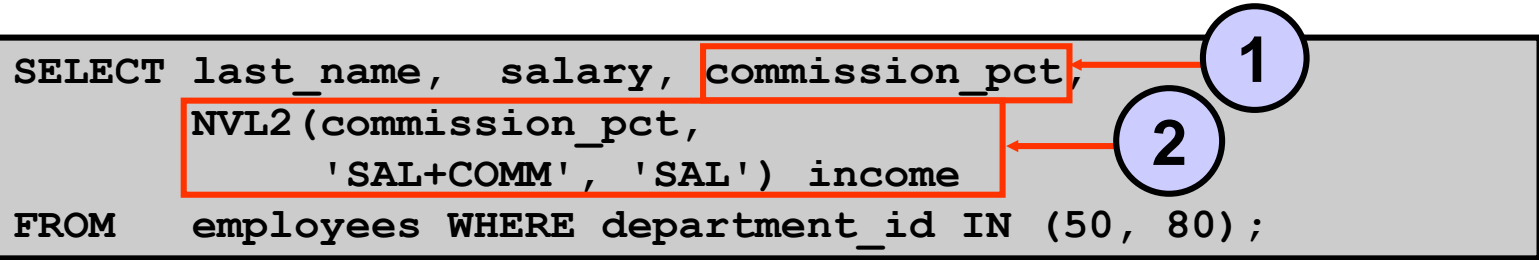
LAST_NAME	SALARY	NVL(COMMISSION_PCT,0)	AN_SAL
King	24000	0	288000
Kochhar	17000	0	204000
De Haan	17000	0	204000
Hunold	9000	0	108000
Ernst	6000	0	72000
Lorentz	4200	0	50400
Mourgos	5800	0	69600
Rajs	3500	0	42000

...

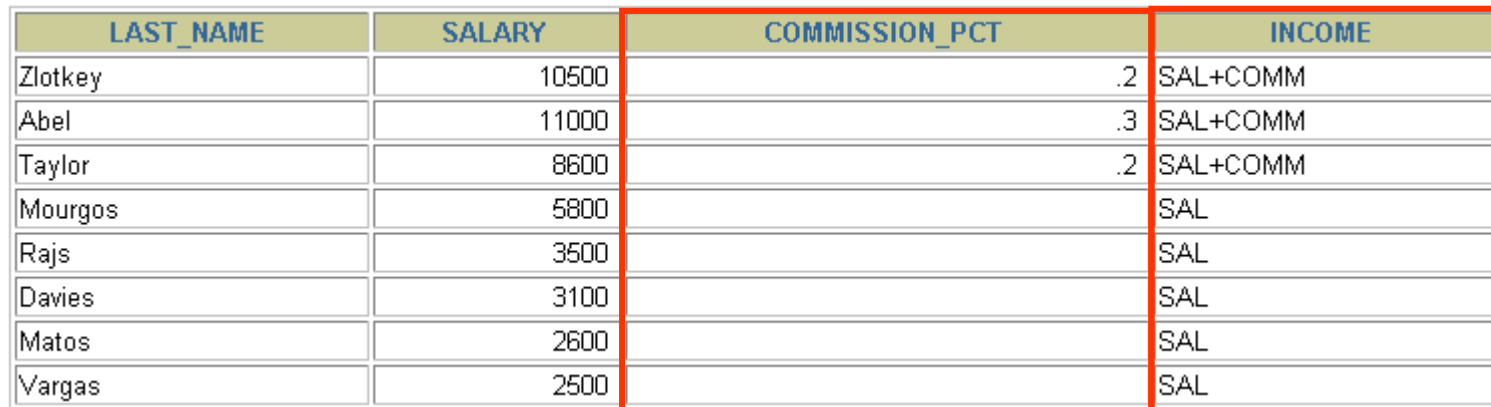
20 rows selected.

Utiliser la fonction NVL2

```
SELECT last_name, salary, commission_pct,  
       NVL2(commission_pct,  
            'SAL+COMM', 'SAL') income  
FROM   employees WHERE department_id IN (50, 80);
```



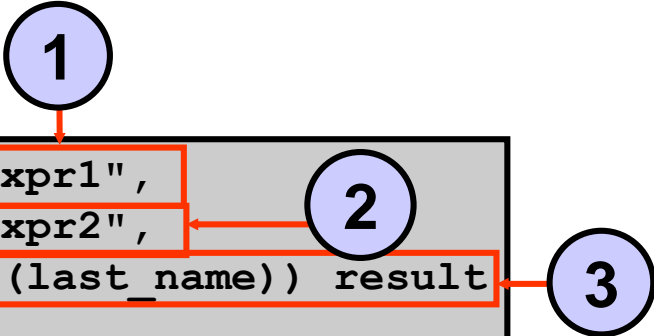
LAST_NAME	SALARY	COMMISSION_PCT	INCOME
Zlotkey	10500	.2	SAL+COMM
Abel	11000	.3	SAL+COMM
Taylor	8600	.2	SAL+COMM
Mourgos	5800		SAL
Rajs	3500		SAL
Davies	3100		SAL
Matos	2600		SAL
Vargas	2500		SAL




8 rows selected.

Utiliser la fonction NULLIF

```
SELECT first_name, LENGTH(first_name) "expr1",  
       last_name, LENGTH(last_name) "expr2",  
       NULLIF(LENGTH(first_name), LENGTH(last_name)) result  
FROM employees;
```



FIRST_NAME	expr1	LAST_NAME	expr2	RESULT
Steven	6	King	4	6
Neena	5	Kochhar	7	5
Lex	3	De Haan	7	3
Alexander	9	Hunold	6	9
Bruce	5	Ernst	5	
Diana	5	Lorentz	7	5
Kevin	5	Mourgos	7	5
Trenna	6	Rajs	4	6
Curtis	6	Davies	6	



...

20 rows selected.

Utiliser la fonction COALESCE

- L'avantage de la fonction COALESCE par rapport à la fonction NVL est que la fonction COALESCE peut prendre plusieurs valeurs alternatives.
- Si la première expression n'est pas NULL, la fonction COALESCE renvoie cette expression ; dans le cas contraire, elle exécute une opération COALESCE sur les expressions restantes.

Utiliser la fonction COALESCE

```
SELECT last_name,  
       COALESCE(manager_id,commission_pct, -1) comm  
FROM   employees  
ORDER BY commission_pct;
```

LAST_NAME	COMM
Grant	149
Zlotkey	100
Taylor	149
Abel	149
King	-1
Kochhar	100
De Haan	100

...

20 rows selected.

Synthèse

Ce TP vous a permis d'apprendre à :

- **effectuer des calculs sur des données à l'aide de fonctions**
- **modifier des données individuelles à l'aide de fonctions**
- **modifier des formats de date pour un affichage à l'aide de fonctions**
- **convertir des types de données de colonne à l'aide de fonctions**
- **utiliser les fonctions NVL**