

Cours base de données

CHAPITRE 5 : SQL

Dr Coulibaly Tiekoura

PLAN DU CHAPITRE

1. Extraire des données à l'aide de l'instruction SQL SELECT
2. Restreindre et trier les données
3. **Utiliser des fonctions monolignes afin de personnaliser la sortie**
4. afficher des données agrégées à l'aide des fonctions de groupe
5. Afficher des données de plusieurs tables
6. Utiliser des sous -interrogations
7. Utiliser des opérateurs ensemblistes

PLAN DU CHAPITRE (SUITE)

8. Manipuler des données
9. Utiliser des instructions LDD pour créer et gérer des tables

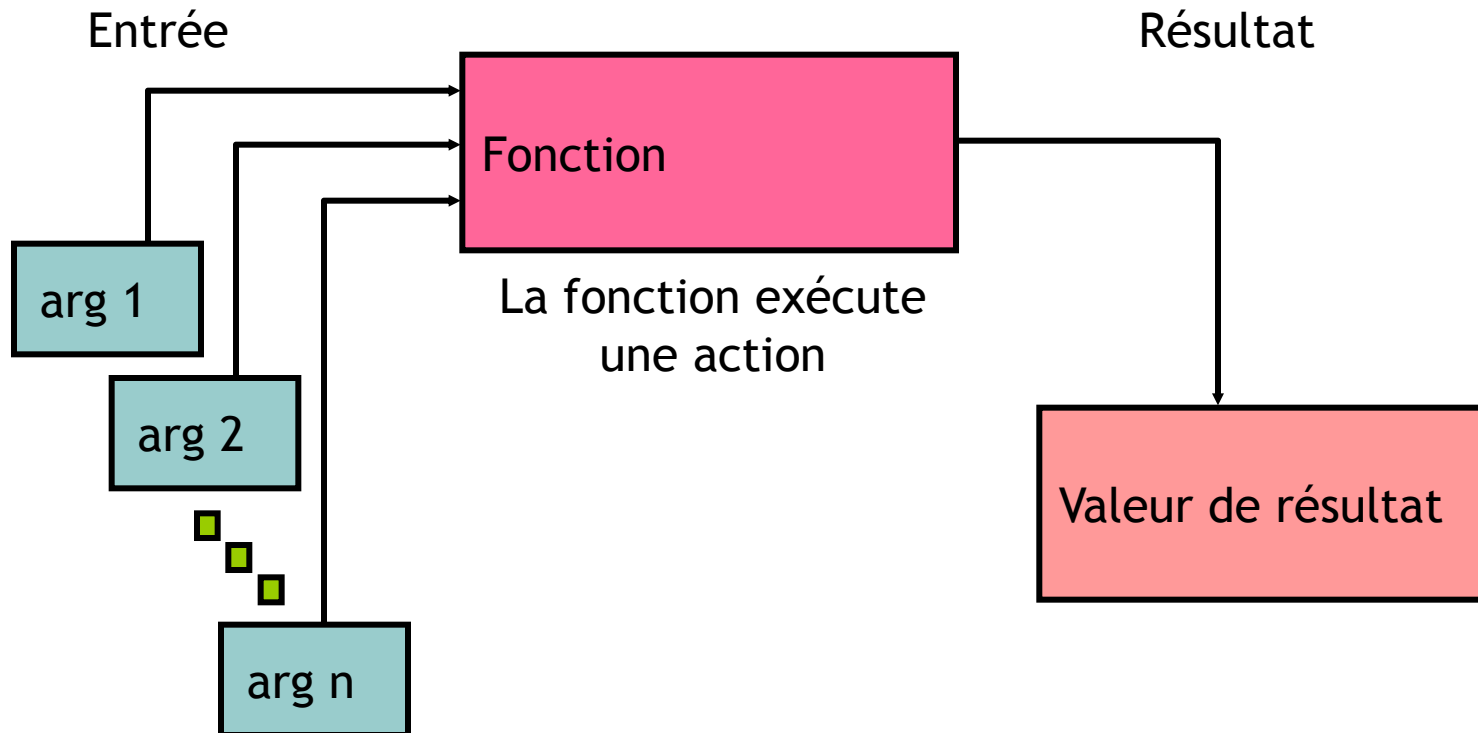
Cours base de données

CHAPITRE 5.3:
**LES FONCTIONS
MONOLIGNES**

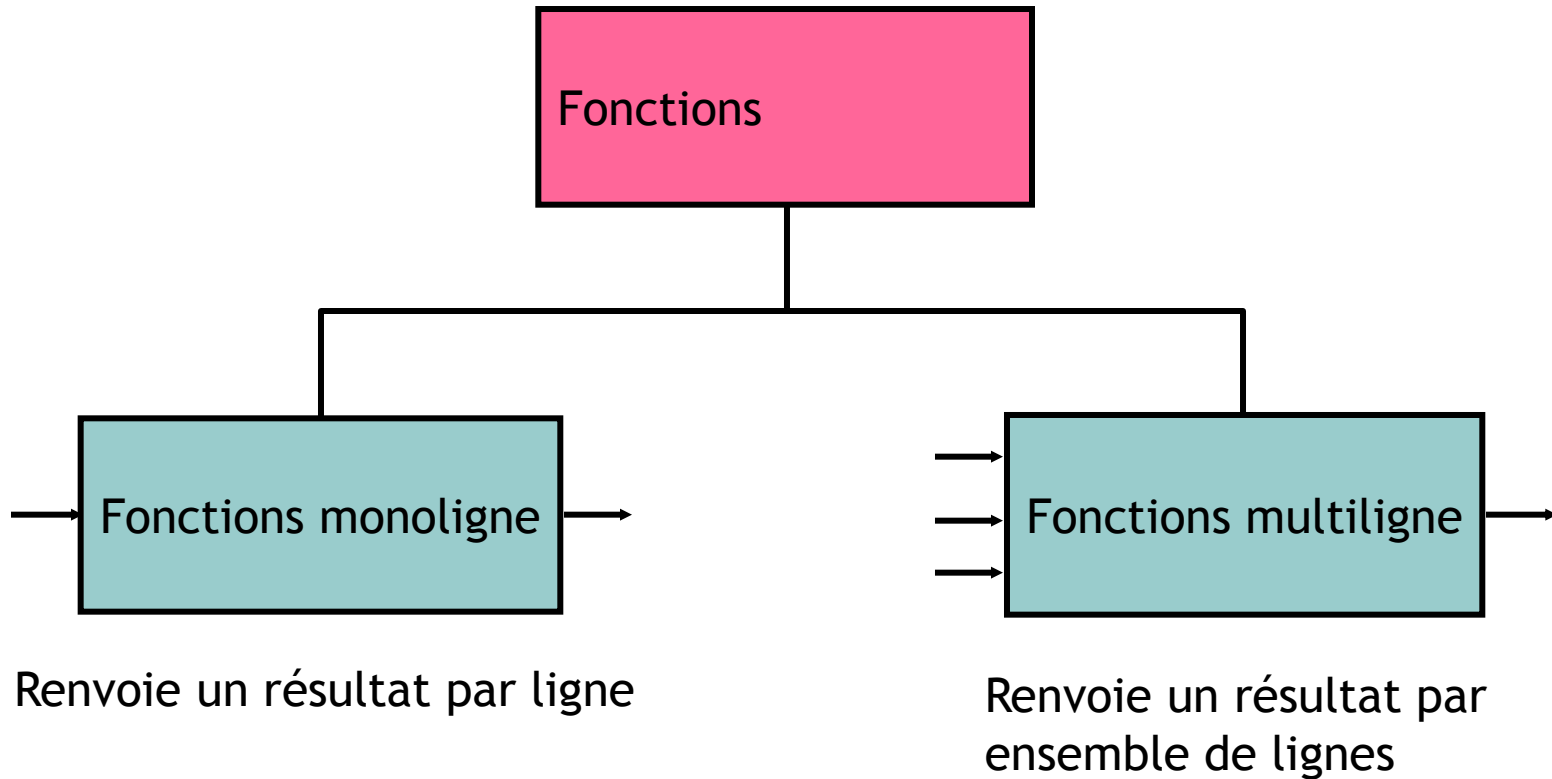
OBJECTIFS

- ◉ Décrire les divers types de fonction du langage SQL.
- ◉ Utiliser des fonctions de type caractère, des fonctions numériques et des fonctions de date dans des instructions SELECT.
- ◉ Décrire l'utilisation des fonctions de conversion.

FONCTIONS SQL



DEUX TYPES DE FONCTIONS SQL

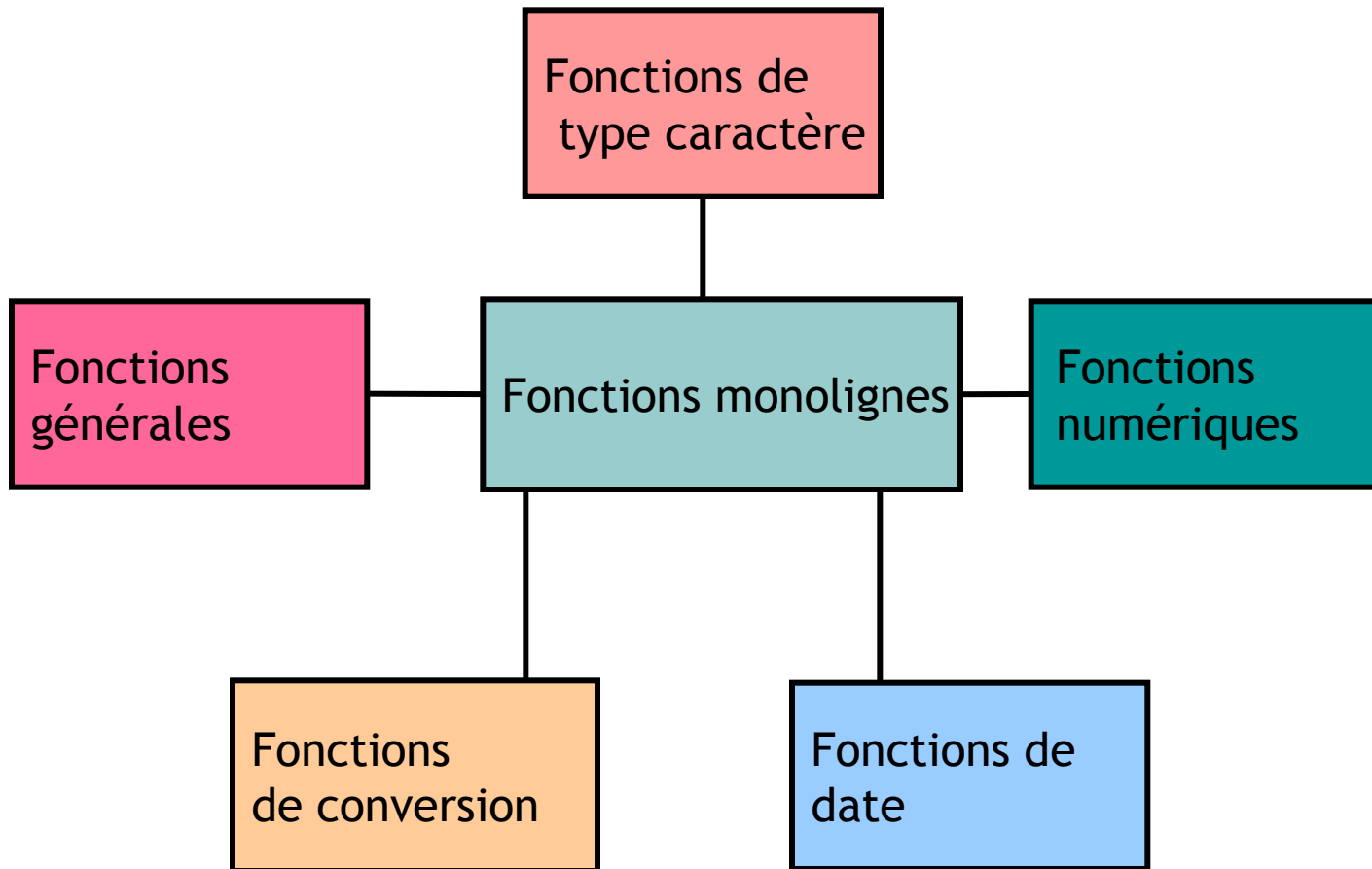


FONCTIONS MONOLIGNES

- ◉ Les fonctions monoligne :
 - ◉ Manipulent des données
 - ◉ Acceptent des arguments et renvoient une seule valeur.
 - ◉ Opèrent sur chaque ligne renvoyée.
 - ◉ Renvoient un seul résultat par ligne.
 - ◉ Peuvent modifier le type de données
 - ◉ Peuvent être imbriquées
 - ◉ Acceptent des arguments pouvant être une colonne ou une expression.

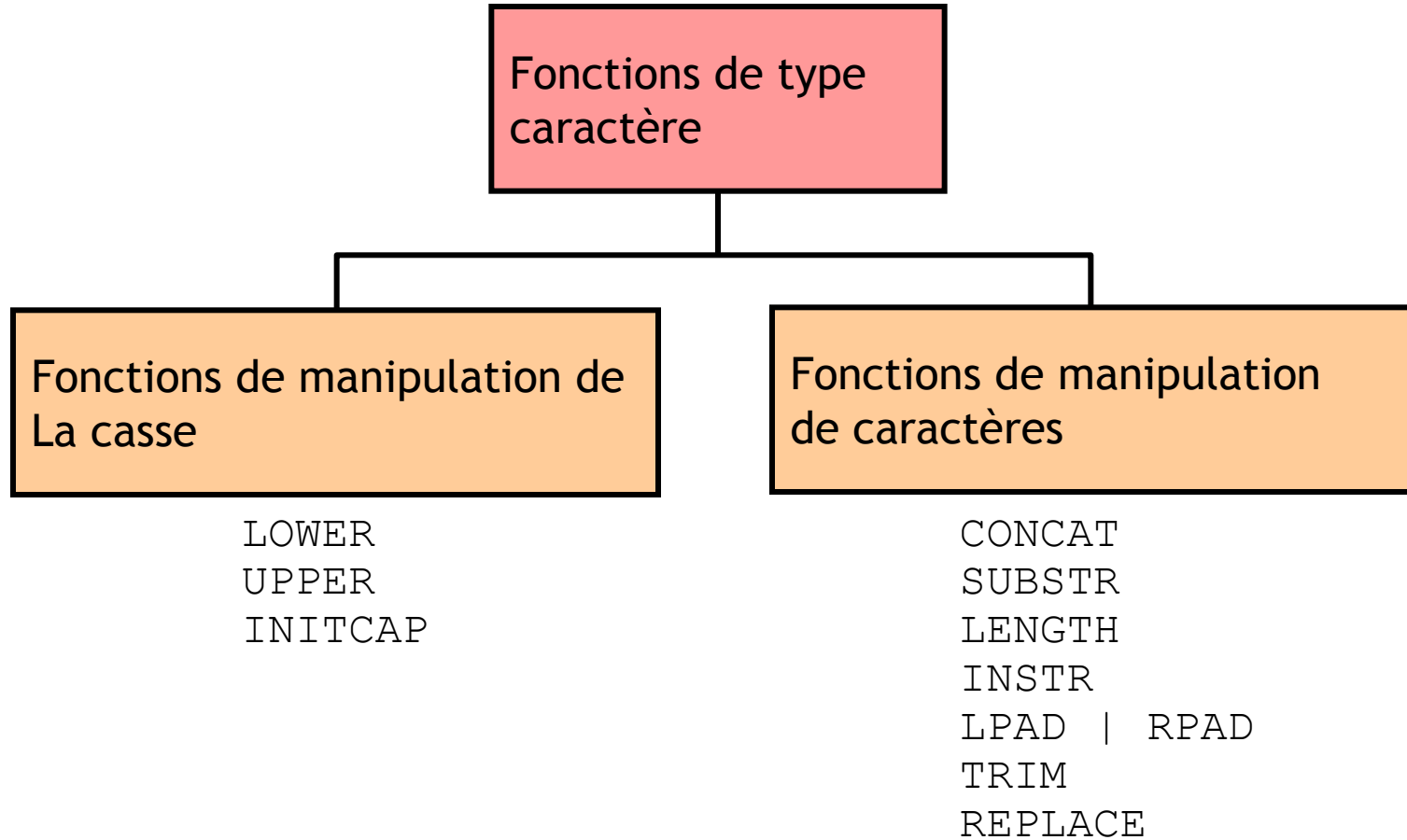
```
function_name [(arg1, arg2,...)]
```


FONCTIONS MONOLIGNE



PARTIE 1

FONCTIONS DE TYPE CARACTÈRE



FONCTIONS DE MANIPULATION DE LA CASSE

- ◉ Ces fonctions convertissent la casse de chaînes de caractères :

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

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

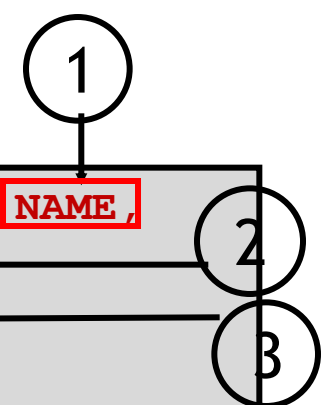
FONCTIONS DE MANIPULATION DE CARACTÈRES

- ◉ Ces fonctions manipulent des caractères:


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

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 de la division.

Function	Result
ROUND (45.926, 2)	45.93
TRUNC (45.926, 2)	45.92
MOD (1600, 300)	100

UTILISER LA FONCTION ROUND

The diagram illustrates the use of the ROUND function in SQL. It shows a query and its results, with numbered annotations explaining the components.

Query:

```
SELECT ROUND(45.923, 2), ROUND(45.923, 0),  
       ROUND(45.923, -1)  
FROM DUAL;
```

Results:

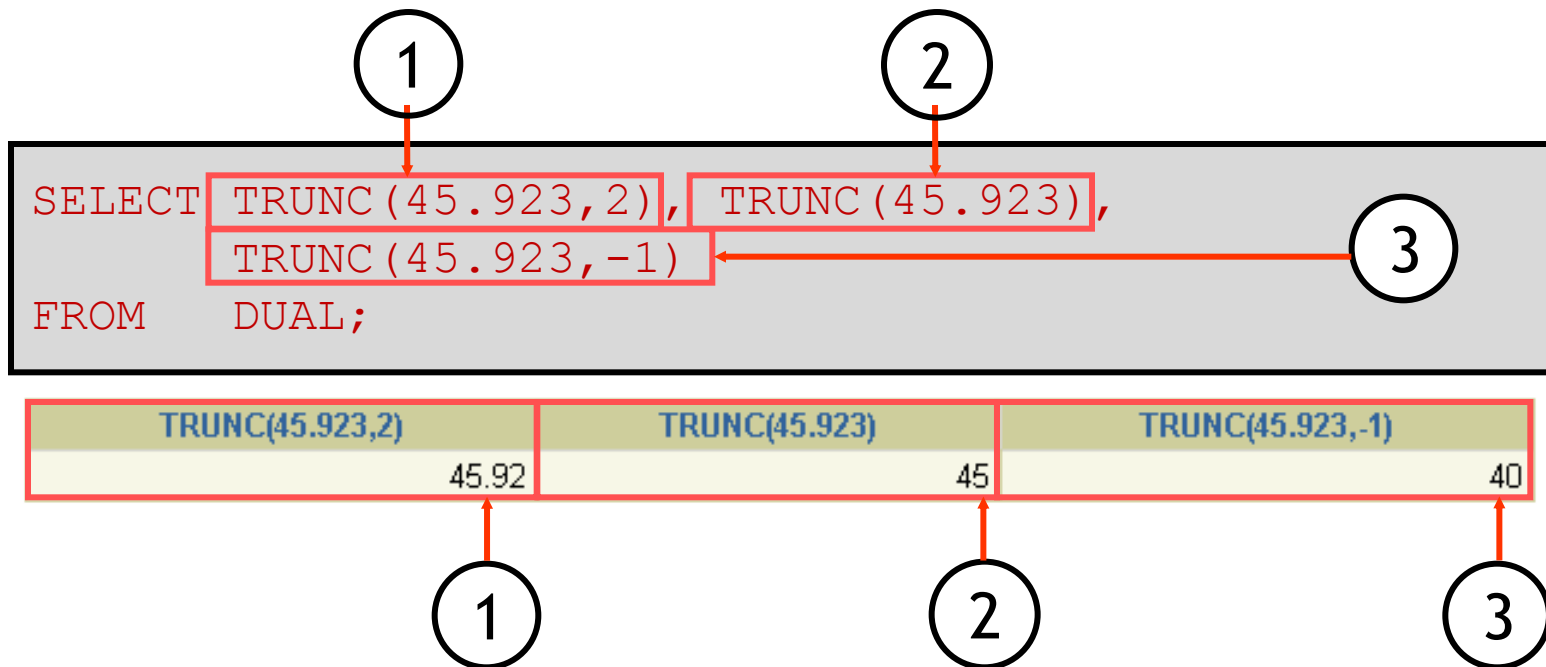
ROUND(45.923,2)	ROUND(45.923,0)	ROUND(45.923,-1)
45.92	46	50

Annotations:

- 1: Points to the first argument of the ROUND function (45.923).
- 2: Points to the second argument of the ROUND function (2, 0, or -1).
- 3: Points to the ROUND function itself.

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

UTILISER LA FONCTION TRUNC



UTILISER LA FONCTION MOD

- Pour tous les employés dont le poste est Sales Representative (SA_REP), calculer le reste du salaire après division par 5000 :

```
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 numériques interne : siècle, année, mois, jour, heures, minutes et secondes.
- ◉ Le format de date par défaut est DD-MON-RR.

```
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
- ◎ Calcul arithmétique 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.

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

Function	Result
MONTHS_BETWEEN	Nombre de mois entre deux dates
ADD_MONTHS	Ajout d'un mois 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

Function	Result
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'

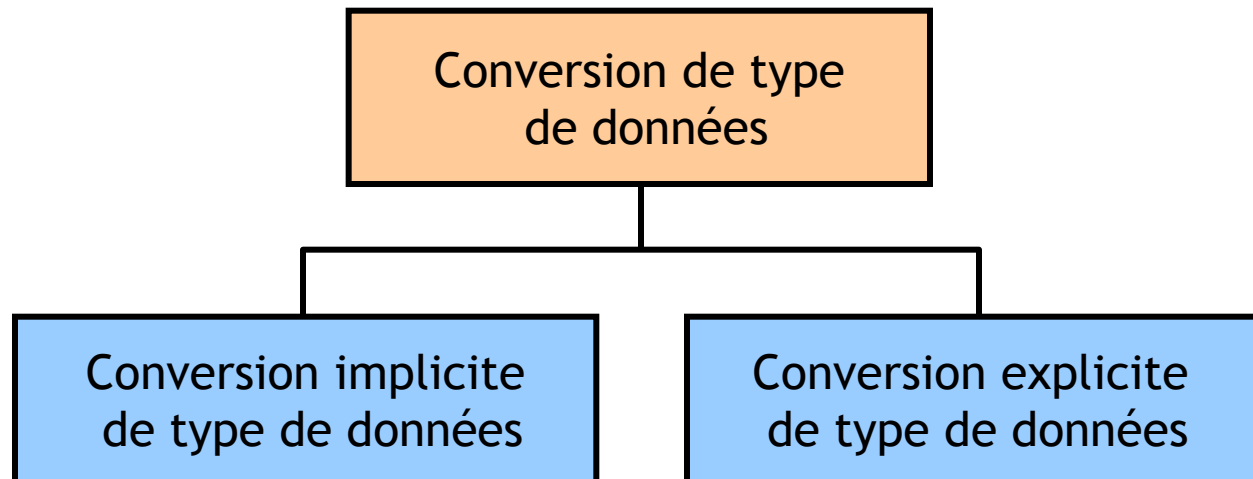
UTILISER DES FONCTIONS DE DATE

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

Function	Result
<code>ROUND (SYSDATE , 'MONTH')</code>	<code>01-AUG-03</code>
<code>ROUND (SYSDATE , 'YEAR')</code>	<code>01-JAN-04</code>
<code>TRUNC (SYSDATE , 'MONTH')</code>	<code>01-JUL-03</code>
<code>TRUNC (SYSDATE , 'YEAR')</code>	<code>01-JAN-03</code>

PARTIE 2

FONCTIONS DE CONVERSION



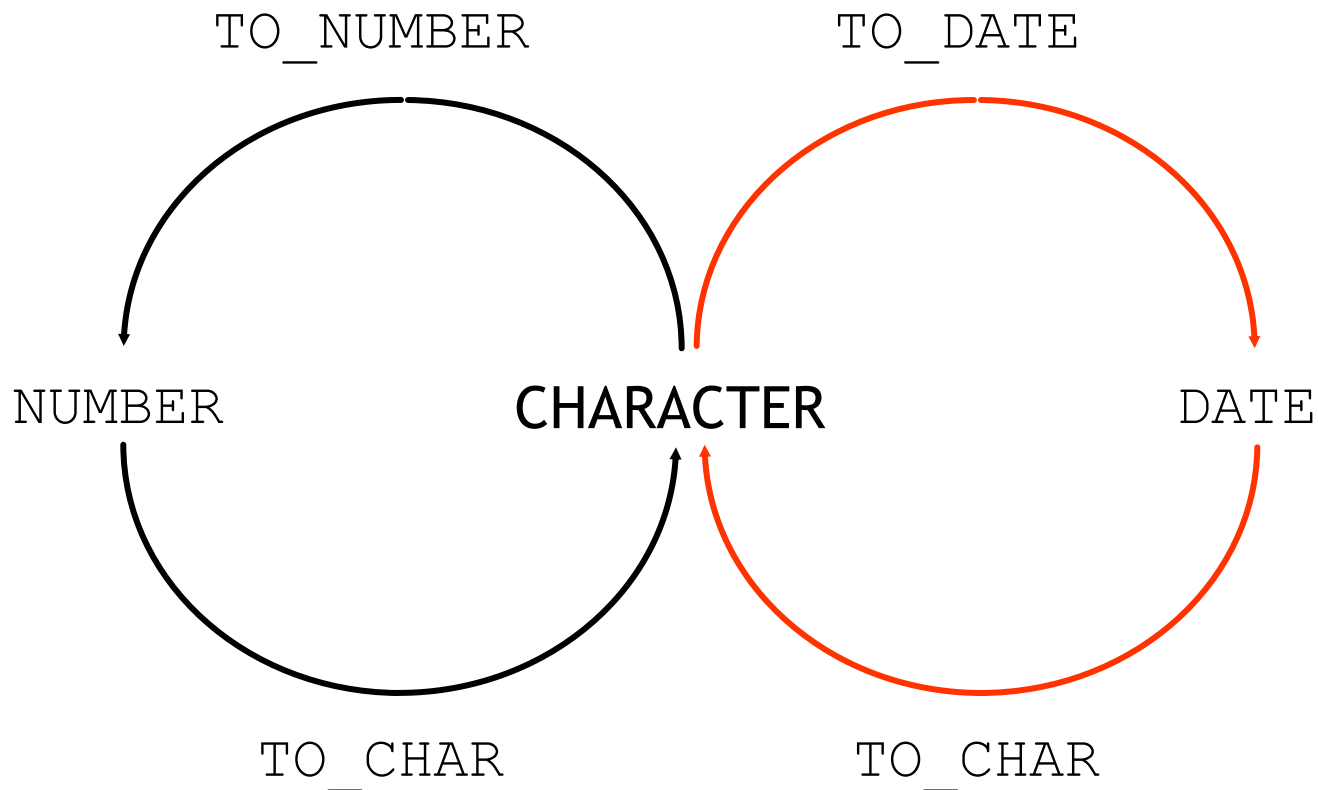
CONVERSION IMPLICITE DE TYPE DE DONNÉES

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

From	To
VARCHAR2 or CHAR	NUMBER
VARCHAR2 or CHAR	DATE

- L'expression `hire_date > '01-Jan-90'` entraîne la conversion implicite de la chaîne '01-Jan-90' en date.
- L'expression `salary = '2000'` entraîne la conversion implicite de la chaîne '2000' en valeur numérique 2000.

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

Element	Result
YYYY	Année complète en 4 chiffres
YY	Année en deux 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 formattent la partie heure de la date:

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

- Ajouter 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.

UTILISEZ LA FONCTION TO_CHAR AVEC DES NOMBRES

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

- 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:

Element	Result
9	Représente un nombre
0	Force l'affichage d'un zéro
\$	Insère un signe dollar
L	Utilise le symbole monétaire local
.	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 LA FONCTION 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')
```

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)

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.

1

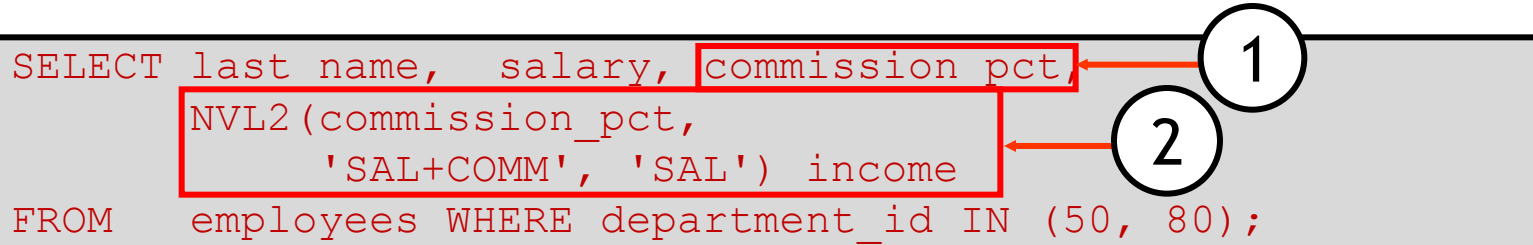
2

1

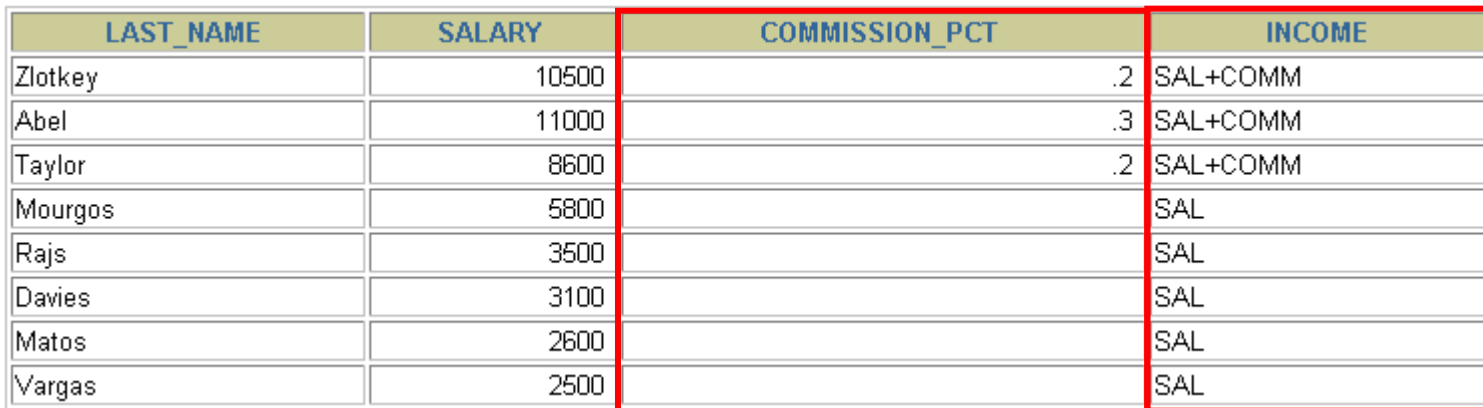
2

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.

SYNTHÈSE

- ◎ Ce chapitre vous a permis d'apprendre à:
 - Effectuer des calculs sur des données.
 - Modifier des données individuelles.
 - Manipuler la sortie de groupes de lignes.
 - Modifier des formats de date pour un affichage.
 - Convertir des types de données de colonnes.
 - Utiliser les fonctions NVL.