

GUIDE DE RÉVISION - FONCTIONS SQL

STRUCTURE DE BASE

```
SELECT [DISTINCT] {*, column [alias], ...}  
FROM table  
[WHERE condition(s)]  
[ORDER BY {column, expr, alias} [ASC|DESC]];
```

1. FONCTIONS CARACTÈRES

A. Transformation Casse

Fonction	Description	Exemple	Résultat
LOWER ()	Convertit en minuscules	LOWER('Bonjour')	bonjour
UPPER ()	Convertit en majuscules	UPPER('Bonjour')	BONJOUR
INITCAP ()	Première lettre en majuscule	INITCAP('bonjour monde')	Bonjour Monde

B. Manipulation Texte

Fonction	Description	Exemple	Résultat
CONCAT ()	Fusionne deux chaînes	CONCAT('SQL', 'Plus')	SQLPlus
SUBSTR ()	Extrait une portion	SUBSTR('SQL Plus', 1, 3)	SQL
LENGTH ()	Compte les caractères	LENGTH('SQL')	3
INSTR ()	Position d'une sous-chaîne	INSTR('SQL', 'S')	1
LPAD ()	Complète à gauche	LPAD('SQL', 5, '*')	**SQL

2. FONCTIONS NUMÉRIQUES

Fonction	Description	Exemple	Résultat
ROUND ()	Arrondit	ROUND(125.815, 1)	125.8
MOD ()	Reste division	MOD(15, 4)	3

3. FONCTIONS DATE

A. Opérations

```
-- Ajouter des jours
SELECT hiredate + 7 FROM employees; -- Ajoute 7 jours

-- Différence entre dates
SELECT DATEDIFF(SYSDATE(), hiredate) FROM employees;
```

B. Formats Date Courants

Format	Signification	Exemple
--------	---------------	---------

YYYY	Année 4 chiffres	2024
------	------------------	------

MM	Mois 2 chiffres	03
----	-----------------	----

MONTH	Nom du mois	MARS
-------	-------------	------

DD	Jour 2 chiffres	09
----	-----------------	----

DY	Jour abrégé	LUN
----	-------------	-----

4. FONCTIONS CONVERSION

A. Conversions Principales

```
-- Date vers Texte
TO_CHAR(date_col, 'DD-MM-YYYY')

-- Texte vers Nombre
TO_NUMBER('123.45')

-- Texte vers Date
TO_DATE('2024-03-09', 'YYYY-MM-DD')
```

5. FONCTIONS SPÉCIALES

A. Gestion des NULL

```
-- NVL : Remplace NULL par une valeur
SELECT ename, NVL(commission, 0) FROM emp;
```

B. Conditions

```
-- CASE
SELECT ename,
       CASE
         WHEN salary > 3000 THEN 'Élevé'
         WHEN salary > 2000 THEN 'Moyen'
         ELSE 'Bas'
       END as niveau_salaire
FROM employees;

-- DECODE (Oracle)
SELECT job,
       DECODE(job,
              'DEV', salary * 1.1,
              'MANAGER', salary * 1.2,
              salary) as nouveau_salaire
FROM employees;
```

6. IMBRICATION DE FONCTIONS

```
-- Exemple 1: Multiple transformation
SELECT UPPER(SUBSTR(ename, 1, 3)) FROM emp;

-- Exemple 2: Conversion et formatage
SELECT TO_CHAR(ROUND(salary * 1.1), '$999,999')
FROM employees;
```

POINTS CLÉS À RETENIR

1. Les fonctions mono-ligne retournent UN résultat par ligne
2. Les types de données doivent correspondre dans les conversions
3. L'imbrication se fait de l'intérieur vers l'extérieur
4. Toujours vérifier la compatibilité des types de données

EXERCICES TYPES

```
-- 1. Formatage nom employé
SELECT CONCAT(UPPER(SUBSTR(ename, 1, 1)),
              LOWER(SUBSTR(ename, 2)))
FROM emp;

-- 2. Calcul ancienneté
SELECT ename,
       ROUND(MONTHS_BETWEEN(SYSDATE, hiredate)/12) as années
FROM emp;

-- 3. Rapport salaire
SELECT ename,
       CASE
         WHEN sal > (SELECT AVG(sal) FROM emp)
         THEN 'Au-dessus moyenne'
         ELSE 'Sous moyenne'
       END as statut_salaire
FROM emp;
```