

Cours base de données

# CHAPITRE 5 : SQL

Dr Coulibaly K. 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 mono lignes 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 5.1:

# EXTRAIRE DES DONNÉES À L'AIDE DE L'INSTRUCTION SQL SELECT

# OBJECTIFS

- ◉ Décrire des fonctionnalités des instructions SQL SELECT.
- ◉ Exécuter des instructions SELECT de base.

# FONCTIONNALITÉS DES INSTRUCTIONS SELECT

**Projection**

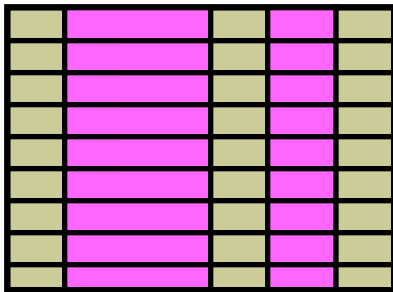



Table 1

**Sélection**

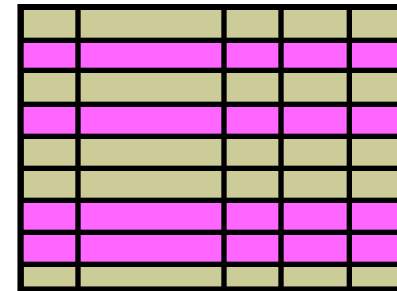



Table 1

**Jointure**

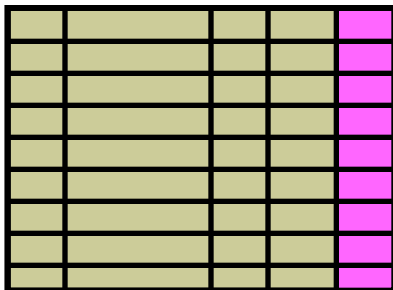



Table 1

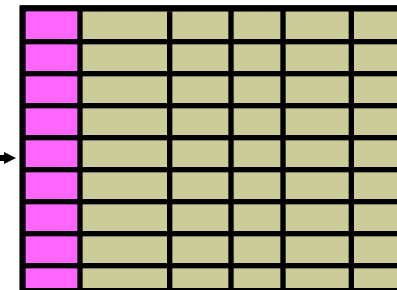



Table 2

# INSTRUCTION SELECT DE BASE

```
SELECT * | { [DISTINCT] column | expression [alias] , ... }  
FROM      table;
```

- ◉ L'instruction SELECT identifie les colonnes à afficher.
- ◉ La clause FROM identifie la table contenant ces colonnes.

# SÉLECTIONNER TOUTES LES COLONNES

```
SELECT *
```

```
FROM departments ;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

8 rows selected.



# SÉLECTIONNER DES COLONNES SPÉCIFIQUES

```
SELECT department_id, location_id
FROM departments;
```

DEPARTMENT_ID	LOCATION_ID
10	1700
20	1800
50	1500
60	1400
80	2500
90	1700
110	1700
190	1700

8 rows selected.

# ECRIRE DES INSTRUCTIONS SQL

- ⦿ Les instructions SQL ne distinguent pas les majuscules des minuscules.
- ⦿ Les instructions SQL peuvent être écrites sur une ou plusieurs lignes.
- ⦿ Les mots-clés ne peuvent pas être abrégés ou répartis sur plusieurs lignes.
- ⦿ Les clauses sont généralement placées sur des lignes distinctes.

# EXPRESSIONS ARITHMÉTIQUES

- Créez les expressions avec des données de type nombre et date à l'aide d'opérateurs arithmétiques

Opérateur	Description
<b>+</b>	<b>Ajouter</b>
<b>-</b>	<b>Soustraire</b>
<b>*</b>	<b>Multiplier</b>
<b>/</b>	<b>Diviser</b>

# UTILISEZ DES OPÉRATEURS ARITHMÉTIQUES

```
SELECT last_name, salary, salary + 300
FROM employees;
```

LAST_NAME	SALARY	SALARY+300
King	24000	24300
Kochhar	17000	17300
De Haan	17000	17300
Hunold	9000	9300
Ernst	6000	6300

...  
20 rows selected.

# PRIORITÉS DES OPÉRATEURS

```
SELECT last_name, salary, 12*salary+100
FROM employees;
```

①

LAST_NAME	SALARY	12*SALARY+100
King	24000	288100
Kochhar	17000	204100
De Haan	17000	204100

...

20 rows selected.

```
SELECT last_name, salary, 12*(salary+100)
FROM employees;
```

②

LAST_NAME	SALARY	12*(SALARY+100)
King	24000	289200
Kochhar	17000	205200
De Haan	17000	205200

...

20 rows selected.

# DÉFINIR UNE VALEUR NULL

- Une valeur NULL est une valeur qui n'est pas disponible, pas affectée ou inconnue .
- Une valeur NULL est différente d'un zéro ou d'un espace.

```
SELECT last_name, job_id, salary, commission_pct  
FROM employees;
```

LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
King	AD_PRES	24000	
Kochhar	AD_VP	17000	
...			
Zlotkey	SA_MAN	10500	.2
Abel	SA_REP	11000	.3
Taylor	SA_REP	8600	.2
...			
Gietz	AC_ACCOUNT	8300	

20 rows selected.

# VALEURS NULL DANS LES EXPRESSIONS ARITHMÉTIQUES

- Les expressions arithmétiques contenant une valeur NULL renvoient une valeur NULL.

```
SELECT last_name, 12*salary*commission_pct
FROM employees;
```

LAST_NAME	12*SALARY*COMMISSION_PCT
King	
Kochhar	
...	
Zlotkey	25200
Abel	39600
Taylor	20640
...	
Gietz	

20 rows selected.

# DÉFINIR UN ALIAS DE COLONNE

## ◎ Un alias de colonne :

- Renomme un entête de colonne
- Est utile avec les calculs
- Suit immédiatement le nom d'une colonne (le mot clé facultatif AS peut également être utilisé entre le nom de la colonne et l'alias)
- Nécessité des guillemets s'il contient des espaces ou des caractères spéciaux (# \$), ou s'il distingue les majuscules des minuscules.



# UTILISER DES ALIAS DE COLONNE

```
SELECT last_name AS name, commission_pct comm
FROM employees;
```

NAME	COMM
King	
Kochhar	
De Haan	

...

20 rows selected.

```
SELECT last_name "Name", salary*12 "Annual Salary"
FROM employees;
```

Name	Annual Salary
King	288000
Kochhar	204000
De Haan	204000

...

20 rows selected.

# OPÉRATEUR DE CONCATÉNATION

## ◉ Un opérateur de concaténation :

- Lie des colonnes ou des chaînes de caractères à d'autres colonnes.
- Est représenté par deux barres verticales (||).
- Crée une colonne résultante qui est une expression de type caractère.

```
SELECT    last_name||job_id AS "Employees"  
FROM      employees;
```

Employees
KingAD_PRES
KochharAD_VP
De HaanAD_VP
...

20 rows selected.

# CHAÎNES DE CARACTÈRES LITTÉRALES

- ◉ Un littéral est un caractère, un nombre ou une date inclus dans l'instruction SELECT.
- ◉ Les valeurs littérales de type date et caractère doivent être incluses entre apostrophes.
- ◉ Chaque chaîne de caractères est sortie une fois pour chaque ligne renvoyée.

# UTILISER DES CHAÎNES DE CARACTÈRES LITTÉRALES

```
SELECT last_name || ' is a ' || job_id  
       AS "Employee Details"  
FROM   employees;
```

Employee Details
King is a AD_PRES
Kochhar is a AD_VP
De Haan is a AD_VP
Hunold is a IT_PROG
Ernst is a IT_PROG
Lorentz is a IT_PROG
Mourgos is a ST_MAN
Rajs is a ST_CLERK

...

20 rows selected.

# OPÉRATEUR DE DÉLIMITATION (Q) ALTERNATIF

- ◉ Indiquer votre propre délimiteur de chaînes de caractères. Exp : [], {}, () ou <>
- ◉ Choisir n'importe quel délimiteur
- ◉ Améliorer la lisibilité et la simplicité d'utilisation

```
SELECT department name ||  
       q'[, it's assigned Manager Id: ]'  
       || manager_id  
       AS "Department and Manager"  
FROM departments;
```

Department and Manager
Administration, it's assigned manager ID: 200
Marketing, it's assigned manager ID: 201
Shipping, it's assigned manager ID: 124

...

8 rows selected.

# LIGNES EN DOUBLE

- Par défaut, les interrogations renvoient toutes les lignes, y compris les lignes en double.

```
SELECT department_id  
FROM employees;
```

1

DEPARTMENT_ID	
	90
	90
	90

...

20 rows selected.

```
SELECT DISTINCT department_id  
FROM employees;
```

2

DEPARTMENT_ID	
	10
	20
	50

...

8 rows selected.

# AFFICHER LA STRUCTURE D'UNE TABLE

- Utilisez la commande DESCRIBE pour afficher la structure d'une table.

```
DESC[RIBE] nom_table
```

# AFFICHER LA STRUCTURE D'UNE TABLE

```
DESCRIBE employees
```

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)



# SYNTHÈSE

- ◎ Ce chapitre vous a permis d'apprendre à :
  - Ecrire une instruction SELECT qui :
    - Renvoie toutes les lignes et colonnes
    - Renvoie des colonnes particulières d'une table.
    - Utilise des alias de colonne pour afficher des entêtes plus évocateurs

```
SELECT * | { [DISTINCT] column | expression [alias], ... }  
FROM table;
```