

# **Extraire des données à l'aide de l'instruction SQL SELECT**

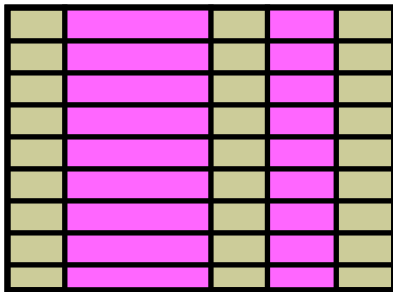
# Objectifs

**A la fin de ce chapitre, vous pourrez :**

- **décrire les fonctionnalités des instructions SQL `SELECT`**
- **exécuter une instruction `SELECT` de base**

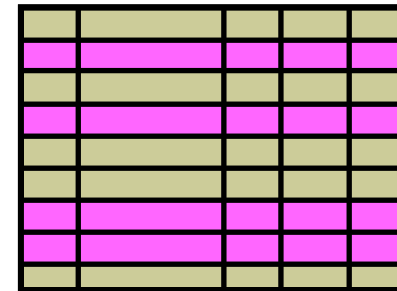
# Fonctionnalités des instructions SQL SELECT

**Projection**



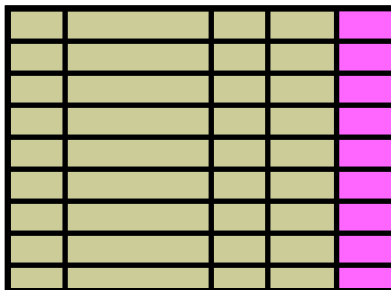

**Table 1**

**Sélection**

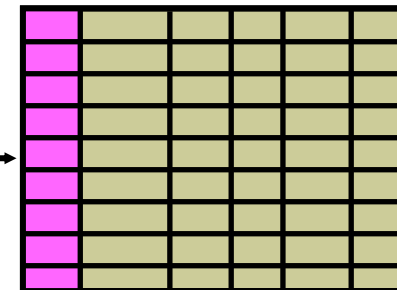



**Table 1**

**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.**
- **Des indentations sont utilisées pour améliorer la lisibilité.**
- **Les instructions SQL peuvent être terminées par un point-virgule (;). Les points-virgules sont obligatoires si vous exécutez plusieurs instructions SQL.**

# Expressions arithmétiques

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

Opérateur	Description
+	Ajouter
-	Soustraire
*	Multiplier
/	Diviser



# Utiliser 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.

# Priorité 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, inconnue ou inapplicable.
- 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;
```

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

20 rows selected.

# Définir un alias de colonne

**Un alias de colonne :**

- **Renomme un en-tête 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)**
- **Nécessite 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.

# 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.

# 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 *iSQL\*Plus* DESCRIBE pour afficher la structure d'une table :

```
DESC[RIBE] tablename
```

# 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)