



SMI-S5

# SGBD ORACLE : LID-Partie 1

Pr. Issam QAFFOU

Laboratoire Ingénierie des Systèmes d'Information  
Département d'Informatique  
FSSM-UCA

# Langage d'interrogation des données

- ❑ **Une requête** permet de rechercher des données dans une ou plusieurs tables à partir de critères simples ou complexes.
- ❑ **Une requête** est une instruction `SELECT`.

# Syntaxe SELECT

**SELECT** [ { **DISTINCT** | **UNIQUE** } | **ALL** ] *listeColonnes*  
**FROM** *nomTable1* [*,nomTable2*]...  
[ **WHERE** *condition* ]  
[ **GROUP BY** *col1, col2..*]  
[ **HAVING** *condition* ]  
[ { **UNION** | **UNION ALL** | **INTERSECT** | **MINUS** } ( *sousRequête* )]  
[ **ORDER BY** *col1* ] ;

# Projection : éléments du SELECT

Extraction de toutes les colonnes :

```
SELECT * FROM Pilote;
```

Pilote

brevet	nom	nbHVol	compa
PL-1	Gratien Viel	450	AF
PL-2	Didier Donsez	0	AF
PL-3	Richard Grin	1000	SING
PL-4	Placide Fresnais	2450	CAST
PL-5	Daniel Vielle		AF

# Projection : éléments du SELECT

## Extraction de certaines colonnes :

**SELECT compa, brevet FROM Pilote;**

COMPA	BREVET
-----	-----
AF	PL-1
AF	PL-2
SING	PL-3
CAST	PL-4
AF	PL-5

# Projection : éléments du SELECT

## Alias : AS

Les alias permettent de renommer des colonnes à l’affichage ou des tables dans la requête.

```
SELECT aliasPilotes.compa AS c1,  
aliasPilotes.nom  
FROM Pilote aliasPilotes;
```

C1	NOM
AF	Gratien Viel
AF	Didier Donsez
SING	Richard Grin
CAST	Placide Fresnais
AF	Daniel Vielle

# Projection : éléments du SELECT

## Duplicatas

```
SELECT compa  
FROM Pilote;
```

```
COMPA  
-----  
AF  
AF  
SING  
CAST  
AF
```

```
SELECT DISTINCT(compa)  
FROM Pilote;
```

```
COMPA  
-----  
AF  
CAST  
SING
```

# Projection : éléments du SELECT

## Éléments calculés

- Il est possible d'évaluer et d'afficher simultanément des expressions dans la clause SELECT (types NUMBER, DATE et INTERVAL).

### Exemple :

```
SELECT brevet, nbHVol,  
       nbHVol*nbHVol AS auCarré,  
       10*nbHVol+5/2  
FROM Pilote;
```



# Projection : éléments du SELECT

## Ordre

- Pour trier le résultat d'une requête, il faut spécifier la clause d'ordonnancement par **ORDER BY**
- **ASC** ou **DESC** : tri ascendant ou descendant (par défaut ASC).

## Exemple :

```
SELECT brevet, nom  
FROM Pilote  
ORDER BY nom DESC;
```

# Projection : éléments du SELECT

## Concaténation

Il est possible de concaténer des expressions par l'opérateur de concaténation **||**

La colonne résultante est considérée comme une chaîne de caractères.

### Exemple :

```
SELECT brevet,  
nom || ' ' || compa AS "Embauche"  
FROM Pilote;
```

# Projection : éléments du SELECT

*SELECT [ { DISTINCT / UNIQUE } / ALL ] listeColonnes  
FROM nomTable [aliasTable]  
[ clauseOrdonnancement ] ;*

- ❑ DISTINCT et UNIQUE jouent le même rôle : ne pas prendre en compte les duplicatas.
- ❑ ALL : prend en compte les duplicatas (option par défaut).
- ❑ *ListeColonnes* : { \* / *expression1* [[AS] *alias1* ] [, *expression2* [[AS] *alias2* ] ... }.
- ❑ FROM : désigne la table (qui porte un alias ou non) à interroger.
- ❑ *clauseOrdonnancement* : tri sur une ou plusieurs colonnes ou expressions.

# Restriction : WHERE

La **clause WHERE** limite la recherche aux enregistrements qui respectent une **condition** simple ou complexe.

La condition de la clause WHERE est composée de colonnes, d'expressions, de constantes liées entre eux par des opérateurs:

- de comparaison (>, =, <, >=, <=, <>) ;
- logiques (NOT, AND ou OR) ;
- intégrés (BETWEEN, IN, LIKE, IS NULL).

## Clause WHERE : Exemples

Afficher les clients dont la date de naissance n'est pas le 25 décembre 1960 ?

```
SELECT *  
FROM clients  
WHERE naissance <> '25/12/1960';
```

## Clause WHERE : Exemples

Afficher les clients dont on connait le code postal ?

```
SELECT *  
FROM clients  
WHERE code_postal IS NOT NULL
```

## Clause WHERE : Exemples

Afficher les articles bleus dont le prix est supérieur ou égale à 100 ou inférieur strictement à 10 ?

```
SELECT *  
FROM articles  
WHERE couleur='bleu'  
      AND (prix >= 100 OR prix < 10) ;
```



## Clause WHERE : Exemples

Afficher les articles dont le prix est entre 10 et 100 ?

```
SELECT *
```

```
FROM articles
```

```
WHERE prix BETWEEN 10 AND 100;
```



## Clause WHERE : Exemples

Afficher les clients qui ne s'appellent ni 'Salah' ni 'Salam' ni 'Alami' ?

```
SELECT *
```

```
FROM clients
```

```
WHERE nom NOT IN ('Salah', 'Salam', 'Alami');
```

## Clause WHERE : Exemples

Afficher les clients dont le nom commence par A?

```
SELECT *  
FROM clients  
WHERE nom LIKE 'A%';
```

## Clause WHERE : Exemples

Afficher les clients dont le nom ne contient pas m?

```
SELECT *  
FROM clients  
WHERE nom NOT LIKE '%m%';
```