

---

# Le langage SQL pour Oracle

## - partie 1 : SQL comme LDD

---

# SQL : Introduction

- SQL : Structured Query Language
- langage de gestion de bases de données relationnelles pour
  - Définir les données (LDD)
  - interroger la base de données (Langage de requêtes)
  - manipuler les données (LMD)
  - contrôler l'accès aux données (LCD)

# SQL comme LDD

## ■ Identificateurs

- ❑ Lettre suivie par : lettre ou chiffre ou \_ ou # ou \$
- ❑ Chaîne de caractères entre guillemets
- ❑ Maximum 30 caractères
- ❑ Différent d'un mot clé :

ASSERT, ASSIGN, AUDIT, COMMENT, DATE, DECIMAL,  
DEFINITION, FILE, FORMAT, INDEX, LIST, MODE,  
OPTION, PARTITION, PRIVILEGE, PUBLIC, SELECT,  
SESSION, SET, TABLE

- pas de distinction entre majuscules et minuscules

# SQL comme LDD

- Tables :
  - relations d'un schéma relationnel stockées
  - table : formée de lignes et de colonnes
  - SQL2 : nom d'une table précédée du nom d'un schéma ORACLE :
    - nom de schéma remplacé par le nom d'utilisateur qui a créé la table
    - par défaut, le schéma est le nom de l'utilisateur connecté

# SQL comme LDD

- Tables : colonnes
  - ❑ toutes les données d'une colonne sont du même type
  - ❑ identificateur unique pour les colonnes d'une même table
  - ❑ 2 colonnes dans 2 tables différentes peuvent avoir le même nom
  - ❑ nom complet d'une colonne comprend le nom complet de la table à laquelle elle appartient
  - ❑ exemple : `DEPARTEMENTS.DEPARTEMENT ID`  
ou `HR.DEPARTEMENTS.DEPARTEMENT ID`

# SQL comme LDD

- Types de données SQL2 :
  - ❑ types pour les chaînes de caractères
  - ❑ types numériques
  - ❑ types temporels (date, heure, . . .)
  - ❑ SQL2 n'a pas de type pour les données volumineuses (images , sons)
  - ❑ SQL2 ne permet pas à l'utilisateur de créer ses propres types

# SQL comme LDD

- Types pour les chaînes de caractères :
  - CHAR(taille)
    - Chaînes de caractères de longueur fixe
    - codage en longueur fixe : remplissage de blancs
    - taille comprise entre 1 et 2000 octets
  - VARCHAR(taille\_max) mais de préférence VARCHAR2(taille\_max)
    - Chaînes de caractères de longueur variable
    - taille comprise entre 1 et 4000 octets
  - constantes
    - Chaînes de caractères entre guillemets.

# SQL comme LDD

- Types pour les chaînes de caractères : exemples
  - NCHAR(5) : chaînes de 5 caractères
  - VARCHAR2(20) : chaînes de 20 caractères au plus
  - 'Administration', 'Marketing'

# SQL comme LDD

- Types numériques SQL2 :
  - types numériques pour les entiers :
    - SMALLINT pour 2 octets
    - INTEGER pour 4 octets
  - types numériques pour les décimaux à virgule flottante :
    - REAL
    - DOUBLE PRECISION ou FLOAT
  - types numériques pour les décimaux à virgule fixe :
    - DECIMAL(nb chiffres, nb décimales)
    - NUMERIC(nb chiffres, nb décimales)
  - Constantes
    - exemples : 43.8, -13, 5.3E-6

# SQL comme LDD

- Types numériques Oracle :
  - ❑ ORACLE accepte tous les types numériques de SQL2 mais il les traduit dans ses propres types
  - ❑ NUMBER : nombre en virgule flottante avec jusqu'à 38 chiffres significatifs
  - ❑ NUMBER(nb chiffres, nb décimales) : nombre en virgule fixe

# SQL comme LDD

- Types temporels SQL2 :
  - DATE : pour les dates
  - TIME : pour les heures, minutes et secondes
  - TIMESTAMP : pour un moment précis : date et heure, minutes et secondes (précision jusqu'à la microseconde).

# SQL comme LDD

- Types temporels Oracle :
  - DATE remplace DATE et TIME de SQL2
  - DATE correspond à une date avec précision jusqu'à la seconde
  - Constantes :
    - Exemples : '1/05/2007' ou '1 MAY 2007'

# SQL comme LDD

- Type booléen SQL2 :
  - BIT
    - Pour enregistrer la valeur d'un bit
    - Exemples : BIT(1), BIT(4)
  - Non supporté par Oracle.

# SQL comme LDD

- Exemple jouet :
  - Schéma de relations :
    - AVION(NumAV, NomAV, CAP, LOC)
    - PILOTE(NumPIL, NomPIL, VILLE)
    - VOL(NumVOL, NumAV, NumPIL, Vd, Va, Hd, Ha)

# SQL comme LDD

- Les trois tables correspondantes :

AVION				PILOTE		
NumAV	NomAV	CAP	LOC	NumPIL	NomPIL	VILLE
100	Airbus	300	Nice	1	Laurent	Nice
101	Airbus	300	Paris	2	Sophie	Paris
102	Carav	200	Toulouse	3	Claude	Grenoble

VOL						
NumVOL	NumAV	NumPIL	Vd	Va	Hd	Ha
it100	100	1	Nice	Paris	7	8
it101	100	2	Paris	Toulouse	11	12
it102	101	1	Paris	Nice	12	13
it103	102	3	Grenoble	Toulouse	9	11
it104	101	3	Toulouse	Grenoble	17	18

# SQL comme LDD

- AVION(NumAV, NomAV, CAP, LOC)
  - numAV NUMBER(4)
  - NomAV VARCHAR2(20)
  - CAP NUMBER(4)
  - LOC VARCHAR2(15)
- PILOTE(NumPIL, NomPIL, VILLE)
  - NumPIL NUMBER(4)
  - NomPIL VARCHAR2(20)
  - VILLE VARCHAR2(15)
- VOL(NumVOL, NumAV, NumPIL, Vd, Va, Hd, Ha)
  - NumVOL VARCHAR2(5)
  - NumAV NUMBER(4)
  - NumPIL NUMBER(4)
  - Vd VARCHAR2(15)
  - Va VARCHAR2(15)
  - Hd DATE
  - Ha DATE

# SQL comme LDD

- Création de table :
  - Syntaxe :
    - CREATE TABLE nom de table (liste de définition\_de\_colonne, [liste de contrainte\_de\_table]) ;
    - définition\_de\_colonne ::= nom\_de\_colonne  
(nom\_de\_domaine ou type)  
[liste de contrainte\_de\_colonne]  
[DEFAULT valeur\_par\_défaut]

# SQL comme LDD

- Création de table : contrainte de colonne
  - `Contrainte_de_colonne ::= [CONSTRAINT nom]  
type_de_contrainte_de_colonne`
  - `type_de_contrainte_de_colonne ::=`  
PRIMARY KEY ou  
NOT NULL ou  
UNIQUE ou  
CHECK(condition\_sur\_valeur) ou REFERENCES  
nom\_de\_table(nom\_de\_colonne)

# SQL comme LDD

- Création de table contrainte de table :
  - `contrainte_de_table ::= [CONSTRAINT nom]  
type_de_contrainte_de_table`
  - `type_de_contrainte_de_table ::=`  
PRIMARY KEY (liste de `nom_de_colonne`) ou  
NOT NULL (liste de `nom_de_colonne`) ou  
UNIQUE (liste de `nom_de_colonne`) ou  
CHECK (`condition_sur_ligne`) ou  
FOREIGN KEY liste de `nom_de_colonne`  
REFERENCES `nom de de table` (liste de  
`nom_de_colonne`)

# SQL comme LDD

- Exemple : création de la table avion à partir du schéma :
- AVION(NumAV, NomAV, CAP, LOC)
  - CREATE TABLE AVION (  
    NumAV NUMBER(4) CONSTRAINT PK\_AVION PRIMARY KEY,  
    NomAV VARCHAR2(20),  
    CAP NUMBER(4) CONSTRAINT Dom\_CAP\_AVION CHECK (CAP ≥ 4),  
    LOC VARCHAR2(15)  
);

# SQL comme LDD

- Création de la table VOL à partir du schéma :  
VOL(NumVOL, NumAV, NumPIL, Vd, Va, Hd, Ha)
  - ❑ Contraintes de colonnes ?
  - ❑ Contraintes de table ?

# SQL comme LDD

```
■ CREATE TABLE VOL (  
    NumVOL VARCHAR2(5),  
    NumAV NUMBER(4),  
    NumPIL NUMBER(4),  
    Vd VARCHAR2(15) NOT NULL,  
    Va VARCHAR2(15) NOT NULL,  
    Hd DATE,  
    Ha DATE,  
    CONSTRAINT PK_VOL PRIMARY KEY(NumVol),  
    CONSTRAINT FK_VOL_NumAV FOREIGN KEY (NumAV) REFERENCES AVION,  
    CONSTRAINT FK_VOL_NumPIL FOREIGN KEY(NumPIL) REFERENCES PILOTE,  
    CONSTRAINT C1_VOL CHECK (Vd <> Va),  
    CONSTRAINT C2_VOL CHECK (Hd < Ha)  
);
```

# SQL comme LDD

- Suppression de table :  
    DROP TABLE nom;
- Quand une table est supprimée, ORACLE :
  - ❑ efface tous les index qui y sont attachés quelque soit le propriétaire
  - ❑ efface tous les privilèges qui y sont attachés  
    MAIS les vues et les synonymes se référant à cette table  
    ne sont pas supprimés

# SQL comme LDD

- Modification de table :  
`ALTER TABLE nom_de_table modification_de_table ;`
- modification de table ::=  
`ADD COLUMN définition_de_colonne`  
`ADD CONSTRAINT contrainte_de_table`  
`DROP COLUMN nom_de_colonne`  
`DROP CONSTRAINT nom_de_contrainte`

# SQL comme LDD

- Exemple : Ajout d'une colonne à la table vols de schéma :

```
VOL(NumVOL,NumAV,NumPIL,Vd,Va,Hd,Ha)
```

```
ALTER TABLE VOL ADD COLUMN COUT_VOL  
NUMBER(8) ;
```

le schéma devient :

```
VOL(NumVOL,NumAV,NumPIL,Vd,Va,Hd,Ha, COUT  
VOL)
```

# SQL comme LDD

- Insertion de lignes dans une table :

```
INSERT INTO nom_de_table [liste_de_colonnes] VALUES  
liste_de_valeurs;
```

ou

```
INSERT INTO nom_de_table [liste_de_colonnes] requête ;
```

# SQL comme LDD

## ■ Exemples :

- ❑ ajouter un avion dans la table AVION en respectant l'ordre des colonnes  
INSERT INTO AVION  
VALUES (100, 'Airbus', 200, 'Paris') ;
- ❑ ajouter un avion dans la table AVION sans connaître l'ordre  
INSERT INTO AVION (NumAV, CAP, LOC, NomAV)  
VALUES (101, 200, 'Paris', 'Airbus') ;
- ❑ ajouter un avion dans la table avions dont la localisation est INDEFINI  
INSERT INTO AVION (NumAV, NomAV, CAP)  
VALUES (102, 'Airbus', 200);  
ou  
INSERT INTO AVION  
VALUES (102, 'Airbus', 200, NULL) ;

# SQL comme LDD

- Suppression de lignes d'une table

```
DELETE [FROM] nom de table  
[WHERE condition] ;
```

- Exemples :

- vider la table AVION

```
DELETE FROM AVION;
```

- supprimer de la table AVION tous les avions dont la capacité est inférieur à 100

```
DELETE FROM AVION  
WHERE CAP < 100 ;
```

# SQL comme LDD

- Modification de lignes dans une table

```
UPDATE nom_de_table  
SET liste_expression_colonne  
[WHERE condition] ;
```

Expression\_colonne ::=  
nom\_de\_colonne = expression ou  
nom\_de\_colonne = requête

- Exemple :

- modifier la capacité de l'avion num éro 100

```
UPDATE AVION SET CAP = 300 WHERE NumAV = 100 ;
```