

Cours 5  
Langage SQL  
LDD  
Pr. M. Machkour

# Objectifs

- Présentation du langage SQL
- Étudier le langage de définition des données
  - La commande create
  - La commande alter
  - La commande drop

- Présentation du langage SQL
  - SQL = Structured Query Language : langage d'interrogation de bases de données relationnelles

- Présentation du langage SQL

- SQL a remplacé

SEQUEL=Structured English as Query

Language=premier langage relationnel utilisé  
par le SGBD System-R de IBM

## ■ **Les normes du langage SQL**

- SQL devient une norme ANSI en 1986 SQL/86
- SQL devient une norme ISO en 1987 SQL/86
- Une autre norme SQL/89

Expression des contraintes d'intégrité :

Primary key, check, references, default

- Une autre norme en 1992 SQL2

Expression des opérateurs ensemblistes: intersect ,  
minus...outer join, cascade, date, temps

- La norme SQL3 (1999): les types de données abstraits, notion de rôle, les concepts de l'objet
- La norme SQL4(2003)...

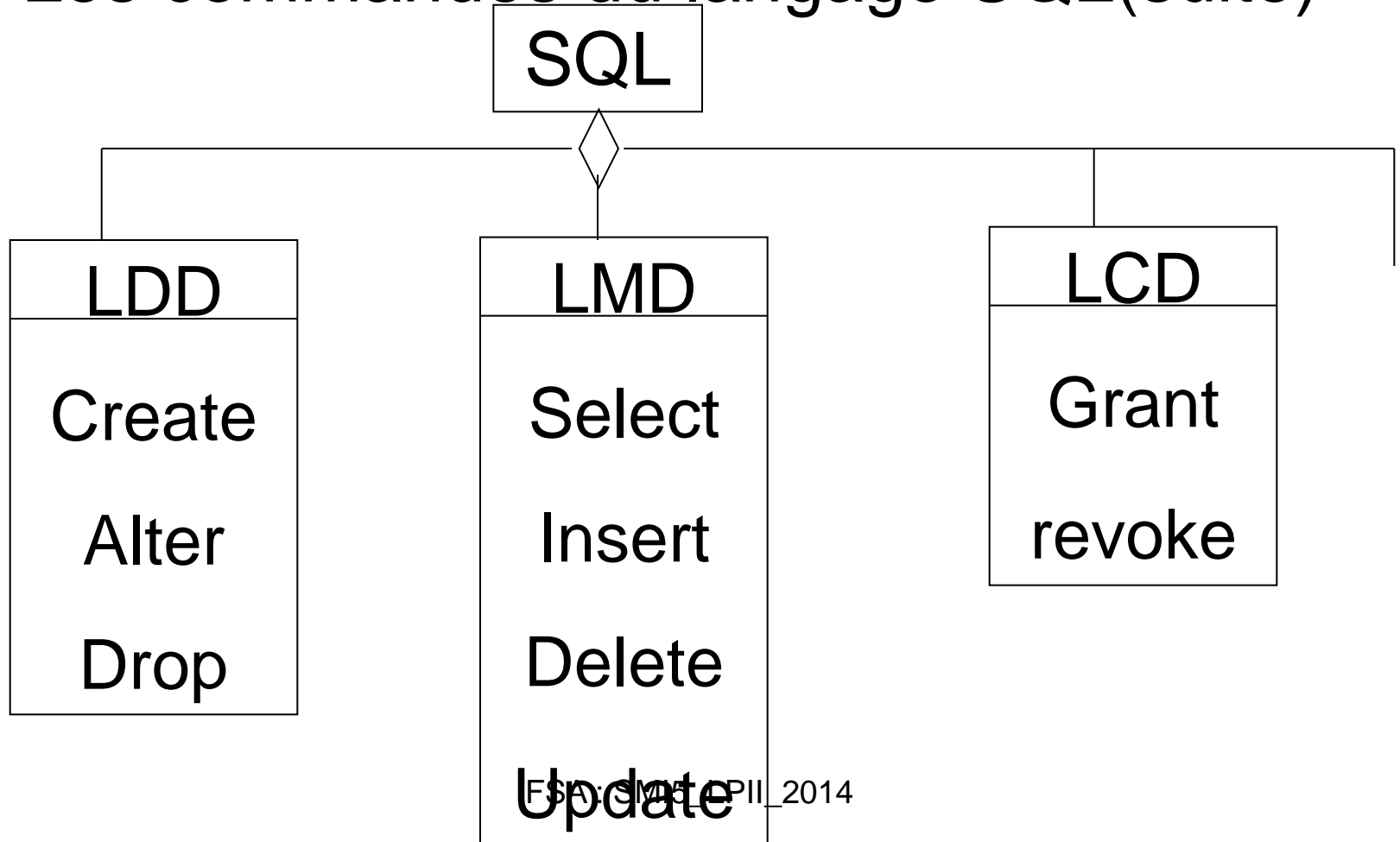
- Type du langage SQL
  - SQL est un langage :
    - de haut niveau
    - déclaratif = formulation de la requête et le SGBD utilise les moyens et les procédures adéquats pour obtenir le résultat => détails de stockage ne sont pas pris en considération.

- Les commandes du langage SQL

SQL assure entre autres :

- la création et la modification (structure) des objets de bases de données et les contraintes associées(create, alter, drop),
- l'interrogation et la maintenance(insertion suppression et modification) des bases de données(select, insert, delete, update,
- la sécurité des bases de données(contrôle des accès: grant, revoke).

- Les commandes du langage SQL(suite)





- La commande create  
=>Création des objets de base de données

Create:

- Base de données
- Table
- utilisateur
- Index
- Vues
- rôles
- Schéma

## ■ Création des tables

Table=relation

```
Create table nomTable (  
  champ1 type [default valParDéfaut] [contrainte_colonne],  
  champ2 type [default valParDéfaut] [contrainte_colonne],  
  ...  
  [Contrainte_table1,]  
  [Contrainte_table2,]  
  ...  
);
```

- **Création des tables**

Table=relation

```
Create table nomTable (  
  champ1 type,  
  champ2 type,  
  ...  
);
```

## ■ Exemple

```
Create table emp(  
  nemp number(4),  
  nom varchar2(20),  
  prénom varchar2(20),  
  genre char,  
  salaire number(10,2),  
  nserv number(2)  
);
```

- **Création des tables**

Table=relation

```
Create table nomTable (  
  champ1 type [contrainte_colonne],  
  champ2 type [contrainte_colonne],  
  ...  
);
```

## ■ Exemple

```
Create table emp(  
  nemp number(4) primary key,  
  nom varchar2(20),  
  prénom varchar2(20),  
  genre char,  
  salaire number(10,2),  
  nserv number(2)  
);
```

## ■ Exemple

```
Create table emp(  
  nemp number(4) primary key,  
  nom varchar2(20),  
  prénom varchar2(20),  
  genre char,  
  salaire number(10,2),  
  nserv number(2) references service(nserv)  
);
```

## ■ Exemple

```
Create table emp(  
  nemp number(4) primary key,  
  nom varchar2(20),  
  prénom varchar2(20),  
  genre char,  
  salaire number(10,2),  
  nserv number(2) references service  
);
```



## ■ Exemple

```
Create table emp(  
  nemp number(4) primary key,  
  nom varchar2(20),  
  prénom varchar2(20),  
  genre char,  
  salaire number(10,2),  
  nserv references service(nserv)  
);
```

## ■ Exemple

```
Create table emp(  
  nemp number(4) primary key,  
  nom varchar2(20),  
  prénom varchar2(20),  
  genre char,  
  salaire number(10,2),  
  nserv references service  
);
```

## ■ Exemple

```
Create table emp(  
  nemp number(4) primary key,  
  nom varchar2(20),  
  prénom varchar2(20),  
  genre char,  
  salaire number(10,2),  
  nserv number(2) references service(nserv) on  
  delete cascade);
```

## ■ Exemple

```
Create table emp(  
  nemp number(4) primary key,  
  nom varchar2(20),  
  prénom varchar2(20),  
  genre char,  
  salaire number(10,2),  
  nserv number(2) references service(nserv) on  
  update cascade);
```

## ■ Exemple

```
Create table emp(  
  nemp number(4) primary key,  
  nom varchar2(20),  
  prénom varchar2(20),  
  genre char,  
  salaire number(10,2),  
  nserv number(2) references service(nserv) on  
  delete set null);
```

## ■ Exemple

```
Create table emp(  
  nemp number(4) primary key,  
  nom varchar2(20),  
  prénom varchar2(20),  
  genre char,  
  salaire number(10,2),  
  nserv number(2) references service(nserv) on  
  update set null);
```

## ■ Exemple

```
Create table emp(  
  nemp number(4) primary key,  
  nom varchar2(20),  
  prénom varchar2(20),  
  genre char check(genre in('f','m','F','M')),  
  salaire number(10,2),  
  nserv number(2)  
);
```

## ■ Exemple

```
Create table emp(  
  nemp number(4) primary key,  
  nom varchar2(20),  
  prénom varchar2(20),  
  genre char check(genre in('f','m','F','M')),  
  salaire number(10,2),  
  nserv number(2) references service(nserv) on  
  delete cascade);
```



## ■ Exemple

```
Create table emp(  
  nemp number(4) primary key,  
  nom varchar2(20),  
  prénom varchar2(20),  
  genre char check(genre in('f','m','F','M'),  
  salaire number(10,2),  
  nserv number(2) references service(nserv)  
);
```

## ■ Exemple

```
Create table emp(  
  nemp number(4) primary key,  
  nom varchar2(20),  
  prénom varchar2(20),  
  genre char check(genre in('f','m','F','M')),  
  salaire number(10,2),  
  nserv number(2) references service(nserv) on  
  delete cascade);
```

## ■ Création des tables

Table=relation

```
Create table nomTable (  
  champ1 type [ default valeur ] [contrainte_colonne],  
  champ2 type [ default valeur ] [contrainte_colonne],  
  ...  
);
```

## ■ Exemple

```
Create table emp(  
  nemp number(4) primary key,  
  nom varchar2(20),  
  prénom varchar2(20),  
  genre char default 'm',  
  salaire number(10,2) default 5000,  
  nserv number(2) references service(nserv) on  
  delete cascade);
```

- Description de la syntaxe
  - nomTable et champi sont des identificateurs dont la taille  $\leq 30$  et commencent par une lettre , \_ , \$ ou #.  
La casse n'est pas respectée.
  - type représente un type prédéfini ou défini par l'utilisateur

## ■ Les types prédéfinis en Oracle

### ➤ Les types caractères

- Les constantes caractères littérales sont écrites entre apostrophes: 'ali'
- Le type `char(n)`=chaîne de caractères de longueur fixe, n est la taille maximale de la chaîne( $n \leq 2000$ )  
la valeur par défaut de n est 1.  
genre `char`=genre `char(1)`.
- Le type `varchar2(n)`=chaîne de caractères de longueur variable, n est la taille maximale de la chaîne( $n \leq 4000$ ).
- Le type `varchar(n)`=`varchar2(n)`.

## ■ Les types prédéfinis en Oracle

### ➤ Les types numériques

- **number** = nombres réels  $\geq$  et  $\leq 0$  sur 40 chiffres.  
Les valeurs absolues des nombres sont limitées entre  $1.0E-130$  et  $1.0E126$ .

EX: salaire number

**Les constantes numériques : 12, 12.3 , 12.9e-3**

- Les types prédéfinis en Oracle
  - Les types numériques
    - `number(p,d)` = les nombre sur p chiffres avec d chiffres après la virgule représentée par point(.).  
p peut aller à 38.

EX: salaire number(10,2)



- Les types prédéfinis en Oracle
  - Les types numériques
    - `number(p)` représente les entiers sur p chiffres.  
C'est équivalent à `number(p,0)`.

EX:

`nemp number(4)`

- Les types supportés(IBM, ANSI)
  - decimal équivalent à number(38).
  - decimal(n,p)
  - integer,int, smallint équivalent à number(38).
  - float(n)
  - Real

## ■ Les types prédéfinis en Oracle

### ➤ Les types date

- date

EX:

dateNaissance date

- Valeur littérale de date

'12/11/2005' ou

'12-11-2005'

## ■ Les types prédéfinis en Oracle

### ➤ Les types date

- timestamp[(fractionDeSeconde)]  
contient date et heure

- fractionDeSeconde nombre de chiffres dans la fraction de secondes ≤ 9. par défaut est 6.

EX :

moment\_début timestamp(3)

Constantes littérales:

'12/11/2005 10:30:20,100' ou

'12/11/2005 :10:30:20,100'

## ■ Exercice

Donner les commandes de création des tables suivantes:

- Filière(codeFilière, libellé,département)
- Etudiant(codeEtudiant, nom, prénom, dateNaissance, codeFilière)

- Description de la syntaxe(suite)  
ContrainteColonne=[**constraint** nomContrainte]  
TypeContrainteColonne

TypeContrainteColonne =  
primary key|  
unique |  
[not] null |  
Check (condition) |  
References nomTable[(colonne) ] [on delete  
cascade|on delete set null]  
**[enable|disable]**

## ■ Exemple

```
Create table emp(  
  nemp number(4) constraint cleEmp primary key ,  
  Nom varchar2(20) constraint uniNom unique,  
  Prénom varchar2(20),  
  Salaire number(10,2),  
  Genre char constraint valsGenre check(lower(genre)  
    in('f','m')),  
  Nserv number(2) constraint clEtrEmpServ references  
    service(nserv) on delete cascade);
```

## ■ Exemple

```
Create table emp(  
  nemp number(4) constraint cleEmp primary key disable,  
  Nom varchar2(20) constraint uniNom unique,  
  Prénom varchar2(20),  
  Salaire number(10,2),  
  Genre char constraint valsGenre check(lower(genre)  
    in('f','m')),  
  Nserv number(2) constraint clEtrEmpServ references  
    service(nserv) on delete cascade);
```



## ■ Exercice

Donner les commandes de création des tables suivantes avec nom des contraintes:

- Filière(codeFilière, libellé,département)
- Etudiant(codeEtudiant, nom, prénom, dateNaissance, codeFilière)

- Exemple

```
Create table emp(  
  nemp number(4) constraint clEmpl primary key ,  
  Nom varchar2(20),  
  Prénom varchar2(20),  
  Salaire number(10,2),  
  Genre char constraint valsGenre  
  check(lower(genre) in('f','m'))),  
  Nserv number(2) constraint clEtrEmpServ  
  references service(nserv) on delete cascade);
```

- Exemple

Create table emp(nemp number(4) primary key,  
Nom varchar2(20),  
Prénom varchar2(20),  
Genre char check(genre in('f','m','F','M'),  
Salaire number(10,2),  
Nserv references service(nserv) on delete  
cascade);

## ■ Création des tables(suite)

Table=relation

Create **table** nomTable (

champ1 type [**default** valParDéfaut] [contrainte\_colonne],

champ2 type [**default** valParDéfaut] [contrainte\_colonne],

...

[Contrainte\_table1,]

[Contrainte\_table2,]

...

);

- Description de la syntaxe(suite)

contrainteTable=[constraint nomContrainte]

typeDeContrainteTable

typeDeContrainteTable=

primary key (col1,col2...)

| Foreign key(col1,col2...) references

nomTable[(col1,col2)] [on delete cascade| on delete set null]

| Unique(col1,col2...)

| Check(condition)

[enable|disable]

## ■ Exemple

```
Create table emp(nemp number(4),  
Nom varchar2(20),  
Prénom varchar2(20),  
Salaire number(10,2),  
Genre char,  
Nserv number(2),  
primary key(nemp),  
foreign key(nserv) references service(nserv) on delete  
cascade,  
check(genre in('f','m')));
```

## ■ Exemple

Create table emp(nemp number(4),

Nom varchar2(20),

Prénom varchar2(20),

Salaire number(10,2),

Genre char,

Nserv number(2),

constraint clEmpl primary key(nemp),

Contraint clEtrEmpServ foreign key(nserv) references  
service(nserv) on delete cascade,

constraint valsGenre check(lower(genre) in('f','m')));

## ■ Exercice

Donner les commandes de création des tables suivantes avec contraintes table non nommées:

- Filière(codeFilière, libellé,département)
- Etudiant(codeEtudiant, nom, prénom, dateNaissance, codeFilière)



## ■ Exercice

Donner les commandes de création des tables suivantes avec contraintes table nommées:

- Filière(codeFilière, libellé,département)
- Etudiant(codeEtudiant, nom, prénom, dateNaissance, codeFilière)

- Création d'une table globale temporaire  
Create global temporary table nomTable(  
...)  
On commit delete rows|preserve rows

## ■ Index

Définition d'index

index = Moyen ou structure pour accélérer le temps de réponse dans une base de données.

## ■ Inconvénients d'index

- Espace mémoire suppl=lien de stockage des valeurs de la colonne indexée et les adresses des lignes associées
- Ralentissement de MAJ

- Création d'index

Create index

nomIndex on nomTable( col1 [asc|desc] )

- Exemple

Create index idxnom on emp(nom asc)

- Création d'index

Create unique index

nomIndex on nomTable( col1 [asc|desc] )

- Exemple

Create unique index idxnom on emp(nom asc)



- Création d'index

Create index

```
nomIndex on nomTable( col1 [asc|desc]  
[, col2 [asc|desc]...])
```

- Exemple

Create index idxnomprénom on emp(nom asc,  
prénom desc)

- Création d'index

Create [unique] index

nomIndex on nomTable( expression [asc|desc] )

- Exemple

Create index idxnom on emp(lower(nom) asc)

- Création d'index (syntaxe générale)

Create [unique] index

nomIndex on nomTable( col1|expression1 [asc|desc]  
[, col2|expression2 [asc|desc]...])

- Note.

Pour voir les objets créés, vous pouvez consulter la vue: `user_objects` (`object_name`, `object_Type...`)

Pour voir les autres champs de cette vue, utiliser `desc user_objects`.

Vous pouvez aussi utiliser les vues `user_tables`, `user_indexes`, `user_constraints...`

- Création des vues

## Syntaxe

Create [or replace] [force|no force] view  
nomVue[(listeDeChamps)]

As

requêteSélection

With check option [constraint nomContrainte] | [with  
read only];

## Exemple

Create view test

as

select \* from emp where sal<10000;

- Création de schéma
- Create schema authorization  
nom\_schema
  - CréationDesTables
  - CréationDesVues
  - OctroiDePrivilèges



- Création de BD

Create database nomBase

Datafile...

Logfile...

.

.

.

- La commande alter

Alter=commande de modification des  
**structures** des objets de BD

- Modification de table

Alter table nomTable

Rename to nouveauNomDeTable

- Exercice  
renommer le nom de la table emp par  
employe

- Modification de table

Alter table nomTable

Rename column nomColonne to  
nouveauNomColonne

- Exercice

renommer la colonne nemp la table emp  
par codeEmp

- Modification de table

Alter table nomTable

Add champ type [**default** valParDéfaut]  
[contrainte\_colonne]

- Exercice

Ajouter le champ email à la table étudiant.



- Exercice

Ajouter les champs email et ville à la table étudiant.

- Modification de table

Alter table nomTable

Add [constraint nomContrainte]  
typeDeContrainteTable

- Exercice

Ajouter la contrainte (non nommée) de clé primaire à la table employé: prendre le champ codeEmp.

- Exercice

Ajouter la contrainte (nommée) de clé primaire à la table employé: prendre le champ codeEmp.

- Exercice

Ajouter la contrainte (non nommée) de clé étrangère à la table employé: prendre le champ codeService avec la table service.

- Exercice

Ajouter la contrainte (nommée) de clé étrangère à la table employé: prendre le champ codeService avec la table service.

- Modification de table

Alter table nomTable

Add(champ1 type...|contrainteTable1,  
champ2 type...|contrainteTable2...)

- Modification de table  
Alter table nomTable  
Drop column champ



- Exercice

Supprimer le champ email de la table étudiant.

- Modification de table  
Alter table nomTable  
Drop column champ [cascade constraints]

- Exercice

Supprimer le champ codefilière de la table filière et les contraintes associées.

- Modification de table  
Alter table nomTable  
Drop (champ1, champ2, ...) [cascade  
constraints]

- Modification de table  
Alter table nomTable  
Drop constraint nomContrainte [cascade]

- Exercice

Supprimer la contrainte cleEmp de la table emp.

- Modification de table  
Alter table nomTable  
Drop primary key [cascade]

- Exercice

Supprimer la contrainte de clé primaire de la table étudiant.



- Modification de table

Alter table nomTable

Drop unique(champ1[,champ2...]) [cascade]

- Exercice

Supprimer la contrainte d'unicité du champ nom de la table étudiant.

- Modification de table

Alter table nomTable

Modify champ [type] [default  
valeurParDefaut] [contrainteColonne]

Modify constraint nomContrainte  
enable|disable [exceptions into  
nomTableExceptions] [cascade]

Modify (champ1...|constraint  
nomContrainte1...,...)

**Enable**  
**Disable**

**Unique(listeChamps)**  
**Primary key**  
**Constraint nomContrainte**

**[Exceptions into nomTableExceptions]**  
**[cascade]**

- Exemples

- Alter table emp rename to empl;
- Alter table emp disable primary key;
- Alter table emp modify constraint clé enable exceptions into exc;

Le schéma de exc est

exc(ligne, propriétaire, table, contrainte)

- Alter table emp add email varchar2(50);
- Alter table emp drop column email;
- Alter table service disable primary key cascade;

- Modification d'un index

Alter index nomIndex

Enable

Disable

Rename to nouveauNomIndex

unusable

- Examples
  - Alter index ind rename to ind1
  - Alter index ind1 disable

- La commande drop

## Suppression des objets de BD

- Base de données
- Table
- Index
- Vues
- rôles
- Schéma



- Suppression d'une table

Drop table nomTable [cascade Constraints];

- Suppression d'un index  
Drop index nomIndex;

- Suppression d'une vue  
Drop view nomVue;

- Suppression d'une BD
- Drop database nomBase;