



COURS 1

Système de gestion de base de données, CREATE, INSERT, DISTINCT, ORDER BY

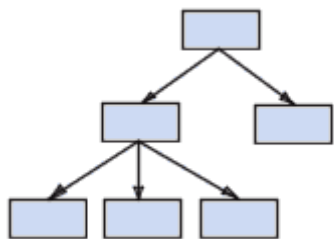
BINV1030-1

Gestion des données : bases

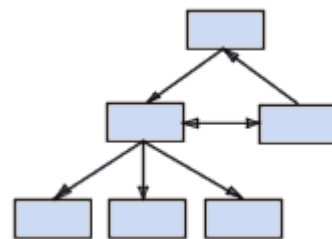


Bases de données

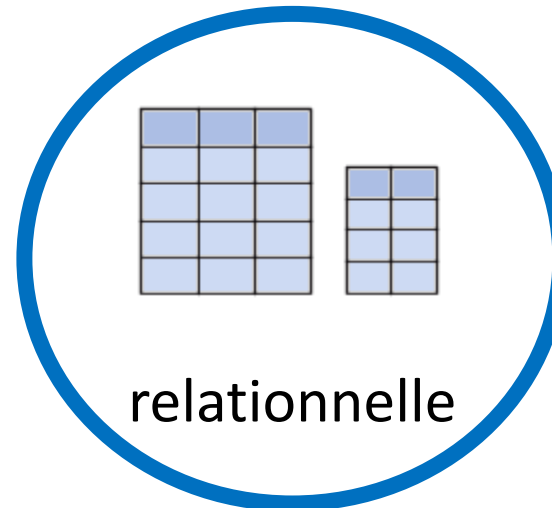
- Gérer des grandes quantités d'information
 - Clients, fournisseurs, ...
 - Commandes, livraisons, facturations, ...
 - Listes d'articles, catalogues de produits, ...
- Il existe différents types de bases de données :



hiérarchique



en réseau



relationnelle

NoSQL



La gestion des bases de données

- **SGBD** : Système de Gestion de Bases de Données
- En anglais DBMS : DataBase Management System
- La gestion se fait via le langage **SQL** : Structured Query Language
 - Permet l'accès aux données de façon souple
 - Autorise un accès aux informations à plusieurs utilisateurs
 - Permet de manipuler les données présentes dans la base de données : insertion, suppression, modification



Bases de données relationnelles (p. 1)

- Une BD relationnelle est composée de tables
 - Colonnes : **attributs** ou **champs**
 - Lignes : **tuples**, **enregistrements** ou **records** (en anglais)
- Exemple : une table contenant les clients d'une entreprise
 - 8 attributs : noms des colonnes
 - 3 enregistrements : clients

clients
num
nom
prenom
adresse
cp
ville
tel
numtva

num	nom	prenom	adresse	cp	ville	tel	numtva
1	Dupuis	Benoit	clos Chapelle aux Champs, 43	1200	Bruxelles	02/764.46.46	BE413770425
3	Dupuis	Luc	rue Lambert Fortune, 2	1300	Wavre	010/34.65.67	
18	Van Moer	Willy	rue Sainte Anne, 6	1300	Wavre		



Création de tables : CREATE (p. 3)

```
CREATE TABLE clients
```

```
(num          INT          NOT NULL,  
 nom          VARCHAR(25)  NOT NULL,  
 prenom       VARCHAR(15)  NOT NULL,  
 adresse      VARCHAR(30)  NOT NULL,  
 cp           INT          NOT NULL,  
 ville        VARCHAR(20)  NOT NULL,  
 tel          VARCHAR(11)   NULL,  
 numtva       VARCHAR(10)  NULL)
```

clients
num NN
nom NN
prenom NN
adresse NN
cp NN
ville NN
tel
numtva



Création de tables : types de données (p. 5)

clients
num NN
nom NN
prenom NN
adresse NN
cp NN
ville NN
tel
numtva

- Comme en Java, il existe différents types de données en SQL

```
CREATE TABLE clients
(num          INT          NOT NULL,
 nom         VARCHAR(25)  NOT NULL,
 prenom      VARCHAR(15)  NOT NULL,
 adresse     VARCHAR(30)  NOT NULL,
 cp          INT          NOT NULL,
 ville       VARCHAR(20)  NOT NULL,
 tel         VARCHAR(11)  NULL,
 numtva      VARCHAR(10)  NULL)
```

- Voici les types existant en PostgreSQL :



PostgreSQL

bigint, bit, bit varying, boolean, char, character varying (varchar), character (char), date, double precision, integer (int), interval, numeric, decimal, real, smallint, time (avec et sans fuseau horaire), timestamp (avec et sans fuseau horaire), xml.



Création de tables : NULL ou NOT NULL

clients	
num	NN
nom	NN
prenom	NN
adresse	NN
cp	NN
ville	NN
tel	
numtva	

- Quelle est la différence entre une valeur qui est NULL ou NOT NULL ?

CREATE TABLE

(num
nom
prenom
adresse
cp
ville
tel
numtva

clients	
num	NN
nom	NN
prenom	NN
adresse	NN
cp	NN
ville	NN
tel	
numtva	

obligatoires

NOT NULL,
NOT NULL,
NOT NULL,
NOT NULL,
NOT NULL,
NOT NULL,
NULL,
NULL)



NULL n'est pas
zéro ni une
chaine vide !!!

- NULL → la valeur peut être omise (= laissée vide)
- NOT NULL → la valeur doit être spécifiée obligatoirement



Création de tables : conventions (p. 4)

clients	
num	NN
nom	NN
prenom	NN
adresse	NN
cp	NN
ville	NN
tel	
numtva	

■ Quelles sont les conventions SQL ?

```
CREATE TABLE clients
(num          INT          NOT NULL,
nom          VARCHAR(25)   NOT NULL,
prenom       VARCHAR(15)   NOT NULL,
adresse      VARCHAR(30)   NOT NULL,
cp           INT          NOT NULL,
ville        VARCHAR(20)   NOT NULL,
tel          VARCHAR(11)   NULL,
numtva       VARCHAR(10)   NULL)
```

ceci_est_un_nom_correct

Mais ça p@s du tout !

- Mots réservés de l'instruction SQL : en MAJUSCULES
- Tous les autres mots en minuscules
- Pas de caractères spéciaux, ni accentués, ni espaces, ...



Clé primaire (p. 5)

- La clé primaire ou Primary Key (PK) est un identifiant unique pour un enregistrement

num	nom	prenom	adresse	cp	ville	tel	numtva
1	Dupuis	Benoit	clos Chapelle aux Champs, 43	1200	Bruxelles	02/764.46.46	BE413770425
3	Dupuis	Luc	rue Lambert Fortune, 2	1300	Wavre	010/34.65.67	
18	Van Moer	Willy	rue Sainte Anne, 6	1300	Wavre		
3	Dupuis	Luc	Rue L. Fortune, 7	1000	Bruxelles	02/7644654	

```
CREATE TABLE clients
(num          INT          PRIMARY KEY,
 nom          VARCHAR(25)  NOT NULL,
 prenom       VARCHAR(15)  NOT NULL,
 adresse      VARCHAR(30)  NOT NULL,
 cp           INT          NOT NULL,
 ville        VARCHAR(20)  NOT NULL,
 tel          VARCHAR(11)
 numtva       VARCHAR(10)
)
```

clients	
PK	<u>num</u>
	nom NN
	prenom NN
	adresse NN
	cp NN
	ville NN
	tel
	numtva



Insertion de données : INSERT (p. 6)

- L'insertion de données se fait au moyen du mot clé INSERT

```
INSERT INTO nom_table  
[ (nom_colonne {, nom_colonne}) ]  
VALUES (valeur {, valeur})
```

- Conventions :
 - Mot en MAJUSCULES = clause ou élément de l'instruction (mot réservé)
 - [] → élément facultatif
 - {...} → élément facultatif qui peut être répété
 - | → désigne une alternative (OU exclusif)



```
INSERT INTO nom_table  
[(nom_colonne {, nom_colonne})]  
VALUES (valeur {, valeur})
```

Insertion de données : exemples (p. 7)

num	nom	prenom	adresse	cp	ville	tel	numtva
-----	-----	--------	---------	----	-------	-----	--------

```
INSERT INTO clients (num, nom, prenom, adresse, cp, ville)  
VALUES (18, 'Van Moer', 'Willy', 'rue Sainte Anne, 6', 1300, 'Wavre')
```

```
INSERT INTO clients (nom, prenom, numtva, num, tel, adresse, cp, ville)  
VALUES ('Dupuis', 'Luc', NULL, 3, '010/34.65.67', 'rue Lambert Fortune,  
2', 1300, 'Wavre')
```

```
INSERT INTO clients  
VALUES (1, 'Dupuis', 'Benoit', 'clos Chapelle aux Champs, 43', 1200,  
'Bruxelles', '02/764.46.46', NULL)
```

num	nom	prenom	adresse	cp	ville	tel	numtva
18	Van Moer	Willy	Rue Sainte Anne, 6	1300	Wavre		
3	Dupuis	Luc	Rue Lambert Fortune, 2	1300	Wavre	010/34.65.67	
1	Dupuis	Benoit	Clos Chapelle aux Champs, 43	1200	Bruxelles	0499123456	



Consultation de données : SELECT (p. 7)

- La consultation de données se fait au moyen du mot clé SELECT

```
SELECT * | nom_colonne {, nom_colonne...}  
FROM nom_table {, nom_table ...}  
WHERE nom_condition
```

- Conventions :
 - Mot en MAJUSCULES = clause ou élément de l'instruction (mot réservé)
 - [] → élément facultatif
 - {...} → élément facultatif qui peut être répété
 - | → désigne une alternative (OU exclusif)



```
SELECT * | nom_colonne {, nom_colonne...}  
FROM nom_table {, nom_table ...}  
WHERE nom_condition
```

Consultation de données : exemple (p. 8)

```
SELECT *  
FROM clients
```

num	nom	prenom	adresse	cp	ville	tel	numtva
1	Dupuis	Benoit	clos Chapelle aux Champs, 43	1200	Bruxelles	02/764.46.46	BE413770425
3	Dupuis	Luc	rue Lambert Fortune, 2	1300	Wavre	010/34.65.67	
18	Van Moer	Willy	rue Sainte Anne, 6	1300	Wavre		



Ce résultat est un ensemble. L'ordre aurait donc pu être différent !



```
SELECT * | nom_colonne {, nom_colonne...}  
FROM nom_table {, nom_table ...}  
WHERE nom_condition
```

Consultation de données : exemples (p. 8)

```
SELECT num, nom, prenom, tel, cp  
FROM clients
```

num	nom	prenom	tel	cp
1	Dupuis	Benoit	02/764.46.46	1200
3	Dupuis	Luc	010/34.65.67	1300
18	Van Moer	Willy		1300

```
SELECT cp FROM clients
```

cp
1200
1300
1300



DISTINCT (p. 8)

- Le mot clé DISTINCT permet d'éviter qu'un résultat se répète

```
SELECT cp  
FROM clients
```



cp
1200
1300
1300

```
SELECT DISTINCT cp  
FROM clients
```



cp
1200
1300

- DISTINCT porte sur toutes les colonnes du SELECT

```
SELECT DISTINCT nom, prenom  
FROM clients
```



nom	prenom
Dupuis	Benoit
Dupuis	Luc
Van Moer	Willy



Conditions simples (p. 9)

```
WHERE nom_colonne opérateur value
```

- Opérateurs : = | < | > | <> | != | IS | IS NOT

```
SELECT * FROM clients
```

- WHERE cp = 1300
- WHERE ville <> 'Wavre'
- WHERE numtva **IS** NULL
- WHERE tel **IS NOT** NULL



Conditions générales (p. 9)

- Opérateurs : **AND** | **OR** | **NOT**
- Afficher tous les clients qui habitent Bruxelles ou qui se prénomment Benoit :

```
SELECT *
```

```
FROM clients
```

```
WHERE ville = 'Bruxelles' OR prenom = 'Benoit'
```

- **OR** est un OU non-exclusif !



Priorité des opérateurs (p. 9)

- Comment se lisent les instructions suivantes ?

C1 OR C2 AND C3

➤ (C1 OR C2) AND C3

➤ C1 OR (C2 AND C3)

NOT C1 AND C2

➤ NOT (C1 AND C2)

➤ (NOT C1) AND C2



NOT



AND



OR



Exercice

- Quels sont les clients qui habitent Wavre ou Waterloo et dont le prénom est Luc ?
Donnez leurs num, nom et prénom.

clients	
PK	<u>num</u>
	nom NN prenom NN adresse NN cp NN ville NN tel numtva



IS (NOT) NULL (p. 10)



NULL ne peut jamais vérifier une égalité !! Ni une inégalité

- On utilisera les mots clés **IS** et **IS NOT**
- Lister tous les clients qui n'ont pas de numéro de TVA :

```
SELECT * FROM clients
```

```
WHERE numtva IS NULL
```

num	nom	prenom	adresse	cp	ville	tel	numtva
3	Dupuis	Luc	Rue Lambert Fortune, 2	1300	Wavre	010/34.65.67	
18	Van Moer	Willy	Rue Sainte Anne, 6	1300	Wavre		



IS (NOT) NULL



NULL ne peut jamais vérifier une égalité !! Ni une inégalité

```
SELECT * FROM clients  
WHERE numtva > 'BE000000'
```

Sous-entend :

```
AND numtva IS NOT NULL
```

→ En comparant `numtva` avec une valeur, tous les enregistrements pour lesquels `numtva` n'a pas de valeur sont exclus du résultat



IS (NOT) NULL

- Quels sont les albums dont le dessinateur est différent du coloriste ?

```
SELECT *
```

```
FROM bd1.albums
```

```
WHERE coloriste <> dessinateur
```

- Il y aura automatiquement 1 coloriste et 1 dessinateur dans chaque ligne du résultat !
- On demande un dessinateur et un coloriste puisqu'ils doivent être différents
- **Ils ne peuvent donc pas être NULL** puisqu'on va comparer leurs valeurs



Exercices

- Quels sont tous les clients qui habitent Wavre et qui n'ont pas de numéro de TVA ?
Donnez leurs num, nom et prénom.

clients	
PK	<u>num</u>
	nom NN prenom NN adresse NN cp NN ville NN tel numtva



Clause de tri : ORDER BY (p. 13)

- ORDER BY définit l'ordre de présentation des résultats de l'instruction SELECT

```
SELECT [DISTINCT] * | [nom_table.] nom_colonne  
        {, [nom_table.] nom_colonne...}
```

```
FROM    nom_table {, nom_table...}
```

```
WHERE    nom_condition
```

```
ORDER BY nom_colonne {, nom_colonne...}
```

- L'attribut (ou la colonne) utilisé pour trier est la **clé de tri**



Clause de tri : exemples

► `SELECT * FROM clients ORDER BY nom ASC`

num	nom	prenom	adresse	cp	ville	tel	numtva
1	Dupuis	Benoit	Clos Chapelle Champs, 43	1200	Bruxelles	02/764.46.46	BE413770425
3	Dupuis	Luc	Rue Lambert Fortune, 2	1300	Wavre	010/34.65.67	
18	Van Moer	Willy	Rue Sainte Anne, 6	1300	Wavre		

► `SELECT * FROM clients ORDER BY prenom DESC`

num	nom	prenom	adresse	cp	ville	tel	numtva
18	Van Moer	Willy	Rue Sainte Anne, 6	1300	Wavre		
3	Dupuis	Luc	Rue Lambert Fortune, 2	1300	Wavre	010/34.65.67	
1	Dupuis	Benoit	Clos Chapelle Champs, 43	1200	Bruxelles	02/764.46.46	BE413770425



Clause de tri : exemples

- Lister tous les clients par ordre alphabétique des noms et anti-alphabétique des prénoms ;

```
SELECT * FROM clients
```

```
ORDER BY nom, prenom DESC
```

num	nom	prenom	adresse	cp	ville	tel	numtva
3	Dupuis	Luc	Rue Lambert Fortune, 2	1300	Wavre	010/34.65.67	
1	Dupuis	Benoit	Clos Chapelle Champs, 43	1200	Bruxelles	02/764.46.46	BE413770425
18	Van Moer	Willy	Rue Sainte Anne, 6	1300	Wavre		



Exercices

- Quels sont les différents noms de clients présents dans la base de données ? Triez-les par ordre alphabétique.

clients	
PK	<u>num</u>
	nom NN prenom NN adresse NN cp NN ville NN tel numtva



Exercices

- A quelles adresses avons-nous des clients dont le code postal est 1200 ? Donnez leur adresse et leur code postal.

clients	
PK	<u>num</u>
	nom NN prenom NN adresse NN cp NN ville NN tel numtva



Exercices

- Listez tous les noms des clients dont le numéro de tva est supérieur à 'BE000000' et n'est pas NULL

clients	
PK	<u>num</u>
	nom NN prenom NN adresse NN cp NN ville NN tel numtva



Conclusion

- Étudier toute la matière jusqu'à la page 11 comprise (avant `LIKE`) + la page 14 (`ORDER BY`)
- Semaine 1 :
 - Prise en main de DataGrip
 - Exercices de `CREATE TABLE`
 - Exercices de `SELECT`
- Semaine 3 : interro de SQL !