université PARIS-SACLAY | IUT DE SCEAUX

Traitement numérique des données

SQL-QBE-Access

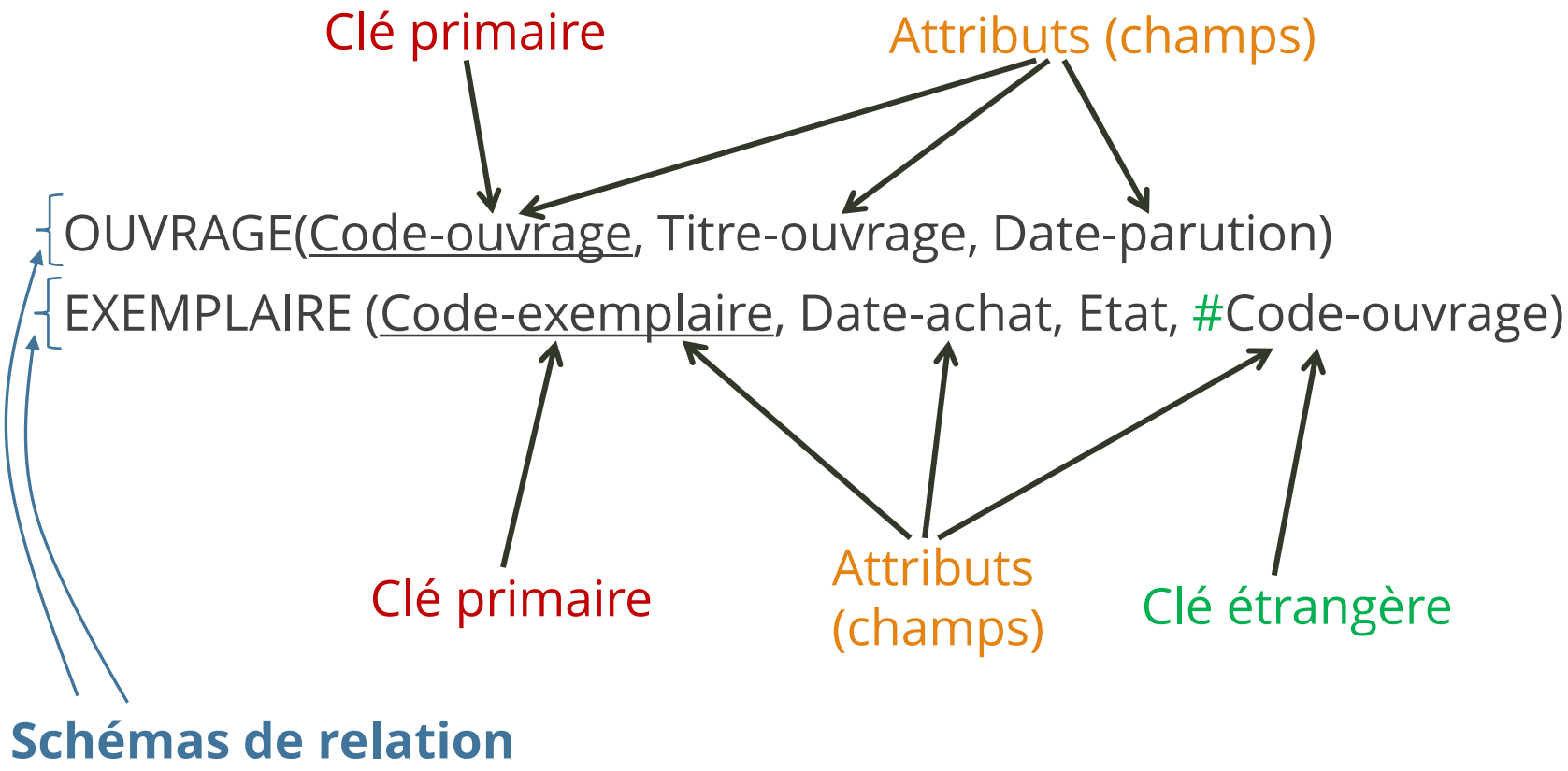
R5.04

PARTIE « INFO »

Evangelos BAMPAS

evangelos.bampas@universite-paris-saclay.fr

Modèle relationnel : rappels



Relations : (Tables de données)

OUVRAGE

Code-ouvrage	Titre-ouvrage	Date-parution
46	The catcher in the rye	1951
81	Millénium	2005
24	La vie en rosalie	2022
59	The Da Vinci code	2003

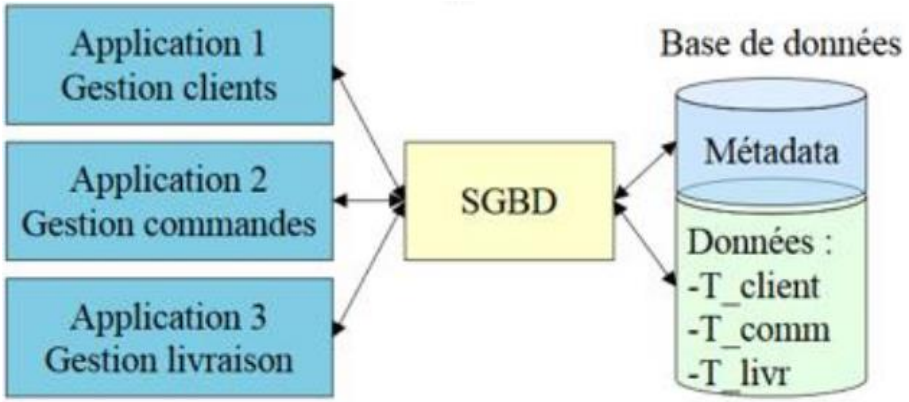
EXEMPLAIRE

Code-exemplaire	Date-achat	Etat	Code-ouvrage
22	22/08/2010	Bon	46
23	07/03/2011	Très bon	46
24	12/12/2005	Bon	81
25	31/10/2022	Neuf	24
26	31/10/2022	Neuf	24

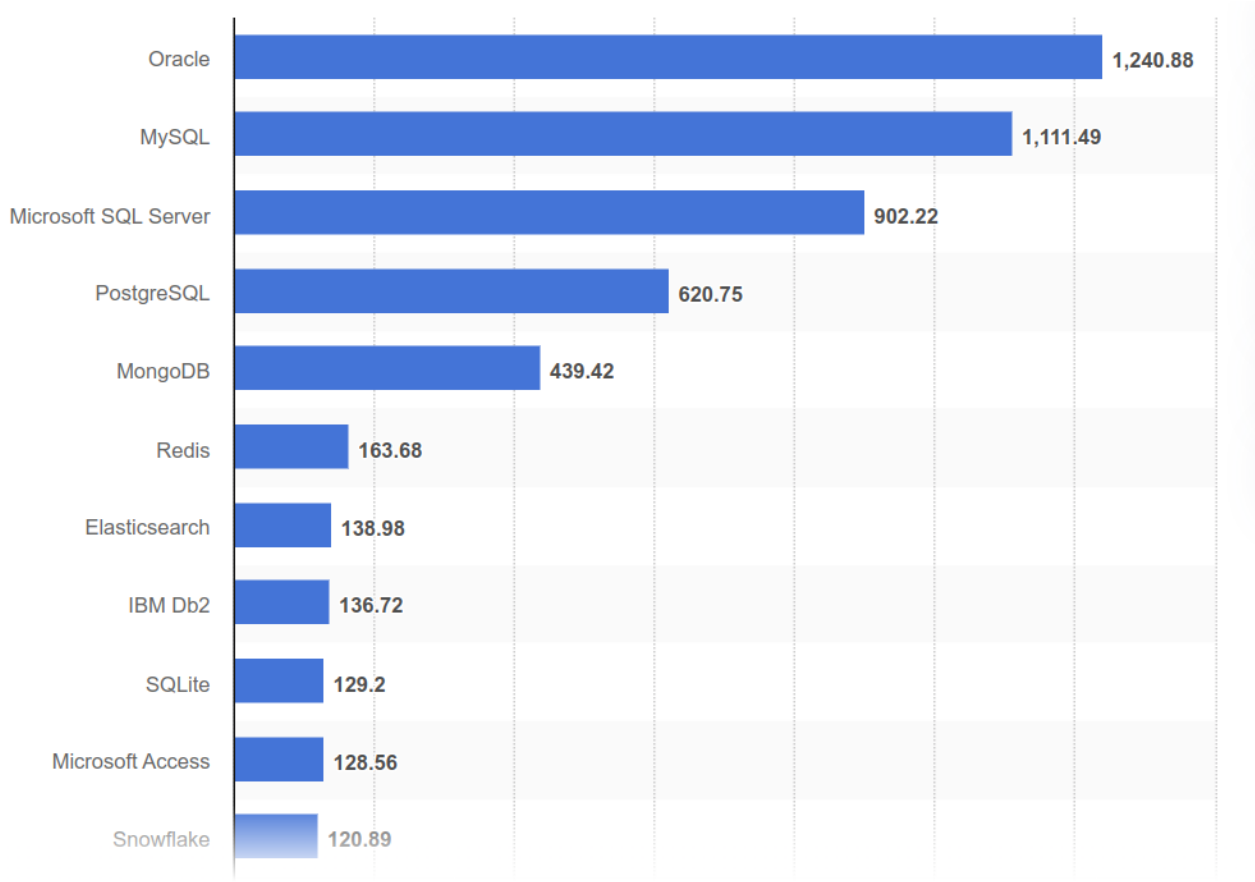
Système de Gestion de Bases de Données (SGBD)

- **SGBD** : Un **logiciel système** permettant de stocker, manipuler, gérer, et partager les données dans une base de données.

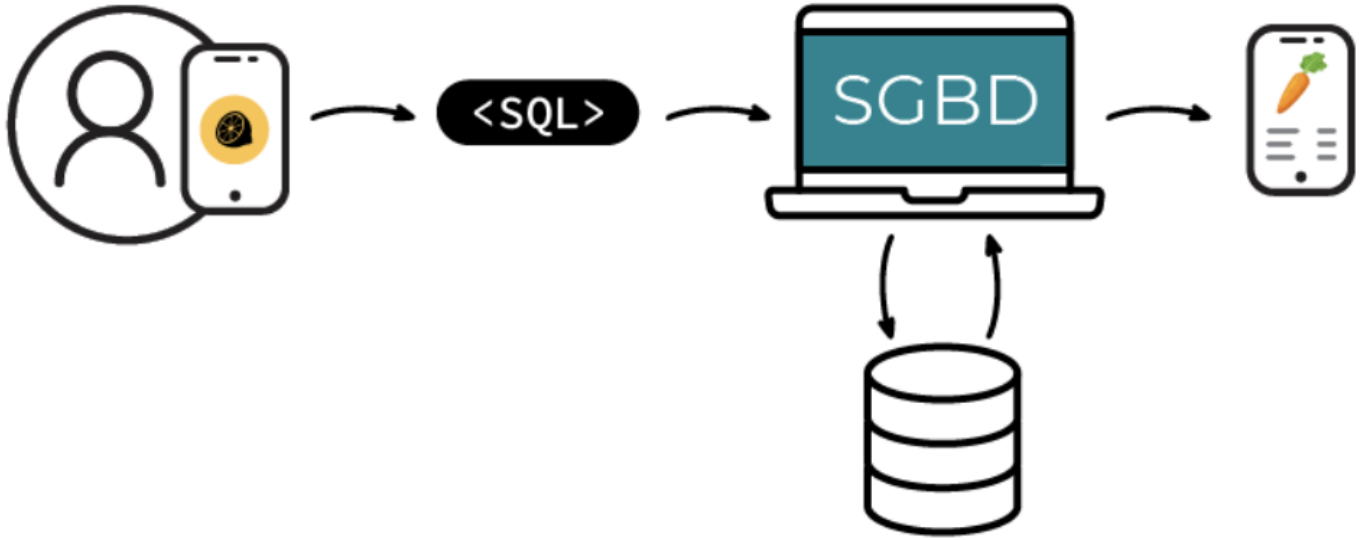
- Aspects principaux :
  - Indépendance physique et logique des données
  - Manipulation des données à l'aide d'un langage informatique
  - Efficacité d'accès
  - Administration centralisée
  - Cohérence
  - Partageabilité



SGBD du marché (septembre 2023)



Interaction avec un SGBD



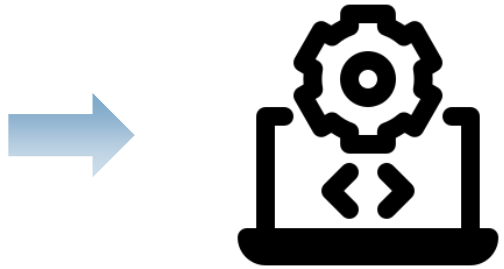
SQL

SQL : Structured Query Language

- « Langage de requêtes structurées »
- Langage normalisé : supporté par de nombreux SGBD
- Interface entre les applications et les SGBD relationnels
- Les requêtes sont fournies au SGBD sous forme de texte
  - Suivant une spécification précise (langage informatique)
  - Langage **déclaratif** : décrire **Quoi** sans dire **Comment**

```
SELECT Code-exemplaire, Date-achat, Code-ouvrage
FROM EXEMPLAIRE
WHERE Code-ouvrage=46
```

Requête SQL



Moteur SGBD

Table résultat

Code-exemplaire	Date-achat	Code-ouvrage
22	22/08/2010	46
23	07/03/2011	46

Modèle de données pour les exemples

- Pour tous les exemples on va supposer le modèle de données suivant :

  CLIENT(idCli, nom, ville)

  PRODUIT(idPro, description, marque, prix)

  VENTE(#idCli, #idPro, date, qte)

CLIENT

idCli	nom	ville
1	Alice Dupont	Paris
2	Bob Martin	Marseille
3	Carol Smith	Lyon
4	David Johnson	Toulouse
5	Émilie Davis	Paris
6	François Leclerc	Bordeaux
7	Gabrielle Dubois	Paris
8	Henri Lefebvre	Nantes
9	Isabelle Moreau	Strasbourg
10	Jean Dupuis	Montpellier

PRODUIT

idPro	description	marque	prix
1	Téléviseur Smart 4K	LG	499.99
2	Ordinateur Portable	Acer	799.99
3	Smartphone Haut de Gamme	Samsung	899.99
4	Machine à Café	Philips	89.99
5	Appareil Photo Numérique	Canon	349.99
6	Console de Jeu	Sony	399.99
7	Réfrigérateur	Bosch	549.99
8	Aspirateur Robot	Bosch	299.99
9	Tablette Numérique	Samsung	299.99
10	Enceinte Bluetooth	JBL	69.99

VENTE

idCli	idPro	date	qte
1	1	01/10/2022	5
1	3	01/10/2023	3
5	1	03/11/2023	2
6	4	04/11/2021	4
4	5	05/11/2021	6
5	9	06/11/2021	1
6	7	07/11/2021	8
2	6	01/11/2023	1
2	7	01/11/2023	1
2	8	11/11/2023	1

Projections
(afficher certaines colonnes d'une table)

- Exemple 1 : Donner les noms, marques et prix des produits

```
SELECT P.description, P.marque, P.prix
FROM   PRODUIT P
```

PRODUIT P

idPro	description	marque	prix
1	Téléviseur Smart 4K	LG	499.99
2	Ordinateur Portable	Acer	799.99
3	Smartphone Haut de Gamme	Samsung	899.99
4	Machine à Café	Philips	89.99
5	Appareil Photo Numérique	Canon	349.99
6	Console de Jeu	Sony	399.99
7	Réfrigérateur	Bosch	549.99
8	Aspirateur Robot	Bosch	299.99
9	Tablette Numérique	Samsung	299.99
10	Enceinte Bluetooth	JBL	69.99

Résultat

description	marque	prix
Téléviseur Smart 4K	LG	499.99
Ordinateur Portable	Acer	799.99
Smartphone Haut de Gamme	Samsung	899.99
Machine à Café	Philips	89.99
Appareil Photo Numérique	Canon	349.99
Console de Jeu	Sony	399.99
Réfrigérateur	Bosch	549.99
Aspirateur Robot	Bosch	299.99
Tablette Numérique	Samsung	299.99
Enceinte Bluetooth	JBL	69.99

Synonyme de table

- Exemple 1 : Donner les noms, marques et prix des produits

```
SELECT P.description, P.marque, P.prix
FROM   PRODUIT P
```

- Dans l'exemple, nous avons introduit dans la clause FROM un **synonyme** du nom de table (P).
  - Les noms de table ou les synonymes peuvent être utilisés dans SELECT pour préfixer les noms de colonnes.
  - Les préfixes ne sont obligatoires que dans des cas particuliers ; leur emploi est cependant conseillé par souci de clarté.
  - Dans ce cours, nous allons toujours utiliser les synonymes.

Doublons

- Le résultat d'une requête peut contenir des doublons (lignes identiques).
- Pour éliminer les doublons, on utilise le mot clé DISTINCT.
- Exemple 2 : Donner les différentes marques de produit

```
SELECT DISTINCT P.marque
FROM   PRODUIT P
```

PRODUIT P

idPro	description	marque	prix
1	Téléviseur Smart 4K	LG	499.99
2	Ordinateur Portable	Acer	799.99
3	Smartphone Haut de Gamme	Samsung	899.99
4	Machine à Café	Philips	89.99
5	Appareil Photo Numérique	Canon	349.99
6	Console de Jeu	Sony	399.99
7	Réfrigérateur	Bosch	549.99
8	Aspirateur Robot	Bosch	299.99
9	Tablette Numérique	Samsung	299.99
10	Enceinte Bluetooth	JBL	69.99

Projection

marque
LG
Acer
Samsung
Philips
Canon
Sony
Bosch
~~Bosch~~
~~Samsung~~
JBL

Résultat

marque
LG
Acer
Samsung
Philips
Canon
Sony
Bosch
JBL

Projections avec colonnes calculées

- Exemple 3 : Donner les références des produits et leurs prix majorés de 20 %

```
SELECT P.idPro, P.prix*1.20
FROM   PRODUIT P
```

PRODUIT P

idPro	description	marque	prix	P.prix*1.20
1	Téléviseur Sma	LG	499.99	599.99
2	Ordinateur Por	Acer	799.99	959.99
3	Smartphone Ha	Samsung	899.99	1079.99
4	Machine à Cafe	Philips	89.99	107.99
5	Appareil Photo	Canon	349.99	419.99
6	Console de Jeu	Sony	399.99	479.99
7	Réfrigérateur	Bosch	549.99	659.99
8	Aspirateur Rob	Bosch	299.99	359.99
9	Tablette Numé	Samsung	299.99	359.99
10	Enceinte Bluetc	JBL	69.99	83.99

Résultat

idPro	P.prix*1.20
1	599.99
2	959.99
3	1079.99
4	107.99
5	419.99
6	479.99
7	659.99
8	359.99
9	359.99
10	83.99

Il est possible d'effectuer des opérations arithmétiques simples (+, -, *, /) sur les colonnes extraites.

Renommer une colonne avec AS

- Il est possible de renommer une colonne en ajoutant AS nouveau_nom dans la clause SELECT.
- Exemple 3bis : Donner les références des produits et leurs prix majorés de 20 %

```
SELECT P.idPro, P.prix*1.20 AS 'Prix TTC'
FROM   PRODUIT P
```

PRODUIT P

idPro	description	marque	prix	Prix TTC
1	Téléviseur Sma	LG	499.99	599.99
2	Ordinateur Por	Acer	799.99	959.99
3	Smartphone Ha	Samsung	899.99	1079.99
4	Machine à Cafe	Philips	89.99	107.99
5	Appareil Photo	Canon	349.99	419.99
6	Console de Jeu	Sony	399.99	479.99
7	Réfrigérateur	Bosch	549.99	659.99
8	Aspirateur Rob	Bosch	299.99	359.99
9	Tablette Numéi	Samsung	299.99	359.99
10	Enceinte Bluetc	JBL	69.99	83.99

Résultat

idPro	Prix TTC
1	599.99
2	959.99
3	1079.99
4	107.99
5	419.99
6	479.99
7	659.99
8	359.99
9	359.99
10	83.99

Renvoyer toutes les colonnes d'une table

- Un astérisque (*) dans SELECT permet de lister tous les attributs.

- Exemple 4 : Donner tous les renseignements sur les clients

```
SELECT *
FROM   CLIENT
```

Résultat

idCli	nom	ville
1	Alice Dupont	Paris
2	Bob Martin	Marseille
3	Carol Smith	Lyon
4	David Johnson	Toulouse
5	Émilie Davis	Paris
6	François Lecle	Bordeaux
7	Gabrielle Dubo	Paris
8	Henri Lefebvre	Nantes
9	Isabelle Morea	Strasbourg
10	Jean Dupuis	Montpellier

Restrictions
(afficher certaines lignes d'une table)

- Exemple 5 : Donner les descriptions des produits de marque Bosch

```
SELECT P.description
FROM   PRODUIT P
WHERE  P.marque='Bosch'
```

La clause WHERE spécifie une **condition de recherche** (restriction de ligne).

PRODUIT P

idPro	description	marque	prix
1	Téléviseur Smart 4K	LG	499.99
2	Ordinateur Portable	Acer	799.99
3	Smartphone Haut de Gamme	Samsung	899.99
4	Machine à Café	Philips	89.99
5	Appareil Photo Numérique	Canon	349.99
6	Console de Jeu	Sony	399.99
7	Réfrigérateur	Bosch	549.99
8	Aspirateur Robot	Bosch	299.99
9	Tablette Numérique	Samsung	299.99
10	Enceinte Bluetooth	JBL	69.99

Restriction

idPro	description	marque	prix
7	Réfrigérateur	Bosch	549.99
8	Aspirateur Robot	Bosch	299.99

Résultat

description
Réfrigérateur
Aspirateur Robot

Critères de recherche dans la clause WHERE

- Critères de recherche disponibles en SQL :
  - Égalité ou inégalité avec une valeur (=, <>, <, >, <=, >=)
  - LIKE (comparaison partielle)
  - BETWEEN … AND … (attribut entre deux valeurs)
  - IN (valeur d'attribut comprise dans une liste des valeurs)
  - IS NULL ou IS NOT NULL (comparaison avec NULL)
  - Critères composés à l'aide des connecteurs logiques AND, OR, NOT

Restrictions : exemples

- Exemple 6 : Lister les clients dont le nom comporte la lettre A en 2e position

```
SELECT *
FROM   CLIENT C
WHERE  C.nom LIKE '_A%'
```

  - LIKE permet d'effectuer une recherche sur un modèle particulier.
  - _ remplace n'importe quel caractère (un seul caractère)
  - % remplace n'importe quelle suite de caractères (un nombre quelconque de caractères)

- Exemple 7 : Lister les produits dont le prix est compris entre 200€ et 600€

```
SELECT  *
FROM    PRODUIT P
WHERE P.prix BETWEEN 200 AND 600
```

- Exemple 8 : Lister les produits de marque Apple, IBM, ou Dec

```
SELECT *
FROM PRODUIT P
WHERE P.marque IN ('Apple','IBM',
'Dec')
```

- Exemple 9 : Lister les produits dont le prix est inconnu

```
SELECT  *
FROM PRODUIT P
WHERE P.prix IS NULL
```

- Exemple 10 : Lister les produits de marque IBM dont le prix est inférieur à 1000€

```
SELECT  *
FROM PRODUIT P
WHERE P.marque='IBM' AND P.prix<1000
```

Tri

- La clause ORDER BY permet de spécifier les colonnes définissant les critères de tri.
- Le tri se fera d'abord selon la première colonne spécifiée, puis selon la deuxième colonne, etc.
- Exemple 11 : Lister les produits en les triant par marques et à l'intérieur d'une marque par prix décroissants

```
SELECT      *
FROM        PRODUIT P
ORDER BY    P.marque, P.prix DESC
```

- L'ordre de tri est précisé par ASC (croissant), ou DESC (décroissant) ; par défaut ASC.

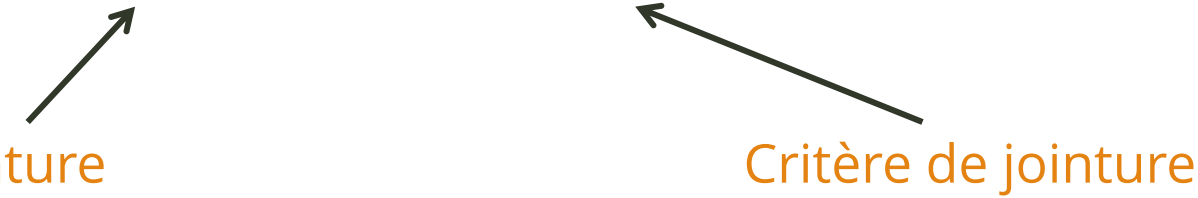
Résultat

idPro	description	marque	prix
2	Ordinateur Portable	Acer	799.99
7	Réfrigérateur	Bosch	549.99
8	Aspirateur Robot	Bosch	299.99
5	Appareil Photo Numérique	Canon	349.99
10	Enceinte Bluetooth	JBL	69.99
1	Téléviseur Smart 4K	LG	499.99
4	Machine à Café	Philips	89.99
3	Smartphone Haut de Gamme	Samsung	899.99
9	Tablette Numérique	Samsung	299.99
6	Console de Jeu	Sony	399.99

Jointures

- Les jointures en SQL permettent de croiser plusieurs tables dans une même requête.
- On peut ainsi exploiter la puissance d'un SGBD relationnel pour obtenir des résultats qui combinent les données de plusieurs tables de manière efficace.
- Exemple 12 : Donner les références (idPro) et les descriptions des produits vendus
  - La description des produits est dans la table PRODUIT
  - Les informations des ventes sont dans la table VENTE
  - On est obligé donc de croiser les deux tables à l'aide d'une jointure
  - Requête SQL :

```
SELECT P.idPro, P.description
FROM   PRODUIT P INNER JOIN VENTE V ON P.idPro=V.idPro
```



PRODUIT P

idPro	description	marque	prix
1	Téléviseur Smart 4K	LG	499.99
2	Ordinateur Portable	Acer	799.99
3	Smartphone Haut de	Samsung	899.99
4	Machine à Café	Philips	89.99
5	Appareil Photo Numé	Canon	349.99
6	Console de Jeu	Sony	399.99
7	Réfrigérateur	Bosch	549.99
8	Aspirateur Robot	Bosch	299.99
9	Tablette Numérique	Samsung	299.99
10	Enceinte Bluetooth	JBL	69.99

VENTE V

idCli	idPro	date	qte
1	1	01/10/2022	5
1	3	01/10/2023	3
5	1	03/11/2023	2
6	4	04/11/2021	4
4	5	05/11/2021	6
5	9	06/11/2021	1
6	7	07/11/2021	8
2	6	01/11/2023	1
2	7	01/11/2023	1
2	8	11/11/2023	1

```
SELECT P.idPro, P.description
FROM   PRODUIT P INNER JOIN VENTE V ON P.idPro=V.idPro
```

La table de jointure contient **toutes les paires des lignes** des deux tables qui **vérifient le critère de jointure** (P.idPro=V.idPro dans l'exemple)

PRODUIT P INNER JOIN VENTE V ON P.idPro=V.idPro

P.idPro	P.description	P.marque	P.prix	V.idCli	V.idPro	V.date	V.qte
1	Téléviseur Smart 4K	LG	499.99	1	1	01/10/2022	5
1	Téléviseur Smart 4K	LG	499.99	5	1	03/11/2023	2
3	Smartphone Haut de	Samsung	899.99	1	3	01/10/2023	3
4	Machine à Café	Philips	89.99	6	4	04/11/2021	4
5	Appareil Photo Numé	Canon	349.99	4	5	05/11/2021	6
6	Console de Jeu	Sony	399.99	2	6	01/11/2023	1
7	Réfrigérateur	Bosch	549.99	6	7	07/11/2021	8
7	Réfrigérateur	Bosch	549.99	2	7	01/11/2023	1
8	Aspirateur Robot	Bosch	299.99	2	8	11/11/2023	1
9	Tablette Numérique	Samsung	299.99	5	9	06/11/2021	1

P.idPro = V.idPro

Résultat

P.idPro	P.description
1	Téléviseur Smart 4K
1	Téléviseur Smart 4K
3	Smartphone Haut de Gamme
4	Machine à Café
5	Appareil Photo Numérique
6	Console de Jeu
7	Réfrigérateur
7	Réfrigérateur
8	Aspirateur Robot
9	Tablette Numérique

Jointures à 3 tables

- Exemple 13 : Donner les noms des clients qui ont acheté un produit de type « Téléviseur Smart 4K »
  - La description des produits est dans la table PRODUIT
  - Les noms des clients sont dans la table CLIENT
  - Les informations des ventes sont dans la table VENTE
  - Requête SQL :

```
SELECT C.nom
FROM   (PRODUIT P INNER JOIN VENTE V ON P.idPro=V.idPro) INNER JOIN
CLIENT C ON C.idCli=V.idCli
WHERE  P.description='Téléviseur Smart 4K'
```

PRODUIT P INNER JOIN VENTE V ON P.idPro=V.idPro

P.idPro	P.description	P.marque	P.prix	V.idCli	V.idPro	V.date	V.qte
1	Téléviseur Smart 4K	LG	499.99	1	1	01/10/2022	5
1	Téléviseur Smart 4K	LG	499.99	5	1	03/11/2023	2
3	Smartphone Haut de	Samsung	899.99	1	3	01/10/2023	3
4	Machine à Café	Philips	89.99	6	4	04/11/2021	4
5	Appareil Photo Numé	Canon	349.99	4	5	05/11/2021	6
6	Console de Jeu	Sony	399.99	2	6	01/11/2023	1
7	Réfrigérateur	Bosch	549.99	6	7	07/11/2021	8
7	Réfrigérateur	Bosch	549.99	2	7	01/11/2023	1
8	Aspirateur Robot	Bosch	299.99	2	8	11/11/2023	1
9	Tablette Numérique	Samsung	299.99	5	9	06/11/2021	1

CLIENT

idCli	nom	ville
1	Alice Dupont	Paris
2	Bob Martin	Marseille
3	Carol Smith	Lyon
4	David Johnson	Toulouse
5	Émilie Davis	Paris
6	François Leclerc	Bordeaux
7	Gabrielle Dubois	Paris
8	Henri Lefebvre	Nantes
9	Isabelle Moreau	Strasbourg
10	Jean Dupuis	Montpellier

(PRODUIT P INNER JOIN VENTE V ON P.idPro=V.idPro) INNER JOIN CLIENT C ON C.idCli=V.idCli

P.idPro	P.description	P.marque	P.prix	V.idCli	V.idPro	V.date	V.qte	C.idCli	C.nom	C.ville
1	Téléviseur Sma	LG	499.99	1	1	01/10/2022	5	1	Alice Dupont	Paris
1	Téléviseur Sma	LG	499.99	5	1	03/11/2023	2	5	Émilie Davis	Paris
3	Smartphone Ha	Samsung	899.99	1	3	01/10/2023	3	1	Alice Dupont	Paris
4	Machine à Café	Philips	89.99	6	4	04/11/2021	4	6	François Leclei	Bordeaux
5	Appareil Photo	Canon	349.99	4	5	05/11/2021	6	4	David Johnson	Toulouse
6	Console de Jeu	Sony	399.99	2	6	01/11/2023	1	2	Bob Martin	Marseille
7	Réfrigérateur	Bosch	549.99	6	7	07/11/2021	8	6	François Leclei	Bordeaux
7	Réfrigérateur	Bosch	549.99	2	7	01/11/2023	1	2	Bob Martin	Marseille
8	Aspirateur Rob	Bosch	299.99	2	8	11/11/2023	1	2	Bob Martin	Marseille
9	Tablette Numéi	Samsung	299.99	5	9	06/11/2021	1	5	Émilie Davis	Paris

Résultat final de la requête : (appliquer la restriction, puis la projection sur la table de jointure)

```
  SELECT C.nom
  FROM   PRODUIT P INNER JOIN VENTE V ON P.idPro=V.idPro INNER JOIN
CLIENT C ON C.idCli=V.idCli
  WHERE  P.designation='Téléviseur Smart 4K'
```

Résultat

C.nom
Alice Dupont
Émilie Davis

Auto-jointure

- Exemple 14 : Donner les noms des clients de la même ville que Gabrielle Dubois
  - Pour le couplage des villes, on va utiliser la jointure de la table CLIENT avec elle-même.
  - Pour distinguer les 2 copies de la table, il faut introduire 2 synonymes différents (C1 et C2) de la table CLIENT.
  - Requête SQL :

```
SELECT C2.nom
FROM   CLIENT C1 INNER JOIN CLIENT C2 ON C1.ville=C2.ville
WHERE  C1.nom= 'Gabrielle Dubois'
```

CLIENT C1 (restreint sur nom='Gabrielle Dubois')

idCli	nom	ville
7	Gabrielle Dubois	Paris

CLIENT C2

idCli	nom	ville
1	Alice Dupont	Paris
2	Bob Martin	Marseille
3	Carol Smith	Lyon
4	David Johnson	Toulouse
5	Émilie Davis	Paris
6	François Leclerc	Bordeaux
7	Gabrielle Dubois	Paris
8	Henri Lefebvre	Nantes
9	Isabelle Moreau	Strasbourg
10	Jean Dupuis	Montpellier

C1.ville = C2.ville

Auto-jointure

C1.idCli	C1.nom	C1.ville	C2.idCli	C2.nom	C2.ville
7	Gabrielle Dubois	Paris	1	Alice Dupont	Paris
7	Gabrielle Dubois	Paris	5	Émilie Davis	Paris
7	Gabrielle Dubois	Paris	7	Gabrielle Dubois	Paris

Résultat final : `SELECT C2.nom` sur la table de l'auto-jointure

C2.nom
Alice Dupont
Émilie Davis
Gabrielle Dubois

Fonctions d'agrégation

- Permettent d'effectuer des opérations statistiques sur un ensemble de lignes.
- Idéales pour effectuer quelques statistiques de base sur des tables :
  - MIN : Récupérer la valeur la plus petite
  - MAX : Récupérer la valeur la plus grande
  - AVG : Calculer la moyenne
  - SUM : Calculer la somme
  - COUNT : Compter

Fonctions d'agrégation : exemples

- Exemple 15 : Compter toutes les lignes dans la table CLIENT

```
SELECT COUNT(*)
FROM CLIENT
```

COUNT(*)
10

- Exemple 16 : Donner le nombre total de clients ayant acheté au moins une fois

```
SELECT COUNT(DISTINCT idCli)
FROM VENTE
```

COUNT(DISTINCT idCli)
5

  - Le mot DISTINCT élimine les doublons avant d'appliquer la function COUNT.

- Exemple 17 : Donner le nombre total de « Téléviseur Smart 4K » vendus

```
SELECT SUM(V.qte)
FROM VENTE V INNER JOIN PRODUIT P ON V.idPro=P.idPro
WHERE P.description= 'Téléviseur Smart 4K'
```

SUM(V.qte)
7

Fonctions d'agrégation avec groupes

- La clause GROUP BY permet de partitionner une table en plusieurs groupes.
- Les fonctions d'agrégation opèrent sur chaque groupe.
- Exemple 18 : Donner le nombre de clients par ville

- Sans GROUP BY :

```
SELECT COUNT(*)
FROM CLIENT C
```

- Résultat :

COUNT(*)
10

- Ne donne pas l'information **par ville**

- Avec GROUP BY :

```
SELECT COUNT(*)
FROM CLIENT C
GROUP BY C.ville
```

- Résultat :

COUNT(*)
1
1
1
1
1
3
1
1

- Avec GROUP BY et affichage de la ville :

```
SELECT C.ville, COUNT(*)
FROM CLIENT C
GROUP BY C.ville
```

- Résultat :

ville	COUNT(*)
Bordeaux	1
Lyon	1
Marseille	1
Montpellier	1
Nantes	1
Paris	3
Strasbourg	1
Toulouse	1

Autres exemples avec GROUP BY

- Exemple 19 : Donner pour chaque référence de produit la quantité totale vendue

```
SELECT V.idPro, SUM(V.qte)
FROM VENTE V
GROUP BY V.idPro
```

idPro	SUM(V.qte)
1	7
3	3
4	4
5	6
6	1
7	9
8	1
9	1

- Exemple 20 : Donner la quantité totale achetée par chaque client

```
SELECT V.idCli, SUM(V.qte)
FROM VENTE V
GROUP BY V.idCli
```

idCli	SUM(V.qte)
1	8
2	3
4	6
5	3
6	12

Restrictions de groupes

- La clause HAVING permet de spécifier une condition de restriction des groupes.
- La clause HAVING sert à éliminer certains groupes.
  - À distinguer de WHERE, qui sert à éliminer des lignes.
- Exemple 21 : Donner les noms des marques dont le prix moyen des produits est <500€.

```
SELECT P.marque, AVG(P.prix)
FROM PRODUIT P
GROUP BY P.marque
HAVING AVG(P.prix)<500
```

- Exemple 22 : Donner les références des produits achetés en quantité>10, plus que 50 fois.

```
SELECT V.idPro, V.qte, COUNT(*)
FROM VENTE V
WHERE V.qte>10
GROUP BY V.idPro
HAVING COUNT(*)>50
```

La forme générale de SELECT

SELECT	**[DISTINCT]** *liste d'attributs, expressions*
FROM	*liste de tables ou jointure*
WHERE	*condition de restriction de lignes*
GROUP BY	*attributs de partitionnement*
HAVING	*condition de restriction de groups*
ORDER BY	*liste de colonnes* **[ASC** *ou* **DESC]**

- Exemple 23 : Donner les idPro, les prix, les marques et la quantité maximum vendue de tous les produits Samsung ou LG, dont la quantité totale vendue est supérieure à 500 et dont les quantités vendues sont >10 par vente. Trier par quantité max vendue décroissante.

```
SELECT P.idPro, P.prix, P.marque, MAX(V.qte)
FROM PRODUIT P INNER JOIN VENTE V ON
P.idPro=V.idPro
WHERE P.marque IN ('Samsung', 'LG') AND V.qte>10
GROUP BY P.idPro, P.prix, P.marque
HAVING SUM(V.qte)>500
ORDER BY MAX(V.qte) DESC
```

Construction du résultat d'une requête SELECT

- Du seul point de vue logique, on peut considérer que le résultat d'un SELECT est construit suivant les étapes :
  1. FROM

     La clause FROM est évaluée de manière à produire une nouvelle table.
  2. WHERE

     Le résultat de l'étape 1 est réduit par élimination de toutes les lignes qui ne satisfont pas la condition de WHERE.
  3. GROUP BY

     Le résultat de l'étape 2 est partitionné selon les valeurs des colonnes dont le nom figure dans la clause GROUP BY.
  4. HAVING

     Les groupes ne satisfaisant pas la condition HAVING sont éliminés du résultat de l'étape 3.
  5. SELECT

     Chaque groupe génère une seule ligne du résultat. Les colonnes ne figurant pas dans la clause SELECT sont éliminées.
  6. ORDER BY

     Les critères de tri de la clause ORDER BY sont appliqués.

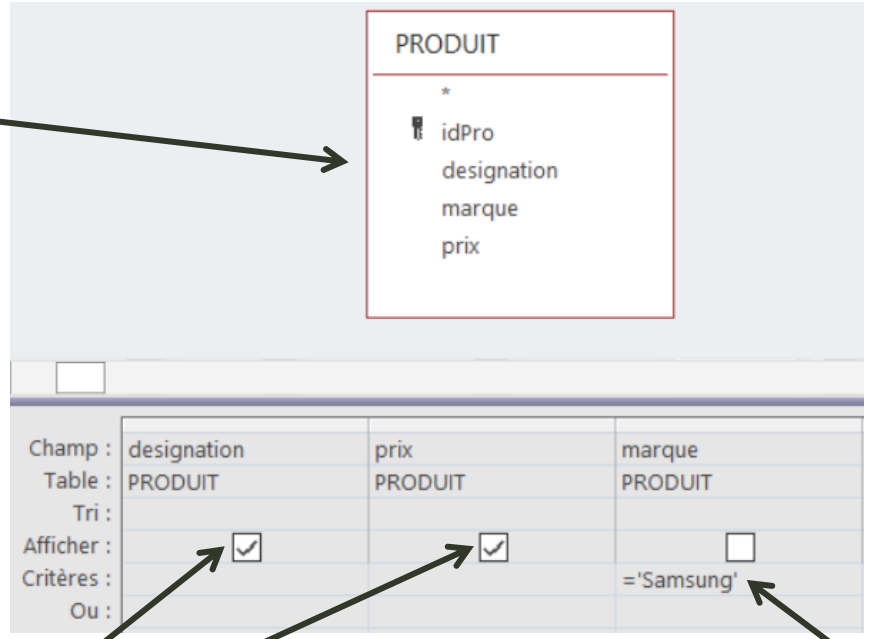
QBE

QBE : Query By Example

- Langage graphique de construction de requêtes SQL.
- Principe : Formulation de la requête en spécifiant la forme de la réponse.
- Formulation graphique, à la souris.
- Supporté par Access comme assistant de création de requêtes.
- Moins complet que SQL.
- Une requête QBE est traduite en interne en SQL, puis exécutée par le SGBD.

QBE : exemple 1

Tables utilisées (FROM)



- Se traduit à la requête SQL suivante :

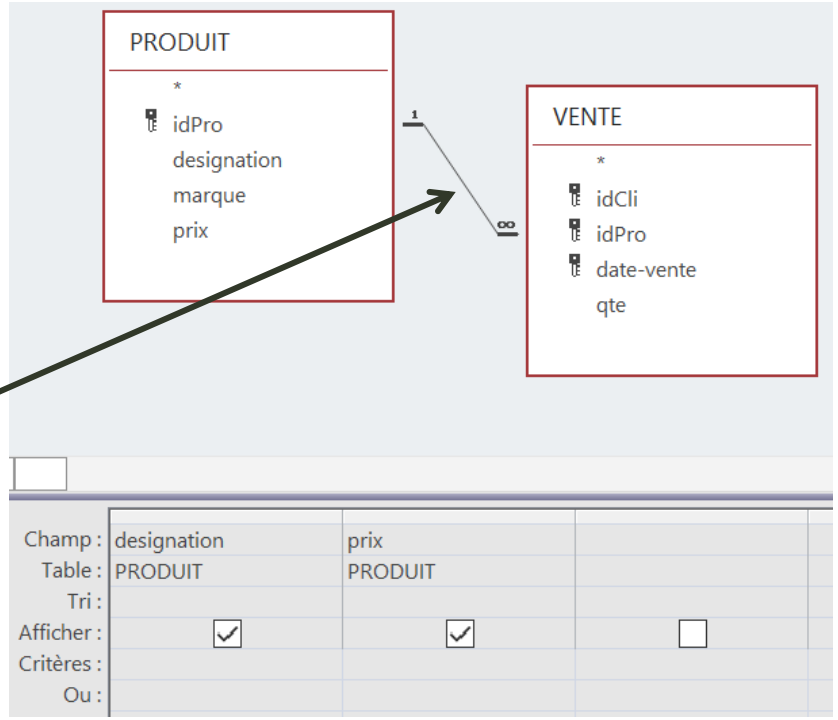
```
SELECT PRODUIT.designation, PRODUIT.prix
FROM PRODUIT
WHERE (((PRODUIT.marque)='Samsung'))
```

Attributs du résultat (SELECT)

Attributs avec une condition (WHERE)

QBE : exemple 2

Jointure entre les tables PRODUIT et VENTE



- Se traduit à la requête SQL suivante :

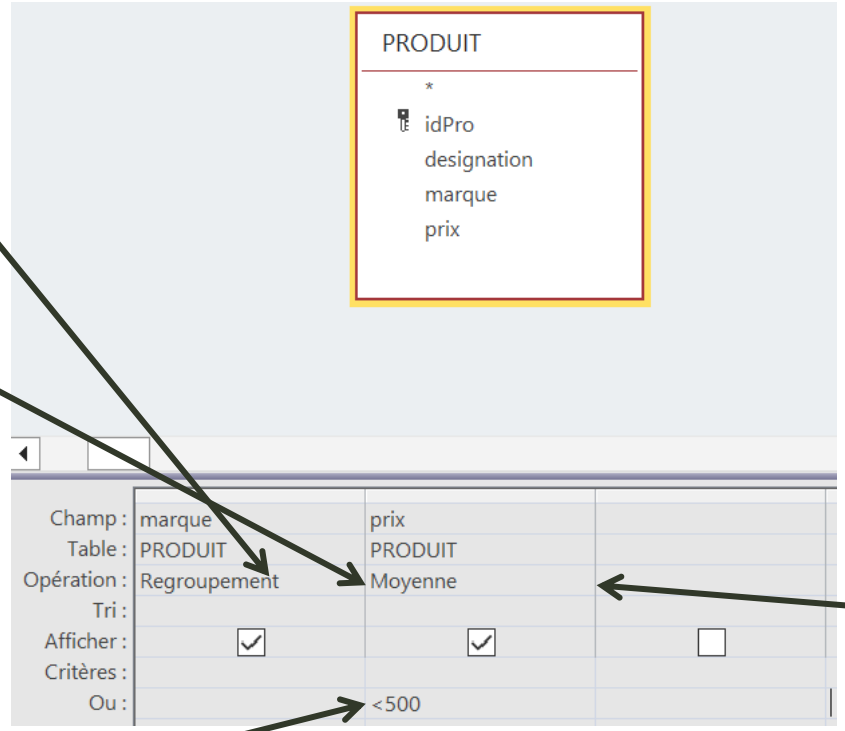
```
SELECT PRODUIT.designation, PRODUIT.prix
FROM PRODUIT INNER JOIN VENTE ON
PRODUIT.idPro = VENTE.idPro
```

QBE : exemple 3

Attribut de partitionnement (GROUP BY)

Fonction d'agrégation

Condition de groupe (HAVING)



- Se traduit à la requête SQL suivante :

```
SELECT PRODUIT.marque, Avg(PRODUIT.prix) AS MoyenneDeprix
FROM PRODUIT
GROUP BY PRODUIT.marque
HAVING (((Avg(PRODUIT.prix))<500))
```

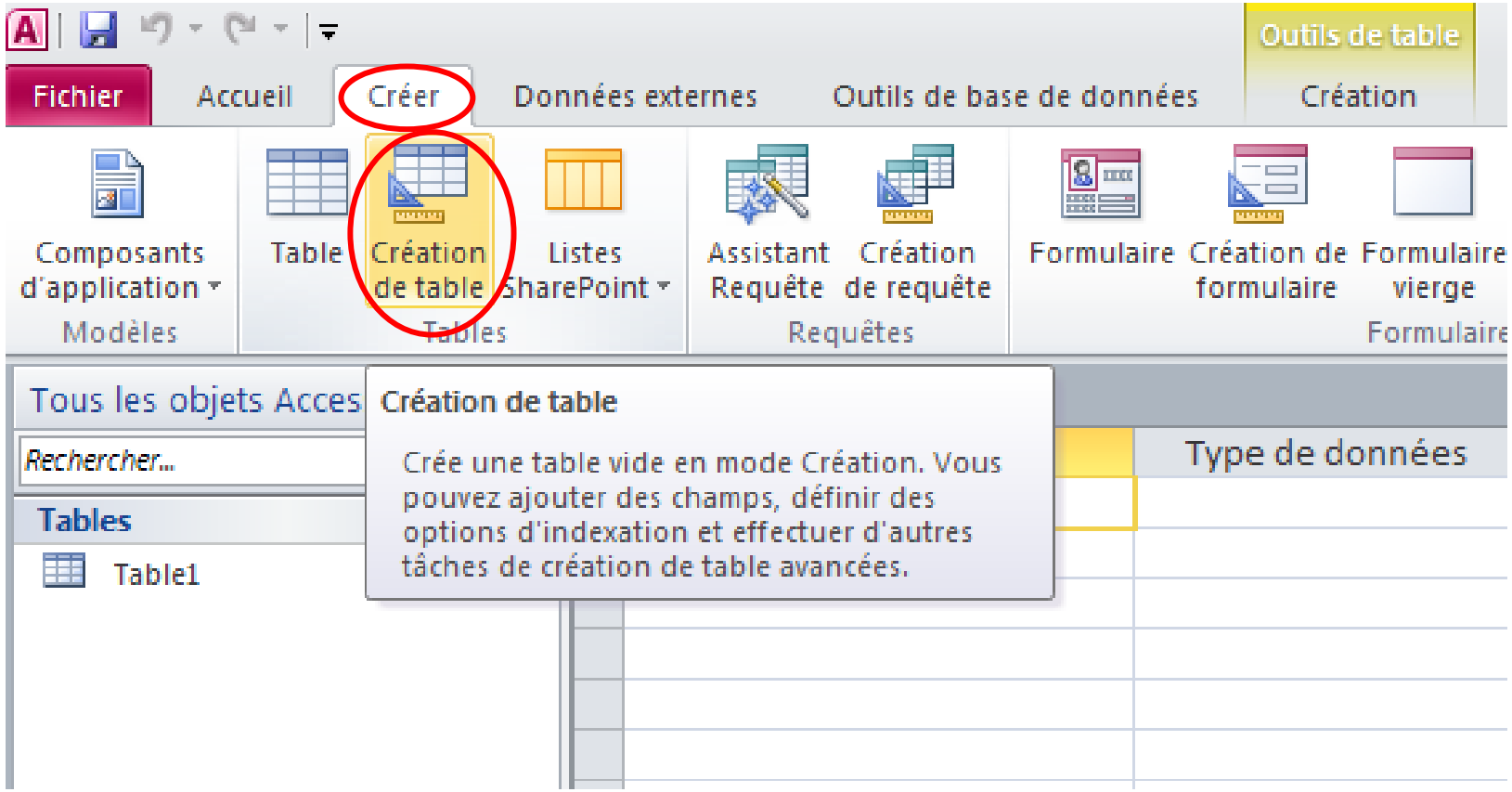
La ligne « Opération » s'affiche en cliquant sur le bouton « Totaux », dans le menu « Conception de requêtes »

Access : fonctionnalités de base

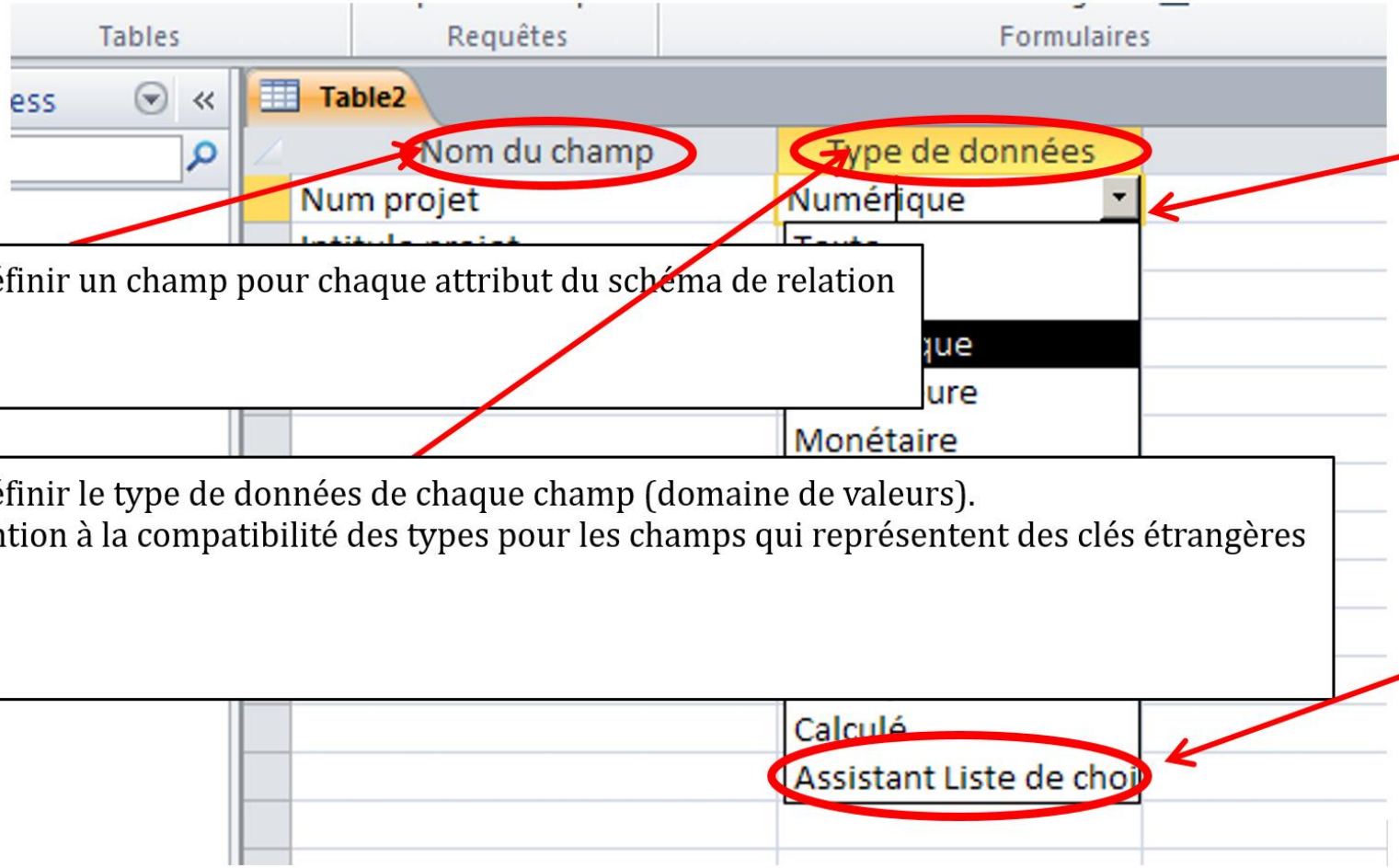
Que faire une fois que j'ai créé une base de données vide …

- Schéma de relation ⇨ Table dans le SGBD
  - Pour chaque attribut (champ), spécifier son **type de données**
  - Si besoin, créer des listes de choix pour les valeurs de certains attributs
- Clés primaires ⇨ Spécifiées lors de la création d'une table
- Clés étrangères ⇨ Liens (relations) entre les tables
  - Intégrité référentielle
  - Mises à jour en cascade (typiquement OUI)
  - Suppressions en cascade (NON)
- Créer des formulaires pour la saisie de données
- Créer des états d'impression (rapports)
- Créer des requêtes SQL pour les opérations courantes

Créer une table



Définir les noms de champs et les types de données



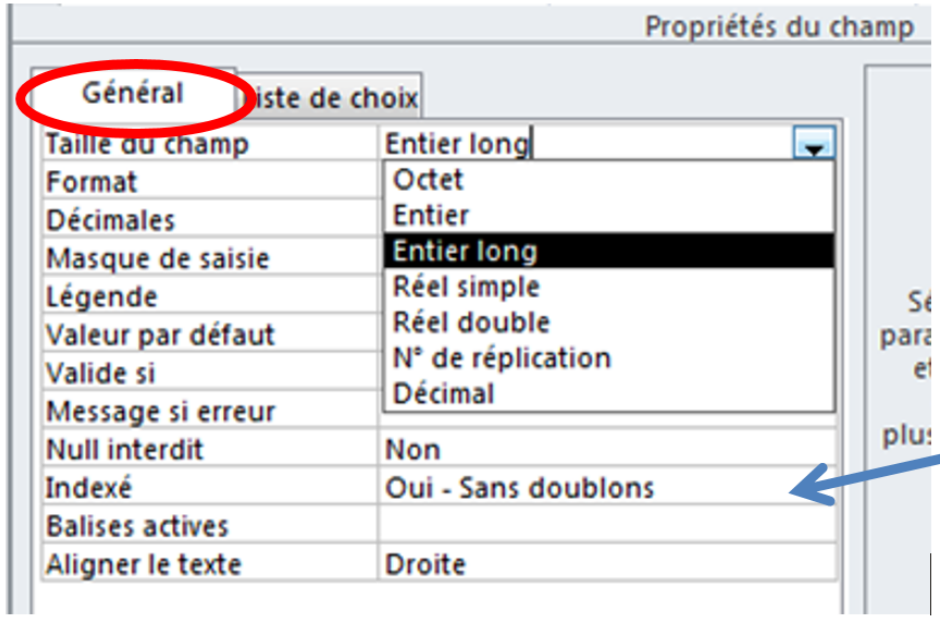
1- Définir un champ pour chaque attribut du schéma de relation

2- Définir le type de données de chaque champ (domaine de valeurs).
Attention à la compatibilité des types pour les champs qui représentent des clés étrangères

Liste des types de données disponibles

*Pour définir une **liste de choix***

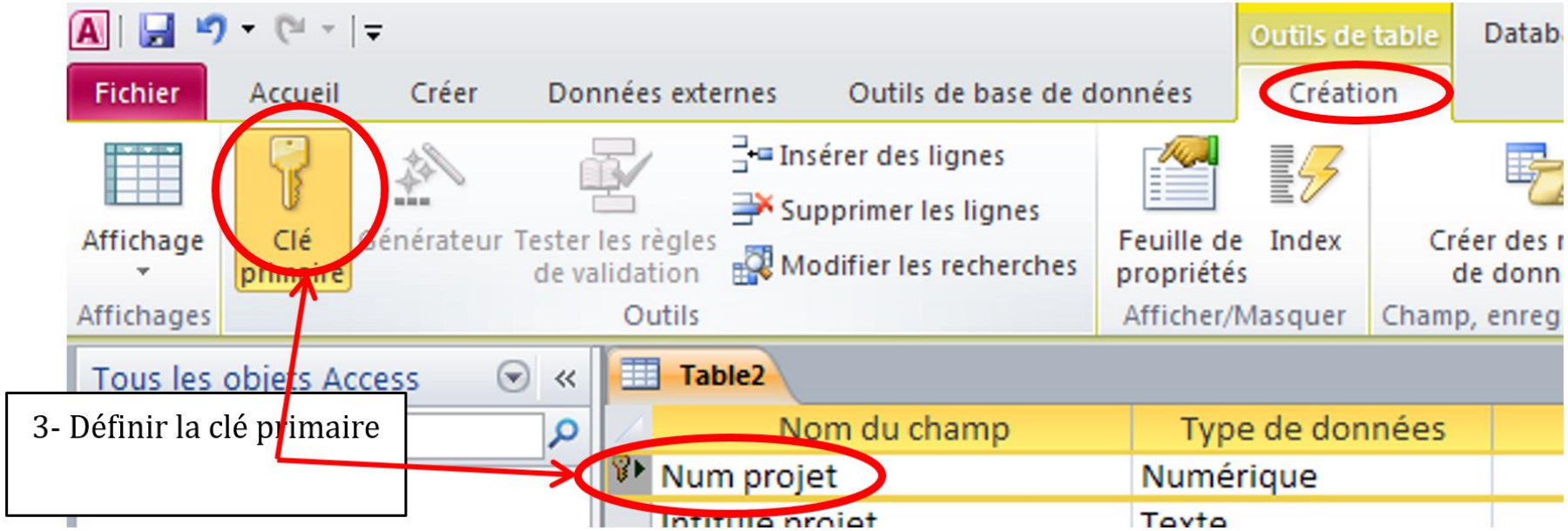
Contraindre les valeurs d'un champ (au-delà de son type de données)



Taille du champ
- Choix pour le type Numérique
- pour le type texte : 255 caractères par défaut, à préciser si l'on veut le réduire

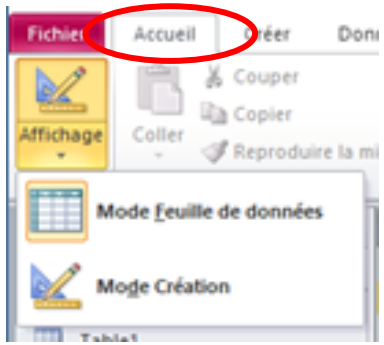
« oui-sans doublons » pour un champ clé primaire

Définir la clé primaire



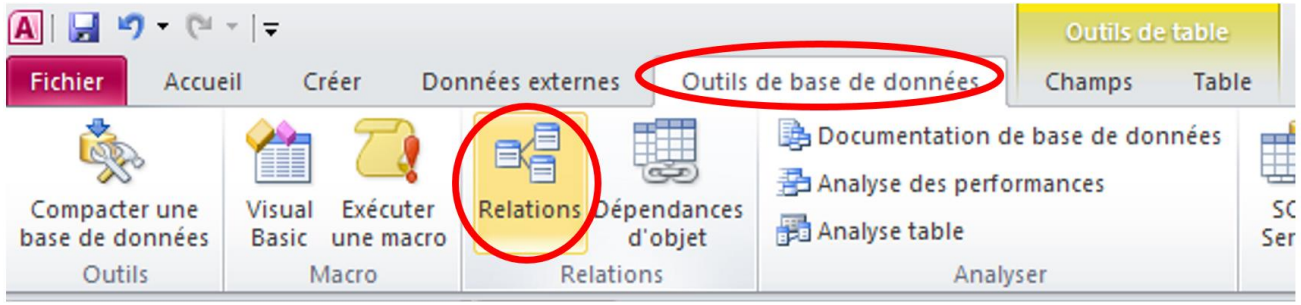
- Une clé primaire peut être composée de plusieurs attributs :
  - Dans ce cas il faut sélectionner simultanément les différents champs (touche MAJ) puis cliquer sur l'icône de clé.

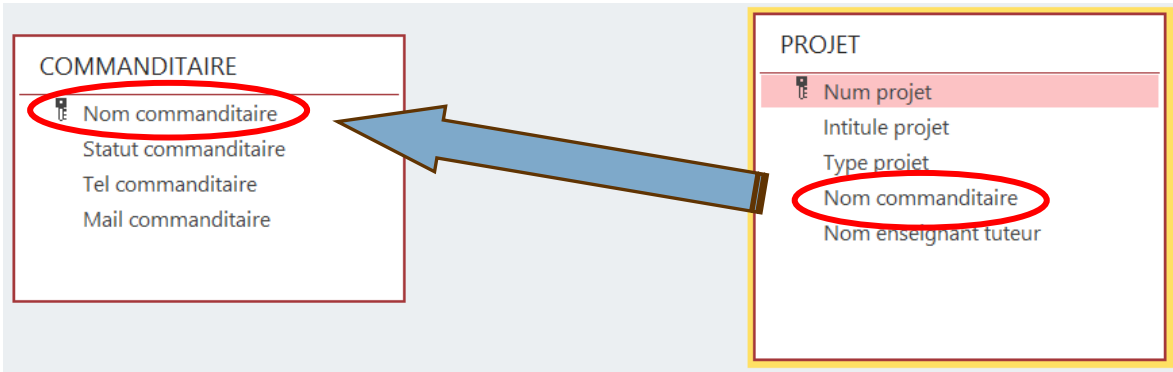
Afficher une table



- Mode Feuille de données
  - Permet de visualiser et modifier les **données** de la table.
- Mode Création
  - Permet de définir la **structure** de la table (champs, types de données, clé primaire).

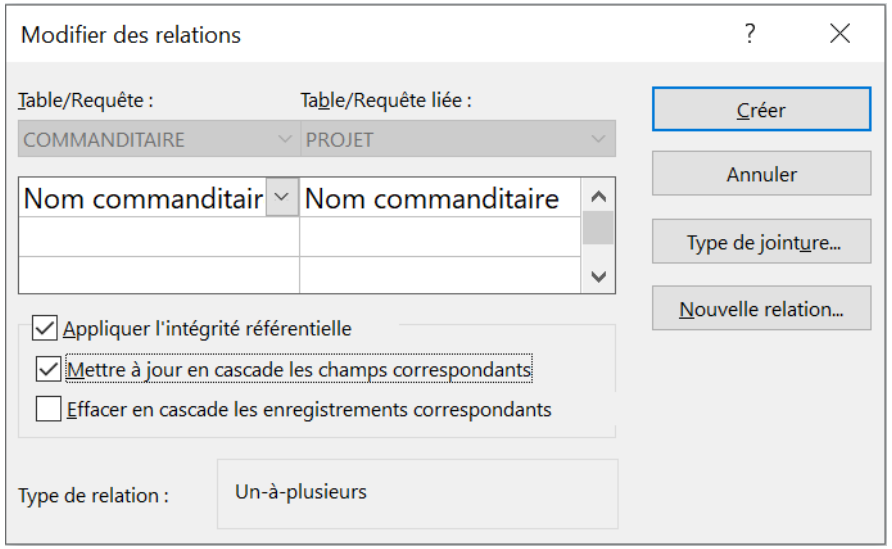
Définir les relations entre les tables (clés étrangères)

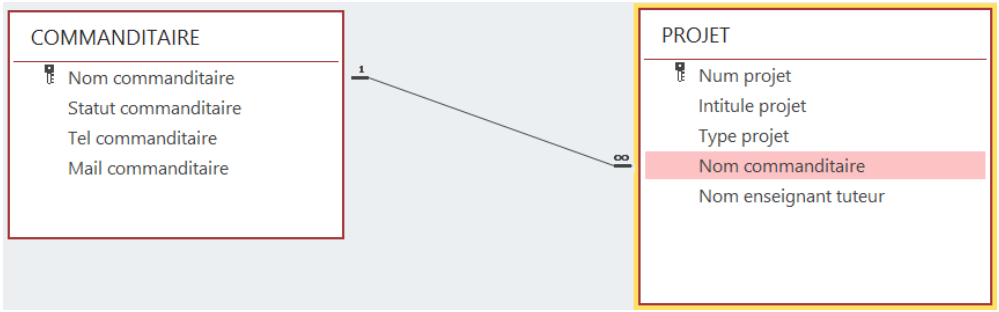




- Faites glisser le champ clé étrangère vers le champ clé primaire.
  - **Pas dans l'autre sens !**

Définir l'intégrité référentielle

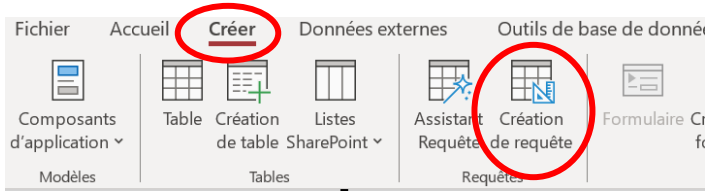




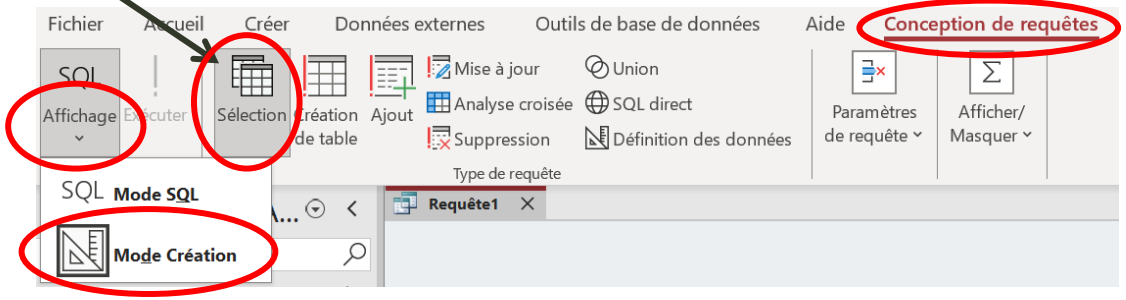
- Dans la fenêtre qui apparaît :
  - Vérifiez les deux champs qui seront reliés
  - Appliquez l'intégrité référentielle
    - **Attention : Les deux champs doivent avoir exactement le même type de données**
  - Activez les mises à jour en cascade

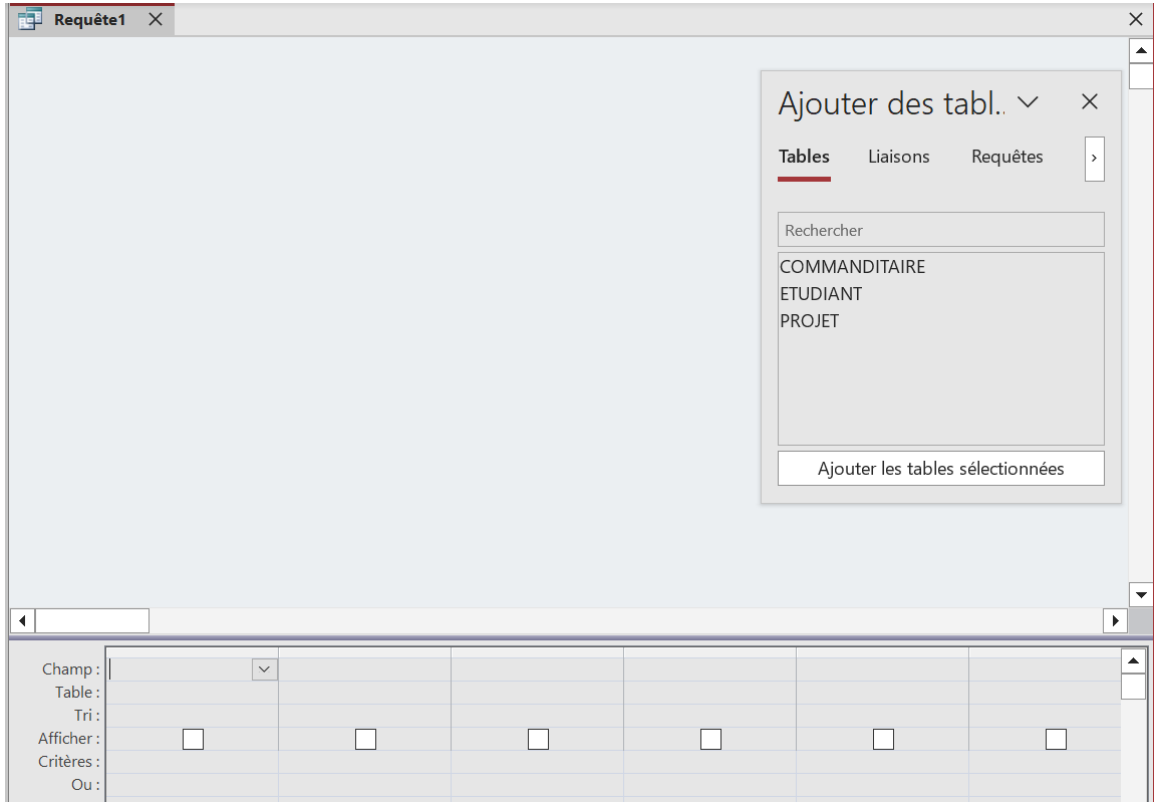
- Les deux tables sont reliées.

Créer des requêtes QBE



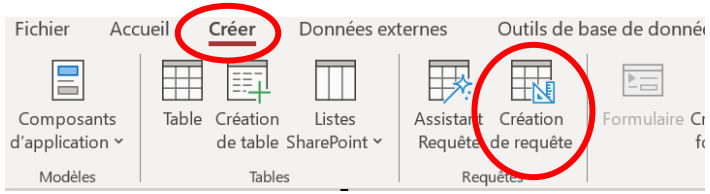
Construire une requête SELECT



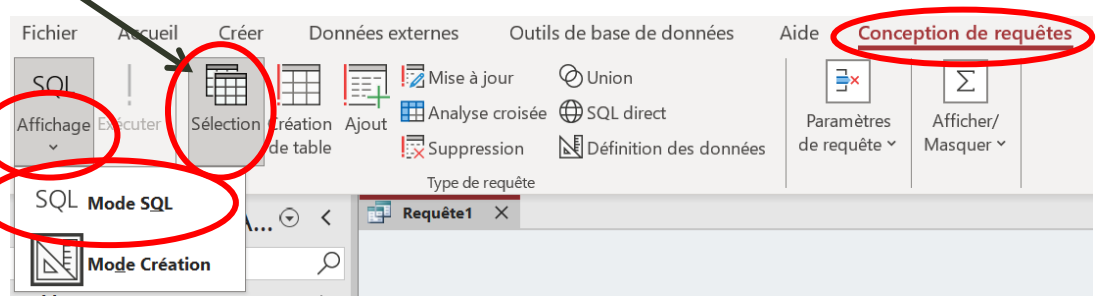


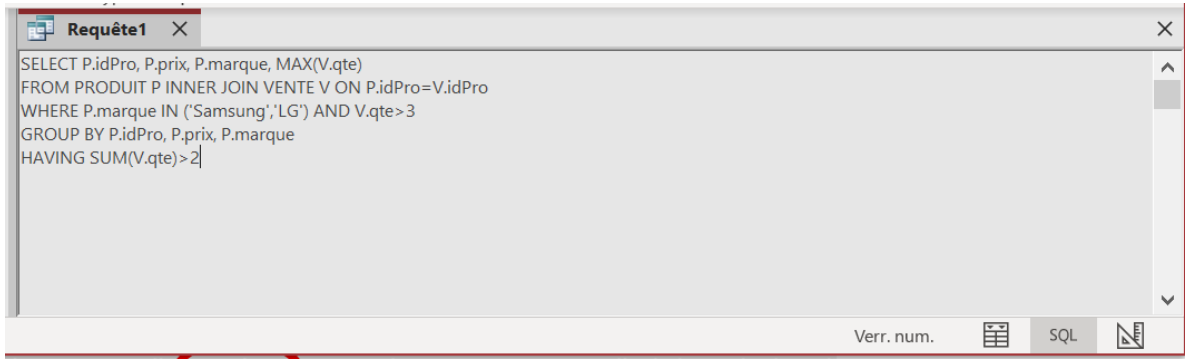
- En mode Création :
  - Éditeur visuel QBE

Créer des requêtes SQL



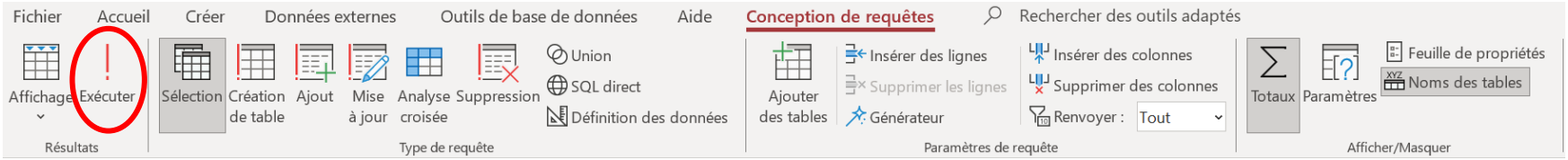
Construire une requête SELECT





- En mode SQL :
  - Écrire directement une requête SQL

Afficher le résultat d'une requête



- Pour tester votre requête, ou pour afficher le résultat, il suffit de cliquer sur « Exécuter ».
- Vous basculez en mode Feuille de données et le résultat apparaît !