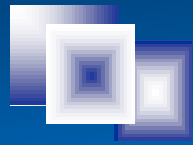


Bases de données

Pr Boubker Sbihi
Enseignant-chercheur
Bsbih@hotmail.com



Plan



Introduction

Gestion des Bases de données

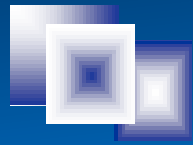
Gestion des Tables

Requêtes de sélection

Fonctions

Bibliographie

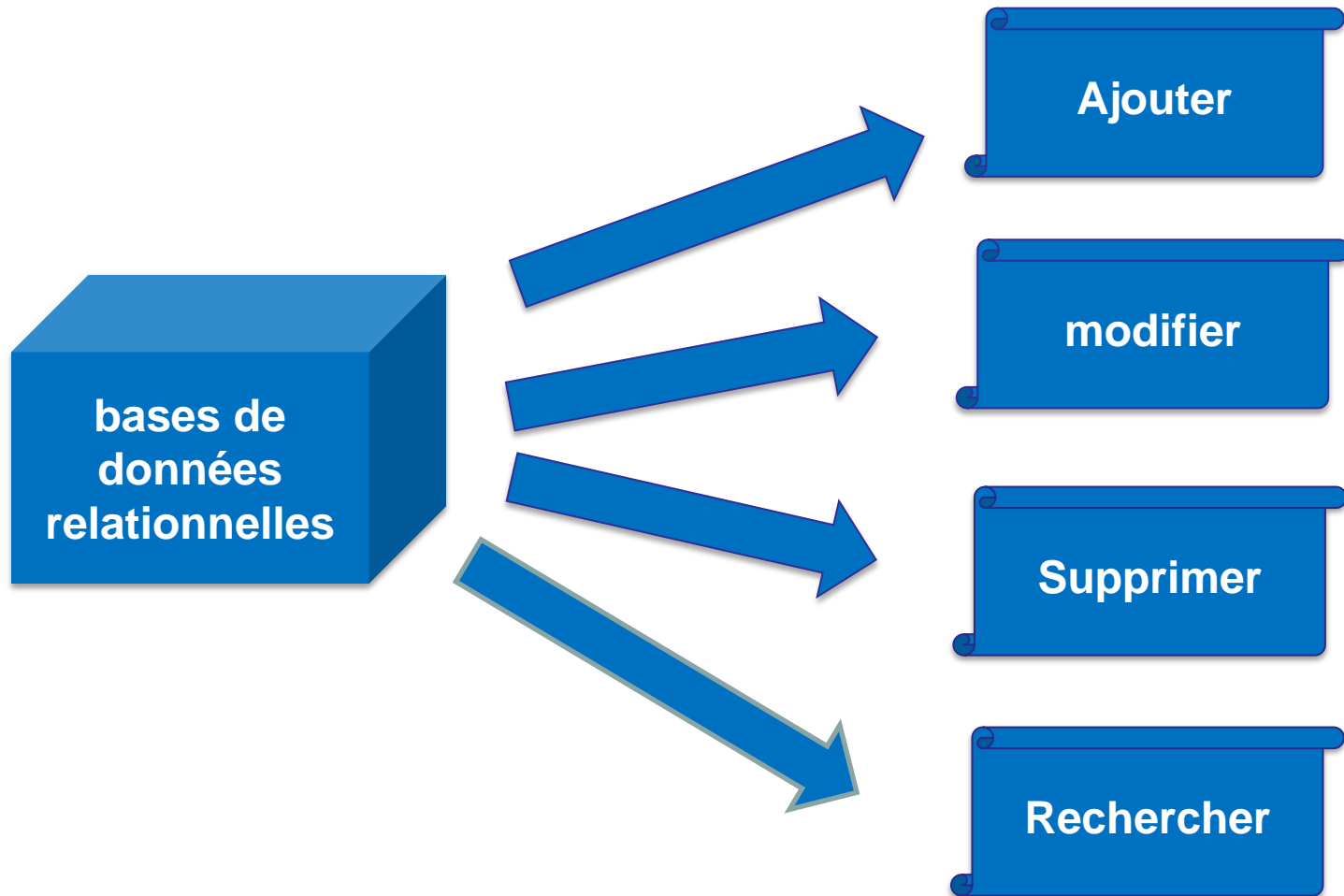
Introduction

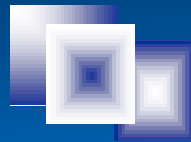


Introduction

- ✓ **SQL (Structured Query Language)** est un langage évolué normalisé de manipulation et d'exploitation des bases de données relationnelles
- ✓ Il comprend un ensemble de commandes
- ✓ Il est basé sur l'algèbre relationnelle

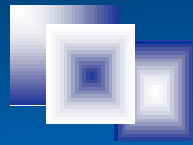
Objectifs





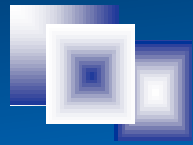
Historique

- ✓ Développé par IBM en 1974
- ✓ Auteurs : Donald D. Chamberlin et Raymond F. Boyce
- ✓ Normalisé en 87 par ANSI de SQL
- ✓ Dernière version normalisée SQL:2011
- ✓ Centaines de produits supportant SQL



Caractéristiques

- ✓ Proche du langage humain
- ✓ Utilise des mots simples en anglais
- ✓ Facile et simple à apprendre
- ✓ Conçu pour les non informaticiens



Caractéristiques

- ✓ Défini indépendamment des plates formes
- ✓ Langage de requêtes pour les SGBD
- ✓ Migration entre les SGBD
- ✓ Pas de différence entre majuscule et minuscule

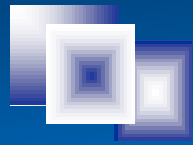
Dimensions

Dimensions

Langage de définition de données (LDD)

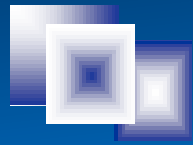
Langage de manipulation des données (LMD)

Langage de contrôle des données (LCD)



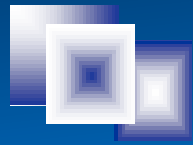
Dimensions

- ✓ Langage de définition de données (LDD) :
permet la description de la structure de la
base : tables, vues, index, attributs
 - ✓ Create
 - ✓ Alter
 - ✓ Drop



Dimensions

- ✓ Langage de manipulation des données (LMD) qui permet la manipulation des tables et des vues
 - ✓ Select
 - ✓ Update
 - ✓ Insert
 - ✓ Delete



Dimensions

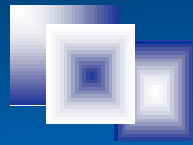
- ✓ Langage de contrôle des données pour gérer la confidentialité (LCD) qui permet de gérer les primitives de gestion des transactions et des privilèges d'accès aux données
 - ✓ Grant
 - ✓ Revoke



Dimensions

LDD	LMD	LCD
CREATE	SELECT	GRANT
DROP	INSERT	REVOKE
ALTER	DELETE	CONNECT
	UPDATE	COMMIT
		ROLLBACK
		SET

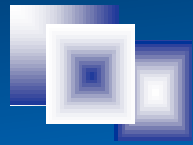
Gestion des Bases de données



Objets BD

Les objets de la base de données sont de trois types :

- ✓ Les tables
- ✓ Les vues
- ✓ Les indexes

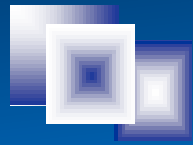


Création d'une BD

CREATE DATABASE nom-database

✓ nom-database : nom de la base de données

CREATE DATABASE DB_PERSONNE



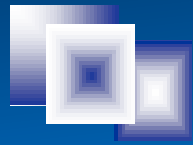
Création d'une BD

DROP DATABASE nom-database

✓ nom-database : nom de la base de données

DROP database DB_PERSONNE

Gestion des Tables



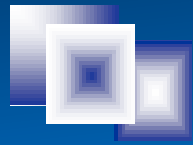
Création d'une TABLE

CREATE TABLE nom-table

(Attribut 1 type 1, Attribut 2 type 2,....

Attribut n type n)

✓ nom-table : nom de la table



Création d'une TABLE

```
CREATE TABLE Personne  
(  
  CIN VARCHAR(15) NOT NULL PRIMARY KEY,  
  Nom VARCHAR(40),  
  Prenom VARCHAR(40),  
);
```



Modification d'une TABLE

ALTER TABLE nom-table ADD COLUMN

Attribut type;

- ✓ nom-table : nom de la table
- ✓ ADD COLUMN ou DROP COLUMN



Modification d'une TABLE

```
ALTER TABLE Personne ADD COLUMN Age  
INTEGER NULL;
```

```
ALTER TABLE Personne DROP COLUMN Age;
```

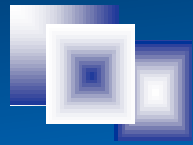


Suppression d'une TABLE

DROP TABLE nom-table

✓ nom-table : nom de la table

DROP TABLE **Personne**



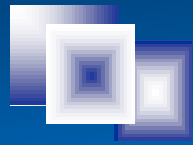
Création d'une FK

```
CREATE TABLE Auteur  
(  
  CIN VARCHAR(15) NOT NULL PRIMARY KEY,  
  Nom VARCHAR(40),  
  Prenom VARCHAR(40),  
);
```




Création d'une FK

```
CREATE TABLE LIVRE (  
ISBN VARCHAR(15) NOT NULL,  
TITRE VARCHAR(200),  
TYPE VARCHAR(40),  
CIN_AUT VARCHAR(15),  
PRIMARY KEY (ISBN),  
FOREIGN KEY (CIN_AUT) REFERENCES  
AUTEUR(CIN) );
```



Insertion

- ✓ La commande **INSERT INTO** est utilisée pour ajouter des enregistrements dans une base de données
- ✓ **VALUES** pour inclure les données



Insertion

```
INSERT                INTO                TABLE  
(champ1,champ2,champn)                VALUES  
(val1,val2,valn)
```

- ✓ Table est le nom de la table dans laquelle nous souhaitons insérer les données
- ✓ champ1 et champ2 sont les noms des champs
- ✓ val1,val2,valn sont les valeurs à insérer

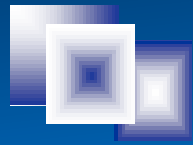


Insertion

```
INSERT INTO Personne (cin,nom,prenom) VALUES  
('1','Azizi','Ali')
```

```
INSERT INTO Personne (cin,nom,prenom) VALUES  
(‘2’,‘Ahmadi’,‘Aziz’)
```

```
INSERT INTO Personne (cin,nom,prenom) VALUES  
(‘3’,‘hilali’,‘rachid’)
```



Mise à jour

- ✓ La commande **UPDATE** est utilisée pour mettre à jour des enregistrements.
- ✓ Elle s'utilise en association avec **SET** et un opérateur.



Mise à jour

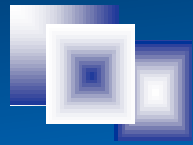
UPDATE Table SET (champ1opérateur
valeur1) WHERE champx OPERATEUR
valx

- ✓ Table est le nom de la table dans laquelle nous souhaitons mettre à jour
- ✓ champ1 et champ2 sont les noms des champs
- ✓ val1, val2, valn sont les valeurs



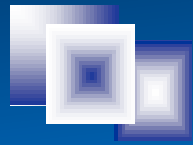
Mise à jour

```
UPDATE nom SET nom='Aaoui' WHERE cin='1'
```



Suppression

- ✓ La fonction **DELETE** est utilisée dans le but de supprimer des enregistrements
- ✓ Elle est utilisée avec l' instruction **FROM** et impose la condition **WHERE**



Suppression

DELETE FROM Table WHERE champn
opérateur valeur

- ✓ Table est le nom de la table dans laquelle nous souhaitons mettre à jour
- ✓ champ1 et champ2 sont les noms des champs
- ✓ val1, val2, valn sont les valeurs



Suppression

```
DELETE FROM Personne WHERE nom='azizi'
```

Requêtes de sélection



Sélection

Les requêtes sont structurées pour répondre à ces quatre rubriques comme suit :

SELECT je veux telles choses

FROM A prendre de telles sources

WHERE Ayant telles caractéristiques

ORDER BY ordonnées de telle manière



Sélection

```
SELECT champ1,champ2 FROM Table  
WHERE condition
```

- ✓ champ1 et champ2 sont les noms des champs
- ✓ Table est le nom de la table dans laquelle nous souhaitons extraire les données
- ✓ WHERE ou LIKE est optionnel et indique une condition

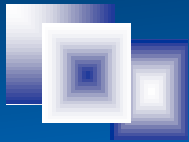


Sélection

```
SELECT Nom, Prénom FROM Personne WHERE  
Nom="Alaoui"
```

```
SELECT Nom as 'noms des amis' FROM Personne  
LIKE 'A%'
```

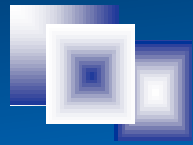
```
SELECT * FROM Personne WHERE Nom="Alaoui"
```



Sélection distincte

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```

```
SELECT DISTINCT Nom FROM Personne LIKE 'A%'
```

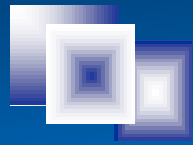


Tri

On utilise l'instruction **ORDER BY** pour trier les enregistrements par ordre alphabétiquement ou numérique

Il faut ajouter **ASC** pour un tri ascendant ou **DSC** pour un tri descendant.

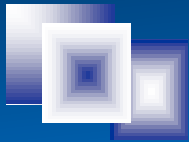

```
SELECT Nom FROM Personne WHERE Prenom LIKE  
'A%' order by cin asc
```



Jointure

```
SELECT champ1,champ2 FROM table1 T1,  
table1 T2 WHERE condition
```

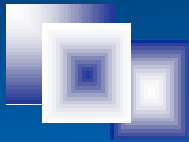
- ✓ champ1 et champ2 sont les noms des champs
- ✓ Table 1 et 2 : est le nom de la table dans laquelle nous souhaitons extraire les données
- ✓ WHERE ou LIKE est optionnel et indique une condition



Jointure

```
SELECT Nom_Aut,Tit_Liv FROM Auteur A,Livre L  
WHERE Num_Aut=1;
```

Fonctions



Fonctions

- ✓ AND & OR : et et ou
- ✓ Between : Entre
- ✓ MIN : Minimum
- ✓ MAX : Maximum
- ✓ AVG: Moyenne
- ✓ Count : nombre
- ✓ SUM : Somme



Fonctions

SELECT MIN(column_name)

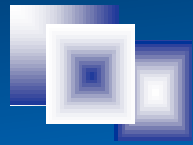
FROM table WHERE condition;

- ✓ Table est le nom de la table dans laquelle nous souhaitons extraire les données
- ✓ WHERE est optionnel et indique une condition



Fonctions

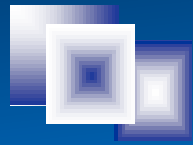
```
SELECT MAX(age)  
FROM Personne  
WHERE Nom is not null;
```



AND,OR,NOT

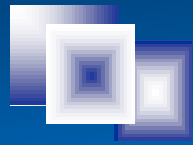
SELECT column1, column2, ...

FROM table WHERE (condition1 AND
condition2) or (NOT condition);



BETWEEN

```
SELECT column_name(s)  
  
FROM table_name  
  
WHERE column_name BETWEEN value1  
AND value2;
```



IN

```
SELECT column_name(s)
```

```
FROM table_name
```

```
WHERE column_name IN (value1, value2,  
...);
```

```
SELECT column_name(s)
```

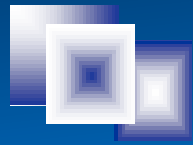
```
FROM table_name
```

```
WHERE column_name IN (NVL SELECT);
```



Bibliographie

- SQL Synthèse de cours et exercices, Frédéric Brouard, Christian Soutou, Rudi Bruchez, Pearson Education, 2012
- Bases de données de la modélisation au SQL, Laurent Audibert - Ellipses, 2009
- Administrez vos bases de données avec MySQL, Chantal Gribaumont, 2012



Bibliographie

- <https://www.w3schools.com/sql/trysql.asp>
- <https://www.w3schools.com/sql/>
<http://www.netalya.com/fr/sql.asp>
- <http://sql.sh/>
- <https://sql.developpez.com/>
- <https://openclassrooms.com/courses/.../introduction-au-langage-sql>



Fin

MERCI

POUR VOTRE AIMABLE

ATTENTION