19/10/2022 – Base de données SQL

Compétence 4 : Gérer des données

UE1.4 :

* R1.05 - introduction aux bases de données et SQL
* R1.06 - Mathématique discrète = 60%
* R1.09 – Economies
* SAE S1.04 - Création d’une base de données = 40%

UE2.4 :

* R2.06 – Exploitation d’une base de données (SQL avancées)
* R2.08 – Gestion = 62 %
* R2.12 – Anglais
* SAE S2.04 – Exploitation d’une base de données = 38 %

# Introduction aux bases de données et SQL

Sommaire :

1. Introduction
2. Modèle relationnel
3. Théorie de normalisation
4. Modèle entité / association
5. Introduction à SQL

## Bases de données Relationnelles : Introduction

**Définitions**: Une base de données est un réservoir structuré de données partageables par plusieurs utilisateurs.

**L’approche BD**: Elle permet de répondre à 3 questions :

Q1) Comment organiser (structurer) les données pour qu’on puisse les stocker

Q2) Comment mettre à jour (opérations d’ajout, de suppression et modification) de manière fiable

Q3) Comment consulter de manière efficace les données

**Le modèle de Données** : Le Modèle de données est l’outil permettant d’assurer les trois fonctions de structuration, de mise à jour et de consultation des données

Le modèle le plus connu est le modèle relationnel.

**Le SGBD** : Le Système de Gestion de Bases de Données est une implantation du modèle de données (notamment le modèle relationnel) : Oracle (architecture lourde), MySQL (architecture moyenne), SQLite (architecture légère – embarquée), …

## Modèle relationnel

0/ Rappels mathématiques

* Produit cartésien (x) : Soient A et B deux ensemble finis, A x B = { (a , b) | a E A, b E B }

Exemple :

A = {a1, a2, a3}

B = {b1, b2}

A x B = {(a1, b1) , (a1, b2) , (a2, b1) , (a2, b2) …

A x B x C = {…}

n-uplet -> Tuple

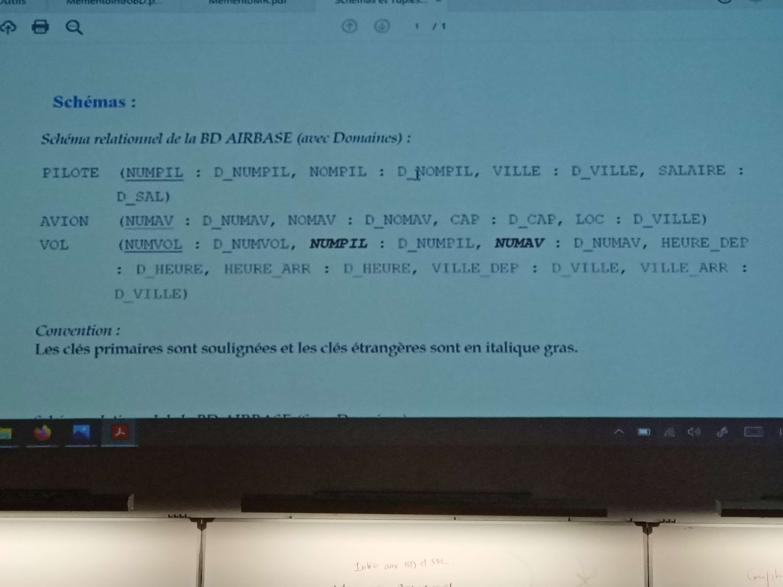
## Concept Relationnels

* **Domaine :** Ensemble de valeurs ayant une signification. Chaque domaine est caractérisé par un type scalaire et possède un nom.
* **Relation :** Sous ensemble du produit cartésien de n domaines.
* **Tuple :** n-uplet, un élément d’une relation. Il représente un objet ou un lien entre objets.
* **Attribut :** Rôle joué par un domaine dans une relation.
* **Clef primaire (Primary Key) :** Un attribut ou un groupe d’attributs identifiant de manière unique les tuples d’une relation.
* **Clef étrangère (Foreign Key) :** Un attribut ou un groupe d’attributs clef primaire dans une autre relation.
* **Schéma d’une relation :** Nom de la relation et la liste de ses attributs.
* **Extension d’une relation :** L’ensemble des tuples d’une relation.
* **Schéma relationnel :** L’union des schémas des relations ?
* **Bases de données Relationnelles :** L’union des extensions des relations.

Exemple :

Vol < D\_NUMVOL (Numvol) x D\_VILLE (VILLE\_DEP) x D\_VILLE (VILLE\_ARR) x D\_HEURE (HEURE\_DEP) x D\_HEURE (HEURE\_ARR)

Vol (Numvol, VILLE\_DEP, VILLE\_ARR, HEURE\_DEP, HEURE\_ARR) Schéma d’une relation



## Image

## Contraintes d’intégrités relationnelles

* **Contraintes de domaine :** Chaque valeur affectée à un attribut doit être dans le domaine de l’attribut.
* **Contrainte de relation/clé :** Chaque valeur d’une clef primaire doit être unique et obligatoire.
* **Contrainte de référence en ajout :** Chaque valeur affectée à une clef étrangère doit exister dans les valeurs de la clef primaire associée.
* **Contrainte de référence en suppression :** On ne peut pas supprimer un tuple si la valeur de sa clef primaire existe dans les valeurs de la clef étrangère associée.

## Algèbre Relationnelle

### Opérateurs relationnels :

PROJECTION :

RT = relation temporaire

Syntaxe : RT = PROJECTION (R / Liste des attributs)

Sémantique : RT = { t(A1, A2, …, AP) | tel que E R} A1, A2, …, AP (Liste des attributs)

Exemple : RT = PROJECTION (PILOTE | NOMPIL, SAL)

Exemple : RT = PROJECTION (R, A)

|  |
| --- |
| A |
| a1 |
| a2 |

RT =

SELECTION :

Syntaxe : RT = SELECTION (R, / Condition simple) Condition de type attribut constante

JOINTURE, DIVISION

### Opérateurs ensemblistes :

UNION, INTERSECTION, DIFFERENCE