

# 1 Base de données et SGBD

## 1.1 Base de données



### Définition 1

Une **base de données** est un ensemble d'informations structurées mémorisées sur un support persistant.

## 1.2 Système de Gestion de Bases de Données



### Définition 2

Un **Système de Gestion de Bases de Données (SGBD)** est un système informatique qui assure la gestion de l'ensemble des informations stockées dans une base de données. Il prend en charge :

- ☞ le *contrôle d'accès* aux fichiers de la base en gérant les accès concurrents ;
- ☞ la garantie de la *persistance* et de l'*intégrité* des données ;
- ☞ les *interactions* avec les applications et les utilisateurs, grâce à des langages d'interrogation et de manipulation de données comme **SQL**, en optimisant les requêtes et le stockage des données.

Un **SGBD** contrôle l'accès aux fichiers physiques de la base de données à travers une architecture *client / serveur*.

# 2 Le modèle relationnel

## 2.1 Table ou relation



### Définition 3 Table ou relation

Table / Relation postes

nom attribut / nom colonne

attribut / colonne

nuplet / ligne

| num_serie | nom_poste   | nom_modele     | fabricant | annee | nom_système  | version  |
|-----------|-------------|----------------|-----------|-------|--------------|----------|
| S2069FST  | Portable-01 | hp probook 400 | hp        | 2019  | fedora linux | 36       |
| JQ192AFH  | PORTABLE-02 | latitude e7470 | dell      | 2018  | debian       | bullseye |
| N1J4HSMY  | PORTABLE-03 | hp probook 600 | hp        | 2021  | ubuntu       | 18.04    |
| GENS27BP  | PORTABLE-04 | hp probook 400 | hp        | 2020  | debian       | buster   |
| PW5A4Z5Z  | Portable-05 | lenovo x270    | lenovo    | 2017  | debian       | buster   |
| ZM067CMA  | portable-06 | hp probook 600 | hp        | 2021  | debian       | bullseye |
| CEQFQIDJ  | PORTABLE-07 | hp probook 600 | hp        | 2021  | ubuntu       | 20.04    |
| Y4HWLYV7  | PORTABLE-08 | lenovo x260    | lenovo    | 2016  | fedora linux | 36       |
| U8VDRSNO  | Portable-09 | lenovo x270    | lenovo    | 2017  | debian       | bullseye |
| 6NW82QRC  | PORTABLE-10 | hp probook 400 | hp        | 2019  | ubuntu       | 20.04    |
| O5BHIVUN  | PORTABLE-11 | hp probook 600 | hp        | 2021  | ubuntu       | 20.04    |
| LLJ62SOI  | Portable-12 | lenovo x270    | lenovo    | 2017  | ubuntu       | 18.04    |
| DCY6QKHW  | portable-13 | hp probook 400 | hp        | 2018  | debian       | bullseye |
| OLUGMXT7  | PORTABLE-14 | hp probook 400 | hp        | 2019  | ubuntu       | 20.04    |

Dans le *modèle relationnel* (apparu en 1970 dans un article de **E.F. Codd**), une base de données est organisée sous la forme d'une ou plusieurs **tables**, appelées **relations** dans la formalisation mathématique.

| Représentation par relation | Représentation par table |
|-----------------------------|--------------------------|
| Relation                    | Table                    |
| Nuplet                      | Ligne                    |
| Nom d'attribut              | Nom de colonne           |
| Valeur d'attribut           | Cellule                  |
| Domaine                     | Type                     |



## Définition 4 Schéma relationnel

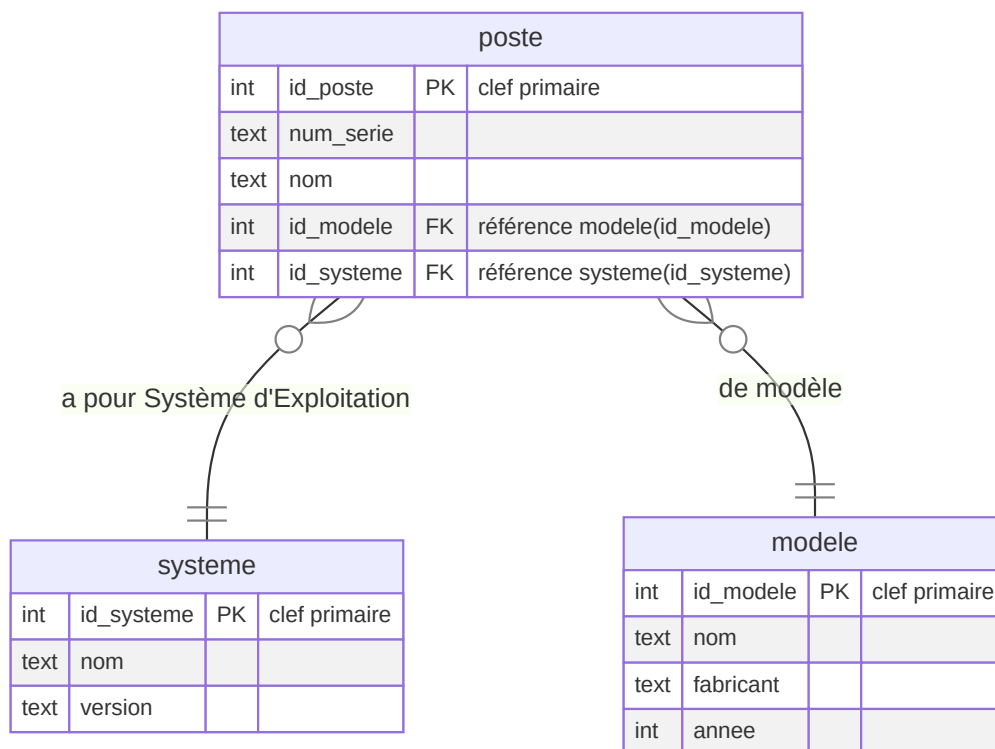
Une relation peut être décrite par : son **nom**, un nom distinct pour chaque **attribut** et le **domaine** de valeur de chaque attribut.

Cette description peut être résumée dans un **schéma relationnel**.

Par exemple la **table** du graphique 1 est une **relation**, nommons la *postes*, de schéma :

```
postes(num_serie: TEXT, nom_poste: TEXT, nom_modele: TEXT, fabricant: TEXT,
annee: INT, nom_systeme: TEXT, version: TEXT)
```

Le **schéma relationnel d'une base** constituée de plusieurs relations est l'ensemble des schémas de ces relations. On peut le représenter sous la forme d'un diagramme faisant apparaître les dépendances entre les différentes relations.



## 2.2 Qualité d'une base de données relationnelle



### Propriété 1 *Qualité d'un schéma relationnel*

Un **schéma relationnel** d'une base de données doit vérifier certains principes :

- ☞ *Unicité* : chaque **relation** ne peut contenir deux **nuplets** identiques ;
- ☞ *Contrainte de domaine* : la valeur d'un **attribut** doit appartenir au **domaine** de l'attribut ;
- ☞ *Atomicité* : une valeur d'un **attribut** est une donnée atomique, elle ne peut contenir d'autres données. Les **types** des attributs sont donc simples (entier, flottant, chaînes de caractères, booléens ...) et ne peuvent être structurés (listes ...) ;
- ☞ *Non redondance* : une donnée ne doit être représentée qu'une fois dans une table et des données identiques ne doivent pas se trouver dans des tables différentes. Une entité du monde réel ne doit être représentée que par un seul **nuplet**.

## 2.3 Clef primaire, clef étrangère



### Définition 5 *Clef d'une relation*

- ☞ Étant donné une relation R, une **clef** de R est un attribut (ou un ensemble d'attributs) de R tel que :
  - **Condition 1** : l'attribut a une valeur pour tout nuplet de la relation R ;
  - **Condition 2** il ne peut exister deux nuplets de R avec la même valeur pour cet attribut.

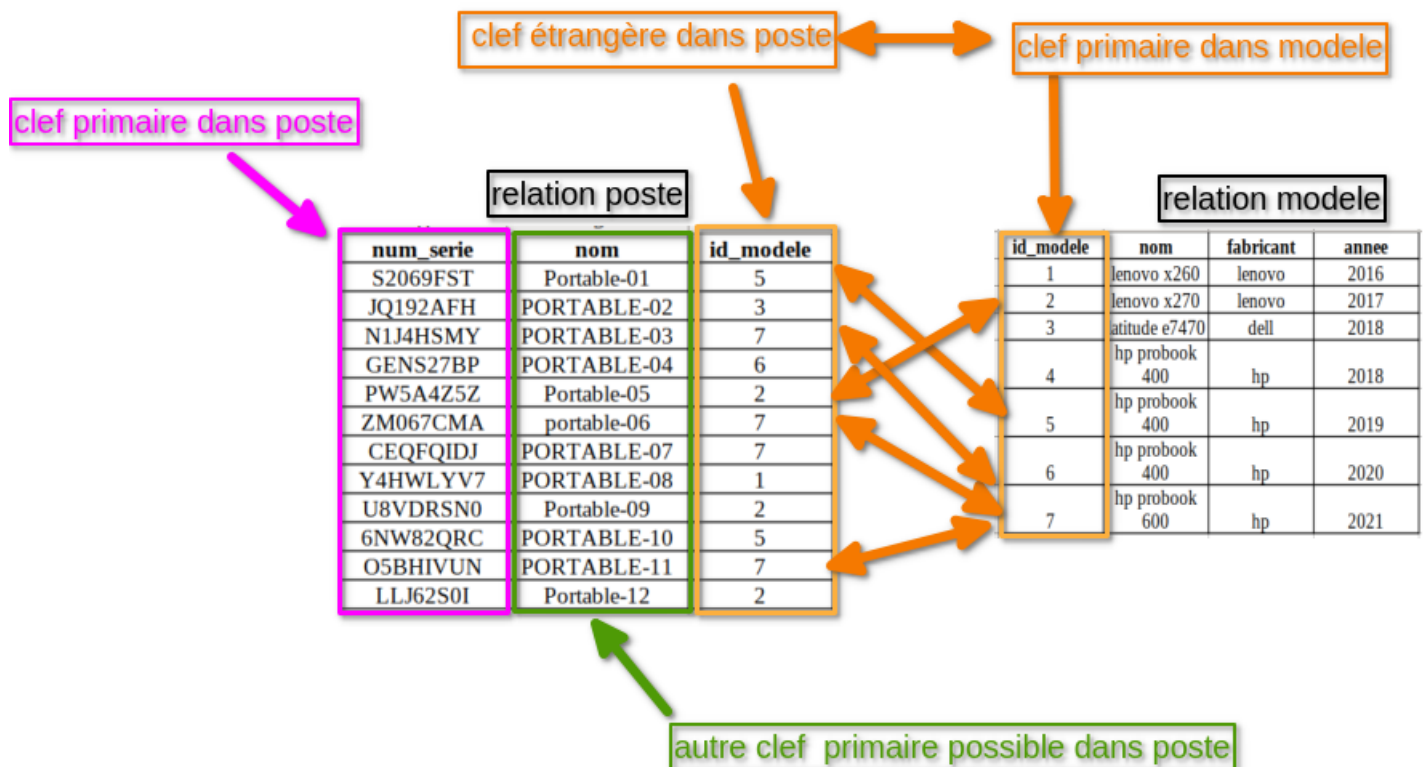
Si on parle de table, une **clef** d'une table est une colonne qui ne contient pas valeur non renseignée et telle qu'à un instant donné, il ne peut exister plus d'une ligne dans la table ayant une valeur fixée pour cette colonne.

- ☞ La connaissance de la valeur des attributs d'une clef suffit à déterminer complètement le nuplet (la ligne) dans la relation (la table).
- ☞ Le choix d'une clef minimale en nombre d'attributs détermine une **clef primaire** pour la relation.



### Définition 6 *Clef étrangère d'une relation*

Étant donné deux relations R et S, une **clef étrangère** de R est un attribut de R qui référence une clef primaire de S, c'est-à-dire qu'une valeur de la clef étrangère pour un nuplet de R doit être une valeur de clef primaire pour un nuplet de S.



## 2.4 Contraintes d'intégrité



### Propriété 2 Contraintes d'intégrité

Lors de la création d'une base de données relationnelle, on doit déclarer son schéma en précisant pour chaque relation les domaines de ses attributs, la ou les clefs primaires et les éventuelles clefs étrangères. On définit ainsi des contraintes qui seront vérifiées par le SGBD lors de chaque transaction sur la base :

- ☞ *Contrainte de domaine* : la valeur d'un **attribut** doit toujours appartenir au **domaine** de cet attribut;
- ☞ *Contrainte d'unicité (ou de relation)* : une valeur de **clef** ne peut apparaître qu'une fois dans une **relation**;
- ☞ *Contrainte d'intégrité référentielle* :
  - la valeur d'une **clef étrangère** doit toujours être également une des valeurs de la clef référencée;
  - une clef étrangère ne peut être une valeur qui n'est pas clé primaire de la relation à laquelle on se réfère;
  - une valeur de clef primaire ne peut être changée dans la relation primaire si le nuplet correspondant possède des nuplets liés par clef étrangère dans une autre relation;
  - un nuplet de la relation primaire ne peut être effacé s'il possède des nuplets liés dans une autre relation de la base par une clef étrangère.