

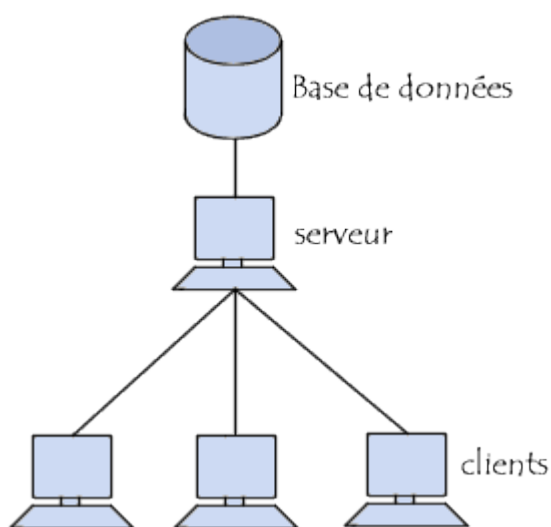
PARTIE I. **INTRODUCTION AUX BASES DE DONNEES**

1.1. CONCEPT DE BASE ET NOTION DE SGBD

En informatique de gestion, les bases de données ont connu depuis longtemps un développement et ont pris une place essentielle dans les systèmes informatiques pour **stocker et bien gérer les masses de données relatives aux activités des entreprises (financières, commerciales, techniques, production etc)**.

Ce concept a été maintes fois défini de façons différentes et on peut proposer la définition suivante (cf Delobel).

Une base de données (en anglais DB : Database) représente un ensemble structuré de données enregistrées sur des supports physiques accessibles par l'ordinateur pour satisfaire simultanément plusieurs utilisateurs de façon sélective et en temps opportun. Ces données doivent pouvoir être utilisées par des programmes, par des utilisateurs différents. Ainsi, la notion de base de données est généralement couplée à celle de **réseau**, afin de pouvoir mettre en commun ces informations, d'où le nom de **base**.



Par exemple, on donne une fiche des commandes des clients d'une société commerciale.

N°Cli	Nom	Adres	N°Pro	Design	PU.	Qte
C1	Rabe	Tana	P1	Riz	100	10
C1	Rabe	Tana	P2	Blé	200	15
C1	Rabe	Tana	P3	Pomme	300	20
C1	Rabe	Tana	P4	Arachide	400	10
C1	Rabe	Tana	P5	Fruit	500	40
C2	Jean	Diego	P1	Riz	100	50
C2	Jean	Diego	P2	Blé	200	60
C2	Jean	Diego	P3	Pomme	3000	70
C3	Rivo	Tuléar	P6	Prune	1000	20

Remarques :

Cette fiche présente des inconvénients. En effet,

- 1) il y a des répétitions (redondances) d'informations par exemple, les code, nom et adresse des clients, de même, les code, désignation et prix des produits
- 2) Si un même client commande n produits, les informations concernant ce client se répètent n fois dans la fiche
- 3) Si un même produit est commandé par n clients, les informations concernant ce produit se répètent n fois dans la fiche
- 4) Si on modifie un nom d'un client, on le modifie dans toutes les lignes de la fiche
- 5) Si on supprime la commande du client C3, les informations concernant le produit P6 sera perdu.
- 6) Si on ajoute une commande d'un nouveau client C4, celui-ci doit commander un produit de code donné qui ne peut pas être nul.

Pour remédier à ces inconvénients, on peut décomposer la fiche initiale en 3 sous-fiches CLIENT, PRODUIT et COMMANDE suivantes :

FICHE CLIENT

N°CLI	Nom	Adres
C1	Rabe	Tana
C2	Jean	Diego
C3	Rivo	Tuléar

FICHE PRODUIT

N°PRO	Designation	Prix U.
P1	Riz	1000
P2	Blé	2000
P3	Pomme	3000
P4	Arachide	4000
P5	Fruit	5000
P6	Prune	1000

FICHE COMMANDE

N°CLI	N°PRO	Qtecom
C1	P1	10
C1	P2	15
C1	P3	20
C1	P4	10
C1	P5	40
C2	P1	50
C2	P2	60
C2	P3	70
C3	P6	20

Dans ce cas, on peut insérer un nouveau client sans qu'il commande un produit et modifier le

nom d'un client une seule fois dans la fiche CLIENT et on peut faire la même opération dans la fiche PRODUIT.

L'ensemble de ces 3 fiches bien décomposées (structurées) forme **une base de données**.

Une base de données est alors une entité dans laquelle il est possible de stocker des données de façon structurée et avec le moins de redondance possible.

1.2. UTILITE ET GESTION D'UNE BASE DE DONNEES

Une base de données est faite pour enregistrer des informations, des faits, des événements qui surviennent dans la vie d'un organisme et pour les restituer à la demande ou bien tirer des conclusions en rapprochant plus faits élémentaires les uns des autres.

Une base de données permet de mettre des données à la disposition d'utilisateurs pour une consultation, une saisie ou bien une mise à jour, tout en s'assurant des droits accordés à ces derniers.

Une base de données peut être locale, c'est-à-dire utilisable sur une machine par un utilisateur, ou bien **répartie ou distribuée**, c'est-à-dire que les informations sont stockées sur des machines distantes et accessibles par réseau. Une base peut être aussi :

embarquée : Une base de données *embarquée* (anglais *embedded*) est un SGBD sous forme de composant logiciel (fichier) qui peut être incorporé dans un logiciel applicatif, contrairement à un SGBD client-serveur dans lequel un processus traite les requêtes

spatiale : utilisée par les applications informatiques SIG. Ce type de bd permet le stockage d'informations géométriques telles que des points, des lignes, des surfaces et des volumes.

NoSQL : conçu avec l'apparition de l'Internet dans les années 2000 : de services Internet a nécessité des moyens techniques adaptés à des besoins sans précédent quant au nombre d'utilisateurs et à la quantité de données. Prévus pour la répartition de charge (anglais *load balancing*), NoSQL ont fait le compromis de ne pas mettre en œuvre certaines fonctionnalités classiques des SGBD en vue d'obtenir la puissance de calcul et la scalabilité nécessaire aux populaires services web de e-commerce, de recherche ou de réseau

L'avantage majeur de l'utilisation de bases de données est la possibilité de pouvoir être accédées par plusieurs utilisateurs simultanément.

Par exemple, la base de données d'une société d'automobiles contiendra les informations ayant un rapport avec la gestion de la société considérée. Elle présente donc, par essence même, **un caractère de confidentialité** et ne pourra être consultée que par les membres de la société. .

La base de données de gestion universitaire archive des informations sur les enseignants, les étudiants, les enseignements et les conditions d'inscription à un enseignement. Ces informations vont pouvoir être partagées et utilisés par:

- le Service de la Scolarité qui gère les inscriptions des étudiants et la préparation des emplois du temps;
- les services financiers qui détermineront les budgets en fonction des effectifs;
- et les enseignants.

Le fait de partager ces données est le reflet d'une évolution dans l'utilisation des systèmes informatiques. On parle généralement de système d'information pour désigner toute la structure regroupant les moyens mis en place pour pouvoir partager des données. La création d'une base de données rend possible alors *la centralisation, la coordination, l'intégration et la diffusion de l'information archivée*.

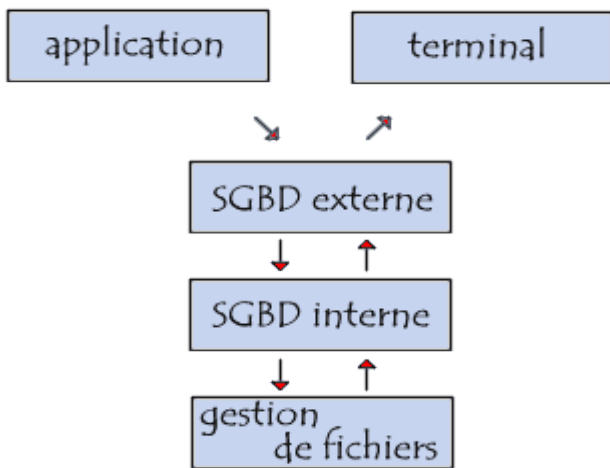
Le logiciel permettant à l'utilisateur d'interagir avec la base de données s'appelle **le Système de Gestion de Base de Données (SGBD.)** ou en anglais DBMS (Database management system)

Un SGBD *représente un ensemble coordonné de programmes qui permet de décrire, mémoriser, manipuler, traiter les ensembles de données constituant la base.*

Les données sont typiquement manipulées par un logiciel applicatif qui fait appel aux services du SGBD pour manipuler la base de données.

Ils sont utilisés pour de nombreuses applications informatiques, notamment les guichets automatiques bancaires, les logiciels de réservation, les bibliothèques numériques, le e-commerce,, les ressources humaines, la gestion de la relation client et la logistique. Les logiciels d'inventaire, les *progiciels de gestion intégrés* ou la plupart des blogs et sites web.

Les SGBD se distinguent des systèmes de fichiers par le fait qu'ils permettent la description des données(définition des noms, formats, caractéristiques) de manière séparée de leur utilisation (mise à jour et recherche).



Le SGBD peut se décomposer en trois sous-systèmes :

- le système de gestion de fichiers :
il permet le stockage des informations sur un support physique
- le SGBD interne :
il gère l'ordonnancement des informations
- le SGBD externe :
il représente l'interface avec l'utilisateur

Exemple des principaux SGBD: MSACCESS, MySQL, POSTGRESQL, ORACLE , IBM DB2 , Ingres , Interbase, Microsoft SQL server , Microsoft FoxPro , Sybase etc

1.3. PROBLEMATIQUE DE CONCEPTION D'UNE BASE DE DONNEES

Concevoir une BD représente un processus très complexe où entrent en jeu de très nombreux paramètres. Cette conception n'est possible que si l'on décompose le processus en différentes étapes.

Ainsi, la description d'une base de données peut se faire à différents niveaux permettant d'assurer les objectifs suivants:

- ⇒ indépendance structure de stockage et structure des données
- ⇒ indépendance applications et structure de données.

Pour assurer les deux types d'indépendance précédents, on s'accorde sur une décomposition en 3 niveaux :

- *niveau externe avec les schémas externes qui correspondent à des différents groupes d'utilisateurs;*
- *niveau conceptuel avec le schéma conceptuel;*
- *niveau interne ou physique avec le schéma interne ou physique .*

Dans une telle représentation, seul le niveau interne a une existence matérielle tandis que les autres correspondent en des termes abstraits.

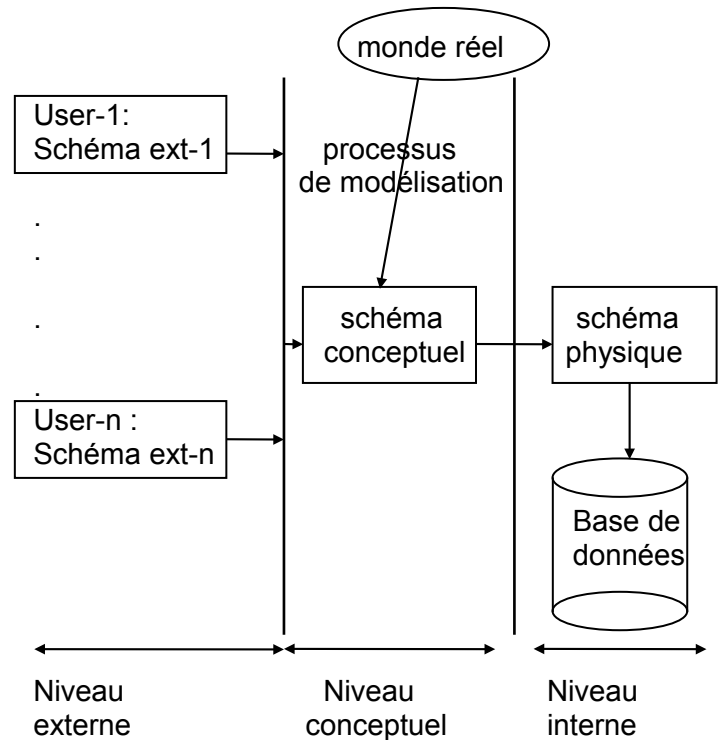


Fig 1.2. Les différents niveaux de représentation d'une base de données

♦ Le niveau conceptuel

Le niveau conceptuel est la partie fondamentale de l'architecture d'une BD. Il a pour but de décrire en termes abstraits mais fidèles une certaine réalité d'une organisation et ses processus de gestion.

Le schéma conceptuel est le résultat du processus de modélisation du monde réel en utilisant un modèle des données approprié.

Pour construire un schéma conceptuel, il est commode de regrouper les objets du monde réel en classes d'objets de même nature dont chacune sera désignée par un nom par exemple: les classes CLIENT et PRODUIT. Le modèle doit permettre aussi de décrire les liens logiques (associations) qui peuvent exister entre classes d'objets, par exemple : ACHETER est un lien entre les classes CLIENT et PRODUIT.

" Par définition, un schéma conceptuel représente donc la description des classes d'éléments de l'univers de base de données et des relations existantes entre ces classes ainsi que la description des règles de gestion et de certaines contraintes d'intégrité".

Les contraintes d'intégrité correspondent à des propriétés qui devront toujours être vérifiées dans la base quelques soient les valeurs enregistrées. Elles servent pour contrôler la validité des données de la base et assurer alors la fiabilité des informations, par exemple :

- les notes doivent être comprises entre 0 et 20.
- les étudiants inscrits sont âgés pas plus de 25 ans.

En général, le schéma conceptuel donne une vision globale de la BD répondant aux besoins de l'entreprise et il sera élaboré sans tenir compte de l'aspect technique (logiciel et matériel), ni du mode de stockage et ni du mode d'accès et sans se référer aux conditions d'utilisation par tel ou tel traitement. Ce schéma décrit la structure de la base indépendamment de son implantation

♦ **Le niveau interne**

Le niveau interne a pour but de spécifier comment les données seront stockées sur les supports physiques (organes périphériques) de l'ordinateur. Par exemple, les données de la base sont enregistrées sur un disque dur ou sur les bandes magnétiques. Ce niveau décrit alors le modèle de stockage des données et les fonctions d'accès

A ce niveau, les classes d'objets ainsi que les liens décrits au niveau conceptuel se transforment en fichiers physiques dont l'ensemble formera la base de données de l'entreprise.

♦ **Le niveau externe**

Ce niveau correspond aux différentes vues des utilisateurs. Chaque schéma externe donne une vue sur le schéma conceptuel à une classe d'utilisateurs

Chaque groupe de travail utilisant les données possède une description des données perçues de la manière dont il les voit dans ses programmes d'application selon les besoins.

Finalement, il tient à souligner que *pour une base particulière, il existe un seul schéma interne avec un modèle unique et un seul schéma conceptuel avec un modèle unique, mais il existe par contre, en général, plusieurs schémas externes avec plusieurs modèles.*

1.4. Indépendance données - programmes

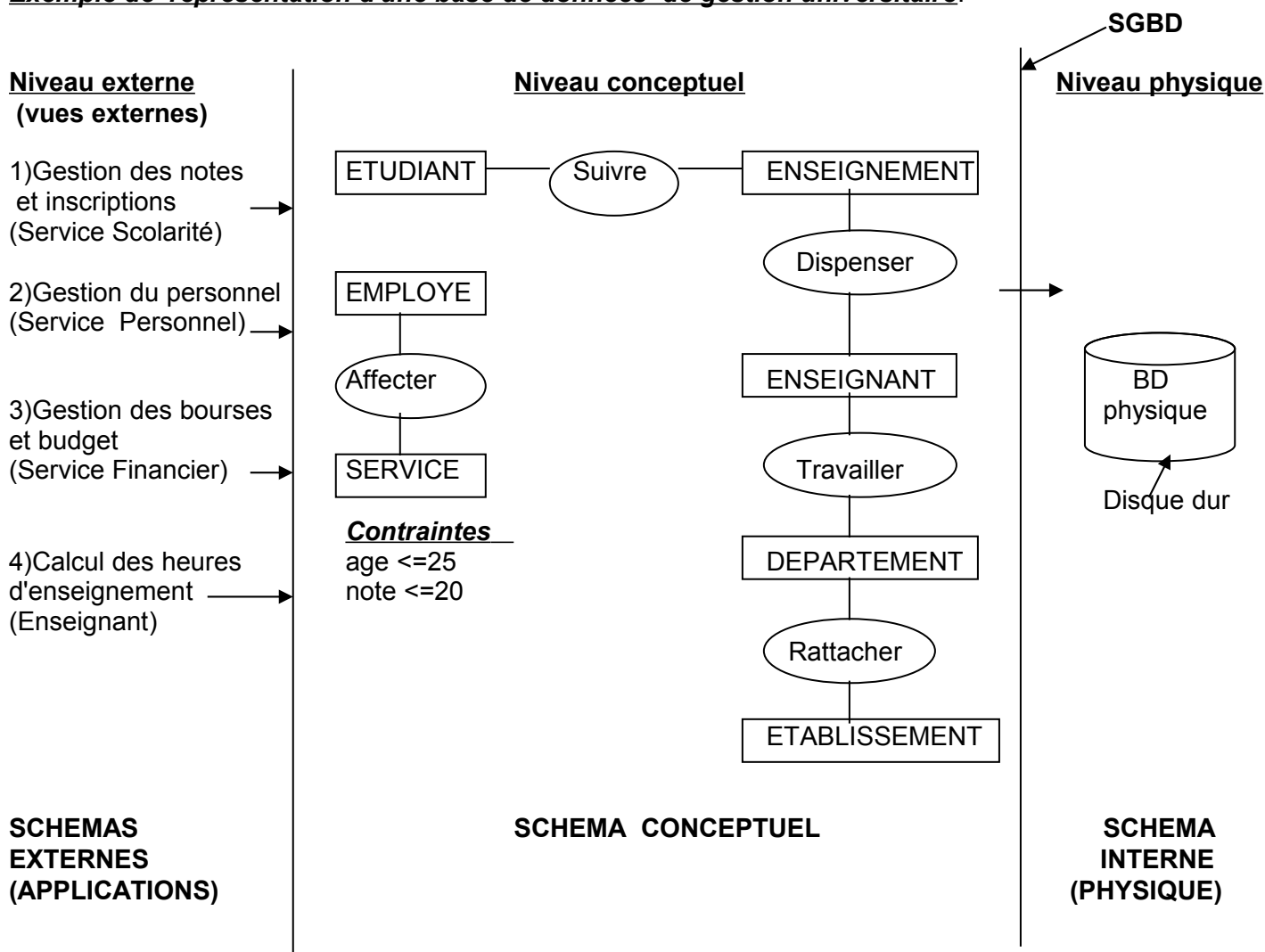
L'architecture à trois niveaux permet de supporter le concept d'indépendance données - programmes, c'est à dire la capacité de modifier le schéma de la base de données à un niveau donné, sans remettre en cause le schéma aux niveaux supérieurs :

- **indépendance logique** : on peut changer le niveau conceptuel sans remettre en cause les schémas externes ou les programmes d'application. L'ajout ou le retrait de nouveaux

concepts ne doit pas modifier des éléments qui n'y font pas explicitement référence,

- **indépendance physique** : on peut changer le schéma physique sans remettre en cause le schéma conceptuel (et les schémas externes). On peut modifier l'organisation physique des fichiers, rajouter ou supprimer des méthodes d'accès.

Exemple de représentation d'une base de données de gestion universitaire:



a) ETUDIANT , ENSEIGNEMENT ,ENSEIGNANT, DEPARTEMENT, ETABLISSEMENT, SERVICE ET EMPLOYE sont **des classes d'objets ou entités**

b) SUIVRE ,DISPENSER ,TRAVAILLER , RATTACHER, AFFECTER sont **des associations**

Les critères de choix du SGBD les plus importants sont la fiabilité, la performance, la conformité aux normes, la palette d'ordinateurs supportés, et la facilité d'utilisation

1.5. FONCTIONNALITES D'UN SGBD

En général, un SGBD doit atteindre les objectifs suivants

1- Description

Le SGBD doit mettre à la disposition des utilisateurs un outil pour décrire l'ensemble des données qui seront stockées dans la base de données . Il s'agit d'un langage de définition des données **(LDD) qui est un langage** descriptif supportant un modèle et permettant de décrire les données d'une base de données d'une manière assimilable par une machine. La description d'un ensemble de données particulier, correspondant à une application, à l'aide d'un LDD donne naissance à un **schéma**.

2- Contrôler la redondance d'informations

Dans une base de données ,une information doit être physiquement implantée une seule fois : il ne pourra pas y avoir duplication de l'information. L'élimination de la redondance permet d'avoir une vision globale des données et notamment de mettre en évidence les liens entre les données.

3- Partage des données

Une base de données doit permettre d'accéder la même information par plusieurs utilisateurs en même temps. Le SGBD doit inclure un mécanisme de contrôle de la concurrence basé sur des techniques de verrouillage des données (pour éviter par exemple qu'on puisse lire une information qu'on est en train de mettre à jour).

Le partage des données se fait également par la notion de vue utilisateur, qui permet de définir pour chaque classe d'utilisateurs la portion de la base

de données qui l'intéresse (et dans la forme qui l'intéresse).

3. Indépendance des programmes par rapport aux données

L'indépendance des données-programmes correspond à la nécessité de diminuer les coûts de maintenance des programmes d'application: les modifications apportées à la structure seront introduites par un changement du monde réel et non pas pour une application particulière: ceci permet à de nombreuses applications de partager les mêmes données.

4- Gérer les autorisations d'accès

Une base de données étant multi-utilisateurs, se pose le problème de la confidentialité des données. Des droits doivent être gérés sur les données, droits de lecture, mise à jour, création, ... qui permettent d'affiner la notion de vue utilisateur.

Il apparaît nécessaire, lorsque une base de données est partagée par un nombre d'utilisateurs, de s'assurer que les informations confidentielles ne pourront être accessibles qu'aux personnes réellement autorisées. Pour cela, le SGBD doit offrir des mécanismes permettant de vérifier les **droits d'accès** des utilisateurs. Par exemple, une personne travaillant dans le service de la scolarité ne devra pas être autorisée à accéder aux informations dans le fichier des enseignants contenant, par exemple, l'historique de ses services et son indice de rémunération.

5- Offrir des interfaces d'accès multiples

Un SGBD doit offrir plusieurs interfaces d'accès, correspondant aux différents types d'utilisateurs pouvant s'adresser à lui. Cette fonction a pour but d'offrir à l'utilisateur une interaction avec la base de données sous forme d'un dialogue pour rechercher et sélectionner et modifier des données. On trouve des interfaces orientées utilisateur final (langages de requêtes déclaratifs comme SQL avec mise en oeuvre graphique, interface de type formulaire, ...) ou bien orientées programmeurs d'applications (interface avec des langages de programmation classiques).

6- Vérifier les contraintes d'intégrité

Plus la masse d'informations enregistrées dans une base de données est grande, plus le risque que la donnée enregistrée soit erronée par rapport à la réalité est grand. Pour diminuer ce risque, le SGBD doit offrir à l'utilisateur la possibilité de définir des règles qui permettent de maintenir l'intégrité de la base de données. Ces règles sont appelées "**contraintes d'intégrité**". Elles correspondent à des propriétés qui devront toujours être vérifiées dans la base, quelques que soient les valeurs enregistrées.

Par exemple, dans la base universitaire décrite précédemment, on devra s'assurer que les notes des étudiants obtenues dans un enseignement sont comprises entre 0 et 20. Ou bien qu'un étudiant ne peut être inscrit à un enseignement que s'il est âgé moins de

25 ans et possède certains diplômes. Ces contraintes d'intégrité correspondent à des contrôles que le SGBD pourra effectuer sur les données.

Un SGBD doit veiller à ce que les applications respectent ces règles lors des modifications des données et doit ainsi assurer la **cohérence** des données.

Un schéma de base de données se compose d'une description des données et de leurs relations ainsi que d'un ensemble de contraintes d'intégrité. Une contrainte d'intégrité est une propriété de l'application à modéliser qui renforce la connaissance que l'on en a. On peut classer les contraintes d'intégrité, en contraintes structurelles (un employé a un chef et un seul par exemple) et contraintes dynamiques (un salaire ne peut diminuer). Les SGBD commerciaux supportent automatiquement un certain nombre de contraintes structurelles, mais ne prennent pas en compte les contraintes dynamiques (elles doivent être codées dans les programmes d'application).

8- Assurer la sécurité et la reprise après panne

Une base de données est souvent vitale dans le fonctionnement d'une organisation, et il n'est pas tolérable qu'une panne puisse remettre en cause son fonctionnement de manière durable. Les SGBD fournissent des mécanismes pour assurer cette sécurité. Le premier mécanisme est celui de transaction qui permet d'assurer un comportement atomique à une séquence d'actions (elle s'effectue complètement avec succès ou elle est annulée). Une transaction est une séquence d'opérations qui fait passer la base de données d'un état cohérent à un nouvel état cohérent. L'exemple typique est celui du débit-crédit pour la gestion d'une carte bancaire. Ce mécanisme permet de s'affranchir des petites pannes (style coupure de courant).

En ce qui concerne les risques liés aux pannes disques, les SGBD s'appuient sur un mécanisme de journalisation qui permet de régénérer une base de données automatiquement à partir d'une version de sauvegarde et du journal des mouvements.

Les SGBD possèdent des outils qui servent à créer des compte-rendus (*reports*), des écrans pour la saisie des informations, importer et exporter les données de et vers la base de données, et manipuler le catalogue.

Ces outils sont utilisés par l'administrateur de bases de données pour effectuer des sauvegardes, des restaurations de données, autoriser ou interdire l'accès à certaines informations, et effectuer des modifications du contenu de la base de données - création, lecture, modification et suppression d'informations, abrégé **CRUD** (anglais *create, read, update, delete*).

Ces outils servent également à surveiller l'activité du moteur et effectuer des opérations de *tuning*

1.6. MISE EN OEUVRE D'UN SGBD

Il existe plusieurs catégories d'utilisateurs des SGBD, parmi lesquelles il y a l'administrateur de bases de données (*database administrator* abr. *DBA*), le concepteur de base de données, le développeur, ainsi que les utilisateurs. Dans un SGBD à trois niveaux de schéma, les administrateurs de données ont trois rôles :

- ⇒ **administrateur d'entreprise** correspondant à la définition du schéma conceptuel
- ⇒ **administrateur d'application** correspondant à la définition des schémas externes et des règles de correspondance externe à conceptuel
- ⇒ **administrateur de base de données** correspondant à la définition du schéma interne et des règles de correspondance interne à conceptuel.

L'administrateur de bases de données est un expert en SGBD, il s'occupe d'installer et de maintenir le SGBD ainsi que les outils annexes qui l'accompagnent. Il est la personne responsable de l'intégrité, de la sécurité, de la disponibilité des informations contenues dans les bases de données ainsi que de la performance du SGBD. Il protège les informations contre les accidents dus à des mauvaises manipulations, des erreurs de programmation, des utilisations malveillantes ou des pannes qui entraîneraient des détériorations du contenu des bases de données. Pour ce faire, l'administrateur de base de données autorise ou interdit l'accès aux informations et surveille l'activité du SGBD. Il effectue régulièrement des copies de sauvegarde en vue de permettre la [récupération de données](#) qui ont été perdues ou détériorées et effectue des réglages de *tuning* en vue d'améliorer la performance du SGBD. L'administrateur utilise les outils d'administration de base de données ou le langage de commande du SGBD

Le concepteur de base de données (anglais *database designer*) est la personne qui identifie les informations qui seront enregistrées dans la base de données, les relations entre ces informations et les contraintes telles que la présence ou non de redondance.

Les développeurs créent des logiciels applicatifs dans un langage de programmation de haut niveau. Chaque logiciel cible une activité en particulier - par exemple retrouver des livres dans une bibliothèque - et est destiné aux autres usagers du SGBD.

Les utilisateurs autorisés par l'administrateur peuvent accéder aux données et les manipuler, rechercher en utilisant le langage de commande du SGBD ou à travers un logiciel applicatif.