Bases de données

1 Bibliographie

[1] https://www.apprendre-en-ligne.net/info/database/index.html

[2] http://cerig.efpg.inpg.fr/tutoriel/bases-de-donnees

[3] www.developpez.com
https://cyril-gruau.developpez.com/uml/tutoriel/ConceptionBD

[4] INRIA Canal U Bases de données relationnelles
https://www.canal-u.tv/producteurs/inria/cours_en_ligne/bases_de_donnees_relationnelles

[5] CNAM Bases de données relationnelles : Comprendre pour maîtriser
https://www.fun-mooc.fr/fr/cours/bases-de-donnees-relationnelles-comprendre-pour-maitriser/

[6] FLOT/MOOC : Bases de données relationnelles
http://flot.sillages.info/?portfolio=bases-de-donnees-relationnelles

[7] Phlippe Rigaux : Cours de bases de données – Modèles et langages
http://sql.bdpedia.fr/

2 Introduction

Dans les entreprises, on manipule souvent des données ayant la même structure. Prenons l'exemple de la liste des membres du personnel : pour chaque personne, on enregistre le nom, le prénom, le sexe, la date de naissance, l'adresse, la fonction dans l'entreprise, etc. Toutes ces données ont la même structure ; si elles sont gérées par des moyens informatiques, on dit qu'elles constituent une …

Définition

Une base de données (**BD**) est .

On rajoute parfois deux conditions supplémentaires à la définition précédente

- : la base contient toutes les informations requises pour le service que l'on en attend.
- : la même information n'est présente qu'une seule fois (pas de doublons)

Reprenons l'exemple de la base de données du personnel. Elle est utilisée pour la paye mensuelle, pour l'avancement, les mutations, les mises à la retraite, etc. L'exhaustivité est indispensable pour le personnel, car la personne qui est absente de la base… n'est pas payée. L'unicité est importante pour l'employeur, car la personne qui est enregistré deux fois… risque de toucher double paye

L'organisation logique des données se fait selon un modèle de données et la structure physique des fichiers comporte des destinés à accélérer les opérations de **recherche et de tri**. Le est aujourd'hui le plus utilisé parce qu'il permet .

Le logiciel qui manipule les bases de données est appelé . Il permet d'organiser, de contrôler, de consulter et de modifier

la base de données. Les opérations sont parfois formulées dans un [redacted] tel que [redacted], qui est le plus connu et employé pour les modèles relationnels.

Quatre exemples de bases de données

Tous les exemples ci-dessous proviennent du site www.apprendre-en-ligne.net.

- ✓ **Seshat**

Seshat est une base de données de mathématiciens. Comme vous pourrez le constater, cette base de données permet de retrouver rapidement un mathématicien selon divers critères : nom, année de naissance, année de mort, jour de naissance, ville de naissance ou de mort, signe astrologique, pays, etc. Elle permet aussi de trouver tous les contemporains d'un mathématicien. La mise en page peut aussi se changer aisément et est la même pour tous les mathématiciens.

En fait, les pages que vous verrez n'existent pas réellement : elles sont construites à chaque requête d'après les informations figurant dans la base de données. De même, la chronologie des mathématiciens est construite en visitant toute la base de données. Cela signifie que si l'on ajoute ou enlève un mathématicien à la base de données, le tableau chronologique sera automatiquement modifié lors de la prochaine requête.

- ✓ **Les blogs**

Les blogs sont aussi basés sur une base de données. Les contenus des billets sont stockés dans un champ, ainsi que des informations comme le sujet, le nombre de lectures, la date, etc. Quand on ouvre le blog, le système affiche les *n* derniers billets, par ordre antéchronologique.

- ✓ **Les quiz interactifs**

Pour les quiz aussi, les questions sont stockées dans une base de données, avec les réponses, la difficulté, le chapitre concerné, etc. Les questions sont ensuite choisies par le système au hasard ou selon certains critères.

- ✓ **Le défi Turing**

Le défi Turing est une base de données d'exercices de programmation. Les questions sont contenues dans une base de données, ainsi que les membres du défi. On garde en mémoire les problèmes résolus par chaque membre, ce qui permet d'établir un classement. Enfin, un forum permet aux membres de partager leurs solutions : c'est aussi géré avec une base de données.

3 Un 1er exemple pour tout expliquer rapidement

Replongeons-nous avant les années 1980, lorsque la téléphonie mobile n'existait pas. Nous allons imaginer un opérateur de téléphonie qui veut créer une base de données pour gérer ses clients.

Les clients peuvent obtenir plusieurs numéros de téléphone auprès de cet opérateur. Pour chacun de ces numéros, l'opérateur enregistrera les informations nécessaires pour établir une facture mensuelle détaillée.

Quelles sont ces données nécessaires ?

Il faudra tout d'abord identifier de manière non équivoque le client : on enregistrera donc son nom, son prénom et son adresse. On stockera aussi le numéro de téléphone qu'il a utilisé (rappelons qu'il peut en avoir plusieurs ; on fera une facture par numéro).
Il faudra aussi connaître, pour calculer les prix de la communication, le numéro appelé, la date et l'heure de l'appel (pour établir la facture détaillée) ainsi que la durée de l'appel. Enfin, comme le prix de l'appel peut varier selon l'heure et le numéro appelé, on mentionnera le tarif appliqué.

3.1 Première approche : un tableur

Une première idée pour implémenter une base de données consiste à utiliser un tableur. En effet, une base de données « plate » peut se voir comme un simple tableau:

- les colonnes représenteront des _____ (nom, prénom, numéro de téléphone, etc.),
- les lignes des _____ (ici des appels).

Par exemple, voici un extrait de cette base de données :

Nom	Prénom	Adresse	Numéro de tél.	No appelé	Date et heure de l'appel	Durée de l'appel (sec)	Tarif (CHF/min)
Camé	Léon	Petit-Chêne 3 2900 Porrentruy	0324660010	00333246634217	10.2.2010 14:32	455	0.70
Camé	Léon	Petit-Chêne 3 2900 Porrentruy	0324660010	0324711230	17.2.2010 18:37	62	0.20
Camé	Léon	Petit-Chêne 3 2900 Porrentruy	0324669987	0324711230	23.2.2010 9:21	145	0.20
Camé	Léon	Grand-Rue 49 2800 Delémont	0324221022	0214537124	12.2.2010 19:01	302	0.10
...	...	...	...	...	...	...	

L'avantage de cette méthode est sa simplicité : tout ce qui concerne un appel tient sur une ligne, et tous les appels sont répertoriés dans une table.
Cette approche a aussi des inconvénients :

- ✓ Les informations sont _____ : l'adresse du client apparaît plusieurs fois, encombrant inutilement la mémoire. Il aurait été plus judicieux d'avoir une

autre table où l'on aurait enregistré les coordonnées des clients une fois pour toutes.

- ✓ Pour identifier un client, on a besoin de trois champs. Il aurait été plus simple d'associer à chaque client un (ce que toutes les entreprises font).
- ✓ Si un utilisateur change d'adresse, il faut reporter ce changement sur chaque ligne où il apparaît, sous peine d'avoir une table incohérente.
- ✓ Un tableur n'est tout simplement pas fait pour gérer une grande base de données.

3.2 Deuxième approche : base de données relationnelle

Une autre solution est d'utiliser
On parle alors de . La liaison de différentes tables se fera en utilisant un logiciel de gestion de bases de données (**SGBD**).
Reprenons l'exemple de l'entreprise de téléphonie. On va utiliser trois tables au lieu d'une : la première contiendra les coordonnées des clients, la seconde les numéros de téléphone des clients, et la troisième la liste des appels.

Id_client	Nom	Prénom	Adresse
156	Camé	Léon	Grand-Rue 49 2800 Delémont
10234	Camé	Léon	Petit-Chêne 3 2900 Porrentruy
...	...	...	...

Table des clients

Numéro	Id_client
0324221022	156
0324660010	10234
0324669987	10234
...	...

Table des numéros

La table des numéros permet de faire la liaison entre un abonné et son ou ses numéros de téléphone. Elle est indispensable du fait qu'un abonné peut avoir plusieurs numéros de téléphone différents.
Si cela n'avait pas été le cas, on aurait simplement pu ajouter un champ « numéro » à la table des clients. Ce champ aurait alors pu être utilisé comme identifiant et remplacer le champ « Id_Client ».

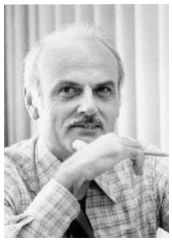
Id_appel	Numéro appelant	No appelé	Date et heure de l'appel	Durée de l'appel (sec)	Tarif (CHF / min)
123465	0324660010	00333246634217	10.2.2010 14:32	455	1.00
145674	0324660010	0324711230	17.2.2010 18:37	62	0.20
162346	0324669987	0324711230	23.2.2010 9:21	145	0.20
124369	0324221022	0214537124	12.2.2010 19:01	302	0.10
	...	...	...	...	...

Table des appels

Cette manière de faire est moins lisible au premier coup d'œil : il faut par exemple consulter deux tables pour connaître le propriétaire d'un numéro de téléphone.
Mais cet inconvénient est vite balayé par les avantages :

- ✓ Les informations ne sont plus redondantes : par exemple, l'adresse d'un client apparaît une seule fois et est donc facilement modifiable.
- ✓ Un client est identifié par un numéro, ce qui évite des confusions.
- ✓ Il est très facile de tirer la liste des clients, puisqu'ils sont tous rassemblés dans une table.

4 Les bases de données relationnelles



C'est l'article d'Edgar Frank **Codd** (1923-2003) « *A Relational Model of Data for Large Shared Data Banks* », *CACM 13, No. 6, June 1970* qui fonde le modèle relationnel, mais une première description de ce modèle avait déjà été publiée l'année précédente dans un rapport technique d'IBM : « *Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks* », *IBM Research Report RJ599*.

Tables

Dans les bases de données relationnelles, une _____ est un ensemble de données organisées sous forme d'un tableau où les colonnes correspondent à des _____ et les lignes à des _____, également appelés **entrées**.
La notion de table est apparue dans les années 1970 chez IBM avec l'algèbre relationnelle qui est une théorie mathématique en relation avec la théorie des ensembles. Cette théorie a pour but de clarifier et de faciliter l'utilisation d'une base de données.

Conception d'une table

Lors de la conception d'une base de données relationnelle, il est important de clairement **définir toutes les tables** qui la composeront et les **différentes relations qui les lient**, de manière à pouvoir dresser le **schéma conceptuel** qui permettra de

décrire le fonctionnement de la base de données avant de la mettre « physiquement » en place.
On distinguera les , qui contiendront divers champs contenant des informations, et les , qui feront les liens entre deux tables courantes.

Contenu d'une table

Par nature, **chaque colonne** d'une table, également nommé « champ », doit contenir des **données d'un même type** et ce champ doit être nommé. Chaque table est l'implémentation physique d'une relation entre les différents champs. Chaque correspondance est définie par une ligne de la table. Il y a certaines règles à respecter, notamment le fait qu'il faille mettre un pour chaque enregistrement dans la table. Il y a deux possibilités :

- ✓ Mettre un identifiant qui s'auto-incrémente au fur et à mesure des données entrées (par exemple « Id_appel » dans la « table des appels »)
- ✓ Choisir un identifiant unique (par exemple « Id_client » dans la « table des clients »).

Travail sur une table

Il y a deux niveaux de travail sur une table :

- ✓ un niveau de d'une table, qui permet de définir, lier, et contraindre les données via un langage de définition de données ;
- ✓ un niveau de d'une table, qui permet d'ajouter, supprimer, rechercher des données via un langage de manipulation de données

Actuellement, le langage standardisé pour travailler sur les tables est le . Il est utilisé avec quelques variantes sur la plupart des systèmes de gestion de bases de données. Nous en reparlerons plus loin.

5 Modèle entité-association

Le (aussi appelé « modèle entité -relation ») est un type de très utilisé pour les bases de données, notamment les bases de données relationnelles. Il a été inventé par Peter Pin-Shan **Chen** en 1975 (né en 1947

Il est destiné à clarifier l'organisation des données dans les bases de données relationnelles en notifiant :

- ✓
  - Ce sont des objets concrets que l'on peut identifier (client, livre, individu, voiture, etc.).

- On peut représenter un ensemble d'entités de la réalité par une entité type (un client pour l'ensemble des clients).
- Ces entités sont caractérisées par leurs [redacted] (pour un client : nom, prénom, adresse, ...). Parmi ces attributs, on définit un [redacted] qui va permettre de caractériser de **façon unique** l'entité dans l'ensemble (un numéro de client).

✓ [redacted]

- Elles représentent **les liens existant** entre une ou plusieurs entités.
- Elles sont caractérisées par un nom, une propriété d'association et éventuellement des attributs.

✓ [redacted]

- Le [redacted] de la relation (ou [redacted] de la relation) est le nombre d'entités qui sont impliquées dans cette relation.
- La [redacted] (d'une entité par rapport à une relation) exprime le nombre de participations possibles d'une entité à une relation. Comme c'est un nombre variable, on note la cardinalité minimum (0 ou 1) et maximum pour chaque entité. La seule difficulté est de se poser la question dans le bon sens.

  Dans notre exemple, autour de l'association POSSEDER :
  - Côté abonnés la question est : « un abonné peut posséder combien de numéros de téléphone ?>.
    La réponse est : « un abonné peut avoir de 1 à plusieurs numéros de téléphone »
  - Côté numéros de téléphone, la question est : « un numéro peut être possédé par (ou appartenir à) combien de clients ? ».
    La réponse est : « le numéro de téléphone n'appartient qu'à un client ou aucun client (numéro pas encore attribué ou plus attribué). » Il ne peut pas avoir 0 numéro, car dans ce cas il ne serait plus client de l'opérateur.

5.1 Représentation graphique

✓ Les **entités** sont représentées dans des **rectangles** et s'écrivent en lettres **majuscules**. Utiliser de préférence un **nom commun au pluriel** (par exemple : CLIENTS)

✓ **L'identifiant** d'une entité (aussi appelé « clé primaire ») est le **premier attribut cité et est souligné**. Les autres attributs sont placés à la suite.

✓ Les **relations** sont placées dans des **losanges** avec leurs attributs éventuels. Utiliser de préférence un **verbe à l'infinitif**.

✓ Les **cardinalités** sont placées **à côté de l'entité** qu'elles caractérisent.

5.2 Démarche de conception

Voici une méthode possible pour réaliser un schéma entité-association :

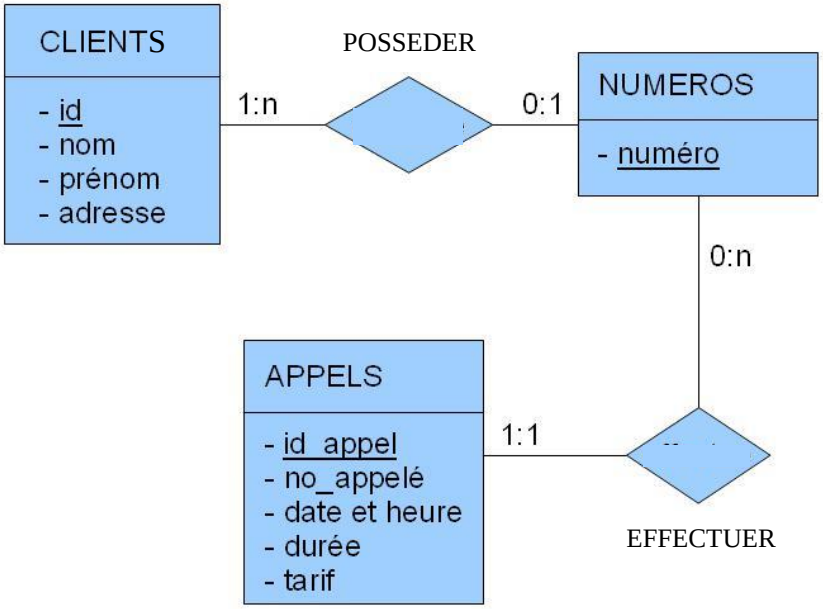
1) établir la liste des **entités**,
2) déterminer les **attributs** de chaque entité en choisissant un identifiant,
3) établir les **relations** entre les différentes entités et définir les cardinalités.

5.3 Exemple

Reprenons l'exemple de notre opérateur de téléphonie fixe.

1) Il y a 3 entités : les clients, les numéros de téléphone et les appels.
2) Les attributs :
   - Pour les clients : id, nom, prénom, adresse.
   - Pour les numéros de téléphone : le numéro.
   - Pour les appels : le numéro appelé, la date, la durée et le tarif.
3) Les relations et les cardinalités :
   - Un client [redacted] numéros.
   - Un numéro [redacted] client.
   - Un numéro [redacted] appels.
   - Un appel [redacted] numéro.

Représentation graphique



On peut classer les relations en 3 types selon leur cardinalité maximale :

- les relations (par exemple 1:1 – 0:1)
- les relations (c'est le cas des deux relations ci-dessus)
- les relations (par exemple 0:n – 1:n)

6 Traduction du schéma conceptuel en un modèle relationnel

Les règles principales de transformation d'un schéma conceptuel entité-association en un schéma relationnel sont :

6.1 Règle I

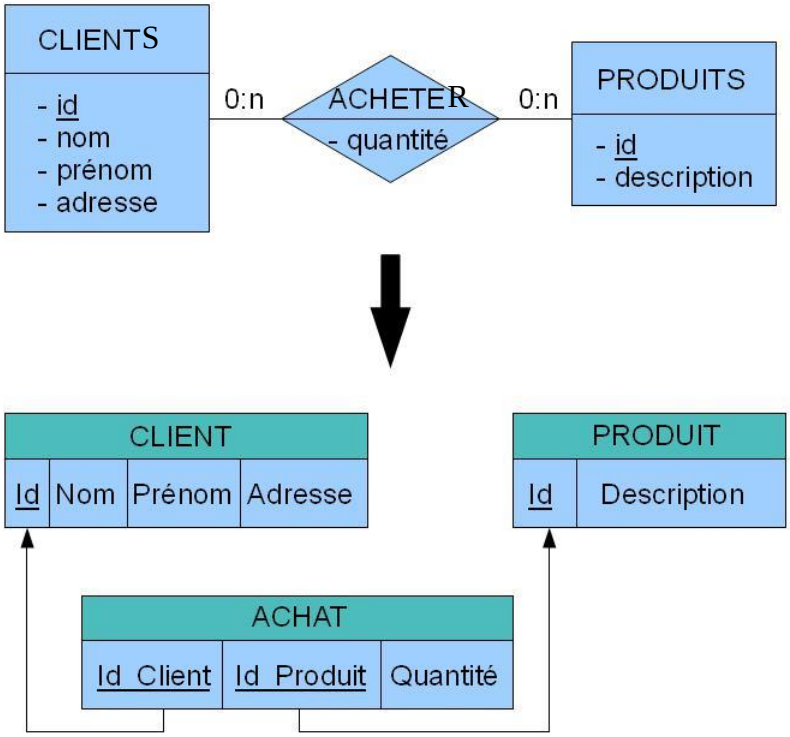
Toute est traduite en une **table relationnelle** dont les caractéristiques sont les suivantes :

- le nom de la table est le nom de l'entité ;
- la clé de la table est l'identifiant de l'entité ;
- les autres attributs de la table forment les autres colonnes de la table.

6.2 Règle II

Toute est traduite en une **table relationnelle** dont les caractéristiques sont les suivantes :

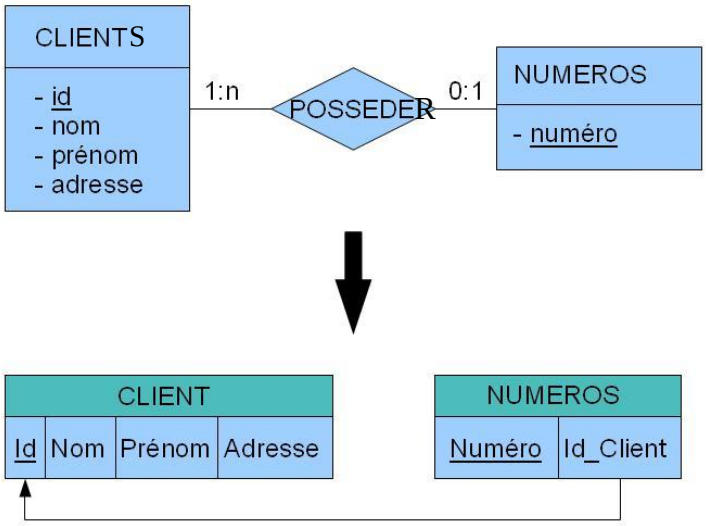
- le nom de la table est le nom de la relation ;
- la clé de la table est formée par « l'addition » des identifiants des entités participant à la relation ;
- les attributs spécifiques de la relation forment les autres colonnes de la table.



6.3 Règle III

Toute [redacted] est traduite :

1) soit par **un report de clé** (voir schéma ci-dessous) : l'identifiant de l'entité participant à la relation côté *N* est ajoutée comme colonne supplémentaire à la table représentant l'autre entité. Cette colonne est parfois appelée [redacted]. Le cas échéant, les attributs spécifiques à la relation sont eux aussi ajoutés à la même table ;



2) soit par une table spécifique dont les caractéristiques sont les suivantes :

- le nom de la table est le nom de la relation ;

- la clé de la table est l'identifiant de l'entité participant à la relation côté 1 ;
- les attributs spécifiques de la relation forment les autres colonnes de la table.

6.4 Règle IV

Toute [redacted] est traduite, au choix, par l'une des trois solutions suivantes :

- choix 1 : fusion des tables des entités qu'elle relie ;
- choix 2 : report de clé d'une table dans l'autre (voir ci-dessous) ;
- choix 3 : création d'une table spécifique reliant les clés des deux entités.