

Présentation de la notion de base de données

Table des matières

I. Contexte	3
II. Comprendre les notions de BDD relationnelle	3
A. Introduction aux bases de données relationnelles.....	3
B. Database Management System (DBMS) ou SGBD.....	4
C. Le modèle relationnel de données.....	5
III. Comprendre les notions de normes de standardisation	6
A. Les principes de base de la normalisation des BDD	6
B. La première forme normale (1FN)	6
C. La deuxième forme normale (2FN)	7
D. La troisième forme normale (3FN)	9
E. Autres formes	10
F. Exercice : Quiz	12
IV. L'essentiel	13
V. Auto-évaluation	14
A. Exercice	14
Solutions des exercices	14

I. Contexte

Durée : 1 h

Environnement de travail : aucun

Prérequis : aucun

Contexte

L'informatique a une finalité principale : créer des outils pour automatiser des processus, notamment dans une entreprise. Et pour répondre à ce besoin, une compétence est fondamentale : la gestion des données. En effet, les besoins courants sont souvent : gérer des articles, des clients, des salariés, etc. Finalement, ce ne sont que des données. Auparavant, ces bases de données étaient souvent gérées sur papier, et il a fallu informatiser tout cela. Aujourd'hui, la plupart des bases de données sont informatisées, et il nous faut accéder à ces données depuis nos logiciels.

Nous allons dans ce cours voir la création de ces bases de données informatiques, et comment nous pouvons les organiser. Nous allons notamment aborder la notion de base de données « *relationnelle* », une base de données majoritairement utilisée aujourd'hui dans n'importe quel logiciel de gestion.

II. Comprendre les notions de BDD relationnelle

A. Introduction aux bases de données relationnelles

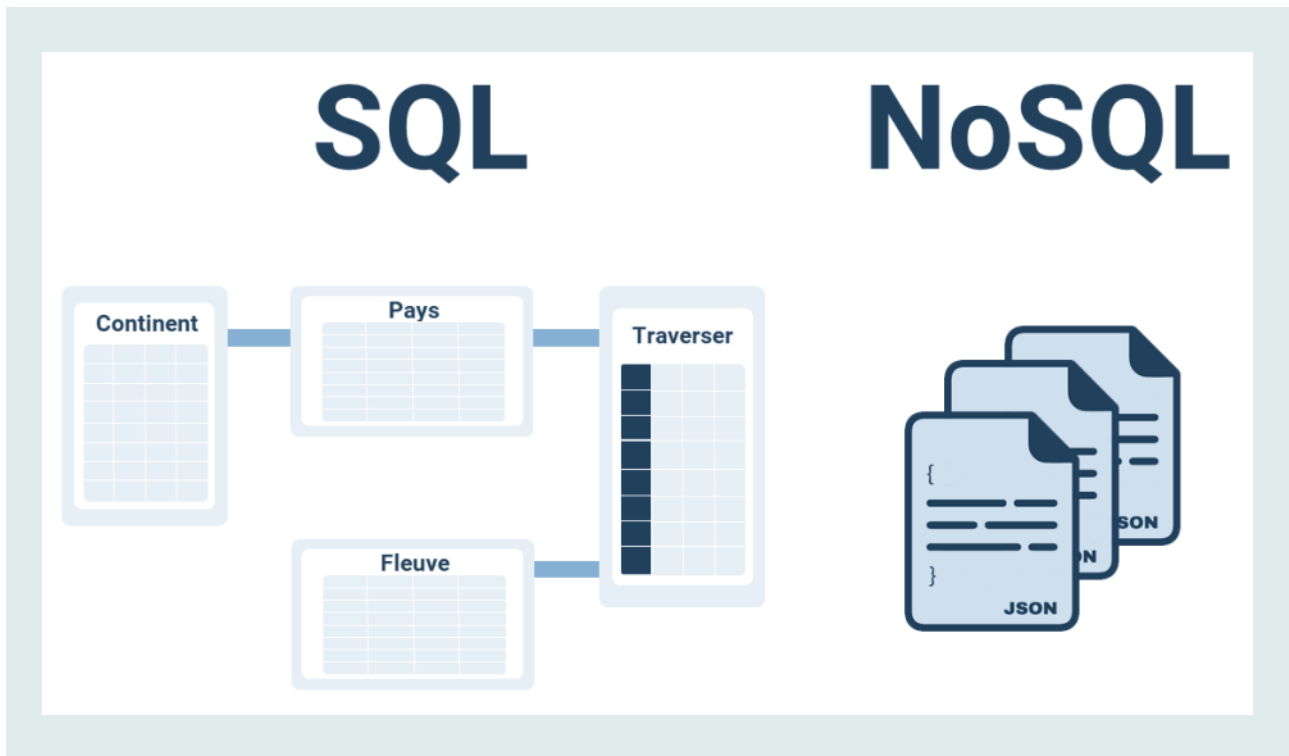
Définition Une base de données

Il existe une base de données dès que des données sont stockées. Elles sont conçues pour stocker, gérer et récupérer des informations. Elles peuvent stocker des informations sur des produits, des personnes, des commandes, ou tout autre type de données nécessaires pour les opérations d'une entreprise ou d'une organisation. Une base de données peut être aussi simple qu'une feuille de calcul contenant des lignes et des colonnes, ou aussi complexe qu'une base de données d'entreprise qui stocke des informations provenant de plusieurs systèmes et applications.

Définition Une base de données relationnelle

Une base de données relationnelle (BDD) est une base de données qui représente les données sous forme de tables, où chaque ligne représente une entité (par exemple, un employé dans une entreprise) et chaque colonne représente un attribut de cette entité (par exemple, le nom de l'employé, l'adresse, etc.). Nous pouvons imaginer pour cela différents tableaux Excel.

Dans l'image ci-dessous, nous avons une image d'une base de données relationnelle à gauche, et non relationnelle à droite. Nous comprendrons plus tard ce qu'est le SQL et le NoSQL. Nous voyons qu'une base de données relationnelle lie les données entre elles, alors qu'une base de données non-relationnelle est une liste de fichiers non liés entre eux.



Un peu d'histoire

Les bases de données relationnelles, aussi connues sous l'acronyme RDBMS (pour « *Relational Database Management System* »), ou SGBDR en français, sont le pilier de nombreux systèmes d'information modernes. Le concept de base de données relationnelle a été introduit par Edgar F. Codd, un chercheur d'IBM, dans un article révolutionnaire publié en 1970. Avant cette époque, les bases de données étaient généralement hiérarchiques ou basées sur un modèle de réseau, et la gestion des données était complexe et peu flexible.

La proposition de Edgar F. Codd est basée sur le principe du modèle relationnel des données, qui représente les données sous forme de tables composées de lignes et de colonnes. Chaque ligne de la table représente une entité unique (une « *occurrence* »), et chaque colonne représente une caractéristique de cette entité. La force de ce modèle est sa simplicité et son uniformité : toutes les opérations de données sont effectuées en manipulant ces tables de manière cohérente.

Une autre innovation clé du modèle relationnel est l'utilisation de la **logique de prédicat** et de l'**algèbre relationnelle** pour interroger les données. Cela a conduit à la création du SQL (*Structured Query Language*), qui est devenu le langage de requête standard pour les bases de données relationnelles.

Depuis l'introduction du modèle relationnel, de nombreux systèmes de gestion de bases de données relationnelles ont été développés, comme Oracle, MySQL, SQL Server de Microsoft et PostgreSQL. Ces systèmes permettent de gérer des volumes de données toujours plus importants et de soutenir une gamme de plus en plus large d'applications.

Malgré l'émergence de nouveaux types de bases de données au cours des dernières décennies, comme les bases de données NoSQL et les bases de données de graphes, les bases de données relationnelles restent dominantes dans de nombreux domaines en raison de leur robustesse, de leur maturité et de leur capacité à garantir la cohérence des données.

B. Database Management System (DBMS) ou SGBD

Un système de gestion de base de données (DBMS ou SGBD en français) est un logiciel qui sert d'interface entre la base de données et l'utilisateur ou les applications. Il fournit des moyens pour créer, récupérer, mettre à jour et gérer les données dans la base de données. Des exemples de DBMS relationnels incluent MySQL, Oracle et PostgreSQL.

Un DBMS fournit également des mécanismes pour maintenir l'intégrité des données, gérer le contrôle d'accès, gérer les transactions et récupérer les données en cas de panne système.

C. Le modèle relationnel de données

Définition

Le modèle relationnel de données est une **manière d'organiser et de structurer les données**. Le modèle relationnel de données repose sur le concept de relation, qui peut être visualisé comme des table de données avec des lignes et des colonnes, reliées les unes aux autres. Les lignes représentent des enregistrements, également appelés « *occurrences* », et les colonnes représentent des attributs, appelés aussi « *propriétés* ».

Une base de données est dite « *relationnelle* » lorsque les tables peuvent être liées entre elles.

En soit, nous pouvons imaginer des tableaux contenant des données reliés entre eux. C'est une vue simpliste, mais assez proche d'une base de données relationnelle.

Une table

Une table est donc un tableau contenant des données. Ses colonnes sont ses attributs, et ses lignes ses occurrences.

Par exemple, nous pourrions avoir une table « *Étudiant* » qui a les attributs « *ID* », « *Nom* » et « *Matière* ».

ID	Nom	Matière
1	Alice	Informatique
2	Bob	Anglais Littérature
3	Eve	Math

Dans cette table, chaque occurrence représente un étudiant, avec un ID unique, un nom et une matière.

Une occurrence

Une occurrence dans une base de données relationnelle est une **liste ordonnée de valeurs**. Chaque occurrence dans une table contient une valeur pour chaque attribut. Ces occurrences peuvent être visualisées comme des lignes dans une table. Par exemple, considérons une table « *Étudiant* » avec les attributs « *ID* », « *Nom* » et « *Matière* ». Un exemple d'occurrence pourrait être (1, 'Alice', 'Informatique'), où 1 est l'ID de l'étudiant, 'Alice' est le nom de l'étudiant, et 'Informatique' est la matière de l'étudiant.

Il est important de noter que dans une relation, on ne doit jamais avoir 2 occurrences identiques. Chaque occurrence est unique et identifiable par sa clé primaire, qui est un attribut ou un ensemble d'attributs spécifiques qui identifient de manière unique chaque occurrence dans une relation. On l'appelle souvent ID. C'est souvent un nombre unique qui s'incrémente.

Les clés dans le modèle relationnel

Une clé dans un modèle relationnel est un **attribut** ou un ensemble d'attributs qui identifient de manière unique une occurrence dans une relation. Il existe 2 types principaux de clés dans le modèle relationnel : les clés primaires et les clés étrangères.

- **Clé primaire**

La clé primaire est un attribut qui identifie de manière unique chaque occurrence dans une relation. Dans notre exemple « *Étudiant* », l'ID pourrait être la clé primaire, car il est unique pour chaque étudiant.

- **Clé étrangère**

Une clé étrangère est un attribut qui crée un lien entre les données de 2 relations. Une clé étrangère dans une table correspond à la clé primaire d'une autre relation. Par exemple, si nous avons une autre table « *Cours* » avec les attributs « id », « nom » et « EtudiantID », le « EtudiantID » dans la table, « *Cours* » serait une clé étrangère qui correspond à la clé primaire « ID » dans la table on « *Étudiant* ».

III. Comprendre les notions de normes de standardisation

A. Les principes de base de la normalisation des BDD

B. La première forme normale (1FN)

Définition

La première forme normale (1FN) est une étape fondamentale de la normalisation des bases de données relationnelles. C'est un processus qui aide à éliminer les redondances et les anomalies et à améliorer la précision et l'efficacité des opérations de la base de données.

Une table est dite en première forme normale (1FN) si tous ses attributs (colonnes) ont des **valeurs atomiques**, c'est-à-dire indivisibles. En d'autres termes, chaque cellule de la table doit contenir une valeur unique et non une collection ou un ensemble de valeurs. Cela assure que chaque élément d'information est stocké dans sa propre cellule, facilitant ainsi les opérations de requête et de mise à jour des données.

La mise en œuvre de la 1FN implique généralement le fractionnement des tables qui contiennent des attributs multivalués ou composites en tables distinctes, de manière à ce que chaque valeur ait son propre enregistrement.

Exemple

Pour illustrer la première forme normale, considérons un exemple de table non normalisée qui stocke des informations sur les étudiants et les cours auxquels ils sont inscrits :

EtudiantID	EtudiantNom	Cours
1	John Doe	Math, Anglais, Science
2	Jane Smith	Anglais, Art
3	Mark Johnson	Science, Math

Cette table n'est pas en première forme normale parce que la colonne « *Cours* » contient plusieurs valeurs pour chaque étudiant. Pour la transformer en 1FN, nous devons éliminer les valeurs multiples de la colonne « *Cours* » de telle sorte que chaque ligne contienne une seule valeur pour cet attribut.

Cela peut être réalisé en créant une nouvelle ligne pour chaque valeur de cours pour chaque étudiant, comme suit :

EtudiantID	EtudiantNom	Cours
1	John Doe	Math
1	John Doe	Anglais
1	John Doe	Science
2	Jane Smith	Anglais
2	Jane Smith	Art
3	Mark Johnson	Science
3	Mark Johnson	Math

Maintenant, chaque cellule de la table contient une seule valeur, et la table est en première forme normale.

Les avantages de la 1FN

L'application de la première forme normale a plusieurs avantages :

1. **Facilité de requête** : comme chaque élément d'information est stocké dans sa propre cellule, il est plus facile de construire et d'exécuter des requêtes pour récupérer ou manipuler les données.
2. **Intégrité des données** : la 1FN réduit les anomalies de mise à jour (insertion, suppression et modification) en éliminant les duplications de données.
3. **Flexibilité** : la 1FN facilite l'ajout de nouvelles lignes et colonnes à la base de données sans perturber l'ensemble du système.

En résumé, la première forme normale est une étape essentielle pour assurer une gestion efficace des données dans une base de données relationnelle. Elle favorise la clarté, l'intégrité et la flexibilité des données, tout en facilitant le processus de requête.

C. La deuxième forme normale (2FN)

Définition

Après la première forme normale (1FN), l'étape suivante dans la normalisation des bases de données relationnelles est la deuxième forme normale (2FN). La 2FN traite principalement des problèmes liés aux dépendances partielles des clés primaires composées.

Une table est dite en deuxième forme normale (2FN) si elle est en 1FN et si tous ses attributs non-clés sont fonctionnellement dépendants de la clé primaire. Cela signifie que chaque colonne non-clé doit dépendre de l'ensemble de la clé primaire, pas seulement d'une partie de celle-ci.

La 2FN est généralement pertinente pour les tables qui ont des clés primaires composées, c'est-à-dire des clés primaires qui se composent de plusieurs colonnes. Si une table a une clé primaire simple, elle est déjà en 2FN si elle est en 1FN.

Exemple

Pour illustrer la deuxième forme normale, prenons un exemple de table qui contient des informations sur les notes des étudiants dans différents cours. Supposons que la clé primaire de cette table soit une clé composée (EtudiantID, CoursID).

EtudiantID	CoursID	EtudiantNom	CoursNom	Note
1	101	John Doe	Math	A
1	102	John Doe	Anglais	B
2	101	Jane Smith	Math	B
2	103	Jane Smith	Science	A

Cette table est en 1FN, mais elle n'est pas en 2FN, car les attributs non-clés « *EtudiantNom* » et « *CoursNom* » dépendent uniquement d'une partie de la clé primaire. Plus précisément, « *EtudiantNom* » dépend uniquement de « *EtudiantID* » et « *CoursNom* » qui dépendent de « *CoursID* ». Ces dépendances partielles sont une violation de la 2FN.

Pour mettre cette table en 2FN, nous devons la décomposer en plusieurs tables pour éliminer les dépendances partielles. Nous pouvons créer une table « *Etudiants* », une table « *Cours* » et une table « *Notes* » comme suit :

Table « *Etudiants* » :

EtudiantID	EtudiantNom
1	John Doe
2	Jane Smith

Table « *Cours* » :

CoursID	CoursNom
101	Math
102	Anglais
103	Science

Table « *Notes* » :

EtudiantID	CoursID	Note
1	101	A
1	102	B
2	101	B
2	103	A

Maintenant, chaque attribut non-clé dépend entièrement de la clé primaire dans chaque table, et la base de données est en deuxième forme normale.

L'application de la deuxième forme normale a plusieurs avantages :

1. **Réduction des redondances** : la 2FN aide à éliminer la redondance des données en veillant à ce que chaque information ne soit stockée qu'une seule fois.
2. **Amélioration de l'intégrité des données** : en éliminant les dépendances partielles, la 2FN rend les données plus cohérentes et précises.
3. **Facilité de modification** : avec moins de redondances, les modifications des données sont plus faciles à gérer.

En résumé, la deuxième forme normale permet d'éviter les problèmes liés aux dépendances partielles et d'améliorer l'efficacité et la précision des opérations de la base de données.

D. La troisième forme normale (3FN)

Définition

La troisième forme normale (3FN) est une autre étape importante dans le processus de normalisation des bases de données relationnelles. Après avoir abordé les problèmes de redondance des données et de dépendances partielles en première et deuxième forme normale, la 3FN s'attaque à un autre type de problème : les dépendances transitives.

Une table est dite en troisième forme normale (3FN) si elle est en 2FN et si tous ses attributs non-clés sont non transitivement dépendants de la clé primaire. En d'autres termes, aucune colonne non-clé ne doit dépendre d'autres colonnes non-clés.

Une dépendance transitive se produit lorsqu'un attribut non-clé dépend d'un autre attribut non-clé qui, à son tour, dépend de la clé primaire. Pour être en 3FN, une table doit être restructurée pour éliminer toutes ces dépendances transitives.

Définition

La dépendance transitive

Le concept de dépendance transitive est essentiel à comprendre pour saisir l'essence de la 3FN. Il s'agit d'un phénomène où un attribut dépend d'un autre attribut par le biais d'un attribut intermédiaire. Cela peut mener à des anomalies de mise à jour, d'insertion et de suppression, en particulier dans les bases de données avec de nombreux attributs interdépendants.

Exemple

Pour illustrer la troisième forme normale, prenons un exemple de table qui contient des informations sur les étudiants, les cours qu'ils suivent et les professeurs qui enseignent ces cours :

EtudiantID	CoursID	Note	Professeur
1	101	A	Prof. X
1	102	B	Prof. Y
2	101	B	Prof. X
2	103	A	Prof. Z

Cette table est en 2FN, mais elle n'est pas en 3FN, car l'attribut non-clé « *Professeur* » dépend de l'attribut non-clé « *CoursID* », une dépendance transitive. Pour la mettre en 3FN, nous devons éliminer cette dépendance transitive en fractionnant la table en 2 tables, l'une pour « *Notes* » et l'autre pour « *Cours* », comme suit :

Table « *Notes* » :

EtudiantID	CoursID	Note
1	101	A
1	102	B
2	101	B
2	103	A

Table « *Cours* » :

CoursID	Professeur
101	Prof. X
102	Prof. Y
103	Prof. Z

Maintenant, chaque attribut non-clé dépend directement de la clé primaire dans chaque table, et la base de données est en troisième forme normale.

L'application de la troisième forme normale présente plusieurs avantages :

1. **Réduction des redondances** : en éliminant les dépendances transitives, la 3FN minimise la redondance des données et le risque d'incohérences.
2. **Amélioration de l'intégrité des données** : la 3FN améliore l'intégrité des données en garantissant que chaque attribut non-clé dépende directement de la clé primaire.
3. **Facilité de modification** : avec moins de redondances et une structure de données plus claire, les modifications de la base de données sont plus faciles à gérer.

En résumé, la troisième forme normale permet d'éviter les problèmes liés aux dépendances transitives et d'améliorer l'efficacité et la précision des opérations de la base de données.

E. Autres formes

Définition Boyce-Codd Normal Form

Après avoir abordé les 3 premières formes normales, la prochaine étape dans le processus de normalisation des bases de données est la « *Boyce-Codd Normal Form* » (BCNF). La BCNF est une version plus stricte de la troisième forme normale (3FN). Elle a été développée par Raymond F. Boyce et Edgar F. Codd pour résoudre certaines anomalies qui ne peuvent pas être traitées par la 3FN.

Une table est en BCNF si, pour toute dépendance fonctionnelle non triviale $X \rightarrow Y$, X est une super-clé. Autrement dit, l'attribut à gauche d'une dépendance fonctionnelle doit toujours être une super-clé. Cela garantit que les dépendances fonctionnelles sont prédéfinies et contrôlées, éliminant ainsi davantage d'anomalies potentielles que la 3FN ne peut pas résoudre.

La différence entre 3FN et BCNF

La principale différence entre la 3FN et la BCNF est que la 3FN autorise les dépendances fonctionnelles où un attribut non-clé dépend d'une partie d'une clé candidate, tandis que la BCNF l'interdit. En d'autres termes, la BCNF est plus stricte en matière de dépendances fonctionnelles que la 3FN.

Exemple

Prenons un autre exemple d'une table qui répertorie les élèves inscrits à différents cours :

EtudiantID	Nom	CoursID	CoursNom
1	Bob	C1	Maths
1	Bob	C2	Anglais
2	Alice	C1	Maths
2	Alice	C3	Science
3	Max	C2	Anglais
3	Max	C3	Science

Dans cette table, la clé primaire est (*EtudiantID*, *CoursID*), et il existe une dépendance fonctionnelle (*CoursID*) → *CoursNom*. Cependant, cette dépendance viole la BCNF car *CoursID* n'est pas une super-clé. Pour mettre cette table en BCNF, nous devons la diviser en 2 tables, une pour les étudiants et une pour les cours, comme suit :

Table « *Enrollment* » :

EtudiantID	Nom	CoursID
1	Bob	C1
1	Bob	C2
2	Alice	C1
2	Alice	C3
3	Max	C2
3	Max	C3

Table « Cours » :

CoursID	CoursNom
C1	Maths
C2	Anglais
C3	Science

Chaque table est maintenant en BCNF, car toutes les dépendances fonctionnelles sont déterminées par la super-clé.

La BCNF offre plusieurs avantages importants pour la gestion des bases de données.

1. **Élimination des anomalies** : la BCNF peut éliminer plus d'anomalies que la 3FN en restreignant davantage les dépendances fonctionnelles.
2. **Amélioration de la performance** : en éliminant les dépendances superflues, la BCNF peut réduire la taille de la base de données et améliorer les performances.
3. **Préservation de l'intégrité des données** : en garantissant que toutes les dépendances sont contrôlées et déterminées par les super-clés, la BCNF préserve l'intégrité des données.

La BCNF est une étape importante dans la normalisation des bases de données qui offre une meilleure gestion des dépendances fonctionnelles et une réduction des anomalies. Cependant, il est important de noter que la mise en œuvre de la BCNF peut parfois entraîner une perte d'information si elle n'est pas gérée correctement, donc elle doit être utilisée avec précaution. Dans l'ensemble, la BCNF, lorsqu'elle est correctement appliquée, peut offrir une structure de base de données plus solide et plus performante.

F. Exercice : Quiz

[solution n°1 p.15]

Question 1

Qu'est-ce qu'une base de données relationnelle ?

- ☐ Une base de données qui stocke les données sous forme de tables avec des lignes et des colonnes
- ☐ Une base de données qui stocke les données sous forme de graphes
- ☐ Une base de données qui stocke les données sous forme de documents semi-structurés

Question 2

Qu'est-ce qu'une clé primaire dans le modèle relationnel ?

- ☐ Un attribut qui identifie de manière unique chaque occurrence dans une relation
- ☐ Un attribut qui crée un lien entre les données de 2 relations
- ☐ Un attribut qui représente une caractéristique d'une entité

Question 3

Qu'est-ce que la première forme normale (1FN) ?

- ☐ Une étape fondamentale de la normalisation des bases de données relationnelles qui garantit que chaque cellule d'une table contient une valeur atomique.
- ☐ Une étape de la normalisation qui garantit que chaque colonne non-clé dépend de l'ensemble de la clé primaire.
- ☐ Une étape de la normalisation qui garantit que chaque attribut non-clé est non transitivement dépendant de la clé primaire.

Question 4

Qu'est-ce que la deuxième forme normale (2FN) ?

- ☐ Une étape de la normalisation qui garantit que chaque cellule d'une table contient une valeur unique
- ☐ Une étape de la normalisation qui garantit que chaque colonne non-clé dépend de l'ensemble de la clé primaire
- ☐ Une étape de la normalisation qui garantit que chaque attribut non-clé est non transitivement dépendant de la clé primaire

Question 5

Quelle est la principale différence entre la troisième forme normale (3FN) et la Boyce-Codd Normal Form (BCNF) ?

- ☐ La BCNF permet les dépendances fonctionnelles partielles, contrairement à la 3FN
- ☐ La 3FN permet les dépendances fonctionnelles partielles, contrairement à la BCNF
- ☐ La BCNF est plus stricte en matière de dépendances fonctionnelles que la 3FN

IV. L'essentiel

Les bases de données sont omniprésentes dans le milieu du développement, savoir réaliser des bases de données de bonne qualité est donc une qualité essentielle pour devenir un bon développeur. Ce cours sur les notions fondamentales des bases de données a fourni une vue d'ensemble approfondie de la gestion des données dans un environnement informatique. Nous avons appris que les bases de données sont conçues pour stocker, gérer et récupérer des informations, qu'elles soient aussi simples que des feuilles de calcul ou aussi complexes que des bases de données d'entreprise.

Nous avons découvert les bases du modèle relationnel, où chaque table représente une entité et chaque colonne un attribut, ainsi que la distinction entre les clés primaires et étrangères pour garantir l'intégrité et la cohérence des données. Les différentes formes de normalisation ont été explorées en détail : la première forme normale (1FN) pour éliminer les redondances, la deuxième forme normale (2FN) pour gérer les dépendances partielles, et la troisième forme normale (3FN) pour traiter les dépendances transitives. Nous avons abordé la Boyce-Codd Normal Form (BCNF), une version plus stricte de la 3FN, qui élimine davantage d'anomalies potentielles en contrôlant les dépendances fonctionnelles.

V. Auto-évaluation

A. Exercice

Considérez la table suivante représentant les commandes d'une entreprise. On admet que l'on ne peut commander qu'une seule référence d'article par commande :

Table : Commandes

Numéro de commande	Date de commande	Client	Produit	Quantité	Prix unitaire
C001	15 mai 2022	John	Téléphone	5	500
C002	20 juin 2022	Lisa	Protection	3	15
C003	15 mai 2022	John	Protection	2	15
C004	10 juillet 2022	Emma	Téléphone	4	500

Question 1

[solution n°2 p.16]

Cette table utilise-t-elle un modèle de base de données relationnelle ?

Question 2


[solution n°3 p.16]

Montrez comment cette table pourrait être transformée en respectant la norme 3NF.

Solutions des exercices


Exercice p. 12 Solution n°1**Question 1**

Qu'est-ce qu'une base de données relationnelle ?

- ☒ Une base de données qui stocke les données sous forme de tables avec des lignes et des colonnes
- ☐ Une base de données qui stocke les données sous forme de graphes
- ☐ Une base de données qui stocke les données sous forme de documents semi-structurés
-  Une base de données relationnelle représente les données sous forme de tables, où chaque ligne représente une entité et chaque colonne représente un attribut de cette entité. Ces tables peuvent être liées entre elles.


Question 2

Qu'est-ce qu'une clé primaire dans le modèle relationnel ?

- ☒ Un attribut qui identifie de manière unique chaque occurrence dans une relation
- ☐ Un attribut qui crée un lien entre les données de 2 relations
- ☐ Un attribut qui représente une caractéristique d'une entité
-  Une clé primaire est un attribut qui identifie de manière unique chaque occurrence dans une relation. Généralement, c'est ce que l'on appelle ID, un nombre ou une chaîne de caractère généré, avec une valeur unique dans la table.


Question 3

Qu'est-ce que la première forme normale (1FN) ?

- ☒ Une étape fondamentale de la normalisation des bases de données relationnelles qui garantit que chaque cellule d'une table contient une valeur atomique.
- ☐ Une étape de la normalisation qui garantit que chaque colonne non-clé dépend de l'ensemble de la clé primaire.
- ☐ Une étape de la normalisation qui garantit que chaque attribut non-clé est non transitivement dépendant de la clé primaire.
-  La première forme normale (1FN) garantit que chaque cellule d'une table contient une valeur atomique, éliminant ainsi les valeurs multiples dans une seule cellule. Par exemple, on ne pourra pas avoir la valeur « français, anglais, espagnol » dans une colonne langues parlées, car il y aurait plusieurs valeurs dans une cellule.

Question 4

Qu'est-ce que la deuxième forme normale (2FN) ?

- ☐ Une étape de la normalisation qui garantit que chaque cellule d'une table contient une valeur unique
- ☒ Une étape de la normalisation qui garantit que chaque colonne non-clé dépend de l'ensemble de la clé primaire
- ☐ Une étape de la normalisation qui garantit que chaque attribut non-clé est non transitivement dépendant de la clé primaire
-  La deuxième forme normale (2FN) garantit que chaque colonne non-clé dépende de l'ensemble de la clé primaire, éliminant ainsi les dépendances partielles des clés primaires composées.

Question 5

Quelle est la principale différence entre la troisième forme normale (3FN) et la Boyce-Codd Normal Form (BCNF) ?

- ☐ La BCNF permet les dépendances fonctionnelles partielles, contrairement à la 3FN
- ☐ La 3FN permet les dépendances fonctionnelles partielles, contrairement à la BCNF
- ☒ La BCNF est plus stricte en matière de dépendances fonctionnelles que la 3FN

Q La principale différence entre la 3FN et la BCNF est que la BCNF est plus stricte en matière de dépendances fonctionnelles. La BCNF n'autorise pas les dépendances fonctionnelles où un attribut non-clé dépend seulement d'une partie d'une clé candidate.

p. 14 Solution n°2

La table des commandes donnée ne respecte pas complètement le modèle de base de données relationnelles. Nous pouvons le remarquer au fait qu'il y a redondance de données. Les colonnes « *Client* » et « *Produit* » contiennent des valeurs répétées.

Pour rendre la table conforme au modèle relationnel, il serait préférable de normaliser les données en utilisant des relations entre les différentes entités (Commande, Client, Produit).

p. 14 Solution n°3

Table : Commandes

Numéro de commande	Date de commande	Client	Produit	Quantité
C001	15 mai 2022	1	1	5
C002	20 juin 2022	2	2	3
C003	15 mai 2022	1	2	2
C004	10 juillet 2022	3	1	4

Table : Produits

IdProduit	Produit	Prix unitaire
1	Téléphone	500
2	Protection	15

Table : Client

Numéro client	Nom
1	John
2	Lisa
3	Emma