|  |  |
| --- | --- |
|  | **Conception & Développement Informatique**  **Concevoir une base de données**  Apprentissage | Théorie |

**La méthode Merise**

Étude de cas simple : Gestion d’employés

Contenu

[Les étapes de la démarche Merise 1](#_Toc46145161)

[Étude de cas : Gestion d’employés 2](#_Toc46145162)

[Analyse : L’interview 3](#_Toc46145163)

# Les étapes de la démarche Merise

*"Pour concevoir un système d'information robuste, il faut, en premier lieu, identifier les données à traiter et les relations entre ces données."*

|  |  |
| --- | --- |
| **A**  **N**  **A**  **L**  **Y**  **S**  **E** | 1. **Recueillir les informations**    1. L'interview    2. Les documents    3. Les contraintes sur les données 2. **Constituer le dictionnaire des données**    1. Repérer les données à représenter    2. Supprimer toute donnée "calculée"    3. Préciser les contraintes liées à chaque donnée |
| **C**  **O**  **N**  **C**  **E**  **P**  **T**  **I**  **O**  **N** | 1. **Établir le modèle conceptuel**    1. Repérer et créer les entités    2. Attribuer à chaque entité un identifiant et compléter le dictionnaire des données    3. Placer les propriétés dans les entités    4. Repérer et placer les associations 2. **Valider le modèle**    1. Identifier les dépendances fonctionnelles    2. Respect de la 1èreForme Normale (1FN)    3. Respect de la2èmeForme Normale (2FN)    4. Respect de la3ème Forme Normale (3FN) 3. **Transformer le modèle conceptuel en modèle logique**    1. Respect des règles de transformation    2. Les entités et associations deviennent des relations    3. Les identifiants deviennent des clés primaires    4. Les associations impliquent la création de clés étrangères dans les relations |
| **R**  **É**  **A**  **L**  **I**  **S**  **A**  **T**  **I**  **O**  **N** | 1. **Implémenter le modèle physique**    1. Le modèle logique devient un schéma relationnel    2. Les relations deviennent des tables    3. Les attributs deviennent des champs (ou colonnes)    4. Les clés primaires impliquent la création de contraintes d'unicité    5. Les clés étrangères impliquent la création de contraintes d'intégrité relationnelle    6. Choix du Système de Gestion de Base de Données    7. Le schéma est implémenté dans un langage relationnel (création des tables et des contraintes)    8. Le schéma est testé et validé par un ou plusieurs jeux de données (tests d'intégrité)    9. Les processus métiers et de sécurité sont implémentés (procédures stockées, déclencheurs…)    10. L'ensemble est testé en conditions réelles (tests fonctionnels)    11. Livraison |

# Étude de cas : Gestion d’employés

Une entreprise est une entité possédant un nom, un numéro SIREN, un directeur, des managers et des employés.

Une personne est caractérisée par un nom et un prénom.

Un employé est une personne disposant d'un numéro d'employé (identifiant) et d'une adresse email auto calculée et non modifiable. Un tel objet permet également de connaitre le rôle (la fonction) de l'employé au sein de son entreprise.

Les employés sont intégrés à des équipes gérées par des managers eux-mêmes des employés. Les managers gèrent leur propre liste d'employés mais ne peuvent pas y intégrer d'autres managers. Tous les employés associés à un manager possèdent le même rôle (la même fonction) que le manager au sein de l'entreprise. Plus concrètement, un manager possédant le rôle "développeur" est un manager de développeurs. Un employé peut faire partie de plusieurs équipes à la fois si toutes les conditions précédentes sont respectées.

Un directeur supervise le tout. Il est le manager des managers et peut interagir avec (et agir sur) toutes les équipes de l'entreprise. Il possède également la compétence pour ajouter ou supprimer un employé de l'entreprise (sauf lui-même, bien entendu). Il est possible de modifier l'adresse email d'un directeur.

Pour finir, tous les employés font partie d’un département (logistique, comptabilité, informatique…). Un département est caractérisé par un nom, un numéro de bâtiment et un étage. Chaque employé est associé à un seul département.

Une équipe peut accueillir des employés de différents départements.

On souhaite aujourd’hui informatiser la gestion des employés dans l’entreprise.

# Analyse : L’interview

Les bâtiments sont référencés de quelle manière ?

Par numéro.

Un manager peut-il gérer plusieurs équipes ?

# Le dictionnaire des données

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entité** | **Mnémonique** | **Signification** | **Type (longueur)** | **Contraintes** |
| **Compagnies** | Co\_name | Nom de l’entreprise | Alphanumérique (50) | Obligatoire |
|  | Co\_siren | Numéro de siren de l’entreprise | Numérique (11) | Identifiant auto incrémenté |
| **Employees** | Emp\_name | Nom de l’employé | Alphanumérique (50) | Obligatoire |
|  | Emp\_firstname | Prenom de l’employé | Alphanumérique (50) | Facultatif |
|  | Emp\_email | Mail de l’employé | Alphanumérique (100) | Obligatoire |
|  | Emp\_role | Rôle de l’employé | Alphanumérique (50) | Obligatoire |
|  | Emp\_id | Numéro d’employé | Numérique (11) | Obligatoire |
| **department** | Dept\_name | Nom du département | Alphanumérique (50) | Obligatoire |
|  | Dept\_building | Bâtiment | Numérique (3) | Obligatoire |
|  | Dept\_floor | Étage | Numérique (3) | Obligatoire |

# Règles de gestion

Un employé est embauché dans **une** entreprise et un seul.

Une entreprise peu embauchée **un ou plusieurs** employés.

Un manager est un employé.

Un manager gère **un ou plusieurs** employés mais ne peut pas gérer d’autre manager.

Un employé est un gérer par **un ou plusieurs** manager.

Un manageur ne peut gérer que des employés qui ont le même rôle que lui-même.

Le directeur est un employé.

Le directeur est un manager.

Le directeur manage **un ou plusieurs** managers.

Un manager est managé par **un seul et un seul** directeur.

En employé est intégré à **un seu**l département.

Un département intègre **un ou plusieurs** employés.

Dépendance fonctionnelle

# Modèle conceptuel

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Co\_siren** | **Emp\_id** | **Dept\_name** |
| **Co\_siren** |  | 1 |  |
| **Co\_name** | 1 |  |  |
| **Emp\_id** |  |  |  |
| **Emp\_name** |  | 1 |  |
| **Emp\_firstname** |  | 1 |  |
| **Emp\_email** |  | 1 |  |
| **Emp\_role** |  | 1 |  |
| **Dept\_name** |  | 1 |  |
| **Dept\_building** |  |  | 1 |
| **Dept\_floor** |  |  | 1 |

co\_siren -> co\_name

emp\_id -> emp\_name, emp\_firstname, emp\_email, emp\_role

dept\_name -> dept\_building, dept\_floor

--- FIN DU DOCUMENT ---

<http://www.arfp.asso.fr>