

# Modèle Conceptuel de Données (MCD) – Parc Informatique / Connexions des Utilisateurs

## Contexte

L'objectif est de modéliser un **système de suivi des connexions** dans un réseau d'entreprise.

Chaque :

- **Utilisateur** peut se connecter à un ou plusieurs **ordinateurs**.
- **Connexion** est enregistrée avec une **date**, une **heure de début** et une **heure de fin**.

## Approche Initiale : Sans MCD

Une équipe a débuté le projet directement en SQL, sans modélisation préalable.

Table unique initiale

```
CREATE TABLE connexion (  
  id_connexion INT PRIMARY KEY,  
  nom_utilisateur VARCHAR(100),  
  prenom_utilisateur VARCHAR(100),  
  service VARCHAR(100),  
  email VARCHAR(100),  
  nom_ordinateur VARCHAR(100),  
  adresse_ip VARCHAR(15),  
  systeme_exploitation VARCHAR(50),  
  date_installation DATE,  
  date_connexion DATETIME,  
  heure_debut TIME,  
  heure_fin TIME  
);
```

Extrait de données

id_connexion	nom_utilisateur	prenom_utilisateur	service	email	nom_ordinateur	adresse_ip	systeme_exploitation	
1	Dupont	Jean	Comptabilité	j.dupont@entreprise.fr	PC-COMPTA-01	192.168.1.10	Windows 10	:
2	Martin	Sophie	RH	s.martin@entreprise.fr	PC-RH-05	192.168.1.20	Windows 11	:
3	Dupont	Jean	Comptabilité	j.dupont@entreprise.fr	PC-COMPTA-01	192.168.1.10	Windows 10	:
4	Durand	Pierre	Informatique	p.durand@entreprise.fr	PC-INFO-03	192.168.1.30	Linux Ubuntu	:

Problèmes observés

- **Forte redondance** des informations utilisateur et ordinateur.
- **Pas de séparation logique** des entités métier.
- **Aucune contrainte d'intégrité** (pas de clés étrangères).
- **Maintenance difficile**, risque d'incohérences.

## Solution : Modélisation avec un MCD

Afin de corriger ces problèmes, l'équipe a décidé de modéliser les données à l'aide d'un **MCD** en identifiant trois entités principales :

- **UTILISATEUR**
- **ORDINATEUR**
- **CONNEXION** (entité d'association)

MCD (Diagramme Mermaid)

```
erDiagram
    UTILISATEUR ||--o{ CONNEXION : "effectue"
    ORDINATEUR ||--o{ CONNEXION : "cible"

    UTILISATEUR {
```

```
    int id_utilisateur PK
    string nom
    string prenom
    string service
    string email
}

ORDINATEUR {
    int id_ordinateur PK
    string nom_ordinateur
    string adresse_ip
    string systeme_exploitation
    date date_installation
}

CONNEXION {
    int id_connexion PK
    datetime date_connexion
    time heure_debut
    time heure_fin
    int id_utilisateur FK
    int id_ordinateur FK
}
```

Objectifs du MCD

- Identifier les **entités** et **attributs clés**
- Définir les **relations** entre entités
- Clarifier les **cardinalités**
- Assurer la **cohérence et l'intégrité des données**
- Servir de **base de validation** pour les parties prenantes

Composants du MCD

Élément	Description
Entité	Élément du métier (ex : <b>Utilisateur</b> )
Attribut	Propriété d'une entité (ex : nom, email)
Association	Relation entre deux entités (ex : "utilise un ordinateur")
Cardinalité	Contraintes de participation (ex : 1,1 ou 0,n)

Exemple de structure normalisée

Entités

- **UTILISATEUR** : employés de l'entreprise.
- **ORDINATEUR** : postes du réseau.
- **CONNEXION** : événements de connexion.

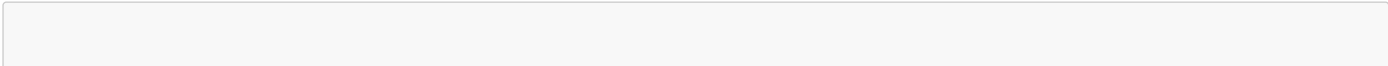
Attributs

Entité	Attributs principaux
UTILISATEUR	nom, prénom, service, email
ORDINATEUR	nom, IP, OS, date d'installation
CONNEXION	date, heure de début, heure de fin

Associations & Cardinalités

Relation	Cardinalité
Utilisateur → Connexion	0,n (peut faire plusieurs connexions)
Ordinateur → Connexion	0,n (peut être utilisé plusieurs fois)
Connexion → Utilisateur/Ordinateur	1,1 (chaque connexion a un seul utilisateur et un seul ordinateur)

Représentation textuelle (Merise)



```
ENTITE UTILISATEUR (#id_utilisateur: INT, nom: VARCHAR, prenom: VARCHAR, service: VARCHAR, email: VARCHAR)
ENTITE ORDINATEUR (#id_ordinateur: INT, nom_ordinateur: VARCHAR, adresse_ip: VARCHAR, systeme_exploitation: VARCHAR,
date_installation: DATE)
ENTITE CONNEXION (#id_connexion: INT, date_connexion: DATE, heure_debut: TIME, heure_fin: TIME)

ASSOCIATION effectuee (UTILISATEUR 0,n --- 1,1 CONNEXION)
ASSOCIATION cible (ORDINATEUR 0,n --- 1,1 CONNEXION)
```

Pourquoi **CONNEXION** est une entité ?

- Contient des attributs propres (date, heure début/fin).
- Permet de tracer l'historique des connexions.
- Nécessite des clés étrangères vers **UTILISATEUR** et **ORDINATEUR**.

Avantages du MCD

Avantage	Bénéfices concrets
Clarté métier	Vision claire et partagée des besoins
Modularité	Séparation des entités et des responsabilités
Intégrité des données	Réduction des erreurs et incohérences
Évolutivité	Ajout d'attributs ou d'entités facilité
Communication	Document commun pour développeurs et métiers

Passage au MLD (Modèle Logique de Données)

- **MCD → MLD** : on traduit les entités, leurs attributs et les associations (relations) en **tables** avec **clés primaires** et **clés étrangères**.

1. Table **UTILISATEUR**

```
CREATE TABLE Utilisateur (
  id_utilisateur INT PRIMARY KEY,
  nom VARCHAR(100) NOT NULL,
  prenom VARCHAR(100) NOT NULL,
  service VARCHAR(100),
  email VARCHAR(100) UNIQUE
);
```

2. Table **ORDINATEUR**

```
CREATE TABLE Ordinateur (
  id_ordinateur INT PRIMARY KEY,
  nom_ordinateur VARCHAR(100) NOT NULL,
  adresse_ip VARCHAR(15) NOT NULL,
  systeme_exploitation VARCHAR(50),
  date_installation DATE
);
```

3. Table **CONNEXION**

```
CREATE TABLE Connexion (
  id_connexion INT PRIMARY KEY,
  date_connexion DATETIME NOT NULL,
  heure_debut TIME,
  heure_fin TIME,
  id_utilisateur INT NOT NULL,
  id_ordinateur INT NOT NULL,
  FOREIGN KEY (id_utilisateur) REFERENCES Utilisateur(id_utilisateur),
  FOREIGN KEY (id_ordinateur) REFERENCES Ordinateur(id_ordinateur)
);
```

Schéma Physique (MPD – SQL)

```

erDiagram
    UTILISATEUR ||--o{ CONNEXION : "effectue"
    ORDINATEUR ||--o{ CONNEXION : "cible"

    UTILISATEUR {
        int id_utilisateur PK
        string nom
        string prenom
        string service
        string email UNIQUE
    }

    ORDINATEUR {
        int id_ordinateur PK
        string nom_ordinateur
        string adresse_ip
        string systeme_exploitation
        date date_installation
    }

    CONNEXION {
        int id_connexion PK
        datetime date_connexion
        time heure_debut
        time heure_fin
        int id_utilisateur FK
        int id_ordinateur FK
    }

```

## Pourquoi faire un MCD avant le SQL ?

Créer un **Modèle Conceptuel de Données (MCD)** avant de passer à l'implémentation SQL présente de nombreux avantages :

Objectif	Avec MCD	Sans MCD
<b>Compréhension des besoins</b>	Validation claire avec les utilisateurs métiers	Risque de malentendus ou oublis fonctionnels
<b>Conception structurée</b>	Base propre, sans redondances	Tables mal conçues, données dupliquées
<b>Définition des relations</b>	Cardinalités explicites, associations bien modélisées	Liens oubliés ou ambigus
<b>Maintenance évolutive</b>	Facile à modifier ou enrichir	Modifications complexes et risquées
<b>Qualité du code SQL</b>	Génération de tables claires, contraintes précises	Implémentation fragile, manque de cohérence
<b>Gains de temps sur le long terme</b>	Moins de corrections, plus de stabilité	Révisions fréquentes, perte de productivité

### Exemple concret

Sans MCD, la première version de la base stockait plusieurs fois les mêmes informations utilisateur ou ordinateur. Cela aurait entraîné :

- des erreurs de mise à jour,
- une croissance inutile du volume de données,
- une mauvaise performance des requêtes.

## Outils pour créer un MCD

### Gratuits

- [Lucidchart](#)
- [Draw.io](#)
- [Merise.fr \(MCD Helper\)](#)

### Payants

- PowerAMC
- Microsoft Visio

## Conclusion

Le **Modèle Conceptuel de Données (MCD)** est une **étape indispensable** pour tout projet de base de données :

1. Il permet de **structurer l'information** autour des entités métier réelles.
2. Il évite les erreurs classiques de conception (redondances, oublis de liens, mauvaise normalisation).
3. Il facilite la **communication entre les équipes techniques et métiers**.
4. Il rend la base **plus maintenable, évolutive et performante**.

5. Il sert de **référence documentaire** tout au long du projet.

---

Résumé

Étape du projet	Avec MCD	Sans MCD
Cadrage des besoins	Clarifié avec les utilisateurs	Risques d'oublis
Modélisation	Propre et normalisée	Données mal organisées
Implémentation SQL	Représentation fidèle du métier	Tables incohérentes ou redondantes
Maintenance et évolution	Facilement adaptable	Risque de régressions ou d'incohérences
Performance et optimisation	Accès rapide, structure claire	Requêtes lentes, base surchargée