**Onglet 1**

****

***Conception et Implémentation d’une Base de Données pour la Gestion des Incidents de Sécurité.***

**Réalisé par: Encadré par:**

**BENBOUZIANE Imane Mme GONTIER**

**DIARRA Mamadou**

**ANANI Semh**

**Formation : M1 CARE – Administrateurs des réseaux des Entreprises**

**Sommaire :**

[1. Introduction 2](#_b0nvf4kp2sqj)

[1.1 Contexte du projet 2](#_407syylisxy5)

[1.2 Objectifs du projet 3](#_rmtlub7vgr8f)

[1.3 Méthodologie 3](#_ap0vcy19dcvf)

[2. Cahier des charges fonctionnel 4](#_vyxmhqmheiig)

[2.1 Données à gérer : 4](#_sa2kolpwa2mb)

[2.2 Règles de gestion 5](#_fk4mc76jh9vr)

[3. Conception de la base de données 6](#_oijlpn1wlrog)

[3.1 Modèle conceptuel de données 6](#_1zjgy8najfww)

[3.2 Modèle logique relationnel (MLD) 7](#_8u6potw6tled)

[4. Implémentation dans le SGBR 7](#_8q883h59qm8s)

[4.1 Scripts SQL de création des tables et contraintes : 8](#_1gtn6heu7yny)

[5. Jeu de données : 9](#_ytp8hdgzsmdy)

[5.1 Jeu de données d’exemple : 9](#_eaye3oxz0z47)

[6. Ensemble de requêtes SQL documentées : 17](#_rift8gydxant)

[7. Proposition de vues (rapports analytiques) simulant un tableau de bord SOC 20](#_in1z3np2ae4x)

[7.1 Vue 1 : Nombre d’incident par niveau de gravité 20](#_9ald4bigzv7)

[7.2. Vue 2 : Nombre d’incident par type de menace 20](#_6xu54dywagu4)

[7.3. Vue 3 : Actifs les plus attaqués 21](#_mcy0yizwnpi)

[7.4 Vue 4 : Incidents liés aux vulnérabilités 21](#_26tuva53xbic)

[7.5. Vue 5 : Temps moyen de résolution des incidents 22](#_65kt2geznc4e)

[7.6 Vue 6 : Temps moyen de résolution des incidents par équipe 22](#_z9y913z91ru9)

[7.7 Vue 7 : Actions correctives par statut 23](#_n0z28jdg1l3f)

[7.8 Vue 8 : Incident par source d’alerte 23](#_uowfr088eulz)

[7.9 Vue 9 : Incidents non résolu 24](#_ul3751ftasjx)

[8. Conclusion 24](#_nlovbfrqp8zi)

[9. Annexe : 26](#_1s579feguqie)

[9.1 Script création de tables : 26](#_ca0nplxe2pmu)

[9.2 Script création jeu de donnée : 28](#_x2hklm1kdugz)

## 

# **1. Introduction**

## **1.1 Contexte du projet**

L’organisation MIG, acteur majeur dans les services financiers, fait face à une intensification des menaces cybernétiques : ransomwares, phishing ciblé, intrusions réseau, exfiltration de données et compromissions internes.

Pour renforcer son Security Operations Center (SOC), MIG souhaite mettre en place une base de données centralisée capable de recenser, classifier et corréler les incidents de sécurité.

L’objectif du projet est de concevoir et d’implémenter une base de données sur la gestion des incidents liés à des cyberattaques.

## **1.2 Objectifs du projet**

1. Définir un cahier des charges clair pour la base de données.
2. Concevoir un MCD et un MLD adaptés à la problématique.
3. Implémenter la base dans un SGBD relationnel via des scripts SQL.
4. Fournir un jeu de données représentatif pour les tests.
5. Élaborer des requêtes SQL
6. Créer des vues et proposer un rapport analytique permettant d’extraire des indicateurs utiles pour un SOC.

## **1.3 Méthodologie**

La méthodologie de conception suivie repose sur plusieurs étapes :

1. Identification des besoins et règles de gestion.
2. Conception d’un modèle conceptuel de données (MCD).
3. Traduction en modèle relationnel (MLD).
4. Implémentation dans un SGBDR (MySQL/PHPMyadmin).
5. Réalisation de requêtes SQL avancées et vues analytiques.

# **2. Cahier des charges fonctionnel**

## 

## Le système d’information de MIG doit disposer d’une base de données centralisée qui lui permette de gérer efficacement les incidents de cybersécurité.

## L’objectif principal est de fournir un outil de qualité et sécurisé qui garantisse :

## **Le suivi opérationnel** : chaque incident doit être enregistré avec ses particularités (gravité, type, description, date, état) pour permettre une totale gestion de son cycle de vie.

## **L’analyse technique** : les incidents doivent être associés aux actifs concernés, aux menaces détectées ainsi qu’aux vulnérabilités connues (CVE, score CVSS), pour évaluer leur criticité.

## **La dimension organisationnelle** : la base de donnée doit prendre en compte la formation des équipes de sécurité et les rôles et responsabilités de leurs membres, afin d’identifier clairement les acteurs impliqués dans la résolution.

## **Le suivi des actions correctives** : chaque correction mise en place (planification, exécution, résultats) doit être documentée, pour permettre des retours d’expérience et d’améliorer les processus prochainement mis en place.

## Cette base de données n’est pas seulement un registre qui contient des incidents, mais un outil décisionnel qui va permettre au SOC et à la direction sécurité d’évaluer les risques, de prioriser les actions et de mesurer l’efficacité des mesures de protection.

## **2.1 Données à gérer :**

**Table Incident :** *répertorie les incidents*

* id\_Incident INT : clé primaire
* type\_incident VARCHAR(50) : nom de l’incident
* niveau\_gravité VARCHAR(50) : gravité de l’incident (faible, moyen, élevé, critique)
* date\_detection DATE : date à laquelle l’incident a été détecté
* date\_resolution DATE : date à laquelle l’incident a été résolue (peut être NULL)
* statut VARCHAR(50) : niveau de résolution (en cours ou résolu)
* description VARCHAR(50)

**Table Menaces :** *répertorie les menaces qui peuvent mener à des incidents*

* id\_menace INT : clé primaire
* nom\_menace VARCHAR(50) : exemple : malware….
* description VARCHAR(50)

**Table Actifs :** *répertorie les différents actifs qui peuvent être affecté par un incident*

* id\_actif INT : clé primaire
* type\_actifs VARCHAR(50) : nom des actifs (serveur, poste de travail, application, base de données, réseau)
* localisation VARCHAR(50) : site, agence, salle
* criticité VARCHAR(50) : faible, moyenne, élevée, critique, très critique

**Table Vulnérabilités :** *répertorie les vulnérabilités qui peuvent causer des incidents*

* id\_vulnérabilité INT : clé primaire
* CVE VARCHAR(50) : identifier la vulnérabilité
* CVSS FLOAT : score compris entre 0 et 10

**Table Sources d'alerte :** *répertorie les source qui alertent sur les incidents*

* id\_source INT : clé primaire
* type\_source VARCHAR(50) : (SIEM, IDS, antivirus, logs système)
* outil VARCHAR(50)
* description VARCHAR(50)

**Table Équipes :** *répertorie les équipes qui s’occupent des incidents*

* id\_équipe INT : clé primaire
* nom\_equipe VARCHAR(50)
* domaine\_expertise VARCHAR(50) : dans l’équipe est spécialisé (SOC, réseau, systèmes, forensic…)
* contact VARCHAR(50) : l’adresse mail pour joindre l’équipe

**Table Membres :** *répertorie les membres qui constituent les différentes équipes*

* id\_membre INT : clé primaire
* nom VARCHAR(50)
* prenom VARCHAR(50)
* mail VARCHAR(50) : contacte de la personne
* téléphone INT : contact de la personne

**Table Actions correctives :** *répertorie les actions réalisés par les équipes pour résoudre les incidents*

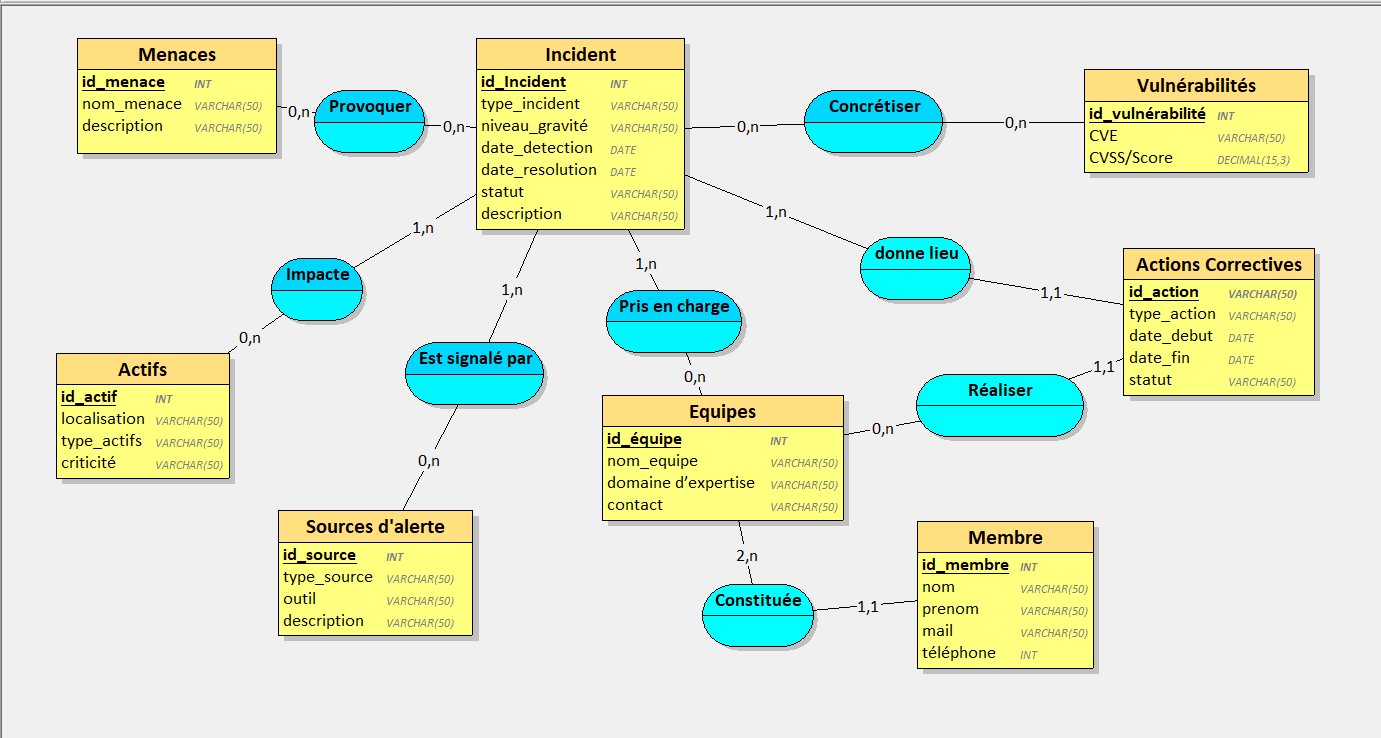
* id\_action INT : clé primaire
* type\_action VARCHAR(50) : nom de l’action
* date\_debut DATE
* date\_fin DATE
* statut VARCHAR(50) : niveau d'exécution de l’action (planifiée, en cours, terminée)

## **2.2 Règles de gestion**

* **Incident ↔ Actif** : un incident **impacte** au moins un actif (**1..N**) et un actif peut être impacté par zéro ou plusieurs incidents (**0..N**).
* **Incident ↔Vulnérabilité** : un incident peut être **concrétiser** zéro ou plusieurs vulnérabilités **(0..N)** et une vulnérabilité peut se **concrétiser** par plusieurs incidents (**0..N**).
* **Incident** ↔ **Source d’alerte** : un incident peut être **signalé** par une ou plusieurs sources **(1..N)** et chaque source peut signaler plusieurs incidents (**0..N**).
* **Incident** ↔ **Équipe** : un incident est **pris en charge** par au moins une équipe (**1..N**) et une équipe peut gérer plusieurs incidents (**0..N**).
* **Incident** ↔ **Action corrective** : un incident peut donner lieu à plusieurs actions correctives (**1..N**) et chaque action est réalisée pour exactement un incident (**1..1**).
* **Menace** ↔ **Incident :** Une menace peut **provoquer** zéro ou plusieurs incidents (**0..N**) et un incident peut concerner zéro ou plusieurs menaces (**0..N**).
* **Equipe** ↔ **Membre** : une équipe est constituée de plusieurs membres (**2..N**) et chaque membre ne fait partie que d’une seule équipe (**1..1**).
* **Equipe** ↔ **Action Corrective** : une équipe peut réaliser des actions correctives pour gérer un incident (**0..N**) et chaque action corrective est réalisée par une seule équipe (**1..1**).

# **3. Conception de la base de données**

## **3.1 Modèle conceptuel de données**

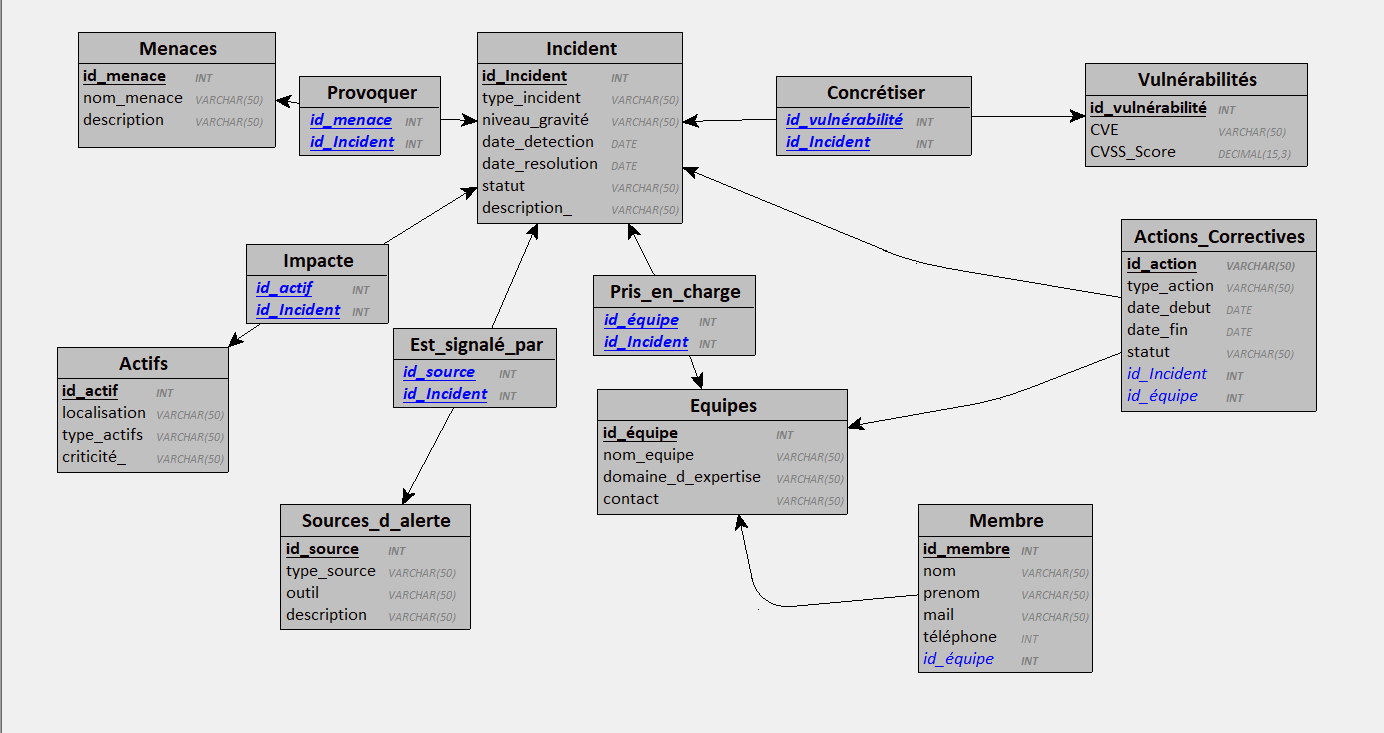
****

Le MCD représente l’ensemble des entités nécessaires à la gestion des incidents et leurs relations :

* Un **incident** est lié à un ou plusieurs actifs.
* Une **menace** peut concerner plusieurs incidents.
* Une **vulnérabilité** peut être exploitée dans plusieurs attaques.
* Une **équipe** est responsable de plusieurs incidents et se compose de plusieurs membres.
* Chaque **action** **corrective** est rattachée à un incident et planifiée dans le temps.

Le MCD permet ainsi d’avoir une vision claire et indépendante du langage informatique, facilitant la compréhension aussi bien pour les équipes techniques que pour les responsables métiers.

## **3.2 Modèle logique relationnel (MLD)**

****

Le MLD traduit le MCD en tables relationnelles.  
Chaque entité devient une table, et chaque relation est matérialisée par une clé étrangère ou une table d’association.

Les règles suivantes ont guidé sa conception :

* Respect de l’intégrité référentielle avec des clés étrangères et des contraintes ON DELETE CASCADE.
* Normalisation des données pour éviter les doublons et garantir la cohérence.
* Extensibilité du modèle afin de permettre l’ajout futur de nouvelles entités (par exemple : journaux de détection automatisée, indicateurs de performance, logs d’outils SIEM).

# **4. Implémentation dans le SGBR**

L’implémentation de la base a été réalisée en SQL.  
Les scripts fournis permettent :

* La création des tables principales (Incidents, Menaces, Vulnérabilités, Actifs, Équipes, Membres, Actions).
* La mise en place des contraintes d’intégrité (PRIMARY KEY, FOREIGN KEY, CHECK, UNIQUE).
* La création de tables d’association pour gérer les relations de type plusieurs-à-plusieurs.

Nous avons choisi MySQL comme SGBDR, associé à phpMyAdmin pour la gestion et la visualisation, en raison de sa simplicité d’administration, de sa compatibilité multiplateforme et de sa large adoption dans le monde professionnel.

## **4.1 Scripts SQL de création des tables et contraintes :**

Afin de mettre en place une base de données permettant de gérer efficacement les incidents de sécurité, différents scripts SQL ont été élaborés. Ces scripts définissent les tables principales, leurs attributs, ainsi que les relations entre elles. L’objectif est d’assurer une structure cohérente, normalisée et respectant l’intégrité des données (**voir annexe Script création de tables**).

Les premières tables créées concernent les éléments essentiels du système :

* Menaces : stocke les différentes menaces identifiées, avec un identifiant unique, un nom et une description.
* Actifs : recense les actifs de l’organisation, avec leur localisation, leur type et leur niveau de criticité.
* Vulnérabilités : contient les vulnérabilités connues, identifiées notamment par leur référence CVE et un score CVSS.
* Sources d’alerte : regroupe les outils ou systèmes qui détectent les incidents et envoient des alertes.
* Équipes : décrit les différentes équipes intervenant en cas d’incident, avec leurs domaines d’expertise et leurs contacts.
* Incidents : centralise les informations sur chaque incident (type, gravité, dates, statut et description).
* Membres : représente les personnes appartenant à une équipe donnée, avec leurs coordonnées.
* Actions correctives : liste les actions entreprises pour résoudre un incident, avec leur état d’avancement.

Chaque table est définie avec une clé primaire (PRIMARY KEY) et des contraintes adaptées, telles que l’unicité de certaines colonnes (UNIQUE), des valeurs contrôlées (CHECK), et la gestion des relations (FOREIGN KEY).

Pour modéliser les relations entre les différentes entités, plusieurs tables d’association ont été créées :

* Provoquer : relie une menace à un incident.
* Est\_signalé\_par : associe un incident à une source d’alerte.
* Impacter : indique quels actifs sont affectés par un incident.
* Pris\_en\_charge : relie une équipe aux incidents dont elle est responsable.
* Concrétiser : relie une vulnérabilité aux incidents dans lesquels elle s’est matérialisée.

Ces tables possèdent des clés primaires composites, ainsi que des clés étrangères permettant de maintenir la cohérence des données grâce aux options ON DELETE CASCADE et ON UPDATE CASCADE.

Les contraintes mises en place garantissent :

* l’unicité des données sensibles (ex. : noms de menace, adresses e-mail),
* la validité des valeurs (ex. : niveaux de criticité ou de gravité prédéfinis),
* la cohérence entre les entités grâce aux relations hiérarchisées.

Ainsi, la base de données obtenue est à la fois robuste, normalisée et adaptée à la gestion d’incidents de sécurité informatique.

# **5. Jeu de données :**

Le script complet d’insertion des données est disponible en annexe et sert à peupler la base pour tester les requêtes et les vues analytiques.

Afin de tester et de valider la structure de la base de données, un script SQL d’insertion de données a été rédigé (**voir annexe Script création jeu de données**).  
Ce jeu de données permet de :

* renseigner quelques menaces types,
* ajouter des actifs avec différents niveaux de criticité,
* insérer des vulnérabilités identifiées par des codes CVE,
* définir des équipes et leurs membres,
* simuler des incidents avec leurs dates, niveaux de gravité et statuts,
* associer des actions correctives aux incidents.

Ces données fictives facilitent la vérification des contraintes, des relations entre les tables et du bon fonctionnement des requêtes qui seront appliquées par la suite.

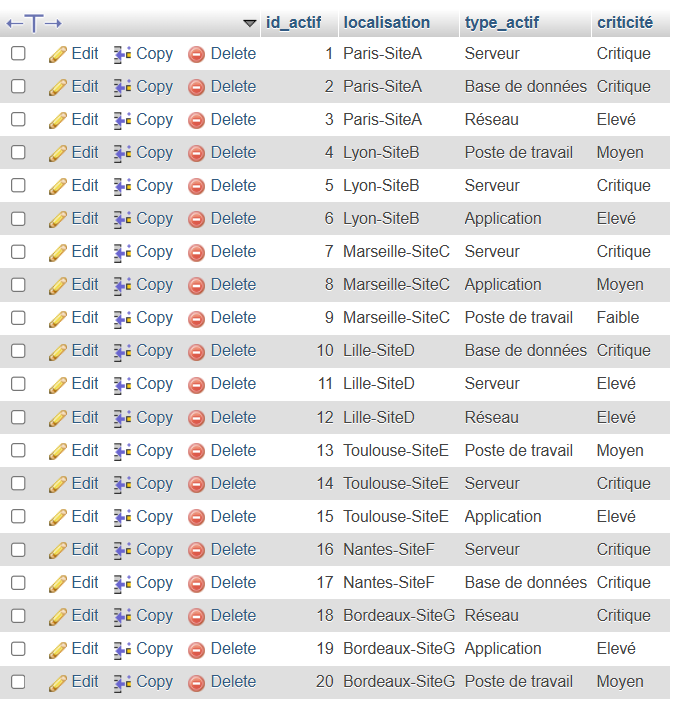
## **5.1 Jeu de données d’exemple :**

Pour tester la base de données, un jeu de données fictif a été injecté. Ces données couvrent plusieurs cas d’usage :

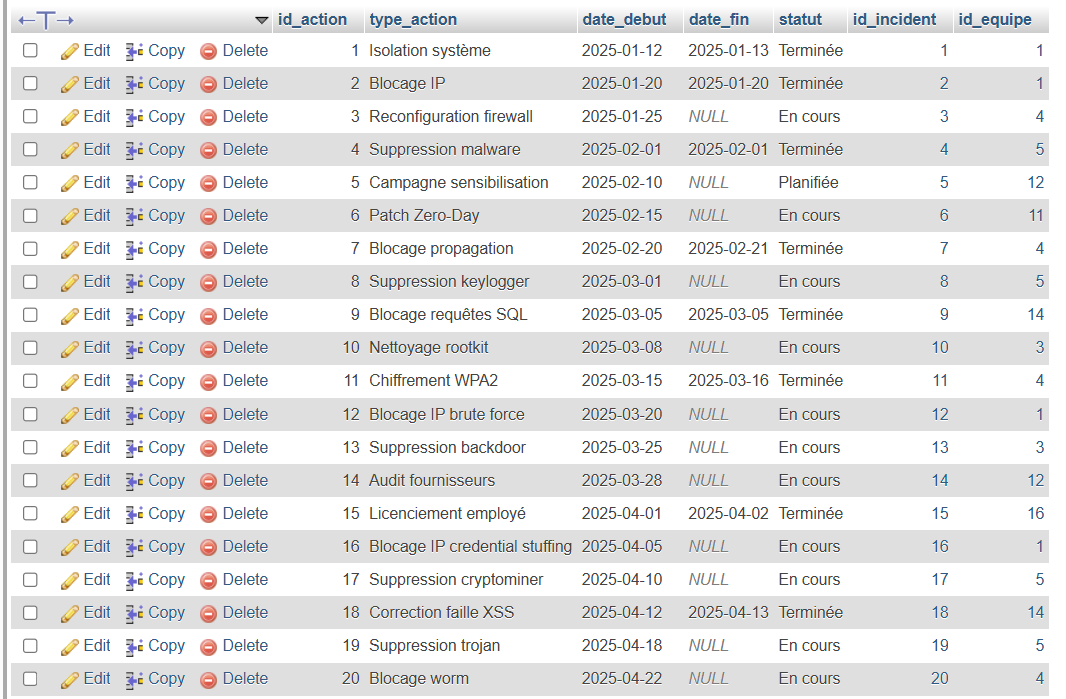
* **Incidents critiques** : par exemple, une attaque par ransomware sur un serveur de fichiers stratégiques.
* **Menaces fréquentes** : phishing, malware, injection SQL.
* **Vulnérabilités connues** : CVE avec score CVSS élevé, permettant d’établir des corrélations entre les incidents et les failles exploitées.
* **Actifs stratégiques** : serveurs applicatifs, bases de données clients, postes utilisateurs.
* **Équipes de sécurité** : SOC, administrateurs systèmes et réseaux, analystes sécurité.
* **Actions correctives** : application de patchs, analyse forensique, restauration de données, durcissement des configurations.

Ce jeu de données permet de valider la structure de la base et de tester les requêtes SQL sur des scénarios réalistes.

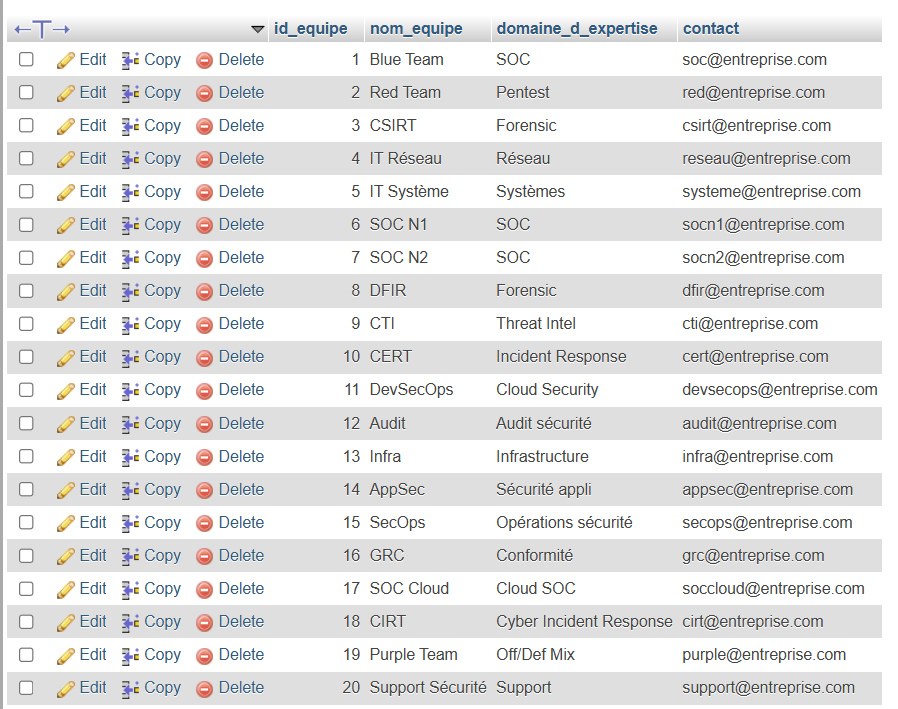
**Table Actifs :**



**Table Actions\_Correctives :**



**Table Equipes :**



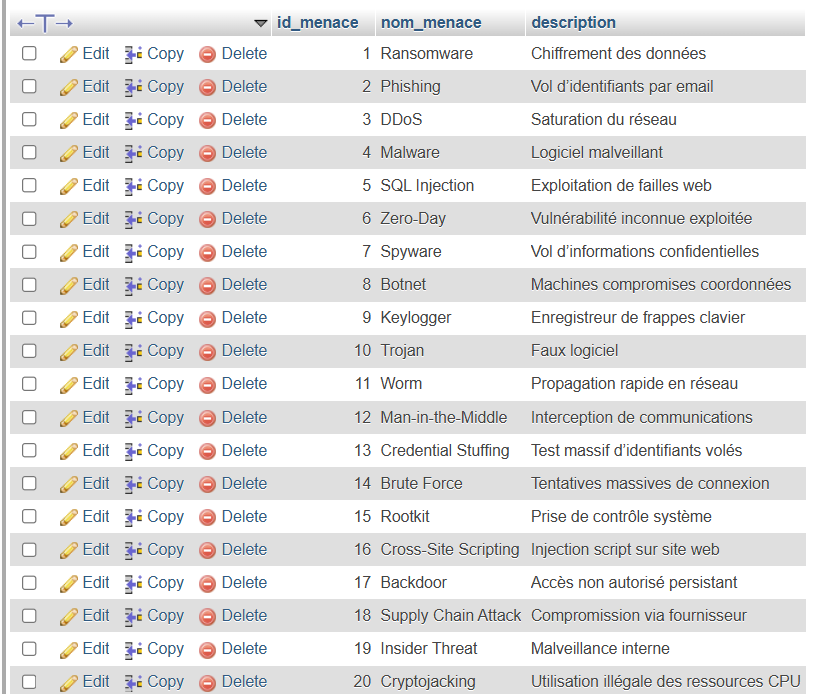
**Table Membres :**

****

**Table Incident :**

****

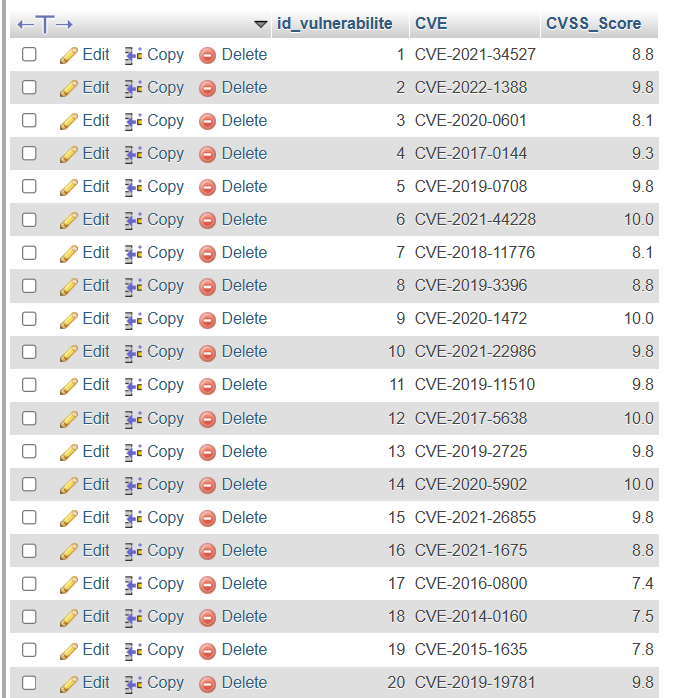
**Table Menaces :**

****

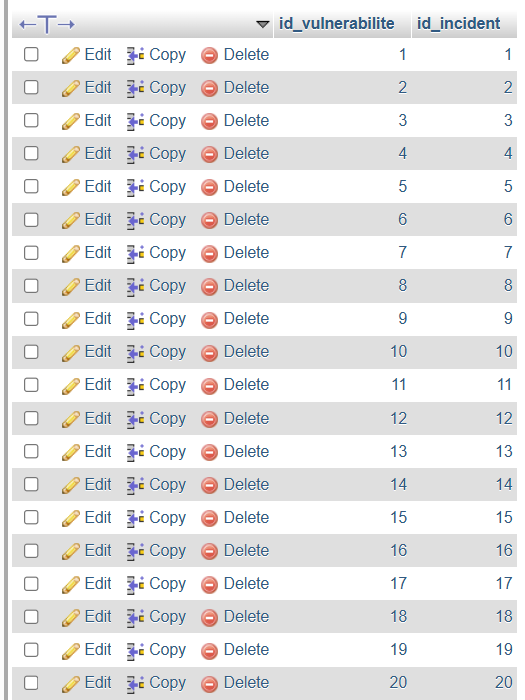
**Table Source\_d\_alerte :**

****

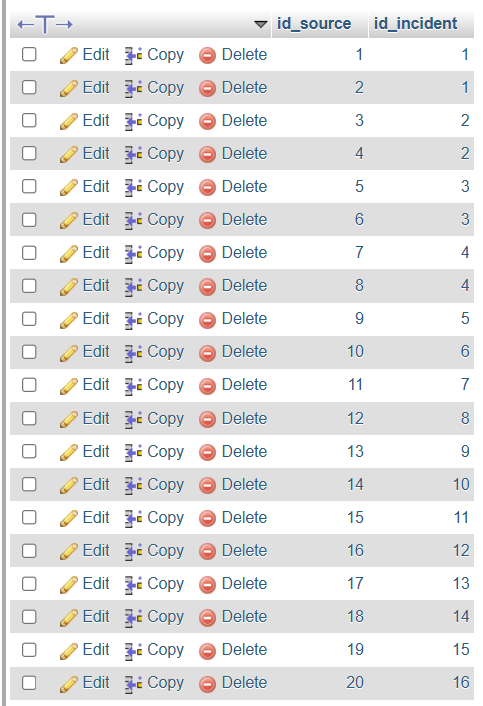
**Table Vulnerabilites :**

****

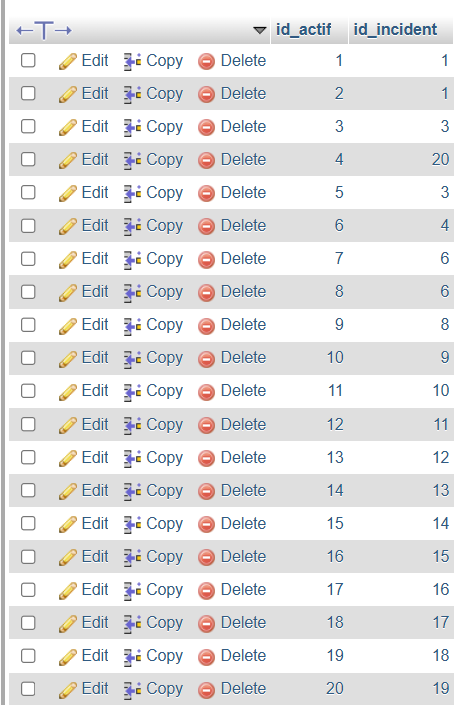
**Table Concretiser :**

****

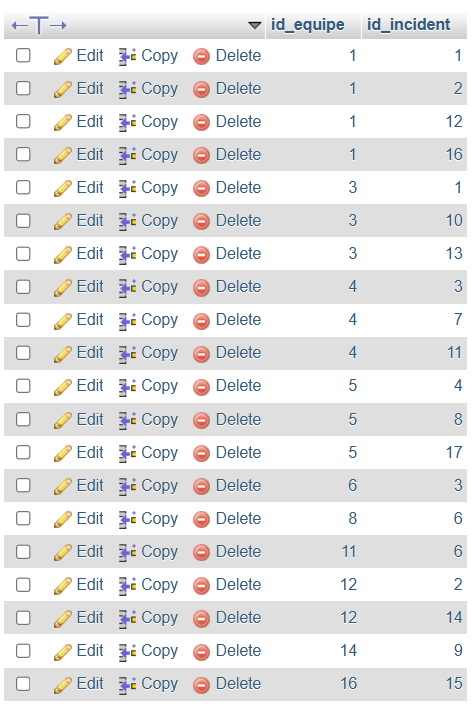
**Table Est\_signale\_par :**

****

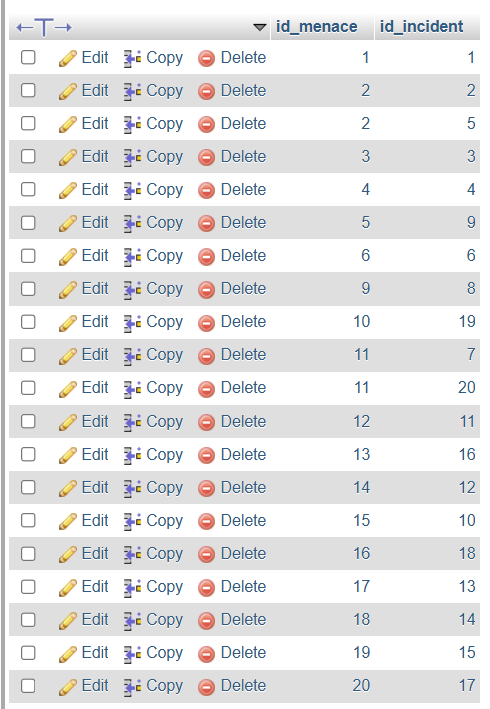
**Table Impacter :**

****

**Table Pris\_en\_charge :**

****

**Table Provoquer :**

****

# **6. Ensemble de requêtes SQL documentées :**

Plusieurs requêtes ont été définies pour répondre à des besoins opérationnels précis :

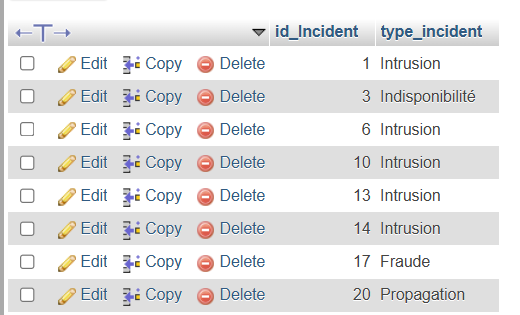
* **Lister les incidents critiques non résolus** : pour identifier les menaces nécessitant une intervention immédiate.
* **Identifier les actifs les plus attaqués afin** de repérer les points sensibles de l’infrastructure.
* **Mesurer la performance des équipes** : calcul du temps moyen de résolution par équipe, utile comme indicateur de performance (KPI).
* **Corriger incidents et vulnérabilités** : relier les attaques aux failles exploitées pour orienter les actions correctives.

Ces requêtes constituent la base d’un futur tableau de bord de suivi de la cybersécurité.

Lister les incidents critiques non résolus :

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) id\_Incident, type\_incident FROM incident

WHERE niveau\_gravite = 'Critique' [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html%23operator_and) statut <> 'Résolu';



Identifier les actifs les plus attaqués au cours d’une période donnée :

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) A.type\_actif, [COUNT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_count)(I.id\_Incident) AS actif\_plus\_attaque

FROM Actifs A

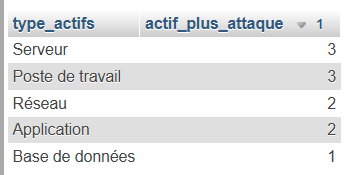
JOIN Impacter IM ON A.id\_actif = IM.id\_actif

JOIN Incident I ON IM.id\_Incident = I.id\_Incident

WHERE I.date\_detection >= '2025-03-06'

GROUP BY A.type\_actif

ORDER BY actif\_plus\_attaque DESC;



Calculer le temps moyen de résolution des incidents par équipe :

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) E.id\_equipe, E.nom\_equipe, [AVG](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_avg)(I.date\_resolution - I.date\_detection) AS delai\_moyen\_resolution

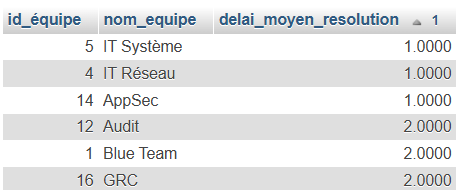
FROM Equipes E JOIN Pris\_en\_charge P ON E.id\_equipe = P.id\_equipe

JOIN Incident I ON P.id\_Incident = I.id\_Incident

WHERE I.statut = 'Résolu'

GROUP BY E.id\_equipe, E.nom\_equipe

ORDER BY delai\_moyen\_resolution;



Corréler les incidents avec les vulnérabilités connues (CVE) :

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) I.id\_Incident, I.type\_incident, V.id\_vulnerabilite, V.CVE FROM Incident I

JOIN Concretiser C ON I.id\_Incident = C.id\_Incident

JOIN Vulnerabilites V ON C.id\_vulnerabilite = V.id\_vulnerabilite

ORDER BY I.date\_detection DESC;



# **7. Proposition de vues (rapports analytiques) simulant un tableau de bord SOC**

On a décidé de créer huit vues pour pouvoir simuler au mieux un tableau de bord SOC (Security Operations Center).

## **7.1 Vue 1 : Nombre d’incident par niveau de gravité**

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) vue\_incidents\_par\_gravite AS

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [COUNT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_count)(id\_Incident) AS nb\_incidents, niveau\_gravite FROM Incident

GROUP BY niveau\_gravite

ORDER BY nb\_incidents DESC;



## **7.2. Vue 2 : Nombre d’incident par type de menace**

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) vue\_incidents\_par\_menace AS

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) M.nom\_menace, [COUNT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_count)(P.id\_Incident) AS nb\_incidents

FROM Menaces M

JOIN Provoquer P ON M.id\_menace = P.id\_menace

GROUP BY M.nom\_menace

ORDER BY nb\_incidents DESC;

****

## **7.3. Vue 3 : Actifs les plus attaqués**

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) vue\_actifs\_plus\_attaques AS

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) A.type\_actif, [COUNT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_count)(I.id\_Incident) AS actif\_plus\_attaque

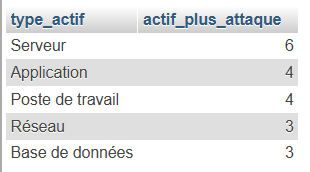
FROM Actifs A

JOIN Impacter IM ON A.id\_actif = IM.id\_actif

JOIN Incident I ON IM.id\_Incident = I.id\_Incident

GROUP BY A.type\_actif

ORDER BY actif\_plus\_attaque DESC;

****

## **7.4 Vue 4 : Incidents liés aux vulnérabilités**

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) vue\_incident\_lies\_aux\_vulnerabilites AS

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) V.CVE, V.CVSS\_Score, [COUNT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_count)(I.id\_Incident) AS nb\_incidents

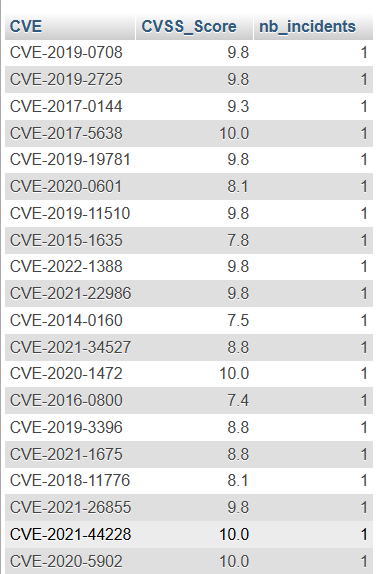
FROM Incident I

JOIN Concretiser C ON I.id\_Incident = C.id\_Incident

JOIN Vulnerabilites V ON C.id\_vulnerabilite = V.id\_vulnerabilite

GROUP BY V.CVE, V.CVSS\_Score

ORDER BY nb\_incidents DESC;



## **7.5. Vue 5 : Temps moyen de résolution des incidents**

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) vue\_temps\_moyen\_resolution\_incident AS

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [AVG](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_avg)(date\_resolution - date\_detection) AS delai\_moyen\_resolution\_en\_jour

FROM Incident

WHERE statut = 'Résolu'

ORDER BY delai\_moyen\_resolution\_en\_jour;



## **7.6 Vue 6 : Temps moyen de résolution des incidents par équipe**

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) vue\_temps\_moyen\_resolution\_incident\_par\_equipe AS

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) E.id\_equipe, E.nom\_equipe, [AVG](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_avg)(I.date\_resolution - I.date\_detection) AS delai\_moyen\_resolution FROM Equipes E

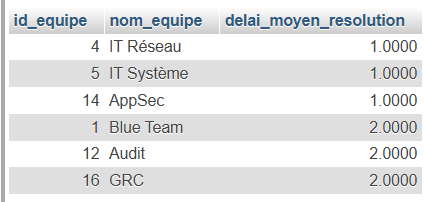
JOIN Pris\_en\_charge P ON E.id\_equipe = P.id\_equipe

JOIN Incident I ON P.id\_Incident = I.id\_Incident

WHERE I.statut = 'Résolu'

GROUP BY E.id\_equipe, E.nom\_equipe

ORDER BY delai\_moyen\_resolution;



## **7.7 Vue 7 : Actions correctives par statut**

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) vue\_action\_par\_statut AS

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) statut, [COUNT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_count)(id\_action) AS nb\_actions

FROM actions\_correctives

GROUP BY statut

ORDER BY nb\_actions DESC;

****

## **7.8 Vue 8 : Incident par source d’alerte**

[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) vue\_incidents\_par\_source\_d\_alerte AS

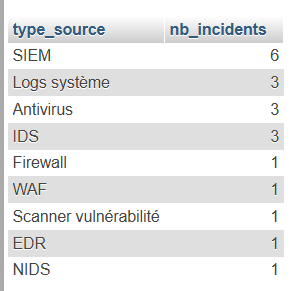
[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) S.type\_source, [COUNT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_count)(E.id\_Incident) AS nb\_incidents

FROM sources\_d\_alerte S

JOIN est\_signale\_par E ON S.id\_source = E.id\_source

GROUP BY S.type\_source

ORDER BY nb\_incidents DESC;

****

## **7.9 Vue 9 : Incidents non résolu**

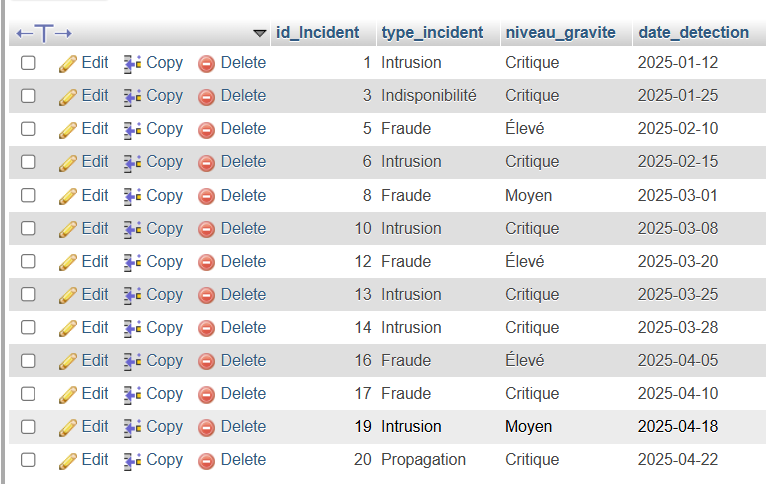
[CREATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) [VIEW](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-view.html) vue\_incident\_non\_resolu AS

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) id\_Incident, type\_incident, niveau\_gravite, date\_detection

FROM incident

WHERE statut <> 'Résolu'

ORDER BY date\_detection;



# **8. Conclusion**

La conception et l’implémentation de cette base de données pour la gestion des incidents de cybersécurité constituent un bon projet par rapport aux attentes actuelles de l’organisation MIG.

Face à la multiplication des menaces (ransomwares, phishing, intrusions réseau, exfiltration de données, compromissions internes), la centralisation des informations dans un système unique est un point essentiel pour améliorer la qualité des analyses réalisées par le SOC.

Le projet a permis d’atteindre plusieurs objectifs clés :

* **Centralisation des données** : l’ensemble des incidents, menaces, vulnérabilités, actifs et actions correctives est désormais géré dans une base unique et cohérente.
* **Suivi opérationnel structuré** : chaque incident est tracé depuis sa détection jusqu’à sa résolution, en intégrant les équipes responsables et les mesures appliquées.
* **Capacité analytique renforcée** : grâce aux requêtes SQL et aux vues analytiques, il est désormais possible de générer des rapports précis (incidents critiques, actifs les plus ciblés, temps moyen de résolution, etc.), offrant aux analystes SOC et aux responsables sécurité des indicateurs clairs pour la prise de décision.
* **Perspectives d’évolution** : la base a été pensée pour être extensible, avec la possibilité d’intégrer des flux automatiques depuis des outils externes (SIEM, IDS/IPS, antivirus), ou encore de connecter des solutions de visualisation comme Grafana, Power BI ou Kibana afin de produire de véritables tableaux de bord dynamiques.

Au-delà de l’aspect purement technique, ce projet souligne l’importance d’une démarche structurée dans la gestion des incidents de sécurité : classification, corrélation, priorisation et suivi des actions. Une telle approche permet non seulement de répondre aux incidents en cours, mais aussi de mieux anticiper les futures menaces.

En conclusion, cette base de données constitue un début essentiel dans la mise en place d’un système global de pilotage de la cybersécurité. Elle offre à MIG un outil solide et évolutif, capable de soutenir son SOC dans ses missions quotidiennes, tout en ouvrant la voie à des évolutions futures telles que l’automatisation des réponses, l’intégration de l’intelligence artificielle pour la détection d’anomalies, ou encore la mise en place de tableaux de bord stratégiques pour la direction générale.

# **9. Annexe :**

## **9.1 Script création de tables :**

CREATE TABLE Menaces(

id\_menace INT PRIMARY KEY AUTO\_INCREMENT, -- Identifiant unique de la menace

nom\_menace VARCHAR(50) NOT NULL UNIQUE, -- Nom unique de la menace

description VARCHAR(450) NOT NULL -- Description obligatoire

);

CREATE TABLE Actifs (

id\_actif INT PRIMARY KEY AUTO\_INCREMENT, -- Identifiant unique de l'actif

localisation VARCHAR(100) NOT NULL, -- Localisation obligatoire

type\_actif VARCHAR(50) NOT NULL , -- Type obligatoire avec valeurs autorisées

criticité VARCHAR(50) NOT NULL CHECK (criticité IN ('Faible','Moyen','Elevé','Critique'))

-- Criticité obligatoire et restreinte à ces valeurs

);

CREATE TABLE Vulnerabilites (

id\_vulnerabilite INT PRIMARY KEY AUTO\_INCREMENT, -- Identifiant unique

CVE VARCHAR(50) NOT NULL, -- CVE obligatoire

CVSS\_Score DECIMAL(3,1) NOT NULL CHECK (CVSS\_Score >= 0 AND CVSS\_Score <= 10)

-- Score CVSS obligatoire entre 0 et 10

);

CREATE TABLE Sources\_d\_alerte (

id\_source INT PRIMARY KEY AUTO\_INCREMENT, -- Identifiant unique

type\_source VARCHAR(50) NOT NULL, -- Type de source obligatoire (SIEM, IDS…)

outil VARCHAR(100) NOT NULL, -- Nom de l'outil ayant généré l'alerte

description VARCHAR(450) NOT NULL -- Message ou résumé de l'alerte

);

CREATE TABLE Equipes (

id\_equipe INT PRIMARY KEY AUTO\_INCREMENT, -- Identifiant unique

nom\_equipe VARCHAR(50) NOT NULL UNIQUE, -- Nom de l'équipe unique

domaine\_d\_expertise VARCHAR(50) NOT NULL, -- Spécialité (SOC, Réseau, Forensic…)

contact VARCHAR(100) NOT NULL UNIQUE -- Contact mail unique

);

CREATE TABLE Incident (

id\_Incident INT PRIMARY KEY AUTO\_INCREMENT, -- Identifiant unique

type\_incident VARCHAR(50) NOT NULL, -- Type obligatoire (intrusion, malware…)

niveau\_gravite VARCHAR(50) NOT NULL CHECK (niveau\_gravite IN ('Faible', 'Moyen', 'Elevé', 'Critique')),

date\_detection DATE NOT NULL, -- Date de détection obligatoire

date\_resolution DATE, -- Date de résolution non obligatoire

statut VARCHAR(50) NOT NULL CHECK (statut IN ('En cours','Résolu')), -- Statut obligatoire

description VARCHAR(450) NOT NULL -- Description obligatoire

);

CREATE TABLE Membre (

id\_membre INT PRIMARY KEY AUTO\_INCREMENT, -- Identifiant unique

nom VARCHAR(50) NOT NULL, -- Nom obligatoire

prenom VARCHAR(50) NOT NULL, -- Prénom obligatoire

mail VARCHAR(100) NOT NULL UNIQUE, -- Mail unique obligatoire

telephone VARCHAR(20) NOT NULL, -- Téléphone obligatoire

id\_equipe INT NOT NULL, -- Chaque membre appartient à une seule équipe (1..1)

FOREIGN KEY (id\_equipe) REFERENCES Equipes(id\_equipe)

ON DELETE CASCADE

ON UPDATE CASCADE

);

CREATE TABLE Actions\_Correctives (

id\_action INT PRIMARY KEY AUTO\_INCREMENT, -- Identifiant unique

type\_action VARCHAR(50) NOT NULL, -- Type de l'action réalisé

date\_debut DATE NOT NULL, -- Date du début de l'action obligatoire

date\_fin DATE, -- Date de fin l'action

statut VARCHAR(50) NOT NULL CHECK (statut IN ('Planifiée','En cours','Terminée')),

id\_incident INT NOT NULL, -- Chaque action est liée à exactement un incident (1..1)

id\_equipe INT NOT NULL, -- Chaque action est réalisée par une seule équipe (1..1)

FOREIGN KEY (id\_incident) REFERENCES Incident(id\_incident)

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (id\_equipe) REFERENCES Equipes(id\_equipe)

ON DELETE CASCADE

ON UPDATE CASCADE

);

-- ==============================

-- Tables d'associations (relations N-N)

-- ==============================

-- Menace -> Incident (0..N / 0..N)

CREATE TABLE Provoquer (

id\_menace INT NOT NULL,

id\_incident INT NOT NULL,

PRIMARY KEY (id\_menace, id\_incident),

FOREIGN KEY (id\_menace) REFERENCES Menaces(id\_menace) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (id\_incident) REFERENCES Incident(id\_incident) ON DELETE CASCADE ON UPDATE CASCADE

);

-- Incident -> Source d’alerte (1..N / 0..N)

CREATE TABLE Est\_signale\_par (

id\_source INT NOT NULL,

id\_incident INT NOT NULL,

PRIMARY KEY (id\_source, id\_incident),

FOREIGN KEY (id\_source) REFERENCES Sources\_d\_alerte(id\_source) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (id\_incident) REFERENCES Incident(id\_incident) ON DELETE CASCADE ON UPDATE CASCADE

);

-- Incident -> Actif (1..N / 0..N)

CREATE TABLE Impacter (

id\_actif INT NOT NULL,

id\_incident INT NOT NULL,

PRIMARY KEY (id\_actif, id\_incident),

FOREIGN KEY (id\_actif) REFERENCES Actifs(id\_actif) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (id\_incident) REFERENCES Incident(id\_incident) ON DELETE CASCADE ON UPDATE CASCADE

);

-- Incident -> Équipe (1..N / 0..N)

CREATE TABLE Pris\_en\_charge (

id\_equipe INT NOT NULL,

id\_incident INT NOT NULL,

PRIMARY KEY (id\_equipe, id\_incident),

FOREIGN KEY (id\_equipe) REFERENCES Equipes(id\_equipe) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (id\_incident) REFERENCES Incident(id\_incident) ON DELETE CASCADE ON UPDATE CASCADE

);

-- Incident -> Vulnérabilité (0..N / 0..N)

CREATE TABLE Concretiser (

id\_vulnerabilite INT NOT NULL,

id\_incident INT NOT NULL,

PRIMARY KEY (id\_vulnerabilite, id\_incident),

FOREIGN KEY (id\_vulnerabilite) REFERENCES Vulnerabilites(id\_vulnerabilite) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (id\_incident) REFERENCES Incident(id\_incident) ON DELETE CASCADE ON UPDATE CASCADE

);

## **9.2 Script création jeu de donnée :**

-- Table Menaces

INSERT INTO Menaces VALUES

(1,'Ransomware','Chiffrement des données'),

(2,'Phishing','Vol d’identifiants par email'),

(3,'DDoS','Saturation du réseau'),

(4,'Malware','Logiciel malveillant'),

(5,'SQL Injection','Exploitation de failles web'),

(6,'Zero-Day','Vulnérabilité inconnue exploitée'),

(7,'Spyware','Vol d’informations confidentielles'),

(8,'Botnet','Machines compromises coordonnées'),

(9,'Keylogger','Enregistreur de frappes clavier'),

(10,'Trojan','Faux logiciel'),

(11,'Worm','Propagation rapide en réseau'),

(12,'Man-in-the-Middle','Interception de communications'),

(13,'Credential Stuffing','Test massif d’identifiants volés'),

(14,'Brute Force','Tentatives massives de connexion'),

(15,'Rootkit','Prise de contrôle système'),

(16,'Cross-Site Scripting','Injection script sur site web'),

(17,'Backdoor','Accès non autorisé persistant'),

(18,'Supply Chain Attack','Compromission via fournisseur'),

(19,'Insider Threat','Malveillance interne'),

(20,'Cryptojacking','Utilisation illégale des ressources CPU');

-- Table Actifs

INSERT INTO Actifs VALUES

(1,'Paris-SiteA','Serveur','Critique'),

(2,'Paris-SiteA','Base de données','Critique'),

(3,'Paris-SiteA','Réseau','Elevé'),

(4,'Lyon-SiteB','Poste de travail','Moyen'),

(5,'Lyon-SiteB','Serveur','Critique'),

(6,'Lyon-SiteB','Application','Elevé'),

(7,'Marseille-SiteC','Serveur','Critique'),

(8,'Marseille-SiteC','Application','Moyen'),

(9,'Marseille-SiteC','Poste de travail','Faible'),

(10,'Lille-SiteD','Base de données','Critique'),

(11,'Lille-SiteD','Serveur','Elevé'),

(12,'Lille-SiteD','Réseau','Elevé'),

(13,'Toulouse-SiteE','Poste de travail','Moyen'),

(14,'Toulouse-SiteE','Serveur','Critique'),

(15,'Toulouse-SiteE','Application','Elevé'),

(16,'Nantes-SiteF','Serveur','Critique'),

(17,'Nantes-SiteF','Base de données','Critique'),

(18,'Bordeaux-SiteG','Réseau','Critique'),

(19,'Bordeaux-SiteG','Application','Elevé'),

(20,'Bordeaux-SiteG','Poste de travail','Moyen');

-- Table Vulnérabilités

INSERT INTO Vulnerabilites VALUES

(1,'CVE-2021-34527',8.800),

(2,'CVE-2022-1388',9.800),

(3,'CVE-2020-0601',8.100),

(4,'CVE-2017-0144',9.300),

(5,'CVE-2019-0708',9.800),

(6,'CVE-2021-44228',10.000),

(7,'CVE-2018-11776',8.100),

(8,'CVE-2019-3396',8.800),

(9,'CVE-2020-1472',10.000),

(10,'CVE-2021-22986',9.800),

(11,'CVE-2019-11510',9.800),

(12,'CVE-2017-5638',10.000),

(13,'CVE-2019-2725',9.800),

(14,'CVE-2020-5902',10.000),

(15,'CVE-2021-26855',9.800),

(16,'CVE-2021-1675',8.800),

(17,'CVE-2016-0800',7.400),

(18,'CVE-2014-0160',7.500),

(19,'CVE-2015-1635',7.800),

(20,'CVE-2019-19781',9.800);

-- Table Sources\_d\_alerte

INSERT INTO Sources\_d\_alerte VALUES

(1,'SIEM','Splunk','Corrélation événements'),

(2,'IDS','Snort','Détection intrusion réseau'),

(3,'Antivirus','Kaspersky','Détection malware poste client'),

(4,'Logs système','Windows Event','Collecte journaux Windows'),

(5,'Firewall','Palo Alto','Blocage trafic suspect'),

(6,'SIEM','ELK Stack','Analyse centralisée logs'),

(7,'IDS','Suricata','Détection trafic réseau'),

(8,'Antivirus','Bitdefender','Détection malware'),

(9,'Logs système','Syslog Linux','Journalisation OS Linux'),

(10,'SIEM','QRadar','Analyse anomalies'),

(11,'NIDS','Zeek','Analyse trafic réseau'),

(12,'EDR','CrowdStrike','Protection endpoint'),

(13,'SIEM','ArcSight','Gestion sécurité'),

(14,'Scanner vulnérabilité','Nessus','Détection failles'),

(15,'WAF','F5','Protection appli web'),

(16,'SIEM','Graylog','Analyse centralisée'),

(17,'IDS','Bro','Analyse comportement réseau'),

(18,'Antivirus','McAfee','Détection virus'),

(19,'Logs système','Sysmon','Monitoring Windows avancé'),

(20,'SIEM','Azure Sentinel','Cloud SIEM');

-- Table Equipes

INSERT INTO Equipes VALUES

(1,'Blue Team','SOC','soc@entreprise.com'),

(2,'Red Team','Pentest','red@entreprise.com'),

(3,'CSIRT','Forensic','csirt@entreprise.com'),

(4,'IT Réseau','Réseau','reseau@entreprise.com'),

(5,'IT Système','Systèmes','systeme@entreprise.com'),

(6,'SOC N1','SOC','socn1@entreprise.com'),

(7,'SOC N2','SOC','socn2@entreprise.com'),

(8,'DFIR','Forensic','dfir@entreprise.com'),

(9,'CTI','Threat Intel','cti@entreprise.com'),

(10,'CERT','Incident Response','cert@entreprise.com'),

(11,'DevSecOps','Cloud Security','devsecops@entreprise.com'),

(12,'Audit','Audit sécurité','audit@entreprise.com'),

(13,'Infra','Infrastructure','infra@entreprise.com'),

(14,'AppSec','Sécurité appli','appsec@entreprise.com'),

(15,'SecOps','Opérations sécurité','secops@entreprise.com'),

(16,'GRC','Conformité','grc@entreprise.com'),

(17,'SOC Cloud','Cloud SOC','soccloud@entreprise.com'),

(18,'CIRT','Cyber Incident Response','cirt@entreprise.com'),

(19,'Purple Team','Off/Def Mix','purple@entreprise.com'),

(20,'Support Sécurité','Support','support@entreprise.com');

-- Table Membre

INSERT INTO Membre VALUES

(1,'Martin','Paul','paul.martin@entreprise.com',123456789,1),

(2,'Dupont','Alice','alice.dupont@entreprise.com',987654321,1),

(3,'Durand','Louis','louis.durand@entreprise.com',147258369,2),

(4,'Moreau','Emma','emma.moreau@entreprise.com',369258147,2),

(5,'Bernard','Lucas','lucas.bernard@entreprise.com',741852963,3),

(6,'Petit','Chloé','chloe.petit@entreprise.com',963852741,3),

(7,'Robert','Léo','leo.robert@entreprise.com',951357456,4),

(8,'Richard','Manon','manon.richard@entreprise.com',456789123,4),

(9,'Durant','Hugo','hugo.durant@entreprise.com',258369147,5),

(10,'Leroy','Sarah','sarah.leroy@entreprise.com',357159456,5),

(11,'Simon','Camille','camille.simon@entreprise.com',654987321,6),

(12,'Fournier','Mathis','mathis.fournier@entreprise.com',321654987,6),

(13,'David','Lina','lina.david@entreprise.com',852741963,7),

(14,'Garnier','Noah','noah.garnier@entreprise.com',753951456,7),

(15,'Roux','Eva','eva.roux@entreprise.com',369147258,8),

(16,'Vincent','Tom','tom.vincent@entreprise.com',147369258,8),

(17,'Henry','Jade','jade.henry@entreprise.com',951456753,9),

(18,'Masson','Clara','clara.masson@entreprise.com',357258159,9),

(19,'Blanc','Adam','adam.blanc@entreprise.com',159357258,10),

(20,'Guerin','Nina','nina.guerin@entreprise.com',258147369,10);

-- Table Incident

INSERT INTO Incident VALUES

(1,'Intrusion','Critique','2025-01-12',NULL,'En cours','Serveur compromis'),

(2,'Fraude','Élevé','2025-01-20','2025-01-22','Résolu','Vol de comptes utilisateurs'),

(3,'Indisponibilité','Critique','2025-01-25',NULL,'En cours','Réseau saturé DDoS'),

(4,'Intrusion','Moyen','2025-02-01','2025-02-02','Résolu','Malware isolé'),

(5,'Fraude','Élevé','2025-02-10',NULL,'En cours','Campagne phishing'),

(6,'Intrusion','Critique','2025-02-15',NULL,'En cours','Exploitation Zero-Day'),

(7,'Propagation','Élevé','2025-02-20','2025-02-21','Résolu','Ver réseau stoppé'),

(8,'Fraude','Moyen','2025-03-01',NULL,'En cours','Keylogger détecté'),

(9,'Intrusion','Élevé','2025-03-05','2025-03-06','Résolu','Injection SQL stoppée'),

(10,'Intrusion','Critique','2025-03-08',NULL,'En cours','Rootkit détecté'),

(11,'Fraude','Moyen','2025-03-15','2025-03-16','Résolu','MITM sur wifi invité'),

(12,'Fraude','Élevé','2025-03-20',NULL,'En cours','Brute force massifs'),

(13,'Intrusion','Critique','2025-03-25',NULL,'En cours','Backdoor trouvée'),

(14,'Intrusion','Critique','2025-03-28',NULL,'En cours','Supply chain compromise'),

(15,'Fraude','Moyen','2025-04-01','2025-04-03','Résolu','Employé malveillant'),

(16,'Fraude','Élevé','2025-04-05',NULL,'En cours','Credential stuffing détecté'),

(17,'Fraude','Critique','2025-04-10',NULL,'En cours','Cryptojacking serveur'),

(18,'Intrusion','Élevé','2025-04-12','2025-04-14','Résolu','XSS exploité'),

(19,'Intrusion','Moyen','2025-04-18',NULL,'En cours','Trojan découvert'),

(20,'Propagation','Critique','2025-04-22',NULL,'En cours','Worm en propagation');

-- Table Actions\_Correctives

INSERT INTO Actions\_Correctives VALUES

(1,'Isolation système','2025-01-12','2025-01-13','Terminée',1,1),

(2,'Blocage IP','2025-01-20','2025-01-20','Terminée',2,1),

(3,'Reconfiguration firewall','2025-01-25',NULL,'En cours',3,4),

(4,'Suppression malware','2025-02-01','2025-02-01','Terminée',4,5),

(5,'Campagne sensibilisation','2025-02-10',NULL,'Planifiée',5,12),

(6,'Patch Zero-Day','2025-02-15',NULL,'En cours',6,11),

(7,'Blocage propagation','2025-02-20','2025-02-21','Terminée',7,4),

(8,'Suppression keylogger','2025-03-01',NULL,'En cours',8,5),

(9,'Blocage requêtes SQL','2025-03-05','2025-03-05','Terminée',9,14),

(10,'Nettoyage rootkit','2025-03-08',NULL,'En cours',10,3),

(11,'Chiffrement WPA2','2025-03-15','2025-03-16','Terminée',11,4),

(12,'Blocage IP brute force','2025-03-20',NULL,'En cours',12,1),

(13,'Suppression backdoor','2025-03-25',NULL,'En cours',13,3),

(14,'Audit fournisseurs','2025-03-28',NULL,'En cours',14,12),

(15,'Licenciement employé','2025-04-01','2025-04-02','Terminée',15,16),

(16,'Blocage IP credential stuffing','2025-04-05',NULL,'En cours',16,1),

(17,'Suppression cryptominer','2025-04-10',NULL,'En cours',17,5),

(18,'Correction faille XSS','2025-04-12','2025-04-13','Terminée',18,14),

(19,'Suppression trojan','2025-04-18',NULL,'En cours',19,5),

(20,'Blocage worm','2025-04-22',NULL,'En cours',20,4);

-- Table Provoquer (Menaces ↔ Incident)

INSERT INTO Provoquer VALUES

(1,1),(2,2),(3,3),(4,4),(2,5),

(6,6),(11,7),(9,8),(5,9),(15,10),

(12,11),(14,12),(17,13),(18,14),(19,15),

(13,16),(20,17),(16,18),(10,19),(11,20);

-- Table Impacte (Actifs ↔ Incident)

INSERT INTO Impacter VALUES

(1,1),(2,1),(3,3),(5,3),(6,4),

(7,6),(8,6),(9,8),(10,9),(11,10),

(12,11),(13,12),(14,13),(15,14),(16,15),

(17,16),(18,17),(19,18),(20,19),(4,20);

-- Table Est\_signalé\_par (Sources\_d\_alerte ↔ Incident)

INSERT INTO Est\_signale\_par VALUES

(1,1),(2,1),(3,2),(4,2),(5,3),

(6,3),(7,4),(8,4),(9,5),(10,6),

(11,7),(12,8),(13,9),(14,10),(15,11),

(16,12),(17,13),(18,14),(19,15),(20,16);

-- Table Pris\_en\_charge (Equipes ↔ Incident)

INSERT INTO Pris\_en\_charge VALUES

(1,1),(3,1),(1,2),(12,2),(4,3),

(6,3),(5,4),(11,6),(8,6),(4,7),

(5,8),(14,9),(3,10),(4,11),(1,12),

(3,13),(12,14),(16,15),(1,16),(5,17);

-- Table Concretiser (Vulnérabilités ↔ Incident)

INSERT INTO Concretiser VALUES

(1,1),(2,2),(3,3),(4,4),(5,5),

(6,6),(7,7),(8,8),(9,9),(10,10),

(11,11),(12,12),(13,13),(14,14),(15,15),

(16,16),(17,17),(18,18),(19,19),(20,20);

Onglet 2