

Compte Rendu du Projet de Base de Données : L'Univers de Zelda

Zakariya BOUMENAD Antoine ABSI

October 5, 2025

Contents

1	Introduction	2
1.1	Accès à la Base de données	2
2	Modèle Conceptuel de Données (MCD)	3
2.1	Schéma	3
2.2	Justification des cardinalités	3
2.3	Graphe des dépendances	4
3	Modèle Relationnel de Données (MRD)	5
3.1	Analyse des tables du MRD	5
3.2	Gestion du conflit lié à <i>nom</i>	7
3.3	Conclusion	7
4	Base de données	8
4.1	Création des tables vides	8
4.2	Remplissage des tables	11
5	Scénarios d'utilisation et requêtes	15

1 Introduction

Ce projet a pour objectif la création d'une base de données SQL permettant de gérer des informations relatives à l'univers du jeu vidéo Zelda. Cette base de données devra être conçue de manière à respecter les principes de la modélisation relationnelle tout en garantissant la conformité aux bonnes pratiques de conception des bases de données. Nous allons passer par plusieurs étapes : l'élaboration du Modèle Conceptuel de Données (MCD), sa transformation en Modèle Relationnel de Données (MRD), la mise en place de la base de données, et enfin des scénarios d'utilisation avec des requêtes SQL correspondantes.

1.1 Accès à la Base de données

- **Groupe:** Groupe 4 - 1
- **Nom de la base:** db_gr04_binome01
- **Utilisateur:** gr04_binome01
- **Mot de passe:** lihosytuny

Pour rappel, vous pouvez accéder à cette base via l'url: <https://gsiddev1.mines-albi.fr/adminer/>

En choisissant:

- **Système:** PostgreSQL
- **Serveur:** localhost

2 Modèle Conceptuel de Données (MCD)

Le Modèle Conceptuel de Données (MCD) est conçu en respectant la troisième forme normale (3NF), afin d'assurer l'intégrité des données tout en minimisant les redondances. Le MCD présente les entités du système ainsi que leurs relations.

2.1 Schéma

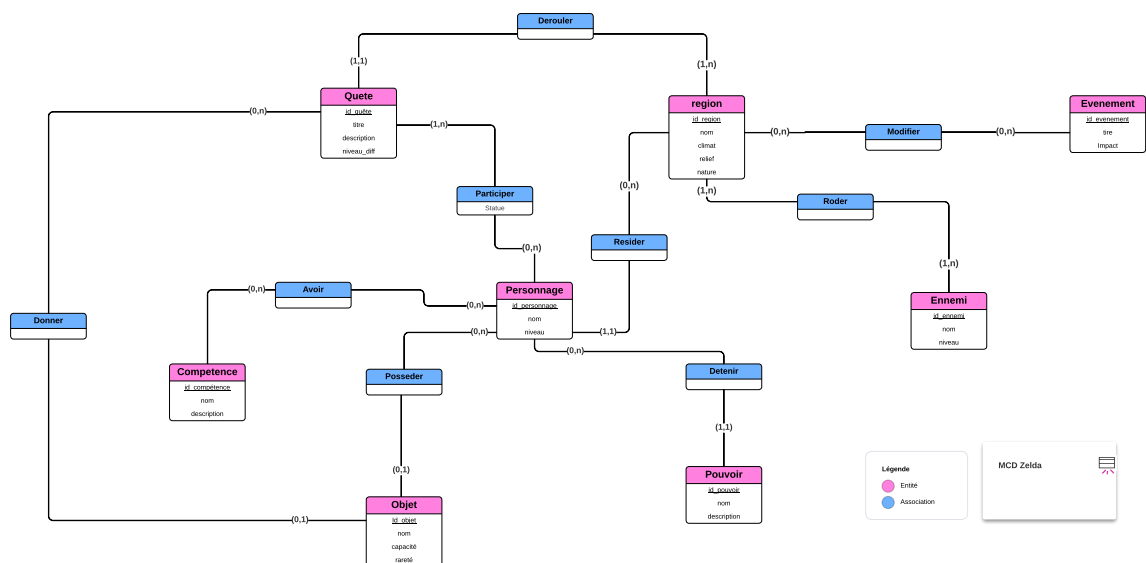


Figure 1: MCD Zelda

2.2 Justification des cardinalités

Relation	Entité 1	Entité 2	Justification des cardinalités
Résider	Personnage	Région	Un personnage réside dans une seule région (1,1), mais une région peut avoir plusieurs personnages ou être vide (0,N).
Détenir	Personnage	Pouvoir	Un personnage peut posséder plusieurs pouvoirs (0,N), et chaque pouvoir doit être lié à un personnage unique (1,1).
Posséder	Personnage	Objet	Un personnage peut posséder plusieurs objets ou aucun (0,N), et un objet peut être possédé par un seul personnage ou aucun (0,1).
Donner	Quête	Objet	Une quête peut nécessiter plusieurs objets ou aucun (0,N), et un objet peut être lié à une seule quête ou aucune (0,1).
Participer	Personnage	Quête	Un personnage peut participer à plusieurs quêtes ou aucune (0,N), et une quête doit impliquer au moins un participant (1,N).
Dérouler	Quête	Région	Une quête doit se dérouler dans au moins une région (1,N), et une région peut accueillir plusieurs quêtes ou aucune (1,N).
Modifier	Événement	Région	Un événement peut modifier plusieurs régions ou aucune (0,N), et une région peut être impactée par plusieurs événements ou aucun (0,N).
Rôder	Ennemi	Région	Un ennemi rôde dans au moins une région (1,N), et une région doit contenir au moins un ennemi (1,N).
Avoir	Personnage	Compétence	Un personnage peut avoir plusieurs compétences ou aucune (0,N), et une compétence peut être partagée par plusieurs personnages ou aucun (0,N).

Table 1: Résumé des relations et cardinalités du MCD Zelda

2.3 Graphe des dépendances

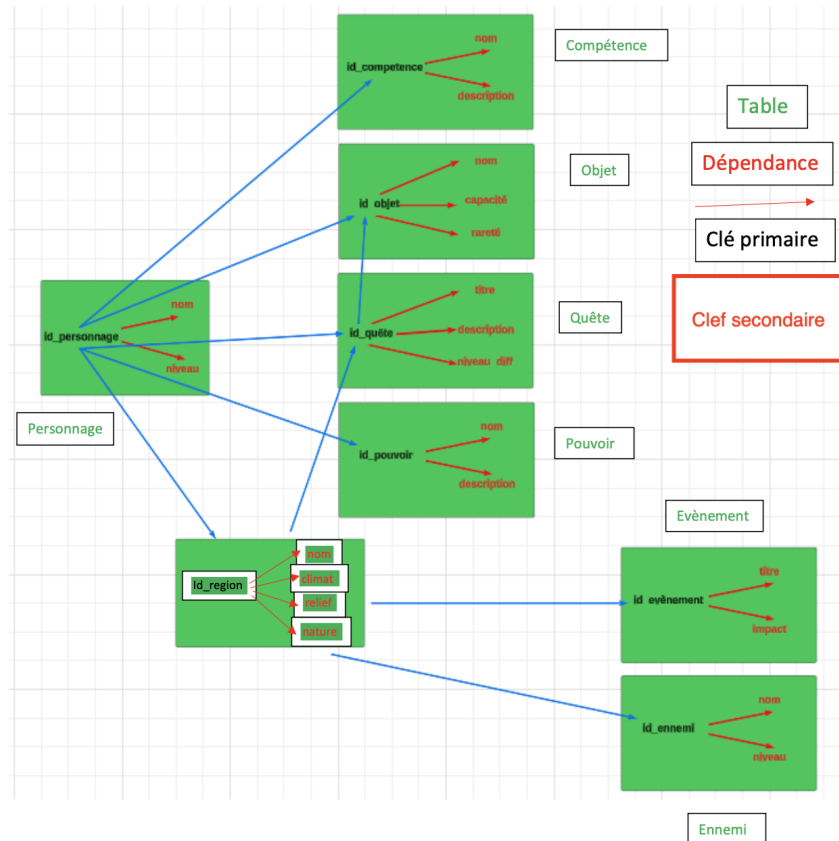


Figure 2: Graphe des dépendances

3 Modèle Relationnel de Données (MRD)

Le modèle relationnel est basé sur le MCD. Voici les tables correspondantes :

3.1 Analyse des tables du MRD

Table Région

Attributs : (*id_region*, *nom*, *climat*, *relief*, *nature*)

- *id_region* est la clé primaire.

- Les attributs *nom*, *climat*, *relief*, *nature* dépendent directement de *id_region* (pas de dépendances transitives).

Conclusion : La table est en 3NF.

Table Modifier

Attributs : (*#id_region*, *#id_evénement*)

- Clé primaire composée : (*#id_region*, *#id_evénement*).

- Les deux attributs sont directement liés à la clé primaire (aucune dépendance transitive).

Conclusion : La table est en 3NF.

Table Événement

Attributs : (*id_evenement, titre, type, impact*)

- *id_evenement* est la clé primaire.
- Tous les autres attributs (*titre, type, impact*) dépendent directement de *id_evenement*.

Conclusion : La table est en 3NF.

Table Ennemi

Attributs : (*id_ennemi, nom, niveau*)

- *id_ennemi* est la clé primaire.
- Les attributs *nom* et *niveau* dépendent directement de *id_ennemi*.

Conclusion : La table est en 3NF.

Table Rôder

Attributs : (*#id_ennemi, #id_region*)

- Clé primaire composée : (*#id_ennemi, #id_region*).
- Pas de dépendances transitives.

Conclusion : La table est en 3NF.

Table Personnage

Attributs : (*id_personnage, nom, niveau, #id_region*)

- *id_personnage* est la clé primaire.
- Tous les autres attributs dépendent directement de *id_personnage*, y compris *nom*.

Conclusion : La table est en 3NF.

Table Objet

Attributs : (*id_objet, nom, pouvoir, rareté, #id_personnage, #id_quête*)

- *id_objet* est la clé primaire.
- Tous les autres attributs, y compris *nom*, dépendent directement de *id_objet*.

Conclusion : La table est en 3NF.

Table Quête

Attributs : (*id_quête, titre, description, niveau_difficulté, #id_region*)

- *id_quête* est la clé primaire.
- Tous les autres attributs dépendent directement de *id_quête*.

Conclusion : La table est en 3NF.

Table Participer

Attributs : (*#id_quête, #id_personnage, statue*)

- Clé primaire composée : (*#id_quête, #id_personnage*).
- *statue* dépend directement de la clé primaire.

Conclusion : La table est en 3NF.

Table Pouvoir

Attributs : (*id_pouvoir*, *nom*, *description*, *#id_personnage*)

- *id_pouvoir* est la clé primaire.
- Les autres attributs dépendent directement de *id_pouvoir*.

Conclusion : La table est en 3NF.

Table Avoir

Attributs : (*#id_compétence*, *#id_personnage*)

- Clé primaire composée : (*#id_compétence*, *#id_personnage*).
- Pas de dépendances transitives.

Conclusion : La table est en 3NF.

Table Compétence

Attributs : (*id_compétence*, *nom*, *description*, *rareté*, *capacité*)

- *id_compétence* est la clé primaire.
- Tous les autres attributs dépendent directement de *id_compétence*.

Conclusion : La table est en 3NF.

3.2 Gestion du conflit lié à *nom*

Pour garantir que toutes les tables respectent la 3NF, la variable *nom* ne doit pas servir de dépendance pour un autre attribut si elle n'est pas une clé primaire. Cela est respecté dans ce MRD car :

- Dans toutes les tables, *nom* est utilisé comme un attribut descriptif qui dépend uniquement de la clé primaire de chaque table.
- Aucune dépendance transitive ou violation de la 3NF ne découle de cet usage.

3.3 Conclusion

Le MCD et le MRD sont conformes à la 3NF.

4 Base de données

La base de données respecte le MRD, et chaque table est dotée de types de données appropriés.

4.1 Création des tables vides

Requêtes SQL pour la gestion des tables

Suppression des tables existantes

```
1 DROP TABLE IF EXISTS Avoir CASCADE;
2 DROP TABLE IF EXISTS Participer CASCADE;
3 DROP TABLE IF EXISTS Roder CASCADE;
4 DROP TABLE IF EXISTS Modifier CASCADE;
5 DROP TABLE IF EXISTS Pouvoir CASCADE;
6 DROP TABLE IF EXISTS Objet CASCADE;
7 DROP TABLE IF EXISTS Personnage CASCADE;
8 DROP TABLE IF EXISTS Qute CASCADE;
9 DROP TABLE IF EXISTS Region CASCADE;
10 DROP TABLE IF EXISTS Competence CASCADE;
11 DROP TABLE IF EXISTS Ennemi CASCADE;
12 DROP TABLE IF EXISTS Evenement CASCADE;
```

Création des tables

Table Evenement

```
1 CREATE TABLE Evenement(
2 id_evenement SERIAL NOT NULL,
3 titre VARCHAR(255) NOT NULL,
4 impacte VARCHAR(255) NOT NULL,
5 PRIMARY KEY (id_evenement)
6 );
```

Table Ennemi

```
1 CREATE TABLE Ennemi(
2 id_ennemi SERIAL NOT NULL,
3 nom VARCHAR(255) NOT NULL,
4 niveau INT NOT NULL,
5 PRIMARY KEY (id_ennemi)
6 );
```

Table Competence


```

1 CREATE TABLE Competence(
2 id_competence SERIAL NOT NULL,
3 nom VARCHAR(255) NOT NULL,
4 capacite VARCHAR(255) NOT NULL,
5 rarete VARCHAR(255) NOT NULL,
6 PRIMARY KEY (id_competence)
7 );

```

Table Region

```

1 CREATE TABLE Region(
2 id_region SERIAL NOT NULL,
3 nom VARCHAR(255) NOT NULL,
4 climat VARCHAR(255) NOT NULL,
5 relief VARCHAR(255) NOT NULL,
6 nature VARCHAR(255) NOT NULL,
7 PRIMARY KEY (id_region)
8 );

```

Table Quête

```

1 CREATE TABLE Qute(
2 id_qute SERIAL NOT NULL,
3 titre VARCHAR(255) NOT NULL,
4 description VARCHAR(255) NOT NULL,
5 niveau_diff INT NOT NULL,
6 id_region INT NOT NULL,
7 PRIMARY KEY (id_qute),
8 FOREIGN KEY (id_region) REFERENCES Region (id_region)
9 );

```

Table Personnage

```

1 CREATE TABLE Personnage(
2 id_personnage SERIAL NOT NULL,
3 nom VARCHAR(255) NOT NULL,
4 niveau INT NOT NULL,
5 id_region INT NOT NULL,
6 PRIMARY KEY (id_personnage),
7 FOREIGN KEY (id_region) REFERENCES Region (id_region)
8 );

```

Table Objet

```

1 CREATE TABLE Objet(
2 id_objet SERIAL NOT NULL,

```

```

3 nom VARCHAR(255) NOT NULL,
4 capacite VARCHAR(255) NOT NULL,
5 rarete VARCHAR(255) NOT NULL,
6 id_personnage INT NOT NULL,
7 id_qute INT NOT NULL,
8 PRIMARY KEY (id_objet),
9 FOREIGN KEY (id_personnage) REFERENCES Personnage (id_personnage),
10 FOREIGN KEY (id_qute) REFERENCES Qute (id_qute)
11 );

```

Table Pouvoir

```

1 CREATE TABLE Pouvoir(
2 id_pouvoir SERIAL NOT NULL,
3 nom VARCHAR(255) NOT NULL,
4 description VARCHAR(255) NOT NULL,
5 id_personnage INT NOT NULL,
6 PRIMARY KEY (id_pouvoir),
7 FOREIGN KEY (id_personnage) REFERENCES Personnage (id_personnage)
8 );

```

Table Participer

```

1 CREATE TABLE Participer(
2 id_qute INT NOT NULL,
3 id_personnage INT NOT NULL,
4 statue VARCHAR(255) NOT NULL,
5 FOREIGN KEY (id_personnage) REFERENCES Personnage (id_personnage),
6 FOREIGN KEY (id_qute) REFERENCES Qute (id_qute)
7 );

```

Table Avoir

```

1 CREATE TABLE Avoir(
2 id_competence INT NOT NULL,
3 id_personnage INT NOT NULL,
4 FOREIGN KEY (id_personnage) REFERENCES Personnage (id_personnage),
5 FOREIGN KEY (id_competence) REFERENCES Competence (id_competence)
6 );

```

Table Modifier

```

1 CREATE TABLE Modifier(
2 id_region INTEGER NOT NULL,
3 id_evenement INTEGER NOT NULL,
4 CONSTRAINT modifier_region_fkey FOREIGN KEY (id_region) REFERENCES Region (
    id_region),

```

```

5 CONSTRAINT modifier_evenement_fkey FOREIGN KEY (id_evenement) REFERENCES
  Evenement (id_evenement)
6 );

```

Table Roder

```

1 CREATE TABLE Roder(
2 id_region INTEGER NOT NULL,
3 id_ennemi INTEGER NOT NULL,
4 CONSTRAINT roder_region_fkey FOREIGN KEY (id_region) REFERENCES Region (
  id_region),
5 CONSTRAINT roder_ennemi_fkey FOREIGN KEY (id_ennemi) REFERENCES Ennemi (
  id_ennemi)
6 );

```

4.2 Remplissage des tables

Nous avons choisit de remplir chaque table avec 10 items

Requêtes SQL pour remplir les tables

Table competence

```

1 INSERT INTO competence (nom, capacite, rarete) VALUES
2 ('Tir 1 arc', 'Tireur précis', 'Rare'),
3 ('Matrise de 1 pe', 'Matre piste', 'Epique'),
4 ('Alchimie', 'Cratation de potions', 'Rare'),
5 ('Discretion', 'Dplacement silencieux', 'Lgendaire'),
6 ('Magie de feu', 'Lancer de flammes', 'Epique'),
7 ('Gurison', 'Rgnration de sant', 'Rare'),
8 ('Magie de glace', 'Conglation instantane', 'Epique'),
9 ('Magie de la lumire', 'Bouclier lumineux', 'Epique'),
10 ('Invocation de fes', 'Appelle une fe', 'Lgendaire'),
11 ('Saut hroque', 'Double saut avec rotation', 'Rare');

```

Table ennemi

```

1 INSERT INTO ennemi (nom, niveau) VALUES
2 ('Moblin', 1),
3 ('Bokoblin', 2),
4 ('Lizalfos', 5),
5 ('Stalfos', 3),
6 ('Ganondorf', 10),
7 ('Dark Link', 9),
8 ('ReDead', 6),
9 ('Hinox', 8),
10 ('Keese', 2),

```

```
11 ('Gohma', 7);
```

Table evenement

```
1 INSERT INTO evenement (titre, impacte) VALUES
2 ('Lune de sang', 'Ressuscite les ennemis'),
3 ('Eruption volcanique', 'Renforce la magie de feu'),
4 ('Pluie torrentielle', 'Rduit l agilit'),
5 ('Tremblement de terre', 'Affaiblit les structures'),
6 ('Eclipse solaire', 'Rduit la magie'),
7 ('Tempte de sable', 'Rduit la visibilit'),
8 ('Aurore borale', 'Renforce la gurison'),
9 ('Vagues gantes', 'Dsactive les ports'),
10 ('Chute d toiles', 'Augmente les objets rares'),
11 ('Jour sans fin', 'Renforce les competences');
```

Table region

```
1 INSERT INTO region (nom, climat, relief, nature) VALUES
2 ('Plaine d Hyrule', 'Tempr', 'Plaines', 'Herbe'),
3 ('Dsert Gerudo', 'Aride', 'Dunes', 'Cactus'),
4 ('Montagne de la Mort', 'Chaud', 'Volcan', 'Lave'),
5 ('Fort de Firone', 'Humide', 'Fort', 'Tropiques'),
6 ('Valle de Zora', 'Froid', 'Cascade', 'Rivire'),
7 ('Village Cocorico', 'Tempr', 'Colline', 'Prairies'),
8 ('Marcages Korogu', 'Humide', 'Boues', 'Marcages'),
9 ('le de l Eveil', 'Doux', 'Plage', 'Cocotiers'),
10 ('Temple du Temps', 'Tempr', 'Ruines', 'Anciennes structures'),
11 ('Forteresse Maudite', 'Sombre', 'Collines', 'Terrains hostiles');
```

Table personnage

```
1 INSERT INTO personnage (nom, niveau, id_region) VALUES
2 ('Link', 10, 1),
3 ('Zelda', 9, 2),
4 ('Mipha', 7, 3),
5 ('Daruk', 6, 4),
6 ('Revali', 8, 5),
7 ('Urbosa', 9, 6),
8 ('Impa', 7, 7),
9 ('Tingle', 3, 8),
10 ('Epona', 4, 9),
11 ('Rauru', 8, 10);
```

Table quête

```

1 INSERT INTO "qute" (titre, description, niveau_diff, id_region) VALUES
2 ('Sauver Zelda', 'Empcher Ganondorf d enlever Zelda.', 10, 1),
3 ('Trouver l Epe de Lgende', 'Rcuprer l pe dans le Temple du Temps.', 9, 10),
4 ('Vaincre le Roi Moblin', 'Eliminer le Roi Moblin dans la Forteresse Maudite.'
5 , 8, 9),
6 ('Escorte de caravane', 'Amener des provisions au Village Cocorico.', 3, 6),
7 ('Activer la Tour de Firone', 'Rparer la tour dans la Fort de Firone.', 5, 4),
8 ('Apaiser Vah Ruta', 'Calmer la Bte Divine Vah Ruta.', 7, 3),
9 ('Djouer l embuscade', 'Protger le Village des attaques.', 6, 7),
10 ('Rallumer les flammes sacres', 'Rallumer les flammes du Mont de la Mort.', 9,
11 5),
12 ('Traverser les marcages', 'Trouver un passage scuris.', 4, 7),
13 ('Protger l Archipel', 'liminer les ennemis sur l le de l Eveil.', 4, 8);

```

Table pouvoir

```

1 INSERT INTO pouvoir (nom, description, id_personnage) VALUES
2 ('Bouclier Hylien', 'Augmente la dfense contre les projectiles.', 1),
3 ('Pouvoir de lumire', 'Inflige des dgts massifs aux ennemis des tnbres.', 2),
4 ('Bndiction aquatique', 'Soigne les blessures au contact de l eau.', 3),
5 ('Force brute', 'Permet de soulever des rochers normes.', 4),
6 ('Vitesse du vent', 'Augmente la vitesse de dplacement.', 5),
7 ('Danse Gerudo', 'Distrait les ennemis proches.', 6),
8 ('Lame spectrale', 'Frappe avec une lame fantomatique.', 7),
9 ('Explosion magique', 'Inflige une onde de choc magique.', 8),
10 ('Invocation d Epona', 'Invoque un cheval fidle.', 9),
11 ('Rgnration sacre', 'Restaure lentement la vie.', 10);

```

Table objet

```

1 INSERT INTO objet (nom, capacite, rarete, id_personnage, id_qute) VALUES
2 ('Epe de Lgende', 'Augmente les dgts contre les ennemis des tnbres.', '
3 Lgendaire', 1, 2),
4 ('Arc de Lumire', 'Tire des flches de lumire.', 'Epique', 2, 1),
5 ('Trident Zora', 'Amliore les attaques sous l eau.', 'Rare', 3, 6),
6 ('Bouclier des Gorons', 'Protge contre les ruptions volcaniques.', 'Rare', 4,
7 5),
8 ('Plume Rito', 'Rduit les dgts des chutes.', 'Rare', 5, 9),
9 ('Epe Gerudo', 'Augmente les dgts au corps corps.', 'Epique', 6, 10),
10 ('Grimoire Korogu', 'Permet d utiliser des pouvoirs forestiers.', 'Rare', 7,
11 8),
12 ('Cape d ombre', 'Rend invisible pendant un court moment.', 'Lgendaire', 8, 3),
13 ('Harnais d Epona', 'Augmente la vitesse du cheval.', 'Rare', 9, 4),
14 ('Bracelet de force', 'Double la puissance physique.', 'Epique', 10, 7);

```

Table avoir

```
1 INSERT INTO avoir (id_competence, id_personnage) VALUES
2 (1, 1), (2, 2), (3, 3), (4, 4), (5, 5),
3 (6, 6), (7, 7), (8, 8), (9, 9), (10, 10);
```

Table modifier

```
1 INSERT INTO modifier (id_region, id_evenement) VALUES
2 (1, 1), (2, 2), (3, 3), (4, 4), (5, 5),
3 (6, 6), (7, 7), (8, 8), (9, 9), (10, 10);
```

Table participer

```
1 INSERT INTO participer (id_qute, id_personnage, statue) VALUES
2 (1, 1, 'Termin'), (2, 2, 'En cours'), (3, 3, 'Echou'),
3 (4, 4, 'Termin'), (5, 5, 'Termin'), (6, 6, 'En cours'),
4 (7, 7, 'Echou'), (8, 8, 'Termin'), (9, 9, 'En cours'),
5 (10, 10, 'Termin');
```

Table roder

```
1 INSERT INTO roder (id_region, id_ennemi) VALUES
2 (1, 1), (2, 2), (3, 3), (4, 4), (5, 5),
3 (6, 6), (7, 7), (8, 8), (9, 9), (10, 10);
```

5 Scénarios d'utilisation et requêtes

Requêtes SQL avec indexation cohérente

1. Listez tous les personnages et leurs niveaux

```
1 SELECT nom, niveau
2 FROM personnage;
```

2. Affichez les compétences et leur rareté

```
1 SELECT nom, rarete
2 FROM competence;
```

3. Trier les ennemis par leur niveau, du plus fort au plus faible

```
1 SELECT nom, niveau
2 FROM ennemi
3 ORDER BY niveau DESC;
```

4. Récupérez les titres et impacts des événements

```
1 SELECT titre, impacte
2 FROM evenement;
```

5. Trouver les objets de rareté "Rare"

```
1 SELECT nom, capacite
2 FROM objet
3 WHERE rarete = 'Rare';
```

6. Afficher les personnages qui participent aux quêtes avec le titre de la quête

```
1 SELECT p.nom AS personnage, q.titre AS qute
2 FROM participer pa
3 JOIN personnage p ON pa.id_personnage = p.id_personnage
4 JOIN "qute" q ON pa.id_qute = q.id_qute;
```

7. Trouver tous les objets possédés par "Link"

```
1 SELECT o.nom, o.capacite
2 FROM objet o
3 JOIN personnage p ON o.id_personnage = p.id_personnage
4 WHERE p.nom = 'Link';
```

8. Lister les compétences "Légendaires" et les personnages correspondants

```
1 SELECT p.nom AS personnage, c.nom AS competence
2 FROM avoir a
3 JOIN personnage p ON a.id_personnage = p.id_personnage
4 JOIN competence c ON a.id_competence = c.id_competence
5 WHERE c.rarete = 'Lgendaire';
```

9. Récupérez les quêtes disponibles dans la région "Plaine d'Hyrule"

```
1 SELECT q.titre, q.description
2 FROM "qute" q
3 JOIN region r ON q.id_region = r.id_region
4 WHERE r.nom = 'Plaine d Hyrule';
```

10. Afficher les ennemis présents dans chaque région

```
1 SELECT r.nom AS region, e.nom AS ennemi
2 FROM roder ro
3 JOIN region r ON ro.id_region = r.id_region
4 JOIN ennemi e ON ro.id_ennemi = e.id_ennemi;
```

11. Augmenter de 1 le niveau des personnages qui participent à des quêtes trop difficiles pour eux

```
1 UPDATE personnage
2 SET niveau = niveau + 1
3 WHERE id_personnage IN (
4     SELECT p.id_personnage
5     FROM participer p
6     JOIN "qute" q ON p.id_qute = q.id_qute
7     WHERE q.niveau_diff > (SELECT niveau FROM personnage WHERE id_personnage =
8         p.id_personnage)
9 );
```


12. Lister les régions où des événements renforcent quelque chose

```
1 SELECT r.nom, e.titre
2 FROM modifier m
3 JOIN region r ON m.id_region = r.id_region
4 JOIN evenement e ON m.id_evenement = e.id_evenement
5 WHERE e.impacte LIKE 'Renforce%';
```

13. Afficher tous les personnages et leurs pouvoirs

```
1 SELECT p.nom AS personnage, po.nom AS pouvoir, po.description
2 FROM pouvoir po
3 JOIN personnage p ON po.id_personnage = p.id_personnage;
```

14. Trouver les quêtes associées à des objets de rareté "Légendaire"

```
1 SELECT q.titre, o.nom AS objet
2 FROM "qute" q
3 JOIN objet o ON q.id_qute = o.id_qute
4 WHERE o.rarete = 'Lgendaire';
```

15. Lister les personnages ayant des compétences "Épiques" ou "Légendaires"

```
1 SELECT DISTINCT p.nom
2 FROM personnage p
3 JOIN avoir a ON p.id_personnage = a.id_personnage
4 JOIN competence c ON a.id_competence = c.id_competence
5 WHERE c.rarete IN ('Epique', 'Lgendaire');
```

16. Trouver les personnages qui possèdent un objet que "Zelda" possède également

```
1 SELECT DISTINCT p1.nom
2 FROM personnage p1
3 JOIN objet o1 ON p1.id_personnage = o1.id_personnage
4 WHERE o1.nom IN (
5     SELECT o2.nom
6     FROM objet o2
7     JOIN personnage p2 ON o2.id_personnage = p2.id_personnage
8     WHERE p2.nom = 'Zelda'
9 );
```

17. Supprimer les participations des personnages ayant échoué à des quêtes trop difficiles

```
1 DELETE FROM participer
2 WHERE statue = 'Echou'
3 AND id_personnage IN (
4     SELECT p.id_personnage
5     FROM participer p
6     JOIN "qute" q ON p.id_qute = q.id_qute
7     WHERE q.niveau_diff > (SELECT niveau FROM personnage WHERE id_personnage =
8         p.id_personnage)
9 );
```

18. Lister les régions qui n'ont aucune quête associée

```
1 SELECT r.nom
2 FROM region r
3 WHERE NOT EXISTS (
4     SELECT 1
5     FROM "qute" q
6     WHERE q.id_region = r.id_region
7 );
```

19. Afficher les personnages et leurs objets, triés par rareté décroissante

```
1 SELECT p.nom AS personnage, o.nom AS objet, o.rarete
2 FROM objet o
3 JOIN personnage p ON o.id_personnage = p.id_personnage
4 ORDER BY o.rarete DESC;
```

20. Afficher les personnages et leurs compétences triées par rareté décroissante

```
1 SELECT p.nom AS personnage, c.nom AS competence, c.rarete
2 FROM avoir a
3 JOIN personnage p ON a.id_personnage = p.id_personnage
4 JOIN competence c ON a.id_competence = c.id_competence
5 ORDER BY c.rarete DESC;
```

21. Listez les noms de tous les personnages dans la base de données

```
1 SELECT nom
2 FROM personnage;
```

22. Ajoutez un nouvel objet nommé "Potion de Mana" et un pouvoir de "restauration de magie"

```
1 INSERT INTO objet (nom, capacite, rarete, id_personnage, id_quete)
2 VALUES ('Potion de Mana', 'restauration de magie', 'Rare', 1, 1);
```

23. Supprimez un ennemi appelé "Gobelin" du registre des ennemis

```
1 DELETE FROM ennemi
2 WHERE nom = 'Gobelin';
```

24. Augmentez de 1 le niveau d'un personnage nommé "Link"

```
1 UPDATE personnage
2 SET niveau = niveau + 1
3 WHERE nom = 'Link';
```

25. Listez les personnages avec leur région de résidence

```
1 SELECT p.nom AS personnage, r.nom AS region
2 FROM personnage p
3 JOIN region r ON p.id_region = r.id_region;
```

26. Ajoutez une région "Château d'Hyrule" si elle n'existe pas

```
1 INSERT INTO region (nom, climat, relief, nature)
2 VALUES ('Chateau d Hyrule', 'Tempr', 'Montagneux', 'Chateau');
```

27. Listez des objets rares possédés par un personnage nommé "Link"

```
1 SELECT o.nom, o.rarete
2 FROM objet o
3 JOIN personnage p ON o.id_personnage = p.id_personnage
4 WHERE p.nom = 'Link' AND o.rarete = 'Rare';
```

28. Augmentez le niveau de tous les personnages qui participent à une quête d'un niveau supérieur à leur propre niveau

```

1 UPDATE personnage
2 SET niveau = niveau + 1
3 WHERE id_personnage IN (
4     SELECT p.id_personnage
5     FROM participer p
6     JOIN "qute" q ON p.id_qute = q.id_qute
7     WHERE q.niveau_diff > (SELECT niveau FROM personnage WHERE id_personnage =
8         p.id_personnage)
9 );

```

29. Listez les personnages qui possèdent un objet que "Zelda" possède également

```

1 SELECT DISTINCT p1.nom AS personnage
2 FROM personnage p1
3 JOIN objet o1 ON p1.id_personnage = o1.id_personnage
4 WHERE o1.nom IN (
5     SELECT o2.nom
6     FROM objet o2
7     JOIN personnage p2 ON o2.id_personnage = p2.id_personnage
8     WHERE p2.nom = 'Zelda'
9 );

```

30. Supprimez toutes les participations aux quêtes pour les personnages ayant échoué une quête d'un niveau supérieur au leur

```

1 DELETE FROM participer
2 WHERE statue = 'Echou'
3 AND id_personnage IN (
4     SELECT p.id_personnage
5     FROM participer p
6     JOIN "qute" q ON p.id_qute = q.id_qute
7     WHERE q.niveau_diff > (SELECT niveau FROM personnage WHERE id_personnage =
8         p.id_personnage)
9 );

```