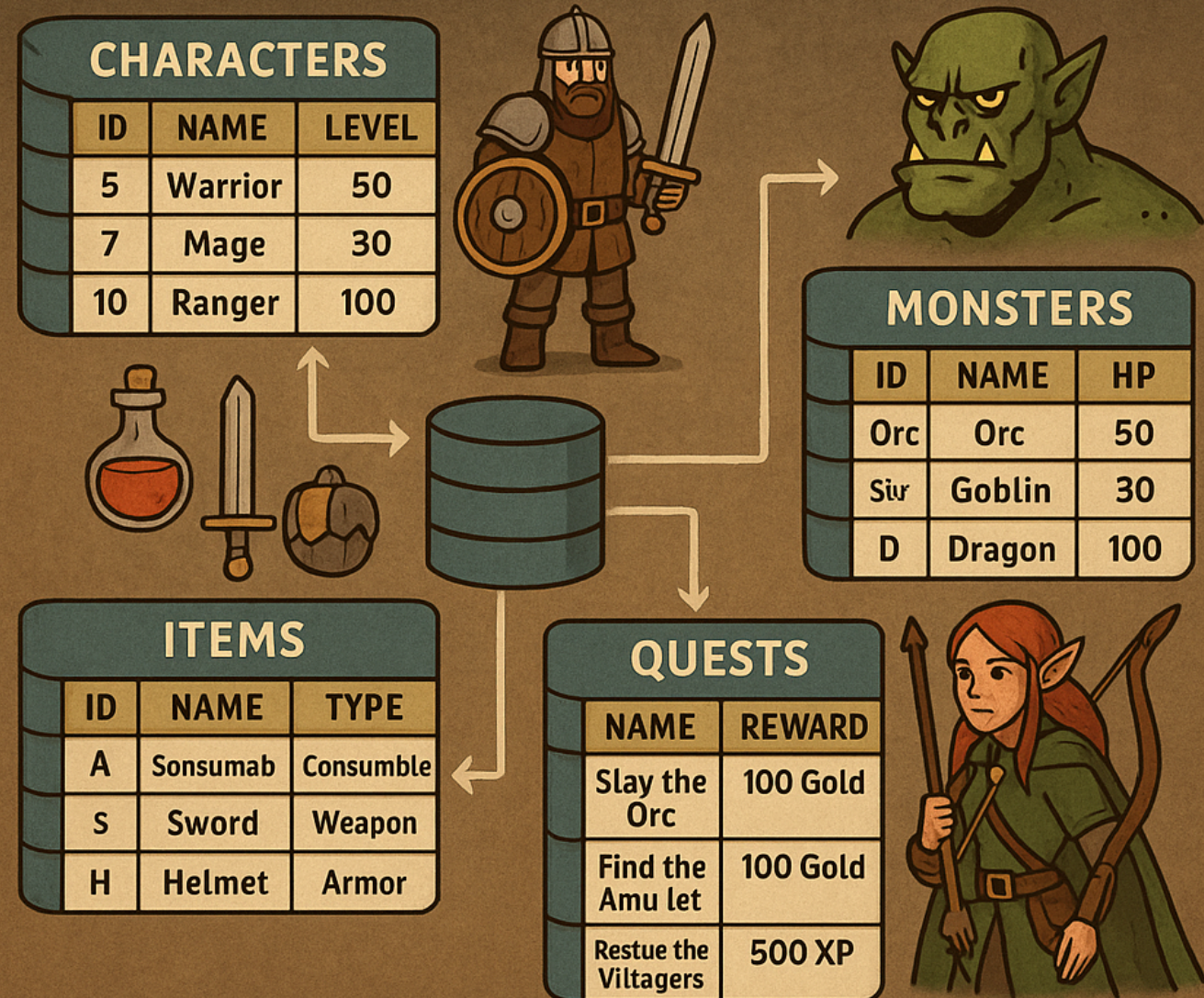


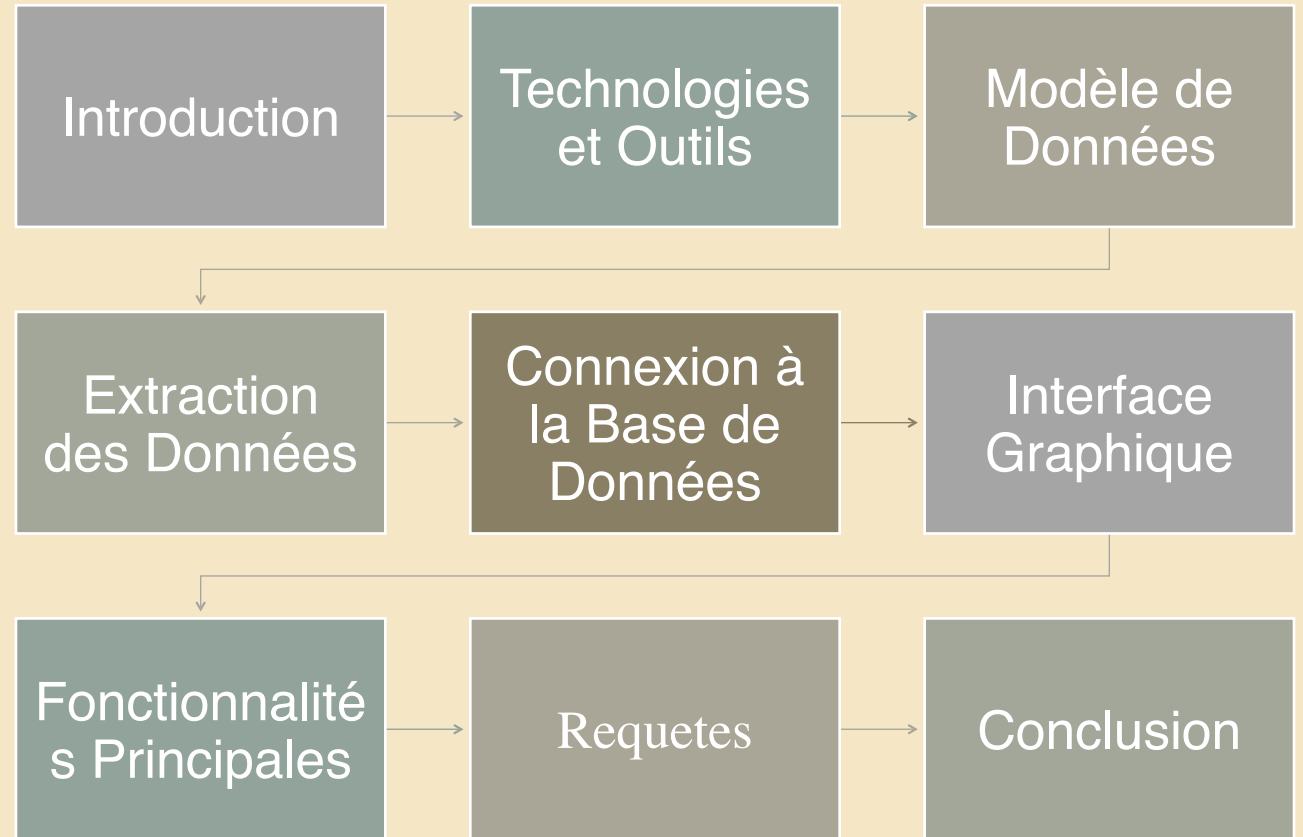
# PROJET DE SYSTÈME D'INVENTAI RE RPG

*Présenté par: MEHDI  
EL MARDAH,  
HASSAN AMRI, David  
Poplawski, Dat Bui The*

# ROLE-PLAYING GAME DATABASE SYSTEM

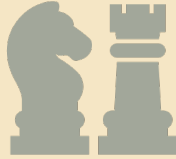


# PLAN





# INTRODUCTION



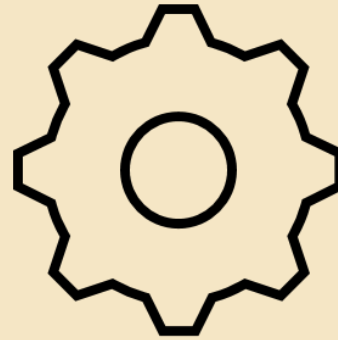
Projet: Gestion d'inventaire  
pour un jeu de rôle (RPG)



Objectif: Permettre aux  
joueurs de visualiser et  
gérer leur inventaire



Fonctionnalités:  
chargement des données,  
gestion des personnages,  
quêtes, et items



# TECHNOLOGIES ET OUTILS



Python 3.x



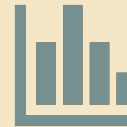
MySQL 8+



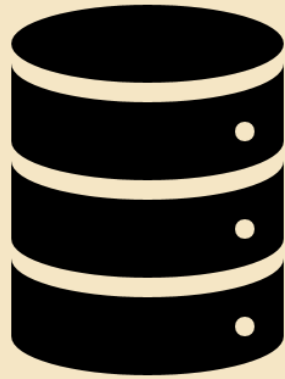
mysql-connector-  
python



python-pptx pour  
la génération de  
présentations



Tkinter (ou autre)  
pour l'interface  
graphique



# MODÈLE DE DONNÉES

- Tables principales:
  - - Player (ID, NomUtilisateur, Niveau, XP, Monnaie, SlotsInventaire)
  - - Character (ID, Nom, Classe, Caractéristiques, Référence Player)
  - - Monster (ID, Nom, Type, Niveau, PointsDeVie)
  - - NPC (ID, Nom, Rôle, Dialogue)
  - - Quest (ID, NomQuête, Objectif, Récompense)
  - - Item (ID, NomObjet, Type, Attributs)
- Relations: Un joueur peut avoir plusieurs personnages, quêtes, items



# EXTRACTION DES DONNÉES

- Fichiers source:
  - - players.csv (chargé par loadPlayerData)
  - - monsters.xml (chargé par loadMonsterData)
  - - npcs.json (chargé par loadNpcData)
  - - quests.xml (chargé par loadQuestData)
  - - Autres CSV pour items, caractéristiques, etc.
- Validation et insertion dans MySQL via mysql-connector
- Détails des méthodes d'extraction :
  - - loadPlayerData : Lecture de players.csv avec le module csv: parsing ligne par ligne, validation des champs (types, valeurs attendues), création d'objets Player et insertion via requêtes SQL.
  - - loadMonsterData : Parsing de monsters.xml avec xml.etree.ElementTree: itération sur les éléments <monster>, extraction des attributs, validation et insertion dans la table Monster.
  - - loadNpcData : Chargement de npcs.json avec le module json: parcours des objets, fusion des champs nom et caractères, vérification des formats, insertion dans la table NPC.
  - - loadQuestData : Parsing de quests.xml similaire à monsters.xml: extraction des balises <quest>, récupération des champs (objectif, récompense), validation et insertion dans la table Quest.
  - - Extraction des items : Lecture des fichiers CSV correspondants: parsing avec csv, gestion des attributs spécifiques aux items (type, attributs), validation et insertion dans la table Item.
  - - Validation des données : Vérification globale après extraction: contrôle de cohérence (références entre tables), gestion des erreurs (logs, rejet des enregistrements invalides).

# CONNEXION À LA BASE DE DONNÉES

- Classe DataBase (src/db\_utils/DataBase.py)
  - - Méthode connectToDatabase(): établir connexion MySQL
  - - Méthode close\_connection(): fermer la connexion
- Utilisation du cursor pour exécuter les requêtes d'insertion et de consultation

# INTERFACE GRAPHIQUE

- Fichiers principaux:
  - - MainGui.py: point d'entrée de l'application GUI
  - - MenuManager.py: gestion des menus principaux
  - - Écrans: intro\_screen.py, Quest\_screen.py, profile.py, etc.
- Modèles: Player, Character, Monster, NPC, Quest (dans gui/models)
- Service: gestion des opérations CRUD sur la base de données

# FONCTIONNALITÉS PRINCIPALES

- Charger et afficher l'inventaire du joueur
- Afficher les quêtes disponibles et en cours
- Gestion des interactions avec les PNJ (NPC)
- Ajout et suppression d'objets dans l'inventaire
- Mise à jour des niveaux et de l'XP des personnages



# REQUETES

- 1 Les 10 joueurs ayant le plus d'or.
- 2 Le joueur ayant le plus de personnages de la même classe.
- 3 La quête ayant la plus grosse récompense en or par niveau de difficulté.
- 4 Le PNJ possédant l'inventaire contenant les objets dont la valeur en or cumulée est la plus importante.
- 5 Le type d'objet (arme, armure, potion ou relique le plus souvent offert en récompense de quêtes de niveau 5.
- 6 Les monstres avec les meilleures récompenses en valeur en or cumulée en fonction de leurs points de vie

# 1

- SELECT ID , UserName ,W a l l e t C r e d i t s AS Gold
- FROM P l a y e r
- ORDER BY W a l l e t C r e d i t s DESC
- LIMIT 1 0 ;

## 2

- SELECT p . UserName , c . Class , COUNT( \* ) AS NumberOfCharacters
- FROM P l a y e r p
- JOIN Ch aract erTa ble c ON p . ID = c . PlayerID
- GROUP BY p . UserName , c . C l a s s
- ORDER BY NumberOfCharacters DESC
- LIMIT 1 ;

### 3

- SELECT q . D i f f i c u l t y L e v e l , q . QuestName , qg . GoldAmount
- FROM Quest q , QuestGold qg
- WHERE q . ID = qg . QuestID
- AND ( q . D i f f i c u l t y L e v e l , qg . GoldAmount ) IN (
- SELECT q2 . D i f f i c u l t y L e v e l , MAX( qg2 . GoldAmount )
- FROM Quest q2 , QuestGold qg2
- WHERE q2 . ID = qg2 . QuestID
- GROUP BY q2 . D i f f i c u l t y L e v e l
- ) ;

## 4

- SELECT n . ID , n . NpcName ,
- SUM( i . Quantity \* o . P r i c e ) AS V a l e u r T o t a l e FROM NPC AS n
- JOIN NPCInventory AS i ON i . NPCID = n . ID
- JOIN ObjectTest AS o ON o . ObjectName = i . ObjectName
- GROUP BY n . ID , n . NpcName
- ORDER BY V a l e u r T o t a l e DESC
- LIMIT 1 ;

# 5

- SELECT o . Type , COUNT( \* ) AS Frequency
- FROM Quest q
- JOIN Quest\_Objects qo ON q . ID = qo . QuestID
- JOIN ObjectTest o ON qo . ObjectName = o . ObjectName
- WHERE q . D i f f i c u l t y L e v e l = 5
- GROUP BY o . Type
- ORDER BY Frequency DESC
- LIMIT 1

## 6

- SELECT bi.BeastName, bi.LifePoints,
- IF NULL(mg.GoldAmount, 0) + IFNULL(SUM(ot.Price \* r.Quantity), 0) AS TotalGoldValue,
- ROUND((IFNULL(mg.GoldAmount, 0) + IFNULL(SUM(ot.Price \* r.Quantity), 0)) / bi.LifePoints, 2) AS GoldPerHP
- FROM Bestiary bi
- LEFT JOIN MonsterGold mg ON bi.ID = mg.MonsterID
- LEFT JOIN Rewards r ON bi.ID = r.MonsterID
- LEFT JOIN ObjectTest ot ON r.ObjectName = ot.ObjectName
- GROUP BY bi.ID, bi.BeastName, bi.LifePoints
- ORDER BY GoldPerHP DESC
- LIMIT 10;

# CONCLUSION



- Projet fonctionnel de gestion d'inventaire RPG
- Mise en place d'une base de données relationnelle MySQL
- Interface graphique pour une interaction utilisateur intuitive
- Possibilités d'évolution: ajouter multijoueur, meilleure gestion des quêtes

**QUESTIONS?**