

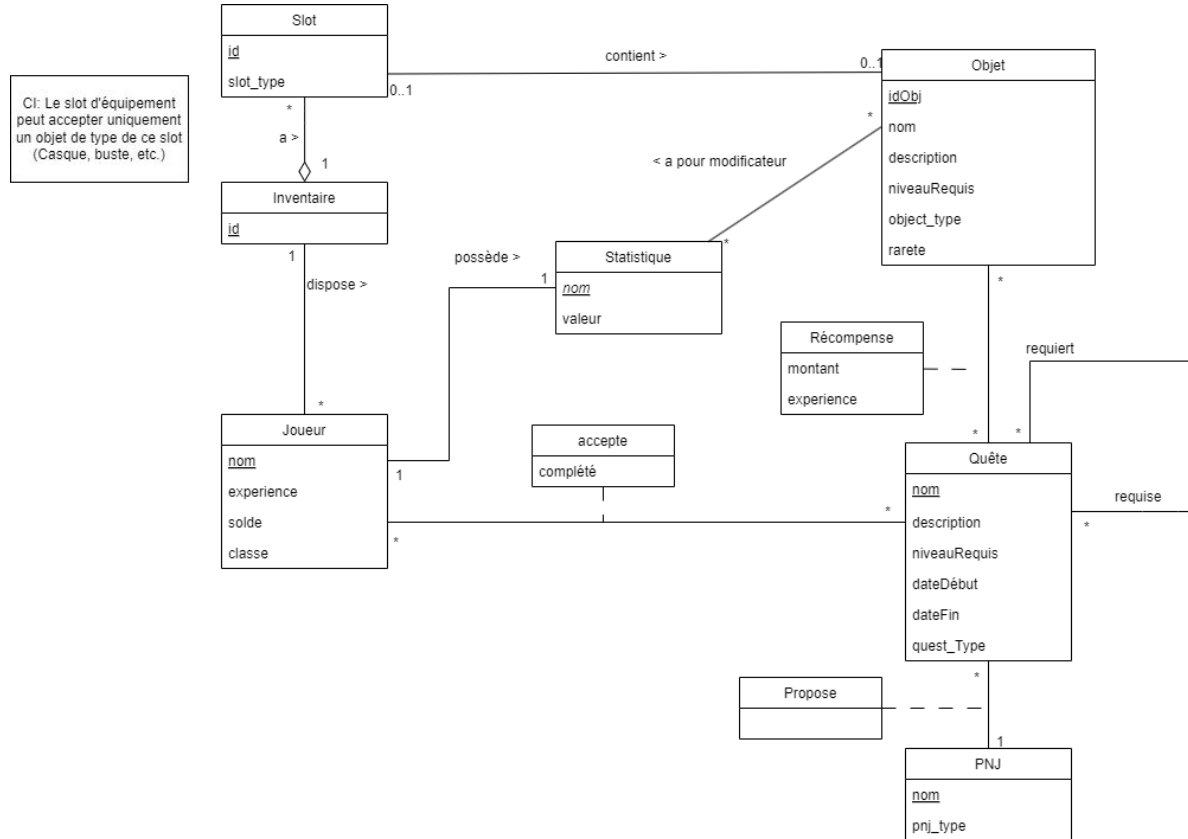
Présentation BDR

Florian Duruz, Yoann Changanagui

? Problématique ?



Schéma conceptuelle



Trigger/Vue/Fonction Armurerie

```
-- Vue pour lister tous les joueurs avec leurs inventaires et les objets qu'ils possèdent
CREATE OR REPLACE VIEW vw_joueurs_inventaire AS
SELECT joueur.nom AS nom_joueur, o.*, s.type AS type_slot FROM slot AS s
    INNER JOIN inventaire
    ON s.id_inventaire = inventaire.id
    INNER JOIN joueur
    ON inventaire.id = joueur.id_inventaire
    INNER JOIN objet AS o
    ON s.id_objet = o.id
    INNER JOIN statistique
    ON o.id_statistique = statistique.id;
```

Trigger/Vue/Fonction Armurerie (2)

```
-- Vue pour lister toutes les statistiques des joueurs
CREATE VIEW vw_joueurs_statistiques AS
SELECT joueur.nom AS nom_joueur, s.* FROM joueur
    INNER JOIN statistique s
    ON joueur.id_statistique = s.id;
```

Trigger/Vue/Fonction Armurerie (3)

-- Trigger to update the statistics of a player when an object is updated in his inventory OK

```
create function update_player_statistic() returns trigger
```

```
language plpgsql
```

```
as
```

```
$$
```

```
BEGIN
```

```
UPDATE statistique
```

```
SET intelligence = (SELECT SUM(s.intelligence)
```

```
FROM Slot sl
```

```
JOIN Objet o ON sl.id_objet = o.id
```

```
JOIN Statistique s ON o.id_statistique = s.id
```

```
WHERE sl.id_inventaire = NEW.id_inventaire AND sl.type = 'Equipe
```

```
ment'),
```

```
agilite = (SELECT SUM(s.agilite)
```

```
FROM Slot sl
```

```
JOIN Objet o ON sl.id_objet = o.id
```

```
JOIN Statistique s ON o.id_statistique = s.id
```

```
WHERE sl.id_inventaire = NEW.id_inventaire AND sl.type = 'Equipe
```

```
ment'),
```

```
force = (SELECT SUM(s.force)
```

```
FROM Slot sl
```

```
JOIN Objet o ON sl.id_objet = o.id
```

```
JOIN Statistique s ON o.id_statistique = s.id
```

```
WHERE sl.id_inventaire = NEW.id_inventaire AND sl.type = 'Equipe
```

```
ment'),
```

```
endurance = (SELECT SUM(s.endurance)
```

```
FROM Slot sl
```

```
WHERE id = NEW.id_inventaire;
```

```
RETURN NEW;
```

```
END;
```

```
CREATE TRIGGER trg_update_player_statistic
```

```
AFTER UPDATE ON Slot
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION update_player_statistic();
```

Trigger/Fonction Quêtes

```
-- Get Available quests for a player
CREATE OR REPLACE FUNCTION get_available_quests_for_player(
    player_name VARCHAR,
    player_experience DECIMAL
)
RETURNS TABLE(
    nom VARCHAR,
    description VARCHAR,
    niveauRequis DECIMAL,
    dateDebut TIMESTAMP,
    dateFin TIMESTAMP,
    type quest_type,
    nom_quete_requise VARCHAR
)
LANGUAGE plpgsql
AS $$
BEGIN
    RETURN QUERY
    SELECT q.*
    FROM Quete q
    LEFT JOIN Accepte a ON a.nom_quete = q.nom
    WHERE a.nom_quete IS NULL
    AND q.niveauRequis <= player_experience
    AND (q.nom_quete_requise IS NULL OR q.nom_quete_requise IN (
        SELECT nom_quete
        FROM Accepte
        WHERE nom_joueur = player_name AND complete = true
    ));
END;
$$;
```


Trigger/Fonction Quêtes

```
--Check des dépendences des quête du joueur
CREATE OR REPLACE FUNCTION check_and_delete_dependent_quest()
RETURNS TRIGGER AS
$$
BEGIN
    DELETE FROM Accepte
    WHERE nom_quete IN (
        SELECT nom
        FROM Quete
        WHERE nom_quete_requise = OLD.nom_quete
    )
    AND nom_joueur = OLD.nom_joueur;
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER delete_dependent_quest
AFTER DELETE ON Accepte
FOR EACH ROW
EXECUTE FUNCTION check_and_delete_dependent_quest();
```

Technologies Utilisées

- Docker
- Github
- Java, JavaLin, jte
- PostgreSQL
- Drawio

Démo de l'application

Défi Rencontré

- Erreurs dans la conception de la base de données demandant de faire des micros-ajustements à chaque nouvelle étape.
- Apprentissage de javalin (*peu de documentation*)
- Apprentissage de jte permettant le rendu dynamique de nos pages HTML
- Temps de développement très long en plus de devoir ajuster nos triggers/vues, BDD, etc.

Synthèse des fonctionnalités : Quêtes joueurs

- Listage des quêtes : disponible, en cours, complétées
- Gestions des dépendances entre quêtes
- Ajout manuel de quête à un joueur
- Marquer une quête comme complétée via un bouton
- Suppression d'une quête en cours ou complétée (pour tester les triggers)

Synthèse des fonctionnalités : Armurerie

- Lister l'inventaire d'un joueur (Equipements et sac)
- Lister les statistiques d'un joueur
- Equiper un objet de l'inventaire (*Ajout des statistiques de l'objet au joueur*)
- Déséquiper un objet de l'inventaire (*Retrait des statistiques au joueur*)
- Le joueur peut ajouter les objets qu'il n'a pas dans son inventaire

Conclusion