

Titre: Livrable 3_Projet_BDD

Participants: Martrenchar Luc, Doyhenart Xan, Abouleila Selim

- Le prompt utilisé pour le LLM pour insérer des données d'exemple, et le script d'insertion généré.

```
-- 1. Insertion dans les tables "parent" (sans clés étrangères)

-- Table Jeu
INSERT INTO Jeu (id_jeu, nom, description, annee_publication, min_joueurs, max_joueurs, temps_jeu, age_min)
VALUES
  (1, 'Catan', 'Un jeu de placement et de commerce sur une île en développement.', 1995, 3, 4, 60, 10),
  (2, 'Ticket to Ride', 'Un jeu de construction de lignes de chemin de fer à travers différents pays.', 2004, 2, 5, 45, 8),
  (3, '7 Wonders', 'Un jeu de draft de cartes pour construire les plus grandes merveilles de l'Antiquité.', 2010, 2, 7, 30, 10),
  (4, 'Carcassonne', 'Un jeu de placement de tuiles pour construire un paysage médiéval.', 2000, 2, 5, 45, 8);

-- Table Utilisateur
INSERT INTO Utilisateur (id_utilisateur, pseudo, email, mot_de_passe, date_inscription)
VALUES
  (1, 'Alice', 'alice@example.com', 'motdepasse1', '2023-01-15'),
  (2, 'Bob', 'bob@example.com', 'motdepasse2', '2023-02-10'),
  (3, 'Charlie', 'charlie@example.com', 'motdepasse3', '2023-03-20'),
  (4, 'Diana', 'diana@example.com', 'motdepasse4', '2023-04-01');

-- Table Contributeur
INSERT INTO Contributeur (id_contributeur, nom, role)
VALUES
  (1, 'Klaus Teuber', 'designer'),
  (2, 'Days of Wonder', 'editeur'),
  (3, 'Alan R. Moon', 'designer');
```

```

-- Table Metadata
-- type_metadata exemples : 'categorie', 'mecanique', 'famille', 'tag', 'theme', etc.
INSERT INTO Metadata (id_metadata, nom, type_metadata)
VALUES
    (1, 'Stratégie', 'categorie'),
    (2, 'Famille', 'categorie'),
    (3, 'Drafting', 'mecanique'),
    (4, 'Placement de tuiles', 'mecanique'),
    (5, 'Négociation', 'mecanique'),
    (6, 'Transport', 'theme');

-- Table Badge
INSERT INTO Badge (id_badge, titre, description)
VALUES
    (1, 'Top Reviewer', 'Attribué aux utilisateurs qui rédigent de nombreux commentaires de qualité.'),
    (2, 'Early Adopter', 'Attribué aux utilisateurs qui rejoignent la plateforme durant sa première année.'),
    (3, 'Veteran Member', 'Attribué aux utilisateurs inscrits depuis plus de deux ans.');
```

```

-- Table Commentaire
INSERT INTO Commentaire (id_commentaire, id_jeu, id_utilisateur, contenu, note, date_creation)
VALUES
    (1, 1, 1, 'Excellent jeu pour découvrir le jeu de plateau moderne.', 4.5, '2023-02-01 10:15:00'), -- Catan par Alice
    (2, 1, 2, 'Très sympa mais peut durer un peu trop longtemps.', 4.0, '2023-02-05 14:30:00'), -- Catan par Bob
    (3, 2, 3, 'Parfait pour jouer en famille, très accessible.', 4.2, '2023-03-05 11:00:00'), -- Ticket to Ride par Charlie
    (4, 3, 1, 'Le système de draft est très plaisant et rapide.', 4.8, '2023-03-15 09:20:00'), -- 7 Wonders par Alice
    (5, 4, 4, 'Facile à apprendre, se joue rapidement et plaît à tout le monde.', 4.1, '2023-03-25 16:45:00'); -- Carcassonne par Diana

-- Table JeuContributeur (Association N:N entre Jeu et Contributeur)
-- Quelques exemples de liens
INSERT INTO JeuContributeur (id_jeu, id_contributeur)
VALUES
    -- Catan
    (1, 1), -- Klaus Teuber (designer)
    (1, 6), -- Asmodee (éditeur)
    -- Ticket to Ride
    (2, 3), -- Alan R. Moon (designer)
    (2, 2), -- Days of Wonder (éditeur)
    -- 7 Wonders
    (3, 4), -- Antoine Bauza (designer)
    (3, 5), -- Repos Production (éditeur)
    -- Carcassonne
    (4, 6), -- Asmodee (éditeur) -- pas exemple

-- Table JeuMetadata (Association N:N)
-- Montre comment les jeux sont catégorisés
INSERT INTO JeuMetadata (id_jeu, id_metadata)
VALUES
    -- Catan
    (1, 1), -- Stratégie
    (1, 5), -- Négociation (possibilité d'échanger des ressources)
    -- Ticket to Ride
    (2, 2), -- Famille
    (2, 6), -- Transport (trains)
    -- 7 Wonders
    (3, 1), -- Stratégie
    (3, 3), -- Drafting
    -- Carcassonne
    (4, 2), -- Famille
    (4, 4); -- Placement de tuiles

```

```
-- Table UtilisateurBadge (Association N:N entre Utilisateur et Badge)
INSERT INTO UtilisateurBadge (id_utilisateur, id_badge, date_attribution)
VALUES
  (1, 1, '2023-03-01'), -- Alice reçoit le badge "Top Reviewer"
  (2, 2, '2023-02-15'), -- Bob reçoit le badge "Early Adopter"
  (3, 2, '2023-03-21'), -- Charlie reçoit aussi le badge "Early Adopter"
  (1, 3, '2025-01-01'); -- Alice reçoit "Veteran Member" (exemple fictif)
```

- Documentation des déclencheurs, procédures stockées et vues
- Vous devez créer au moins trois objets de chaque type (trigger, index, etc.).

Ajout de tables supplémentaire

```
-- Table Inventaire
CREATE TABLE IF NOT EXISTS Inventaire (
  id_exemplaire INT AUTO_INCREMENT PRIMARY KEY,
  id_jeu INT NOT NULL,
  statut ENUM('disponible','loué','réservé') DEFAULT 'disponible',
  FOREIGN KEY (id_jeu) REFERENCES Jeu(id_jeu) ON DELETE CASCADE
) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS Location (
  id_location INT AUTO_INCREMENT PRIMARY KEY,
  id_exemplaire INT NOT NULL,
  id_utilisateur INT NOT NULL,
  date_debut DATE NOT NULL,
  date_fin DATE NULL,
  duree_location INT DEFAULT 0,
  FOREIGN KEY (id_exemplaire) REFERENCES Inventaire(id_exemplaire) ON DELETE CASCADE,
  FOREIGN KEY (id_utilisateur) REFERENCES Utilisateur(id_utilisateur) ON DELETE CASCADE
) ENGINE=InnoDB;
```

Insertion dans ces tables supplémentaire

```
INSERT INTO Inventaire (id_jeu, statut)
VALUES
```

```
-- Deux exemplaires de Catan (id_jeu = 1)
```

```
(1, 'disponible'),
```

```
(1, 'disponible'),
```

```
-- Deux exemplaires de Ticket to Ride (id_jeu = 2)
```

```
(2, 'disponible'),
```

```
(2, 'loué'),
```

```
-- Un exemplaire de 7 Wonders (id_jeu = 3)
```

```
(3, 'disponible'),
```

```
-- Un exemplaire de Carcassonne (id_jeu = 4)
```

```
(4, 'disponible');
```

```
/*
Note:
- 'date_fin' = NULL indique que le jeu n'est pas encore retourné.
- 'duree_location' peut être mise à 0 ou calculée via un trigger.
- Les 'id_exemplaire' ci-dessous doivent correspondre aux valeurs AUTO_INCREMENT
  réellement générées dans Inventaire.
*/

-- Exemple 1: Un exemplaire déjà retourné (id_exemplaire=4)
--      loué par l'utilisateur #2 (Bob) du 2023-01-10 au 2023-01-15.
INSERT INTO Location (id_exemplaire, id_utilisateur, date_debut, date_fin, duree_location)
VALUES
    (4, 2, '2023-01-10', '2023-01-15', 5);

-- Exemple 2: Un exemplaire en cours de location (id_exemplaire=2)
--      loué par l'utilisateur #3 (Charlie) depuis le 2023-03-01, pas encore rendu.
INSERT INTO Location (id_exemplaire, id_utilisateur, date_debut, date_fin, duree_location)
VALUES
    (2, 3, '2023-03-01', NULL, 0);
```

Creation des VUES

```
USE BoardTrust;
CREATE OR REPLACE VIEW vue_jeux_disponibles AS
SELECT
    i.id_exemplaire,
    j.nom AS nom_jeu,
    j.description,
    i.statut
FROM Inventaire i
JOIN Jeu j ON i.id_jeu = j.id_jeu
WHERE i.statut = 'disponible';
```

```
CREATE OR REPLACE VIEW vue_historique_locations AS
SELECT
    loc.id_location,
    loc.date_debut,
    loc.date_fin,
    loc.duree_location,
    u.pseudo AS loueur,
    j.nom AS nom_jeu
FROM Location loc
JOIN Utilisateur u ON loc.id_utilisateur = u.id_utilisateur
JOIN Inventaire i ON loc.id_exemplaire = i.id_exemplaire
JOIN Jeu j ON i.id_jeu = j.id_jeu;
```

```

CREATE OR REPLACE VIEW vue_top_utilisateurs AS
SELECT
    u.id_utilisateur,
    u.pseudo
    -- Supprimer nombre_emprunts si vous ne l'avez pas
FROM Utilisateur u
ORDER BY u.id_utilisateur DESC; -- Ou un autre critère

-- Afficher la liste de toutes les vues (et tables)
SHOW FULL TABLES WHERE Table_type = 'VIEW';

```

Affichage du contenu

	Tables_in_boardtrust	Table_type
►	vue_historique_locations	VIEW
	vue_jeux_disponibles	VIEW
	vue_top_utilisateurs	VIEW

	id_exemplaire	nom_jeu	description	statut
►	1	Catan	Un jeu de placement et de commerce sur une île...	disponible
	2	Catan	Un jeu de placement et de commerce sur une île...	disponible
	3	Ticket to Ride	Un jeu de construction de lignes de chemin de f...	disponible
	5	7 Wonders	Un jeu de draft de cartes pour construire les plu...	disponible
	6	Carcassonne	Un jeu de placement de tuiles pour construire u...	disponible

	id_location	date_debut	date_fin	duree_location	loueur	nom_jeu
►	1	2023-01-10	2023-01-15	5	Bob	Ticket to Ride
	2	2023-03-01	NULL	0	Charlie	Catan

	id_utilisateur	pseudo
►	4	Diana
	3	Charlie
	2	Bob
	1	Alice

Creation des indexes

```
USE BoardTrust;

-- 1) Index sur le pseudo de l'utilisateur pour les recherches rapides
CREATE INDEX idx_utilisateur_pseudo
  ON Utilisateur (pseudo);

-- 2) Index sur l'id_jeu dans la table Commentaire
--   utile si on fait souvent des "SELECT ... FROM Commentaire WHERE id_jeu = ..."
CREATE INDEX idx_commentaire_id_jeu
  ON Commentaire (id_jeu);

-- 3) Index composite (multiple colonnes) dans la table Location
--   pour optimiser la recherche des locations d'un utilisateur par date
CREATE INDEX idx_location_utilisateur_date
  ON Location (id_utilisateur, date_debut);
```

Creation des triggers

```
CREATE TRIGGER trg_location_ai_inventaire
AFTER INSERT
ON Location
FOR EACH ROW
) BEGIN
    UPDATE Inventaire
    SET statut = 'loué'
    WHERE id_exemplaire = NEW.id_exemplaire;
- END $$

DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE TRIGGER trg_location_ai_utilisateur
AFTER INSERT
ON Location
FOR EACH ROW
BEGIN
    UPDATE Utilisateur
    SET nombre_emprunts = nombre_emprunts + 1
    WHERE id_utilisateur = NEW.id_utilisateur;
END $$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE TRIGGER trg_location_au_retour
AFTER UPDATE
ON Location
FOR EACH ROW
) BEGIN
    -- Si la date_fin vient d'être renseignée, on considère que l'utilisateur a rendu le jeu
) IF NEW.date_fin IS NOT NULL AND OLD.date_fin IS NULL THEN
    -- 1) On remet l'exemplaire à "disponible"
    UPDATE Inventaire
    SET statut = 'disponible'
    WHERE id_exemplaire = NEW.id_exemplaire;

    -- 2) On calcule la duree_location = date_fin - date_debut
    UPDATE Location
    SET duree_location = DATEDIFF(NEW.date_fin, NEW.date_debut)
    WHERE id_location = NEW.id_location;
    END IF;
END $$
```

```
DELIMITER ;
```

Creation des procédures :

```
DELIMITER $$
```

```
CREATE PROCEDURE sp_louer_jeu (  
    IN p_id_exemplaire INT,  
    IN p_id_utilisateur INT  
)  
BEGIN  
    INSERT INTO Location (id_exemplaire, id_utilisateur, date_debut, date_fin)  
    VALUES (p_id_exemplaire, p_id_utilisateur, CURRENT_DATE(), NULL);  
  
    -- Les triggers s'occuperont de mettre l'exemplaire à "loué"  
    -- et d'incrémenter le compteur de l'utilisateur  
END $$
```

```
DELIMITER ;
```

```
CALL sp_louer_jeu(1, 2);
```

```
DELIMITER $$
```

```

CREATE PROCEDURE sp_retourner_jeu (
    IN p_id_location INT
)
BEGIN
    UPDATE Location
    SET date_fin = CURRENT_DATE()
    WHERE id_location = p_id_location;

    -- Le trigger AFTER UPDATE calcule la durée + repasse l'exemplaire en "disponible"
END $$

DELIMITER ;
CALL sp_retourner_jeu(5);

DELIMITER $$

CREATE FUNCTION fn_get_average_rating (
    p_id_jeu INT
)
RETURNS FLOAT
DETERMINISTIC
BEGIN
    DECLARE avg_note FLOAT;

    SELECT AVG(note) INTO avg_note
    FROM Commentaire
    WHERE id_jeu = p_id_jeu;

    RETURN IFNULL(avg_note, 0); -- renvoie 0 si aucune note
END $$

DELIMITER ;
SELECT fn_get_average_rating(1) AS moyenne_catan;

```

Affichage des procédures:

Procedure	sql_mode	Create Procedure	character_set_client	collation_connection	Database Collation
▶ sp_louer_jeu	ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE...	CREATE DEFINER='root'@'localhost' PROCE...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci

Procedure	sql_mode	Create Procedure	character_set_client	collation_connection	Database Collation
▶ sp_retourner_jeu	ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE...	CREATE DEFINER='root'@'localhost' PROCE...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci

Function	sql_mode	Create Function	character_set_client	collation_connection	Database Collation
▶ fn_get_average_rating	ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE...	CREATE DEFINER='root'@'localhost' FUNCTI...	utf8mb4	utf8mb4_0900_ai_ci	utf8mb4_0900_ai_ci

Procédure / Fonction : Le nom de la procédure (ex. `sp_louer_jeu`) ou de la fonction (`fn_get_average_rating`).

sql_mode : Les modes SQL activés (ex. `ONLY_FULL_GROUP_BY`, `STRICT_TRANS_TABLES`). Ils influencent la manière dont MySQL traite certaines requêtes et validations.

Create Procedure / Create Function : La définition exacte de l'objet, incluant le `DEFINER` et les instructions SQL.

character_set_client n : Les paramètres d'encodage et de collation utilisés lors de la création de la procédure/fonction. En MySQL 8, `utf8mb4` (et `utf8mb4_0900_ai_ci`) est le jeu de caractères par défaut, gérant l'intégralité d'Unicode.

Database Collation : La collation par défaut de la base de données (ici, également `utf8mb4_0900_ai_ci`).