**Rapport du Projet de**

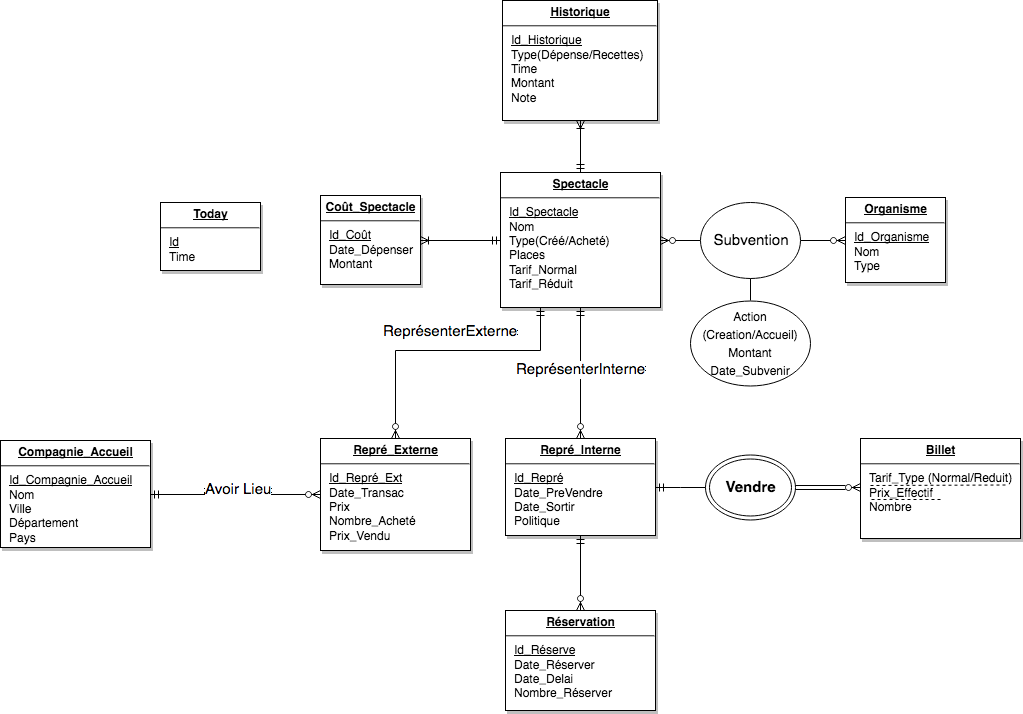
**« Base de Données Avancée (PostgreSQL) »**

Université Paris Diderot

21502157 FAN Yi-Zhe | 21505130 HSIEH Yung-Kun

CODE : [*git@moule.informatique.univ-paris-diderot.fr:Yizhe/project-bda-m1-2017.git*](mailto:git@moule.informatique.univ-paris-diderot.fr:Yizhe/project-bda-m1-2017.git)

**Les Diagrammes de Modèle Entité-Association**



**Les Schémas et Contraints**

**Spectacle ( IdSpectacle, Nom, Places, Type, TarifNormal, TarifRéduit )**

Check constraints:

  "spectacle\_places\_check" CHECK (places >= 0)

  "spectacle\_tarif\_normal\_check" CHECK (tarif\_normal >= 0::numeric)

  "spectacle\_tarif\_reduit\_check" CHECK (tarif\_reduit >= 0::numeric)

  "spectacle\_type\_check" CHECK (type = ANY (ARRAY[0, 1]))

**Cout\_Spectacle ( Id\_Coût, #IdSpectacle, Date\_Dépenser, Montant )**

Check constraints:

  "cout\_spectacle\_montant\_check" CHECK (montant > 0::numeric)

Foreign-key constraints:

  "cout\_spectacle\_id\_spectacle\_fkey" FOREIGN KEY (id\_spectacle) REFERENCES spectacle(id\_spectacle)

Triggers:

1. cout\_achete\_checker BEFORE INSERT OR UPDATE ON cout\_spectacle FOR EACH ROW EXECUTE PROCEDURE check\_cout\_achete()
2. historique\_cout\_modifier AFTER INSERT OR DELETE OR UPDATE ON cout\_spectacle FOR EACH ROW EXECUTE PROCEDURE modify\_cout\_historique()

**Organismes ( Id\_Organisme, Nom, Type )**

**Subvention ( #Id\_Spectacle, #Id\_Organisme, Action, Montant, Date\_Subvenir)**

Check constraints:

  "subvention\_action\_check" CHECK (action::text = ANY (ARRAY['creation'::character varying, 'accueil'::character varying]::text[]))

  "subvention\_montant\_check" CHECK (montant > 0::numeric)

Foreign-key constraints:

  "subvention\_id\_organisme\_fkey" FOREIGN KEY (id\_organisme) REFERENCES organisme(id\_organisme)

  "subvention\_id\_spectacle\_fkey" FOREIGN KEY (id\_spectacle) REFERENCES spectacle(id\_spectacle)

Triggers:

1. historique\_subvenir\_modifier AFTER INSERT OR DELETE OR UPDATE ON subvention FOR EACH ROW EXECUTE PROCEDURE modify\_subvenir\_historique()
2. subvenir\_action\_checker BEFORE INSERT OR UPDATE ON subvention FOR EACH ROW EXECUTE PROCEDURE check\_subvenir\_action()

**Repré\_Interne ( Id\_Repré, #IdSpectacle, Date\_PreVendre, Date\_Sortir, Politique )**

Check constraints:

  "repre\_interne\_check" CHECK (date\_sortir > date\_prevendre)

  "repre\_interne\_politique\_check" CHECK (politique >= 0)

Foreign-key constraints:

  "repre\_interne\_id\_spectacle\_fkey" FOREIGN KEY (id\_spectacle) REFERENCES spectacle(id\_spectacle)

**Billet ( #Id\_Repré, Tarif\_Type, Prix\_Effectif, Nombre )**

Check constraints:

  "billet\_nombre\_check" CHECK (nombre >= 0)

  "billet\_tarif\_type\_check" CHECK (tarif\_type = ANY (ARRAY[0, 1]))

Foreign-key constraints:

  "billet\_id\_repre\_fkey" FOREIGN KEY (id\_repre) REFERENCES repre\_interne(id\_repre)

Triggers:

1. on\_billet\_changer AFTER INSERT OR DELETE OR UPDATE ON billet FOR EACH ROW EXECUTE PROCEDURE on\_billet\_change()
2. on\_billet\_preprocessor BEFORE INSERT OR UPDATE ON billet FOR EACH ROW EXECUTE PROCEDURE on\_billet\_preprocess()

**Historiques ( Id\_Historique, #Id\_Spectacle, Type, Time, Montant, Note )**

Check constraints:

  "historique\_type\_check" CHECK (type = ANY (ARRAY[0, 1]))

Foreign-key constraints:

  "historique\_id\_spectacle\_fkey" FOREIGN KEY (id\_spectacle) REFERENCES spectacle(id\_spectacle)

**Repré\_Externe ( Id\_Repré\_Ext, #Id\_Spectacle, #Id\_Compagnie\_Accueil, Date\_Transac, Prix, Nombre\_Acheté, Prix\_Vendu )**

Check constraints:

  "repre\_externe\_nombre\_achete\_check" CHECK (nombre\_achete > 0)

  "repre\_externe\_prix\_check" CHECK (prix > 0::numeric)

  "repre\_externe\_prix\_check1" CHECK (prix > 0::numeric)

Foreign-key constraints:

  "repre\_externe\_id\_compagnie\_accueil\_fkey" FOREIGN KEY (id\_compagnie\_accueil) REFERENCES compagnie\_accueil(id\_compagnie\_accueil)

  "repre\_externe\_id\_spectacle\_fkey" FOREIGN KEY (id\_spectacle) REFERENCES spectacle(id\_spectacle)

Triggers:

1. historique\_repre\_externe\_modifier AFTER INSERT OR DELETE OR UPDATE ON repre\_externe FOR EACH ROW EXECUTE PROCEDURE modify\_repre\_externe\_historique()
2. type\_checker\_prix\_modifier BEFORE INSERT OR UPDATE ON repre\_externe FOR EACH ROW EXECUTE PROCEDURE check\_type\_modify\_prix()

**Compagnie\_Accueil ( Id\_Compagnie\_Accueil, Nom, Ville, Département, Pays )**

**Today ( Id, time )**

**Les Fonctions Commerciales**

**FUNCTION reserver (idRepre INTEGER, nombre INTEGER) RETURNS INTEGER**

Cette fonction sert à faire des réservations de *nombre* de places pour une représentation *idRepre*. Elle renvoie -1 si la réservation échoue, sinon elle renvoie un numéro pour la réservation.

**FUNCTION calc\_nombre\_place\_dans\_billet (idRepre INTEGER) RETURNS INTEGER**

Cette fonction sert à calculer le nombre des places vendus (places réservés exclues).

**FUNCTION calc\_nombre\_place\_dans\_reserv (idRepre INTEGER) RETURNS INTEGER**

Cette fonction sert à calculer le nombre des places réservés (places vendus exclues) pour la représentation *idRepre*.

**FUNCTION get\_seat\_sold\_percentage (idRepre INTEGER) RETURNS numeric (2,2)**

Cette fonction sert à calculer le pourcentage de places vendus (places réservés exclues) pour la représentation *idRepre*.

**FUNCTION get\_seat\_full\_percentage (idRepre INTEGER) RETURNS numeric (2,2)**

Cette fonction sert à calculer le pourcentage de places totales (places réservées et places vendues) pour la représentation *idRepre*.

**FUNCTION get\_current\_ticket\_price (idRepre INTEGER, tarifType INTEGER) RETURNS numeric (8,2)**

Cette fonction sert à calculer le prix actuel du billet selon le politique et le type de tarif (Réduit = 1, Normal = 0) du spectacle pour la représentation *idRepre.*

**FUNCTION pay\_reservation (myIdReserve INTEGER, tarifNormal INTEGER, tarifReduit INTEGER) RETURNS numeric(8,2)**

Cette fonction sert à payer une réservation de numéro *myIdReserve*. Vous devez indiquer le nombre de billets tarif normal *tarifNormal* et tarif réduit *tarifReduit*.

**FUNCTION account (idSpectacle integer, searchType integer) RETURNS numeric(8,2)**

Cette fonction sert à calculer la recette (*searchType*=1) ou la dépense (*searchType*=0) ou le gain (*searchType*=2) du spectacle *idSpectacle* (somme de plusieurs représentations).

**FUNCTION account\_by\_repre (idRepre integer, tarifType integer) RETURNS numeric(8,2)**

Cette fonction sert à calculer la somme d’argent totale correspondant aux billets de types différents vendu (Réduit = 1, Normal = 0 pour *tarifType*) pour de la représentation *idRepre*.

**FUNCTION account\_by\_date(idSpectacle integer, dateFrom date, dateTo date) RETURNS numeric(8,2)**

Cette fonction sert à calculer le gain du spectacle *idSpectacle* (somme de plusieurs représentations) entre la date *dateFrom* et la date *dateTo*.

**FUNCTION balance\_sheet(searchType integer, dateFrom date, dateTo date) RETURNS setof historique**

Cette fonction fournit une liste des recettes (*searchType*=1) ou des dépenses (*searchType*=0) ou les deux (*searchType*=2) des représentations entre la date *dateFrom* et la date *dateTo*.