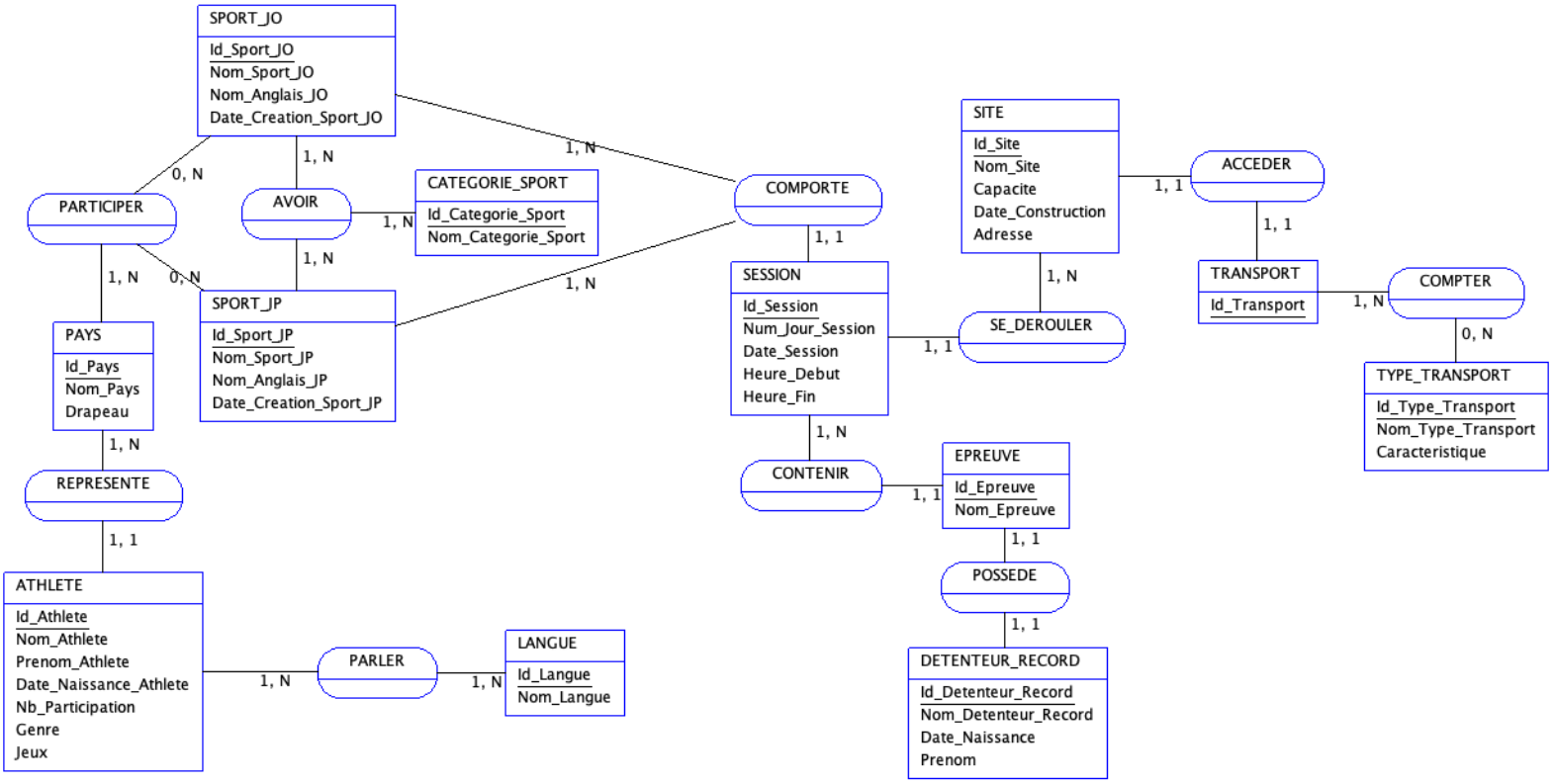
Projet Base de Données

Présenté par : Rakotoarison Hasina Sitraka, Danielevic Naoˆmie, Cointe Julien, Guillaume Fasquelle

---

Sommaire :

Modèle entité association (EA) :



Modèle relationnel :

Image :

```
EPREUVE (Id_Epreuve, Nom_Epreuve, #Id_Session, #detenteur_record_id_detenteur_record)
SITE (Id_Site, Nom_Site, Capacite_Site, Date_Construction, Adresse_Site, #transport_id_transport)
SESSION (Id_Session, Num_Jour_Session, Date_Session, Heure_Debut, Heure_Fin, #Id_Site)
TRANSPORT (Id_Transport, #site_id_site)
SPORT_JP (Id_Sport_JP, Nom_Sport_JP, Nom_Anglais_JP, Date_Creation_Sport_JP)
CATEGORIE_SPORT (Id_Categorie_Sport, Nom_Categorie_Sport)
TYPE_TRANSPORT (Id_Type_Transport, Nom_Type_Transport, Caracteristique)
PAYS (Id_Pays, Nom_Pays, Drapeau_PAYS)
DETENTEUR_RECORD (Id_Detenteur_Record, Nom_Detenteur_Record, Date_Naissance, Prenom_DetenteurRecord, #epreuve_id_epreuve)
SPORT_JO (Id_Sport_JO_SPORT_JO, Nom_Sport_JO_SPORT_JO, Nom_Anglais_JO_SPORT_JO, Date_Creation_Sport_JO_SPORT_JO)
ATHLETE (Id_Athlete, Nom_Athlete, Prenom_Athlete, Date_Naissance_Athlete, Nb_Participation_ATHLETE, Genre_ATHLETE, Jeux_ATHLETE, #Id_Pays)
LANGUE (Id_Langue_LANGUE, Nom_Langue_LANGUE)
COMPORTE (Id_Sport_JP, Id_Session, Id_Sport_JO_SPORT_JO)
AVOIR (Id_Sport_JP, Id_Categorie_Sport, Id_Sport_JO_SPORT_JO)
COMPTER (Id_Transport, Id_Type_Transport)
PARTICIPER (Id_Pays, Id_Sport_JP, Id_Sport_JO_SPORT_JO)
PARLER (Id_Langue_LANGUE, Id_Athlete)
```

Texte :

EPREUVE (Id_Epreuve, Nom_Epreuve, #Id_Session, #detenteur_record_id_detenteur_record)
SITE (Id_Site, Nom_Site, Capacite_Site, Date_Construction, Adresse_Site, #transport_id_transport)
SESSION (Id_Session, Num_Jour_Session, Date_Session, Heure_Debut, Heure_Fin, #Id_Site)
TRANSPORT (Id_Transport, #site_id_site)
SPORT_JP (Id_Sport_JP, Nom_Sport_JP, Nom_Anglais_JP, Date_Creation_Sport_JP)
CATEGORIE_SPORT (Id_Categorie_Sport, Nom_Categorie_Sport)
TYPE_TRANSPORT (Id_Type_Transport, Nom_Type_Transport, Caracteristique)
PAYS (Id_Pays, Nom_Pays, Drapeau_PAYS)
DETENTEUR_RECORD (Id_Detenteur_Record, Nom_Detenteur_Record, Date_Naissance, Prenom_DetenteurRecord, #epreuve_id_epreuve)
SPORT_JO (Id_Sport_JO_SPORT_JO, Nom_Sport_JO_SPORT_JO, Nom_Anglais_JO_SPORT_JO, Date_Creation_Sport_JO_SPORT_JO)
ATHLETE (Id_Athlete, Nom_Athlete, Prenom_Athlete, Date_Naissance_Athlete, Nb_Participation_ATHLETE, Genre_ATHLETE, Jeux_ATHLETE, #Id_Pays)
LANGUE (Id_Langue_LANGUE, Nom_Langue_LANGUE)
COMPORTE (Id_Sport_JP, Id_Session, Id_Sport_JO_SPORT_JO)
AVOIR (Id_Sport_JP, Id_Categorie_Sport, Id_Sport_JO_SPORT_JO)
COMPTER (Id_Transport, Id_Type_Transport)
PARTICIPER (Id_Pays, Id_Sport_JP, Id_Sport_JO_SPORT_JO)
PARLER (Id_Langue_LANGUE, Id_Athlete)

Dictionnaire :

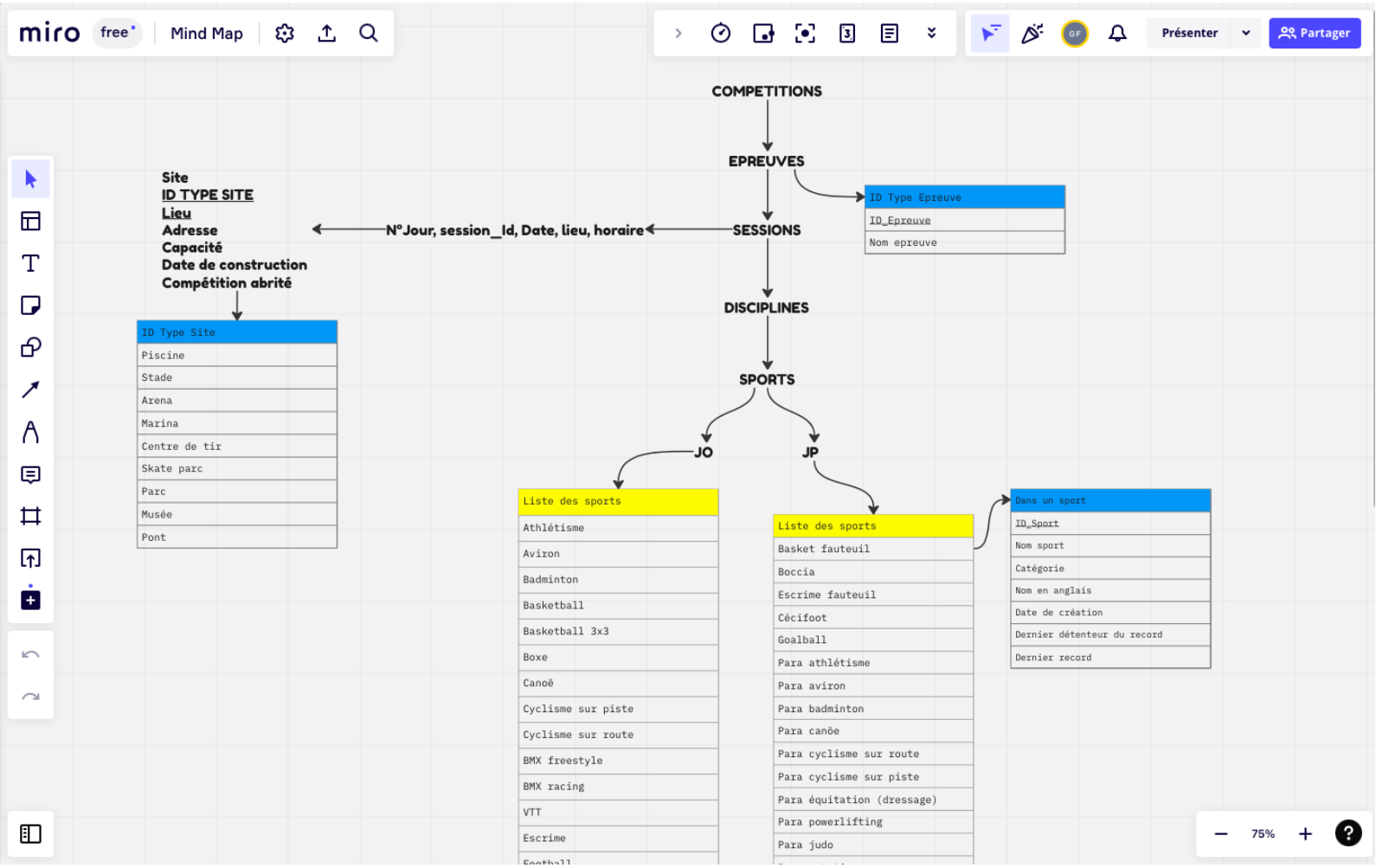
NOM	ID	TYPE	ENTITÉ
Id_Athlete	Id_Athlete	INT_AUTO_INCREMENT	ATHLETE
Nom_Athlete	Nom_Athlete	VARCHAR	ATHLETE
Prenom_Athlete	Prenom_Athlete	VARCHAR	ATHLETE
Date_Naissance_Athlete	Date_Naissance_Athlete	DATE	ATHLETE
Nb_Participation_Athlete	Nb_Participation_Athlete	INT_AUTO_INCREMENT	ATHLETE
Genre	Genre	CHAR	ATHLETE
Jeux	Jeux	VARCHAR	ATHLETE
Id_Categorie_Sport	Id_Categorie_Sport	INT_AUTO_INCREMENT	CATEGORIE_SPORT
Nom_Categorie_Sport	Nom_Categorie_Sport	VARCHAR	CATEGORIE_SPORT
Id_Detenteur_Record	Id_Detenteur_Record	INT_AUTO_INCREMENT	DETENTEUR_RECORD
Date_Naissance	Date_Naissance	DATE	DETENTEUR_RECORD
Prenom	Prenom	VARCHAR	DETENTEUR_RECORD
Nom_Epreuve	Nom_Epreuve	VARCHAR	EPREUVE
Id_Epreuve	Id_Epreuve	INT_AUTO_INCREMENT	EPREUVE
Id_Langue	Id_Langue	INT_AUTO_INCREMENT	LANGUE
Nom_Langue	Nom_Langue	VARCHAR	LANGUE
Id_Pays	Id_Pays	INT_AUTO_INCREMENT	PAYS
Nom_Pays	Nom_Pays	VARCHAR	PAYS
Drapeau	Drapeau	VARCHAR	PAYS
Id_Session	Id_Session	INT_AUTO_INCREMENT	SESSION
Num_Jour_Session	Num_Jour_Session	INT	SESSION
Heure_Debut	Heure_Debut	TIME	SESSION
Heure_Fin	Heure_Fin	TIME	SESSION
Date_Session	Date_Session	DATE	SESSION
Id_Site	Id_Site	INT_AUTO_INCREMENT	SITE
Nom_Site	Nom_Site	VARCHAR	SITE

Capacite	Capacite	INT	SITE
Date_Construction	Date_Construction	DATE	SITE
Adresse	Adresse_Site	VARCHAR	SITE
Id_Sport_JO	Id_Sport_JO	INT_AUTO_INCREMENT	SPORT_JO
Nom_Sport_JO	Nom_Sport_JO	VARCHAR	SPORT_JO
Nom_Anglais_JO	Nom_Anglais_JO	VARCHAR	SPORT_JO
Date_Creation_Sport_JO	Date_Creation_Sport_JO	DATE	SPORT_JO
Id_Sport_JP	Id_Sport_JP	INT_AUTO_INCREMENT	SPORT_JP
Nom_Sport_JP	Nom_Sport_JP	VARCHAR	SPORT_JP
Date_Creation_Sport_JP	Date_Creation_Sport_JP	DATE	SPORT_JP
Nom_Anglais_JP	Nom_Anglais_JP	VARCHAR	SPORT_JP
Id_Transport	Id_Transport	INT_AUTO_INCREMENT	TRANSPORT
Id_Type_Transport	Id_Type_Transport	INT_AUTO_INCREMENT	TYPE_TRANSPORT
Nom_Type_Transport	Nom_Type_Transport	VARCHAR	TYPE_TRANSPORT
Caracteristique	Caracteristique	VARCHAR	TYPE_TRANSPORT

Outils utilisés :

Miro :

Avant de commencer la réalisation de notre modèle entité association nous avons d'abord eu plusieurs réflexions lors de séances de brainstorming afin que chaque membre du groupe puisse exposer ses idées sur la manière de concevoir et de visualiser le modèle entité association.
Grâce à Miro qui est une plateforme de collaboration en ligne qui se présente sous forme d'un tableau blanc interactif sur lequel on peut créer différents éléments comme des tableaux, des formes géométriques.
Ce qui nous a permis de faire une première ébauche succincte de notre modèle EA.

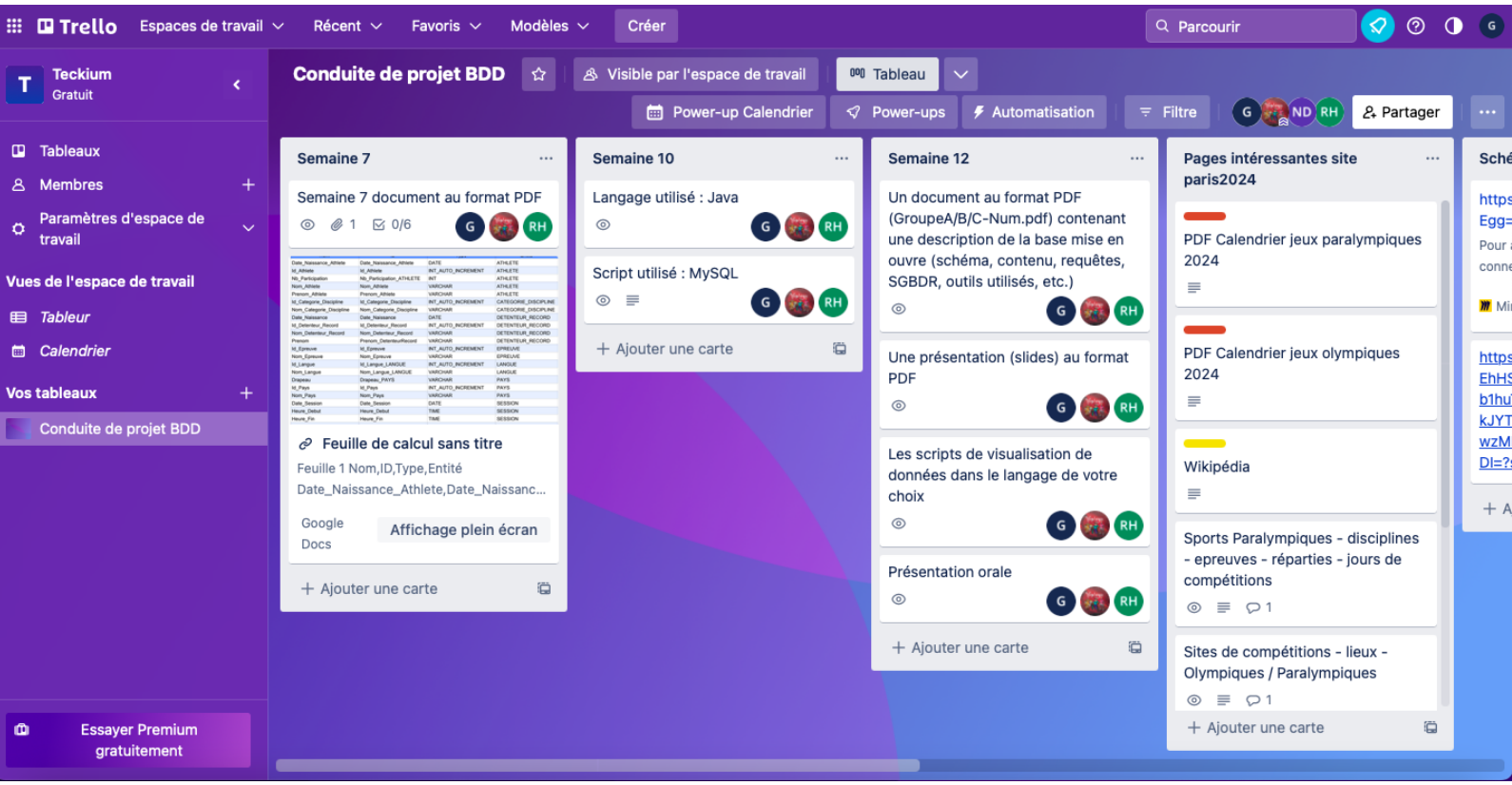


AnalyseSI :

Pour la création du modèle entité association et du modèle relationnel nous avons choisi d'utiliser le logiciel “AnalyseSI” pour sa simplicité d’utilisation.
Il suffit de sélectionner les différents blocs entité association que l’on souhaite créer puis de les relier entre eux et enfin d’ajouter les différents attributs et cardinalité.
Une fois le modèle entité association achevé nous pouvons générer le modèle relationnel associé ainsi que le dictionnaire.

Trello :

Afin de nous organiser nous avons utilisé “Trello” qui est un outil de gestion de projet en ligne qui permet de modéliser les projets sous forme de planches en listant des cartes qui représentent les différentes tâches du projet.
Grâce à cet outil chaque membre du groupe a accès aux différentes ressources (lien des sites web) et peut voir toutes les modifications qui ont été apportées (ajout de nouvelle ressource, tâche à terminer, nouvelle tâche à effectuer).



PostgreSQL :

Afin de créer notre base de données nous avons dans un premier temps consulté sur internet les différents Systèmes de Gestion de Bases de Données Relationnelles (SGBDR) avec pour chacun leurs avantages et inconvénients.
Au départ nous comptions utiliser MySQL cependant il n'était pas possible d'utiliser une séquence chose qui était indispensable pour nous puis avec Oracle le problème étant que nous ne connaissions pas la limitation de taille de la base de données dont on avait le droit avec notre version.
Finalement, notre choix s'est arrêté sur "PostgreSQL" qui correspondait le mieux pour créer notre base de données.

Tomcat :

Pour le site web qui va permettre de visualiser notre base de données nous avons choisi d'utiliser "Tomcat" qui est un serveur web open source qui permet d'exécuter des applications web basées sur Java, facile à configurer et à déployer.

Scrapping :

Génération de fichier .txt :

Pour construire notre base de données nous avons réalisé un algorithme de "scrapping" qui permet de récolter les données nécessaires à la construction de notre base et de générer un fichier .txt.

Génération de fichier .sql :

Nous avons réalisé un deuxième algorithme qui permet à partir du fichier .txt généré avec le premier algorithme de générer un fichier .sql.

Mode de fonctionnement :

L’algorithme de “scrapping” fonctionne de la façon suivante:

Dans un premier temps il faut indiquer l’URL du site web dont on veut prélever les données, il faut également indiquer les balises HTML du site qui contient les informations souhaitées. L'algorithme va alors en sortit créer un fichier .txt qui comprend nos données.

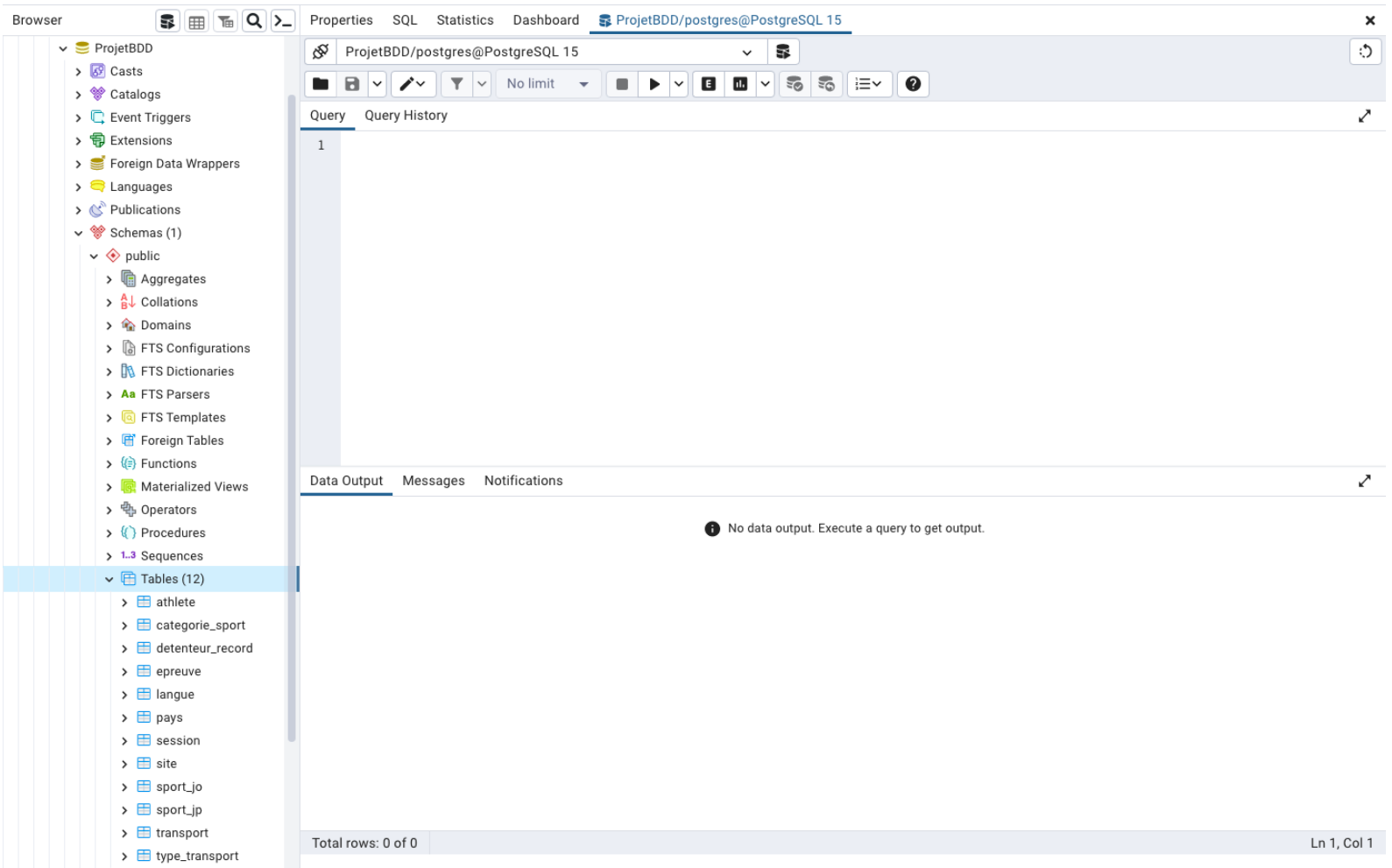
Puis pour finir, le fichier .txt généré va être lu par un le deuxième algorithme qui lui est chargé de créer un fichier .sql qui contiendra les requêtes SQL d’insertion de données dans la table choisie.

Les deux algorithmes ont été écrits en Java.

Base de donnée :

Présentation :

Voici une capture d'écran de notre base de données sur pgAdmin 4.



Sur cette capture d'écran de pgAdmin 4 on peut voir le nom de notre base de données qui s'appelle "ProjetBDD" dans le menu déroulant en cherchant "Tables" on peut voir les différentes tables qui composent notre table de données.
Sur la droite, l'onglet "Query" qui nous permet de saisir nos différentes requêtes SQL.
En bas le résultat de la requête.

En cliquant sur les différentes tables on peut également visualiser les différents attributs qui composent notre table.

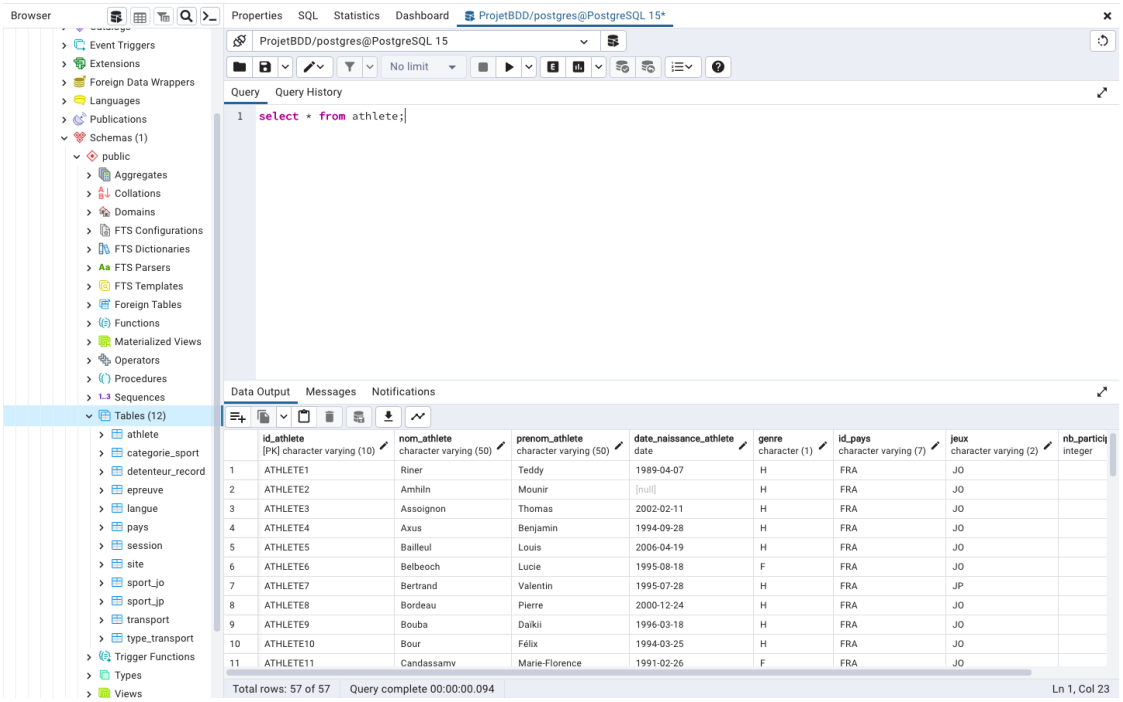
Quelque requêtes SQL :

Code couleurs :

ROUGE : syntaxe langage SQL
BLEU : nom des tables
VERT : nom des attributs

SELECT * FROM athlete;

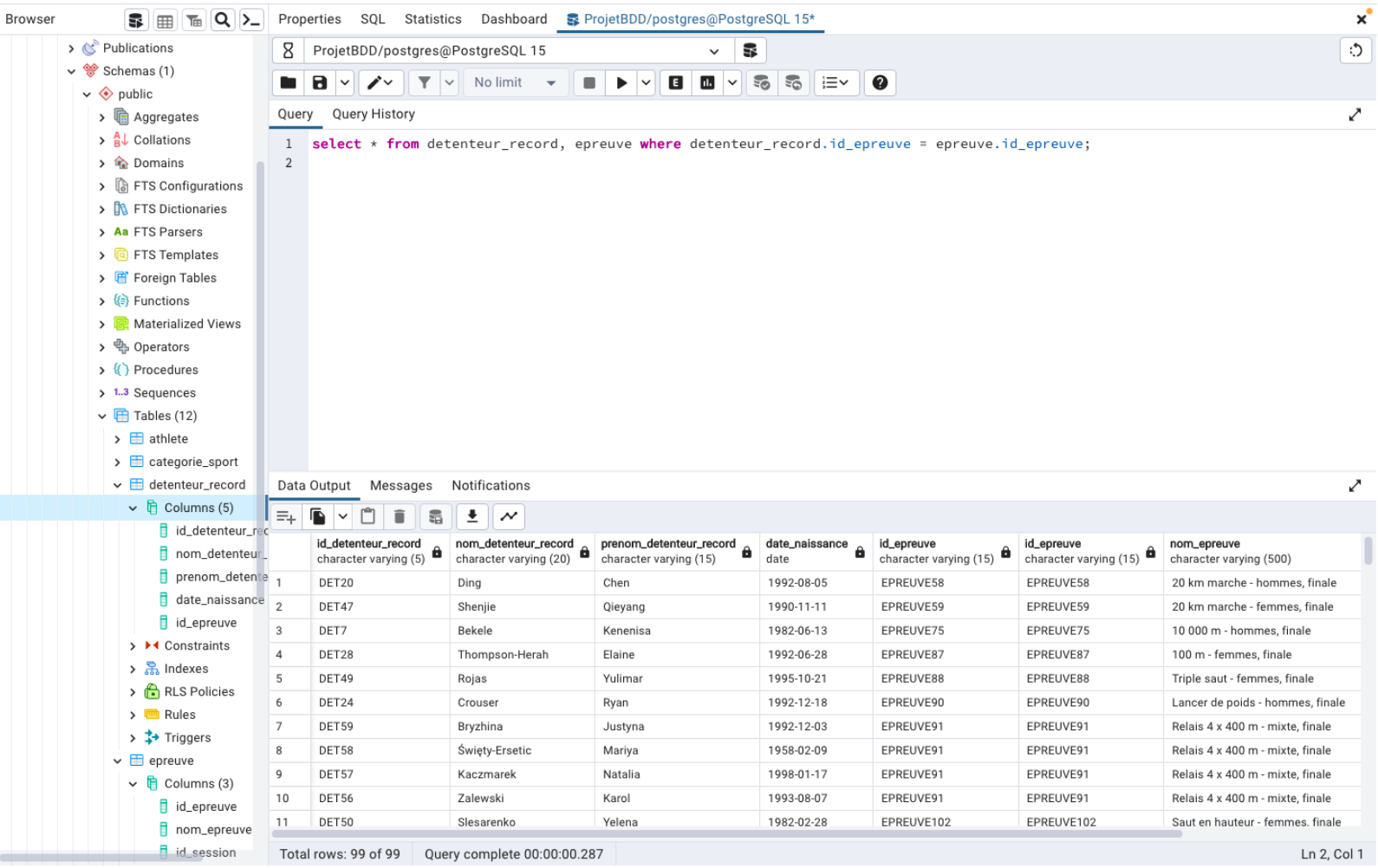
Cette requête permet d'afficher tous les attributs de la table athlete.



	id_athlete [PK] character varying (10)	nom_athlete character varying (50)	prenom_athlete character varying (50)	date_naissance_athlete date	genre character (1)	id_pays character varying (7)	jeux character varying (2)	nb_partici integer
1	ATHLETE1	Riner	Teddy	1989-04-07	H	FRA	JO	
2	ATHLETE2	Amhiln	Mounir	[null]	H	FRA	JO	
3	ATHLETE3	Assoignon	Thomas	2002-02-11	H	FRA	JO	
4	ATHLETE4	Axus	Benjamin	1994-09-28	H	FRA	JO	
5	ATHLETE5	Bailleul	Louis	2006-04-19	H	FRA	JO	
6	ATHLETE6	Belbeoch	Lucie	1995-08-18	F	FRA	JO	
7	ATHLETE7	Bertrand	Valentin	1995-07-28	H	FRA	JP	
8	ATHLETE8	Bordeau	Pierre	2000-12-24	H	FRA	JO	
9	ATHLETE9	Bouba	Daïkii	1996-03-18	H	FRA	JO	
10	ATHLETE10	Bour	Félix	1994-03-25	H	FRA	JO	
11	ATHLETE11	Candassamy	Marie-Florence	1991-02-26	F	FRA	JO	

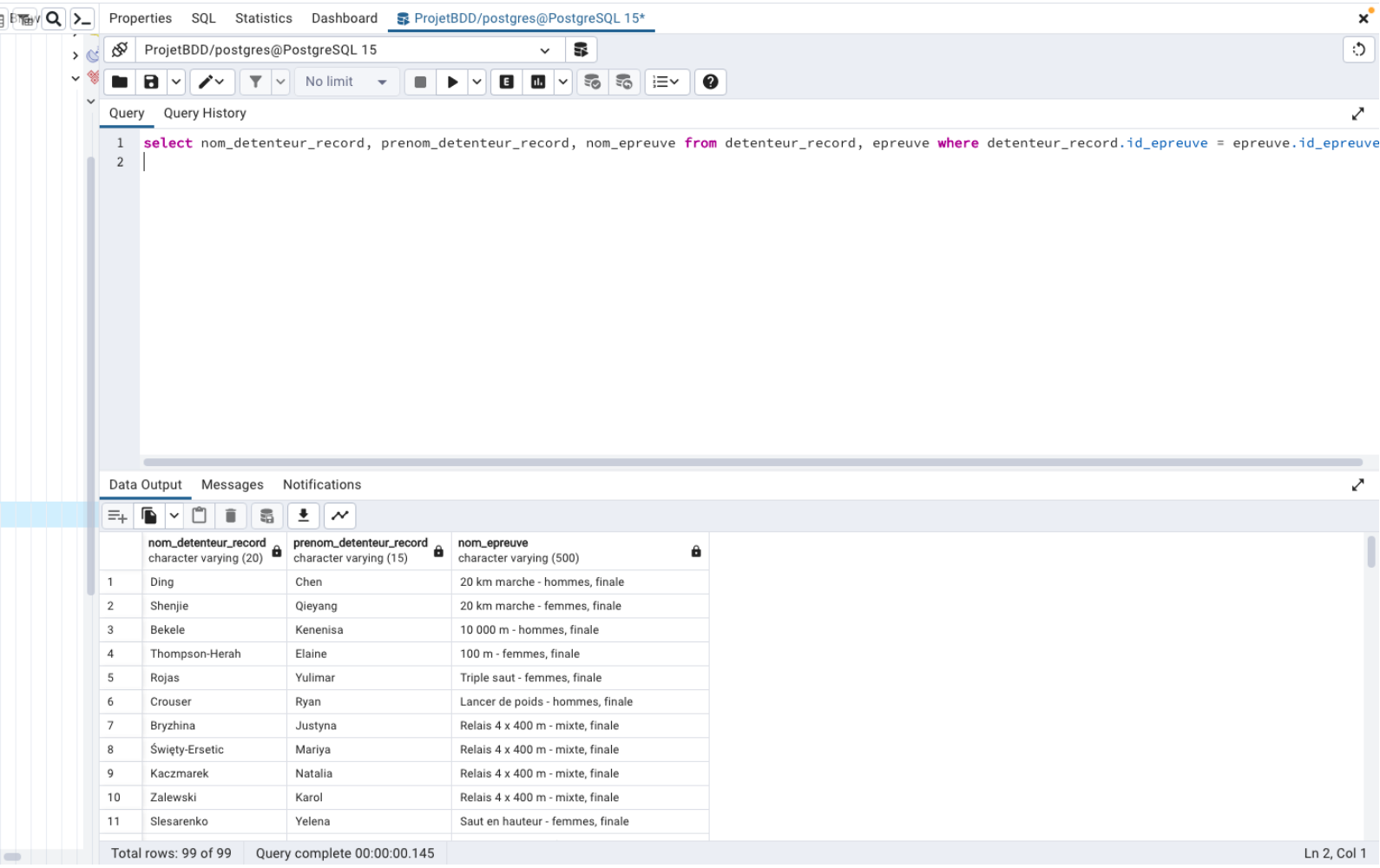
SELECT * FROM detenteur_record, epreuve WHERE detenteur_record.id_epreuve = epreuve.id_epreuve;

Cette requête permet d'afficher tous les attributs des tables detenteur_record et epreuve qui respecte la condition de la jointure.
Ici, on peut visualiser les détenteurs de record avec le nom de l'épreuve associée.



On peut très bien remplacer “*” par nom_detenteur_record, prenom_detenteur_record, nom_epreuve
afin d'obtenir uniquement le nom et prénom du sportif avec le nom de l’épreuve

```
SELECT nom_detenteur_record, prenom_detenteur_record, nom_epreuve FROM detenteur_record, epreuve WHERE detenteur_record.id_epreuve = epreuve.id_epreuve;
```



	nom_detenteur_record character varying (20)	prenom_detenteur_record character varying (15)	nom_epreuve character varying (500)
1	Ding	Chen	20 km marche - hommes, finale
2	Shenjie	Qieyang	20 km marche - femmes, finale
3	Bekele	Kenenisa	10 000 m - hommes, finale
4	Thompson-Herah	Elaine	100 m - femmes, finale
5	Rojas	Yulimar	Triple saut - femmes, finale
6	Crouser	Ryan	Lancer de poids - hommes, finale
7	Bryzhina	Justyna	Relais 4 x 400 m - mixte, finale
8	Święty-Ersetic	Mariya	Relais 4 x 400 m - mixte, finale
9	Kaczmarek	Natalia	Relais 4 x 400 m - mixte, finale
10	Zalewski	Karol	Relais 4 x 400 m - mixte, finale
11	Slesarenko	Yelena	Saut en hauteur - femmes, finale

Site web :

Afin de pouvoir visualiser notre base de données nous avons développé un site web qui permet d'accéder aux différentes tables et attributs qui la composent puis d'effectuer quelques recherches.

De plus Java, JSP (Jakarta Server Pages où JavaServer Pages) et Tomcat sont trois technologies souvent utilisées en développement web car elles fournissent un cadre robuste et évolutif pour le développement d'applications web.

Serveur web :

Pour héberger le site web on utilise Tomcat qui comme évoqué précédemment est un serveur web open source qui permet d'exécuter des applications web basées sur Java ce qui est notre cas.

Script JSP :

Nous avons choisi d'utiliser Jakarta Server Pages (JSP) car il permet de créer des pages web dynamiques en incluant du code Java, des balises HTML et des éléments de présentation CSS.

Le script ci-dessous envoie des requête à la base de donnée pour les récupérer sous une liste puis les affiche dans une table sur une page HTML .

```jsp
<%@ page import="java.sql.*, connexion.*, bdd.*, client.*" %>
<html>
  <head>
    <title>Titre de la page</title>
  </head>
  <body>
    <%
        Fonction f = new Fonction();
        Connection con = null;
        Deconnexion decon = new Deconnexion();
        BDTable bd = new BDTable();
        Site site = new Site();
        Connexion c = null;
        try {
            System.out.println("tafiditra anaty try");
            c = new Connexion();
            c.connecter();
            con = c.getCon();
            java.sql.Statement stmt = con.createStatement();
            System.out.println("tonga");

            String requete = "select * from Site";
            System.out.println(requete);
            BDTable[] lSite = f.findWthReq(requete, site, con);

            out.println("length " + lSite.length);
            Site[] listeSite = new Site[lSite.length];

            out.println("<div>");
                out.println("<table border='1px'>");
                    out.println("<tr><th>ID</th><th>Nom_Pays</th><th>Dreapeau</th></tr>");
                for (int i = 0; i < lSite.length; i++) {
                    listeSite[i] = (Site) lSite[i];
                    out.println("<tr><td>" + listeSite[i].getId_Pays() + "</td><td>" + listeSite[i].getNom_Pays() + "</td><td><img src ="' + listeSite[i].getDreapeau() + '"></td></tr>");
                }
                out.println("</table>");
                out.println("</div>");
        } catch (SQLException e1) {
            out.println(e1.getMessage());
            if (c.getCon() != null) {
                con.rollback();
            }
        } catch (Exception e2) {
            out.println(e2.getMessage());
        } finally {
            if (c.getCon() != null) {
                decon.deconnect(c);
            }
        }
```

Fonctionnalité sur le site web :

Le site web a pour but de permettre la visualisation des entités de la base de données (tables et attributs) et de faire une recherche avancée sur certaines tables.

Quelques requêtes sur le site web :