Est-ce que on peut jouer au tennis demain ?

Charte graphique (site web et appli)

Web + bdd ou capteurs + web

**CAPTEURS :**

Comment mettre en place le broker

Quel matériel à prendre

Quels capteurs à prendre

WAP ? (Protocole de transfert de données sans fil)

Comment faire lien tablette broker ?

Capteur vibration gyroscope...

Quel taille l’écran ?

Doit pouvoir être modifié à distance (télécommande pour changer les points) ?

**MDP BDD :**

*Code phpmyadmin :*

lcr-mm

Azerty.1@

*Code alwaysdata :*

User : menn@lacroixrouge-brest.fr

Mdp user : 7qc2m97qc2m9

*Code ftp :*

Hôte : ftp-lcr-mm.alwaysdata.net

User : lcr-mm

Mdp : Azerty.1@

<https://lcr-mm.alwaysdata.net/api.php>

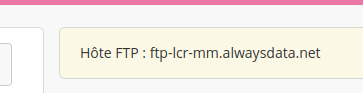
mysql -h mysql-lcr-mm.alwaysdata.net -u lcr-mm -p lcr-mm\_tcb

INSERT INTO Entraineur (nom, prenom) VALUES ('testt','test');

INSERT INTO Statistiques VALUES (1, 1, 1, 0, 1);

**Total match joués :**

SELECT id\_personne, SUM(matchs\_joues) AS total\_matchs\_joues, SUM(matchs\_gagnes) AS total\_matchs\_gagnes, SUM(matchs\_perdus) AS total\_matchs\_perdus FROM matches WHERE id\_personne = 3 GROUP BY id\_personne;



Logiciel pour tester les pen test OWASP ZAP & SQLmap

**Reflexion BDD : looping.exe**

Table résultat/score de personne vers match

Tej table stat ou mettre info dans résultat

Remplir jouer – score / mode de jeu:

Lire table prsn

**- Double authentification !!**

**- Codage mdp (mdp provisoire puis modif par user)**

**- Type BDD (pas que relationnelle)**

**- cacher mdp dans API (1/2)**

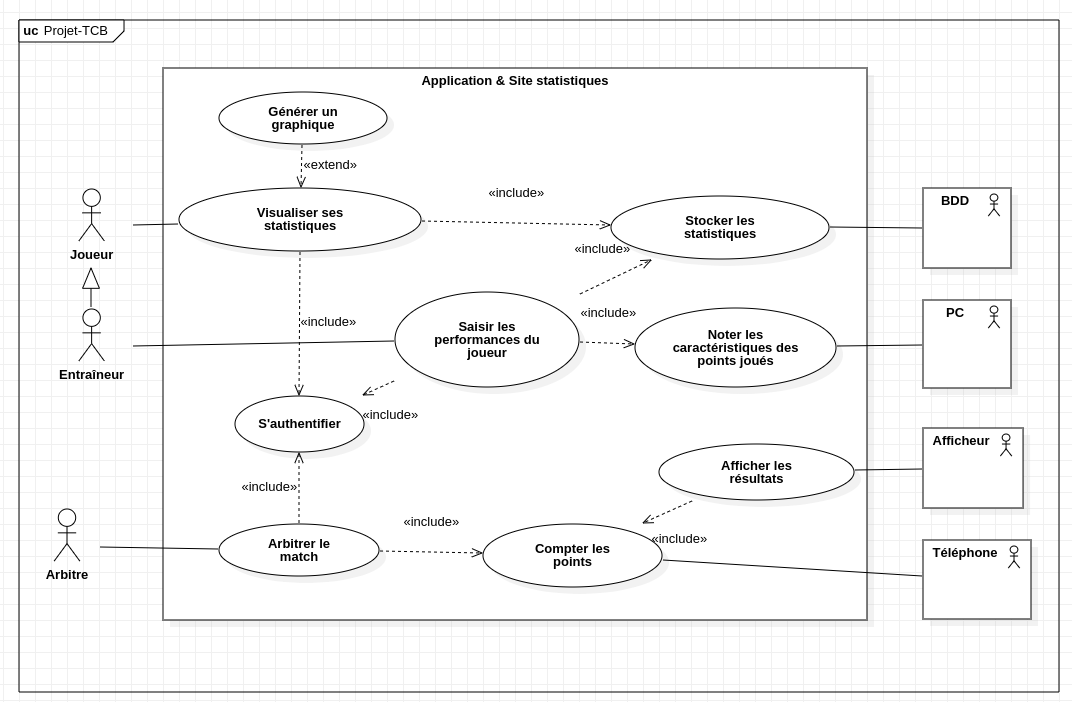
**- regular expression (user format attendu)**

**- Passer en API avec Andy**

https://www.looping-mcd.fr/

**Liste appli pentest :**

Kali / nmap / metasploit / Wireshark / John The Ripper /Hashcat /Hydra / Burp Suite / OWASP ZAP / SQLmap / aircrack-ng



|  |  |  |
| --- | --- | --- |
| ***Nom CU: ...*** | ***Référence : CUx*** | ***Nom*** |
| Pré-condition(s) | *1. Connexion à l’Application*  *2. Connexion au Site Web* | |
| Scénario nominal | *1. Arbitrage du match – Comptage des points*  *2. Complétion de la feuille de match* | |
| Post-condition(s) | *1. Envoie de la feuille de match (données) vers BDD*  *2. Mise à jour de la BDD* | |

**QUESTIONS :**

* Qui gère la table matches / les 2
* Statistiques du joueur unique et stat entre 2 joueurs
* OU stat joueur unique et stat 2 joueurs comparé dans 1 match
* OU stat 1 joueur unique sur 1 match en particulier
* Aller au TCB (feuille de match exacte ou demandé)
* Rajouter champ mdp dans Personne ? Oui
* Serveur web BDD ? oui

**QUESTIONS TCB :**

* Exemple feuille de match
* Point d’accès WI-FI

**MODIF BDD :**

* Ajouter API en acteurs secondaire faire une relation avec sa fonction (ex = API => chiffrement, requetes SQL...)

Test mdp crypté (AES-256-CBC):

Recup mdp crypté

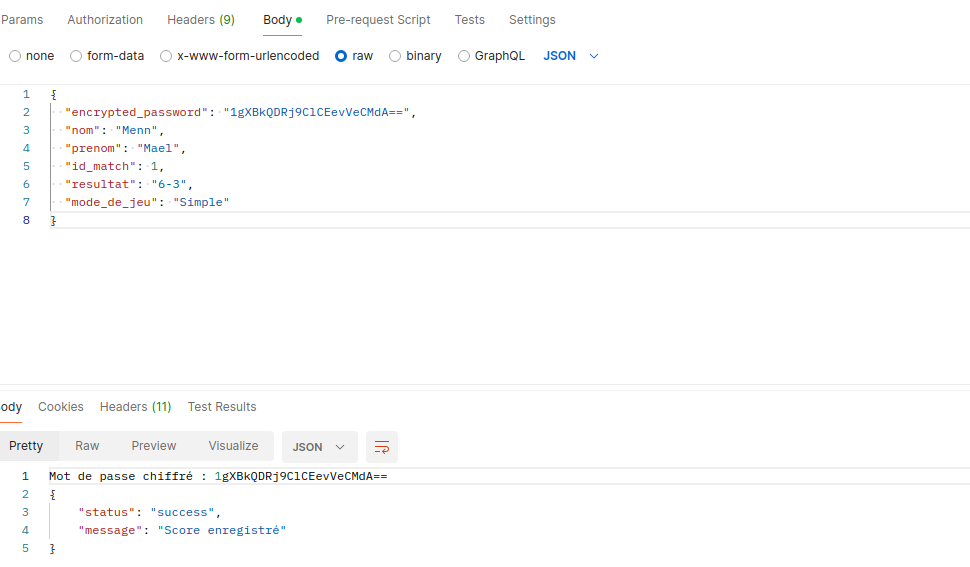
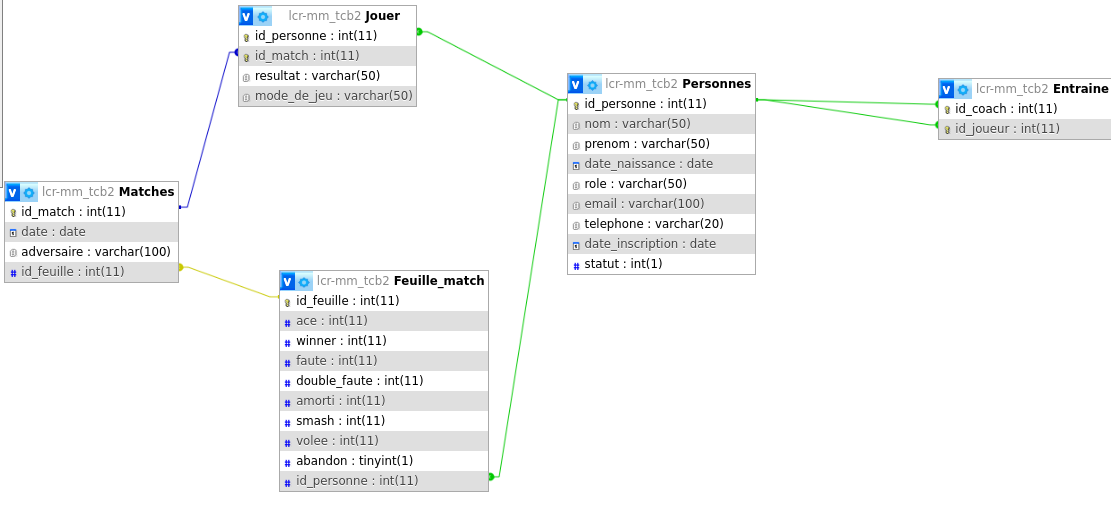
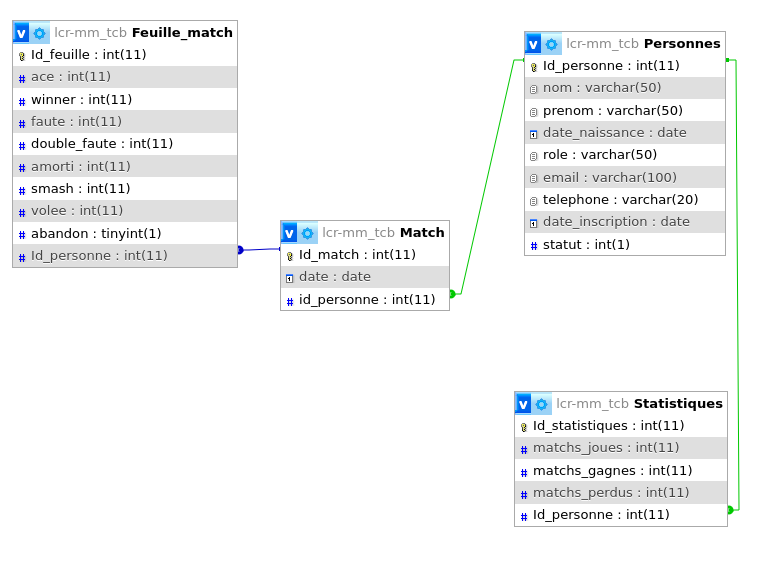
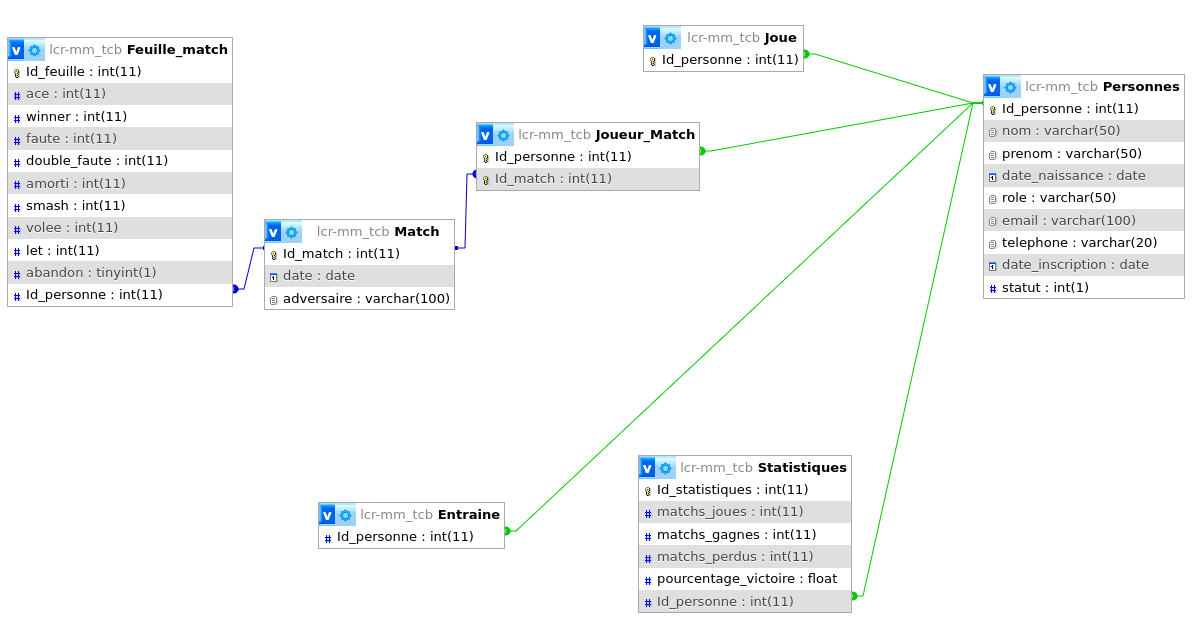
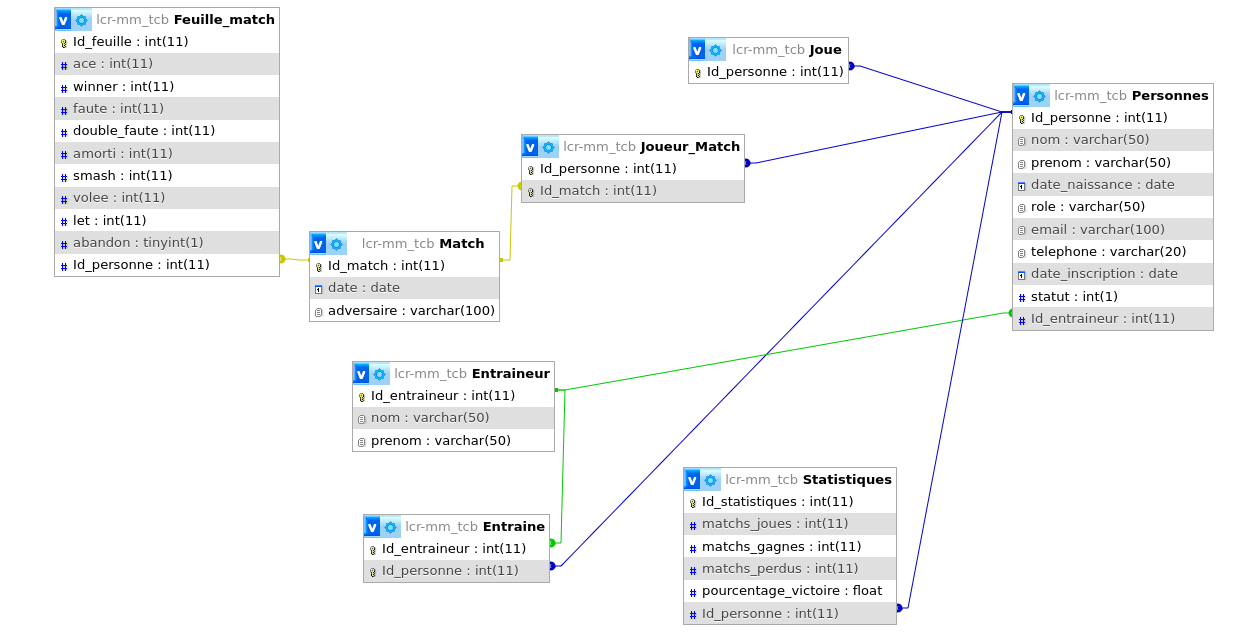
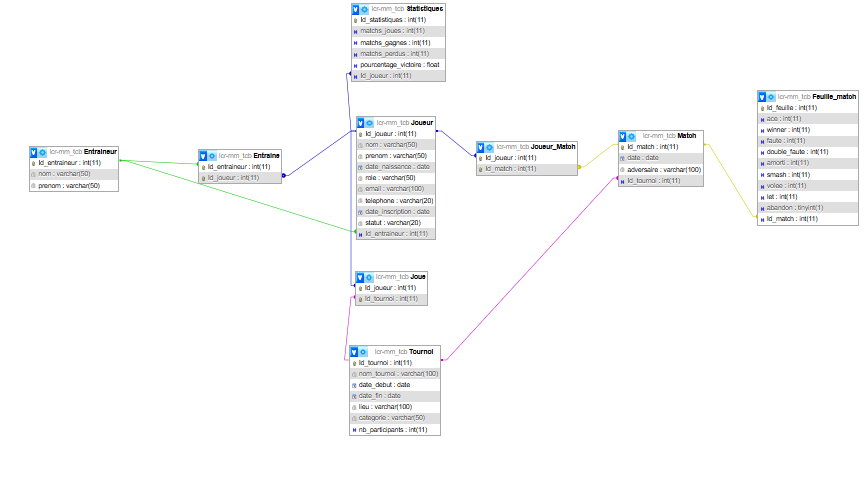
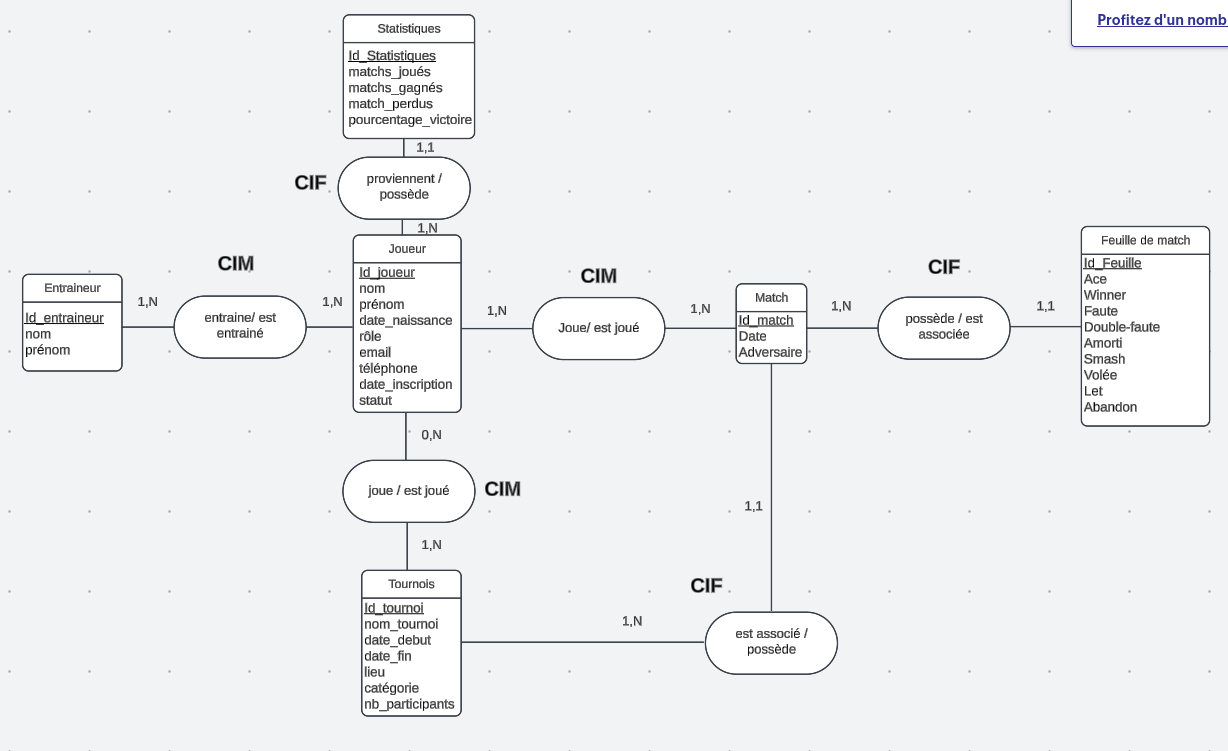
<https://lcr-mm.alwaysdata.net/API/encrypt_password.php>

**Recup id joueur avec mdp crypté**

<https://lcr-mm.alwaysdata.net/API/api.php?encrypted_password=1gXBkQDRj9ClCEevVeCMdA==&nom=Menn&prenom=Mael>

2 eme API

<https://lcr-mm.alwaysdata.net/API_id_prsn/score.php?encrypted_password=1gXBkQDRj9ClCEevVeCMdA==&nom=Menn&prenom=Mael&id_match=1&resultat=2-1&mode_de_jeu=Normal>



## Les différents types de bases de données

On peut classer les bases de données selon différentes approches : en fonction de leur contenu (bibliographique, texte, chiffres, images…) ou selon leur approche organisationnelle, qui est souvent celle retenue en informatique.

* **La BDD hiérarchique**. Souvent présentée sous forme d’arbre avec ses ramifications, il s’agit du tout premier programme qui a permis de structurer l’information de façon hiérarchique. Ici, chaque enregistrement dépend d’un seul enregistrement, et chaque niveau d’enregistrement découle sur un ensemble de catégories plus petites.
* **La BDD réseau**. Dans ce cas, contrairement à la BDD hiérarchique, un objet peut avoir plusieurs objets parents et plusieurs objets enfants, ce qui permet de s’approcher du monde réel plus fidèlement. Des liens multiples sont ainsi créés entre les ensembles, permettant une vitesse et une polyvalence qui ont permis leur adoption massive.
* **La BDD SQL ou relationnelle**. C’est la plus connue et la plus pratiquée actuellement. Reposant sur l’algèbre relationnel, elle a pour fonction de modéliser facilement les systèmes du monde réel, et de créer des bases de données à la fois simples à maintenir et à faire évoluer. Constituées d’un ensemble de tableaux, ces bases de données contiennent des données classées par catégorie. L’API standard pour ces BDD est le Structured Query Language (SQL).
* **La BDD orientée objet**. Cette typologie de base de données est encore en cours d’élaboration. Elle est focalisée sur la base de données des objets en tant que concept de programmation qui va permettre de simplifier la création de logiciels.
* **La BDD orientée texte**. La “flat file database” se présente sous la forme d’un fichier .txt ou .ini, qui est soit un fichier texte, soit un fichier combinant du texte avec un fichier binaire. Chaque ligne ne comporte généralement qu’un seul enregistrement.
* **La BDD distribuée**. Ce type de base de données présente des portions stockées au sein de différents endroits physiques, avec un traitement réparti ou répliqué entre différents points d’un réseau. Elle peut être homogène ou hétérogène : soit tous les emplacements physiques fonctionnent avec le même hardware et tournent sous le même système d’exploitation et les mêmes applications de bases de données, soit ils varient entre différents endroits physiques.
* **La BDD cloud**. Optimisée ou directement créée pour les environnements virtualisés, elle peut être relative à un cloud privé, un cloud public ou un cloud hybride. Ses avantages sont multiples : paiement pour la capacité de stockage et la bande passante en fonction de l’usage, changement d’échelle sur demande, disponibilité plus élevée…
* **La BDD NoSQL**. Poussées par l’essor du Big Data, elles sont utiles pour les larges ensembles de données distribuées, et parfaites pour analyser des quantités importantes de données non structurées, ou stockées sur plusieurs serveurs cloud virtuels.
* **La BDD orientée graph**. C’est un type de database NoSQL qui utilise la théorie des graphes pour stocker, cartographier et effectuer des requêtes sur les relations entre les données.

**RGPD :**

## **1. Identifier et Documenter les Données Traitées**

Le RGPD impose de savoir quelles données sont collectées, comment elles sont stockées et utilisées.

* Lister toutes les **données personnelles** collectées (nom, email, IP, cookies, etc.).
* Identifier les **finalités** de la collecte (ex. création de compte, statistiques, marketing).
* Vérifier la **durée de conservation** des données.
* S’assurer que seules les données **strictement nécessaires** sont collectées (**principe de minimisation**).

Tenir un **registre des traitements** (obligatoire pour les entreprises de plus de 250 employés ou traitant des données sensibles).