

# Bases de données relationnelles & Applications WEB

Document Réponses

Nom	Prénom
BAUDONNIERE	Mathis
GERBOUIN	Gabriel
POMMEREY	Guillaume
TOIN	Rodolphe

## Question 1 : Le Schéma conceptuel de la base de données

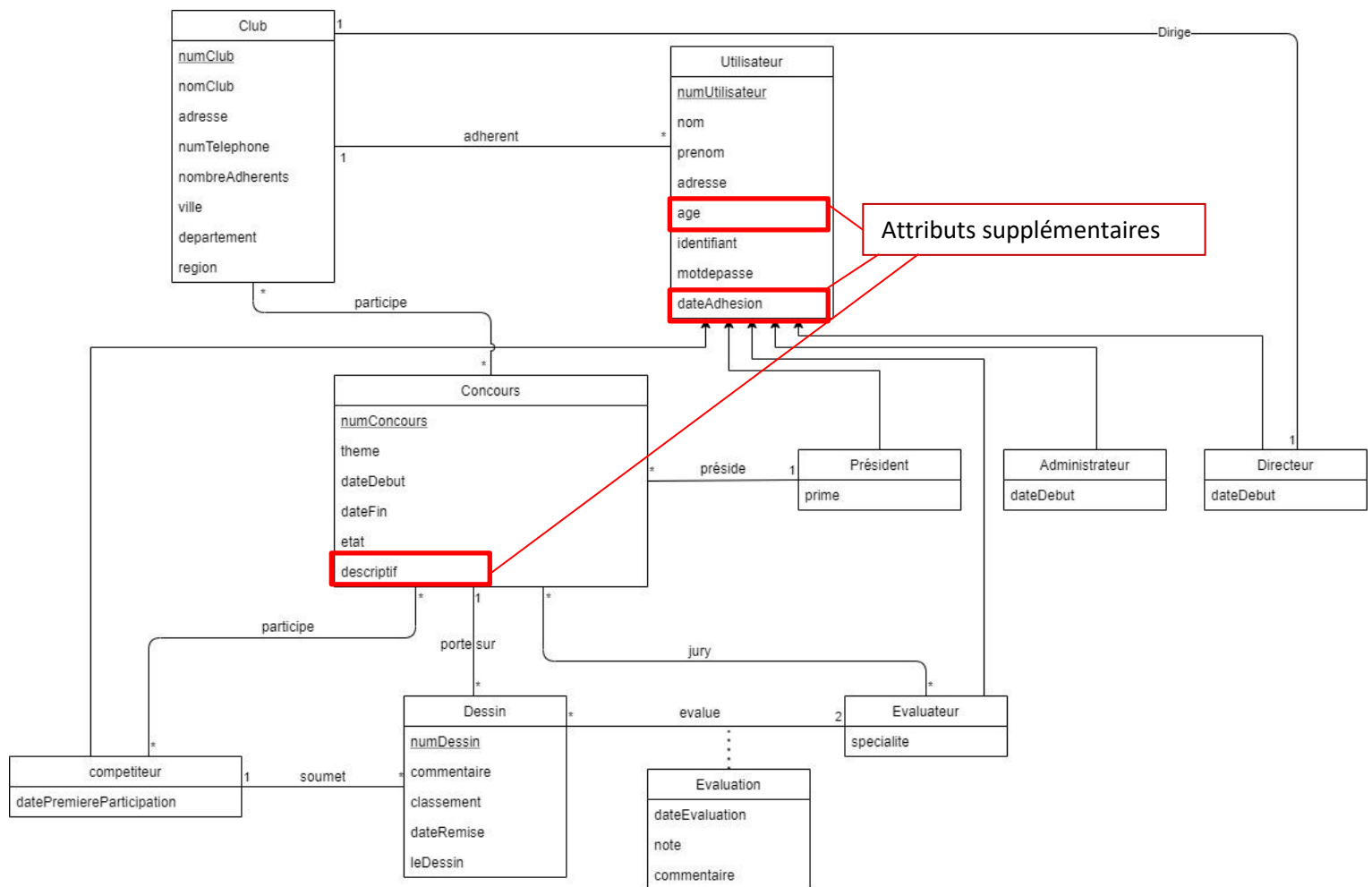


Figure 1 - Schéma conceptuel de notre BDD

Ce schéma conceptuel représente le contenu de la base en termes conceptuels, indépendamment de toute considération informatique et doit respecter un ensemble de contraintes donné par le projet. On retrouve sur ce schéma :

- Les entités
- Attributs
- Le Type de relation : cardinalités
- L'identifiant des liens

## Question 2 : Analyse du schéma conceptuel

Pour chacune des contraintes suivantes, nous précisons lesquelles sont explicitement définies dans le schéma conceptuel proposé précédemment.

- Un évaluateur ne pourra pas évaluer plus de deux dessins d'un même compétiteur dans un même concours, et ne pourra pas évaluer plus de 8 dessins en tout dans un même concours.  
Cette contrainte n'est pas représentée dans le schéma car elle concerne un concours en particulier.
- Un dessin doit être évalué par deux évaluateurs qui constituent un Jury.  
Cette contrainte est représentée par les cardinalités des liens « jury » (\* \*) et « évalue » (\* 2).
- Tout président d'un concours ne pourra pas être évaluateur ou compétiteur de ce même concours.  
Cette contrainte est représentée par les classes « Évaluateur » et « Compétiteur » qui sont distinctes. De plus, elles ne sont pas reliées à la classe « Dessin » de la même manière.
- Tout compétiteur ne peut pas déposer plus de trois dessins dans un même concours donné.  
Cette contrainte n'est pas représentée dans le schéma car elle concerne un concours en particulier.
- Un concours pour être organisé doit mobiliser au moins 2 régions et au moins 2 départements de chaque région. De plus, chaque département impliqué dans un concours doit mobiliser au moins 2 clubs (de la même ville ou pas).  
Cette contrainte n'est pas représentée dans le schéma car elle fait allusion à un attribut d'une classe et non pas une classe complète.
- Un évaluateur ne peut pas évaluer des dessins de son Club.  
Cette contrainte n'est pas représentée dans le schéma car le schéma ne montre pas l'appartenance des évaluateurs à un club en particulier.
- Tout club qui participe à un concours donné doit mobiliser au moins 6 compétiteurs et 3 évaluateurs.  
Cette contrainte n'est pas représentée dans le schéma puisque les tables « Compétiteur » et « Évaluateur » ne sont pas reliées à la table « Club ».
- Pour être président d'un concours, cinq années de licence au moins sont nécessaires.  
Cette contrainte n'est pas représentée dans le schéma.
- Tout dessin est proposé par un seul compétiteur.  
Cette contrainte est représentée par la cardinalité « 1 » du lien « soumet » entre « Compétiteur » et « Dessin ».
- Un évaluateur ne peut pas évaluer des dessins de son Club.  
Cette contrainte n'est pas représentée dans le schéma car le schéma ne montre pas l'appartenance des évaluateurs à un club en particulier.
- Le président d'un concours ne peut pas être d'une même région des clubs participants au concours.  
Cette contrainte n'est pas représentée dans le schéma car le schéma ne montre pas l'appartenance des présidents à une région en particulier.
- Pour être évaluateur dans un concours, un membre doit avoir au moins deux années de licence dans un club.  
Cette contrainte n'est pas représentée dans le schéma.

## Question 3 : Le schéma logique

### a) Formalisme textuel pour le schéma logique

#### Tables principales :

Concours (numConcours, numPresident\*, theme, descriptif, dateDebut, dateFin, etat : enum = 'pas commence', 'en cours', 'attente', 'resultat', 'evalue')

Club (numClub, nomClub, adresse, numTéléphone, nombreAdhérents, ville, département, région)

Utilisateur (numUtilisateur, numClub\*, nom, prenom, age, adresse, identifiant, motDePasse, dateAdhesion)

Président (numPresident\*, prime)

Evaluateur (numEvaluateur\*, specialite)

Competiteur (numCompetiteur\*, datePremParticipation)

Administrateur (numAdministrateur\*, dateDébut)

Directeur (numDirecteur\*, dateDébut)

Dessin (numDessin, numConcours\*, numCompetiteur\*, commentaire, classement, dateRemise, leDessin)

#### Tables relations :

Dirige (numClub\*, numDirecteur\*) (Cas particulier 1X1, la création de la table Dirige est adéquate)

Evaluation (numDessin\*, numEvaluateur\*, dateEvaluation, note, commentaire)

ParticipeClub (numClub\*, numConcours\*)

ParticipeCompetiteur (numCompetiteur\*, numConcours\*)

MembreJury (numConcours\*, numEvaluateur\*)

Ce modèle relationnel permet d'apporter du détail sur l'utilisation des clés étrangères entre les tables, sans référence à un langage de programmation particulier. On y retrouve :

- **Clé primaire** : Ensemble minimum d'attributs qui permet de distinguer chaque table de le rapport à tous les autres. Il est obligatoire que chaque table ait une clé primaire.
- **Clé étrangère** : fait référence à la clé primaire d'une autre table et permet un lien logique.

## b) Formalisme graphique pour le schéma logique

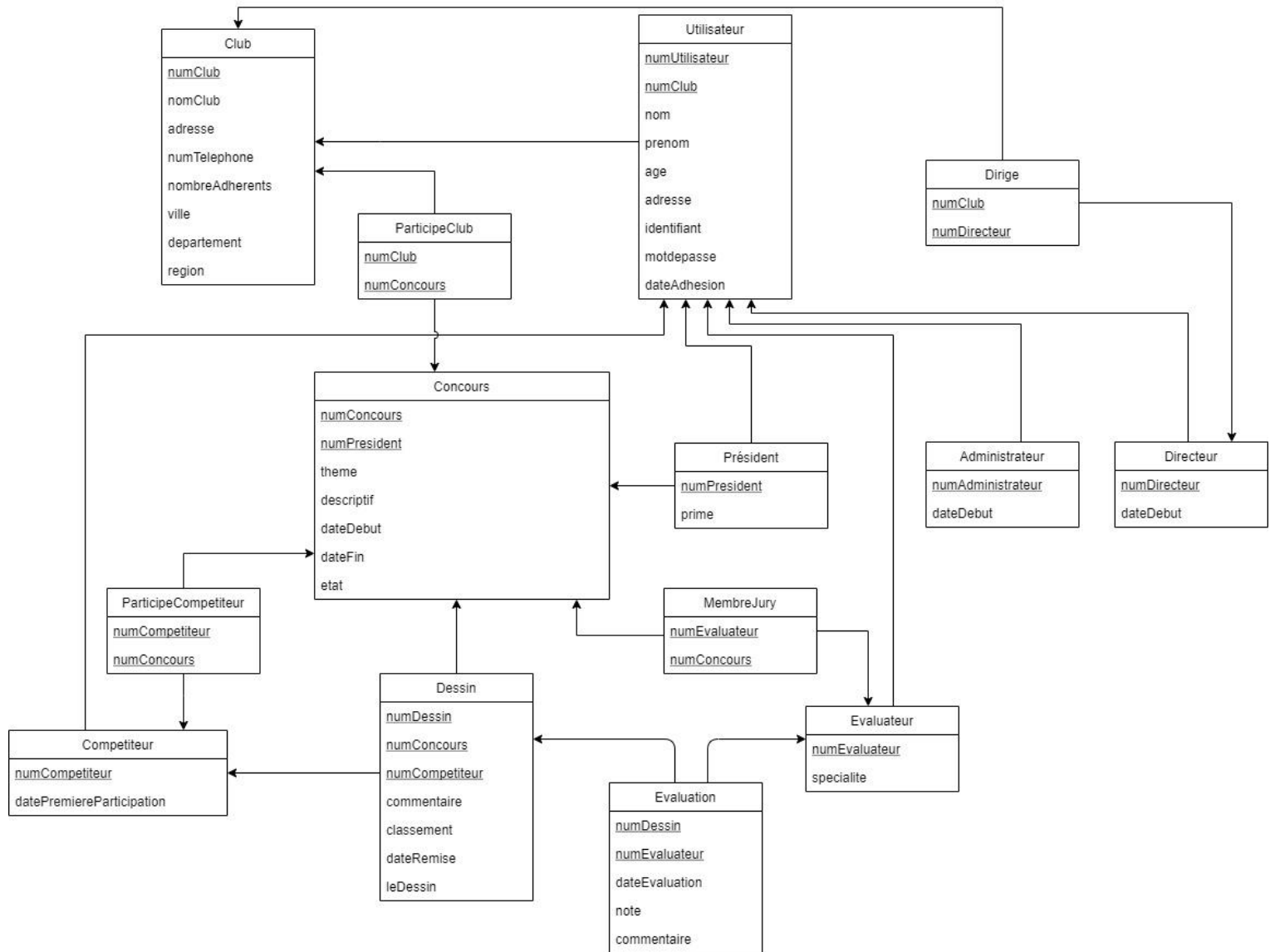


Figure 2- Schéma logique de notre BDD

## Question 4 : Le schéma physique

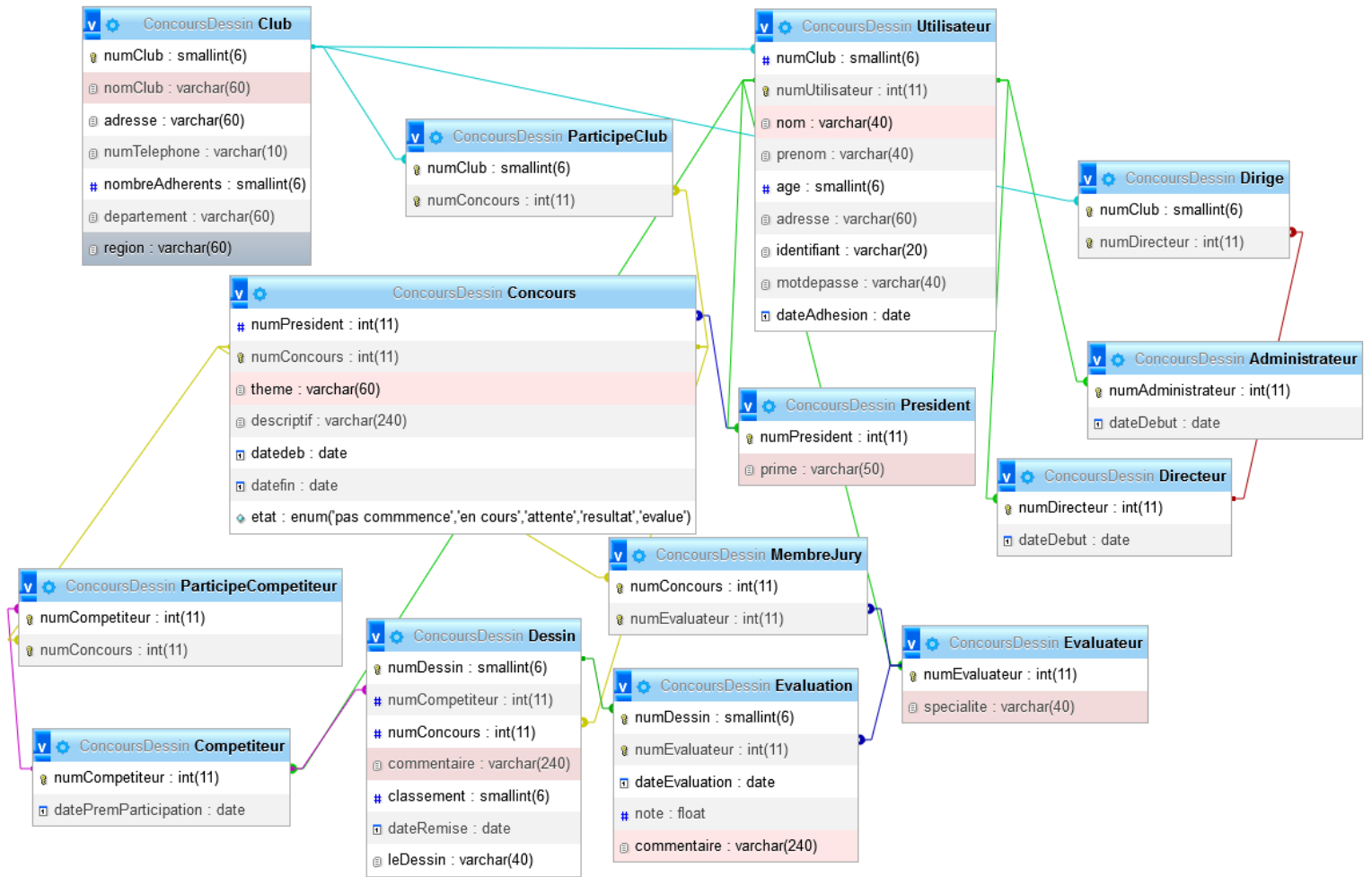


Figure 3- Schéma physique de notre BDD

## Question 5 : Remplissage de la BD

Pour le remplissage de notre base de données nous avons créé une feuille de tableau Excel par classe. Nous avons prêté une attention particulière au respect des contraintes demandées par le projet. Cette partie très sensible du projet devait être effectuée avec soin si nous espérions travailler avec des données logiques et cohérentes. Nous avons obtenu avec beaucoup d'efforts un fichier Excel capable de nous fournir l'entièreté des données nécessaires au projet tout en respectant les contraintes données, comme les suivantes :

- Le premier utilisateur de chaque club est le directeur de club.
- Sur le même principe les 3 utilisateurs suivant le directeur sont les évaluateurs.
- Enfin les utilisateurs restants sont les compétiteurs.

Précisément, cela revient à ordonner les numéros d'utilisateurs de la manière suivante :

numUtilisateur	Rôle
1	Admin
2 / 12 / 22 / 32 ..... 152	Le premier de chaque club est directeur de club
3 4 5 / 13 14 15 .... 153 154 155	Evaluateur, les trois premiers après le Directeur
6 7 8 9 10 11 / 16 17 18 19 20 21 ...	Compétiteurs, les utilisateurs restants
De 2 à 161	Le Président, choisis parmi l'entièreté des utilisateurs

## Question 6 : Interrogation de la Bases de Données (Requêtes SQL)

**Requête 1 :** Afficher le nom et l'adresse et l'âge de tous les compétiteurs qui ont participé dans un concours en 2021. Vous afficherez aussi la description du concours la date de début et la date de fin. Le club du compétiteur, le département et la région.

```
SELECT Utilisateur.nom, Utilisateur.adresse, Utilisateur.age, Concours.descriptif,
Concours.datedeb, Concours.datefin, Club.nomClub as "Club", Club.departement, Club.region
FROM Utilisateur, Concours, Club, Competiteur, ParticipeCompetiteur, ParticipeClub
Where Utilisateur.numUtilisateur = Competiteur.numCompetiteur
AND Competiteur.numCompetiteur = ParticipeCompetiteur.numCompetiteur -- On s assure que l
on parle bien du même utilisateur
AND ParticipeCompetiteur.numConcours = Concours.numConcours -- On s assure qu'on
sélectionne un concours de ce même utilisateur
AND ParticipeClub.numClub = Utilisateur.numClub -- On sélectionne le numéro de club
de l utilisateur
AND ParticipeClub.numConcours = Concours.numConcours
AND Utilisateur.numClub = Club.numClub -- On lie la table Utilisateur et la table
Club par leur clé numClub
-- Fin de la sélection, début des conditions
AND YEAR(Concours.datedeb) = 2021 -- YEAR(une_date) permet de retourner l'année uniquement
```

**Requête 2 :** Afficher par ordre croissant de la note tous les dessins qui ont été évalués en 2022. Vous afficherez les informations suivantes : le numéro du dessin et la note attribuée, le nom du compétiteur, la description du concours et le thème du concours.

```
SELECT Evaluation.numDessin, Evaluation.note, Utilisateur.nom, Concours.descriptif,
Concours.theme
FROM Evaluation, Utilisateur, Concours, Dessin, Competiteur
WHERE Evaluation.numDessin = Dessin.numDessin -- On lie la table Evaluation a la table
Dessin
AND Dessin.numConcours = Concours.numConcours -- On lie la table Dessin a la table
Concours
AND Dessin.numCompetiteur = Competiteur.numCompetiteur -- On lie la table Dessin a la
table Competiteur
AND Competiteur.numCompetiteur = Utilisateur.numUtilisateur -- On lie la table Competiteur
a la table Evalueur
AND YEAR(Dessin.dateRemise) = 2022 -- YEAR(une_date) permet de retourner l'année
uniquement
ORDER BY Evaluation.note ASC; -- Tri dans l'ordre ascendant des notes
```



**Requête 3 :** Pour cette requête on vous demande d'afficher des informations sur tous les dessins qui ont été évalués et qui sont stockés dans la base. Voici les informations qu'on souhaite voir affichés : le numéro, l'année, la description du concours dans lequel le dessin a été évalué ; le nom du compétiteur ayant proposé le dessin ; le numéro et le commentaire du dessin fait par le compétiteur ; la note et le commentaire de l'évaluation ; le nom de l'évaluateur.

```
SELECT Concours.numConcours, YEAR(Concours.datedeb) AS année, Concours.descriptif AS
DescriptifConcours, User_Competiteur.nom AS NomCompetiteur, Dessin.numDessin,
Dessin.commentaire AS CommentaireDessin, Evaluation.note, Evaluation.commentaire AS
CommentaireNote, User_Evaluateur.nom AS NomEvalueur
FROM Concours, Utilisateur AS User_Competiteur, Utilisateur AS User_Evaluateur, Dessin,
Evaluation, Evalueur, Competiteur, ParticipeCompetiteur
WHERE User_Competiteur.numUtilisateur = Competiteur.numCompetiteur -- On lie la table
Utilisateur (pour le competiteur) a la table Competiteur
AND Competiteur.numCompetiteur = Dessin.numCompetiteur -- On lie la table Competiteur a la
table Dessin
AND Dessin.numConcours = Concours.numConcours -- On lie la table Dessin a la table
Concours
AND ParticipeCompetiteur.numConcours = Concours.numConcours -- On lie la table
ParticipeCompetiteur a la table Concours
AND User_Evaluateur.numUtilisateur = Evalueur.numEvalueur -- On lie la table
Utilisateur (pour l evalueur) a la table Evalueur
AND Evalueur.numEvalueur = Evaluation.numEvalueur -- On lie la table Evalueur a la
table Evaluation
AND Evaluation.numDessin = Dessin.numDessin -- On lie la table Evaluation a la table
Dessin
AND Competiteur.numCompetiteur = ParticipeCompetiteur.numCompetiteur -- On lie la table
Competiteur a la table ParticipeCompetiteur
AND Competiteur.numCompetiteur = ParticipeCompetiteur.numCompetiteur -- On lie la table
Competiteur a la table ParticipeCompetiteur
AND Evaluation.note IS NOT NULL; -- On exclue les dessins non notés
```

**Requête 4 :** Afficher le nom, prénom et âge des compétiteurs qui ont participé à tous les concours qui ont été organisés.

```
-- Suppression de la table TempTableRequete4 si elle n'a pas ete supprimee auparavant
drop table if exists TempTableRequete4;

-- Creation de la table temporaire TempTableRequete4
Create Table TempTableRequete4
(
nomUtil          Varchar (60) NOT NULL,
prenomUtil       Varchar (60) NOT NULL,
age              smallint check (age between 0 and 150),
nbrConcours      smallint NOT NULL);

-- Premiere requete, pour selectionner le nom, prenom et age de chaque competiteur, et le
nombre de participations pour chaque competiteur
-- Requete inseree dans la table temporaire TempTableRequete4
INSERT INTO TempTableRequete4
SELECT Utilisateur.nom, Utilisateur.prenom, Utilisateur.age,
COUNT(ParticipeCompetiteur.numConcours) AS nbrConcours
FROM Utilisateur, Competiteur, ParticipeCompetiteur, Concours
WHERE Utilisateur.numUtilisateur = Competiteur.numCompetiteur -- On lie la table
Utilisateur à la table Competiteur
AND Competiteur.numCompetiteur = ParticipeCompetiteur.numCompetiteur -- On lie la table
Competiteur à la table ParticipeCompetiteur
AND ParticipeCompetiteur.numConcours = Concours.numConcours -- On lie la table
ParticipeCompetiteur à la table Concours
GROUP BY ParticipeCompetiteur.numCompetiteur; -- On regroupe les résultats par rapport au
numéro de competiteur

-- Seconde requete, pour selectionner uniquement les competiteurs stockes dans la table
TempTableRequete4 qui ont participe a tous les concours
SELECT *
FROM TempTableRequete4
WHERE nbrConcours = 8;

-- Suppression de la table TempTableRequete4 en fin de requete
drop table if exists TempTableRequete4;
```

**Requête 5 :** Afficher le nom et la moyenne des notes de la région qui a la meilleure moyenne des notes des dessins proposés.

```
SELECT Club.region, AVG(Evaluation.note) as moyenne
FROM Club, Evaluation, Utilisateur, Competiteur, Dessin
WHERE Club.numClub = Utilisateur.numClub -- On lie la table Club à la table Utilisateur
AND Utilisateur.numUtilisateur = Competiteur.numCompetiteur -- On lie la table Utilisateur
à la table Competiteur
AND Competiteur.numCompetiteur = Dessin.numCompetiteur -- On lie la table Competiteur à la
table Dessin
AND Dessin.numDessin = Evaluation.numDessin -- On lie la table Dessin à la table
Evaluation
GROUP BY Club.region -- Regroupe les résultats par région
ORDER BY AVG(Evaluation.note) DESC -- Trie les résultats par note moyenne décroissante
LIMIT 1 -- On ne garde que le 1er resultat (la meilleure moyenne)
```

**Requête 6 :** Afficher le nombre de dessins évalués par chaque évaluateur, pour chaque concours. Afficher le nombre de dessins, le nom de l'évaluateur, le numéro du concours.

```
SELECT Utilisateur.nom, Utilisateur.prenom, COUNT(Evaluation.note) AS nombreDeNotes,
Concours.numConcours
FROM Utilisateur, Evaluation, Concours, Dessin, Evalueur
WHERE Evaluation.numDessin = Dessin.numDessin -- On lie la table Evaluation et la table
Dessin
AND Dessin.numConcours = Concours.numConcours -- On lie la table Dessin et la table
Concours
AND Evaluation.numEvalueur = Evalueur.numEvalueur -- On lie la table Evaluation et
la table Evalueur
AND Evalueur.numEvalueur = Utilisateur.numUtilisateur -- On lie la table Evalueur et
la table Utilisateur
GROUP BY Evaluation.numEvalueur, Concours.numConcours -- On regroupe les résultats par
rapport au numéro d'évaluateur et au numéro de concours
ORDER BY Concours.numConcours ASC; -- On trie les résultats par numero de concours
croissant
```

**Requête 7 :** Afficher le nom de l'évaluateur le plus sévère pour chaque concours ; la moyenne des notes qu'il a attribuées ; le numéro du concours évalué.

```
-- Suppression de la table TempTableRequete7 si elle n'a pas ete supprimee auparavant
drop table if exists TempTableRequete7;

-- Creation de la table temporaire TempTableRequete7
Create Table TempTableRequete7
(
nomEval          Varchar (60) NOT NULL,
prenomEval       Varchar (60) NOT NULL,
moyenne          float check (moyenne between 0 and 20),
numConcours      smallint NOT NULL);

-- Premiere requete, pour selectionner le nom et prenom de chaque evaluateur, et la
moyenne qu'il a donne pour tous les concours
-- Requete inseree dans la table temporaire TempTableRequete7
INSERT INTO TempTableRequete7
SELECT Utilisateur.nom, Utilisateur.prenom, AVG(Evaluation.note) AS moyenne,
Concours.numConcours
FROM Utilisateur, Evaluation, Concours, Dessin, Evalueur
WHERE Evaluation.numDessin = Dessin.numDessin -- On lie la table Evaluation et la table
Dessin
AND Dessin.numConcours = Concours.numConcours -- On lie la table Dessin et la table
Concours
AND Evaluation.numEvalueur = Evalueur.numEvalueur -- On lie la table Evaluation et
la table Evalueur
AND Evalueur.numEvalueur = Utilisateur.numUtilisateur -- On lie la table Evalueur et
la table Utilisateur
GROUP BY Evaluation.numEvalueur, Concours.numConcours -- On regroupe les résultats par
rapport au numéro d'évaluateur et au numéro de concours
ORDER BY AVG(Evaluation.note) ASC; -- On trie les résultats par note moyenne croissante

-- Seconde requete, pour selectionner la moyenne minimale par concours
SELECT nomEval, prenomEval, MIN(moyenne) AS moyenneMinimale, numConcours
FROM TempTableRequete7
GROUP BY numConcours; -- On regroupe les résultats par rapport au numéro de concours

-- Suppression de la table TempTableRequete7 en fin de requete
drop table if exists TempTableRequete7;
```

**Requête 8 :** Afficher le nom et la moyenne des notes de chaque club, tous concours confondus. Afficher également le nombre total de dessins réalisés par chacun de ces clubs.

```
SELECT Club.nomClub, AVG(Evaluation.note) AS moyenne, COUNT(Dessin.numDessin) AS
nombreDeDessins
FROM Club, Evaluation, Dessin, Concours, ParticipeClub
WHERE Evaluation.numDessin = Dessin.numDessin -- On lie la table Evaluation et la table
Dessin
AND Dessin.numConcours = Concours.numConcours -- On lie la table Dessin et la table
Concours
AND Concours.numConcours = ParticipeClub.numConcours -- On lie la table Concours et la
table ParticipeClub
AND ParticipeClub.numClub = Club.numClub -- On lie la table ParticipeClub et la table Club
GROUP BY Club.nomClub -- On regroupe les résultats par rapport au nom de Club
ORDER BY AVG(Evaluation.note) ASC -- On trie les résultats par note moyenne croissante
```

## Question 7 : Une interface WEB pour le Système d'information

L'interface que nous avons créée permet réaliser quelques interrogations simples de notre base de données. Nous sommes restés simple en termes de design, même si notre fichier *style.css* a été enrichi pour pouvoir convenablement afficher nos tableaux de résultats, nos boutons et notre header selon un design uniformisé. Ci-dessous une présentation brève des principales briques de notre interface web :

### 1) La page d'identification

Notre page d'identification permet de se connecter à l'application web. Deux comptes sont permis :

- Compte utilisateur :
  - identifiant : user
  - mot de passe : userpass
- Compte administrateur
  - identifiant : admin
  - mot de passe : adminpass

Si un couple identifiant/mot de passe n'est pas correct alors un message d'erreur sera affiché. En cas de connexion réussie, la variable `$_SESSION` sera agrémenté d'un champ « user » qui stockera l'identifiant de l'utilisateur connecté et d'un champ booléen « is\_admin » qui permet de mémoriser si l'utilisateur connecté possède les droits d'administration.

Le formulaire renvoie vers la page *check\_login.php* qui vérifiera le couple identifiant/mot de passe. Si l'identification est approuvée, l'utilisateur sera renvoyé vers *index.php*, sinon vers *login.php*.



The screenshot displays a web interface for login. At the top, a purple header bar contains the 'eseo' logo with the tagline 'INNOVATION MAKES SENSE' on the left, and navigation links 'Accueil' and 'Connexion' on the right. The main content area has a light green background with the title 'Connexion' centered. Below the title is a purple rounded rectangle containing a login form. The form has two input fields: the first contains the email 'rodolphe.toin@reseau.eseo.fr' and the second is masked with dots. Below these fields, a red error message reads 'Identifiant/mot de passe incorrect'. At the bottom of the form is a dark grey button labeled 'Connexion'.

Figure 4- Page "login.php"

## 2) Le header

Le header est actualisé en fonction de l'utilisateur actuellement connecté. Un lien « Déconnexion » est également ajouté afin de pouvoir se déconnecter (fichier *logout.php*).

Le lien « Requête » est également ajouté pour pouvoir accéder à tout moment à la page *requestHub.php*.



Figure 5- header.php

## 3) Les pages de requêtage


La page *requestHub.php* présente 4 boutons permettant d'accéder aux pages *request1.php*, *request2.php*, *request3.php* et *user\_stats.php*.

Les pages *request1.php*, *request2.php*, *request3.php* présentent les résultats des 3 requêtes personnalisées imposées dans le cadre de ce projet. Le but de la requête vient titrer la page, la requête effectuée est affichée suivi du résultat retourné, affiché dans un tableau.


La page *user\_stats.php* accessible via le bouton « Statistiques utilisateurs » permet d'afficher des données numériques calculées sur l'utilisateur.

- Une première requête permet de remplir la liste déroulante « Utilisateur » avec les noms des différents utilisateurs présents dans la base
- En appuyant sur le bouton « Afficher les statistiques » une seconde requête est effectuée à l'aide du nom sélectionné dans le menu déroulant. Plusieurs grandeurs sont calculées, à savoir le nombre de concours auquel l'utilisateur a participé, le nombre total de dessins qu'il a soumis, sa moyenne totale et ses notes minimale et maximale. Ci-dessous la requête utilisée :

```
SELECT nom, prenom, Count(DISTINCT theme) as nbrConcours, COUNT(note) as  
nbrParticipations, ROUND(AVG(note), 2) as moyenne, MIN(note) as minimum, MAX(note) as  
maximum  
FROM Dessin, Evaluation, Competiteur, Utilisateur, Concours  
WHERE Dessin.numCompetiteur = Competiteur.numCompetiteur  
AND Dessin.numDessin = Evaluation.numDessin  
AND Competiteur.numCompetiteur = Utilisateur.numUtilisateur  
AND Concours.numConcours = Dessin.numConcours  
AND nom LIKE ?
```



[Accueil](#)
[Requêtes](#)

 admin | 
 [Déconnexion](#)

**Afficher le nom et la moyenne des notes de chaque club, tous concours confondus. Afficher également le nombre total de dessins réalisés par chacun de ces clubs.**


```

SELECT Club.nomClub, AVG(Evaluation.note) AS moyenne, COUNT(Dessin.numDessin) AS
nombreDeDessins FROM Club, Evaluation, Dessin, Concours, ParticipeClub WHERE
Evaluation.numDessin = Dessin.numDessin AND Dessin.numConcours = Concours.numConcours AND
Concours.numConcours = ParticipeClub.numConcours AND ParticipeClub.numClub = Club.numClub
GROUP BY Club.nomClub ORDER BY AVG(Evaluation.note) ASC
        
```


Club	Moyenne (ordre croissant)	Nbr total de dessins
Club_8	10.109337408709443	1147
Club_6	10.109337408709443	1147
Club_5	10.109337408709443	1147
Club_7	10.109337408709443	1147
Club_3	10.119616062567289	1146
Club_2	10.119616062567289	1146
Club_1	10.119616062567289	1146
Club_4	10.119616062567289	1146
Club_14	10.125179046132159	1145
Club_16	10.125179046132159	1145
Club_13	10.125179046132159	1145
Club_15	10.125179046132159	1145
Club_9	10.13548951724974	1144
Club_12	10.13548951724974	1144
Club_11	10.13548951724974	1144
Club_10	10.13548951724974	1144

Retour

Figure 6- Page "request3.php"



[Accueil](#)
[Requêtes](#)

 admin | 
 [Déconnexion](#)

### Statistiques des utilisateurs

Utilisateur: 

Afficher les statistiques

Nom	Prénom	Nbr total de concours	Nbr de participations	Moyenne	Pire note	Meilleur note
Gaudin	Véronique	6	72	10.21	2.64	15.83

Retour

Figure 7- Page "user\_stats.php"