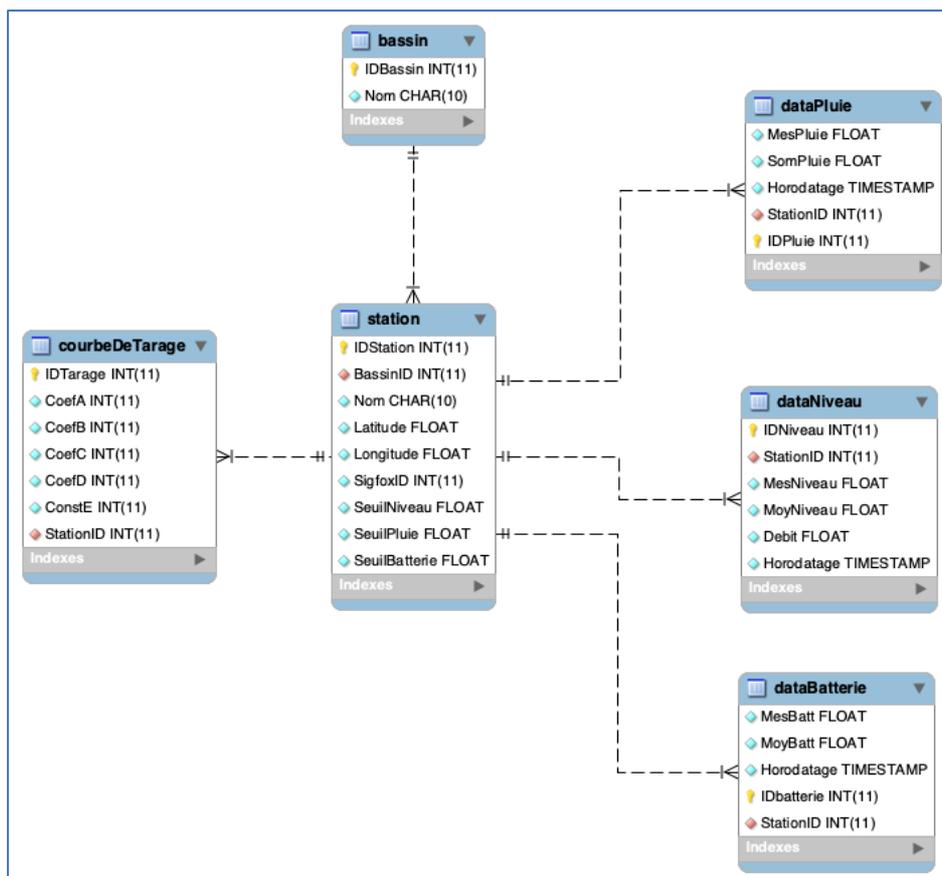


CORRIGE ACTIVITES BASE DE DONNEE

Q16. Préciser, à l'aide des documentations PP8 et PP9, le code SQL permettant de créer la table **dataNiveau**.

```
CREATE TABLE dataNiveau (  
  IDniveau int(10) NOT NULL AUTO_INCREMENT,  
  StationID int(10) NOT NULL, MesNiveau float DEFAULT NULL, MoyNiveau  
  float DEFAULT NULL, Debit float DEFAULT NULL,  
  Horodatage timestamp NULL DEFAULT CURRENT_TIMESTAMP, PRIMARY KEY  
  (IDniveau),  
  CONSTRAINT FOREIGN KEY (StationID) REFERENCES station (IDstation) );
```



```

-- Création de la table 'bassin'
CREATE TABLE bassin (
  IDBassin INT(11) NOT NULL,
  Nom CHAR(10),
  PRIMARY KEY (IDBassin)
);

-- Création de la table 'station'
CREATE TABLE station (
  IDStation INT(11) NOT NULL,
  BassinID INT(11),
  Nom CHAR(10),
  Latitude FLOAT,
  Longitude FLOAT,
  SigfoxID INT(11),
  SeuilNiveau FLOAT,
  SeuilPluie FLOAT,
  SeuilBatterie FLOAT,
  PRIMARY KEY (IDStation),
  FOREIGN KEY (BassinID) REFERENCES bassin(IDBassin)
);

-- Création de la table 'courbeDeTarage'
CREATE TABLE courbeDeTarage (
  IDTarage INT(11) NOT NULL,
  CoefA INT(11),
  CoefB INT(11),
  CoefC INT(11),
  CoefD INT(11),
  ConstE INT(11),
  StationID INT(11),
  PRIMARY KEY (IDTarage),
  FOREIGN KEY (StationID) REFERENCES station(IDStation)
);

-- Création de la table 'dataPluie'
CREATE TABLE dataPluie (
  IDPluie INT(11) NOT NULL,
  MesPluie FLOAT,
  SomPluie FLOAT,
  Horodatage TIMESTAMP,
  StationID INT(11),
  PRIMARY KEY (IDPluie),
  FOREIGN KEY (StationID) REFERENCES station(IDStation)
);

-- Création de la table 'dataNiveau'
CREATE TABLE dataNiveau (
  IDNiveau INT(11) NOT NULL,
  StationID INT(11),
  MesNiveau FLOAT,
  MoyNiveau FLOAT,
  Debit FLOAT,
  Horodatage TIMESTAMP,
  PRIMARY KEY (IDNiveau),
  FOREIGN KEY (StationID) REFERENCES station(IDStation));

```

```
-- Création de la table 'dataBatterie'
CREATE TABLE dataBatterie (
  IDBatterie INT(11) NOT NULL,
  MesBatt FLOAT,
  MoyBatt FLOAT,
  Horodatage TIMESTAMP,
  StationID INT(11),
  PRIMARY KEY (IDBatterie),
  FOREIGN KEY (StationID) REFERENCES station(IDStation)
);
```

Q17. Préciser, à l'aide des documentations PP8 et PP9, le code SQL permettant d'insérer un nouvel enregistrement dans la table **dataNiveau** avec les valeurs

suivantes : **IDstation** = 1, niveau instantané = 0,17 m, niveau moyen = 0,16 m, débit instantané = 0,6 m³/s.

```
INSERT INTO dataNiveau (StationID, MesNiveau, MoyNiveau, Debit)
VALUES (1, 0.17, 0.16, 0.6);
```

Q18. Expliquer, à l'aide des documentations PP8 et PP9, la raison de cette erreur et proposer une solution au problème. Les requêtes SQL ne sont pas demandées.

C'est à cause d'une contrainte de clé étrangère. Le champ station .IDStation sert de clé étrangère dans les tables dataPluie, dataNiveau, dataBatterie et courbeDeTarage.

Il faut d'abord supprimer toutes les références à cette station dans les tables dataPluie, dataNiveau, dataBatterie et courbeDetarage.

Q19. Préciser, à l'aide des documentations PP8 et PP9, le code SQL permettant de sélectionner toutes les moyennes de niveaux d'eau de la station qui s'appelle

« Wimille » en commençant par la plus récente.

```
SELECT MoyNiveau FROM dataNiveau, station WHERE station.IDstation =
dataNiveau.StationID AND station.Nom= 'Wimille'
ORDER BY Horodatage DESC
```

Ou

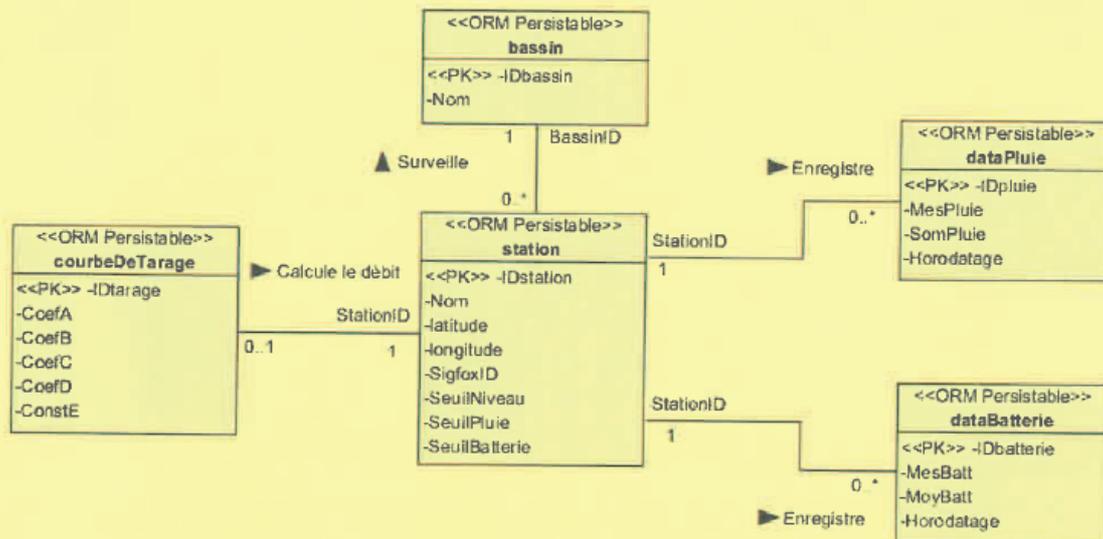
```
SELECT MoyNiveau FROM dataNiveau INNER JOIN station ON
dataNiveau.StationID=station.IDStation WHERE station.Nom='Wimille'
ORDER by Horodatage DESC
```

La première méthode était plus courante dans les anciennes versions de SQL, mais elle est généralement considérée comme moins lisible et moins préférable par rapport à la syntaxe JOIN explicite, surtout lorsque la jointure implique plusieurs tables.

INNER JOIN renvoie les lignes où il y a une correspondance dans les deux tables. Si une ligne de l'une des tables n'a pas de correspondant dans l'autre table, cette ligne n'est pas incluse dans le résultat final.

Documentation PP8 : Base de données « vigicrues »

Le service web de collecte enregistre les données des stations de mesures dans la base de données MySQL « vigicrues ». Cette base de données est décrite par le schéma conceptuel suivant (notation UML2) :



Ce modèle conceptuel génèrera le code SQL suivant pour créer la table dataBatterie :

```
CREATE TABLE databatterie (
    IDbatterie int(10) NOT NULL AUTO_INCREMENT,
    StationID int(10) NOT NULL,
    MesBatt float DEFAULT NULL,
    MoyBatt float DEFAULT NULL,
    Horodatage timestamp NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (IDbatterie),
    CONSTRAINT FOREIGN KEY (StationID) REFERENCES station
(IDstation)
);
```

Le champ Horodatage des tables est rempli automatiquement par Mysql avec la date courante lors de chaque insertion d'un nouvel enregistrement.

La table courbeDeTarage contient les coefficients de la fonction polynomiale d'ordre 4 qui interpole la courbe de tarage établie par l'hydrologue.

Débit = CoefA * Niveau⁴ + CoefB * Niveau³ + CoefC * Niveau² + CoefD * Niveau + ConstE

Documentation PP9 : Rappel des syntaxes SQL de Mysql

Utiliser (rendre active) une base de données existante :	<code>USE nom_de_la_base ;</code>
Créer une base de données :	<code>CREATE DATABASE nom_de_la_base ;</code>
Supprimer une base de données	<code>DROP DATABASE nom_de_la_base ;</code>
Créer une table dans la base de données active:	<code>CREATE TABLE nomTable (id INT NOT NULL AUTO_INCREMENT, champ1 DOUBLE, champ2 VARCHAR, champ3 TIMESTAMP NOT NULL, ..., PRIMARY KEY(id)) ;</code>
Sélectionner toutes les informations de la table :	<code>SELECT * FROM nomTable ;</code>
Sélectionner seulement les informations d'un champ :	<code>SELECT nomChamp FROM nomTable ;</code>
Sélectionner tous les champs de la table nomTable correspondant à deux critères.	<code>SELECT * FROM nomTable WHERE nomChamp1 = 'poste' AND nomChamp3 < 12 ;</code>
Sélectionner sur plusieurs tables nomTable1.nomChamp1 est clé primaire. nomTable2.nomChamp4 est une clé étrangère vers nomTable1.	<code>SELECT * FROM nomTable1, nomTable2 WHERE nom_table1.nomChamp1 = nom_table2.nomChamp4 ;</code>
Trier les résultats par ordre croissant	<code>SELECT nomChamp FROM nomTable ORDER BY nomChamp ASC ;</code>
Trier les résultats par ordre décroissant	<code>SELECT nomChamp FROM nomTable ORDER BY nomChamp DESC ;</code>
Écrire une nouvelle entrée dans une table de BDD :	<code>INSERT INTO nomTable(champ1, champ2) VALUES('valeur_texte1', 0.867) ;</code>
Modifier les informations de l'entrée dont le champ id = 51 :	<code>UPDATE nomTable SET nomChamp1=10, valeur2=32 WHERE id=51 ;</code>
Supprimer une entrée dans une table	<code>DELETE FROM nomTable WHERE champ1='valeur1' ;</code>

Q19. Expliquez le rôle de la clé primaire « idMesure » de la table « mesures ».

Dans une base de données relationnelle, une clé primaire est une contrainte d'unicité qui permet d'identifier de manière unique un enregistrement dans une table.

Q20. Écrire en langage SQL la requête permettant d'écrire cette nouvelle acquisition dans la base de données.

```
INSERT INTO mesures(tension_pan, courant_pan, energie, tension_bat,
temperature) VALUES( 24.0, 18.8, 321.2, 23.8, 34.0) ;
```

Q21. En vous aidant des documentations PP4 et PP5, donner la requête SQL permettant de créer la table MPPT dans base de données nommée « solaire »

```
CREATE TABLE MPPT (idMPPT INT NOT NULL AUTO_INCREMENT,
CAN_ID INT, nom VARCHAR, PRIMARY KEY (idMPPT) ) ;
```

Q22. Quel champ doit-on ajouter à la table « mesures » afin de modéliser la relation entre les mesures et la carte MPPT les ayant effectuées.

Un champ idMPPT de même type que celui de la table MPPT

Q23. Compléter le schéma de la base de données en indiquant le nouveau champ de la table « mesures » ainsi que les cardinalités entre la table « mesures » et la table « MPPT ».

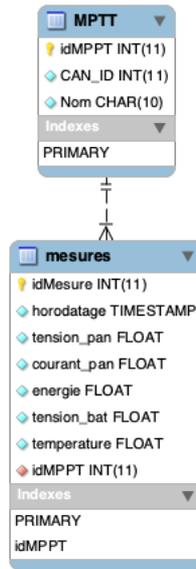
MPPT		mesures		utilisateurs
<u>idMPPT</u>	1	<u>idMesure</u>		<u>idUser</u>
CAN_ID		horodatage		login
nom		tension_pan		mdp
		courant_pan		
		energie		
		tension_bat		
	N	temperature		
		idMPPT		

Q24. Donner le rôle du champ que vous avez ajouté à la table « mesures ».

clé étrangère

Q25. Écrire en langage SQL la requête permettant de récupérer les valeurs de l'énergie liée au MPPT dont le champ « CAN_ID » a pour valeur 3.

```
select energie from mesures, MPPT where
mesures.idMPPT=MPPT.idMPPT and idCAN=3 ;
```



```
SELECT mesures.energie FROM mesures INNER JOIN MPPT ON  
mesures.idMPPT=MPPT.idMPPT WHERE MPPT.CAN_ID=3
```