

Base de données électrochimiques

1. Introduction

Il s'agit de construire et d'utiliser une base de données SQL comportant les données suivantes :

- ▷ Éléments chimiques.
- ▷ Composés chimiques.
- ▷ Demi-équations redox et potentiels standard (en solution aqueuse).
- ▷ Courbes courant-potentiel expérimentales.

2. Structure de la base

2.a. Table elements

Toutes les tables sont préfixées par `ch`. Voici la définition SQL de cette table :

```
CREATE TABLE IF NOT EXISTS 'ch_elements' (  
  'Symbole' char(2),  
  'Nom' tinytext,  
  'Z' tinyint(3),  
  'M' float(9,6),  
  'Groupe' tinyint(2),  
  'Periode' tinyint(2),  
  'Bloc' tinytext,  
  'Orbitales' tinytext,  
  'Electroneg' float(9,6),  
  'Eionisation' float(9,6),  
  PRIMARY KEY ('Symbole')  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_cs;
```

Pour toutes les tables, on choisit un encodage des caractères `latin1` et un interclassement `latin1_general_cs`, qui permet de faire des requêtes sensibles à la casse (pour différencier par exemple CO et Co).

Voici la signification de ses champs :

- ▷ **Symbole** : symbole de l'élément chimique (maximum 2 caractères), qui constitue la clé primaire.
- ▷ **Nom** : nom de l'élément.
- ▷ **Z** : numéro atomique.
- ▷ **M** : masse molaire (en g/mol).
- ▷ **Groupe** : groupe (colonne du tableau périodique).
- ▷ **Periode** : période (ligne du tableau périodique).
- ▷ **Bloc** : bloc (s,p,d ou f).

- ▷ Orbitales : structure électronique de la couche de valence.
- ▷ Electroneg : électronégativité de Pauling.
- ▷ Einonisation : énergie de première ionisation (en eV).

Voici par exemple un enregistrement pour l'élément Silicium :

```
'Si', 'Silicium', 14, 28.090000, 14, 3, 'p', '3s23p2', 1.900000, 8.150000
```

Cette table est déjà remplie (même s'il manque quelques structures électroniques).

2.b. Table composes

Cette table contient des composés chimiques (ou des ions). Elle peut aussi contenir des éléments simples, par exemple les éléments métalliques. Voici sa définition :

```
CREATE TABLE IF NOT EXISTS 'ch_composes' (
  'Symbole' char(20),
  'Charge' tinyint(2),
  PRIMARY KEY ('Symbole')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_cs;
```

Voici la signification de ses champs :

- ▷ Symbole : symbole du composé. Les charges sont entre parenthèses. Ce champ constitue la clé primaire.
- ▷ Charge : nombre de charges.

Voici par exemple l'enregistrement pour deux ions et pour l'eau :

```
'Fe(2+)', 2
'SbO(+)', 1
'H2O', 0
```

2.c. Table compositions

Cette table contient la composition des composés définis dans la table composes, c'est-à-dire les éléments chimiques et leur nombre.

Voici la définition de la table :

```
CREATE TABLE IF NOT EXISTS 'ch_compositions' (
  'Id' smallint(6) NOT NULL AUTO_INCREMENT,
  'Compose' char(20), /* clé table ch_composes (Symbole) */
  'Element' char(2), /* clé table ch_elements (Symbole) */
  'Nombre' tinyint(2),
  PRIMARY KEY ('Id')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_cs;
```

Voici la signification de ses champs :

- ▷ Id : identifiant de l'enregistrement (auto incrémenté), sans signification particulière, mais constituant la clé primaire.
- ▷ Compose : composé, clé étrangère sur le champ `ch_composes.Symbole`.
- ▷ Element : élément, clé étrangère sur le champ `ch_elements.Symbole`.
- ▷ Nombre : nombre d'éléments.

On remarque que les deux clés étrangères font référence à des clés primaires des tables `ch_composes` et `ch_elements`.

Voici en exemple les enregistrements pour l'eau et pour un ion :

```
4, 'H2O', 'H', 2
5, 'H2O', 'O', 1
11, 'SbO(+)', 'Sb', 1
12, 'SbO(+)', 'O', 1
```

2.d. Table potentiels

Cette table contient les demi-équations redox et les potentiels redox correspondants. Voici sa définition :

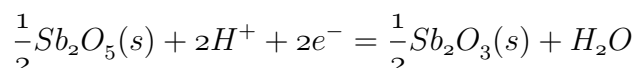
```
CREATE TABLE IF NOT EXISTS 'ch_potentiels' (
  'Id' smallint(6) NOT NULL AUTO_INCREMENT,
  'Famille' char(2),
  'Oxydant' char(20), /* clé table ch_elements (Symbole) */
  'NOxydant' float(4,2),
  'EtatOxydant' char(2),
  'Reducteur' char(20), /* clé table ch_composes (Symbole) */
  'NReducteur' float(4,2),
  'EtatReducteur' char(2),
  'NElectrons' float(4,2),
  'CompOx' char(20), /* clé table ch_composes (Symbole) */
  'NCompOx' float(4,2),
  'EtatCompOx' char(2),
  'CompRed' char(20), /* clé table ch_composes (Symbole) */
  'NCompRed' float(4,2),
  'EtatCompRed' char(2),
  'Potentiel' float(4,2), /* en V */
  PRIMARY KEY ('Id')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_cs;
```

Voici la signification de ses champs :

- ▷ Id : identifiant auto incrémenté, qui constitue la clé primaire.
- ▷ Famille : élément principal, clé étrangère sur le champ `ch_elements.Symbole`.
- ▷ Oxydant : symbole de l'oxydant, clé étrangère sur le champ `ch_composes.Symbole`.
- ▷ NOxydant : coefficient stœchiométrique de l'oxydant.
- ▷ EtatOxydant : état physique de l'oxydant (s,g), que l'on laisse vide s'il est soluble.

- ▷ **Reducteur** : symbole du réducteur, clé étrangère sur le champ `ch_composes.Symbole`.
- ▷ **NReducteur** : coefficient stœchiométrique du réducteur.
- ▷ **EtatReducteur** : état physique du réducteur.
- ▷ **NElectrons** : nombre d'électrons échangés.
- ▷ **CompOx** : composé complémentaire associé à l'oxydant.
- ▷ **NCompOx** : coefficient stœchiométrique du composé complémentaire associé à l'oxydant.
- ▷ **EtatCompOx** : état physique du composé complémentaire associé à l'oxydant.
- ▷ **CompRed** : composé complémentaire associé au réducteur.
- ▷ **NCompRed** : coefficient stœchiométrique du composé complémentaire associé au réducteur.
- ▷ **EtatCompRed** : état physique du composé complémentaire associé au réducteur.
- ▷ **Potentiel** : potentiel standard (à pH=0) en V.

Pour la réaction suivante :



voici l'enregistrement :

```
2, 'Sb', 'Sb205', 0.50, 's', 'Sb203', 0.50, 's', 2.00, 'H(+)', 2.00, NULL, 'H2O', 1.
```

Les composés doivent être présents dans la table `ch_composes`.

2.e. Table solutions

Cette table contient les noms de solutions aqueuses.

```
CREATE TABLE IF NOT EXISTS 'ch_solutions' (
  'Id' smallint(6) NOT NULL AUTO_INCREMENT,
  'Nom' char(30),
  PRIMARY KEY ('Id')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_cs;
```

- ▷ **Id** : identifiant auto incrémenté, qui constitue la clé primaire.
- ▷ **Nom** : nom de la solution.

2.f. Table compositions solutions

Cette table contient la composition des solutions aqueuses :

```
CREATE TABLE IF NOT EXISTS 'ch_compositions_solutions' (
  'Id' smallint(6) NOT NULL AUTO_INCREMENT,
  'Solution' char(30), /*clé table ch_solutions (Nom)*/
  'Compose' char(20), /*clé table ch_composes (Symbole)*/
  'Concentration' float(6,3), /* en mol/L */
  PRIMARY KEY ('Id')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_cs;
```

- ▷ **Id** : identifiant auto incrémenté, qui constitue la clé primaire.
- ▷ **Solution** : clé étrangère sur le champ `ch_solutions.Nom`.
- ▷ **Composé** : composé chimique en solution, clé étrangère sur le champ `ch_composes.Symbole`.
- ▷ **Concentration** : concentration du composé en mol/L.

Voici par exemple les renregistrements pour une solution de chlorure de sodium :

```
1, 'NaCl30', 'Na(+)', 0.510
2, 'NaCl30', 'Cl(-)', 0.510
```

2.g. Tables electrodes

Cette table contient les électrodes utilisées dans les expériences de courbe intensité-potentiel.

```
CREATE TABLE IF NOT EXISTS 'ch_electrodes' (
  'Id' smallint(6) NOT NULL AUTO_INCREMENT,
  'Materiau' char(30),
  'Surface' float(6,3), /* en mm^2 */
  PRIMARY KEY ('Id')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_cs;
```

- ▷ **Id** : identifiant auto incrémenté, qui constitue la clé primaire.
- ▷ **Materiau** : matériau de l'électrode.
- ▷ **Surface** : surface immergée en mm^2 .

Voici un exemple :

```
2, 'Fer', 38.000
```

2.h. Tables experiencesIE

Cette table contient les expériences de courbe intensité-potentiel.

```
CREATE TABLE IF NOT EXISTS 'ch_experiencesIE' (
  'Id' int(6) NOT NULL AUTO_INCREMENT,
  'Solution' char(30), /*clé table ch_solutions (Nom) */
  'Electrode' smallint(6), /* clé table ch_electrodes (Id) */
  'DureePoint' float(4,1), /* en s */
  'Observations' char(200),
  'Auteur' char(30),
  PRIMARY KEY ('Id')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_cs;
```

- ▷ **Id** : identifiant auto incrémenté, qui constitue la clé primaire.
- ▷ **Solution** : solution aqueuse, clé étrangère sur le champ `ch_solutions.Nom`.

- ▷ **Electrode** : électrode, clé étrangère sur le champ `ch_electrodes.Id`.
- ▷ **DureePoint** : durée de chaque mesure en secondes.
- ▷ **Observations** : observations expérimentales ou particularités de l'expérience.
- ▷ **Auteur** : auteur de l'expérience.

Voici un exemple :

```
1, 'NaCl30', 2, 10.0, '', 'F. Legrand'
```

2.i. Table courbesIE

Cette table contient les courbes intensité-potentielle, c'est-à-dire les points expérimentaux des expériences de courbe intensité-potentielle.

```
CREATE TABLE IF NOT EXISTS 'ch_courbesIE' (  
  'Id' int(6) NOT NULL AUTO_INCREMENT,  
  'Experience' int(6), /* clé table ch_experiencesIE (Id) */  
  'Tension' float(4,3), /* en V */  
  'Courant' float(5,1), /* en mA */  
  PRIMARY KEY ('Id')  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_cs;
```

- ▷ **Id** : identifiant auto incrémenté, qui constitue la clé primaire.
- ▷ **Experience** : clé étrangère sur le champ `ch_experiencesIE.Id`.
- ▷ **Tension** : potentiel en V.
- ▷ **Courant** : courant en mA.

Voici par exemple les premiers enregistrements pour la courbe `Id=1`

```
1, 1, -1.478, -40.7  
2, 1, -1.460, -37.6  
3, 1, -1.441, -34.6  
4, 1, -1.426, -32.5  
5, 1, -1.409, -29.5  
6, 1, -1.395, -27.4
```

3. Accès à la base de données

La base de données MySQL est hébergée en local sur le serveur S1PEDA. Son nom est `mysqlmp2`.

L'accès à l'interface phpmyadmin se fait dans un navigateur web à l'adresse `S1PEDA/phpmyadmin` avec son identifiant et mot de passe personnels (le même que celui des sessions windows). Depuis cette interface, on pourra consulter les tables, remplir certaines tables, ou faire des requêtes SQL.

En Python 2.7, l'accès se fait avec le module `MySQLdb`. Documentation du module : [MySQLdb User's Guide](#). Voici comment se fait la connexion à la base :

```
import MySQLdb
db = MySQLdb.connect (host = 'S1PEDA',
                      user = 'mysqlmp2',
                      passwd = 'romesole',
                      db = 'mysqlmp2')
dbc=db.cursor()
```

En salle de TP, on utilisera Python 3.5 et le module `mysql-connector`. Documentation : [MySQL Connector](#). La connexion se fait de la manière suivante :

```
import mysql.connector
db = mysql.connector.connect (host = 'S1PEDA',
                              user = 'mysqlmp2',
                              passwd = 'romesole',
                              db = 'mysqlmp2')
dbc=db.cursor()
```

Voici comment effectuer une requête :

```
dbc.execute("SELECT Symbole,M FROM ch_elements")
```

Pour récupérer les résultats de la requête dans un tableau :

```
result = dbc.fetchall()
```

Les lignes du tableau sont lues dans une boucle :

```
for row in result:
    Symbole = row[0]
    M = row[1]
    ....
```

Pour les requêtes effectuant une écriture (avec `INSERT`), il peut être nécessaire de récupérer la dernière valeur de l'identifiant auto incrémenté (qui est souvent la clé primaire) :

```
Id = dbc.lastrowid
```

Remarque : l'accès avec un script python depuis un client n'est généralement pas possible pour une base de données internet (pour des raisons de sécurité). Dans ce cas, il faudra utiliser un script python CGI tournant sur le serveur, ou un script PHP. Du côté du client, la communication avec ces scripts se fait le plus souvent par des pages HTML contenant du script JAVASCRIPT.

4. Travaux pratiques : requêtes sur des données

4.a. Éléments et composés chimiques

- ▷ Extraire tous les éléments (afficher les symboles) classés par électronégativité croissante, en affichant cette électronégativité.
- ▷ Extraire les éléments d'une période donnée, en affichant leur structure électronique et leur électronégativité.
- ▷ Extraire la liste des éléments chimiques (avec leur nombre) d'un composé chimique.
- ▷ Extraire tous les composés qui contiennent un élément donné.
- ▷ Extraire tous les composés qui contiennent un élément donné, en affichant leur charge et en les classant par charge croissante.
- ▷ Extraire les éléments dont l'électronégativité est inférieure à celle d'un élément donné.
- ▷ Extraire les composés comportant des éléments d'une période donnée.

4.b. Réactions électrochimiques

- ▷ Extraire les couples redox dont le potentiel standard se trouve dans un intervalle donné.
- ▷ Extraire l'oxydant, le réducteur, et le nombre d'électrons échangés, pour toutes les réactions électrochimiques d'une famille donnée.
- ▷ Extraire la liste des éléments chimiques de tous les composés chimiques qui interviennent dans les réactions d'une famille donnée.
- ▷ Extraire tous les oxydants contenant un élément chimique donné, en affichant aussi le réducteur correspondant et le nombre d'électrons échangés.

4.c. Traitements de données

Voici des traitements plus poussés, à réaliser en Python 3.5 :

- ▷ Tracer l'énergie d'ionisation des éléments en fonction du nombre de charges Z .
- ▷ Écrire les équations des réactions électrochimiques pour une famille donnée, en affichant pour chacune le potentiel standard.
- ▷ Écrire un script qui vérifie l'équilibre des éléments chimiques et des charges pour toutes les réactions électrochimiques.

5. Courbes intensité-potentiel

5.a. Insertion des expériences intensité-potentiel

Les données (potentiel et courant) d'une expérience doivent être dans un fichier texte, soit disposées en colonne, soit disposées en ligne. Lorsqu'on remplit le fichier texte à la main (utiliser Notepad), il est plus commode de les entrer en colonne, sous le format suivant :

```
E1 I1  
E2 I2
```


E3 I3

Sur une ligne, les données sont séparées par une espace ou une tabulation. Le potentiel doit être en Volts, l'intensité du courant doit être en mA.

Avant d'insérer les données d'une expérience, il faut consulter la table `ch_electrodes` pour connaître l'identifiant numérique de l'électrode utilisée et la table `ch_solutions` pour connaître le nom de la solution utilisée.

Le script python suivant permet d'insérer toutes les données d'une expérience :

[entrerExperienceIE.py](#) Ces expériences seront réalisées en TP de chimie puis insérées à l'issu du TP.

5.b. Tracé des courbes intensité-potentiel

Le script Python suivant permet d'extraire les données d'expériences afin de tracer des courbes intensité-potentiel.

[tracerCourbeIE.py](#) Exercice : compléter le script pour qu'il trace sur la même figure toutes les courbes intensité-potentiel correspondant à une électrode et une solution données.