جامعة مولاي إسماعيل
UNIVERSITÉ MOULAY ISMAÏL
كلية العلوم والتقنيات
FACULTÉ DES SCIENCES ET TECHNIQUES

Module : THL et Compilation (I513)
Filière : LST GL S5
Professeur : N. Mouhni
Année universitaire : 2025-2026

Interpréteur des requêtes du langage SQL simplifié

A rendre avant le : Mercredi 26 Novembre 2025

Travail : En monôme, binôme ou trinôme

1 Contexte et Objectifs

1.1 Présentation du Projet

Objectif

Vous devez développer un **interpréteur de requêtes SQL simplifiées en langage C** utilisant **Flex** (analyseur lexical) et **Bison** (analyseur syntaxique). L'interpréteur doit traiter des commandes de manipulation et d'interrogation de bases de données relationnelles.

Important

Note importante : Vous ne développez PAS un vrai SGBD qui exécute les requêtes. Votre programme doit **analyser, vérifier et afficher des statistiques** sur les requêtes, mais ne stocke pas réellement les données des tables.

1.2 Objectifs Pédagogiques

Ce projet vous permettra de :

✓ Maîtriser l'utilisation de **Flex et Bison**
✓ Comprendre l'analyse lexicale et syntaxique dans un contexte pratique
✓ Implémenter des **actions sémantiques**
✓ Gérer une **table des symboles**
✓ Détecter et signaler les erreurs de manière claire

2 Description du Langage : GLSimpleSQL

2.1 Commandes Supportées

Votre interpréteur doit reconnaître et analyser les commandes SQL suivantes :

2.1.1 CREATE TABLE

Crée une nouvelle table avec ses champs et types.

```
CREATE TABLE nom_table (
    champ1 type1,
    champ2 type2,
    ...
);
```

Listing 1 – Syntaxe CREATE TABLE

Exemple :

```
CREATE TABLE Client (
    numClt INT,
    nom VARCHAR(50),
    prenom VARCHAR(50),
    age INT
);
```

2.1.2 INSERT INTO

Insère des données dans une table.

```
INSERT INTO nom_table VALUES (valeur1, valeur2, ...);
INSERT INTO nom_table (champ1, champ2) VALUES (valeur1, valeur2);
```

Listing 2 – Syntaxe INSERT INTO

Exemples :

```
INSERT INTO Client VALUES (1, 'Dupont', 'Jean', 35);
INSERT INTO Client (numClt, nom) VALUES (2, 'Martin');
```

2.1.3 SELECT

Interroge les données d'une table.

```
SELECT * FROM nom_table;
SELECT champ1, champ2 FROM nom_table;
SELECT * FROM nom_table WHERE condition;
SELECT champ1, champ2 FROM nom_table WHERE condition;
```

Listing 3 – Syntaxe SELECT

Exemples :

```
SELECT * FROM Client;
SELECT nom, prenom FROM Client WHERE numClt = 2;
SELECT * FROM Client WHERE age > 30 AND nom = 'Dupont';
```

2.1.4 UPDATE

Modifie des données existantes.

```
UPDATE nom_table SET champ1 = valeur WHERE condition;
UPDATE nom_table SET champ1 = valeur1, champ2 = valeur2 WHERE
   condition;
```

Listing 4 – Syntaxe UPDATE

Exemples :

```
UPDATE Client SET age = 36 WHERE numClt = 1;
UPDATE Client SET nom = 'Durand', age = 40 WHERE numClt = 2;
```

2.1.5 DELETE

Supprime des données d'une table.

```
DELETE FROM nom_table WHERE condition;
DELETE FROM nom_table;
```

Listing 5 – Syntaxe DELETE

Exemples :

```
DELETE FROM Client WHERE age < 18;
DELETE FROM Client;
```

2.1.6 DROP TABLE

Supprime une table complète.

```
DROP TABLE nom_table;
```

Listing 6 – Syntaxe DROP TABLE

Exemple :

```
DROP TABLE Client;
```

2.2 Types de Données Supportés

TABLE 1 – Types de données GLSimpleSQL

Type	Description
`INT`	Entiers
`FLOAT`	Nombres réels
`VARCHAR(n)`	Chaînes de caractères de longueur maximale n
`BOOL`	Booléens (TRUE/FALSE)

2.3 Opérateurs et Conditions

— **Opérateurs de comparaison :** `=`, `!=`, `<`, `>`, `<=`, `>=`

— **Opérateurs logiques :** `AND`, `OR`, `NOT`

3 Travail Demandé

3.1 Phase 1 : Analyse Lexicale (Flex)

3.1.1 Spécifications Requises

1. Reconnaissance des tokens :

— **Mots-clés SQL :** SELECT, FROM, WHERE, INSERT, CREATE, UPDATE, DELETE, DROP, TABLE, INTO, VALUES, SET, AND, OR, NOT

— **Types de données :** INT, FLOAT, VARCHAR, BOOL

— **Opérateurs :** =, !=, <, >, <=, >=

— **Identificateurs :** noms de tables et de champs (lettres, chiffres, underscore)

— **Constantes :**

  — Entières : 123, -45

  — Réelles : 3.14, -0.5

  — Chaînes : 'texte', "texte"

  — Booléennes : TRUE, FALSE

— **Symboles :** (,), „ ;, *

2. Gestion des éléments à ignorer :

— Commentaires SQL :

  — Commentaires sur une ligne : `- commentaire`

  — Commentaires multi-lignes : `/* commentaire */`

3. Gestion des erreurs lexicales :

— Caractères invalides

— Chaînes non terminées

3.2 Phase 2 : Analyse Syntaxique (Bison)

3.2.1 Spécifications Requises

1. Grammaire formelle :

Vous devez écrire la grammaire BNF/EBNF complète de GLSimpleSQL. Exemple de début de grammaire :

Extrait de grammaire BNF

```
<requete> ::= <create_table>
            | <insert_into>
            | <select>
            | <update>
            | <delete>
            | <drop_table>

<select> ::= SELECT <liste_champs> FROM <nom_table>
           | SELECT <liste_champs> FROM <nom_table> WHERE <condition>

<liste_champs> ::= *
                 | <nom_champ>
                 | <nom_champ> , <liste_champs>
```

2. Règles de production Bison :

— Règles pour chaque type de requêtes

— Construction d'une structure de données (AST ou autre) représentant la requête

3. Gestion des erreurs syntaxiques

3.3 Phase 3 : Actions Sémantiques

3.3.1 A. Statistiques sur les Requêtes

Votre programme doit afficher des statistiques détaillées :
Pour les requêtes SELECT :

— Nom de la table interrogée

— Nombre de champs sélectionnés (liste des noms)

— Présence d'une clause WHERE (OUI/NON)

— Nombre de conditions dans le WHERE

— Nombre d'opérateurs logiques utilisés (AND, OR)

Pour les requêtes INSERT :

— Nom de la table

— Nombre de valeurs à insérer

— Vérification de la correspondance nombre de champs/valeurs

Pour les requêtes UPDATE :

— Nom de la table

— Nombre de champs à modifier

— Présence d'une clause WHERE

Exemple d'affichage attendu :

```
Requête SELECT analysée :
  - Table : Client
  - Nombre de champs : 2 (nom, prenom)
  - Clause WHERE : OUI
  - Nombre de conditions : 1
  - Opérateurs logiques : 0
```

3.3.2 B. Table des Symboles

Implémentez une table des symboles pour stocker :

— Les tables créées avec leurs champs et types

— Pour chaque table : nom, liste des champs, types associés

3.3.3 C. Vérifications Sémantiques

Vérifications obligatoires :

1. Table inexistante dans une requête SELECT, INSERT, UPDATE, DELETE
2. Champ inexistant dans une table
3. Incohérence entre nombre de champs et nombre de valeurs (INSERT)
4. Tentative de créer une table déjà existante
5. Tentative de supprimer une table inexistante (DROP)
6. Utilisation invalide de * : `SELECT *, nom FROM table`

3.3.4 D. Messages d'Erreur

Les messages doivent être clairs et précis :

Exemples de messages attendus

```
ERREUR SÉMANTIQUE ligne 5 :
    La table 'Produit' n'existe pas.

ERREUR SÉMANTIQUE ligne 8 :
    Le champ 'prix' n'existe pas dans la table 'Client'.

ERREUR SÉMANTIQUE ligne 12 :
    INSERT INTO Client : 3 valeurs fournies mais 4 champs attendus.
```

4 Exemples de Tests Requis

4.1 Tests de Base (Obligatoires)

```
1 -- Test 1 : Cr ation et insertion
2 CREATE TABLE Etudiant (
3     id INT,
```

```
    nom VARCHAR(50),
    age INT
);

INSERT INTO Etudiant VALUES (1, 'Diallo', 20);
INSERT INTO Etudiant (id, nom) VALUES (2, 'Sow');
```

Listing 7 – Test 1 : Création et insertion

```
-- Test 2 : S lections vari es
SELECT * FROM Etudiant;
SELECT nom FROM Etudiant;
SELECT nom, age FROM Etudiant WHERE age > 18;
SELECT * FROM Etudiant WHERE id = 1 AND age < 25;
```

Listing 8 – Test 2 : Sélections variées

```
-- Test 3 : Modifications
UPDATE Etudiant SET age = 21 WHERE id = 1;
UPDATE Etudiant SET nom = 'Ba', age = 22 WHERE id = 2;
```

Listing 9 – Test 3 : Modifications

```
-- Test 4 : Suppressions
DELETE FROM Etudiant WHERE age < 18;
DROP TABLE Etudiant;
```

Listing 10 – Test 4 : Suppressions

4.2 Tests d'Erreurs (Obligatoires)

```
-- Erreur : table inexistante
SELECT * FROM Produit;

-- Erreur : champ inexistant
SELECT prix FROM Etudiant;

-- Erreur : nombre de valeurs incorrect
INSERT INTO Etudiant VALUES (3, 'Kane');

-- Erreur : table d j  existante
CREATE TABLE Etudiant (id INT);
CREATE TABLE Etudiant (num INT);

-- Erreur : syntaxe invalide
SELECT FROM Etudiant;
SELECT * Etudiant;
```

Listing 11 – Tests détection d'erreurs

5 Livrables

Chaque membre de l'équipe doit déposer une copie sur Google Classroom des livrables suivants :

1. Un dossier zippé contenant votre code, avec un fichier readMe.txt décrivant la structure de votre projet
2. Une Vidéo montrant l'exécution et le test (ne dépassant pas 5min)
3. La grammaire formelle complète du langage GLSimpleSQL dans un fichier "Grammaire.pdf"
4. Un document détaillant votre travail "Rapport.pdf"

6 Critères d'Évaluation

6.1 Détail des Critères

TABLE 2 – Critères d'évaluation

Critère	**Détails**
Analyseur lexical	Reconnaissance correcte des tokens, gestion des erreurs
Analyseur syntaxique	Grammaire complète, gestion des erreurs
Actions sémantiques	Compteurs, vérifications, messages d'erreur
Tests	Qualité et exhaustivité des tests
Documentation (Rapport)	Clarté, complétude, qualité rédactionnelle
Présentation	Présentation vidéo enregistrant la simulation
Réponses aux questions	Réponses aux questions le jour de l'examen final

6.2 Pénalités

— Retard (par jour) : -1 point

— Plagiat : **0/20 pour tous les membres du groupe**

7 Ressources Utiles

7.1 Documentation

— Manuel Flex : `https://westes.github.io/flex/manual/`

— Manuel Bison : `https://www.gnu.org/software/bison/manual/`

7.2 Outils

— Compilateur GCC

— Flex et Bison

— un IDE , e.g Visual Studio code