# Proposition de Projet - MTI820

## FR2SQL

Génération de requêtes SQL à partir d'un langage naturel en français

**WILHELMY Felix**
Département de génie
LOG & TI
ÉTS, Montréal, Canada
WILF15099506

**NOUBISSI KOM Carmen Wilfred**
Département de génie
LOG & TI
ÉTS, Montréal, Canada
NOUC20329101

**LAAZIRI Othman**
Département de génie
LOG & TI
ÉTS, Montréal, Canada
LAAO82010107

# 1 Problématique

Les organisations, afin de permettre aux utilisateurs non-experts en base de données d'interagir de manière intuitive avec celle-ci, s'appuient de plus en plus sur des interfaces conversationnelles. Le succès de ces systèmes dépend fortement de la qualité du composant NL2SQL (Natural Language to SQL). Malgré les progrès récents, la littérature reste dominée par des approches anglophones et peu adaptées au français ainsi qu'aux environnements BI. Ce travail consiste à développer un système capable de comprendre et de traduire efficacement une requête formulée en langage naturel en une requête SQL précise et exécutable, tout en gérant la diversité des expressions, des ambiguïtés et des contextes afin de rendre l'accès aux données plus accessible et plus fluide pour tous les utilisateurs et particulièrement ceux qui parlent français.

**Objectif** Notre projet vise donc à concevoir un outil francophone qui

- facilite l'accès aux données analytiques pour des utilisateurs non techniques ;
- accélère l'extraction d'informations pertinentes à partir de requêtes en langage naturel ;
- maximise la valeur tirée des données disponibles au sein de l'organisation.

# 2 Méthodologie

## 2.1 Architecture générale

L'architecture du système repose sur un pipeline modulaire implémenté en **Python** à l'aide de la bibliothèque `PyTorch Lightning`, qui facilite la structuration et la reproductibilité des boucles d'apprentissage. Le modèle de base est **LLaMA 3 Instruct-8B**, un grand modèle de langage (LLM)

open-source de 8 milliards de paramètres pré-entraîné pour suivre des instructions. Afin de rendre l'entraînement réalisable sur les ressources à notre disposition, le modèle sera quantifié (*quantized*) en **4 bits** à l'aide de la bibliothèque `BitsAndBytes`, puis *fine-tune* selon la méthode **QLoRA** (Quantized Low-Rank Adapter) [1].

## 2.2 Préparation des données (Semaine 7 & 8)

Le jeu de données pour l'entrainement sera SPIDER-FR, une version traduite en français de Spider [2], qui contient des paires (question naturelle, requête SQL) sur des schémas de bases relationnelles complexes.

Chaque entrée de notre système aura la structure suivante :

```
{
  "question_fr": "Quelle est la moyenne des salaires des employés ?",
  "sql": "SELECT AVG(salary) FROM employees",
  "db_id": "employee_db"
}
```

Un module de chargement dédié aligne ces questions à leur schéma relationnel (fichier SQLite) pour l'encodage.

## 2.3 Entraînement du modèle (Semaine 9 & 10)

L'objectif est d'apprendre au modèle à générer une requête SQL correcte à partir d'une question formulée en français, en tenant compte du schéma de la base de données concernée. Pour cela, nous utilisons la méthode `QLoRA` (Quantized Low-Rank Adapter), une approche d'affinage efficace qui n'ajuste qu'un sous-ensemble restreint des paramètres du modèle pré-entraîné. Cette méthode combine :

— une quantification **4 bits** via la bibliothèque `BitsAndBytes`, permettant de réduire considérablement l'utilisation mémoire tout en préservant la performance ;

— un entraînement structuré à l'aide de `PyTorch Lightning`, facilitant la gestion des boucles d'apprentissage, du suivi expérimental et de la reproductibilité.

## 2.4 Évaluation et métriques (Semaine 11)

La qualité du modèle est évaluée avec quatre métriques principales, standard dans la littérature NL2SQL :

— **Execution Accuracy** : pourcentage de requêtes générées produisant le même résultat que la requête cible sur la base ;
— **Exact Match** : taux de correspondance exacte entre la requête prédite et celle attendue (au niveau chaîne de caractères) ;
— **Valid SQL Rate** : proportion de requêtes valides (syntaxiquement exécutables) ;
— **VES** (Valid Efficiency Score) : indicateur composite qui pénalise les erreurs syntaxiques, les lenteurs et les requêtes inefficaces.

### 2.4.1 Outils d'evaluations

L'évaluation sera probablement automatisée via script proposé par [2] :

`https://github.com/taoyds/test-suite-sql-eval`

Nous utiliserons également PICARD (Parsing Incrementally-Constrained Autoregressive Decoder) pour contraindre la génération au sein de la grammaire SQL, garantissant que les requêtes produites sont valides dès la phase de décodage.

## 2.5 Infrastructure de déploiement et d'entraînement

Les expérimentations sont réalisées sur deux plateformes :

— **Google Colab Pro+** : pour les itérations rapides, l'entraînement LoRA sur GPU T4/A100 ;

— **Postes personnels** : pour la préparation des données, la validation, l'export de modèles, et les tests légers.

# 3 Calendrier de Planification

Table 1 présente les jalons prévus pour la session.

TABLE 1 – Calendrier Prévisionnel du Projet

| **Semaine** | **Tâches** |
|---|---|
| 6 | Remise de la proposition de projet [Tous]<br>Configuration de l'environment de developpement [Tous]<br>Configuration du repositoire github pour le projet [Felix] |
| 7 & 8 | Implémentation d'un lien schéma-question (extraire les mots clee des demande language naturel) [Felix]<br>Extraction et introspection des schémas (SQLite) (avec les liens, extraire les schema des tables de la base de donnees pertinant a la requete) [Othman]<br>Générateur de contexte pour LLM (a partir des schema de table et de la requete de l'utilisateur) [Wilfred] |
| 9 & 10 | Mise en place du pipeline de fine-tuning QLoRA pour notre model [Wilfred]<br>Script de logging + tracking [Othman]<br>Chargement LLaMA-3 Instruct-8B (4-bit) et fine-tuning sur Google Colab [Felix] |
| 11 | Intégration de PICARD dans le pipeline [Wilfred]<br>Integration des batteries de tests et d'evaluation du model [Felix]<br>Debut de redaction du rapport [Othman] |
| 12 | Analyse & visualisation des resultats [Felix]<br>Rédaction du rapport final (méthodologie, figures, tableaux de métriques) [Tous]<br>Création des slides et preparation pour l'oral [Wilfred & Othman] |

# 4 Structure Prévisionnelle du Rapport Final

1. Résumé
2. Introduction
3. Problématique
4. Objectifs
5. Analyse des besoins
6. Modèle dimensionnel et sources de données
   (a) Modèle dimensionnel adapté
   (b) Description de Spider-FR et des données internes Forester
7. Architecture de la solution
   (a) Plan d'architecture haut niveau
   (b) Description des technologies employées (PyTorch Lightning, QLoRA, PICARD, etc.)
8. Méthodologie détaillée
   (a) Pipeline modulaire et prétraitement
   (b) Entraînement et configuration (hyperparamètres, quantification)
   (c) Évaluation et métriques (Execution Accuracy, Exact Match, Valid SQL Rate, VES)
9. Résultats et analyse
   (a) Résultats globaux
   (b) Analyse par schéma et tests multilingues
   (c) Impact des stratégies de post-traitement (PICARD)
10. Discussion
    (a) Points forts et points faibles de la preuve de concept
    (b) Limites et améliorations possibles
    (c) Perspectives d'intégration en environnement BI réel
11. Conclusion

# Références


[1] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora : Efficient finetuning of quantized llms," 2023. [Online]. Available : https://arxiv.org/abs/2305.14314

[2] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Roman, Z. Zhang, and D. Radev, "Spider : A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task," 2019. [Online]. Available : https://arxiv.org/abs/1809.08887

[3] N. Busany, E. Hadar, H. Hadad, G. Rosenblum, Z. Maszlanka, O. Akhigbe, and D. Amyot, "Automating business intelligence requirements with generative ai and semantic search," 2024. [Online]. Available : https://arxiv.org/abs/2412.07668

[4] J. Jiang, H. Xie, Y. Shen, Z. Zhang, M. Lei, Y. Zheng, Y. Fang, C. Li, D. Huang, W. Zhang, Y. Li, X. Yang, B. Cui, and P. Chen, "Siriusbi : Building end-to-end business intelligence enhanced by large language models," 2024. [Online]. Available : https://arxiv.org/abs/2411.06102

[5] X. Liu, S. Shen, B. Li, P. Ma, R. Jiang, Y. Zhang, J. Fan, G. Li, N. Tang, and Y. Luo, "A survey of nl2sql with large language models : Where are we, and where are we going ?" 2025. [Online]. Available : https://arxiv.org/abs/2408.05109

[6] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models : A survey," 2025. [Online]. Available : https://arxiv.org/abs/2402.06196

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available : https://arxiv.org/abs/1706.03762