

# IFT2935: Base de données

## Données

Avant de démarrer l'application, il faut avoir préalablement créer les tables et insérer les données à partir de `sqlplus`:

```
@sql/final/LDD.sql
@sql/final/LMD.sql
COMMIT; -- applique les changements
```

## Installation

Le projet est développé sur Oracle Database 11g, alors il vous faudra satisfaire quelques dépendances:

- [Oracle Instant Client](#)
- [cx\\_Oracle](#)
  - Python 2.7 (Python 3+ ne fonctionne pas)

Les bases de données Oracle du DIRO roulent sous la version 10.2.0.1, alors vous pouvez installer la plus récente des 10.2 (10.2.0.5 à ma dernière consultation).

Il faut s'enregistrer pour télécharger le client et le SDK, Oracle fournit les archives zip suivantes:

- `basic` version 10.2.\*
- `sdk` version 10.2.\*

Il faut extraire les archives dans le même répertoire `instantclient_10_2` et exporter la variable d'environnement `ORACLE_HOME` vers ce répertoire.

```
export ORACLE_HOME=$(pwd)/instantclient_10_2
export LD_LIBRARY_PATH=$(pwd)/instantclient_10_2
```

Vous devez créer un lien symbolique pour la librairie `libclntsh.so`.

```
ln -s libclntsh.so.10.1 libclntsh.so
```

Une fois `ORACLE_HOME` et `LD_LIBRARY_PATH` exportés, `cx_Oracle`, `PyYAML` et `flask` peuvent s'installer facilement à partir de `pip`:

```
pip install --user cx_Oracle PyYAML flask
```

Pour paramétrer l'exécution, ajoutez un fichier de configuration YAML config.yml:

```
database:
  dsn:
    host: delphes.iro.umontreal.ca
    port: 1521
    sid: a05
  user:           # mettre votre usager au DIRO
  password:       # mot de passe (voir les notes)
SECRET_KEY: "secrey key" # clé secrète pour la session
```

L'indentation doit être respectée et les entrées **user** et **password** correspondent aux accès pour Oracle.

Ensuite, vous pouvez démarrer l'application:

```
python app.py
```