

Compte rendu de projet - Advanced DataBase Management

Par : Théo BIGAND, Roméo AGOSTINO & Mathieu AUDIBERT | **CSDS1**

Tables des matières

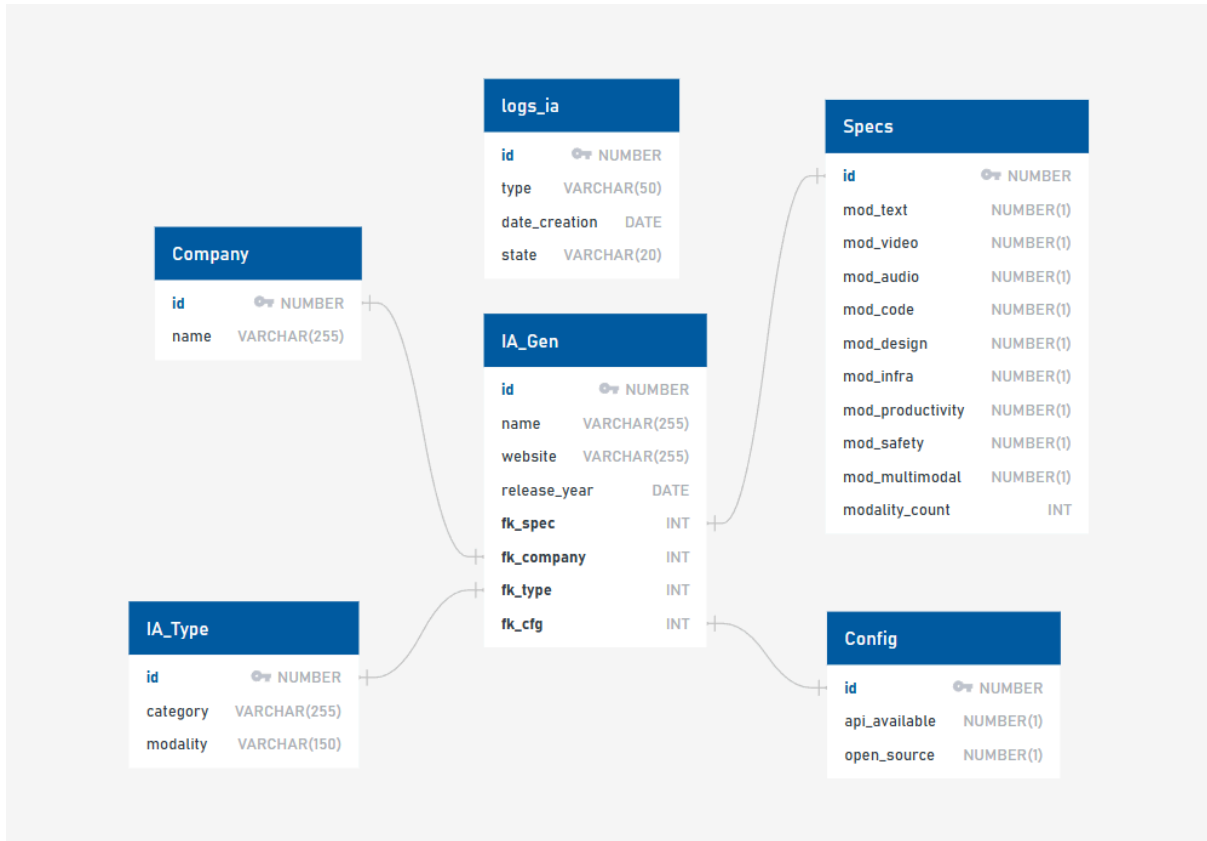
- 1. Liens utiles 2
- 2. Contexte 2
- 3. Schémas relationnels 3
- 4. Sécurité & gestion des utilisateurs 4
- 5. Requêtes & optimisation 5
- 6. Interface graphiques (bonus) 6

1. Liens utiles

Dataset : kaggle.com/datasets/wardabilal/global-generative-ai-tools-landscape-2025
 Dataset au format CSV : [ADVDB](#)
 Repository : <https://github.com/MathieuAudibert/advdb>
 Pour générer nos mots de passe : <https://generateurdemotdepasse.fr/>
 Support de présentation: [ADV-DB / Presentation](#)

2. Schémas relationnels

Nous sommes partis sur un schéma relationnel simple (similaire à un **MPD**) :



Et pour les contraintes, nous avons :

Des NOT NULL sur IA_Gen.name, logs_ia.type, logs_ia.date_creation, [Company.name](#) et enfin IA_Type.category.

Des foreign Keys dans la table IA_Gen : fk_specs, fk_company, fk_iatype, fk_cfg

Des Primary key sur tous les ID de chaque table

3. Contexte

Afin que nous puissions tous travailler en parallèle sur une même base de données, nous avons utilisé un **serveur personnel** (utilisant la distrib Ubuntu Server) sur lequel nous avons créé nos users (theo, mathieu & romeo). Nous nous connectons à ce serveur à l'aide d'ssh (pour des raisons de sécurité, nous cacherons l'ip) :

```
Mathieu@DESKTOP-KK11NT3 MSYS ~
$ ssh -p 2222 mathieu@ip
```

Pour plus de contexte, nous avons d'abord partitionné notre disque dur externe sur lequel nous avons créé un **point de montage** (répertoire) /data/engines/oracle que nous avons rendu permanent grâce à fstab :

```
prodigieux@eteris-cloud-provider:~$ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.39.3).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): p
Disk /dev/sdb: 931,51 GiB, 1000204886016 bytes, 1953525168 sectors
Disk model: Transcend
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x03149842

Device Boot Start End Sectors Size Id Type
/dev/sdb1 * 2048 1953521663 1953519616 931,5G 7 HPFS/NTFS/exFAT

Command (m for help): d
Selected partition 1
Partition 1 has been deleted.

Command (m for help): 1
1: unknown command

Command (m for help): p

Disk /dev/sdb: 931,51 GiB, 1000204886016 bytes, 1953525168 sectors
Disk model: Transcend
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x03149842

Command (m for help): n
Partition type
  p primary (0 primary, 0 extended, 4 free)
  e extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-1953525167, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-1953525167, default 1953525167): +200G
Created a new partition 1 of type 'Linux' and of size 200 GiB.
Partition #1 contains a ntfs signature.

Do you want to remove the signature? [Y]es/[N]o: █
```

```
prodigieux@eteris-cloud-provider:~$ sudo mkfs.ext4 /dev/sdb1
mke2fs 1.47.0 (5-Feb-2023)
/dev/sdb1 contains a ntfs file system labelled 'Transcend'
Proceed anyway? (y,N) y
Creating filesystem with 52428800 4k blocks and 13107200 inodes
Filesystem UUID: 2cce6683-5075-49be-bbe3-f1448e7e511a
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: done

prodigieux@eteris-cloud-provider:~$ sudo mkdir -p /data

prodigieux@eteris-cloud-provider:~$ sudo mount /dev/sdb1 /data

prodigieux@eteris-cloud-provider:~$
```

```
prodigieux@eteris-cloud-provider:~$ sudo nano /etc/fstab
```

```
GNU nano 7.2
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/e46cb45f-5107-4240-b4d5-4d6ea2de2119 / ext4 defaults 0 1
# /boot/efi was on /dev/sda1 during curtin installation
/dev/disk/by-uuid/1255-8FFD /boot/efi vfat defaults 0 1
/swap.img none swap sw 0 0
UUID=2cce6683-5075-49be-bbe3-f1448e7e511a /data ext4 defaults 0 2
```

Depuis ce point de montage, nous avons créé un répertoire
/data/projects/advdb sur lequel nous avons donné les **droits** à notre user
oracle :

```
prodigieux@eteris-cloud-provider:~$ sudo mkdir -p /data/engines/oracle

prodigieux@eteris-cloud-provider:~$ sudo mkdir -p /data/projects/advdb/db/oracle

prodigieux@eteris-cloud-provider:~$ sudo chown -R oracle:oinstall /data/engines/oracle

prodigieux@eteris-cloud-provider:~$ sudo chown -R oracle:oinstall /data/projects/advdb/db/oracle
```

Linux server a besoin de **plusieurs noyaux spécifiques** pour faire tourner oracle, les voici :

```
prodigieux@eteris-cloud-provider:~$ sudo apt update
Réception de :1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Atteint :2 http://archive.ubuntu.com/ubuntu noble InRelease
Réception de :3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Réception de :4 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Réception de :5 https://download.docker.com/linux/ubuntu noble InRelease [48,5 kB]
Réception de :6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1 378 kB]
Réception de :7 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [224 kB]
Réception de :8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21,6 kB]
Réception de :9 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [9 488 B]
Réception de :10 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2 266 kB]
Réception de :11 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1 675 kB]
Réception de :12 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [518 kB]
Réception de :13 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Réception de :14 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [916 kB]
Réception de :15 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [207 kB]
Réception de :16 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [71,5 kB]
Réception de :17 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [19,4 kB]
Réception de :18 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Réception de :19 http://archive.ubuntu.com/ubuntu noble-updates/main Translation-en [309 kB]
Réception de :20 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Réception de :21 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15,8 kB]
Réception de :22 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2 404 kB]
Réception de :23 http://archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [549 kB]
Réception de :24 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Réception de :25 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1 501 kB]
Réception de :26 http://archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [304 kB]
Réception de :27 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [378 kB]
Réception de :28 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [31,4 kB]
Réception de :29 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Réception de :30 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7 144 B]
Réception de :31 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Réception de :32 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [29,2 kB]
Réception de :33 http://archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [17,6 kB]
Réception de :34 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [11,0 kB]
Réception de :35 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Réception de :36 https://download.docker.com/linux/ubuntu noble/stable amd64 Packages [40,2 kB]
13,5 Mo réceptionnés en 3s (5 091 ko/s)

Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
48 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.
```

[illegible]

Ensuite nous avons **installé l'archive** OracleDB Linux sur notre machine locale que nous avons **transféré sur le serveur** grâce à scp (équivalent de SshFileTransferProtocol SFTP) :

```
Mathieu@DESKTOP-KK11NT3 MSYS ~
$ scp chemin/local/LINUX.X64_193000_db_home.zip oracle@ip:/data/engines/oracle
```

Il suffit ensuite de dézipper l'archive.

Une fois cela fait, à l'aide de docker on installe Oracle et nous avons **fini la configuration d'Oracle** sur un serveur utilisant Ubuntu Server !

```
prodigieux@eteris-cloud-provider:/tmp/docker-images/OracleDatabase/SingleInstance/dockerfiles$ sudo ./buildContainerImage.sh -v 19.3.0 -e
/var/run/cdi
Swarm: inactive
Runtimes: io.containerd.runc.v2 runc
Default Runtime: runc
Init Binary: docker-init
containerd version: b98a3aace656320842a23f4a392a33f46af97866
runc version: v1.3.0-0-g4ca628d1
init version: de40ad0
Security Options:
  apparmor
  seccomp
    Profile: builtin
  cgroupns
Kernel Version: 6.8.0-87-generic
Operating System: Ubuntu 24.04.3 LTS
OSType: linux
Architecture: x86_64
CPUs: 4
Total Memory: 7.678GiB
Name: eteris-cloud-provider
ID: 501ad0c2-0e1b-43c7-8c2a-a3f2a89266a7
Docker Root Dir: /var/lib/docker
Debug Mode: false
Experimental: false
Insecure Registries:
  ::1/28
  127.0.0.0/8
Live Restore Enabled: false

=====
Building image 'oracle/database:19.3.0-ee' ...
[+] Building 10.8s (4/14)
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 5.12kB
=> WARN: FromAsCasing: 'as' and 'FROM' keywords' casing do not match (line 25)
=> [internal] load metadata for docker.io/library/oraclelinux:7-slim
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [base 1/4] FROM docker.io/library/oraclelinux:7-slim@sha256:1add6ed8602ea996528110fe75f4b03c2ca7ffdb9497148dbb46c3cc9ce6acd
=> => resolve docker.io/library/oraclelinux:7-slim@sha256:1add6ed8602ea996528110fe75f4b03c2ca7ffdb9497148dbb46c3cc9ce6acd
=> => sha256:05211a081d592eb56daf16b0651eb55b233a65466492b28345804309b45caf21 585B / 585B
=> => sha256:38299ae2e819f6ec4befee45312f809ca241167aa554dd79e0a4baf505b88d21 50.49MB / 50.49MB
=> => sha256:1add6ed8602ea996528110fe75f4b03c2ca7ffdb9497148dbb46c3cc9ce6acd 2.38kB / 2.38kB
=> => sha256:645371a66c08ed18b7aacf217c964f0db70c26ac87f3d8816d7ae24160f178e4 1.03kB / 1.03kB
=> => extracting sha256:38299ae2e819f6ec4befee45312f809ca241167aa554dd79e0a4baf505b88d21
=> [internal] load build context
=> => transferring context: 792.04MB
```

```
prodigieux@eteris-cloud-provider:/tmp/docker-images/OracleDatabase/SingleInstance/dockerfiles$ sudo docker run --name oracle-advdb \
-e ORACLE_PWD="oracleadvdb" \
-e ORACLE_EDITION="EE" \
-e ORACLE_SID="ADVDB" \
-p 1521:1521 \
-v /data/projects/advdb/db/oracle:/opt/oracle/oradata \
-d oracle/database:19.3.0-ee
26b972f6a8cd1c6febfa9bccaa0a161a03f293c104e4217ae5c3dde4c4dbb82e
```

```
SQL> SQL> Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
The Oracle base remains unchanged with value /opt/oracle
The Oracle base remains unchanged with value /opt/oracle
#####
DATABASE IS READY TO USE!
#####
The following output is now a tail of the alert.log:
2025-12-11T21:11:18.211121+00:00
ALTER SYSTEM SET local_listener='' SCOPE=BOTH;
ALTER PLUGGABLE DATABASE ORCLPDB1 SAVE STATE
Completed: ALTER PLUGGABLE DATABASE ORCLPDB1 SAVE STATE
XDB initialized.
2025-12-11T21:11:23.848296+00:00
TABLE SYS.WRP$_REPORTS: ADDED INTERVAL PARTITION SYS_P201 (5824) VALUES LESS THAN (TO_DATE(' 2025-12-12 01:00:00', 'YYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
TABLE SYS.WRP$_DETAILS: ADDED INTERVAL PARTITION SYS_P202 (5824) VALUES LESS THAN (TO_DATE(' 2025-12-12 01:00:00', 'YYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
TABLE SYS.WRP$_REPORTS_TIME_BANDS: ADDED INTERVAL PARTITION SYS_P205 (5823) VALUES LESS THAN (TO_DATE(' 2025-12-11 01:00:00', 'YYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
```

```

bash-4.2$ sqlplus sys/oracleadvdb@ADVDB as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Dec 11 21:16:43 2025
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> show pdbs

      CON_ID  CON_NAME                                OPEN MODE  RESTRICTED
-----
          2  PDB$SEED                                READ ONLY    NO
          3  ORCLPDB1                                READ WRITE   NO
SQL>

```

4. Sécurité et gestion des utilisateurs

En termes d'utilisateurs nous avons de base :

- sys
- oracle (utilisé lors de l'installation d'Oracle)

Nous avons ajoutés les utilisateurs suivants sur ORCLPDB1 :

- admin_orclpdb1
- u_rw_orclpdb1
- u_r_orclpdb1

Nous avons notés les utilisateurs de la façon suivante :

- <u> : pour user
- <-rw/-r> : pour les droits de l'utilisateur
- <_orclpdb1> : pour la base de données auquel il a les droits

```

SQL> CREATE USER admin_orclpdb1 IDENTIFIED BY "2F:@?RipG7E)49vp";

User created.

SQL>

```

```

SQL> CREATE USER u_rw_orclpdb1 IDENTIFIED BY "n4g$~8xnLF86C~G:";

User created.

```

```
SQL> CREATE USER u_r_orclpdb1 IDENTIFIED BY ",-pe,z2P!D9R4m6Z";  
User created.
```

Afin de respecter les **normes de sécurité** nous leurs avons ajoutés ces profils :

```
SQL> CREATE PROFILE ur_profile LIMIT connect_time 5 failed_login_attempts 3;  
Profile created.
```

```
SQL> █
```

```
SQL> CREATE PROFILE urw_profile LIMIT connect_time 5 failed_login_attempts 5 password_life_time 31;  
Profile created.
```

```
SQL> █
```

Les 2 profils ont les limites suivantes :

- 5 minutes de connexion maximum
- 3 et 5 tentatives de connexion
- Le mot de passe de l'utilisateur doit être changé tous les 31 jours.

Et chaque user a des **privilèges associés** :

- Pour admin, les droits d'administration sur la bdd
- Pour u-rw les droits de lecture/écriture
- Pour u-r les droits de lecture

```
SQL> GRANT ALL PRIVILEGES TO admin_orclpdb1;  
  
Grant succeeded.
```

Afin de garder ce rendu cohérent, la partie ou nous créons les tables se trouve dans la section d'après, mais, nous les avons créées après avoir attribué tous les privilèges à admin_orclpdb1. Ensuite nous avons créés 2 rôles afin de pouvoir attribuer les privilèges à nos utilisateurs actuels et nos futurs utilisateurs/tables plus facilement.


```
SQL> CREATE ROLE rw;

Role created.

SQL> CREATE ROLE r;

Role created.

SQL> █
```

```
SQL> GRANT SELECT, INSERT, UPDATE, DELETE ON Company TO rw;
GRANT SELECT, INSERT, UPDATE, DELETE ON logs_ia TO rw;
GRANT SELECT, INSERT, UPDATE, DELETE ON IA_Type TO rw;
GRANT SELECT, INSERT, UPDATE, DELETE ON Specs TO rw;
GRANT SELECT, INSERT, UPDATE, DELETE ON Config TO rw;
GRANT SELECT, INSERT, UPDATE, DELETE ON IA_Gen TO rw;
```

```
SQL>
Grant succeeded.

SQL>
Grant succeeded.

SQL>
Grant succeeded.

SQL>
Grant succeeded.

SQL> SQL> SQL> █
```

```
SQL> GRANT SELECT ON Company TO r;
GRANT SELECT ON logs_ia TO r;
GRANT SELECT ON IA_Type TO r;
GRANT SELECT ON Specs TO r;
GRANT SELECT ON Config TO r;
GRANT SELECT ON IA_Gen TO r;
Grant succeeded.
```

```
SQL>
Grant succeeded.
```

```
SQL>
Grant succeeded.
```

```
SQL>
Grant succeeded.
```

```
SQL>
Grant succeeded.
```

```
SQL> █
```

```
SQL> GRANT rw TO u_rw_orclpdb1;

Grant succeeded.
```

```
SQL> GRANT r TO u_r_orclpdb1;

Grant succeeded.

SQL> █
```

Nous avons aussi décidé d'utiliser des mdp respectant la **complexité des mots de passe** recommandé par l'ANSSI (Agence Nationale de la Sécurité des Services d'Information) qui est :

- Avoir une longueur d'au moins **16 caractères**
- Au minimum **1 caractère spécial**
- Au minimum **1 chiffre**
- Utiliser des **MAJUSCULES/minuscules**

Nous avons généré ces mots de passe pour chacun de nos utilisateurs à l'aide de sites spécialisés (<https://generateurdemotdepasse.fr/>).

Voici un exemple de mot de passe généré : CQ6Yd4cw2w87S5iY

De plus, nous ne voulions pas connaître nos mots de passe ainsi que les taper prends trop de temps, nous avons donc ajouté un script dans le répertoire personnel de nos utilisateurs (Ubuntu) /home/xyz sous la forme de fichiers cachés .advdb_connect.sh qui lorsqu'il sont exécutés, enregistres les mots de passe et les noms de nos utilisateurs de la bdd Oracle dans nos **variables d'environnement** (qui sont réinitialisés à chaque déconnexion) et se connecte à la base de donnée.

```
mathieu@eteris-cloud-provider:~$ cat advdb_connect.sh
#!/bin/bash

if [ "$1" == "admin" ]; then
    sudo docker exec -it --user oracle oracle-advdb bash -c "
        source ~/.bashrc
        sqlplus / as sysdba
    "
else
    DB_CONNECT_STRING='(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=ADVDB)))'
    sudo docker exec -it oracle-advdb sqlplus "$DB_USER/$DB_PASS@$DB_CONNECT_STRING"
fi
```

```
export ADM_ORCLPDB1='admin_orclpdb1'
export ADM_PASSWD='2F:@?RipG7E)49vp'
export URW_ORCLPDB1='u-rw_orclpdb1'
export URW_PASSWD='n4g$~8xnLF86C~G:'
export UR_ORCLPDB1='u-r_orclpdb1'
export UR_PASSWD=',-pe,z2P!D9R4m6Z'
```

5. Requêtes et optimisation

Création des tables :

```
CREATE SEQUENCE logs_seq START WITH 1 INCREMENT BY 1 CACHE 20;
CREATE SEQUENCE company_seq START WITH 1 INCREMENT BY 1 CACHE 20;
CREATE SEQUENCE iatype_seq START WITH 1 INCREMENT BY 1 CACHE 20;
CREATE SEQUENCE cfg_seq START WITH 1 INCREMENT BY 1 CACHE 20;
CREATE SEQUENCE specs_seq START WITH 1 INCREMENT BY 1 CACHE 20;
CREATE SEQUENCE iagen_seq START WITH 1 INCREMENT BY 1 CACHE 20;
```

CACHE 20 est la valeur par défaut, c'est suffisant et ça permet une petite amélioration de l'optimisation

```
CREATE TABLE logs_ia (
    id NUMBER DEFAULT logs_seq.nextval PRIMARY KEY,
    type VARCHAR(50) NOT NULL,
    date_creation DATE NOT NULL,
    state VARCHAR(20));

CREATE TABLE Company (
    id NUMBER DEFAULT company_seq.nextval PRIMARY KEY,
    name VARCHAR(255) NOT NULL);

CREATE TABLE IA_Type (
    id NUMBER DEFAULT iatype_seq.nextval PRIMARY KEY,
    category VARCHAR(255) NOT NULL,
    modality VARCHAR(150));

CREATE TABLE Config (
    id NUMBER DEFAULT cfg_seq.nextval PRIMARY KEY,
    api_available NUMBER(1),
    open_source NUMBER(1));

CREATE TABLE Specs (
    id NUMBER DEFAULT specs_seq.nextval PRIMARY KEY,
    mod_text NUMBER(1),
    mod_image NUMBER(1),
    mod_video NUMBER(1),
    mod_audio NUMBER(1),
    mod_code NUMBER(1),
    mod_design NUMBER(1),
    mod_infra NUMBER(1),
    mod_productivity NUMBER(1),
    mod_safety NUMBER(1),
```

```
mod_multimodal NUMBER(1),  
modality_count INT);
```

```
CREATE TABLE IA_Gen (  
    id NUMBER DEFAULT iagen_seq.nextval PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    website VARCHAR(255),  
    release_year DATE,  
    fk_specs INT,  
    fk_company INT,  
    fk_iatype INT,  
    fk_cfg INT,  
    FOREIGN KEY (fk_specs) REFERENCES Specs(id),  
    FOREIGN KEY (fk_company) REFERENCES Company(id),  
    FOREIGN KEY (fk_iatype) REFERENCES IA_Type(id),  
    FOREIGN KEY (fk_cfg) REFERENCES Config(id));  
COMMIT;
```

Conception de requêtes SQL classiques :

- 1) les IA qui sont à la fois disponibles via API et OPEN SOURCE

```
SELECT g.name, g.release_year  
FROM IA_Gen g  
JOIN Config cfg ON g.fk_cfg = cfg.id  
WHERE cfg.api_available=1 AND cfg.open_source=1;
```

Optimisation avec :

```
CREATE INDEX idx_cfg_api_open  
ON Config (api_available, open_source, id);
```

rdrails

Guardrails AI (01-DEC-23
framework)

LMQL	01-DEC-23
DeepSeek R1	01-DEC-25
Mistral Large	01-DEC-23

NAME	RELEASE_Y
-----	-----
Llama 3.1	01-DEC-23
Qwen 2.5	01-DEC-25
Gemma 2	01-DEC-24
DBRX	01-DEC-24
Phi-3	01-DEC-24
MPT-7B	01-DEC-23
Mixtral 8x7B	01-DEC-23
Mistral Codestr	01-DEC-23
al	

Code Llama	01-DEC-23
------------	-----------

NAME	RELEASE_Y
-----	-----
Llama Guard	01-DEC-23
Transformers	01-DEC-18
Diffusers	01-DEC-22
Gradio	01-DEC-19
Hugging Face Sp	01-DEC-21
aces	

ComfyUI	01-DEC-23
Black Forest La	01-DEC-24
bs FLUX	

NAME	RELEASE_Y
-----	-----
Coqui TTS	01-DEC-21
TruLens	01-DEC-23
Ollama	01-DEC-23
WhisperX	01-DEC-22
Silero VAD	01-DEC-20
Papers with Cod	01-DEC-18
e SOTA	

Kaggle Models	01-DEC-23
OpenHands	01-DEC-24
LlamaGuard 2	01-DEC-23

41 rows selected.

SQL> █

Avant optimisation :

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		16	512	4 (0)	00:00:01
* 1	HASH JOIN		16	512	4 (0)	00:00:01
* 2	TABLE ACCESS FULL	CONFIG	1	9	2 (0)	00:00:01
3	TABLE ACCESS FULL	IA_GEN	113	2599	2 (0)	00:00:01

Après optimisation :

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		16	512	2 (0)	00:00:01
1	NESTED LOOPS		16	512	2 (0)	00:00:01
2	TABLE ACCESS FULL	IA_GEN	113	2599	2 (0)	00:00:01
* 3	INDEX RANGE SCAN	IDX_CFG_API_OPEN	1	9	0 (0)	00:00:01

2) Nombre d'IA par entreprise

```
SELECT c.name, COUNT(g.id) ia_count
FROM Company c
JOIN IA_Gen g ON g.fk_company = c.id
GROUP BY c.name
ORDER BY ia_count DESC;
```

Optimisation avec :

```
CREATE INDEX idx_iagen_company ON IA_Gen(fk_company);
```

COMPANY

IA_COUNT

OpenAI
9

Meta
5

Mistral AI
4

COMPANY

IA_COUNT

Hugging Face
4

Community
3

Anthropic
3

COMPANY

IA_COUNT

Adobe
3

Google
2

xAI
2

Avant optimisation :

```
--
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
----	-----------	------	------	-------	-------------	------

```
--
```

PLAN_TABLE_OUTPUT

0	SELECT STATEMENT		83	1577	5 (20)	00:00:01
1	SORT ORDER BY		83	1577	5 (20)	00:00:01
2	HASH GROUP BY		83	1577	5 (20)	00:00:01
* 3	HASH JOIN		113	2147	4 (0)	00:00:01

PLAN_TABLE_OUTPUT

4	VIEW	VW_GBF_7	83	1328	2 (0)	00:00:01
5	TABLE ACCESS FULL	COMPANY	83	996	2 (0)	00:00:01
6	TABLE ACCESS FULL	IA_GEN	113	339	2 (0)	00:00:01

Après optimisation :

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
Time					

PLAN_TABLE_OUTPUT					
0	SELECT STATEMENT		83	1577	3 (34)
00:00:01					
1	SORT ORDER BY		83	1577	3 (34)
00:00:01					
2	HASH GROUP BY		83	1577	3 (34)
00:00:01					
3	NESTED LOOPS		113	2147	2 (0)
00:00:01					
PLAN_TABLE_OUTPUT					
4	VIEW	VW_GBF_7	83	1328	2 (0)
00:00:01					
5	TABLE ACCESS FULL	COMPANY	83	996	2 (0)
00:00:01					
* 6	INDEX RANGE SCAN	IDX_IAGEN_COMPANY	1	3	0 (0)
00:00:01					

3) IA sorties après la moyenne des années de sortie

```
SELECT name,
       EXTRACT(YEAR FROM release_year) AS release_year
FROM IA_Gen
WHERE EXTRACT(YEAR FROM release_year) >
      (SELECT AVG(EXTRACT(YEAR FROM release_year)) FROM
IA_Gen);
```

Optimisation avec :

```
CREATE INDEX idx_iagen_release_year
ON IA_Gen (EXTRACT(YEAR FROM release_year));
```

NAME	RELEASE_YEAR
Claude	2023
Gemini	2023
DALL·E 3	2023
Pika	2023
Suno	2023
Cursor	2023
Sourcegraph Cody	2023

Amazon Q Developer	2023
--------------------	------

NAME	RELEASE_YEAR
JetBrains AI Assistant	2023
pgvector	2023
Deepgram	2023
NVIDIA NeMo Guardrails	2023

Lakera Guard	2023
Guardrails AI (2023

NAME	RELEASE_YEAR
framework)	
LMQL	2023
OpenRouter	2023
Vercel v0	2023
Framer AI	2023
Notion AI	2023
xAI Grok	2023
Mistral Large	2023
Llama 3.1	2023
MPT-7B	2023

NAME	RELEASE_YEAR
Mixtral 8x7B	2023
GPT-4o	2023
Claude 3.7 Sonnet	2023
Mistral Codestral	2023
Code Llama	2023

Avant optimisation :

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		6	120	4 (0)	00:00:01
* 1	TABLE ACCESS FULL	IA_GEN	6	120	2 (0)	00:00:01
2	SORT AGGREGATE		1	8		
3	TABLE ACCESS FULL	IA_GEN	113	904	2 (0)	00:00:01

Après optimisation :

```

-----
| Id | Operation                                | Name                                | Rows | B
ytes | Cost (%CPU)| Time      |
-----
PLAN_TABLE_OUTPUT
|  0 | SELECT STATEMENT                                |                                |    6 |
150 |      2   (0)| 00:00:01 |
|  1 | TABLE ACCESS BY INDEX ROWID BATCHED| IA_GEN                                |    6 |
150 |      1   (0)| 00:00:01 |
|*  2 | INDEX RANGE SCAN                                | IDX_IAGEN_RELEASE_YEAR |    1 |
|      1   (0)| 00:00:01 |
|  3 | SORT AGGREGATE                                |                                |    1 |
13 |                                |
PLAN_TABLE_OUTPUT
|  4 | INDEX FULL SCAN                                | IDX_IAGEN_RELEASE_YEAR |  113 |
1469 |      1   (0)| 00:00:01 |
-----

```

4) Nombre d'IA par catégorie fonctionnelle

```

SELECT t.category,
       COUNT(g.id) AS total_ia
FROM   IA_Type t
JOIN   IA_Gen g ON g.fk_iatype = t.id
GROUP BY t.category
ORDER BY total_ia DESC;

```

Optimisation avec :

```

CREATE INDEX idx_iagen_iatype ON IA_Gen(fk_iatype);

```

2	3	4	5	6
CATEGORY				

TOTAL_IA				

LLMs & Chat Assistants				
32				
Other				
27				
Video Gen & Editing				
13				
CATEGORY				

TOTAL_IA				

Image Gen & Editing				
12				
Search & RAG				
6				
Audio/Music/TTS				
6				
CATEGORY				

TOTAL_IA				

Safety & Guardrails				
5				
Design & UI				
4				
Infra & Inference				
3				
CATEGORY				

TOTAL_IA				

Evaluation & Benchmarks				
2				
Speech-to-Text (ASR)				

Avant optimisation :

```
--
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
----	-----------	------	------	-------	-------------	------

```
--
```

PLAN_TABLE_OUTPUT

0	SELECT STATEMENT		13	338	5 (20)	00:00:01
1	SORT ORDER BY		13	338	5 (20)	00:00:01
2	HASH GROUP BY		13	338	5 (20)	00:00:01
* 3	HASH JOIN		113	2938	4 (0)	00:00:01

PLAN_TABLE_OUTPUT

4	VIEW	VW_GBF_7	25	575	2 (0)	00:00:01
5	TABLE ACCESS FULL	IA_TYPE	25	500	2 (0)	00:00:01
6	TABLE ACCESS FULL	IA_GEN	113	339	2 (0)	00:00:01

```
--
```

Après optimisation :

```

-----
| Id | Operation                      | Name                      | Rows | Bytes | Cost (%CPU)|
Time |
-----
PLAN_TABLE_OUTPUT
-----
| 0 | SELECT STATEMENT                |                          | 13   | 338   | 3   (34)|
00:00:01 |
| 1 | SORT ORDER BY                  |                          | 13   | 338   | 3   (34)|
00:00:01 |
| 2 | HASH GROUP BY                  |                          | 13   | 338   | 3   (34)|
00:00:01 |
| 3 | NESTED LOOPS                   |                          | 113  | 2938  | 2   (0)|
00:00:01 |
PLAN_TABLE_OUTPUT
-----
| 4 | VIEW                            | VW_GBF_7                 | 25   | 575   | 2   (0)|
00:00:01 |
| 5 | TABLE ACCESS FULL             | IA_TYPE                   | 25   | 500   | 2   (0)|
00:00:01 |
|* 6 | INDEX RANGE SCAN               | IDX_IAGEN_IATYPE         | 5    | 15    | 0   (0)|
00:00:01 |

```

5) Classement des entreprises selon le nombre d'IA

```

SELECT company,
       ia_count,
       RANK() OVER (ORDER BY ia_count DESC) AS rank_company
FROM (
    SELECT c.name AS company,
           COUNT(g.id) AS ia_count
    FROM Company c
    LEFT JOIN IA_Gen g ON g.fk_company = c.id
    GROUP BY c.name
);

```

Optimisation avec :

```

CREATE MATERIALIZED VIEW mv_company_rank AS
SELECT c.name AS company,
       COUNT(g.id) AS ia_count
FROM Company c

```

```
LEFT JOIN IA_Gen g ON g.fk_company = c.id  
GROUP BY c.name;
```

```
SELECT company,  
       ia_count,  
       RANK() OVER (ORDER BY ia_count DESC) AS rank_company  
FROM mv_company_rank;
```

COMPANY	
IA_COUNT	RANK_COMPANY
OpenAI	91
Meta	52
Mistral AI	43
COMPANY	
IA_COUNT	RANK_COMPANY
Hugging Face	43
Community	35
Anthropic	35
COMPANY	
IA_COUNT	RANK_COMPANY
Adobe	35
Google	28
xAI	28
COMPANY	
IA_COUNT	RANK_COMPANY
Canva	28
Amazon Web Services	28
Google Cloud	

Avant optimisation :

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		83	1577	6 (34)	00:00:01
1	WINDOW SORT		83	1577	6 (34)	00:00:01
2	HASH GROUP BY		83	1577	6 (34)	00:00:01
* 3	HASH JOIN OUTER		113	2147	4 (0)	00:00:01
4	TABLE ACCESS FULL	COMPANY	83	996	2 (0)	00:00:01
5	TABLE ACCESS FULL	IA_GEN	113	791	2 (0)	00:00:01

Après optimisation :

```

-----
| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |
-----
| 0 | SELECT STATEMENT | | 83 | 996 | 3 (34)| 00:00:01 |
| 1 | WINDOW SORT | | 83 | 996 | 3 (34)| 00:00:01 |
| 2 | MAT_VIEW ACCESS FULL | MV_COMPANY_RANK | 83 | 996 | 2 (0)| 00:00:01 |
-----
PLAN_TABLE_OUTPUT
-----
| 0 | SELECT STATEMENT | | 83 | 996 | 3 (34)| 00:00:01 |
| 1 | WINDOW SORT | | 83 | 996 | 3 (34)| 00:00:01 |
| 2 | MAT_VIEW ACCESS FULL | MV_COMPANY_RANK | 83 | 996 | 2 (0)| 00:00:01 |
-----

```

6) IA multimodales avec leur entreprise et type

```

SELECT g.name AS ia_name,
       c.name AS company,
       t.category,
       t.modality
FROM IA_Gen g
JOIN Company c ON g.fk_company = c.id
JOIN IA_Type t ON g.fk_iatype = t.id
WHERE EXISTS (
    SELECT 1
    FROM Specs s
    WHERE s.id = g.fk_specs
          AND s.mod_multimodal = 1
);

```

Optimisation avec :

```

CREATE INDEX idx_iagen_fkspecs ON IA_Gen(fk_specs);

```

IA_NAME

COMPANY

CATEGORY

MODALITY

ChatGPT

OpenAI

LLMs & Chat Assistants

multimodal

IA_NAME

COMPANY

CATEGORY

MODALITY

OpenAI o3-mini

OpenAI

LLMs & Chat Assistants

multimodal

IA_NAME

COMPANY

CATEGORY

MODALITY

GPT-4o

OpenAI

LLMs & Chat Assistants

multimodal

IA_NAME

COMPANY

CATEGORY

MODALITY

Avant optimisation :

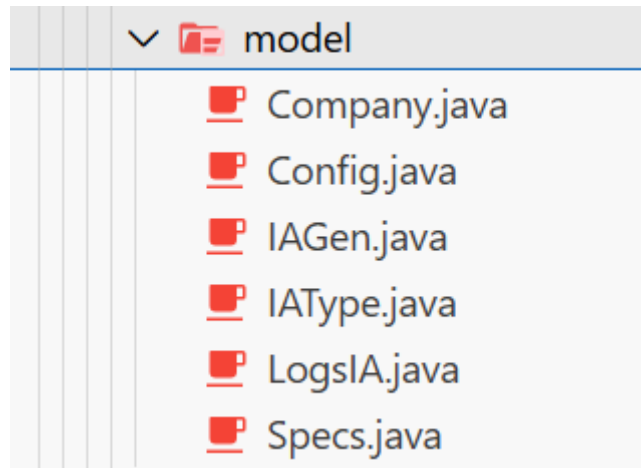
```
--
| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time
|
-----
PLAN_TABLE_OUTPUT
| 0 | SELECT STATEMENT | | 11 | 715 | 8 (0)| 00:00:01
|
|* 1 | HASH JOIN | | 11 | 715 | 8 (0)| 00:00:01
|
|* 2 | HASH JOIN | | 11 | 583 | 6 (0)| 00:00:01
|
| 3 | MERGE JOIN CARTESIAN | | 25 | 800 | 4 (0)| 00:00:01
|
PLAN_TABLE_OUTPUT
|* 4 | TABLE ACCESS FULL | SPECS | 1 | 6 | 2 (0)| 00:00:01
|
| 5 | BUFFER SORT | | 25 | 650 | 2 (0)| 00:00:01
|
| 6 | TABLE ACCESS FULL | IA_TYPE | 25 | 650 | 2 (0)| 00:00:01
|
| 7 | TABLE ACCESS FULL | IA_GEN | 113 | 2373 | 2 (0)| 00:00:01
|
PLAN_TABLE_OUTPUT
|
| 8 | TABLE ACCESS FULL | COMPANY | 83 | 996 | 2 (0)| 00:00:01
|
-----
```

Après optimisation :

Id	Operation	Name	Rows	Bytes	Cost
0	SELECT STATEMENT		11	715	
1	HASH JOIN		11	715	
2	HASH JOIN		11	583	
3	NESTED LOOPS		11	297	
4	NESTED LOOPS		11	297	
5	TABLE ACCESS FULL	SPECS	1	6	
6	INDEX RANGE SCAN	IDX_IAGEN_FKSPECS	11		
7	TABLE ACCESS BY INDEX ROWID	IA_GEN	11	231	
8	TABLE ACCESS FULL	IA_TYPE	25	650	
9	TABLE ACCESS FULL	COMPANY	83	996	

6. Interface graphique (bonus)

Nous avons utilisé Java Springboot. Après avoir installé les dépendances et avoir rendu propre l'architecture du projet, nous avons créé les models suivants :



Les models reprennent les informations des tables afin de pouvoir etre modifiés
exemple avec Config.java

```
front > src > main > java > com > advdb > front > moc  
1  package com.advdb.front.model;  
2  
3  public record Config(  
4      Long id,  
5      boolean apiAvailable,  
6      boolean openSource  
7  ) {}
```

Ensuite, nous avons configuré un tunnel ssh afin de pouvoir accéder au serveur distant a l'aide de la librairie jsch & ssh :

```
public class SshTunnelConfig {

    private static final Logger logger = LoggerFactory.getLogger(SshTunnelConfig.class);

    @Value("${ssh.tunnel.host}")
    private String sshHost;

    @Value("${ssh.tunnel.port}")
    private int sshPort;

    @Value("${ssh.tunnel.username}")
    private String sshUsername;

    @Value("${ssh.tunnel.password}")
    private String sshPassword;

    @Value("${ssh.tunnel.local-port}")
    private int localPort;

    @Value("${ssh.tunnel.remote-host}")
    private String remoteHost;

    @Value("${ssh.tunnel.remote-port}")
    private int remotePort;

    private Session session;

    @PostConstruct
    public void startTunnel() {
        try {
            logger.info(format: "tunnel SSH vers {}", sshPort);

            JSch jsch = new JSch();
            session = jsch.getSession(sshUsername, sshHost, sshPort);

            session.setPassword(sshPassword);

            Properties config = new Properties();
            config.put(key: "StrictHostKeyChecking", value: "no");
            session.setConfig(config);

            session.connect();
            logger.info(msg: "connexion ssh établie");

            session.setPortForwardingL(localPort, remoteHost, remotePort);
            logger.info(format: "localhost:{} -> {}:{} via {}", localPort, remoteHost, remotePort, s

        } catch (JSchException e) {
            logger.error(msg: "erreur", e);
            throw new RuntimeException(message: "state:", e);
        }
    }
}
```

Ensuite, une fois que le tunnel est créé, nous avons créé une classe de connexion à la base de donnée

```
11  @Configuration
12  public class DbConnection {
13      @Value("${oracle.db.url}")
14      private String dbUrl;
15
16      @Value("${oracle.db.username}")
17      private String username;
18
19      @Value("${oracle.db.password}")
20      private String passwd;
21
22      @Value("${oracle.db.driver}")
23      private String driver;
24
25      @Bean
26      @Primary
27      public DataSource dataSource(){
28          return DataSourceBuilder.create()
29              .url(dbUrl)
30              .username(username)
31              .password(passwd)
32              .driverClassName(driver)
33              .build();
34      }
35  }
```

Une fois la configuration finie, nous avons ensuite créé notre controller :

```
@Controller
@RequestMapping("/ias")
public class IAGenController {
    @Autowired
    private JdbcTemplate jdbcTemplate;

    private LocalDateTime dateCreation;

    @GetMapping
    public String listIAs(Model model) {
        String sql = "SELECT * FROM IA_Gen";
        List<IAGen> ias = jdbcTemplate.query(sql, (rs, rowNum) -> new IAGen(
            rs.getLong(columnLabel: "id"),
            rs.getString(columnLabel: "name"),
            rs.getString(columnLabel: "website"),
            rs.getDate(columnLabel: "release_year") != null ? rs.getDate(columnLabel: "release_year") : null,
            rs.getLong(columnLabel: "fk_specs"),
            rs.getLong(columnLabel: "fk_company"),
            rs.getLong(columnLabel: "fk_iatype"),
            rs.getLong(columnLabel: "fk_cfg")
        ));
        model.addAttribute("ias", ias);
        return "ia-list";
    }
}
```

Enfin, ce controller appelle notre vue que nous avons fait générer afin de gagner du temps

```
front > src > main > resources > templates > ia-list.html > html
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4     <title>Gestion des IA</title>
5     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/boot
6 </head>
7 <body class="container mt-5">
8     <h2 class="mb-4">Répertoire des IA Génératives (PDB1)</h2>
9
10    <div class="card mb-4">
11        <div class="card-body">
12            <h5 class="card-title">Ajouter une IA</h5>
13            <form th:action="@{/ias/add}" method="post" class="row g-3">
14                <div class="col-auto">
15                    <input type="text" name="name" class="form-control" placeholder="Nom de
16                </div>
17                <div class="col-auto">
18                    <input type="text" name="website" class="form-control" placeholder="Sit
19                </div>
20                <div class="col-auto">
21                    <button type="submit" class="btn btn-primary">Enregistrer</button>
22                </div>
23            </form>
24        </div>
25    </div>
```

Et voici ce que donne notre front :

localhost:7777/ias

Répertoire des IA Génératives (PDB1)

Ajouter une IA

Nom de l'IA Site Web Enregistrer

ID	Nom	Site Web	Actions
1	ChatGPT	https://chatgpt.com	Supprimer
2	Claude	https://claude.ai	Supprimer
3	Gemini	https://gemini.google.com	Supprimer
4	Midjourney	https://www.midjourney.com	Supprimer
5	Stable Diffusion	https://stability.ai/stable-image	Supprimer
6	DALL-E 3	https://openai.com/index/dall-e-3-system-card	Supprimer
7	Runway Gen-3	https://runwayml.com	Supprimer

On peut ajouter des ia gen (seulement sur la table IA_Gen, faute de temps) et les supprimer (cascade).