

ADV-DB

MACSIN4A0325: Advanced DB Management





Table des matières

1

Introduction

Présentation du dataset

2

Modélisation

Schéma, clés primaires

3

Contexte

Environnement

4

Sécurité

Privilèges, profiles

5

Optimisation

Requêtes SQL

6

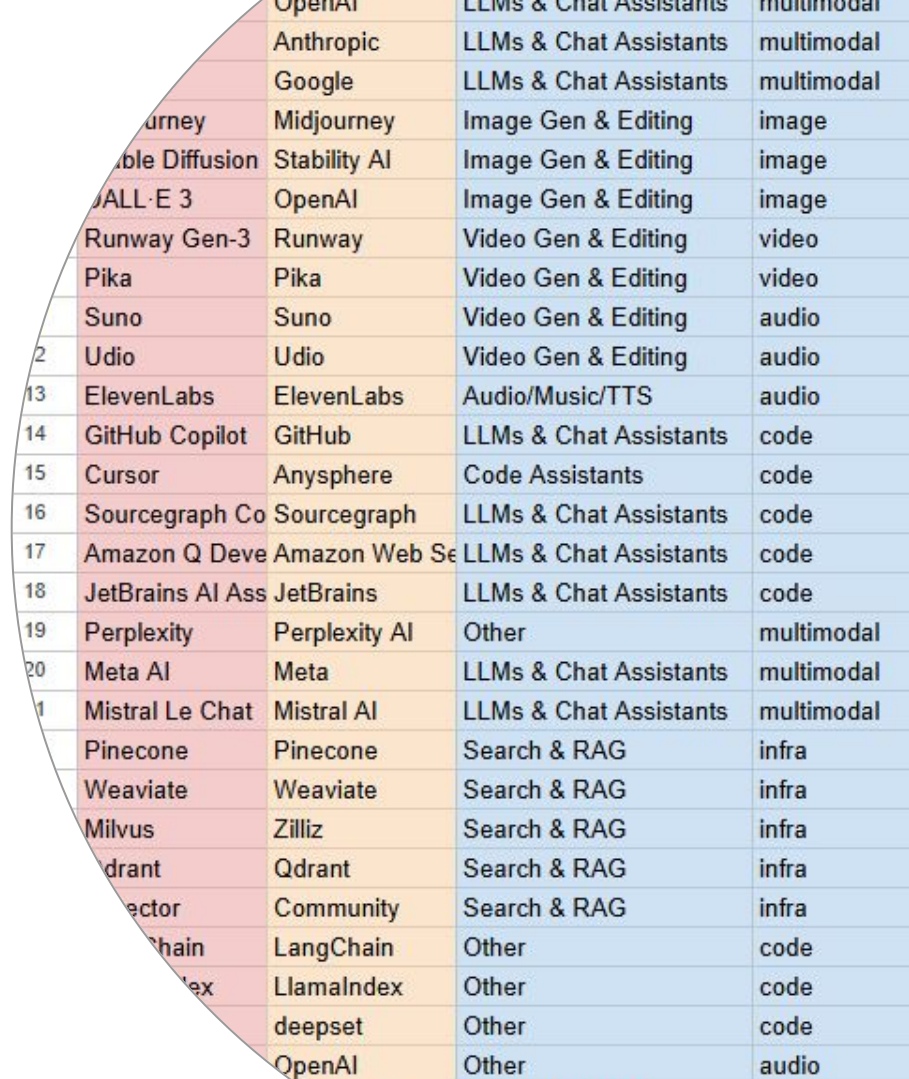
Conclusion

Conclusion et bonus



01 - Introduction

Présentation du dataset



1	OpenAI	LLMs & Chat Assistants	multimodal
2	Anthropic	LLMs & Chat Assistants	multimodal
3	Google	LLMs & Chat Assistants	multimodal
4	Midjourney	Image Gen & Editing	image
5	Stability AI	Image Gen & Editing	image
6	OpenAI	Image Gen & Editing	image
7	Runway	Video Gen & Editing	video
8	Pika	Video Gen & Editing	video
9	Suno	Video Gen & Editing	audio
10	Udio	Video Gen & Editing	audio
11	ElevenLabs	Audio/Music/TTS	audio
12	GitHub Copilot	LLMs & Chat Assistants	code
13	Cursor	Code Assistants	code
14	Sourcegraph	LLMs & Chat Assistants	code
15	Amazon Q Developer	LLMs & Chat Assistants	code
16	JetBrains AI Assistant	LLMs & Chat Assistants	code
17	Perplexity	Other	multimodal
18	Meta AI	LLMs & Chat Assistants	multimodal
19	Mistral AI	LLMs & Chat Assistants	multimodal
20	Pinecone	Search & RAG	infra
21	Weaviate	Search & RAG	infra
22	Milvus	Search & RAG	infra
23	Qdrant	Search & RAG	infra
24	Community	Search & RAG	infra
25	LangChain	Other	code
26	LlamaIndex	Other	code
27	deepset	Other	code
28	OpenAI	Other	audio

45%

des modèles sont sortis en **2023**

67%

des IA gen ne sont **PAS** open-source

27%

sont des **LLMS**



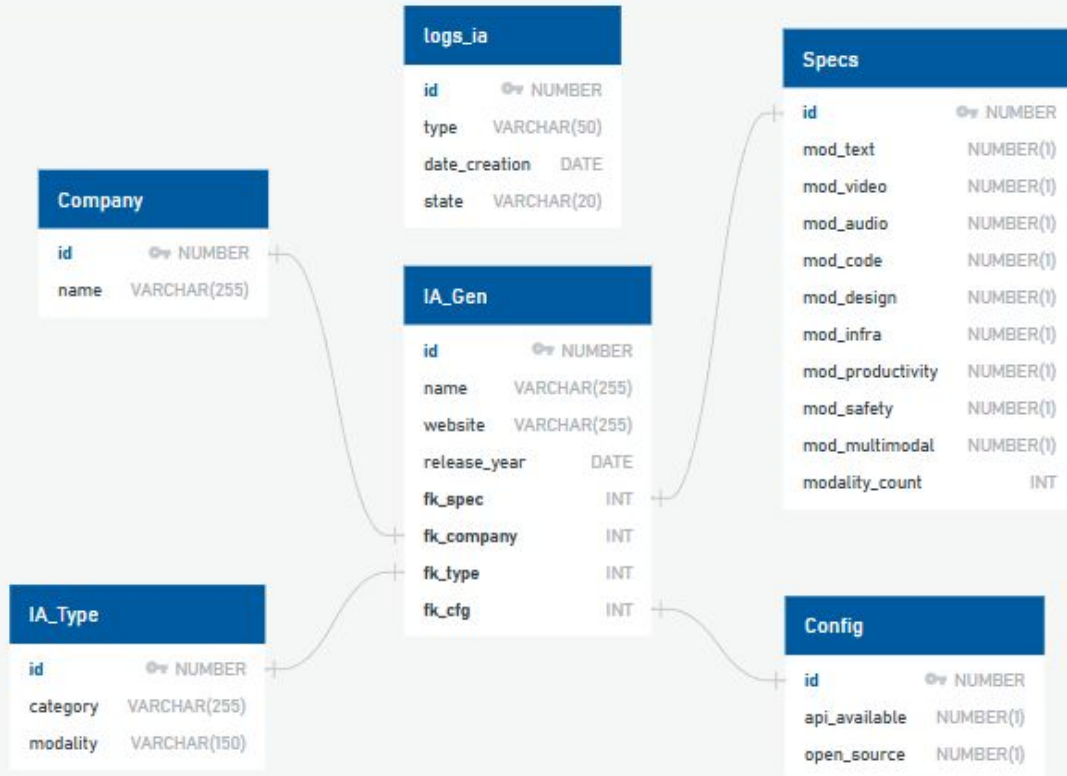
A propos du dataset

115 lignes au **total**

0% de **mismatched** ou **missing**

22 colonnes





02 - Modélisation



03 - Contexte

Environnement

04 - Sécurité

Gestion des users, privilèges



Création des users

```
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.
```

NOMBRE DE CARACTÈRES	UNIQUEMENT DES CHIFFRES	LETTRES MINUSCULES	LETTRES MINUSCULES ET MAJUSCULES	LETTRES MINUSCULES ET MAJUSCULES + CHIFFRES	LETTRES MINUSCULES ET MAJUSCULES + CHIFFRES + CARACTÈRES SPÉCIAUX
4	IMMÉDIATEMENT	IMMÉDIATEMENT	IMMÉDIATEMENT	IMMÉDIATEMENT	IMMÉDIATEMENT
6	IMMÉDIATEMENT	IMMÉDIATEMENT	IMMÉDIATEMENT	1 sec	5 sec
8	IMMÉDIATEMENT	5 sec	22 min	1 heure	9 heures
10	IMMÉDIATEMENT	58 min	1 mois	7 mois	5 ans
12	45 sec	3 semaines	300 ans	2000 ans	34 000 ans
14	41 min	51 ans	800 000 ans	9 millions d'années	200 millions d'années

*source : SCSP Community (Seasoned Cyber Security Professionals)

```
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'
```



Profiles, rôles & privilèges

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

Profile created.

```
SQL> █
```

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

Profile created.

```
SQL> █
```



Profiles, rôles & privilèges

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

```
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 rw TO u_rw_orclpdb1;
```

Grant succeeded.

```
SQL> GRANT r TO u_r_orclpdb1;
```

Grant succeeded.

```
SQL> █
```

```
SQL>  
Grant succeeded.
```

```
SQL>  
Grant succeeded.
```

```
SQL>  
Grant succeeded.
```

```
SQL>  
Grant succeeded.
```

```
SQL>  
Grant succeeded.
```

```
SQL> SQL> SQL> █
```

05 - Optimisation

Requêtes, index, vues



Optimisation par **cache**

```
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;
```

Optimisation des séquence avec pré-allocation des valeurs



Optimisation par Index

1ère requête - Les IA qui sont a la fois disponibles via API & 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;
```

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

Résultat :

Coût : 4

Coût après l'optimisation : 2

```
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
Gemini 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
al            01-DEC-23

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
aces          01-DEC-21

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

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
e SOTA        01-DEC-18

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

41 rows selected.

SQL>
```

Optimisation par Index

2 ème requête - Nombre d'IA par entreprise :

```
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  
ORDER BY ia_count DESC;  
  
CREATE INDEX idx_cfg_api_open  
ON Config (api_available, open_source, id);
```

Résultat :

Coût : 5

Coût après l'optimisation : 2

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



Optimisation par Index

3ème requête - IA sorties après la moyenne des années de sorties :

```
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);
```

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

Résultat :

Coût : 4

Coût après l'optimisation : 2

NAME	RELEASE_YEAR
Claude	2023
Gemini	2023
DALL·E 3	2023
Pika	2023
Suno	2023
Cursor	2023
Sourcegraph Cody	2023
Amazon Q Developer	2023
JetBrains AI Assistant	2023
pgvector	2023
Deepgram	2023
NVIDIA NeMo Guardrails	2023
Lakera Guardrails AI (framework)	2023
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
Mixtral 8x7B	2023
GPT-4o	2023
Claude 3.7 Sonnet	2023
Mistral Codestral	2023
Code Llama	2023

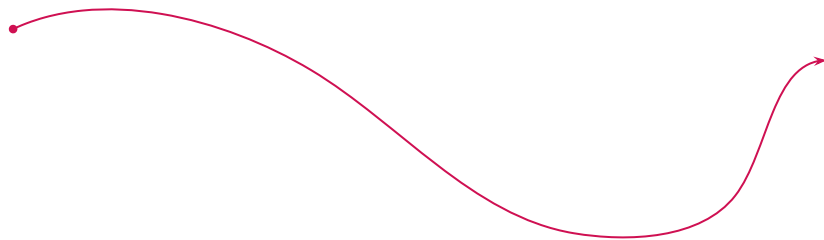
Optimisation par Index

4ème requête - 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;
```

```
CREATE INDEX idx_iagen_iatype ON IA_Gen(fk_iatype);
```

Résultat :



Coût : 5

Coût après l'optimisation : 3

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)

Optimisation par Index

5ème requête - IA multimodales avec leur entreprise et leur types :

```
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  
JOIN Specs s ON g.fk_specs = s.id  
WHERE s.mod_multimodal = 1;
```

```
CREATE INDEX idx_specs_multimodal ON Specs(mod_multimodal);
```

Résultat :

Coût : 8

Coût après l'optimisation : 7

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

Optimisation par vues

6ème requête - 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);
```

```
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;
```

Résultat sur la slide suivante

Coût : 6

Coût après l'optimisation : 3



Optimisation par vues

COMPANY		

IA_COUNT	RANK_COMPANY	

OpenAI	9	1
Meta	5	2
Mistral AI	4	3
COMPANY		

IA_COUNT	RANK_COMPANY	

Hugging Face	4	3
Community	3	5
Anthropic	3	5
COMPANY		

IA_COUNT	RANK_COMPANY	

Adobe	3	5
Google	2	8
xAI	2	8
COMPANY		

IA_COUNT	RANK_COMPANY	

Canva	2	8
Amazon Web Services	2	8
Google Cloud		





06 - Conclusion

Remerciements & Bonus

Bonus



```
SshTunnelConfig.java

@Configuration
@ConditionalOnProperty(name = "ssh.tunnel.enabled", havingValue = "true")
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;
```

```
DbConnection.java

@Configuration
public class DbConnection {

    @Value("${oracle.db.url}")
    private String dbUrl;

    @Value("${oracle.db.username}")
    private String username;

    @Value("${oracle.db.password}")
    private String passwd;

    @Value("${oracle.db.driver}")
    private String driver;

    @Bean
    @Primary
    public DataSource dataSource(){
        return DataSourceBuilder.create()
            .url(dbUrl)
            .username(username)
            .password(passwd)
            .driverClassName(driver)
            .build();
    }
}
```

Demo



Merci de nous avoir écoutés !

Liens :

<https://www.kaggle.com/datasets/wardabilal/global-generative-ai-tools-landscape-2025>

<https://github.com/MathieuAudibert/advdb>

https://docs.google.com/document/d/14W2J3o2_DF9VccvgDo429PwX9op7GCDTn-Hi6Aufqh0/edit?tab=t.0

BIGAND Théo, AGOSTINO Roméo & AUDIBERT Mathieu

