

# SUPABASE WITH SD4LE

WED, MAY 29, 2024

윤태영

# SD4LE? SUPABASE?



# SD4LE? SUPABASE?



# CONTENTS

- 어쩌서 DB를 쓰려는 것이지?
- Supabase란 무엇인가?
- SD4LE with Supabase

어째서 DB를 쓰려는 것이지?

# 어째서 DB를 쓰려는 것이지?

평가하기

성명 [redacted] 소속 교육조직>대학>학부(과)>컴퓨터공학과 문제풀이 현황

번호	평가문제 지문
1	기계,기구 및 설비점검사항 중 정지 중 점검사항으로 잘못된 것은? <input type="radio"/> 급유의 이상유무 <input type="radio"/> 전동기 스위치 이상유무 <input type="radio"/> 겹지상대 이상유무 <input type="radio"/> 과열현상이나 스파크 발생유무
2	생활습관병의 종류가 아닌 것은? <input type="radio"/> 고혈압 <input type="radio"/> 당뇨병 <input type="radio"/> 알 <input type="radio"/> 비만
3	VDT와 관련된 직결적 유해요인에 해당하지 않는 것은? <input type="radio"/> 반복성 부하 <input type="radio"/> 정적인 부하 <input type="radio"/> 지속적 부하 <input type="radio"/> 유해여인이 포함된 취미활동
4	다음 중 영상표시 단말기를 뜻하는 VDT에 속하지 않는 것은?

제출하기

※연구실 안전교육 수강완료 후 평가를 실시하여 60 점 이상이어야 안전교육 이수됩니다. 평가하기

테스트 당시 12번 중 1번 꼴로 통과 최저 점수를 못 넘는 것을 확인함.

# 어째서 DB를 쓰려는 것이지?



11/12 -> "와 프로그램 잘 돌아가네요 ㅎㅎ 굿"

# 어째서 DB를 쓰려는 것이지?



1/12 -> "뭐야 안 되잖아요 프로그램 수준 ㅈㅈ"



# 어째서 DB를 쓰려는 것이지?

## Accessing GPT-4, GPT-4 Turbo and GPT-4o in the OpenAI API

### GPT-4 Turbo vs. GPT-4o

GPT-4o has the same high intelligence but is faster, cheaper, and has higher rate limits than GPT-4 Turbo. Specifically:

- Pricing: GPT-4o is 50% cheaper than GPT-4 Turbo, coming in at \$5/M input and \$15/M output tokens).
- Rate limits: GPT-4o's rate limits are 5x higher than GPT-4 Turbo—up to 10 million tokens per minute.
- Speed: GPT-4o is 2x as fast as GPT-4 Turbo.
- Vision: GPT-4o's vision capabilities perform better than GPT-4 Turbo in evals related to vision capabilities.
- Multilingual: GPT-4o has improved support for non-English languages over GPT-4 Turbo.

GPT-4o가 GPT-4 Turbo보다 2배 빠르고, 비용도 절반이라는데...

문제 푸는 애를 GPT-3.5 Turbo → GPT-4o 바꿔버려?

# 어째서 DB를 쓰려는 것이지?



'환율도 높은데, 실행 횟수 & 문제 발생 횟수 카운트하도록 해놓고  
나중에 봐서 심각하다 싶으면 그때 가서 바꾸자 ㅎㅎ'

(GPT-4 시리즈로 돌렸을 때 얼마가 나올지 모르기 때문)

# 어째서 DB를 쓰려는 것이지?



문제 생길 때마다  
파일 내 데이터 확인하고, +1 시켜서 다시 또 파일 업로드하는 기능  
구현할 생각에 벌써부터 속이 답답

# 어째서 DB를 쓰려는 것이지?



"안 되겠다 DB 써서 간단하게 쓱싹 구현해 봅시다"

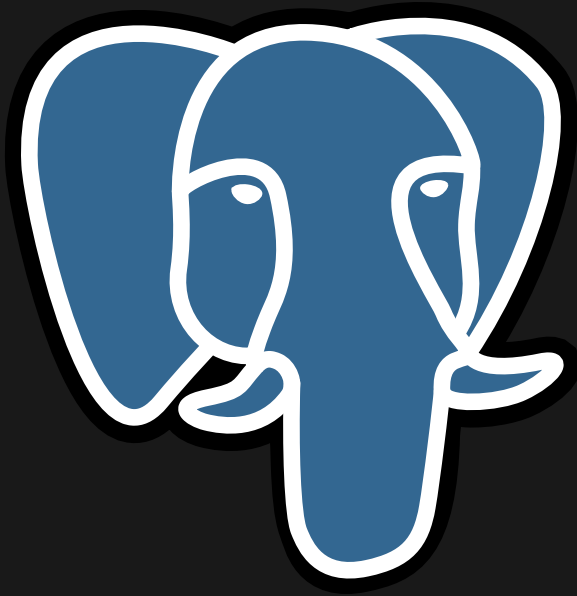
**SUPABASE란 무엇인가?**

# SUPABASE란 무엇인가?



데이터베이스 호스팅 서비스 like Firebase

# SUPABASE란 무엇인가?



PostgreSQL[포스트그레스 쿼엘] 기반

# SUPABASE란 무엇인가?

비교	Supabase	Firebase
소스코드	오픈소스	비공개
비활성시 중지	1주일	없음
프로젝트 수	2개	5~10개
DB	PostgreSQL, 500MB 용량	NoSQL, 1GiB 용량
사용자 인증	MAU 5만명, 소셜 로그인 업체 카카오 포함 19개	MAU 5만명, 소셜 로그인 업체 5개
서버리스 함수	5십만 호출	2백만 호출, GCP로 이전됨.
저장소	1GB 용량, 월 2GB 대역폭	5GB 용량, 일 1GB 대역폭



# SUPABASE란 무엇인가?

파이어베이스로 간단한 웹사이트 만들었는데 재밌다

---

메모장 사이트 만들.

나혼자 쓸건데 돈 안나가겠지?

# SUPABASE란 무엇인가?

파이어 베이스는 쓰는 회사들 별로 없음 ㅋㅋ ... mongodb 로 마찬가지로 사람들 잘 안씀. AWS RDS 같은거나 해야지

요즘에는또 **supabase** 라는거 쓰는 추세임. 수파베이스 라는 오픈소스 데이터베이스인데 이게 파이어베이스 대체하려고 만든거라고 하더라

- ↳ 오 ㅋㅋ 한번 봐야겠다

- ↳ 대충 봤는데 딱 내가 찾던거인듯, 다음에는 이거 써봐야지

# SUPABASE란 무엇인가?



[Product](#) ▾ [Developers](#) ▾ [Pricing](#) [Docs](#) [Blog](#)

67.2K

[Dashboard](#)

Alternative

## Supabase vs Firebase

2022-05-26 • 4 minute read



Ant Wilson  
CTO and Co-Founder

# SUPABASE란 무엇인가?

## What is Supabase?

Supabase is an open source firebase **alternative**, but instead of being built around a document-based datastore, Supabase offers a relational database management system called PostgreSQL. This comes with a few advantages:

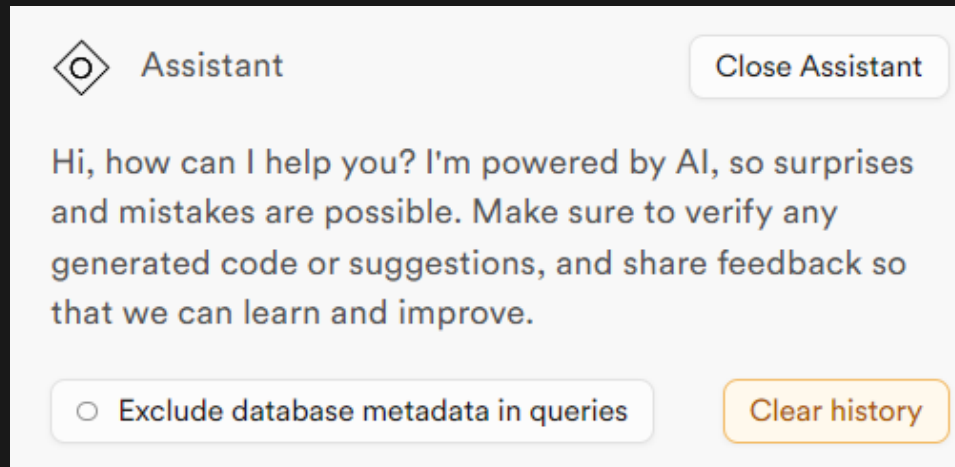
# SUPABASE란 무엇인가?

The screenshot displays the Supabase Table Editor interface. On the left, a sidebar shows the 'public' schema with a list of tables: 'error\_cnt', 'is\_user\_not\_in\_list\_able', 'latest\_version', 'unwelcome\_txt', and 'user\_data'. The 'user\_data' table is selected. The main area shows the table structure and data. The table has columns: 'id' (int8), 'user\_id' (text), 'updated\_at' (timestamp), 'is\_whitelisted' (bool), 'created\_at' (timestamp), and 'last\_used\_at' (timestamp). There are 5 records in the table, all with 'is\_whitelisted' set to 'TRUE' and 'created\_at' set to '2024-05-29 15:59:47.053013'. The 'last\_used\_at' column is NULL for all records. The interface includes a 'Filter' button, a 'Sorted by 1 rule' dropdown, an 'Insert' button, and a 'RLS disabled' status indicator. The bottom of the interface shows pagination information: 'Page 1 of 1', '100 rows', and '5 records'. There are also 'Refresh', 'Data', and 'Definition' buttons at the bottom right.

	id	user_id	updated_at	is_whitelisted	created_at	last_used_at
<input type="checkbox"/>	1	20770000	2024-05-29 15:59:47.053013	TRUE	2024-05-29 15:59:47.053013	NULL
<input type="checkbox"/>	2	20780000	2024-05-29 15:59:47.053013	TRUE	2024-05-29 15:59:47.053013	NULL
<input type="checkbox"/>	3	20790000	2024-05-29 15:59:47.053013	TRUE	2024-05-29 15:59:47.053013	NULL
<input type="checkbox"/>	4	20800000	2024-05-29 15:59:47.053013	TRUE	2024-05-29 15:59:47.053013	NULL
<input type="checkbox"/>	5	20810000	2024-05-29 15:59:47.053013	TRUE	2024-05-29 15:59:47.053013	NULL

(Supabase 웹사이트 내 프로젝트 관리 콘솔)

# SUPABASE란 무엇인가?



와! AI!

# SUPABASE란 무엇인가?



You 2 minutes ago

오늘날 포르쉐에서 판매하고 있는 차종들에 대한 데이터베이스를 만드려고 한다. 적절한 SQL 코드를 작성해 다오.



Assistant 2 minutes ago

```
-- Create table for models
CREATE TABLE
models (
  id BIGINT PRIMARY
KEY GENERATED ALWAYS AS IDENTITY,
  NAME TEXT NOT NULL,
  YEAR INTEGER NOT NULL
);
-- Create table for engines with foreign key

CREATE TABLE
engines (
  id BIGINT PRIMARY
KEY GENERATED ALWAYS AS IDENTITY,
  model_id BIGINT REFERENCES
models (id),
  TYPE TEXT NOT NULL,
  horsepower INTEGER NOT NULL
);
-- Create table for features with foreign key
```

```
29 -- Insert data into models table
30 INSERT INTO
31 models (NAME, YEAR)
32 VALUES
33 ('911', 2021),
34 ('Cayenne', 2021),
35 ('Panamera', 2021),
36 ('Macan', 2021),
37 ('Taycan', 2021),
38 ('718 Cayman', 2021),
39 ('718 Boxster', 2021),
40 ('911 Turbo S', 2021),
41 ('911 GT3', 2021),
42 ('Panamera Turbo S', 2021);
43
44 -- Insert data into engines table with model_id referencing models table
45 INSERT INTO
46 engines (
```

# SQL OF SD4LE

```
1 SET TIMEZONE = 'Asia/Seoul';
2
3 -- Drop table if it exists
4 DROP TABLE IF EXISTS user_data;
5 DROP TABLE IF EXISTS latest_version;
6 DROP TABLE IF EXISTS is_user_not_in_list_able;
7 DROP TABLE IF EXISTS unwelcome_txt;
8 DROP TABLE IF EXISTS error_cnt;
9
10 -- Create user_data table
11 CREATE TABLE user_data (
12     id BIGINT PRIMARY KEY GENERATED ALWAYS AS IDENTITY,
13     user_id TEXT,
14     is_whitelisted BOOLEAN DEFAULT TRUE,
15     created_at TIMESTAMP DEFAULT NOW(),
```

어시스턴트에게 기존의 FTP 파일 구조 설명 → 어시스턴트가 SQL 코드 작성  
→ 코드 적절하게 수정







# SQL OF SD4LE

```
54 -- Insert data into user_data table
55 INSERT INTO user_data (user_id) VALUES ('20770000');
56 INSERT INTO user_data (user_id) VALUES ('20780000');
57 INSERT INTO user_data (user_id) VALUES ('20790000');
58 INSERT INTO user_data (user_id) VALUES ('20800000');
59 INSERT INTO user_data (user_id) VALUES ('20810000');
60
61 -- Insert data into latest_version table
62 INSERT INTO latest_version (version) VALUES ('1.1.0');
63
64 -- Insert data into is_user_not_in_list_able table
65 INSERT INTO is_user_not_in_list_able (status) VALUES (1);
66
67 -- Insert data into unwelcome_txt table
68 INSERT INTO unwelcome_txt (message) VALUES ('프로그램 사용 권한이 없습니다.\n개발자에게 문의하세요');
```

어시스턴트에게 기존의 FTP 파일 구조 설명 → 어시스턴트가 SQL 코드 작성  
→ 코드 적절하게 수정

**SD4LE WITH SUPABASE**

# SD4LE WITH SUPABASE

파일명	크기	파일 유형	최종 수정 ^
..			
 unwelcome_txt.dat	77	Probe D...	2024-05-02 오후 11:25:06
 is_user_not_in_list_able....	1	Probe D...	2024-05-02 오후 11:30:06
 lastest_version.dat	5	Probe D...	2024-05-03 오전 12:08:53
 user_data.dat	204	Probe D...	2024-05-03 오후 2:53:59

지금까지 사용했던 파일들-

# SD4LE WITH SUPABASE

user\_data

♂️

◆

#

id

int8

◇

user\_id

text

◇

is\_whitelisted

bool

◇

created\_at

timestamp

◇

updated\_at

timestamp

◇

last\_used\_at

timestamp

latest\_version

♂️

◆

#

id

int4

◇

version

text

◇

updated\_at

timestamp

is\_user\_not\_in\_list\_able

♂️

◆

#

id

int4

◇

status

int4

◇

updated\_at

timestamp

unwelcome\_txt

♂️

◆

#

id

int4

◇

message

text

◇

updated\_at

timestamp

error\_cnt

♂️

◆

#

id

int4

◇

error\_cnt

int4

◇

updated\_at

timestamp

이것으로 대체되었습니다.

**FETCH TABLE DATA**

# FETCH TABLE DATA

```
1 from supabase import create_client, Client
2
3 # Supabase 프로젝트 URL과 API 키 설정
4 supabase_url = '(프로젝트 URL)'
5 supabase_key = '(API 키)'
6
7 # Supabase 클라이언트 생성
8 supabase: Client = create_client(supabase_url, supabase_key)
9
10 # 테이블 데이터 조회 함수
11 def fetch_table_data(table_name):
12     response = supabase.table(table_name).select("*").execute()
13     return response.data
14
15 # 각 테이블의 데이터 조회 및 출력
```

# FETCH TABLE DATA

```
9
10 # 테이블 데이터 조회 함수
11 def fetch_table_data(table_name):
12     response = supabase.table(table_name).select("*").execute()
13     return response.data
14
15 # 각 테이블의 데이터 조회 및 출력
16 tables = ['user_data', 'latest_version', 'is_user_not_in_list_able', 'unwelcome_txt', 'error_cr
17
18 for table in tables:
19     data = fetch_table_data(table)
20     print(f"Data from {table}:")
21     for row in data:
22         print(row)
23     print("\n")
```

# FETCH TABLE DATA

Data from user\_data:

```
{'id': 1, 'user_id': '20770000', 'is_whitelisted': True, 'created_at': '2024-05-29T14:54:32.457822', 'updated_at': '2024-05-29T14:54:32.457822', 'last_used_at': None}
{'id': 2, 'user_id': '20780000', 'is_whitelisted': True, 'created_at': '2024-05-29T14:54:32.457822', 'updated_at': '2024-05-29T14:54:32.457822', 'last_used_at': None}
{'id': 3, 'user_id': '20790000', 'is_whitelisted': True, 'created_at': '2024-05-29T14:54:32.457822', 'updated_at': '2024-05-29T14:54:32.457822', 'last_used_at': None}
{'id': 4, 'user_id': '20800000', 'is_whitelisted': True, 'created_at': '2024-05-29T14:54:32.457822', 'updated_at': '2024-05-29T14:54:32.457822', 'last_used_at': None}
{'id': 5, 'user_id': '20810000', 'is_whitelisted': True, 'created_at': '2024-05-29T14:54:32.457822', 'updated_at': '2024-05-29T14:54:32.457822', 'last_used_at': None}
```

Data from latest\_version:

```
{'id': 1, 'version': '1.1.0', 'updated_at': '2024-05-29T14:54:32.457822'}
```

Data from is\_user\_not\_in\_list\_able:

```
{'id': 1, 'status': 1, 'updated_at': '2024-05-29T14:54:32.457822'}
```

Data from unwelcome\_txt:

```
{'id': 1, 'message': '프로그램 사용 권한이 없습니다.\r\n개발자에게 문의하세요.', 'updated_at': '2024-05-29T14:54:32.457822'}
```

Data from error\_cnt:

```
{'id': 1, 'error_cnt': 0, 'execution_cnt': 0, 'updated_at': '2024-05-29T14:54:32.457822'}
```



**UPDATE TABLE DATA**

# UPDATE TABLE DATA

```
1 # latest_version 데이터 업데이트 함수
2 def update_latest_version(new_version):
3     current_time = datetime.now().isoformat()
4     response = supabase.table('latest_version').update({'version': new_version, 'updated_at': cu
5     return response.data
6
7 # version 데이터 업데이트
8 new_version = '1.2.0'
9 update_response = update_latest_version(new_version)
10 print("Updated latest_version:")
11 print(update_response)
12
13 # error_cnt & execution_cnt 값 1 증가 함수
14 def increment_error_cnt():
15     current_time = datetime.now().isoformat()
```

# UPDATE TABLE DATA

```
10 print("Updated latest_version:")
11 print(update_response)
12
13 # error_cnt & execution_cnt 값 1 증가 함수
14 def increment_error_cnt():
15     current_time = datetime.now().isoformat()
16     new_error_cnt = fetch_table_data('error_cnt')[0]['error_cnt'] + 1
17     new_execution_cnt = fetch_table_data('error_cnt')[0]['execution_cnt'] + 1
18     response = supabase.table('error_cnt').update({'error_cnt': new_error_cnt, 'execution_cnt': new_execution_cnt})
19     return response.data
20
21 # error_cnt & execution_cnt 데이터 업데이트
22 update_response = increment_error_cnt()
23 print("\nIncremented error_cnt:")
24 print(update_response)
```

# UPDATE TABLE DATA

Updated latest\_version:

```
[{'id': 1, 'version': '1.2.0', 'updated_at': '2024-05-29T16:03:17.645521'}]
```

Updated error\_cnt:

```
[{'id': 1, 'error_cnt': 3, 'execution_cnt': 3, 'updated_at': '2024-05-29T16:03:17.822284'}]
```

**감사합니다**