
제5회 FOSSLight Community Day

코딩 머글이 FOSSLight를 고쳤어요!

LINE+ Open Source Program Office

이서연

우리 팀을 소개합니다.

(내맘대로 머글이라고 불러서 미안.. 근데 맞잖아 ...)

머글 1

컴퓨터 공학 전공
그러나 개발자로서의 경험 없음
오로지 OSPO, 그 뿐....

머글 2

컴퓨터 공학 전공
그러나 개발자로서의 경험 없음
오로지 OSPO, 그 뿐....

머글 3

독일어 전공 귀염둥이

우리의 FOSSLight를 소개합니다

인프라

Web app: VM Rocky Linux

DB: Mysql

(한번 아찔한 사고가 있었어서 DB를 분리 운영 / 매일 백업)

Version - Before

1.6.1

(~~많이 아픈 손가락~~)

(~~그치만 손을 쓸수 없던...~~)



우리의 FOSSLight를 소개합니다

인프라

Web app: VM Rocky Linux
DB: Mysql

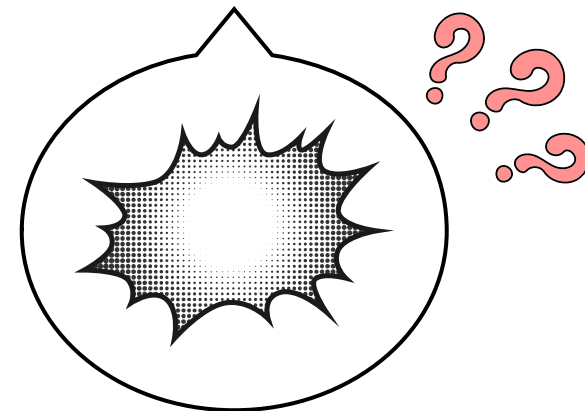
Version - Before

1.6.1



Version - After

2.3.0



어떻게? - 1

역할 분담

머글 1

DB 업그레이드 담당

머글 2

웹앱 업그레이드 담당

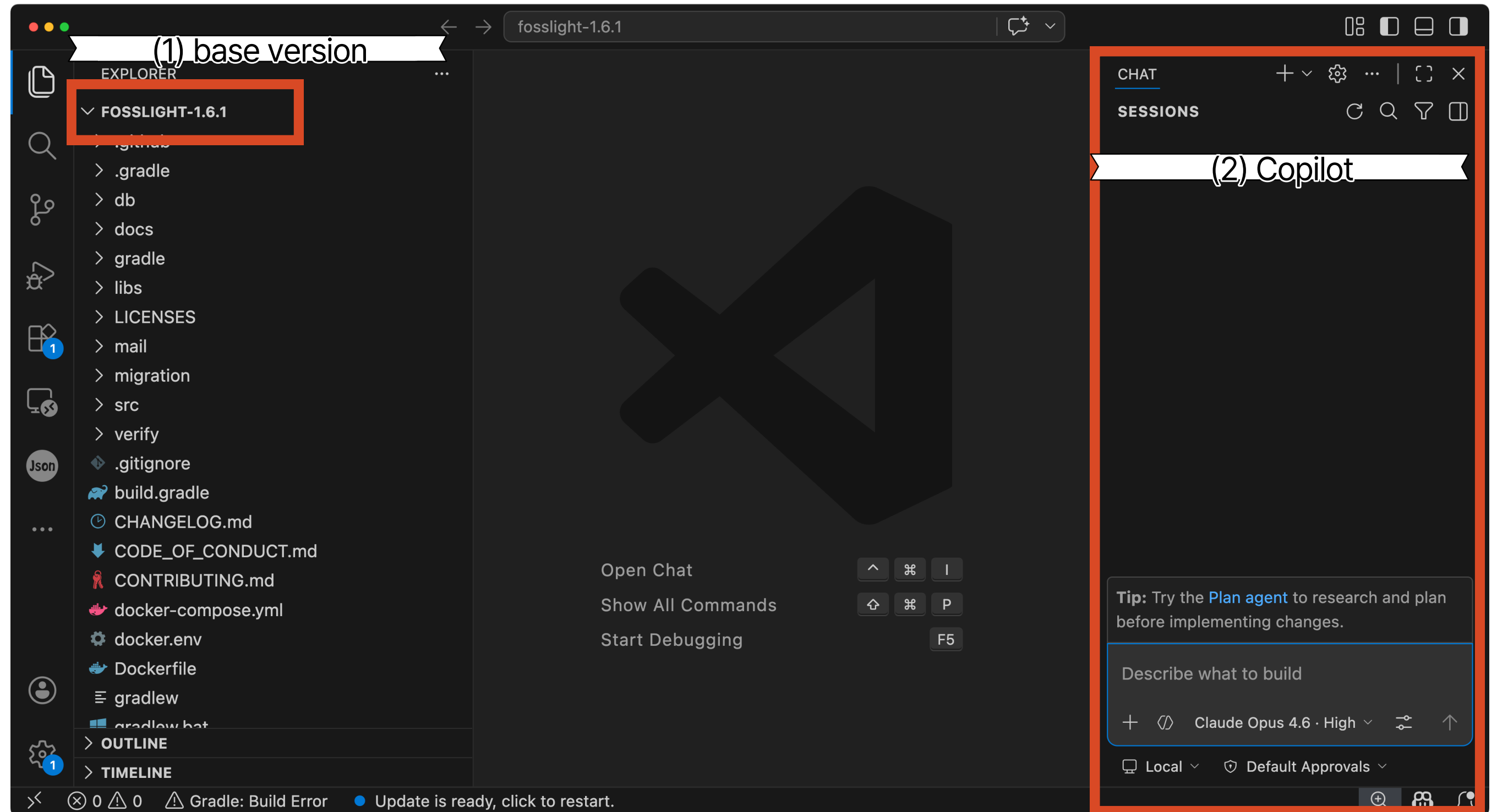
머글 3

기능 테스트



어떻게? - 2

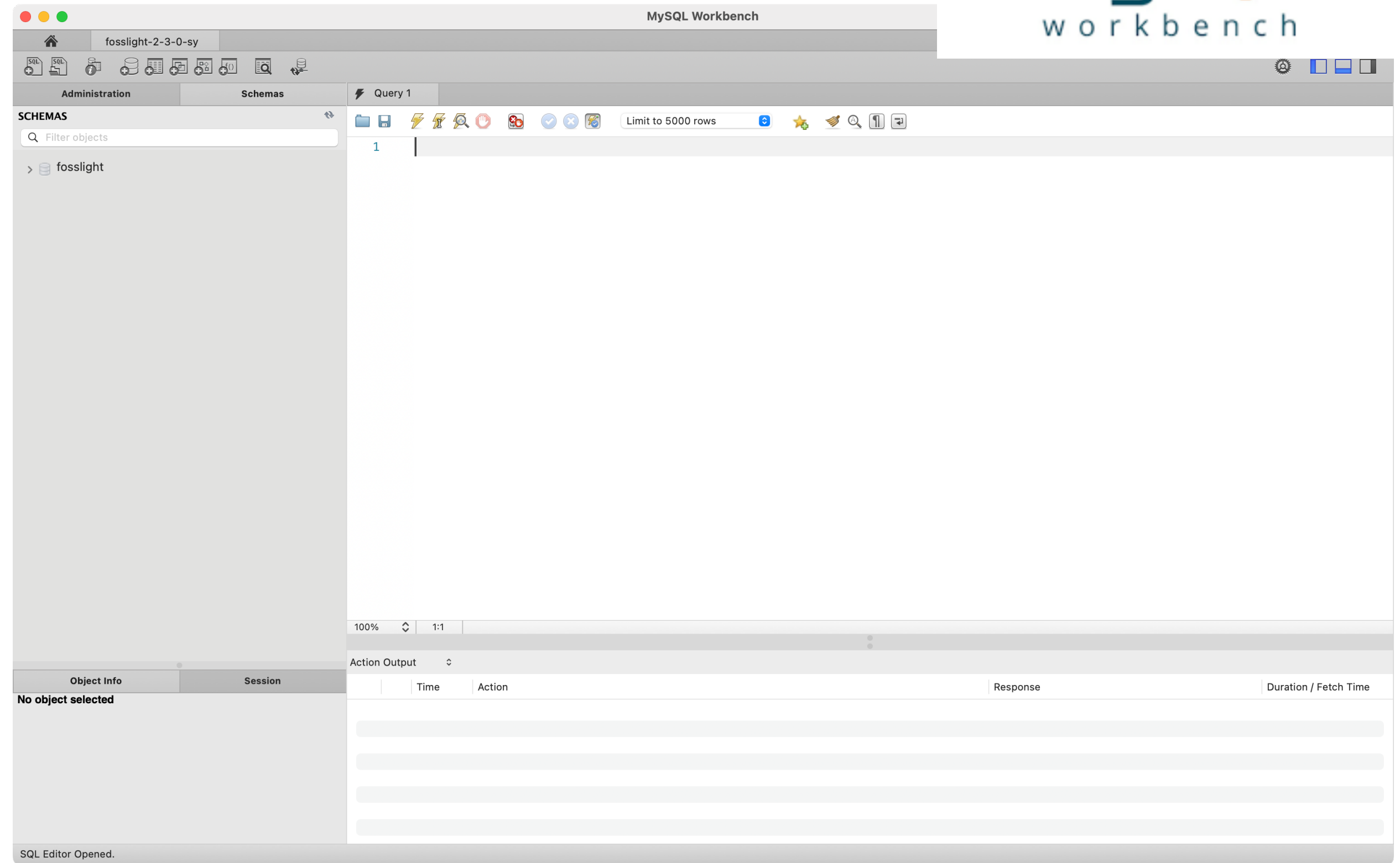
작업 전 기본 셋팅





어떻게? - 2

작업 전 기본 셋팅



어떻게? - 3

기본 작업 흐름

STEP 1

- 기존 DB 접속
 - 기존 스키마 dump (백업)
 - 마이그레이션 쿼리 실행
(migration/migration/scripts/20240725150921_update_v2.0.0.sql)
 - 마이그레이션한 스키마 dump



STEP 2

- 새 DB 접속
 - 초기화
 - 초기화 쿼리 실행
(db/initdb.d/fosslight_create.sql)
 - 백업 dump를 import
 - 문제가 있는지 확인



STEP 3

- 문제 발생시
 - Copilot을 이용해서 원인 파악
 - "create.sql을 실행했는데 mysql에서 이런 에러가 났어. 원인이 뭘까?"
- 해결 방안을 확인하여 쿼리 수정



Visual Studio Code

다시 돌아가서 반복

문제 해결 사례 1

SEQUENCE 문법 (v2.1.1)

문제 발생

```
ERROR 1064 (42000) at line 5634: You have an error in your SQL syntax;
check the manual that corresponds to your MySQL server version for the
right syntax to use near 'CREATE SEQUENCE NEXT_FILE_ID START WITH 1
INCREMENT BY 1' at line 1
```

[fosslight](#) / [db](#) / [initdb.d](#) / [fosslight_create.sql](#)

Code

Blame

Executable File · 5681 lines (5541 loc) · 661 KB

5633

... 5634 **CREATE SEQUENCE NEXT_FILE_ID** START WITH 1 INCREMENT BY 1;|

5635

문제 해결 사례 1

SEQUENCE 문법

원인 파악

← MYSQL SYNTAX ERROR TROUBLESHOOTING

Start Over

이 쿼리를 mysql에서 실행하는 도중에 이런 에러가 발생했어. 원인이 뭔지 확인해줘.

You have an error in your SQL syntax;
check the manual that corresponds to your MySQL server version for the right syntax to
use near 'CREATE SEQUENCE NEXT_FILE_ID START WITH 1 INCREMENT BY 1' at line 1

fosslight_create.sql

문제 해결 사례 1

SEQUENCE 문법

원인 파악

Analyzed SQL syntax error in MySQL execution

`CREATE SEQUENCE` 는 MySQL에서 지원하지 않는 구문입니다.

이 SQL 파일은 MariaDB 10.5.8에서 생성된 것으로(파일 상단 주석 참조), MariaDB는 10.3부터 `CREATE SEQUENCE` 를 지원하지만 MySQL은 이 구문을 지원하지 않습니다.

해결 방법:

1. MariaDB를 사용 — 이 프로젝트의 `docker-compose.yml` 도 MariaDB를 기반으로 구성되어 있으므로, MySQL 대신 MariaDB를 사용하는 것이 권장됩니다.
2. MySQL에서 실행해야 한다면 — `CREATE SEQUENCE` 구문을 제거하고, 대신 `AUTO_INCREMENT` 컬럼이 있는 테이블로 시퀀스를 대체할 수 있습니다:

```
-- CREATE SEQUENCE 대신:  
CREATE TABLE NEXT_FILE_ID_SEQ (  
  id BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY  
) AUTO_INCREMENT = 1;
```

문제 해결 사례 1

SEQUENCE 문법

원인 파악

Analyzed SQL syntax error in MySQL execution

`CREATE SEQUENCE` 는 MySQL에서 지원하지 않는 구문입니다.

이 SQL 파일은 MariaDB 10.5.8에서 생성된 것으로(파일 상단 주석 참조), MariaDB는 10.3부터 `CREATE SEQUENCE` 를 지원하지만 MySQL은 이 구문을 지원하지 않습니다.

해결 방법:

1. MariaDB를 사용 — 이 프로젝트의 `docker-compose.yml` 도 MariaDB를 기반으로 구성되어 있으므로, MySQL 대신 MariaDB를 사용하는 것이 권장됩니다.
2. MySQL에서 실행해야 한다면 — `CREATE SEQUENCE` 구문을 제거하고, `BIGINT` 컬럼이 있는 테이블로 시퀀스를 대체할 수 있습니다:

```
-- CREATE SEQUENCE 대신:  
CREATE TABLE NEXT_FILE_ID_SEQ (  
  id BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY  
) AUTO_INCREMENT = 1;
```

해봤지만
소용 없음...

문제 해결 사례 1

SEQUENCE 문법

원인 파악

Analyzed SQL syntax error in MySQL execution

`CREATE SEQUENCE` 는 MySQL에서 지원하지 않는 구문입니다.

이 SQL 파일은 MariaDB 10.5.8에서 생성된 것으로(파일 상단 주석 참조), MariaDB는 `CREATE SEQUENCE` 를 지원하지만 MySQL은 이 구문을 지원하지 않습니다.

해결 방법:

1. MariaDB를 사용 — 이 프로젝트의 `docker-compose.yml` 파일을 보면 MariaDB를 사용하는 것이 권장됩니다.
2. MySQL에서 실행해야 한다면 — `CREATE SEQUENCE` 구문을 지원하지 않는 테이블로 시퀀스를 대체할 수 있습니다:

```
-- CREATE SEQUENCE 대신:  
CREATE TABLE NEXT_FILE_ID_SEQ (  
  id BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY  
) AUTO_INCREMENT = 1;
```


그러나 힌트를 얻을 수 있었음!
이 쿼리는
NEXT_FILE_ID_SEQ라는 테이블에서
id라는 것을 1씩 증가시켜주는 것이구나!

문제 해결 사례 1

SEQUENCE 문법

원인 파악

이 라인을 주석처리하고 웹 앱 레벨에서 해결할 수 없을까? 그러니까 id라는 값이 1씩 증가하도록 저장하면 되잖아.

 fosslight_create.sql

문제 해결 사례 1

SEQUENCE 문법

원인 파악

Searched for NEXT_FILE_ID and getFileId, reviewed 2 files

현재 구조를 정리하면:

1. DB 레벨: MariaDB의 `CREATE SEQUENCE NEXT_FILE_ID` 로 시퀀스를 만들고, `NEXTVAL(NEXT_FILE_ID)` 로 다음 ID를 가져옴
2. 앱 레벨: `FileMapper.xml:129-132` 에서 `NEXTVAL()` 을 호출

흥미롭게도, 이미 주석 처리된 대안 코드가

`FileMapper.xml:131-132` 에 있습니다:

```
<select id="getFileId" resultType="java.lang.Str
    SELECT NEXTVAL(NEXT_FILE_ID)
<!--      SELECT IFNULL(MAX(FILE_ID), 0)+1 AS FILE
          FROM T2_FILE -->
</select>
```

문제 해결 사례 1

SEQUENCE 문법

문제 해결

즉, 웹 앱 레벨에서 해결 가능합니다. 변경 방법은:

1. DB: CREATE SEQUENCE 줄은 이미 주석 처리되어 있으므로 그대로 둬. 마이그레이션 스크립트(`20241108014959_update_v2.1.1.sql:78-83`)의 시퀀스 관련 줄도 주석 처리 필요.
2. `FileMapper.xml` : 현재/주석을 뒤집으면 됨:

```
<select id="getFileId" resultType="java.lang.Str
<!-- SELECT NEXTVAL(NEXT_FILE_ID) -->
      SELECT IFNULL(MAX(FILE_ID), 0)+1 AS FILE_ID
      FROM T2_FILE
</select>
```

fosslight / src / main / resources / oss / fosslight / repository / FileMapper.xml

Code

Blame

349 lines (316 loc) · 9.75 KB

```
128
129     <select id="getFileId" resultType="java.lang.String">
130         SELECT NEXTVAL(NEXT_FILE_ID)
131     <!-- SELECT IFNULL(MAX(FILE_ID), 0)+1 AS FILE_ID
132         FROM T2_FILE -->
133     </select>
```

이걸 남겨주신분은 천사이신가요

아 물론 코파일럿 너도 잘했지~

문제 해결 사례 2

인덱스 길이 문제 (v2.1.1)

문제 발생

```
ERROR 1071 (42000) at line 4: Specified key was too long; max key length is 767 bytes
```

[fosslight](#) / [migration](#) / [migration](#) / [scripts](#) / [20241108014959_update_v2.1.1.sql](#) 

 FOSSLight-dev Fix migration script

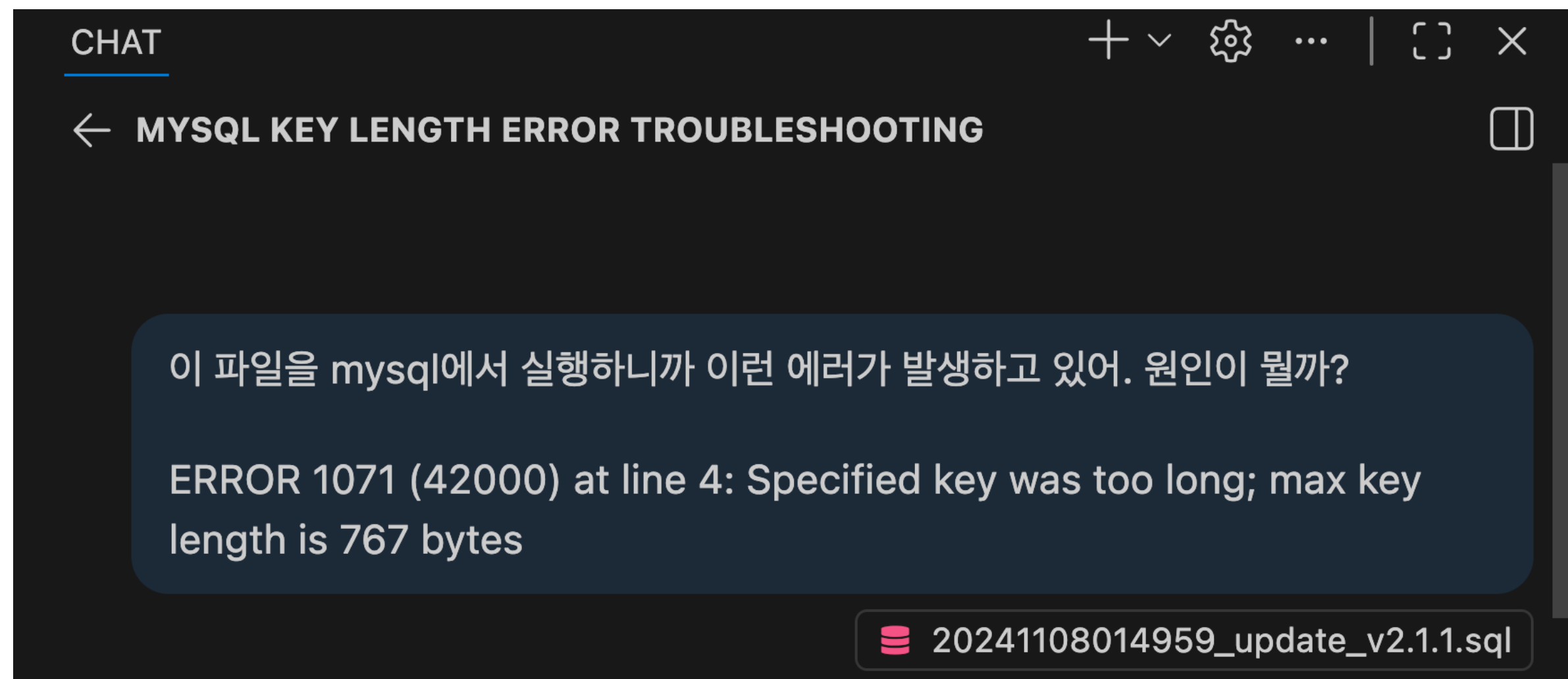
Code Blame 83 lines (69 loc) · 5.63 KB

```
1 -- // update_v2.1.1
2 -- Migration SQL that makes the change goes here.
3
4 ALTER TABLE `OSS_DOWNLOADLOCATION` DROP INDEX `DOWNLOAD_LOCATION`;
5 ALTER TABLE `OSS_DOWNLOADLOCATION` ADD INDEX `DOWNLOAD_LOCATION` (`DOWNLOAD_LOCATION`);
6
```

문제 해결 사례 2

인덱스 길이 문제

원인 파악



문제 해결 사례 2

인덱스 길이 문제

원인 파악

```
← MYSQL KEY LENGTH ERROR TROUBLESHOOTING
테이블 정의를 확인해보겠습니다.

Searched for text DOWNLOAD_LOCATION ( **/fossilight_create.sql ), 16 results

OSS_DOWNLOADLOCATION 테이블의 DOWNLOAD_LOCATION 컬럼이 ** varchar(2000) **으
로 정의되어 있습니다.

원인 요약:

MySQL InnoDB에서 인덱스 키의 최대 길이는 기본적으로 767 bytes입니다. utf8mb4 문
자셋을 사용하면 한 글자당 4바이트이므로:

    varchar(2000) × 4 = 8000 bytes >> 767 bytes

utf8 이더라도 2000 × 3 = 6000 bytes로 여전히 초과합니다.
```

뭐요..?

문제 해결 사례 2

인덱스 길이 문제

원인 파악

해결 방법은 두 가지입니다:

1. **prefix index** 사용 — 인덱스를 컬럼 앞부분 일부만 걸도록 변경:

```
ALTER TABLE `OSS_DOWNLOADLOCATION` ADD INDEX `DOWNLOAD_LOCATION`
```

2. **innodb_large_prefix** 활성화 (MySQL 5.6 이하인 경우) — **ROW_FORMAT**을 **DYNAMIC** 또는 **COMPRESSED** 로 변경하고 **innodb_large_prefix=ON** 설정. MySQL 5.7.7+에서는 기본 활성화되어 있지만, **ROW_FORMAT**이 **COMPACT** / **REDUNDANT** 이면 여전히 767 제한이 적용됩니다.

실제로 `fossight_create.sql:2132` 의 초기 테이블 정의에서도 동일하게 **KEY** **DOWNLOAD_LOCATION (DOWNLOAD_LOCATION)** 으로 되어 있어서, 신규 설치 시에도 **ROW_FORMAT**이 맞지 않으면 같은 에러가 발생할 수 있습니다.

가장 간단한 수정은 migration SQL의 line 4를 **prefix index**로 바꾸는 것입니다.



Claude Opus 4.6 · 3x

제가요.....??????????????????

문제 해결 사례 2

인덱스 길이 문제

원인 파악

해결 방법은 두 가지입니다:

1. **prefix index** 사용 — 인덱스를 컬럼 앞부분 일부만 걸도록 변경:

```
ALTER TABLE `OSS_DOWNLOADLOCATION` ADD INDEX `DOWNLOAD_LOCATION`
```

2. **innodb_large_prefix** 활성화 (MySQL 5.6 이하인 경우) — **ROW_FORMAT**을 **DYNAMIC** 또는 **COMPRESSED** 로 변경하고 **innodb_large_prefix** 설정. MySQL 5.7.7+에서는 기본 활성화되어 있지만, **ROW_FORMAT=DYNAMIC** 이면 여전히 767 제한이 적용됩니다.

실제로 `fosslight_create.sql:2132` 의 초기 테이블 `DOWNLOAD_LOCATION (DOWNLOAD_LOCATION)` 으로 `ROW_FORMAT`이 맞지 않으면 같은 에러가 발생할 수 있습니다.

일단 백업이 있으니까
해보기나 할까...

가장 간단한 수정은 migration SQL의 line 4를 **prefix index**로 바꾸는 것입니다.



Claude Opus 4.6 • 3x

제가요.....??????????????????

문제 해결 사례 2

인덱스 길이 문제

원인 파악

이걸 이렇게 자른다고 해도 데이터 읽어오는데에 문제는 없을까?

2초 동안 생각함

대체로 읽어오는 데 문제는 없습니다. 다만 인덱스의 효율과 제약 의미가 달라집니다.

핵심은 이겁니다:

- 데이터 자체는 컬럼 전체값이 그대로 저장됩니다.
- (768) 은 인덱스에 앞 768자만 저장하겠다는 뜻입니다.
- 그래서 `SELECT DOWNLOAD_LOCATION ...` 으로 값을 읽으면 전체 문자열이 정상적으로 나옵니다.
- 잘리는 것은 데이터가 아니라 인덱스 키입니다.

호오오오옥시 모르니까
다른 친구에게도 물어보기

문제 해결 사례 2

인덱스 길이 문제

문제 해결

Line 4 주석처리:

```
-- ALTER TABLE `OSS_DOWNLOADLOCATION` DROP INDEX `DOWNLOAD_LOCATION`;
```

Line 5 수정 (인덱스 길이 제한 추가):

-- 수정 전:

```
-- ALTER TABLE `OSS_DOWNLOADLOCATION` ADD INDEX `DOWNLOAD_LOCATION` (`DOWNLOAD_LOCATION`);
```

-- 수정 후:

```
ALTER TABLE `OSS_DOWNLOADLOCATION` ADD INDEX `DOWNLOAD_LOCATION` (`DOWNLOAD_LOCATION` (768));
```

이유:

- MySQL의 인덱스 길이 제한(최대 767 bytes for InnoDB)을 고려
- DOWNLOAD_LOCATION 컬럼이 긴 문자열을 포함할 수 있어 prefix 인덱스 사용

결과 1

버전 업그레이드 완료

업그레이드 이력

1.6.1 → 2.0.0 pre → 2.0.0 → 2.0.2 → 2.1.0 → 2.1.1 → 2.2.0 → 2.3.0

결과 2

FOSSLight에 기여!

이번 작업 중 기여 이력

 **Fix typo: gird to grid** ✓

#1123 by syleeeee Contributor was merged on Feb 24  1 of 5 tasks

 **Fix migration script that deletes table data** ✓

#1110 by syleeeee Contributor was merged on Dec 12, 2025  1 of 5 tasks

 **Remove migration script that deletes table data** ✓

#1109 by syleeeee Contributor was closed on Dec 1, 2025  1 of 5 tasks

 **Fix bug in migration scripts** ✓ bug fix

#1107 by syleeeee Contributor was merged on Nov 28, 2025  1 of 5 tasks

 **Fix OSS Rename performance issues and Project Search UI bugs** ✓

#1114 by dongs0 was merged on Dec 12, 2025  1 of 5 tasks

 **fix: improve OSS nickname search and autocomplete** ✓

#1113 by dongs0 was merged on Dec 12, 2025  1 of 5 tasks

 **fix: improve UI grid refresh and search functionality** ✓

#1112 by dongs0 was closed on Dec 8, 2025  5 tasks

결론

이게 어떻게 가능했나?

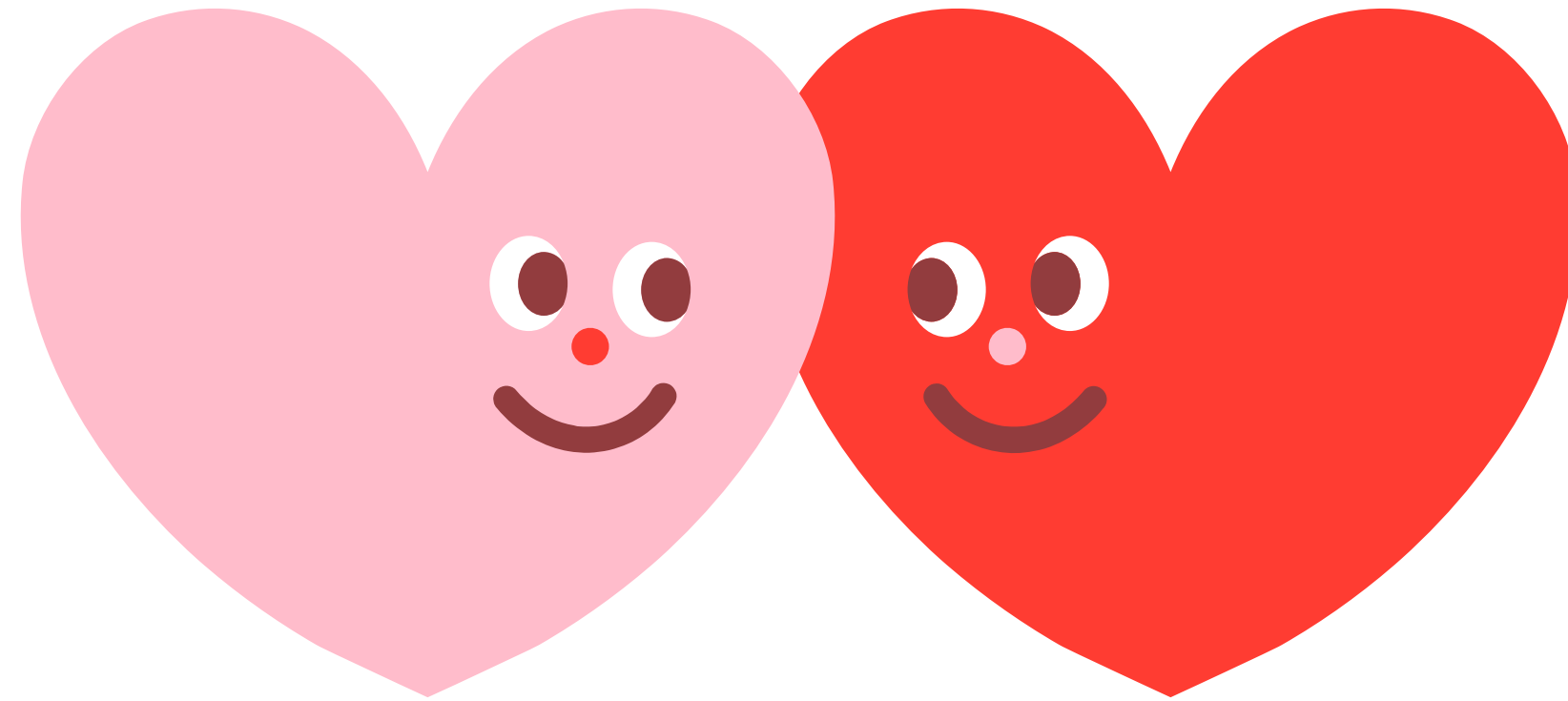
01. 이해할 때까지 물어보자

INNODB? 몰라도 됩니다.
확신이 없으면 여러번 물어봅시다.
여러 AI에게 물어봐서 검증하고 이해합시다.
내가 이해할 때까지!!!

02. 오픈소스 오픈소스!

오픈소스가 아니었다면?
아예 원인 파악을 할 엄두도 안냈을지도...
오픈소스니까 AI에게 보여주고 원인 파악도 시키고
물음표 살인마랑 가두코
자체 원인 파악, 재시도가 가능!
심지어 기여하고 뿌듯하기 가능!!!

결론



앞으로도 업그레이드 때마다 잘 부탁드립니다 ♥
오픈소스로 유지해주시느라 늘 수고가 많으십니다!
PR 리뷰 해주세용
