

# 카카오톡 대화 분석기

## 1. 프로그램 소개

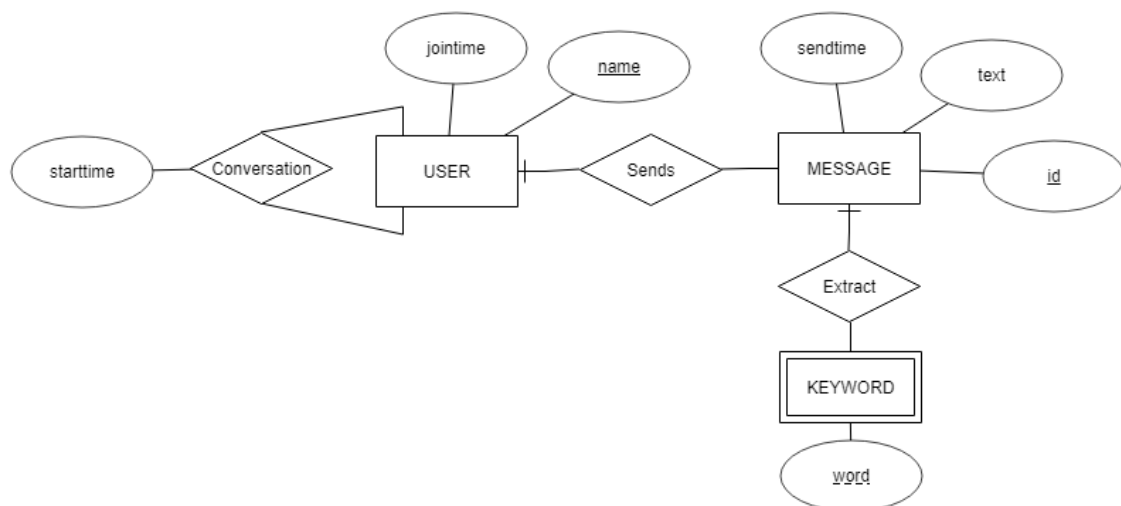
데이터베이스를 응용해 어떤 프로그램을 만들어볼까 고민하던중 카카오톡 대화 분석이라는 주제가 떠올랐다. 주변에서 흔하게 찾아볼 수 있는 빅데이터중 하나이고 분석을 통해 사람간의 관계를 어느 정도 유추할 수 있을 것이라는 생각이 들었다. 기존의 카카오톡 분석 프로그램중 유명한 것은 TEXTAT과 Kakanalyzer이 있다. 두 프로그램 모두 안드로이드 어플리케이션이다. 그런데 TEXTAT은 일대일 대화를 기반으로 감정을 분석하는 프로그램이고 Kakanalyzer는 단톡방의 분석을 지원하지만 주어지는 정보가 점유율과 가장 많이 사용된 단어밖에 없다. 이번에 내가 만드는 프로그램은 점유율, 키워드를 포함해 특정 기간 내의 점유율, 시간대별 점유율, 가장 톡방이 활발했던 날짜를 확인할 수 있는 기능을 제공한다. 또한 특정 사람의 대화를 누가 많이 받아줬는지, 또 특정 사람이 누구의 대화를 잘 받아줬는지도 분석할 수 있다. 이를 통해 단톡방 멤버들의 친밀도도 어느 정도 파악할 수 있다.

## 2. 개발 환경

Python 3.6 + konlpy 모듈 0.4버전 + MySQL 5.7버전을 이용했다. konlpy는 자연어처리를 통해 문장에서 키워드를 추출하기 위해 사용했다.

## 3. ER 다이어그램, Relation, MySQL

E/R diagram은 아래와 같다.



Entity set USER는 사용자의 정보가 들어있다. 외부적인 정보 없이 대화 로그로만 알 수 있는 정보를 통해 USER의 attribute를 정하려다보니 이름(name)과 최초로 챗을 보낸 시간(jointime)이 결정되었다. 또한 그렇기 때문에 단톡방에 있더라도 한 번도 채팅을 하지 않았을 경우 유저로 등록되지 않는다. key는 이름이다.

Entity set MESSAGE는 메시지의 내용(text)과 메시지를 전송한 시간(sendtime)과 그 메시지의 ID가 들어있다. ID는 시간 순으로 1번부터 차례로 부여한다. key는 ID이다.

Entity set KEYWORD는 MESSAGE에 종속되어있는 weak entity set이다. 단어(word)를 attribute로 가지고 key는 단어(word)와 메시지의 ID이다.

Relation Conversation은 USER 내부의 Many to Many relation이다. 참가자 A가 톡을 보낸 후, 10분내로 A가 아닌 다른 참가자 B가 톡을 보냈을 경우 A와 B 사이에 대화가 이루어졌다고 판단한다. 이 Relation은 대화의 시작 시간(starttime)을 attribute로 가진다.

Relation Sends는 One to Many relation이다. 메시지를 누가 작성했는지의 관계를 나타낸다.

Relation Extract는 One to Many relation이다. 단어가 어떤 메시지로부터 추출되었는지의 관계를 나타낸다.

이 ER diagram을 토대로 변환된 Relation들에 대해 FD, BCNF, 3NF를 따져보겠다.

i) MESSAGE(id, text, sendtime)

nontrivial FD는 id → text, sendtime이 존재한다. (id)는 super key이므로 BCNF, 3NF를 만족한다. (카카오톡 대화 로그는 분 단위로만 시간을 기록하기 때문에 1분 동안에 동일한 사람에게서 동일한 내용이 전송될 수 있다. 그렇기 때문에 sendtime이 key가 될 수 없는 것이다.)

ii) USER(name, jointime)

nontrivial FD는 name → jointime이 존재한다. (name)은 super key이므로 BCNF, 3NF를 만족한다.

iii) KEYWORD(word, id)

nontrivial FD가 존재하지 않는다. BCNF, 3NF를 만족한다.

iv) CONVERSATION(sender, receiver, starttime)

애초에 중복된 sender, receiver, starttime가 들어올 수 있기 때문에 key도 없고 FD 또한 없다.

MESSAGE와 같이 id를 도입하면 key를 만들 수 있겠지만 CONVERSATION table은 프로그램 내에서 횟수를 셀 때에만 쓰이기 때문에 굳이 id를 도입할 필요를 못 느꼈고, 최종적으로 key가 없는 형태로 두었다.

v) SENDS(name, id)

nontrivial FD가 존재하지 않는다. BCNF, 3NF를 만족한다.

실제로 MySQL에서는

```

1 • CREATE table USER(
2     name nvarchar(20),
3     jointime datetime,
4     PRIMARY KEY(name)
5 );
6 • CREATE table MESSAGE(
7     id int,
8     msgtext nvarchar(500),
9     sendtime datetime,
10    PRIMARY KEY(id)
11 );
12 • CREATE table KEYWORD (
13     word nvarchar(40),
14     id int,
15     FOREIGN KEY (id) REFERENCES MESSAGE(id),
16     PRIMARY KEY(word, id)
17 );
18 • CREATE table SENDS(
19     sender nvarchar(20),
20     id int,
21     FOREIGN KEY (id) REFERENCES MESSAGE(id)
22 );
23 • CREATE table CONVERSATION(
24     sender nvarchar(20),
25     receiver nvarchar(20),
26     starttime datetime
27 );

```

으로 구현했다.

#### 4. 프로그램 작동 방식

사용자는 우선 프로그램을 구동하기 전, 카카오톡에서 대화 내보내기 기능을 통해 아래와 같은 대화 로그를 미리 파일로 저장해두고 있어야 한다.

```

2016년 2월 5일 오전 11:20, [프로젝트] : ㅇ
2016년 2월 5일 오전 11:20, [프로젝트] : 약
2016년 2월 5일 오전 11:20, [프로젝트] : 특이 4개야
2016년 2월 5일 오전 11:21, [프로젝트] : 공지록 잡담록 16톡
2016년 2월 5일 오전 11:21, [프로젝트] : [프로젝트] : [프로젝트]

```

또한 데이터베이스에 미리 table들을 create 해두어야 한다.

프로그램이 시작되면 우선 ODBC 드라이버를 이용해 현재 컴퓨터에서 실행중인 데이터베이스와 연동한다. 그리고 konlpy 모듈을 로딩한다. 이후에 메뉴를 출력한다.

```
카카오톡 대화 분석기
로딩중...
```

- 
1. 새로운 대화 불러오기
  2. 특정 기간 내의 대화 참여자 랭킹 보기
  3. 시간대별 대화 참여자 랭킹 보기
  4. 특정 기간 내의 키워드 랭킹 보기
  5. 가장 톡방이 활발했던 날짜 보기
  6. 특정 사람의 대화를 받아준 사람의 랭킹 보기
  7. 특정 사람이 누구의 대화를 잘 받아줬는지에 대한 랭킹 보기
  8. 불러온 대화 제거하기
- 

```
>
```

이제 사용자에게 입력을 받는데, 1~8이 아닌 다른 문자를 입력할 경우 잘못 입력했다고 사용자에게 알려준다.

- 
1. 새로운 대화 불러오기
  2. 특정 기간 내의 대화 참여자 랭킹 보기
  3. 시간대별 대화 참여자 랭킹 보기
  4. 특정 기간 내의 키워드 랭킹 보기
  5. 가장 톡방이 활발했던 날짜 보기
  6. 특정 사람의 대화를 받아준 사람의 랭킹 보기
  7. 특정 사람이 누구의 대화를 잘 받아줬는지에 대한 랭킹 보기
  8. 불러온 대화 제거하기
- 

```
> sdf
잘못 입력했습니다.
```

- 
1. 새로운 대화 불러오기
  2. 특정 기간 내의 대화 참여자 랭킹 보기
  3. 시간대별 대화 참여자 랭킹 보기
  4. 특정 기간 내의 키워드 랭킹 보기
  5. 가장 톡방이 활발했던 날짜 보기
  6. 특정 사람의 대화를 받아준 사람의 랭킹 보기
  7. 특정 사람이 누구의 대화를 잘 받아줬는지에 대한 랭킹 보기
  8. 불러온 대화 제거하기
- 

```
> 9
잘못 입력했습니다.
```

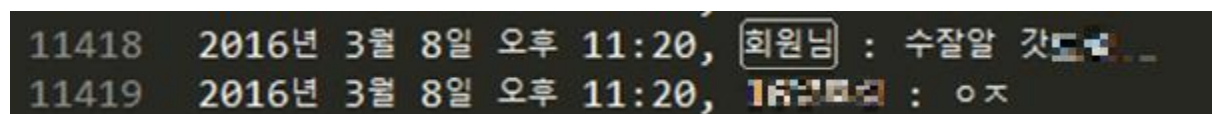
- 
1. 새로운 대화 불러오기
  2. 특정 기간 내의 대화 참여자 랭킹 보기
  3. 시간대별 대화 참여자 랭킹 보기
  4. 특정 기간 내의 키워드 랭킹 보기
  5. 가장 톡방이 활발했던 날짜 보기
  6. 특정 사람의 대화를 받아준 사람의 랭킹 보기
  7. 특정 사람이 누구의 대화를 잘 받아줬는지에 대한 랭킹 보기
  8. 불러온 대화 제거하기
- 

```
>
```

만약 대화를 불러오지 않은 채로 2-7의 기능을 사용하려고 할 경우 로딩된 로그파일이 없다고 알려준다.

```
-----
1. 새로운 대화 불러오기
2. 특정 기간 내의 대화 참여자 랭킹 보기
3. 시간대별 대화 참여자 랭킹 보기
4. 특정 기간 내의 키워드 랭킹 보기
5. 가장 톡방이 활발했던 날짜 보기
6. 특정 사람의 대화를 받아준 사람의 랭킹 보기
7. 특정 사람이 누구의 대화를 잘 받아줬는지에 대한 랭킹 보기
8. 불러온 대화 제거하기
-----
> 2
로딩된 로그파일이 없습니다.
-----
1. 새로운 대화 불러오기
2. 특정 기간 내의 대화 참여자 랭킹 보기
3. 시간대별 대화 참여자 랭킹 보기
4. 특정 기간 내의 키워드 랭킹 보기
5. 가장 톡방이 활발했던 날짜 보기
6. 특정 사람의 대화를 받아준 사람의 랭킹 보기
7. 특정 사람이 누구의 대화를 잘 받아줬는지에 대한 랭킹 보기
8. 불러온 대화 제거하기
-----
>
```

이외에 각 메뉴 안에서 입력을 받을 때에도 try, except, raise를 이용해 오류가 발생하더라도 프로그램이 그대로 종료되어버리지는 않도록 예외처리를 해두었다. 이제 각 기능들을 수행하는 SQL 쿼리를 알아보겠다. 첫 번째를 대화를 불러올 때에는 MESSAGE, SENDS, USER, CONVERSATION 테이블에 INSERT를 한다. 예시 대화와 4개의 insert문은 아래와 같다.



```
INSERT INTO MESSAGE(id, msgtext, sendtime)
VALUES (11418, '수잘알 갓도...', '2016-03-08 23:20:00');
```

```
INSERT INTO SENDS(sender, id)
VALUES ('회원님', 11200);
```

```
INSERT INTO USER(name, jointime)
VALUES ('회원님', '2016-03-08 23:20:00');
```

```
insert into CONVERSATION(sender, receiver, starttime)
VALUES ('회원님', '16@@@', '2016-03-08 23:20:00');
```

이다.

특정 기간 내의 대화 참여자 랭킹을 보는 쿼리는

```
SELECT sender, count(sender) FROM message NATURAL JOIN sends
WHERE '2017-08-01' <= date_format(sendtime, '%Y-%m-%d')
AND date_format(sendtime, '%Y-%m-%d') <= '2017-08-03'
GROUP BY sender ORDER BY count(sender) desc;
```

이다.(2017-08-01일부터 2017-08-03일까지의 랭킹을 보는 예시이다.)

시간대별 대화 참여자의 랭킹을 보는 쿼리는

```
SELECT sender, count(sender) FROM message NATURAL JOIN sends
WHERE 11 <= extract(hour FROM sendtime)
AND extract(hour FROM sendtime) <= 13
GROUP BY sender ORDER BY count(sender) desc;
```

이다. (11시부터 13시까지의 랭킹을 보는 예시이다.)

특정 기간 내의 키워드 랭킹을 보기 위해서는 우선 기간 내의 메시지를 SELECT한 후 konlpy를 이용해 메시지에서 키워드를 분리해 KEYWORD 테이블에 INSERT를 한 뒤, KEYWORD 테이블에서 SELECT를 하는 방식으로 진행된다. 쿼리는

```
SELECT sender, msgtext, sendtime, id
FROM message NATURAL JOIN sends
WHERE '2017-08-27' <= date_format(sendtime, '%Y-%m-%d')
AND date_format(sendtime, '%Y-%m-%d') <= '2017-09-01';
```

```
INSERT INTO keyword(word, id) VALUES ('WORDS', 10323);
```

```
SELECT word, count(word) FROM keyword
GROUP BY word ORDER BY count(word) desc limit 10;
```

이다.(2017-08-27일부터 2017-08-29일 사이의 상위 10개의 키워드에 대한 랭킹을 보는 예시이다.)

가장 독방이 활발했던 날짜를 보는 쿼리는

```
SELECT date_format(sendtime, '%Y-%m-%d'),
count(date_format(sendtime, '%Y-%m-%d'))
FROM message GROUP BY date_format(sendtime, '%Y-%m-%d')
```

```
ORDER BY count(date_format(sendtime, '%Y-%m-%d'))
desc limit 10;
```

이다.(상위 10개의 키워드에 대한 랭킹을 보는 예시이다.)

특정 사람의 대화를 받아준 사람의 랭킹을 보는 쿼리는

```
SELECT receiver, count(receiver)
FROM conversation WHERE sender = '회원님 '
GROUP BY receiver ORDER BY count(receiver) desc;
```

이다.(나에 대한 랭킹을 보는 예시이다.)

7번 기능을 수행하는 쿼리는

```
SELECT sender, count(sender)
FROM conversation WHERE receiver = '회원님 '
GROUP BY sender ORDER BY count(sender) desc;
```

이다.(나에 대한 랭킹을 보는 예시이다.)

8번 기능을 수행하는 쿼리는

```
TRUNCATE TABLE conversation;
TRUNCATE TABLE sends;
TRUNCATE TABLE keyword;
TRUNCATE TABLE message;
TRUNCATE TABLE user;
```

이다.

1번부터 8번까지의 기능을 사용한 스크린샷은 아래와 같다.

- 
1. 새로운 대화 불러오기
  2. 특정 기간 내의 대화 참여자 랭킹 보기
  3. 시간대별 대화 참여자 랭킹 보기
  4. 특정 기간 내의 키워드 랭킹 보기
  5. 가장 독방이 활발했던 날짜 보기
  6. 특정 사람의 대화를 받아준 사람의 랭킹 보기
  7. 특정 사람이 누구의 대화를 잘 받아줬는지에 대한 랭킹 보기
  8. 불러온 대화 제거하기
- 

> 1

로그파일 이름을 입력해주세요.

> 3rdCreativeLine.txt

파싱을 시작합니다.

10000번째 줄까지 파싱 완료했습니다.  
 20000번째 줄까지 파싱 완료했습니다.  
 30000번째 줄까지 파싱 완료했습니다.  
 40000번째 줄까지 파싱 완료했습니다.  
 50000번째 줄까지 파싱 완료했습니다.  
 60000번째 줄까지 파싱 완료했습니다.  
 70000번째 줄까지 파싱 완료했습니다.  
 80000번째 줄까지 파싱 완료했습니다.  
 90000번째 줄까지 파싱 완료했습니다.  
 100000번째 줄까지 파싱 완료했습니다.  
 110000번째 줄까지 파싱 완료했습니다.  
 120000번째 줄까지 파싱 완료했습니다.  
 130000번째 줄까지 파싱 완료했습니다.  
 140000번째 줄까지 파싱 완료했습니다.  
 150000번째 줄까지 파싱 완료했습니다.  
 160000번째 줄까지 파싱 완료했습니다.  
 170000번째 줄까지 파싱 완료했습니다.  
 180000번째 줄까지 파싱 완료했습니다.  
 190000번째 줄까지 파싱 완료했습니다.  
 200000번째 줄까지 파싱 완료했습니다.  
 210000번째 줄까지 파싱 완료했습니다.  
 220000번째 줄까지 파싱 완료했습니다.  
 230000번째 줄까지 파싱 완료했습니다.  
 240000번째 줄까지 파싱 완료했습니다.  
 250000번째 줄까지 파싱 완료했습니다.  
 260000번째 줄까지 파싱 완료했습니다.  
 270000번째 줄까지 파싱 완료했습니다.  
 280000번째 줄까지 파싱 완료했습니다.

파싱을 완료했습니다.

처리한 쿼리 : 263561

작성자가 탈퇴해서 처리하지 못한 쿼리 : 12768

그 외 오류로 처리하지 못한 쿼리 : 10955



1. 새로운 대화 불러오기
2. 특정 기간 내의 대화 참여자 랭킹 보기
3. 시간대별 대화 참여자 랭킹 보기
4. 특정 기간 내의 키워드 랭킹 보기
5. 가장 독방이 활발했던 날짜 보기
6. 특정 사람의 대화를 받아준 사람의 랭킹 보기
7. 특정 사람이 누구의 대화를 잘 받아줬는지에 대한 랭킹 보기
8. 불러온 대화 제거하기

> 2

전체 기간에 대해 확인하겠습니까? (Y/N)

> Y

(0001-01-01) ~ (9999-01-01) 기간의 메시지 : 263561개

순위	이름	갯수	비율
1	김민준	30019	11.39%
2	김민준	22028	8.36%
3	김민준	19522	7.41%
4	김민준	18632	7.07%
5	김민준	17680	6.71%
6	김민준	14773	5.61%
7	김민준	14655	5.56%
8	김민준	13113	4.98%
9	김민준	11659	4.42%
10	김민준	11536	4.38%
11	김민준	11135	4.22%
12	김민준	11111	4.22%
13	김민준	10482	3.98%
14	김민준	9093	3.45%
15	김민준	7782	2.95%
16	김민준	7078	2.69%
17	김민준	5038	1.91%
18	김민준	4936	1.87%
19	김민준	3930	1.49%
20	김민준	3680	1.40%
21	김민준	3642	1.38%
22	김민준	3577	1.36%
23	김민준	2970	1.13%
24	김민준	2281	0.87%
25	김민준	2278	0.86%
26	김민준	787	0.30%
27	김민준	97	0.04%
28	김민준	28	0.01%
29	김민준	18	0.01%
30	김민준	1	0.00%

1. 새로운 대화 불러오기
2. 특정 기간 내의 대화 참여자 랭킹 보기
3. 시간대별 대화 참여자 랭킹 보기
4. 특정 기간 내의 키워드 랭킹 보기
5. 가장 폭방이 활발했던 날짜 보기
6. 특정 사람의 대화를 받아준 사람의 랭킹 보기
7. 특정 사람이 누구의 대화를 잘 받아왔는지에 대한 랭킹 보기
8. 불러온 대화 제거하기

> 4

기간이 길 경우 분석 시간이 오래 걸릴 수 있습니다. 한달 이내의 기간으로 확인하는 것을 권장합니다.  
기간의 시작 날짜와 끝 날짜를 아래 형식대로 입력해주세요

ex) 2017-04-01~2017-04-07

> 2017-11-26~2017-12-02

최대 몇 위 까지의 키워드를 보고싶은지 입력해주세요.

> 30

2017-11-26일 대화 분석을 시작합니다.  
2017-11-27일 대화 분석을 시작합니다.  
2017-11-28일 대화 분석을 시작합니다.  
2017-11-29일 대화 분석을 시작합니다.  
2017-11-30일 대화 분석을 시작합니다.  
2017-12-01일 대화 분석을 시작합니다.  
2017-12-02일 대화 분석을 시작합니다.

(2017-11-26) ~ (2017-12-02) 기간에 추출된 키워드 : 3430개

순위	키워드	갯수	비율
1	폭방	34	0.99%
2	특정	29	0.85%
3	오전	23	0.67%
4	지문	15	0.44%
5	문자	14	0.41%
6	가장	13	0.38%
7	생각	13	0.38%
8	내가	12	0.35%
9	사람	11	0.32%
10	그거	11	0.32%
11	하나	11	0.32%
12	정리	10	0.29%
13	나도	10	0.29%
14	베이	9	0.26%
15	비트	9	0.26%
16	머리	9	0.26%
17	정도	9	0.26%
18	우리	9	0.26%
19	11	9	0.26%
20	원래	8	0.23%
21	이름	8	0.23%
22	이	8	0.23%
23	12	8	0.23%
24	한	8	0.23%
25	있음	8	0.23%
26	어디	8	0.23%
27	시간	8	0.23%
28	수업	8	0.23%
29	내일	7	0.20%
30	게임	7	0.20%

- 
1. 새로운 대화 불러오기
  2. 특정 기간 내의 대화 참여자 랭킹 보기
  3. 시간대별 대화 참여자 랭킹 보기
  4. 특정 기간 내의 키워드 랭킹 보기
  5. 가장 톡방이 활발했던 날짜 보기
  6. 특정 사람의 대화를 받아주 사람의 랭킹 보기
  7. 특정 사람이 누구의 대화를 잘 받아줬는지에 대한 랭킹 보기
  8. 불러온 대화 제거하기
- 

> 5

최대 몇 위까지의 날짜를 보고싶은지 입력해주세요.

> 20

순위	날짜	갯수
1	2016-11-22	3756
2	2016-12-13	3292
3	2016-11-23	3251
4	2017-01-02	3100
5	2016-12-15	2863
6	2016-12-12	2857
7	2016-12-01	2309
8	2016-12-18	2228
9	2016-12-16	2115
10	2016-11-28	2108
11	2016-12-20	2060
12	2017-01-29	2035
13	2016-12-25	1988
14	2017-11-23	1968
15	2017-07-06	1952
16	2017-08-25	1922
17	2016-11-25	1921
18	2017-08-03	1913
19	2017-05-08	1816
20	2017-04-04	1805

- 
1. 새로운 대화 불러오기
  2. 특정 기간 내의 대화 참여자 랭킹 보기
  3. 시간대별 대화 참여자 랭킹 보기
  4. 특정 기간 내의 키워드 랭킹 보기
  5. 가장 톡방이 활발했던 날짜 보기
  6. 특정 사람의 대화를 받아준 사람의 랭킹 보기
  7. 특정 사람이 누구의 대화를 잘 받아줬는지에 대한 랭킹 보기
  8. 불러온 대화 제거하기
- 

> 6  
 확인하고 싶은 사람의 이름을 입력하세요.  
 > 회원님

'회원님'님이 시작한 대화를 받아준 사람			
순위	이름	갯수	비율
1	이정민	313	10.83%
2	이도현	269	9.31%
3	이도민	223	7.72%
4	이도준	195	6.75%
5	이도원	194	6.71%
6	이도준	189	6.54%
7	이도민	182	6.30%
8	이도준	164	5.67%
9	이도원	149	5.16%
10	이도준	138	4.78%
11	이도민	131	4.53%
12	이도준	124	4.29%
13	이도민	103	3.56%
14	이도준	76	2.63%
15	이도원	72	2.49%
16	이도준	68	2.35%
17	이도민	62	2.15%
18	이도준	60	2.08%
19	이도원	42	1.45%
20	이도준	35	1.21%
21	이도민	34	1.18%
22	이도준	29	1.00%
23	이도원	22	0.76%
24	이도준	13	0.45%
25	이도민	2	0.07%
26	이도준	1	0.03%

- 
1. 새로운 대화 불러오기
  2. 특정 기간 내의 대화 참여자 랭킹 보기
  3. 시간대별 대화 참여자 랭킹 보기
  4. 특정 기간 내의 키워드 랭킹 보기
  5. 가장 톡방이 활발했던 날짜 보기
  6. 특정 사람의 대화를 받아준 사람의 랭킹 보기
  7. 특정 사람이 누구의 대화를 잘 받아줬는지에 대한 랭킹 보기
  8. 불러온 대화 제거하기
- 

> 7  
 확인하고 싶은 사람의 이름을 입력하세요.  
 > 회원님

'회원님'님이 대화를 받아준 사람			
순위	이름	갯수	비율
1	김민준	316	11.27%
2	김민준	256	9.13%
3	김민준	254	9.06%
4	김민준	193	6.88%
5	김민준	176	6.27%
6	김민준	170	6.06%
7	김민준	155	5.53%
8	김민준	155	5.53%
9	김민준	136	4.85%
10	김민준	130	4.63%
11	김민준	126	4.49%
12	김민준	111	3.96%
13	김민준	85	3.03%
14	김민준	84	2.99%
15	김민준	82	2.92%
16	김민준	68	2.42%
17	김민준	62	2.21%
18	김민준	62	2.21%
19	김민준	51	1.82%
20	김민준	32	1.14%
21	김민준	31	1.11%
22	김민준	27	0.96%
23	김민준	21	0.75%
24	김민준	17	0.61%
25	김민준	3	0.11%
26	김민준	1	0.04%
27	김민준	1	0.04%

```
-----
1. 새로운 대화 불러오기
2. 특정 기간 내의 대화 참여자 랭킹 보기
3. 시간대별 대화 참여자 랭킹 보기
4. 특정 기간 내의 키워드 랭킹 보기
5. 가장 독방이 활발했던 날짜 보기
6. 특정 사람의 대화를 받아준 사람의 랭킹 보기
7. 특정 사람이 누구의 대화를 잘 받아줬는지에 대한 랭킹 보기
8. 불러온 대화 제거하기
-----
> 8
대화의 제거가 완료되었습니다.
```

## 5. 후기 및 개선사항

많은 양의 데이터를 MySQL로 처리해보니 파이썬에서 직접 dictionary 등의 자료구조를 이용해 데이터를 가지고 있는 것 보다 MySQL을 활용하는 것이 작업하기 훨씬 편했고 확장성의 면에서도 큰 도움이 되었다. 시간이 더 많았다면 웹서비스로 구현을 하고 싶었으나 그렇지 못한 것이 아쉽고 방학때 시간이 된다면 Django나 Flask를 이용해 웹서비스로 구현을 해보고 싶다.

또한 지금은 혼자 사용할 프로그램이니 보안에 크게 신경을 안 쓰고 구현했는데, 실제로는 현재의 프로그램은 SQL injection이 일어날 수 있는 여지가 굉장히 많다. 상용 프로그램으로 이 프로그램을 개선할 경우에는 SQL injection을 염두에 두고 코드를 개선해야할 것이다.