23-1 Database System Team Project1

21900156 김예준 / 22000796 함상훈 / 22100579 이진주

How to attack this problem?

Our first step is to analyze the website, https://kubic.handong.edu/.

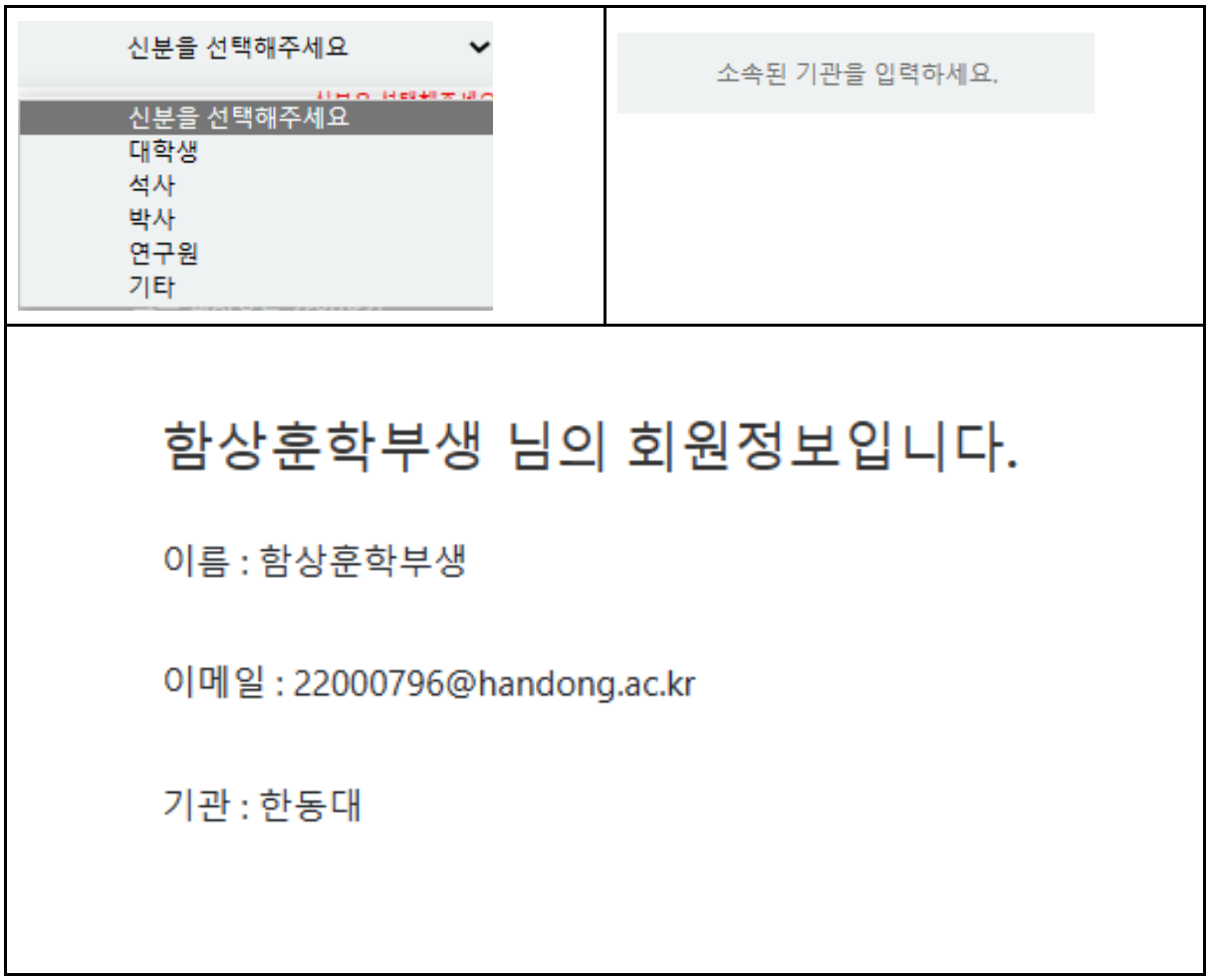
We are able to find attributes that allow us to find the document, such as doc_type, title, post,title,first_char, published_institution.

We do a search through the search bar, and we get results like the one in the following image.

We were able to find attributes like post_title, post_date, post_writer, and abstract that make up the documents.

And if we click on the documentation and look closely, we can find attributes like published_institution, published_institutuion_url, original_url, post_body.

When we sign up, it asks for the information such as occupation, institute, and email. The email also contains information about the name.



When we look at the Announcements, FAQs, and Q&As in the Community tab, we find the following properties: docId, category, title, content, userName, regDate and content.

글 번호		글 제목	작성자	등록일
2	검토중	글 쓰기가 안됩니다.	전여훈	21-02-26
1	완료	회원 약관은 어디서 확인하나요?	전여훈	21-02-26

버그가 있어요

작성자: 전여훈

등록일: 23-05-21

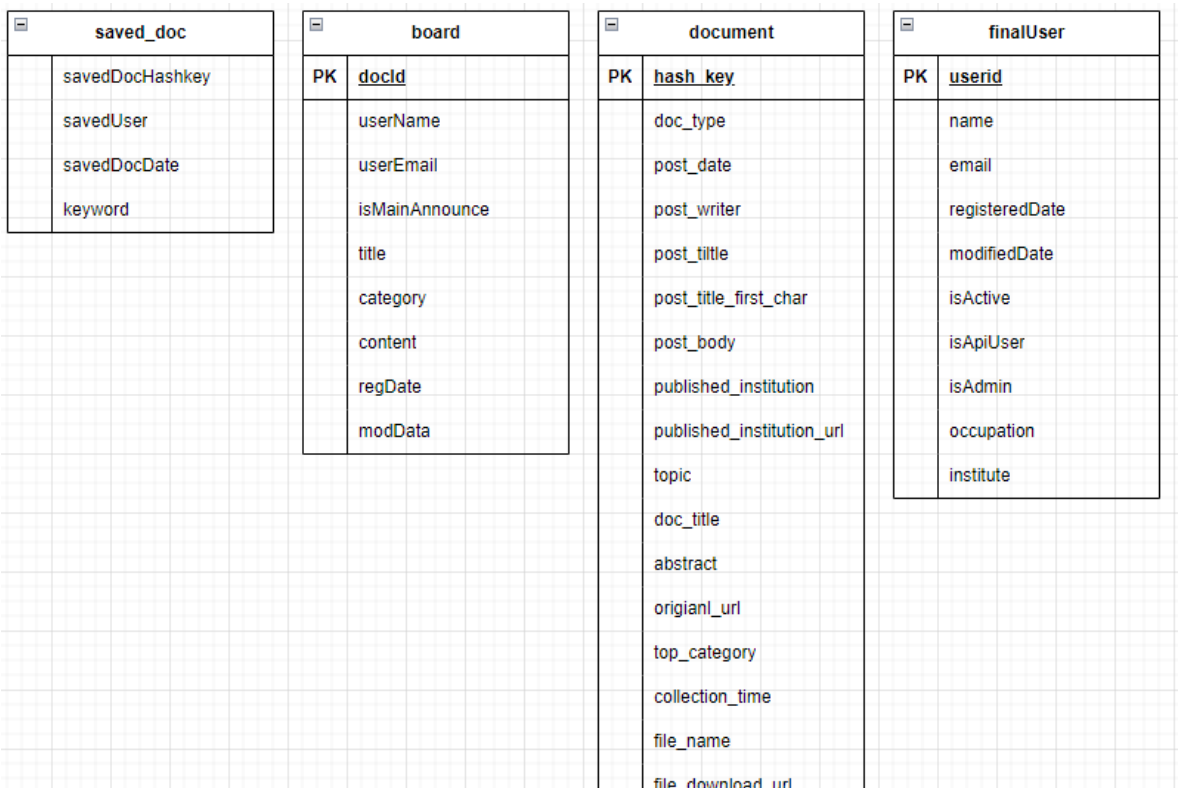
홈페이지 사용시 버그가 발생하면 관리자에게 문의 바랍니다.

When we log in and search a document, we find the ability to save the document.

This shows us that to save the document use keyword, savedUser, and savedDocDate.

The second step is to find out if the attribute we haven't identified is associated with other attributes.After running the query and comparing the properties not found on the homepage. Also, normalization principles were consulted to discover and reflect dependencies between columns.

According to the above steps, we are able to create the following table.



The next step is to find unnecessary duplicates.
We used various queries with ‘distinct’ to find duplicates like this.

```
SELECT COUNT(DISTINCT [column name]) FROM [origin table];
```

If the result of this kind of query is significantly less than the whole count, we made a new table for that column called ‘[column name]Mapping’.

The mapping tables consist of the de-duplicated target columns from the original table and the primary keys of smaller capacity (e.g., small int) assigned to each unique data of the columns . And the columns separated by the mapping from the origin table are replaced by the columns of the primary key of the mapping table. then, part of the size becomes like this example.

origin table		origin table + mapping table
35000 * varchar(255)	>	35000 * smallint + 100 * (smallint + varchar(255))

We expect the mapping table to minimize the size of the overlapping parts, making the overall data smaller.

Finally, we adjust whole data types and sizes. in the row data, there were many columns defined with non-efficient domain type.
We changed the domain to the most appropriate data type, taking into account the maximum length of the data and the diversity of each data length in each column.

For example, for strings with large length variation within 255 characters, we chose the 'VARCHAR' type. But for strings where all the data is 22 characters long, we chose 'CHAR'. For strings longer than 255 characters, we chose the 'MEDIUM TEXT' type.

hello haha! i am JJ wanna go home…… welcome to HGU! null . .	VARCHAR(255)
21800156 22000796 22100579 . .	CHAR(8)

And for numeric data types, we chose 'TINY INT', 'SMALLINT', etc. as appropriate, considering the maximum value. We were pleasantly surprised at how much space this process could save.

0~255	TINY INT
255~65535	SMALL INT
65535~4294967295	INT

This is the final result of our effort.

DDL Query

document
create table document as select doc_type, post_date, post_title, post_body, hash_key, doc_title, abstract, original_url, collection_time, file_name, file_download_url, file_content, file_id, post_title_first_char, topic, top_category, published_institution, post_writer from kubicdb group by doc_type, post_date, post_title, post_body, hash_key, doc_title, abstract, original_url, collection_time, file_name, file_download_url, file_content, file_id, post_title_first_char, topic, top_category, published_institution, post_writer; alter table document add docType_id tinyint; alter table document add firstChar_id tinyint unsigned; alter table document add topic_id tinyint; alter table document add topCategory_id tinyint unsigned; alter table document add pubInst_id tinyint; alter table document add postWriter_id smallint; alter table document modify post_data varchar(255); alter table document modify post_title varchar(255); alter table document modify post_body text; alter table document modify doc_title varchar(255); alter table document modify original_url text; alter table document modify collection_time char(22); alter table document modify file_name varchar(255); alter table document modify file_download_url text; alter table document modify file_content text; alter table document modify file_id varchar(255);

```
#create a mapping table with them

alter table document drop doc_type;
alter table document drop post_title_first_char;
alter table document drop topic;
alter table document drop top_category;
alter table document drop published_institution;
alter table document drop post_writer;

alter table document add PRIMARY KEY document (hash_key);
alter table savedDoc add CONSTRAINT document_PostfirstCharMapping_firstChar_id_fk
FOREIGN KEYS (firstChar_id) REFERENCES PostfirstCharMapping(firstChar_id);
alter table savedDoc add CONSTRAINT document_topicMapping_topic_id_fk FOREIGN
KEYS (topic_id) REFERENCES topicMapping(topic_id);
alter table savedDoc add CONSTRAINT
document_topCategoryMapping_topCategory_id_fk  FOREIGN KEYS (topCategory_id)
REFERENCES topCategoryMapping(topCategory_id);
alter table savedDoc add CONSTRAINT
document_pubInstitutionMapping_pubInst_id_fk FOREIGN KEYS (pubInst_id)
REFERENCES pubInstitutionMapping(pubInst_id);
alter table savedDoc add CONSTRAINT
document_postWriterMapping_postWriter_id_fk FOREIGN KEYS (postWriter_id)
REFERENCES postWriterMapping(postWriter_id);
```

savedDoc

```
create table savedDoc as
select keyword, savedUser, savedDocDate, savedDocHashKey from kubicdb
group by keyword, savedUser, savedDocDate, savedDocHashKey;

alter table savedDoc add key_id tinyint;
alter table savedDoc add eid tinyint;
alter table savedDoc add savedDocs_id tinyint;
alter table savedDoc add PRIMARY KEY savedDoc (eid, key_id, savedDocs_id);
alter table savedDoc add CONSTRAINT savedDoc_keyword_mapping_key_id_fk
FOREIGN KEYS (key_id) REFERENCES keyword_mapping(key_id);
alter table savedDoc add CONSTRAINT savedDoc_finalUser_eid_fk FOREIGN KEYS
(eid) REFERENCES finalUser(eid);
alter table savedDoc add CONSTRAINT savedDoc_savedDocsMapping_savedDocs_id_fk
FOREIGN KEYS (savedDocs_id) REFERENCES savedDocsMapping(savedDocs_id);
```

```
#create a mapping table with them

alter table savedDoc drop keyword;
alter table savedDoc drop savedUser;
alter table savedDoc drop savedDocDate;
alter table savedDoc drop savedDocHashKey;
```

finalUser

```
create table finalUser as
select distinct userId, name, email, institute, occupation, registeredDate, modifiedDate,
isActive, isApiUser, isAdmin
from kubicdb
where userId is not null
group by userId, name, email, institute, occupation, registeredDate, modifiedDate,
isActive, isApiUser, isAdmin;

alter table finalUser add eid tinyint;
alter table finalUser add occu_id tinyint;
alter table finalUser add inst_id tinyint;
alter table finalUser add PRIMARY KEY finalUser(eid);
alter table finalUser add CONSTRAINT finalUser_occupationMapping_occu_id_fk
FOREIGN KEYS (occu_id) REFERENCES occupationMapping(occu_id);
alter table finalUser add CONSTRAINT finalUser_instituteMapping_inst_id_fk FOREIGN
KEYS (inst_id) REFERENCES instituteMapping(inst_id);

#create a mapping table with them

alter table finalUser drop email;
alter table finalUser drop occupation;
alter table finalUser drop institute;
alter table finalUser modify userId char(24);
alter table finalUser modify name varchar(23);
alter table finalUser modify registeredDate bigint;
alter table finalUser modify modifiedDate bigint;
alter table finalUser modify isActive tinyint;
alter table finalUser modify isApiUser tinyint;
alter table finalUser modify isAdmin tinyint;
```

board

```
create table board as
select title, content, userName, userEmail, isMainAnnounce, regDate, modDate, docId,
category
from kubicdb
where docId is not null
group by title, content, userName, userEmail, userEmail, isMainAnnounce, regDate,
modDate, docId, category;

alter table board add eid tinyint;

#create a mapping table with them

alter table board drop userName;
alter table board drop userEmail;
alter table board modify title varchar(255);
alter table board modify content varchar(255);
alter table board modify isMainAnnounce tinyint;
alter table board modify regDate bigint;
alter table board modify modDate bigint;
alter table board modify docId tinyint;
alter table board modify category varchar(255);

alter table board add PRIMARY KEY board(docId);
alter table board add CONSTRAINT board_emailMapping_eid_fk FOREIGN KEYS (eid)
REFERENCES emailMapping(eid);
```

doc_type_mapping

```
create table doc_type_mapping(
    docType_id tinyint primary key auto_increment,
    doc_type char(5)
);
```

emailMapping

```
create table emailMapping(
    eid tinyint primary key auto_increment,
    email varchar(255)
);
```

instituteMapping
create table instituteMapping(inst_id tinyint primary key auto_increment, institute varchar(100));

keyword_mapping
create table keyword_mapping(key_id tinyint primary key auto_increment, keyword varchar(5));

occupationMapping
create table occupationMapping(occu_id tinyint primary key auto_increment, occupation varchar(255));

PostfirstCharMapping
create table PostfirstCharMapping(firstChar_id tinyint unsigned primary key auto_increment, post_title_first_char(1));

postWriterMapping
create table postWriterMapping(postWrite_id smallint primary key auto_increment, post_writer varchar(255));

pubInstitutionMapping

```
create table pubInstitutionMapping(
    publinst_id tinyintprimary key auto_increment,
    published_institution varchar(255),
    published_institution_url varchar(255)
);
```

savedDocsMapping

```
create table savedDocsMapping(
    savedDocs_id tinyint primary key auto_increment,
    savedDocDate date,
    savedDocHashKey char(20)
);
```

topCategoryMapping

```
create table topCategoryMapping(
    topCategoty_id tinyint unsigned primary key auto_increment,
    top_category varchar(255)
);
```

topicMapping

```
create table topicMapping(
    topic_id tinyint primary key auto_increment,
    topic varchar(255)
);
```

Result for View instruction

. ***#1. total_volume***

```
create view total_volume as
SELECT table_schema AS 'DB16',
ROUND(SUM(data_length+index_length)/1024, 1) AS 'Size(KB)'
FROM information_schema.tables
WHERE table_schema = 'DB16'
```

```
AND TABLE_NAME <> 'kubicdb';
```

WHERE ORDER BY

	DB16	Size(KB)
1	DB16	207904.0

#2. userCount

```
create view userCount as
select COUNT(*) from finalUser;
```

1 row Tx: Auto DDL

WHERE ORDER BY

	COUNT(*)
1	100

#3. boardCount

```
create view boardCount as
select COUNT(*) from board;
```

WHERE ORDER BY

	COUNT(*)
1	4

#4. docCount

```
create view docCount as
select COUNT(*) from document;
```

WHERE ORDER BY

	COUNT(*)
1	30782

#5. instInfo

```
create view instInfo as
select published_institution, count(published_institution) as CNT from document d join
pubInstitutionMapping p on d.pubInst_id = p.pubInst_id
                                                    group by published_institution
                                                    order by count(published_institution) asc;
```

WHERE ORDER BY

	published_institution	CNT
1	한민족통일여성협의회	1
2	평화통일연대	5
3	KYC한국청년연합	8
4	남북사회통합연구원	8
5	고려대 아세아문제연구원	9
6	한국YMCA	10
7	대한불교조계종 민족공동체추진본부	11
8	원주시민연대	12
9	평화재단	12
10	통일부이산가족찾기	13
11	북한인권정보센터	13
12	서울대학교 의과대학 통일의학센터	14
13	북한SDGs데이터포털	14
14	한양대학교 평화연구소	15
15	교육부 통통평화학교	15
16	평화통일동포연합	18

#6. my_docs_2019_keyword_count

```
create view my_docs_2019_keyword_count as
select keyword, count(keyword) as keyword_count from savedDoc d
    join keyword_mapping k on d.key_id = k.key_id
    join DB16.savedDocsMapping sDM on d.savedDocs_id = sDM.savedDocs_id
where savedDocDate like '2019%'
group by keyword
order by count(keyword) desc;
```

WHERE ORDER BY

	keyword	keyword_count
1	정책	98
2	교육	97
3	토론	95
4	발표	94
5	연구	93
6	자료	86
7	인물조사	82
8	논문	79
9	법률	78
10	보고서	78

#7. policy_writer_count

```
create view policy_writer_count as
select post_writer, count(post_writer) as data_count from document d
    left join postWriterMapping pWM on d.postWriter_id = pWM.postWriter_id
    left join savedDocsMapping sDM on d.hash_key = sDM.savedDocHashKey
    left join savedDoc sD on sD.savedDocs_id = sDM.savedDocs_id
    left join keyword_mapping k on sD.key_id = k.key_id
where keyword = '토론'
```

```
group by post_writer
order by count(post_writer) desc;
```

WHERE ORDER BY

	post_writer	data_count
1	재나이스피플	39
2	국가안보전략연구원	29
3	서보혁	22
4	동아시아연구원	21
5	조한범	20
6	관리자	20
7	외교안보연구소	18
8	세종연구소	17
9	이규창	16
10	김병권	14
11	박형중	11
12	통일맞이	8
13	한국국방연구원	7
14	KDB 미래전략연구소	6
15	사무국	6
16	국방대학교 안보문제연구소	6

#8. inst_data_max_status

```
create view inst_data_max_status as
with users(eid, institute, occupation) as (
    select f.eid, institute, occupation from finalUser f
     join instituteMapping iM on f.inst_id = iM.inst_id
     join occupationMapping oM on f.occu_id = oM.occu_id
)
select institute,
       (select occupation from users
                    where institute = u.institute
                    group by occupation
                    order by count(occupation) desc
                    limit 1) as max_status,
       (select count(sD.savedDocs_id)
        from savedDoc sD
            left join users us on sD.eid = us.eid
            left join occupationMapping o on us.occupation = o.occupation
        where sD.eid = us.eid and us.institute = u.institute and us.occupation = (
            select occupation from users
                    where institute = u.institute
                    group by occupation
                    order by count(occupation) desc
                    limit 1)) as max_status_count,
       (select count(sD.savedDocs_id)
        from savedDoc sD left join users us on sD.eid = us.eid
        where sD.eid = us.eid and us.institute = u.institute) as data_count
from users u
    left join savedDoc sD on u.eid = sD.eid
group by institute
order by count(institute) desc;
```

WHERE ORDER BY

	institute	max_status	max_status_count	data_count
1	한동대학교	대학생	1956	2043
2	서울대학교	대학생	319	347
3	연세대학교	대학생	115	115
4	한동대	대학생	90	90
5	개인	기타	30	30
6	재단법인통일과나눔	기타	30	30
7	고려대학교	대학생	30	30
8	한동	대학생	30	30
9	Handong Univ	대학생	30	30
10	전산전자공학부	대학생	30	30
11	KUBIC Middleware팀	대학생	30	30
12	HGU	대학생	30	30
13	숭실대학교	박사	30	30
14	숭실대	박사	30	30
15	숭실대 대학원	박사	29	29

#9. checkDoc

```
create view checkDoc as
select post_title, post_writer, published_institution, post_date, top_category from
document d
    left join postWriterMapping pWM on d.postWriter_id = pWM.postWriter_id
    left join pubInstitutionMapping pIM on d.pubInst_id = pIM.pubInst_id
    left join topCategoryMapping tCM on d.topCategory_id = tCM.topCategory_id
order by post_date desc
limit 5;
```

WHERE ORDER BY

	post_title	post_writer	published_institution	post_date	top_category
1	인류사 3500년 중 전쟁 없었던 해는 270년에 불과 전쟁 가능성에 대비해야만 평화	통일과나눔	통일과나눔	2023-03-31	통일 스터디
2	논평 성평등 삭제하고 살만하지 않은 사회를 공고하게 만드는 저출산 정책 전면	adminwmp	평화를만드는여성회	2023-03-31	성명서및보도자료
3	정전협정 70년을 맞이하여 한반도 평화를 바라는 여성들의 입장	adminwmp	평화를만드는여성회	2023-03-30	성명서및보도자료
4	자료집 강제동원 정부 해법 관련 긴급 국회 토론회	겨레하나	겨레하나	2023-03-28	자료실/일반자료실
5	성명서 북 연이은 단거리탄도미사일 발사를 규탄한다	관리자	한국자유총연맹	2023-03-27	보도자료 성명서/ 북한

#10. category_Count

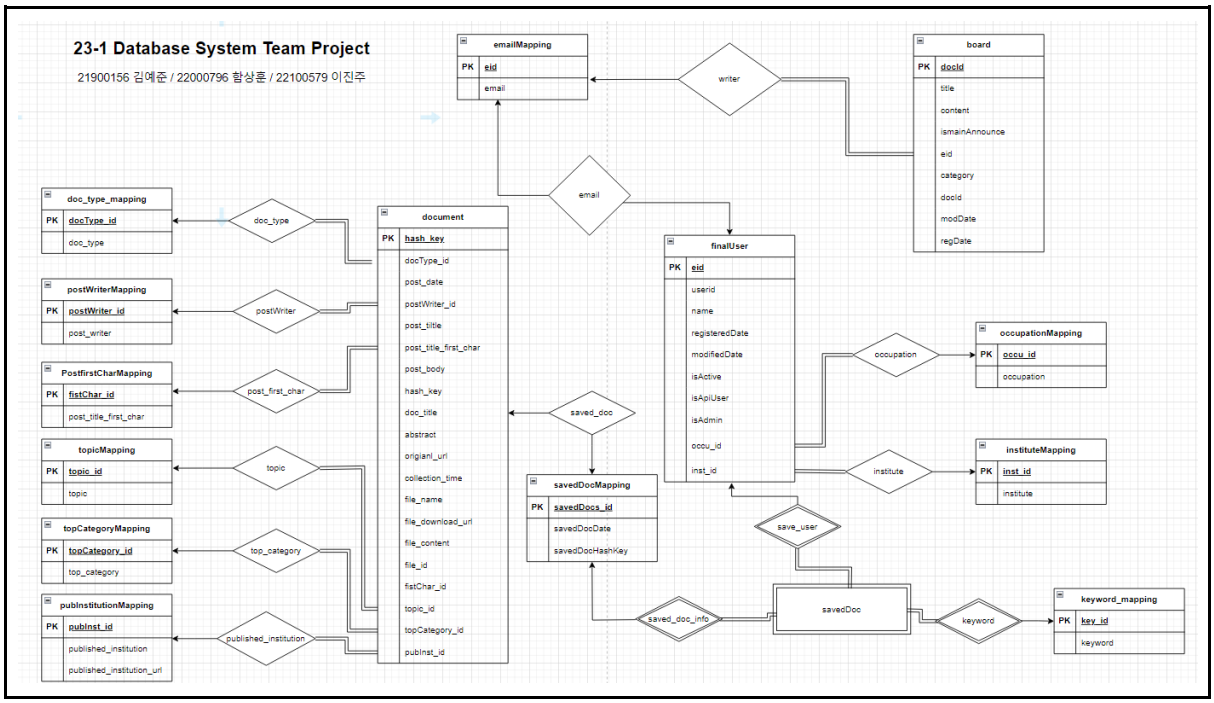
```
create view category_Count as
select top_category, count(top_category) as category_count, rank() over (order by
count(top_category) desc) as category_rank
from document d
    left join topCategoryMapping tCM on d.topCategory_id = tCM.topCategory_id
group by top_category
order by count(top_category) desc;
```

158 rows | Tx: Auto | DDL

WHERE | ORDER BY

	top_category	category_count	category_rank
1	연구자료	7982	1
2	전체자료	5396	2
3	정부자료	2868	3
4	언론자료	1549	4
5	통일부 발간자료	1333	5
6	통일부 발간물	908	6
7	현안분석-온라인시리즈	586	7
8	정기간행물-주간통일정세	545	8
9	통일문제 이해	541	9
10	참고자료	500	10
11	북한소식	453	11
12	북한이해	383	12
13	자료실	357	13

ER-diagram



This is the ER-Diagram we draw.

Summary of the database size and table sizes

- database size is 216160.0 KB
- table sizes are below image
-

	TABLE_SCHEMA	TABLE_NAME	data(KB)	idx(KB)
1	DB16	PostfirstCharMapping	16.0	0.0
2	DB16	board	16.0	16.0
3	DB16	doc_type_mapping	16.0	0.0
4	DB16	document	207568.0	7760.0
5	DB16	emailMapping	16.0	0.0
6	DB16	finalUser	16.0	32.0
7	DB16	instituteMapping	16.0	0.0
8	DB16	keyword_mapping	16.0	0.0
9	DB16	occupationMapping	16.0	0.0
10	DB16	postWriterMapping	384.0	0.0
11	DB16	pubInstitutionMapping	16.0	0.0
12	DB16	savedDoc	96.0	112.0
13	DB16	savedDocsMapping	16.0	0.0
14	DB16	topCategoryMapping	16.0	0.0
15	DB16	topicMapping	16.0	0.0