Term Project Phase 2 Report

Database System Section 02 Team 7

21700675 Jihoon Jeong

21800506 Donggyu Lee

22000546 Yeeun Lee

1. Goal of Project

"Design and implement a database instance that is efficient in time"

2. Approach

1) Denormalization

Denormalization between two tables may decrease the time for a certain task. However, such denormalization can increase the time for other tasks at the same time. Furthermore, it will increase the size of the database.

Considering those factors and our database schema, we decide not to denormalize our database.

2) Indexing

- Referenced Function
  - explain (https://dev.mysql.com/doc/refman/8.0/en/explain.html)

    : EXPLAINING statement is used to obtain a query execution plan. One can use this statement by putting 'EXPLAIN' right in front of the query that you want to check its execution plan.

    Example)

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	147	100.00	Using where
1	PRIMARY	user	NULL	ref	status_index	status_index	1023	A.status	24	100.00	NULL
2	DERIVED	user	NULL	index	status_index	status_index	1023	NULL	147	100.00	Using index

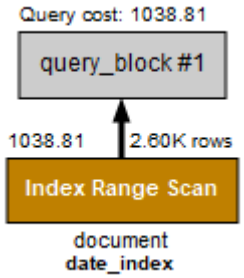
By looking at the type column one can know which method was applied to get the data from the table which means that it is possible to know whether indexing is applied or not. Furthermore, from the key column one can know which index was used and from rows one can know how many rows were considered during the execution.

- profiling (https://dev.mysql.com/doc/refman/8.0/en/show-profile.html)
  : SHOW PROFILE statements display profiling information that indicates resource usage for statements executed during the course of the current session.

Query_ID	Duration	Query
38	0.02649950	WITH A AS (SELECT compareDocID FROM similarity WHERE docID = (SELECT I

By looking at the Duration column, one can know the execution time of a query. It is used to compare the time efficiency of completing each task.

- Execution Plan
  : One can check the visualized execution plan of a query by using the Execution Plan function provided by MySQL Workbench.



Information such as number of referenced rows, query cost, the way of getting data from a table(eg. Index Range Scan, Full Scan).

- Steps
  → Write Query for each task
  → Measure the duration(Duration before applying index) of each task through profiling.

→ Create and apply proper index to certain table to increase the efficiency of certain task

→ Measure the duration(Duration after applying index) of each task through profiling. In addition, check whether the index is used or not in the query by using the 'explain' command in MySQL. If indexing was not applied even after its creation, consider modifying the query so that indexing can be applied and check it again.

→ Compare the duration before and after the index application to determine whether to apply the index or not.

3. Results

1) Solution Query, Results and the Execution Time for Each Task

- Task 1
  - Query

```
SELECT topic, COUNT(*) cnt,
RANK() OVER (ORDER BY COUNT(*) DESC) rank_topic
FROM document JOIN search ON document.hash_key = search.hashKey
WHERE topic IS NOT NULL
GROUP BY topic;
```

  - Result

	topic	cnt	rank_topic
1	문화	5184	1
2	정치	1429	2
3	경제	761	3
4	사회	579	4
5	국제	535	5
6	IT_과학	449	6
7	스포츠	58	7

- Execution Time Before Indexing

	Query_ID	Duration	Query
▶	23	0.04469525	SELECT topic, count(*) cnt, RANK() OVER (ORDER BY count(*) DESC) rank_topic F

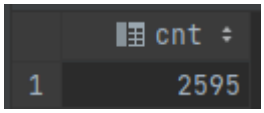
- Execution Time After Indexing

	Query_ID	Duration	Query
▶	434	0.04261975	SELECT topic, count(*) cnt, RANK() OVER (ORDER BY count(*) DESC) rank_topic

- Task 2
  - Query

```
SELECT COUNT(*) FROM document
WHERE post_date >= '2019' AND post_date <= '2022';
```

  - Result



	cnt
1	2595

  - Execution Time Before Indexing

	Query_ID	Duration	Query
▶	24	0.01849400	select count(*) from document where post_date >= '2019' AND post_date <= '2022'

  - Execution Time After Indexing

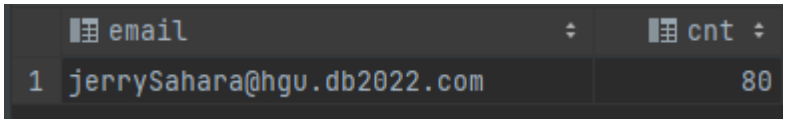
	Query_ID	Duration	Query
▶	47	0.00517600	select count(*) from document where post_date >= '2019' AND post_date <= '2022'

- Task 3
  - Query

```
SELECT A.email, COUNT(*) cnt
FROM user A JOIN saved B ON A.userID = B.userID
WHERE savedDate >= '2022-01-01' AND savedDate <= '2022-12-31'
GROUP BY A.userID
ORDER BY cnt DESC
```

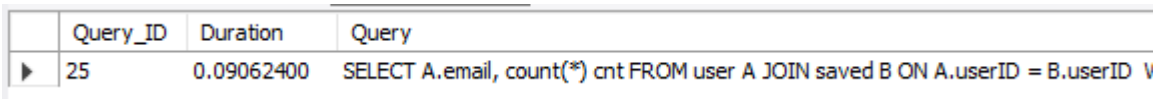
```
LIMIT 1;
```

- Result



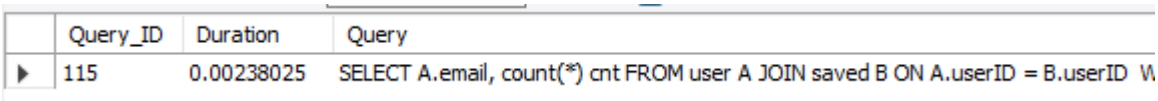
	email	cnt
1	jerrySahara@hgu.db2022.com	80

- Execution Time Before Indexing



	Query_ID	Duration	Query
▶	25	0.09062400	SELECT A.email, count(*) cnt FROM user A JOIN saved B ON A.userID = B.userID W

- Execution Time After Indexing

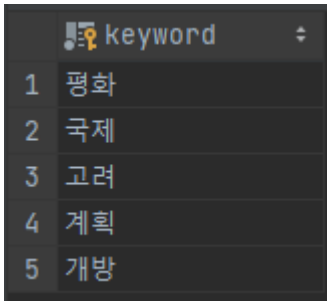


	Query_ID	Duration	Query
▶	115	0.00238025	SELECT A.email, count(*) cnt FROM user A JOIN saved B ON A.userID = B.userID W

- Task 4
  - Query

```
SELECT keyword
FROM
(SELECT keyword, COUNT(*) CNT
    FROM saved
    WHERE userID = (WITH A AS (SELECT userID, COUNT(*) AS cnt
                FROM saved
                WHERE savedDate >= '2022-01-01' AND savedDate <=
'2022-12-31'
                GROUP BY userID)
                SELECT userID FROM A
                WHERE cnt = (SELECT MAX(cnt) FROM A))
GROUP BY keyword) AS B
ORDER BY CNT DESC
LIMIT 5;
```

- Result



	keyword
1	평화
2	국제
3	고려
4	계획
5	개방

- Execution Time Before Indexing

	Query_ID	Duration	Query
▶	37	0.11720950	SELECT keyword FROM (SELECT keyword, COUNT(*) CNT FROM saved WHERE us

- Execution Time After Indexing

	Query_ID	Duration	Query
▶	182	0.04837575	SELECT keyword FROM (SELECT keyword, COUNT(*) CNT FROM saved WHERE us

- Task 5
  - Query

```
WITH A AS (
    SELECT compareDocID FROM similarity
    WHERE docID = (SELECT hash_key
        FROM document
        WHERE post_date LIKE '2014%' AND
        LENGTH(post_title) = (SELECT MAX(LENGTH(post_title))
                FROM document WHERE post_date LIKE '2014%'))
    AND docID <> compareDocID
    ORDER BY Score DESC
    LIMIT 3)
SELECT post_title, post_writer
FROM A JOIN document ON A.compareDocID = document.hash_key ;
```

- Result

	post_title	post_writer
1	中國-臺灣間 交流.協力 現況	孫仁燮
2	북한 외화벌이 추세와 전망	김석진
3	주간국방논단 제1624-2호: 국방 분야 민간지원 활용에 있어, 호주의 '성과기반계약(PBC)' 제도 분	장지홍, 김진호

- Execution Time Before Indexing

	Query_ID	Duration	Query
▶	38	0.02649950	WITH A AS (SELECT compareDocID FROM similarity WHERE docID = (SELECT

- Execution Time After Indexing

	Query_ID	Duration	Query
▶	180	0.01146850	WITH A AS (SELECT compareDocID FROM similarity WHERE docID = (SELECT

- Task 6
  - Query

```
WITH A AS (SELECT status FROM user
    GROUP BY status HAVING COUNT(status) = 1)
SELECT name, inst, email, user.status
FROM user JOIN A USING(status);
```

  - Result

	name	inst	email	status
1	Charles Johnson	한동대학교	charlesJohnson@hgu.db2022.com	박사
2	Reginald Spaulding	한동대학교	reginaldSpaulding@hgu.db2022.com	연구원
3	Jesse Kubik	재단법인통일과ㄴ	jesseKubik@hgu.db2022.com	기타
4	David Dugue	한동대학교	davidDugue@hgu.db2022.com	석사

  - Execution Time Before Indexing

	Query_ID	Duration	Query
▶	39	0.00216500	with A as (select status from user group by status having c

  - Execution Time After Indexing

	Query_ID	Duration	Query
▶	206	0.00133050	with A as (select status from user group by status having count(status) =

- Task 7
  - Query
    - Before applying indexing

```
SELECT tfidfWord
FROM frequency
JOIN (SELECT hash_key
      FROM document
      WHERE post_writer LIKE '%송인호%') AS A
ON frequency.docID = A.hash_key
ORDER BY Score DESC
LIMIT 5;
```

    - After applying indexing (Updated query for indexing)

```
SELECT tfidfWord
FROM frequency
JOIN (SELECT hash_key
      FROM document
      WHERE MATCH(post_writer) AGAINST ('송인호')) AS A
ON frequency.docID = A.hash_key
ORDER BY Score DESC
LIMIT 5;
```

  - Result

	tfidfWord
1	급속
2	결론
3	가치
4	개념
5	개선

  - Execution Time Before Indexing

	Query_ID	Duration	Query
▶	42	0.01673350	SELECT tfidfWord FROM frequency JOIN (SELECT hash_key FROM d

- Execution Time After Indexing

Query_ID	Duration	Query
236	0.00148000	SELECT tfidfWord FROM frequency JOIN (SELECT hash_key FROM do

- Task 8
  - Query

```
WITH D AS (
  WITH C AS (
    WITH B AS (
      WITH A AS (
       SELECT docID FROM frequency WHERE tfidfWord = '국민'
         ORDER BY Score DESC LIMIT 1)
       SELECT post_writer FROM A JOIN document
       ON A.docID = hash_key)
    SELECT hash_key FROM document JOIN B
    ON document.post_writer = B.post_writer)
  SELECT * FROM similarity JOIN C ON similarity.docID = C.hash_key
  AND similarity.docID <> similarity.compareDocID)
SELECT post_title, post_writer, post_date
FROM document JOIN D ON document.hash_key = D.compareDocID
ORDER BY Score DESC
LIMIT 5;
```

  - Result

	post_title	post_writer	post_date
1	South Koreans and Their Neighbors 2019	아산정책연구원	2019-04-26
2	한미 외교장관 전화통화	외교부	2017-09-15
3	South Korea's Middle-Power Diplomacy on the Post-2015 Development A…	Taekyoon Kim	2015-10-13
4	Northeast Asia and the International Criminal Court: Causes and Con…	Alexander Dukalskis	2015-04-08
5	북 비무장지대(DMZ) 지뢰도발 관련 대북경고성명	국방부	2015-08-10

  - Execution Time Before Indexing

	Query_ID	Duration	Query
▶	43	0.35989025	explain with D as (with C as (with B as (with A as (select docID from frequency

- Execution Time After Indexing

	Query_ID	Duration	Query
▶	288	0.00984075	with D as (with C as (with B as (with A as (select docID from frequency where tf

- Task 9
  - Query

```
WITH A AS
        (SELECT topic, COUNT(topic) CNT2015
        FROM document JOIN search
          ON document.hash_key = search.hashKey
        WHERE post_date LIKE '2015%'
        GROUP BY topic),
B AS
        (SELECT topic, COUNT(topic) CNT2020
        FROM document JOIN search
          ON document.hash_key = search.hashKey
        WHERE post_date LIKE '2020%'
        GROUP BY topic)
SELECT topic, CNT2015, IFNULL(CNT2020, 0) CNT2020
FROM A LEFT OUTER JOIN B USING(topic)
WHERE topic IS NOT NULL;
```

  - Result

	topic	CNT2015	CNT2020
1	문화	100	623
2	정치	114	50
3	사회	4	5
4	IT_과학	27	86
5	경제	20	4
6	스포츠	2	0
7	국제	7	8

- Execution Time Before Indexing

Query_ID	Duration	Query
44	0.06848500	with A as (select topic, count(topic) CNT2015 from document join search on docum

- Execution Time After Indexing

Query_ID	Duration	Query
356	0.01907550	with A as (select topic, count(topic) CNT2015 from document join search on d

- Task 10
  - Query

```
SELECT tfidfWord, COUNT(tfidfWord) AS word_count,
RANK() OVER (ORDER BY count(tfidfWord) DESC) AS word_rank
FROM frequency
GROUP BY tfidfWord;
```

  - Result (Cut in line 6)

	tfidfWord	word_count	word_rank
1	경제	6911	1
2	가능	6377	2
3	국가	6252	3
4	결과	5884	4
5	관계	5883	5
6	경우	5845	6

  - Execution Time Before Indexing

	Query_ID	Duration	Query
▶	45	3.91295525	select tfidfWord, count(tfidfWord) as word_count, rank() over (order by count(t

- Execution Time After Indexing

	Query_ID	Duration	Query
▶	329	2.98089400	select tfidfWord, count(tfidfWord) as word_count, rank() over (order by count(t

2) List of Added Index

- document
  - date_index (task 2, 5, 9)
    : post_date index
  - writer_index (task 8)
    : post_writer index
  - writer_fulltext_index (task 7)
    : post_writer fulltext index

- frequency
  - tdidf_index (task 10)
    : tfidfWord index
  - score_tfidf_index (task 8)
    : Score + tfidfWord index

- saved
  - date_id_index (task 3, 4)
    : savedDate + userID index

- search
  - topic_index (task 1)
    : topic index
  - key_topic_index (task 9)
    : topic + hashKey index

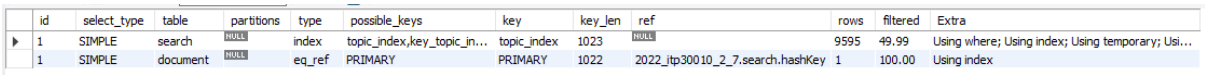
- user
    - status_index (task 6)

      : status index

4. Justification of Indexing

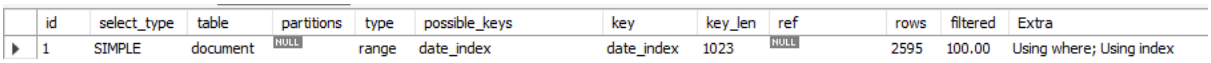
In the process of selecting specific columns to be used as the search_key of the indexes, the focus was on the column mainly used in the JOIN operation and the WHERE clause. After creating indexes on the selected columns based on the above criteria, the execution time before and after creation was compared. The index was adopted only if there was an improvement in the execution time (Adopted indexes are listed above). Therefore, it can be said that there was a logical and reasonable process for selecting and creating indexes.

- Task 1



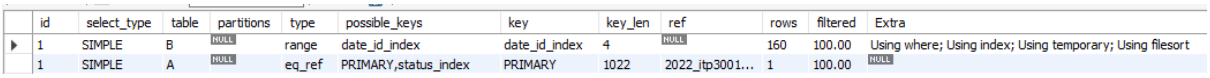
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	search	NULL	index	topic_index,key_topic_in...	topic_index	1023	NULL	9595	49.99	Using where; Using index; Using temporary; Usi...
1	SIMPLE	document	NULL	eq_ref	PRIMARY	PRIMARY	1022	2022_itp30010_2_7.search.hashKey	1	100.00	Using index

- Task 2



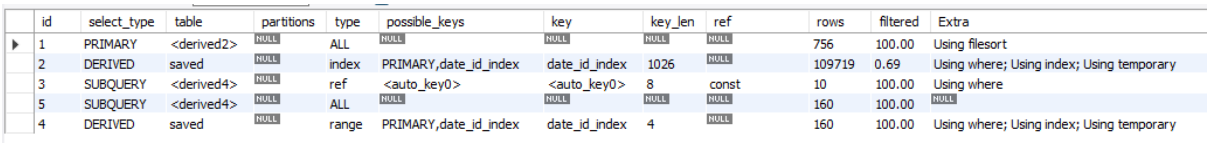
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	document	NULL	range	date_index	date_index	1023	NULL	2595	100.00	Using where; Using index

- Task 3



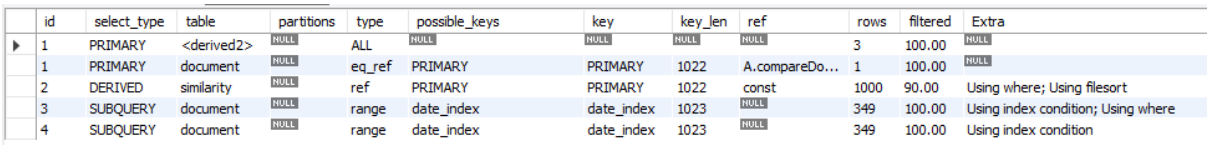
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	B	NULL	range	date_id_index	date_id_index	4	NULL	160	100.00	Using where; Using index; Using temporary; Using filesort
1	SIMPLE	A	NULL	eq_ref	PRIMARY,status_index	PRIMARY	1022	2022_itp3001...	1	100.00	NULL

- Task 4



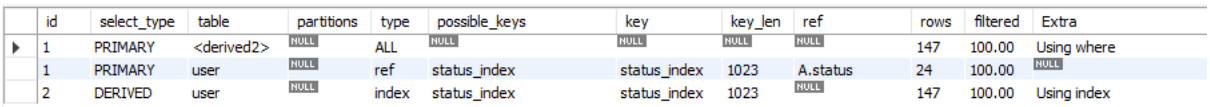
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	756	100.00	Using filesort
2	DERIVED	saved	NULL	index	PRIMARY,date_id_index	date_id_index	1026	NULL	109719	0.69	Using where; Using index; Using temporary
3	SUBQUERY	<derived4>	NULL	ref	<auto_key0>	<auto_key0>	8	const	10	100.00	Using where
5	SUBQUERY	<derived4>	NULL	ALL	NULL	NULL	NULL	NULL	160	100.00	NULL
4	DERIVED	saved	NULL	range	PRIMARY,date_id_index	date_id_index	4	NULL	160	100.00	Using where; Using index; Using temporary

- Task 5



id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	3	100.00	NULL
1	PRIMARY	document	NULL	eq_ref	PRIMARY	PRIMARY	1022	A.compareDo...	1	100.00	NULL
2	DERIVED	similarity	NULL	ref	PRIMARY	PRIMARY	1022	const	1000	90.00	Using where; Using filesort
3	SUBQUERY	document	NULL	range	date_index	date_index	1023	NULL	349	100.00	Using index condition; Using where
4	SUBQUERY	document	NULL	range	date_index	date_index	1023	NULL	349	100.00	Using index condition

- Task 6



id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	147	100.00	Using where
1	PRIMARY	user	NULL	ref	status_index	status_index	1023	A.status	24	100.00	NULL
2	DERIVED	user	NULL	index	status_index	status_index	1023	NULL	147	100.00	Using index

- Task 7

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	document	NULL	fulltext	PRIMARY,writer_fulltext...	writer_fulltext_index	0	const	1	100.00	Using where; Ft_hints: no_ranking; Using temporary; Using f...
1	SIMPLE	frequency	NULL	ref	PRIMARY	PRIMARY	1022	2022_itp3001...	66	100.00	NULL

- Task 8

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	<derived5>	NULL	system	NULL	NULL	NULL	NULL	1	100.00	Using temporary; Using filesort
1	PRIMARY	document	NULL	const	PRIMARY,writer_index,w...	PRIMARY	1022	const	1	100.00	NULL
1	PRIMARY	document	NULL	ref	PRIMARY,writer_index,w...	writer_index	1023	const	4	100.00	Using index
1	PRIMARY	similarity	NULL	ref	PRIMARY	PRIMARY	1022	2022_itp3001...	995	90.00	Using where
1	PRIMARY	document	NULL	eq_ref	PRIMARY	PRIMARY	1022	2022_itp3001...	1	100.00	NULL
5	DERIVED	frequency	NULL	index	tfidf_index,score_tfidf_i...	score_tfidf_index	1031	NULL	104	0.95	Using where; Backward index scan; Using index

- Task 9

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	850	100.00	NULL
1	PRIMARY	<derived3>	NULL	ALL	NULL	NULL	NULL	NULL	1210	100.00	Using where; Using join buffer (hash join)
3	DERIVED	document	NULL	range	PRIMARY,date_index	date_index	1023	NULL	1208	100.00	Using where; Using index; Using temporary
3	DERIVED	search	NULL	ref	topic_index,key_topic_in...	key_topic_index	1022	2022_itp3001...	1	100.00	Using index
2	DERIVED	document	NULL	range	PRIMARY,date_index	date_index	1023	NULL	1698	100.00	Using where; Using index; Using temporary
2	DERIVED	search	NULL	ref	topic_index,key_topic_in...	key_topic_index	1022	2022_itp3001...	1	49.99	Using where; Using index

- Task 10

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	frequency	NULL	index	PRIMARY,tfidf_index,sco...	tfidf_index	1022	NULL	813763	100.00	Using index; Using temporary; Using filesort

5. Summary of the database size and table sizes (in Kilobytes)

1) Table sizes

- Query

```
SELECT TABLE_SCHEMA, TABLE_NAME,
ROUND(DATA_LENGTH/(1024), 1) AS 'data(KB)',
ROUND(INDEX_LENGTH/(1024), 1) AS 'idx(KB)'
FROM information_schema.tables
WHERE TABLE_TYPE = 'BASE TABLE'
AND TABLE_SCHEMA = '2022_itp30010_2_7';
```

- Result

	TABLE_SCHEMA	TABLE_NAME	data(KB)	idx(KB)
▶	2022_itp30010_2_7	bulletin	80.0	16.0
	2022_itp30010_2_7	document	267840.0	5168.0
	2022_itp30010_2_7	frequency	108544.0	0.0
	2022_itp30010_2_7	institution	16.0	0.0
	2022_itp30010_2_7	saved	16992.0	7744.0
	2022_itp30010_2_7	search	1552.0	1552.0
	2022_itp30010_2_7	similarity	72448.0	0.0
	2022_itp30010_2_7	user	48.0	0.0

2) Database size

- Query

```
SELECT table_schema AS 'DatabaseName',
ROUND(SUM(data_length+index_length)/1024, 1) AS 'Size(KB)'
FROM information_schema.tables
WHERE table_schema = '2022_itp30010_2_7'
GROUP BY table_schema;
```

- Result

	DatabaseName	Size(KB)
▶	2022_itp30010_2_7	482000.0