Reporting Tables Analysis

Sprint 4: MagazineFeeds, Sentiment_Analysis_keywords, and SentimentAnlyResult Stored Procedure Analysis

Overview

This report explains the structure and relationships between the tables MagazineFeeds, Sentiment_Analysis_keywords, SentimentAnlyResult stored procedures. It describes what each table stores, the indexes they use, what issues were found, and how to improve them.

MagazineFeeds

The MagazineFeeds table keeps information about each magazine article, such as the link of the article, name of the magazine and link of the magazine. It is a master list of all the magazine articles.

Column Name	Data Type	Description
URL	VARCHAR(1000) (Primary Key)	A unique link to the article. (Not NULL)
MagazineName	VARCHAR(150)	The name of the magazine.
MagazineLink	VARCHAR(150)	Link of the magazine in which the article was published

Indexes

Index Name	Type	Purpose
PRIMARY	PRIMARY, BTree on URL	The main index automatically created on the URL column. The link is unique and is searchable.

URL_UNIQUE	BTree on URL	Adds a uniqueness to the url column

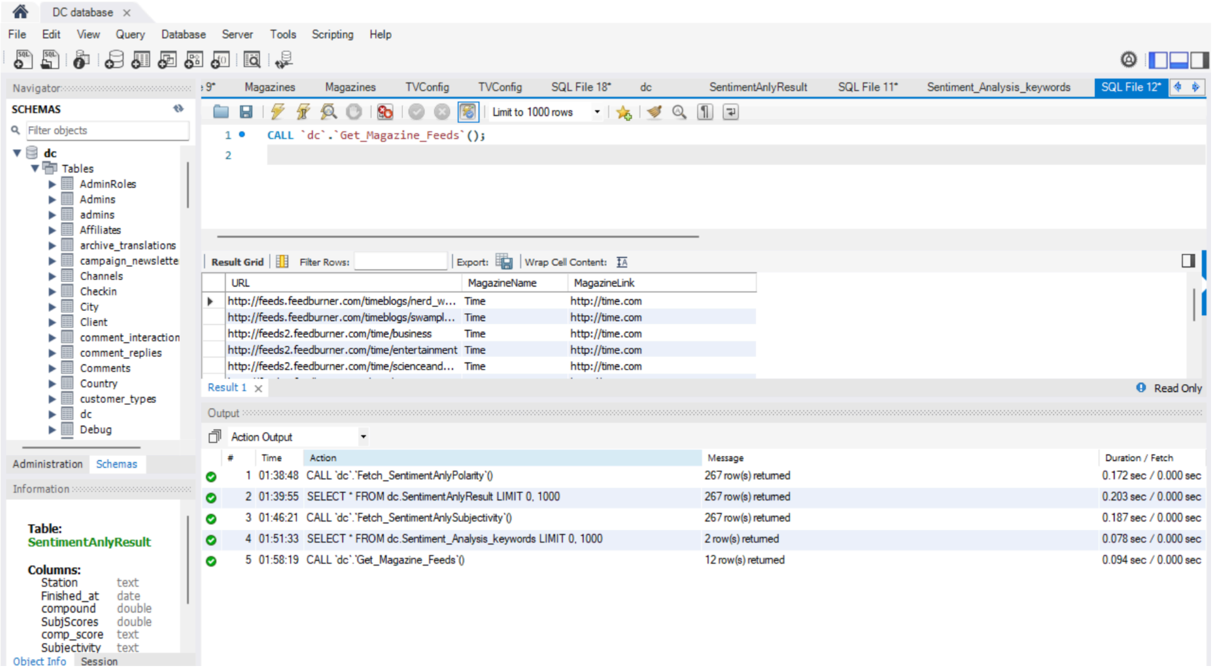
Stored Procedures

Name	Purpose
Get_Magazine_Feeds	Returns the tables with url of the article, name and link of the magazine

Observations / Issues

1. **The primary key being the url**
   URLs are long text fields and using them as primary keys leads to slow indexing and higher memory usage.
2. **URL_UNIQUE index on the same column as the PRIMARY**
   The PRIMARY KEY already implies uniqueness. The UNIQUE constraint adds a duplicate index enforcing the same rule.
3. **No foreign keys. The table is not connected to anything.**
   It is very hard to get information about this table and no other tables can access this table neither for receiving or storing data. The data is isolated.

Stored Procedure Analysis:



```
CALL `dc`.`Get_Magazine_Feeds`();
```

URL	MagazineName	MagazineLink
http://feeds.feedburner.com/timeblogs/nerd_w...	Time	http://time.com
http://feeds.feedburner.com/timeblogs/swampl...	Time	http://time.com
http://feeds2.feedburner.com/time/business	Time	http://time.com
http://feeds2.feedburner.com/time/entertainment	Time	http://time.com
http://feeds2.feedburner.com/time/scienceand...	Time	http://time.com

```
#  Time      Action                                                    Message              Duration / Fetch
1  01:38:48  CALL `dc`.`Fetch_SentimentAnlyPolarity`()                  267 row(s) returned  0.172 sec / 0.000 sec
2  01:39:55  SELECT * FROM dc.SentimentAnlyResult LIMIT 0, 1000         267 row(s) returned  0.203 sec / 0.000 sec
3  01:46:21  CALL `dc`.`Fetch_SentimentAnlySubjectivity`()              267 row(s) returned  0.187 sec / 0.000 sec
4  01:51:33  SELECT * FROM dc.Sentiment_Analysis_keywords LIMIT 0, 1000 2 row(s) returned    0.078 sec / 0.000 sec
5  01:58:19  CALL `dc`.`Get_Magazine_Feeds`()                           12 row(s) returned   0.094 sec / 0.000 sec
```

Table: SentimentAnlyResult

Columns:

Station	text
Finished_at	date
compound	double
SubjScores	double
comp_score	text
Subjectivity	text

Returned the tables with url of the article, name and link of the magazine

Recommendations

- URL_UNIQUE index can be dropped
- Reconsider the usage of URL as the PRIMARY index
- Reoptimize the connection between the tables and add foreign keys accordingly.

Sentiment_Analysis_Keywords [TO BE DELETED]

The **Sentiment_Analysis_Keywords** table keeps information about the content of the media, especially the keywords associated with them. It has ID, keyword, and submission time. Currently, it only holds testing data that previous semesters' teams had once used.

Column Name	Data Type	Description
ID	INT	A unique id assigned to these keywords
Keyword	VARCHAR(20)	The keywords associated with it.
Submission_Time	datetime	Time when the content was analyzed

Indexes

Index Name	Type	Purpose
ID	PRIMARY	The main index automatically created on the ID column.

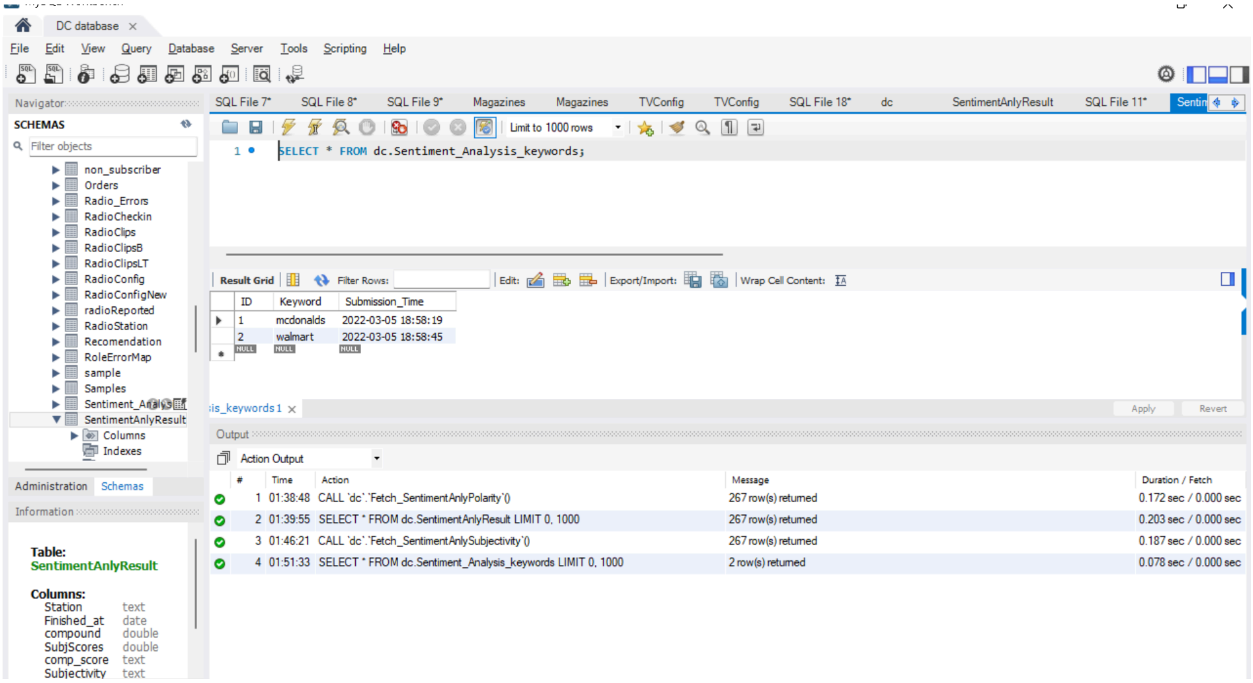
Observations / Issues

1. **No relationships with source tables:**

The table is not linked to any feed table (e.g., NewspaperFeeds or MagazineFeeds). As a result, it's unclear which article or content each keyword belongs to.

2. **Limited keyword size:**
   The Keyword column uses VARCHAR(20), which may be too short to store multi-word or compound keywords used in sentiment analysis.
3. **No indexing on Keyword or Submission_Time:**
   Queries filtering by keyword or analyzing time-based patterns will perform slowly without indexes on these columns.
4. **No foreign key constraints:**
   Without foreign keys to the analyzed content (article or feed), there's no referential integrity between keywords and their sources.

Data query:



```
SELECT * FROM dc.Sentiment_Analysis_keywords;
```

ID	Keyword	Submission_Time
1	mcdonalds	2022-03-05 18:58:19
2	walmart	2022-03-05 18:58:45
NULL	NULL	NULL

```
#  Time      Action                                                        Message              Duration / Fetch
1  01:38:48  CALL `dc`.`Fetch_SentimentAnlyPolarity`()                     267 row(s) returned  0.172 sec / 0.000 sec
2  01:39:55  SELECT * FROM dc.SentimentAnlyResult LIMIT 0, 1000            267 row(s) returned  0.203 sec / 0.000 sec
3  01:46:21  CALL `dc`.`Fetch_SentimentAnlySubjectivity`()                 267 row(s) returned  0.187 sec / 0.000 sec
4  01:51:33  SELECT * FROM dc.Sentiment_Analysis_keywords LIMIT 0, 1000    2 row(s) returned    0.078 sec / 0.000 sec
```

Table:
SentimentAnlyResult

Columns:

```
Station      text
Finished_at  date
compound     double
SubjScores   double
comp_score   text
Subjectivity text
```

Observation: The query worked and returned 2 rows in 0.078

Issue: There are only 2 rows of data in this table and because the data is isolated most of it is outdated.

Recommendations

- Add a FeedID or ArticleID column that references either NewspaperFeeds or MagazineFeeds depending on the source.
- Add indexes on Keyword and Submission_Time to improve query performance.

- Increase the Keyword column size to VARCHAR(100) for flexibility.
- Add NOT NULL constraints to Keyword and Submission_Time.

SentimentAnlyResult

Purpose

The **SentimentAnlyResult** table keeps information about the results of the sentiment analysis done on media content (specifically radio).

Column Name	Data Type	Description
Station	Text	Name or identifier of the radio station from which the analyzed content originated. Used to group or filter sentiment results by source.
Finished_at	date	The date when the sentiment analysis process for this record was completed. Useful for tracking analysis timelines or historical trends.
compound	double	The **overall sentiment score** calculated by the sentiment analysis algorithm. Typically ranges from -1 (most negative) to +1 (most positive).
SubjScores	double	The **subjectivity score**, representing how subjective or opinionated the content is (e.g., 0 = objective, 1 = highly subjective).
comp_scores	text	Raw or component sentiment scores (e.g., individual positive/negative/neutral values) stored as text or JSON for detailed inspection.

Subjectivity	text	Qualitative label representing the level of subjectivity (Objective, Subjective, Mixed). Often derived from the SubjScores value.

Indexes: Has no indexes

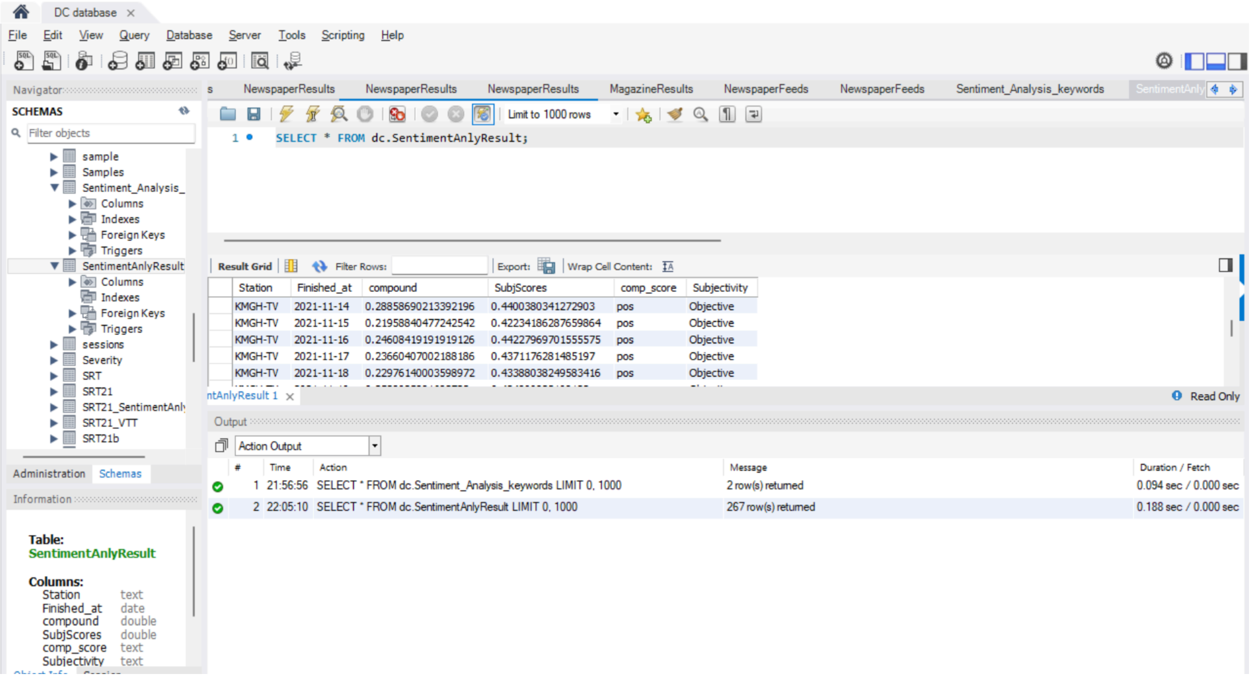
Stored Procedures

Name	Purpose
Fetch_SentimentAnlyPolarity	Returns the formatted date as a table from the SentimentAnlyResult. Along with component sentiment and subjectivity scores.
Fetch_SentimentAnlySubjectivity	Returns the formatted date as a table from the SentimentAnlyResult. Along with subjectivity scores and a subjectivity categorical bucket.

Observations / Issues

1. **No relationship with feed tables:**
   The Station field is stored as text and does not reference NewspaperFeeds or MagazineFeeds. This breaks data integrity and makes it impossible to determine which feed each sentiment result belongs to.
2. **Lack of indexing:**
   The table has no indexes, which will degrade performance as the number of records grows.
3. **Data type inconsistencies:**
   Fields like comp_scores and Subjectivity are stored as text but contain structured or numerical data that could be stored as JSON or numeric types.

Data query:



```sql
SELECT * FROM dc.SentimentAnlyResult;
```

Station	Finished_at	compound	SubjScores	comp_score	Subjectivity
KMGH-TV	2021-11-14	0.28858690213392196	0.4400380341272903	pos	Objective
KMGH-TV	2021-11-15	0.21958840477242542	0.42234186287659864	pos	Objective
KMGH-TV	2021-11-16	0.24608419191919126	0.44227969701555575	pos	Objective
KMGH-TV	2021-11-17	0.23660407002188186	0.4371176281485197	pos	Objective
KMGH-TV	2021-11-18	0.22976140003598972	0.43388038249583416	pos	Objective

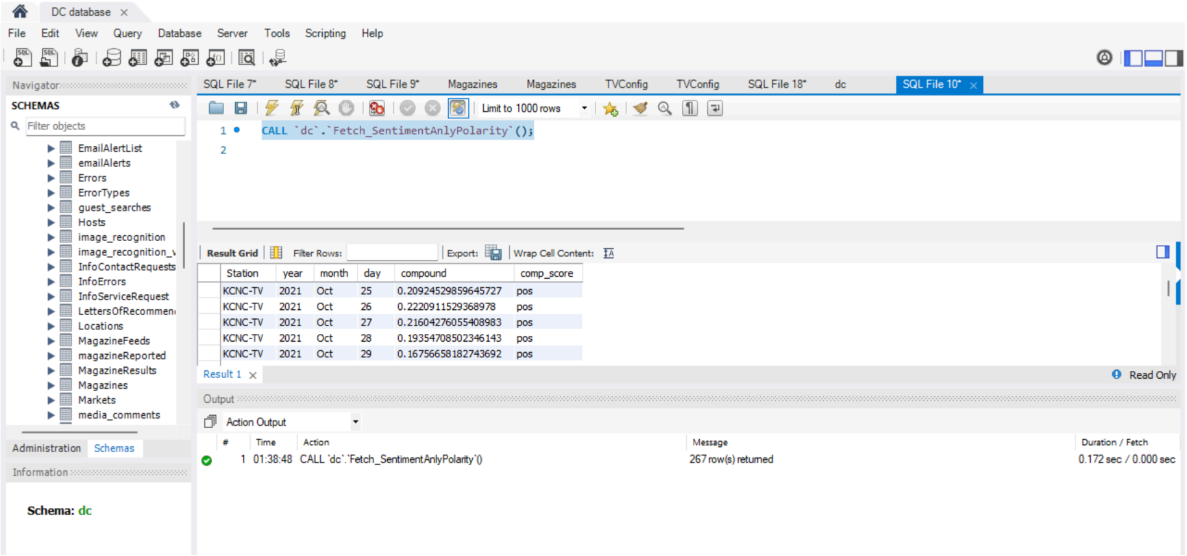
```
#  Time      Action                                                    Message            Duration / Fetch
1  21:56:56  SELECT * FROM dc.Sentiment_Analysis_keywords LIMIT 0, 1000  2 row(s) returned    0.094 sec / 0.000 sec
2  22:05:10  SELECT * FROM dc.SentimentAnlyResult LIMIT 0, 1000         267 row(s) returned  0.188 sec / 0.000 sec
```

Table: SentimentAnlyResult

Columns:

Station	text
Finished_at	date
compound	double
SubjScores	double
comp_score	text
Subjectivity	text

Stored Procedures:

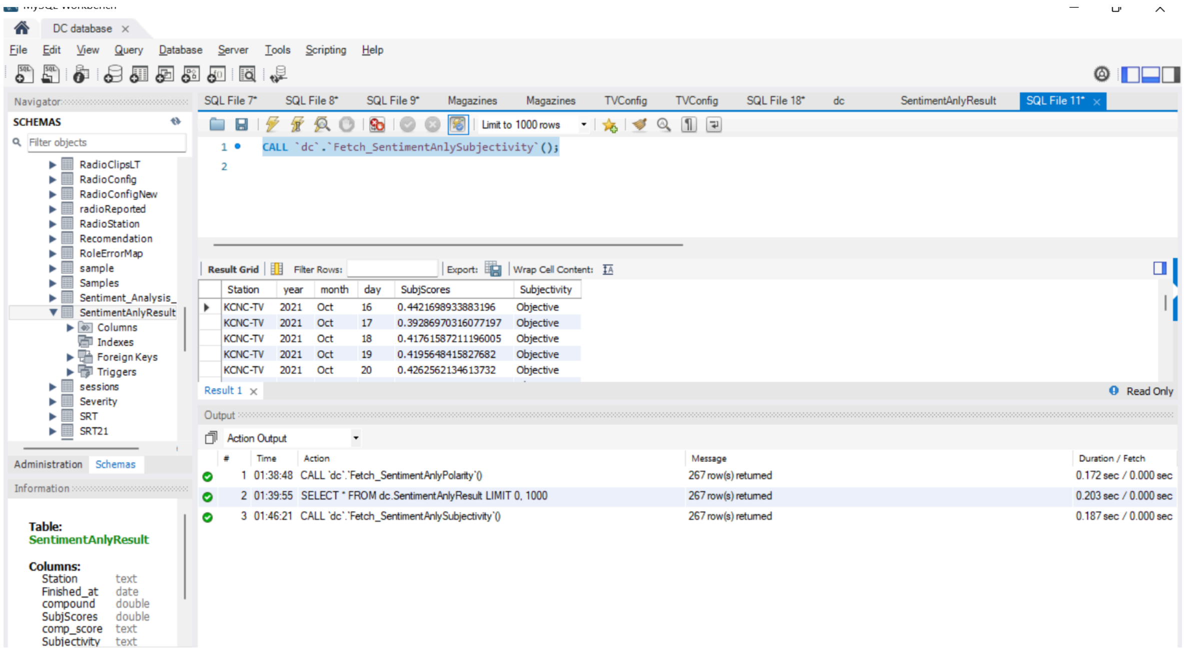


```sql
CALL `dc`.`Fetch_SentimentAnlyPolarity`();
```

Station	year	month	day	compound	comp_score
KCNC-TV	2021	Oct	25	0.20924529859645727	pos
KCNC-TV	2021	Oct	26	0.2220911529368978	pos
KCNC-TV	2021	Oct	27	0.21604276055408983	pos
KCNC-TV	2021	Oct	28	0.19354708502346143	pos
KCNC-TV	2021	Oct	29	0.16756658182743692	pos

```
#  Time      Action                                    Message              Duration / Fetch
1  01:38:48  CALL `dc`.`Fetch_SentimentAnlyPolarity`()  267 row(s) returned  0.172 sec / 0.000 sec
```

Schema: dc



```
CALL `dc`.`Fetch_SentimentAnlySubjectivity`();
```

Station	year	month	day	SubjScores	Subjectivity
KCNC-TV	2021	Oct	16	0.4421698933883196	Objective
KCNC-TV	2021	Oct	17	0.39286970316077197	Objective
KCNC-TV	2021	Oct	18	0.41761587211196005	Objective
KCNC-TV	2021	Oct	19	0.4195648415827682	Objective
KCNC-TV	2021	Oct	20	0.4262562134613732	Objective

#	Time	Action	Message	Duration / Fetch
1	01:38:48	CALL 'dc'.'Fetch_SentimentAnlyPolarity'()	267 row(s) returned	0.172 sec / 0.000 sec
2	01:39:55	SELECT * FROM dc.SentimentAnlyResult LIMIT 0, 1000	267 row(s) returned	0.203 sec / 0.000 sec
3	01:46:21	CALL 'dc'.'Fetch_SentimentAnlySubjectivity'()	267 row(s) returned	0.187 sec / 0.000 sec

Table: SentimentAnlyResult

Columns:

Station	text
Finished_at	date
compound	double
SubjScores	double
comp_score	text
Subjectivity	text

Recommendations

- Add a FeedID column referencing NewspaperFeeds or MagazineFeeds to associate results with articles.
- Add indexes on FeedID and Finished_at to optimize filtering and lookups.
- Convert comp_scores to JSON and Subjectivity to ENUM ('Objective', 'Subjective', 'Mixed').
- Add NOT NULL constraints to key columns (FeedID, compound, Finished_at).
- Add foreign key relationships to Sentiment_Analysis_Keywords to link results with the keywords used.
- Add created_at and updated_at timestamps.

Move **Categories** into its own table for better organization and easier searches.

Recommended Structure

- Create a unified relationship structure where feeds (newspaper and magazine) act as the core data sources, and sentiment/keyword tables depend on them.
- MagazineFeeds: Act as master content sources.
- Sentiment_Analysis_Keywords: Links to feed tables through FeedID (foreign key).
- SentimentAnlyResult: Links to both FeedID (source article) and KeywordID (keyword used).

One-to-Many Relationships:

- One article (in NewspaperFeeds or MagazineFeeds) → Many sentiment keywords
- One keyword → Many sentiment results.
- One article → Many sentiment results (indirectly via keywords).

Benefits of This Structure

- Enables sentiment analysis across all article types.
- Ensures referential integrity (no orphan sentiment or keyword entries).
- Simplifies analytical queries (e.g., sentiment trends by publisher or keyword).
- Improves maintainability and scalability as data volume grows.

Data Type and Index Improvements

- Use INT AUTO_INCREMENT for all ID fields.
- Convert URL fields from primary to unique indexes.
- Change Finished_at and Submission_Time to DATETIME.
- Replace long TEXT fields with VARCHAR(255) where appropriate.
- Use ENUM or JSON for categorical and structured sentiment data.

Indexes to Add

- On FeedID in Sentiment_Analysis_Keywords and SentimentAnlyResult
- On Keyword in Sentiment_Analysis_Keywords
- On Finished_at in SentimentAnlyResult
- On ArticleID in both feed tables

Recommendations Summary

1. Introduce surrogate keys (ArticleID, FeedID, KeywordID, ResultID).
2. Establish foreign key relationships among all four tables.
3. Convert text timestamps to DATETIME.
4. Add NOT NULL constraints on essential columns.
5. Simplify indexes by keeping only the most relevant.
6. Normalize repeated large text fields.
7. Delete the Sentiment_Analysis_Keywords table

Conclusion

The reporting tables provide useful data independently but lack the relational structure needed for efficient cross analysis. By establishing foreign key relationships, and optimizing data types, indexes, and the stored procedures the database can support faster queries, better integrity, and scalable analytics across newspaper and magazine content.