

Reporting Tables Analysis

Sprint 5: Magazines, Newspapers and Markets Analysis

Overview

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

Magazines

The Magazines table keeps information about each magazine, name of the magazine, link of the magazine and records of where the magazine is published. It is a master list of all the magazines.

Column Name	Data Type	Description
ID	INT	A unique ID.
URL	VARCHAR(1000)	Link of the magazine.
MagazineName	VARCHAR(150)	The name of the magazine.
Country	VARCHAR(150)	Country in which the magazine is published
City	VARCHAR(150)	City in which the magazine is published
StateProvince	VARCHAR(100)	State in which the magazine is published

Indexes

Index Name	Type	Purpose
PRIMARY	PRIMARY, BTree on ID	The main index automatically created on the ID column. The ID is unique and is searchable.
MagazineName_UNIQUE	BTree	Adds a uniqueness to the MagazineName column. Makes the name of the magazines searchable.

Stored Procedures

Name	Purpose
Get_Magazines	Returns the names of magazines from the magazines table
Magazine_Search_Emailalerts	Searches the email alerts sent using location
Magazine_Cleanup	Cleans out MagazineResults table using number of days
Insert_Magazine	Inserts a new row in the magazines table.

Observations / Issues

1. No Validation on URL Quality

URL is stored as VARCHAR(1000) with no format validation.
The table may accumulate invalid or malformed URLs.

2. Potential Duplicate Geographic Data

Country, StateProvince, and City are stored as free-text, increasing risk of inconsistencies (e.g., “USA”, “U.S.A.”, “United States”).

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.

Data query:

The screenshot shows the SQL Server Management Studio interface. The left pane displays the database schema, including tables like ErrorTypes, Host, and dc.Magazines. The right pane shows a query window with the following content:

```
File Edit View Query Database Server Tools Scripting Help
Navigator: DC database
Schemas: Filter objects
Table: SentimentOnlyResult
Columns: Station text, Finished_at date, compound double, SubScores double, comp_score text, Subjectivity text
Result Grid: ID, MagazineName, Country, StateProvince, City, URL
Output: Action Output
```

Result Grid:

ID	MagazineName	Country	StateProvince	City	URL
1	Time	United States	New York	New York City	http://tme.com
2	WIRED	United States	California	San Francisco	https://www.wired.com/
3	Rolling Stone	United States	New York	New York City	https://www.rollingstone.com/
4	New York Magazine	United States	New York	New York City	http://nymag.com/
5	The New Yorker	United States	New York	New York City	https://www.newyorker.com/

Action Output:

#	Time	Action	Message	Duration / Fetch
1	01:38:48	CALL `dc`.`Fetch_SentimentOnlyPolarity`()	267 row(s) returned	0.172 sec / 0.000 sec
2	01:39:55	SELECT * FROM dc.SentimentOnlyResult LIMIT 0, 1000	267 row(s) returned	0.203 sec / 0.000 sec
3	01:46:21	CALL `dc`.`Fetch_SentimentOnlySubjectivity`()	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
6	02:16:10	SELECT * FROM dc.Magazines LIMIT 0, 1000	15 row(s) returned	0.109 sec / 0.047 sec

Data Query Speed: Returned 15 rows in 0.109 seconds

Stored Procedures Analysis

The screenshot shows the SQL Server Management Studio interface. The left pane displays the database schema, including tables like dc.Magazines and dc.Newspapers. The right pane shows a query window with the following content:

```
File Edit View Query Database Server Tools Scripting Help
Navigator: DC database
Schemas: Filter objects
Table: Newspapers
Columns: ID int AI, NewspaperName varchar, Country varchar, StateProvince varchar, City varchar, URI varchar
Result Grid: MagazineName
Output: Action Output
```

Result Grid:

MagazineName
AARP The Magazine
Cosmos
ESPN The Magazine
Game Informer
Good Housekeeping

Action Output:

#	Time	Action	Message	Duration / Fetch
1	01:38:48	CALL `dc`.`Fetch_SentimentOnlyPolarity`()	267 row(s) returned	0.172 sec / 0.000 sec
2	01:39:55	SELECT * FROM dc.SentimentOnlyResult LIMIT 0, 1000	267 row(s) returned	0.203 sec / 0.000 sec
3	01:46:21	CALL `dc`.`Fetch_SentimentOnlySubjectivity`()	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
6	02:16:10	SELECT * FROM dc.Magazines LIMIT 0, 1000	15 row(s) returned	0.109 sec / 0.047 sec
7	10:54:58	SELECT * FROM dc.Newspapers LIMIT 0, 1000	37 row(s) returned	0.125 sec / 0.000 sec
8	11:21:06	CALL `dc`.`Get_Magazines`()	15 row(s) returned	0.141 sec / 0.000 sec

Schemas

Name: Magazine_Search

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

```

28     WHERE  (All_Magazines_param OR Selected_Magazine_param = Magazine) AND
29         MATCH (Summary, Title, Categories) AGAINST (Search_Query_param IN BOOLEAN MODE)
30         AND Magazine = MagazineName
31         AND ((Selected_City = City) OR
32             (All_Cities AND (Selected_State_Province = StateProvince)) OR
33             ((All_Cities AND All_States) AND (Selected_Country = Country)) OR
34             (All_Cities AND All_States AND All_Countries))
35         AND ((STR_TO_DATE(Start_DateTime, '%m/%d/%Y %h:%i %p') <= PublishDate) OR (Start_DateTime = ''))

```

Routine

Output:

#	Time	Action	Message	Duration / Fetch
1	16:51:04	SELECT * FROM dc.MagazineResults LIMIT 0, 1000	659 row(s) returned	0.484 sec / 2.797 sec

Procedure: Magazine_Search

Object Info Session

Schemas

Name: Magazine_Cleanup

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

```

1 •  CREATE DEFINER='root'@'localhost' PROCEDURE `Magazine_Cleanup`(
2     Days          INT
3 )
4 BEGIN
5     DELETE FROM dc.MagazineResults
6     WHERE (DATEDIFF(NOW(), AddedDate) >= Days) AND (ID <> 0);
7 END

```

Routine

Output:

#	Time	Action	Message	Duration / Fetch
1	16:51:04	SELECT * FROM dc.MagazineResults LIMIT 0, 1000	659 row(s) returned	0.484 sec / 2.797 sec

Procedure: MagazineResult

```

Name: Magazine_Search_Emailalerts
The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:
31 WHERE (All_Magazines_param OR Selected_Magazine_param = Magazine) AND
32 MATCH (Summary, Title, Categories) AGAINST (Search_Query_param IN BOOLEAN MODE)
33 AND Magazine = MagazineName
34 AND ((Selected_City = City) OR
35 (All_Cities AND (Selected_State_Province = StateProvince)) OR
36 ((All_Cities AND All_States) AND (Selected_Country = Country)) OR
37 (All_Cities AND All_States AND All_Countries))
38 AND ((STR_TO_DATE(Start_DateTime, '%m/%d/%Y %h:%i %p') <= PublishDate) OR (Start_DateTime = ''))

```

Output

#	Time	Action	Message	Duration / Fetch
1	16:51:04	SELECT * FROM dc.MagazineResults LIMIT 0, 1000	659 row(s) returned	0.484 sec / 2.797 sec

- The country/state are not normalized

Recommendations

- Add NOT NULL + basic data quality constraints. Mark MagazineName, URL, Country, and City as NOT NULL. Enforce minimum length or non-empty checks.
- Add a CHECK constraint or enforce format validation inside Insert_Magazine.
- Move Country, StateProvince, and City to reference tables with foreign keys.

Newspapers

The Newspapers table keeps information about each newspapers, name of the newspapers, link of the newspaper and records of where the newspaper is published. It is a master list of all the newspapers.

Column Name	Data Type	Description
ID	INT	A unique ID.
URL	VARCHAR(1000)	Link of the newspaper.

NewspaperName	VARCHAR(150)	The name of the newspaper.
Country	VARCHAR(150)	Country in which the newspaper is published
City	VARCHAR(150)	City in which the newspaper is published
StateProvince	VARCHAR(100)	State in which the newspaper is published

Indexes

Index Name	Type	Purpose
PRIMARY	PRIMARY, BTree on ID	The main index automatically created on the ID column. The ID is unique and is searchable.
Newspaper Name_UNI QUE	BTree	Adds a uniqueness to the NewspaperName column. Makes the name of the newspapers searchable.

Stored Procedures

Get_Newspapers	Returns the names of newspapers from the magazines table
Newspaper_Search_Emailalerts	Searches the email alerts sent using location
Newspaper_Cleanup	Cleans out NewspaperResults table using number of days
Insert_Newspaper	Inserts a new row in the Newspaper table.

Observations / Issues

1. No Validation on URL Quality

URL is stored as VARCHAR(1000) with no format validation.

The table may accumulate invalid or malformed URLs.

2. Potential Duplicate Geographic Data

Country, StateProvince, and City are stored as free-text, increasing risk of inconsistencies (e.g., "USA", "U.S.A.", "United States").

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.

Data query:

The screenshot shows a database management interface with the following details:

- Schemas:** A tree view of schemas including EmailAlertList, emailAlerts, Errors, ErrorTypes, guest_searches, Hosts, image_recognition, image_recognition_v, InfoContactRequests, InfoErrors, InfoServiceRequest, LettersOfRecommen, Locations, MagazineFeeds, magazineReported, MagazineResults, Magazines, Markets, and media_comments.
- Query Bar:** Shows the query `SELECT * FROM dc.Newspapers`.
- Result Grid:** Displays the results of the query in a table format:

ID	NewspaperName	Country	StateProvince	City	URL
1	The New York Times	United States	New York	New York City	https://www.nytimes.com/
2	The Washington Post	United States	District of Columbia	Washington D.C.	https://www.washingtonpost.com/
3	The Wall Street Journal	United States	New York	New York City	https://www.wsj.com/
4	Los Angeles Times	United States	California	Los Angeles	http://www.latimes.com/
5	Chicago Tribune	United States	Illinois	Chicago	http://www.chicagotribune.com

- Output:** Shows the action history with one entry: "1 20:19:13 SELECT * FROM dc.Newspapers LIMIT 0..1000" which returned 37 rows in 0.094 sec / 0.000 sec.

Observation: The query worked and returned 37 rows in 0.094

Stored Procedures:

SCHEMAS

Filter objects

- Get_Alert_List
- Get_Alert_List_Cron
- get_channel_id
- Get_Channels
- get_channels_with_c
- Get_Cities
- Get_Countries
- Get_ID_and_GUID
- Get_ID_and_GUID_1
- Get_ID_and_GUID_1_
- Get_InfoContactReq
- Get_Magazine_Feed:
- Get_Magazine
- Get_Newspaper_Fee
- Get_Newspapers
- Get_Radio_config
- Get_Radio_configNe
- Get_Radios
- Get_Roles

Administration Schemas

Information

Procedure:
Get_Newspapers

Object Info Session

Query Completed

```
1 • call dc.Get_Newspapers();
2
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result 1 x Read Only

NewspaperName
Chicago Sun-Times
Chicago Tribune
China Daily
Chosun Ilbo
Cincinnati Enquirer
Der Spiegel

Action Output

#	Time	Action	Message	Duration / Fetch
1	20.19.13	SELECT * FROM dc.Newspapers LIMIT 0, 1000	37 row(s) returned	0.094 sec / 0.000 sec
2	20.21.15	call dc.Get_Newspapers()	37 row(s) returned	0.078 sec / 0.000 sec

SCHEMAS

Filter objects

- InsertComment
- InsertTranslation
- Magazine_Cleanup
- Magazine_Search
- Magazine_Search_En
- MagazineResult
- Newspaper_Cleanup
- Newspaper_Search
- Newspaper_Search_J
- NewspaperResult
- Radio_Search
- Radio_Search_Email
- Relevance_Search
- Search_Engine
- Search_TV
- Search_TV_es
- Simple_Search
- Update_Admin_Pass
- Update_Admin_Role

Administration Schemas

Information

Procedure:
Magazine_Search

Object Info Session

Query Completed

Name: Newspaper_Cleanup

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `Newspaper_Cleanup`(
2     Days
3 )
4 BEGIN
5     DELETE FROM dc.NewspaperResults
6     WHERE (DATEDIFF(NOW(), AddedDate) >= Days) AND (ID <> 0);
7 END
```

Routine

Action Output

#	Time	Action	Message	Duration / Fetch
1	16.51.04	SELECT * FROM dc.MagazineResults LIMIT 0, 1000	659 row(s) returned	0.484 sec / 2.797 sec

The screenshot shows the MySQL Workbench interface with two open windows for stored procedures.

Top Window (Get_Newspapers):

- Name:** Get_Newspapers
- DDL:**

```

CREATE PROCEDURE `Get_Newspapers`()
BEGIN
    SELECT * FROM dc.Newspapers;
END

```
- Parameters:**

Search_Query_param	[IN] TEXT
Selected_Newspaper_param	[IN] TEXT
All_Papers_param	[IN] BOOL
Selected_Country	[IN] TEXT
All_Countries	[IN] BOOL
Selected_State_Province	[IN] TEXT
All_States	[IN] BOOL
Selected_City	[IN] TEXT
All_Cities	[IN] BOOL
Start_DateTime	[IN] TEXT
End_DateTime	[IN] TEXT
Limit_param	[IN] INT
emailAlertID	[IN] INT
- Output:**

#	Time	Action	Message	Duration / Fetch
1	20:19:13	SELECT * FROM dc.Newspapers	37 row(s) returned	0.094 sec / 0.000 sec
2	20:21:15	call dc.Get_Newspapers()	37 row(s) returned	0.078 sec / 0.000 sec

Bottom Window (Insert_Newspaper):

- Name:** Insert_Newspaper
- DDL:**

```

CREATE DEFINER='root'@'localhost' PROCEDURE `Insert_Newspaper`(
    New_Newspaper_Name TEXT,
    URL_Param TEXT,
    RSS_Param TEXT,
    Country_Param TEXT,
    State_Province_Param TEXT,
    City_Param TEXT
)
BEGIN
    INSERT INTO dc.Newspapers (NewspaperName, Country, StateProvince, City, URL)
    VALUES (New_Newspaper_Name, Country_Param, State_Province_Param, City_Param, URL_Param)
    ON DUPLICATE KEY UPDATE URL = URL_Param;

```
- Output:**

#	Time	Action	Message	Duration / Fetch
1	20:19:13	SELECT * FROM dc.Newspapers LIMIT 0, 1000	37 row(s) returned	0.094 sec / 0.000 sec
2	20:21:15	call dc.Get_Newspapers()	37 row(s) returned	0.078 sec / 0.000 sec

Recommendations

- Add NOT NULL + Basic Data Quality Constraints. Mark NewspaperName, URL, Country, and City as NOT NULL. Enforce minimum length or non-empty checks.
- Add a CHECK constraint or enforce format validation inside Insert_Newspaper.
- Move Country, StateProvince, and City to reference tables with foreign keys.

Markets

Purpose

The **Markets** table was a temporary table to store location of alerts that were sent.

Column Name	Data Type	Description
id	INT	Name or identifier of the location from where media is taken.
Country	VARCHAR(20)	Country of origin
State	VARCHAR(2)	State/ province of origin
City	VARCHAR(40)	City of origin

Indexes:

Index Name	Type	Purpose
PRIMARY	PRIMARY, BTree on Country, State, City	The main index created on these columns is unique
id	BTree	Is a auto generated unique id

Stored Procedures: No stored procedures

Observations / Issues

1. Purpose Is Unclear and is redundant to Locations tables

The table is described as temporary, yet it exists as a normal table with indexes.
Unclear whether it should still exist or has already been replaced.

2. PRIMARY KEY Is Incorrectly Defined

The PRIMARY KEY is on (Country, State, City) instead of on id.

The screenshot shows the Oracle SQL Developer interface. The Navigator pane on the left lists schemas like TVConfig, TVConfigNew, TvSTT, TVTT, updated_Email_Alert, user_preferences, user_roles, users, videos, VTT, and VTTB. The SQL Worksheet at the top contains the query: `SELECT * FROM dc.Markets;`. The Result Grid below shows the following data:

	id	Country	State	City
▶	1	US	CO	Denver
▶	2	US	OH	Columbus
*	NULL	NULL	NULL	NULL

The Output window at the bottom displays the execution log for the query:

Action	Time	Message	Duration / Fetch
1 20:19:13 SELECT * FROM dc.Newspapers LIMIT 0, 1000		37 row(s) returned	0.094 sec / 0.000 sec
2 20:21:15 call dc.Get_Newspapers()		37 row(s) returned	0.078 sec / 0.000 sec
3 20:25:27 SELECT * FROM dc.Markets LIMIT 0, 1000		2 row(s) returned	0.094 sec / 0.000 sec
4 20:25:28 SELECT * FROM dc.Markets LIMIT 0, 1000		2 row(s) returned	0.078 sec / 0.000 sec

No other tests can be run on this data.

Recommendations

- Clarify whether the table should still exist. If obsolete it can be dropped.
- Fix the primary key make sure it is only on id
- Add basic metadata columns like date and time of creation and updates.

Recommended Structure

- Standardize locations across ALL tables. The tables all store Country, StateProvince/State, and City as free-text fields with inconsistent lengths. A unified structure improves data quality, querying, and joins.
- Add a URL format CHECK constraint or validate in Insert_Magazine and Insert_Newspaper.
- Mark Name columns as NOT NULL
- Add foreign keys for Magazines and Newspapers to connect Feeds table

Recommendations Summary

1. Standardize and normalize all geographic fields.
2. Implement NOT NULL constraints for core fields.
3. Improve URL validation.
4. Add foreign keys to reference tables when possible.
5. Improve stored procedures to enforce data quality.

Magazines / Newspapers

1. Fix isolated data problem by linking location fields to normalized tables.
2. Improve constraints to reduce duplicates.
3. Add indexing for location-based queries.

Markets

1. Clarify whether the table is still needed.
2. Correct the primary key.
3. Add metadata fields (CreatedAt, UpdatedAt).
4. Drop table if obsolete or merged with Locations.

Conclusion

The Magazines, Newspapers, and Markets tables provide essential media source and location data, but the current structure allows inconsistencies, duplicated geographic values, outdated entries, and limited connectivity to other parts of the database. By standardizing the location fields, improving data types and constraints, correcting indexes, and enhancing stored procedures, the database becomes more reliable and easier to maintain.