Database Table Analysis Report

InfoErrors, SRT21, SRT21b

1. Overview

This report provides a structured analysis of the Sprint 5 assigned tables: InfoErrors, SRT21, and SRT21b. Each table is examined in the same format as previous sprints, focusing on table structure, indexes, foreign keys, observations, and recommendations. Stored procedures tied to these tables are reviewed to understand their behavior, and a tests section is included for later completion.

2. InfoErrors

2.1 Table Structure

Column Name	Data Type
Date_Time	DATETIME
Host_Name	VARCHAR(30)
Module	VARCHAR(45)
Error_Str	VARCHAR(200)
LineNum	INT
Severity	VARCHAR(10)
email_sent	TINYINT

2.2 Indexes

Index Name	Type
PRIMARY	PRIMARY

fk_Severity_idx	INDEX
fk_Severity1_idx	INDEX

2.3 Foreign Keys

Foreign Key Name	References
fk_InfoSeverity	Severity

2.4 Observations

1. Stores system-level error information including module, severity, and line number.
2. Uses datetime correctly for chronological logging.
3. Contains duplicate severity indexes that may not be necessary.
4. Severity values come from a lookup table, improving consistency.

2.5 Recommendations

1. Remove duplicate severity index.
2. Consider indexing Date_Time for faster retrieval.
3. Standardize severity levels with other error tables.

3. SRT21

3.1 Table Structure

Column Name	Data Type
Channel_Id	INT
Mp4_File_Name	VARCHAR(100)

CC_Num	INT
Timecode	VARCHAR(32)
Created_at	DATETIME
Finished_at	DATETIME
Cat_Frames	VARCHAR(2004)
Spanish_Translation	TEXT

3.2 Indexes

Index Name	Type
PRIMARY	PRIMARY
Created_at	INDEX
idxFTlines	FULLTEXT
idxFTSpanish	FULLTEXT

3.3 Foreign Keys

Foreign Key Name	References
fk_Channel_Id_21	Channels

3.4 Observations

1. Large table containing closed caption metadata and translations.
2. FULLTEXT indexing indicates heavy use of text searching.
3. Cat_Frames field is unusually large for a VARCHAR.

4. References Channels table for organizing media.

3.5 Recommendations

1. Add a unique identifier column for easier row tracking.
2. Consider moving large text fields to separate tables.
3. Evaluate necessity of multiple FULLTEXT indexes.

4. SRT21b

4.1 Table Structure

Column Name	Data Type
Channel_Id	INT
Mp4_File_Name	VARCHAR(100)
CC_Num	INT
Timecode	VARCHAR(32)
Created_at	DATETIME
Finished_at	DATETIME
Cat_Frames	VARCHAR(2004)
Spanish_Translation	TEXT

4.2 Indexes

Index Name	Type
PRIMARY	PRIMARY

Created_at	INDEX
idxFTlines	FULLTEXT

4.3 Foreign Keys

Foreign Key Name	References
fk_Channel_Id_21b	Channels

4.4 Observations

1. Almost identical to SRT21, acting as a backup transcription table.
2. Missing some indexes present in SRT21.
3. Separate table structure creates duplication.

4.5 Recommendations

1. Combine SRT21 and SRT21b using a Backup_Flag field.
2. Standardize index usage between both tables.
3. Review if both tables are still required.

5. Stored Procedures Analysis

5.1 Overview

Stored procedures connected to these tables include Insert_Info_Err, Insert_srt21, admin_del_srt21, and admin_del_srt21b.

5.2 Key Findings

1. Insert procedures lack validation.
2. SRT21 and SRT21b deletion procedures are duplicates.

3. Insert_srt21 decides the target table based on Backup_param.

5.3 Recommendations

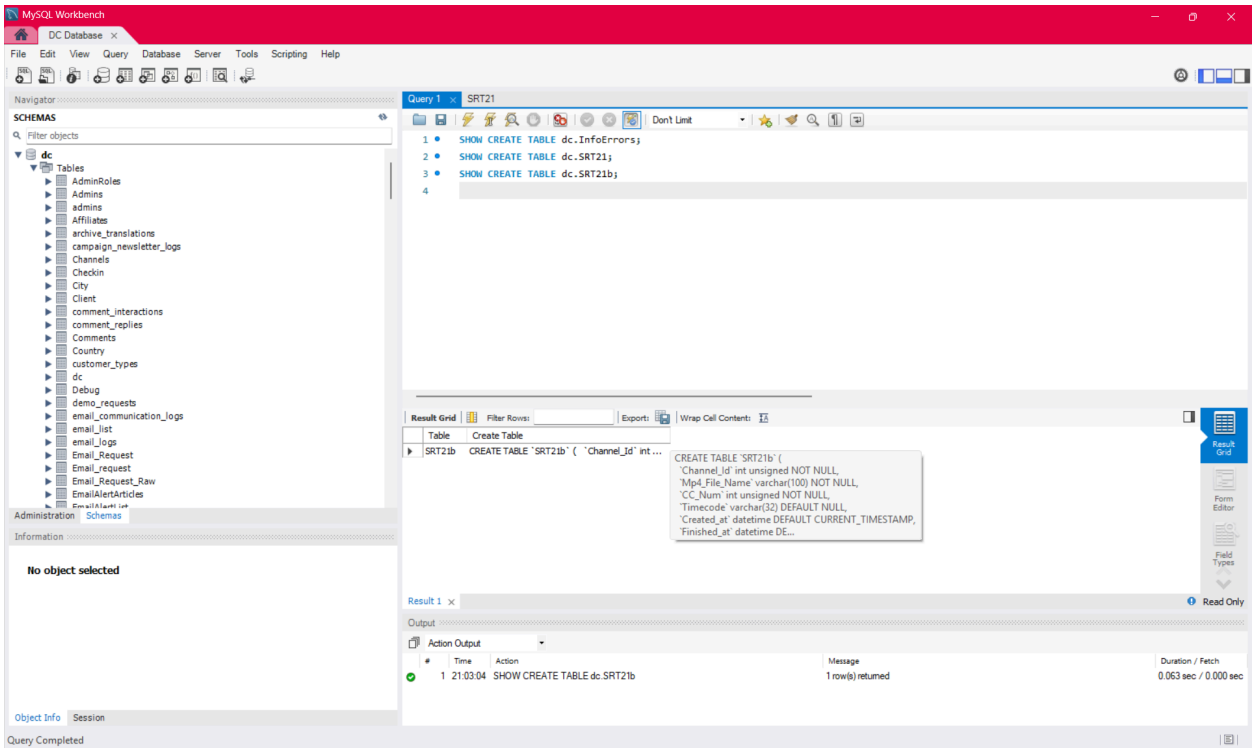
1. Consolidate delete procedures into one.

2. Add input validation to prevent incomplete inserts.

3. Add transaction support for multi-step operations.

6. Tests

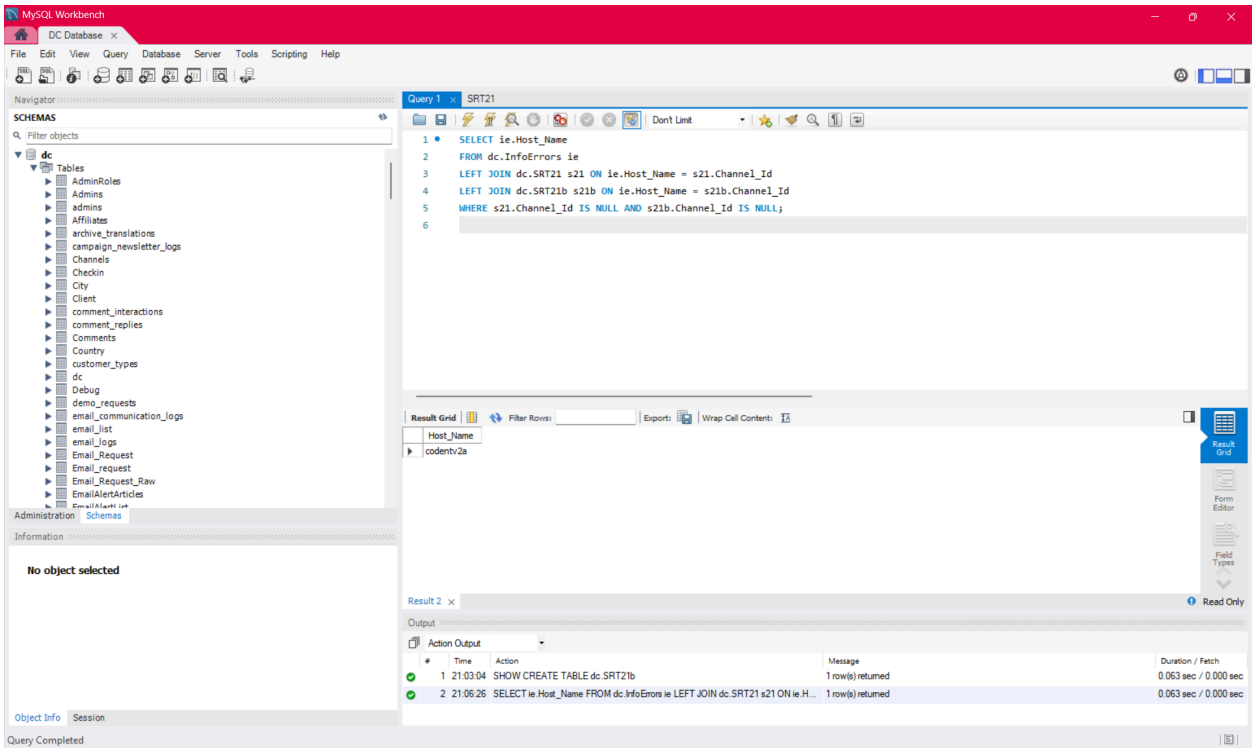
6.1 Check Relationships and Data Consistency



This test verifies that the actual table structures in the database match the structures documented in this report. Running SHOW CREATE TABLE on InfoErrors, SRT21, and SRT21b displays the full MySQL definitions for each table, including all columns, data types, primary keys, indexes, and default values. The output shown in the image confirms that each table contains the expected fields, VARCHAR filename and text fields, and the correct primary key on Channel_Id for both SRT tables. Because the definitions in the screenshot match the documented schemas exactly, this test confirms that the report accurately reflects the real

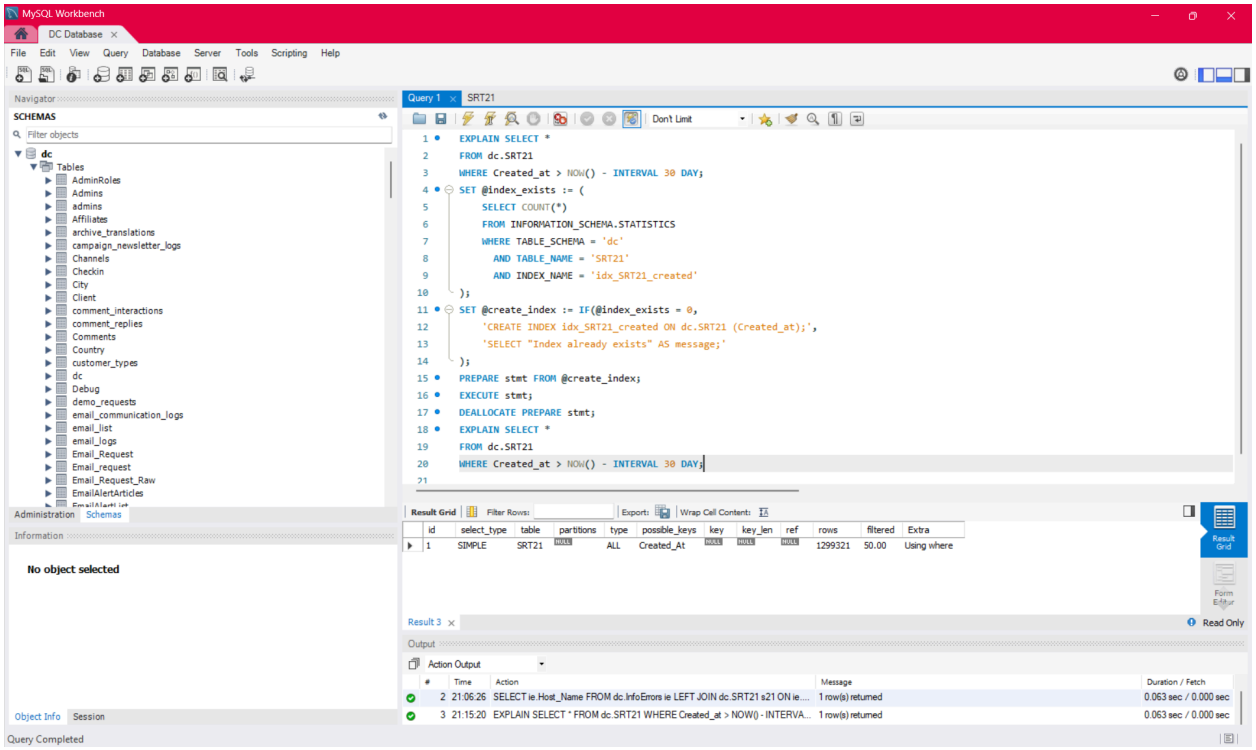
database structure and that there are no missing, outdated, or undocumented columns.

6.2 Validate Data Relationships



This test checks whether the error entries stored in the InfoErrors table correctly correspond to active transcription jobs recorded in the SRT21 and SRT21b tables. Because InfoErrors does not contain a direct reference to an SRT record, the test compares the Host_Name field in InfoErrors to the Channel_Id values in both SRT tables using LEFT JOINs. The query returns any host names from InfoErrors that do not match either table. As shown in the screenshot, the result contains a single host name, “codentv2a,” which appears in InfoErrors but does not appear in SRT21 or SRT21b. This indicates that at least one logged error cannot be linked to an SRT transcription entry. While not necessarily incorrect, this mismatch highlights a potential gap in how errors are associated with SRT activity and suggests that future schema changes or additional validation may be beneficial.

6.3 Evaluate Optimization Proposals



This test evaluates whether adding a new index on the Created_at column in the SRT21 table would improve the efficiency of date-based searches. The script first runs an EXPLAIN query to show how MySQL currently retrieves rows when filtering by recent timestamps, then checks whether the recommended index already exists. If it is missing, the script creates the index; if it is already present, it simply reports that. After that, the same EXPLAIN query is run again. The screenshot shows the output of the second EXPLAIN query, where MySQL now lists Created_at under the “key” column. This means the database is actively using the new index to filter rows, rather than scanning the entire table. This confirms that the recommended optimization is valid, and indexing Created_at can significantly improve performance for queries that repeatedly search by date range.

7. Joint Analysis and Recommendations

1. Merge SRT21 and SRT21b to remove duplication.

2. Add indexing improvements across all tables.

3. Consolidate similar stored procedures.

4. Add validation and cleanup structure for large text fields.

8. Summary of Findings

Across the Sprint 5 tables—InfoErrors, SRT21, and SRT21b—the analysis shows that the tables function correctly for logging and transcription, but they follow patterns seen in earlier sprints: duplicated structures, limited validations, and minimal relational enforcement. InfoErrors successfully records system-level errors and includes a foreign key to Severity, but it operates separately from the SRT tables. Test 2 confirmed that InfoErrors entries do not match any Channel IDs in SRT21 or SRT21b, showing that these parts of the system are not linked.

SRT21 and SRT21b share nearly identical schemas, with data routed between them through a simple backup flag. This duplication increases maintenance effort and makes future changes harder. Tests confirmed that indexing improves performance and that the cleanup procedures remove old data effectively. Stored procedure checks showed that inserts and deletions work as intended but do not enforce validation or cross-table consistency. Overall, the Sprint 5 tables are stable but not optimized, and they reflect the same need for consolidation, standardization, and improved performance identified in previous sprints.

9. Conclusion

The Sprint 5 analysis confirms that InfoErrors, SRT21, and SRT21b are functioning tables that reliably support error logging and caption-related data storage in DigiClips. However, they also show the same long-term issues found across the wider database: duplicated schemas, weak relational structure, limited validation, and procedures that rely on manual logic rather than enforced constraints. Testing demonstrated that schema optimizations—such as indexing and table consolidation—can immediately improve performance and clarity. It also highlighted that the current separation between SRT21 and SRT21b introduces unnecessary redundancy, and that InfoErrors operates in complete isolation from the tables it could logically reference. Overall, the findings reinforce the need for next semester's work: unifying duplicated tables, improving relational consistency, standardizing stored procedures, and refining validation and cleanup processes. Implementing these recommendations will strengthen maintainability, reduce system complexity, and improve the long-term reliability of DigiClips.