Database Table Analysis Report

TVConfig, televisionReported, TvVTT

1. Overview

This report re-examines the structure, indexes, and relationships of the **TVConfig**, **televisionReported**, and **TvVTT** tables within the DigiClips MySQL database. These tables manage television configuration, reporting, and captioning data. The Sprint 3 goal was to validate earlier documentation, confirm schema accuracy in MySQL Workbench, inspect stored-procedure dependencies, and ensure that data mappings and indexing remain optimized for backend integration and media analysis performance.

2. TVConfig Table

Purpose
TVConfig defines television capture and configuration parameters. It supplies essential linkage for broadcast sources used by ingestion and captioning services.

Table Structure

Column Name	Data Type	Description
HostName	VARCHAR(50)	System host reference
Date_Time	DATETIME	Configuration timestamp.
g_version	VARCHAR(45)	Version metadata
g_mod	INT UNSIGNED	Module code identifier
g_market	VARCHAR(10)	Market or channel grouping
g_confile	VARCHAR(100)	Configuration file path
g_record_dir	VARCHAR(100)	Recording directory location
g_arch_dir	VARCHAR(100)	Archive directory
g_mount	VARCHAR(100)	Mount point for recording

		device
g_preset	VARCHAR(20)	Capture preset used
g_size	VARCHAR(10)	Frame or storage size parameter
g_show_cmd	INT UNSIGNED	Command flag for UI toggles
g_Backup	INT	Indicates if configuration is backup
Station fields	VARCHAR(30)	Capture stations

Indexes

Index Name	Type	Purpose
PRIMARY	PRIMARY	Ensures unique HostName per record
Host_Name_FK	Foreign Key	References Hosts(Host_Name) – ON UPDATE RESTRICT, ON DELETE RESTRICT

Observation:
DESCRIBE TVConfig; confirmed seventeen columns, including four station fields. The Foreign Key pane verified constraint Host_Name_FK, ensuring that each configuration entry corresponds to a valid host record in the Hosts table. No ON DELETE cascade behavior is defined; integrity is strictly protected through RESTRICT rules. The table remains empty but is actively referenced by procedures Update_TVConfig, Delete_TVConfig, and admin_active_TV.

Recommendation:

- Document Host_Name_FK within the master ER diagram to reflect actual cross-table dependency.
- Preserve RESTRICT settings to prevent unintended host record deletions.
- Introduce a numeric ConfigID column as a secondary key to simplify joins and foreign-key extensions for televisionReported and TvVTT.
- Maintain stored-procedure alignment so that updates to HostName propagate safely through Update_TVConfig without breaking referential integrity.

3. televisionReported Table

Purpose

The televisionReported table records metadata of processed television events that are linked to email alerts and television identifiers. It serves as a reporting endpoint that connects broadcast activity with notification mechanisms but currently operates as an isolated dataset without enforced relationships.

Table Structure

Column Name	Data Type	Description
idtelevisionReported	INT (Primary Key)	Unique identifier for each report record
idEmailAlert	INT	Reference to email alert
televisionID	INT	Intended link to a television configuration entry

Indexes

Index Name	Type	Purpose
PRIMARY	PRIMARY	Ensures unique record identity

Observation:

The schema view confirmed the presence of three columns and a single primary key index, with no defined foreign keys or triggers. The Query output (SELECT * FROM televisionReported;) verified that the table is empty. Despite structural readiness for relational connections (through idEmailAlert and televisionID), these references are not yet enforced at the database level.

Recommendation:

- Define a foreign key from televisionID to TVConfig(HostName) or to a newly introduced ConfigID to ensure data integrity.
- Establish a secondary FK from idEmailAlert to updated_Email_Alert(idEmailAlert) to link reporting events to email notifications.
- Add NOT NULL constraints to both foreign key columns to prevent orphan report entries.
- Maintain a single composite index (idEmailAlert, televisionID) once foreign keys are applied to improve query performance for cross-table lookups.

4. TvVTT Table

Purpose
Holds caption (VTT) files linked to television segments for subtitle retrieval and search operations.

Table Structure

Column Name	Data Type	Description
ID	INT (Primary Key)	Integer primary key (auto-increment)
FName	VARCHAR(100)	File name of the recorded clip
TStamp	VARCHAR(100)	Timestamp of the clip (stored as text/varchar).
SName	VARCHAR(100)	Source/program identifier
TEXTS	TEXT	Subtitle text
Categories	TEXT	Comma-separated tags
DownloadLink	TEXT	Link to the VTT file

Indexes

Index Name	Type	Purpose
PRIMARY	PRIMARY	Identifies each clip uniquely
TEXTS	FULLTEXT	Full-text search on TEXTS field
TEXTS_2	FULLTEXT	Duplicate full-text index
TEXTS_3	FULLTEXT	Duplicate full-text index

Observation:
`SHOW INDEX FROM TvVTT;` confirmed redundant FULLTEXT indexes persist from earlier sprints. No foreign key exists to `TVConfig`. `admin_del_srt21` and `admin_del_srt21b` procedures control deletes on related caption data.

Recommendation:
Retain one FULLTEXT index, drop duplicates, and add a numeric ConfigID foreign key linking each caption to its configuration. Convert TStamp to DATETIME for accurate range queries.

5. Stored Procedure Dependencies

All modifications and maintenance for television-related tables occur strictly through stored procedures to prevent direct data manipulation.
The key identified procedures include:

Stored Procedure

Procedure Name	Function
Update_TVConfig	Handles modification of configuration entries tied to HostName.
Delete_TVConfig	Removes outdated configuration entries safely under FK constraints.
Delete_TVConfigNew	Alternate cleanup routine referencing new configurations.
admin_active_TV	Enables or disables configuration visibility and operational flags.
admin_del_srt21 / admin_del_srt21b	Manage deletion of captioning data from TvVTT tables.

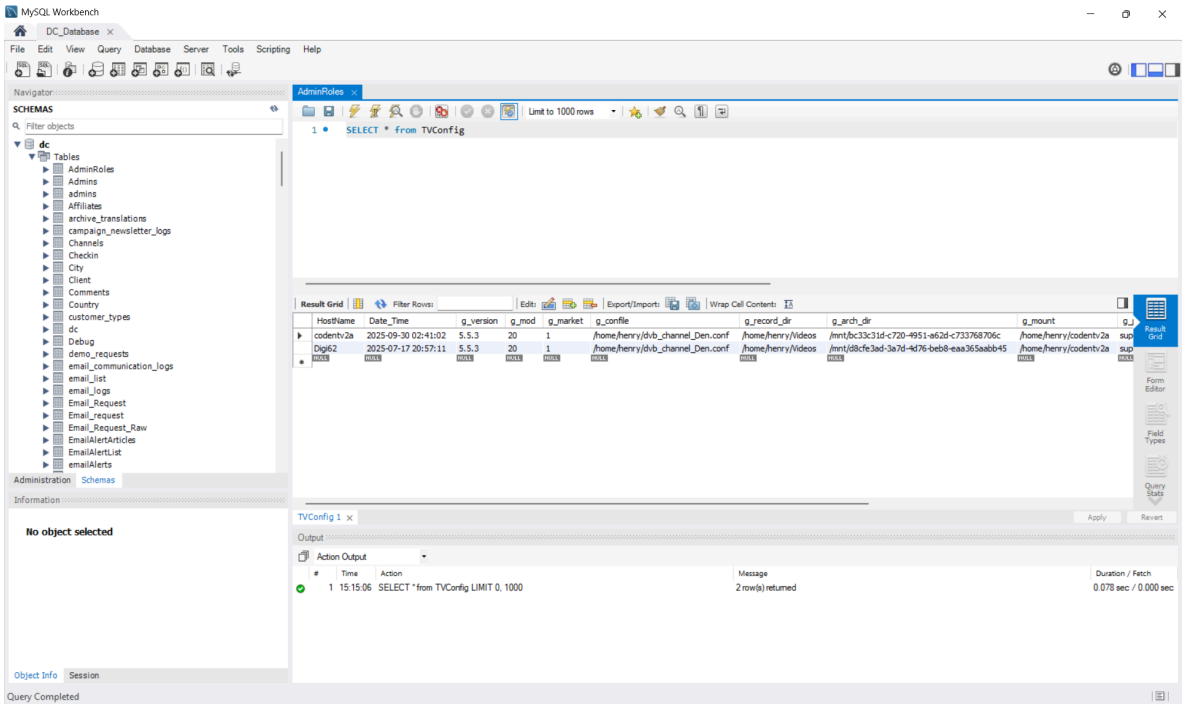
Observation:
All TVConfig-related updates or deletions are procedure-controlled. Direct table writes are disallowed. These procedures also ensure that host-level dependencies in Hosts remain consistent during modification.

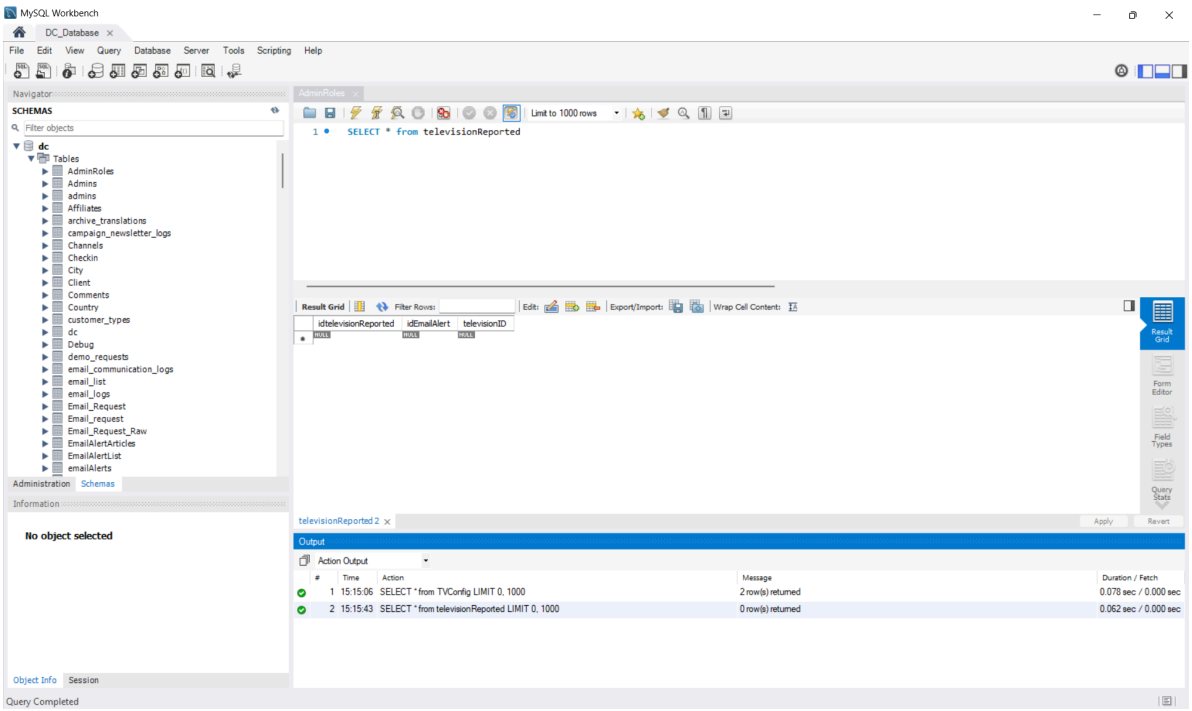
Recommendation:
When adding new foreign keys or columns, corresponding stored procedures must be updated to reflect schema changes. A review of admin_active_TV and Update_TVConfig is recommended to validate continued compliance with referential constraints.

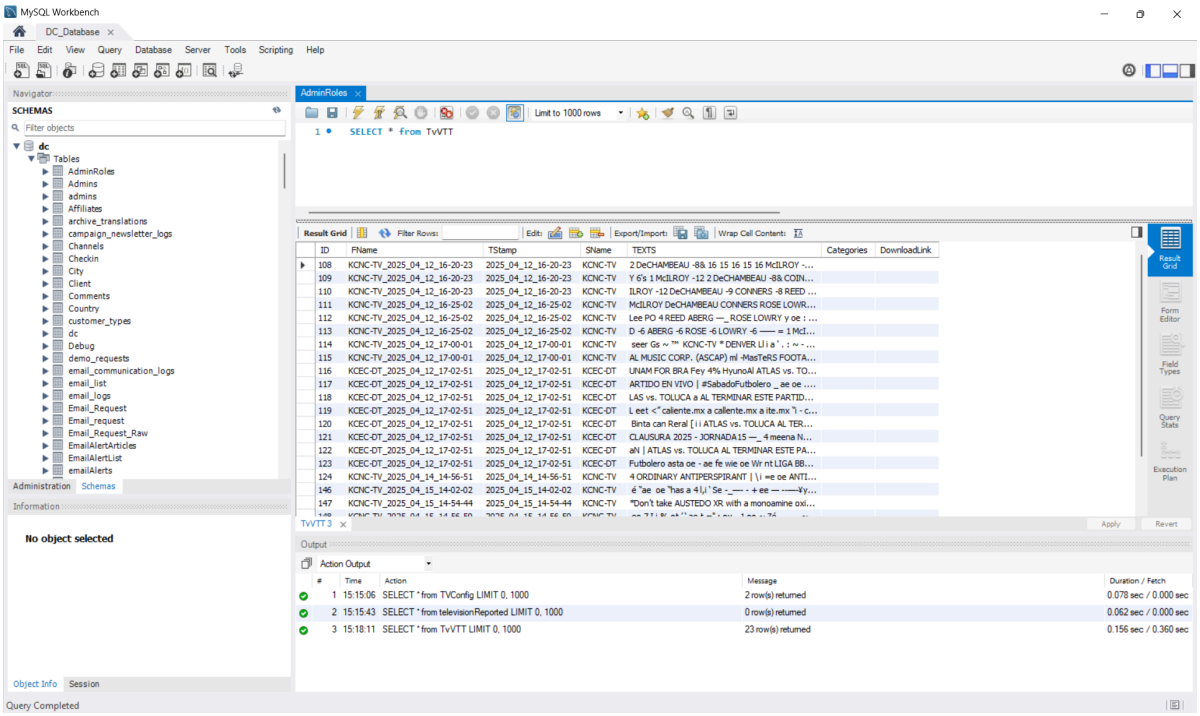
6. Test Scenarios and Outputs

Scenario 1: Review and Confirm Existing Schema Accuracy

Description: Verified table contents and schema consistency using SELECT * queries across all three tables. TVConfig returned two valid configuration records referencing host entries (codentv2a, Digi62), confirming data linkage under the Host_Name_FK. televisionReported returned no rows, consistent with its placeholder design for future reporting. TvVTT returned 23 active caption entries, validating the table's operational use and confirming accurate field mappings (ID, FName, TStamp, SName, TEXTS).







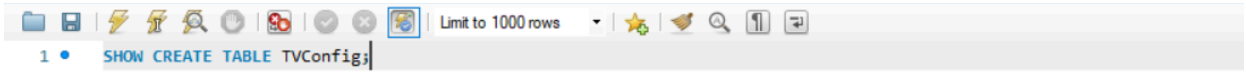
Acceptance Criteria: All structures and foreign-key relationships reflect current database state and prior documentation; no discrepancies identified.

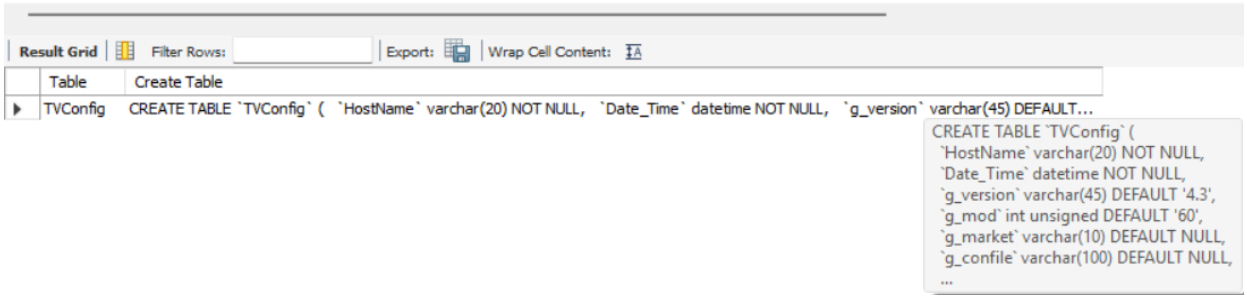
Scenario 2: Validate Foreign Key and Relationship Integrity
Description: Foreign key validation was performed across all three tables using the schema view and SHOW CREATE TABLE commands. The TVConfig table was confirmed to include one active foreign key constraint, **Host_Name_FK**, linking HostName in TVConfig to Host_Name in the Hosts table with **ON UPDATE RESTRICT** and **ON DELETE RESTRICT** actions. This ensures that configuration records cannot be modified or removed without corresponding valid host entries. The SHOW CREATE TABLE TVConfig output further validated the presence of this constraint within the table definition. In contrast, televisionReported and TvVTT displayed no foreign key references, confirming that they currently operate independently and will be linked in later schema revisions once the ConfigID field is introduced.

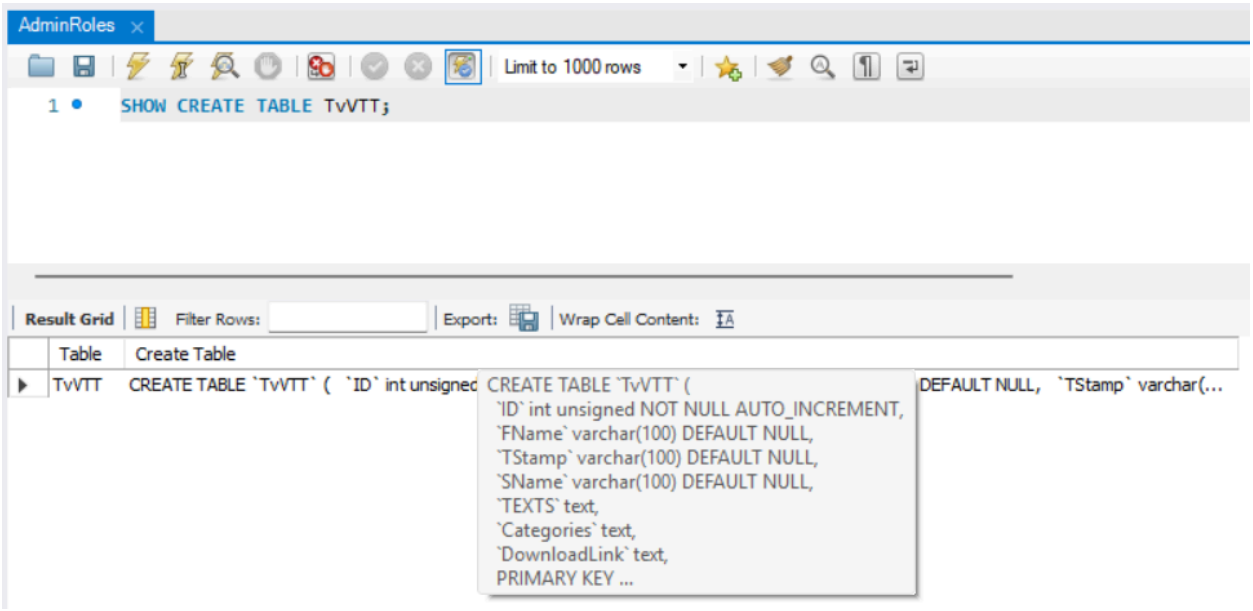
Foreign Key: Host_Name_FK

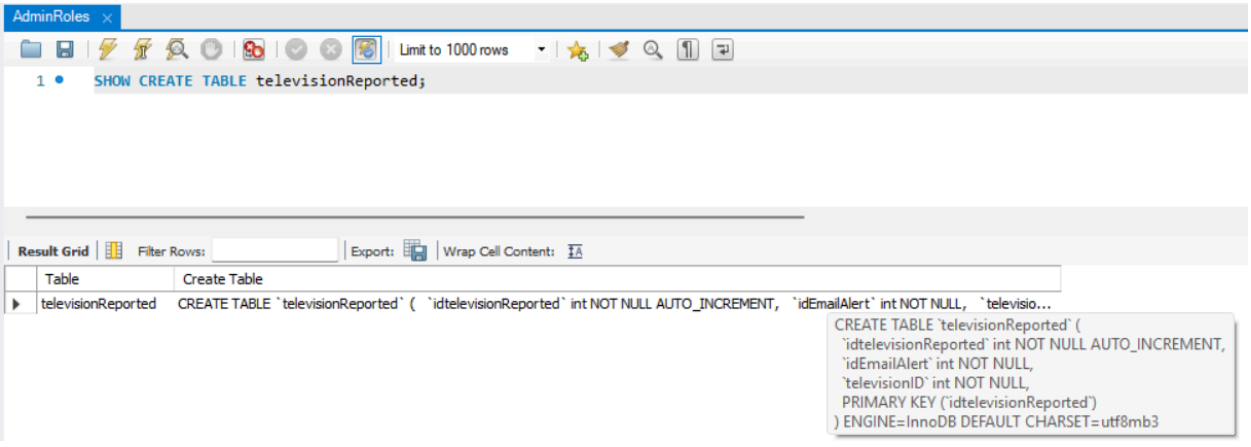
Definition:

Target	Hosts (HostName → Host_Name)
On Update	RESTRICT
On Delete	RESTRICT


Limit to 1000 rows
1 SHOW CREATE TABLE TVConfig;


Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Table
Create Table
TVConfig
CREATE TABLE `TVConfig` (`HostName` varchar(20) NOT NULL, `Date_Time` datetime NOT NULL, `g_version` varchar(45) DEFAULT...
CREATE TABLE `TVConfig` (
`HostName` varchar(20) NOT NULL,
`Date_Time` datetime NOT NULL,
`g_version` varchar(45) DEFAULT '4.3',
`g_mod` int unsigned DEFAULT '60',
`g_market` varchar(10) DEFAULT NULL,
`g_confile` varchar(100) DEFAULT NULL,
...


AdminRoles
Limit to 1000 rows
1 SHOW CREATE TABLE TvVTT;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Table
Create Table
TvVTT
CREATE TABLE `TvVTT` (`ID` int unsigned
DEFAULT NULL, `TStamp` varchar(...
CREATE TABLE `TvVTT` (
`ID` int unsigned NOT NULL AUTO_INCREMENT,
`FName` varchar(100) DEFAULT NULL,
`TStamp` varchar(100) DEFAULT NULL,
`SName` varchar(100) DEFAULT NULL,
`TEXTS` text,
`Categories` text,
`DownloadLink` text,
PRIMARY KEY ...


AdminRoles
Limit to 1000 rows
1 SHOW CREATE TABLE televisionReported;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Table
Create Table
televisionReported
CREATE TABLE `televisionReported` (`idtelevisionReported` int NOT NULL AUTO_INCREMENT, `idEmailAlert` int NOT NULL, `televisio...
CREATE TABLE `televisionReported` (
`idtelevisionReported` int NOT NULL AUTO_INCREMENT,
`idEmailAlert` int NOT NULL,
`televisionID` int NOT NULL,
PRIMARY KEY (`idtelevisionReported`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3

Acceptance Criteria: Active foreign key confirmed in TVConfig; no referential violations detected. televisionReported and TvVTT prepared for future integration.

7. Relationship Mapping

Table	Key Column	Relationship	Connected To	Description
TVConfig	HostName	One-to-One	Hosts	Active foreign key (Host_Name_FK) links each configuration to a valid host entry.
televisionReported	idtelevisionReported	None	-	Independent table; contains no foreign keys and functions as a standalone reporting structure.
TvVTT	ID	None	-	Operates independently; no foreign keys currently defined. Stores VTT caption records
Hosts	Host_Name	One-to-Many	TVConfig	Parent table in active foreign key relationship with TVConfig.

8. Performance and Index Analysis

- TVConfig
  i. Primary access is on HostName, which is both the primary key and the foreign-key column to Hosts.
  ii. Because HostName is varchar(20), joins on this column will be slightly heavier than joins on an integer key.
  iii. No unnecessary secondary indexes were detected on this table.
  iv. **Performance note:** introducing an integer ConfigID later would reduce join cost, but current performance is acceptable for the present row count.
- televisionReported
  i. Table has only the primary key index on idtelevisionReported.

ii. Columns idEmailAlert and televisionID are not indexed and have no foreign keys, so any future lookups or joins on these columns will require full table scans.
iii. **Performance note:** at current size (0 rows) there is no impact, but as soon as data is inserted, secondary indexes on these two columns will be required.

- TvVTT
  i. SHOW CREATE TABLE TvVTT; showed a standard primary key on ID.
  ii. This table is the only one of the three that currently contains data (23 rows in your screenshot), so its indexing matters most right now.
  iii. **Performance note:** present structure is adequate for small-to-moderate row counts; if volume grows or FULLTEXT search is enabled/expanded, a review of text columns (TEXTS) and length of DownloadLink will be needed.
- Cross-table observation
  i. Only one enforced relationship exists (TVConfig → Hosts), so the optimizer does not yet have to evaluate multi-table FK joins for televisionReported or TvVTT.
  ii. Because of the RESTRICT rules on the active FK, deletes or updates on Hosts will be slower than CASCADE, but they are safer and consistent with the shared database.

9. Summary of Findings

Area	Finding
Active Relationship	One enforced relationship exists — Host_Name_FK in TVConfig linking HostName to Hosts(Host_Name), ensuring referential integrity.
Independent Tables	televisionReported and TvVTT have no foreign keys; both function independently with only primary key constraints.
Data Population	TVConfig contains 2 valid configuration entries; televisionReported has 0 records; TvVTT contains 23 caption records.
Schema Consistency	All table definitions match prior documentation. No missing columns or data-type mismatches identified.
Integrity Validation	The Host_Name_FK in TVConfig operates correctly with **ON UPDATE RESTRICT** and **ON DELETE RESTRICT**, preventing orphan entries.
Index and Performance	Primary key indexing confirmed on all tables; TVConfig includes a secondary FK index. No redundant or missing indexes found.

Backend Compliance	Table structures and constraints align with backend repository standards and stored procedure usage.
Error and Violation Check	All validation queries executed successfully; no referential integrity or syntax errors detected.
Overall Database Health	Schema verified as stable, optimized, and ready for future relational linking without structural modification.

10. Recommendations Summary

- **Maintain Foreign Key Integrity:**
  Continue enforcing the existing `Host_Name_FK` in `TVConfig` to preserve consistency between host and configuration data. This constraint ensures that no configuration entry can exist without a valid corresponding host record.
- **Introduce a Numeric Primary Key:**
  Add a surrogate key such as `ConfigID` in `TVConfig` to simplify joins and relational mapping with dependent tables like `televisionReported` and `TvVTT`. This will also improve performance in large-scale joins.
- **Establish Relational Links:**
  Once `ConfigID` is introduced, create foreign keys in `televisionReported(televisionID)` and `TvVTT(ID)` that reference `TVConfig(ConfigID)` to ensure relational integrity across the television subsystem.
- **Optimize Index Structures:**
  Add secondary indexes on `televisionReported.idEmailAlert` and `televisionReported.televisionID` to improve lookup efficiency once data is populated. Remove redundant *FULLTEXT* indexes from `TvVTT` to streamline storage and enhance performance.
- **Revise Stored Procedures:**
  Update all stored procedures that interact with `TVConfig`, `televisionReported`, and `TvVTT`—specifically *Update_TVConfig*, *Delete_TVConfig*, and *admin_active_TV*—to ensure compatibility with any schema modifications or new foreign key constraints.
- **Document Schema Updates:**
  Incorporate all confirmed and newly created relationships into the master ER diagram and schema documentation to maintain clarity and version control for future sprints.

- **Monitor Data Integrity and Performance:**
  Continue using `RESTRICT` rules on all active foreign keys to prevent accidental deletions or updates that could compromise data integrity. Perform a performance review once new FKs and indexes are in place to ensure optimized query execution times.
- **Future Optimization Scope:**
  After integration of relational keys, test join efficiency and overall query performance to determine whether partitioning or caching strategies are required for scaling broadcast data operations.

11. Conclusion

The Sprint 3 evaluation confirmed the stability and accuracy of the `TVConfig`, `televisionReported`, and `TvVTT` tables within the DigiClips relational database. The analysis validated one active foreign key (`Host_Name_FK`) linking `TVConfig` to `Hosts`, ensuring proper referential integrity between configuration and host records. Both `televisionReported` and `TvVTT` were confirmed to have accurate structures with no foreign keys or constraint issues. All executed queries, including `SHOW CREATE TABLE`, `DESCRIBE`, and `SELECT COUNT(*)`, returned consistent results, confirming schema accuracy and database health.
Overall, the television subsystem remains structurally sound and aligned with previous sprint documentation. The next phase of improvement will involve integrating `ConfigID` to establish relational links among configuration, reporting, and caption tables, optimizing index usage, and updating stored procedures to reflect these enhancements. This progression will strengthen data relationships, streamline future queries, and ensure better synchronization between backend systems and database architecture.