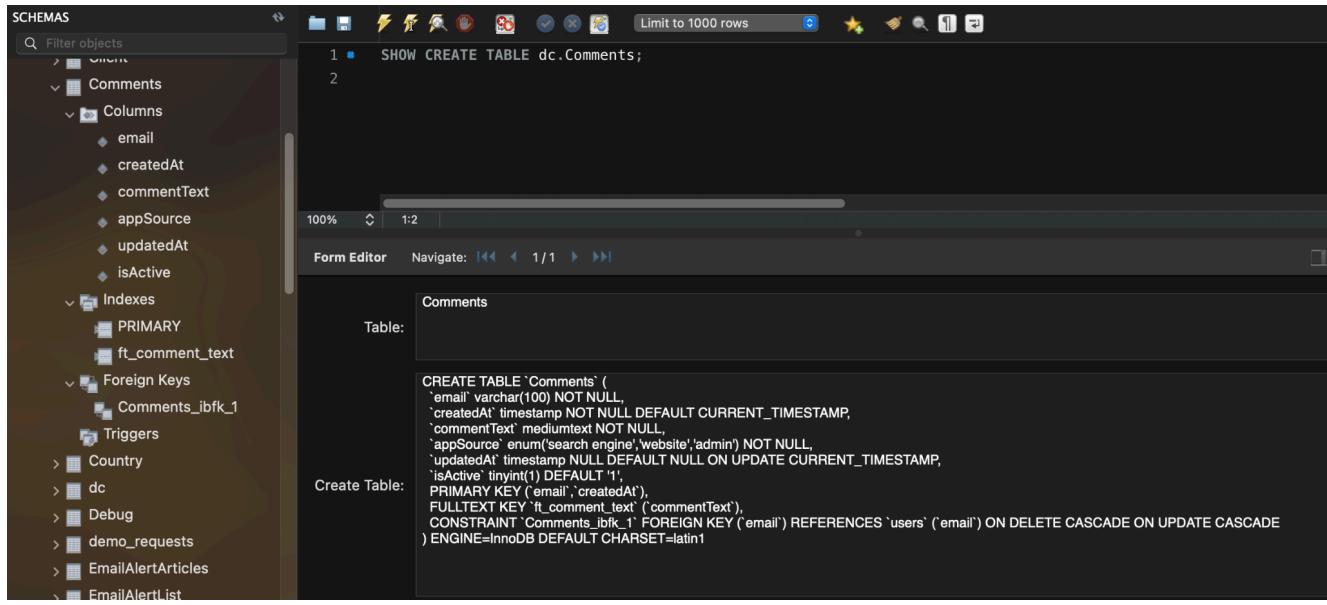


## Overview

The **Comments** table stores feedback, text entries, and notes made by users or system processes. It keeps details such as who made the comment (email), what the comment says, where it came from (app source), and when it was created or updated. It also tracks whether each comment is active.

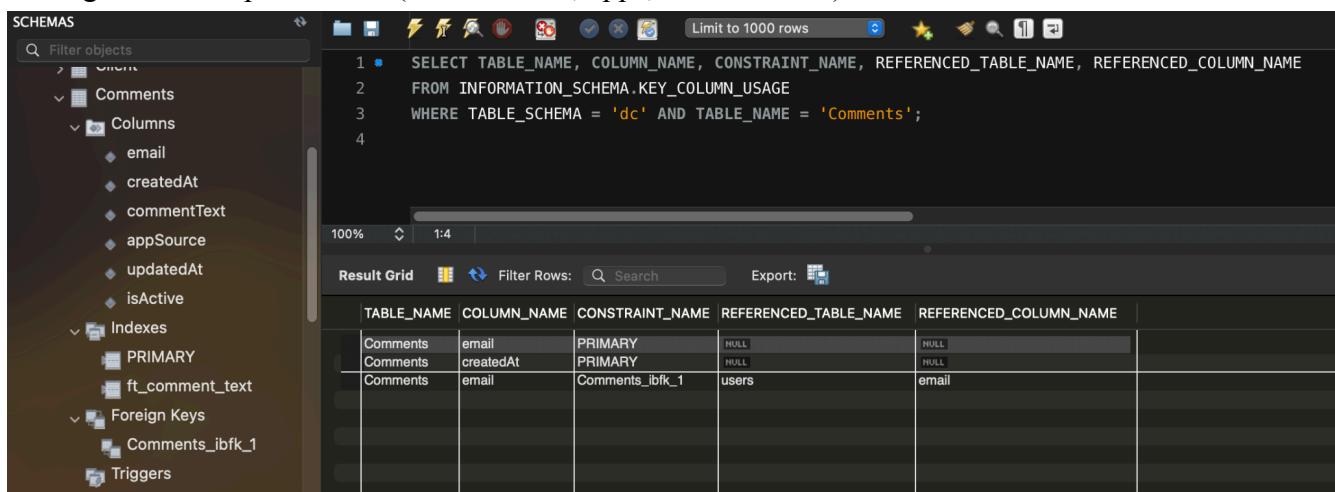


A screenshot of MySQL Workbench showing the creation of the **Comments** table. The left pane shows the database schema with the **Comments** table selected. The right pane displays the SQL code for creating the table:

```
CREATE TABLE `Comments` (
  `email` varchar(100) NOT NULL,
  `createdAt` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `commentText` mediumtext NOT NULL,
  `appSource` enum('search engine','website','admin') NOT NULL,
  `updatedAt` timestamp NULL DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
  `isActive` tinyint(1) DEFAULT '1',
  PRIMARY KEY (`email`),
  FULLTEXT KEY `ft_comment_text` (`commentText`),
  CONSTRAINT `Comments_ibfk_1` FOREIGN KEY (`email`) REFERENCES `users` (`email`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

## Purpose

The purpose of the **Comments** table is to record and manage user or system-generated comments for different applications and platforms. It supports data tracking for reviews, feedback, or alerts coming from multiple sources (like websites, apps, or broadcasts).



A screenshot of MySQL Workbench showing the usage of the **Comments** table across other tables. The left pane shows the database schema with the **Comments** table selected. The right pane displays the SQL code and results of a query to find key column usages:

```
1 • SELECT TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME, REFERENCED_TABLE_NAME, REFERENCED_COLUMN_NAME
2 FROM INFORMATION_SCHEMA.KEY_COLUMN_USAGE
3 WHERE TABLE_SCHEMA = 'dc' AND TABLE_NAME = 'Comments';
4
```

The results show the following key column usages:

TABLE_NAME	COLUMN_NAME	CONSTRAINT_NAME	REFERENCED_TABLE_NAME	REFERENCED_COLUMN_NAME
Comments	email	PRIMARY	HULL	HULL
Comments	createdAt	PRIMARY	HULL	HULL
Comments	email	Comments_ibfk_1	users	email

## Table Columns

Column Name	Data Type	Description
email	VARCHAR(100)	The email address of the user who made the comment.
createdAt	TIMESTAMP	The date and time when the comment was first created
commentText	TEXT	The actual content of the comment
appSource	ENUM	The platform where the comment was made (for example, Web, Mobile, or API).
updatedAt	TIMESTAMP	The date and time when the comment was last updated.
isActive	TINYINT	A flag showing if the comment is active (1 = active, 0 = inactive)

```

SCHEMAS
Filter objects
> [ ] Schema
  > [ ] Comments
    > [ ] Columns
      > [ ] email
      > [ ] createdAt
      > [ ] commentText
      > [ ] appSource
      > [ ] updatedAt
      > [ ] isActive
    > [ ] Indexes
      > [ ] PRIMARY
      > [ ] ft_comment_text
    > [ ] Foreign Keys
      > [ ] Comments_ibfk_1
    > [ ] Triggers

1 • DESCRIBE dc.Comments;
2 |
```

Result Grid

Field	Type	Null	Key	Default	Extra
email	varchar(100)	NO	PRI	NULL	
createdAt	timestamp	NO	PRI	CURRENT_TIMESTAMP	DEFAULT_GENERATED
commentText	mediumtext	NO	MUL	NULL	
appSource	enum('search engine','website','admin')	NO		NULL	
updatedAt	timestamp	YES		NULL	on update CURRENT_TIMESTAMP
isActive	tinyint(1)	YES		1	

## Indexes and Keys

- Primary Key: email, createdAt
- Foreign Keys:
  - Fk\_comment\_text: full text index to speed up searches through commentText

The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' tree view is expanded to show the 'Comments' table under the 'dc' schema. The 'Indexes' node is selected, revealing two primary keys ('PRIMARY') and one full-text index ('ft\_comment\_text'). The main pane displays the output of the SQL command 'SHOW INDEXES FROM dc.Comments;'. The result grid shows three rows corresponding to the table's structure:

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Last_update
Comments	0	PRIMARY	1	email	A	1			HULL	BTREE			YES	
Comments	0	PRIMARY	2	createdAt	A	2			HULL	BTREE			YES	
Comments	1	ft_comment_text	1	commentText	HULL	2			HULL	FULLTEXT			YES	

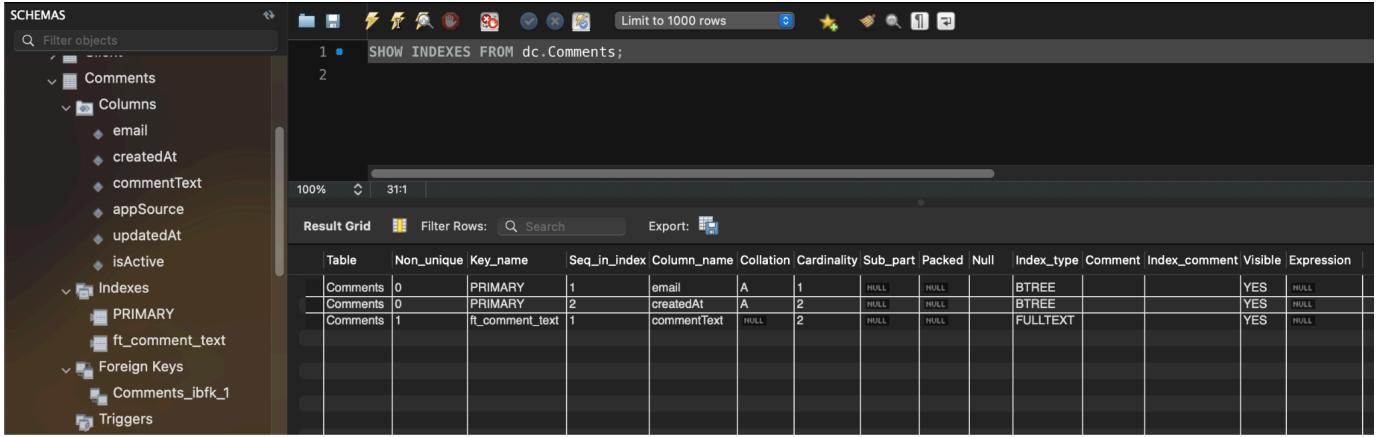
## Observations and Issues

1. **Table Structure is Clean:** The table contains all necessary fields for tracking and managing comments. There are no missing required columns, and data types are appropriate for the stored information.

The screenshot shows the MySQL Workbench interface. The 'Schemas' tree view is expanded to show the 'Comments' table under the 'dc' schema. The 'Indexes' node is selected, revealing two primary keys ('PRIMARY') and one full-text index ('ft\_comment\_text'). The main pane displays the output of the SQL command 'DESCRIBE dc.Comments;'. The result grid shows the table's structure with detailed column definitions:

Field	Type	Null	Key	Default	Extra
email	varchar(100)	NO	PRI	NULL	
createdAt	timestamp	NO	PRI	CURRENT_TIMESTAMP	DEFAULT_GENERATED
commentText	mediumtext	NO	MUL	NULL	
appSource	enum('search engine','website','admin')	NO		NULL	
updatedAt	timestamp	YES		NULL	on update CURRENT_TIMESTAMP
isActive	tinyint(1)	YES		1	

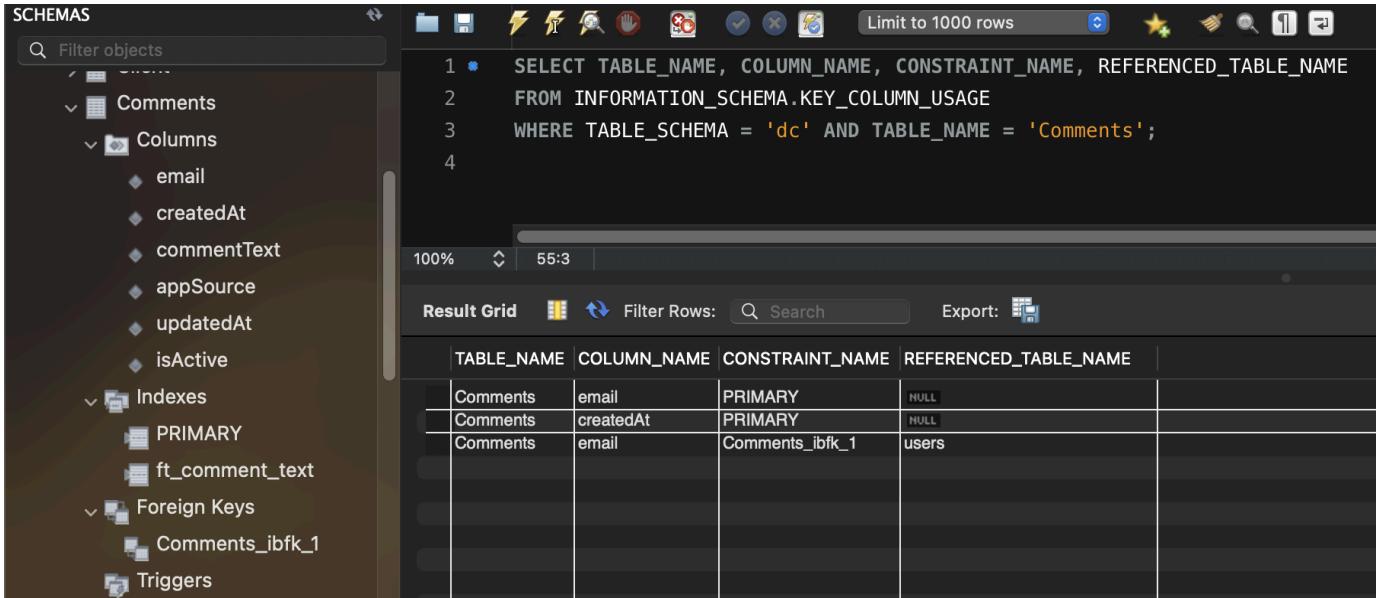
2. **Full Text Search Enabled:** The **ft\_comment\_text** index allows searching within the **commentText** field efficiently. This improves lookup speed for keyword or phrase searches.



```
1 • SHOW INDEXES FROM dc.Comments;
2
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
Comments	0	PRIMARY	1	email	A	1				BTREE			YES	HULL
Comments	0	PRIMARY	2	createdAt	A	2				BTREE			YES	HULL
Comments	1	ft_comment_text	1	commentText	NUL	2				FULLTEXT			YES	HULL

3. **Foreign Key Relationship Present:** The **Comments** table includes one foreign key constraint named **Comments\_ibfk\_1**. This links the **email** column in **dc.Comments** to the **email** column (or corresponding key) in the **users** table. This ensures that every comment is tied to a valid user record, improving data consistency and traceability.



```
1 • SELECT TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME, REFERENCED_TABLE_NAME
2   FROM INFORMATION_SCHEMA.KEY_COLUMN_USAGE
3  WHERE TABLE_SCHEMA = 'dc' AND TABLE_NAME = 'Comments';
4
```

TABLE_NAME	COLUMN_NAME	CONSTRAINT_NAME	REFERENCED_TABLE_NAME
Comments	email	PRIMARY	
Comments	createdAt	PRIMARY	
Comments	email	Comments_ibfk_1	users

## **Recommendations:**

1. **Validate Email References:** Since `email` is now a foreign key linked to the `users` table (`Comments_ibfk_1`), ensure all inserted emails exist in `users`.
2. **Determine how Timestamps are Set:** Look and see how the timestamps are currently set. Ensure that they automatically populate to ensure consistency in time tracking for creation and edit of comments.
3. **Regular Backup:** Since comments can grow large quickly, back up this table regularly to improve performance.
4. **Archive Old Comments:** Archive or delete rows where `inActive=0` and `updatedAt` is older than 2 years old. This allows for more efficient search operations