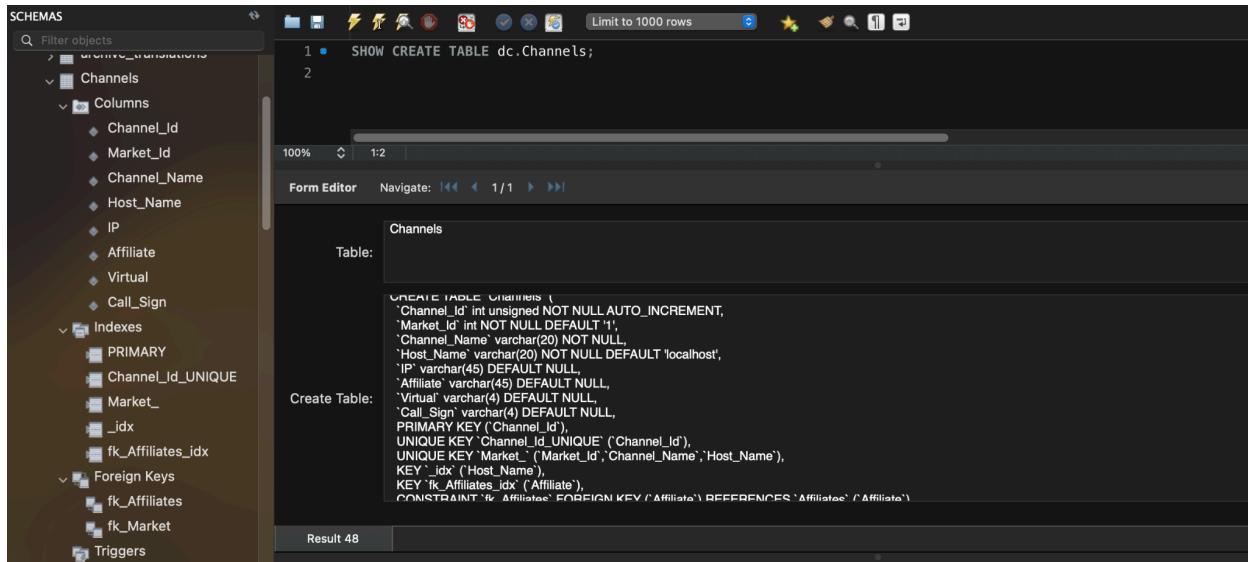


Overview

The **Channels** table stores details about each broadcast channel, including the channel's name, market location, affiliate network, and related technical information like IP address and call sign. It connects to two other tables, **Markets** and **Affiliates**, through foreign keys, which help link each channel to its market and affiliate.



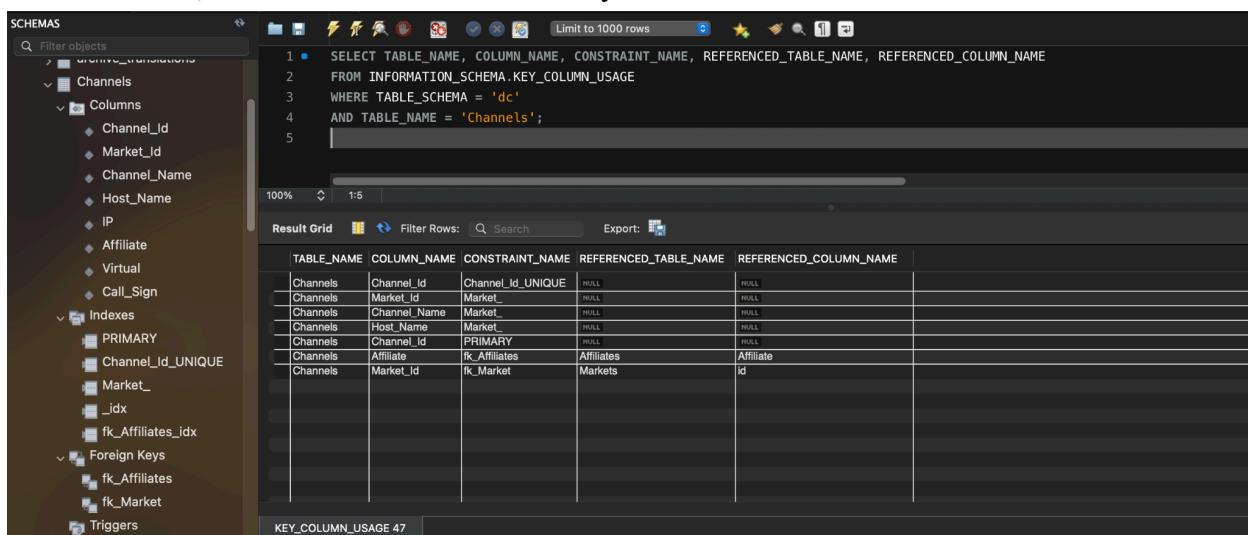
The screenshot shows the MySQL Workbench interface with the 'SCHEMAS' tree on the left. The 'Channels' table is selected under the 'Tables' node. The central pane displays the SQL code for creating the 'Channels' table:

```
1 • SHOW CREATE TABLE dc.Channels;
2

CREATE TABLE `Channels` (
  `Channel_Id` int unsigned NOT NULL AUTO_INCREMENT,
  `Market_Id` int NOT NULL DEFAULT '1',
  `Channel_Name` varchar(20) NOT NULL,
  `Host_Name` varchar(20) NOT NULL DEFAULT 'localhost',
  `IP` varchar(45) DEFAULT NULL,
  `Affiliate` varchar(45) DEFAULT NULL,
  `Virtual` varchar(4) DEFAULT NULL,
  `Call_Sign` varchar(4) DEFAULT NULL,
  PRIMARY KEY (`Channel_Id`),
  UNIQUE KEY `Channel_Id_UNIQUE` (`Channel_Id`),
  UNIQUE KEY `Market_` (`Market_Id`, `Channel_Name`, `Host_Name`),
  KEY `idx` (`Host_Name`),
  KEY `fk_Affiliates_idx` (`Affiliate`),
  CONSTRAINT `fk_Affiliates` FOREIGN KEY (`Affiliate`) REFERENCES `Affiliates` (`Affiliate`)
)
Result 48
```

Purpose

The purpose of the **Channels** table is to organize all information related to broadcast channels. Each record represents one channel, showing which market it belongs to, what affiliate it's connected with, and how it's identified in the system.



The screenshot shows the MySQL Workbench interface with the 'SCHEMAS' tree on the left. The 'Channels' table is selected under the 'Tables' node. The central pane displays the SQL code for selecting key column usage:

```
1 • SELECT TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME, REFERENCED_TABLE_NAME, REFERENCED_COLUMN_NAME
  FROM INFORMATION_SCHEMA.KEY_COLUMN_USAGE
  WHERE TABLE_SCHEMA = 'dc'
  AND TABLE_NAME = 'Channels';
5
```

The bottom pane shows the results of the query, listing the columns and their constraints:

TABLE_NAME	COLUMN_NAME	CONSTRAINT_NAME	REFERENCED_TABLE_NAME	REFERENCED_COLUMN_NAME
Channels	Channel_Id	Channel_Id_UNIQUE	NULL	NULL
Channels	Market_Id	Market_	NULL	NULL
Channels	Channel_Name	Market	NULL	NULL
Channels	Host_Name	Market	NULL	NULL
Channels	Channel_Id	PRIMARY	NULL	NULL
Channels	Affiliate	fk_Affiliates	Affiliates	Affiliate
Channels	Market_Id	fk_Market	Markets	id

KEY_COLUMN_USAGE 47

Table Columns

Column Name	Data Type	Description
Channel_Id	INT UNSIGNED AUTO_INCREMENT	The unique ID number for each channel.
Market_Id	INT NOT NULL DEFAULT 1	Links each channel to a record in the Markets table.
Channel_Name	VARCHAR(20) NOT NULL	The name of the channel (for example, KDVR or KCEC).
Host_Name	VARCHAR(20) NOT NULL DEFAULT 'localhost'	The network or server name for the channel.
IP	VARCHAR(45)	The IP address linked to the channel.
Affiliate	VARCHAR(45)	Connects the channel to its affiliate network (from the Affiliates table)
Virtual	VARCHAR(45)	The virtual number of the channel
Call_Sign	VARCHAR(45)	The call sign that identifies the station publicly

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree is visible, showing the 'dc' schema with a 'Channels' table selected. Under 'Channels', there are 'Columns', 'Indexes', and 'Foreign Keys'. The 'Columns' section lists the columns: Channel_Id, Market_Id, Channel_Name, Host_Name, IP, Affiliate, Virtual, and Call_Sign. On the right, the main pane displays the results of the DESCRIBE query for the 'Channels' table:

```

1 • DESCRIBE dc.Channels;
2
100% 22:1
Result Grid Filter Rows: Search Export:

```

Field	Type	Null	Key	Default	Extra
Channel_Id	int unsigned	NO	PRI	NULL	auto_increment
Market_Id	int	NO	MUL	1	
Channel_Name	varchar(20)	NO		NULL	
Host_Name	varchar(20)	NO	MUL	localhost	
IP	varchar(45)	YES		NULL	
Affiliate	varchar(45)	YES	MUL	NULL	
Virtual	varchar(4)	YES		NULL	
Call_Sign	varchar(4)	YES		NULL	

Result 49

Indexes and Keys

- Primary Key: Channel_Id
- Foreign Keys:
 - fk_Market → linked to Markets(id)
 - fk_Affiliates → likes to Affiliates(Affiliate)
- Other Indexes: Market_, _idx, fk_Affiliates_idx
 - Purpose: Faster Searching

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The left sidebar shows the database structure with the following objects:
 - Affiliates
 - archive_translations
 - Channels
 - Columns
 - Channel_Id
 - Market_Id
 - Channel_Name
 - Host_Name
 - IP
 - Affiliate
 - Virtual
 - Call_Sign
 - Indexes
 - PRIMARY
 - Channel_Id_UNIQUE
 - Market_
 - _idx
 - fk_Affiliates_idx
 - Foreign Keys
 - fk_Affiliates
 - fk_Market
 - Triggers- Query Editor:** The top pane displays the SQL command: `SHOW INDEXES FROM dc.Channels;`
- Result Grid:** The bottom pane shows the results of the query in a grid format. The columns are: Table, Non_unique, Key_name, Seq_in_Index, Column_name, Collation, Cardinality, Sub_part, Packed, Null, Index_type, Comment, Index_comment, Visible, Expression.
- Results:** The grid contains the following data:

Table	Non_unique	Key_name	Seq_in_Index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
Channels	0	PRIMARY	1	Channel_Id	A	25				BTREE			YES	NULL
Channels	0	Channel_Id_UNIQUE	1	Channel_Id	A	25				BTREE			YES	NULL
Channels	0	Market_	1	Market_Id	A	1				BTREE			YES	NULL
Channels	0	Market_	2	Channel_Name	A	8				BTREE			YES	NULL
Channels	0	Market_	3	Host_Name	A	25				BTREE			YES	NULL
Channels	1	_idx	1	Host_Name	A	8				BTREE			YES	NULL
Channels	1	fk_Affiliates_idx	1	Affiliate	A	1			YES	BTREE			YES	NULL

Observations and Issues

1. **Possible Missing Affiliate Links:** Some channels in dc.Channels have Affiliate names that do not exist in the dc.Affiliates table. These are orphaned records since the link has been broken.

The screenshot shows a database interface with a schema browser on the left and a query editor on the right. The schema browser lists tables like archive_translations, Channels, and Markets, along with their columns and indexes. The query editor contains the following SQL code:

```
1 • SELECT channels.Channel_Id, channels.Market_Id, channels.Affiliate
  FROM dc.Channels channels
  LEFT JOIN dc.Markets markets ON channels.Market_Id = markets.id
  LEFT JOIN dc.Affiliates affiliate ON channels.Affiliate = affiliate.Affiliate
 WHERE markets.id IS NULL OR affiliate.Affiliate IS NULL;
```

The execution plan on the right is a nested loop join. It starts with a Full Table Scan on the 'channels' table (cost 2.75, 25 rows). This feeds into a Unique Key Lookup for 'markets.id' (cost 27.5, 1 row). The result of this lookup then feeds into another Unique Key Lookup for 'affiliate.PRIMARY' (cost 27.5, 1 row). Finally, a nested loop join (cost 30.25) combines the results from the first two lookups. The total query cost is 57.75. The results are grouped under 'query_block #1'.

2. **Market Links are Valid:** All Market_Id values in Channels match records in the Markets table.

The screenshot shows a database interface with a schema browser on the left and a query editor on the right. The schema browser lists tables like archive_translations, Channels, and Markets. The query editor contains the following SQL code:

```
1 • SELECT * FROM dc.Markets WHERE id = 1;
```

The results grid shows a single row of data for Market ID 1:

id	Country	State	City
1	US	CO	Denver

3. **Duplicate Channel Names:** Some channel names appear more than once. These might represent subchannels or duplicate data.

The screenshot shows a database interface with a sidebar titled "SCHEMAS". The "Channels" schema is selected, displaying its columns: Channel_Id, Market_Id, Channel_Name, Host_Name, IP, Affiliate, Virtual, Call_Sign. Below the columns are Indexes (PRIMARY, Channel_Id_UNIQUE, Market__idx, fk_Affiliates_idx) and Foreign Keys (fk_Affiliates, fk_Market). Triggers are also listed. On the right, a query editor window displays a SQL SELECT statement to find duplicate channel names:

```

1 • SELECT Channel_Name, COUNT(*)
2 FROM dc.Channels
3 GROUP BY Channel_Name
4 HAVING COUNT(*) > 1;
5

```

The results grid shows the following data:

	Channel_Name	COUNT(*)
		2
	KCEC	3
	KCEC-DT	2
	KCNC-TV	4
	KDEN-DT	4
	KDVR-DT	4
	KMGH-TV	3
	KTVD-DT	3
	KUSA-HD	4
	KWGN-DT	4

Recommendations:

1. **Affiliate Fixes:** Check over to determine if there are missing affiliate links. If there are, add missing affiliate names to the dc.Affiliates table so all foreign key links are valid
2. **Check Duplicates:** Review repeated channel names to confirm if they are valid or if they are duplicates that should be removed
3. **Regular Checks:** Run these queries once a month to maintain clean and accurate data,