LOGARG – Database Documentation

MySQL Schema, Relationships, and Example Data

1. Overview

This database stores topics, sources, statements, arguments, and their logical links. The structure supports argument trees used in the LOGARG platform.

- **topics** – one record per discussion topic; links to a root argument.
- **sources** – bibliographic references for arguments and statements.
- **statements** – stores text for claims, justifications, and rebuttals.
- **arguments** – links a root claim to its source.
- **premises** – connects an argument with its supporting or counter statements.

2. Table Definitions

arguments

```
CREATE TABLE `arguments` (
  `id` int NOT NULL AUTO_INCREMENT,
  `claim` int NOT NULL,
  `source` varchar(255) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `claim` (`claim`),
  KEY `source` (`source`),
  CONSTRAINT `arguments_ibfk_1` FOREIGN KEY (`claim`) REFERENCES
     `statements` (`id`),
  CONSTRAINT `arguments_ibfk_2` FOREIGN KEY (`source`) REFERENCES
      `sources` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=
   utf8mb4_0900_ai_ci;
```

premises

```
CREATE TABLE `premises` (
  `argument` int NOT NULL,
  `premise` int NOT NULL,
  PRIMARY KEY (`argument`,`premise`),
  KEY `premise` (`premise`),
  CONSTRAINT `premises_ibfk_1` FOREIGN KEY (`argument`)
     REFERENCES `arguments` (`id`),
  CONSTRAINT `premises_ibfk_2` FOREIGN KEY (`premise`) REFERENCES
      `statements` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=
   utf8mb4_0900_ai_ci;
```

statements

```
CREATE TABLE `statements` (
  `id` int NOT NULL AUTO_INCREMENT,
  `text` varchar(1024) NOT NULL,
  `counter_statement` int DEFAULT NULL,
  `source` varchar(255) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `counterStatement` (`counter_statement`),
  KEY `source` (`source`),
  CONSTRAINT `statements_ibfk_1` FOREIGN KEY (`counter_statement
     `) REFERENCES `statements` (`id`),
  CONSTRAINT `statements_ibfk_2` FOREIGN KEY (`source`)
     REFERENCES `sources` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=
   utf8mb4_0900_ai_ci;
```

sources

```
CREATE TABLE `sources` (
  `name` varchar(255) NOT NULL,
  `text` varchar(1024) DEFAULT NULL,
  `url` varchar(1024) DEFAULT NULL,
  PRIMARY KEY (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=
   utf8mb4_0900_ai_ci;
```

topics

```
CREATE TABLE `topics` (
  `id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  `argument_id` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `argument_id` (`argument_id`),
  CONSTRAINT `topics_ibfk_1` FOREIGN KEY (`argument_id`)
     REFERENCES `arguments` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=
   utf8mb4_0900_ai_ci;
```

3. Relationships Summary

Relationship	Description
topics → arguments	One topic can link to one root argument.
arguments → statements	Each argument links to one root claim.
arguments → sources	Each argument has a source (foreign key).

statements → sources	Each statement has a reference source.
premises → arguments/statements	Connects arguments with statements (supports tree structure).
statements → statements	Self-join via counter_statement for rebuttals.

4. Sample Data Inserts

1. Sources

```
INSERT INTO sources (name, text, url) VALUES
  ('Nature2021', 'Study on media and STEM interest', 'https://
     example.org/nature-2021'),
  ('WHO2020', 'Report on sedentary lifestyle risks', 'https://
     example.org/who-2020'),
  ('MetaReview', 'Meta analysis of learning from TV', 'https://
     example.org/meta-review');
```

2. Statements

```
INSERT INTO statements (id, text, counter_statement, source)
   VALUES
  (1, 'Television has an overall positive impact on society.',
     NULL, 'Nature2021'),
  (2, 'Documentaries and science shows increase interest in STEM.
     ', NULL, 'Nature2021'),
  (3, 'Passive viewing rarely leads to sustained learning.', 2, '
     MetaReview'),
  (4, 'Teacher-guided formats convert interest into genuine
     learning.', 3, 'Nature2021'),
  (5, 'Excessive TV correlates with sedentary lifestyle and
     health risks.', 1, 'WHO2020');
```

3. Argument

```
INSERT INTO arguments (id, claim, source) VALUES
  (100, 1, 'Nature2021');
```

4. Premises

```
INSERT INTO premises (argument, premise) VALUES
  (100, 2),
  (100, 3),
  (100, 4),
  (100, 5);
```

5. Topic

```
INSERT INTO topics (id, name, argument_id) VALUES
  (10, 'Television', 100);
```

5. Example Queries

Get the root claim for a topic

```
SELECT t.name AS topic,
       a.id AS argument_id,
       s.id AS claim_id,
       s.text AS claim_text,
       s.source AS claim_source
FROM topics t
JOIN arguments a ON a.id = t.argument_id
JOIN statements s ON s.id = a.claim
WHERE t.name = 'Television';
```

Get all premises for an argument

```
SELECT p.argument,
       s.id AS statement_id,
       s.text AS statement_text,
       s.counter_statement,
       s.source
FROM premises p
JOIN statements s ON s.id = p.premise
WHERE p.argument = 100
ORDER BY s.id;
```

Show counter relationships

```
SELECT child.id AS child_id,
       child.text AS child_text,
       parent.id AS counters_which_statement_id,
       parent.text AS counters_which_statement_text
FROM statements child
LEFT JOIN statements parent ON child.counter_statement = parent.
   id
WHERE child.id IN (2,3,4,5)
ORDER BY child.id;
```

6. Notes and Conventions

- `sources.name` is the primary key used across both arguments and statements.

- `counter_statement` forms self-links for rebuttals or opposing claims.
- `premises` connects arguments with their supporting or counter statements.
- Multiple arguments per topic are possible; `topics.argument_id` can be NULL if needed.