**Project : Amazon Product Reviews**

**Advanced Database Management System Project - MariaDB**

**Presented by**

Bhargav Rishi Medisetti

**Database Description:**

MariaDB: MariaDB is an open-source relational database management system (RDBMS) that originated as a fork of MySQL. Designed for scalability, reliability, and performance, it has gained popularity among developers and enterprises worldwide. It offers features like high availability through clustering, built-in replication, and advanced storage engines to handle diverse workloads. MariaDB supports SQL standards and integrates seamlessly with various programming languages, making it versatile for different application requirements. Its robust security features include encryption, role-based access controls, and audit plugins, ensuring data protection. Additionally, MariaDB is actively maintained by a global community, ensuring regular updates and new feature rollouts.

MariaDB is employed to manage the database of Amazon Product Reviews, for it is powerful and effective and can be used without cost as an open-source relational database management system compatible with MySQL. This will provide a structured way of storing and retrieving Amazon product review data, thereby effectively managing the data at the product, user, and review levels. The schema would have three tables, namely, `users`, `products`, and `reviews`, with primary and foreign keys across these tables to ensure integrity among the data. Each table is thus designed to hold both unique identifiers and essential attributes, such as user information, product descriptions, and review text. This will, in turn, allow MariaDB to perform the CRUD operations: Create, Read, Update, and Delete, hence managing and analyzing the product reviews data efficiently for scenarios where this would be applicable, such as online retail platforms that require user-generated content, ratings of products, and detailed information on products.

**Dataset Description:**

The Amazon sales dataset with over 1,000 products covers quintessential features for ratings and reviews. In other words, one could take a comprehensive view about the performance of each product on Amazon as it contains product ID, name, category, pricing information including both actual and discounted prices, discount percentage, average rating, and rating count. For each product, there is also a brief description, URL to the image, and a link to the official page on Amazon.

This dataset also stores review data left by users, where each review includes attributes such as the ID of the review, title, detailed content, and the ID and name of the user who submitted the review. These kinds of features allow for deeper analysis on the feedback about products for insights into the preferences and sentiments of users.

We divided this dataset into three interrelated tables in our database: `products`, `users`, and `reviews`. The products table contains information on the product of the particular product ID, product name, category, and all pricing details. It acts as a source for identifying the products. The users table holds information specific to each user, like user ID and username, unique to each user in the dataset. It finally has a reviews table, which stores review-specific information, such as the ID of the review, title, and content, along with foreign keys to associated products and users, namely Product\_id and User\_id. This makes every review uniquely identify one product and one user. The structure will enforce data integrity and efficiently enable CRUD on each of these entities independently or together, if necessary.

**Description of the product:**

The Amazon Product Reviews database shall be fully transactional in nature, meaning that it is consistent and always reliable to operate across the product, users, and reviews tables. It should be optimized for an e-commerce environment in which ACID compliance-atomicity, consistency, isolation, and durability-must be strictly observed so as to guarantee integrity and reliability of data in frequently occurring transactions. Each table plays its role in maintaining these properties of a transaction within the database.

**Products Table**: Each product's critical information, like product\_id, product\_name, category, pricing details, and customer feedback metrics of rating and rating\_count, is kept in the products table. This will be designed atomic, where each update would be both valid and complete, whether it is a price adjustment or a rating change. This includes constraints on attributes, making sure data entry is valid, like discounted\_price and actual\_price; indexed product\_id for fast lookups to enable real-time access and update.

**Users Table:** The 'users' table holds data on every customer or reviewer. These include unique identifiers like user\_id and user\_name. It guarantees data integrity since there is a primary key imposed on the column of user\_id, which prevents duplication of entry and hence ensures one version of the truth about any given user. This table serves as a reference for user\_id in the reviews table whenever users create or update their reviews, ensuring referential integrity. The transactional isolation in the users table prevents any conflict in the case of simultaneous updates, such as in the case of changing a user's profile.

**Reviews Table:** This table serves as the bridge between products and users; it includes information about reviews, which includes review\_id, review\_title, review\_content, with an association to product\_id and user\_id for each. Each review is uniquely associated with one product and one user; this reinforces one-to-one relationships, which are necessary in maintaining data integrity. Foreign keys here would be the product\_id and user\_id of the reviews table, ensuring relational integrity across transactions. That is, each review should be connected to an existing product and user. The design of such a schema can allow transactional isolation to prevent frequent creation, reading, or updating of reviews from data inconsistencies.

**ACID Compliance Across Tables:**

Transactions across these tables maintain ACID properties

**Atomicity**: Since operations span multiple tables, adding a new review with a link to an existing user and product either happens or does not happen; partial data update is avoided and doesn't end up being inconsistent.

**Consistency**: Through the constraints like primary and foreign keys on the schema, all transactions leave the database in a valid state.

**Isolation** allows multiple transactions to execute concurrently without interference. For example, several users can simultaneously post or update their reviews; meanwhile, there will not be any case of data corruption or race conditions.

**Durability** ensures committed transactions, such as new reviews, user updates, or product pricing, are written to storage in a way that ensures data survives crashes or sudden shutdowns.

**High-Volume, Real-Time Transaction Support:**

This database structure will support real-time data updates across products, users, and reviews. Changes in data reflect instantly in a user-facing application. Use of indexed primary and foreign keys makes it efficient for the database to handle high-volume requests hence fast retrieval and modification of data if needed.

This generally ensures the transactional design of the whole Amazon Product Reviews database to keep product, user, and review data accurate and reliable. Thus, making such data available to support an e-commerce platform like Amazon.

**Data Structure Description**

### **Products Table:**

The products table is designed to store detailed information about products sold on Amazon. This table includes various attributes that describe each product, such as its unique identifier, name, category, pricing details, ratings, and additional information. Below is a detailed explanation of each field in the products table:

* product\_id - **Datatype**: varchar(100)
  + **Description**: This column stores the unique identifier for each product. It is a variable-length string with a maximum length of 100 characters. .
* product\_name - **Datatype**: varchar(255)
  + **Description**: This column stores the name of the product. It is a variable-length string with a maximum length of 255 characters. This field is mandatory.
* category - **Datatype**: varchar(100)
  + **Description**: This column stores the category to which the product belongs. It is a variable-length string with a maximum length of 100 characters.
* discounted\_price - **Datatype**: decimal(10,2)
  + **Description**: This column stores the discounted price of the product.
* actual\_price - **Datatype**: decimal(10,2)
  + **Description**: This column stores the actual price of the product before any discounts.
* discount\_percentage - **Datatype**: decimal(5,2)
  + **Description**: This column stores the discount percentage applied to the product.
* rating - **Datatype**: decimal(3,2)
  + **Description**: This column stores the average rating of the product.
* rating\_count - **Datatype**: int
  + **Description**: This column stores the number of ratings the product has received. It is an integer.
* About\_product - **Datatype**: varchar(2000)
  + **Description**: This column stores a description or additional information about the product. It is a variable-length string with a maximum length of 2000 characters.

### **Users Table:**

The users table is designed to store information about users who have signed up on Amazon. This table includes attributes that uniquely identify each user and store their names. Below is a detailed explanation of each field in the users table:

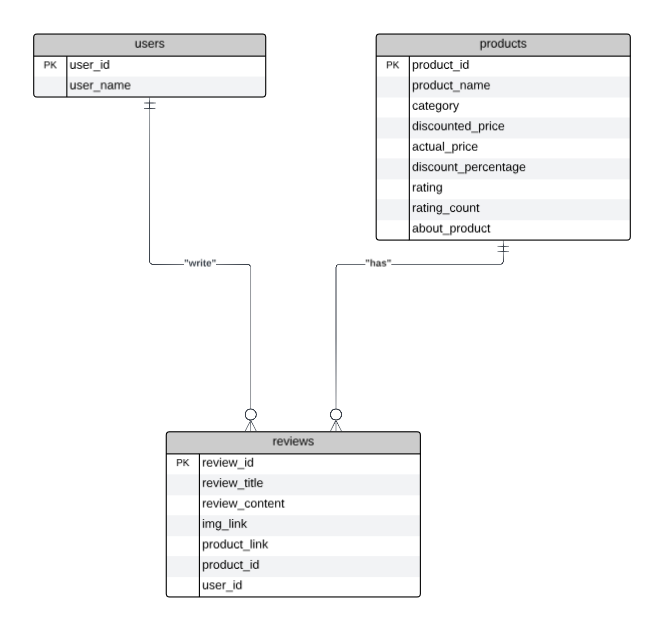
* User\_id - **Type**: varchar(100)
  + **Description**: This column stores the unique identifier for each user. It is a variable-length string with a maximum length of 100 characters. This field is the primary key, ensuring that each user has a unique ID.
* User\_name - **Type**: varchar(255)
  + **Description**: This column stores the name of the user. It is a variable-length string with a maximum length of 255 characters.

### **Reviews Table:**

The reviews table is designed to store information about product reviews. This table includes various attributes that describe each review, such as its unique identifier, title, content, associated product, and user details. Below is a detailed explanation of each field in the reviews table:

* Review\_id - **Type**: varchar(100)
  + **Description**: This column stores the unique identifier for each review. It is a variable-length string with a maximum length of 100 characters. This field is the primary key, ensuring that each review has a unique ID.
* Review\_title - **Type**: varchar(255)
  + **Description**: This column stores the title of the review. It is a variable-length string with a maximum length of 255 characters.
* Review\_content - **Type**: varchar(5000)
  + **Description**: This column stores the content of the review. It is a variable-length string with a maximum length of 5000 characters.
* Img\_link - **Type**: varchar(2083)
  + **Description**: This column stores the URL of an image associated with the review. It is a variable-length string with a maximum length of 2083 characters.
* product\_link - **Type**: varchar(2083)
  + **Description**: This column stores the URL of the product being reviewed. It is a variable-length string with a maximum length of 2083 characters.
* product\_id - **Type**: varchar(100)
  + **Description**: This column stores the unique identifier of the product being reviewed. It is a variable-length string with a maximum length of 100 characters. This field is a foreign key that references the product\_id in the products table.
* user\_id - **Type**: varchar(100)
  + **Description**: This column stores the unique identifier of the user who wrote the review. It is a variable-length string with a maximum length of 100 characters. This field is a foreign key that references the user\_id in the users table.

**Entity Relationship Diagram:**



**CRUD Operations**

**—--Creating Tables**

**CREATE DATABASE project;**

**USE DATABASE project;**

**—--Creating table products: This table is designed to store details about products sold on Amazon**

**CREATE TABLE products (**

**product\_id VARCHAR(100) NOT NULL,**

**product\_name VARCHAR(255) NOT NULL,**

**category VARCHAR(100),**

**discounted\_price DECIMAL(10, 2),**

**actual\_price DECIMAL(10, 2),**

**discount\_percentage DECIMAL(5, 2),**

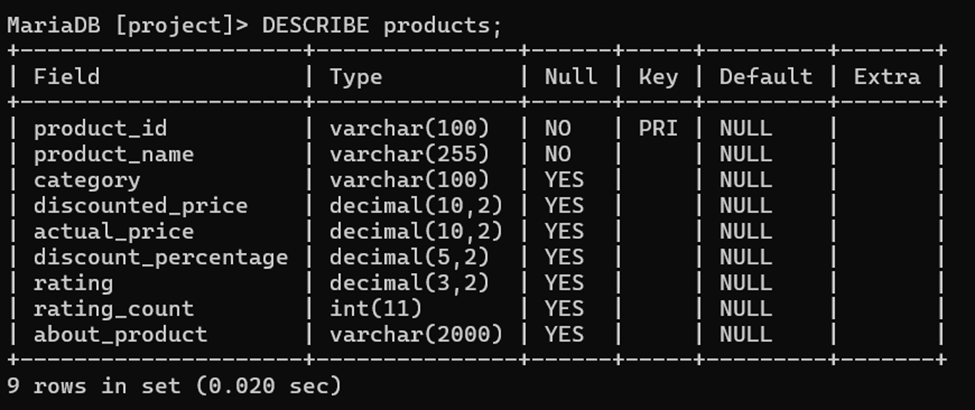
**rating DECIMAL(3, 2),**

**rating\_count INT,**

**about\_product VARCHAR(2000),**

**PRIMARY KEY (product\_id)**

**);**

****

**—--Creating table users:**

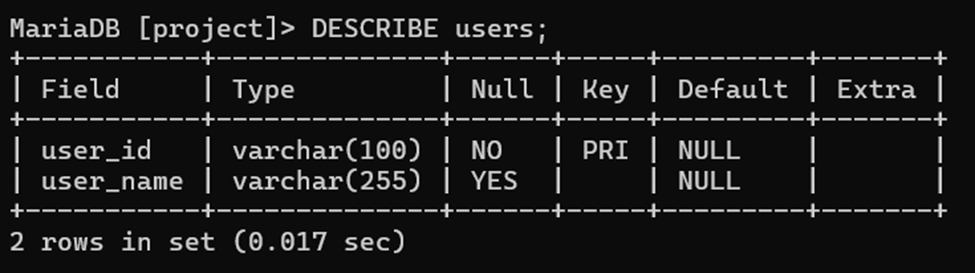
**CREATE TABLE users (**

**user\_id VARCHAR(100) NOT NULL,**

**user\_name VARCHAR(255),**

**PRIMARY KEY (user\_id)**

**);**

****

**—--Creating table reviews:**

**CREATE TABLE reviews (**

**review\_id VARCHAR(100),**

**review\_title VARCHAR(255),**

**review\_content VARCHAR(5000),**

**img\_link VARCHAR(2083),**

**product\_link VARCHAR(2083),**

**product\_id VARCHAR(100) NOT NULL,**

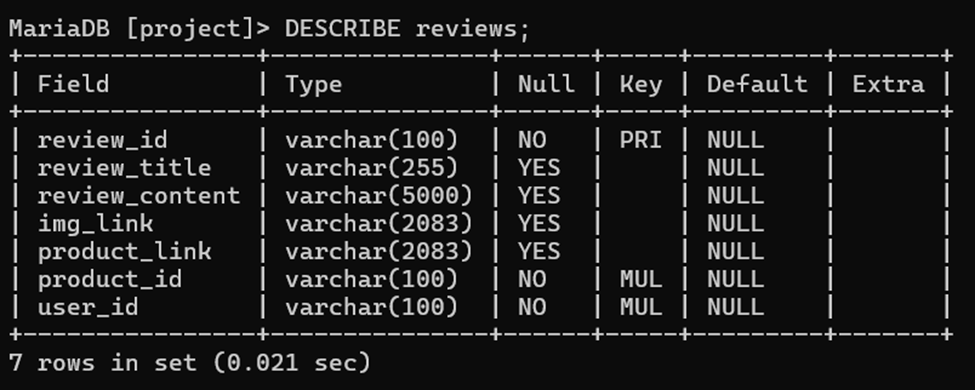
**user\_id VARCHAR(100) NOT NULL,**

**PRIMARY KEY (review\_id),**

**CONSTRAINT fk\_product FOREIGN KEY (product\_id) REFERENCES products(product\_id),**

**CONSTRAINT fk\_user FOREIGN KEY (user\_id) REFERENCES users(user\_id)**

**);**

****

****

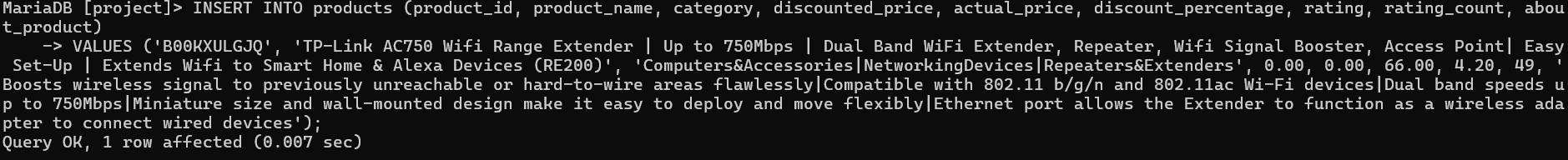
**INSERT STATEMENTS**

**PRODUCTS Table:**

**—--Insert a new product:**

**INSERT INTO products (product\_id, product\_name, category, discounted\_price, actual\_price, discount\_percentage, rating, rating\_count, about\_product)**

**VALUES ('B00KXULGJQ', 'TP-Link AC750 Wifi Range Extender | Up to 750Mbps | Dual Band WiFi Extender, Repeater, Wifi Signal Booster, Access Point| Easy Set-Up | Extends Wifi to Smart Home & Alexa Devices (RE200)', 'Computers&Accessories|NetworkingDevices|Repeaters&Extenders', 0.00, 0.00, 66.00, 4.20, 49, 'Boosts wireless signal to previously unreachable or hard-to-wire areas flawlessly|Compatible with 802.11 b/g/n and 802.11ac Wi-Fi devices|Dual band speeds up to 750Mbps|Miniature size and wall-mounted design make it easy to deploy and move flexibly|Ethernet port allows the Extender to function as a wireless adapter to connect wired devices');**

****

**—-Insert another new product:**

**INSERT INTO products (product\_id, product\_name, category, discounted\_price, actual\_price, discount\_percentage, rating, rating\_count, about\_product)**

**VALUES ('B00NFD0ETQ', 'Logitech G402 Hyperion Fury USB Wired Gaming Mouse, 4,000 DPI, Lightweight, 8 Programmable Buttons, Compatible for PC/Mac - Black', 'Electronics', 0.00, 0.00, 31.00, 4.60, 10, 'HIGH SPEED TRACKING : Fusion engine delivers one of the highest gaming mouse tracking speeds of up to 500 IPS. Requirements : Windows 8, Windows 8.1, Windows 7 or Windows Vista. Length Cable: 2.1 m|ON-THE-FLY DPI : Shift through up to four DPI settings, from pixel-precise targeting (250 DPI) to lightning-fast maneuvers (4000 DPI)');**

**SELECT Statements**

**—--Select all products in a specific category, ordered by rating:**

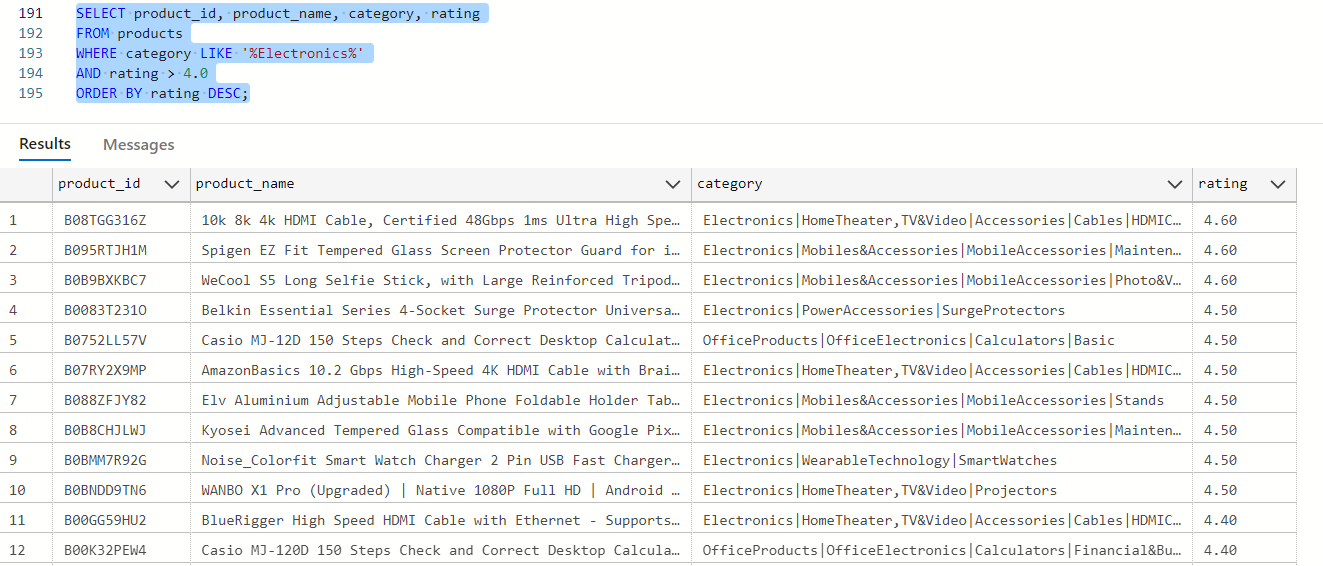
**SELECT product\_id, product\_name, category, rating**

**FROM products**

**WHERE category LIKE '%Electronics%'**

**AND rating > 4.0**

**ORDER BY rating DESC;**

****

**—--Select the average discounted percentage of products in each category, having more than 10 products:**

**SELECT category, AVG(discount\_percentage) AS avg\_discount\_percentage**

**FROM products**

**GROUP BY category**

**HAVING COUNT(product\_id) > 10;**

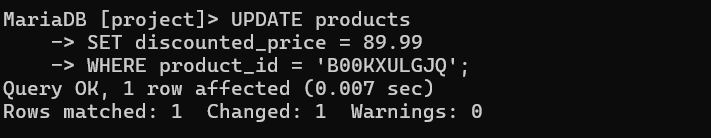
**UPDATE Statements**

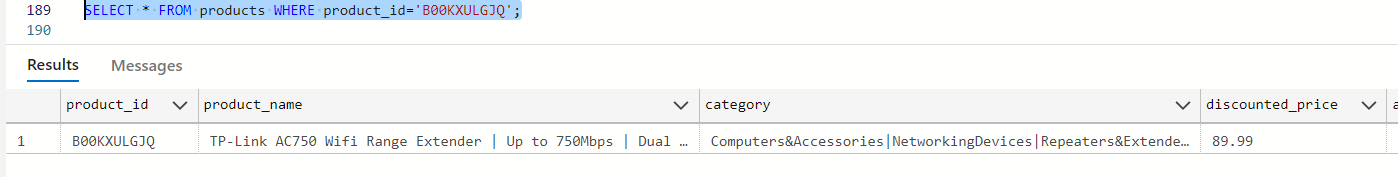
**—--Update the discounted price of a specific product:**

**UPDATE products**

**SET discounted\_price = 89.99**

**WHERE product\_id = 'B00KXULGJQ';**

****

****

**—--Update the rating of products in a specific category:**

**UPDATE products**

**SET rating = 4.8**

**WHERE category = 'Electronics';**

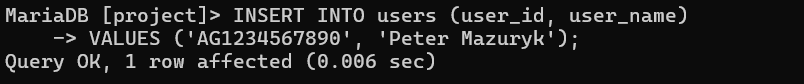
**USERS Table**

**—--INSERT Statements:**

**—--Inserting new users:**

**INSERT INTO users (user\_id, user\_name)**

**VALUES ('AG1234567890', 'Peter Mazuryk');**

****

**INSERT INTO users (user\_id, user\_name)**

**VALUES ('AH0987654321', 'Sean Bezerra');**

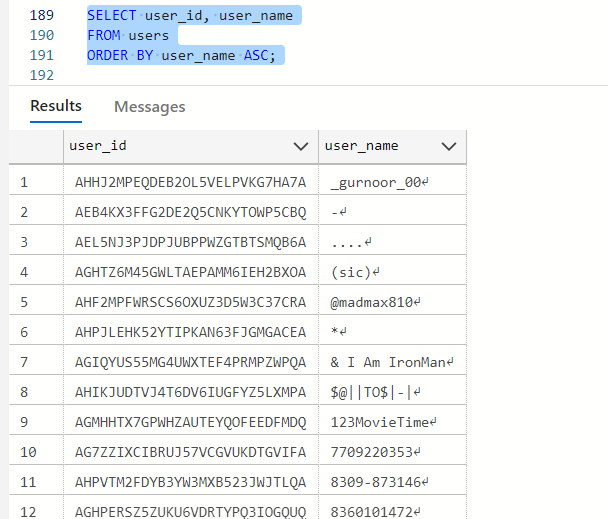
**SELECT Statements**

**—--Select all users ordered by user\_name:**

**SELECT user\_id, user\_name**

**FROM users**

**ORDER BY user\_name ASC;**

****

**—--Select users with a specific user\_id pattern ordered by user\_name in descending order:**

**SELECT user\_id, user\_name**

**FROM users**

**WHERE user\_id LIKE 'AG%'**

**ORDER BY user\_name DESC;**

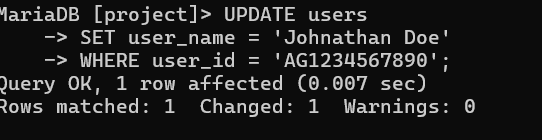
**UPDATE Statements:**

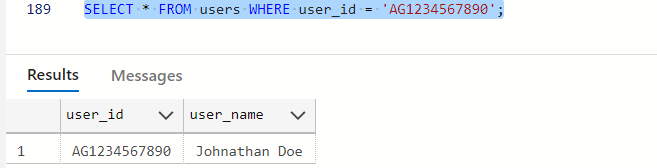
**—--Update the user\_name of a specific user:**

**UPDATE users**

**SET user\_name = 'Johnathan Doe'**

**WHERE user\_id = 'AG1234567890';**

****

****

**—--Update the user\_name of users with a specific user\_id pattern:**

**UPDATE users**

**SET user\_name = 'Anonymous'**

**WHERE user\_id LIKE 'AH%';**

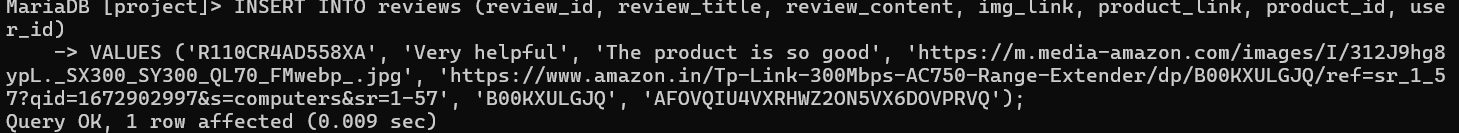
**REVIEWS Table**

**—--INSERT Statements:**

**--Inserting new reviews:**

**INSERT INTO reviews (review\_id, review\_title, review\_content, img\_link, product\_link, product\_id, user\_id)**

**VALUES ('R110CR4AD558XA', 'Very helpful', 'The product is so good', 'https://m.media-amazon.com/images/I/312J9hg8ypL.\_SX300\_SY300\_QL70\_FMwebp\_.jpg', 'https://www.amazon.in/Tp-Link-300Mbps-AC750-Range-Extender/dp/B00KXULGJQ/ref=sr\_1\_57?qid=1672902997&s=computers&sr=1-57', 'B00KXULGJQ', 'AFOVQIU4VXRHWZ2ON5VX6DOVPRVQ');**

****

**INSERT INTO reviews (review\_id, review\_title, review\_content, img\_link, product\_link, product\_id, user\_id)**

**VALUES ('R112HB5700T6SG', 'Good but lack some features', 'Its a good wireless mouse but sometimes lags when I do heavy tasks like video editing.', 'https://m.media-amazon.com/images/W/WEBP\_402378-T2/images/I/31C+JNS-7PL.\_SY300\_SX300\_.jpg', 'https://www.amazon.in/Lenovo-GY50R91293-Wireless-Mouse-Black/dp/B07J2NGB69/ref=sr\_1\_258?qid=1672903007&s=computers&sr=1-258', 'B07J2NGB69', 'AFHAATSERIT56U7QXWGJIL4MO3KQ');**

**SELECT Statements**

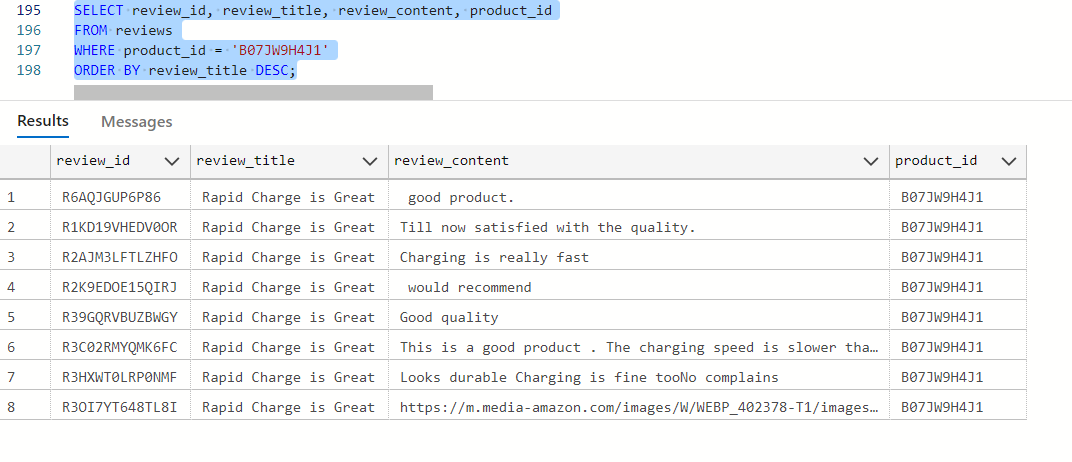
**—--Select all reviews for a specific product, ordered by review\_id:**

**SELECT review\_id, review\_title, review\_content, product\_id**

**FROM reviews**

**WHERE product\_id = 'B07JW9H4J1'**

**ORDER BY review\_title DESC;**

****

**—--Select the number of reviews for each product, having more than 5 reviews:**

**SELECT product\_id, COUNT(1) AS no\_of\_reviews**

**FROM reviews**

**GROUP BY product\_id**

**HAVING no\_of\_reviews > 5;**

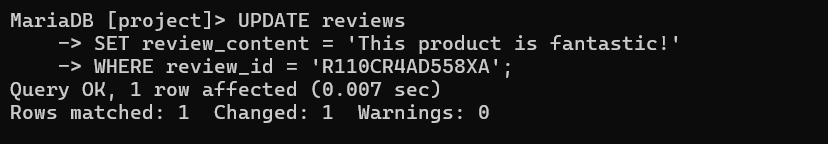
**UPDATE Statements**

**—--Update the review content of a specific review:**

**UPDATE reviews**

**SET review\_content = 'This product is fantastic!'**

**WHERE review\_id = 'R110CR4AD558XA';**

****

****

**--Update the review title of reviews for a specific product:**

**UPDATE reviews**

**SET review\_title = 'Rapid Charge is Great'**

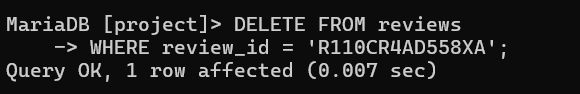
**WHERE product\_id = 'B07JW9H4J1';**

**DELETE Statements:**

**--Delete a specific review:**

**DELETE FROM reviews**

**WHERE review\_id = 'R110CR4AD558XA';**

****

**—--Delete all reviews for a specific product:**

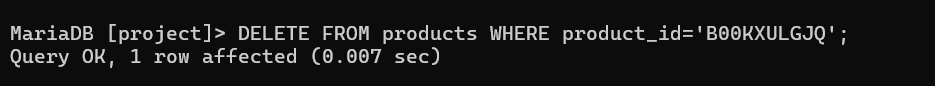
**DELETE FROM reviews**

**WHERE product\_id = 'B07JW9H4J1';**

**--Delete a specific product:**

**DELETE FROM products**

**WHERE product\_id = 'B00KXULGJQ';**

****

**—--Delete all products in a specific category:**

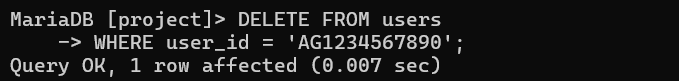
**DELETE FROM products**

**WHERE category = 'Home Appliances';**

**—--Delete a specific user:**

**DELETE FROM users**

**WHERE user\_id = 'AG1234567890';**

****

**—--Delete users with a specific user\_id pattern:**

**DELETE FROM users**

**WHERE user\_id LIKE 'AH%';**