PROJECT

DENIAL OF SERVICE

USING MYSQL

RELATIONAL DATABASE STRUCTURE BASED ON NETWORK SECURITY

Prepared By: Guided By: Anmol Baghel Zakir Hussain

TABLE OF CONTENTS

1. DoS Attack Description.

2. Databases used in this Project.

3. Tables used in each of the Databases.

4. Queries identified by the Network Infra

security team.

5. Final Goal of the Project.

**1. DoS Attack Description.**

**What is a Denial of Service (DoS) Attack?**

A Denial of Service (DoS) attack is a type of cyberattack where an attacker attempts to make a computer or network resource unavailable by overwhelming it with traffic or requests.

The goal of a DoS attack is to exhaust the resource's capacity, making it unable to handle legitimate requests.

Imagine someone trying to shut down a popular restaurant by calling in a huge number of fake orders, overwhelming the kitchen and staff. That's basically what a Denial of Service (DoS) attack is, but instead of a restaurant, it's a computer or network resource that's being targeted.

**Types of DoS Attacks**

There are three main types of DoS attacks, each with its own unique characteristics:

* **Volume-based attacks**: These attacks overwhelm the network with traffic to consume bandwidth. Imagine a massive crowd of people trying to access a website at the same time, causing the server to slow down or crash. This type of attack is often carried out using botnets or compromised devices.
* **Protocol attacks**: These attacks exploit weaknesses in network protocols to consume resources. It's like finding a vulnerability in a system's security software, allowing attackers to gain unauthorized access. Protocol attacks can be particularly devastating, as they can affect multiple systems and networks.
* **Application attacks**: These attacks target specific applications or services to consume resources. For example, an attacker might target a popular online game, overwhelming the server with traffic and causing it to crash. Application attacks can be highly targeted, making them difficult to detect and prevent.

**Common DoS Attack Techniques**

* **Flooding**: Sending a large number of requests or traffic to overwhelm the resource. This is like trying to drink from a firehose, where the sheer volume of water is too much for the system to handle.
* **Buffer overflow**: Sending more data than a buffer can handle, causing it to crash. This is like trying to stuff too many people into a small room, causing it to become overcrowded and unusable.
* **Malformed packets**: Sending packets with incorrect or malicious data to cause errors. This is like sending a package with the wrong address, causing it to get lost in transit.
* **SYN flooding**: Sending a large number of SYN requests to consume server resources. This is like sending a flood of requests to a server, overwhelming it with traffic and causing it to slow down or crash.

**How DoS Attacks are Launched**

DoS attacks can be launched using various tools and techniques, including:

* **Botnets**: Networks of compromised devices used to launch attacks. These devices can be infected with malware, allowing attackers to remotely control them and launch attacks.
* **Malware**: Software designed to harm or exploit systems. Malware can be used to launch DoS attacks, as well as steal sensitive data or disrupt system operations.
* **Scripting**: Using scripts to automate attack processes. This allows attackers to launch complex attacks with minimal effort, making it easier to carry out large-scale DoS attacks.

**Protecting Against DoS Attacks**

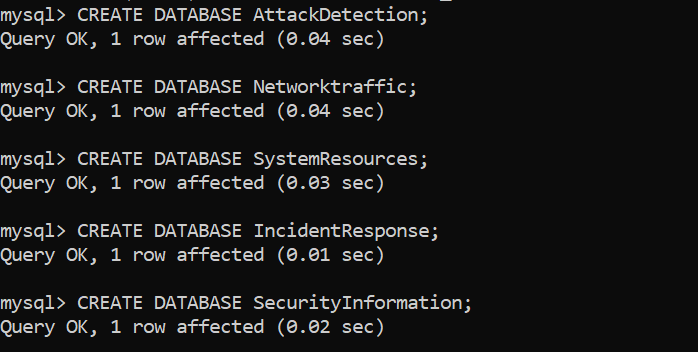
To protect against DoS attacks, organizations can use a variety of techniques, including:

* **Firewalls**: To block malicious traffic and prevent attacks from reaching the system.
* **Intrusion Detection/Prevention Systems (IDS/IPS)**: To detect and prevent attacks in real-time, reducing the risk of system compromise.
* **Load balancing**: To distribute traffic across multiple resources, reducing the risk of overload and improving system performance.
* **Content Delivery Networks (CDNs)**: To cache content and reduce server load, making it more difficult for attackers to launch successful DoS attacks.
* **DDoS mitigation services**: Specialized services that detect and mitigate DoS attacks, providing an additional layer of protection for organizations.

**2. Databases used in this Project.**

**- Create five databases using the below syntax:**

**create database <name of database>;**

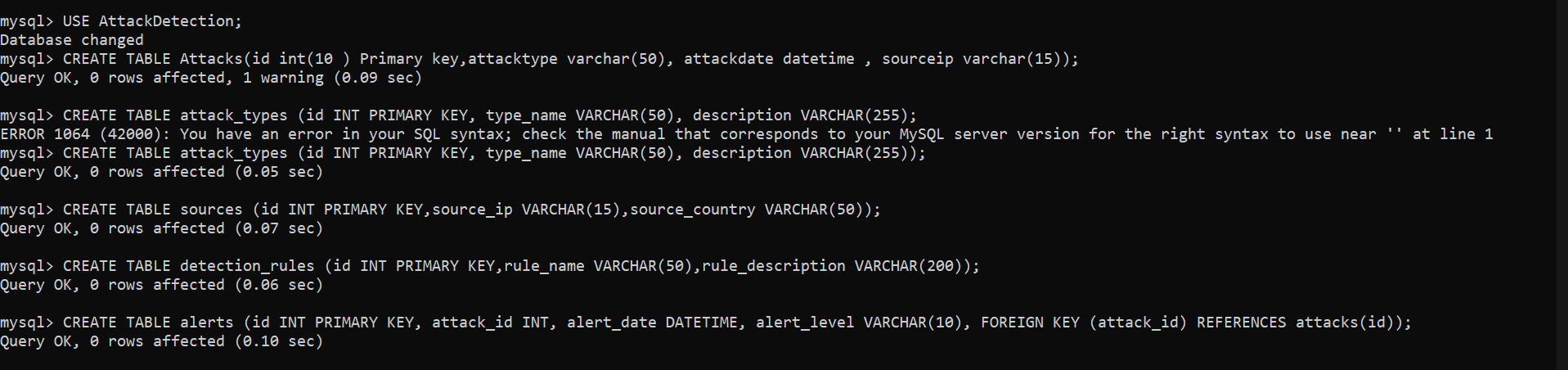
****

**3) Tables used in each of the Databases:**

**Using first database, ‘Attack\_Detection’ ,**

**Create 5 tables namely,**

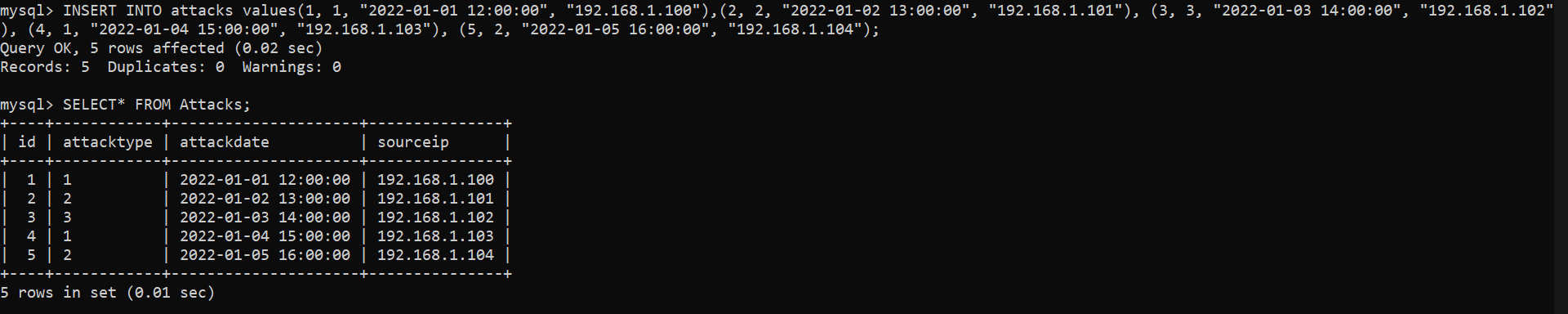
1. Attacks
2. Attack\_types
3. Sources
4. Detection\_rules
5. Alerts



**Inserting data in each tables and displaying the entire table:**

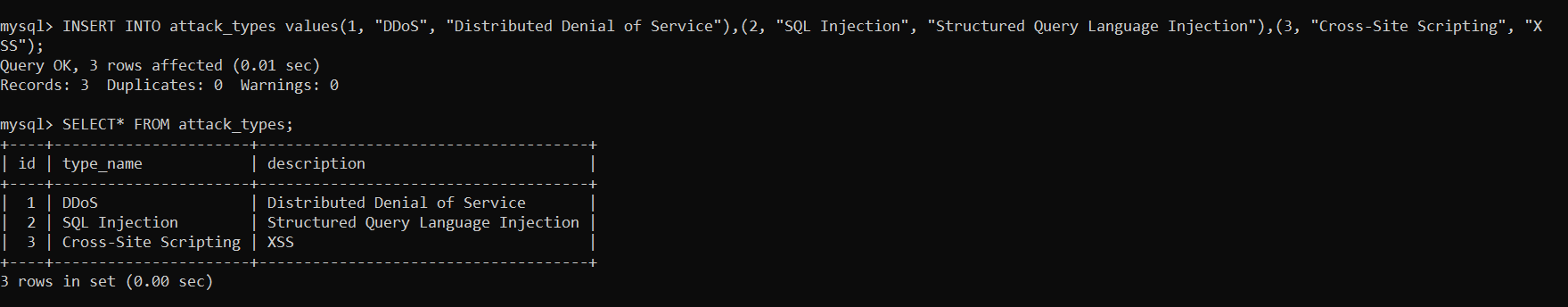
**1.’Attacks’ table:**

This Table Logs all detected cyberattacks, including the type of attack (e.g., DoS, SQL Injection), the date it occurred, and the source IP of the attack. This helps track and analyze patterns of malicious activity.

****

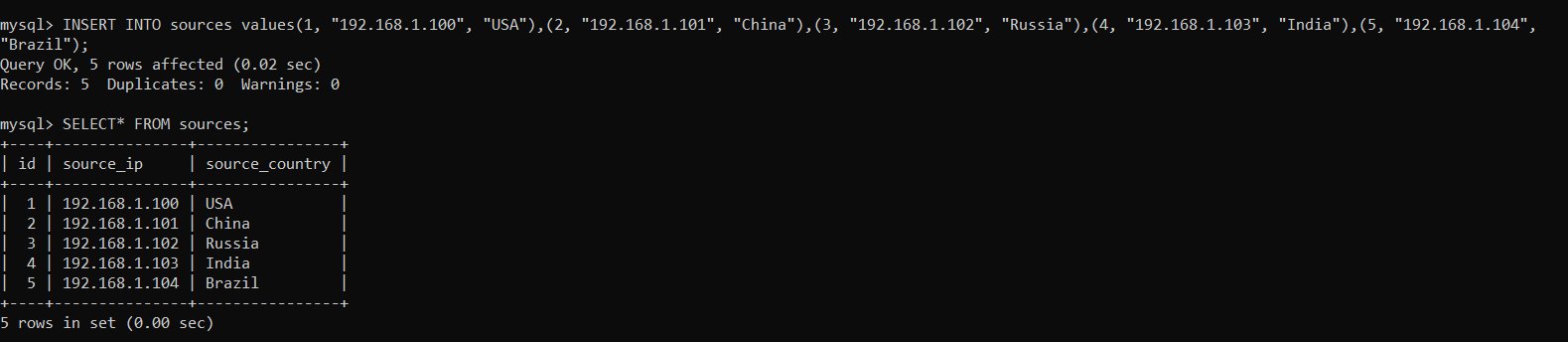
**2.’Attack\_types’ Table:**

This table contains a list of different attack types (e.g., DoS, Brute Force) with a brief description of each. This helps in categorizing incidents for better response and reporting.

****

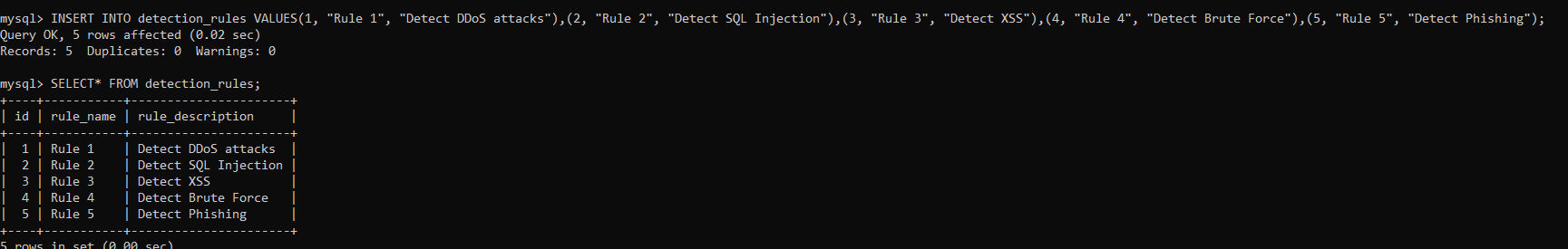
**3.’Sources’ Table:**

This Table records information about the IP addresses from which attacks originated and the associated country. This is useful for identifying attack sources and assessing geographic trends.

****

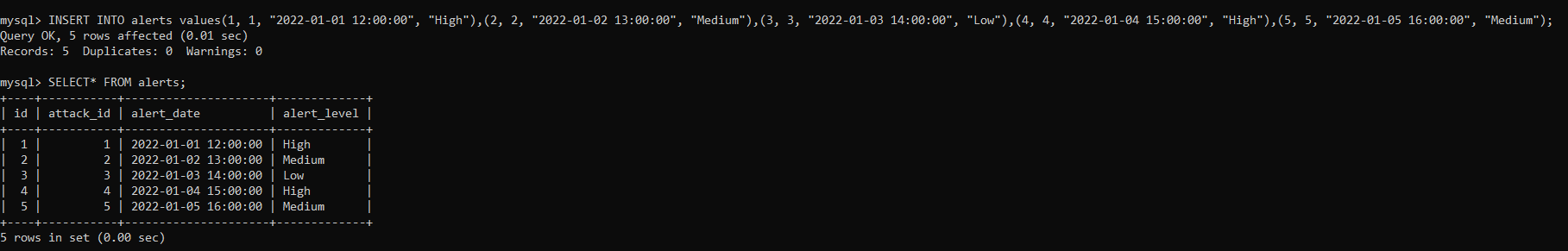
**4.’Detection Rules’ Table:**

This Table defines rules for detecting various types of attacks, such as specific patterns or behaviors. These rules trigger alerts when a match is found, enabling real-time protection.

****

**5.’Alerts’ Table:**

This Table logs alerts triggered by detected attacks, including the severity (e.g., high, medium, low) and timestamp. This table helps prioritize responses based on the severity of the threat.

****

**Using second database, named as ‘Network\_Traffic’,**

**Create 5 tables namely,**

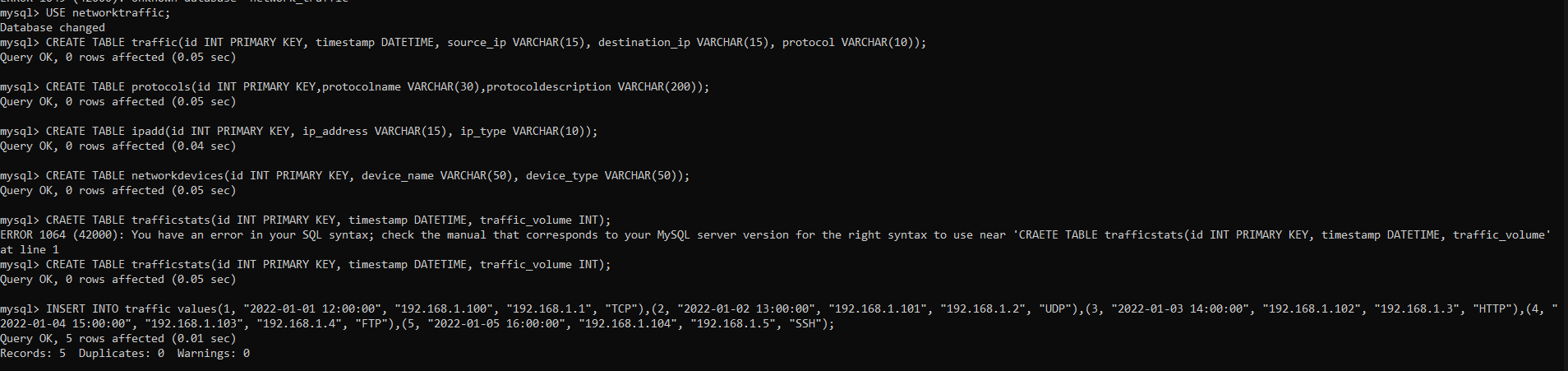
1.Traffic

2.Protocols

3.Ipadd

4.NetworkDevices

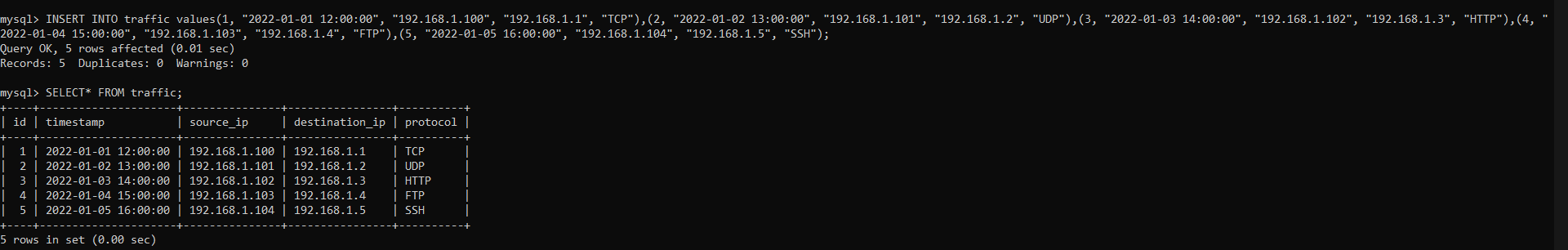
5.TrafficStats



**Inserting data in each tables and displaying the entire table:**

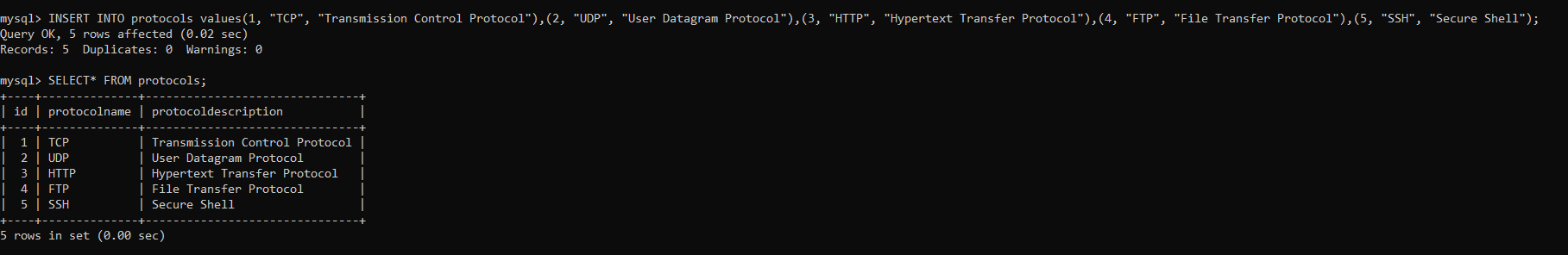
**1.’Traffic’ Table:**

Tracks all network traffic by recording timestamps, source and destination IP addresses, and the protocol used. This table is key to monitoring and analyzing network behavior during attacks.

****

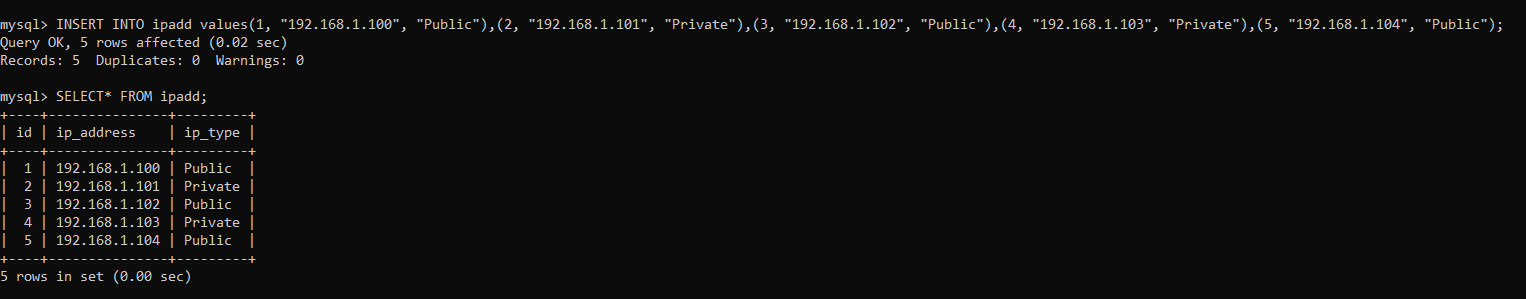
**2.’Protocols’ Table:**

Lists the various network protocols (e.g., TCP, UDP) used in traffic, with a description of each. This helps in understanding which protocols might be targeted in attacks.

****

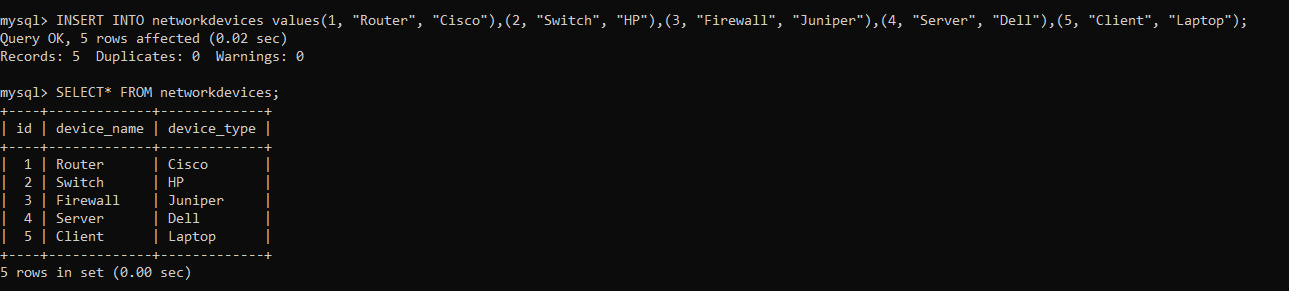
**3.’Ipadd’ Table:**

Stores both public and private IP addresses in the network, classifying them by type. This aids in differentiating internal versus external traffic and helps identify anomalies.

****

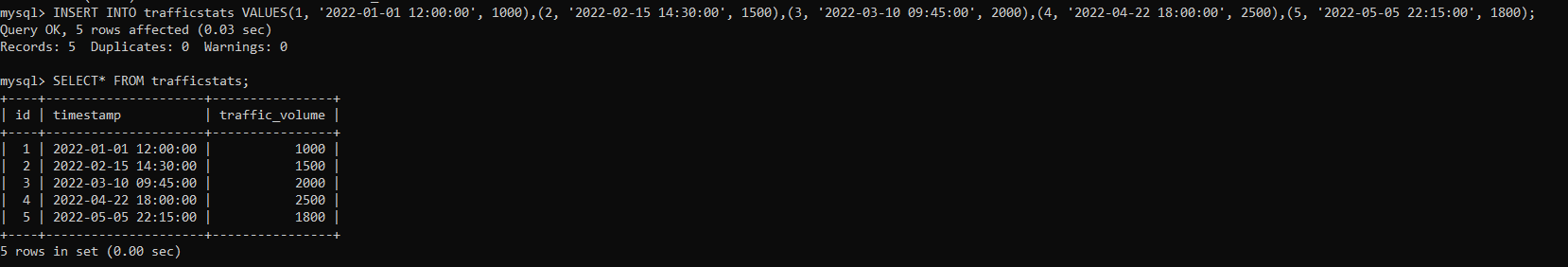
**4.’NetworkDevices’ Table:**

Contains details about the devices used in the network, such as routers, switches, and firewalls, with information on their types and manufacturers. This table helps in mapping the network infrastructure.

****

**5.’TrafficStats’ Table:**

Logs the volume of traffic flowing through the network at specific timestamps, which is useful for spotting unusual spikes in traffic that could indicate an attack like a DoS.

****

**Using Third database, named as ‘System\_Resources’,**

**Create 5 tables namely,**

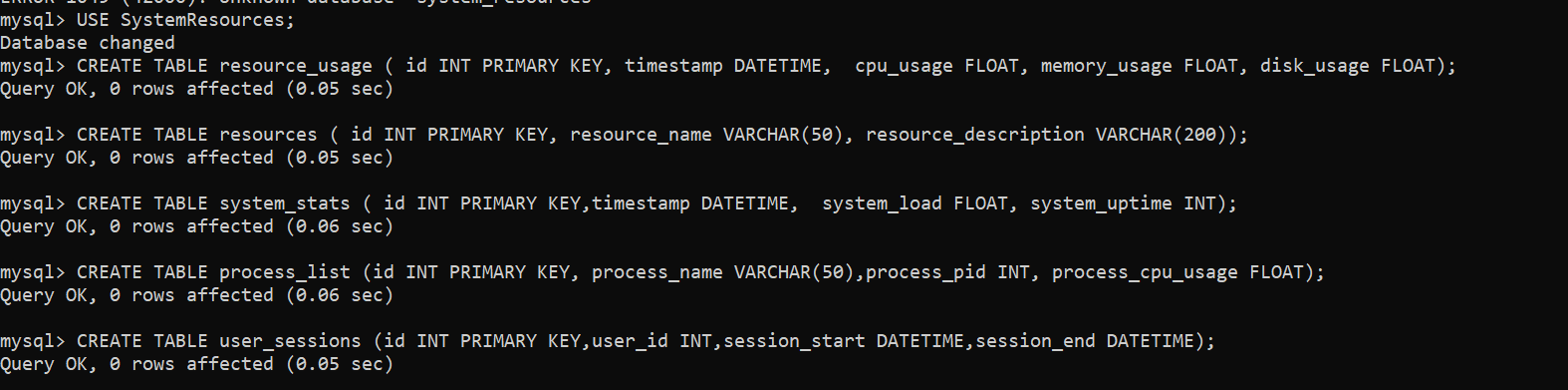
1.Resource\_usage

2.Resources

3.System\_stats

4.Process\_list

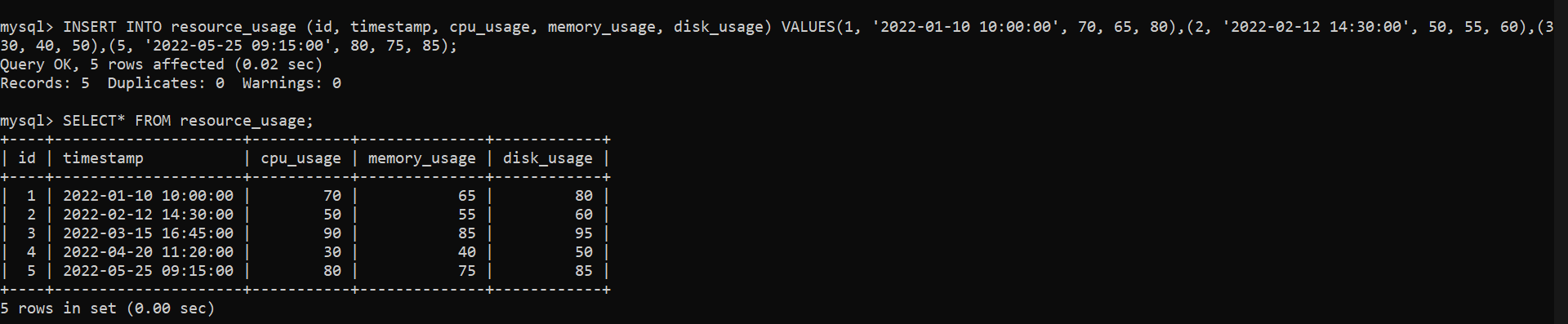
5.User\_sessions

****

**Inserting data in each tables and displaying the entire table:**

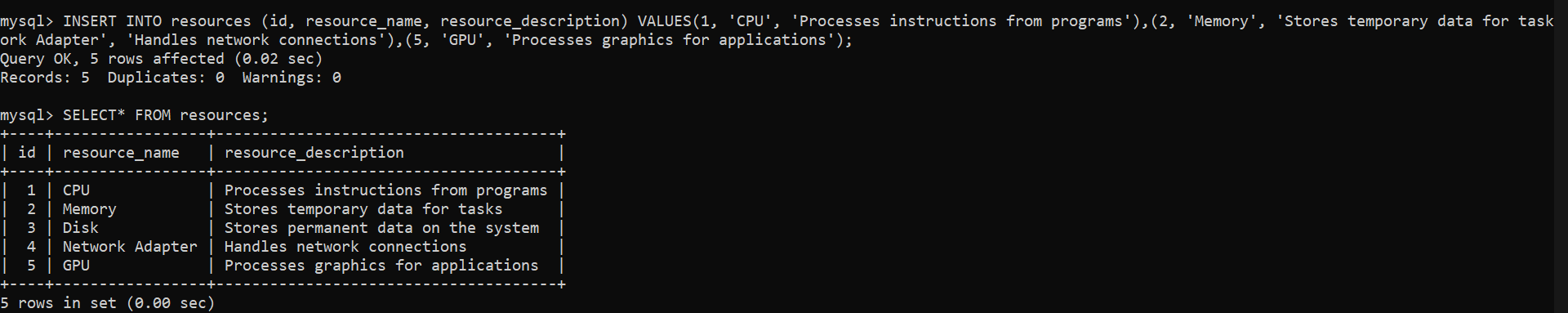
1.Resource\_usage:

Tracks the usage of critical system resources like CPU, memory, and disk space over time. Monitoring this data helps identify resource exhaustion during attacks.



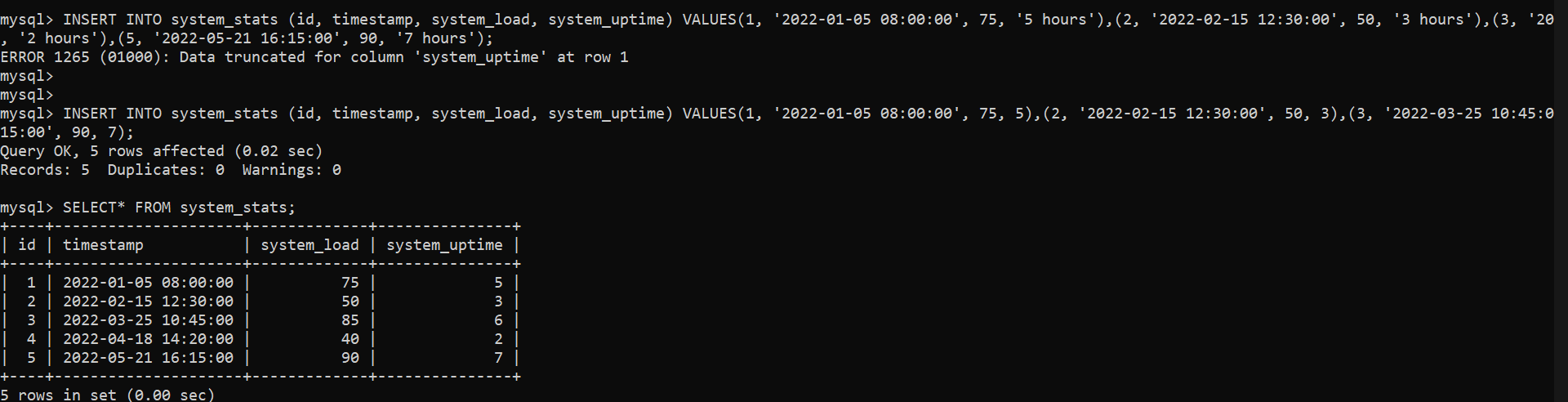
2.Resources:

It contains descriptions of system components, such as servers or databases, detailing their purpose and usage. This helps in inventory management and resource tracking.



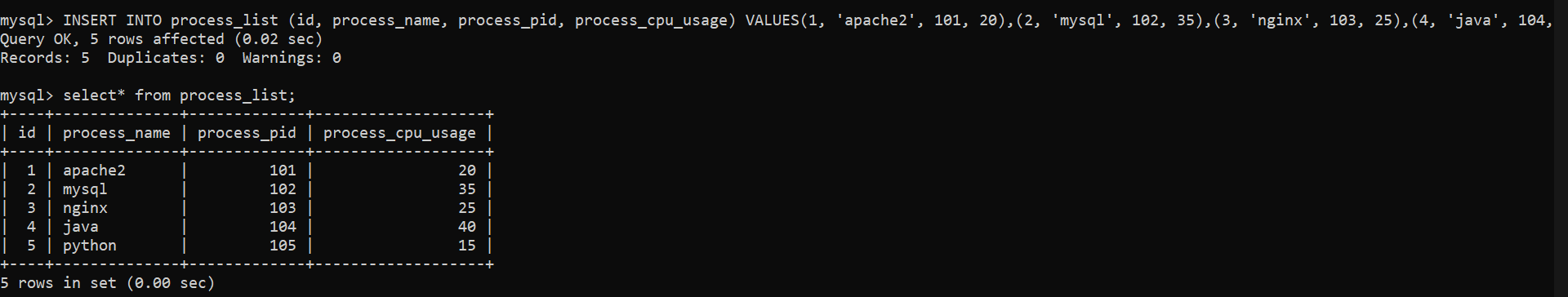
3.System\_stats:

Logs overall system performance metrics, including system load and uptime, at regular intervals. This table is crucial for understanding system health and identifying performance bottlenecks.



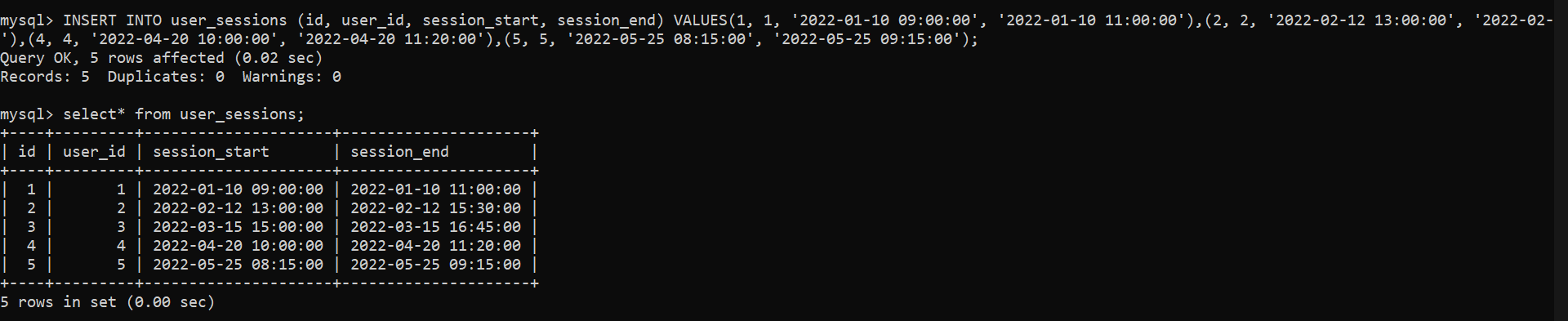
4.Process\_list:

Records running system processes, including process names, process IDs (PID), and the CPU usage for each process. This helps monitor what processes might be impacted or exploited during an attack.



5.User\_sessions:

This Table tracks the start and end times of user sessions on the system. Monitoring user activity helps detect unauthorized access or prolonged session durations that could signal a security threat.



**Using Fourth database, named as ‘Incident\_Response’,**

Create 5 tables namely,

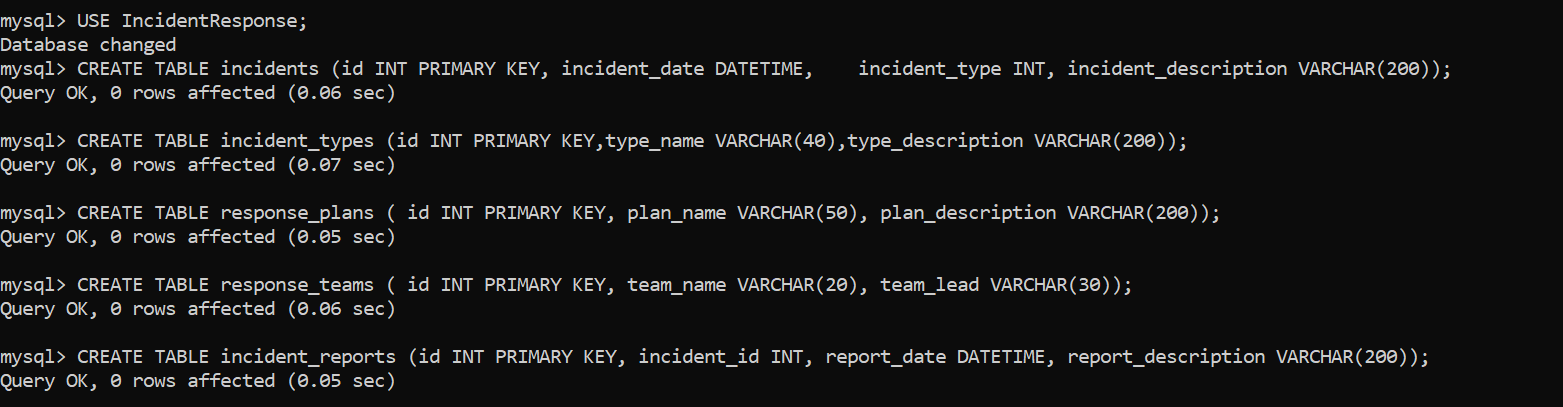
1. Incidents

2.Incident\_type

3.Response\_plans

4.Response\_teams

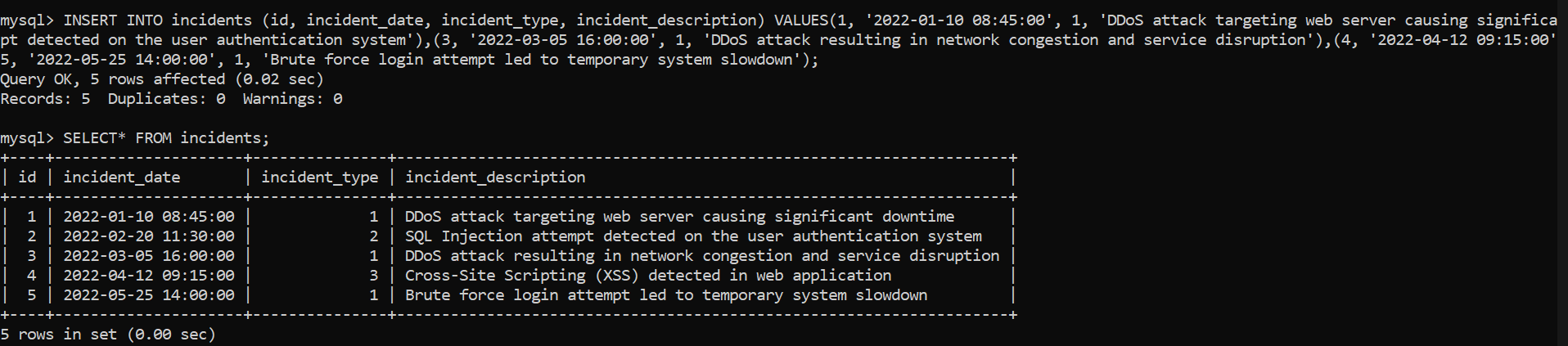
5.Incident\_reports

****

**Inserting data in each table and displaying the entire table:**

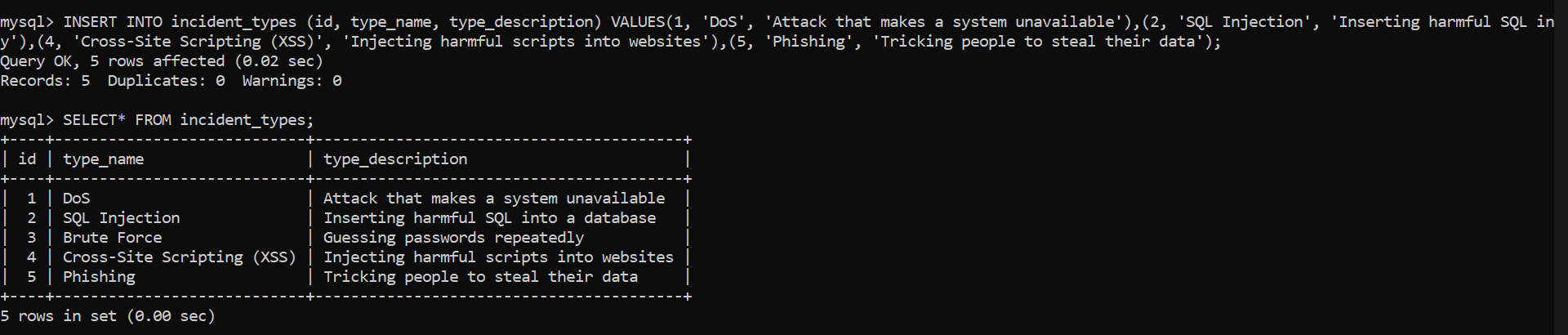
**1. Incidents:**

Logs all security incidents, recording the date, type, and a brief description of the incident. This is essential for maintaining a history of breaches and responses**.**

****

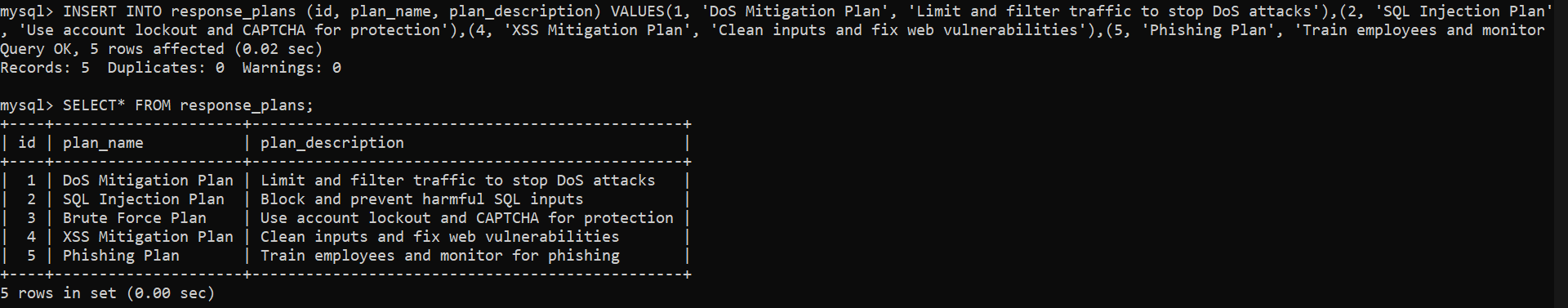
**2.Incident\_type:**

It categorizes incidents into types like DoS, SQL Injection, and Brute Force, with descriptions for each. This helps in quick identification and classification of incidents for a faster response.

****

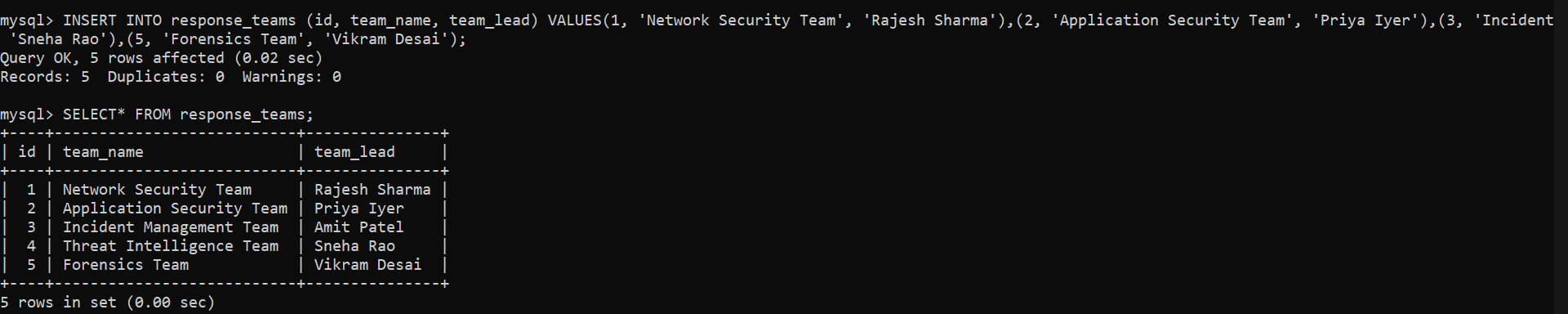
**3.Response\_plans:**

This Table contains predefined response plans for different types of incidents, outlining steps to mitigate the attack and restore normal operations. This ensures a consistent and structured response to threats.

****

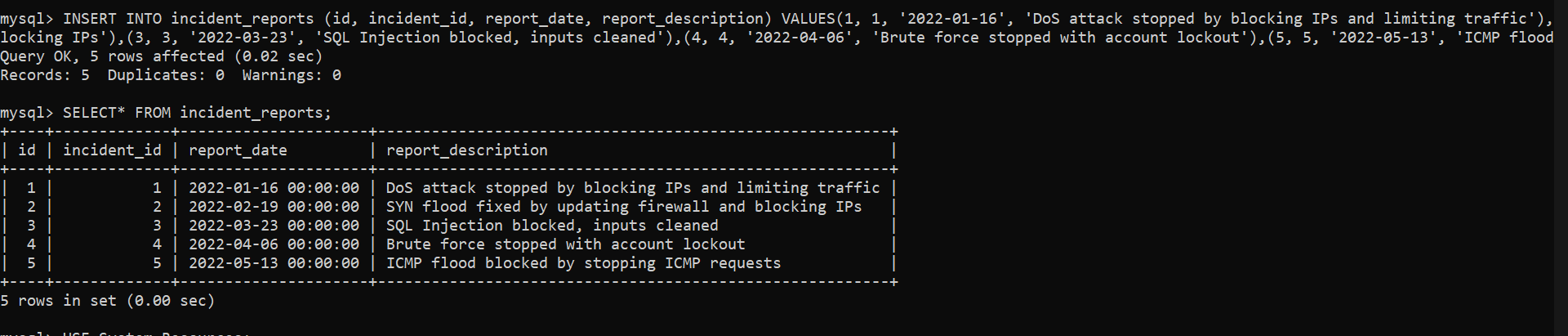
**4.Response\_teams:**

Lists the teams responsible for handling incidents, including the team names and the team leads. This helps in assigning roles and coordinating responses during an attack.

****

**5.Incident\_reports:**

This Table stores reports generated after incidents, including the response actions taken, the date the report was created, and a brief description of the outcome. These reports are valuable for post-incident analysis.



**Using Fifth database, named as ‘Security\_Information’,**

**Create 5 tables namely,**

1.Vulnerabilities

2.Patches

3.Security\_Advisories

4.Threat\_Intelligence

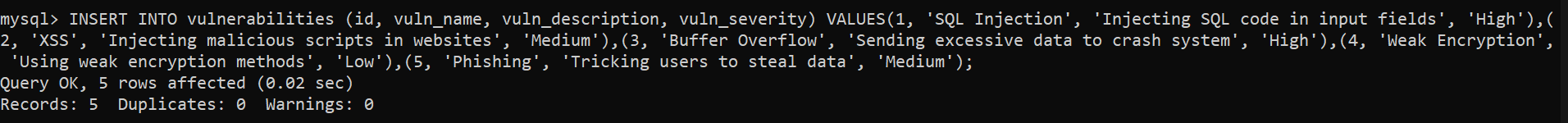
5.Security\_incidents

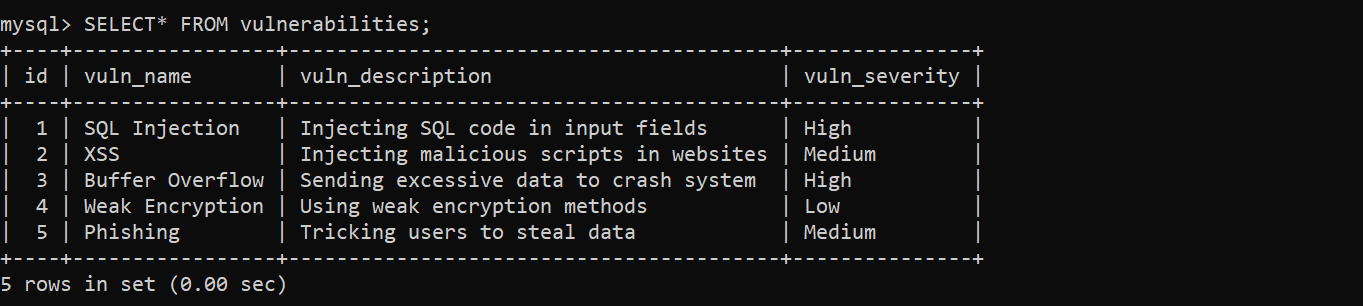


**Inserting data in each tables and displaying the entire table:**

1. Vulnerabilities:

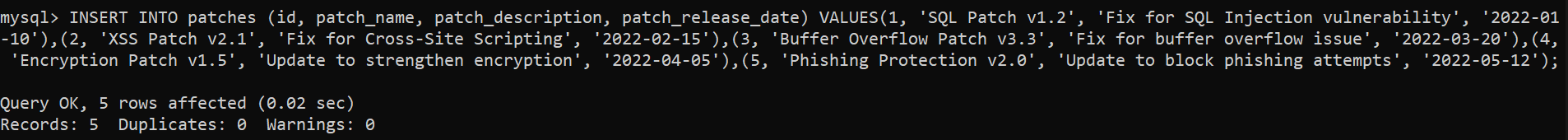
It lists known vulnerabilities with severity and descriptions. This helps prioritize patching and mitigation efforts based on the risk posed by each vulnerability.

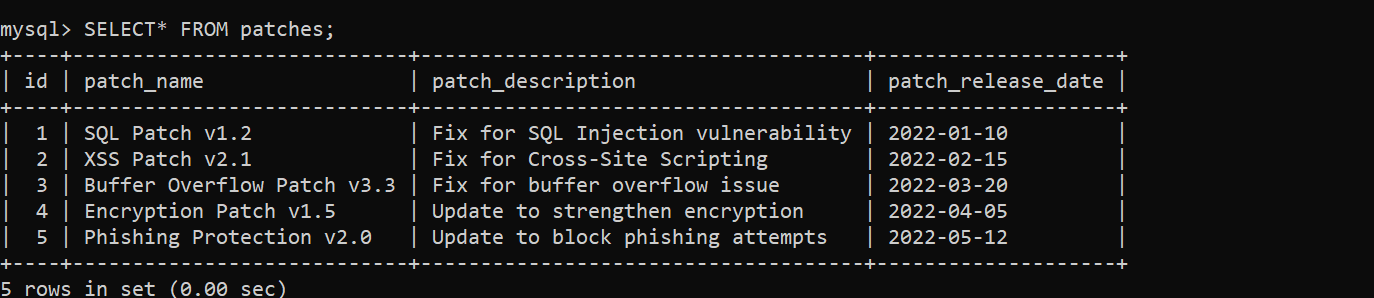




2.Patches:

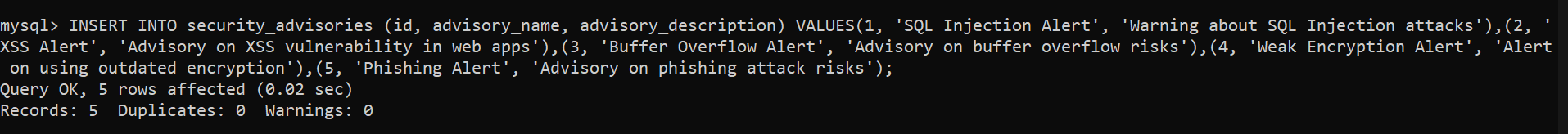
This Table stores information on patches released to fix vulnerabilities, including the patch name, description, and release date. This table is critical for tracking system updates and ensuring vulnerabilities are addressed.

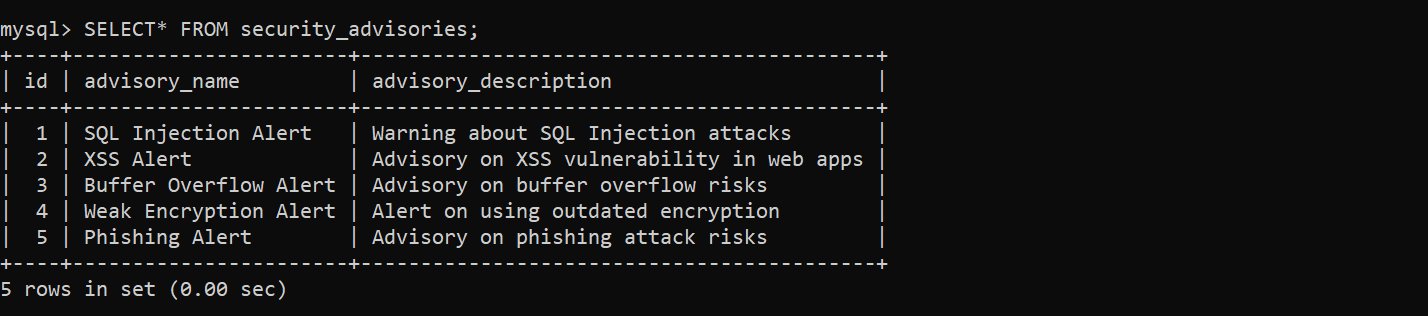




3.Security\_Advisories:

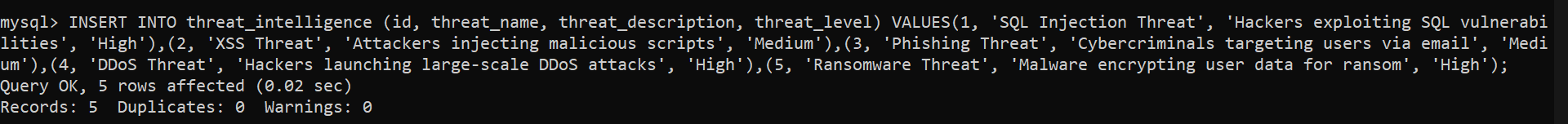
It logs advisories issued to inform about emerging threats, including details on how to mitigate or respond to those threats. This helps keep the security team informed of the latest risks.

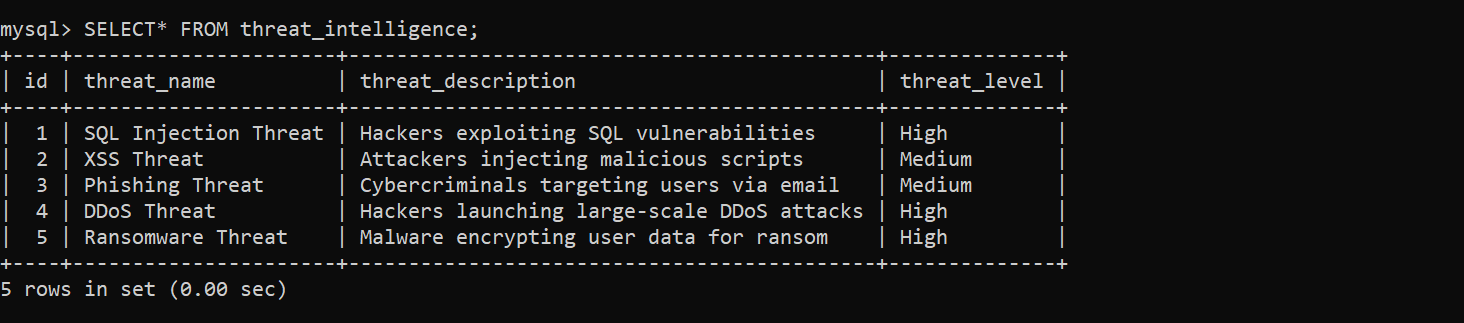




4.Threat\_Intelligence:

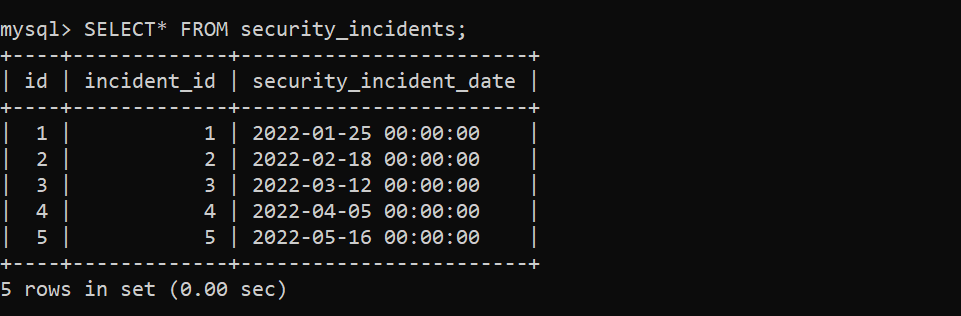
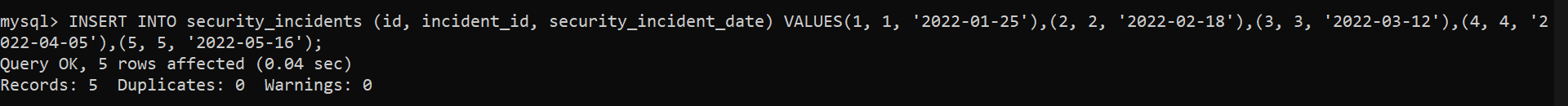
This Table contains intelligence on current and potential security threats, with descriptions and threat levels. This information aids in proactive defense planning and resource allocation.





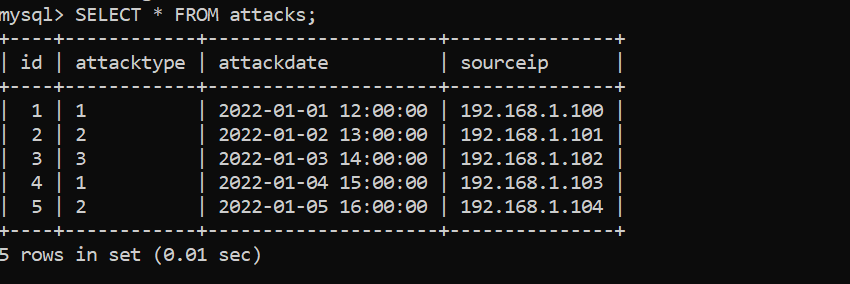
5.Security\_incidents:

It tracks all security incidents, linking them to the main incident ID and recording the date of the incident. This helps maintain a comprehensive record of security breaches for analysis and reporting.

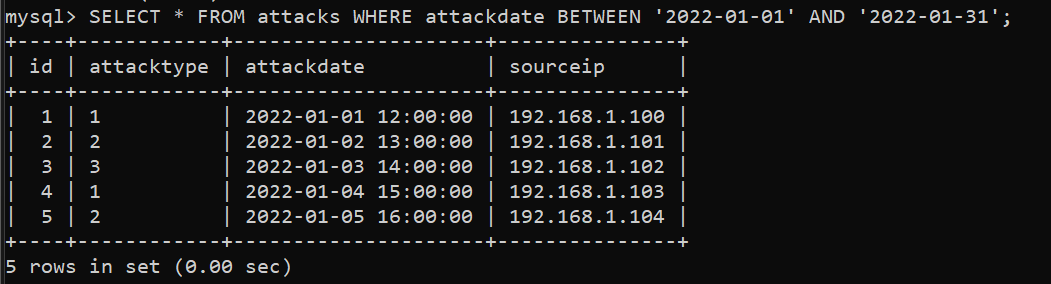


4) Queries identified by the Network Infra security team:

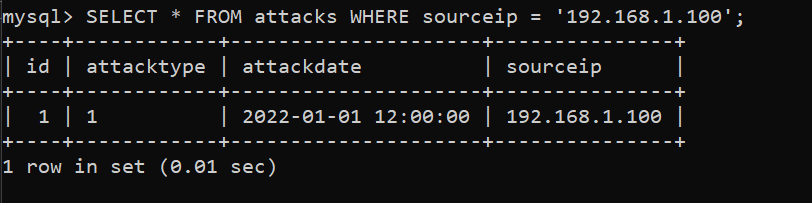
Retrieve all data of attacks from Attacks table:



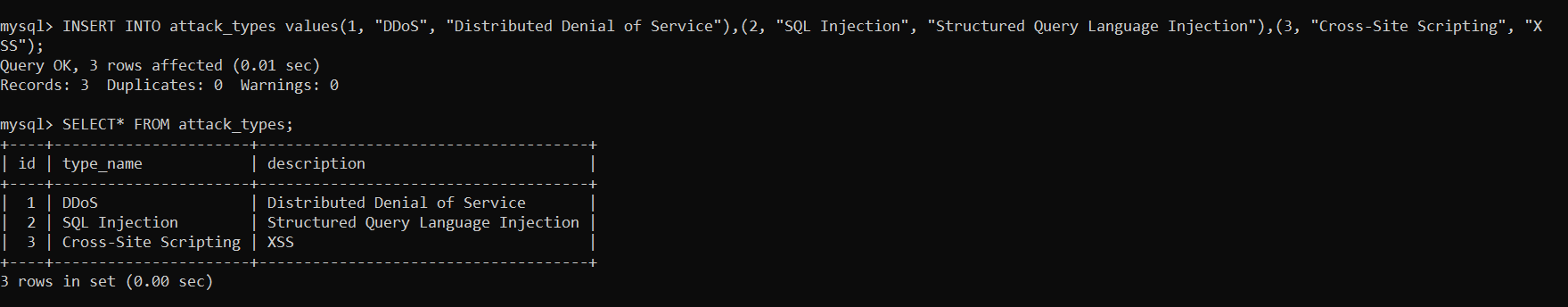
Retrieve attacks by date range from Attacks table:



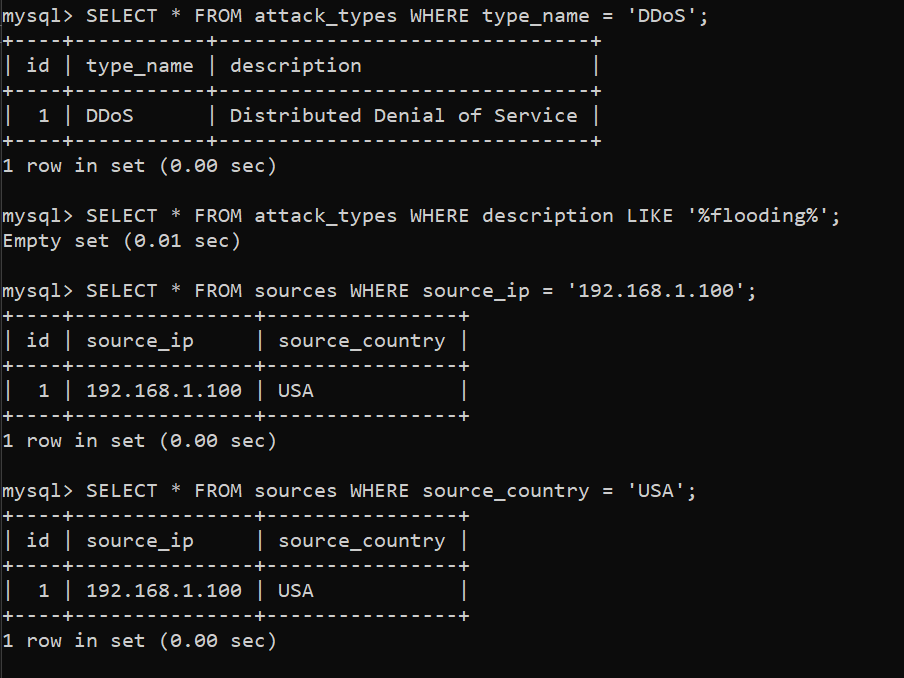
Retrieve attacks by source IP from Attack table:



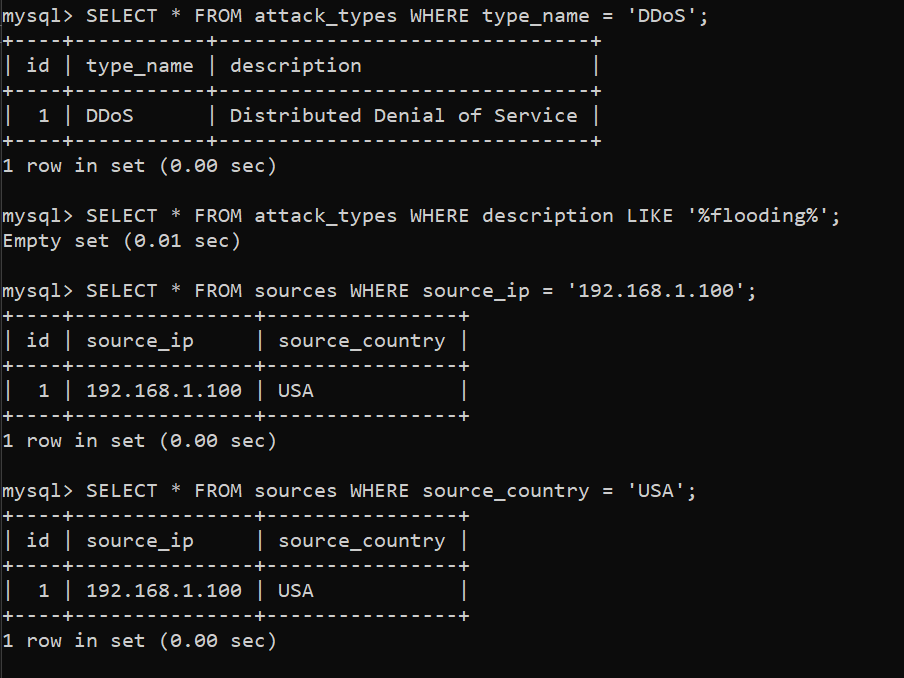
Retrieve all attack types Attack\_types Table:



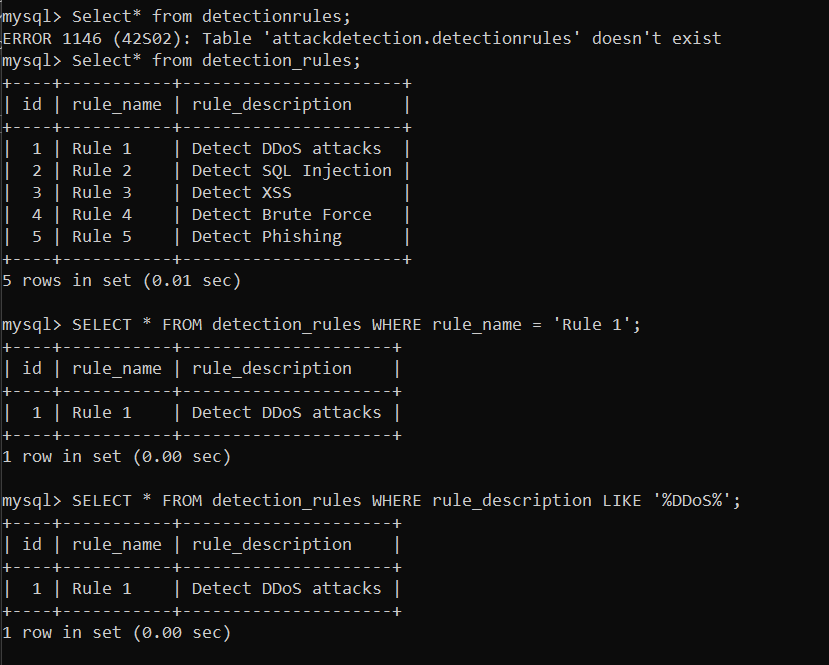
Retrieve all sources by source ip from Sources Table:



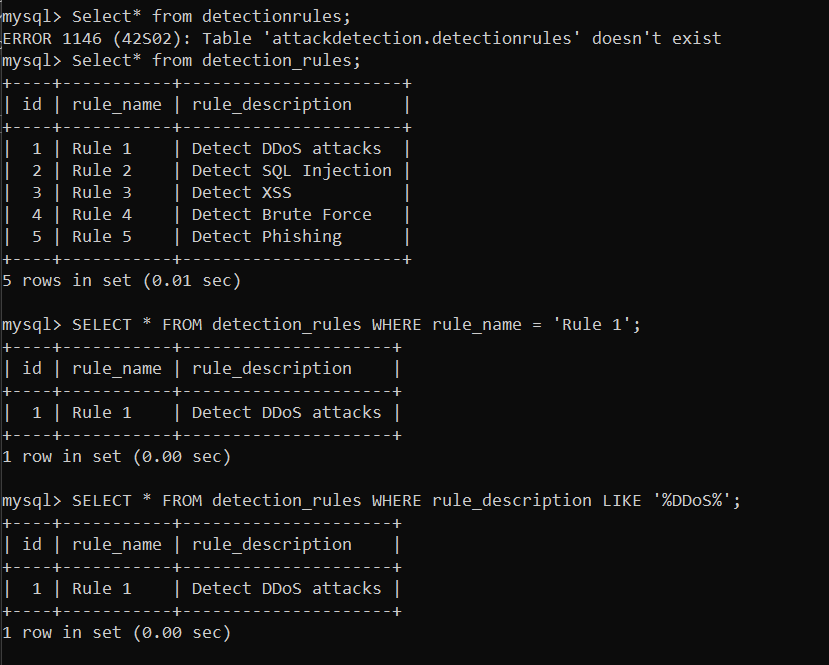
Retrieve all sources by country from Sources Table:



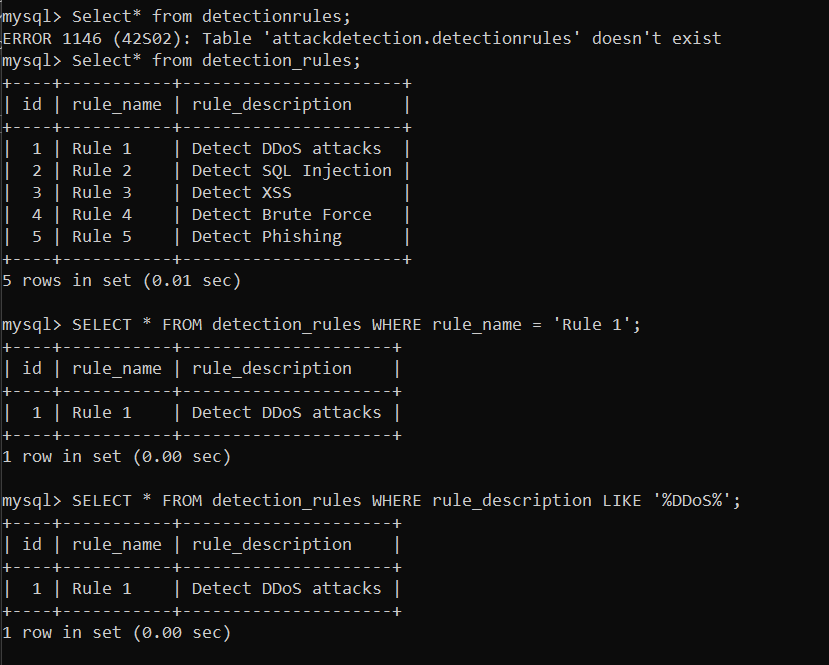
Retrieve all detection rules from Detection rules Table:



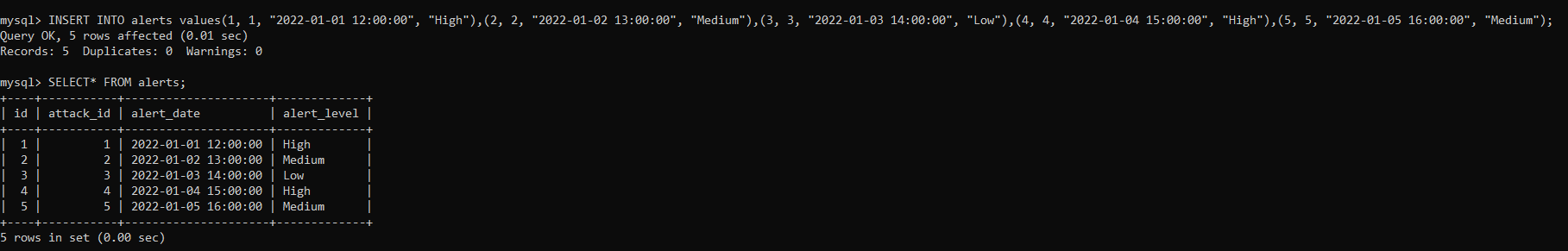
Retrieve all detection rules by name from Detection rules Table:



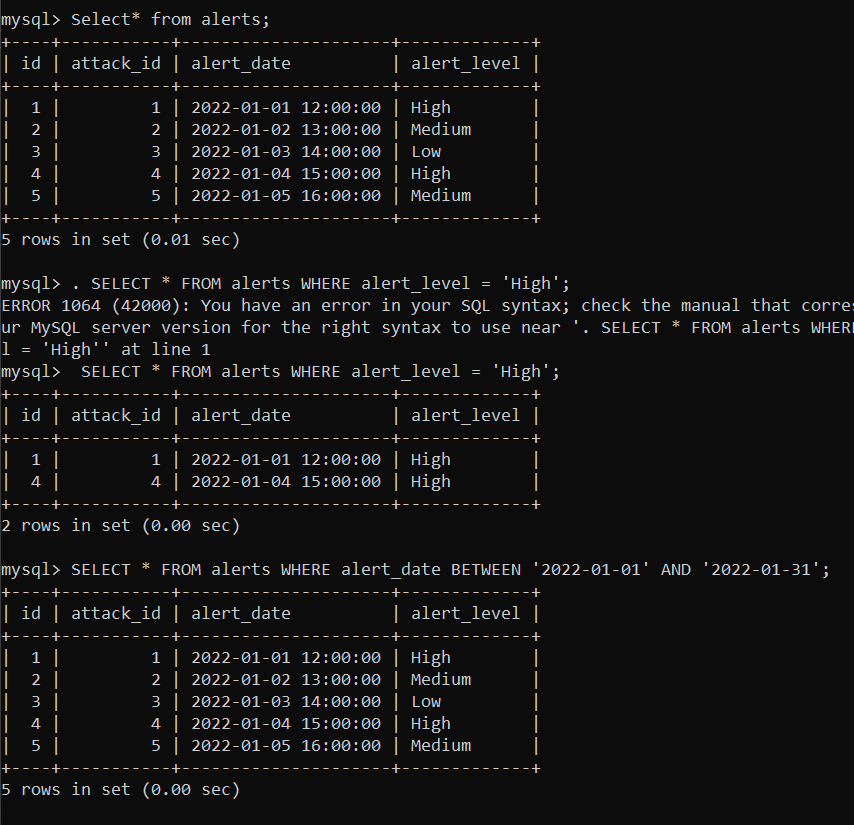
Retrieve all detection rules by description from Detection rules Table:



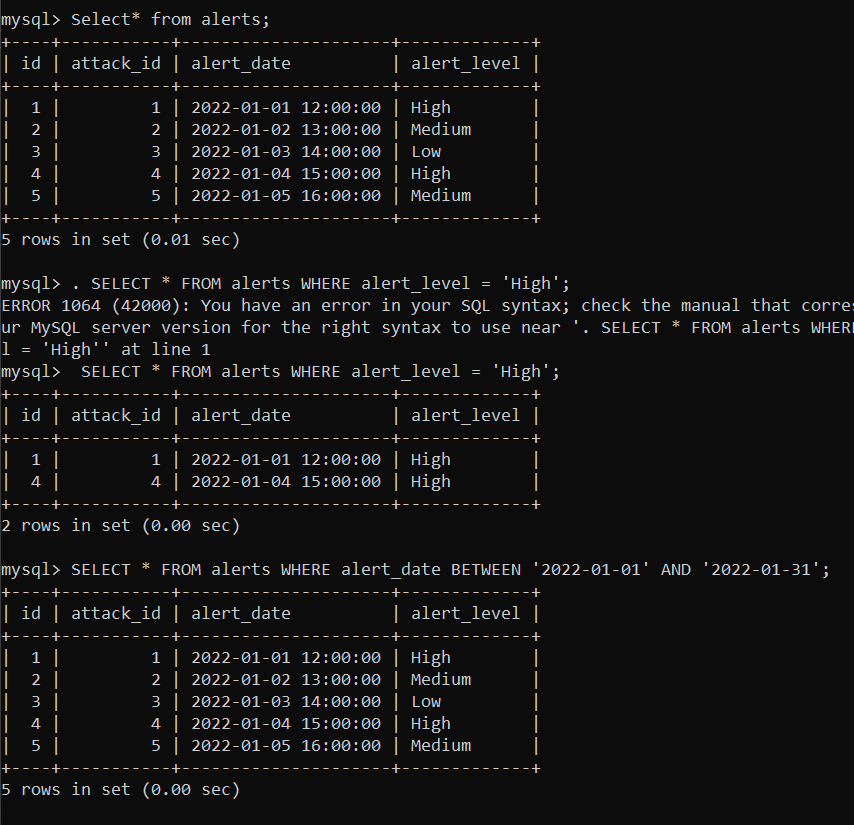
Retrieve all alerts from Alerts Table:



Retrieve alerts by level From Alerts table:



Retrieve alerts by Date from Alerts Table:



**Final Goal of the Project.:**

This project's main goal is to create a flexible, secure, and robust system capable of identify, track, and lessen cyberattacks—especially Denial-of-service attacks. Through leveraging dynamic detection criteria, real-time alerts, and advanced analytics, the system will enhance cybersecurity overall and protect vital assets from constantly changing threats.