

Sri Lanka Institute of Information Technology



University Library Management System

Group Assignment

Submitted by – group 32

Date of submission : October 15, 2024

IE2042 – Database Management Systems for Security.

B.Sc. (Hons) in Information Technology

specializing in Cyber Security.

Contents

1.	Introduction	4
2.	Assumption	5
3.	ER diagram and schema	6
4.	Table create	8
5.	Data insertion.....	22
6.	Triggers	52
7.	Views	54
8.	Indexes	56
9.	Procedures	57
10.	Analysis of 2 database vulnerabilities	59
10.2.	SQL injections.....	59
10.2.	Denial of Service (DoS) attacks.....	64
11.	Mitigation and countermeasure suggestions	67
11.1.	SQL Injection countermeasures :	67
11.2.	Dos Attack	72

Group Assignment - WBS

Database Design, Implementation and Security

	Student ID 1	Student ID 2	Student ID 3	Student ID 4
Task 1				
Properly Documented Assumptions	✓	✓	✓	✓
ERD and Logical Model	✓	✓	✓	✓
Normalization	✓	✓	✓	✓
Table Implementations	✓	✓	✓	✓
Constraints Implementation	✓	✓	✓	✓
2 Triggers		✓		
2 Views	✓			
2 Indexes				✓
2 Stored Procedures			✓	
Task 2				
Description and analysis of 2 database vulnerabilities	✓	✓		
Mitigation and countermeasure suggestions			✓	✓
Total				

IT23406094 JAYASINGHE S.D	IT23262904 LOKUVITHANA N T	IT23265592 RAJAPAKSHA R M P U	IT23285224 BANDARA B. R. K. Y
Member 1	Member 2	Member 3	Member 4

1. Introduction

We analyzed those requirements and created a platform where users can borrow and reserve items who love to explore and expand their knowledge. This system provides a wide range of services. People can also utilize this system to donate items such as books and journals.

A database must be able to manage a wide range of data. The database is required because it is hard to manage data by writing and typing them in a file. Therefore, a computer-based database is required for the system. Users and library staff members can save time and manage data easier with database. Furthermore, since the data is stored in a database, there is no risk of losing it.

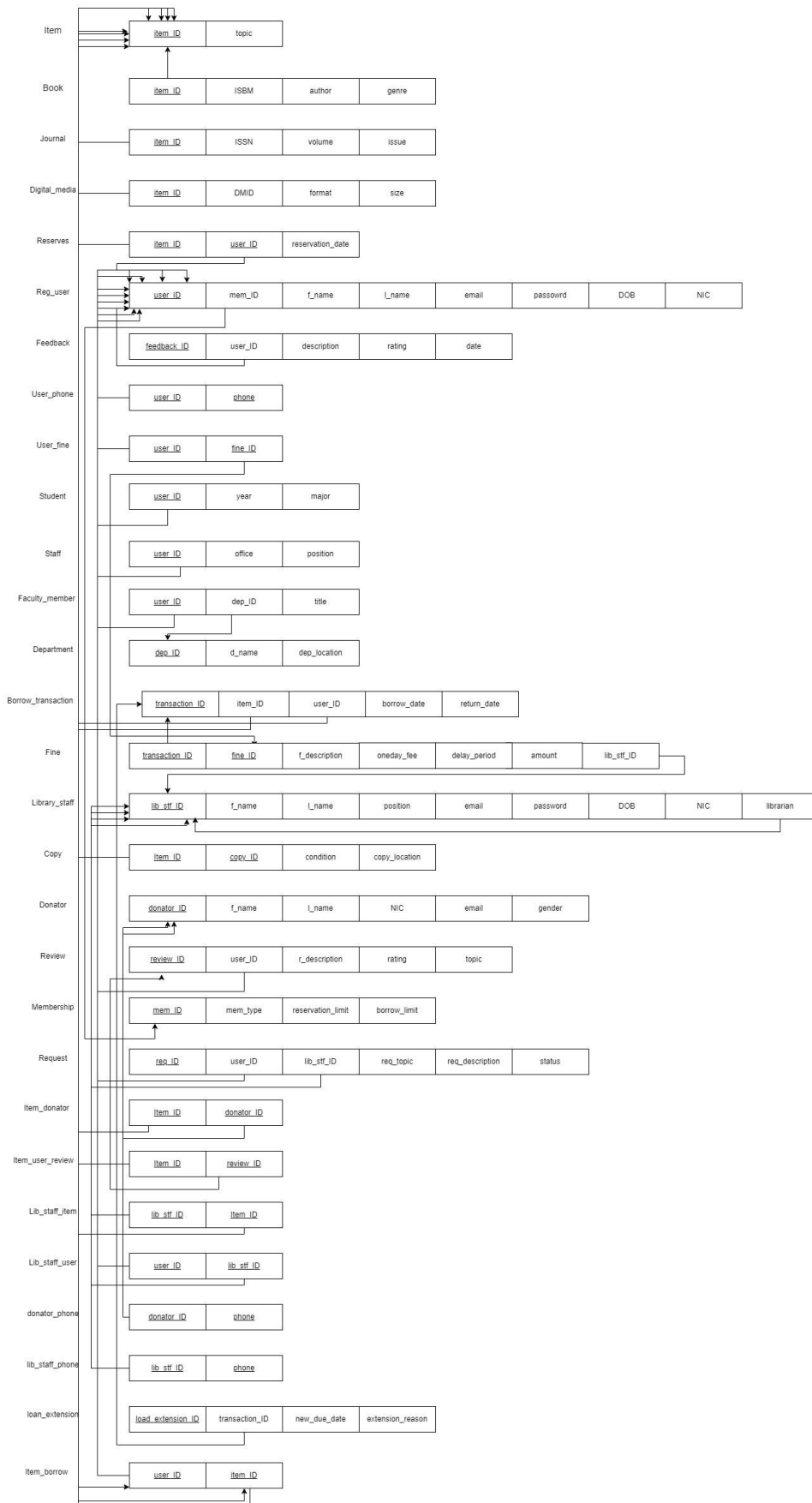
Once all of this was done based on the assumptions that were made SQL queries were written to make the tables that were associated with the database system and demo values were input so that views, triggers, indexes, and procedures could be coded. Triggers were made to check for certain conditions before an insertion to the table takes place. Views were created by finding the user roles within the scenario so that access levels can be controlled. Procedures were then written to fulfill the questions that were asked in the assignment.

In this assignment two main types of database attacks used by threat actors are discussed in brief and they are SQL injection and denial of Service attacks. The way in which these attacks are performed and what the benefit is for the attackers are also listed below.

Other than the attack types of preventive measures that can be taken against these attacks are also mentioned and these security practices can be used by people to make their database stronger and less vulnerable to attacks.

2. Assumption

- Users can get services from this library system by online s well as physically due to that reason this is a virtual library management system.
- A student or a staff member can be a faculty member, so this user entity's sub entities are overlapping.
- Without Item, copy cannot be existing so copy is a weak entity
- The librarian manages other staff members and librarian is also a member of library staff so, it is a recursive entity.
- Donators can donate one or more items and items can be donated by one or more donators.
- In this online library system users are registered users.
- All user must has membership and according to that membership they have reservation and borrowing limit due to their membership type
- If user wants any requests such as Turnitin or any other services they can request through this system
- The amount of fine table is automatically calculated when the "f_description" is "late return" only



4. Table create

table.sql - DESK...QU3JA2\User (51))

```
-- create Department table
CREATE TABLE Department (
    dep_ID CHAR(6) NOT NULL,
    d_name VARCHAR(30) NOT NULL,
    dep_location VARCHAR(30) NOT NULL,

    CONSTRAINT Department_PK PRIMARY KEY (dep_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:26:34.1837099+05:30

```
--create Membership table
CREATE TABLE Membership (
    mem_ID CHAR(6) NOT NULL,
    mem_type VARCHAR(20) NOT NULL,
    reservation_limit INT NOT NULL,
    borrow_limit INT NOT NULL,

    CONSTRAINT Membership_PK PRIMARY KEY (mem_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:29:01.5127047+05:30


```
table.sql - DESK...QU3JA2\User (51))  X
--create Reg_user table
CREATE TABLE Reg_user (
    user_ID CHAR(6) NOT NULL,
    mem_ID CHAR(6) NOT NULL,
    f_name VARCHAR(30) NOT NULL,
    l_name VARCHAR(30) NOT NULL,
    email VARCHAR(60) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL, --allows to store hashed passwords
    DOB date NOT NULL,
    NIC VARCHAR(12) UNIQUE NOT NULL,

    CONSTRAINT User_pk PRIMARY KEY (user_ID),
    CONSTRAINT UserMemID_fk FOREIGN KEY (mem_ID) REFERENCES Membership(mem_ID)
);

146 %
Messages
Commands completed successfully.

Completion time: 2024-10-15T12:29:41.9080756+05:30
```

```
table.sql - DESK...QU3JA2\User (51))  X
--create Library_staff table
CREATE TABLE Library_staff (
    lib_stf_ID CHAR(6) NOT NULL,
    f_name VARCHAR(30) NOT NULL,
    l_name VARCHAR(30) NOT NULL,
    position VARCHAR(20) NOT NULL,
    email VARCHAR(60) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL, --allows to store hashed passwords
    DOB date NOT NULL,
    NIC VARCHAR(12) UNIQUE NOT NULL,
    librarian_ID CHAR(6), -- Recursive foreign key to link a supervisor (can be NULL if no supervisor)

    CONSTRAINT Library_staff_pk PRIMARY KEY (lib_stf_ID),
    CONSTRAINT Library_staff_librarian_fk FOREIGN KEY (librarian_ID) REFERENCES Library_staff(lib_stf_ID)
);

146 %
Messages
Commands completed successfully.

Completion time: 2024-10-15T12:30:22.5293615+05:30
```

```
-- Donator table
```

```
CREATE TABLE Donator (  
    donator_ID CHAR(6) NOT NULL,  
    f_name VARCHAR(30) NOT NULL,  
    l_name VARCHAR(20) NOT NULL,  
    NIC VARCHAR(12) UNIQUE NOT NULL,  
    email VARCHAR(50) UNIQUE NOT NULL,  
    gender VARCHAR(10) NOT NULL,  
  
    CONSTRAINT PK_Donator PRIMARY KEY (donator_ID)  
);
```

146 %

 Messages


Commands completed successfully.

Completion time: 2024-10-15T12:30:55.7264042+05:30

```
--create Item table
```

```
CREATE TABLE Item (  
    item_ID CHAR(6) NOT NULL,  
    title VARCHAR(30) NOT NULL,  
  
    CONSTRAINT Item_pk PRIMARY KEY (item_ID)  
);
```

146 %

 Messages

Commands completed successfully.

Completion time: 2024-10-15T12:31:19.2933026+05:30

```
--create Book table
CREATE TABLE Book (
    item_ID CHAR(6) NOT NULL,
    ISBN CHAR(6) UNIQUE NOT NULL,
    author VARCHAR(30) NOT NULL,
    genre VARCHAR(30) NOT NULL,

    CONSTRAINT Book_pk PRIMARY KEY (item_ID),
    CONSTRAINT Book_FK FOREIGN KEY (item_ID) REFERENCES Item (item_ID)
);
```

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:31:43.6809017+05:30

```
--create Journal table
CREATE TABLE Journal (
    item_ID CHAR(6) NOT NULL,
    ISSN CHAR(6) UNIQUE NOT NULL,
    volume VARCHAR(30) NOT NULL,
    issue VARCHAR(50) NOT NULL,

    CONSTRAINT Journal_pk PRIMARY KEY (item_ID),
    CONSTRAINT Journal_FK FOREIGN KEY (item_ID) REFERENCES Item (item_ID)
);
```

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:32:04.9728652+05:30

```
-- Digital_media table
CREATE TABLE Digital_media (
    item_ID CHAR(6) NOT NULL,
    DMID CHAR(6) UNIQUE NOT NULL,
    format VARCHAR(30) NOT NULL,
    size VARCHAR(10) NOT NULL,

    CONSTRAINT PK_Digital_media PRIMARY KEY (item_ID),
    CONSTRAINT Digital_media_FK FOREIGN KEY (item_ID) REFERENCES Item (item_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:32:33.8234292+05:30

```
-- CREATE Copy table
CREATE TABLE Copy
(
    copy_ID char (6) NOT NULL,
    item_ID char(6) NOT NULL,
    condition varchar(30) NOT NULL,
    copy_location varchar(20) NOT NULL,

    CONSTRAINT Copy_PK PRIMARY KEY (item_ID,copy_ID),
    CONSTRAINT Copy_FK FOREIGN KEY (item_ID) REFERENCES Item (item_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:33:45.5959225+05:30

```
--create Item_borrow table
CREATE TABLE Item_borrow (
    user_ID CHAR(6) NOT NULL,
    item_ID CHAR(6) NOT NULL,

    CONSTRAINT Item_borrow_pk PRIMARY KEY (user_ID, item_ID),
    CONSTRAINT Item_borrowUserID_fk FOREIGN KEY (user_ID) REFERENCES Reg_user(user_ID),
    CONSTRAINT Item_borrowItemID_fk FOREIGN KEY (item_ID) REFERENCES Item(item_ID)
);
```

```
-- Borrow_transaction table
CREATE TABLE Borrow_transaction (
    transaction_ID CHAR(6) NOT NULL,
    item_ID CHAR(6) NOT NULL,
    user_ID CHAR(6) NOT NULL,
    borrow_date DATE DEFAULT GETDATE() NOT NULL,
    return_date DATE NULL,

    CONSTRAINT PK_Transaction PRIMARY KEY (transaction_ID),
    CONSTRAINT TransactionUser_ID_FK FOREIGN KEY (user_ID) REFERENCES Reg_user (user_ID),
    CONSTRAINT TransactionItem_ID_FK FOREIGN KEY (item_ID) REFERENCES Item (item_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:35:08.0657968+05:30

```
--create fine table
CREATE TABLE Fine (
    transaction_ID CHAR(6) NOT NULL,
    fine_ID CHAR (6) NOT NULL,
    f_description VARCHAR(50) NOT NULL,
    oneday_fee MONEY,
    delay_period INT,
    amount MONEY,
    lib_stf_ID CHAR(6) NOT NULL,

    CONSTRAINT Fine_PK PRIMARY KEY (transaction_ID,fine_ID),
    CONSTRAINT FineTransactionID_FK FOREIGN KEY (transaction_ID) REFERENCES Borrow_transaction(transaction_ID),
    CONSTRAINT FineLib_stf_ID_FK FOREIGN KEY (lib_stf_ID) REFERENCES Library_staff(lib_stf_ID),
    CONSTRAINT Fine_fine_ID_UNIQUE UNIQUE (fine_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:35:32.5300562+05:30

```
--create User_fine table
CREATE TABLE User_fine (
    user_ID CHAR(6) NOT NULL,
    fine_ID CHAR(6) NOT NULL,

    CONSTRAINT User_fine_pk PRIMARY KEY (user_ID, fine_ID),
    CONSTRAINT User_fineUserID_fk FOREIGN KEY (user_ID) REFERENCES Reg_user(user_ID) ,
    CONSTRAINT User_fineFineID_fk FOREIGN KEY (fine_ID) REFERENCES Fine(fine_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:36:38.2067519+05:30

```
--create User_phone table
CREATE TABLE User_phone (
    user_ID CHAR(6) NOT NULL,
    phone CHAR(10) NOT NULL,

    CONSTRAINT User_phone_pk PRIMARY KEY (user_ID, phone),
    CONSTRAINT User_phoneUserID_fk FOREIGN KEY (user_ID) REFERENCES Reg_user(user_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:37:03.9827137+05:30

```
--create reserves table
CREATE TABLE User_item (
    item_ID CHAR(6) NOT NULL,
    user_ID CHAR(6) NOT NULL,
    reservation_date DATE NOT NULL,

    CONSTRAINT User_item_pk PRIMARY KEY (item_ID, user_ID),
    CONSTRAINT User_itemItemID_fk FOREIGN KEY (item_ID) REFERENCES Item(item_ID),
    CONSTRAINT User_itemUserID_fk FOREIGN KEY (user_ID) REFERENCES Reg_user(user_ID)
);
```

6 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:37:38.6073725+05:30

```
--create feedback table
CREATE TABLE feedback (
    feedback_ID CHAR(6) NOT NULL,
    user_ID CHAR(6) NOT NULL,
    description VARCHAR(50) NOT NULL,
    rating CHAR(1) CHECK (rating IN ('1', '2', '3', '4', '5')) NOT NULL,
    date DATE NOT NULL,

    CONSTRAINT feedback_pk PRIMARY KEY (feedback_ID),
    CONSTRAINT feedbackUserID_fk FOREIGN KEY (user_ID) REFERENCES Reg_user(user_ID)
);
```

6 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:38:01.8814322+05:30

```
--create student table
CREATE TABLE student (
  user_ID CHAR(6) NOT NULL,
  year CHAR(1) CHECK (year IN ('1', '2', '3', '4')) NOT NULL,
  major VARCHAR(30) NOT NULL,

  CONSTRAINT student_pk PRIMARY KEY (user_ID),
  CONSTRAINT studentUserID_fk FOREIGN KEY (user_ID) REFERENCES Reg_user(user_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:39:09.3441689+05:30

```
-- Faculty_member table
CREATE TABLE Faculty_member(
  user_ID CHAR(6) NOT NULL,
  Dep_ID CHAR(6) NOT NULL,
  title VARCHAR(20) NOT NULL,

  CONSTRAINT PK_Faculty_member PRIMARY KEY (user_ID),
  CONSTRAINT Faculty_memberuser_ID_FK FOREIGN KEY(user_ID) REFERENCES Reg_user(user_ID),
  CONSTRAINT Faculty_memberDep_ID_FK FOREIGN KEY(Dep_ID) REFERENCES Department(Dep_ID)
);
```

16 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:39:27.0434435+05:30


```
-- Staff table
CREATE TABLE Staff (
    user_ID char(6) NOT NULL,
    office varchar(20) NOT NULL,
    position varchar(20) NOT NULL,
    CONSTRAINT Staff_PK PRIMARY KEY (user_ID),
    CONSTRAINT Staff_FK FOREIGN KEY (user_ID) REFERENCES Reg_user (user_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:39:59.2829385+05:30

```
--create donator_phone table
CREATE TABLE donator_phone (
    donator_ID CHAR(6) NOT NULL,
    phone CHAR(10) NOT NULL,
    CONSTRAINT donator_phone_pk PRIMARY KEY (donator_ID, phone),
    CONSTRAINT donator_phoneDonator_ID_FK FOREIGN KEY (donator_ID) REFERENCES Donator (donator_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:40:19.6423721+05:30

```
-- Item_review table
CREATE TABLE Item_review
(
    review_ID char (6) NOT NULL,
    item_ID char(6) NOT NULL,

    CONSTRAINT Item_review_PK PRIMARY KEY (review_ID,item_ID),
    CONSTRAINT Item_review_review_ID_FK FOREIGN KEY (review_ID) REFERENCES Review (review_ID),
    CONSTRAINT Item_review_item_ID_FK FOREIGN KEY (item_ID) REFERENCES Item (item_ID)
);

-- Loan extension table
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:43:07.6670138+05:30

```
-- Item_review table
CREATE TABLE Item_review
(
    review_ID char (6) NOT NULL,
    item_ID char(6) NOT NULL,

    CONSTRAINT Item_review_PK PRIMARY KEY (review_ID,item_ID),
    CONSTRAINT Item_review_review_ID_FK FOREIGN KEY (review_ID) REFERENCES Review (review_ID),
    CONSTRAINT Item_review_item_ID_FK FOREIGN KEY (item_ID) REFERENCES Item (item_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:41:09.0077362+05:30

```
-- Loan_extension table
CREATE TABLE Loan_extension
(
    loan_extension_ID char (6) NOT NULL,
    transaction_ID char (6) NOT NULL,
    new_due_date date NOT NULL,
    extension_reason varchar (50) NOT NULL,

    CONSTRAINT Loan_extension_PK PRIMARY KEY (loan_extension_ID),
    CONSTRAINT Loan_extension_FK FOREIGN KEY (transaction_ID) REFERENCES Borrow_transaction (transaction_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:41:24.8735172+05:30

```
-- lib_staff_user table
CREATE TABLE lib_staff_user (
    user_ID CHAR(6) NOT NULL,
    lib_stf_ID CHAR(6) NOT NULL,

    CONSTRAINT PK_lib_staff_user PRIMARY KEY (lib_stf_ID, user_ID),
    CONSTRAINT lib_staff_user_ID_fk FOREIGN KEY(user_ID) REFERENCES Reg_user(user_ID),
    CONSTRAINT lib_staff_userlib_stf_ID_FK FOREIGN KEY(lib_stf_ID) REFERENCES Library_staff(lib_stf_ID)
);
```

16 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:41:43.9827551+05:30

```
-- Request table
CREATE TABLE Request (
  req_ID CHAR(6) NOT NULL,
  user_ID CHAR(6) NOT NULL,
  lib_stf_ID CHAR(6) NOT NULL,
  req_topic VARCHAR(30) NOT NULL,
  req_description VARCHAR(50) NOT NULL,
  status VARCHAR(10) NOT NULL,

  CONSTRAINT PK_Request PRIMARY KEY (req_ID),
  CONSTRAINT RequestUser_ID_FK FOREIGN KEY(user_ID) REFERENCES Reg_user(user_ID),
  CONSTRAINT RequestLib_stf_ID_FK FOREIGN KEY(lib_stf_ID) REFERENCES Library_staff(lib_stf_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:42:03.3179956+05:30

```
--create Lib_staff_item table
CREATE TABLE Lib_staff_item
(
  item_ID char (6) NOT NULL,
  lib_stf_ID char(6) NOT NULL,

  CONSTRAINT Lib_staff_item_PK PRIMARY KEY (item_ID,lib_stf_ID),
  CONSTRAINT Lib_staff_item_ID_FK FOREIGN KEY (item_ID) REFERENCES Item (item_ID),
  CONSTRAINT Lib_staff_item_lib_stf_ID_FK FOREIGN KEY (lib_stf_ID) REFERENCES Library_staff (lib_stf_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:42:26.1962040+05:30

```
--create Lib_stf_phone table
CREATE TABLE Lib_stf_phone (
    lib_stf_ID CHAR(6) NOT NULL,
    phone VARCHAR(10) NOT NULL,

    CONSTRAINT Lib_stf_phone_PK PRIMARY KEY (lib_stf_ID,phone),
    CONSTRAINT Lib_stf_phoneLib_stf_ID_FK FOREIGN KEY (lib_stf_ID) REFERENCES Library_staff (lib_stf_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:42:50.2464076+05:30

```
--create Item_donator table
CREATE TABLE Item_donator (
    donator_ID CHAR(6) NOT NULL,
    item_ID char(6) NOT NULL,

    CONSTRAINT Item_donator_PK PRIMARY KEY (item_ID,donator_ID),
    CONSTRAINT Item_donatorItem_ID_FK FOREIGN KEY (item_ID) REFERENCES Item (item_ID),
    CONSTRAINT Item_donatorDonator_ID_FK FOREIGN KEY (donator_ID) REFERENCES Donator (donator_ID)
);
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T12:43:07.6670138+05:30

5. Data insertion

```
insert.sql - DESK...QU3JA2\User (61))  × table.sql - DESK...QU3JA2\User (51))*  
- INSERT INTO Department (dep_ID, d_name, dep_location) VALUES  
  ('DEP001', 'Computing', 'Building A');  
- INSERT INTO Department (dep_ID, d_name, dep_location) VALUES  
  ('DEP002', 'Business', 'Building B');  
- INSERT INTO Department (dep_ID, d_name, dep_location) VALUES  
  ('DEP003', 'Engineering', 'Building C');  
- INSERT INTO Department (dep_ID, d_name, dep_location) VALUES  
  ('DEP004', 'Law', 'Building D');  
- INSERT INTO Department (dep_ID, d_name, dep_location) VALUES  
  ('DEP005', 'Hotel management', 'Building E');  
146 %  
Messages  
  
(1 row affected)  
  
(1 row affected)  
  
(1 row affected)  
  
(1 row affected)  
  
(1 row affected)  
  
Completion time: 2024-10-15T12:45:38.7315782+05:30
```

insert.sql - DESK...QU3JA2\User (61)) × table.sql - DESK...QU3JA2\User (51))*

```
INSERT INTO Membership (mem_ID, mem_type, reservation_limit, borrow_limit) VALUES
('M00001', 'Student', 3, 3);

INSERT INTO Membership (mem_ID, mem_type, reservation_limit, borrow_limit) VALUES
('M00002', 'Assistant Lecturer', 5, 5);

INSERT INTO Membership (mem_ID, mem_type, reservation_limit, borrow_limit) VALUES
('M00003', 'Alumni', 5, 5);

INSERT INTO Membership (mem_ID, mem_type, reservation_limit, borrow_limit) VALUES
('M00004', 'LIC', 6, 6);

INSERT INTO Membership (mem_ID, mem_type, reservation_limit, borrow_limit) VALUES
('M00005', 'Lecturer', 5, 5);
```

146 %

Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T12:46:21.7160962+05:30

```

INSERT INTO Reg_user (user_ID, mem_ID, f_name, l_name, email, password, DOB, NIC) VALUES
('IT3701', 'M00001', 'Sanduni', 'Rajapaksha', 'sanduni1@gmail.com', 'tkkTsx9202HDGT@%ffg', '2003-02-15', '200356104307');

INSERT INTO Reg_user (user_ID, mem_ID, f_name, l_name, email, password, DOB, NIC) VALUES
('IT7062', 'M00001', 'Krishna', 'Silva', 'krish111@gmail.com', 'krtFn$389fVggjt573%*9fjdjs', '2000-08-10', '200034600121');

INSERT INTO Reg_user (user_ID, mem_ID, f_name, l_name, email, password, DOB, NIC) VALUES
('L90701', 'M00002', 'Samanthi', 'Lokuvithana', 'samanthi88@gmail.com', 'lxct%2SL!!99ghl547', '1998-02-15', '981234567V');

INSERT INTO Reg_user (user_ID, mem_ID, f_name, l_name, email, password, DOB, NIC) VALUES
('L07062', 'M00004', 'Aruni', 'Dissanayake', 'arunD@gmail.com', 'krtFn$389fVggjt573%*9fjdjs', '1985-08-10', '851234567V');

INSERT INTO Reg_user (user_ID, mem_ID, f_name, l_name, email, password, DOB, NIC) VALUES
('A07503', 'M00003', 'Amal', 'Danujaya', 'amaldanujaya99@gmail.com', 'jaIntHcbTr@^37H3222', '1999-04-21', '992345678V');

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T12:47:49.5088639+05:30

```

```

insert.sql - DESK_QU3JA2\User (61)  table.sql - DESK_QU3JA2\User (51)*

INSERT INTO Reg_user (user_ID, mem_ID, f_name, l_name, email, password, DOB, NIC) VALUES
('EN8054', 'M00001', 'Kaveesha', 'Gimhani', 'kaveeshagimhani2003@gmail.com', 'npnfb6D912J%3W8$', '2002-12-11', '200259266102');

INSERT INTO Reg_user (user_ID, mem_ID, f_name, l_name, email, password, DOB, NIC) VALUES
('BM7503', 'M00001', 'Nadun', 'Jayasinghe', 'Nadun.j@gmail.com', 'securePass!123', '1999-06-15', '990123456V');

INSERT INTO Reg_user (user_ID, mem_ID, f_name, l_name, email, password, DOB, NIC) VALUES
('HM0701', 'M00001', 'Jane', 'Dias', 'jane.dias@gmail.com', 'myPass@2024', '1999-09-20', '990145456V');

INSERT INTO Reg_user (user_ID, mem_ID, f_name, l_name, email, password, DOB, NIC) VALUES
('L77503', 'M00002', 'Nimal', 'Perera', 'nimal.perera@gmail.com', 'pass1234', '1988-03-12', '880123456V');

INSERT INTO Reg_user (user_ID, mem_ID, f_name, l_name, email, password, DOB, NIC) VALUES
('L94753', 'M00002', 'Anu', 'Fernando', 'anu.fernando@gmail.com', 'mysecurepass!12', '1996-05-08', '960123456V');

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T12:48:56.4435423+05:30

```



```

insert.sql - DESK...QU3JA2\User (611)  X table.sql - DESK...QU3JA2\User (511)*
INSERT INTO Library_staff (lib_stf_ID, f_name, l_name, position, email, password, DOB, NIC, librarian_ID)
VALUES ('LS0101', 'Ama', 'Devindi', 'Librarian', 'Ama@gmail.com', 'daltmvWLE5832)$33', '1980-01-22', '801234567V', NULL);

INSERT INTO Library_staff (lib_stf_ID, f_name, l_name, position, email, password, DOB, NIC, librarian_ID) VALUES
('LS8902', 'Avindi', 'Abeyrathna', 'Assistant', 'Avindi@gmail.com', '4cnrj485uajGBf$@RT', '1989-07-13', '891234567V', 'LS0101');

INSERT INTO Library_staff (lib_stf_ID, f_name, l_name, position, email, password, DOB, NIC, librarian_ID) VALUES
('LS7113', 'Kaushi', 'Rathnayake', 'Assistant', 'Kaushi@gmail.com', 'cjr%38T55chv', '1995-10-05', '951234567V', 'LS0101');

INSERT INTO Library_staff (lib_stf_ID, f_name, l_name, position, email, password, DOB, NIC, librarian_ID) VALUES
('LS8734', 'Sanduni', 'Wickramasinghe', 'Technician', 'Sanduni@gmail.com', 'krj$@DHMY@%5', '1992-09-30', '921234567V', 'LS0101');

INSERT INTO Library_staff (lib_stf_ID, f_name, l_name, position, email, password, DOB, NIC, librarian_ID) VALUES
('LS1285', 'Nadun', 'Ranasinghe', 'Clerk', 'Nadun@gmail.com', 'fjr48D6djb', '1990-03-25', '901234567V', 'LS0101');

select * from library_staff

146 %
Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T12:49:31.8843679+05:30
  
```

```

INSERT INTO Donator (donator_ID, f_name, l_name, NIC, email, gender) VALUES
('D73501', 'Kavindu', 'Weerasinghe', '861234567V', 'Kavindu58@gmail.com', 'Male');

INSERT INTO Donator (donator_ID, f_name, l_name, NIC, email, gender) VALUES
('D88502', 'Dineth', 'Rathnayake', '981234567V', 'DinethR7@gmail.com', 'Male');

INSERT INTO Donator (donator_ID, f_name, l_name, NIC, email, gender) VALUES
('D82473', 'Pasindu', 'Mendis', '951234567V', 'PasinduM23@gmail.com', 'Male');

INSERT INTO Donator (donator_ID, f_name, l_name, NIC, email, gender) VALUES
('D91634', 'Heshani', 'vithanage', '771234567V', 'HeshaniV2097@gmail.com', 'Female');

INSERT INTO Donator (donator_ID, f_name, l_name, NIC, email, gender) VALUES
('D09205', 'Kaveesha', 'Devinda', '921234567V', 'Kaveesha23@gmail.com', 'Male');

146 %
Messages

(1 row affected)

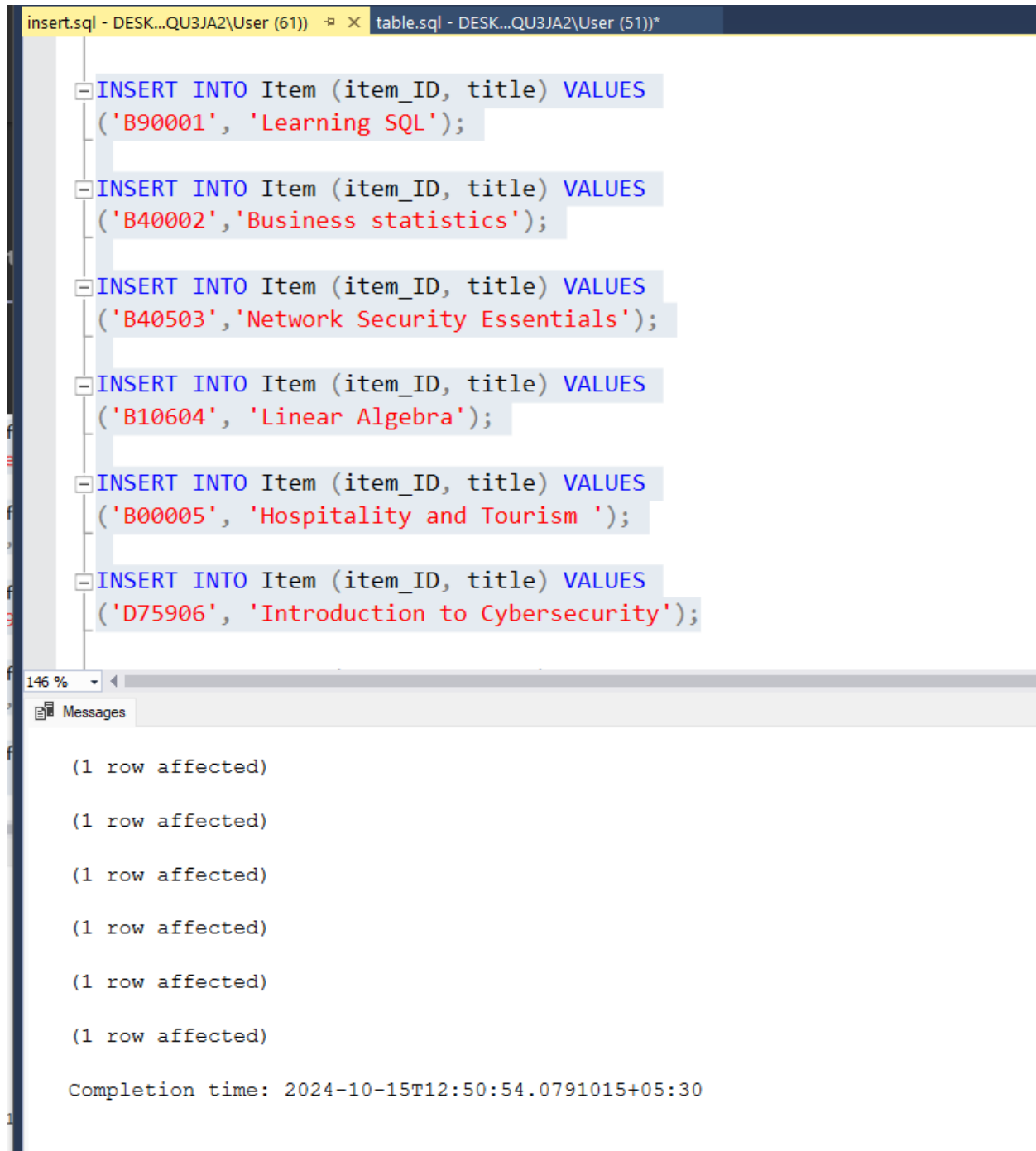
(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T12:50:20.6196885+05:30
  
```



```
insert.sql - DESK...QU3JA2\User (61)) × table.sql - DESK...QU3JA2\User (51))*  
  
INSERT INTO Item (item_ID, title) VALUES  
('B90001', 'Learning SQL');  
  
INSERT INTO Item (item_ID, title) VALUES  
('B40002', 'Business statistics');  
  
INSERT INTO Item (item_ID, title) VALUES  
('B40503', 'Network Security Essentials');  
  
INSERT INTO Item (item_ID, title) VALUES  
('B10604', 'Linear Algebra');  
  
INSERT INTO Item (item_ID, title) VALUES  
('B00005', 'Hospitality and Tourism ');  
  
INSERT INTO Item (item_ID, title) VALUES  
('D75906', 'Introduction to Cybersecurity');
```

146 %

Messages


(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)

Completion time: 2024-10-15T12:50:54.0791015+05:30

insert.sql - DESK...QU3JA2\User (61)) × table.sql - DESK...QU3JA2\User (51))*

```
- INSERT INTO Item (item_ID, title) VALUES  
  ('D00007', 'Hotel Management Essential');  
  
- INSERT INTO Item (item_ID, title) VALUES  
  ('J84908', 'Hospitality & Tourism Research');  
  
- INSERT INTO Item (item_ID, title) VALUES  
  ('J25309', 'Journal of Network Security');  
  
- INSERT INTO Item (item_ID, title) VALUES  
  ('J70010', 'Journal of Business Analytics');  
  
- INSERT INTO Item (item_ID, title) VALUES  
  ('J80101', 'Applied Linear Algebra');  
  
- INSERT INTO Item (item_ID, title) VALUES  
  ('D50694', 'SQL for Data Analysis');
```

146 %

 Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T12:51:18.9290620+05:30

```
- INSERT INTO Item (item_ID, title) VALUES  
  ('D90756', 'Introduction to Algebra Video');  
- INSERT INTO Item (item_ID, title) VALUES  
  ('D80555', 'Business Statistics Course');  
- INSERT INTO Item (item_ID, title) VALUES  
  ('J58282', 'Computational Mathematics');
```

146 %

 Messages

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T12:51:45.4089683+05:30

```
- INSERT INTO Book (item_ID, ISBN, author, genre) VALUES  
  ('B90001', '123456', 'Alan Beaulieu', 'Computing');  
- INSERT INTO Book (item_ID, ISBN, author, genre) VALUES  
  ('B40002', '654321', 'George Orwell', 'Business');  
- INSERT INTO Book (item_ID, ISBN, author, genre) VALUES  
  ('B40503', '789012', 'William Stallings', 'Technology');  
- INSERT INTO Book (item_ID, ISBN, author, genre) VALUES  
  ('B10604', '890123', 'David Lay', 'Mathematics');  
- INSERT INTO Book (item_ID, ISBN, author, genre) VALUES  
  ('B00005', '345678', 'Harper Lee', 'Hospitality');
```

146 %

Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T12:52:10.7885848+05:30

```
INSERT INTO Journal (item_ID, ISSN, volume, issue) VALUES
('J84908', '112233', 'Vol. 10', 'Issue 2');

INSERT INTO Journal (item_ID, ISSN, volume, issue) VALUES
('J25309', '223344', 'Vol. 5', 'Issue 1');

INSERT INTO Journal (item_ID, ISSN, volume, issue) VALUES
('J70010', '334455', 'Vol. 3', 'Issue 4');

INSERT INTO Journal (item_ID, ISSN, volume, issue) VALUES
('J80101', '337455', 'Vol. 3', 'Issue 4');

INSERT INTO Journal (item_ID, ISSN, volume, issue) VALUES
('J58282', '445566', 'Vol. 7', 'Issue 3');
```

146 %

 Messages

(1 row affected)

(1 row affected)

(1 row affected)


(1 row affected)

(1 row affected)

Completion time: 2024-10-15T12:52:47.4504779+05:30

```
- INSERT INTO Digital_media (item_ID, DMID, format, size) VALUES  
  ('D75906', 'DM001', 'MP4', '10GB');  
- INSERT INTO Digital_media (item_ID, DMID, format, size) VALUES  
  ('D00007', 'DM002', 'MP4', '8GB');  
- INSERT INTO Digital_media (item_ID, DMID, format, size) VALUES  
  ('D50694', 'DM003', 'MP4', '2600MB');  
- INSERT INTO Digital_media (item_ID, DMID, format, size) VALUES  
  ('D90756', 'DM004', 'MP3', '900MB');  
- INSERT INTO Digital_media (item_ID, DMID, format, size) VALUES  
  ('D80555', 'DM005', 'AVI', '700MB');
```

146 %

 Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T12:53:12.7735132+05:30

```
- INSERT INTO Copy (copy_ID, item_ID, condition, copy_location) VALUES  
('C87501', 'J58282', 'New', 'IEE xplore');  
- INSERT INTO Copy (copy_ID, item_ID, condition, copy_location) VALUES  
('C39662', 'D75906', 'New', 'One drive');  
- INSERT INTO Copy (copy_ID, item_ID, condition, copy_location) VALUES  
('C92563', 'B90001', 'New', 'Shelf C');  
- INSERT INTO Copy (copy_ID, item_ID, condition, copy_location) VALUES  
('C84154', 'J80101', 'New', 'IEE xplore');  
- INSERT INTO Copy (copy_ID, item_ID, condition, copy_location) VALUES  
('C93575', 'B40002', 'Damaged', 'Shelf E');
```

146 %

 Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T12:53:38.5332599+05:30


```
- INSERT INTO Item_borrow (user_ID, item_ID) VALUES  
  ('IT3701', 'J70010');  
- INSERT INTO Item_borrow (user_ID, item_ID) VALUES  
  ('L90701', 'B10604');  
- INSERT INTO Item_borrow (user_ID, item_ID) VALUES  
  ('L07062', 'D80555');  
- INSERT INTO Item_borrow (user_ID, item_ID) VALUES  
  ('A07503', 'B10604');  
- INSERT INTO Item_borrow (user_ID, item_ID) VALUES  
  ('EN8054', 'J84908');
```

6 %

Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T12:54:49.7019263+05:30

```
INSERT INTO Borrow_transaction (transaction_ID, item_ID, user_ID, borrow_date) VALUES  
( 'T73451', 'B90001', 'IT3701', '2024-09-01');  
  
INSERT INTO Borrow_transaction (transaction_ID, item_ID, user_ID, borrow_date) VALUES  
( 'T08302', 'B40002', 'L90701', '2024-10-05');  
  
INSERT INTO Borrow_transaction (transaction_ID, item_ID, user_ID, borrow_date) VALUES  
( 'T86323', 'B40503', 'L07062', '2024-05-07');  
  
INSERT INTO Borrow_transaction (transaction_ID, item_ID, user_ID, borrow_date) VALUES  
( 'T94254', 'J25309', 'A07503', '2024-06-10');  
  
INSERT INTO Borrow_transaction (transaction_ID, item_ID, user_ID, borrow_date) VALUES  
( 'T00225', 'B10604', 'EN8054', '2024-03-12');
```

146 %

Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T12:55:31.0278456+05:30

```
INSERT INTO Fine (transaction_ID, fine_ID, f_description, oneday_fee, delay_period, lib_stf_ID) VALUES ('T73451', 'F58371', 'Late return', 100.00, 4, 'LS8902');  
INSERT INTO Fine (transaction_ID, fine_ID, f_description, amount, lib_stf_ID) VALUES ('T08302', 'F48552', 'Book damaged', 5500.00, 'LS7113');  
INSERT INTO Fine (transaction_ID, fine_ID, f_description, amount, lib_stf_ID) VALUES ('T86323', 'F39453', 'Lost item', 10000.00, 'LS0101');  
INSERT INTO Fine (transaction_ID, fine_ID, f_description, oneday_fee, delay_period, lib_stf_ID) VALUES ('T08302', 'F48592', 'Late return', 100.00, 5, 'LS7113');  
INSERT INTO Fine (transaction_ID, fine_ID, f_description, amount, lib_stf_ID) VALUES ('T00225', 'F55555', 'Missing page', 1500.00, 'LS8902');
```

146 %

Messages

(1 row affected)

(1 row affected)

(1 row affected)


(1 row affected)

(1 row affected)

Completion time: 2024-10-15T12:59:13.8660411+05:30

```
INSERT INTO User_fine (user_ID, fine_ID) VALUES  
( 'IT3701', 'F58371' );  
  
INSERT INTO User_fine (user_ID, fine_ID) VALUES  
( 'L90701', 'F48552' );  
  
INSERT INTO User_fine (user_ID, fine_ID) VALUES  
( 'L07062', 'F39453' );  
  
INSERT INTO User_fine (user_ID, fine_ID) VALUES  
( 'A07503', 'F48592' );  
  
INSERT INTO User_fine (user_ID, fine_ID) VALUES  
( 'EN8054', 'F55555' );
```

146 %

 Messages

(1 row affected)

(1 row affected)

(1 row affected)


(1 row affected)

(1 row affected)

Completion time: 2024-10-15T13:00:55.7197448+05:30

```
INSERT INTO User_phone (user_ID, phone) VALUES  
( 'IT3701', '0711234567' );  
  
INSERT INTO User_phone (user_ID, phone) VALUES  
( 'L90701', '0721234567' );  
  
INSERT INTO User_phone (user_ID, phone) VALUES  
( 'L07062', '0777654321' );  
  
INSERT INTO User_phone (user_ID, phone) VALUES  
( 'A07503', '0769988777' );  
  
INSERT INTO User_phone (user_ID, phone) VALUES  
( 'EN8054', '0701122334' );  
  
select * from User_phone
```

146 %

 Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T13:00:55.7197448+05:30

```
- INSERT INTO User_item (item_ID, user_ID, reservation_date) VALUES  
('B10604', 'IT3701', '2024-10-01');  
  
- INSERT INTO User_item (item_ID, user_ID, reservation_date) VALUES  
('D75906', 'L90701', '2024-09-20');  
  
- INSERT INTO User_item (item_ID, user_ID, reservation_date) VALUES  
('J70010', 'L07062', '2024-08-15');  
  
- INSERT INTO User_item (item_ID, user_ID, reservation_date) VALUES  
('B40503', 'A07503', '2024-07-10');  
  
- INSERT INTO User_item (item_ID, user_ID, reservation_date) VALUES  
('D00007', 'EN8054', '2024-06-25');
```

essages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T13:02:15.6886009+05:30

```
[- INSERT INTO feedback (feedback_ID, user_ID, description, rating, date) VALUES
  ('FB001', 'IT3701', 'Excellent service', '5', '2024-10-10');
[- INSERT INTO feedback (feedback_ID, user_ID, description, rating, date) VALUES
  ('FB002', 'L90701', 'Good collection of books', '4', '2024-09-15');
[- INSERT INTO feedback (feedback_ID, user_ID, description, rating, date) VALUES
  ('FB003', 'L07062', 'Helpful staff', '5', '2024-08-20');
[- INSERT INTO feedback (feedback_ID, user_ID, description, rating, date) VALUES
  ('FB004', 'A07503', 'More digital media needed', '3', '2024-07-18');
[- INSERT INTO feedback (feedback_ID, user_ID, description, rating, date) VALUES
  ('FB005', 'EN8054', 'Satisfactory experience', '4', '2024-06-12');
```

146 %

Messages

(1 row affected)

(1 row affected)

(1 row affected)


(1 row affected)

(1 row affected)

Completion time: 2024-10-15T13:02:43.6461671+05:30

```
- INSERT INTO student (user_ID, year, major) VALUES  
  ('IT3701', '3', 'Information Technology');  
  
- INSERT INTO student (user_ID, year, major) VALUES  
  ('EN8054', '4', 'Engineering');  
  
- INSERT INTO student (user_ID, year, major) VALUES  
  ('BM7503', '2', 'Business');  
  
- INSERT INTO student (user_ID, year, major) VALUES  
  ('HM0701', '1', 'Hotel Management');  
  
- INSERT INTO student (user_ID, year, major) VALUES  
  ('IT7062', '3', 'Information Technology');
```

146 %

 Messages

(1 row affected)

(1 row affected)

(1 row affected)


(1 row affected)

(1 row affected)

Completion time: 2024-10-15T13:03:07.3258287+05:30


```
[- INSERT INTO Faculty_member (user_ID, Dep_ID, title) VALUES  
  ('L90701', 'DEP004', 'SE lecturer');  
[- INSERT INTO Faculty_member (user_ID, Dep_ID, title) VALUES  
  ('L07062', 'DEP003', 'CS Lecturer');  
[- INSERT INTO Faculty_member (user_ID, Dep_ID, title) VALUES  
  ('BM7503', 'DEP002', 'Management');  
[- INSERT INTO Faculty_member (user_ID, Dep_ID, title) VALUES  
  ('IT3701', 'DEP001', 'Netrowking');  
[- INSERT INTO Faculty_member (user_ID, Dep_ID, title) VALUES  
  ('HM0701', 'DEP005', 'Cookery');
```

146 %

 Messages

(1 row affected)

(1 row affected)

(1 row affected)


(1 row affected)

(1 row affected)

Completion time: 2024-10-15T13:03:36.1382142+05:30

```
- INSERT INTO Staff (user_ID, office, position) VALUES  
('L90701', 'Office A', 'Lecturer');  
  
- INSERT INTO Staff (user_ID, office, position) VALUES  
('L07062', 'Office B', 'Assistant Lecturer');  
  
- INSERT INTO Staff (user_ID, office, position) VALUES  
('A07503', 'Office C', 'Technician');  
  
- INSERT INTO Staff (user_ID, office, position) VALUES  
('L77503', 'Office C', 'LIC');  
  
- INSERT INTO Staff (user_ID, office, position) VALUES  
('L94753', 'Office C', 'Lecturer');
```

146 %

 Messages

(1 row affected)

(1 row affected)

(1 row affected)


(1 row affected)

(1 row affected)

Completion time: 2024-10-15T13:03:58.3267057+05:30

```
- INSERT INTO donator_phone (donator_ID, phone) VALUES  
  ('D73501', '0719876543');  
- INSERT INTO donator_phone (donator_ID, phone) VALUES  
  ('D88502', '0771234567');  
- INSERT INTO donator_phone (donator_ID, phone) VALUES  
  ('D82473', '0757654321');  
- INSERT INTO donator_phone (donator_ID, phone) VALUES  
  ('D91634', '0723456789');  
- INSERT INTO donator_phone (donator_ID, phone) VALUES  
  ('D09205', '0719988776');
```

146 %

 Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T13:04:32.2386216+05:30

```
INSERT INTO Review (review_ID, topic, r_description, rating, user_ID) VALUES
('R02301', 'SQL Book', 'Very informative', '5', 'IT3701');

INSERT INTO Review (review_ID, topic, r_description, rating, user_ID) VALUES
('R00672', 'Business Journal', 'Well written', '4', 'L90701');

INSERT INTO Review (review_ID, topic, r_description, rating, user_ID) VALUES
('R03043', 'Network Security', 'Needs more detail', '3', 'L07062');

INSERT INTO Review (review_ID, topic, r_description, rating, user_ID) VALUES
('R02034', 'Hospitality Video', 'Engaging content', '4', 'A07503');

INSERT INTO Review (review_ID, topic, r_description, rating, user_ID) VALUES
('R07545', 'Linear Algebra', 'Highly recommended', '5', 'EN8054');
```

146 %

Messages

(1 row affected)

(1 row affected)

(1 row affected)


(1 row affected)

(1 row affected)

Completion time: 2024-10-15T13:05:07.0054876+05:30

```
INSERT INTO Item_review (review_ID, item_ID) VALUES  
('R02301', 'B90001');  
  
INSERT INTO Item_review (review_ID, item_ID) VALUES  
('R00672', 'J70010');  
  
INSERT INTO Item_review (review_ID, item_ID) VALUES  
('R03043', 'B40503');  
  
INSERT INTO Item_review (review_ID, item_ID) VALUES  
('R02034', 'D00007');  
  
INSERT INTO Item_review (review_ID, item_ID) VALUES  
('R07545', 'D75906');
```

146 %

 Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T13:05:33.4772856+05:30

```
INSERT INTO Loan_extension (loan_extension_ID, transaction_ID, new_due_date, extension_reason) VALUES
('LE001', 'T73451', '2024-09-15', 'Research purpose');

INSERT INTO Loan_extension (loan_extension_ID, transaction_ID, new_due_date, extension_reason) VALUES
('LE002', 'T08302', '2024-10-19', 'Additional reading');

INSERT INTO Loan_extension (loan_extension_ID, transaction_ID, new_due_date, extension_reason) VALUES
('LE004', 'T00225', '2024-10-28', 'Additional reading');

INSERT INTO Loan_extension (loan_extension_ID, transaction_ID, new_due_date, extension_reason) VALUES
('LE005', 'T94254', '2024-10-20', 'Additional reading');

INSERT INTO Loan_extension (loan_extension_ID, transaction_ID, new_due_date, extension_reason) VALUES
('LE003', 'T86323', '2024-06-22', 'Delayed report submission');
```

146 %

Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T13:05:58.3898019+05:30

```
[- INSERT INTO lib_staff_user (user_ID, lib_stf_ID) VALUES  
  ('IT3701', 'LS0101');  
[- INSERT INTO lib_staff_user (user_ID, lib_stf_ID) VALUES  
  ('L90701', 'LS0101');  
[- INSERT INTO lib_staff_user (user_ID, lib_stf_ID) VALUES  
  ('L07062', 'LS0101');  
[- INSERT INTO lib_staff_user (user_ID, lib_stf_ID) VALUES  
  ('A07503', 'LS7113');  
[- INSERT INTO lib_staff_user (user_ID, lib_stf_ID) VALUES  
  ('EN8054', 'LS8902');
```

146 %

 Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T13:07:09.8250467+05:30

```
INSERT INTO Request (req_ID, user_ID, lib_stf_ID, req_topic, req_description, status) VALUES
('REQ004', 'A07503', 'LS8734', 'E-books', 'Need access to e-book collection', 'Pending');

INSERT INTO Request (req_ID, user_ID, lib_stf_ID, req_topic, req_description, status) VALUES
('REQ005', 'EN8054', 'LS7113', 'New Journals', 'Request for latest engineering journals', 'Approved');

INSERT INTO Request (req_ID, user_ID, lib_stf_ID, req_topic, req_description, status) VALUES
('REQ006', 'IT3701', 'LS0101', 'Turntin', 'Request to get access for Turntin', 'Pending');

INSERT INTO Request (req_ID, user_ID, lib_stf_ID, req_topic, req_description, status) VALUES
('REQ007', 'L07062', 'LS8902', 'System Issue', 'Issue with digital media access', 'Resolved');

INSERT INTO Request (req_ID, user_ID, lib_stf_ID, req_topic, req_description, status) VALUES
('REQ008', 'L90701', 'LS8734', 'System Issue', 'Cannot access for Turntin', 'Approved');
```


146 %
Messages

(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)

Completion time: 2024-10-15T13:07:42.3448692+05:30


```
- INSERT INTO Lib_staff_item (item_ID, lib_stf_ID) VALUES  
  ('J25309', 'LS8734');  
- INSERT INTO Lib_staff_item (item_ID, lib_stf_ID) VALUES  
  ('D00007', 'LS7113');  
- INSERT INTO Lib_staff_item (item_ID, lib_stf_ID) VALUES  
  ('B40002', 'LS0101');  
- INSERT INTO Lib_staff_item (item_ID, lib_stf_ID) VALUES  
  ('J84908', 'LS8902');  
- INSERT INTO Lib_staff_item (item_ID, lib_stf_ID) VALUES  
  ('B40503', 'LS1285');
```

146 %

 Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T13:08:14.8347684+05:30

```
INSERT INTO Lib_stf_phone (lib_stf_ID, phone) VALUES  
('LS8734', '0719988776');  
  
INSERT INTO Lib_stf_phone (lib_stf_ID, phone) VALUES  
('LS1285', '0705544332');  
  
INSERT INTO Lib_stf_phone (lib_stf_ID, phone) VALUES  
('LS7113', '0765544321');  
  
INSERT INTO Lib_stf_phone (lib_stf_ID, phone) VALUES  
('LS8902', '0714455667');  
  
INSERT INTO Lib_stf_phone (lib_stf_ID, phone) VALUES  
('LS0101', '0722233445');
```

% ▾

Messages

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T13:08:37.3860756+05:30

```
- INSERT INTO Item_donator (donator_ID, item_ID) VALUES  
  ('D91634', 'B10604');  
- INSERT INTO Item_donator (donator_ID, item_ID) VALUES  
  ('D73501', 'J58282');  
- INSERT INTO Item_donator (donator_ID, item_ID) VALUES  
  ('D88502', 'D90756');  
- INSERT INTO Item_donator (donator_ID, item_ID) VALUES  
  ('D09205', 'J84908');  
- INSERT INTO Item_donator (donator_ID, item_ID) VALUES  
  ('D82473', 'B40002');
```

% ▾

Messages

(1 row affected)

(1 row affected)

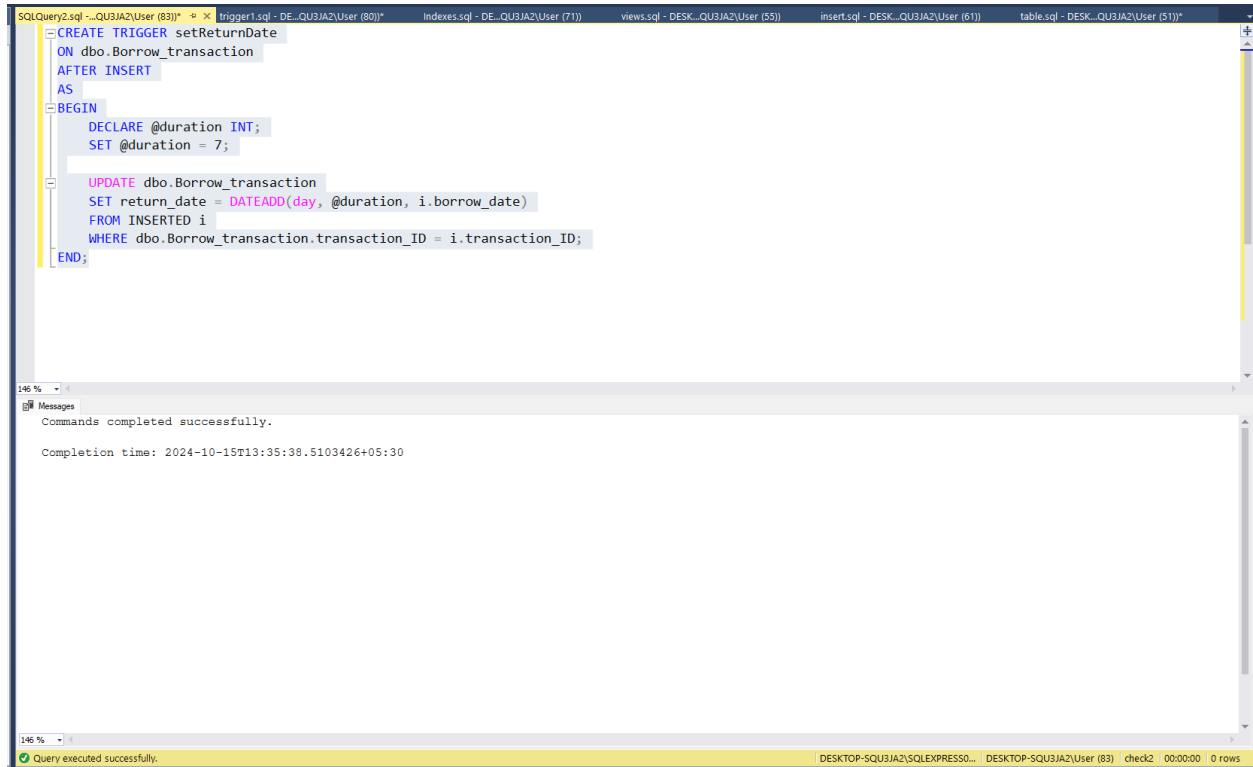
(1 row affected)

(1 row affected)

(1 row affected)

Completion time: 2024-10-15T13:09:03.1646206+05:30

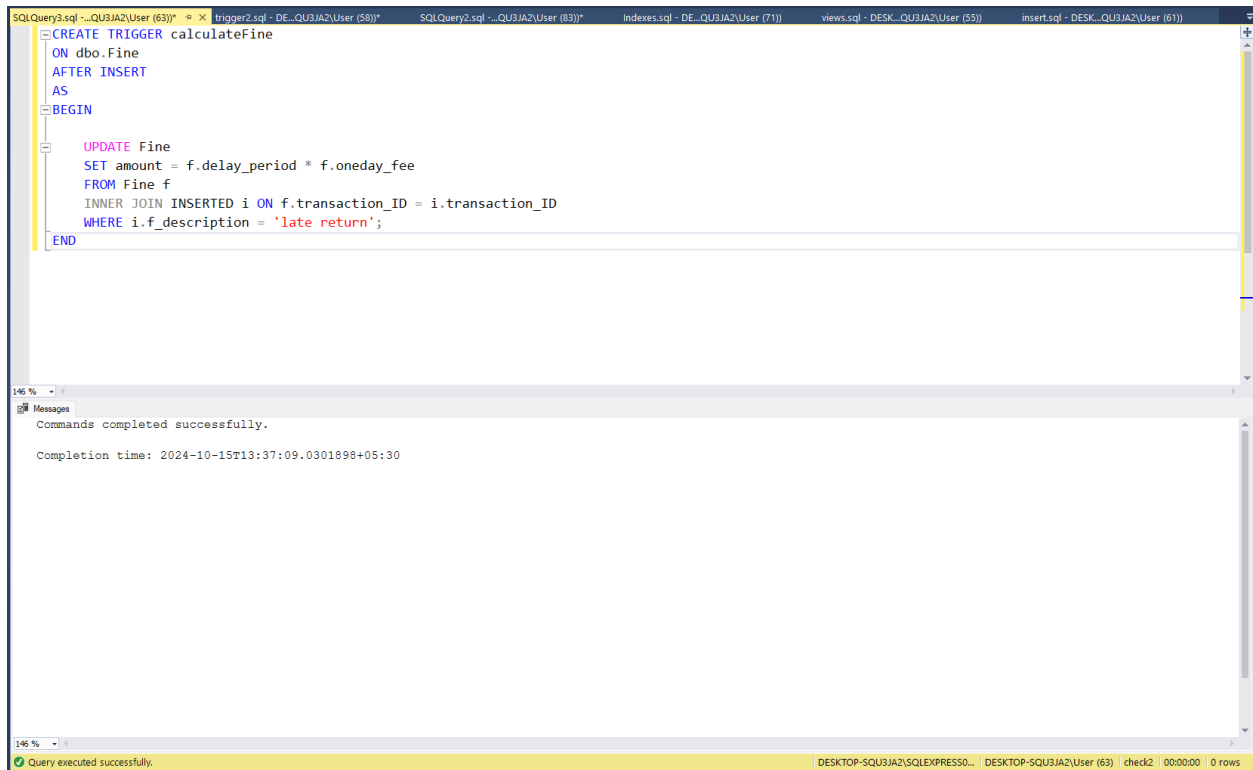
6. Triggers



```
SQLQuery2.sql - QU3JA2\User (83)* * X trigger1.sql - DE...QU3JA2\User (80)* * Indexes.sql - DE...QU3JA2\User (71)* * views.sql - DESK...QU3JA2\User (53)* * insert.sql - DESK...QU3JA2\User (61)* * table.sql - DESK...QU3JA2\User (51)* *  
-- CREATE TRIGGER setReturnDate  
ON dbo.Borrow_transaction  
AFTER INSERT  
AS  
BEGIN  
    DECLARE @duration INT;  
    SET @duration = 7;  
  
    UPDATE dbo.Borrow_transaction  
    SET return_date = DATEADD(day, @duration, i.borrow_date)  
    FROM INSERTED i  
    WHERE dbo.Borrow_transaction.transaction_ID = i.transaction_ID;  
END;
```

146 %
Messages
Commands completed successfully.
Completion time: 2024-10-15T13:35:38.5103426+05:30

146 %
Query executed successfully. | DESKTOP-SQU3JA2\SQLEXPRESSO... | DESKTOP-SQU3JA2\User (83) | check2 | 00:00:00 | 0 rows



```
SQLQuery3.sql - DESKTOP-SQU3JA2\User (63) * X trigger2.sql - DE...QU3JA2\User (58) * SQLQuery2.sql - ...QU3JA2\User (63) * Indexes.sql - DE...QU3JA2\User (71) views.sql - DESK...QU3JA2\User (55) insert.sql - DESK...QU3JA2\User (61)
CREATE TRIGGER calculateFine
ON dbo.Fine
AFTER INSERT
AS
BEGIN
    UPDATE Fine
    SET amount = f.delay_period * f.oneday_fee
    FROM Fine f
    INNER JOIN INSERTED i ON f.transaction_ID = i.transaction_ID
    WHERE i.f_description = 'late return';
END
```

146 %
Messages
Commands completed successfully.
Completion time: 2024-10-15T13:37:09.0301898+05:30

146 %
Query executed successfully. DESKTOP-SQU3JA2\SQLEXPRESSO... | DESKTOP-SQU3JA2\User (63) | check2 | 00:00:00 | 0 rows

7. Views

--This view shows details about members who has fine

```
CREATE VIEW fine_details AS
SELECT uf.user_ID, f.fine_ID, f.amount
FROM User_fine uf, Fine f
WHERE uf.fine_ID = f.fine_ID;
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T13:21:37.0870018+05:30

--This view shows the details about donated items and it's donator details

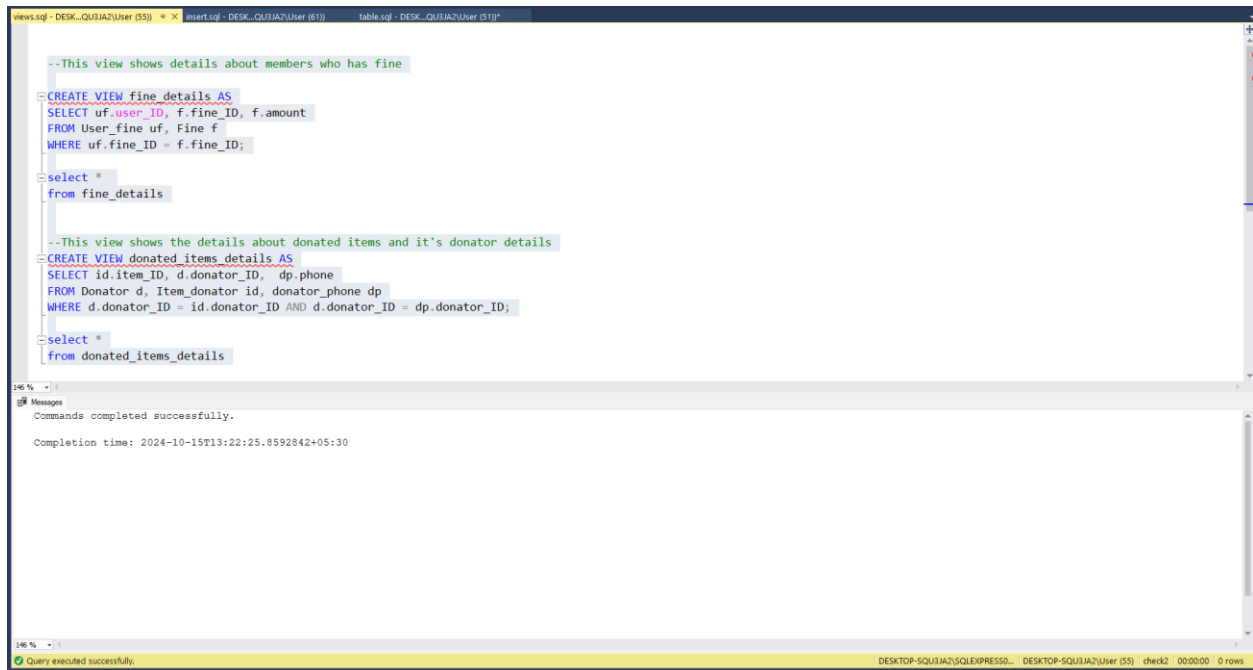
```
CREATE VIEW donated_items_details AS
SELECT id.item_ID, d.donator_ID, dp.phone
FROM Donator d, Item_donator id, donator_phone dp
WHERE d.donator_ID = id.donator_ID AND d.donator_ID = dp.donator_ID;
```

146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T13:22:25.8592842+05:30



```
--This view shows details about members who has fine
CREATE VIEW fine_details AS
SELECT uf.user_ID, f.fine_ID, f.amount
FROM User_fine uf, Fine f
WHERE uf.fine_ID = f.fine_ID;

select *
from fine_details

--This view shows the details about donated items and it's donator details
CREATE VIEW donated_items_details AS
SELECT id.item_ID, d.donator_ID, dp.phone
FROM Donator d, Item_donator id, donator_phone dp
WHERE d.donator_ID = id.donator_ID AND d.donator_ID = dp.donator_ID;

select *
from donated_items_details
```

Messages

Commands completed successfully.

Completion time: 2024-10-15T13:22:25.8592842+05:30

Query executed successfully.

DESKTOP-SQU3IA2\SQLEXPRESS... | DESKTOP-SQU3IA2\User (SS) | check2 | 00:00:00 | 0 rows

8. Indexes

```
-- Create index on item_ID in Item_borrow table
CREATE INDEX idx_item_borrow_ID ON Item_borrow(item_ID);

-- Create index on user_ID in Fine table
CREATE INDEX idx_fine_userID ON Fine(transaction_ID);
```

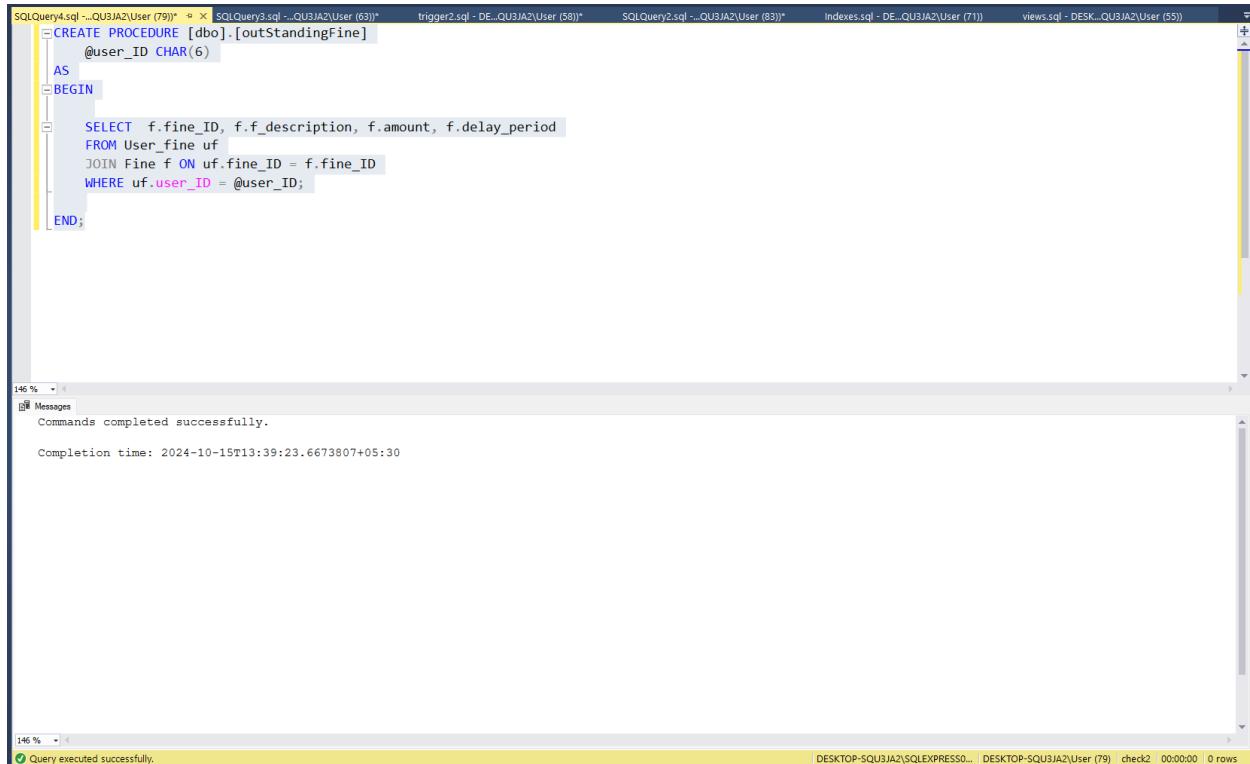
146 %

Messages

Commands completed successfully.

Completion time: 2024-10-15T13:28:42.5001895+05:30

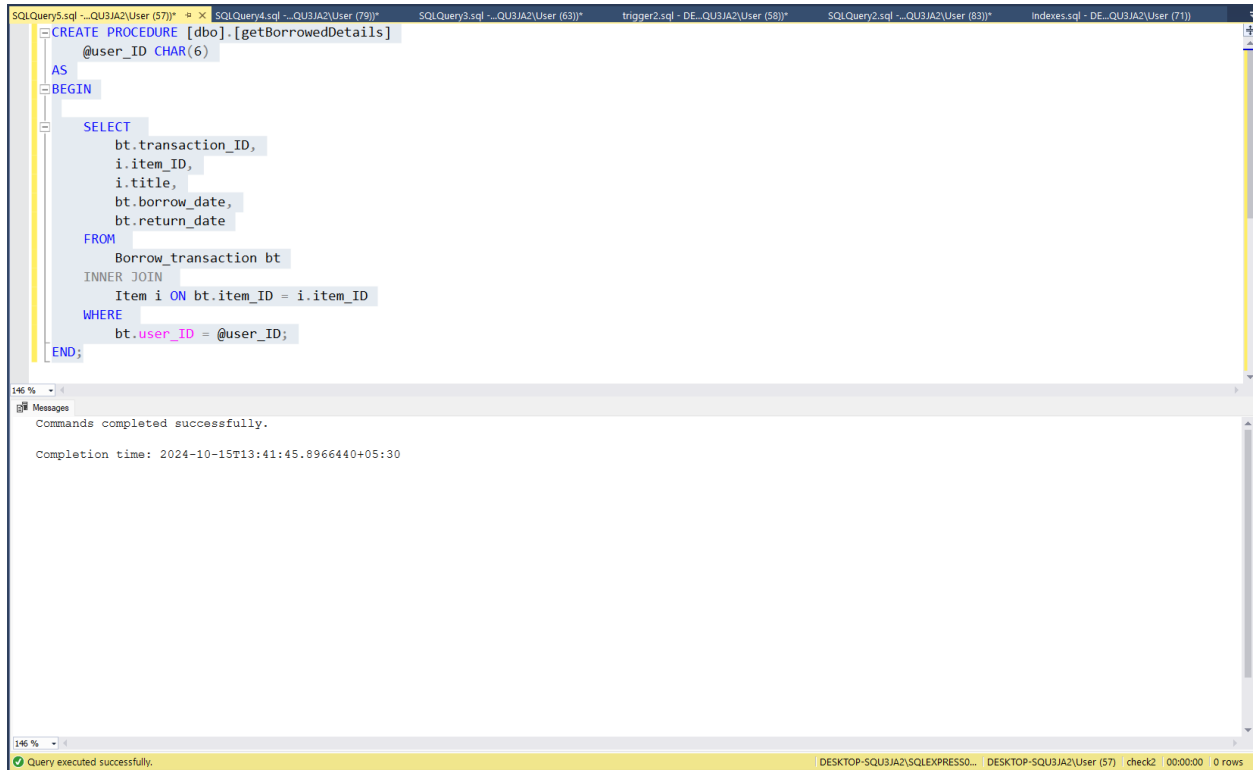
9. Procedures



The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows the SQL code for a stored procedure named [dbo].[outStandingFine]. The code is as follows:

```
CREATE PROCEDURE [dbo].[outStandingFine]
    @user_ID CHAR(6)
AS
BEGIN
    SELECT f.fine_ID, f.f_description, f.amount, f.delay_period
    FROM User_fine uf
    JOIN Fine f ON uf.fine_ID = f.fine_ID
    WHERE uf.user_ID = @user_ID;
END;
```

The bottom pane shows the Messages tab, indicating that the commands completed successfully. The completion time is 2024-10-19T13:39:23.6673807+05:30. The status bar at the bottom indicates that the query was executed successfully.



```
SQLQuery5.sql - QU3JA2\User (57)*  SQLQuery4.sql - QU3JA2\User (79)*  SQLQuery3.sql - QU3JA2\User (63)*  trigger2.sql - DE...QU3JA2\User (58)*  SQLQuery2.sql - QU3JA2\User (83)*  Indexes.sql - DE...QU3JA2\User (71)*
CREATE PROCEDURE [dbo].[getBorrowedDetails]
    @user_ID CHAR(6)
AS
BEGIN
    SELECT
        bt.transaction_ID,
        i.item_ID,
        i.title,
        bt.borrow_date,
        bt.return_date
    FROM
        Borrow_transaction bt
    INNER JOIN
        Item i ON bt.item_ID = i.item_ID
    WHERE
        bt.user_ID = @user_ID;
END;
```

Messages

Commands completed successfully.

Completion time: 2024-10-15T13:41:45.8966440+05:30

Query executed successfully. | DESKTOP-SQU3JA2\SQLEXPRESSO... | DESKTOP-SQU3JA2\User (57) | check2 | 00:00:00 | 0 rows

10. Analysis of 2 database vulnerabilities

10.2. SQL injections

Sql injection is a type of dangerous attack where malicious sql code is injected into an application, allowing attackers to manipulate database remotely .SQL (structured query language) is used to manage and modify data in databases, and an sql injection attack can enable attacker to perform a range of activities on the database server. These activities include stealing or modifying data bypassing authentication, and even gain root access to the system (database).

Techniques

1) Classic SQL injections:

This occurs when an attacker directly injects malicious SQL code into an application's input fields. For an example login page or search bars, to implement database queries. That is directly executed by the backend database. By manipulating user inputs without proper sanitization or validation, the attacker can alter the SQL query.

Examples:

Let's assume web application has the following query for user login:

```
SELECT * FROM users WHERE username = 'username' AND password = 'password' ;
```

An attacker inserts:

' OR '1' = '1' ; --

Since '1' = '1' always evaluates to **true** and -- comments out the rest of the query, the attacker can **bypass authentication**, gaining **unauthorized access**

```
SELECT * FROM users WHERE username = ' ' OR '1' = '1' ; -- ' AND password = ' ' ;
```

2) Blind SQL injections:

In this type of SQL injections where the attacker doesn't directly see the result of their injected queries. An attacker gains knowledge about the database via a successions of true and false conditions. This is more challenge than classical SQL injection because attacker cannot immediately see the output. Blind SQL injections are divided into 2 main types Boolean-based and time-based.

Boolean Based:

If the condition (username = 'admin' AND 1 = 1) is true, the web page loads normally. If the condition is false (username = 'admin' AND 1 = 2), the page may display an error or behave differently

```
SELECT * FROM users WHERE username = 'admin' AND 1 = 1 ; --  
SELECT * FROM users WHERE username = 'admin' AND 1 = 2 ; --
```

Let's assume an attacker want to know the admin's password. He can use following query for it .If the first letter of a password is 'a', page loads normally. If it is not, an attacker can use different letter and try out.

```
SELECT * FROM users WHERE username = 'admin' AND SUBSTRING (password,1,1) = 'a'; --
```

Time Based:

In a time based blind SQL injection attacks attacker purposefully delays SQL queries. Instead of getting a direct output, the attacker makes the database “wait” for a certain amount of time if a condition is true, using the delay to determine the result.

For an example:

If the web page takes 5 seconds to respond, it means the username (admin) exists and if the page loads immediately, attacker can try another username.

```
SELECT * FROM users WHERE username = 'admin' AND IF (1=1, SLEEP(5), 0) ; --
```

3) Union-Based SQL injections

Union-Based SQL injection is an attack particularly used in web applications with the inclusion of Union SQL operators is an SQL injection attack to combine the results of two or more SELECT queries into a single result set. It is useful for retrieving data from various in the database that the application does not have direct access to.

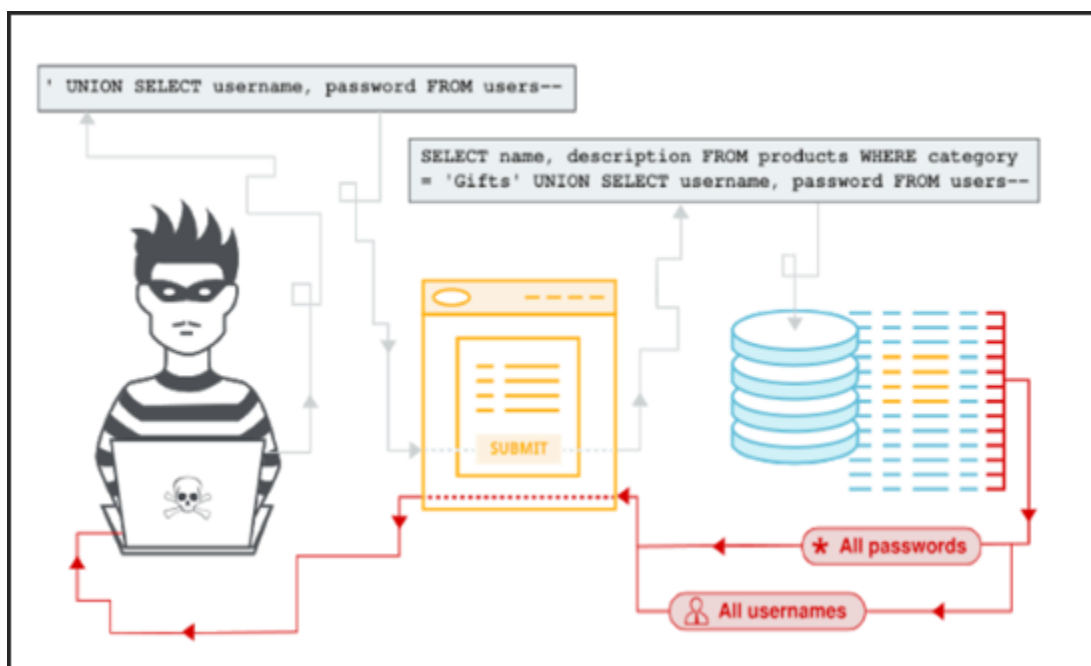
First, attacker can execute a legitimate SQL query to retrieve the data and check whether it works or not. As an example,

```
SELECT * FROM users WHERE id = 1; --
```

After that, an attacker can manipulate the UNION statement and evaluate to merge the output of their own query with the legitimate query.

```
SELECT * FROM users WHERE id = 1 UNION SELECT username, password FROM admin ; --
```

If the attacker's query successful, the database returns a result set that includes data such as username and password from the admin table. Key point is that for a UNION statement to succeed, the number of columns in the users table and admin table must be equal.



Impact

SQL injection attacks have a significant impact on organizations. It often leads to data breaches that expose sensitive information such as financial records and personal information. Therefore, these attacks can compromise the integrity of sensitive data by allowing unauthorized users to make intentional modifications and delete information. In various cases, attackers can bypass authentication, allowing access to system applications, which can lead to further exploitation and system breaches. In addition SQL injections can expose valuable data and information, helping attackers launch more sophisticated and precise attacks. As a result, reputation is damaged and customer trust is damaged. Not only that Business activities are disrupted and legal fines cause significant financial losses. This makes SQL injections a serious threat that requires strict security measures.

10.2. Denial of Service (DoS) attacks

DoS attack is an attempt where the attacker tries to interrupt or shut down the targeted server, service or network by overwhelming it with a flood of illegitimate requests that trigger a crash. This will lead the target to become slow, unresponsive or inaccessible for legitimate users. These attacks exploit the limitations of a system's resources such as bandwidth, processing power and memory by sending excessive requests or malicious commands.

Types of DoS attacks,

1) Flood attacks

These are the attacks where attackers overwhelm a system with excessive traffic, interrupting the legitimate requests.

Ex – Dyn attack 2016

Technique –

- Attackers analyze the data traffic of a system and send an overwhelming number of requests to the server or database, overwhelming its processing capacity.
- To exploit databases, attackers send SQL queries, connection requests or data-intensive operations to exhaust server resources. This can be automated by using tools which are sending queries at a high rate.

Types –

- UDP Flood (User Datagram Protocol) –

Attacker sends lots of UDP packets to random ports on target server. This causes the server to check the ports repeatedly.

- SYN Flood (Synchronization) –

This is an attack focused on TCP protocol, The attacker sends lots of SYN requests and never complete the handshake and the server will have incomplete connections.

Impact –

- By facing a flood attack the system will be get unresponsive because of exhausted resources. And most importantly legitimate users will not be able to interact with the system and if the targets data base is a business driven one there will be a lost.

2) Application Layer Attack

These attacks are focused on the functionality of the system by exploiting complex database queries or inefficient search algorithms.

Technique -

- Application layer attacks (layer 7 attacks) target specific vulnerabilities of a system and these will be more sophisticated and hard to detect.

Attackers may use – SQL injection

Heavy SQL Query

Locking Resources

Impact –

- System will slow down and cause legitimate user requests will be queued or denied because the database is too busy processing the attacker's complex queries.
- Data base will become unresponsive for normal operations and will effectively deny services to legitimate users.
- For financial based databases lost will occur.

3) Protocol Attack

Technique –

- Attackers will exploit weaknesses in the communication protocols that manage network traffic between client, server and database. These attacks will cause network jam or connection failures and prevent legitimate users accessing the database.
- The attacker sends malformed oversized packets that violate the defined size limits in the protocol. And this will cause the server to crash or behave unexpectedly.

Impact –

- Like all other DoS attacks protocol attack also will exhaust the resources and this will cause the server unresponsive.
- Data base could not be able to process legitimate connection requests because available ports are exhausted. And there will be also network congestion.
- By this attack there will be weakened ports available and they might be lead to security breach.

11. Mitigation and countermeasure suggestions

11.1. SQL Injection countermeasures:

1. Prepared Statements:

Parameterized Queries :

- You can use placeholders to input and bind user inputs rather than directly inserting user inputs
- From that you can prevent from malicious inputs from altering database commands because the user inputs are treated as plain data not as the part of query.

Avoid Dynamic Queries

- Don't create database queries by directly concatenating user input data with SQL statements
- Because these dynamic queries can be helped to attacker to manipulate inject malicious commands.

2. Validating and Sanitizing Input

Validation :

- Ensure that users input their inputs according to domain and what is expected.
- User inputs must be fulfilled the given format.
- By enforcing this rules , you can assure that you stop attackers to input harmful data like scripts
- Then you can make sure that validation reduce the risk of SQL injection.

Sanitization :

- Cleaning or modifying the input to remove harmful characters
- If dangerous input sneaks to database by fulfilling validation also, then sanitization can identify that it cannot harm to system by turning it into safe input .
 - Example: Ensure that phone number must contain 10 digits only
 - Ensure that person's name cannot have special characters like “ #,@,*”

3. Stored procedures

- Stored procedure means it is predefined SQL query stored in the database.
- The application only calls the predefined procedure and passes in required data.
- It do not want to write SQL queries every time.
- Then the SQL logic is fixed. Because of that attacker or unauthorized user cannot change SQL logic by user input .
- It decreases the risk of SQL Injection
- These procedures know how to fetch the data safely.

4. Escape special characters

- Convert user input special characters like quotes into harmless versions.
- Because special character of user inputs can be harm to the database .
- Escape special character ensures that they are handled safely and do not interfere with SQL query structure.

Example :

if user inputs as **o'really** then database would safely handle it as **o " really**

Why single quotation mark (') is dangerous in SQL ?

- It defines string literals in SQL queries. If don't handle properly an attacker can insert malicious code including single quotes potentially leading to SQL Injection.
- if attacker includes a single quote in their input , they can hastily close the string and insert malicious SQL queries.

```
select *  
from Registered_User  
where FirstName = '' OR '1' = '1';
```

This query would return all users , because '1' = '1' is always true.

So the double single quotes act as single quotes within the string making it safe to execute in database.

5. Database Security Configuration:

Database User permission

- Limit and control users access to database and give access to database that are only they want.
- So they can only do what is necessary for their job
- Then minimize the damage by restricting their access if they gain control to the database by attackers.

Regular Update

- Keep your database and software patched within latest security updates.
- Then you can ensures vulnerabilities in your software are fixed before attackers can exploit them.

6. Error Handling

- Don't allow to database to show detailed error message to user when something get wrong .
- Because it can be an information to unauthorized user or to attacker to identify the structure of the database.
- You can simply show error message as "Something went wrong " rather than
- "Unknown column DepartmentName" .
- if use detailed error message then attacker can analyze what are the column and tables and all in this database and what are the columns are not there.

7. Database-level security features

- Modern database management software come with built-in security features to detect and prevent harmful queries before execution.
- This act as safety net within the database.

8. Use Web Application Firewalls(WAF)

- WAF is the intermediate device between the user and web application of the given database.
- It scans block incoming malicious request like SQL injection attempts.
- If your code has vulnerabilities , WAF can block them before goes to the database.

9. Train the Experts:

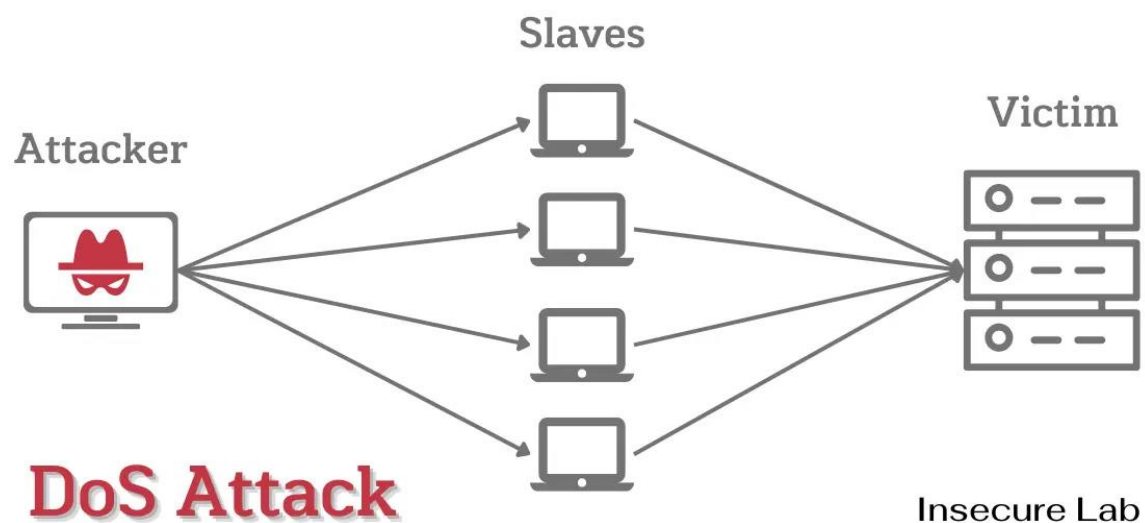
- We can maximize awareness of security by providing security focused training to other non-security professionals as well.
- Then they can be more aware about how to write query more securely .
- It will minimize the damage attackers can do by restricting their access if they gain control.

10. Regular Update

- Keep your database and software patched within latest security updates.
- Then you can ensures vulnerabilities in your software are fixed before attackers can exploit them.

11.2. Dos Attack countermeasures

- A (DoS) attack is a malicious attempt to disrupt the normal functioning of a targeted server, service, or network by overwhelming it with a flood of traffic or by exploiting vulnerabilities. The goal is to render the service unavailable to legitimate users, effectively denying access to the resources or services it provides **Techniques.**



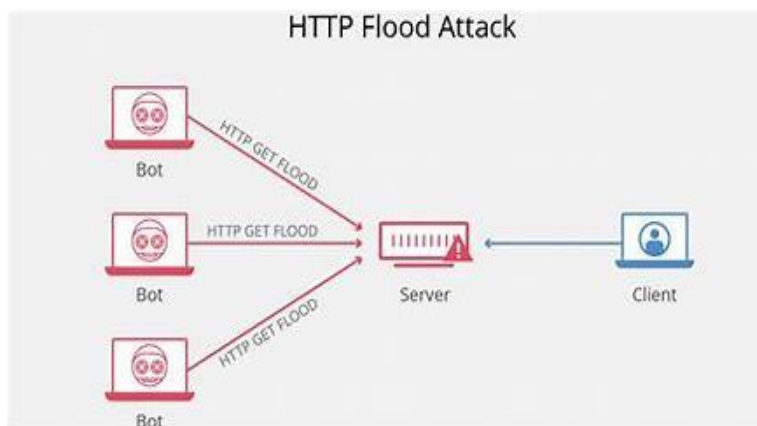
Types of dos attack

- Flood Attack
- Application Layer Attack
- Protocol Attack

Countermeasures of the various types of Dos Attacks

1.Flood Attacks

- **ICMP Flood**
 - **Rate Limiting:** Limit the number of ICMP packets that can be processed.
 - **Filtering:** Block ICMP traffic at the firewall or router, except for legitimate use.
- **SYN Flood**
 - **SYN Cookies:** Use SYN cookies to validate requests without allocating resources until the handshake is complete.
 - **Connection Limits:** Limit the number of concurrent connections from a single IP address.
- **UDP Flood**
 - **UDP Rate Limiting:** Limit the rate of UDP packets from individual IPs.
 - **Filtering:** Block incoming UDP traffic to unused ports.



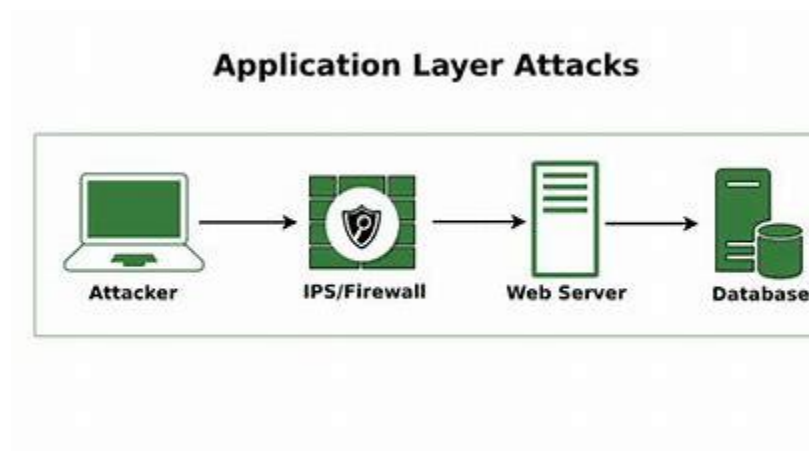
2.Application Layer Attacks

- **HTTP Flood**

- **Web Application Firewall (WAF):** Deploy a WAF to filter out malicious HTTP requests.
- **Caching:** Use caching solutions to reduce server load from repeated requests.

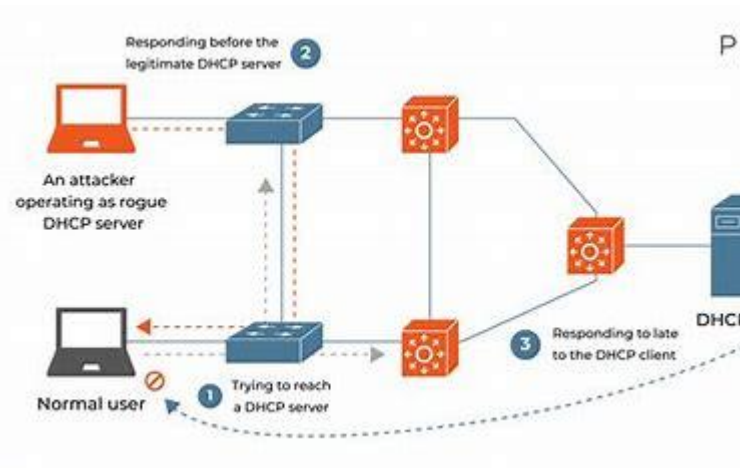
- **Slowloris**

- **Timeout Settings:** Configure timeouts to close idle connections more quickly.
- **Connection Limits:** Limit the number of simultaneous connections from a single IP.



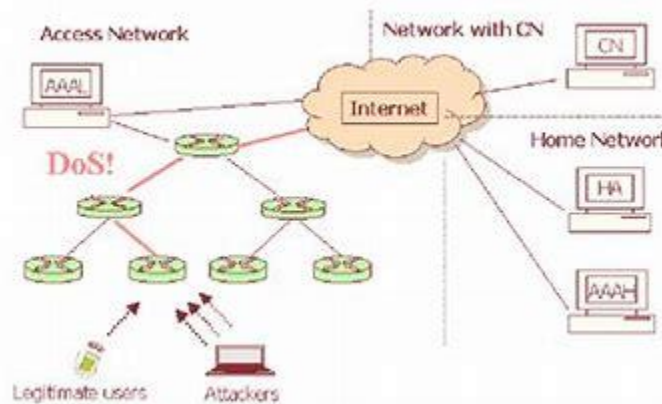
3. Protocol Attacks

- **Ping of Death**
 - **Packet Size Filtering:** Set limits on packet sizes accepted by the server.
 - **IP Filtering:** Block malformed packets at the firewall.
- **Smurf Attack**
 - **Disable IP Spoofing:** Configure routers to prevent IP address spoofing.
 - **Disable ICMP Broadcasts:** Turn off ICMP broadcasts on devices.



4. Resource Exhaustion Attacks

- **DNS Amplification**
 - **Rate Limiting:** Limit the rate of DNS requests from individual IP addresses.
 - **DNSSEC:** Implement DNS Security Extensions to validate requests.
- **NTP Amplification**
 - **Rate Limiting:** Limit responses from NTP servers.
 - **NTP Configuration:** Configure NTP servers to ignore requests from untrusted sources.



By implementing these countermeasures, organizations can significantly enhance their resilience against various types of DoS attacks.