



AMERICAN  
INTERNATIONAL  
UNIVERSITY-  
BANGLADESH

## Advance Database Management System

### Section – B

**Project Name: Acid Victims Management System**

**Course Teacher: JUENA AHMED NOSHIN**

N o	Student Name	Student ID	Contribution
1	Tasfi Islam	21-45504-3	
.			

## **Contents:**

• Introduction .....	3
• Scenario Description .....	3
• ER Diagram .....	4
• Normalization.....	5-7
• Table creation and insertion .....	8-13
• Queries .....	14-40
• Schema Diagram.....	41
• User Interface .....	42
• Relational Algebra .....	43

## Introduction:

It is an application that uses a database to help store, organize, and monitor information regarding victims of acid attacks. Administrators can use it to register victims, keep track of medical treatments, manage case information, and keep an eye on court cases.

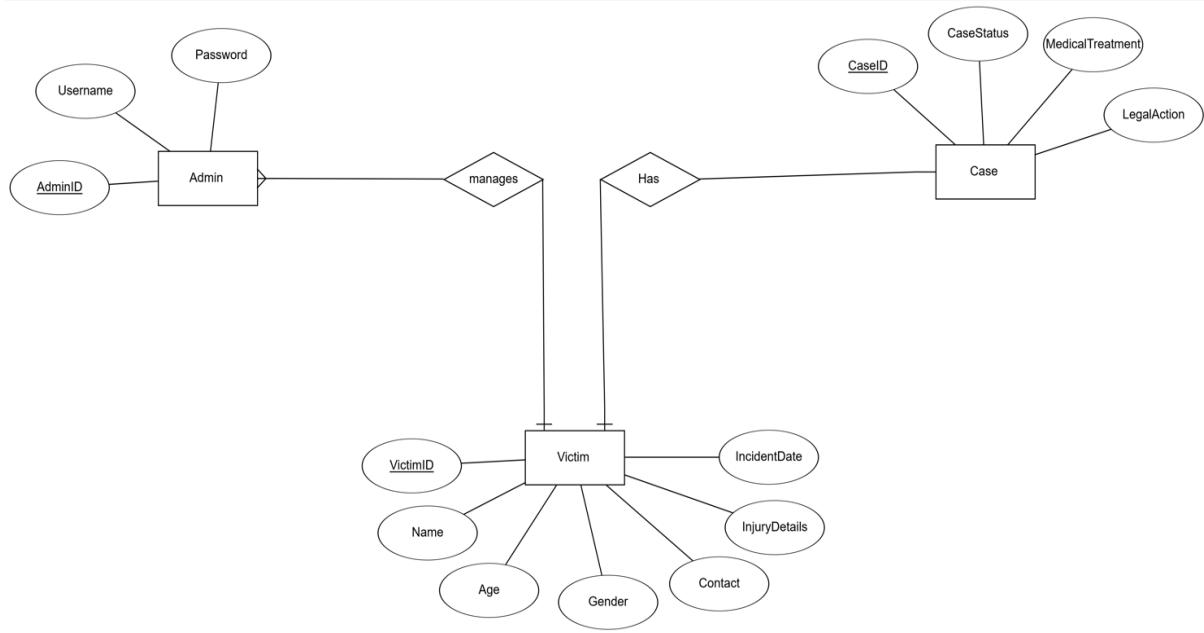
Its features, which include case tracking, victim database administration, and secure admin login, are designed to help victims and expedite the process.

## Scenario Description:

The Acid Victims Management System follows a structured approach. One Administrator manages multiple Victims while maintaining confidentiality and appropriate documentation. Each victim has exactly one case, which keeps track of their medical treatment, legal actions, and case status. The admin is-in- charge of adding and updating victim and case Information while making sure everything is secured and organized. This system provides an efficient way to monitor victims' recovery and legal progress while maintaining data integrity.

## ER Diagram:

---



## **Normalization**

### **Manage**

#### **UNF**

**Manage( AdminID , Username, Password, VictimId, Name, Age, Gender, Contact, InjuryDetails, IncidentDate)**

#### **1NF**

There is no multi valued attribute, Relation already in 1NF.

**( AdminID , Username, Password, VictimId, Name, Age, Gender, Contact, InjuryDetails, IncidentDate)**

## 2NF

1. AdminID , Username, Password
2. VictimId, Name, Age, Gender, Contact, InjuryDetails, IncidentDate

## 3NF

There is no transitive dependencies. Already in **3NF**.

1. AdminID, Username, Password
2. VictimId, Name, Age, Gender, Contact, InjuryDetails, IncidentDate

## Table creation

1. AdminID, Username, Password
2. VictimId, Name, Age, Gender, Contact, InjuryDetails, IncidentDate

**Has**

**UNF**

Has (VictimId, Name, Age, Gender, Contact, InjuryDetails, IncidentDate , CaseID,CaseStatus, Medicaltreatment, LegalAction)

**1NF**

There is no multi valued attribute, Relation already in 1NF.

(VictimId, Name, Age, Gender, Contact, InjuryDetails, IncidentDate , CaseID,CaseStatus, Medicaltreatment, LegalAction)

**2NF**

1. VictimId, Name, Age, Gender, Contact, InjuryDetails, IncidentDate
2. CaseID,CaseStatus, Medicaltreatment, LegalAction

**3NF**

There is no transitive dependencies. Already in **3NF**.

1. VictimId, Name, Age, Gender, Contact, InjuryDetails, IncidentDate

2. CaseID, CaseStatus, MedicalTreatment, LegalAction

### Table creation

1. VictimId, Name, Age, Gender, Contact, InjuryDetails, IncidentDate

2. CaseID, CaseStatus, MedicalTreatment, LegalAction

### Temporary Tables

1. AdminID, Username, Password

2. VictimId, Name, Age, Gender, Contact, InjuryDetails, IncidentDate

~~3. VictimId, Name, Age, Gender, Contact, InjuryDetails, IncidentDate~~

4. CaseID, CaseStatus, MedicalTreatment, LegalAction

## Final Tables

1. AdminID, Username, Password

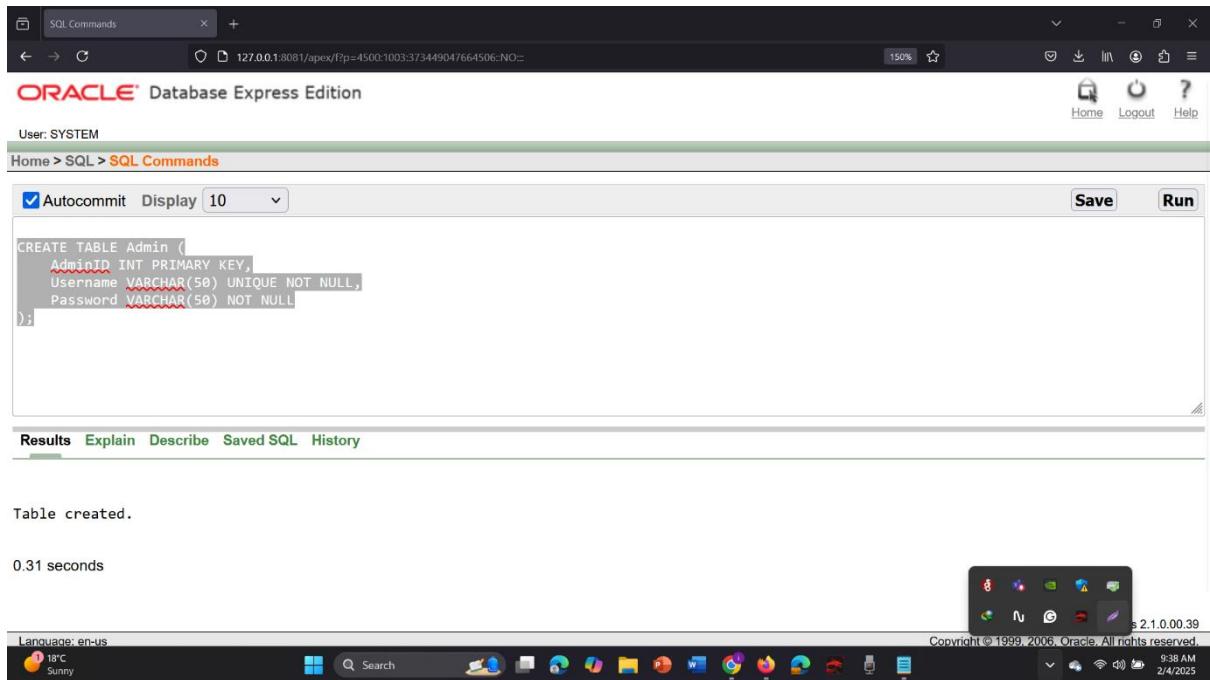
2. VictimId, Name, Age, Gender, Contact, InjuryDetails, IncidentDate

3. CaseID, CaseStatus, MedicalTreatment, LegalAction

## **Table Creation:**

### 1. Admin Table

```
CREATE TABLE Admin (
    AdminID INT PRIMARY KEY,
    Username VARCHAR(50) UNIQUE NOT NULL,
    Password VARCHAR(50) NOT NULL
);
```



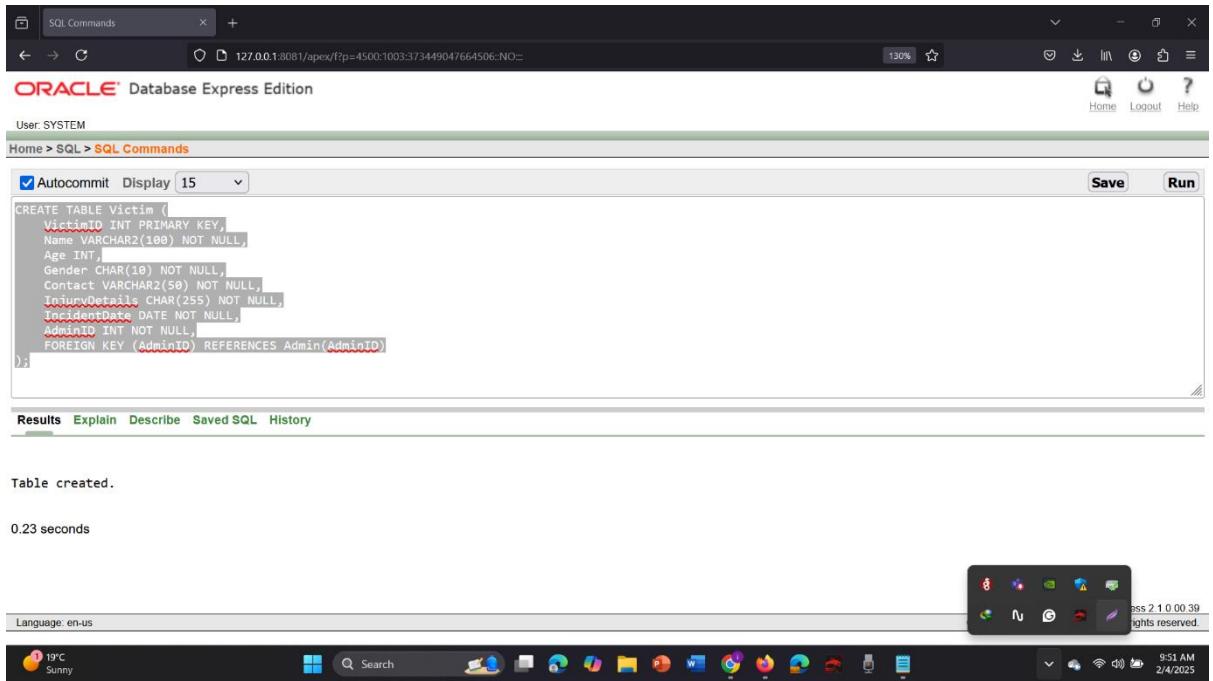
The screenshot shows the Oracle Database Express Edition SQL Commands interface. The URL in the address bar is 127.0.0.1:8081/apex/f?p=4500:1003:373449047664506::NO:::. The page title is "ORACLE® Database Express Edition". The user is SYSTEM. The navigation bar includes Home, Logout, and Help. The main area shows the following SQL code:

```
CREATE TABLE Admin (
    AdminID INT PRIMARY KEY,
    Username VARCHAR(50) UNIQUE NOT NULL,
    Password VARCHAR(50) NOT NULL
);
```

Below the code, the results show "Table created." and a duration of "0.31 seconds". The system tray at the bottom indicates the language is en-us, the weather is 18°C sunny, and the date and time are 2/4/2025 9:38 AM.

## 2. Victim Table

```
CREATE TABLE Victim (
    VictimID INT PRIMARY KEY,
    Name VARCHAR2(100) NOT NULL,
    Age INT,
    Gender CHAR(10) NOT NULL,
    Contact VARCHAR2(50) NOT NULL,
    InjuryDetails CHAR(255) NOT NULL,
    IncidentDate DATE NOT NULL,
    AdminID INT NOT NULL,
    FOREIGN KEY (AdminID) REFERENCES Admin/AdminID
);
```



The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL command entered is:

```
CREATE TABLE Victim (
    VictimID INT PRIMARY KEY,
    Name VARCHAR2(100) NOT NULL,
    Age INT,
    Gender CHAR(10) NOT NULL,
    Contact VARCHAR2(50) NOT NULL,
    InjuryDetails CHAR(255) NOT NULL,
    IncidentDate DATE NOT NULL,
    AdminID INT NOT NULL,
    FOREIGN KEY (AdminID) REFERENCES Admin(AdminID)
);
```

The results show the table was created successfully:

```
Table created.
```

Execution time: 0.23 seconds

The system tray at the bottom right shows the date and time as 2/4/2025, 9:51 AM.

### 3. Case Table

```
CREATE TABLE CaseTable (
    CaseID INT PRIMARY KEY,
    VictimID INT NOT NULL,
    CaseStatus CHAR(15) NOT NULL,
    MedicalTreatment CHAR(255) NOT NULL,
    LegalAction CHAR(255) NOT NULL,
    FOREIGN KEY (VictimID) REFERENCES Victim(VictimID)
);
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The URL in the address bar is 127.0.0.1:8081/apex/f?p=4500:1003:373449047664506..NO. The page title is "User: SYSTEM Home > SQL > SQL Commands". A dropdown menu shows "Autocommit" is checked and "Display" is set to 15. The SQL code entered is:

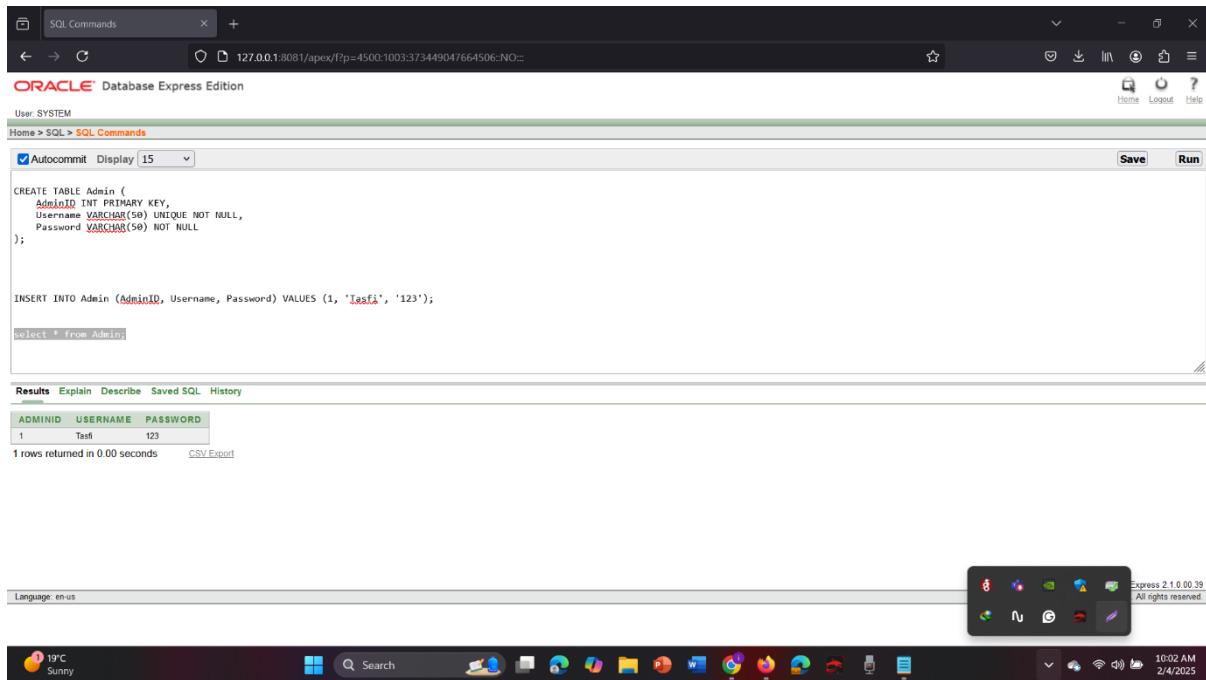
```
CREATE TABLE CaseTable (
    CaseID INT PRIMARY KEY,
    VictimID INT NOT NULL,
    Description CHAR(10) NOT NULL,
    MedicalNotes CHAR(255) NOT NULL,
    LegalActions CHAR(755) NOT NULL,
    FOREIGN KEY (VictimID) REFERENCES Victim(VictimID)
);
```

The results section shows the message "Table created." and a duration of "0.07 seconds". The system tray at the bottom indicates it's 10:04 AM on 2/4/2025, the weather is 19°C and sunny, and the application version is Express 2.1.0.0.39.

## Data Insertion:

### 1. Admin

```
INSERT INTO Admin (AdminID, Username, Password) VALUES (1, 'Tasfi', '123');
```



The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```

CREATE TABLE Admin (
    AdminID INT PRIMARY KEY,
    Username VARCHAR(50) UNIQUE NOT NULL,
    Password VARCHAR(50) NOT NULL
);

INSERT INTO Admin (AdminID, Username, Password) VALUES (1, 'Tasfi', '123');

select * from Admin;

```

The results pane shows a single row inserted:

ADMINID	USERNAME	PASSWORD
1	Tasfi	123

1 rows returned in 0.00 seconds [CSV Export](#)

The system tray at the bottom right shows the Oracle Express icon, the date (2/4/2025), and the time (10:02 AM).

## 2. Victim

INSERT INTO Victim (VictimID, Name, Age, Gender, Contact, InjuryDetails, IncidentDate, AdminID)

VALUES (1, 'Salam', 25, 'Male', '1234567890', 'Face burns from acid attack', TO\_DATE('2024-01-10', 'YYYY-MM-DD'), 1);

INSERT INTO Victim (VictimID, Name, Age, Gender, Contact, InjuryDetails, IncidentDate, AdminID)

VALUES (2, 'Fatema', 30, 'Female', '0987654321', 'Hand and arm injuries', TO\_DATE('2024-01-15', 'YYYY-MM-DD'), 1);

INSERT INTO Victim (VictimID, Name, Age, Gender, Contact, InjuryDetails, IncidentDate, AdminID)

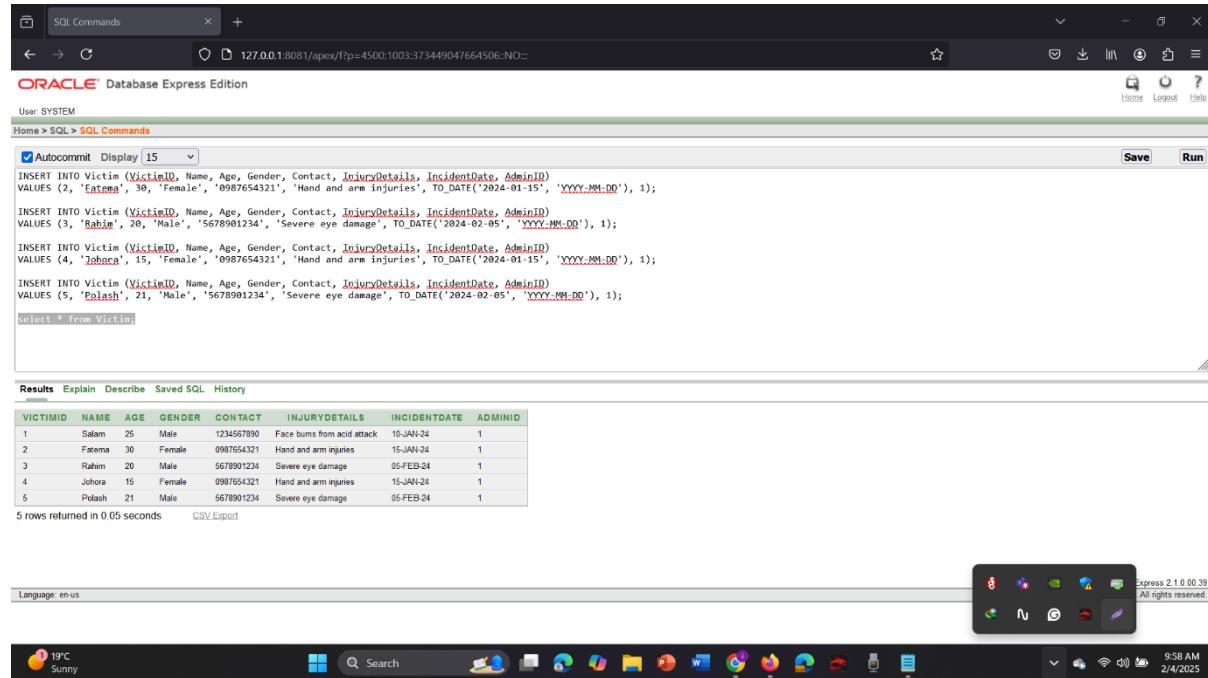
VALUES (3, 'Rahim', 20, 'Male', '5678901234', 'Severe eye damage', TO\_DATE('2024-02-05', 'YYYY-MM-DD'), 1);

INSERT INTO Victim (VictimID, Name, Age, Gender, Contact, InjuryDetails, IncidentDate, AdminID)

VALUES (4, 'Johora', 15, 'Female', '0987654321', 'Hand and arm injuries', TO\_DATE('2024-01-15', 'YYYY-MM-DD'), 1);

```
INSERT INTO Victim (VictimID, Name, Age, Gender, Contact, InjuryDetails, IncidentDate, AdminID)
```

```
VALUES (5, 'Polash', 21, 'Male', '5678901234', 'Severe eye damage', TO_DATE('2024-02-05', 'YYYY-MM-DD'), 1);
```



The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands window, several INSERT statements are run to insert data into the Victim table. The statements include inserting rows for victims Salam, Fatema, Rahim, Johora, and Polash, each with their respective details like name, age, gender, contact, injury details, incident date, and admin ID. After the inserts, a select \* from Victim; statement is run to verify the data.

```
SQL Commands
127.0.0.1:8081/apex?p=4500:1003:373449047664506.NO...
ORACLE Database Express Edition
User: SYSTEM
Home > SQL > SQL Commands
Autocommit Display 15
Save Run
INSERT INTO Victim (VictimID, Name, Age, Gender, Contact, InjuryDetails, IncidentDate, AdminID)
VALUES (2, 'Fatema', 30, 'Female', '0987654321', 'Hand and arm injuries', TO_DATE('2024-01-15', 'YYYY-MM-DD'), 1);
INSERT INTO Victim (VictimID, Name, Age, Gender, Contact, InjuryDetails, IncidentDate, AdminID)
VALUES (3, 'Rahim', 20, 'Male', '5678901234', 'Severe eye damage', TO_DATE('2024-02-05', 'YYYY-MM-DD'), 1);
INSERT INTO Victim (VictimID, Name, Age, Gender, Contact, InjuryDetails, IncidentDate, AdminID)
VALUES (4, 'Johora', 15, 'Female', '0987654321', 'Hand and arm injuries', TO_DATE('2024-01-15', 'YYYY-MM-DD'), 1);
INSERT INTO Victim (VictimID, Name, Age, Gender, Contact, InjuryDetails, IncidentDate, AdminID)
VALUES (5, 'Polash', 21, 'Male', '5678901234', 'Severe eye damage', TO_DATE('2024-02-05', 'YYYY-MM-DD'), 1);
select * from Victim;

Results Explain Describe Saved SQL History
VICTIMID NAME AGE GENDER CONTACT INJURYDETAILS INCIDENTDATE ADMINID
1 Salam 25 Male 1234567890 Face burns from acid attack 10-JAN-24 1
2 Fatema 30 Female 0987654321 Hand and arm injuries 15-JAN-24 1
3 Rahim 20 Male 5678901234 Severe eye damage 05-FEB-24 1
4 Johora 15 Female 0987654321 Hand and arm injuries 15-JAN-24 1
5 Polash 21 Male 5678901234 Severe eye damage 05-FEB-24 1
5 rows returned in 0.05 seconds CSV Export
Language: en-us
9:58 AM 2/4/2025
Windows Taskbar icons: Start, Search, File Explorer, Task View, Edge, Google Chrome, File Manager, Mail, Calendar, Taskbar settings, Network, Battery, Volume, Date/Time.
```

### 3.Case Table

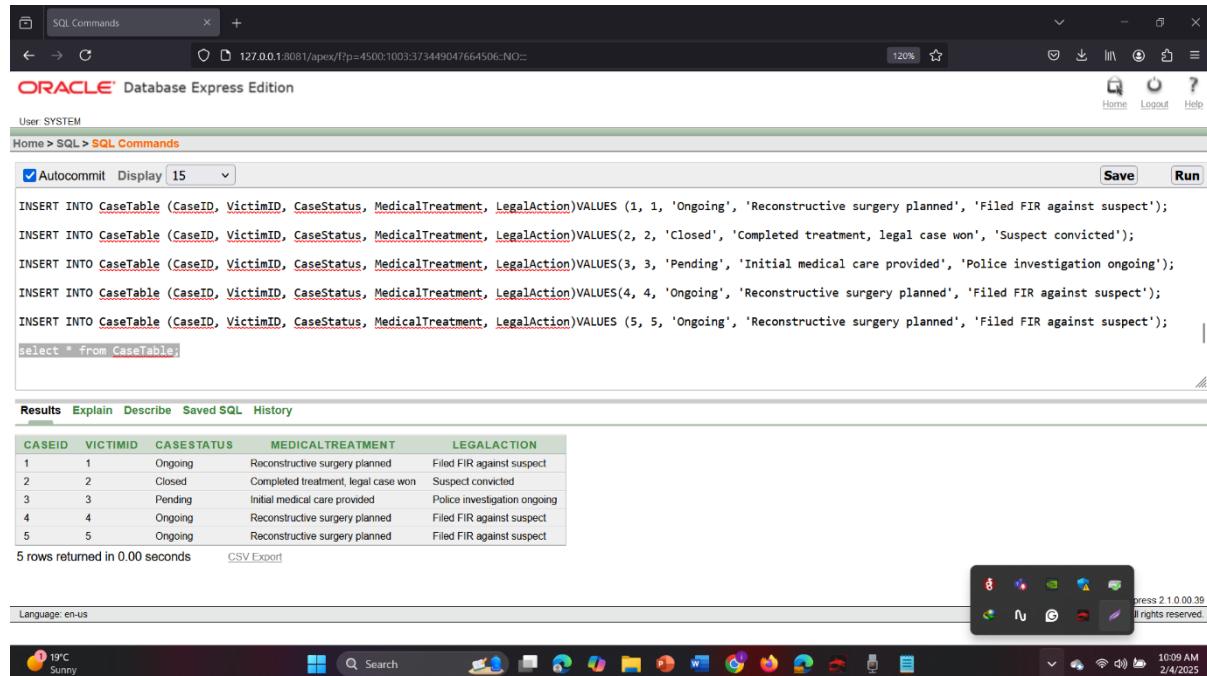
```
INSERT INTO CaseTable (CaseID, VictimID, CaseStatus, MedicalTreatment, LegalAction)VALUES (1, 1, 'Ongoing', 'Reconstructive surgery planned', 'Filed FIR against suspect');
```

```
INSERT INTO CaseTable (CaseID, VictimID, CaseStatus, MedicalTreatment, LegalAction)VALUES(2, 2, 'Closed', 'Completed treatment, legal case won', 'Suspect convicted');
```

```
INSERT INTO CaseTable (CaseID, VictimID, CaseStatus, MedicalTreatment, LegalAction)VALUES(3, 3, 'Pending', 'Initial medical care provided', 'Police investigation ongoing');
```

```
INSERT INTO CaseTable (CaseID, VictimID, CaseStatus, MedicalTreatment, LegalAction)VALUES(4, 4, 'Ongoing', 'Reconstructive surgery planned', 'Filed FIR against suspect');
```

```
INSERT INTO CaseTable (CaseID, VictimID, CaseStatus, MedicalTreatment, LegalAction)VALUES (5, 5, 'Ongoing', 'Reconstructive surgery planned', 'Filed FIR against suspect');
```



```

ORACLE Database Express Edition
User SYSTEM
Home > SQL > SQL Commands

Autocommit Display 15 Save Run
INSERT INTO CaseTable (CaseID, VictimID, CaseStatus, MedicalTreatment, LegalAction)VALUES (1, 1, 'Ongoing', 'Reconstructive surgery planned', 'Filed FIR against suspect');
INSERT INTO CaseTable (CaseID, VictimID, CaseStatus, MedicalTreatment, LegalAction)VALUES(2, 2, 'Closed', 'Completed treatment, legal case won', 'Suspect convicted');
INSERT INTO CaseTable (CaseID, VictimID, CaseStatus, MedicalTreatment, LegalAction)VALUES(3, 3, 'Pending', 'Initial medical care provided', 'Police investigation ongoing');
INSERT INTO CaseTable (CaseID, VictimID, CaseStatus, MedicalTreatment, LegalAction)VALUES(4, 4, 'Ongoing', 'Reconstructive surgery planned', 'Filed FIR against suspect');
INSERT INTO CaseTable (CaseID, VictimID, CaseStatus, MedicalTreatment, LegalAction)VALUES (5, 5, 'Ongoing', 'Reconstructive surgery planned', 'Filed FIR against suspect');

select * from CaseTable;

Results Explain Describe Saved SQL History
CASEID VICTIMID CASESTATUS MEDICALTREATMENT LEGALACTION
1 1 Ongoing Reconstructive surgery planned Filed FIR against suspect
2 2 Closed Completed treatment, legal case won Suspect convicted
3 3 Pending Initial medical care provided Police investigation ongoing
4 4 Ongoing Reconstructive surgery planned Filed FIR against suspect
5 5 Ongoing Reconstructive surgery planned Filed FIR against suspect

5 rows returned in 0.00 seconds CSV Export

Language: en-us

```

## Basic PL/SQL

### 1. Variables

Question 1: Write a PL/SQL block to store a victim's name in a variable and display it.

DECLARE

```
v_victim_name VARCHAR2(100);
```

```
BEGIN
```

```
SELECT Name INTO v_victim_name FROM Victim WHERE VictimID = 1;
```

```

DBMS_OUTPUT.LINE ('Victim Name: ' || v_victim_name);

END;

```

```

DECLARE
    v_victim_name VARCHAR2(100);
BEGIN
    SELECT Name INTO v_victim_name FROM Victim WHERE VictimID = 1;
    DBMS_OUTPUT.PUT_LINE('Victim Name: ' || v_victim_name);
END;

```

Results Explain Describe Saved SQL History

Victim Name: Salam  
Statement processed.  
0.75 seconds

Language: en-us

25°C Haze 11:11 PM 2/4/2025

Question 2: Write a PL/SQL block to count the total number of victims and display it.

Ans:

```

DECLARE
    total_victims NUMBER;
BEGIN
    SELECT COUNT (*) INTO total_victims FROM Victim;
    DBMS_OUTPUT.PUT_LINE ('Total Number of Victims: ' || total_victims);
END;

```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The code entered is:

```

DECLARE
    total_victims NUMBER;
BEGIN
    SELECT COUNT(*) INTO total_victims FROM Victim;
    DBMS_OUTPUT.PUT_LINE('Total Number of Victims: ' || total_victims);
END;

```

The results pane shows the output:

```

Total Number of Victims: 5
Statement processed.

0.50 seconds

```

The system tray at the bottom right shows the date and time as 2/4/2025, 1:18 PM.

## 2. Operators

Question 1: Write a PL/SQL block to determine what a victim's age was 5 years ago and display it.

Ans:

```

DECLARE
    v_age INT;
BEGIN
    SELECT Age INTO v_age FROM Victim
    WHERE VictimID = 1; v_age := v_age - 5;
    DBMS_OUTPUT.PUT_LINE('Victim's age 5 years ago: ' || v_age);
END;

```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The code entered is:

```
DECLARE
    v_age INT;
BEGIN
    SELECT Age INTO v_age FROM Victim WHERE VictimID = 1;
    v_age := v_age - 5;
    DBMS_OUTPUT.PUT_LINE('Victims age 5 years ago: ' || v_age);
END;
```

The results pane shows the output:

```
Victims age 5 years ago: 20
Statement processed.
```

At the bottom, it says "0.09 seconds". The system tray shows the date as 2/4/2025 and the time as 1:35 PM.

Question 2: Write a PL/SQL block to determine what a victim's age will be in 10 years and display it.

Ans:

```
DECLARE
```

```
    v_age INT;
```

```
BEGIN SELECT Age INTO v_age FROM Victim
```

```
WHERE VictimID = 1; v_age := v_age + 10; DBMS_OUTPUT.PUT_LINE('Victims age in
10 years: ' || v_age);
```

```
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The code entered is:

```
DECLARE
  v_age INT;
BEGIN
  SELECT Age INTO v_age FROM Victim WHERE VictimID = 1;
  v_age := v_age + 10;
  DBMS_OUTPUT.PUT_LINE('Victims age in 10 years: ' || v_age);
END;
```

The results pane shows the output:

```
Victims age in 10 years: 35
Statement processed.
```

The status bar at the bottom indicates "0.07 seconds".

### 3. Single-Row Functions

Question 1: Write a PL/SQL block to display a victim's name in uppercase.

Ans:

```
DECLARE
  v_name VARCHAR2(100);
BEGIN
  SELECT UPPER(Name) INTO v_name FROM Victim
  WHERE VictimID = 1;
  DBMS_OUTPUT.PUT_LINE(v_name);
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The code entered is:

```
DECLARE
  v_name VARCHAR2(100);
BEGIN
  SELECT UPPER(Name) INTO v_name FROM Victim WHERE VictimID = 1;
  DBMS_OUTPUT.PUT_LINE(v_name);
END;
```

The results section shows the output:

```
SALAM
Statement processed.
```

Execution time: 0.32 seconds

Question 1: Write a PL/SQL block to display the length of a victim's name.

Ans:

```
DECLARE
  v_length NUMBER;
BEGIN
  SELECT LENGTH(Name) INTO v_length FROM Victim
  WHERE VictimID = 1;
  DBMS_OUTPUT.PUT_LINE(v_length);
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL code entered is:

```
DECLARE
    v_length NUMBER;
BEGIN
    SELECT LENGTH(Name) INTO v_length FROM Victim WHERE VictimID = 1;
    DBMS_OUTPUT.PUT_LINE(v_length);
END;
```

The results show the output of the query:

```
5
Statement processed.
```

The system status bar at the bottom indicates the language is en-us, the date is 2/4/2025, and the time is 1:46 PM.

## 4. Group Functions

Question 1: Write a PL/SQL block to count the total number of cases.

Ans:

```
DECLARE
```

```
    v_count NUMBER;
```

```
BEGIN
```

```
    SELECT COUNT(*) INTO v_count FROM CaseTable;
    DBMS_OUTPUT.PUT_LINE(v_count);
```

```
END;
```

The screenshot shows the Oracle SQL Developer interface. The SQL Commands tab is active, displaying a PL/SQL block. The code is as follows:

```
DECLARE
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count FROM CaseTable;
    DBMS_OUTPUT.PUT_LINE(v_count);
END;
```

After running the code, the results panel shows the output: "5". Below it, the message "Statement processed." is displayed. The status bar at the bottom indicates "0.99 seconds". The system tray shows the date and time as "2/4/2025 1:51 PM".

Question 2: Write a PL/SQL block to find the average age of victims.

Ans:

```
DECLARE
    v_avg NUMBER;
BEGIN
    SELECT AVG(Age) INTO v_avg FROM Victim;
    DBMS_OUTPUT.PUT_LINE(v_avg);
END;
```

The screenshot shows the Oracle SQL Developer interface. The SQL Commands tab is active, displaying the following PL/SQL code:

```
DECLARE
    v_avg NUMBER;
BEGIN
    SELECT AVG(Age) INTO v_avg FROM Victim;
    DBMS_OUTPUT.PUT_LINE(v_avg);
END;
```

Below the code, the results are shown:

```
22.2
Statement processed.
```

Execution details:

```
0.14 seconds
```

At the bottom, the system tray shows the date and time as 2/4/2025 1:53 PM.

## 5. Loops

Question 1: Write a PL/SQL block to print the names of all victims using a FOR loop.

Ans:

```
BEGIN
FOR
v IN (SELECT Name FROM Victim) LOOP
DBMS_OUTPUT.PUT_LINE(v.Name);
END LOOP;
END;
```

The screenshot shows a Microsoft Edge browser window running on Windows 10. The address bar indicates the URL is 127.0.0.1:8081/apex/f?p=4500:1003:373449047664506..NO.. The main content area is a SQL command window titled 'SQL Commands'. It displays the following PL/SQL code:

```
BEGIN
FOR v IN (SELECT Name FROM Victim) LOOP
    DBMS_OUTPUT.PUT_LINE(v.Name);
END LOOP;
END;
```

Below the code, the results of the execution are shown:

```
Salam
Fatema
Rahim
Johora
Polash
```

Message: Statement processed.

Execution time: 0.45 seconds

The system tray at the bottom shows the date and time as 2/4/2025, 2:08 PM. The taskbar also displays various application icons.

## 6. Conditional Statement

Question 1: Write a PL/SQL block to check if a victim's age is greater than 30.

Ans:

```
DECLARE
    v_age INT;
BEGIN
    SELECT Age INTO v_age FROM Victim WHERE VictimID = 1;
    IF v_age > 30
        THEN
            DBMS_OUTPUT.PUT_LINE('Age is greater than 30');
        ELSE
            DBMS_OUTPUT.PUT_LINE('Age is 30 or less');
    END IF;
END;
```

The screenshot shows the Oracle SQL Developer interface. The title bar indicates the connection is to 'TASFI ISLAM (You) | Micros...' at '127.0.0.1:8081/apex/F?p=4500:1003:373449047664506:NO::'. The user is SYSTEM. The main area displays a PL/SQL block:

```
DECLARE
    v_age INT;
BEGIN
    SELECT Age INTO v_age FROM Victim WHERE VictimID = 1;
    IF v_age > 30 THEN
        DBMS_OUTPUT.PUT_LINE('Age is greater than 30');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Age is 30 or less');
    END IF;
END;
```

The results show the output: "Age is 30 or less". Below the results, it says "Statement processed." and "0.21 seconds". The bottom status bar shows the language is "en-us" and the system date and time.

Question 2: Write a PL/SQL block to check if a victim is Male or Female and display a message using a conditional statement.

Ans:

```
DECLARE
    v_gender CHAR(10);
    v_message VARCHAR2(100);
BEGIN
    SELECT Gender INTO v_gender FROM Victim WHERE VictimID = 1;
    IF v_gender = 'Male' THEN
        v_message := 'The victim is a Male.';
    ELSIF v_gender = 'Female' THEN
        v_message := 'The victim is a Female.';
    ELSE
        v_message := 'Gender not specified.';
    END IF;
    DBMS_OUTPUT.PUT_LINE(v_message);
```

END;

The screenshot shows the Oracle Database Express Edition interface. In the main window, a SQL command is being run:

```
DECLARE
    v_gender CHAR(10);
    v_message VARCHAR2(100);
BEGIN
    SELECT Gender INTO v_gender FROM Victim WHERE VictimID = 1;
    IF v_gender = 'Male' THEN
        v_message := 'The victim is a Male.';
    ELSIF v_gender = 'Female' THEN
        v_message := 'The victim is a Female.';
    ELSE
        v_message := 'Gender not specified.';
    END IF;
    DBMS_OUTPUT.PUT_LINE(v_message);
END;
```

The output shows the message "The victim is a Male." followed by "Statement processed." and a timestamp "1.09 seconds".

At the bottom, the system tray displays the Windows taskbar with icons for various applications like File Explorer, Edge, and Control Panel. The date and time are also visible.

## 7. Subqueries

Quistion 1: Write a PL/SQL block to display the name of a victim with an "Ongoing" case.

Ans:

DECLARE

```
v_victim VARCHAR2(100);
```

BEGIN

```
SELECT Name INTO v_victim FROM Victim
```

```
WHERE VictimID = (SELECT VictimID FROM CaseTable WHERE CaseStatus =  
'Ongoing' AND ROWNUM = 1);
```

```
DBMS_OUTPUT.PUT_LINE(v_victim);
```

END;

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The code entered is:

```
DECLARE  
v_victim VARCHAR2(100);  
BEGIN  
SELECT Name INTO v_victim FROM Victim  
WHERE VictimID = (SELECT VictimID FROM CaseTable WHERE CaseStatus = 'Ongoing' AND ROWNUM = 1);  
DBMS_OUTPUT.PUT_LINE(v_victim);  
END;
```

The results pane shows the output:

```
Salam  
Statement processed.  
1.28 seconds
```

The status bar at the bottom right indicates the system is running Oracle Express 2.1.0.0.39, the date is 2/2/2025, and the time is 2:23 PM.

Quistion 2: Write a PL/SQL block to count the number of cases managed by Admin 1.

Ans:

DECLARE

```
v_count NUMBER;
```

BEGIN

```

SELECT COUNT(*) INTO v_count FROM CaseTable
WHERE VictimID IN (SELECT VictimID FROM Victim WHERE AdminID = 1);
DBMS_OUTPUT.PUT_LINE(v_count);

END;

```

```

DECLARE
  v_count NUMBER;
BEGIN
  SELECT COUNT(*) INTO v_count FROM CaseTable
  WHERE VictimID IN (SELECT VictimID FROM Victim WHERE AdminID = 1);
  DBMS_OUTPUT.PUT_LINE(v_count);
END;

```

Results Explain Describe Saved SQL History

5  
Statement processed.  
0.36 seconds

Language: en-us      Express 2.1.0.039  
All rights reserved.

27°C Haze      Search      2/4/2025

## 8.Join

Quistion 1:Write a PL/SQL block to display a victim's name along with their case status.

Ans:

```

DECLARE
  v_victim VARCHAR2(100);
  v_status CHAR(15);

```

```

BEGIN

SELECT v.Name, c.CaseStatus INTO v_victim, v_status

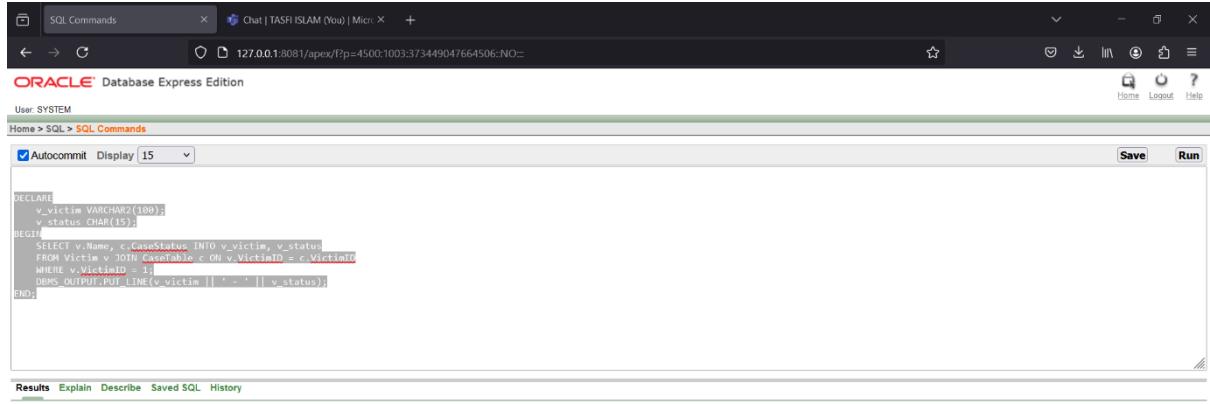
FROM Victim v JOIN CaseTable c ON v.VictimID = c.VictimID

WHERE v.VictimID = 1;

DBMS_OUTPUT.PUT_LINE(v_victim || ' - ' || v_status);

END;

```



```

DECLARE
    v_victim VARCHAR2(100);
    v_status CHAR(15);
BEGIN
    SELECT v.Name, c.CaseStatus INTO v_victim, v_status
    FROM Victim v JOIN CaseTable c ON v.VictimID = c.VictimID
    WHERE v.VictimID = 1;
    DBMS_OUTPUT.PUT_LINE(v_victim || ' - ' || v_status);
END;

```

Results Explain Describe Saved SQL History

Salam - Ongoing  
Statement processed.  
0.78 seconds



Question 2: Write a PL/SQL block to display the admin managing a specific victim.

Ans:

```

DECLARE
    v_admin VARCHAR2(50);

BEGIN
    SELECT a.Username INTO v_admin
    FROM Admin a JOIN Victim v ON a.AdminID = v.AdminID
    WHERE v.VictimID = 3;

    DBMS_OUTPUT.PUT_LINE(v_admin);

END;

```

```

DECLARE
  v_admin VARCHAR2(50);
BEGIN
  SELECT a.Username INTO v_admin
  FROM Admin a JOIN Victim v ON a.AdminID = v.AdminID
  WHERE v.VictimID = 3;
  DBMS_OUTPUT.PUT_LINE(v_admin);
END;

```

Tasfi  
Statement processed.  
1.09 seconds

Language: en-us

Windows taskbar showing various application icons and system status.

## Advance PL/SQL

### 1. Stored Function

**Question 1:** Write a stored function to return the total number of victims.

**Ans:**

```

CREATE OR REPLACE FUNCTION get_total_victims
RETURN NUMBER IS
v_count NUMBER;
BEGIN
SELECT COUNT(*) INTO v_count
FROM Victim;
RETURN v_count;

```

```

END;

DECLARE
    v_total NUMBER;
BEGIN
    v_total := get_total_victims;
    DBMS_OUTPUT.PUT_LINE('Total Number of Victims: ' || v_total);
END;

```

```

CREATE OR REPLACE FUNCTION get_total_victims RETURN NUMBER IS
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count FROM Victim;
    RETURN v_count;
END;

DECLARE
    v_total NUMBER;
BEGIN
    v_total := get_total_victims;
    DBMS_OUTPUT.PUT_LINE('Total Number of Victims: ' || v_total);
END;

```

Total Number of Victims: 5  
Statement processed.  
0.49 seconds

## 2. Stored Procedure

**Question 1:** Write a stored procedure to update a victim's contact number.

Ans:

```

CREATE OR REPLACE PROCEDURE update_victim_contact(p_victim_id INT,
    p_new_contact VARCHAR2)

```

AS

BEGIN

```
UPDATE Victim SET Contact = p_new_contact WHERE VictimID = p_victim_id;  
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The code entered is:

```
CREATE OR REPLACE PROCEDURE update_victim_contact(p_victim_id INT, p_new_contact VARCHAR2)
AS
BEGIN
  UPDATE Victim SET Contact = p_new_contact WHERE VictimID = p_victim_id;
END;
```

The results pane shows the message "Procedure created." and a execution time of "0.31 seconds". The system tray at the bottom right shows the date and time as "2/4/2025 7:36 PM".

### 3. Table-Based Record

**Question 1: Fetch a single victim record and display details using a record.**

**Ans:**

```
DECLARE
```

```
v_victim Victim%ROWTYPE;
```

```

BEGIN
SELECT * INTO v_victim FROM Victim
WHERE VictimID = 1;
DBMS_OUTPUT.PUT_LINE(v_victim.Name || ' - ' || v_victim.Age || ' - ' ||
v_victim.Gender);
END;

```

The screenshot shows the Oracle Database Express Edition interface. In the SQL Commands tab, a PL/SQL block is run. The code declares a cursor, selects a victim from the Victim table where VictimID = 1, and prints the Name, Age, and Gender using DBMS\_OUTPUT.PUT\_LINE. The results show a single row: Salam - 25 - Male. The interface includes tabs for Results, Explain, Describe, Saved SQL, and History.

```

DECLARE
  v_victim Victim%ROWTYPE;
BEGIN
  SELECT * INTO v_victim FROM Victim WHERE VictimID = 1;
  DBMS_OUTPUT.PUT_LINE(v_victim.Name || ' - ' || v_victim.Age || ' - ' || v_victim.Gender);
END;

```

Results

Salam - 25 - Male  
Statement processed.  
0.13 seconds

## 4. Explicit Cursor

Question 1 : Use an explicit cursor to print all case details.

Ans:

```

DECLARE
CURSOR
cur_cases IS SELECT CaseID, CaseStatus FROM CaseTable;
v_case_id INT;
v_status CHAR(15);
BEGIN

```

```

OPEN cur_cases;

LOOP FETCH cur_cases INTO v_case_id, v_status;

EXIT WHEN cur_cases%NOTFOUND;

DBMS_OUTPUT.PUT_LINE('Case ID: ' || v_case_id || ', Status: ' || v_status);

END LOOP;

CLOSE cur_cases;

END;

```

The screenshot shows the Oracle Database Express Edition interface. The SQL Commands window displays the following PL/SQL block:

```

DECLARE
    CURSOR cur_cases IS SELECT CaseID, CaseStatus FROM CaserTable;
    v_case_id INT;
    v_status CHAR(19);
BEGIN
    OPEN cur_cases;
    LOOP
        FETCH cur_cases INTO v_case_id, v_status;
        EXIT WHEN cur_cases%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Case ID: ' || v_case_id || ', Status: ' || v_status);
    END LOOP;
    CLOSE cur_cases;
END;

```

The results pane shows the output of the block:

```

Case ID: 1, Status: Ongoing
Case ID: 2, Status: Closed
Case ID: 3, Status: Pending
Case ID: 4, Status: Ongoing
Case ID: 5, Status: Ongoing
Statement processed.

```

The system status bar at the bottom right indicates "press 2 1 0 0 39" and "all rights reserved".

## 5. Cursor-Based Record

**Question 1:** Use a cursor-based record to print all victim details.

Ans:

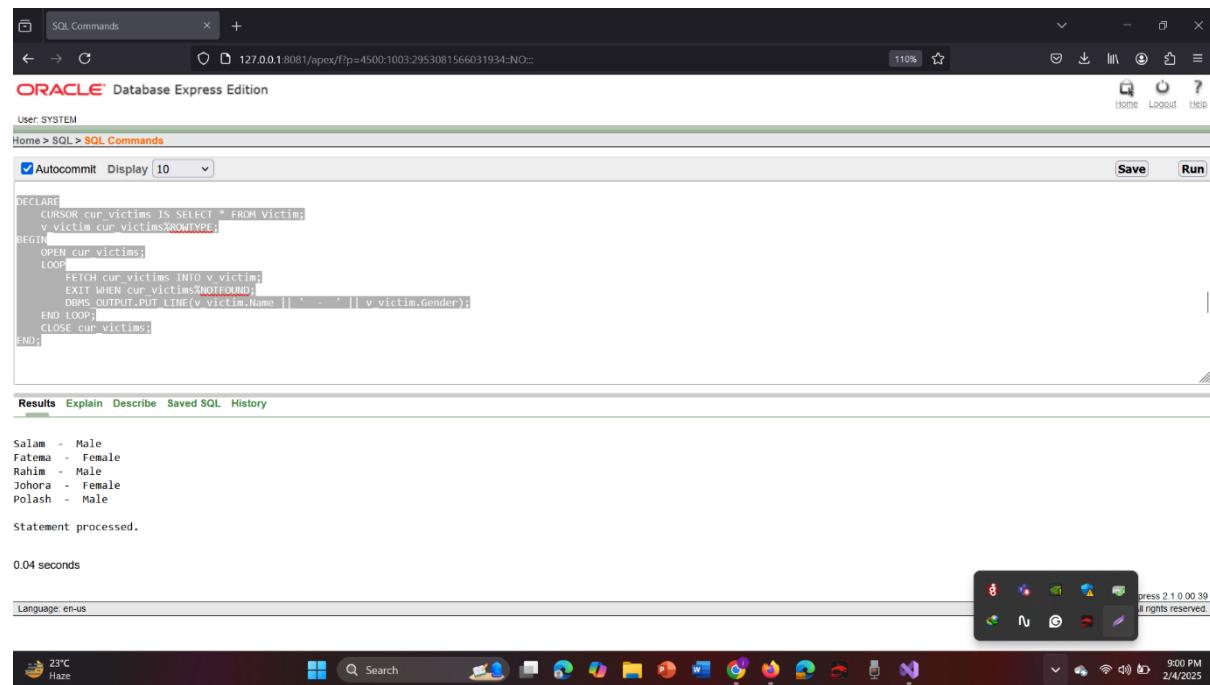
```

DECLARE
    CURSOR cur_victims IS SELECT * FROM Victim;
    v_victim cur_victims%ROWTYPE;
BEGIN
    OPEN cur_victims;

```

LOOP

```
  FETCH cur_victims INTO v_victim;  
  
  EXIT WHEN cur_victims%NOTFOUND;  
  
  DBMS_OUTPUT.PUT_LINE(v_victim.Name || ' - ' || v_victim.Gender);  
  
END LOOP;  
  
CLOSE cur_victims;  
  
END;
```



The screenshot shows the Oracle Database Express Edition SQL Commands interface. The code entered is:

```
DECLARE  
  CURSOR cur_victims IS SELECT * FROM victim;  
  v_victim victim%ROWTYPE;  
BEGIN  
  OPEN cur_victims;  
  LOOP  
    FETCH cur_victims INTO v_victim;  
    EXIT WHEN cur_victims%NOTFOUND;  
    DBMS_OUTPUT.PUT_LINE(v_victim.Name || ' - ' || v_victim.Gender);  
  END LOOP;  
  CLOSE cur_victims;  
END;
```

The results pane displays the output:

```
Salam - Male  
Fatema - Female  
Rahim - Male  
Johora - Female  
Polash - Male  
  
Statement processed.
```

At the bottom, the system tray shows the date and time as 9:00 PM on 2/4/2025.

## 6. Row-Level Trigger

**Question 1:** Create a trigger that logs updates on the Victim table.

**Ans:**

```
CREATE OR REPLACE TRIGGER trg_victim_update
AFTER UPDATE ON Victim
FOR EACH ROW
BEGIN
DBMS_OUTPUT.PUT_LINE('Victim ID ' || :OLD.VictimID || ' was updated.');
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The URL is 127.0.0.1:8081/apex/?p=4500:1003:2953081566031934:NO:. The user is SYSTEM. The code entered is:

```
CREATE OR REPLACE TRIGGER trg_victim_update
AFTER UPDATE ON Victim
FOR EACH ROW
BEGIN
  DBMS_OUTPUT.PUT_LINE('Victim ID ' || :OLD.VictimID || ' was updated.');
END;
```

The results show "Trigger created." and a duration of "3.97 seconds". The system tray shows various icons and the date/time as 2/4/2025 9:10 PM.

## 7. Statement-Level Trigger

**Question 1:** Create a trigger that runs after an insert into the Victim table.

Ans:

```
CREATE OR REPLACE TRIGGER trg_victim_insert
```

```
AFTER INSERT ON Victim
```

```
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE('A new victim record was added.');
```

```
END;
```

The screenshot shows the Oracle Database Express Edition SQL Commands interface. The SQL command entered is:

```
CREATE OR REPLACE TRIGGER trg_victim_insert
AFTER INSERT ON Victim
BEGIN
  DBMS_OUTPUT.PUT_LINE('A new victim record was added.');
END;
```

The results show the trigger was created successfully in 0.52 seconds.

Trigger created.  
0.52 seconds

Language: en-us

Version Express 21.0.0.0.39  
Oracle. All rights reserved.

## 8. Package

**Question 1:** Create a package to get victim details.

**Ans:**

```
CREATE OR REPLACE PACKAGE Victim_Package
```

```
AS FUNCTION get_victim_count
```

```
RETURN NUMBER;
```

```
END Victim_Package;
```

```
CREATE OR REPLACE PACKAGE BODY Victim_Package
```

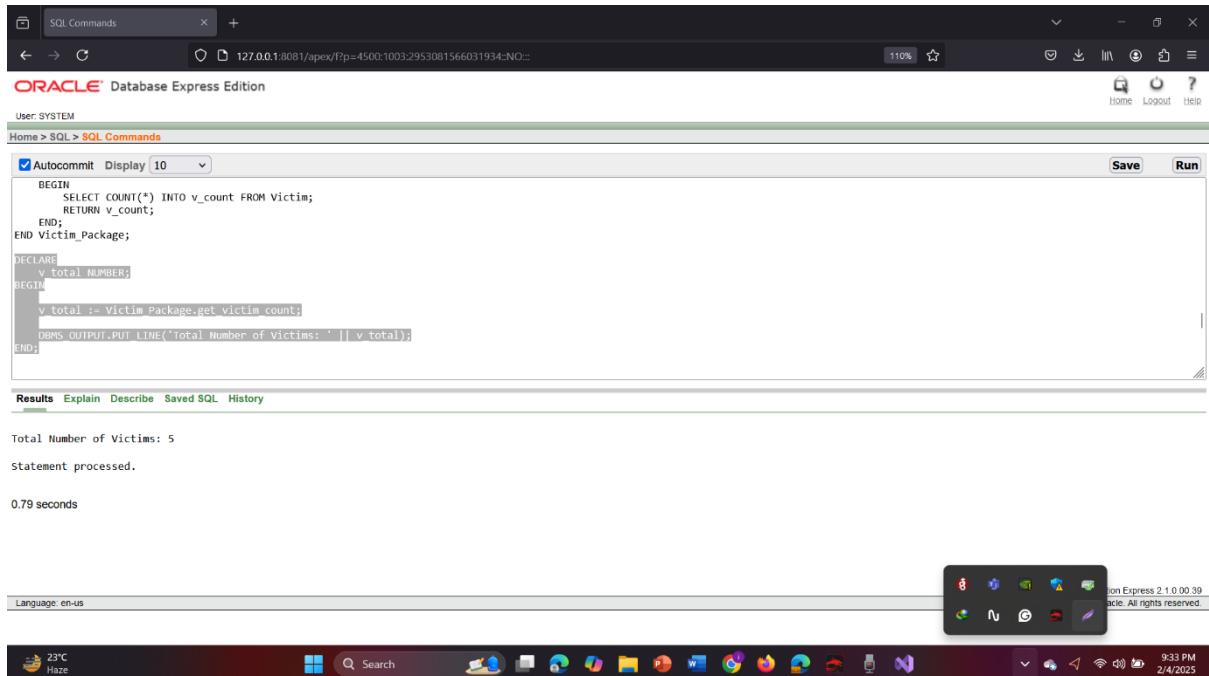
```
AS FUNCTION get_victim_count
```

```
RETURN NUMBER IS v_count NUMBER;
```

```
BEGIN
```

```
SELECT COUNT(*) INTO v_count FROM Victim;
```

```
RETURN v_count;  
END;  
  
END Victim_Package;
```



The screenshot shows the Oracle Database Express Edition SQL Commands interface. The code entered is:

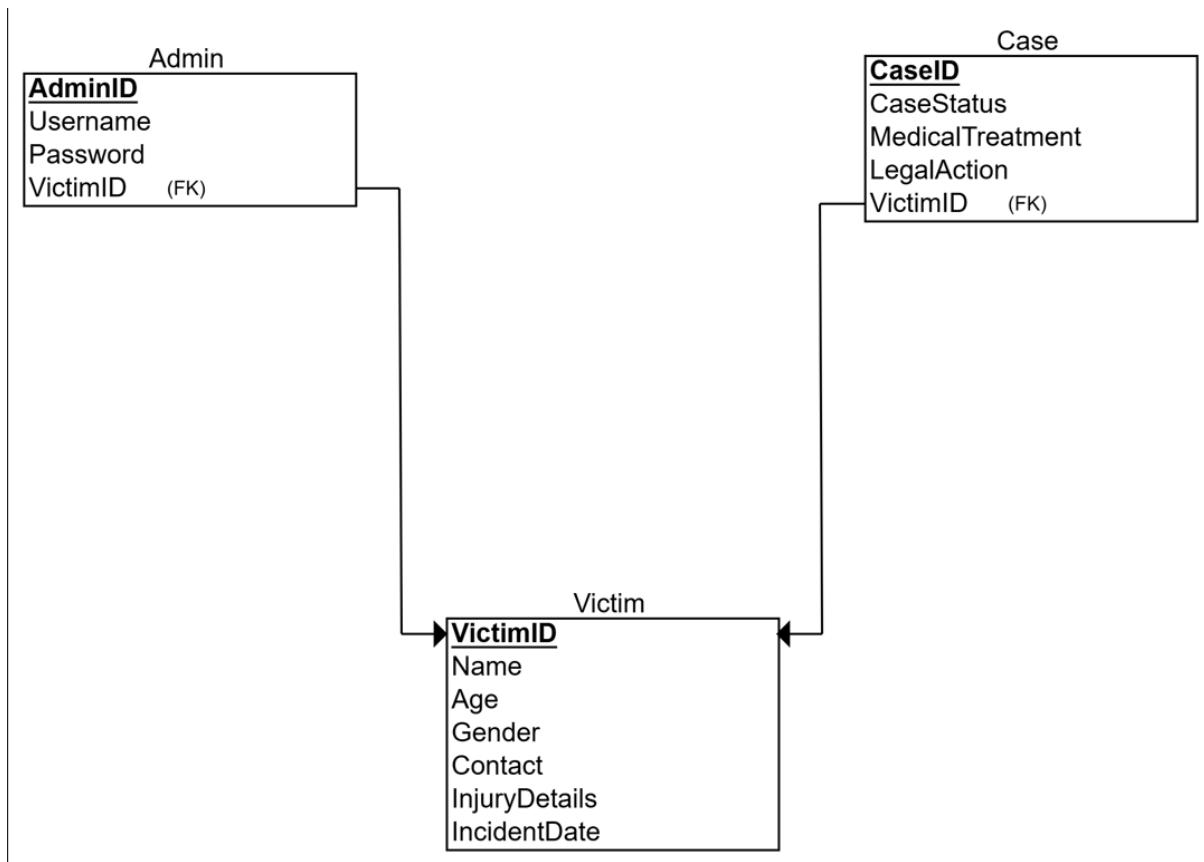
```
BEGIN  
    SELECT COUNT(*) INTO v_count FROM Victim;  
    RETURN v_count;  
END;  
END Victim_Package;  
  
DECLARE  
    v_total NUMBER;  
BEGIN  
    v_total := Victim_Package.get_victim_count;  
    DBMS_OUTPUT.PUT_LINE('Total Number of Victims: ' || v_total);  
END;
```

The results section shows the output:

```
Total Number of Victims: 5  
Statement processed.  
0.79 seconds
```

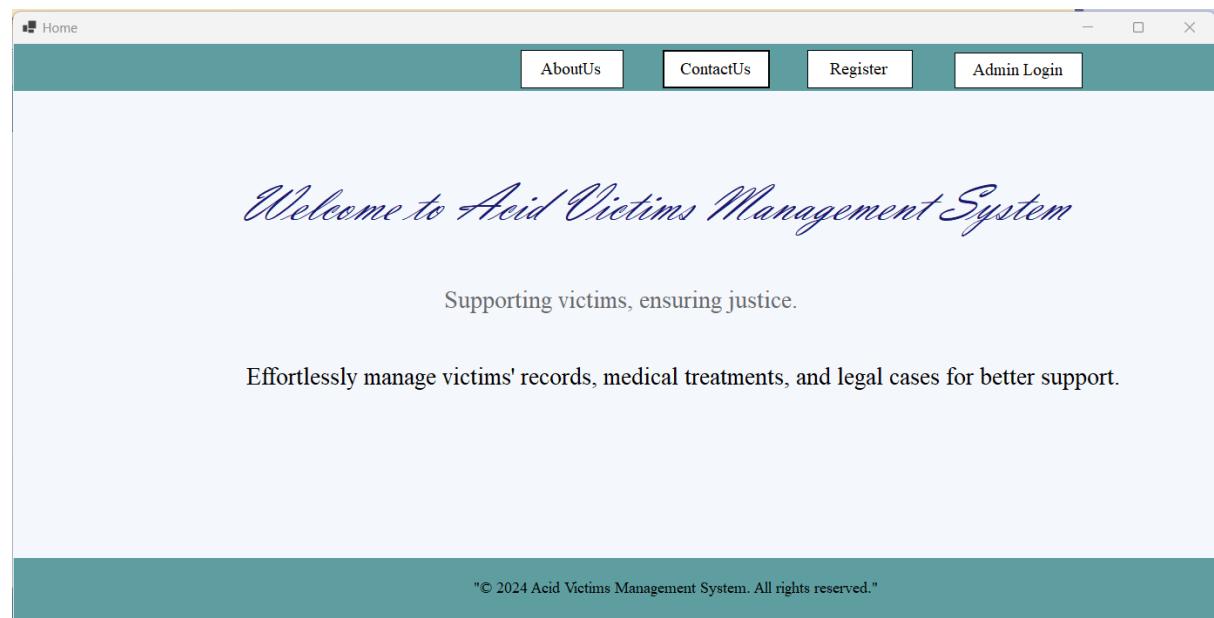
The system tray at the bottom right shows the Oracle Database Express Edition icon, the date (2/4/2025), and the time (9:33 PM).

Schema:

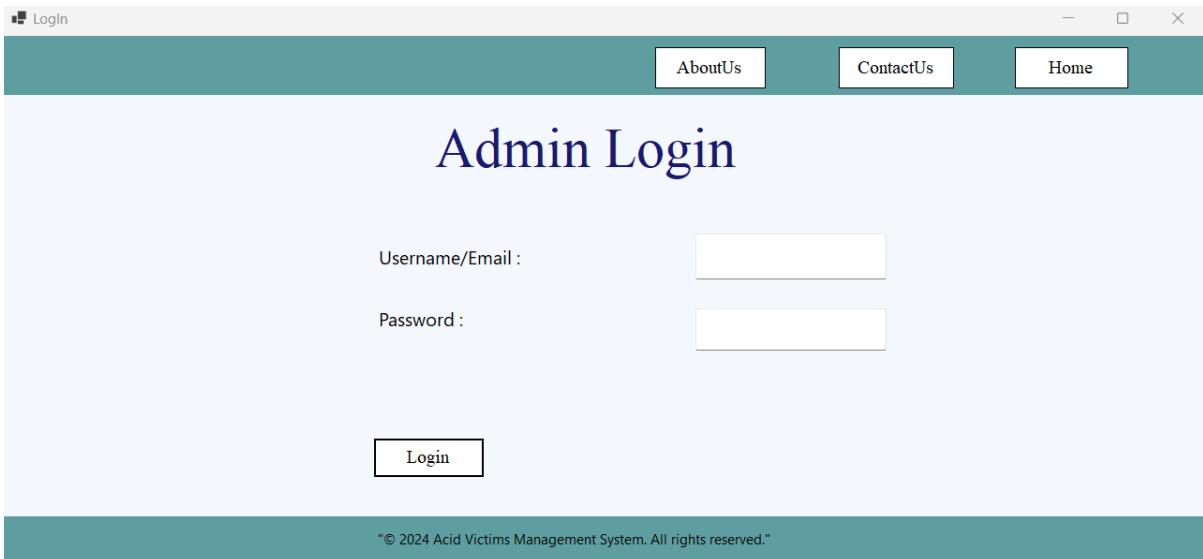


# User Interface

## 1. Home Design



## 2. Login Design



## Relational Algebra :

1. How do you select only those victims with an ongoing case?

Ans:  $\sigma \text{CaseStatus} = \text{'Ongoing'}$  (CaseTable)

2. How do you get only the Victim's IDs and Names?

Ans:  $\pi \text{VictimID, Name}$  (Victim)

3. How do you combine Victims and Cases?

Ans:  $\text{Victim} \bowtie \text{CaseTable}$

4. How do you find victims assigned to a case but not linked to an admin?

Ans:

$\pi \text{ VictimID} (\text{Victim} \bowtie \text{CaseTable}) - \pi \text{ VictimID} (\text{Victim} \bowtie \text{Admin})$

5. How do you find victims who have an active case?

Ans:  $\text{Victim} \bowtie \text{CaseTable}$